



Norges miljø- og
biovitenskapelige
universitet

Masteroppgave 2020 30 stp
Fakultet for realfag og teknologi

Modellering av energibruk i store bygningmasser ved bruk av kunstig intelligens

Modelling of energy consumption in large building
stocks with artificial intelligence

Murat Bastas
Sondre Johannes Bergene
Byggeteknikk og arkitektur – teknologi (Sivilingeniør)



FORORD

Denne masteroppgaven utgjør siste del av studiet som har vart i 2 år. Begge to kom med en bachelorgrad fra Høgskolen i Østfold, som vi også skrev sammen. Vi var derfor allerede sikret et godt samarbeid. Alt gikk etter planen frem til midten av mars, da vi fikk vite at skolene måtte stenges pga. koronaviruset. Vi har hatt hjemmekontor i ca. 2,5 måneder av masterperioden. Hver gang vi tok en tur for å diskutere oppgaven, måtte vi samtidig være ekstra forsiktig med å ha 1-meters avstand. Vi måtte omstille oss, og det var spesielt vanskelig i starten. Det kunne observeres mange mennesker med masker og munnbind, og folk hamstret inn dopapir. Det var tydelig at folk var bekymret og den vanlige hverdagen var snudd opp ned. Men selv i disse tider prøvde vi å holde hodet kaldt, fulgte rådene fra helsemyndighetene og hadde fullt fokus på å fullføre masteroppgaven. Nå som oppgaven er ferdig, er det mange dyktige folk som fortjener takk.

Først og fremst vil vi takke vår eksterne veileder Torfinn Lysfjord (Seksjonssjef i ÅF Advansia), som ga oss muligheten og stilte opp med det som var nødvendig for å gjennomføre denne oppgaven. Deretter vil vi gjerne takke våre interne veiledere Thomas Kringlebotn Thiis (Professor), som har hjulpet oss med å komme fram til ideen bak masteroppgaven. Gode og konstruktive tilbakemeldinger har hjulpet med å holde oss på rett spor. Deretter vil vi gjerne takke Hans Ekkehard Plesser (Professor), som har bidratt veldig mye på den tekniske siden. Vi setter ekstra stor pris på det han bidro med angående maskinlæringen som er en gren innen kunstig intelligens. Dette har generelt sett skapt stor interesse for alle som var involvert i oppgaven.

Ås, 02.06.2020

Murat Bastas

Sondre Johannes Bergene



SAMMENDRAG

I Norge var energiforbruket til husholdninger så mye som 30% av det totale nettoforbruket av elektrisk kraft i 2018. Dette henger sammen med at en stor andel av den totale bygningsmassen består av eldre bygninger med høyt energiforbruk. Denne oppgaven omhandler analyse av energiforbruket av boligmassen til Boligbygg KF og utvikling av en prediksjonsmodell for energiforbruket ved hjelp av kunstig intelligens. Boligbygg er et kommunalt foretak i Oslo og samtidig Norges største utleier av boliger. Sammen med ÅF Advansia, som er et prosjekt- og byggeledelsesfirma, har de inngått en kontrakt om energibesparelse for definerte bygningsmasser. Formålet med analysen er å få en oversikt over hvilken tilstand disse bygningsmassene er i med tanke på energiforbruk. I tillegg er formålet med utvikling av en prediksjonsmodell å kunne stille opp med et godt underlag ved innkjøp av strøm med kort tidshorison. Analysen baserer seg på data fra et utvalg av bygningsmassen til Boligbygg.

I forbindelse med analyse av energiforbruket er de tradisjonelle energi-temperatur kurvene eller såkalt ET-kurver blitt tatt i bruk. Slike kurver viser forholdet mellom energiforbruket og utetemperaturen. I denne oppgaven brukes ET-kurver til å estimere blant annet oppvarmings- og varmtvannsbehov. Videre benyttes maskinlæring til utvikling av en prediksjonsmodell for energiforbruket. Maskinlæring er en type dataanalyse, og er den delen innen kunstig intelligens som omhandler bruk av statistiske metoder til å finne mønstre i store datamengder. I denne oppgaven testes 6 maskinlæringsalgoritmer. Tankegangen er å dele dataene inn i trenings-data og test-data. Modellen bruker trenings-data til å gjenkjenne sammenhengene i dataene, og kan basert på disse, predikere nye verdier. De predikerte verdiene sammenlignes med test-dataene for å kontrollere i hvilken grad modellen har «lært».

Som et resultat av databehandling er det utelatt totalt 44 av 99 bygninger. En av årsakene til dette er for å sikre et felles sammenligningsgrunnlag. I analysen deles resterende bygningene inn etter om oppvarmingsbehovet dekkes av elektrisitet eller fjernvarme. For de førstnevnte bygningene estimeres oppvarmingsbehovet til å være gjennomsnittlig 56 kWh/m² i året, mens for de resterende er tilsvarende estimert forbruk 98 kWh/m² i året. Dette kan skyldes ulik systemvirkningsgrad.

For å få en indikasjon på hvor presis prediksjonsmodellen er, sammenlignes den med predikerte verdier fra ET-kurven. Det kan konkluderes at avviket som oppstår ved prediksjon av døgnverdier blir redusert med ca. 85% ved å ta i bruk kunstig intelligens. Dette viser hvordan kunstig intelligens klarer å benytte informasjonen i dataene på en mer effektiv måte enn den forenklede ET-kurve metoden. I det tilfellet hvor en byggeier skal kjøpe strøm på kraftmarkedet for påfølgende dag, vil det reduserte avviket bidra til reduserte innkjøpskostnader.



ABSTRACT

In Norway, the energy consumption for households was 30% of the total net consumption of electric power in 2018. This is because a large proportion of the total building stock consists of older buildings with high energy consumption. This thesis deals with the analysis of the energy consumption of the housing stock for Boligbygg KS and the development of a prediction model for the energy consumption using artificial intelligence. Boligbygg is a municipal enterprise in Oslo and at the same time Norway's largest landlord of housing. Together with ÅF Advansia, which is a project and construction management company, they have signed an energy saving contract for defined buildings. The purpose of the analysis is to get an overview of the state of these buildings in terms of energy consumption. In addition, the purpose of developing a prediction model is to be able to set up a good foundation when purchasing power with a short time horizon. The analysis is based on data from a selection of Boligbygg's building stock.

In connection with the analysis of energy consumption, the traditional energy-temperature curves or so-called energy signature have been used. Such curves show the relationship between energy consumption and outdoor temperature. In this exercise, energy signature is used to estimate, heating and hot water. Furthermore, machine learning is used to develop a prediction model for energy consumption. Machine learning is a type of data analysis and is the part of artificial intelligence that deals with the use of statistical methods to find patterns in large amounts of data. In this assignment, 6 machine learning algorithms are tested. The idea is to divide the data into training data and test data. The model uses training data to recognize the relationships in the data and, based on these, can predict new values. The predicted values are compared with the test data to check the degree to which the model has "learned".

As a result of data processing, a total of 44 out of 99 buildings have been left out. One of the reasons for this is to ensure a common basis for comparison. In the analysis, the remaining buildings are divided according to whether the heating needs are met by electricity or district heating. For the former buildings, the heating demand is estimated to be 56 kWh / m² per year in average, while for the remaining, the corresponding estimated consumption is 98 kWh / m² per year. This may be due to different system efficiency.

In order to get an indication of how accurate the prediction model is, it is compared with predicted values from the energy signature. It can be concluded that the deviation that occurs when predicting daily values is reduced by approx. 85% by using artificial intelligence. This shows how artificial intelligence manages to use the information in the data more efficiently than the simplified energy signature method. In the case where a building owner is to purchase electricity in the power market for the following day, the reduced deviation will contribute to reduced purchasing costs.



FIGUR- OG TABELLISTE

Figur 1 Presentasjon av SVR-modellen. Kilde: (Carrsco, 2020).....	6
Figur 2 Illustrasjon av korrelasjonskoeffisient	7
Figur 3 Prediksjonsresultater for campusbygningen i Harvard. Kilde: (Yan, Wette, & Xie, 2014).....	8
Figur 4 Kyotopyramiden som viser de energibesparende tiltakene.....	10
Figur 5 Utklipp av tabeller som viser dagens krav til energieffektivitet i bygg. A) viser energiramme, mens b) er oversikt over energiltak som må oppfylles for å oppnå kravet om energieffektivitet. Kilde: (Lovdata, 2020).	11
Figur 6 Utklipp av NS3031, tabell A.1. Energi- og effektbehov for henholdsvis belysning, utstyr og varmtvann. Kilde; (NS3031, 2020)	12
Figur 7 Oversikt på varmebehovet i bygninger - definisjon iht. TEK17.....	16
Figur 8 Ulike ET-kurve modeller. A) er tradisjonell, B) er tilfelle med lite oppvarmingskapasitet og C) er tilfelle med energiforbruk til kjøling.	18
Figur 9 Organisering av kraftmarkedet i Norge. Kilde: (Kraftmarkedet, 2020).....	20
Figur 10 Alle deler som strømgregningen består av.....	20
Figur 11 Prinsippskisse for EPC-modell. Kilde: (Husbanken, 2020).....	22
Figur 12 Årlig energiforbruk pr. kvm. for alle bygg (99 bygg)	25
Figur 13 Bygninger som dekker varmebehovet med elektrisitet, hvor a) viser hvordan forbruket varierer gjennom året, mens b) viser forbruket som en funksjon av utetemperatur.....	26
Figur 14 Bygninger som dekker varmebehovet med fjernvarme, hvor a) viser hvordan forbruket varierer gjennom året, mens b) viser forbruket som en funksjon av utetemperatur.....	27
Figur 15 Illustrasjon av hvordan estimeringene er foretatt.....	29
Figur 16 Alle målerne for bygg som bruker elektrisitet til å dekke varmebehovet	30
Figur 17 Alle målerne for bygg som bruker fjernvarme til å dekke varmebehovet.....	31
Figur 18 Genererer variablene i «DataFrame» for times- og døgnverdier	35
Figur 19 Fordeler dataene i trening- og testsett.....	36
Figur 20 Trener modellen og predikerer.....	36
Figur 21 Databehandlingsresultater, hvor a) viser antall bygg med ulike energiforsyningssystemer, b) viser de utelatte bygningene av ulike årsaker.	39
Figur 22 Totalt årlig energiforbruk pr kvm. for alle bygninger (55 bygninger)	40
Figur 23 Fordeling av a) oppvarmingstid, og b) grunnlast for de ulike bygningene (55 bygninger).....	41
Figur 24 Fordeling av a) balansetemperaturen og b) stigningstallet for de ulike bygningene (55 bygninger)	42
Figur 25 Årlig energiforbruk pr. kvm for bygninger som bruker elektrisitet til oppvarming (37 bygninger)	43
Figur 26 Fordeling av energiforbruk som skal dekke a) oppvarmingsbehovet og b) varmtvannsbehovet for bygninger som bruker elektrisitet til oppvarming (37 bygninger). Estimerte verdier	44
Figur 27 Fordeling av a) balansetemperatur og b) stigningstallet for bygninger som bruker elektrisitet til oppvarming (37 bygninger). Estimerte verdier.....	45
Figur 28 Årlig energiforbruk pr.kvm for bygninger som bruker fjernvarme til oppvarming (18 bygninger)	46
Figur 29 Fordeling av energiforbruk som skal dekke a) oppvarmingsbehovet og b) varmtvannsbehovet for bygninger som bruker fjernvarme til oppvarming (18 bygninger). Estimerte verdier.....	47



Figur 30 Fordeling av a) balansetemperatur og b) stigningstallet for bygninger som bruker fjernvarme til oppvarming (18 bygninger). Estimerte verdier.....	48
Figur 31 Faktisk energiforbruk av oppvarming, varmtvann og annet for bygninger med fjernvarme og bygninger som bruker elektrisitet til oppvarming, sammenlignet med TEK-17. Alle er estimerte verdier	49
Figur 32 Prosent av energiforbruk til oppvarming, varmtvann og annet for bygninger med fjernvarme og bygninger som bruker elektrisitet til oppvarming, sammenlignet med TEK-17. Alle er estimerte verdier.....	49
Figur 33 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) utetemperatur, b) sin av året og c) sin av dagen, basert på timesverdier, for de ulike bygningene (55 bygninger).....	50
Figur 34 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) cos av året, b) cos av dagen og c) helg, basert på timesverdier, for de ulike bygningene (55 bygninger).....	51
Figur 35 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) utetemperatur og b) sin av året, basert på døgnverdier, for de ulike bygningene (55 bygninger)	52
Figur 36 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) cos av året og b) helg, basert på døgnverdier, for de ulike bygningene.....	53
Figur 37 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) utetemperatur, b) cos av dagen og c) sin av dagen, basert på døgnvariasjon, for de ulike bygningene (55 bygninger).....	54
Figur 38 Korrelasjonen mellom energiforbruk og utetemperatur, og cos av året for en tilfeldig valgt bygning. Basert på timesverdier	55
Figur 39 Korrelasjonen mellom energiforbruk og utetemperatur, og cos av året for en tilfeldig valgt bygning. Basert på døgnverdier.....	55
Figur 40 Korrelasjonen mellom energiforbruk og utetemperatur, og sin av dagen for en tilfeldig valgt bygning. Basert på døgnvariasjon.....	56
Figur 41 Prediksjonsresultater med tradisjonell ET-kurve, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE.....	57
Figur 42 Prediksjonsresultater med multipl lineær regresjon-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE.....	58
Figur 43 Prediksjonsresultater med «Kernel Ridge»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE	59
Figur 44 Prediksjonsresultater med «Decision Tree»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE	60
Figur 45 Prediksjonsresultater med «Random Forest»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE	61
Figur 46 Prediksjonsresultater med «Gradient Boost»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE	62
Figur 47 Prediksjonsresultater med «Support Vector Regression»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE.....	63
Figur 48 Prediksjon med lineær regresjon, uten knekkpunkt.....	64
Figur 49 Prediksjon med tradisjonell ET-kurve	65
Figur 50 Prediksjonsresultater for a) timesverdier, b) døgnverdier og c) døgnverdier basert på timesverdier med ET-regressor.....	66
Figur 51 Prediksjonsresultater for a) timesverdier, b) døgnverdier og c) døgnverdier basert på timesverdier med multipl lineær regresjon.....	67



Figur 52 ET-graf med multippel lineær regresjon. Sammenligning av predikerte og de faktiske verdiene.	68
Figur 53 Forholdet mellom målt- og predikert energiforbruk. For multippel lineær regresjon.....	68
Figur 54 Koeffisienten til de ulike variablene for a) timesverdier og b) døgnverdier, til multippel lineær regresjon	69
Figur 55 Prediksjonsresultater for a) timesverdier, b) døgnverdier og c) døgnverdier basert på timesverdier med «Random Forest»	70
Figur 56 ET-graf med «Random Forest». Sammenligning av predikerte og de faktiske verdiene.	71
Figur 57 Forholdet mellom målt- og predikert energiforbruk. For «Random Forest»	71
Figur 58 Koeffisienten til de ulike variablene for a) timesverdier og b) døgnverdier, til «Random Forest».	72

Tabell 1 Energiforbruk etter forbrukerkategori i Norge for 2018 . Kilde (Elektrisitet 2018, 2019) og (Fjernvarme og frikjøling, 2020).....	1
Tabell 2 Termiske egenskapene til de tre mest brukte byggematerialene. Verdier er hentet fra (Thue, 2016).	17
Tabell 3 Tidsforsinkelse for 30 cm tykk vegg med ulike materialer. Kilde (Haase & Andresen, 2007).....	17



TEGNFORKLARING

U_i	U-verdi for et i-element	[W/m ² K]
A_i	Areal for i-element	[m ²]
ψ_k	Kuldebroverdi av k-kuldebro	[W/mK]
l_k	Lengden av k-kuldebroet	[m]
χ_j	Punktkuldebroverdi	[W/K]
T_i	Innetemperaturen	[°C]
T_e	Utetemperaturen	[°C]
T_t	Balansetemperatur i ET-kurve	[°C]
n_{inf}	Luftskifte for infiltrasjon	[h ⁻¹]
V	Volumet	[m ³]
\dot{V}	Gjennomsnittlig ventilasjonsluftmengde	[m ³ /h]
η_T	Temperaturvirkningsgrad for varmegjenvinner	
V_T	Varmetapstall	[W/K]
G	Graddagstallet	[K dg]
T	Antall timer	[h]
C	Varmekapasitet pr. overflateareal	[Ws/Km ²]
C_p	Spesifikk varmekapasitet	[Ws/kgK]
ρ	Densitet	[kg/m ³]
t	Tykkelse	[m]
λ	Varmeledningsevne	[W/mK]
e_0	Grunnlast i ET-kurve	[W/m ²]
s	Stigningstall	
x_n, y_n	Målt datapar	
y_p	Predikert verdi	
e	Residual (feilledd)	



INNHOLDSFORTEGNELSE

FORORD	ii
SAMMENDRAG	iii
ABSTRACT	iv
FIGUR- OG TABELLISTE	v
TEGNFORKLARING	viii
1 INNLEDNING	1
1.1 PROBLEMSTILLING	1
1.2 BAKGRUNN.....	1
1.3 BOLIGBYGG OG ÅF-ADVANISA	2
1.4 FORMÅL	2
1.5 AVGRENSNING	3
2 TEORI	4
2.1 MASKINLÆRING	4
2.1.1 LÆRINGSMETODER	4
2.1.2 ALGORITMER.....	5
2.1.3 KVALITET AV PREDIKSJONSMODELL	7
2.1.4 ANNEN FORSKNING	8
2.1.5 PYTHON.....	9
2.2 ENERGIEFFEKTIVE BYGNINGER	10
2.2.1 BYGGTEKNISK FORSKRIFT.....	11
2.2.2 ENERGISTANDARD – NS3031	12
2.3 ENERGIBALANSE I BYGNINGER	13
2.3.1 ENERGITAP	14
2.3.2 ENERGITILFØRSEL.....	16
2.4 ENERGISIGNATUR	18
2.4.1 BRUK AV ENERGISIGNATUR	18
2.4.2 TIDSOPPLØSNING.....	19
2.5 ENERGIMARKEDET	20
2.5.1 ENERGINETT	21
2.5.2 AMS – MÅLERE.....	21
2.5.3 EPC – «ENERGY PERFORMANCE CONTRACTING».....	22



3 METODE	23
3.1 DATABEHANDLING.....	23
3.1.1 NEDLASTING AV DATA	23
3.1.2 FØRSTE GJENNOMGANG	25
3.2 ANALYSE.....	28
3.2.1 ET-KURVER	28
3.2.2 ESTIMERING AV OPPVARMING, VARMTVANN OG ANNET	30
3.2.3 REGELVERKET – TEK17 OG NS3031.....	32
3.3 PREDIKSJONSMODELLEN	33
3.3.1 VARIABLER.....	33
3.3.2 MASKINLÆRING I PYTHON	35
3.4 DATAKVALITET	38
3.4.1 RELIABILITET	38
3.4.2 VALIDITET	38
4 RESULTAT	39
4.1 DATABEHANDLING.....	39
4.2 ANALYSE.....	40
4.2.1 ALLE BYGNINGENE	40
4.2.2 BYGNINGER MED ELEKTRISK OPPVARMING	43
4.2.3 BYGNINGER MED FJERNVARME.....	46
4.3 VARIABLENE	50
4.3.1 TIMESVERDIER	50
4.3.2 DØGNVERDIER	52
4.3.3 DØGNVARIASJON	54
4.3.4 VARIABLER FOR EN TILFELDIG VALGT BYGNING.....	55
4.4 PREDIKSJON – ALLE BYGNINGER.....	57
4.4.1 PREDIKSJON MED ET-KURVE.....	57
4.4.2 PREDIKSJON MED MASKINLÆRINGSALGORITMER	58
4.5 PREDIKSJON – EN TILFELDIG VALGT BYGNING	64
4.5.1 ET-KURVE	64
4.5.2 MULTIPPEL LINEÆR REGRESJON – ALGORITMEN	67
4.5.3 «RANDOM FOREST» - ALGORITMEN	70



5 DISKUSJON.....	73
5.1 DATABEHANDLING.....	73
5.2 ANALYSE.....	74
5.3 PREDIKSJONMODELLEN	76
5.3.1 VARIABLENE	76
5.3.2 PREDIKSJON MED ET-KURVE.....	77
5.3.3 MULTIPPEL LINEÆR REGRESJON.....	78
5.3.4 RANDOM FOREST.....	79
5.4 VIDERE FORSKNING.....	80
6 KONKLUSJON	81
7 REFERANSER	82
8 VEDLEGG.....	86



1 INNLEDNING

1.1 PROBLEMSTILLING

Denne oppgaven er en analyse av energiforbruket av boligmassen til Boligbygg og utvikling av en prediksjonsmodell for energiforbruket ved hjelp av kunstig intelligens.

1.2 BAKGRUNN

Skjerpede energikrav i byggenæringen i de siste årene har bidratt til god fremgang i veien for å oppnå Norges klimamål før 2030. Til tross for dette er det fortsatt mange eksisterende bygningsmasser som kan ha stor besparingspotensiale. Energiforbruket til slike bygningsmasser kan gi en indikasjon på bespareelsesmulighetene. Slik som det vises i tabell 1, var elektrisitetsforbruket i husholdninger den andre største forbrukergruppen i Norge for 2018. Her kan man se at ca. 30% (Elektrisitet 2018, 2019) av det totale nettoforbruket av elektrisk kraft går til å dekke energibehovet fra husholdninger. Sammenlignet med fjernvarme for tilsvarende forbrukergruppe, gir det en god indikasjon på at de fleste bygningene i Norge benytter fortsatt elektrisitet av stor grad som energikilde.

Forbrukergruppe	Energiforbruket etter forbrukergruppe	
	Elektrisitet [GWh]	Fjernvarme [GWh]
Bergverksdrift/Industri	57 224	978
Tjenestestyring	27 135	3394
Husholdninger	38 516	1374
Total nettoforbruk	127 548	5746

Tabell 1 Energiforbruk etter forbrukerkategori i Norge for 2018. Kilde (Elektrisitet 2018, 2019) og (Fjernvarme og frikjøling, 2020)

Energibehovet fra husholdninger er i første omgang varmebehovet, som skyldes det kalde klimaet her i landet (Fordeling av strømforbruk, 2020). Dette er som regel summen av behovet til oppvarming og varmtvann. Spesielt i de eldre bygningene kan oppvarmingsbehovet utgjøre en betydelig andel av det totale energibehovet. Ved hjelp av f.eks. en EPC-modell kan slike bygninger oppgraderes. Da kan bygningene få en høyere energistandard som klarer å imøtekomme fremtidige mål om reduksjon av bedret privatøkonomi og klimagassutslipp (Kvaale & Jensen, 2011). En EPC-modell (energisparkontrakt) er en type avtale som inngås mellom en byggeier og et energitjenesteforetak. Målet er å oppnå en garantert reduksjon av energiforbruket av eksisterende bygningsmassen, som også er kostnadseffektiv (Hauge, Fredriksen, & Klinski, 2015). Energитjenesteforetaket trenger da et godt analysegrunnlag for å garantere denne reduksjonen. Nå som de fleste har installert AMS-målere (automatiske strømmålere), kan innsamlede data fra disse bidra til et godt analysegrunnlag.



Kunstig intelligens er en informasjonsteknologi som lenge har vært begrenset av prosesseringskraft og datamengde. Siden tilgjengeligheten på disse har fått en enorm økning de siste årene, har utviklingen av kunstig intelligens fått et kraftig oppsving. Selv om denne teknologien har varierende virkningskraft med dagens midler er det en teknologi med enormt potensiale som kan benyttes på alle typer problemer. Det er derfor viktig å forske på denne teknologien i alle felt for å avsløre dens fulle potensiale.

1.3 BOLIGBYGG OG ÅF-ADVANSIA

Boligbygg Oslo KF er et kommunalt foretak samtidig Norges største utleier av boliger (Boligbygg Oslo KF, 2020). ÅF Advansia er et av datterselskapene til ÅF Consult i Norge, og er såkalt energitjenesteforetak i dette tilfellet. Sammen med Boligbygg har de inngått en kontrakt om energibesparelse (EPC-Modell). Kontrakten omhandler reduksjon av energiforbruket i de kommunale bygninger ved hjelp av konkrete tiltak. Dette er snakk om en boligmasse bestående av 99 bygninger. Energiforbruket i bygningen kartlegges ved nærmere analyse av tilhørende data som er disponert av Boligbygg. De konkrete tiltakene er basert på en slik analyse. I samarbeid med ÅF-Advansia skal det i denne oppgaven utføres en mer detaljert analyse av disse dataene. Det er et ønske om å belyse nærmere hva disse dataene kan brukes til.

1.4 FORMÅL

Resultatmålet i denne oppgaven er å få en oversikt over hvilken tilstand bygningsmassen til Boligbygg er i med tanke på energiforbruk. I tillegg er resultatmålet å belyse hvor presist en prediksjon av energiforbruket basert på kunstig intelligens er sammenlignet med klassiske ET-kurver.

Prosessemålet i denne oppgaven er at gruppen vil opparbeide seg kompetanse innen maskinlæring og energiforbruk i bygninger. I tillegg vil gruppen tilegne seg kunnskaper om blant annet godt samarbeid, arbeidsinnsats og moral, oppfølging av fremdriftsplan, møtevirksomhet, det å benytte relevant litteratur og konfliktløsning.

Effektmetålet i denne oppgaven er tilknyttet oppgavens langsiktige virkning i byggbransjen, spesielt energi og miljø i bygg. Nyttiggjørelse av resultater fra denne oppgaven kan fungere som en veiledning til store byggeiere. Spesielt til de som har ambisjoner om å redusere sine kostnader ved innkjøp av energi. I tillegg belyses ET-kurver som metode.



1.5 AVGRENSNING

Problemområdet i denne oppgaven omhandler energi og miljø i bygg. Teorien i denne oppgaven er begrenset til generell energibalanse i bygg med hensyn til tap og tilskudd. Dette krever at leseren har kunnskap innen elementær bygningsfysikkteori. Analysen baserer seg på data fra de definerte bygningsmassene i EPC-kontrakten mellom Boligbygg og ÅF-Advansia. Hvordan disse dataene blir målt og behandlet er ukjent, så de forutsettes at de er pålitelige.

Det er bestemt at oppgaven skal benytte maskinlæring som er en gren innen kunstig intelligens. I denne oppgaven er det avgrenset til å ikke gå dypere inn i de ulike algoritmene i prediksjonsmodellen, kun vise hvilke resultater som kan forventes. Det er benyttet seks maskinlæringsalgoritmer. Teorien som omhandler maskinlæring, avgrenses til metodene som benyttes i oppgaven.



2 TEORI

2.1 MASKINLÆRING

Maskinlæring er en type dataanalyse, og er den delen innen kunstig intelligens som omhandler bruk av statistiske metoder til å finne mønstre i store datamengder. Disse metodene har i dag et bredt spekter i bruksområdet som f.eks. selvdrevne biler, bildegjenkjenning og søkemotorer. Metoden baserer seg på at en modell skal «lære» av innsamlede data i istedenfor å bli programmert. Modellen må naturligvis formes gjennom programmering først, ofte med en matematisk algoritme. Så er det parametere i modellen som justeres basert på dataene. Slike modeller kan blant annet brukes til prediksjon, eller estimering, basert på en eller flere inngangsverdier (variabler). Hva maskinen lærer, altså på hvilken måte modellen påvirkes av dataene, kan ofte være vanskelig for mennesker å forstå. De kan derfor refereres til som svarte bokser (Black Boxes). Hvordan modellen kommer frem til resultatet er ikke lenger viktig, så lenge resultatet er bra (James, Witten, Hastie, & Tibshirani, 2017).

2.1.1 LÆRINGSMETODER

De ulike algoritmene kan kategoriseres etter hvilken læringsmetode som benyttes i modellen. I hovedsak er det tre kategorier; veiledende læring, ikke veiledende læring, eller forsterket læring. Hvilken læringsmetode som benyttes bestemmes som regel av hvilke data som er tilgjengelig. Selv om veiledende læring og ikke veiledende læring er ulike baserer læringen i begge metodene seg på innsamlede data. Forsterket læring baserer seg ikke på innsamlede data. Isteden lar man modellen finne ut av ting selv, ved at algoritmen programmeres til å hele tiden å utføre ulike tilfeldige handlinger. Det defineres en belønning eller en straff dersom modellen tilfeldigvis gjør noe riktig eller galt. På denne måten forsterkes den ønskede oppførselen. Dette kan sammenlignes med å trene opp en hund, og disse modellene anvendes ofte på dataspill som f.eks. sjakk og har vist overmenneskelige ferdigheter på kort tid. Det er da kun kompleksiteten til modellen og prosesseringskraften til maskinen som begrenser resultatene, (Tidemann & Elster, 2020).

Ved veiledende læring er algoritmen programmert spesifikt til å forutsi en utgangsverdi basert på en eller flere inngangsverdier. Under opplæringen får modellen fasiten (utgangsverdien) for en rekke inngangsverdier og lærer på den måten sammenhengene mellom dem. Basert på disse sammenhengene kan modellen predikere med nye inngangsverdier den ikke har sett før. Dette er en mer effektiv måte å «trene» modellen, og er derfor den mest populære og nyttige læringsmetoden innen maskinlæring. Metoden har vist at i enkelte tilfeller, som f.eks. bildegjenkjenning, presterer disse modellene bedre enn mennesker, (Tidemann & Elster, 2020). Bildegjenkjenning er blant de mest betydningsfulle teknologiene innen kunstig intelligens. Det er den viktigste teknologien i selvdrevne biler, men kan også brukes til f.eks. diagnostisering av MR-undersøkelse i medisin. Ulempen med denne læringsmetoden er at den ofte krever store datamengder med god kvalitet og variasjon. Kvaliteten til resultatene blir derfor ofte begrenset av dette istedenfor kompleksiteten til modellen. I følge Elon Musk, grunnlegger av Tesla, er det dette som er den store utfordringen i utviklingen av selvdrevne biler. Selv om teknologien er på plass kreves det eksempler av alle ulike situasjoner i trafikken på film for å trene modellen.



I modeller som benytter ikke veiledende læring, blir det ikke gitt noe fasit (utgangsverdier) for inngangsverdiene. Da er isteden algoritmen programmert til å finne strukturen i dataene selv. Disse modellene får en litt annen natur og brukes derfor til å løse andre typer problemer som f.eks. gruppering av data. Slike ikke-statistiske problemer er mennesker fortsatt mye bedre på enn maskiner. (Tidemann & Elster, 2020)

2.1.2 ALGORITMER

Siden det kan være vanskelig å vite akkurat hva modellen skal lære av dataen, er det ofte umulig å forutsi hvilken maskinlæringsalgoritme som vil gi best resultat. Det krever i så fall lang erfaring i feltet. I praksis sees det derfor ofte at flere ulike algoritmer testes for samme problem. Innenfor veiledende læring kan det likevel skilles mellom to hovedkategorier basert på problemet modellen forsøker å løse; regresjon og klassifikasjon. Problemer knyttet til blant annet bildegjenkjenning er klassifiseringsproblemer. Et eksempel kan være en modell som skal bedømme om en pasient har hjernesvulst eller ikke basert på en MR-undersøkelse. Da er resultatet en klasse. Ved et regresjonsproblem er resultatet et tall eller en rekke av tall. En modell som predikerer det daglige strømforbruket til en bygning, forsøker å løse et regresjonsproblem. De fleste maskinlæringsalgoritmer kan brukes til både regresjon og klassifikasjon, men modellene får naturligvis litt ulik struktur, (James, Witten, Hastie, & Tibshirani, 2017).

Multipel Lineær Regresjon er en velkjent og populær maskinlæringsalgoritme fordi den er enkel og lett å interpretare. I tillegg viser det seg at den passer bra for veldig mange ulike situasjoner og gir gode resultater. Det er ett naturlig sted å begynne dersom man har et regresjons problem, siden også mange andre maskinlæringsalgoritmer baserer seg på denne. Istedenfor å benytte seg av kun en variabel (én lineær sammenheng), benytter metoden seg av flere variabler som vist i formel [1]. Prediksjonsmodellen blir da flerdimensjonal og mye mer fleksibel.

$$y = a + \sum_{j=1}^k b_j * x_j + e \quad [1]$$

I regresjonen benyttes minste kvadrats metode til å justere parameterne a og b for alle variablene. Det er da snakk om å minske kvadratet av feilleddet (residuen) e. En annen maskinlæringsalgoritme som bygger på forskjellige videutviklinger av lineær regresjon, er «Kernel Ridge Regression». Ridge regresjon benytter en litt mindre intuitiv metode enn minste kvadrat av feilleddet til å tilpasse parameterne, men som er veldig nærliggende. Metoden går ut på «krympe» kvadratet av b og tilpasse en ny parameter λ , (James, Witten, Hastie, & Tibshirani, 2017).

En annen tilnærming til regresjonsproblemet er såkalte tre-baserte maskinlæringsalgoritmer, også kalt beslutningstrær «Decision Tree». Disse algoritmene håndterer også flere variabler, men benytter en helt annen metodikk i regresjonen. Med denne metoden deles datasettet opp i mindre segmenter, hvor hvert segment er med på å bestemme utfallet av prediksjonen. Dette kan visualiseres som et tre hvor hver gren er en beslutning, derav navnet beslutningstre. Som regel baserer beslutningen i hvert segment seg på enten snittet eller median av dataene i segmentet. Etter at flere beslutninger har blitt gjort ved hver grensplitt (indre node) ender man opp til slutt med et løv (ytre node) som er en predikert verdi. Selv om dette virker som enkel metode å forstå, ligger kompleksiteten i hvordan datasettet skal inndeles. Med

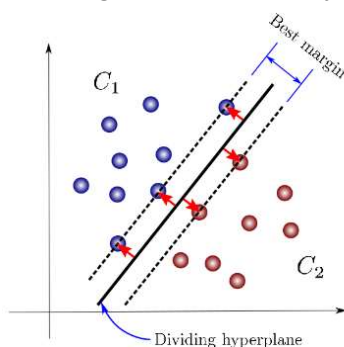
mindre datasettet er veldig lite, finnes det nesten uendelige mange måter å inndeleg datasettet på i ulike trær. Det er derfor umulig for en datamaskin å teste alle for å se hvilke tre som gir best prediksjon. Modeller som baseres på slike beslutningstrær endre da ofte opp med høy varians.

Den mest vanlige metoden som brukes til inndeling av datasettet kalles «recursive binary splitting». Da er tankegangen å starte øverst og velge den inndelingen som fører til det minst avviket. Det tas da ikke høyde for senere inndelinger lengre ned i treet som muligens kunne ført til et bedre tre. Et slikt tre har en tendens til å bli veldig kompleks og derfor overtilpasset, som betyr at den passer veldig godt på treningsdataene, men ikke nødvendig på nye data. Regresjonstrær kan beskrives matematisk av formelen [2].

$$y = \sum_{m=1}^M c_m * 1_{(X \in R_m)} \quad [2]$$

Svakhetene til beslutningstrær er hovedsak høy varians og overtilpassing. For å forbedre begge disse delene er det utviklet en maskinlæringsalgoritme kalt «Random Forest». Den opererer ved å sette opp et mangfold av slike beslutningstrær og basere prediksjonen på et snitt av disse. Måten disse settes opp er ved å hele tiden bruke tilfeldige deler av treningssettet. Inndeling av datasettet i de ulike beslutningstrærne tvinges til å basere seg mer på tilfeldighet nettopp for å skape av treningsdatasettet. Deretter blir det sendt ut en gjennomsnittlig prediksjon for hver enkelt «Decision Trees». Dette er en algoritme som jobber ganske effektivt på store datasett. Den kan blant annet håndtere tusenvis av inngangsvariabler uten å slette noe av variablene, og deretter gi et estimat på hvilke variabler som er nødvendig. En annen algoritme som også baserer seg på beslutningstrær er «Gradient Boost». Denne ML-modellen bygger såkalte «Decision Trees» en om gangen, i motsetning til «Random Forest» som kan bygge flere uavhengige «Decision Trees».

«Support Vector Regression» er annen maskinlæringsalgoritme som bygger på metodene i støttevektormaskiner «Support Vector Machines». Tilsvarende som for minste kvadrats metode, forsøker også denne algoritmen å tilpasse parameterne med regresjon. Men benytter verdien til en annen funksjon til å bestemme parameterne. Parameterne justeres slik at absolutte avviksverdier som er over en viss verdi bidrar til reduksjon i funksjonen. De parameterne som ender opp med høyest funksjonsverdi benyttes. Dette er en utvidelse av marginen brukt i «support vector classifiers», men tilpasset regresjonsproblemet, (James, Witten, Hastie, & Tibshirani, 2017). Støttevektormaskiner er et stort felt innen maskinlæring og ble i utgangspunktet utviklet til å løse klassifiseringsproblemer. Tankegangen i metoden er å sørge for en såkalt beslutningsgrense av et sett med data, for å skille forskjellige klasser og øke marginen mellom en linje og de punktene som er nærmest til linjen (Carrasco, 2020), se Figur 1.



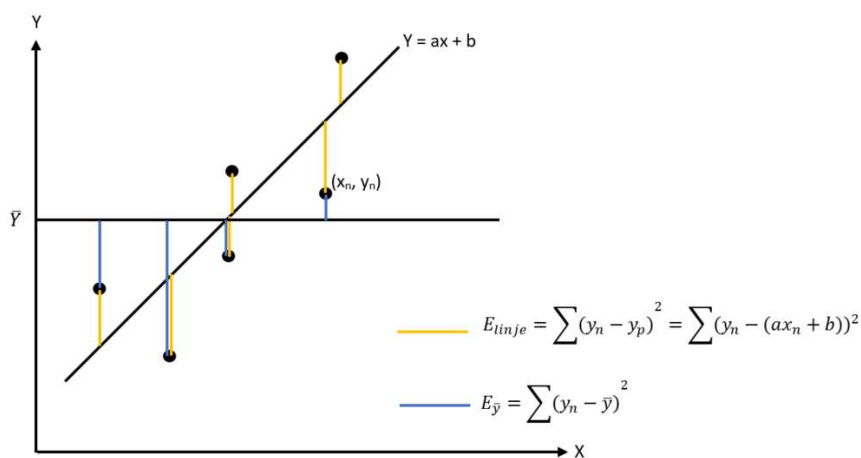
Figur 1 Presentasjon av SVR-modellen. Kilde: (Carrasco, 2020)

2.1.3 KVALITET AV PREDIKSJONSMODELL

Korrelasjon er et mål som skal beskrive hvor mye to størrelser henger sammen (Frøslie, 2020). Dette angis med *korrelasjonskoeffisienten* – r . Korrelasjonskoeffisienten er et dimensjonsløst tall som kan ha en verdi mellom -1 og 1. Negativ fortegn betyr at det er en negativ korrelasjon, i motsetning til positiv korrelasjon. Korrelasjonskoeffisient på f.eks. 0,9 vil bety at det er en positiv korrelasjon, og det er en sterk sammenheng mellom to størrelser (Frøslie, 2020).

$$r = \frac{n\Sigma xy - \Sigma x * \Sigma y}{\sqrt{n(\Sigma x^2) - (\Sigma x)^2} * \sqrt{n(\Sigma y^2) - (\Sigma y)^2}} \quad [3]$$

Problemet er at dette tallet ikke er lett å interpretere, og derfor må man se på «*coefficient of determination*» eller såkalte R^2 -verdi. R^2 -verdi et statistisk mål som beskriver hvor stor andel av variasjonen i y som kan forklares av variasjonen i x . Dette kan bestemmes på flere måter, men tankegangen er å finne hvor stor andel av totalvariasjonen som ikke kan beskrives med regresjonslinjen. Restandelen betyr i dette tilfellet, den delen av variasjonen som kan beskrives med regresjonslinjen. I Figur 2 presenteres regresjonslinjen gjennom punktene med summert feil i forhold til linjen (E_{linje}) og summert feil i forhold til snittverdi av y ($E_{\bar{y}}$).



Figur 2 Illustrasjon av korrelasjonskoeffisient

E_{linje} beskriver hvor mye av variasjonen som ikke kan beskrives med regresjonslinjen, mens $E_{\bar{y}}$ beskriver den totale variasjonen i y . Dette medfører til å bestemme hvor mange prosent av variasjonen i y som kan beskrives med variasjonen i x eller regresjonslinjen på følgende måte;

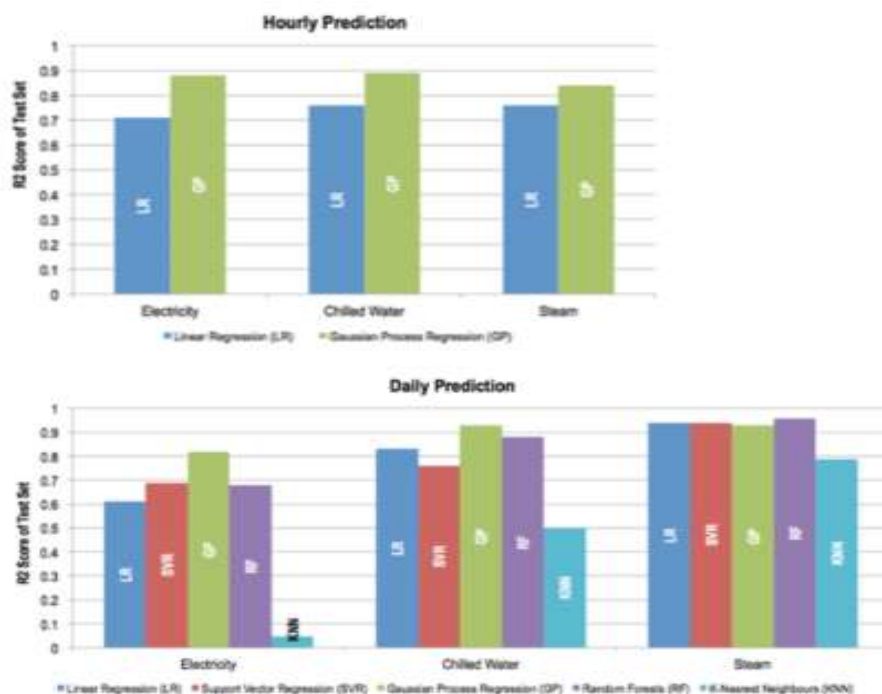
$$R^2 = 1 - \frac{E_{linje}}{E_{\bar{y}}} \quad [4]$$

RMSE-verdi er en annen statistisk mål som beskriver i hvilken grad de faktiske verdiene varierer fra regresjonslinjen, og kalles også for gjennomsnittlige kvadratfeil. RMSE-verdien presenterer avviket med faktiske størrelsen, i motsetning til R^2 -verdi som presenterer kun i form av en prosentsats. Denne verdien er basert på samme prinsipp som ble beskrevet på forrige side, og kan derfor bestemmes på følgende måte;

$$RMSE = \sqrt{\frac{1}{n} * E_{linje}} = \sqrt{\frac{1}{n} * \sum (y_n - y_p)^2} \quad [5]$$

2.1.4 ANNEN FORSKNING

En gruppe med forskere fra CGBC (*CENTER FOR GREEN BUILDINGS AND CITIES*) i Harvard University har forsøkt å utarbeide ulike maskinlæringsmodeller til å predikere energiforbruket til bygningene i Harvard campus. De har valgt se på *Gaussian Process Regression*, *Linear Regression*, *K Nearest Neighbour Regression*, *Random Forest Regression* og *Support Vector Regression* (Yan, Wette, & Xie, 2014). Disse algoritmene ble brukt for å predikere times- og døgnforbruk til de ulike energipostene for ett av bygningene i Harvard campus. Figur 3 viser prediksjonsresultatene for de fem maskinlæringsmodellene. Her har det blitt konkludert at blant annet *Gaussian Process* og *Random Forest* er de to modellene som gir de mest presise prediksjonene av energiforbruket.



Figur 3 Prediksjonsresultater for campusbygningen i Harvard. Kilde: (Yan, Wette, & Xie, 2014)



En annen relevant forskning er utført på en skole i Oslo. Her var målet å analysere ET-kurver og planlegge energiforbruket av oppvarming og varmtvann. I denne analysen hadde forfatterne valgt å inndele ET-kurven i temperaturavhengig og temperaturuavhengig del, for å foreta separat analyse. For den temperaturavhengige delen ble det benyttet ulike maskinlæringsalgoritmer som MLR, SVM, PLS og LASSO. For temperaturuavhengige delen ble det utviklet timesprofiler av energiforbruket. De har foretatt en korrelasjonsanalyse for å avdekke hvilke variabler som kan ha størst innvirkning for begge delene i kurven. Resultatet av dette viser at den høyeste negative korrelasjonen oppstår med utetemperaturen med en verdi lik $-0,516$. Korrelasjon med vind og timesverdier i døgnet viser seg å ha lave verdier, noe som betyr at de ikke har noe betydelig innvirkning på energiforbruket. Generelt sett har de konkludert at MLR (Multipl lineær regresjon)-algoritmen predikerer bedre enn andre de algoritmene (Tereschchenko, Ivanko, Nord, & Sartori, 2019).

2.1.5 PYTHON

Python er et programmeringsspråk som er utviklet av Guido Van Rossum i 1989 (Python, 2020). I denne oppgaven benyttes Python 3.7 i en programvare kalt «Spyder» til all analyse. Programmet inneholder en rekke funksjoner som tillater brukeren å analysere data i tabell format, samt visuelt ved å generere ulike grafer. I tillegg finnes andre funksjoner som kan brukes, og gjøre det enklere ved programmering. De ulike funksjonene hentes fra ulike biblioteker som vanligvis importeres på starten av et program. De mest brukte bibliotekene i denne oppgaven er blant annet *Pandas*, *Matplotlib*, *Numpy*, *Seaborn*, *Scikit-learn* osv. «Scikit-learn» er biblioteket som brukes i forbindelse med maskinlæring. «Scikit-learn» er et stort bibliotek som består av veldig mange funksjoner. Det er i hovedsak dette biblioteket som brukes til å konstruere de ulike modellene og teste ulike maskinlæringsalgoritmer.

Pandas er et av de andre bibliotekene benyttes i stor grad. Dette er på samme måte som «Scikit-learn» et veldig stort bibliotek med mange ulike funksjoner. Biblioteket er utviklet spesielt til å håndtere store data mengder i såkalte *Pandas* datarammer. Det kan også benyttes til å lese og skrive til filer. Med dette menes å laste inn data fra f.eks. CSV-filer og generere nye filer med data. Biblioteket benyttes også til å analysere og generere figurer i kombinasjon med *Matplotlib*.



2.2 ENERGIEFFEKTIVE BYGNINGER

I dag er Norge mer bevisst på energiforbruk og klimapåkjenninger. Dette vil blant annet føre til at myndighetene stiller strengere energikrav. Ved bruk av ulike sertifiseringsverktøy vil da en ambisiøs byggherre klare å tilfredsstille disse kravene, men samtidig oppnå et enda bedre energinivå. Dette gjelder spesielt for nybygg. Da har byggherren allerede fra starten, stor innflytelse på hvor energieffektiv bygningen kan bli. Det er først i eksisterende bygningsmasser det vil være snakk om energieffektive bygg. Dette er noe som skal bli ivaretatt i henhold til energimerkeforskriften gjennom energimerkeordningen. Dette er en ordning som går ut på å utarbeide en energiattest til bygget som skal presentere byggets energistandard (Om energimerkeordning, 2009).

Energieffektivisering av eksisterende bygninger er et viktig tiltak i veien for å oppnå dagens forskrift og fremtidens klimamål. I denne sammenhengen er såkalte *Kyotopyramiden* et viktig og anerkjent verktøy som kan tas i bruk, se Figur 4. Denne pyramiden viser i hvilken rekkefølge de energibesparende tiltakene bør gjennomføres. Formålet er å energieffektivisere et bygg så mye som mulig (Energieffektivisering, 2020). Rekkefølgen på disse tiltakene går nedefra og oppover. I dag kan de ulike tiltakene bli støttet økonomisk av statlige organer. Myndighetene har satt inn en del ressurser for å realisere klimamålene og samtidig motivere byggeiere til å tenke på energieffektivitet ved nybygg eller i eksisterende bygningsmasser. Slike ressurser er blant annet ordninger gjennom Enova og Husbanken.



Figur 4 Kyotopyramiden som viser de energibesparende tiltakene.

2.2.1 BYGGTEKNISK FORSKRIFT

Byggeteknisk forskrift (TEK) har stadig vært under endring, helt siden 1965 og frem til i dag. Myndighetene har først i 1997 utarbeidet en teknisk forskrift med et eget kapittel som omhandlet energikrav i bygninger, i henhold til kommunal- og moderniseringsdepartementet. Frem til i dag har Norge hatt gode miljøambisjoner, noe som har ført til at energikravene i gjeldende kapittel har blitt strengere. Å senke akseptable u-verdien for et vindu fra 2,0 i TEK97 til 0,8 i TEK17, er bare et eksempel på dette. I dag må alle nye bygg tilfredsstillende kravene som er vist på Figur 5. Oppfylte TEK-krav vil samtidig sørge for at bygget oppnår energikarakteren C i henhold til energimerkeordningen. Det er viktig å påpeke at først i TEK17 er det satt konkrete krav til energiltak som skal sørge for at det totale netto energibehovet for bygningen ikke overstiger spesifiserte energirammer. Dette er et alternativ til første ledd i §14-2 (1). Ellers har ikke minimumskravene til energieffektivitet endret seg så betydelig fra TEK10 til TEK17, bortsett ifra minstekravet til lekkasjetall som er på 1,5 i dag i motsetning til 3,0 i TEK10.

Bygningskategori	Totalt netto energibehov [kWh/m² oppvarmet BRA per år]
Småhus, samt fritidsbolig over 150 m ² oppvarmet BRA	100 + 1600/m ² oppvarmet BRA
Boligblokk	95
Barnehage	135
Kontorbygning	115
Skolebygning	110
Universitet/høyskole	125
Sykehus	225 (265)
Sykehjem	195 (230)
Hotellbygning	170
Idrettsbygning	145
Forretningsbygning	180
Kulturbygning	130
Lett industri/verksteder	140 (160)

a)

	Energiltak	Småhus	Boligblokk
1.	U-verdi yttervegg [W/(m ² K)]	≤ 0,18	≤ 0,18
2.	U-verdi tak [W/(m ² K)]	≤ 0,13	≤ 0,13
3.	U-verdi gulv [W/(m ² K)]	≤ 0,10	≤ 0,10
4.	U-verdi vinduer og dører [W/(m ² K)]	≤ 0,80	≤ 0,80
5.	Andel vindus- og dørareal av oppvarmet BRA	≤ 25 %	≤ 25 %
6.	Årsgjennomsnittlig temperaturvirkningsgrad for varmegjenvinner i ventilasjonsanlegg (%)	≥ 80 %	≥ 80 %
7.	Spesifikk vifteeffekt i ventilasjonsanlegg (SFP) [kW/(m ³ /s)]	≤ 1,5	≤ 1,5
8.	Luftlekkasjetall per time ved 50 Pa trykkforskjell	≤ 0,6	≤ 0,6
9.	Normalisert kuldebroverdi, der m ² angis som oppvarmet BRA [W/(m ² K)]	≤ 0,05	≤ 0,07

b)

Figur 5 Utklipp av tabeller som viser dagens krav til energieffektivitet i bygg. A) viser energiramme, mens b) er oversikt over energiltak som må oppfylles for å oppnå kravet om energieffektivitet. Kilde: (Lovdata, 2020).



2.2.2 ENERGISTANDARD – NS3031

NS3031 er en europeisk standard som er tilpasset for dokumentasjon av energikrav i forskrift om tekniske krav til byggverk (TEK). Totale netto energibehovet blir beregnet i henhold til denne standarden. I standarden er det oppgitt normert årlig energibehov spesifisert for tre ulike poster; energibehov til belysning, - utstyr og varmtvann, se figur under.

Bygningskategori	Belysning ^a		Utstyr ^b		Varmtvann ^b	
	W/m ²	kWh/(m ² ·år)	W/m ²	kWh/(m ² ·år)	W/m ²	kWh/(m ² ·år)
Småhus	2,9	17	4	23	5,1	30
Boligblokker	2,9	17	4	23	5,1	30
Barnehager	8	21	2	5	3,8	10
Kontorbygg	8	25	11	34	1,6	5
Skolebygg	10	22	8	12	1,5	4,5

Figur 6 Utklipp av NS3031, tabell A.1. Energi- og effektbehov for henholdsvis belysning, utstyr og varmtvann. Kilde; (NS3031, 2020)

Slik som det observeres fra Figur 6, så er f.eks. normert årlig energibehov for oppvarming av tappevann for boligblokk satt lik 30 kWh/m² oppvarmet BRA per år. Dette tilsvarer ca. 32 % i forhold til kravet i tabell A i Figur 5. På samme måte utgjør belysning ca. 18%, og utstyr ca. 24%. Legg merke til at det ikke er oppgitt noe spesifikke krav om energibehov til romoppvarming. Men dette kan enkelt estimeres med følgende formel, dersom bygningskroppen og graddagstallet for området er kjent;

$$E_{\text{oppvarming}} = V_t * G * T \quad [6]$$

I samme standard blir det også presentert veiledende systemvirkningsgrader for de ulike oppvarmingssystemene i bygningene. Tabell B.10, presenterer systemvirkningsgrader for ulike oppvarmingssystemer for bygg eldre enn 1990. Her er det spesifisert at bygg som har fjernvarmeradiatorer som energiforsyningssystem har en systemvirkningsgrad på 0,86. Termostatstyrt direkte elektrisk oppvarming (panelovner), har en systemvirkningsgrad på 0,98 (NS3031, 2020).



2.3 ENERGIBALANSE I BYGNINGER

For at et bygg skal opprettholde energibalansen må den tilførte effekten eller energien være lik tapt effekt eller energi. Sammenhengen mellom tilskudd og tap er beskrevet i formel [7]. Det er ikke tatt hensyn til varmelagring som vil oppstå pga. termiske egenskaper til de ulike materialene.

$$Q_{sol} + Q_{lys} + Q_{uts} + Q_{per} + Q_{fan} = H_D + H_U + H_G + H_V + H_{inf} \quad [7]$$

der;

Q_{sol} = Solvarme

H_D = Transmisjonstap til det fri

Q_{lys} = Varmer fra lys

H_U = Transmisjonstap til uoppvarmede soner

Q_{uts} = Varmer fra utstyr

H_G = Varmetap mot grunnen

Q_{per} = Varmer fra personer

H_V = Ventilasjonsvarmetap

Q_{fan} = Varmer fra vifter/pumper

H_{inf} = Infiltrasjonsvarmetap

Denne balansen er svært avhengig av en del faktorer som utetemperatur, solinnstråling og vind. For eksempel et bygg i Oslo (3563 graddager) med et oppvarmingsbehov på 400 000 kWh/år, vil ha en konstant forbruksendring på ca. 112 kWh/dg K. Dvs. dersom utetemperaturen i et døgn faller fra 5 °C til 1 °C vil økningen i forbruket tilsvare ca. 448 kWh i døgnet. På en annen side kan vinden i tillegg spille en viktig rolle. En utilstrekkelig tett klimaskjerm og for mye interne åpninger i et høyt boligbygg kan forsterke såkalte skorsteinseffekten. Noe som vil medføre til store luftbevegelser og økt behov for oppvarming (Stack effect, 2020).

Graddagtallet, G, er da et uttrykk som kan tas i bruk i denne sammenhengen for å beregne bygningers oppvarmingsbehov gjennom året. Antall graddager i året bestemmes ved å summere differansen mellom døgnmiddeltemperatur og basistemperatur (17 grader i Norge). Det antas da at «gratisvarme» fra sol, lys, mennesker osv. vil bringe til at innnetemperaturen stiger opp til 20/21 °C, (Thue, 2016). Graddagstallet kan også temperaturkorrigeres dersom energiforbruket skal analyseres over en lengre periode. Dette er for å eliminere klimatiske variasjoner. Dette er også ofte knyttet opp mot fyringsperioden, som her i Norge starter når døgnmiddeltemperaturen synker under 11°C om høsten og slutter når den kommer over 9 °C om våren (Klima i Norge 2100, 2015).



2.3.1 ENERGITAP

Energitapet eller varmetapet i et bygg har stor betydning ved bestemmelse av energibehovet. Med kjennskap til spesifikke varmetapstallet, og antall graddager kan energibehovet til oppvarming bestemmes. De elementære bygningsfysiske varmetapsmekanismer foregår hovedsakelig på følgende måter;

Transmisjonstap er det varmetapet som oppstår pga. temperaturforskjellen over bygningsdelen. Dvs. stor temperaturforskjell vil bidra til økt varmetap. Dette beskrives ved hjelp av U-verdi [$\text{W}/\text{m}^2\text{K}$]. Det er såkalte varmegjennomgangskoeffisienten, og er et mål som skal da beskrive isoleringsevnen til en bygningsdel. Fra 1969 til 1986 ble denne verdien benevnt som k-verdi med en omregningsfaktor i bygningsforskrifter. Først etter 1986 ble dette erstattet til U-verdi uten noe omregningsfaktor (Thue, 2016). Å vite sammensetningen av bygningsdelen i detalj samt temperaturredifferansen, er viktig for å bestemme U-verdien. Ofte må dette beregnes manuelt, men en del materialleverandører oppgir som regel denne verdien. Transmisjonstapet beregnes i henhold til NS3031 punkt 6.1.1.1.1 (10) med følgende formel;

$$H_D = \sum_i U_i * A_i + \sum_k \psi_k * l_k + \sum_j \chi_j \quad [8]$$
$$Q_{trans} = H_d * (T_i - T_e)$$

Infiltrasjonstap er det varmetapet som oppstår pga. luftutveksling gjennom utettheter i klimaskjermen. Drivkreftene til dette tapet er blant annet vind og oppdriftskrefter som oppstår pga. temperaturforskjeller. Derfor er beliggenhet og skjerming av bygget to viktige faktorer som kan ha stor påvirkning til tapet. Byggets tetthet måles ved å sette bygget i under- og overtrykk ved 50 Pa trykkforskjell med hjelp av en vifte, for å observere det totale luftstrømmen inn og ut av bygget. Forholdet mellom denne luftstrømmen og det oppvarmede volumet i bygget, er såkalte lekkasjetallet [h^{-1}]. Noe som da beskriver antall luftomsetninger i løpet av en time. Jo lavere dette tallet er jo tettere er bygget, dermed vil det medføre til redusert varmetap pga. infiltrasjon. Infiltrasjonstapet beregnes i henhold til NS3031 punkt 6.1.1.1.5 (27) med følgende formel;

$$H_{inf} = 0,33 * n_{inf} * V \quad [9]$$
$$Q_{inf} = H_{inf} * (T_i - T_e)$$



Ventilasjonstap er det varmetapet som oppstår via ventilasjon. Dette er avhengig om bygget har naturlig, mekanisk eller balansert ventilasjon. Bygg med naturlig ventilasjon vil ha et varmetap ofte gjennom en veggventil som kan være plassert i ulike rom. Vind og temperaturforskjellen mellom inne- og uteluft er blant annet drivkreftene for dette tapet. Det er ikke noe spesifikke krav til et slikt ventilasjonstap, men det kan ofte antas at naturlig ventilasjon gir 0,4 luftomsetninger av oppvarmede volumet i løpet av en time (Thue, 2016). Mekanisk og balansert ventilasjon kan medføre til et mindre varmetap. Dette skyldes fordi varmegjenvinning er som regel mulig i slike ventilasjoner. Varmegjenvinningen skjer via varmevekslere som sørger for at varmen fra avtrekkslufta benyttes til oppvarming av friskluft. Temperaturvirkningsgrad $[\eta_T]$ er et mål som da brukes i denne sammenhengen. Den sier noe om hvor stor del av varmen fra avtrekksluften som kan gjenvinnes via en varmeveksler. Varmebatteri og roterende gjenvinner er eksempler på slike varmevekslere. Dvs. at en varmeveksler med høy temperaturvirkningsgrad i mekanisk/balansert ventilasjonsanlegg, vil medføre til lite ventilasjonsvarmetap (Ecodesign, 2020). Beregning av ventilasjonsvarmetap er oppgitt i henhold til NS3031 punkt 6.1.1.1.4 (25) med følgende formel;

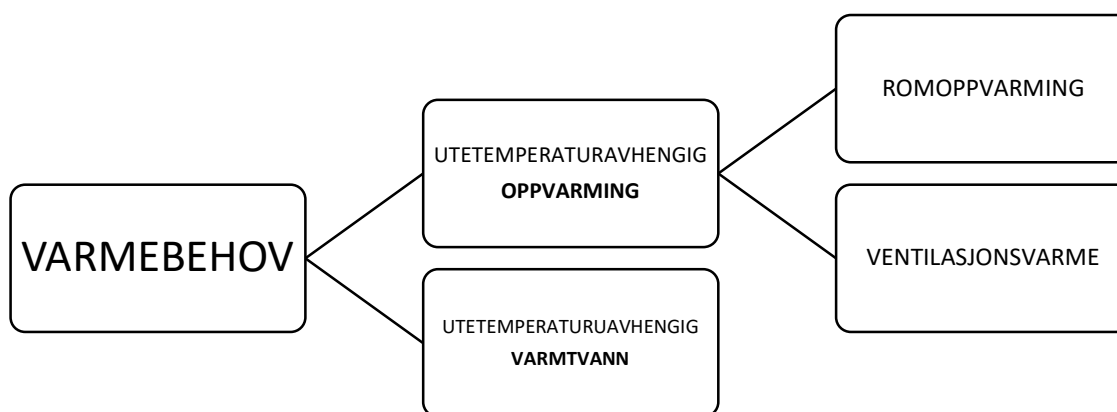
$$H_V = 0,33 * \dot{V}(1 - \eta_T) \quad [10]$$
$$Q_v = H_V * (T_i - T_e)$$

2.3.2 ENERGITILFØRSEL

I ett kalt klima er oppvarming en sentral del i våre bygninger. Gjennom historien har energibruken i husholdninger steget, og spesielt da andel av energiforbruk som er temperaturavhengig (oppvarming), se Figur 7. Denne stigningen har ikke forsettet like mye siden 1990-tallet. Årsaken til dette er mildere vær, mindre vekst i boligarealer pr. person og redusert spesifikk energibruk, ifølge en utgitt rapport i 2011 av Vestlandsforskning (Aall, Simonsen, & Hille, 2011). Temperaturavhengige andelen av energiforbruket ligger omkring 40-65% for boligblokker som er bygd mellom 1951-1998 (Enovas Byggestatistikk 2017, 2017). Alternativt har SINTEF i forbindelse med et EU-prosjekt estimert denne andelen til ca. 64% (Fordeling av strømforbruket, 2020). For å opprettholde varmekomforten og sikre dekning av oppvarmingsbehovet må bygget få tilført varme gjennom et varmedistribusjonsanlegg eller et varmeelement (BKS 421.501, 2017). Lengden eller behovet for oppvarming for et bestemt år er svært avhengig av temperatur, fyringssesongens lengde og graddagstallet for fyringssesongen. Hovedsakelig kan løsninger for romoppvarming deles inn i tre hovedtyper (Dokka & Andresen, 2012):

- Punktoppvarmingsløsninger; panelovner, vedovner eller radiatorer.
- Integrerte oppvarmingsløsninger; vannbåren/elektrisk gulvvarme.
- Luftbårne oppvarmingsløsninger; varmluft via ventilasjonsanlegg eller en varmepumpe.

Et oppvarmingssystem påvirkes særlig av systemvalg og styring. Det er viktig å ha et varmeanlegg som har best mulig virkningsgrad ved de mest aktuelle belastningsnivåene (Thue, 2016). I slike tilfeller er det derfor spesielt viktig at bygget observeres i sin helhet for å kunne vurdere varmeteknisk, og eventuelt spesifisere hvor de aktuelle tiltakene kan iverksettes.



Figur 7 Oversikt på varmebehovet i bygninger - definisjon iht. TEK17



I tillegg til dette får bygget tilført «gratisvarme» fra sol, interne laster og termiske forhold. Den tilførte varmen fra solen påvirkes i stor grad av orientering, avskjerming og størrelsen på vinduene i bygget. Interne laster er lys, utstyr, personer og vifte/pumper. Dette kan påvirkes av blant annet utstyrmengde, antall personer og hva slags aktivitet det foregår i bygget. Termiske forhold er derimot avhengig av de eksponerte termiske massene, dets evne til å lagre og lede termiske energi. Avhengig av samspillet mellom disse termiske egenskapene, kan termiske massen forflytte energibehovet til perioder hvor behovet ikke er så stor (nattetid). Høyere termisk masse kan redusere de daglige temperatursvingningene (tidsforsinkelse) over konstruksjonen, noe som vil sørge til et bedre termisk inn klima og energieffektivt bygg (Haase & Andresen, 2007). Bestemmelse av termiske massen blir uttrykt i form av varmelagringskapasitet til overflatearealet.

$$C = C_p * \rho * t \quad [11]$$

En annen viktig størrelse i denne sammenhengen er temperaturlledningstallet til materialet, også kalt *termisk diffusivitet*. Dette beskriver hvor raskt en temperaturendring forplanter seg i materialet. Høyere termisk diffusivitet, vil medføre til raskere temperaturendring i materialet som igjen betyr hvor raskt materialet lagrer/avgir varmeenergi (Thue, 2016). Dette kan bestemmes ved følgende formel;

$$a = \frac{\lambda}{(C_p * \rho)} \quad [12]$$

Termiske egenskapene til noen av byggematerialene er som følger;

Materiale	Densitet [kg/m ³]	Spesifikk varmekapasitet [Ws/kg*K]	Termisk diffusivitet [m ² /s]
Stål	7800	500	15*10 ⁻⁶
Betong	2300	950	0,8*10 ⁻⁶
Tre	500	2800	0,2*10 ⁻⁶

Tabell 2 Termiske egenskapene til de tre mest brukte byggematerialene. Verdier er hentet fra (Thue, 2016).

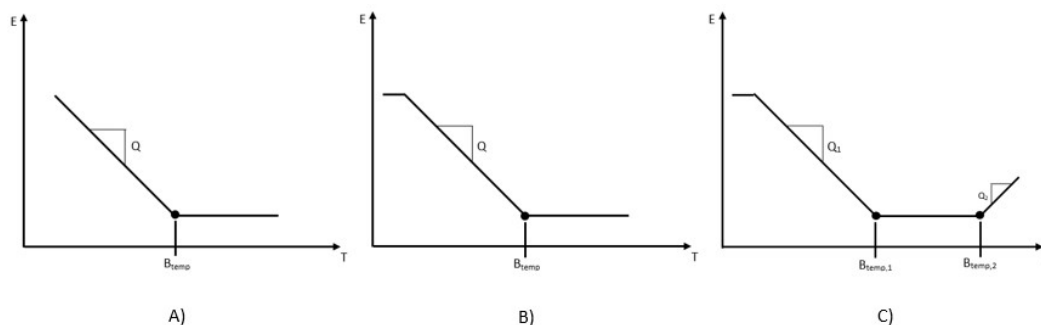
Tabell 3 viser tidsforsinkelsen som kan oppstå over en vegg med 30 cm tykkelse, avhengig av hvilken type materialet.

Material	Tidsforsinkelse [timer]
Soltørket leire	10
Murstein	10
Betong	8

Tabell 3 Tidsforsinkelse for 30 cm tykk vegg med ulike materialer. Kilde (Haase & Andresen, 2007)

2.4 ENERGISIGNATUR

Energisignaturen til et bygg er en slags fingeravtrykk, og kan brukes som et hjelpemiddel i forbindelse med energioppfølging. Dette blir også kalt for byggets ET-kurve, hvor E står for energi og T for temperatur (Graddage, 2019). Kurven skal vise energiforbruket til bygget som funksjon av utetemperatur. Den består hovedsakelig av to deler; stigningsdelen og vannrettedelen. Krysningpunktet for disse delene er såkalt balansetemperatur. Stigningsdelen angir det temperaturavhengige forbruket (Graddage, 2019). Dette forbruket er avhengig av byggets ventilasjon, isoleringsevne, tetthet, brukeratferd, interne varmetilskudd samt plassering mht. solinnstråling. I motsetning til dette er vannrette delen temperaturuavhengig, og skal beskrive energiforbruket til oppvarming av varmtvann. I noen tilfeller vil kurven stige ved høye temperaturer eller flate seg ut ved lave temperaturer. Årsaken til stigningen kan være energiforbruket til romkjøling som kommer ved sommertid. Årsaken til flat-kurve derimot, kan være at oppvarmingssystemet ikke er dimensjonert for ekstremt lave temperaturer eller at det balanserte ventilasjonsanlegget kjøres med utilstrekkelige luftmengder. Se Figur 8 for illustrasjon.



Figur 8 Ulike ET-kurve modeller. A) er tradisjonell, B) er tilfelle med lite oppvarmingskapasitet og C) er tilfelle med energiforbruk til kjøling.

2.4.1 BRUK AV ENERGISIGNATUR

Energisignaturen til et bygg kan være til nytte for å observere hvor mye energi bygget bruker, eventuelt avvik og hvor energieffektiv bygget er (Enova, 2020). Bestemmelse av såkalte kontrollgrenser vil være nødvendig, for å observere eventuelle avvik fra ET-kurven. Disse grensene bør fastsettes for hvert enkelt bygg avhengig av alder og lokasjon (Enova, 2020). Deretter kan slike avvik dokumenteres i form av energibruk [kWh] og antall kroner. Det vil blant annet danne grunnlaget for bestemmelse av tiltak ved rehabilitering/oppgradering av bygningsmassen i et senere tidspunkt.

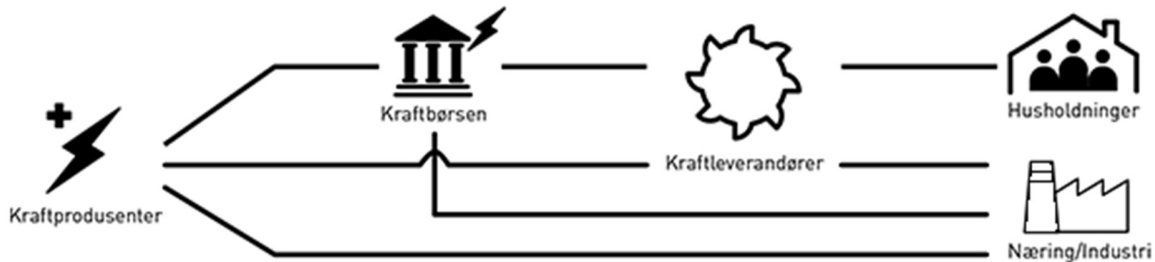


2.4.2 TIDSOPPLØSNING

Energisignaturen kan presenteres i flere tidsoppløsninger; timer, dag, uke, måned og år. Valgt tidsoppløsning kan ha påvirkning på resultatet (Eriksson, Akander, & Moshfegh, 2020). Det sies at energisignaturen for døgnmiddelverdier eller lengre tidsoppløsninger kommer ut med bedre resultat. Dette skyldes av at avvikene mellom individuelle effekter i kortere tidsperioder er blir presentert som en snittverdi over lengre perioder (Fumo & Rafe Biswas, 2020). I motsetning til dette vil en ET-kurve med timesoppløsning være i stand til å bli mer påvirket av værforholdene, noe som vil kreve en mer dynamisk modell for å gi anvendbare resultater (Hammarsten, 2020). I tillegg til dette har det blitt konkludert i en forskning som er utført i Sveits, at det optimale tidsoppløsningen for energisignaturen for et bygg er avhengig av bruksmønsteret til det bygget (Arregi & Garay, 2020). I dette prosjektet har forfatterne valgt å se på tre ulike bygg; Universitetsbygg i UK, Ungdomsskole i Sverige og Sykehus i Spania. Hvert bygg har gjennomgått regresjonsanalyse med tre ulike tidsoppløsninger; timer, daglig og månedlig. Resultatet av deres forskning er at bygninger som er kontinuerlig bruk, som f.eks. sykehus, tillater times/daglige målinger. Mens bygg med som er i diskontinuerlig bruk, bør ha lengre tidsintervaller, for å balansere den forstyrrelses som kan oppstå korrelert med bruksmønsteret i bygget.

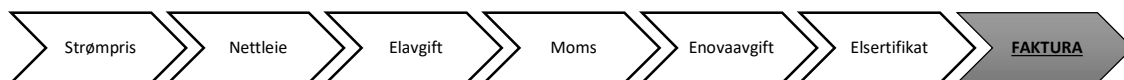
2.5 ENERGIMARKEDET

Energimarkedet i Norge er et markedsbasert kraftsystem. Det er ikke noe form for konkurranse innen nettvirksomhet, siden dette er et naturlig monopol. Alt kjøp og salg av kraft i Norge, foregår på Nord Pool Spot. Dette er kraftbørsen hvor mesteparten av krafthandelen i Norge foregår, og det er her såkalte strømprisen fastsettes. Legg merke til at dette kun er en systempris, eventuelle kostnader som nettleie, MVA, elavgift, osv. kommer i tillegg. Organisering av kraftmarkedet i Norge deles hovedsakelig i 2; engrosmarkedet og sluttbrukermarkedet. Engrosmarkedet er markedet hvor store volumer av kraft kjøpes og selges mellom meglere, nettleverandører eller direkte mellom store forbrukere som f.eks. industri. Det vanligste er å gå via en nettleverandør, men dersom du er en storforbruker (f.eks. industri), så kan du inngå såkalte bilaterale avtaler med kraftprodusentene om kjøp og salg av kraft til fast avtalt pris (Kraftmarkedet, 2020). Sluttbrukermarkedet er der hvor de ulike kraftleverandørene kjøper kraft fra kraftprodusenten, på vegne av deres kunder som i dette tilfellet kan være privatpersoner og/eller mellomstore kunder (små bedrifter). Organiseringen av kraftmarkedet er oppsummert i Figur 9.



Figur 9 Organisering av kraftmarkedet i Norge. Kilde: (Kraftmarkedet, 2020)

Den totale strømrregningen som de ulike privatpersonene og mellomstore kundene mottar består naturligvis av flere ledd (Strømpris og Kraftmarkedet, 2020). Strømprisen er den «spotprisen» som bestemmes på kraftbørsen, resterende ledd er tillegg som kommer fra regjeringen og kraftleverandørene, se Figur 10 for illustrasjon.



Figur 10 Alle deler som strømrregningen består av.



2.5.1 ENERGINETT

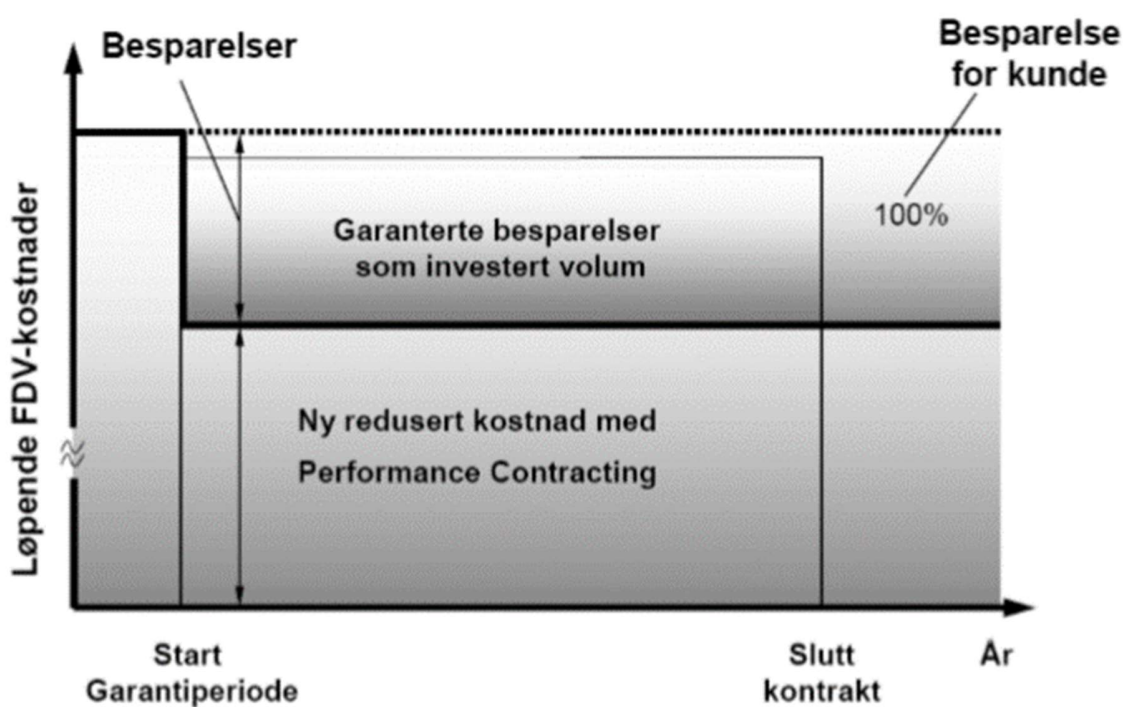
Energinett er et web-basert energioppfølgingssystem. Dette er et system hvor data blir innsendt automatisk fra blant annet kraftselskaper, nettselskaper eller SD-anlegg. Videre får brukeren/organisasjonen muligheten til driftsoppfølging av bygget eller bygningene opp mot hverandre og ha full kontroll over forbruket og eventuelt avvik underveis. I tillegg til energi- og avfallsadministrasjon kan systemet utarbeide enkle rapporter med enkelt brukergrensesnitt for observasjon av variasjoner (Energinett, 2020).

2.5.2 AMS – MÅLERE

AMS-målere er automatiske strømmålere som skal lese inn og sende målerverdier automatisk til Elhub. Elhub er en slags database (mellomstasjon), før målingene blir videresendt til strømleverandøren (Smarte Strømmålere, 2018). Disse målerne blir også kalt for smarte strømmålere. På grunn av geografiske forhold har Norge mulighet til å produsere mye strøm fra vannkraftverkene, men ulempen er arealene dette må distribueres over, spesielt i «rushtiden». Ujevn belastning og økt strømforbruk fra norske husholdninger kan ha negativ påvirkning på nettkapasiteten. Men forbedres ved installering av AMS-målere, fordi de bidrar nettopp til en smartere og fleksibelt energisystem. Slik fleksibel system kan sørge for at forbrukstoppene forflyttes til en tid hvor strømprisen er lavere og belastning på nettet er mindre, (AMS-Målere, 2020).

2.5.3 EPC – «ENERGY PERFORMANCE CONTRACTING»

EPC – Energisparekontrakt er en anerkjent internasjonal modell for garantert og kostnadseffektiv reduksjon av energiforbruket til eksisterende bygningsmasser (Husbanken, 2020). Kontrakten inngås mellom en byggeier, og et foretak som skal tilby energitjenester. Energитjenesteforetaket skal påta seg hele ansvaret for prosjektet; fra analyse- tom. driftsfasen. I løpet av denne avtaleperioden skal foretaket mota det innsparte beløpe. Samtidig vil dette foretaket sitte på risikoen for manglende måloppnåelse. Dersom de ikke oppnår målene eller kontraktsfestet innsparing ikke er oppnådd, plikter energitjenesteforetaket å betale differansen til byggeieren. Derfor vil energitjenesteforetaket følge forbruket nøye ettersom tiltakene utføres fortløpende. Dette for å observere om det fortsatt er potensiale for å oppnå kontraktsfestet besparelse. Etter kontraktperioden vil byggeieren sitte igjen med reduserte FDV-kostnader, noe som kan motivere til nye investering innen energibesparelse. Figur 11 er en illustrasjon for hvordan dagens EPC-prosjekter i Norge bør gjennomføres.



Figur 11 Prinsippkisse for EPC-modell. Kilde: (Husbanken, 2020)



3 METODE

Kvantitative rådataene i dette prosjektet tilhører bygningsmassene fra Boligbygg. Det er totalt 99 bygninger som blir behandlet. Dette kapitlet tar for seg forskningsforløpet og metoden som ble benyttet i databehandlingen, analyse og utvikling av prediksjonsmodell.

3.1 DATABEHANDLING

Før nærmere analyse av bygningene, ble alle dataene kvalitetssikret gjennom en databehandling. Målet med databehandlingen var å ende opp med et datasett med god kvalitet og kontinuitet.

3.1.1 NEDLASTING AV DATA

I første steg i databehandlingen ble det utført kun en enkel observasjon av tilgjengelig rådata i energinett. Gjennom dette ble det identifisert at dataene kommer fra 4 ulike målere i tillegg til utetemperaturen som var lik for alle bygningene. Disse målerne er «EL leiligheter», «EL fellesareal», «Fjernvarme» og «Tappevann VP». Det ble lagt merke til at de ulike målingene startet fra forskjellige tidspunkter. Og dette var også ulikt for noen av bygningene. I tillegg var det kun noen bygninger som hadde data fra alle målingene. Det ble laget en oversikt over alle bygningene for å kartlegge når de ulike målerne var installert og hvilke målere bygningen hadde data fra. Heldigvis var det noe kontinuitet i dette, og det viste seg at de fleste bygninger hadde data fra «EL leiligheter» og «EL fellesareal» for hele 2019. Totalt 9 bygninger hadde data fra «Tappevann VP», men disse målingene startet ulike tider på høsten 2019. Dette ble konkludert som ikke tilstrekkelig nok data, og det ble derfor sett bort i fra disse målingene. Det ble også observert en del bygninger som hadde målere «VP ikke ID klar». Dette tydet på at mange bygningene var i ferd med å få varmepumpe installert i slutten av 2019. Ikke alle bygninger hadde data fra «Fjernvarme»-målingene, men en god del hadde det og alle for hele 2019. Det ble derfor konkludert med å ta med målinger fra «EL leiligheter», «EL fellesareal» og «Fjernvarme» for alle bygninger.

Etter at det ble bestemt hvilke data som skulle brukes, ble dataene fra hvert enkelt bygg filtrert og nedlastet i CSV-format med timesoppløsning. Det ble bestemt at hver CSV-fil skulle ha samme oppsett slik at de skulle bli enklere å jobbe med i Python. Det vil si at hver fil skal inneholde totalt sett 5 kolonner; Tid(Time), EL leiligheter, EL fellesareal, Fjernvarme og Utetemperatur, med 8760 data for hver kolonne. De bygningene som ikke hadde data i hver av disse kolonnen, beholdte kolonnen, men uten verdi. I tillegg ble det laget egne fil med informasjon over bygningene som BRA, energikarakter, byggeår osv. For å kontrollere at nedsalting av data ble riktig og for å utføre enkelt analyser i Python, valgte gruppen å nedlaste totalt 12 bygninger i starten av oppgaven. Det ble plukket 3 bygg fra hver bydel i Oslo; Grorud, Indre Øst, Sentrum og Sør-Vest.

Da dataene for de 12 bygningene var ferdig nedlastet var målet å laste de inn i Python. Det genereres en stor tabell med alle bygningene i for enkelt å kunne arbeide med. Tabellen skulle inneholde data for alle målingene, men også tilleggsinformasjonen om bygningene fra egen fil. Dette kom til å bli en ganske stor



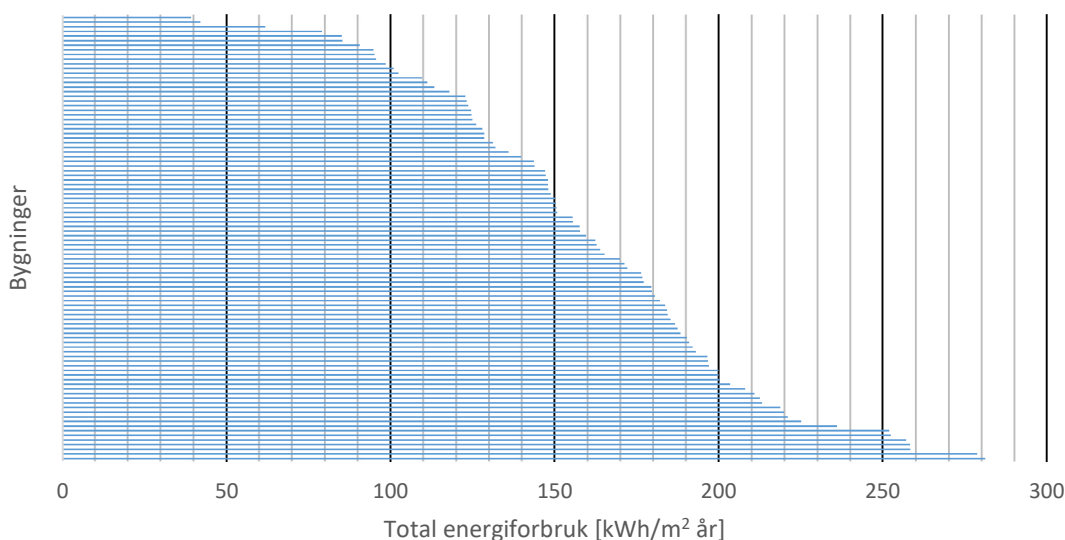
tabell med underkant av 10 millioner data når alle bygningene er med. Til å håndtere en slik tabell ble Pandas benyttet, som er et stort bibliotek i Python utviklet nettopp til å håndtere store datamengder i såkalte datarammer. Pandas inneholder også mange funksjoner som har blitt benyttet til blant annet å laste inn data, generere nye filer og plotting av grafer. Da alt dette var testet ut nærmere og koden var delvis på plass, ble alle de resterende bygningene lastet ned med tilsvarende format.

Til å lese filene og laste inn alle dataene i Python ble Pandas-funksjonen «read_csv» benyttet. Dataene blir da lagret i egne datarammer. Dette ble gjort for alle filene inklusive filen med informasjon over bygningene. Alle datarammene ble lagt under hverandre i en stor dataramme utenom den med informasjon. Den ble, ved hjelp av funksjonen «merge», slått sammen til slutt slik at riktig informasjon havnet til riktig bygning. I tillegg ble det generert to nye kolonner; «Forbruk» og «Forbruk.m2». Den første er summen av alle målerne, altså energiforbruket til bygningen oppgitt i kWh. Den andre deler dette på bygningens BRA og ganger med tusen for å oppgi i W/m². For nærmere beskrivelse av hvordan dette er utført i Python, se vedlegg A.1.

3.1.2 FØRSTE GJENNOMGANG

Da alle dataene var ferdig nedlastet og datarammen var generert i Python, var neste steg og utføre en første gjennomgang av bygningene. Målet med denne gjennomgangen var å få en oversikt og en forståelse av dataene, og fastslå eventuelle feil og mangler. For beskrivelse av koden som ble generert i denne gjennomgangen, se vedlegg A.2.

Først ble det totale årlig energiforbruk pr. kvadratmeter for alle bygninger sammenlignet. Dette er vist i Figur 12. Her ser man en veldig stor variasjon, mye større enn som skulle være forventet. Ifølge det tekniske forskriftet (Byggeteknisk forskrift, 2020) skal boligblokk ikke ha et netto energibehov som overstiger 95 kWh/m² år. Dette gir en indikasjon på hvor bygningene vist i Figur 12 bør ligge. Men siden dette er data over forbruk, i tillegg til at ingen av bygningene er bygget etter TEK17, forventes det ikke at noen bygninger er innenfor dette kravet. Så det at noen bygninger har årlig forbruk på under 100, og til og med under 50 kWh/m² år, tyder på store feil i dataene. På samme måte er det en stor andel bygninger som ligger over 200 kWh/m² år, som er i meste laget av det som skulle forventes.

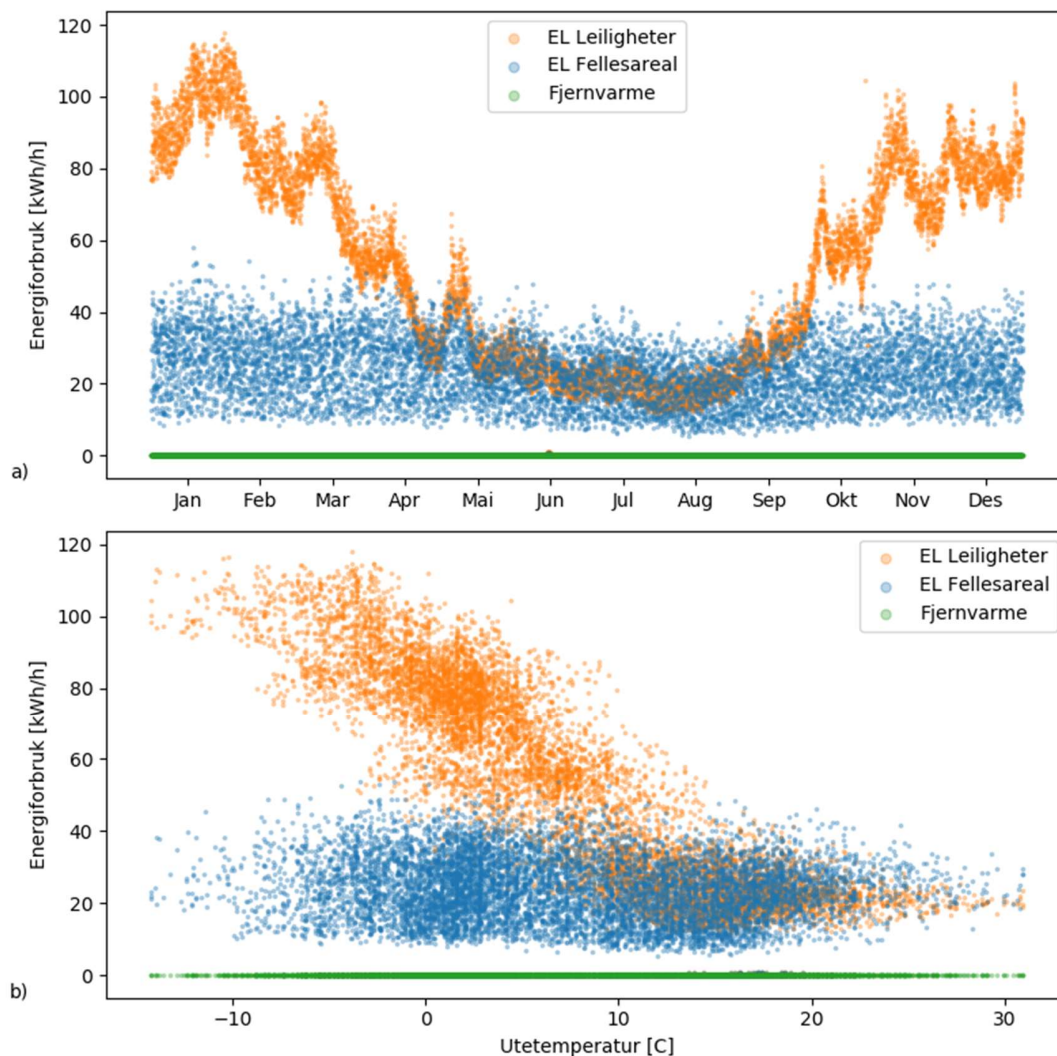


Figur 12 Årlig energiforbruk pr. kvm. for alle bygg (99 bygg)

For å undersøke dette nærmere og kvalitetssikre dataene, ble alle målingene av forbruket nøye gjennomgått. Dette ble gjort ved å generere spredningsplott for alle bygninger, som vist i Figur 13 på neste side. På denne måten var det enkelt å oppdage feil eller mangler i dataene, som medførte at en del bygninger ble utelatt fra videre forskning. Allikevel var det mange bygninger som hadde data som virket fornuftige, men som kom ut med uforklarlige verdier av årlig forbruk pr. kvadratmeter. Det ble derfor i tillegg gjennomgått ENØK-analyser for alle bygninger. Dette er energi-økonomi analyser for hver individuell bygning som gir informasjon om blant annet årlig forbruk, BRA, varmesystem osv. Disse var til stor hjelp for å forklare målinger som tilsynelatende virket fornuftige. Det ble f.eks. observert en del arealer som ikke samsvarte med forbruket, og ved etterforskning av disse oppsto det stor usikkerhet rundt hvilke areal som var gjeldende. Flere bygninger ble utelatt på bakgrunn av dette.

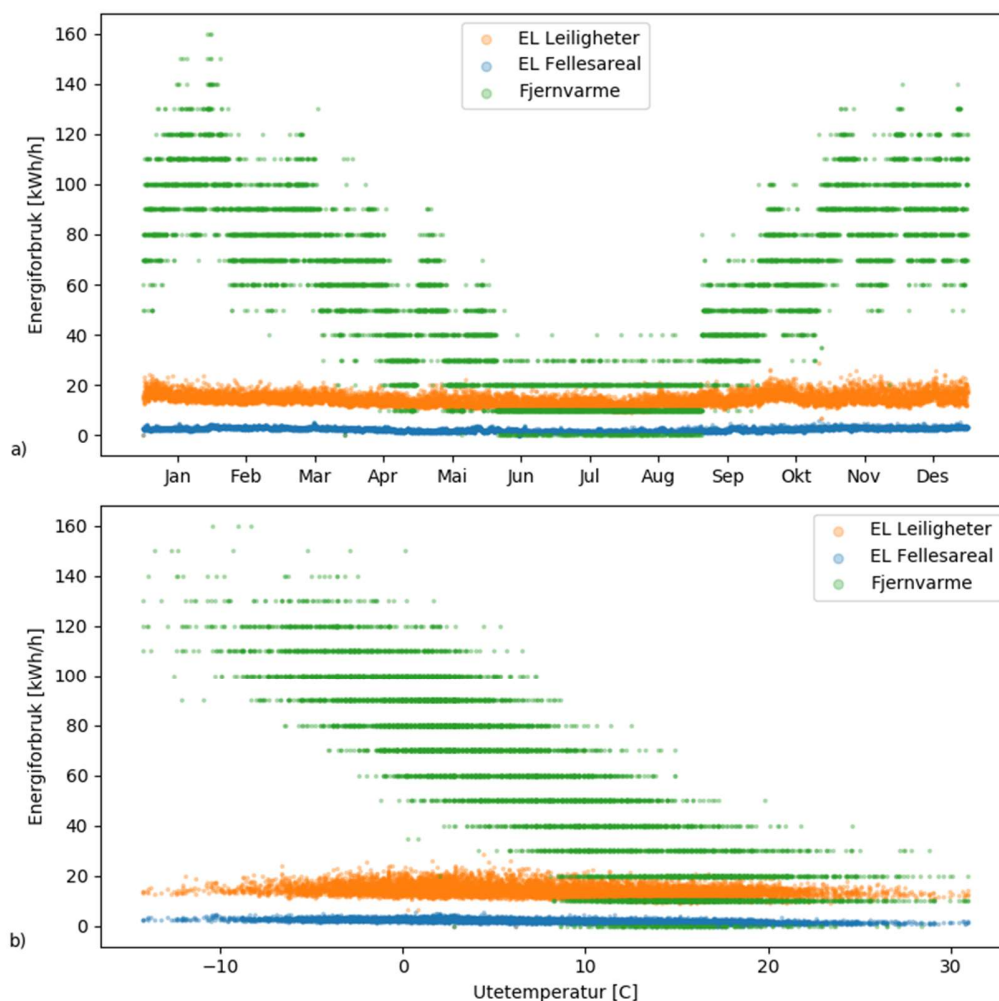


Figur 13 viser to spredningsplott av en tilfeldig valgt bygning, hvor a) viser hvordan forbruket varierer gjennom året, mens b) viser forbruket som en funksjon av utetemperaturen. I tillegg til å observere feil og mangler i dataene, kan man også få en bedre forståelse av hva de ulike dataene inneholder. Selv om dette er en tilfeldig bygning hadde de fleste bygninger, som ikke ble utelatt, tilsvarende grafer. Dette er bygninger som dekker sitt varmebehov med elektrisitet. Det er kun målingene for «EL leiligheter» som varierer gjennom året, med høye verdier om vinteren og lave om sommeren. Det tyder på at oppvarmingen for bygningene ligger i disse målingene. Det kommer også frem av Figur 13 b) når målingene øker ved avtagende utetemperatur. Disse målingene er da temperaturavhengige, selv om det kan se ut som de flater ut ved høyere utetemperaturer. Utflatingen ved høyere utetemperaturer kan indikere en type grunnlast i forbruket som er temperaturuavhengig, bestående typisk av komfyr, kjøleskap, belysning og annet teknisk utstyr. Videre har det blitt informert, fra ekstern veileder, om at «EL fellesareal» inneholder varmtvann. Det kommer frem av figuren at dette forbruket er konstant gjennom året og ikke avhengig av utetemperaturen.



Figur 13 Bygninger som dekker varmebehovet med elektrisitet, hvor a) viser hvordan forbruket varierer gjennom året, mens b) viser forbruket som en funksjon av utetemperaturen

Figur 14 viser samme figur som tidligere, men av en tilfeldig valgt bygning av de som har målinger for «Fjernvarme». Disse bygningene har da et annet type varmesystem. Her blir varmebehovet dekket av fjernvarme via vannbårent varmesystem, som betyr både romoppvarming og varmtvann. Dette fremkommer også av figuren ved at mesteparten av forbruket ligger i «Fjernvarme» målingene. Det er også bare disse målingene som varierer med utetemperatur. Målingene for «EL leiligheter» og «EL fellesareal» er her jevne gjennom hele året og temperaturuavhengige, bestående typisk av teknisk utstyr som nevnt tidligere. En annen observasjon verdt å poengtere er at målingene for fjernvarme viste seg for det meste å ligge i faste intervaller på 10 kWh/h. Dette vil redusere nøyaktigheten til dataene.



Figur 14 Bygninger som dekker varmebehovet med fjernvarme, hvor a) viser hvordan forbruket varierer gjennom året, mens b) viser forbruket som en funksjon av utetemperatur

Det ble også observert noen bygninger med andre type varmesystemer, som f.eks. bygninger med elektrokjeler kombinert med et vannbårent system. Disse bygningene fikk da romoppvarmingen i «EL fellesareal» målingene. Alle disse bygningene med andre varmesystemer ble utelatt fra videre analyse. Grunnen til at «Fjernvarme» målingene ble tatt med videre var fordi en stor andel av bygningene var utstyrt med Fjernvarme.



3.2 ANALYSE

Målet med analysen var å få et bilde av hvilken tilstand boligblokkene er i med tanke på energibehov. Analysemetoden som ble benyttet er basert på ET-kurver som på ulike måter kan bidra i å estimere blant annet bygningens oppvarmingsbehov. Det ble også forsøkt å estimere bygningenes oppvarmingstid og varmtvannsbehov. For å sammenligne bygningene ble forbruket dividert på bygningenes BRA.

I denne analysen er dataene over energiforbruket benyttet til å estimere ulike energibehov. Det er da viktig å skille mellom begreper som energiforbruk og energibehov. Selv om disse er tett knyttet, er det ulikheter ved de som er viktig å poengtere, og som bidrar til at estimatene i denne analysen kun er grove estimater. Når det er snakk om energibehov, er f.eks. systemvirkningsgrad ikke tatt med i beregningen. Fjernvarmesystem har f.eks. lavere systemvirkningsgrad enn i panelovner som vil øke forbruket selv om varmebehovet er det samme. Systemvirkningsgraden kan også gi positiv effekt. En varmpumpe eller et solcellepanel vil kunne bidra til redusert energiforbruk selv om varmebehovet forblir uendret.

3.2.1 ET-KURVER

Målet var å tilpasse ET-kurver for alle bygningene og hente ut verdier for grunnlast, balansetemperatur og stigningstall. I tillegg var det ønskelig å hente ut verdier for R^2 og RMSE slik at modellen senere kunne sammenlignes med andre regresjonsmodeller. Gruppen fikk hjelp av internveileder som utviklet i Python en regresjonsalgoritme kalt «ETRegressor» med dette formålet, se vedlegg B.1. For at algoritmen senere kan sammenlignes med andre maskinlæringsalgoritmer fra Sci-kit Learn biblioteket, ble den programmert med tilsvarende struktur. Den ble derfor også kompatibel med andre funksjoner fra biblioteket.

Formålet med ET-kurven er å skille ut målerdata som er temperaturavhengige, og tilpasse en lineær kurve kun til disse datapunktene. Sagt på en annen måte er dette datapunkter som inneholder energiforbruk til oppvarming. Denne linjen beskriver da korrelasjonen mellom energiforbruket og utetemperaturen og får et stigningstall som er negativ. Dette er ulikt fra en kontinuerlig lineær kurve, som benyttet alle punktene. Den resterende temperatu-uavhengige delen må dermed beskrives på en annen måte enn med utetemperaturen. Det beregnes et gjennomsnitt av disse punktene, også trekkes en linje med denne konstante verdien. Dette tilsvarer da grunnlasten til bygget, og er bygningens energiforbruk til blant annet varmtvann og teknisk utstyr. Krysningpunktet mellom disse linjene er balansetemperaturen som beskriver ved hvilken utetemperatur bygningen starter oppvarming.

«ETRegressor»-klassen som skapes har derfor tre parametere (startverdier); grunnlast (e_0), balansetemperatur (T_t) og stigningstall (s). Siden disse uansett justeres når kurven tilpasses, gjør det ikke noe om de er konstante for de ulike bygningene. For å tilpasse kurvene brukes en funksjon i Python kalt «curve_fit» fra biblioteket «scipy.optimize». Denne funksjonen benytter regresjon til å tilpasse kurver. Når koden kjører tilpasses det da to linjer til datasettet og parameterne blir samtidig justert. Linjene får følgende formel:

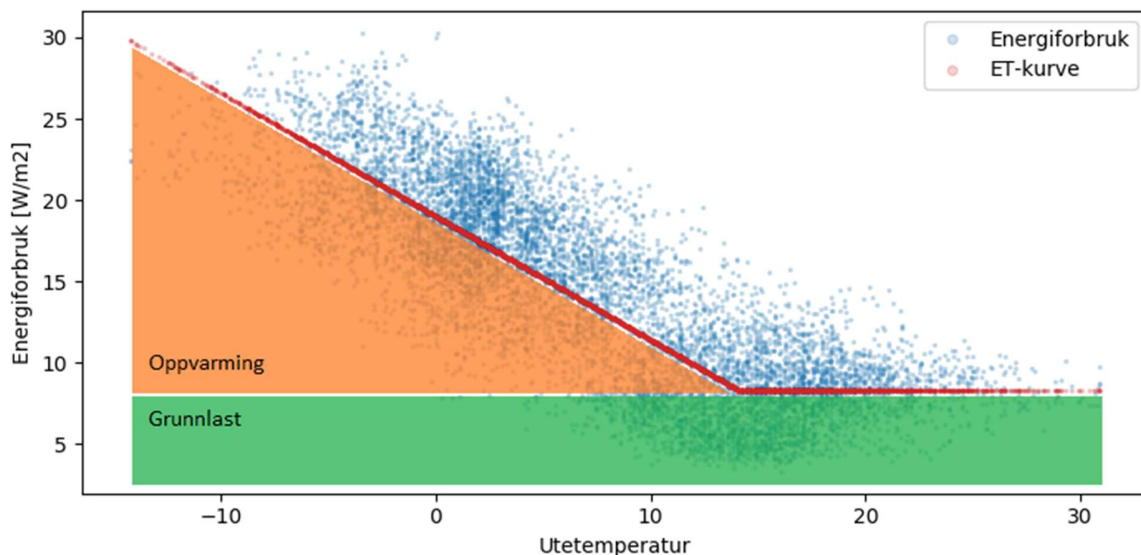
- En lineær kurve for temperaturer mindre enn balansetemperaturen [$T_e < T_t$] med funksjonen;

$$y = s * (T_e - T_t) + e_0 \quad [13]$$

- En konstant kurve for temperaturer over balansetemperaturen [$T_e > T_t$];

$$y = e_0 \quad [14]$$

Etter at parameterne er blitt justert til bygningens målinger kan de benyttes til å estimere oppvarming. Figur 15 beskriver hvordan dette gjøres grafisk. Først må alle datapunkter som er temperaturavhengige defineres, altså de dataene som inneholder oppvarming. Dette kan gjøres på ulike måter, men i denne estimeringen er alle datapunkter som er til venstre og over balansetemperaturen definert som temperaturavhengige. Antallet på disse datapunktene kan gi et estimat på hvor mange timer i året bygningen har oppvarming. Siden grunnlasten defineres som en konstant verdi hver time gjennom hele året kan denne subtraheres fra de temperaturavhengige datapunktene. Da er summen av disse et estimat av oppvarmingen. For nærmere beskrivelse av hvordan beregning er utført i Python, se vedlegg B.2.

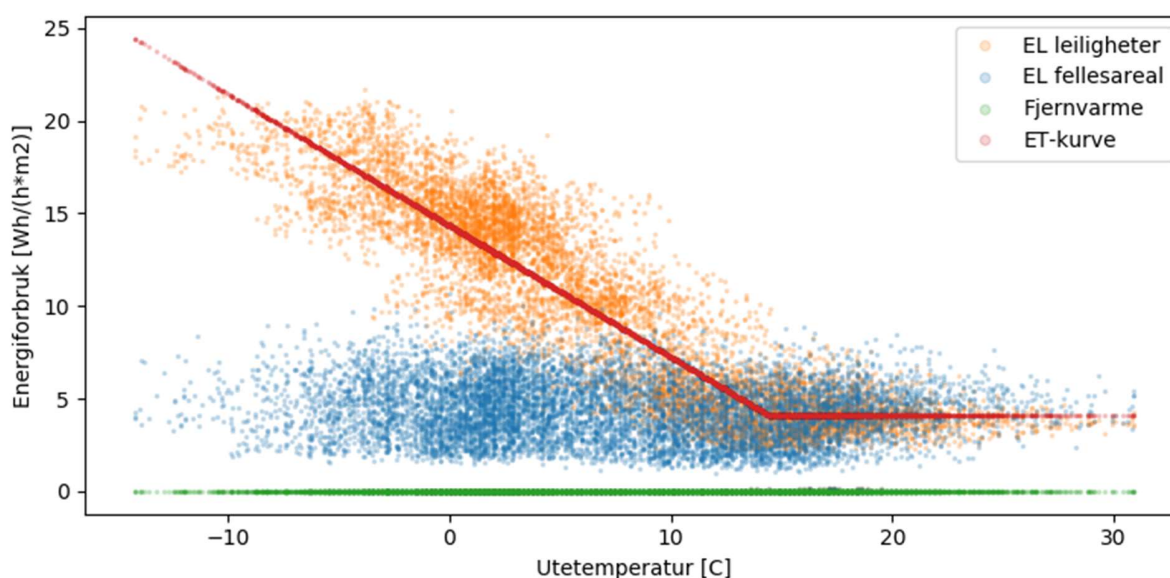


Figur 15 Illustrasjon av hvordan estimeringene er foretatt

3.2.2 ESTIMERING AV OPPVARMING, VARMTVANN OG ANNET

Siden det var blitt oppgitt i grove trekk hva de ulike målingene inneholder, kunne disse i kombinasjon med ET-kurver bidra i å estimere oppvarming, varmtvann og annet. Kategorien «annet» omhandler alt strømforbruk som ikke er en del av varmebehovet. I denne estimeringen var det behov for å skille mellom bygninger som bruker elektrisitet til å dekke varmebehovet og bygninger som bruker fjernvarme. Grunnen til dette er fordi beregningsmetoden blir forskjellige når målingene inneholder ulike deler av energiforbruket. Det var stor usikkerhet over hvilke arealer de ulike målingene skulle fordeles over. For eksempel varmtvannsforbruket ligger i målingene for fellesarealet, men forbruket avhenger jo antall leiligheter og arealet av disse. På bakgrunn av dette ble det bestemt at i denne estimeringen skulle alle målingene deles på bygningenes BRA. Selv om dette fører til upresise estimater, var dette den enkleste løsningen for å danne et felles sammenligningsgrunnlag for alle bygninger.

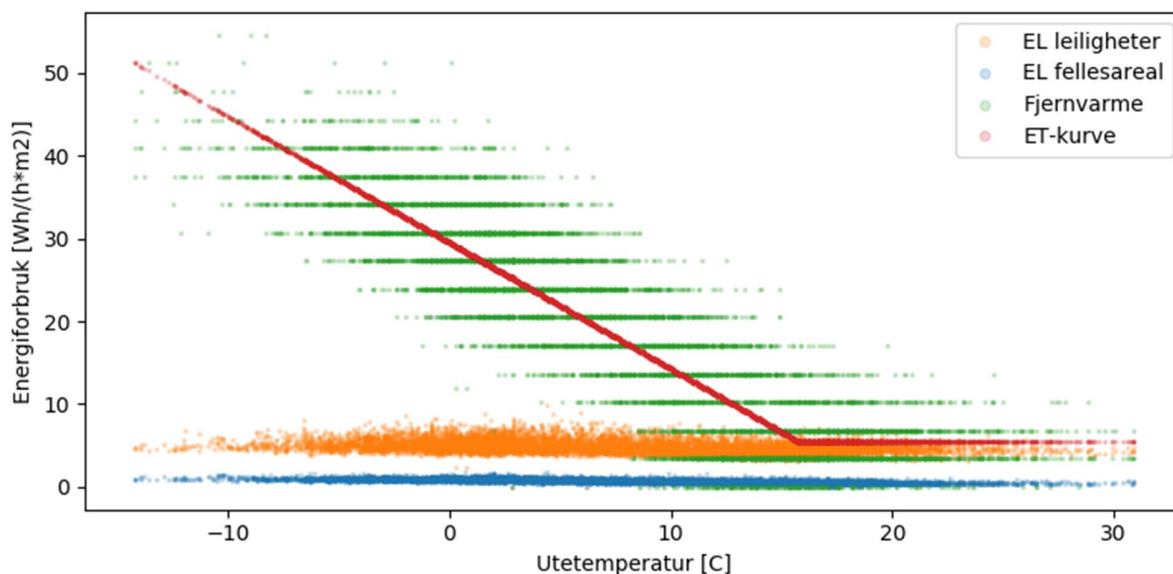
Av Figur 16 er det tydelig at for bygninger som bruker elektrisitet til å dekke varmebehovet ligger oppvarmingen i målingene for «EL leiligheter». Dette varmebehovet dekkes da typisk av panelovner. Ved å benytte en ET-kurve på disse målingene kan man skille oppvarmingen fra grunnlasten som vist i Figur 15. Det forutsettes at all oppvarming ligger i disse målingene. Her blir da grunnlasten det forbruket i leilighetene som ikke er temperaturavhengig. Dette havner da under kategorien «annet». Det er blitt oppgitt at «EL fellesareal» målingene inneholder varmtvann til boligblokken. Summen av disse målingene kan da sees på som et estimat av det årlige varmtvannbehovet til bygningen. Av Figur 16 kan man se at disse målingene er temperaturuavhengige som betyr at det brukes like mye varmtvann hele året. Selv om en liten del av «EL fellesareal» målingene går til lys og annet teknisk utstyr, er det ikke noe enkel måte å skille ut dette på. Det forutsettes derfor at mesteparten av «EL fellesareal» målingene er varmtvann. Eventuell oppvarming av fellesareal blir da definert som varmtvann. For nærmere beskrivelse av hvordan beregning er utført i Python, se vedlegg B.3.



Figur 16 Alle målerne for bygg som bruker elektrisitet til å dekke varmebehovet

For bygninger som dekker varmebehovet med fjernvarme ligger både oppvarmingen og varmtvann i «Fjernvarme» målingene. Mesteparten av energiforbruket ligger da i disse målingene. Det blir derfor benyttet ET-kurver på «Fjernvarme» målingene for å skille ut oppvarmingen, se Figur 17. Den flate, temperatu-uavhengige delen av ET-kurven blir i dette tilfellet da et estimat for varmtvann. Dette er i teorien et bedre estimat av varmtvannsbehovet enn for bygg hvor varmtvannet ligger i «EL fellesareal» målingene, siden det her ikke er noen elektrisk forbruk som forstyrrer dataene. På en annen side har inndelingen av datapunktene i oppvarming og varmtvann blitt mer utfordrende siden datapunktene ligger i intervaller på 10 kWh. Dette kan da medføre til at inndelingen har blitt grovere.

Siden det forutsettes at all oppvarming og varmtvann for bygninger med fjernvarme ligger i «Fjernvarme» målingene, kan summen av de resterende målingene for «EL leiligheter» og «EL fellesareal» kategoriseres som «annet». Eventuell tillegg oppvarming av leiligheter eller fellesareal havner da i denne kategorien. For nærmere beskrivelse av hvordan beregning er utført i Python, se vedlegg B.4.



Figur 17 Alle målerne for bygg som bruker fjernvarme til å dekke varmebehovet



3.2.3 REGELVERKET – TEK17 OG NS3031

Estimatene sammenlignes med kravet i teknisk forskrift §14-21(TEK17) som omhandler krav til energieffektivitet i de ulike bygningskategoriene. I denne oppgaven benyttes boligblokk som bygningskategori. Disse bygningene skal ikke ha et total netto energibehov som overstiger 95 kWh/m² år, i henhold til TEK17. I denne oppgaven benyttes kun målt energiforbruk fra de ulike bygningene, for sammenligning opp mot dette kravet. Forskjellen mellom energiforbruk og energibehov, er nærmere beskrevet i kapittel 3.2. Det forutsettes at målt energiforbruk tilsvarer netto energibehov. Estimatene fra de ulike bygningene blir kontrollert opp mot de forventede estimatene for nye boligblokker i henhold til TEK17 og NS3031. Dette er for å danne et sammenligningsgrunnlag som skal være til hjelp for å observere energiytelsen til de ulike bygningene.

I NS3031 tabell A.1 oppgis forventet energibehov for henholdsvis varmtvann, teknisk utstyr og belysning. Summen av teknisk utstyr og belysning blir betegnet som annet. Sett i forhold til kravet om total netto energibehov i boligblokker kan en prosentsats av både annet, varmtvann og oppvarming bestemmes. Dette skal beskrive den forventede fordelingen av total netto energibehov i henhold til TEK17.

$$\text{Annet} = \frac{40}{95} [\text{kWh}/\text{m}^2] = 41\%$$

$$\text{Varmtvann} = \frac{30}{95} [\text{kWh}/\text{m}^2] = 32\%$$

[15]

$$\text{Oppvarming} = 95 [\text{kWh}/\text{m}^2] - \text{Annet} - \text{Varmtvann} = 27\%$$



3.3 PREDIKSJONSMODELLEN

I prediksjonsanalyse er det ofte snakk om en modell som basert på tidligere erfaringer skal beregne hvilke fremtidige forhold som mest sannsynlig vil inntreffe. Når man skal ta i bruk maskinlæring og mer avanserte modeller i forsøk på å predikere, kan det være vanskelig å vite hvor man skal begynne. Å definere hvilke problem man prøver å løse er alltid lurt. I denne forskningen er det tilgang på mye data fra tidligere energiforbruk. Det er da lurt å benytte seg av denne dataen til å trene en modell til å predikere fremtidige energiforbruk. Det er da snakk om veiledende læring. Neste steg er å se på hvordan resultatene vil se ut. I denne rapporten forsøkes det å predikere en kontinuerlig verdi av fremtidig energiforbruk. Dette er da et regresjonsproblem. Det er fortsatt mange ulike maskinlæringsalgoritmer som kan benyttes i en slik problemstilling. Denne oppgaven tar for seg noen av de mer kjente, som multippel lineær regresjon, beslutningstrær og «Random Forest Regression».

For å få en ide om hvor presis prediksjonsmodellen er sammenlignes den med tradisjonelle ET-kurver. Det er i hovedsak to elementer som er med på å avgjøre kvaliteten til prediksjonsmodellen; hvor godt variablene beskriver variasjonene i energiforbruket, og kompleksiteten til algoritmen. Multippel lineær regresjon er den enkleste maskinlæringsalgoritmen og baserer seg på samme algoritme som ET-kurver. Ved å benytte denne i prediksjonsmodellen, kan virkningen av flere variabler belyses nærmere. Videre testes andre maskinlæringsalgoritmer, med de samme variablene. På den måten kommer virkningene av dem frem.

3.3.1 VARIABLER

ET-kurver kan brukes til prediksjon. Den gir ikke særlig nøyaktig prediksjon for timesverdier (Fumo & Rafe Biswas, 2020), men dersom man ser på døgnverdier kan det gi et greit estimat på det daglige energiforbruket. For å kunne predikere mer nøyaktig enn klassiske ET-kurver, ble det gjort en grundigere forskning på korrelasjonen mellom energiforbruket og utetemperaturen. Dette er en verdi som beskriver den lineære sammenhengen mellom to variabler. Det er viktig å forstå hva hovedproblemet med ET-kurven er og hvor svakheten ligger for å kunne skape en bedre prediksjonsmodell. Når man skal predikere energiforbruk er det i hovedsak to mønstre man forsøker å beskrive. Det første og viktigste mønsteret er den store svingningen som skapes av variasjonene over året. Det andre mønsteret er den litt mindre svingningen som skapes av variasjonene over døgnet. I denne forskningen er det forsøkt å skille mellom disse mønstrene for å se hvor bra utetemperaturen klarer å beskrive disse og eventuelt om det finnes andre variabler som beskriver de bedre.



I denne analysen legges det til 5 nye uavhengig variabler i tillegg til utetemperaturen. Fire av disse er cosinus- og sinusfunksjoner som skal bidra i å beskrive enten variasjonen over året, eller variasjonen over døgnet. Dette er ulikt fra utetemperaturen som forsøker å beskrive begge disse mønstrene samtidig. Den siste er en variabel som enten er 1 eller 0, avhengig av om det er helg eller hverdag. Dette er for å beskrive eventuelle variasjonsmønstre knyttet til energiforbruk i ukedagene. Variablene er som følger;

- Utetemperatur
- Cos av året
- Sin av året
- Cos av dagen
- Sin av dagen
- Helg

I Python er funksjonen «corrcoef» fra biblioteket «Numpy» brukt for å beregne korrelasjonene. Hvordan dette gjøres beskrives nærmere i vedlegg C.1 og C.2. Det er ønskelig å se hvor god korrelasjon det er mellom variablene og de ulike variasjonsmønstrene i energiforbruket. For å få til dette er det forsøkt å skille ut disse variasjonene i dataene. Variablene må korreleres mot energiforbruket for ulike tidsoppløsninger og verdier av dataene. Disse er som følger;

- **Timesverdier:** Her er alle timesverdiene for hele året slik dataene er opprinnelig. Disse dataene inneholder da alle variasjoner i energiforbruket og det snakkes da om den totale variasjonen. Ved å korrelere variablene mot energiforbruket for disse verdiene, forsøker da variablene å beskrive både variasjonen over året og døgnet samtidig.
- **Døgnverdier:** Her summeres alle verdiene over døgnet. I denne prosessen fjernes da variasjonen over døgnet, og man sitter igjen med en beskrivelse av variasjonen over året. For disse verdiene forsøker da variablene å beskrive variasjonen over året.
- **Døgnvariasjon:** Her er formålet å beskrive kun variasjonen over døgnet. For å skille ut denne variasjonen fra dataene, må variasjonen over året (døgnverdier) subtraheres fra den totale variasjonen. Døgnverdier dividert på 24 (timer i døgnet) er gjennomsnittlig timesverdier for døgnet. Ved å subtrahere denne verdien fra alle tilhørende timesverdier, fjerner man variasjonen over året. Man sitter da igjen med et datasett som kun beskriver variasjonen over døgnet, for hele året. Ved å korrelere energiforbruket med variablene for disse verdiene, får man en om hvor godt variablene beskriver variasjonen i energiforbruket over døgnet.



3.3.2 MASKINLÆRING I PYTHON

I denne oppgaven benyttes Python til å programmere prediksjonsmodellen. Denne brukes til å predikere energiforbruket basert på inputvariabler som blant annet utetemperatur, cosinusfunksjoner og sinusfunksjoner. I Python benyttes biblioteket Sci-kit Learn til å lage disse modellene. Dette biblioteket inneholder ferdig funksjoner for de ulike maskinlæringsalgoritmene. En av fordelene med Sci-kit Learn er at alle algoritmene er konstruert på samme måte. Dette gjør det enkelt å bytte ut og prøve ulike algoritmer i modellen med de samme variablene og dataene for energiforbruket. Først ble multipl lineær regresjon benyttet siden dette er den enkleste algoritmen, men også i utviklingen, for å få modellen til å virke. Deretter ble andre algoritmer testet ut. I dette kapitlet beskriver tankegangen bak utvikling av prediksjonsmodellen og eksempler av hvordan dette kan gjøres i Python. Hele skriptet er lagt til som vedlegg C.3 – C.7.

Etter at all data er systematisert i Pandas datarammer (df), kan variablene legges til som vist i Figur 18. Hver kolonne er da en variabel med timesverdier. Når alle variablene er lagt til datarammen, genereres en ny dataramme med døgnerverdier (df_d), altså summen av timesverdiene gjennom døgnet. Dette er for å ha muligheten til å lage en egen modell som benytter døgnerverdier av variablene. Her vil det da ikke gi noe mening å ha med variabler som beskriver døgnervariasjonen.

```
#dataframe
#####
df = dfm.loc[adr, ['Forbruk.m2', 'Utetemperatur']]
df['Cos av året'] = np.cos(2*np.pi*(df.index.dayofyear - 15)/365)
df['Sin av året'] = np.sin(2*np.pi*(df.index.dayofyear - 0)/365)
df['Cos av dagen'] = np.cos(2*np.pi*df.index.hour/24)
df['Sin av dagen'] = np.sin(2*np.pi*df.index.hour/24)
df['Helg'] = ( df.index.dayofweek > 4 ) * 1.0

df_d = df.pivot_table(index=df.index.dayofyear,
                      values=['Forbruk.m2', 'Utetemperatur', 'Helg'])
df_d['Forbruk.m2'] *= 24
df_d['Cos av året'] = np.cos(2*np.pi*(df_d.index - 15)/365)
df_d['Sin av året'] = np.sin(2*np.pi*(df_d.index - 0)/365)
#####
```

Figur 18 Genererer variablene i «DataFrame» for times- og døgnerverdier



Deretter skilles dataene i x og y, hvor x er variablene, mens y er energiforbruket. Dette gjøres også for døgnverdiene. Figur 19 beskriver hvordan dette gjøres i Python. Videre splittes variablene og energiforbruket opp i såkalte treningssett og testsett. Dette er for å forsikre at modellen testes på et sett med variabler den ikke har sett før. Det er vanlig å dele opp dataene i 70% treningsdata og 30% test data.

```
#Standardiserer og splitter i trening/test sett
#####
X, y = df.loc[:, 'Utetemperatur:'], df.loc[:, 'Forbruk.m2']
X_raw, y_raw = df_raw.loc[:, 'Utetemperatur:'], df_raw.loc[:, 'Forbruk.m2']
X_d, y_d = df_d.loc[:, 'Helg:'], df_d.loc[:, 'Forbruk.m2']
X_d_raw, y_d_raw = df_d_raw.loc[:, 'Helg:'], df_d_raw.loc[:, 'Forbruk.m2']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=2)
X_d_train, X_d_test, y_d_train, y_d_test = train_test_split(
    X_d, y_d, test_size=0.3, random_state=2)
#####
```

Figur 19 Fordeler dataene i trening- og testsett

Noen algoritmer, som f.eks. multippel lineær regresjon har behov for at variablene standardiseres. For multippel lineær regresjon kan koeffisientene for de enkelte variablene gi en indikasjon på hvor mye hver variabel bidrar i prediksjonen. For å kunne sammenligne variablene må alle dataene standardiseres. Dette betyr at alle variablene hver for seg skaleres slik at standardavviket blir 1, og funksjonen sentreres rundt 0. I Python benyttes funksjonen «fit_transform» fra biblioteket «StandardScaler» som er et underbibliotek av «sklearn.preprocessing» til å standardisere dataene.

Når en maskinlæringsalgoritme fra Sci-kit Learn tas i bruk, lages det en klasse. Ofte krever algoritmen inputparametere. I tilfelle med multippel lineær regresjon kreves som vist i Figur 20, det ingen inputparametere. Klassen som genereres inneholder mange ulike funksjoner. «Fit»-funksjonen benytter treningssettet til å «trene» opp modellen med såkalt veiledende læring. Hvordan dette gjøres i python beskrives av samme figur. Da gis svaret (energiforbruket) samtidig med tilhørende variabler og koeffisientene tilpasses med regresjon. Det trenes opp en modell basert på timesverdier og en egen modell basert på døgnverdier. Disse modellene får litt ulike egenskaper og brukerområde. Når modellen er tilpasset, predikerer modellen det nye energiforbruket basert på variablene (x) i testsettet. Dette nye datasettet kan sammenlignes med fasiten (y) for å se hvor god modellen er.

```
#Prediksjon med Linear Regression timesverdier
#####
mlr = LinearRegression()
mlr.fit(X_train_std, y_train)
mlr_y_test_pred = pd.DataFrame(mlr.predict(X_test_std), index=X_test.index)
mlr_y_raw_pred = pd.DataFrame(mlr.predict(X_raw_std), index=X_raw_std.index)
mlr_r2 = mlr.score(X_test_std, y_test)
mlr_rmse = np.sqrt(metrics.mean_squared_error(y_test, mlr_y_test_pred))
#####
```

Figur 20 Trener modellen og predikerer



Det finnes flere måter å beskrive hvor presis modellens prediksjon er. Det er vanlig å blant annet benytte korrelasjonskoeffisienten r . Som nevnt tidligere er dette et statistisk mål på sammenhengen mellom to størrelser. I dette tilfelle dreier det seg om sammenhengen mellom test-data og predikert data. Dersom man kvadrerer r , beskriver det hvor stor del av den totale variansen i test-dataene som er «forklart» av regresjonsmodellen. Dette gjøres med funksjonen «score» i Python, som vist i Figur 20. For å beskrive den faktiske feilen modellen gjør, beregnes også RMSE-verdi som er den gjennomsnittlige feilen for hver predikert verdi. Funksjon «mean_squared_error» i Python fra underbiblioteket «metric» beregner MSE-verdien. RMSE-verdien får man da ved å ta roten av denne.

Prediksjonsmodellen benyttes i hovedsak til å predikere timesverdier, siden formålet er benytte så mye informasjon i dataene som mulig til å trene modellen. Prediksjonsmodellen blir også benyttet på døgnverdier av dataene. Da sammenlignes naturligvis de predikerte verdiene mot døgnverdier av «fasiten» av energiforbruket. Det er også mulig å beregne døgnverdier av de predikerte timesverdiene. Da har man på en måte predikert døgnverdiene basert på timesverdiene. Disse kan på samme måte sammenlignes med døgnverdier av «fasiten». Dette kan være en test for å se hvor mye prediksjonsmodellen har lært av timesverdiene sammenlignet med døgnverdier.



3.4 DATAKVALITET

Kvaliteten til de ulike kvantitative dataene bestemmes i form av validitet og reliabilitet. Disse er uttrykk som benyttes i forskningsmetoder for å beskrive kvaliteten til dataene. Reliabilitet beskriver hvor pålitelige data man har i forskningsarbeidet, og i hvilken grad innsamling av data er foretatt. Mens validitet beskriver i hvilken grad dataene er gyldig/relevant for problemstillingen i oppgaven (Grønmo, 2020). I denne oppgaven er kvaliteten til innsamlede data ivaretatt gjennom grundig databehandlingsprosess. Python-scriptene har blitt fullstendig utviklet etter en testperiode med 12 bygg, og har gjennomgått jevnlig kontroll. Kvaliteten til de genererte resultatene fra de ulike Python-scriptene er ivaretatt gjennom de statistiske koeffisientene, som korrelasjonskoeffisient, R^2 -verdi og RMSE-verdi.

3.4.1 RELIABILITET

Som nevnt tidligere blir all rådata hovedsakelig innhentet fra Energinett. I tillegg innhentes data fra eksterne filer tilsendt fra ekstern veileder for denne oppgaven. Disse dataene forutsettes å ha god pålitelighet. For å sikre et godt grunnlag for Python-scriptene ble datainnsamlingen/databehandlingen fra dette tidspunktet foretatt nøyaktig. All rådata fra de ulike målerne for de ulike bygningene ble filtrert og nedlastet i samme form. Datainnsamlingen av de «riktige»/brukbare dataene foregikk hovedsakelig i Python. Her ble omtrentlig halvparten av de nedlastede dataene fjernet pga. lite pålitelighet. Dette er for å sikre påliteligheten til de «genererte dataene» eller resultatene. Et annet viktig faktor er tilnæringsmåten når disse resultatene genereres ved hjelp av Python. Spesielt ved generering av analyseresultater er det foretatt konservative valg. Prediksjonsresultater derimot anses å ha god pålitelighet pga. resultatene blir henholdsvis generert av ferdige maskinlærings biblioteker/pakker i Python.

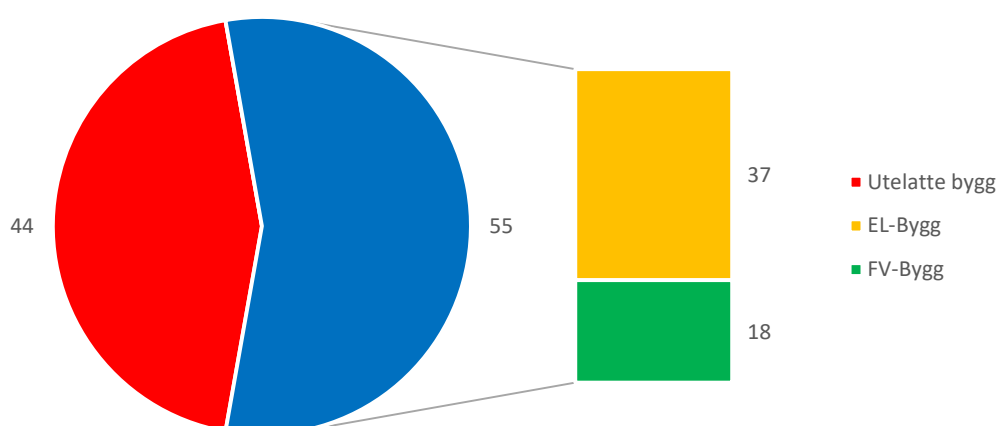
3.4.2 VALIDITET

Her er det viktig å skille mellom indre og ytre validitet. Indre validitet sier noe om i hvilken grad resultatene kan bli påvirket. Ytre validitet sier noe om i hvilken grad resultatene kan generaliseres (Dahlum, 2020). I tillegg til riktig databehandling er indre validiteten sikret i form av egen- og sidemannskontroller. Egenkontrollen har foregått ved å observere nærmere metoden som ble bruk for å generere det aktuelle resultatet. Sidemannskontrollen gikk ut på å se om resultatet gir mening etter første øyekast, og om den er relevant i forhold til hele rapporten og problemstillingen. Ytre validiteten er sikret ved å danne sammenlignbare resultater. Dette er ivaretatt ved å dividere forbruket for hvert bygg med tilhørende arealstørrelser.

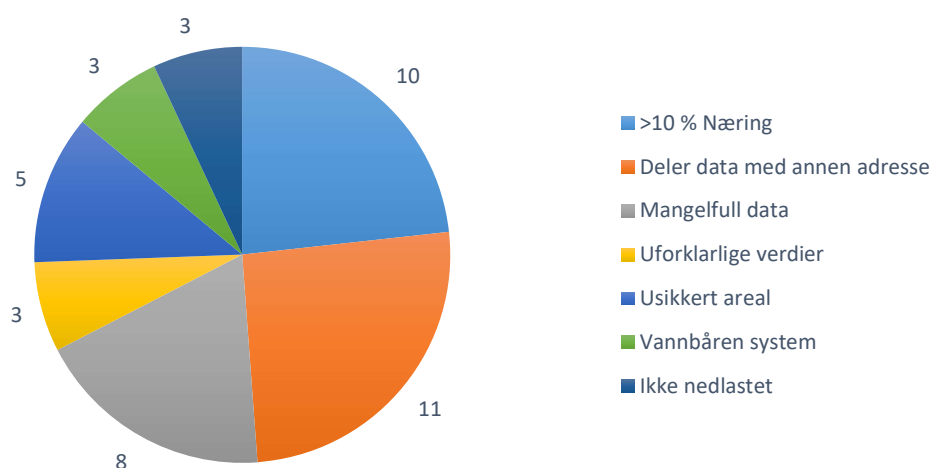
4 RESULTAT

4.1 DATABEHANDLING

Figur 21 presenterer databehandlingsprosessen, hvor a) presenterer hvor mange bygg som skal brukes under analysen basert på energiforsyningssystem, og b) viser antall bygg som er utelatt av ulike årsaker. Totalt 44 av 99 bygninger er fjernet. Enøkanalysene og første gjennomgang av data for hvert bygg har vært grunnlaget for å avklare dette. Her kan man blant annet se at bygg med mer enn 10% næringsareal og bygg med delt data er de to dominerende årsakene. Videre deles de 55 bygningene inn i bygninger som dekker varmebehovet med fjernvarme (FV-bygg), og elektrisitet (EL-bygg).



a)



b)

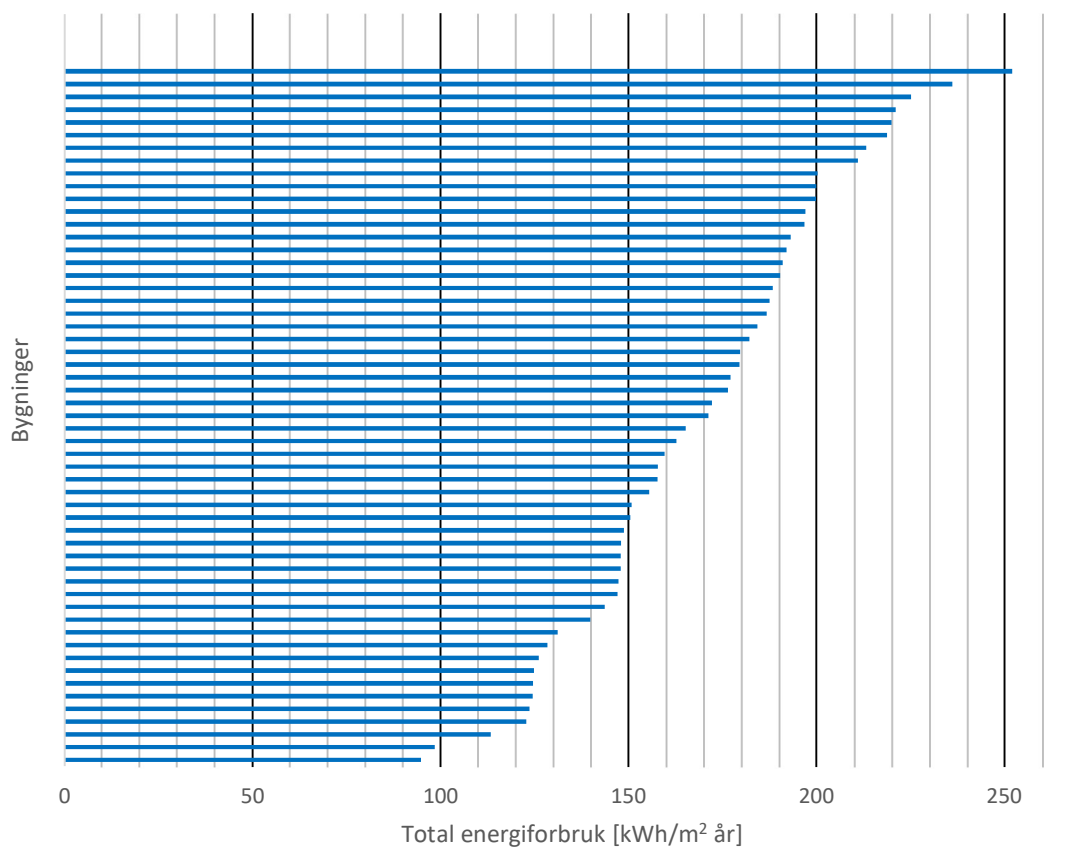
Figur 21 Databehandlingsresultater, hvor a) viser antall bygg med ulike energiforsyningssystemer, b) viser de utelatte bygningene av ulike årsaker.



4.2 ANALYSE

4.2.1 ALLE BYGNINGENE

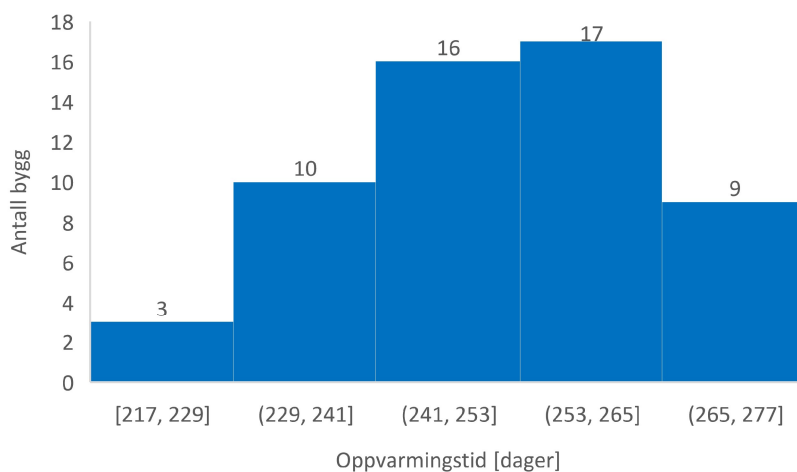
Figur 22 presenterer totalt årlig energiforbruk pr. kvm. for alle de 55 bygningene i analysen. Det viser seg at de fleste bygningene har et årlig energiforbruk på 100-200 kWh/m², med en jevn fordeling. Kun to bygg har et årlig energiforbruk lavere enn 100 kWh/m², og åtte bygg som har mer enn 200 kWh/m².



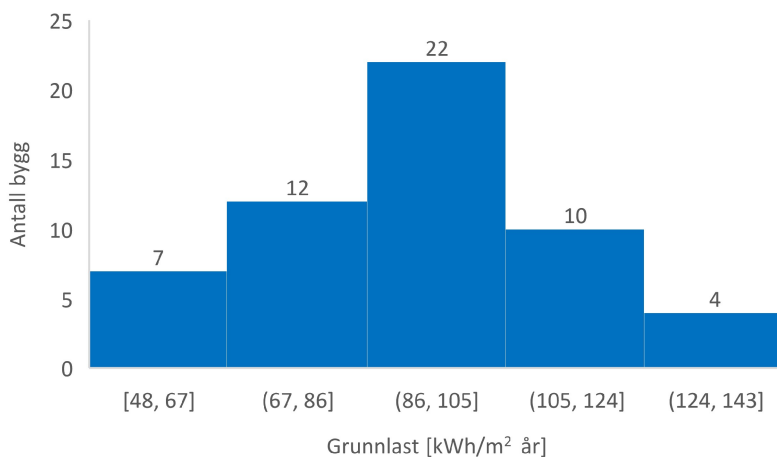
Figur 22 Totalt årlig energiforbruk pr kvm. for alle bygninger (55 bygninger)



Figur 23 a) presenterer oppvarmingstid. De fleste bygningene en oppvarmingstid som ligger mellom 241-265 dager. b) viser grunnlasten for de ulike bygningene. Det viser seg at det er en tydelig normalfordeling, hvor de fleste bygningene bruker mellom 67 – 124 kWh/m² i året til å dekke grunnlasten.



a)

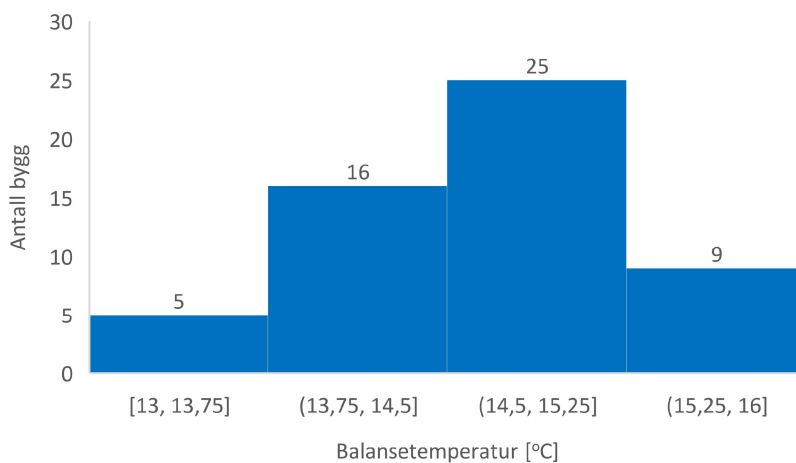


b)

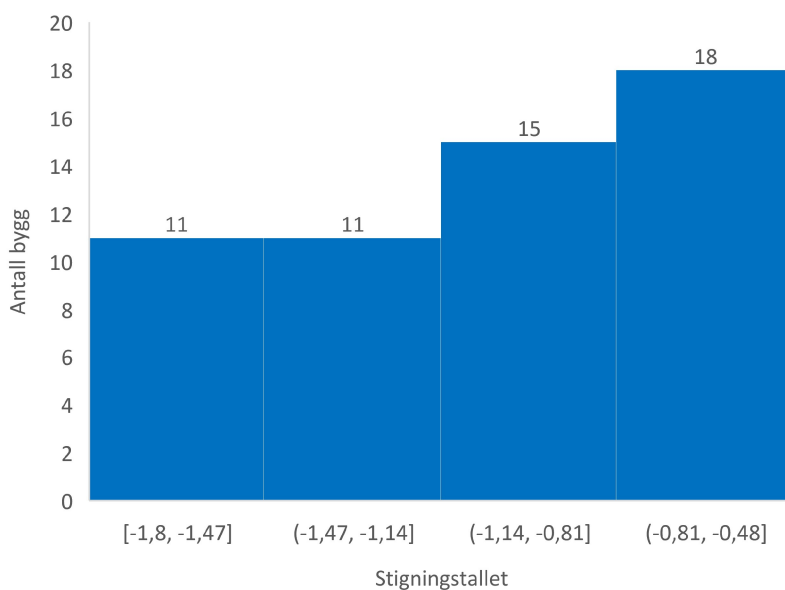
Figur 23 Fordeling av a) oppvarmingstid, og b) grunnlast for de ulike bygningene (55 bygninger)



Videre presenteres i Figur 24 a) fordelingen av balansetemperaturen for de ulike bygningene. Her kan man se en normalfordeling, hvor de fleste bygningene har en balansetemperatur mellom 13-15 °C. b) presenterer stigningstallet for den temperaturavhengige delen i ET-kurven til de ulike bygningene. Legg merke til at 23 bygg har et negativ stigningstall mellom 1,8 – 1,14, i forhold til 32 bygg som ligger mellom 1,14 – 0,5, noe som tyder på en gruppering.



a)



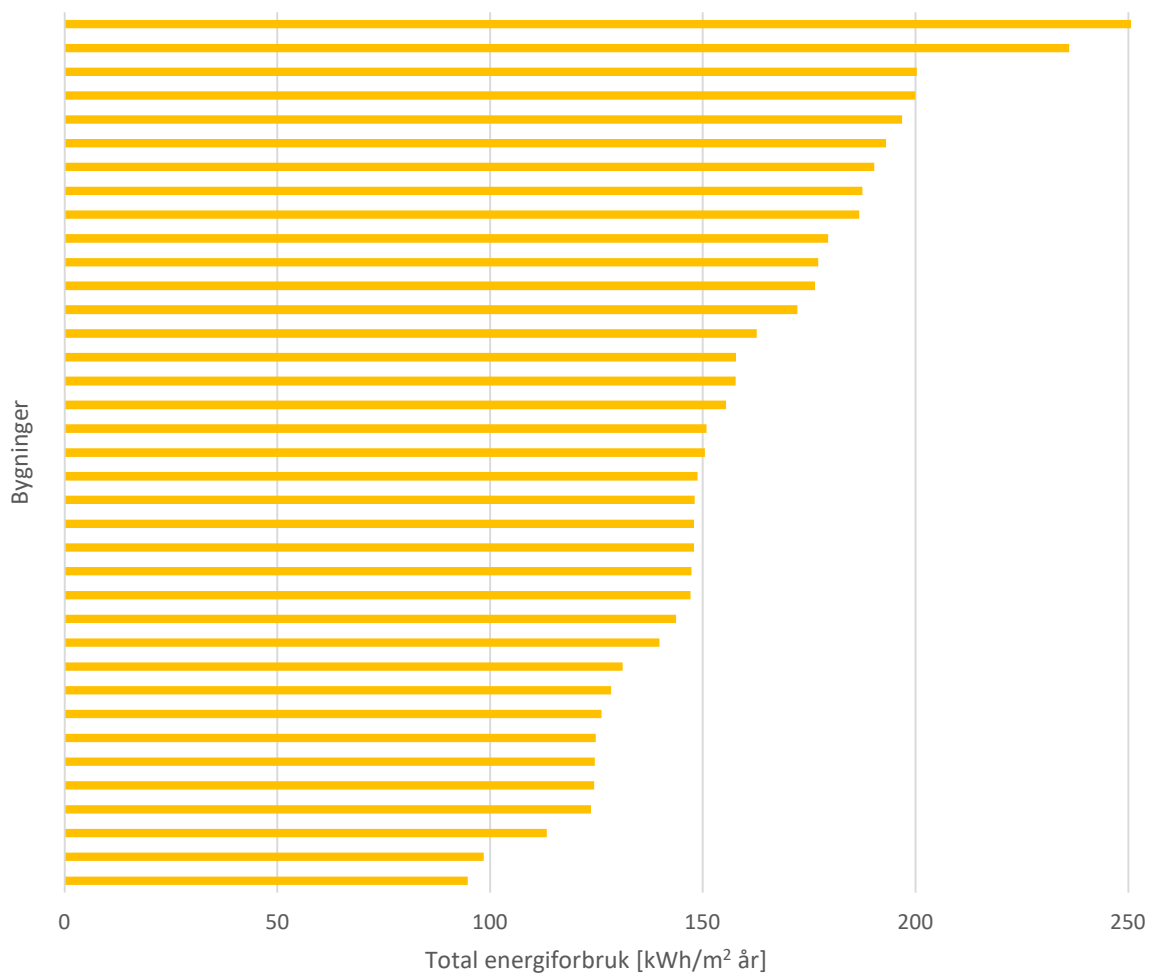
b)

Figur 24 Fordeling av a) balansetemperaturen og b) stigningstallet for de ulike bygningene (55 bygninger)



4.2.2 BYGNINGER MED ELEKTRISK OPPVARMING

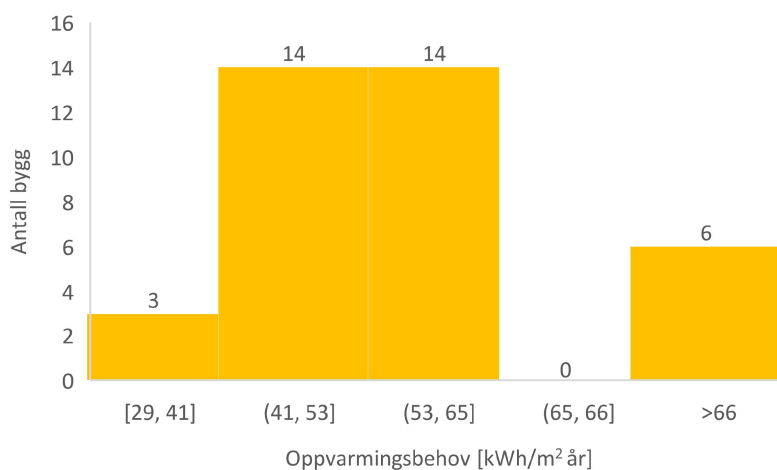
Figur 25 presenterer årlig energiforbruk pr. kvm for de 37 bygningene som bruker elektrisitet til oppvarming. Disse bygningene ser ut til å ligge i den nedre delen av tilsvarende resultat for alle bygninger i Figur 22.



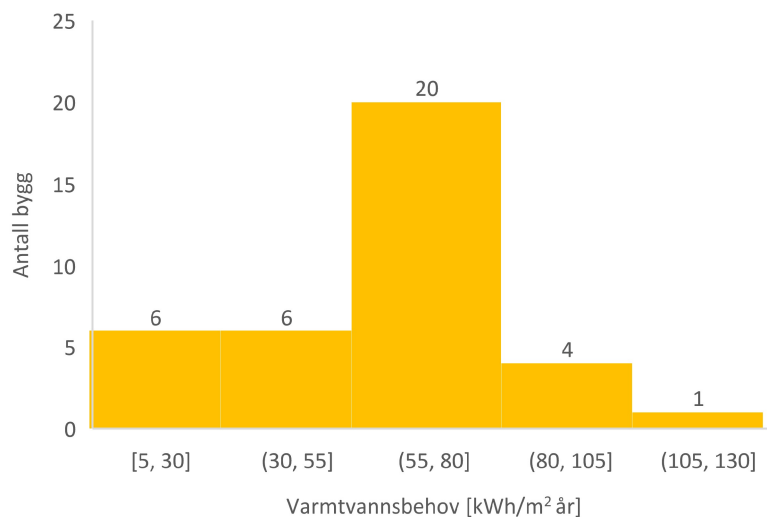
Figur 25 Årlig energiforbruk pr. kvm for bygninger som bruker elektrisitet til oppvarming (37 bygninger)



Figur 26 presenterer estimat av energiforbruk som skal dekke oppvarmings- og varmtvannsbehovet. For de 37 bygningene som bruker elektrisitet til å dekke varmebehovet, viser a) at de fleste bygninger bruker mellom 41 og 65 kWh/m² i året til å dekke oppvarmingsbehovet. b) viser estimert energiforbruk som skal dekke varmtvannsbehovet. De fleste bygninger ligger mellom 55 og 80 kWh/m² i året til å dekke varmtvannsbehovet. Det er stor variasjon i dette estimatet.



a)

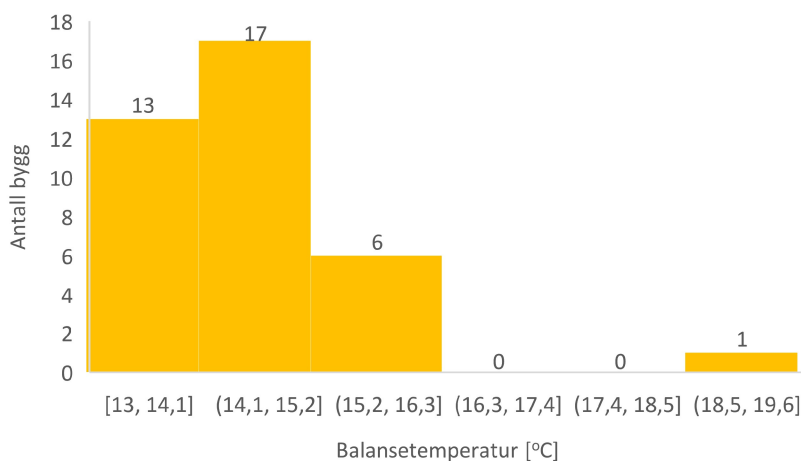


b)

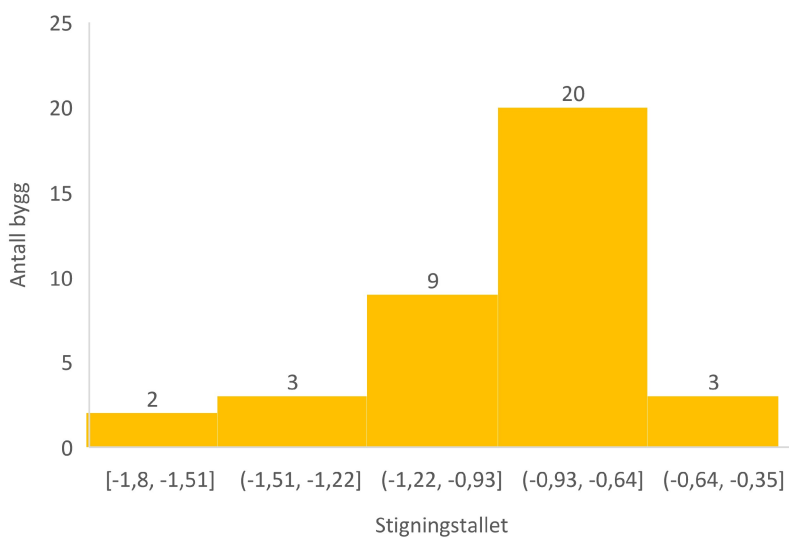
Figur 26 Fordeling av energiforbruk som skal dekke a) oppvarmingsbehovet og b) varmtvannsbehovet for bygninger som bruker elektrisitet til oppvarming (37 bygninger). Estimerte verdier



Figur 27 presenterer a) balansetemperaturen og b) stigningstallet for bygg med elektrisk oppvarming. Én bygning skiller seg ut i a), som tyder på en mulig uteligger i dataene. Stigningstallet virker normalfordelt.



a)



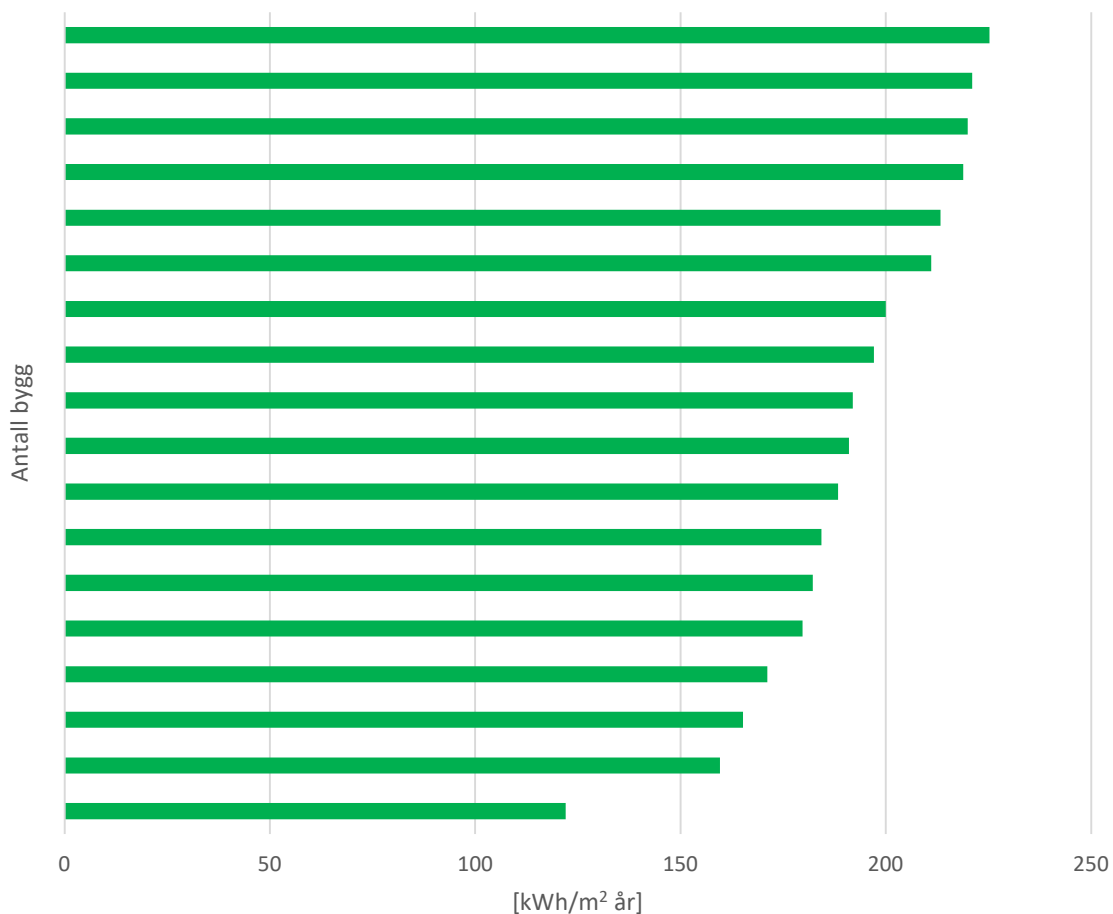
b)

Figur 27 Fordeling av a) balansetemperatur og b) stigningstallet for bygninger som bruker elektrisitet til oppvarming (37 bygninger). Estimerte verdier



4.2.3 BYGNINGER MED FJERNVARME

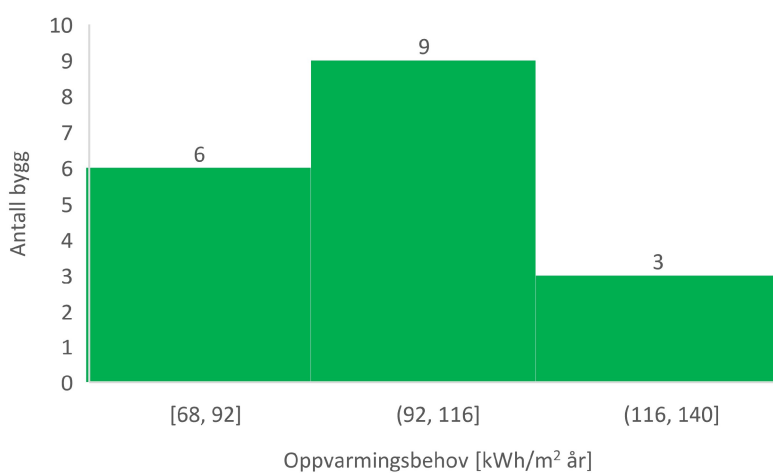
Figur 28 presenterer årlig energiforbruk pr. kvm for de 18 bygningene som bruker fjernvarme til oppvarming. Disse bygningene har generelt et større energiforbruk i forhold til bygninger som bruker elektrisitet til å dekke varmebehovet.



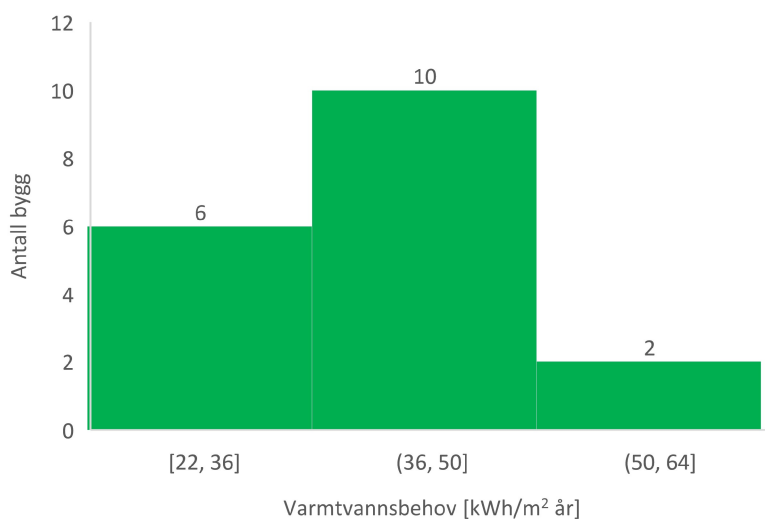
Figur 28 Årlig energiforbruk pr.kvm for bygninger som bruker fjernvarme til oppvarming (18 bygninger)



Figur 29 presenterer estimat av energiforbruk som skal dekke oppvarmings- og varmtvannsbehovet, for de 18 bygningene som bruker fjernvarme til å dekke varmebehovet. a) viser at bygningene bruker alt mellom 68 og 140 kWh/m² i året til å dekke oppvarmingsbehovet. Dette er betydelig mer, sammenlignet med bygninger med elektrisk oppvarming. I tillegg tyder resultatene på stor variasjon i bygningsmassen. b) viser estimert energiforbruk som skal dekke varmtvannsbehovet. Igjen tyder resultatene på høy varians.



a)

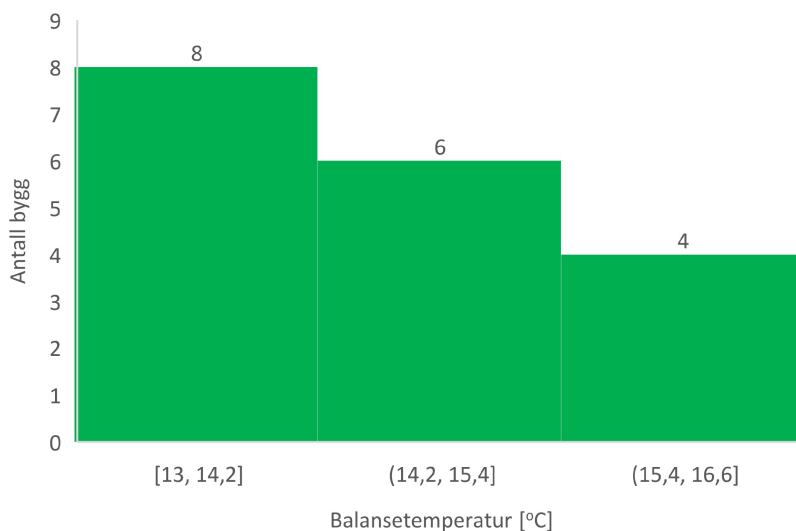


b)

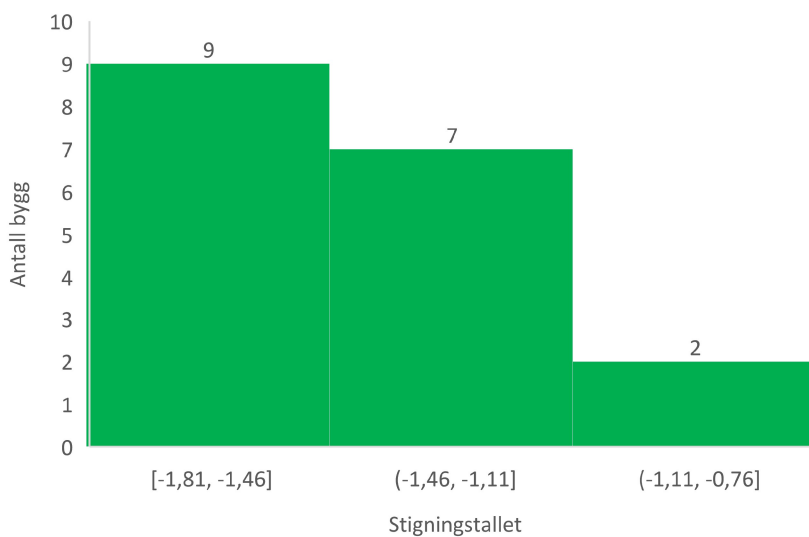
Figur 29 Fordeling av energiforbruk som skal dekke a) oppvarmingsbehovet og b) varmtvannsbehovet for bygninger som bruker fjernvarme til oppvarming (18 bygninger). Estimerte verdier



Figur 30 presenterer a) balansetemperaturen og b) stigningstallet for bygg med fjernvarme. Balansetemperatur samsvarer med verdier for bygninger som dekker varmebehovet med elektrisitet, mens stigningstallet tyder på en vesentlig brattere ET-kurve her.



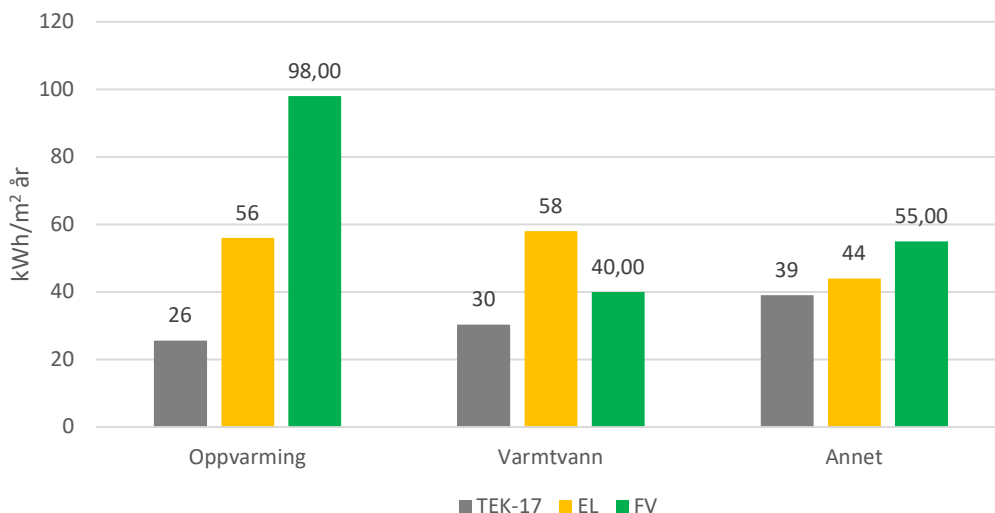
a)



b)

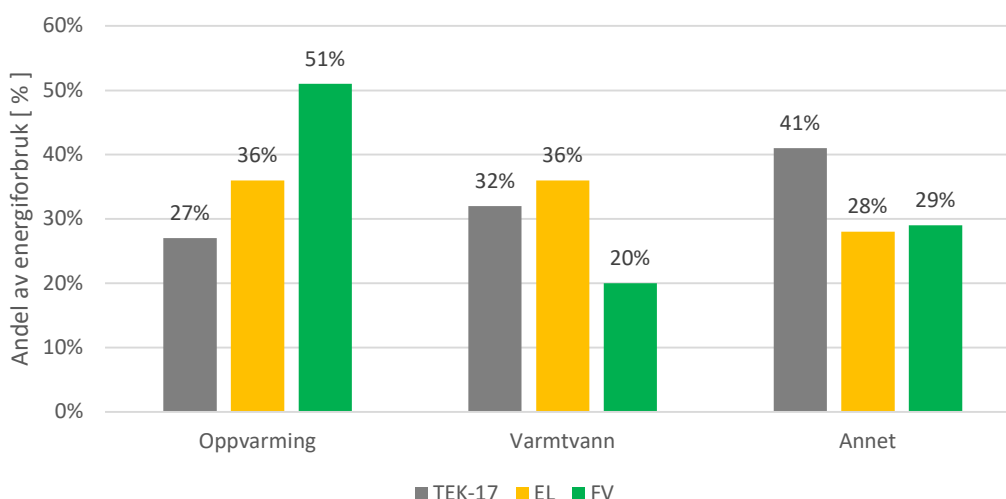
Figur 30 Fordeling av a) balansetemperatur og b) stigningstallet for bygninger som bruker fjernvarme til oppvarming (18 bygninger). Estimerte verdier

Figur 31 viser gjennomsnittlig verdi for estimat av oppvarmingsbehov, varmtvannsbehov og annet for EL-bygg og FV-bygg. Dette sammenlignes med de normerte verdiene fra NS3031/TEK-17 slik som forklart i kapittel 3.2.3. Det er tydelig at både FV-bygg og EL-bygg ligger over TEK-17 kravet til oppvarming. Spesielt bygninger med fjernvarme som har et estimert oppvarmingsbehov på hele 98 kWh/m² i året i gjennomsnitt.



Figur 31 Faktisk energiforbruk av oppvarming, varmtvann og annet for bygninger med fjernvarme og bygninger som bruker elektrisitet til oppvarming, sammenlignet med TEK-17. Alle er estimerte verdier

Figur 32 presenterer tilsvarende estimater som i Figur 31, men i en prosentsats av det totale energiforbruket.



Figur 32 Prosentstørrelse av energiforbruk til oppvarming, varmtvann og annet for bygninger med fjernvarme og bygninger som bruker elektrisitet til oppvarming, sammenlignet med TEK-17. Alle er estimerte verdier

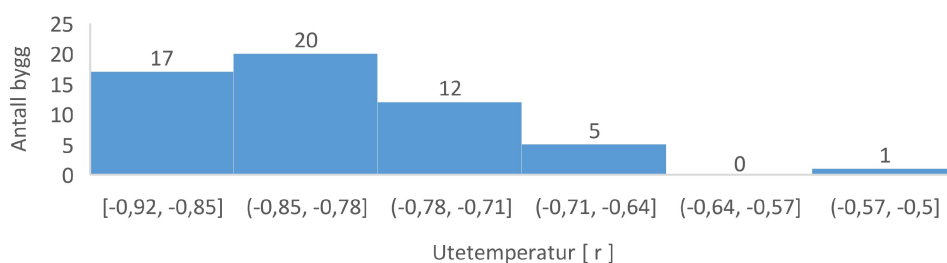


4.3 VARIABLENE

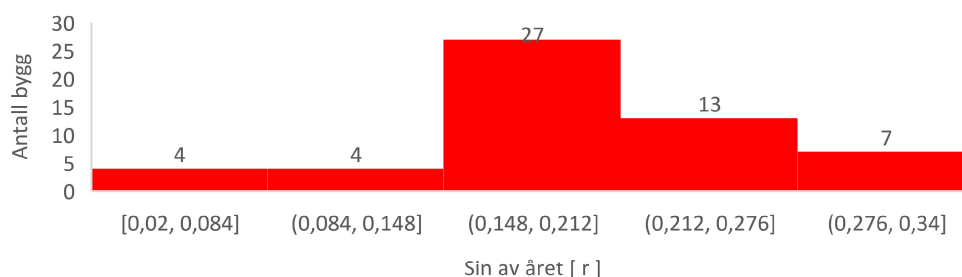
I dette kapittelet vises korrelasjonen r mellom energiforbruket og de ulike variablene. Variablene forsøker å beskrive variasjonen i energiforbruket. Det skiller i hovedsak mellom variasjonen over året og variasjonen over døgnet. Det skiller derfor mellom timesverdier, døgnerverdier og døgnavariasjon i resultatet. Døgnavariasjon er gjennomsnittlige timesverdien over døgnet subtrahert fra tilhørende timesverdi. Generelt viser det seg at de fleste korrelasjonene er normalfordelt.

4.3.1 TIMESVERDIER

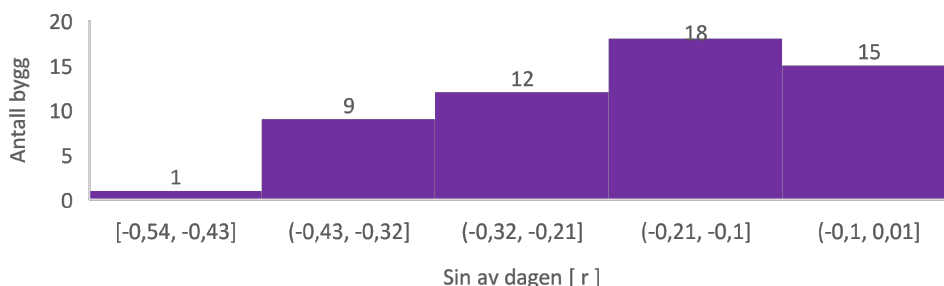
Figur 33 a) viser at korrelasjon mellom energiforbruket og utetemperatur for de fleste bygningene ligger mellom $-0,85$ og $-0,78$. Verdien er negativ som betyr at energiforbruket avtar ved økende temperatur. Variabelen b) sin av året forsøker å beskrive den årlige variasjonen i energiforbruket, mens c) sin av dagen forsøker å beskrive den daglige variasjonen. Disse viser lave verdier.



a)



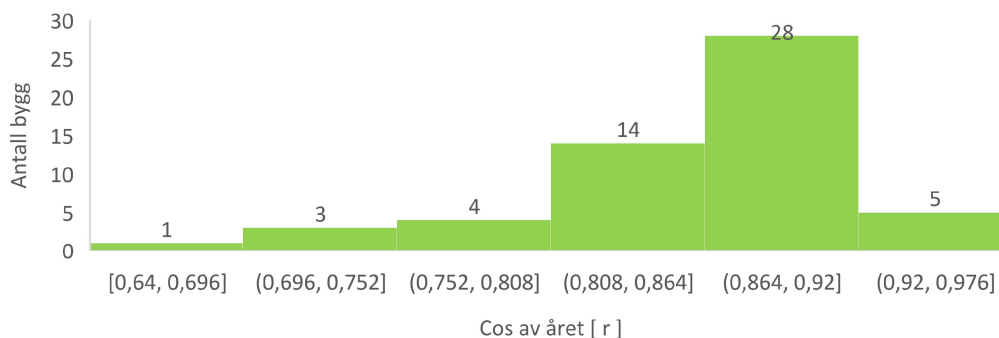
b)



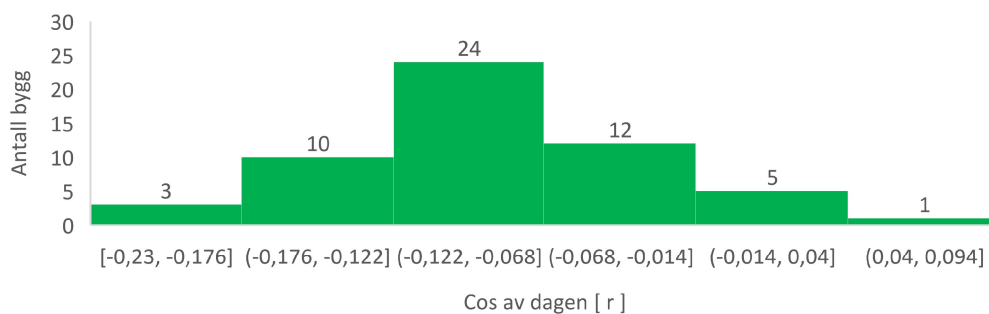
c)

Figur 33 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) utetemperatur, b) sin av året og c) sin av dagen, basert på timesverdier, for de ulike bygningene (55 bygninger)

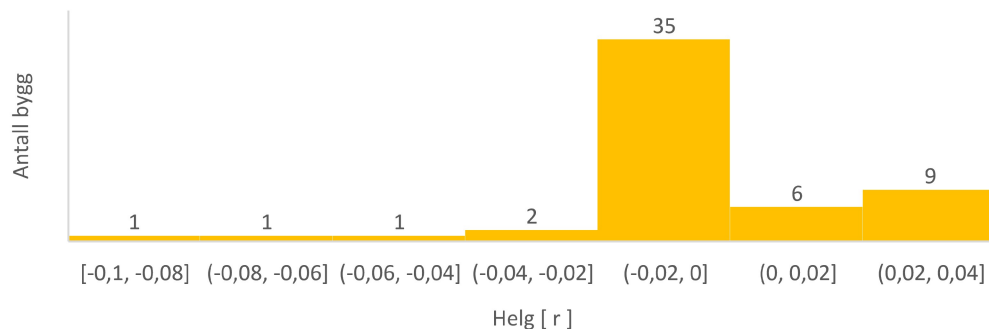
Figur 34 a) presenterer korrelasjonskoeffisient mellom energiforbruk og cos av året, som ligger mellom 0,86 og 0,92 for de fleste bygninger. Dette viser at denne variabelen har en høyere korrelasjon med energiforbruket enn utetemperaturen for timesverdier. Her viser verdien seg å være positiv. Resultater i b) Cos av dagen og c) helg viser seg å ha lav korrelasjon.



a)



b)

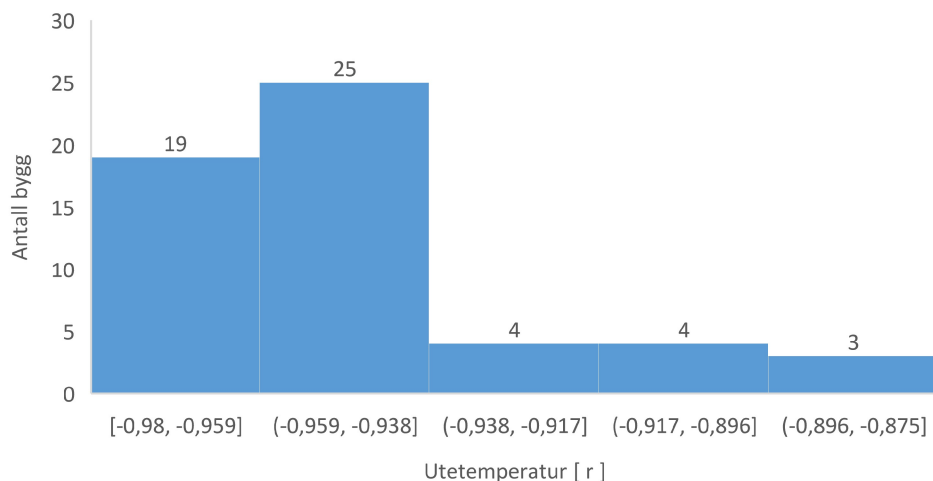


c)

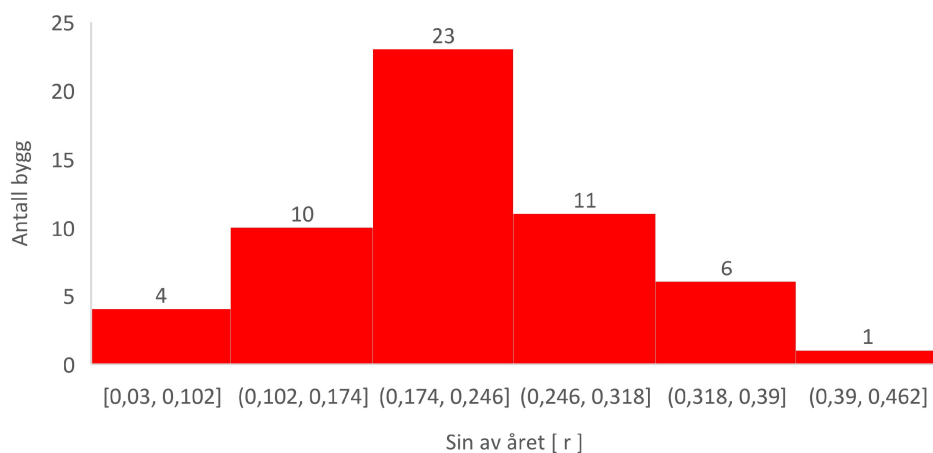
Figur 34 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) cos av året, b) cos av dagen og c) helg, basert på timesverdier, for de ulike bygningene (55 bygninger)

4.3.2 DØGNVERDIER

Figur 35 presenterer korrelasjonskoeffisient mellom energiforbruk og a) utetemperatur, og b) sin av året for alle bygninger, basert på døgnerverdier. Variabler som forsøker å beskrive variasjonen over døgnet blir naturligvis ikke presentert i disse resultatene. Utetemperaturen viser seg å få betydelig høyere verdier sammenlignet med for timesverdier.



a)

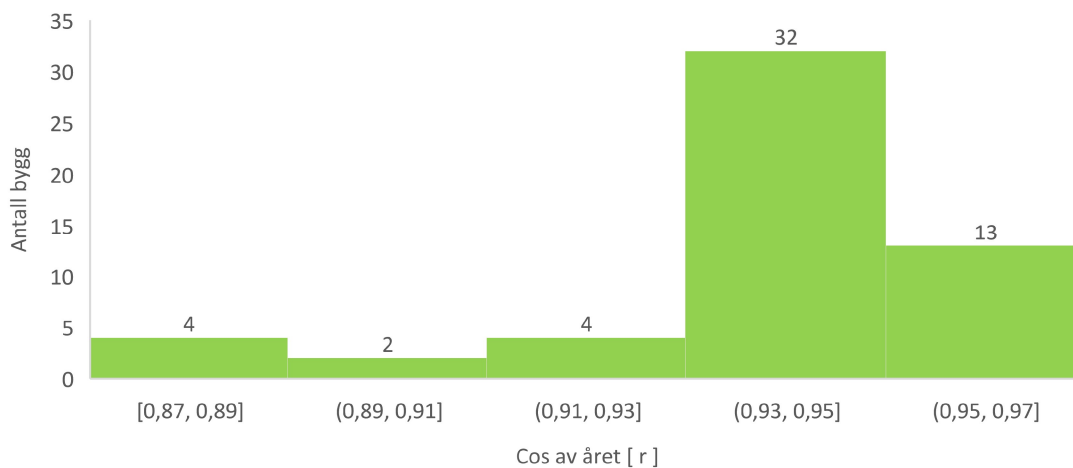


b)

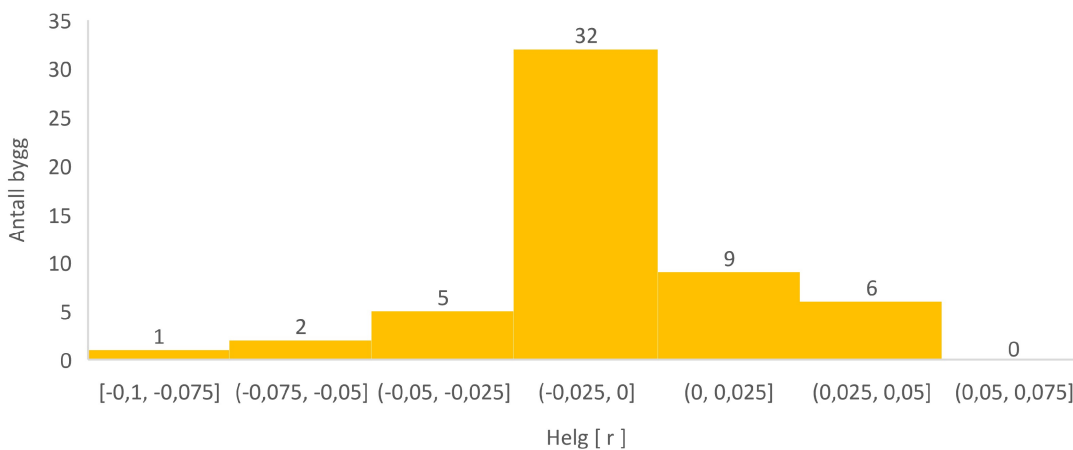
Figur 35 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) utetemperatur og b) sin av året, basert på døgnerverdier, for de ulike bygningene (55 bygninger)



Figur 36 presenterer korrelasjonskoeffisient mellom energiforbruk og a) cos av året, og b) helg for alle bygninger, basert på døgnverdier. Cos av året, som skal beskrive variasjonen over året, får naturligvis høy korrelasjon her. Helg har veldig lav korrelasjon.



a)



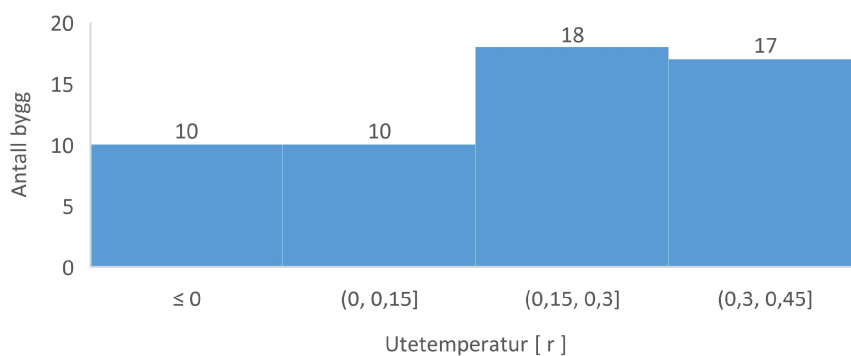
b)

Figur 36 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) cos av året og b) helg, basert på døgnverdier, for de ulike bygningene

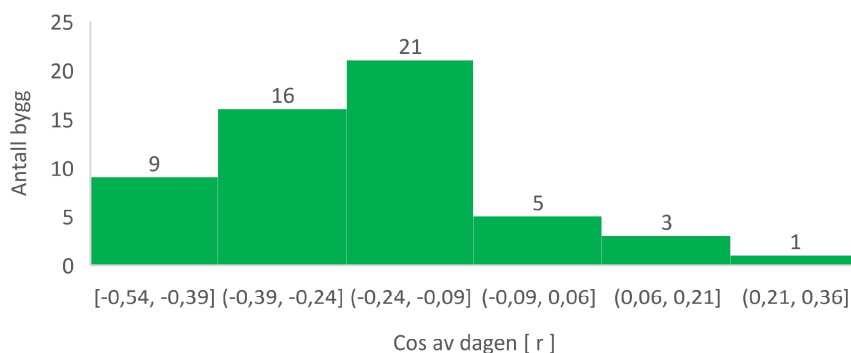


4.3.3 DØGNVARIASJON

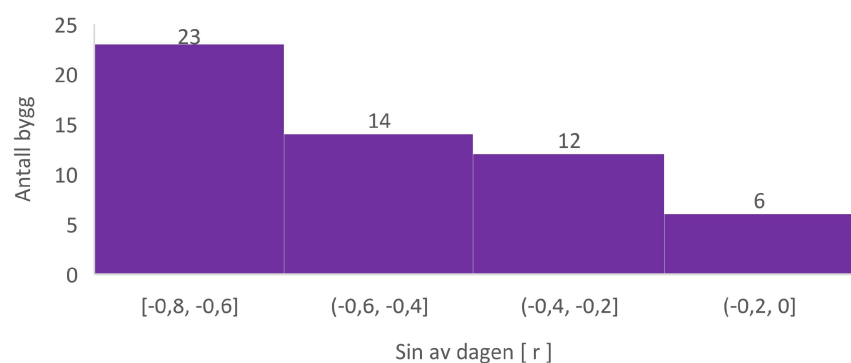
Figur 37 presenterer korrelasjonskoeffisient mellom energiforbruk og a) utetemperaturen, b) cos av dagen, og c) sin av dagen for alle bygninger, basert på døgnvariasjon. Variablene som forsøker å beskrive variasjonen over året, faller naturligvis bort her. Legg merke til utetemperaturen får positiv korrelasjon her, og generelt lavere verdi. Sin av dagen får forholdsvis høye korrelasjoner for de fleste bygninger.



a)



b)

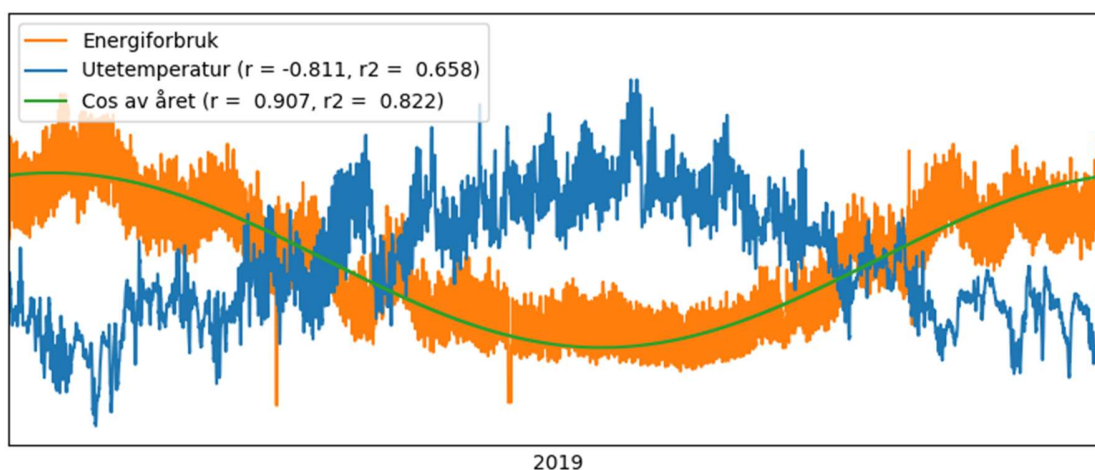


c)

Figur 37 Korrelasjonskoeffisient mellom energiforbruk og variabelen a) utetemperatur, b) cos av dagen og c) sin av dagen, basert på døgnvariasjon, for de ulike bygningene (55 bygninger)

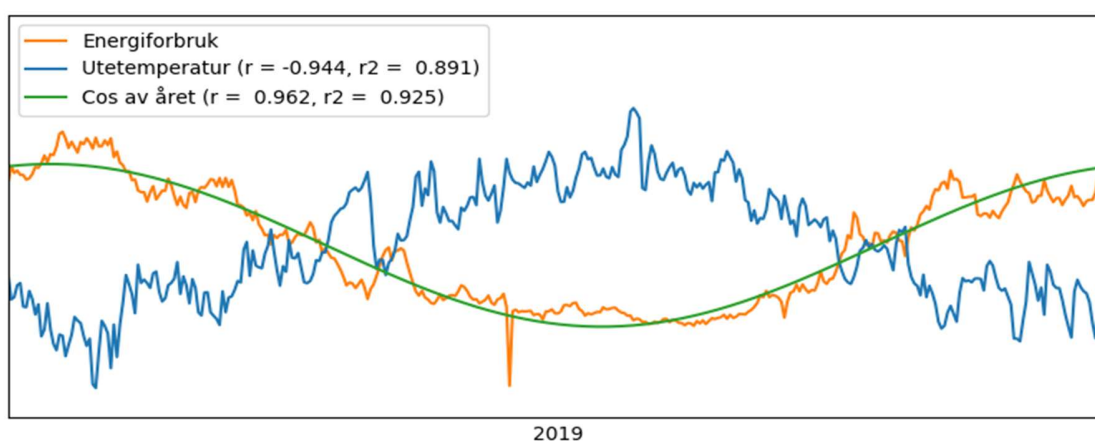
4.3.4 VARIABLER FOR EN TILFELDIG VALGT BYGNING

I dette delkapittelet presenteres de viktigste funnene av korrelasjonsanalysen, som forklares med en tilfeldig valgt bygning. I Figur 38 er utetemperaturen og cos av året plottet sammen med energiforbruket for å få en grafisk forståelse av sammenhengen. I tillegg er kvadratet av korrelasjonen presentert. Denne beskriver da hvor mye av variasjonen i energiforbruket som kan forklares av variablene. Utetemperaturen og cos av året trekkes frem her, fordi de har høyest korrelasjon.



Figur 38 Korrelasjonen mellom energiforbruk og utetemperatur, og cos av året for en tilfeldig valgt bygning. Basert på timesverdier

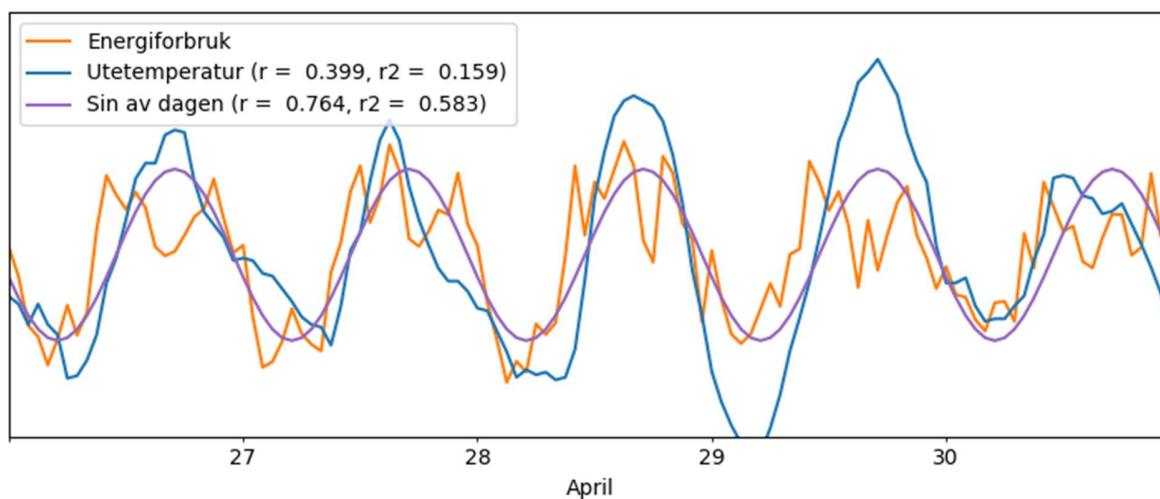
Figur 39 presenterer tilsvarende variabler for døgnverdier.



Figur 39 Korrelasjonen mellom energiforbruk og utetemperatur, og cos av året for en tilfeldig valgt bygning. Basert på døgnverdier



Figur 40 viser døgnvariasjonen. Dette er da en beskrivelse kun av variasjonen over døgnet. Her vises det grafisk at utetemperaturen har fått en positiv korrelasjon. Dette motstrider mot for døgnverdier. I tillegg vises variabelen sin av dagen, som gir best korrelasjon for døgnvariasjon.



Figur 40 Korrelasjonen mellom energiforbruk og utetemperatur, og sin av dagen for en tilfeldig valgt bygning. Basert på døgnvariasjon

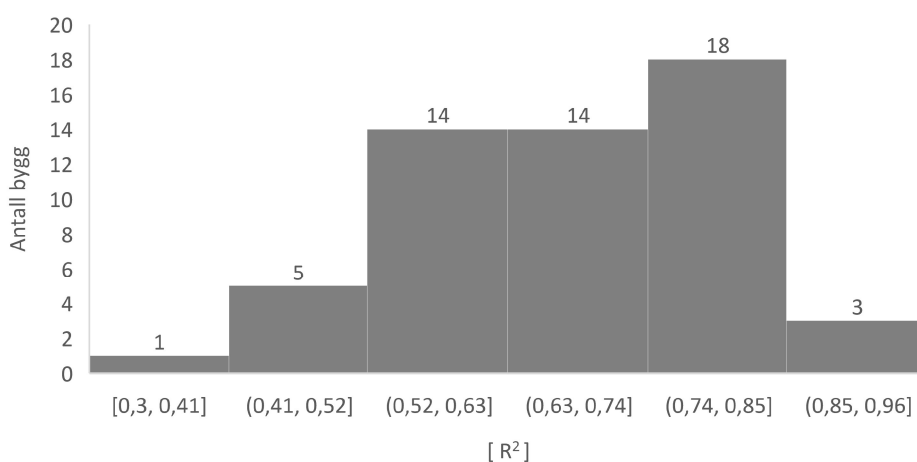


4.4 PREDIKSJON – ALLE BYGNINGER

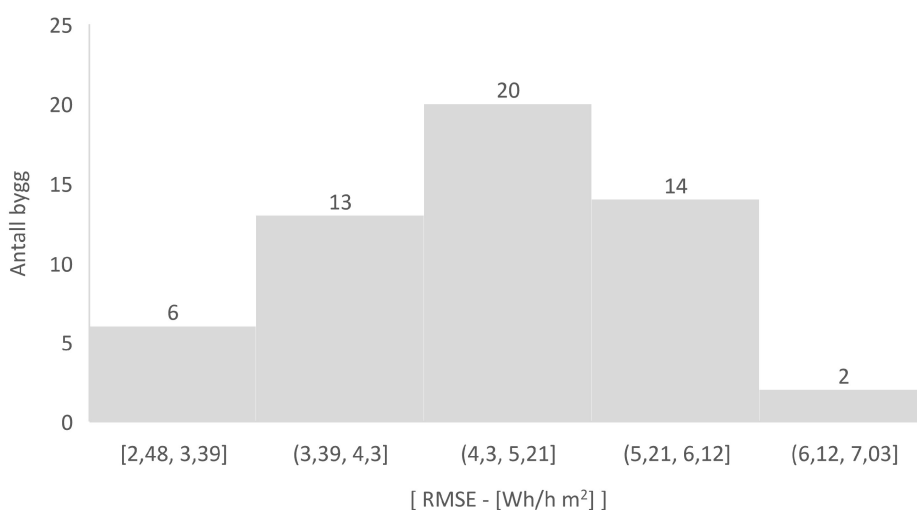
I dette delkapittelet presenteres prediksjonsresultater fra ET-kurve og de ulike maskinlæringsalgoritmene for alle bygningene. Kvaliteten til disse presenteres med R^2 -verdi og RMSE-verdi. R^2 -verdien beskriver hvor stor andel av variasjonen i energiforbruket som kan forklares av prediksjonsmodellen. RMSE-verdien beskriver avviket.

4.4.1 PREDIKSJON MED ET-KURVE

Figur 41 presenterer prediksjonsresultater med tradisjonell ET-kurve, for alle bygningene. Det tyder på en normalfordeling både for R^2 -verdi og RMSE-verdi.



a)



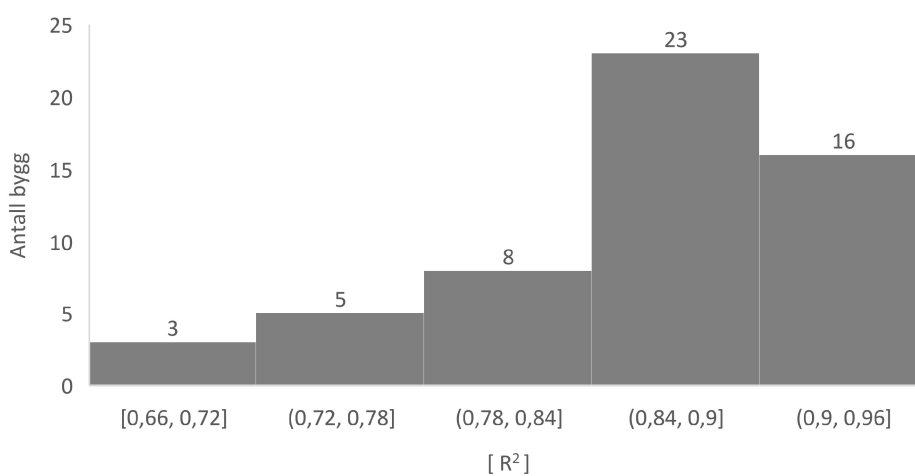
b)

Figur 41 Prediksjonsresultater med tradisjonell ET-kurve, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE

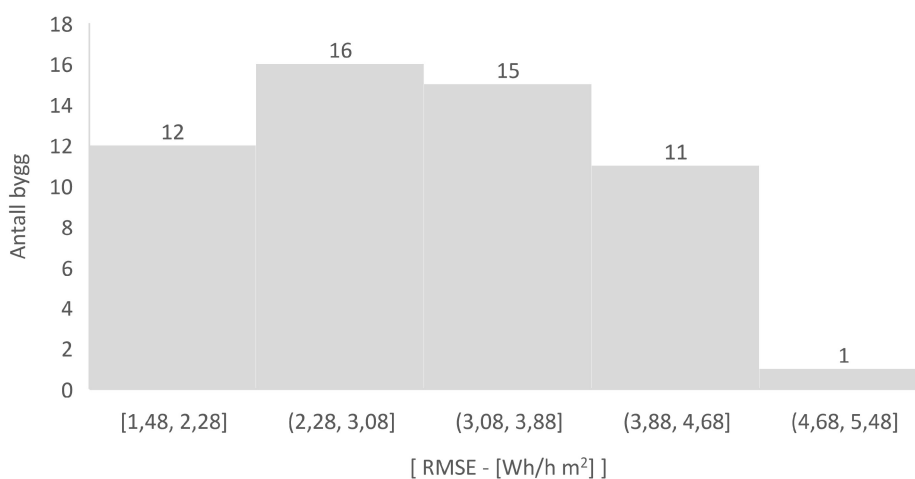
4.4.2 PREDIKSJON MED MASKINLÆRINGSALGORITMER

Her presenteres prediksjon gjort med de ulike maskinlæringsalgoritmene. De samme variablene benyttes i alle prediksjonsmodellene. Dette skaper sammenligningsgrunnlag for de ulike maskinlæringsalgoritmene. Generelt sett kan det observeres en normalfordeling for R^2 - og RMSE-verdiene, for samtlige maskinlæringsalgoritmene. Alle modellene predikerer bedre enn ET-kurven. Det viser seg at «Random Forest»-algoritmen, har den mest presise prediksjonen.

Figur 42 presenterer prediksjonsresultater med multipel lineær regresjon, for alle bygningene.



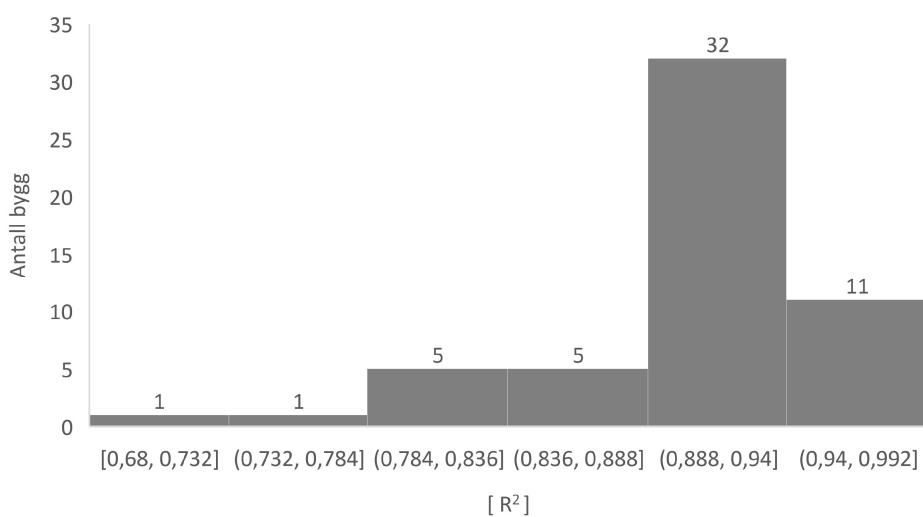
a)



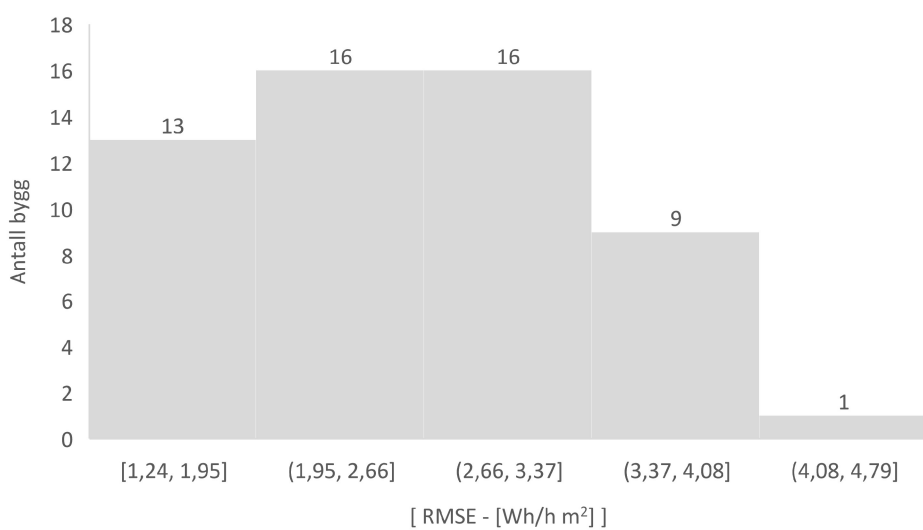
b)

Figur 42 Prediksjonsresultater med multipel lineær regresjon-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE

Figur 43 presenterer prediksjonsresultater med «Kernel Ridge»-algoritmen, for alle bygningene.



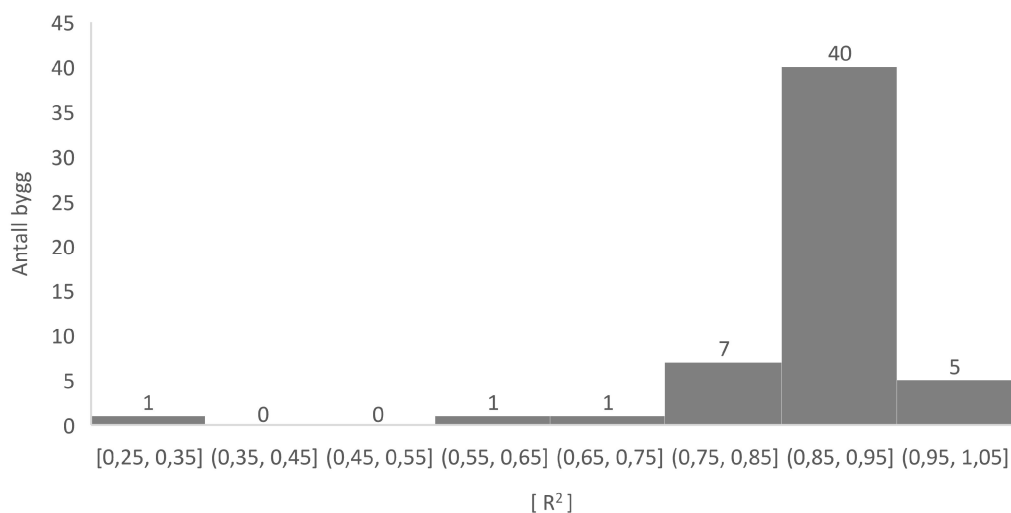
a)



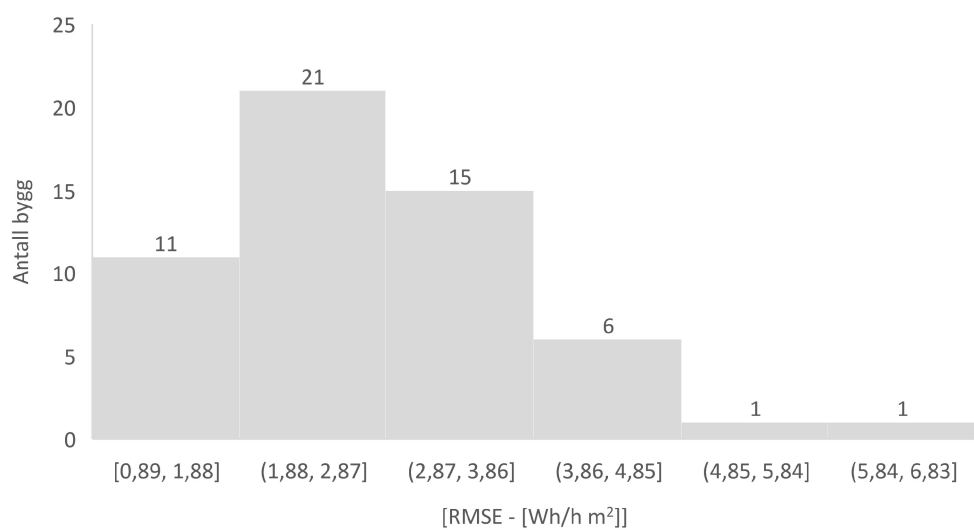
b)

Figur 43 Prediksjonsresultater med «Kernel Ridge»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE

Figur 44 presenterer prediksjonsresultater med «Decision Tree»-algoritmen, for alle bygningene.



a)

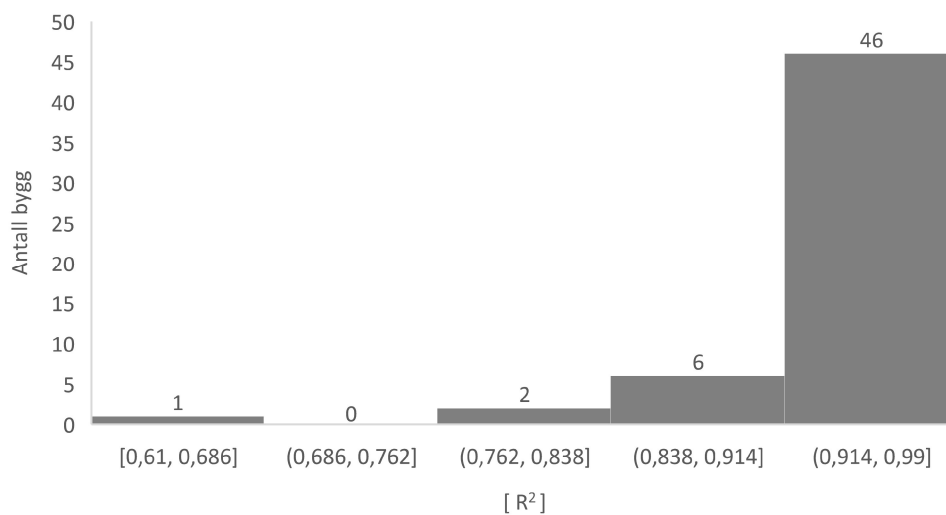


b)

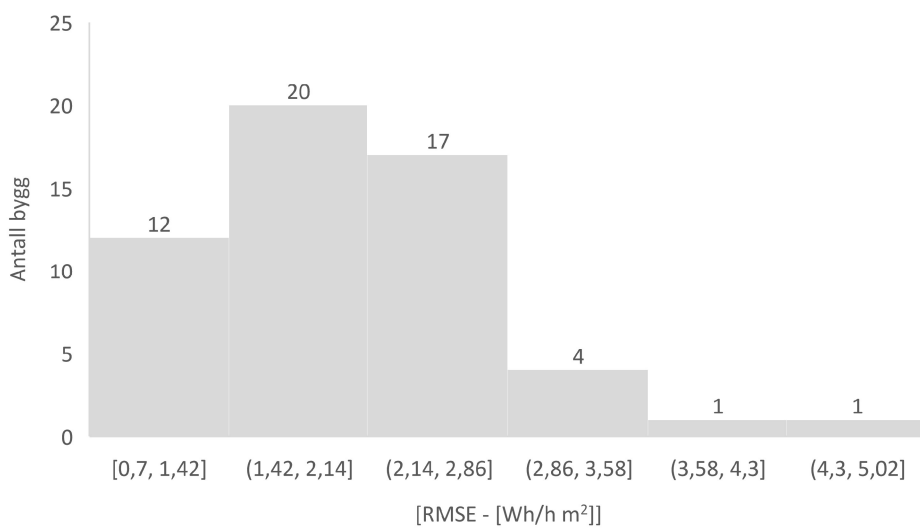
Figur 44 Prediksjonsresultater med «Decision Tree»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE



Figur 45 presenterer prediksjonsresultater med «Random Forest»-algoritmen, for alle bygningene.



a)

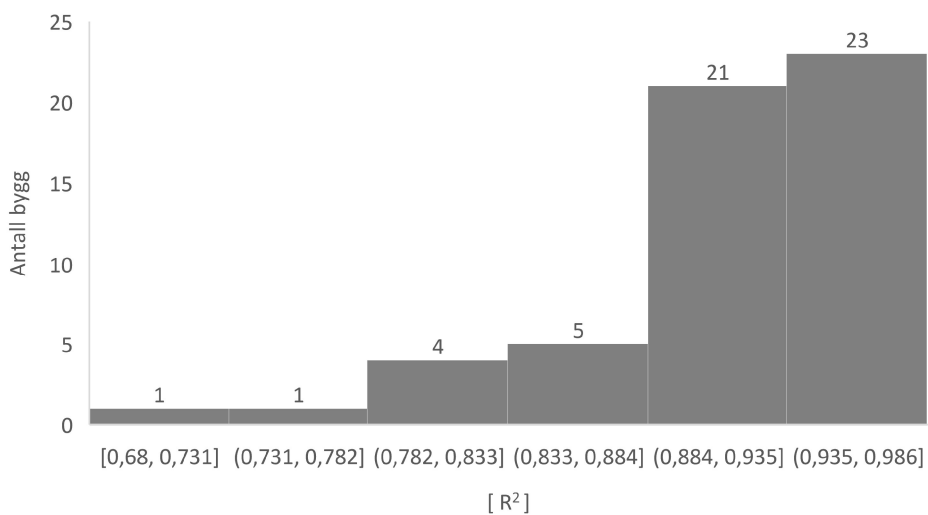


b)

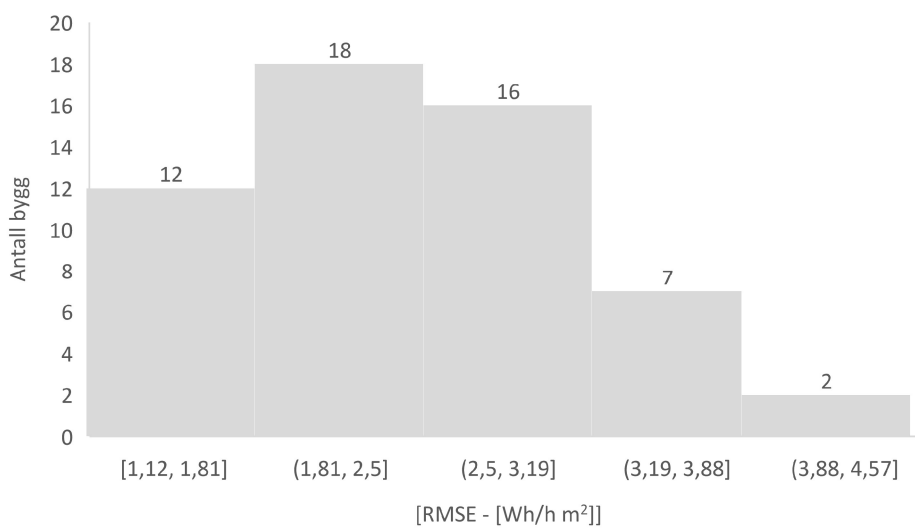
Figur 45 Prediksjonsresultater med «Random Forest»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE



Figur 46 presenterer prediksjonsresultater med «Gradient Boost»-algoritmen, for alle bygningene.



a)

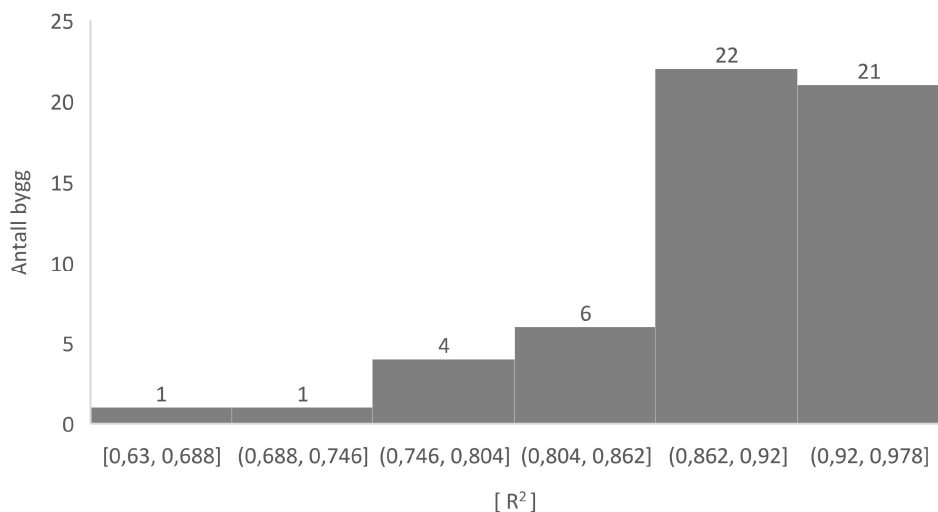


b)

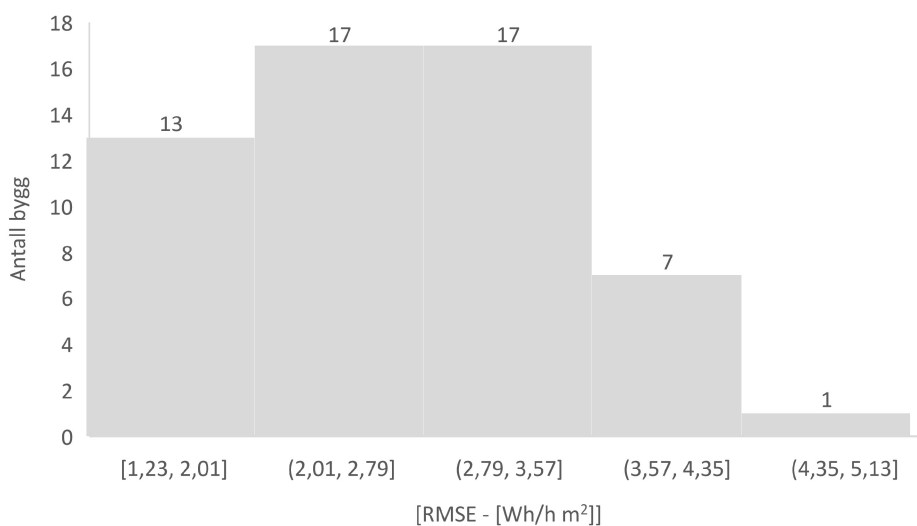
Figur 46 Prediksjonsresultater med «Gradient Boost»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE



Figur 47 presenterer prediksjonsresultater med «Support Vector Regression»-algoritmen, for alle bygningene.



a)



b)

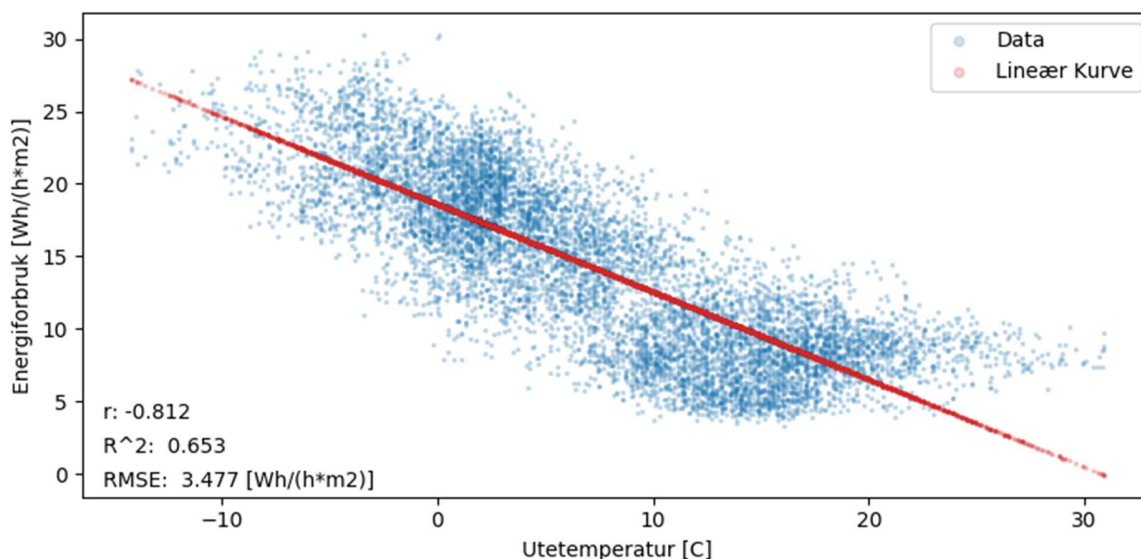
Figur 47 Prediksjonsresultater med «Support Vector Regression»-algoritmen, for alle bygningene. Kvaliteten oppgitt med a) R^2 og b) RMSE

4.5 PREDIKSJON – EN TILFELDIG VALGT BYGNING

For å vise grafisk sammenligningen mellom ET-kurve og prediksjonsmodell basert på maskinlæring, presenteres resultater for en tilfeldig valgt bygning. Multippel lineær regresjon trekkes frem siden den baserer seg på samme algoritme som ET-kurve. Da belyses tydelig virkningen av å legge til flere variabler. I tillegg ønskes det å presentere «Random Forest» nærmere, fordi den har den mest presise prediksjonen, og belyser virkningen av en mer avansert algoritme.

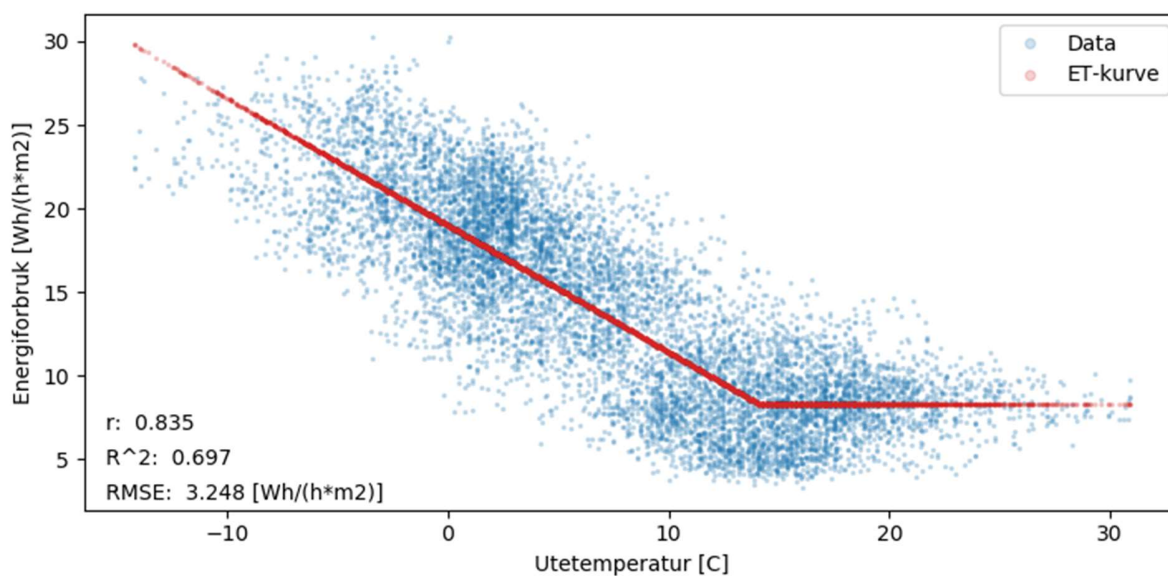
4.5.1 ET-KURVE

Figur 48 viser en enkel lineær regresjonsanalyse av sammenhengen mellom energi og utetemperatur for en tilfeldig valgt bygning i oppgaven. Det er tydelig en lineær sammenheng mellom energiforbruket og utetemperaturen. Den samme korrelasjonsverdien r , kan observeres i Figur 38, som beskriver korrelasjonen mellom energiforbruket og utetemperaturen.



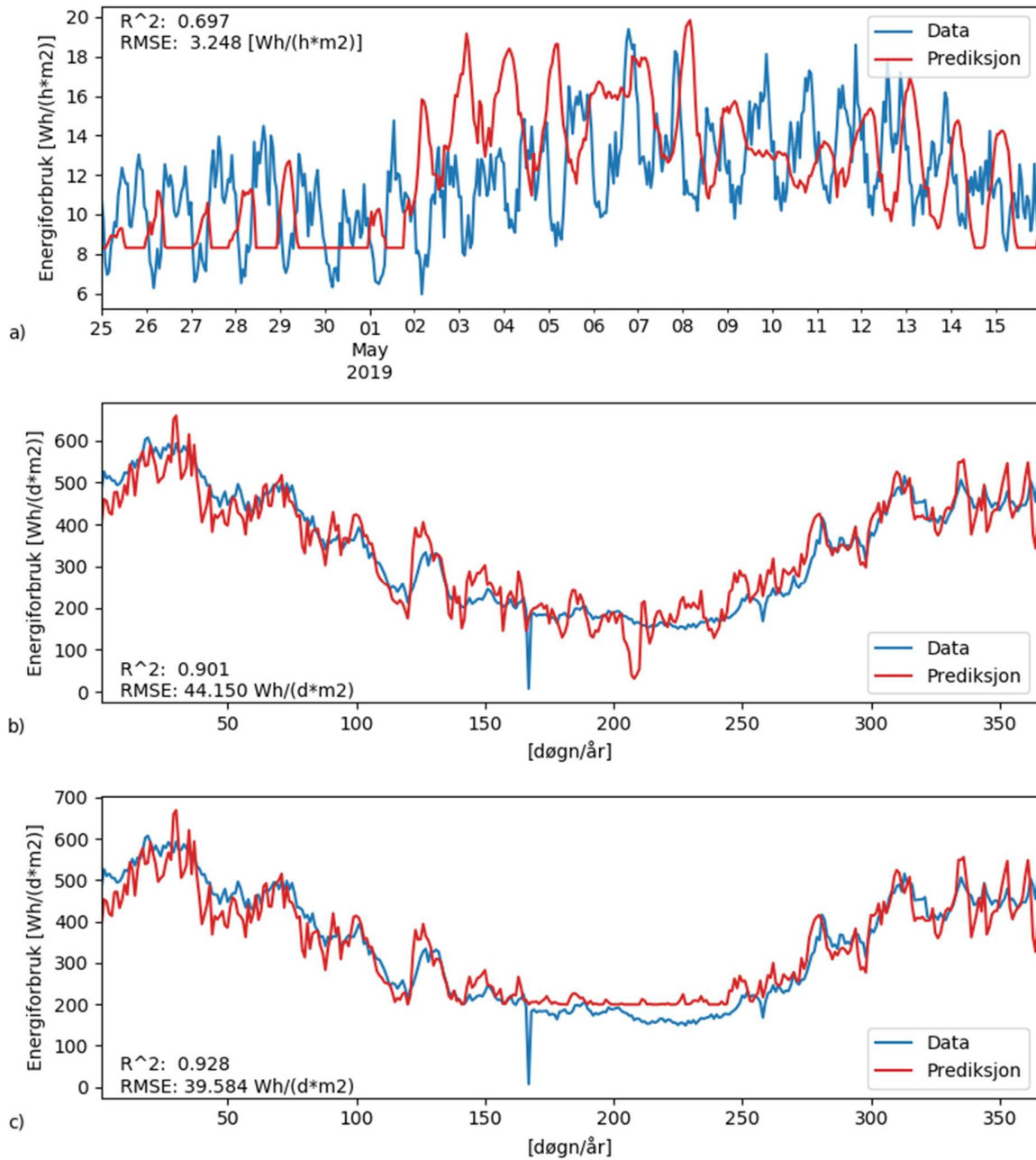
Figur 48 Prediksjon med lineær regresjon, uten knekkpunkt

Ved å introdusere en todelt lineær funksjon med knekkpunkt ved utetemperatur som krever energi til oppvarming forbedres denne prediksjonsmodellen og for form som en tradisjonell ET-kurve. Legg merke til at avviket som oppstår ved prediksjon er nesten uendret, i forhold til prediksjon som er beskrevet Figur 49.



Figur 49 Prediksjon med tradisjonell ET-kurve

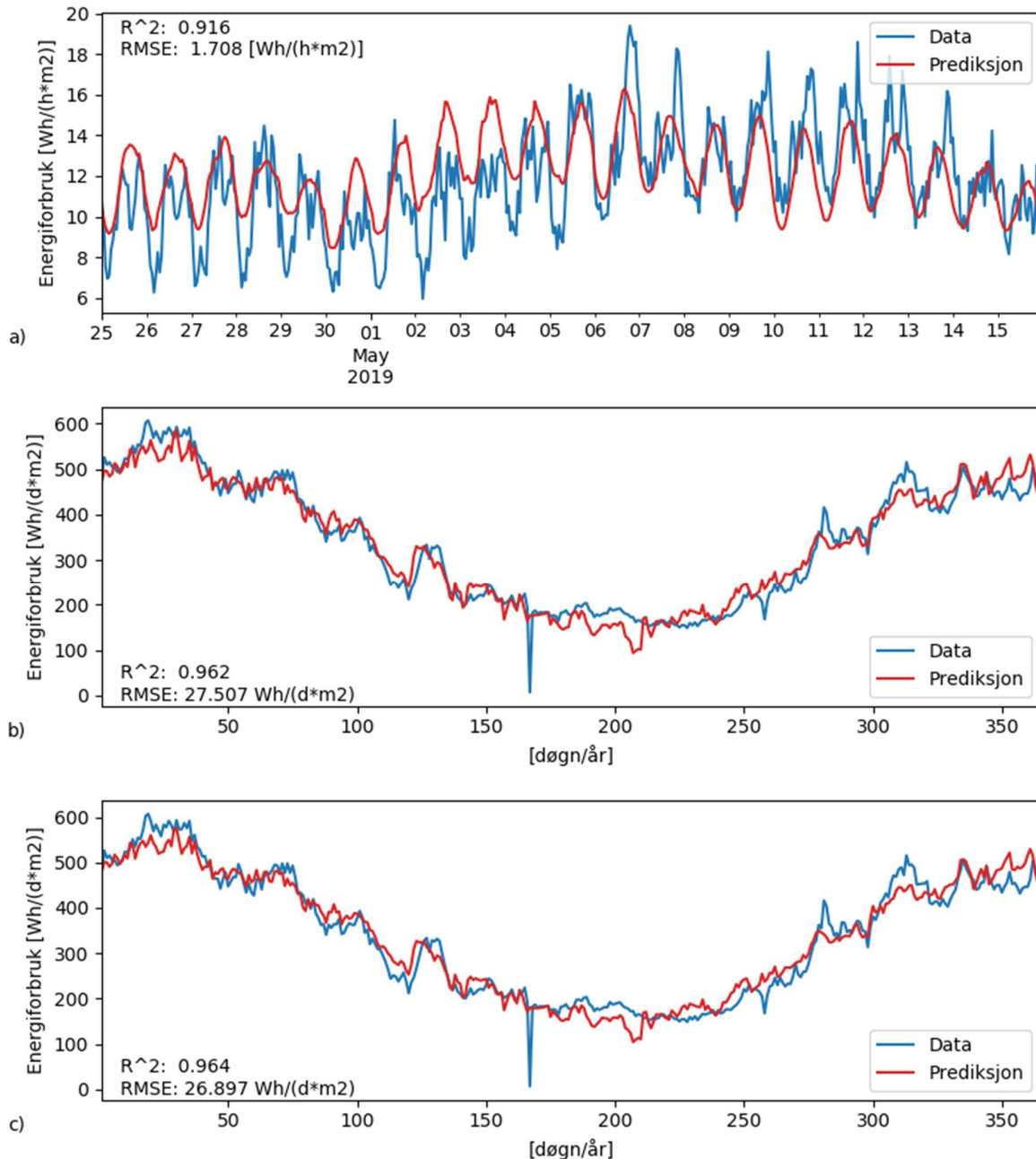
Figur 50 a) viser prediksjonsresultater som vist i Figur 49, men med tid på x-aksen istedenfor utetemperatur. Det er interessant å legge merke til at prediksjonen ikke faller under 8 W/m^2 , men isteden har en konstant verdi her. Dette tilsvarer den temperaturuavhengige delen ET-kurven. R^2 -verdien for døgnverdier kan observeres til å være 0,90 i b), som er en betydelig forbedring sammenlignet med prediksjonsresultater for timesverdier. I b) er døgnverdier beregnet før modellen tilpasses. Modellen kan også tilpasses timesverdiene først som i a) for å så beregne døgnverdier. Dette er vist i c), men denne rekkefølgen viser å ha liten betydning her.



Figur 50 Prediksjonsresultater for a) timesverdier, b) døgnverdier og c) døgnverdier basert på timesverdier med ET-regressor

4.5.2 MULTIPPEL LINEÆR REGRESJON – ALGORITMEN

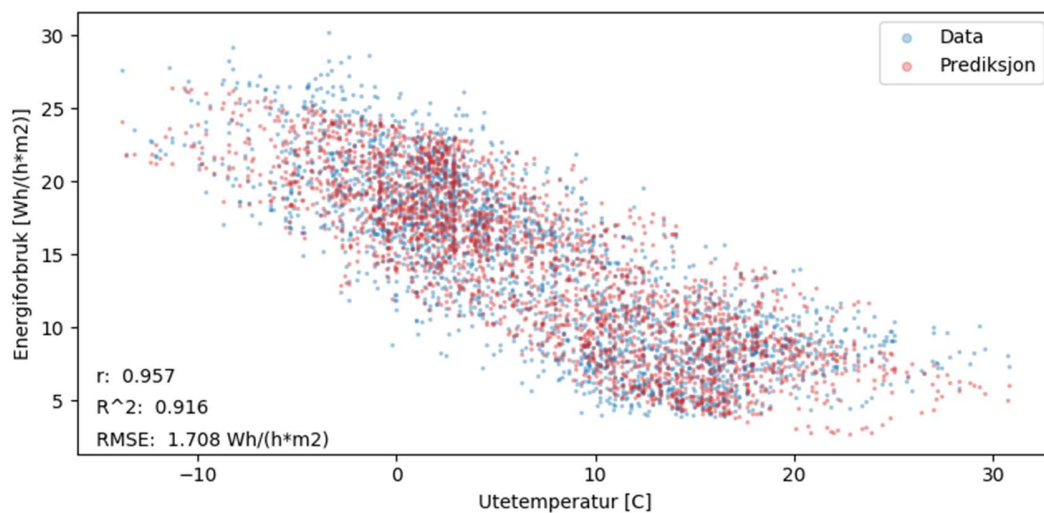
Figur 51 presenterer prediksjonsresultater av a) timesverdier, b) og c) døgnerverdier, for multipel lineær regresjon. Legg merke til at denne prediksjonsmodellen baserer seg på flere variabler ved prediksjon av energiforbruket. Sammenlignet med ET-kurven i Figur 50 som hadde mindre presis prediksjon for timesverdier, er det tydelig at denne modellen er mer presis. Blant annet har RMSE-verdien blitt redusert fra 39,6 Wh/(d*m²) i Figur 50 til 26,9 ved denne modellen, som er en merkbar forbedring.



Figur 51 Prediksjonsresultater for a) timesverdier, b) døgnerverdier og c) døgnerverdier basert på timesverdier med multipel lineær regresjon

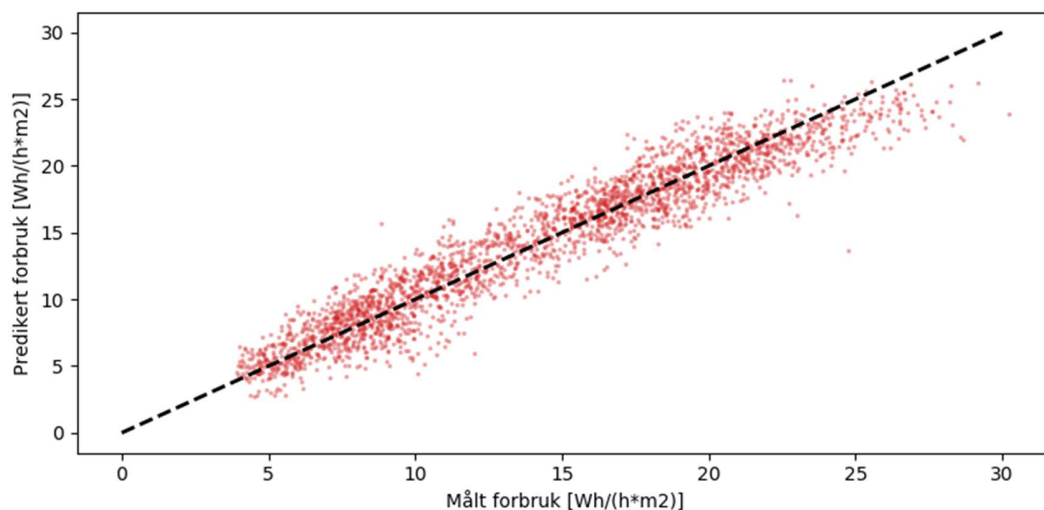


Siden prediksjonen blir flerdimensjonal, er det ikke lenger mulig å se noe fast mønster i ET-grafen. Dette vises i Figur 52. Man kan fremdeles få en indikasjon på hvor godt prediksjonen treffer, og det kan se ut som prediksjonen ikke treffer så bra når utetemperaturen er enten veldig høy, eller veldig lav.



Figur 52 ET-graf med multippel lineær regresjon. Sammenligning av predikerte og de faktiske verdiene.

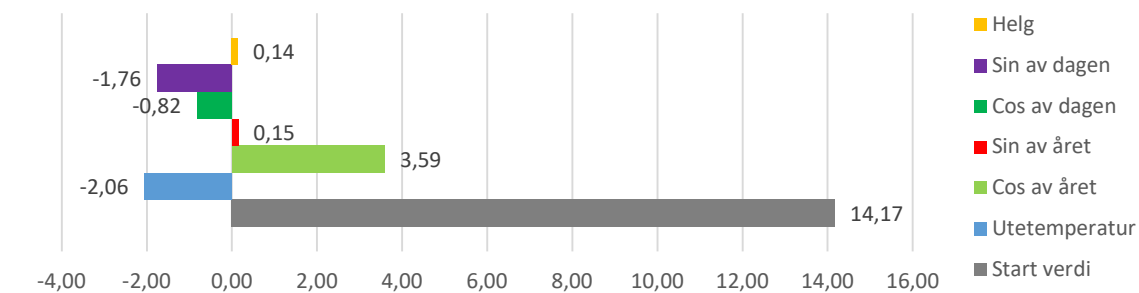
Ved å studere Figur 53, kan man se at dette kan stemme for høyt forbruk, altså lav temperatur, men ikke for lavt forbruk. Dette er en graf som viser målt forbruk mot predikert forbruk, og som bedre beskriver presisjonen av prediksjonen.



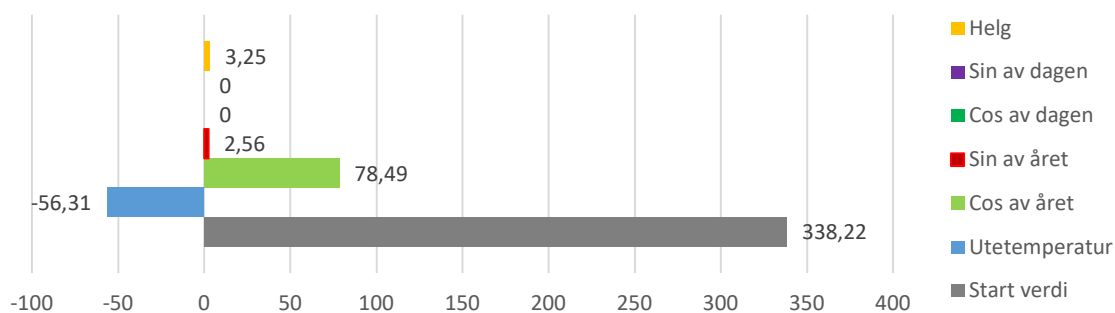
Figur 53 Forholdet mellom målt- og predikert energiforbruk. For multippel lineær regresjon



Figur 54 fremlegger koeffisientene til de ulike variablene for multippel lineær regresjon. Verdiene beskriver i hvilken grad variablene påvirker prediksjonen. Startveriden er den verdien prediksjonen får dersom alle variablene er null. Cos av året er den variabelen som påvirker prediksjonen mest. Variablene som kun beskriver døgnavariasjonen, faller naturligvis bort for b) døgnerverdier. Koeffisientene er ellers ganske likt fordelt for a) timesverdier og døgnerverdier.



a)

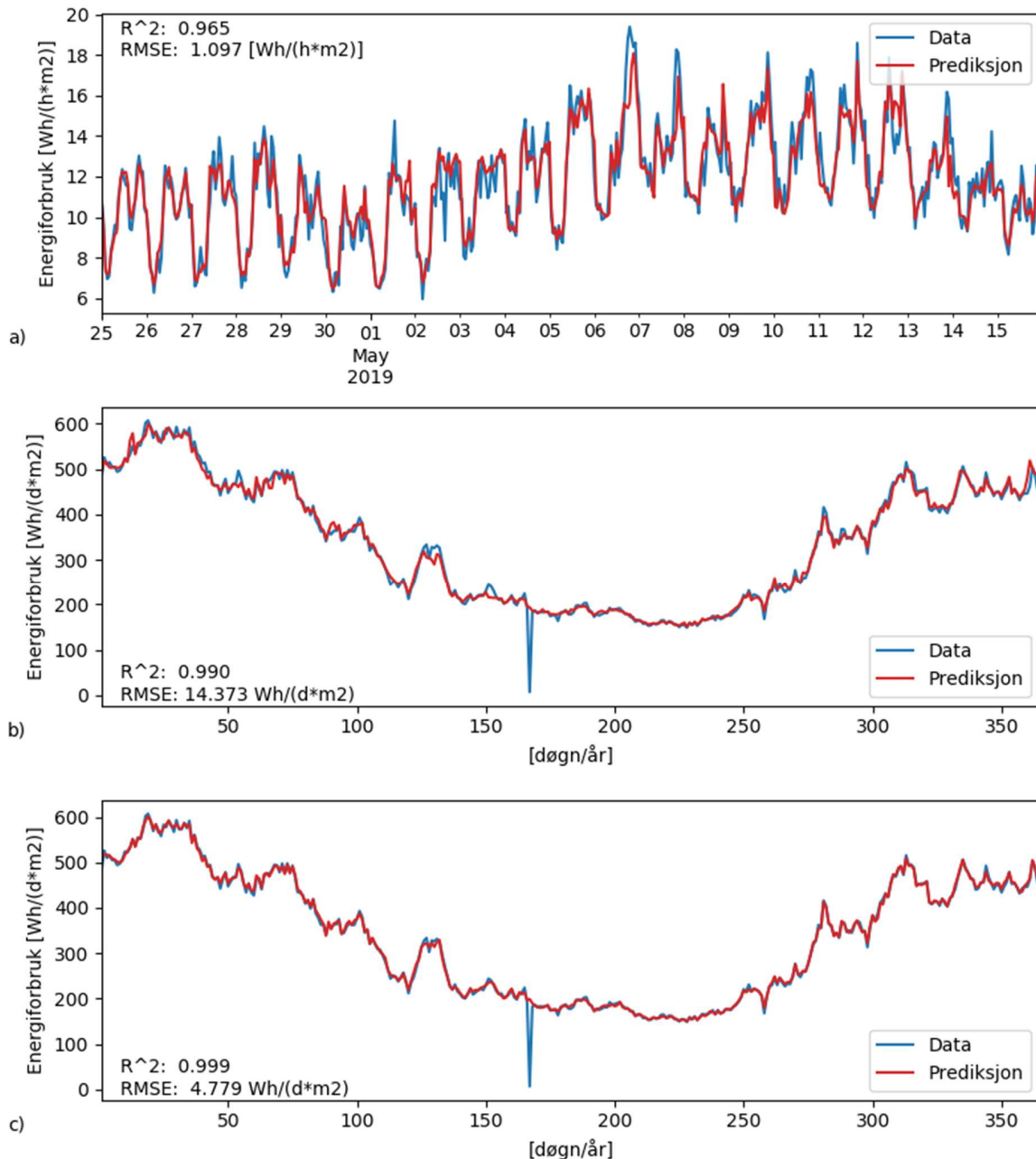


b)

Figur 54 Koeffisienten til de ulike variablene for a) timesverdier og b) døgnerverdier, til multippel lineær regresjon

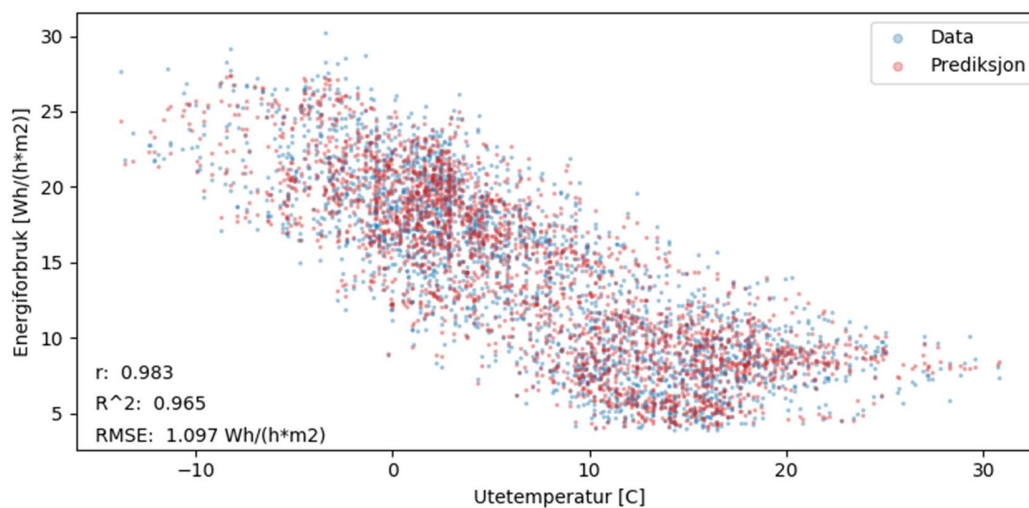
4.5.3 «RANDOM FOREST» - ALGORITMEN

Figur 55 presenterer prediksjonsresultater av a) timesverdier, b) og c) døgnerverdier for «Random Forest»-algoritmen. Man kan med en gang se av grafen at «Random Forest» predikerer mye mer presist sammenlignet med både ET-kurven og multipl lineær regresjon. Det viser seg at «Random Forest» predikerer c) døgnerverdier med en R^2 på 0,999. RMSE-verdien ligger på 4,8 Wh/(d*m²). Det er en betydelig forbedring sammenlignet med ET-kurve og multipl lineær regresjon.



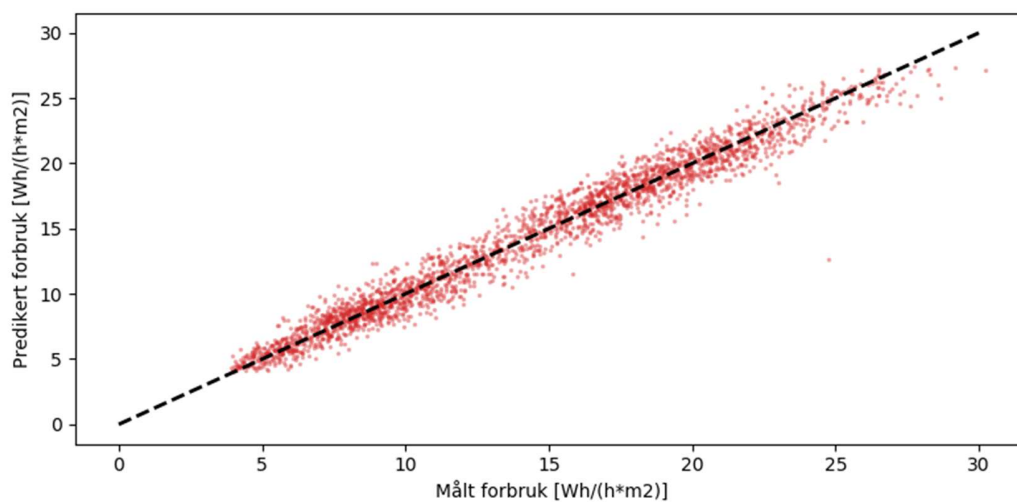
Figur 55 Prediksjonsresultater for a) timesverdier, b) døgnerverdier og c) døgnerverdier basert på timesverdier med «Random Forest»

I Figur 56 kan det heller ikke observeres noe fast mønster ved prediksjon med «Random Forest», slik som for multipel lineær regresjon.



Figur 56 ET-graf med «Random Forest». Sammenligning av predikerte og de faktiske verdiene.

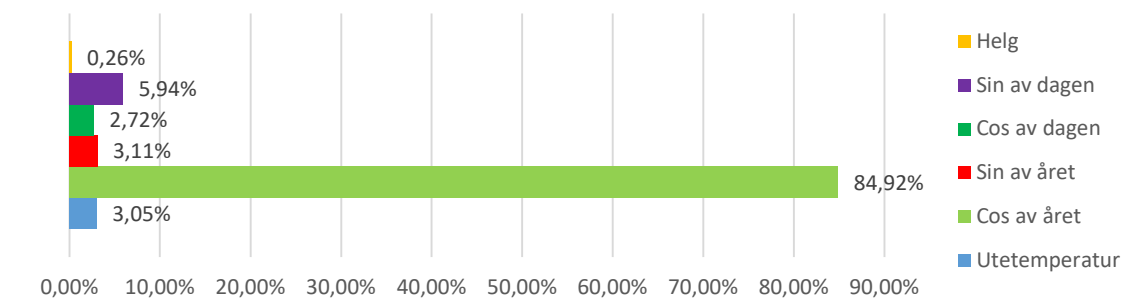
Figur 57 viser at flere av punktene ligger nær den stiplede linjen. Dette er da en annen måte vise at «Random Forest» predikerer bedre sammenlignet med ET-kurve og multipel lineær regresjon.



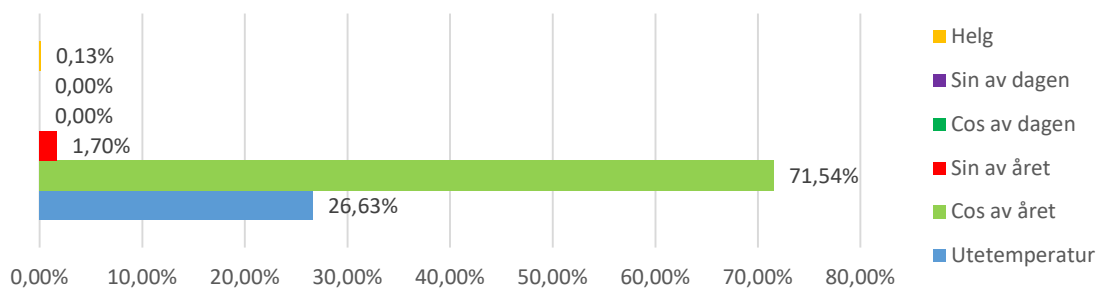
Figur 57 Forholdet mellom målt- og predikert energiforbruk. For «Random Forest»



Figur 58 fremlegger viktigheten til de ulike variablene i prediksjonen med Random Forest. For denne modellen har ikke variablene koeffisienter slik som for multippel lineær regresjon. De ulike variablene får isteden en prosentsats som skal beskrive i hvilken grad de påvirker prediksjonen. Det viser seg at prediksjonen baserer seg i veldig stor grad på cos av året. For b) døgnerverdier får utetemperaturen litt mer å si, i forhold til a) timesverdier.



a)



b)

Figur 58 Koeffisienten til de ulike variablene for a) timesverdier og b) døgnerverdier, til «Random Forest».



5 DISKUSJON

5.1 DATABEHANDLING

Databehandling endte opp med å bli en stor og viktig del av denne oppgaven. Kritisk gjennomgang av all data var viktig for å lage et homogent datasett og sikre et godt sammenligningsgrunnlag for alle bygningene. Det at bygningene f.eks. har ulike oppvarmingssystemer går utover kontinuiteten i dataene. Det er blitt observert at bygninger med ulike oppvarmingssystemer ender opp med ulike målinger. En bygning kan ha romoppvarmingen i målingene for «EL leiligheter», men en annen bygning i målingene for «EL fellesarealer». Dette gjør det vanskelig å danne et felles sammenligningsgrunnlag for alle bygningene. Løsningen ble å avgrense dataene til to oppvarmingssystemene. Altså bygninger som dekker oppvarming med elektrisitet, og bygninger som dekker oppvarmingen med fjernvarme.

Det ble observert at noen bygninger hadde arealer som ikke stemte med forbruket. Dette førte til enten veldig høyt, eller veldig lavt forbruk pr. kvadratmeter. På grunn av dette ble det foretatt en kontroll av alle arealene. I denne analysen kunne arealdataene innhentes fra 3 ulike kilder; Energinett, ekstern Excel-fil mottatt fra ÅF og enøkkanalysene for hvert bygg. Både for å håndtere med riktig areal og for å ha gjennomført en slags arealkontroll ble alle arealdataene fra disse kildene kontrollert opp mot hverandre. Dette resulterte i at å fjerne alle bygg som hadde usikker arealdata, og bygg med næringsareal mer enn 10%. Bygg med næringsarealer mer enn 10% ville forstyrre det totale energiforbruket, og forhindre videre analyse av forbruksmønsteret for selve leilighetene.

Et annet problem som oppsto, var at noen av bygningene delte målinger. Dette var typisk for rekkeblokker og blokker som var bygget sammen i kvartaler, men hadde hver sin adresse. En varmtvannsbereder kunne da stå i kjelleren i en bygning og distribuere varmtvann til resten av kvartalet. Denne bygningen ville da få veldig høye målinger for «EL fellesareal», men de resterende bygningene veldig lite. Dette kunne ha blitt løst ved å f.eks. summere målingene for alle bygningene i kvartalet eller rekken. Siden det ble utfordrende å finne en praktisk løsning for disse tilfellene og det oppsto usikkerhet om akkurat hvilke bygninger det gjaldt, ble det konkludert at disse bygningene skulle utelates fra videre analyse.



5.2 ANALYSE

I henhold til energirammekravet i teknisk forskrift (TEK-17) §14-2 skal det totale netto energibehovet for boligblokker ikke overstige 95 kWh/m^2 per år. Dette er hovedsakelig summen av varmebehovet og interne laster i form av teknisk utstyr, belysning, pumper osv. Sammenlignet med resultater i Figur 22, viser det seg at alle bygninger bortsett fra én ligger over dette kravet. Dette tyder på at energitapet gjennom bygningskroppen til disse bygningene ikke tilfredsstiller dagens krav i TEK17, noe som var forventet. Samtidig er det viktig å poengtere forskjellen mellom energiforbruk og energibehov i denne sammenligningen. Systemvirkningsgraden i energiforsyningssystemet blir sett bort fra i beregningen av energibehovet.

Effekten av ulik systemvirkningsgrad i de ulike energiforsyningssystemene kan nærmere observeres ved å sammenligne Figur 25 og Figur 28. Det viser seg at bygninger med fjernvarme har et høyere energiforbruk enn bygninger som dekker varmebehovet med elektrisitet. I henhold til NS3031 tabell B.10 vil systemtapet for fjernvarme tilsvare 14%, mens for elektriske systemer som panelovner vil dette være lik 2%. Dette betyr i prinsippet at bygninger med fjernvarme må regne med et større behov for levert energi, noe som stemmer i så fall med dataene i denne oppgaven. Det er viktig å påpeke at i praksis kan levert energi fastsettes ved å bruke målt energiforbruk. NS-EN 15603 gir nærmere retningslinjer om dette, men det er utenfor avgrensningene i denne oppgaven. Legg merke til at målt levert energi vil være avhengig av bygningens reelle bruk, og vil derfor avvike fra beregnet levert energi som kun er basert på forutsetninger om ulike forhold i henhold til NS3031.

Balansetemperaturen tilsvarer knekkpunktet i en ET-kurve og er en verdi som forsøker å skille ut den delen som er temperaturavhengig. Den kan derfor gi en indikasjon på ved hvilken utetemperatur de ulike bygningene starter oppvarming om høsten. Teorien sier at i Norge starter dette vanligvis når døgnmiddeltemperaturen synker under 11 C° . I Figur 24 a) fremkommer det at alle bygningene i denne analysen starter fyringssesongen tidligere enn det teorien tilsier. Dette tyder på at disse bygningene har et stort behov for energitilførsel. Det må også tas høyde for at metoden brukt til å gi denne indikasjonen er et grovt estimat. Det er ikke tegn på noen grupperinger blant bygningene for dette estimatet. Fordelingen er normalfordelt.

Stigningstallet er en annen verdi som kan hentes ut av en ET-kurve. Denne beskriver hvor mye energiforbruket endres når utetemperaturen endres. Dette gjelder da for den temperaturavhengige delen i kurven. Energitapet gjennom bygningskroppen i form av transmisjon, infiltrasjon og ventilasjon er en viktig faktor som kan påvirke dette stigningstallet. I tillegg vil brukeratferd, interne varmetilskudd samt plassering med tanke på solinnstråling være andre faktorer som kan påvirke dette. Denne sammenhengen kan i midlertidig være komplisert, som gjør det vanskelig å trekke noe spesifikk konklusjon, men jo dårligere bygningskropp desto brattere stigningstall. Av Figur 24 b) kan det tyde på en gruppering i bygningene med hensyn på dette stigningstallet. Det viser seg at bygninger med fjernvarme har generelt høyere stigningstall enn bygninger som er oppvarmet med elektrisitet, ved å sammenligne resultatene fra Figur 27 b) med Figur 30 b). Dette tyder på at bygninger med fjernvarme kan ha dårligere bygningskropp enn bygninger med elektrisitet. Grunnen til dette henger sammen med at bygninger med fjernvarme også hadde høyere energiforbruk.



Grunnlasten skal være et estimat av alt energiforbruk i bygget som er temperatuavhengig. Dette kan tilsvare den flate delen i ET-kurven. I energibalansen tilsvarer dette oppvarming av tappevann og andre brukerspesifikke forhold, altså «annet». Av Figur 31 presenteres estimerte verdier av varmtvannsbehov og «annet». Bygninger med elektrisk oppvarming har litt høyere varmtvannsbehov, men litt lavere «annet» sammenlignet med bygninger med fjernvarme. Dette kan skyldes beregningsmetodikken, hvor det forklares at en liten del av «annet» ligger i varmtvannsestimatet for bygninger med elektrisk oppvarming. Bygninger med fjernvarme har derfor en mer troverdig fordeling av disse energipostene. Varmtvann og «annet» summert (grunnlasten) blir likt uavhengig av energiforsyningsystem. Dette kommer også frem av Figur 23 b) hvor det vises at grunnlasten er normalfordelt for alle bygninger. Generelt sett ligger alle bygninger i snitt over TEK17 kravet for både varmtvannsbehov og annet. Energiforbruk til oppvarming av tappevann utgjør generelt sett en liten andel av det totale energiforbruket i bygninger.

I Figur 31 kan det observeres at bygninger med fjernvarme har et tydelig høyt oppvarmingsbehov sammenlignet både med bygninger med elektrisk oppvarming og TEK17. Tidligere er det nevnt en forskjell mellom bygninger med fjernvarme og bygninger med elektrisk oppvarming med hensyn på både stigningstall og det totale energiforbruket. Forklaringen til forskjellen ligger i denne energiposten. Akkurat hvorfor bygninger med fjernvarme har så mye høyere oppvarmingsbehov er litt uklar. Forskjellen i systemvirkningsgrad kan ikke være forklaringen alene. I metodikken forklares det at målingene for «fjernvarme» gjelder hele bygninger, inklusiv fellesareal og næring, mens målingene for «EL leiligheter» gjelder kun for leilighetsarealet. Dette, sammen med ulik systemvirkningsgrad, kan være en forklaring på den store forskjellen. Uansett ligger alle bygninger godt over TEK17 kravet for oppvarmingsbehov. I tillegg er det observert en stor varians i estimatene av oppvarming for bygninger med fjernvarme. Årsaken til dette kan være, som beskrevet i metoden, at «Fjernvarme» målingene ligger i faste intervaller på 10 kWh/h.

Oppvarmingsbehovet har en sammenheng med graddagstallet og varmetapstallet til bygget. De viktigste faktorene som kan påvirke varmetapstallet er blant annet varmetap i form av transmisjon og infiltrasjon. Dette betyr at energitiltak som kan redusere varmetapstallet vil være et riktig sted å begynne for disse bygningene. Dette er også noe som samsvarer med det første energibesparende tiltaket i henhold til Kyotopyramiden.



5.3 PREDIKSJONMODELLEN

Det er i hovedsak to elementer som er med på å avgjøre kvaliteten til prediksjonsmodellen; hvor godt variablene beskriver variasjonene i energiforbruket, og kompleksiteten til algoritmen. I denne oppgaven er det avgrenset til å ikke gå dypere inn i algoritmene, kun vise hvilke resultater som kan forventes. Derfor belyses isteden hvilke variabler som kan bidra i å beskrive variasjonen i energiforbruket.

5.3.1 VARIABLENE

Korrelasjonen mellom energiforbruket og de ulike variablene brukt i prediksjonsmodellen er presentert i kapittel 4.3.1. Her kan variansen i korrelasjonen for alle bygningene observeres. Generelt sett oppstår den samme korrelasjonen for de fleste bygninger. Videre diskuteres ulike variabler for en tilfeldig valgt bygning for å bedre kunne forklare poengene som belyses.

I en prediksjonsmodell basert på maskinlæring har variablene en viktig rolle. De skal forsøke å beskrive variasjonen i energiforbruket best mulig. Det er i hovedsak to variasjonsmønstre i energiforbruket det forsøkes å beskrive, variasjonen over året, og variasjonen over døgnet. Den totale variasjonen er når disse beskrives sammen. I Figur 38 vises sammenhengen mellom energiforbruket og utetemperaturen for en tilfeldig valgt bygning med timesverdier. Korrelasjonen er negativ og ligger på $-0,81$. Korrelasjonsresultatene mellom energiforbruket og utetemperaturen for et skolebygg i Oslo, viste seg å være lik $-0,51$. Dette tyder på at energiforbruk i boligblokk har en høyere korrelasjon med utetemperaturen, enn skolebygg. Dette kan indikere at oppvarmingsbehovet for skolebygninger generelt sett ikke er en like stor andel av det totale energiforbruket.

Videre i samme figur at kvadratet av korrelasjonen er $0,65$. Dette betyr at 65% av den totale variasjonen i energiforbruket kan «forklares» med utetemperaturen ved at dette er fellesvarians. De resterende 35% av variasjonen skyldes derfor andre termiske forhold som vind og solinnstråling, men også brukeratferd. I tillegg fremlegges tilsvarende resultat mellom energiforbruket og en cosinuskurve som varierer over året (\cos av året). Denne kan på samme måte «forklare» 82% av den totale variasjonen i energiforbruket. Sammenlignet med utetemperaturen er dette da allerede en bedre modell til å beskrive energiforbruket med timesverdier.

Ved døgnerverdier summeres alle timesverdier for døgnet. Da får man en beskrivelse av variasjonen over året uten forstyrrelsen som forekommer i variasjonen over døgnet. Sammenlignes resultatene i Figur 38 med Figur 39, kan det observeres at korrelasjonen mellom energiforbruket og utetemperaturen har økt. r^2 -verdien har økt med 24% sammenlignet med timesverdier. Dette er en betydelig økning og en indikasjon på at dette ikke bare er på grunn av «støy». Cosinuskurven (\cos av året) har også hatt en økning i r^2 -verdien sammenlignet med timesverdier. Denne økningen er på kun 10% . Økning kan man si er en effekt som oppstår etter fjerning av støy siden det er visst at cosinuskurven ikke har noe korrelasjon med variasjonen i energiforbruket over døgnet. Sammenligner vi disse økningene av r^2 -verdiene har utetemperaturen en økning på over det dobbelte. Årsaken til denne høye økningen kan forklares ved å se nærmere på variasjonen i energiforbruket over døgnet.



Med døgnverdier blir variasjonen i energiforbruket over døgnet fjernet, og man får en beskrivelse av variasjonen over året. En tilsvarende beskrivelse av variasjonen over døgnet kan oppnås ved å subtrahere den gjennomsnittlige timesverdien over døgnet fra tilhørende timesverdi. I Figur 40 kommer det frem at r^2 -verdien mellom utetemperatur og energiforbruket for døgnvariasjon ligger på rundt 0,16 for den samme tilfeldig valgte bygningen. Dette er mye lavere sammenlignet med timesverdier og døgnverdier. En annen viktig observasjon er at fortegnet endres for korrelasjonen. Korrelasjonen er her positiv, mens for døgnverdier var den negativ. Energiforbruket øker ved økende utetemperatur istedenfor å avta, som er den naturlige sammenheng. Energiforbruket og utetemperaturen har da motstridende korrelasjon for henholdsvis variasjon over året og døgnet. Det forklarer den kraftige økningen av r^2 -verdien når variasjonen over døgnet fjernes.

Det har oppstått spekulasjoner rundt hvorfor variasjonen i energiforbruket over døgnet og utetemperaturen får denne korrelasjonen. En av spekulasjonene var om termisk treghet kunne være årsaken. Dette er en effekt som oppstår ved at energi lagres i bygningskroppen under oppvarming. Så når utetemperaturen synker i løp av natta oppstår det en treghet i nedkjølingen av bygget. Den samme treghet gjelder også for oppvarmingen av bygget. Dette fører til at energiforbruket ikke har en direkte sammenheng med utetemperaturen, men er forskjøvet tilsvarende den termiske tregheten. En annen forklaring kan være effekten andre termiske forhold som vind og solinnstråling, som kan påvirke energiforbruket selv om utetemperaturen ikke endres. Men den mest sannsynlige årsaken til dette er brukeradferd. Det brukes mest energi om dagen når folk er våkne som er samtidig med at utetemperaturen er høy. Om natten brukes det lite energi samtidig som utetemperaturen er lav.

Ved å manipulere dataene på denne måten og skille mellom de ulike mønstrene, får man en bedre forståelse av hvorfor utetemperaturen fungerer dårlig til å beskrive den totale variasjonen i energiforbruket. Utfordringen i utviklingen av prediksjonsmodellen er å beskrive variasjonen i energiforbruket over døgnet. Amplituden i denne variasjonen er liten og dermed mer avhengig av andre forhold enn utetemperaturen alene. Variasjonsmønsteret bli derfor mer komplekst. Figur 40 viser r^2 -verdi for døgnvariasjon for en sinuskurve som varierer med døgnet (sin av dagen). Denne variabelen kan «forklare» 58% av variasjonen i energiforbruket over døgnet for det tilfeldig valgte bygningen, og er en viktig variabel i prediksjonsmodellen.

5.3.2 PREDIKSJON MED ET-KURVE

Prediksjon med tradisjonell ET-kurve benytter kun en variabel, nemlig utetemperaturen. Det har blitt nevnt at utetemperaturen kan «forklare» 65% av den totale variasjonen i energiforbruket. Denne beskrivelsen av variasjonen kan oppnås med en enkel lineær funksjon vist i Figur 48. Selv om det er en tydelig lineær sammenheng, kan man se at dette ikke stemmer hele veien. Over en viss utetemperatur er ikke lenger energiforbruket temperaturavhengig, og kan derfor ikke beskrives med utetemperaturen. Målet med en mer presis modell vil da være å beskrive de resterende 35% prosentene som ikke kan beskrives med utetemperaturen alene. Måten ET-kurven løser dette på er ved å legge inn en knekk der hvor energiforbruket ikke lenger kan beskrives av utetemperaturen. Dette illustreres i Figur 49, og det viser seg at denne avanseringen har liten effekt sammenlignet med den enkle lineære funksjonen.

I Figur 50 a) vises hvordan prediksjon gjort med ET-kurve for timesverdier ser ut for en tilfeldig valgt bygning. Det kan se ut som prediksjonen fanger opp noe av døgnvariasjonen, men predikerer motsatt.



Dette henger sammen med at korrelasjonen mellom energiforbruket og utetemperaturen er negativ for variasjonen over året, men positiv for variasjonen over døgnet. Denne modellen operer med kun en variabel og derfor kun en korrelasjon. Ved å operere med døgnerverdier som vist i Figur 50 b) og c) kan prediksjonen derfor bli betydelig bedre. ET-kurven kan «forklare» hele 93% av variasjonen i energiforbruket. Dette er en betydelig forbedring i forhold til å operere med timesverdier. Med en ET-kurve kan man derfor med fordel operere med døgnerverdier. Dette stemmer med annen forskning gjort på ET-kurver. Som beskrevet i teorien vil ulike timesoppløsninger påvirke resultatene og dette avhenger i stor grad av bruksmønstre.

5.3.3 MULTIPPEL LINEÆR REGRESJON

Multipel lineær regresjon er den enkleste av maskinlæringsalgoritmene og benytter samme algoritme som ET-kurve og enkel lineær regresjon, nemlig lineær regresjon. Fordelen med multipel lineær regresjon er at den kan basere seg på flere variabler ved prediksjon av energiforbruket. Disse variablene er sinus- og cosinusfunksjoner som, i tillegg til utetemperaturen, skal bidra i å beskrive de ulike variasjonsmønstrene. Figur 51 viser at prediksjonen med denne modellen er betydelig mer presis enn ET-kurven for timesverdier. Den mest merkbare forskjellen visuelt sett, er at denne modellen kjenner igjen det riktige variasjonsmønsteret over døgnet, som i motsetning til ET-kurven predikerte motsatt. Selv om det fortsatt er en vei igjen å gå, kommer det da tydelig frem her at de andre variablene bidrar stort til å beskrive denne variasjonen. For døgnerverdier er også modellen bedre, men dette kommer ikke like tydelig frem da ET-kurven predikerte forholdvis presist her. Det skal også poengteres at de variablene som forsøker å beskrive variasjonen over døgnet faller bort for denne prediksjonen. Men av Figur 50 c) viser at det uansett er ikke så mye å hente ved å benytte modellen for timesverdier først for å så gjøre om til døgnerverdier.

For å forstå hvordan modellen kommer frem til disse prediksjonene kan Figur 54 benyttes. Her fremlegges koeffisientene til de ulike variablene. Verdien beskriver i hvilken grad variabelen påvirker prediksjonen, og det er korrelasjonen mellom variabelen og energiforbruket som bestemmer koeffisienten. Siden korrelasjonen mellom cosinuskurven og energiforbruket hadde best korrelasjon, får derfor denne variabelen størst koeffisient.



5.3.4 RANDOM FOREST

I denne prediksjonsmodellen benyttes de samme variablene som for multippel lineær regresjon. Forskjellen er at «Random Forest»-algoritmen er mer kompleks. Den benytter seg ikke lengre på lineære sammenhenger, det er en videreutvikling av Decision Tree algoritmen. I denne modellen benyttes flere beslutningstrær og et gjennomsnitt av de individuelle prediksjonene. Dette bidrar blant annet til å fjerne sårbarheten en enkel «Decision Tree» har mot uteliggende data. I Figur 55 a) viser prediksjonen av timesverdier med «Random Forest». Man kan med en gang se av grafen at «Random Forest» predikerer mye mer presist sammenlignet med både ET-kurven og multippel lineær regresjon. R^2 -verdien er her 96,5%. Dette tilsvarer den presisjonen multippel lineær regresjon hadde for prediksjon av døgnverdier, Figur 55 b). Så selv med den hensikt å beholde all informasjonen om de predikerte timesverdiene, predikerer «Random Forest» like bra.

Som for de tidligere modellene er de predikerte timesverdiene gjort om til døgnverdier og sammenlignet med døgnverdiene til energiforbruket. For verken ET-kurven eller multippel lineær regresjon var det ikke stort å hente ved å predikere døgnverdier på denne måten. Dette er ikke tilfelle for «Random Forest». Figur 55 c) viser at ved å predikere døgnverdier på denne måten får man en R^2 -verdi på 0,999. Det betyr at 99,9% av variasjonen i energiforbruket over året kan beskrives med modellen. RMSE-verdien ligger på $4,8 \text{ Wh}/(d \cdot m^2)$. Sammenlignet med ET-kurven som hadde en RMSE-verdi på 40, og multippel lineær regresjon på 27, er dette uten tvil den mest merkbare forbedringen. Med «Random Forest» algoritmen kan man da redusere avviket til prediksjonen for døgnverdier med nesten 90% sammenlignet med ET-kurven. Så uavhengig om man er ute etter timesverdier eller døgnverdier bør det alltid benyttes timesverdier i kombinasjon med denne algoritmen. Dette viser hvordan en mer kompleks modell som «Random Forest» klarer å utnytte de samme variablene på en mye bedre måte. En gruppe med forskere fra CGBC (Center for Green Buildings and Cities) i Harvard University brukte blant annet «Random Forest» algoritmen i en tilsvarende forskning hvor de forsøkte å predikere energiforbruket til et av bygningene i Harvard Campus. De kom også frem til gode resultater ved å benytte denne algoritmen. Dette er da i samsvar med funnene i denne oppgaven. De predikerte også døgnverdier, men ikke basert på timesverdier slik det er gjort i denne oppgaven.

Det er tydelig at «Random Forest» fungerer veldig effektivt både på timesverdier og døgnverdier. Siden «Random Forest» av natur er svært vanskelig å interpretare, er det ikke så lett å si akkurat hvordan prediksjonen blir så god. Med dette går vi inn i «Black Box» prinsippet. Av Figur 58 kan man få en indikasjon på hvordan modellen vektlegger de ulike variablene. Som for multippel lineær regresjon får cos av året høy viktighet. Det er interessant å se hvordan «Random Forest» nedprioriterer utetemperaturen i prediksjon av timesverdier for denne bygningen. Det er tydelig at modellen gjenkjenner at utetemperaturen fungerer dårlig til å beskrive mønstrene for variasjonen i energiforbruket over året og døgnet samtidig. Det er også viktig å poengtere at denne prosentvisen av viktighet kan være misvisende i den forstand at de ulike variablene forsøker å beskrive de ulike mønstrene. Det er like viktig å beskrive døgnvariasjonen godt som variasjonen over året, men prediksjonen vil jo variere mye mer over året enn over døgnet. Det kan sammenlignes med å treffe på blink med et gevær. «Cos av året» vil sørge for at du treffer blinken, som åpenbart er viktig, men «sin av dagen» vil sørge for at du får 10 poeng istedenfor 1. Disse viktighetene kan ikke sammenlignes direkte.



5.4 VIDERE FORSKNING

I forbindelse med problemområdet som belyses i denne oppgaven er det mye det kan forskers videre på. Prediksjonsmodellen kan testes med flere ulike algoritmer, som blant annet «Gaussian Process». Dette er en kompleks algoritme som lå utenfor avgrensningen i denne oppgaven, men som det er blitt vist, har gitt gode resultater i andre forskningsprosjekter tilsvarende denne. Videre kan andre og mer avanserte variabler benyttes i modellen. I denne oppgaven ble det i hovedsak skilt mellom variasjonen i energiforbruket over året og døgnet. Andre variabler som f.eks. værdata kan beskrive helt andre variasjonsmønstre i forbruket.

Med ET-kurver forsøkes det å skille ut datapunkter som er temperaturavhengig. Dette for blant annet å kunne estimere oppvarmingsbehov av bygninger. Denne problemstillingen ble det lagt særdeles merke til i oppgaven og hvor grov inndelingen av datapunkter ble rundt balansetemperaturen (knekkpunkter) i ET-kurven. Dette er et klassifiseringsproblem som potensielt sett kan løses med maskinlæring. I denne oppgaven er det kun sett på regresjonsproblemer i forbindelse med maskinlæring.



6 KONKLUSJON

Denne oppgaven er en analyse av energiforbruket av boligmassen til Boligbygg og utvikling av en prediksjonsmodell for energiforbruket ved hjelp av kunstig intelligens.

Datagrunnlaget fra Boligbygg inneholdt store ulikheter i omfang og kvalitet. Mye av arbeidet med oppgaven var for å skape et homogent datasett og sikre et godt sammenligningsgrunnlag for alle bygningene. Konklusjonen av dette ble å utelate 44 av 99 bygninger, og inndele de resterende bygningene basert på energiforsyningssystem.

For de resterende bygningene sammenlignes det årlige energiforbruket med kravet teknisk forskrift stiller til netto energibehov. Energiforbruket viser seg å ligge over dette kravet for samtlige bygninger, som gir en indikasjon på at klimaskjermen ikke tilfredsstillers TEK17. Det er samtidig viktig å påpeke forskjellen mellom energiforbruk og energibehov i denne sammenligningen, hvor virkningsgraden i energiforsyningssystemet blir sett bort i fra i beregningen av energibehovet. Ulik systemvirkningsgrad konkluderes også med å være en av årsakene til at bygninger med forskjellig energiforsyningssystem har ulikt energiforbruk.

Analysen gjort med ET-kurven viser at dette er et nyttig verktøy i energioppfølging av bygninger. En rekke interessante estimater og resultatene av disse er beskrevet i denne oppgaven. Det er vist at ET-kurver ikke predikerer timesverdier like presist som for døgnverdier av energiforbruket. Årsaken til dette er at forventet sammenheng mellom energiforbruk og utetemperatur ikke gjelder for variasjoner som oppstår over døgnet.

Videre er det utarbeidet en prediksjonsmodell basert på kunstig intelligens som predikerer energiforbruket til bygninger. Sammenlignet med prediksjoner gjort for døgnverdier basert på ET-kurven, viser denne modellen at ved bruk av kunstig intelligens kan avviket til prediksjonen reduseres med ca. 85%. Svakheter ved ET-kurven henger sammen med at en slik modell baserer sin prediksjon kun på utetemperaturen, og det er vist at utetemperaturen fungerer dårlig til å beskrive døgnvariasjonen i energiforbruket.

I det tilfelle hvor en byggeier skal kjøpe strøm på kraftmarkedet for påfølgende dag basert på predikert energiforbruk, må han ta høyde for avviket prediksjonsmodellen gjør. En mer presis modell vil derfor bidra til å redusere innkjøpskostnadene.



7 REFERANSER

- Aall, C., Simonsen, M., & Hille, J. (2011, 12 13). *Energibruk i norske husholdninger*. Hentet fra Vestlandsforskning: <https://www.vestforsk.no/nn/publication/trender-og-drivere-energibruk-i-norske-husholdninger-rapport-til-nve>
- abrenna. (2018, 10 11). *Kjøp/salg på kraftbørsen*. Hentet fra EnerWE: <https://enerwe.no/slik-kjopes-og-selges-den-norske-strommen-pa-kraftborsen/162936>
- AMS-Målere. (2020, 05 05). Hentet fra Enova: <https://www.enova.no/privat/smar-te-strommalere-ams/>
- Arregi, B., & Garay, R. (2020, 05 01). *Regression analysis of the energy consumption of tertiary buildings*. Retrieved from ScienceDirect: <https://www.sciencedirect.com/science/article/pii/S1876610217328886>
- BKS 421.501. (2017, 10). Hentet fra Byggforskserien: https://www.byggforsk.no/dokument/193/termisk_inneklima_betingelser_tilrettelegging_og_maalinger
- Boligbygg Oslo KF. (2020, 03 24). Hentet fra Oslo Kommune: <https://www.oslo.kommune.no/etater-foretak-og-ombud/boligbygg-oslo-kf/#gref>
- Byggeindustrien, EPC. (2017, 11 16). Hentet fra <http://www.bygg.no/article/1334328>
- Byggeteknisk forskrift. (2020, 05 05). Hentet fra Direktoratet for byggkvalitet: <https://dibk.no/byggereglene/byggeteknisk-forskrift-tek17/14/14-2/>
- Carrsco, O. C. (2020, 04 29). *SVM for classification*. Hentet fra Towards DataScience: <https://towardsdatascience.com/support-vector-machines-for-classification-fc7c1565e3>
- Coefficient of determination. (2020, 03 25). Hentet fra Wikipedia: https://en.wikipedia.org/wiki/Coefficient_of_determination
- Dahlum, S. (2020, 05 14). *Validitet*. Hentet fra Store norske leksikon: <https://snl.no/validitet>
- Decision Tree. (2020, 04 29). Hentet fra Wikipedia: https://en.wikipedia.org/wiki/Decision_tree_learning
- Dev, V., & Eden, M. (2020, 04 22). *Random Decison Forest*. Hentet fra Sciencedirect: <https://www.sciencedirect.com/topics/computer-science/random-decision-forest>
- DK statistikk. (2020, 02 05). Hentet fra DST: <https://www.statistikbanken.dk/statbank5a/selectvarval/saveselections.asp>
- Dokka, T. H., & Andresen, I. (2012, 03). *Energieffektive boliger*. Hentet fra Husbanken: https://www.husbanken.no/rapporter/bib_miljo/energieffektive-boliger-for-fremtiden/
- Ecodesign. (2020, 03 24). Hentet fra IV Produkt: <https://www.ivprodukt.no/nyheter/ecodesign-2018>
- Elektrisitet 2018. (2019, 02 28). Hentet fra SSB: <https://www.ssb.no/energi-og-industri/statistikker/elektrisitet/aar/2019-11-28?fane=tabell&sort=nummer&tabell=399857>



- Energieffektivisering*. (2020, 03 17). Hentet fra Paroc:
<https://www.paroc.no/loesninger/byggisolering/energieffektivisering>
- Energinet*. (2020, 03 08). Hentet fra <https://www.energinet.net/site/usermanual>
- Enova. (2020, 03 02). *www.enova.no*. Hentet fra <https://www.enova.no/om-enova/om-organisasjonen/publikasjoner/>:
https://www.enova.no/upload_images/A6C0AB8383754474A36719C76CB5DFE1.pdf
- Enovas Byggestatistikk 2017*. (2017). Hentet fra Enova:
https://www.enova.no/download?objectPath=upload_images/5C6245BC2AD74248BB629BFA95145AA3.pdf&filename=Enovas%20byggstatistikk%202017.pdf
- Eriksson, M., Akander, J., & Moshfegh, B. (2020, 04 25). *ESM*. Hentet fra ScienceDirect:
<https://www.sciencedirect.com/science/article/pii/S0378778819328580>
- Fjernvarme 2018*. (2020, 02 05). Hentet fra SSB: <https://www.ssb.no/fjernvarme>
- Fjernvarme og frikjøling*. (2020, 04 14). Hentet fra Statistisk sentralbyrå: <https://www.ssb.no/energi-og-industri/statistikker/fjernvarme>
- Fordeling av strømforbruk*. (2020, 02 05). Hentet fra SINTEF:
<https://www.sintef.no/globalassets/upload/energi/nyhetsbrev/ny-kunnskap-om-fordeling-av-stromforbruket.pdf>
- Frøslie, K. F. (2020, 05 15). *Korrelasjon*. Hentet fra Store Norske Leksikon: <https://snl.no/korrelasjon>
- Fumo, N., & Rafe Biswas, M. A. (2020, 04 25). *Regression analysis for prediction of residential energy consumption*. Hentet fra Sciencedirect:
<https://www.sciencedirect.com/science/article/pii/S1364032115001884>
- Graddage*. (2019). Hentet fra Evu: <https://evu.dk/wp-content/uploads/2019/06/Graddage.pdf>
- Gradient Boost*. (2020, 05 06). Hentet fra Wkikipedia: https://en.wikipedia.org/wiki/Gradient_boosting
- Grønmo, S. (2020, 04 11). *Kvantitativ metode*. Hentet fra Store norske leksikon:
https://snl.no/kvantitativ_metode
- Haase, M., & Andresen, I. (2007, 09 17). *Thermal Mass Concepts*. Hentet fra SINTEF:
https://www.sintef.no/globalassets/sintef-byggforsk/coin/sintef-reports/sbf-bk-a07030_thermal-mass-activation.pdf
- Hammarsten, S. (2020, 24 04). *A critical appraisal of energy-signature models*. Hentet fra Sciencedirect:
<https://www.sciencedirect.com/science/article/abs/pii/S0306261987900122>
- Hauge, Å. L., Fredriksen, E., & Klinski, M. (2015, 01 21). *Husbanken*. Hentet fra
<http://biblioteket.husbanken.no/arkiv/dok/Komp/Vurdering%20av%20EPC%20energisparekontrakter%20i%20boligselskaper.pdf>
- Husbanken*. (2020, 03 05). Hentet fra <https://www.husbanken.no/Sok.aspx?query=EPC>
- Husbanken boligkvalitet*. (2020, 02 17). Hentet fra <https://nedlasting.husbanken.no/Filer/7b18.pdf>



James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning*. New York: Springer.

Klima i Norge 2100. (2015). Hentet fra cms.met.no:
https://cms.met.no/site/2/klimaservicesenteret/rapporter-og-publikasjoner/_attachment/6616?_ts=14ff3d4eeb8

Kraftmarkedet. (2020, 05 03). Hentet fra Energifakta Norge: <https://energifaktanorge.no/norsk-energiforsyning/kraftmarkedet/>

Kvaale, E., & Jensen, B. (2011, 12 16). *Husbanken, EPC*. Hentet fra <http://biblioteket.husbanken.no/arkiv/dok/HB/EPC-rapport.pdf>

Lovdata. (2020, 03 17). Hentet fra Regjeringen:
https://www.regjeringen.no/contentassets/9dcd44af30c744ea999f0d122fdc1c3c/lti_forskrift_2017-06-19-840.pdf

Miljøstatus. (2020, 03 04). Hentet fra <https://miljostatus.miljodirektoratet.no/tema/klima/norske-utslipp-av-klimagasser/klimagassutslipp-fra-oppvarming-av-bygg/>

Mosland, T. B. (2020, 03 04). *Tekna*. Hentet fra <https://bygg.tekna.no/hvordan-reducere-co2-utslipp/>

Nilsson, S. (2020, 03 18). *Energi och effekt*. Hentet fra Varmahus:
http://www.varmahus.se/energiskola/energi_effekt.php

NS3031. (2020, 04 20). Lysaker, Norge.

Om energimerkeordning. (2009, 08 10). Hentet fra Energimerking:
<https://www.energimerking.no/no/energimerking-bygg/om-energimerkesystemet-og-regelverket/>

Om organisasjonen. (2020, 03 17). Hentet fra Enova: <https://www.enova.no/om-enova/om-organisasjonen/>

Python. (2020, 02 05). Retrieved from Wikipedia: <https://no.wikipedia.org/wiki/Python>

Sander, K. (2019, 09 05). *Reliabilitet*. Hentet fra estude.no: <https://estude.no/reliabilitet/>

Slik fungerer kraftsystemet. (2018, 10 19). Hentet fra Statnett: <https://www.statnett.no/om-statnett/bli-betere-kjent-med-statnett/slik-fungerer-kraftsystemet/>

Smarte Strømmålere. (2018, 03 20). Hentet fra Svorka: <https://svorka.no/smar-te-strommalere-2/>

Stack effect. (2020, 03 19). Hentet fra Wikipedia: https://en.wikipedia.org/wiki/Stack_effect

Strømpris og Kraftmarkedet. (2020, 05 03). Hentet fra Regjeringen:
<https://www.regjeringen.no/no/tema/energi/stromnettet/kraftmarkedet-og-strompris/id2076000/>

Sverige statistikk. (2020, 02 05). Hentet fra SCB:
http://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START__EN__EN0203/SlutAnvSektor/table/tableViewLayout1/



Tereschchenko , T., Ivanko, D., Nord, N., & Sartori, I. (2019). *Analysis of energy signatures and planning of heating and domestic hot water energy use in buildings in Norway*. Hentet fra SINTEF: <https://sintef.brage.unit.no/sintef-xmlui/handle/11250/2611930>

Thue, J. V. (2016). *Bygningsfysikk Grunnlag* . Trondheim: Vigmostad & Bjørke AS.

Tidemann, A., & Elster, A. (2020, April 4). *Maskinlæring: Store Norkes Leksikon*. Hentet fra Store Norke Leksikon: <https://snl.no/maskinlæring>

Vanderpas, J. (2017). *Python Data Science Handbook*. United States of America: O'Reilly Media.

Yan, B., Wette, C., & Xie, W. (2014, 12 18). *Energy Demand Forecasting*. Hentet fra <http://cs109-energy.github.io/>

Ytongsiporex. (2020, 05 06). Hentet fra Ytongsiporex: https://www.ytongsiporex.no/no/docs/279240_Haandbog_Raadgiver_NO.pdf



8 VEDLEGG

Vedlegg A – Python script for første gjennomgang

A.1 – Innlasting av bygninger

A.2 – Første gjennomgang

Vedlegg B – Python script for analyse

B.1 – ET-Regressor

B.2 – Alle bygninger

B.3 – Bygninger med elektrisk oppvarming

B.4 – Bygninger med fjernvarme

Vedlegg C – Python script for prediksjonsmodellen

C.1 – Korrelasjoner for alle bygninger

C.2 – Korrelasjoner for en tilfeldig valgt bygning

C.3 – Maskinlæring

C.4 – Prediksjoner for alle bygninger

C.5 – Prediksjoner for en tilfeldig valgt bygning

C.6 – Prediksjoner med multipel lineær regresjon

C.7 – Prediksjoner med Random Forest

Vedlegg D – Resterende maskinlæringsalgoritmer for det tilfeldig valgt bygning

Vedlegg A:

Python skript for første gjennomgang

A.1 – Innlasting av bygninger

```

# -*- coding: utf-8 -*-
"""
Created on Thu Jan 16 13:36:50 2020

@author: Sondre Bergene og Murat Bastas
"""

#Imports
#####
import pandas as pd
#####

#Genererer DataFrame med alle bygg ufiltrert
#####
def df_ufilt_bygg():
    filer_info = pd.read_csv('C:/Sondre/skole/Master/data/1.ufiltrerte_bygg.csv',
                             sep=';', encoding='latin-1')

    df_list = []
    for e in filer_info['Adresse']:
        df = pd.read_csv(f'C:/Sondre/skole/Master/data/{e}.csv',
                         sep=';', skiprows=3, decimal=',',
                         converters={0: lambda t: pd.to_datetime(
                             t.split('-')[0].strip(), dayfirst=True)})
        df['Adresse'] = e
        df_list.append(df)
    df_ufilt_bygg = pd.concat(df_list)
    df_ufilt_bygg = pd.merge(df_ufilt_bygg, filer_info, on='Adresse')
    df_ufilt_bygg = df_ufilt_bygg.set_index(['Adresse', 'Tid (Time)']).fillna(0)
    df_ufilt_bygg['Forbruk'] = (df_ufilt_bygg['EL leiligheter'] +
                                df_ufilt_bygg['EL fellesareal'] +
                                df_ufilt_bygg['Fjernvarme'])
    df_ufilt_bygg['Forbruk.m2'] = 1000*(df_ufilt_bygg['Forbruk']/
                                         df_ufilt_bygg['Samlet areal'])
    return df_ufilt_bygg
#####

#Genererer DataFrame med alle bygg
#####
def df_alle_bygg():
    filer_info = pd.read_csv('C:/Sondre/skole/Master/data/2.alle_bygg.csv',
                             sep=';', encoding='latin-1')

    df_list = []
    for e in filer_info['Adresse']:
        df = pd.read_csv(f'C:/Sondre/skole/Master/data/{e}.csv',
                         sep=';', skiprows=3, decimal=',',
                         converters={0: lambda t: pd.to_datetime(
                             t.split('-')[0].strip(), dayfirst=True)})
        df['Adresse'] = e
        df_list.append(df)
    df_alle_bygg = pd.concat(df_list)
    df_alle_bygg = pd.merge(df_alle_bygg, filer_info, on='Adresse')
    df_alle_bygg = df_alle_bygg.set_index(['Adresse', 'Tid (Time)']).fillna(0)
    df_alle_bygg['Forbruk'] = (df_alle_bygg['EL leiligheter'] +
                                df_alle_bygg['EL fellesareal'] +
                                df_alle_bygg['Fjernvarme'])
    df_alle_bygg['Forbruk.m2'] = 1000*(df_alle_bygg['Forbruk']/
                                         df_alle_bygg['Samlet areal'])
    return df_alle_bygg
#####

```

```

#Genererer DataFrame med bygninger med elektrisk varmskilde
#####
def df_el_bygg():
    filer_info = pd.read_csv('C:/Sondre/skole/Master/data/3.el_bygg.csv',
                             sep=';', encoding='latin-1')

    df_list = []
    for e in filer_info['Adresse']:
        df = pd.read_csv(f'C:/Sondre/skole/Master/data/{e}.csv',
                         sep=';', skiprows=3, decimal=',',
                         converters={0: lambda t: pd.to_datetime(
                             t.split('-')[0].strip(), dayfirst=True)})

        df['Adresse'] = e
        df_list.append(df)
    df_el_bygg = pd.concat(df_list)
    df_el_bygg = pd.merge(df_el_bygg, filer_info, on='Adresse')
    df_el_bygg = df_el_bygg.set_index(['Adresse', 'Tid (Time)']).fillna(0)
    df_el_bygg['Forbruk'] = (df_el_bygg['EL leiligheter'] +
                             df_el_bygg['EL fellesareal'] +
                             df_el_bygg['Fjernvarme'])
    df_el_bygg['Forbruk.m2'] = 1000*(df_el_bygg['Forbruk']/
                                     df_el_bygg['Samlet areal'])

    return df_el_bygg
#####

#Genererer DataFrame med bygninger med fjernvarme varmskilde
#####
def df_fv_bygg():
    filer_info = pd.read_csv('C:/Sondre/skole/Master/data/4.fjernvarme_bygg.csv',
                             sep=';', encoding='latin-1')

    df_list = []
    for e in filer_info['Adresse']:
        df = pd.read_csv(f'C:/Sondre/skole/Master/data/{e}.csv',
                         sep=';', skiprows=3, decimal=',',
                         converters={0: lambda t: pd.to_datetime(
                             t.split('-')[0].strip(), dayfirst=True)})

        df['Adresse'] = e
        df_list.append(df)
    dfm = pd.concat(df_list)
    df_fv_bygg = pd.merge(dfm, filer_info, on='Adresse')
    df_fv_bygg = df_fv_bygg.set_index(['Adresse', 'Tid (Time)']).fillna(0)
    df_fv_bygg['Forbruk'] = (df_fv_bygg['EL leiligheter'] +
                             df_fv_bygg['EL fellesareal'] +
                             df_fv_bygg['Fjernvarme'])
    df_fv_bygg['Forbruk.m2'] = 1000*(df_fv_bygg['Forbruk']/
                                     df_fv_bygg['Samlet areal'])

    return df_fv_bygg
#####

#Genererer DataFrames
#####
df_ufilt = df_ufilt_bygg()
df_alle = df_alle_bygg()
df_el = df_el_bygg()
df_fv = df_fv_bygg()
#####

```


A.2 – Første gjennomgang

```

# -*- coding: utf-8 -*-
"""
Created on Tue Mar  3 14:40:18 2020

@author: Sondre Bergene og Murat Bastas
"""

#Imports
#####
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
#####

#Innhenter DataFrame
dfm = df_ufilt

#Beregner totalt årlig energiforbruk pr. kvm
#####
tot_consumption = dfm.pivot_table(index='Adresse', values='Forbruk.m2', aggfunc='sum')
tot_consumption.to_excel("C:/Sondre/skole/Master/excel_filer/filtrering/ufiltrerte_årlig_forbruk
#####

#Plotter alle målinger for alle bygg
#####
for adr in dfm.index.get_level_values(0).unique():

    df = dfm.loc[adr, ['EL leiligheter', 'EL fellesareal', 'Fjernvarme', 'Utetemperatur']]
    df['Timer'] = range(len(df.index))

    fig1 = plt.figure(tight_layout=True, figsize=(8, 8))
    gs = gridspec.GridSpec(2,1)

    ax1 = fig1.add_subplot(gs[0, 0])

    df.plot.scatter(ax=ax1, x='Timer', y='EL leiligheter', color='tab:orange',
                    s=3, alpha=0.3)
    df.plot.scatter(ax=ax1, x='Timer', y='EL fellesareal', color='tab:blue', s=3, alpha=0.3)
    df.plot.scatter(ax=ax1, x='Timer', y='Fjernvarme', color='tab:green', s=3, alpha=0.3)
    ax1.legend(['EL Leiligheter', 'EL Fellesareal', 'Fjernvarme'], markerscale=3)
    ax1.set_xlabel('')
    ax1.set_ylabel('Energiforbruk [kWh/h]')
    ax1.set_xticks([365+i*730 for i in range(12)])
    ax1.set_xticklabels(['Jan', 'Feb', 'Mar', 'Apr', 'Mai', 'Jun',
                        'Jul', 'Aug', 'Sep', 'Okt', 'Nov', 'Des'])
    ax1.annotate('a)', xy=(-0.1,0),xycoords='axes fraction')

    ax2 = fig1.add_subplot(gs[1, 0])
    df.plot.scatter(ax=ax2, x='Utetemperatur', y='EL leiligheter', color='tab:orange',
                    s=3, alpha=0.3)
    df.plot.scatter(ax=ax2, x='Utetemperatur', y='EL fellesareal', color='tab:blue', s=3, alpha=0.3)
    df.plot.scatter(ax=ax2, x='Utetemperatur', y='Fjernvarme', color='tab:green', s=3, alpha=0.3)
    ax2.legend(['EL Leiligheter', 'EL Fellesareal', 'Fjernvarme'], markerscale=3)
    ax2.set_ylabel('Energiforbruk [kWh/h]')
    ax2.set_xlabel('Utetemperatur [C]')
    ax2.annotate('b)', xy=(-0.1,0),xycoords='axes fraction')

    plt.savefig(f"C:/Sondre/skole/Master/figurer/filtrering/ufiltrert_alle_målinger/{adr}")
    plt.close()
#####

```


Vedlegg B:
Python skript for analyse

B.1 – ETRegressor

```

# -*- coding: utf-8 -*-
"""
Created on Thu Apr 30 12:11:08 2020

@author: Hans Ekkhard Plesser (Professor)
"""

from sklearn.base import BaseEstimator, RegressorMixin
from sklearn.utils.validation import check_X_y, check_array, check_is_fitted
import scipy.optimize as sopt
import numpy as np

class ETRegressor(BaseEstimator, RegressorMixin):
    def __init__(self, p0=None):
        """
        p0: [e0, Tt, s]
        """
        if p0 is None:
            p0 = [10, 10, -1]
        self.p0 = p0

    @staticmethod
    def et_curve_(T, e0, Tt, s):
        res = np.full_like(T, e0)
        res[T < Tt] += s * ( T[T < Tt] - Tt )
        return res

    def fit(self, X, y):
        if X.ndim == 1:
            X = X[:, np.newaxis]

        self.X_, self.y_ = check_X_y(X, y)
        self.popt_, self.pcov_ = sopt.curve_fit(self.et_curve_, self.X_[:, 0], self.y_, self.p0)
        return self

    def predict(self, X):
        check_is_fitted(self)
        if X.ndim == 1:
            X = X[:, np.newaxis]
        X = check_array(X)
        return self.et_curve_(X, *self.popt_)

    def rmse(self, X, y):
        yp = self.predict(X)
        return np.sqrt(np.mean((y - yp[:, 0])**2))

```

B.2 – Alle bygninger

```

# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 10:19:40 2020

@author: Sondre Bergene og Murat Bastas
"""

# Imports
#####
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.model_selection import train_test_split
from ETRegressor import ETRegressor
from sklearn.linear_model import LinearRegression
#####

#Innhenter DataFrame
dfm = df_alle

#DataFrame som samler inn resultater
nøkkelverdier = pd.DataFrame(columns=['Årlig forbruk pr. kvm [kWh/kvm*år]',
    'Grunnlast [W/kvm]', 'Knekkpunkt [T]', 'Stigningstall',
    'Oppvarmingsbehov [kWh/kvm*år]', 'Oppvarmingsbehov [%]',
    'Oppvarmingstid [t]', 'R^2', 'RMSE [W/kvm]'])

for adr in dfm.index.get_level_values(0).unique():

#Generer dataframe og fjerner uteliggende data
#####
    df = dfm.loc[adr, ['Forbruk.m2', 'Utetemperatur']]

    df = df[(df['Forbruk.m2'] > 3)]
#####

#Splitter i trening/test sett
#####
    X, y = df.loc[:, 'Utetemperatur:'], df.loc[:, 'Forbruk.m2']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=2)
#####

#Prediksjon med ET-kurve timesverdier
#####
    etr = ETRegressor()
    etr.fit(X_train, y_train)
    etr_y_test_pred = pd.DataFrame(etr.predict(X_test['Utetemperatur']), index=X_test.index)
    etr_y_pred = pd.DataFrame(etr.predict(X['Utetemperatur']), index=X.index)
    etr_r2 = etr.score(X_test['Utetemperatur'], y_test)
    etr_rmse = np.sqrt(metrics.mean_squared_error(y_test, etr_y_test_pred))
#####

#Prediksjon med Linear Regression timesverdier
#####
    lr = LinearRegression()
    lr.fit(X_train, y_train)
    lr_y_test_pred = pd.Series(lr.predict(X_test), index=X_test.index)
    lr_y_pred = pd.Series(lr.predict(X), index=X.index)
    lr_r2 = lr.score(X_test, y_test)

```

```

lr_rmse = np.sqrt(metrics.mean_squared_error(y_test, lr_y_test_pred))
#####

#Generer nøkkelverdier
#####
tot_forbruk = df['Forbruk.m2'].sum()/1000
oppvarming = df['Forbruk.m2'][df['Utetemperatur'] < etr.popt_[1]]
oppvarming = oppvarming[oppvarming > etr.popt_[0]]
oppvarmingsbehov = (oppvarming-etr.popt_[0]).sum()/1000
oppvarming_p = oppvarmingsbehov/tot_forbruk
oppvarmingstid = len(oppvarming)
nøkkelverdier.loc[adr] = [tot_forbruk, etr.popt_[0], etr.popt_[1],
                        etr.popt_[2], oppvarmingsbehov, oppvarming_p, oppvarmingstid,
                        etr.score(X_test, y_test), etr.rmse(X_test, y_test)]
#####

#Plot ET-kurve
#####
fig1 = plt.figure(tight_layout=True, figsize=(8, 4))

r_test = np.corrcoef(etr_y_test_pred[0], y_test)

ax1 = fig1.add_subplot()
ax1.plot(X['Utetemperatur'], y, '.', color='tab:blue', markersize=3, alpha=0.2)
ax1.plot(X['Utetemperatur'], etr_y_pred, '.', color='tab:red', markersize=3, alpha=0.2)
ax1.set_ylabel('Energiforbruk [Wh/(h*m2)]')
ax1.set_xlabel('Utetemperatur [C]')
ax1.legend(['Data', 'ET-kurve'], markerscale=3)
ax1.annotate('r: {:.3f}'.format(r_test[0, 1]),
            xy=(0.02,0.16),xycoords='axes fraction')
ax1.annotate('R^2: {:.3f}'.format(etr_r2),
            xy=(0.02,0.09),xycoords='axes fraction')
ax1.annotate('RMSE: {:.3f} [Wh/(h*m2)]'.format(etr_rmse),
            xy=(0.02,0.02),xycoords='axes fraction')

plt.savefig(f"C:/Sondre/skole/Master/figurer/analyse/alle_ET_kurve/{adr}")
plt.close()
#####

#Plot Lineær Regresjon
#####
fig2 = plt.figure(tight_layout=True, figsize=(8, 4))

r = np.corrcoef(df['Forbruk.m2'], df['Utetemperatur'])

ax1 = fig2.add_subplot()
ax1.plot(X['Utetemperatur'], y, '.', color='tab:blue', markersize=3, alpha=0.2)
ax1.plot(X['Utetemperatur'], lr_y_pred, '.', color='tab:red', markersize=3, alpha=0.2)
ax1.set_ylabel('Energiforbruk [Wh/(h*m2)]')
ax1.set_xlabel('Utetemperatur [C]')
ax1.legend(['Data', 'Lineær Kurve'], markerscale=3)
ax1.annotate('r: {:.3f}'.format(r[0, 1]),
            xy=(0.02,0.16),xycoords='axes fraction')
ax1.annotate('R^2: {:.3f}'.format(lr_r2),
            xy=(0.02,0.09),xycoords='axes fraction')
ax1.annotate('RMSE: {:.3f} [Wh/(h*m2)]'.format(lr_rmse),
            xy=(0.02,0.02),xycoords='axes fraction')

plt.savefig(f"C:/Sondre/skole/Master/figurer/analyse/alle_LR_kurve/{adr}")
plt.close()
#####

```

```
#Skriver nøkkelverdier til excel-fil
#####
nøkkkelverdier.to_excel("C:/Sondre/skole/Master/excel_filer/analyse/alle_ET_fit_nøkkkelverdier.xls")
#####
```

B.3 – Bygninger med elektrisk oppvarming


```

# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 10:19:40 2020

@author: Sondre Bergene og Murat Bastas
"""

# Imports
#####
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.model_selection import train_test_split
from ETRegressor import ETRegressor
#####

#Innhenter DataFrame
dfm = df_el

#DataFrame som samler inn resultater
estimates = pd.DataFrame(columns=['Totalt Forbruk', 'Oppvarmingsbehov',
                                  'Varmtvannsbehov', 'Annet', 'Oppvarming_p',
                                  'Varmtvann_p', 'Annet_p', 'Sjekk', 'Stigningstall',
                                  'Balansetemperatur', 'Oppvarmingstid'])

for adr in dfm.index.get_level_values(0).unique():

#dataframe
#####
#generer en df for ett bygg med de parameterne (kolonnene) som testes
df = dfm.loc[adr, ['Forbruk.m2', 'EL leiligheter', 'EL fellesareal',
                  'Fjernvarme', 'Samlet areal', 'Utetemperatur']]
df['EL leiligheter'] = df['EL leiligheter']/df['Samlet areal']
df['EL fellesareal'] = df['EL fellesareal']/df['Samlet areal']
df['Fjernvarme'] = df['Fjernvarme']/df['Samlet areal']
#####

#Splitter i trening/test sett
#####
X, y = df.loc[:, 'Utetemperatur:'], df.loc[:, 'EL leiligheter']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=2)
#####

#Prediksjon med ET-kurve timesverdier
#####
etr = ETRegressor()
etr.fit(X_train, y_train)
etr_y_test_pred = pd.DataFrame(etr.predict(X_test['Utetemperatur']), index=X_test.index)
etr_y_pred = pd.DataFrame(etr.predict(X['Utetemperatur']), index=X.index)
#####

#Generer nøkkelverdier
#####
tot_forbruk = df['Forbruk.m2'].sum()/1000
oppvarming = df['EL leiligheter'][df['Utetemperatur'] < etr.popt_[1]]
oppvarming = oppvarming[oppvarming > etr.popt_[0]]
oppvarmingsbehov = (oppvarming-etr.popt_[0]).sum()
varmtvannsbehov = df['EL fellesareal'].sum()

```

```

annet = etr.popt_[0]*len(df)
oppvarming_p = oppvarmingsbehov/tot_forbruk
varmtvann_p = varmtvannsbehov/tot_forbruk
annet_p = annet/tot_forbruk
oppvarmingstid = len(oppvarming)

sjekk = oppvarming_p+varmtvann_p+annet_p

estimates.loc[adr] = [tot_forbruk, oppvarmingsbehov, varmtvannsbehov, annet,
                    oppvarming_p, varmtvann_p, annet_p, sjekk,
                    etr.popt_[2], etr.popt_[2], oppvarmingstid]
#####

#Plot ET-kurve
#####
fig1 = plt.figure(tight_layout=True, figsize=(8, 4))

r_test = np.corrcoef(etr_y_test_pred[0], y_test)

ax1 = fig1.add_subplot()
ax1.plot(X['Utetemperatur'], df['EL leiligheter']*1000, '.', color='tab:orange', markersize=3)
ax1.plot(X['Utetemperatur'], df['EL fellesareal']*1000, '.', color='tab:blue', markersize=3)
ax1.plot(X['Utetemperatur'], df['Fjernvarme']*1000, '.', color='tab:green', markersize=3, alpha=0.2)
ax1.plot(X['Utetemperatur'], etr_y_pred*1000, '.', color='tab:red', markersize=3, alpha=0.2)
ax1.legend(['EL leiligheter', 'EL fellesareal', 'Fjernvarme', 'ET-kurve'], markerscale=3)
ax1.set_ylabel('Energiforbruk [Wh/(h*m2)]')
ax1.set_xlabel('Utetemperatur [C]')

plt.savefig(f"C:/Sondre/skole/Master/figurer/analyse/el_ET_kurve/{adr}")
plt.close()
#####

#Skriver nøkkelverdier til excel-fil
#####
estimates.to_excel("C:/Sondre/skole/Master/excel_filer/analyse/el_estimater.xlsx")
#####

```

B.4 – Bygninger med Fjernvarme

```

# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 10:19:40 2020

@author: Sondre Bergene og Murat Bastas
"""

# Imports
#####
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.model_selection import train_test_split
from ETRegressor import ETRegressor
#####

#Innhenter DataFrame
dfm = df_fv

#DataFrame som samler inn resultater
estimates = pd.DataFrame(columns=['Totalt Forbruk', 'Oppvarmingsbehov',
                                  'Varmtvannsbehov', 'Annet', 'Oppvarming_p',
                                  'Varmtvann_p', 'Annet_p', 'Sjekk', 'Stigningstall',
                                  'Balansetemperatur', 'Oppvarmingstid'])

for adr in dfm.index.get_level_values(0).unique():

#dataframe
#####
#generer en df for ett bygg med de parameterne (kolonnene) som testes
df = dfm.loc[adr, ['Forbruk.m2', 'EL leiligheter', 'EL fellesareal',
                  'Fjernvarme', 'Samlet areal', 'Utetemperatur']]
df['EL leiligheter'] = df['EL leiligheter']/df['Samlet areal']
df['EL fellesareal'] = df['EL fellesareal']/df['Samlet areal']
df['Fjernvarme'] = df['Fjernvarme']/df['Samlet areal']
#####

#Splitter i trening/test sett
#####
X, y = df.loc[:, 'Utetemperatur:'], df.loc[:, 'Fjernvarme']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=2)
#####

#Prediksjon med ET-kurve timesverdier
#####
etr = ETRegressor()
etr.fit(X_train, y_train)
etr_y_test_pred = pd.DataFrame(etr.predict(X_test['Utetemperatur']), index=X_test.index)
etr_y_pred = pd.DataFrame(etr.predict(X['Utetemperatur']), index=X.index)
#####

#Generer nøkkelverdier
#####
tot_forbruk = df['Forbruk.m2'].sum()/1000
oppvarming = df['Fjernvarme'][df['Utetemperatur'] < etr.popt_[1]]
oppvarming = oppvarming[oppvarming > etr.popt_[0]]
oppvarmingsbehov = (oppvarming-etr.popt_[0]).sum()
varmtvannsbehov = etr.popt_[0]*len(df)

```

```

annet = (df['EL leiligheter']+df['EL fellesareal']).sum()
oppvarming_p = oppvarmingsbehov/tot_forbruk
varmtvann_p = varmtvannsbehov/tot_forbruk
annet_p = annet/tot_forbruk
oppvarmingstid = len(oppvarming)
sjekk = oppvarming_p+varmtvann_p+annet_p
estimates.loc[adr] = [tot_forbruk, oppvarmingsbehov, varmtvannsbehov, annet,
                    oppvarming_p, varmtvann_p, annet_p, sjekk,
                    etr.popt_[2], etr.popt_[2], oppvarmingstid]
#####

#Plot ET-kurve
#####
fig1 = plt.figure(tight_layout=True, figsize=(8, 4))

r_test = np.corrcoef(etr_y_test_pred[0], y_test)

ax1 = fig1.add_subplot()
ax1.plot(X['Utetemperatur'], df['EL leiligheter']*1000, '.', color='tab:orange', markersize=3)
ax1.plot(X['Utetemperatur'], df['EL fellesareal']*1000, '.', color='tab:blue', markersize=3)
ax1.plot(X['Utetemperatur'], df['Fjernvarme']*1000, '.', color='tab:green', markersize=3, alpha=0.2)
ax1.plot(X['Utetemperatur'], etr_y_pred*1000, '.', color='tab:red', markersize=3, alpha=0.2)
ax1.legend(['EL leiligheter', 'EL fellesareal', 'Fjernvarme', 'ET-kurve'], markerscale=3)
ax1.set_ylabel('Energiforbruk [Wh/(h*m2)]')
ax1.set_xlabel('Utetemperatur [C]')

plt.savefig(f"C:/Sondre/skole/Master/figurer/analyse/fv_ET_kurve/{adr}")
plt.close()
#####

#Skriver nøkkelverdier til excel-fil
#####
estimates.to_excel("C:/Sondre/skole/Master/excel_filer/analyse/fv_estimater.xlsx")
#####

```

Vedlegg C: Python skript for analyse

C.1 – Korrelasjoner for alle bygninger

```

# -*- coding: utf-8 -*-
"""
Created on Sun Apr 19 13:58:27 2020

@author: Sondre Bergene og Murat Bastas
"""

#Imports
#####
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
#####

#Innhenter DataFrame
dfm = df_alle

#Lager en skalerer
sc = StandardScaler()

#DataFrame med timesverdier
#####
df = dfm.loc[adr, ['Forbruk', 'Utetemperatur',]]
df['dof'] = df.index.dayofyear
df['cos'] = np.cos(2*np.pi*(df.index.dayofyear - 15)/365)

df_std = pd.DataFrame(sc.fit_transform(df), index=df.index, columns=df.columns)

corr_T = np.corrcoef(df_std['Forbruk'], df_std['Utetemperatur'])
corr_c = np.corrcoef(df_std['Forbruk'], df_std['cos'])
#####

#DataFrame med døgnverdier
#####
df_d = df.groupby(df.index.dayofyear).mean()
df_d.columns = ['Forbruk_d', 'Utetemperatur_d', 'dof', 'cos_d']
df_d_std = pd.DataFrame(sc.fit_transform(df_d), index=df_d.index, columns=df_d.columns)

corr_T_d = np.corrcoef(df_d_std['Forbruk_d'], df_d_std['Utetemperatur_d'])
corr_c_d = np.corrcoef(df_d_std['Forbruk_d'], df_d_std['cos_d'])
#####

#DataFrame med døgnvariasjon
#####
dfmi = df.merge(df_d, on='dof', right_index=True)

dfmi['Forbruk_dv'] = dfmi['Forbruk']-dfmi['Forbruk_d']
dfmi['Utetemperatur_dv'] = dfmi['Utetemperatur']-dfmi['Utetemperatur_d']
df_dv = dfmi.loc[:, ['Forbruk_dv', 'Utetemperatur_dv']]
df_dv['sin'] = np.sin((df.index.hour+13)*2*np.pi/24)

df_dv_std = pd.DataFrame(sc.fit_transform(df_dv), index=df_dv.index, columns=df_dv.columns)

corr_T_dv = np.corrcoef(df_dv_std['Forbruk_dv'], df_dv_std['Utetemperatur_dv'])
corr_s_dv = np.corrcoef(df_dv_std['Forbruk_dv'], df_dv_std['sin'])
#####

#Plot1
#####
fig1 = plt.figure(tight_layout=True, figsize=(8, 3.5))

```

```

ax1 = fig1.add_subplot()
df_std[['Forbruk', 'Utetemperatur', 'cos']].plot(ax=ax1, ylim=[-3,4],
        color=['tab:orange', 'tab:blue', 'tab:green'])
ax1.set_xlabel('2019')
ax1.set_xticks([])
ax1.set_yticks([])
ax1.legend(['Energiforbruk', 'Utetemperatur (r = {:.3f}, r2 = {:.3f})'
        .format(corr_T[0,1], corr_T[0,1]**2),
        'Cos av året (r = {:.3f}, r2 = {:.3f})'
        .format(corr_c[0,1], corr_c[0,1]**2)])

plt.savefig("C:/Sondre/skole/Master/figurer/prediksjon/nk_korrelasjon_timesverdier")
plt.close()
#####

#Plot2
#####
fig2 = plt.figure(tight_layout=True, figsize=(8, 3.5))

ax2 = fig2.add_subplot()
df_d_std[['Forbruk_d', 'Utetemperatur_d', 'cos_d']].plot(ax=ax2, ylim=[-3,4],
        color=['tab:orange', 'tab:blue', 'tab:green'])
ax2.set_xlabel('2019')
ax2.set_xticks([])
ax2.set_yticks([])
ax2.legend(['Energiforbruk', 'Utetemperatur (r = {:.3f}, r2 = {:.3f})'
        .format(corr_T_d[0,1], corr_T_d[0,1]**2),
        'Cos av året (r = {:.3f}, r2 = {:.3f})'
        .format(corr_c_d[0,1], corr_c_d[0,1]**2)])

plt.savefig("C:/Sondre/skole/Master/figurer/prediksjon/nk_korrelasjon_dognverdier")
plt.close()
#####

#Plot3
#####
fig3 = plt.figure(tight_layout=True, figsize=(8, 3.5))

ax3 = fig3.add_subplot()

df_dv_std['2019-04-26':'2019-04-30'].plot(
        ax=ax3, ylim=[-3,4], color=['tab:orange', 'tab:blue', 'tab:purple'])
ax3.set_xlabel('April')
ax3.set_xticks([])
ax3.set_yticks([])
ax3.legend(['Energiforbruk',
        'Utetemperatur (r = {:.3f}, r2 = {:.3f})'
        .format(corr_T_dv[0,1], corr_T_dv[0,1]**2),
        'Sin av dagen (r = {:.3f}, r2 = {:.3f})'
        .format(corr_s_dv[0,1], corr_s_dv[0,1]**2)])

plt.savefig("C:/Sondre/skole/Master/figurer/prediksjon/nk_korrelasjon_dognvariasjon")
plt.close()
#####

```


C.2 – Korrelasjoner for en tilfeldig valgt bygning

```

# -*- coding: utf-8 -*-
"""
Created on Sun Apr 19 13:58:27 2020

@author: Sondre Bergene og Murat Bastas
"""

#Imports
#####
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import numpy as np
import pandas as pd
import seaborn as sns
#####

#Innhenter DataFrame
dfm = df_alle

#DataFrame som samler inn resultater for korrelasjoner
df_r = pd.DataFrame(columns=['Forbruk', 'Utetemperatur',
                             'Dag_cos', 'Dag_sin',
                             'Time_cos', 'Time_sin', 'Helg'])

df_r2 = pd.DataFrame(columns=['Forbruk', 'Utetemperatur',
                              'Dag_cos', 'Dag_sin',
                              'Time_cos', 'Time_sin', 'Helg'])

df_d_r = pd.DataFrame(columns=['Forbruk', 'Utetemperatur',
                               'Dag_cos', 'Dag_sin',
                               'Time_cos', 'Time_sin', 'Helg'])

df_d_r2 = pd.DataFrame(columns=['Forbruk', 'Utetemperatur',
                                'Dag_cos', 'Dag_sin',
                                'Time_cos', 'Time_sin', 'Helg'])

df_dv_r = pd.DataFrame(columns=['Forbruk', 'Utetemperatur',
                                 'Dag_cos', 'Dag_sin',
                                 'Time_cos', 'Time_sin', 'Helg'])

df_dv_r2 = pd.DataFrame(columns=['Forbruk', 'Utetemperatur',
                                  'Dag_cos', 'Dag_sin',
                                  'Time_cos', 'Time_sin', 'Helg'])

for adr in dfm.index.get_level_values(0).unique():

#DataFrame med timesverdier
#####
    df = dfm.loc[adr, ['Forbruk', 'Utetemperatur',]]

    df['Dag_cos'] = np.cos(2*np.pi*(df.index.dayofyear - 15)/365)
    df['Dag_sin'] = np.sin(2*np.pi*(df.index.dayofyear - 0)/365)
    df['Time_cos'] = np.cos(2*np.pi*(df.index.hour + 0)/24)
    df['Time_sin'] = np.sin(2*np.pi*(df.index.hour + 0)/24)
    df['Helg'] = ( df.index.dayofweek > 4 ) * 1.0

    df_r.loc[adr] = np.corrcoef(df.T)[0]
    df_r2.loc[adr] = df_r.loc[adr]**2
#####

```

```

#DataFrame med døgnverdier
#####
df_d = df.groupby(df.index.dayofyear).mean()

df_d_r.loc[adr] = np.corrcoef(df_d.T)[0]
df_d_r2.loc[adr] = df_d_r.loc[adr]**2
#####

#DataFrame med døgnvariasjon
#####
df['dof'] = df.index.dayofyear
df_d['dof'] = df_d.index
dfmi = df.merge(df_d, on='dof', right_index=True)

df_dv = pd.DataFrame({'Forbruk':dfmi['Forbruk_x']-dfmi['Forbruk_y'],
                      'Utetemperatur':dfmi['Utetemperatur_x']-dfmi['Utetemperatur_y'],
                      'Dag_cos':dfmi['Dag_cos_x']-dfmi['Dag_cos_y'],
                      'Dag_sin':dfmi['Dag_sin_x']-dfmi['Dag_sin_y'],
                      'Time_cos':dfmi['Time_cos_x']-dfmi['Time_cos_y'],
                      'Time_sin':dfmi['Time_sin_x']-dfmi['Time_sin_y'],
                      'Helg':dfmi['Helg_x']-dfmi['Helg_y'],})

df_dv_r.loc[adr] = np.corrcoef(df_dv.T)[0]
df_dv_r2.loc[adr] = df_dv_r.loc[adr]**2
#####

#Skriver korrelasjoner til excel-fil
#####
df_r.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_r_variabler_timesverdier.xlsx")
df_r2.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_r2_variabler_timesverdier.xlsx")
df_d_r.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_r_variabler_døgnverdier.xlsx")
df_d_r2.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_r2_variabler_døgnverdier.xlsx")
df_dv_r.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_r_variabler_døgnvariasjon.xlsx")
df_dv_r2.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_r2_variabler_døgnvariasjon.xlsx")
#####

```

C.3 – Maskinl ering

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Mon May 18 20:00:28 2020
```

```
@author: Sondre Bergene og Murat Bastas
```

```
"""
```

```
#Imports
```

```
#####
```

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import seaborn as sns
import numpy as np
from sklearn import metrics
from sklearn.model_selection import train_test_split
from statsmodels.graphics.gofplots import qqplot
```

```
#####
```

```
def plot_predictions(name, mod_t, mod_d, df, df_d):
```

```
#Behandler dataframes
```

```
#####
```

```
#Lager en kopi for å bevare riktig tider før uteliggende data fjernes
```

```
df_raw = df.copy()
df_d_raw = df_d.copy()
```

```
#fjerner uteliggende data
```

```
df = df[(df['Forbruk.m2'] > 3)]
df_d = df_d[(df_d['Forbruk.m2'] > (3*24))]
```

```
#####
```

```
#Splitter i trening/test sett
```

```
#####
```

```
X, y = df.loc[:, 'Utetemperatur:'], df.loc[:, 'Forbruk.m2']
X_raw, y_raw = df_raw.loc[:, 'Utetemperatur:'], df_raw.loc[:, 'Forbruk.m2']
X_d, y_d = df_d.loc[:, 'Helg:'], df_d.loc[:, 'Forbruk.m2']
X_d_raw, y_d_raw = df_d_raw.loc[:, 'Helg:'], df_d_raw.loc[:, 'Forbruk.m2']
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=2)
X_d_train, X_d_test, y_d_train, y_d_test = train_test_split(
    X_d, y_d, test_size=0.3, random_state=2)
```

```
#####
```

```
#Prediksjon med timesverdier
```

```
#####
```

```
mod_t.fit(X_train, y_train)
y_test_pred = pd.DataFrame(mod_t.predict(X_test), index=X_test.index)
y_raw_pred = pd.DataFrame(mod_t.predict(X_raw), index=X_raw.index)
r2 = mod_t.score(X_test, y_test)
rmse = np.sqrt(metrics.mean_squared_error(y_test, y_test_pred))
```

```
#####
```

```
#Prediksjon med døgnverdier
```

```
#####
```

```
mod_d.fit(X_d_train, y_d_train)
y_d_test_pred = pd.DataFrame(mod_d.predict(X_d_test), index=X_d_test.index)
y_d_raw_pred = pd.DataFrame(mod_d.predict(X_d_raw), index=X_d_raw.index)
r2_d = mod_d.score(X_d_test, y_d_test)
rmse_d = np.sqrt(metrics.mean_squared_error(y_d_test, y_d_test_pred))
```

```

#####
#Prediksjon for døgsnittverdier 2 med timesverdi-modellen
#####
y_d2_raw_pred = y_raw_pred.groupby(y_raw_pred.index.dayofyear).sum()
y_d2_test_pred = pd.Series([float(y_d2_raw_pred.loc[i]) for i in y_d_test.index],
                           index=y_d_test.index)
r2_d2 = np.corrcoef(y_d_test, y_d2_test_pred)**2
rmse_d2 = np.sqrt(metrics.mean_squared_error(y_d_test, y_d2_test_pred))
#####

#Lager liste med resultater
#####
predictions = [r2, rmse, r2_d, rmse_d, r2_d2[0, 1], rmse_d2]
#####

#Plot1
#####
fig1 = plt.figure(tight_layout=True, figsize=(8, 9))

gs = gridspec.GridSpec(3,1)

ax1 = fig1.add_subplot(gs[0, 0])
y_raw['2019-04-25':'2019-05-15'].plot(ax=ax1)
y_raw_pred['2019-04-25':'2019-05-15'].plot(ax=ax1, color='tab:red')
ax1.legend(['Data', 'Prediksjon'], loc='upper right')
ax1.set_xlabel('')
ax1.set_ylabel('Energiforbruk [Wh/(h*m2)]')
ax1.annotate('R^2: {:.3f}'.format(r2),
             xy=(0.02,0.93),xycoords='axes fraction')
ax1.annotate('RMSE: {:.3f} [Wh/(h*m2)]'.format(rmse),
             xy=(0.02,0.86),xycoords='axes fraction')
ax1.annotate('a)', xy=(-0.1,-0.1),xycoords='axes fraction')

ax2 = fig1.add_subplot(gs[1, 0])
y_d_raw.plot(ax=ax2)
y_d_raw_pred[0].plot(ax=ax2, color='tab:red')
ax2.legend(['Data', 'Prediksjon'], loc='lower right')
ax2.set_xlabel('[døgn/år]')
ax2.set_ylabel('Energiforbruk [Wh/(d*m2)]')
ax2.annotate('R^2: {:.3f}'.format(r2_d),
             xy=(0.02,0.09),xycoords='axes fraction')
ax2.annotate('RMSE: {:.3f} Wh/(d*m2)'.format(rmse_d),
             xy=(0.02,0.02),xycoords='axes fraction')
ax2.annotate('b)', xy=(-0.1,-0.1),xycoords='axes fraction')

ax3 = fig1.add_subplot(gs[2, 0])
y_d_raw.plot(ax=ax3)
y_d2_raw_pred.plot(ax=ax3, color='tab:red')
ax3.legend(['Data', 'Prediksjon'], loc='lower right')
ax3.set_xlabel('[døgn/år]')
ax3.set_ylabel('Energiforbruk [Wh/(d*m2)]')
ax3.annotate('R^2: {:.3f}'.format(r2_d2[0, 1]),
             xy=(0.02,0.09),xycoords='axes fraction')
ax3.annotate('RMSE: {:.3f} Wh/(d*m2)'.format(rmse_d2),
             xy=(0.02,0.02),xycoords='axes fraction')
ax3.annotate('c)', xy=(-0.1,-0.1),xycoords='axes fraction')

plt.savefig(f"C:/Sondre/skole/Master/figurer/prediksjon/{name}/nk_{name}_1")
plt.close()
#####

```

```

#Plot2 Prediksjon mot målt verdi
#####
fig2 = plt.figure(tight_layout=True, figsize=(8, 4))

r = np.corrcoef(y_test_pred[0], y_test)

ax1 = fig2.add_subplot()
ax1.plot(y_test, y_test_pred, '.', color='tab:red', markersize=3, alpha=0.3)
ax1.plot([0, 30], [0, 30], 'k--', lw=2)
ax1.set_ylabel('Predikert forbruk [Wh/(h*m2)]')
ax1.set_xlabel('Målt forbruk [Wh/(h*m2)]')

plt.savefig(f"C:/Sondre/skole/Master/figurer/prediksjon/{name}/nk_{name}_2")
plt.close()
#####

#Plot3 ET-graf
#####
fig3 = plt.figure(tight_layout=True, figsize=(8, 4))

r = np.corrcoef(y_test_pred[0], y_test)

ax1 = fig3.add_subplot()
ax1.plot(X_test['Utetemperatur'], y_test, '.', color='tab:blue', markersize=3, alpha=0.3)
ax1.plot(X_test['Utetemperatur'], y_test_pred, '.', color='tab:red', markersize=3, alpha=0.3)
ax1.set_ylabel('Energiforbruk [Wh/(h*m2)]')
ax1.set_xlabel('Utetemperatur [C]')
ax1.legend(['Data', 'Prediksjon'], markerscale=3)
ax1.annotate('r: {:.6.3f}'.format(r[0, 1]),
             xy=(0.02, 0.16), xycoords='axes fraction')
ax1.annotate('R^2: {:.6.3f}'.format(r2),
             xy=(0.02, 0.09), xycoords='axes fraction')
ax1.annotate('RMSE: {:.6.3f} Wh/(h*m2)'.format(rmse),
             xy=(0.02, 0.02), xycoords='axes fraction')

plt.savefig(f"C:/Sondre/skole/Master/figurer/prediksjon/{name}/nk_{name}_3")
plt.close()
#####

#Plot4
#####
fig4 = plt.figure(tight_layout=True, figsize=(8, 4))

gs = gridspec.GridSpec(2, 2)

res_test = y_test_pred[0] - y_test
res_raw = y_raw_pred[0] - y_raw

ax1 = fig4.add_subplot(gs[0, 0])
sns.distplot(res_test, ax=ax1);
ax1.set_title('Fordeling av prediksjonsfeil');
ax1.set_xlabel('Forbruk predikert - Forbruk målt')

ax2 = fig4.add_subplot(gs[0, 1])
qqplot(res_test, line='s', ax=ax2);
ax2.set_title('Kvantiler av målt vs predikert forbruk')

ax3 = fig4.add_subplot(gs[1, 0:])
pd.plotting.autocorrelation_plot(res_raw, marker='.', ax=ax3);
ax3.set_ylim(-0.1, 0.12);

```

```
pmon = np.arange(0, 53, 4);
mlags = pmon * 7 * 24
ax3.set_xticks(mlags);
ax3.set_xticklabels(pmon);
ax3.set_xlabel('Lag [weeks]');
ax3.set_ylabel('');
ax3.set_title('Autokorrelasjon til residuene')

plt.savefig(f"C:/Sondre/skole/Master/figurer/prediksjon/{name}/nk_{name}_4")
plt.close()
#####

return predictions
```


C.4 – Prediksjoner for alle bygninger

```

# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 10:19:40 2020

@author: Sondre Bergene og Murat Bastas
"""

# Imports
#####
import numpy as np
import pandas as pd
from plot_predictions import plot_predictions
from ETRegressor import ETRegressor
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.kernel_ridge import KernelRidge
#####

#Innhenter DataFrame
dfm = df_alle

#DataFrame som samler inn resultater
pred_lr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                               'R2_d2', 'RMSE_d2'])
pred_etr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                 'R2_d2', 'RMSE_d2'])
pred_mlr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                 'R2_d2', 'RMSE_d2'])
pred_kr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])
pred_dt = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])
pred_rf = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])
pred_gb = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])
pred_svr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                 'R2_d2', 'RMSE_d2'])

for adr in dfm.index.get_level_values(0).unique():

#DataFrame med timseverdier
#####
    df = dfm.loc[adr, ['Forbruk.m2', 'Utetemperatur']]
    df['Dag_cos'] = np.cos(2*np.pi*(df.index.dayofyear - 15)/365)
    df['Dag_sin'] = np.sin(2*np.pi*(df.index.dayofyear - 0)/365)
    df['Time_cos'] = np.cos(2*np.pi*(df.index.hour+0)/24)
    df['Time_sin'] = np.sin(2*np.pi*(df.index.hour+0)/24)
    df['Helg'] = ( df.index.dayofweek > 4 ) * 1.0
#####

#DataFrame med døgnverdier
#####
    df_d = df.pivot_table(index=df.index.dayofyear, values=['Forbruk.m2',
                                                            'Utetemperatur', 'Helg'])

    df_d['Forbruk.m2'] *= 24
    df_d['cos'] = np.cos(2*np.pi*(df_d.index - 15)/365)
    df_d['sin'] = np.sin(2*np.pi*(df_d.index - 0)/365)

```

```

df_T = df[['Forbruk.m2', 'Utetemperatur']]
df_d_T = df_d[['Forbruk.m2', 'Utetemperatur']]
#####

#Prediksjon med Linear Regression med kunn utetemperatur
#####
lr = 'Lineær Regresjon'
lr_t = LinearRegression()
lr_d = LinearRegression()
pred_lr.loc[adr] = plot_predictions(lr, lr_t, lr_d, df_T, df_d_T)
#####

#Prediksjon med ET-kurve
#####
etr = 'ET-kurve'
etr_t = ETRegressor()
etr_d = ETRegressor()
pred_etr.loc[adr] = plot_predictions(etr, etr_t, etr_d, df_T, df_d_T)
#####

#Prediksjon med Multippel Linear Regression
#####
mlr = 'Multippel Lineær Regresjon'
mlr_t = LinearRegression()
mlr_d = LinearRegression()
pred_mlr.loc[adr] = plot_predictions(mlr, mlr_t, mlr_d, df, df_d)
#####

#Prediksjon med Decision Tree Regression
#####
dt = 'Decision Tree'
dt_t = DecisionTreeRegressor(random_state = 0)
dt_d = DecisionTreeRegressor(random_state = 0)
pred_dt.loc[adr] = plot_predictions(dt, dt_t, dt_d, df, df_d)
#####

#Prediksjon med Random Forest
#####
rf = 'Random Forest'
rf_t = RandomForestRegressor(n_estimators=300, random_state=9, n_jobs=-1)
rf_d = RandomForestRegressor(n_estimators=300, random_state=9, n_jobs=-1)
pred_rf.loc[adr] = plot_predictions(rf, rf_t, rf_d, df, df_d)
#####

#Prediksjon med Gradient Boost Regression
#####
bg = 'Gradient Boost'
gb_t = GradientBoostingRegressor(random_state=1, n_estimators=50)
gb_d = GradientBoostingRegressor(random_state=1, n_estimators=50)
pred_gb.loc[adr] = plot_predictions(bg, gb_t, gb_d, df, df_d)
#####

#Prediksjon med Support Vector Regression
#####
svr = 'Support Vector'
svr_t = SVR(kernel = 'rbf', gamma='auto')
svr_d = SVR(kernel = 'rbf', gamma='auto')
pred_svr.loc[adr] = plot_predictions(svr, svr_t, svr_d, df, df_d)
#####

```

#Prediksjon med Kernel Ridge Regression

#####

```
kr = 'Kernel Ridge'  
kr_t = KernelRidge(kernel='rbf', gamma=0.1)  
kr_d = KernelRidge(kernel='rbf', gamma=0.1)  
pred_kr.loc[adr] = plot_predictions(kr, kr_t, kr_d, df, df_d)
```

#####

#####

```
pred_lr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_lr.xlsx")  
pred_etr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_etr.xlsx")  
pred_mlr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_mlr.xlsx")  
pred_kr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_kr.xlsx")  
pred_dt.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_dt.xlsx")  
pred_rf.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_rf.xlsx")  
pred_gb.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_gb.xlsx")  
pred_svr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_svr.xlsx")
```

#####

C.5 – Prediksjoner for en tilfeldig valgt bygning

```

# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 10:19:40 2020

@author: Sondre Bergene og Murat Bastas
"""

# Imports
#####
import numpy as np
import pandas as pd
from plot_predictions import plot_predictions
from ETRegressor import ETRegressor
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.kernel_ridge import KernelRidge
#####

#Innhenter DataFrame
dfm = df_alle

#DataFrame som samler inn resultater
pred_lr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                               'R2_d2', 'RMSE_d2'])
pred_etr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                 'R2_d2', 'RMSE_d2'])
pred_mlr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                 'R2_d2', 'RMSE_d2'])
pred_kr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])
pred_dt = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])
pred_rf = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])
pred_gb = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])
pred_svr = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                 'R2_d2', 'RMSE_d2'])

#DataFrame med timseverdier
#####
df = dfm.loc[adr, ['Forbruk.m2', 'Utetemperatur']]
df['Dag_cos'] = np.cos(2*np.pi*(df.index.dayofyear - 15)/365)
df['Dag_sin'] = np.sin(2*np.pi*(df.index.dayofyear - 0)/365)
df['Time_cos'] = np.cos(2*np.pi*(df.index.hour+0)/24)
df['Time_sin'] = np.sin(2*np.pi*(df.index.hour+0)/24)
df['Helg'] = ( df.index.dayofweek > 4 ) * 1.0
#####

#DataFrame med døgnverdier
#####
df_d = df.pivot_table(index=df.index.dayofyear, values=['Forbruk.m2',
                                                         'Utetemperatur', 'Helg'])
df_d['Forbruk.m2'] *= 24
df_d['cos'] = np.cos(2*np.pi*(df_d.index - 15)/365)
df_d['sin'] = np.sin(2*np.pi*(df_d.index - 0)/365)

```

```

df_T = df[['Forbruk.m2', 'Utetemperatur']]
df_d_T = df_d[['Forbruk.m2', 'Utetemperatur']]
#####

#Prediksjon med Linear Regression med kunn utetemperatur
#####
lr = 'Lineær Regresjon'
lr_t = LinearRegression()
lr_d = LinearRegression()
pred_lr.loc[adr] = plot_predictions(lr, lr_t, lr_d, df_T, df_d_T)
#####

#Prediksjon med ET-kurve
#####
etr = 'ET-kurve'
etr_t = ETRegressor()
etr_d = ETRegressor()
pred_etr.loc[adr] = plot_predictions(etr, etr_t, etr_d, df_T, df_d_T)
#####

#Prediksjon med Multippel Linear Regression
#####
mlr = 'Multippel Lineær Regresjon'
mlr_t = LinearRegression()
mlr_d = LinearRegression()
pred_mlr.loc[adr] = plot_predictions(mlr, mlr_t, mlr_d, df, df_d)
#####

#Prediksjon med Decision Tree Regression
#####
dt = 'Decision Tree'
dt_t = DecisionTreeRegressor(random_state = 0)
dt_d = DecisionTreeRegressor(random_state = 0)
pred_dt.loc[adr] = plot_predictions(dt, dt_t, dt_d, df, df_d)
#####

#Prediksjon med Random Forest
#####
rf = 'Random Forest'
rf_t = RandomForestRegressor(n_estimators=300, random_state=9, n_jobs=-1)
rf_d = RandomForestRegressor(n_estimators=300, random_state=9, n_jobs=-1)
pred_rf.loc[adr] = plot_predictions(rf, rf_t, rf_d, df, df_d)
#####

#Prediksjon med Gradient Boost Regression
#####
bg = 'Gradient Boost'
gb_t = GradientBoostingRegressor(random_state=1, n_estimators=50)
gb_d = GradientBoostingRegressor(random_state=1, n_estimators=50)
pred_gb.loc[adr] = plot_predictions(bg, gb_t, gb_d, df, df_d)
#####

#Prediksjon med Support Vector Regression
#####
svr = 'Support Vector'
svr_t = SVR(kernel = 'rbf', gamma='auto')
svr_d = SVR(kernel = 'rbf', gamma='auto')
pred_svr.loc[adr] = plot_predictions(svr, svr_t, svr_d, df, df_d)
#####

#Prediksjon med Kernel Ridge Regression

```

```
#####  
kr = 'Kernel Ridge'  
kr_t = KernelRidge(kernel='rbf', gamma=0.1)  
kr_d = KernelRidge(kernel='rbf', gamma=0.1)  
pred_kr.loc[adr] = plot_predictions(kr, kr_t, kr_d, df, df_d)  
#####  
  
#####  
pred_lr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_lr.xlsx")  
pred_etr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_etr.xlsx")  
pred_mlr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_mlr.xlsx")  
pred_kr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_kr.xlsx")  
pred_dt.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_dt.xlsx")  
pred_rf.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_rf.xlsx")  
pred_gb.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_gb.xlsx")  
pred_svr.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_svr.xlsx")  
#####
```


C.6 – Prediksjoner med multippel lineær regresjon

```

# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 10:19:40 2020

@author: Sondre Bergene og Murat Bastas
"""

# Imports
#####
import pandas as pd
import numpy as np
from sklearn import metrics
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
#####

#Innhenter DataFrame
dfm = df_alle

#DataFrame som samler inn resultater av prediksjonen
pred_res = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])

#DataFrame som samler inn resultater av variabler for timesverdier
var_hour = pd.DataFrame(columns=['Startverdi', 'Utetemperatur', 'Cos av året',
                                'Sin av året', 'Cos av dagen', 'Sin av dagen', 'Helg'])

#DataFrame som samler inn resultater av variabler for døgnverdier
var_day = pd.DataFrame(columns=['Startverdi', 'Utetemperatur', 'Cos av året',
                                'Sin av året', 'Helg'])

for adr in dfm.index.get_level_values(0).unique():

#DataFrame
#####
    df = dfm.loc[adr, ['Forbruk.m2', 'Utetemperatur']]
    df['Cos av året'] = np.cos(2*np.pi*(df.index.dayofyear - 15)/365)
    df['Sin av året'] = np.sin(2*np.pi*(df.index.dayofyear - 0)/365)
    df['Cos av dagen'] = np.cos(2*np.pi*(df.index.hour+0)/24)
    df['Sin av dagen'] = np.sin(2*np.pi*(df.index.hour+0)/24)
    df['Helg'] = ( df.index.dayofweek > 4 ) * 1.0

    df_d = df.pivot_table(index=df.index.dayofyear,
                           values=['Forbruk.m2', 'Utetemperatur', 'Helg'])
    df_d['Forbruk.m2'] *= 24
    df_d['Cos av året'] = np.cos(2*np.pi*(df_d.index - 15)/365)
    df_d['Sin av året'] = np.sin(2*np.pi*(df_d.index - 0)/365)
#####

#Lager en kopi for å bevare riktig tider før uteliggende data fjernes
#####
    df_raw = df.copy()
    df_d_raw = df_d.copy()

    df = df[(df['Forbruk.m2'] > 3)]
    df_d = df_d[(df_d['Forbruk.m2'] > (3*24))]
#####

#Standardiserer og splitter i trening/test sett
#####

```

```

X, y = df.loc[:, 'Utetemperatur'], df.loc[:, 'Forbruk.m2']
X_raw, y_raw = df_raw.loc[:, 'Utetemperatur'], df_raw.loc[:, 'Forbruk.m2']
X_d, y_d = df_d.loc[:, 'Helg'], df_d.loc[:, 'Forbruk.m2']
X_d_raw, y_d_raw = df_d_raw.loc[:, 'Helg'], df_d_raw.loc[:, 'Forbruk.m2']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
                                                    random_state=2)
X_d_train, X_d_test, y_d_train, y_d_test = train_test_split(X_d, y_d,
                                                            test_size=0.3, random_state=2)

sc = StandardScaler()
X_train_std = pd.DataFrame(sc.fit_transform(X_train),
                            index=X_train.index, columns=X_train.columns)
X_test_std = pd.DataFrame(sc.fit_transform(X_test), index=X_test.index,
                           columns=X_test.columns)
X_raw_std = pd.DataFrame(sc.fit_transform(X_raw), index=X_raw.index,
                          columns=X_raw.columns)
X_d_train_std = pd.DataFrame(sc.fit_transform(X_d_train),
                              index=X_d_train.index, columns=X_d_train.columns)
X_d_test_std = pd.DataFrame(sc.fit_transform(X_d_test),
                              index=X_d_test.index, columns=X_d_test.columns)
X_d_raw_std = pd.DataFrame(sc.fit_transform(X_d_raw),
                              index=X_d_raw.index, columns=X_d_raw.columns)
#####

#Prediksjon med Linear Regression timesverdier
#####
mlr = LinearRegression()
mlr.fit(X_train_std, y_train)
mlr_y_test_pred = pd.DataFrame(mlr.predict(X_test_std), index=X_test.index)
mlr_y_raw_pred = pd.DataFrame(mlr.predict(X_raw_std), index=X_raw_std.index)
mlr_r2 = mlr.score(X_test_std, y_test)
mlr_rmse = np.sqrt(metrics.mean_squared_error(y_test, mlr_y_test_pred))
#####

#Prediksjon med Linear Regression døgnverdier
#####
mlr_d = LinearRegression()
mlr_d.fit(X_d_train_std, y_d_train)
mlr_d_y_test_pred = pd.DataFrame(mlr_d.predict(X_d_test_std), index=X_d_test_std.index)
mlr_d_y_raw_pred = pd.DataFrame(mlr_d.predict(X_d_raw_std), index=X_d_raw_std.index)
mlr_d_r2 = mlr_d.score(X_d_test_std, y_d_test)
mlr_d_rmse = np.sqrt(metrics.mean_squared_error(y_d_test, mlr_d_y_test_pred))
#####

#Prediksjon med Linear Regression døgsnittverdier 2 med timesverdi-modellen
#####
mlr_d_y_raw_pred2 = mlr_y_raw_pred.groupby(mlr_y_raw_pred.index.dayofyear).sum()
mlr_d_y_test_pred2 = pd.Series([float(mlr_d_y_raw_pred2.loc[i]) for i in y_d_test.index], index=y_d_test.index)
mlr_d2_r2 = np.corrcoef(y_d_test, mlr_d_y_test_pred2)**2
mlr_d2_rmse = np.sqrt(metrics.mean_squared_error(y_d_test, mlr_d_y_test_pred2))
#####

#Samler resultatene av prediksjonene og variablene for alle byggene i DataFrames
#####
pred_res.loc[adr] = mlr_r2, mlr_rmse, mlr_d_r2, mlr_d_rmse, mlr_d2_r2[0, 1], mlr_d2_rmse

var_hour.loc[adr] = mlr.intercept_, *mlr.coef_
var_day.loc[adr] = mlr_d.intercept_, *mlr_d.coef_
#####

```

```
#Skriver til excel fil resultater av prediksjonen og variabler
#####
var_hour.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_variabler_MLR_timesverdier
var_day.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_variabler_MLR_døgnverdier.)
pred_res.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_MLR.xlsx")
#####
```

C.7 – Prediksjoner med Random Forest

```

# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 10:19:40 2020

@author: Sondre Bergene og Murat Bastas
"""

# Imports
#####
import pandas as pd
import numpy as np
from sklearn import metrics
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
#####

#Innhenter DataFrame
dfm = df_alle

#DataFrame som samler inn resultater av prediksjonen
pred_res = pd.DataFrame(columns=['R2', 'RMSE', 'R2_d1', 'RMSE_d1',
                                'R2_d2', 'RMSE_d2'])

#DataFrame som samler inn resultater av variabler for timesverdier
var_hour = pd.DataFrame(columns=['Utetemperatur', 'Cos av året',
                                'Sin av året', 'Cos av dagen', 'Sin av dagen', 'Helg'])

#DataFrame som samler inn resultater av variabler for døgnverdier
var_day = pd.DataFrame(columns=['Utetemperatur', 'Cos av året',
                                'Sin av året', 'Helg'])

for adr in dfm.index.get_level_values(0).unique():

#dataframe
#####
    df = dfm.loc[adr, ['Forbruk.m2', 'Utetemperatur']]
    df['Cos av året'] = np.cos(2*np.pi*(df.index.dayofyear - 15)/365)
    df['Sin av året'] = np.sin(2*np.pi*(df.index.dayofyear - 0)/365)
    df['Cos av dagen'] = np.cos(2*np.pi*df.index.hour/24)
    df['Sin av dagen'] = np.sin(2*np.pi*df.index.hour/24)
    df['Helg'] = ( df.index.dayofweek > 4 ) * 1.0

    df_d = df.pivot_table(index=df.index.dayofyear,
                            values=['Forbruk.m2', 'Utetemperatur', 'Helg'])
    df_d['Forbruk.m2'] *= 24
    df_d['Cos av året'] = np.cos(2*np.pi*(df_d.index - 15)/365)
    df_d['Sin av året'] = np.sin(2*np.pi*(df_d.index - 0)/365)
#####

#Lager en kopi for å bevare riktig tider før uteliggende data fjernes
#####
    df_raw = df.copy()
    df_d_raw = df_d.copy()

    df = df[(df['Forbruk.m2'] > 3)]
    df_d = df_d[(df_d['Forbruk.m2'] > (3*24))]
#####

#Standardiserer og splitter i trening/test sett
#####

```

```

X, y = df.loc[:, 'Utetemperatur:'], df.loc[:, 'Forbruk.m2']
X_raw, y_raw = df_raw.loc[:, 'Utetemperatur:'], df_raw.loc[:, 'Forbruk.m2']
X_d, y_d = df_d.loc[:, 'Helg:'], df_d.loc[:, 'Forbruk.m2']
X_d_raw, y_d_raw = df_d_raw.loc[:, 'Helg:'], df_d_raw.loc[:, 'Forbruk.m2']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=2)
X_d_train, X_d_test, y_d_train, y_d_test = train_test_split(
    X_d, y_d, test_size=0.3, random_state=2)
#####

#Prediksjon med Random Forest timesverdier
#####
rf = RandomForestRegressor(n_estimators=300, random_state=9, n_jobs=-1)
rf.fit(X_train, y_train)
rf_y_test_pred = pd.DataFrame(rf.predict(X_test), index=X_test.index)
rf_y_raw_pred = pd.DataFrame(rf.predict(X_raw), index=X_raw.index)
rf_r2 = rf.score(X_test, y_test)
rf_rmse = np.sqrt(metrics.mean_squared_error(y_test, rf_y_test_pred))
#####

#Prediksjon med Random Forest døgnsnittverdier 1
#####
rf_d = RandomForestRegressor(n_estimators=300, random_state=9, n_jobs=-1)
rf_d.fit(X_d_train, y_d_train)
rf_d_y_test_pred = pd.DataFrame(rf_d.predict(X_d_test), index=X_d_test.index)
rf_d_y_raw_pred = pd.DataFrame(rf_d.predict(X_d_raw), index=X_d_raw.index)
rf_d_r2 = rf_d.score(X_d_test, y_d_test)
rf_d_rmse = np.sqrt(metrics.mean_squared_error(y_d_test, rf_d_y_test_pred))
#####

#Prediksjon Random Forest for døgnsnittverdier 2 med timesverdi-modellen
#####
rf_d_y_raw_pred2 = rf_y_raw_pred.groupby(rf_y_raw_pred.index.dayofyear).sum()
rf_d_y_test_pred2 = pd.Series([float(rf_d_y_raw_pred2.loc[i]) for i in y_d_test.index], index=y_d_test.index)
rf_d2_r2 = np.corrcoef(y_d_test, rf_d_y_test_pred2)**2
rf_d2_rmse = np.sqrt(metrics.mean_squared_error(y_d_test, rf_d_y_test_pred2))
#####

#Samler resultatene av prediksjonene og variablene for alle byggene i DataFrames
#####
pred_res.loc[adr] = rf_r2, rf_rmse, rf_d_r2, rf_d_rmse, rf_d2_r2[0, 1], rf_d2_rmse

var_hour.loc[adr] = rf.feature_importances_
var_day.loc[adr] = rf_d.feature_importances_
#####

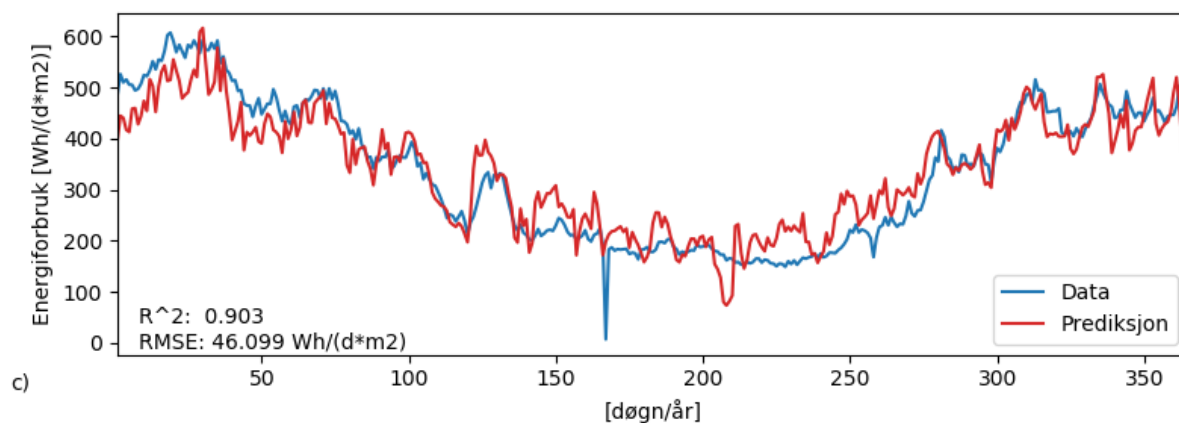
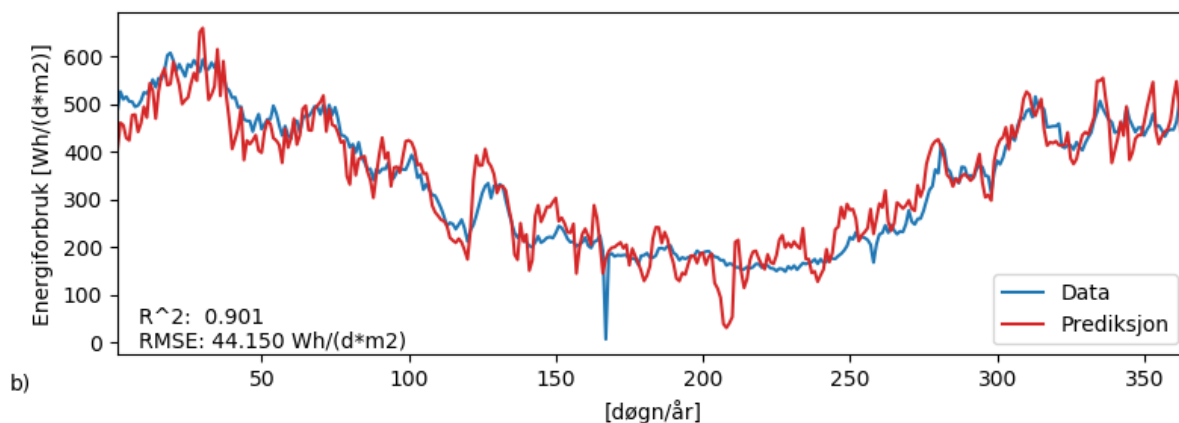
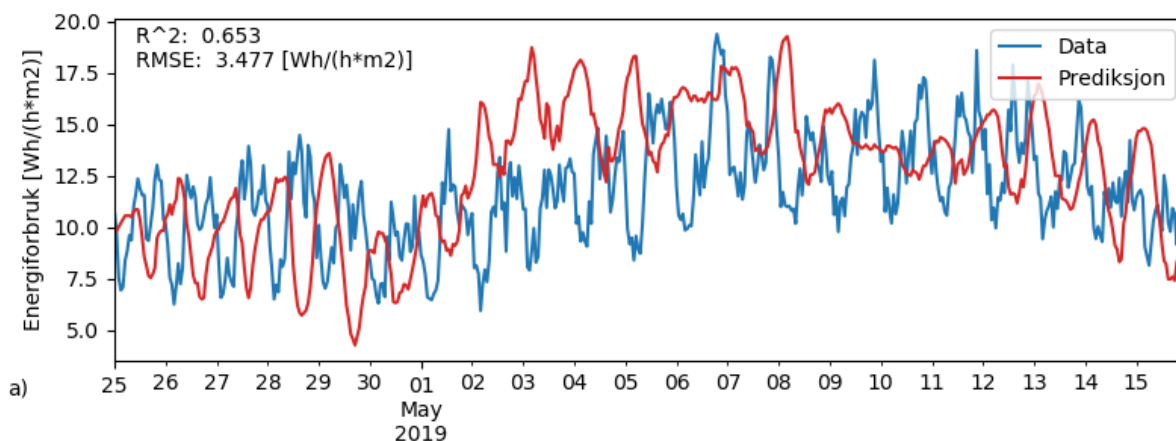
#Skriver til excelfil resultater av prediksjonen og variabler
#####
var_hour.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_variabler_RF_timesverdier.xlsx")
var_day.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_variabler_RF_døgnverdier.xlsx")
pred_res.to_excel("C:/Sondre/skole/Master/excel_filer/prediksjon/alle_prediksjoner_RF.xlsx")
#####

```

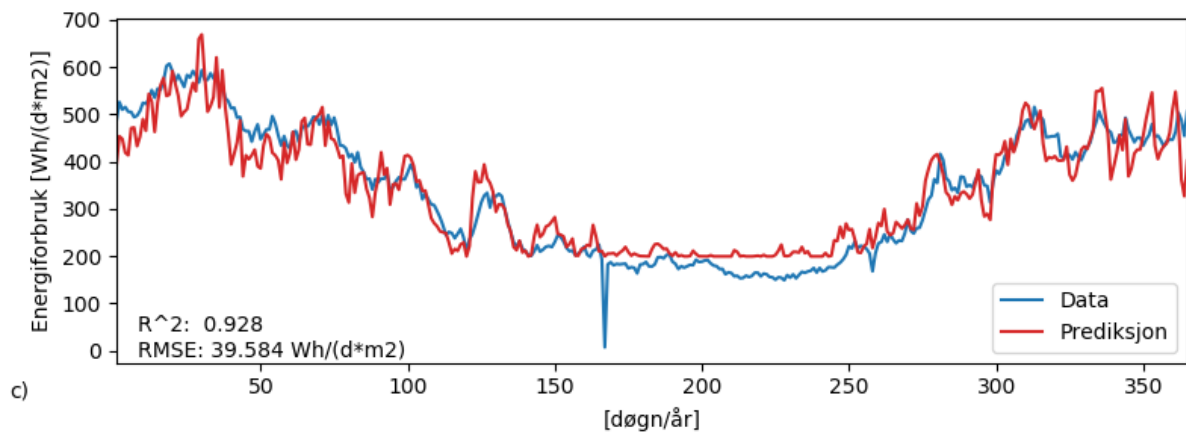
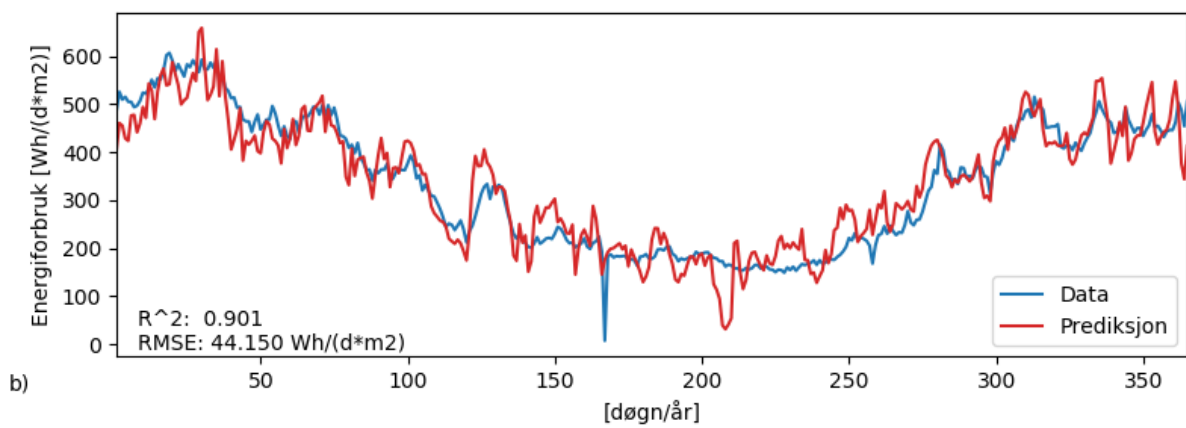
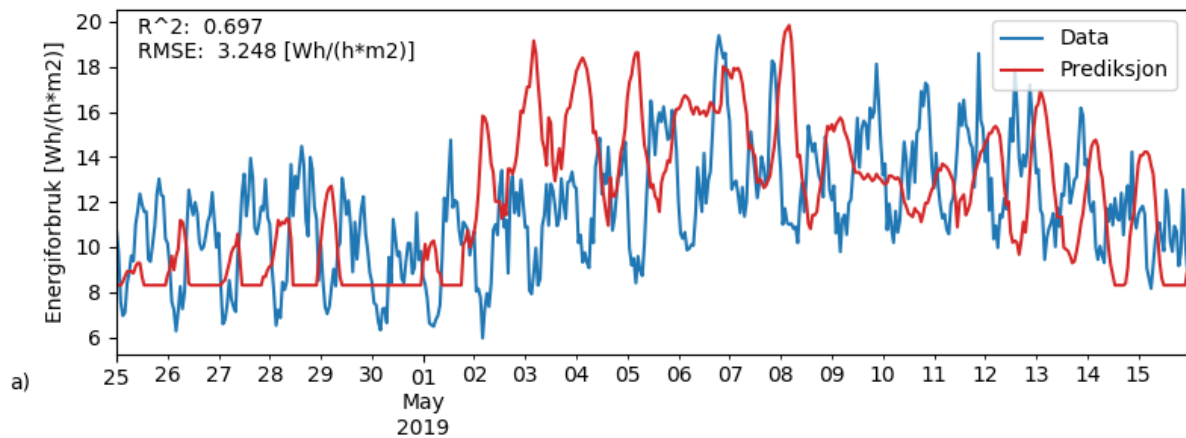
Vedlegg D:

Resterende maskinlæringsalgoritmer for den tilfeldig valgte bygningen

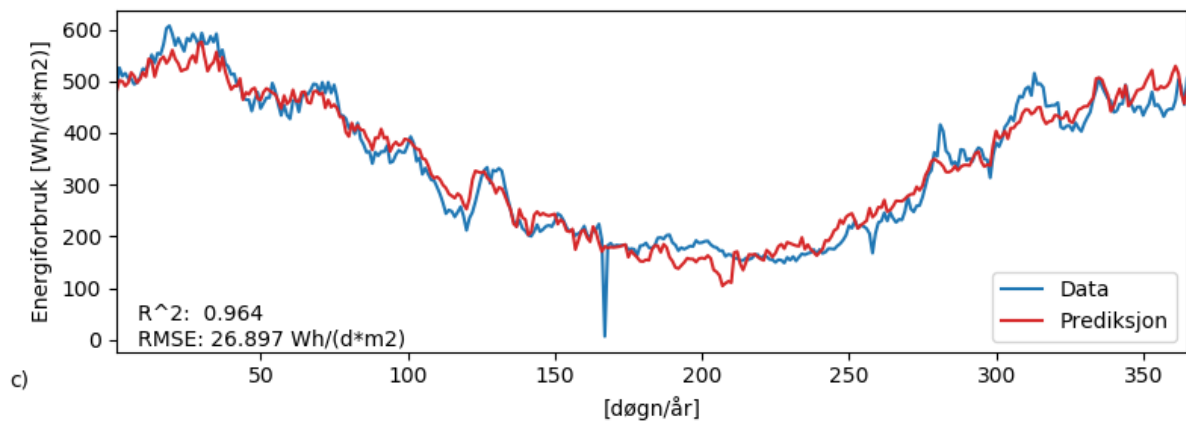
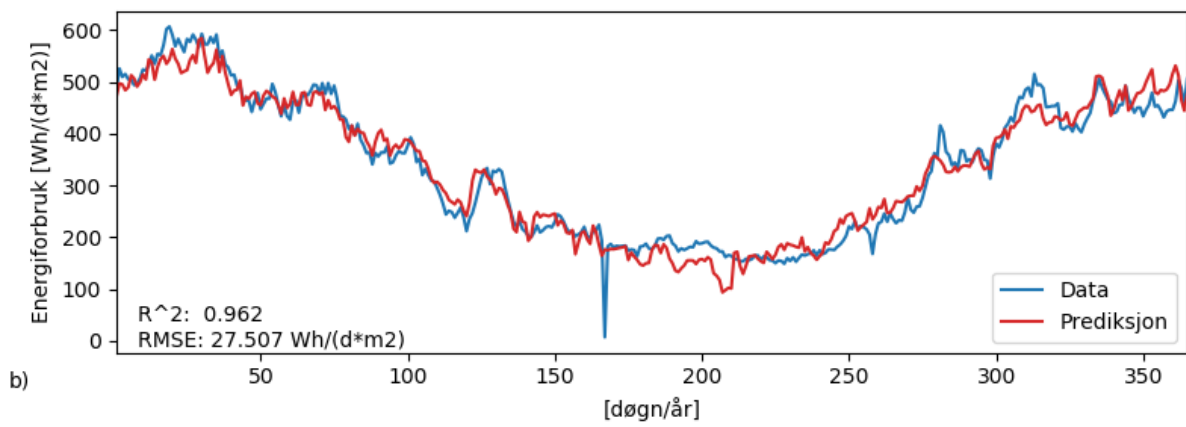
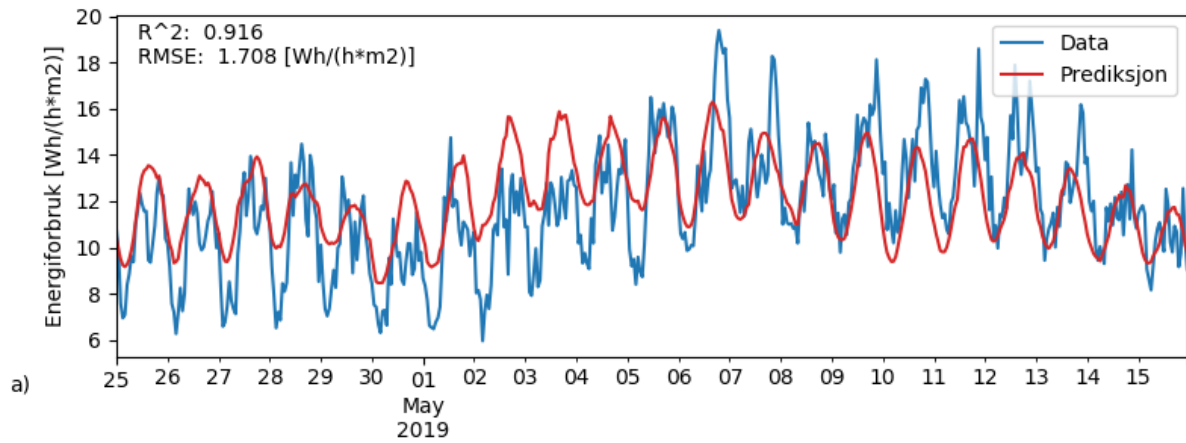
LINEÆR REGRESJON



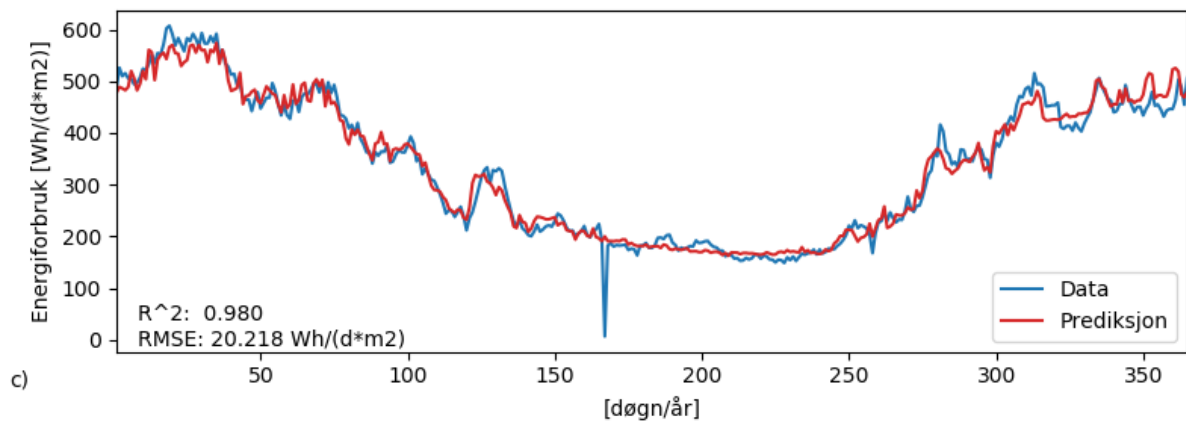
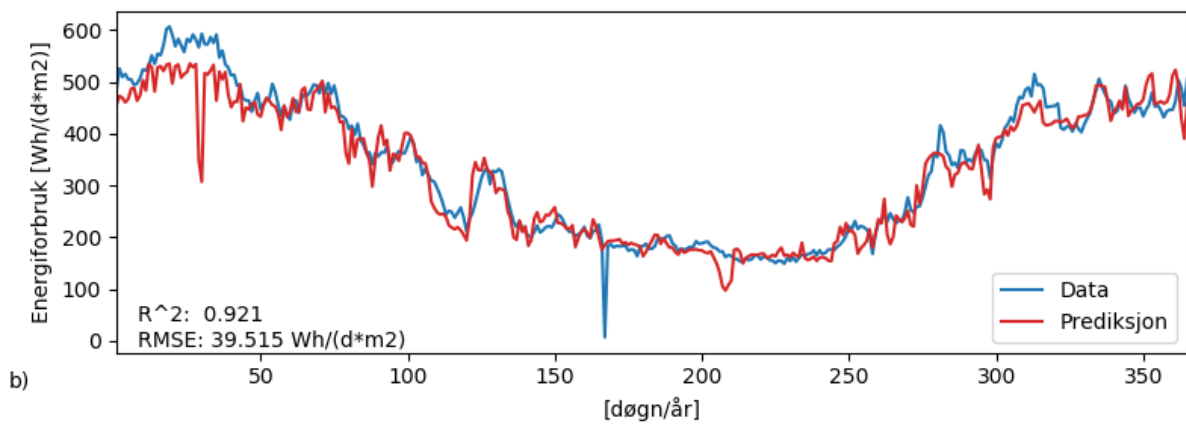
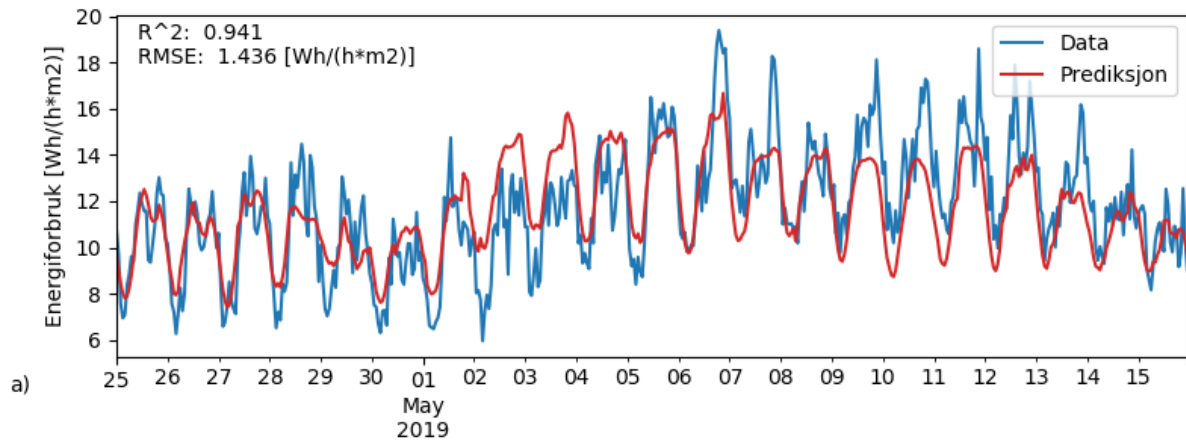
ET-KURVE



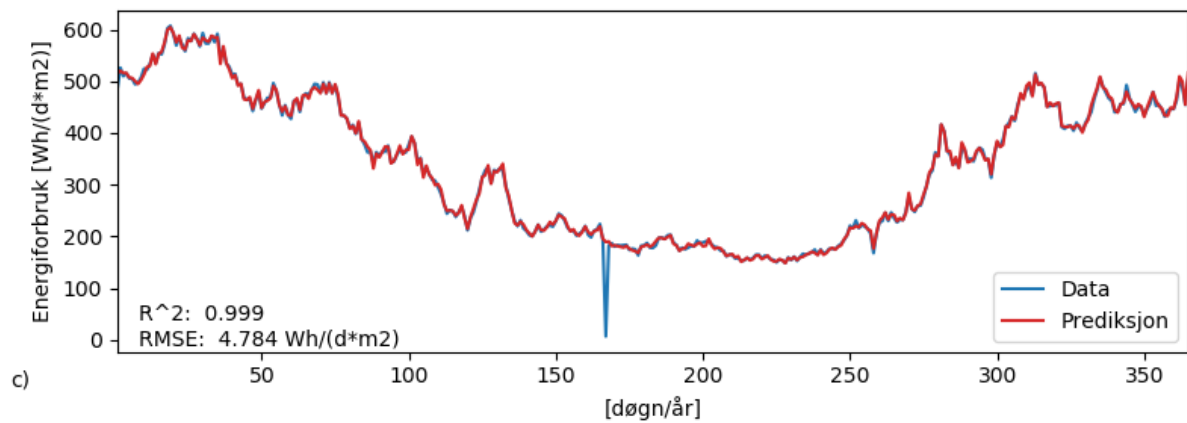
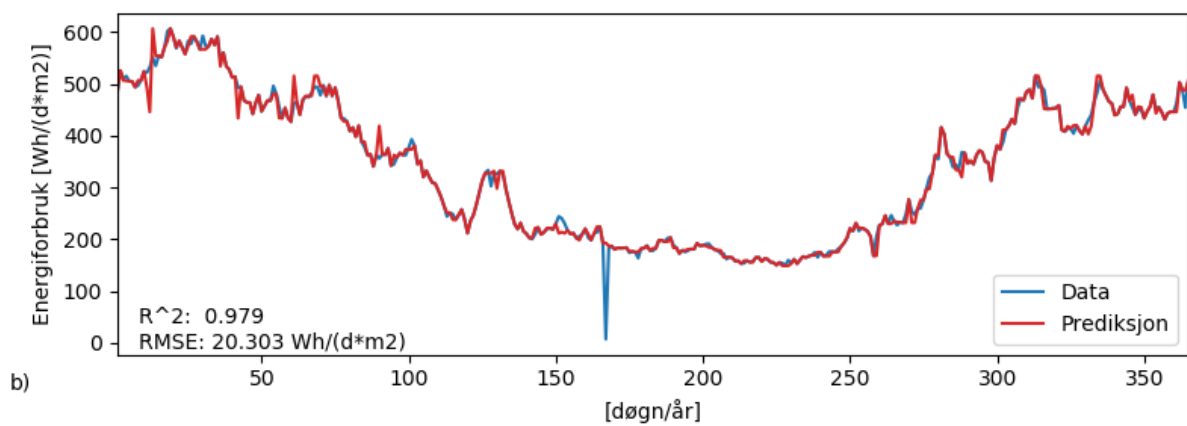
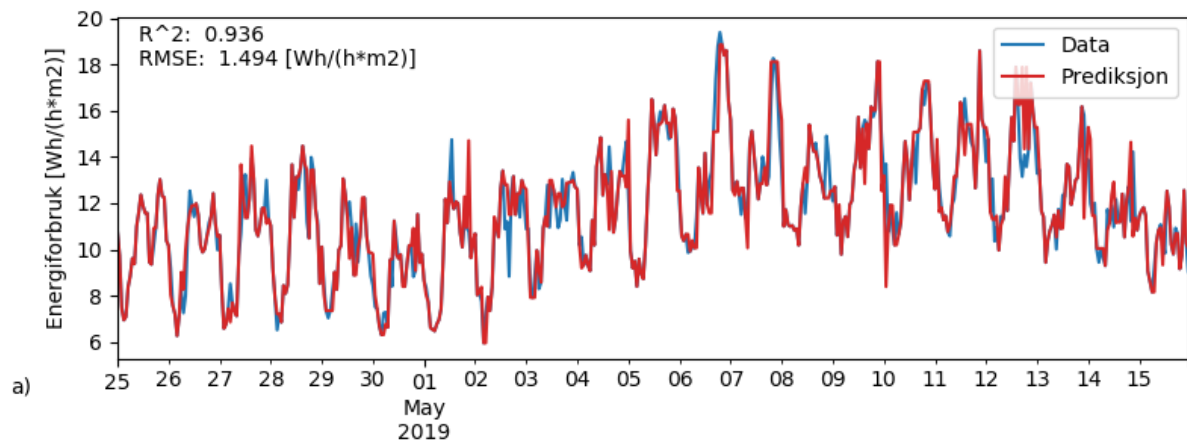
MULTIPPEL LINEÆR REGRESJON



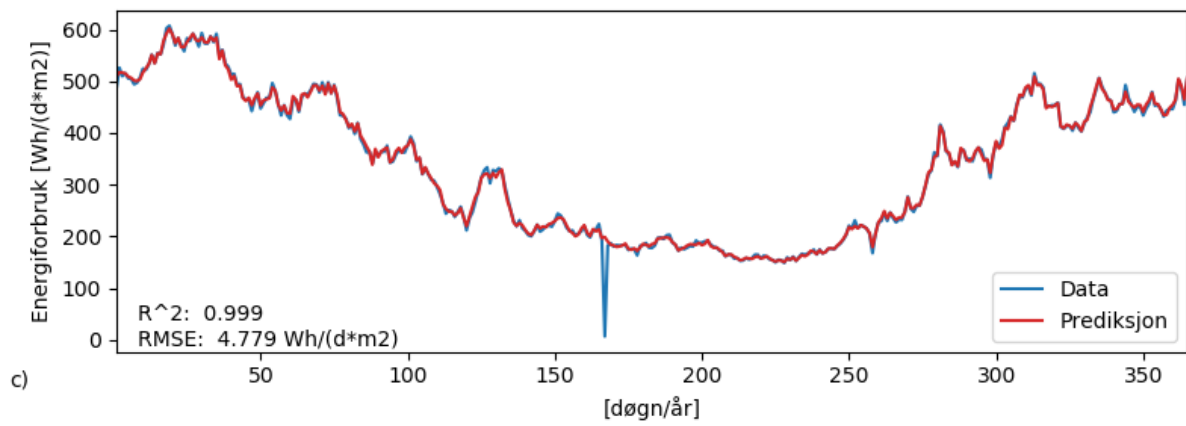
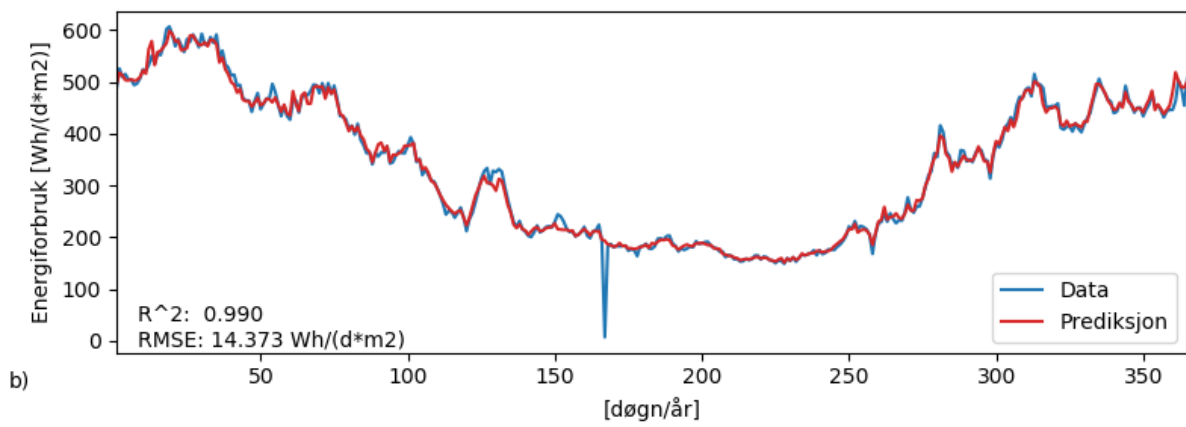
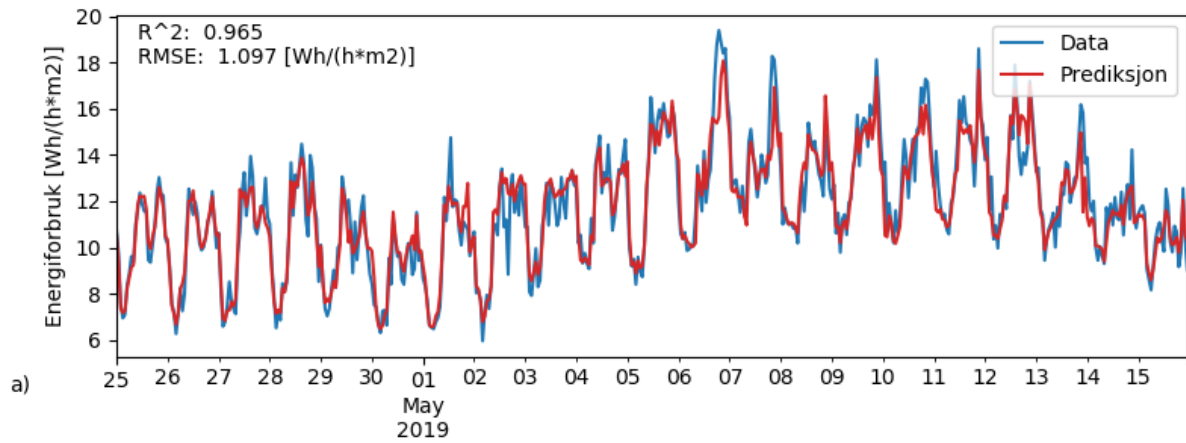
KERNAL RIDGE REGRESSION



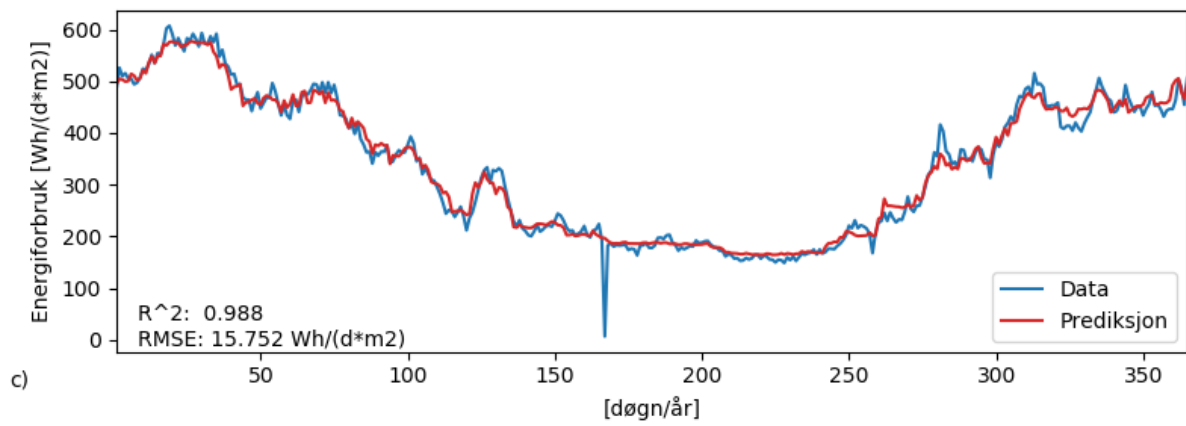
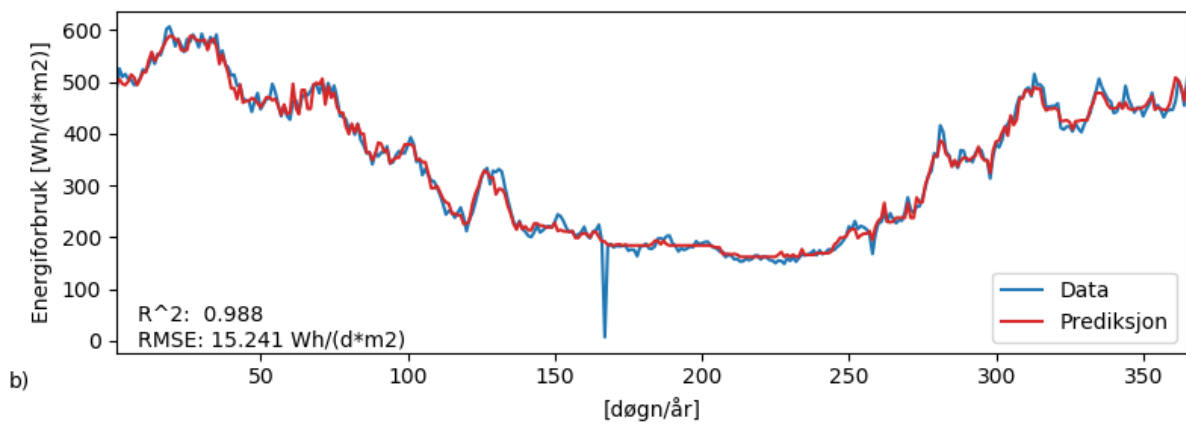
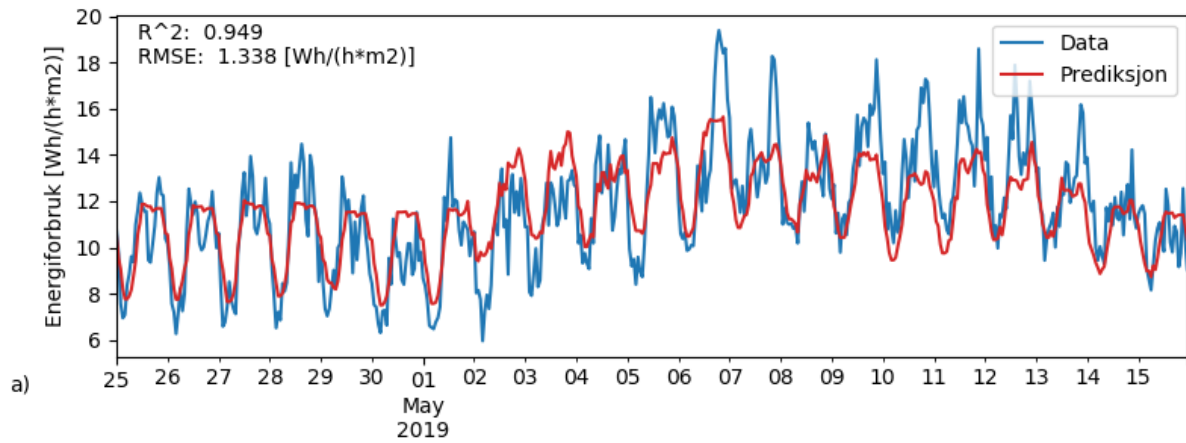
DECISION TREE REGRESSION



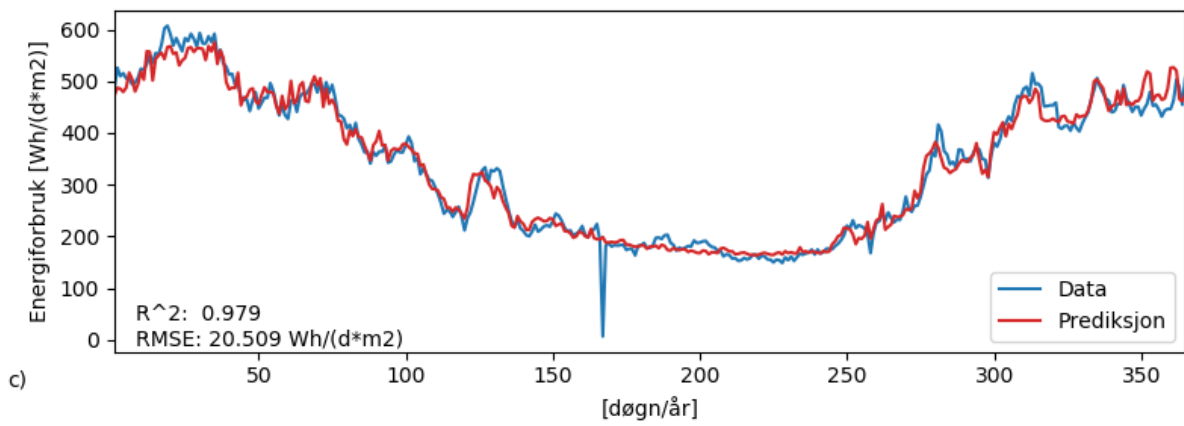
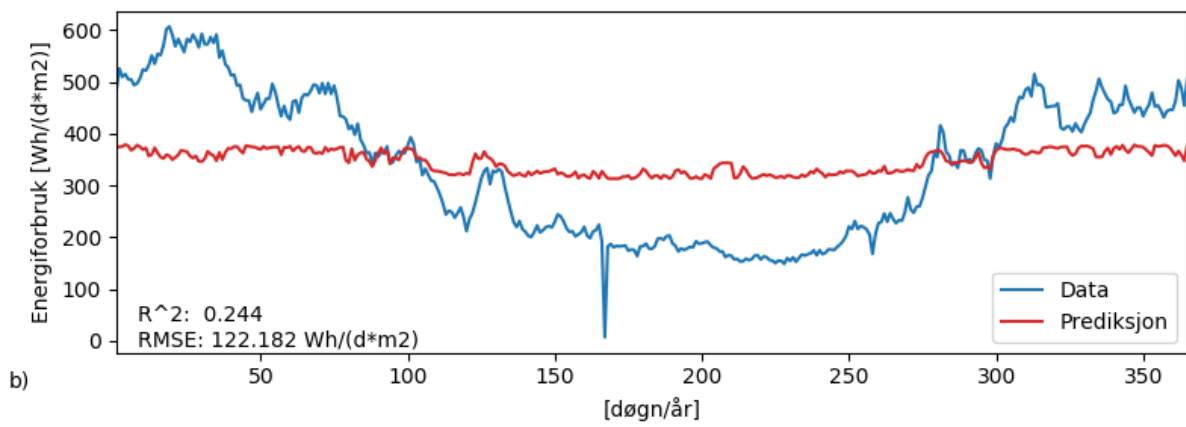
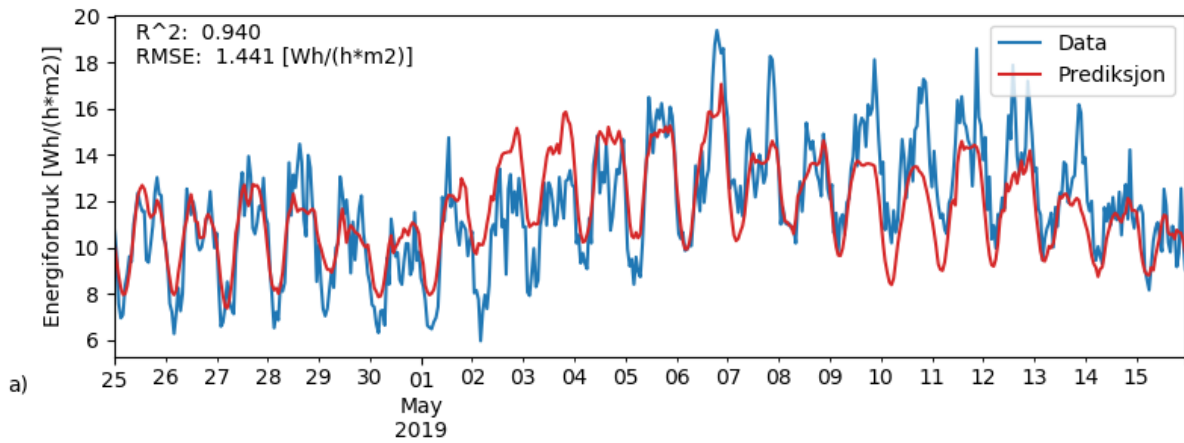
RANDOM FOREST REGRESSION



GRADIENT BOOST REGRESSION



SUPPORT VECTOR REGRESSION





Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway