



Norges miljø- og  
biovitenskapelige  
universitet

**Masteroppgave 2023 30 stp**  
Fakultet for realfag og teknologi

# **Automatisk deteksjon av myr med bruk av hyperspektrale bilder og maskinlæring**

Automatic detection of mires using hyperspectral  
images machine learning

Håvard Ullebust Hagen  
Geomatikk

# Sammendrag

Myra i Norge kan sees på dens utbredte karbonlager og det biologiske mangfoldet den inneholder. Myra over våre skoggrensener er dårlig kartlagt, og i denne oppgaven undersøkes det om det er mulig å detektere myra automatisk. Undersøkelsen benytter seg av en metode som klassifiserer egenskaper i hyperspektrale data. Metoden baserer seg på å utnytte diverse maskinlæringsmodeller til å kunne klassifisere myra. Maskinlæringsmodellene blir trent ved hjelp av et arealressurskart, der det er tilstrekkelig mengde klassifisert myr.

Maskinlæringsmodellene som testes er klassifiseringsalgoritmene Support Vector Machine (SVM), Random Forest Classifier (RFC), Logistic Regression (LogReg) og K-Nearest Neighbors (KNN). Denne oppgaven ser på diverse evalueringsmål for modellene for å veie dem opp mot hverandre og se hvilke av de som presterer best. De ulike evalueringmålene som ble brukt i oppgaven var Matthews korrelasjonskoeffisient (MCC), kappa-koeffisient, klassifiseringsrapporten til modellene, modellenes forvirringsmatrise, AUC-verdi og modellenes klassifiseringsnøyaktighet.

Ved å undersøke klassifiseringsalgoritmene sine evalueringsmål, viser det seg at det er mulig å gjennomføre en vellykket, aksepterbar og tilfredsstillende klassifisering av myr. SVM-modellen presterte best i de fleste evalueringmålene. Den presterte med klassifiseringsnøyaktighet på 0.811, en MCC på 0.632, kappa-koeffisient på 0.632 og en AUC score på 0.87. Disse verdiene forteller at myrarealene som klassifiseres av modellene samsvarer godt med myrarealene i treningsdataene.

# Abstract

The mires in Norway can be appreciated for its extensive carbon storage and the biodiversity it contains. The mires beyond our forest borders are poorly mapped, and this study examines the possibility of automatically detecting mires. The investigation utilizes a method that classifies features in hyperspectral data. The method relies on various machine learning models to classify mires. The machine learning models are trained on a land resource map that includes enough classified mires.

The machine learning models tested in this study include the classification algorithms Support Vector Machine (SVM), Random Forest Classifier (RFC), Logistic Regression (LogReg), and K-Nearest Neighbors (KNN). In this study, various evaluation metrics are used to compare and determine which models perform the best. The evaluation metrics employed in the study include the Matthews correlation coefficient (MCC), kappa coefficient, classification report of the models, confusion matrices of the models, AUC value, and classification accuracy of the models.

By examining the evaluation metrics of the classification algorithms, the results show that it is possible to achieve a successful, acceptable, and potentially good classification or automatic detection of mires. The classification results of SVM determined that it performed the best in most of the evaluation metrics compared to the other models. It achieved a classification accuracy of 0.811, an MCC of 0.632, a kappa coefficient of 0.632, and an AUC score of 0.87. These values indicate that the mire areas classified by the models correspond well with the mire areas in the training data.

# Forord

Denne masteroppgaven er skrevet på Institutt for matematisk realfag og teknologi. Oppgaven er min avslutning på mastergradstudiet i geomatikk ved Norges miljø- og biovitenskapelige universitet (NMBU) i Ås. Oppgaven er gitt av- og utført med NIBIO, samt at den er veiledet av Nicolai Munsterhjelm fra NIBIO og Misganu Debella-Gilo fra NMBU.

Jeg ønsker å takke Nicolai Munsterhjelm for en særdeles effektiv og tålmodig veiledning. Jeg takker for raske tilbakemeldinger, som kom ofte og var uvurderlige.

Takk til Misganu Debella-Gilo for god og uvurderlig hjelp til skriving av oppgaven.

Takk til NIBIO for dataanskaffelse.

*Norges miljø- og biovitenskapelige universitet*

*Ås, 15. Juni 2023*

*Håvard Ullebust Hagen*

# Innhold

## Innhold

Sammendrag.....	1
Abstract.....	2
Forord.....	3
Ordliste.....	8
1. Introduksjon .....	9
1.1 Bakgrunn .....	9
1.2 Problemstilling og formål.....	10
2. Teori.....	11
2.1 Myr .....	11
2.2 Arealressurskart AR5.....	11
2.3 Myr kartlegging med bruk av fjernmåling .....	12
2.4 Bruk av multispektrale bilder til klassifisering .....	12
2.5 Bruk av hyperspektrale bilder til klassifisering .....	14
2.6 Preprosessering av hyperspektrale bilder.....	15
2.6.1 Dimensjonsreduisering.....	15
2.6.1.1 Prinsipalkomponentanalyse .....	15
2.6.2 Datasettbalansering .....	16
2.7 Klassifisering.....	17
2.7.1 SVM.....	18
2.7.2 Random Forest Classifier .....	20
2.7.3 K-Nærmeste Nabo.....	22
2.7.4 Logistic Regression.....	23
2.8 Evaluering av klassifiseringens nøyaktighet.....	24
3. Metode.....	31
3.1 Datasett .....	31
3.1.1 Arealressurskart AR5 .....	31
3.1.2 HDF5 Hyperspektrale data.....	35

3.1.3 Skydekkemaske.....	37
3.1.4 AR5-kartmaske.....	37
3.2 Datasett avledet fra hyperspektrale data.....	38
3.2.1 Prinsipalkomponent analysert hyperspektrale data.....	38
3.2.2 Maskert og skydekkemaskert data.....	40
3.3 Produkter avledet fra arealressurskart.....	40
3.4.1 Undersampling.....	41
3.4.2 Train-test split.....	41
3.4.3 Klassifisering – valg av klassifiseringsmodell.....	42
3.4.4 Valg av hyperparametere.....	42
4. Resultater.....	44
4.1 Klassifiseringsrapporten til modellene med fokus på myrklassen.....	44
4.2 Klassifiseringsrapporten til modellene med fokus på den «ikke-klassifiserte» klassen.....	45
4.3 Nøyaktighet og klassifiseringsfeil.....	45
4.4 Forvirringsmatrisene.....	46
4.5 Kappa koeffisient og MCC-score.....	49
4.6 Area Under the Curve (AUC).....	50
5. Diskusjon.....	51
5.1 Valg av hyperparametere.....	51
5.2 AR5 som treningsdata.....	52
5.3 Diskusjon av resultater fordelt på evalueringsmål.....	53
5.4 Valg av klassifiseringsmodell.....	57
6. Konklusjon.....	58
7. Referanser.....	59

# Tabelliste

Tabell 1. Tabellen beskriver hvordan man kan tolke AUC-verdi i forhold til klassifikasjonsevaluering.....	27
Tabell 2. Tolkningen av Kappa-koeffisienten og hvordan denne beskriver sammenhengen mellom observasjoner og prediksjoner.....	28
Tabell 3. Hvordan matthews korrelasjonskoeffisient (MCC) kan tolkes i forhold til korrelasjonen mellom modellens predikerte klasser og den faktiske klassen .....	29
tabell 4. Ulike kodeverdier benyttet i AR5, deres egenskapsverdi, og definisjonen av disse.....	31
tabell 5. De forskjellige masketyperne påført datasettet, tilhørende pikselverdier og pikselverdiens gyldighet for masketyper.....	38
Tabell 6. Hyperparametere for SVM, funnet av randomsearchCV .....	43
Tabell 7. Hyperparametere for RandomForestClassifier, funnet av randomsearchCV .....	43
Tabell 8. Hyperparametere for LogisticRegression, funnet av gridsearchCV .....	43
Tabell 9. Hyperparametere for K-Nearest-Neighbor, funnet av gridsearchCV .....	43
Tabell 10. Klassifiseringsmodellens evalueringsmål fra modellens klassifiseringsrapport for myrklassen. SVM vises med fargen blå, RandomForest med fargen grønn, Logistisk Regresjon med fargen grå, K-Nearest neighbors med en lys brun farge og K-Nearest neighbors uten undersampling vises med fargen rød.....	44
Tabell 11. Klassifiseringsmodellens evalueringsmål fra modellens klassifiseringsrapport for den «ikke-klassifiserte» arealklassen. SVM, RandomForest, Logistisk Regresjon og K-Nearest neighbors representeres av de samme fargene som i tabell 10.....	45
Tabell 12. Viser klassifiseringsmodellens Nøyaktighet og klassifiseringsfeil. SVM, RandomForest, Logistisk Regresjon og K-Nearest neighbors representeres av de samme fargene som i tabell 10.....	45
Tabell 13. Klassifiseringsmodellens MCC-score og Kappa-score. SVM, RandomForest, Logistisk Regresjon og K-Nearest neighbors representeres av de samme fargene som i tabell 10.....	49
Tabell 14. Klassifiseringsmodellens AUC-score. SVM, RandomForest, Logistisk Regresjon og K-Nearest neighbors representeres av de samme fargene som i tabell 10.....	50

# Figurliste

Figur 1. Aksene $x_1$ og $x_2$ i figuren representerer det originale datasettets variabler og PC1 og PC2 er prinsippal komponentene.....	16
Figur 2. Figuren visualiserer hvordan SVM danner den maksimale marginen. Dimensjoner høyere enn 2 er utfordrende å visualisere (Raschka & Mirjalili, 2017). ....	19
Figur 3. Figuren viser klassifiseringsprosessen til en RFC -modell: d=nye data som skal klassifiseres (Belgiu & Drăguț, 2016).....	21
Figur 4. figuren illustrerer en forvirringsmatrise der: TP = sanne positive hendelser, FN = Falske negative hendelser, FP = falske positive hendelser, TN = Sanne negative hendelser.....	25
Figur 5 AR5-kart hvor kun klasse 99- ikke klassifisert er inkludert. Alle piksler med en verdi over 0 anes som klasse 99 .....	33
Figur 6. AR5-kart hvor klasse 60 - myr er inkludert. Alle piksler med pikselverdi over 0 anees som myr-piksler. Piksler som synes, og ikke er mørkeblå som store deler av bildet, har pikselverdi over 0 .....	34
Figur 7. AR5-kartet der alle klasser er inkludert, pikselfargen representerer klassene i AR5-kartet. ....	34
Figur 8. Figuren viser en illustrasjon av det første bandet fra VNIR-kuben som inneholdt brukbare data.....	35
Figur 9. Figuren viser en illustrasjon av det første bandet fra SWIR-kuben som inneholdt brukbare data.....	36
Figur 10. Figuren viser en illustrasjon av bånd 1 som mangler data, der alle pikselverdiene er 0. Band 2 og 238, som er det siste båndet, følger samme trend. ....	36
Figur 11 X-aksen i figuren representerer den forklarte variansen og y aksen i figuren representerer antall komponenter.....	39
Figur 12. Figuren illustrerer de første 6 komponentene tilpasset oppgavens datasett...	40
Figur 13.FORVIRRINGSMATRISSE FOR LOGREG: Celle 0-0 = Sann Positiv (SP), Celle 0-1 = Falsk Negativ (FN), Celle 1-0 = Falsk Positiv (FP), Celle 1-1 Sann Negativ (SN) (Se figur 4). Tallene i figuren viser til antall hendelser for hver av resultatene.....	46
Figur 14. Forvirringsmatrise for KNN: Celle 0-0 = Sann Positiv (SP), Celle 0-1 = Falsk Negativ (FN), Celle 1-0 = Falsk Positiv (FP), Celle 1-1 Sann Negativ (SN) (Se figur 4). Tallene i figuren viser til antall hendelser for hver av resultatene .....	47
Figur 15. Forvirringsmatrise for SVM: Celle 0-0 = Sann Positiv (SP), Celle 0-1 = Falsk Negativ (FN), Celle 1-0 = Falsk Positiv (FP), Celle 1-1 Sann Negativ (SN) (Se figur 4). Tallene i figuren viser til antall hendelser for hver av resultatene .....	48
Figur 16 Forvirringsmatrise for RFC: Celle 0-0 = Sann Positiv (SP), Celle 0-1 = Falsk Negativ (FN), Celle 1-0 = Falsk Positiv (FP), Celle 1-1 Sann Negativ (SN) (Se figur 4). Tallene i figuren viser til antall hendelser for hver av resultatene .....	49
Figur 17. Modellenes ROC-kurve. X-aksen representerer den falske positive raten. Y-aksen representerer den Sanne positive raten. ....	50



# Ordliste

Forkortelse	Engelsk	Norsk
SVM	Support vector machine	Støtte-vector maskin
RFC	Random Forest Classifier	Tilfeldig skog klassifiseringsystem
LogReg	Logistic Regression	Logistisk regresjon
KNN	K-Nearest Neighbor	K-nærmeste nabo
MCC	Matthews Correlation Coefficient	Matthews korrelasjons koeffisient
AUC	Area Under the Curve	Område under kurven
ROC	Receiver Operating Characteristic	
FKB		Felles KartdataBase
PCA	Principal Component Analysis	Prinsipalkomponentanalyse/ Hovedkomponentanalyse
VNIR	Visible Near InfraRed	Synlig nær-infrarødt lys
SWIR	ShortWave InraRed	Kortbølget infrarødt lys
TP	True Positive	Sann positiv
TN	True Negative	Sann negativ
FP	False Positive	Falsk positiv
FN	False Negative	Falsk negativ

# 1. Introduksjon

## 1.1 Bakgrunn

Myra er en viktig naturressurs for Norge på grunn av dens egenskap til å lagre karbon, rens vann og dempe flom samtidig som den er opprettholder et stort biologisk mangfold. Myra er en type våtmark, der NINA (Norsk Institutt for Naturforskning) påstår at Norge består av 10% våtmark hvor myra står for 9% av. Norges dypeste myrer er på rundt 10 meter, der myrer kan vokse med 1 mm organisk materiale hvert år. Samtidig som myra dekker en estimert 9% av landarealet til Norge inneholder den også svært mye karbon. NINA estimerer at myra i Norge inneholder opptil 50Kg per kvadratmeter, som utgjør 50 000 tonn per kvadratkilometer. NINA går videre med at det er et lav-estimat og de virkelige verdiene kan være høyere (Kyrkjeeide, 2020).

Tradisjonelt sett har kartleggingen av myr krevd fysisk kartlegging av områder som mistenkes for å være myr. Denne fysiske kartleggingen skjer med manuell tolkning av flybilder. Denne prosessen er ofte tidskrevende, spesielt hvis det mistenkte området befinner seg langt unna kjørbare veier, da det blir utfordrende å verifisere om området faktisk er myr. På dette grunnlaget er det behov for mer tid- og kostnadseffektive metoder for å oppdage myr.

En teoretisk mulig metode er å bruke hyperspektrale bilder for å oppdage myr. Hyperspektrale data gir et rikt spektrum av data som kan brukes til å identifisere karakteristiske spektrale signaturer som er typiske for myr. Disse signaturer kan brukes til å utvikle algoritmer og klassifiseringsmetoder som kan brukes til å kartlegge og detektere myrområder på en enklere og mer effektiv måte.

Imidlertid er det fortsatt utfordringer knyttet til å oppdage myr ved bruken av hyperspektrale data. Noen av utfordringene inkluderer variasjoner i spektrale signaturer som er påvirket av forskjellige faktorer som høye datadimensjoner, vegetasjon, klassebalanse i datasettene, skydekke eller skygger. Det kan også være vanskelig å skille mellom myr og andre våtmarker som har lignende spektrale signaturer. På de hyperspektrale bildene er det også få piksler som utgir myr i kontrast

til andre piksler som tilsvarer andre naturtyper. Dette kan utgjøre en stor utfordring når det kommer til klassifiseringsalgoritmens egenskap til å oppdage myra på.

## **1.2 Problemstilling og formål**

I denne masteroppgaven vil det undersøkes om det er mulig å detektere myr automatisk ved hjelp av hyperspektrale data. Målet med oppgaven er å undersøke om det er mulig å utvikle en eller flere maskinlæringsalgoritmer som kan klassifisere myr med akseptable eller gode evalueringsmål. Det er på bakgrunn av det manuelle arbeidet som kreves for å kartlegge myr sammen med myras potensiale for karbongassutslipp, det er ønskelig å effektivisere myrkartleggingen.

## 2. Teori

### 2.1 Myr

Myrer kan dannes på ulike måter, men felles for all myr er at det alltid er et behov for høy fuktighet og underskudd av oksygen under nedbrytning av organisk materiale, slik at det dannes torv. Vann samles ofte opp i høytliggende områder eller flate sletter, noe som gjør det godt egnet til å danne en myr. Nedbør er en vanlig årsak til at vann samles opp, og det er derfor vanlig å finne myrer i områder det er mye nedbør. Andre faktorer, som smeltevann fra snø og is eller flom, kan også bidra til dannelsen av myrer.

Den største trusselen for myra i Norge er drenering, nedbygging og torvuttak. Å vedlikeholde myra sin høye vannstand er særdeles viktig for at myra skal opprettholde sine karbonlagrende egenskaper. Ødeleggelsen av myr tilsvarte omtrent 10% av Norges totale karbondioksidutslipp i 2013 (Kyrkjeeide, 2020). Estimater baserer seg på et areal på 3618 kvadratkilometer, mens NINA mener at arealet som faktisk ble drenert i 2013 var nærmere 7000 kvadratkilometer. En eventuell løsning for å restaurere myra kan være ved å heve vannstanden, som kan gjennomføres ved å tette grøfter. Tetting av grøfter er en svært tidskrevende og arbeidskraftig prosess, det vil derfor lønne seg å forhindre nedbyggingen av myr. For å forhindre nedbygginga av myra er det derfor viktig å ha en nøyaktig oversikt over hvor myra befinner seg.

### 2.2 Arealressurskart AR5

AR5 er et nasjonalt kartdatasett som dekker hele landet. Det beskriver arealressursene ut fra produksjonsgrunnlaget for jord- og skogbruk. Datasettets hovedformål er å dekke nødvendige behov innenfor arealplanlegging, landbruksforvaltning, landbruksnæringer og lovforvaltninger. I AR5 er landarealene fordelt i polygoner som blir beskrevet av spesifikke verdier. Disse verdiene er spesifikke areal typer, skogbonitet, treslag og grunnforhold. Hovedinndelingen for AR5 er arealtype (Kartverket, 2022).

AR5 er en del av, forvaltes og organiseres gjennom Felles KartdataBase (FKB), på lik måte med andre kartdata som setter søkelys på detalj (Kartverket, 2022). AR5

datasettet må kontinuerlig oppdateres med ny informasjon basert på opplysninger fra saksbehandlinger, synfaring eller flybilder.

## **2.3 Myr kartlegging med bruk av fjernmåling**

Et av hovedproblemene med å kartlegge myr er at myrområder kan strekke seg over store avstander. I Norge kan myra befinne seg på urbane områder, som på fjellet eller langt unna kjørbare veger. Dette gjør det utfordrende å fysisk kartlegge myrområder i Norge. Ved å bruke fjernmåling kan en kartlegge myra ved å bruke bilder fra fly eller satellitter, slik at en slipper å oppsøke området fysisk.

Fjernmåling åpner muligheten til å samle inn mye data om myr over store landarealer ved å bruke spektrale sensorer festet til et fly, en satellitt eller en drone. Fjernmåling gjør det mulig å samle informasjon om blant annet vegetasjonstype, vanninnhold, topografi og spektrale signaturer fra myra og dens omgivelser.

Det har tidligere vært forsøk å arealklassifisere myr ved bruk av fjernmåling. I fjor skrev Osmoen en masteroppgave som undersøkte om det var mulig å utføre en vellykket arealklassifisering av myr med styrt klassifisering, ved bruk av ortofoto og laserdata som grunnlagsdata (Osmoen, 2022). I oppgaven utnytter Osmoen en styrt klassifisering med hjelp av klassifiseringsalgoritmen Support Vector Machine. Oppgaven til Osmoen oppnådde en total nøyaktighet på 83% og en Kappa-koeffisient på 0.72. Dette er resultater som Osmoen anser som gode, og oppgaven konkluderer dermed at det er mulig å gjennomføre en vellykket arealklassifisering ved hjelp av objektbasert styrt klassifisering. Osmoen setter også fokus på at det er viktig å sette krav til gode treningsdata (Osmoen, 2022).

Basert på resultatene til Osmoen kan en påstå at potensialet innenfor myrkartlegging ved bruk av fjernmåling er lovende.

## **2.4 Bruk av multispektrale bilder til klassifisering**

Klassifisering ved bruk av multispektrale bilder, innebærer at piksler, eller områder i et bilde forbeholdes en klasse eller materiale. Vanligvis gjøres dette med en

maskinlæringsalgoritme, som er trent på et datasett der de ønskelige klassene er kjent. Det spesielle med multispektrale bilder er deres evne til å se forskjellen på objekter i bildene basert på objektenes refleksjonsegenskaper i ulike bånd, noe som kan resultere i bedre klassifiseringsresultater.

Multispektrale bilder er særdeles nyttig innen applikasjonsområder der det er behov for en nøyaktig identifikasjon og klassifisering av materiale eller objekter. Noen eksempler på disse bruksområdene kan være jordbrukskartlegging eller urban planlegging. Samtidig som det er mange positive sider ved bruken av multispektrale bilder til klassifisering, er det også noen utfordringer knyttet til det. Et av kravene til en god klassifisering er kvaliteten på treningsdatasettet som brukes av klassifikasjonsalgoritmen til å trene algoritmen på klassene (Osmon, 2022). En må også velge en passende klassifikasjonsalgoritme. En kompleks oppgave krever ofte en mer kompleks algoritme som SVM, samtidig som en simpel oppgave kan nytte seg av en mindre regnekraftig algoritme som lineær regresjon.

Multispektrale bilder har tidligere blitt brukt til diverse formål innen fjernmåling. I 2020 skrev Ingvild Martini en masteroppgave om bruken av multispektrale bilder og laserdata til å finne Blågrønn faktor (BGF) i Fredrikstad. BGF stiller krav til bygningsprosjekter for å oppnå en bedre overvannshåndtering og bevare det biologiske mangfoldet. Oppgaven til Martini sammenligner sin egen bruk av multispektrale bilder mot oppgaven til Kailainathan (2019), som brukte hyperspektrale bilder for å finne BGF på eksisterende bymiljø (Martini, 2020).

Oppgaven til Martini konkluderte med at presisjonen på resultatene avtar ved overgangen fra hyperspektrale bilder til multispektrale bilder. Oppgavene viser at det gjennomsnittlige avviket i BGF økte med 0.027 ved å utnytte multispektrale bilder sammenlignet med hyperspektrale. Videre forklarer oppgaven at ved bruken av multispektrale bilder, så kan oppgavens metode utnyttes flere steder og over større områder enn ved bruken av hyperspektrale bilder. Dette er mulig ettersom metoden er mindre kompleks, og at det eksisterer multispektrale bilder i nesten hele Norge (Martini, 2020).

## 2.5 Bruk av hyperspektrale bilder til klassifisering

Hyperspektrale bilder inneholder informasjon om objekter i smale bånd av det elektromagnetiske spekteret. Vanlige bilder inneholder tre bånd; rød, grønn og blå. Hyperspektrale bilder kan inneholde hundrevis av bånd, der hvert bånd kan tilsvare en smal bølgelengde som kan gi detaljert informasjon om et objekts spektrale signatur. Hyperspektrale bilder er som regel laget på lik måte med multispektrale bilder, som ved bruken av fly, droner eller satellitter. Forskjellen mellom multispektrale og hyperspektrale bilder ligger i sensoren som er brukt for å fange bildene (Lu et al, 2020). Multispektrale sensorer samler inn et begrenset antall bånd innenfor det elektromagnetiske spektrumet, der antall bånd vanligvis er rundt noen titall. Båndene tatt ifra en multispektral sensor inneholder vanligvis informasjon om det synlige lyset og noen infrarøde områder. Ettersom hyperspektrale sensorer kan fange informasjon på flere hundre bånd kan de fange mer sensitive spektralsignaturer, som er nyttig når en ønsker å klassifisere objekter som er vanskelig å oppdage med synlig lys (Lu et al, 2020).

Ved overgangen fra multispektrale bilde til hyperspektrale bilder til klassifisering kan det observeres at nøyaktigheten til modellen forbedres, mens prosesseringstiden øker. Samtidig må hyperspektrale bilder også gjennomgå en mer omfattende preprosessering enn multispektrale bilder (Martini, 2020). Båndantallet i hyperspektrale bilder omgjør datasettet som er avledet av bildene til et høydimensjonalt datasett, og håndteringen av disse dimensjonene er en del av preprosesseringsstegene som må til før det brukes til klassifisering. En metode som ofte brukes for dimensjonsreduisering i datasett er en prinsipalkomponentanalyse (Raschka and Mirjalili, 2017).

Videre kan hyperspektrale bilder være påvirket av usikkerheter. Hyperspektrale bilder kan være mer utsatt for støy i atmosfæren, der den vanligste formen er skyer, eller skydekke, og skygger (Sedano et al, 2011). Ved klassifisering der det er ønskelig å klassifisere underrepresenterte klasser, som arealklassen myr, vil det være viktig å oppnå en god klassebalanse.

## 2.6 Preprosessering av hyperspektrale bilder

Preprosessering er å forbedre rådata til behandling før de kan brukes i en maskinlæringsmodell. Preprosessering er en kritisk del av maskinlæringsprosessen, da den kan påvirke ytelsen og nøyaktigheten til modellen. Preprosessering kan være atmosfærekorreksjoner, dimensjonsreduksjon eller datasettbalansering (Huang et al, 2015). Valg av preprosesseringsteknikker avhenger av datasettet og problemet som skal løses.

### 2.6.1 Dimensjonsredusering

Med dimensjonsredusering så er det et ønske om å redusere datasettets kompleksitet for å oppnå best mulig informasjon om datasettet. Datasett som brukes i maskinlæring er ofte komplekse med mange funksjoner og variabler. Ved å bruke dimensjonsreduseringsteknikker som PCA eller LDA, kan de bidra til å håndtere kompleksiteten og hente ut den mest relevante informasjonen (Raschka & Mirjalili, 2017). Ettersom hovedkomponentanalyse (PCA) og lineær diskriminantanalyse (LDA) reduserer antall dimensjoner i datasettet, samtidig som de mest informative dataene bevares, vil dette bidra til å avdekke skjulte mønster og sammenhenger i dataene. Dette kan dermed gi en bedre innsikt i datasettets struktur (Raschka & Mirjalili, 2017).

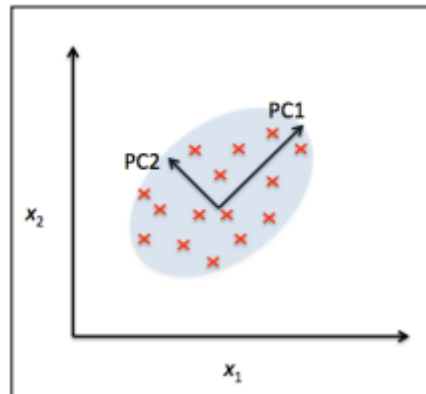
#### 2.6.1.1 Prinsipalkomponentanalyse

Prinsipalkomponentanalyse eller Principal Component Analysis (PCA) er en lineær transformasjonsteknikk som ikke er styrt. PCA er mest kjent for sin brukbarhet innenfor dimensjonsredusering. PCA hjelper til å oppdage skjulte spor i datasett, basert på sammenhenger mellom datasettets egenskaper (Raschka & Mirjalili, 2017).

Hovedmålet til PCA er å finne retningen til den maksimale variansen i høy-dimensjonal data, for å så projisere det på et nytt rom med like mange eller færre dimensjoner enn det opprinnelige datasettet. Retningens maksimale varians blir satt til å være den første prinsipalkomponenten (PC1). Den andre prinsipalkomponenten (PC2) vil inneholde



nest mest varians og stå vinkelrett på prinsipalkomponenten med mest varians (PC1), (figur 1). Dersom det oppstår flere prinsipalkomponenter, vil de følge samme trend som den andre prinsipalkomponenten (PC2). Mesteparten av informasjonen som eksisterer i datasettet vil befinne seg i de første komponentene (Raschka & Mirjalili, 2017).



FIGUR 1. AKSENE  $x_1$  OG  $x_2$  I FIGUREN REPRESENTERER DET ORIGINALE DATASETTETS VARIABLER OG PC1 OG PC2 ER PRINSIPAL KOMPONENTENE.

## 2.6.2 Datasettbalansering

I maskinl ring er det vanlig at noen klasser er overrepresentert og andre er sv rt underrepresentert. Om det er  nskelig   klassifisere en av klassene som er over eller underrepresenterte kan en bruke datasettbalansering eller en klassebalansering, ettersom klassifiseringsn yaktigheten kan v re sv rt svekket av ubalansen (Bae et al, 2021). Dataene kan v re samlet inn over store arealer hvor de fleste arealtypene er skog eller gress, men det kan ogs  v re myr p  mindre areal. Ettersom datasettet er ubalansert med tanke p  klassene kan dette f re til en ubalansert l rings situasjon, der modellen kan overvurdere de overrepresenterte klassene og undervurdere de underrepresenterte klassene. For   h ndtere dataskjevheten brukes datasettbalansering (Bae et al, 2021). Det gj res ved   oversample eller undersample datapunkter fra de ulike klassene, slik at de representeres likt i datasettet. Modellen vil dermed oppn  en mer rettferdig og p litelig evne til   klassifisere. Undersampling fjerner eksempler fra de overrepresenterte klassene i datasettet. Undersampling kan gj res ved   tilfeldig fjerne eksemplene via tilfeldig undersampling eller ved centroidsundersampling der det velges ut hvilke eksempler som fjernes (Zheng et al, 2021).

RandomUnderSampler er en undersamplingsteknikk som kan benyttes i Python for å oppnå en balanse i datasettet. RandomUnderSampler reduserer antallet eksempler i de overrepresenterte klassene ved tilfeldige valg. Dette vil medføre at antallet eksempler i alle klassene blir mer balanserte. RandomUnderSampler kan bidra til å forbedre klassifiseringsalgoritmens evne til å lære fra alle klassene og til slutt gi et mer rettferdig og pålitelig klassifiseringsresultat (Zheng et al, 2021).

## 2.7 Klassifisering

Maskinlæring kan tilby muligheten til å blant annet automatisk klassifisere objekter eller arealtyper. De to mest brukte klassifiseringsmetodene er styrt klassifisering og ikke-styrt klassifisering. Styrt klassifisering benytter treningsdata til å trene en klassifiseringsalgoritme. Disse dataene inneholder informasjon om egenskapene til utvalgte objekter eller arealtyper som representerer de ulike klassene i et datasett. Med dette kan modellen lære å klassifisere nye objekter eller arealtyper i datasettet. For å evaluere nøyaktigheten til klassifiseringsresultatet brukes valideringsdataen fra klassifiseringen.

Ved styrt klassifisering innføres det såkalt fasitdata eller treningsdata til bruk i både trening og validering av modellen. Modellen lærer å generalisere fra treningsdataene for å kunne klassifisere nye data. Noen eksempler på klassifiseringsmodeller som utnytter styrt klassifisering kan være støtteverksnettverk (SVM) (Raschka & Mirjalili, 2017). Tilfeldig Skog klassifisering (RFC) eller Logistisk Regresjon (LogReg). Et eksempel på en instansbasert klassifiseringsmodell, som ikke nødvendigvis trenger treningsdata er K-Nærmeste Nabo (KNN) (Ostfeld & Salomons, 2005).

Etter at en klassifiseringsmodell er valgt, er det viktig å velge riktige hyperparametere for modellen. Hyperparametere kan bidra til å øke modellens prestasjon, men kan også bidra til å overtilpasse eller undertilpasse modellen til treningsdataene (Feurer & Hutter, 2019). For å finne de best mulige hyperparameterne til en klassifiseringsmodell kan en utnytte hyperparameteroptimaliseringsmetoder som et rutenettsøk (GridSearchCV i Python syntaks) eller et tilfeldig søk (RandomizedSearchCV i Python syntaks).

Hyperparameteroptimalisering er en metode som utnyttes for å forbedre klassifiseringsnøyaktigheten. Den forbedrede klassifiseringsnøyaktigheten kommer av en justering av algoritmens parametere. (Raschka & Mirjalili, 2017). Justeringen er iterativ, og det er utfordrende å forutsi hvilke av hyperparameterverdiene som vil gi den beste klassifiseringsnøyaktigheten før justeringen. (Feurer & Hutter, 2019). For hver kombinasjon av hyperparameterverdier som velges for å oppnå bedre klassifisering, så må modellen kjøres på nytt. Ettersom modellen må kjøres på de nye verdiene, som skjer mange ganger i løpet av en hyperparameteroptimalisering, vil prosesseringstiden være lang, og store datasett gjør prosesseringstiden enda lengre.

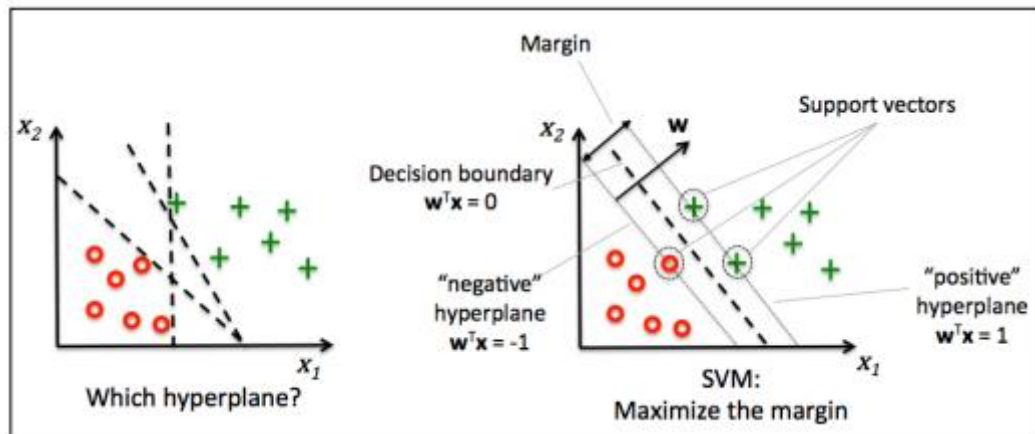
Rutenettsøk er den mest brukte hyperparameteroptimaliseringsmetoden. Med denne metoden definerer brukeren et sett med verdier for hver hyperparameter som skal optimaliseres, der modellen skal trenes og evalueres for alle kombinasjoner av hyperparameterverdier (Feurer & Hutter, 2019). Svakheten til rutenettsøk er prosesseringstiden som øker eksponentielt med antall brukerdefinerte hyperparameterverdier.

En kan alternativt utnytte et tilfeldig søk dersom prosesseringstiden til en klassifiseringsmodell blir for lang. Et tilfeldig søk velger istedenfor et tilfeldig utvalg av hyperparameterkombinasjoner fra brukerdefinerte hyperparameterverdier. Tilfeldig søk virker ofte i hovedsak bedre enn et rutenettsøk i praksis, ettersom noen hyperparameterverdier har større betydning enn andre, som viser seg ofte kan være tilfeldig (Bergstra & Bengio, 2012).

## **2.7.1 SVM**

Støtteverksnettverk eller «support vector machine» (SVM) ofte også kalt for et «Støttevektor klassifiseringssystem (Support Vector Classifier (SVC))» er en populær maskinlæringsalgoritme innen styrt klassifisering. Formålet med SVM er å maksimere en margin som defineres som en avstand mellom to hyperplan, i et høydimensjonalt rom (Suthaharan, 2016). Hyperplanet skal kunne best mulig separere de ulike klassenes egenskaper fra hverandre. Ved å maksimere avstanden mellom klassene dannes det best mulige hyperplanet (Suthaharan, 2016). Ved å analysere og identifisere treningsdataene som eksisterer i de ytre grensene til hver klasse, som ofte omtales som

støttevektorer (support vectors), kan de beste hyperplanene dannes. I en situasjon der det er to dimensjoner, vil det separerende hyperplanet være den stiplede linjen i figur 3 som separerer planet. Marginen SVM har som formål å maksimere, vil være en vinkelrett avstand mellom det separerende hyperplanet og støttevektorene som er nærmest det separerte hyperplanet (Raschka & Mirjalili, 2017).



FIGUR 2. FIGUREN VISUALISERER HVORDAN SVM DANNER DEN MAKSIMALE MARGINEN. DIMENSJONER HØYERE ENN 2 ER UTFORDRENDE Å VISUALISERE (RASCHKA & MIRJALILI, 2017).

SVM er kjent for å håndtere komplekse ikke-lineære separasjonsproblemer ved bruk av ulike kjernefunksjoner som kan omforme dataene til et rom i en høyere dimensjon. SVM er blant annet populært innen bildeklassifisering (Gewali et al, 2018), tekstklassifisering eller medisinsk diagnostikk. Ved bruk av SVM er det viktig å velge riktig kjernefunksjon og å velge riktige hyperparametere for å oppnå best mulig klassifiseringsresultat. De hyperparameterne som oftest justeres er parameterne kjernefunksjonen, C og klassevekten.

## Kjerne

Kjernefunksjonen (Kernel i Python syntaks) bestemmer den matematiske funksjonen som brukes for å omgjøre dataene til et rom i en høyere dimensjon. Kernel-valget avhenger av dataene sin underliggende struktur. Vanlige kernel-typer som utnyttes i SVM er lineær-kernel, polynomisk-kernel og RBF (radial basis function) kernel.

## C parameteren

C-parameteren styrer hvordan feilklassifisering av treningsdataene straffes. Dersom C har en høy verdi, kan det føre til en strengere straff for feilklassifisering, som kan øke kompleksiteten til modellen med en smal margin mellom klassene. En lavere verdi av C kan føre til en kan oppnå flere feilklassifiseringer, men også en enklere modell med en bredere margin mellom klassene. Valget av C parameteren avhenger derfor av hvor viktig det er å unngå feilklassifisering.

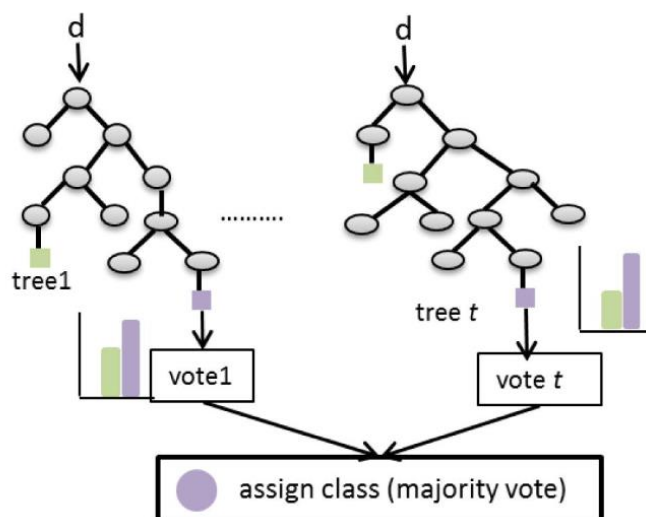
## Klassevekt

For å behandle ubalanserte klasser i et datasett, kan klassevekten (`class_weight` i Python syntaks) parameteren utnyttes. Ved å utnytte `class_weight` kan en tildele hver klasse i datasettet en vekt, slik at det er en god balanse mellom klassene under treningen av modellen. `Class_weight` er vanligvis satt som balansert (`balanced`).

### 2.7.2 Random Forest Classifier

En annen populær maskinlæringsalgoritme som ofte brukes til klassifisering er et Tilfeldig Skog Klassifiseringssystem (Random Forest Classifier (RFC). RFC-modellen består av en kombinasjon av ulike og individuelle beslutningstrær. Samlingen av disse beslutningstrærne blir ofte omtalt som en skog. For hver eneste beslutning som tas i en RFC, foretas det en avstemning blant alle beslutningstrærne i skogen. Den endelige klassen blir bestemt av flertallet av stemmene (Figur.3) (Belgiu & Drăguț, 2016).

## B CLASSIFICATION PHASE



FIGUR 3. FIGUREN VISER KLASSIFISERINGSPROSESSEN TIL EN RFC -MODELL: D=NYE DATA SOM SKAL KLASSIFISERES (BELGIU & DRÄGUȚ, 2016).

Beslutningstrærne lages ved å dra en delmengde av treningsdataene gjennom en erstatningsprosess (bagging approach.) En bagging approach går ut på at den samme delmengden kan bli gjenvalgt flere ganger, der andre delmengder kan ende opp med å aldri bli valgt.

RFC blir ofte brukt ved fjernmåling, da dens høye nøyaktighet på klassifikasjonene er nødvendig (Belgiu & Drăguț, 2016). Klassifiseringsmodellen er også mindre sensitiv til kvaliteten på treningsprøvene og til overtilpassing (overfitting) enn andre ofte brukte maskinlæringsklassifiseringsmodeller. Den lave sensitiviteten kommer fra antallet beslutningstrær som produseres ved å tilfeldig velge en delmengde av treningsdataene og en delmengde av variabler for å dele hvert tre-node (Belgiu & Drăguț, 2016).

Hyperparameterne som ofte finjusteres for å oppnå best mulig modellprestasjon for RFC, kan være antall estimatorer, maksimal dybde eller maksimale egenskaper.

## Antall estimatorer

Antall estimatorer (`N_estimators` i Python syntaks) er antall beslutningstrær i en random forest. En høy verdi for `N_estimators` kan bidra til en sterkere og mer pålitelig modell, men den kan også øke treningstiden.

## Maksimal dybde

Maksimal dybde (`max_depth` i Python syntaks) er den maksimale dybden til hvert beslutningstre. Ved å justere `max_depth` kan du justere kompleksiteten og kapasiteten til modellen. Ved en høy verdi av `max_depth` kan modellen oppdage komplekse mønstre, det kan føre til en overtilpassing.

## Maksimale egenskaper

Maksimale egenskaper (`Max_features` i Python syntaks) er antall funksjoner som skal tas til betraktning før hver node splittes. Ved å angi `max_features` som en lavere verdi kan det føre til variasjon mellom trærne i en random forest, noe som kan redusere overtilpassing.

### 2.7.3 K-Nærmeste Nabo

K-nærmeste Nabo (K-Nearest Neighbor, KNN) er en «case-basert læringsmetode», som forbeholder alle treningsdata til klassifiseringsprosessen. I case-basert læringsmetode, eller instans-basert læring innenfor maskinlæring, er kunnskapen til metoden basert på tidligere erfaringer eller saker (cases). Dette gjør den istedenfor å bygge en modell som representerer dataene. Den lærer derfor av tidligere observasjoner (Guo et al, 2003).

Når et nytt datapunkt skal klassifiseres, blir det sammenlignet med de eksisterende treningsdatapunktene basert på en spesifikk avstandsmåling, der euklidisk avstand ofte utnyttes. KNN-algoritmen finner de K nærmeste treningsdatapunktene (naboene) til det ukjente datapunktet basert på avstanden, hvor K er en parameter styrt av brukeren (Guo et al, 2003).

Klassifiseringen av det ukjente datapunktet blir så avgjort ved å ta flertallet av klassene blant de  $K$  nærmeste naboene. Med andre ord, hvis de fleste av de  $K$  nærmeste naboene tilhører en bestemt klasse, blir det ukjente datapunktet også klassifisert tilhørende den klassen.

## 2.7.4 Logistic Regression

Logistisk regresjon (Logistic Regression (LogReg)) er en maskinlæringsalgoritme som brukes ved klassifiseringsproblemer. Den tar for seg binære og lineære klassifiseringsproblemer (Raschka & Mirjalili, 2017). Med en logistisk regresjon modelleres sannsynligheten for at gitt data tilhører en spesifikk klasse. Logistisk regresjon er relativt rask sammenlignet med andre styrte klassifiseringsteknikker som SVM, men dette kan føre til svakheter i modellens nøyaktighet. Hurtigheten og den svekkede nøyaktigheten til modellen kommer av at den er enklere enn andre modeller som SVM (Raschka & Mirjalili, 2017). Den deler også et lignende problem med lineær regresjon, hvor begge modellene er for enkle og egner seg dårlig til spesielle og komplekse sammenhenger mellom variabler i datasett (Chang, 2020).

En logistisk regresjon er i all hovedsak en lineær modell som skapes med å tilpasse en logistisk funksjon til treningsdataene. Funksjonen inntar en lineær kombinasjon av attributtverdiene og utnytter en sigmoidfunksjon til å produsere en sannsynlighet som befinner seg som en reel verdi mellom 0 eller 1 (Raschka & Mirjalili, 2017). Det utnyttes flere parametere for å kalibrere regresjonsmodellen i maskinlæring, der parametere som oftest justeres er  $C$  og  $L1/L2$

### C-verdi

$C$ -verdien er en regulariseringsparameter som utnyttes for å begrense den logistiske regresjonsmodellen. Mindre  $C$ -verdier øker regulariseringen, der større verdier vil svekke påvirkningen fra regulariseringen. Dersom små  $C$ -verdier utnyttes, så vil det resultere i en enklere modell som ofte kan føre til en «underfitting» (en



undertilpassing) av dataen. Ved bruken av større C-verdier kan modellen oppnå en «overfitting» (en overtilpassing) av dataen.

## L1 og L2

L1 og L2 er reguleringsparametere som straffer kompleksiteten til modellen, som videre fører til en forhindring av overtilpassing. L1-regularisering (LASSO-regularisering) gjør det mulig for modellen å velge de mest informasjonsrike attributtene, samtidig som den ignorerer de mindre informasjonsrike og irrelevante attributtene. L2-regularisering (Ridge-regularisering) bidrar til å forhindre at modellen blir for kompleks og spesifikk for treningsdataene.

## 2.8 Evaluering av klassifiseringens nøyaktighet

For å få et godt innblikk i hvor vellykket klassifiseringen har vært er det viktig å gjøre en nøyte evaluering av klassifiseringsresultatene. Flere forskjellige nøyaktighetsmål kan utnyttes for å evaluere klassifiseringen.

- Forvirringsmatrise
- Klassifiseringsfeil og nøyaktighet
- Klassifiseringsrapporten
- Area Under the Curve og Receiver Operating Characteristic (AUC-ROC)
- Kappa koeffisient
- Matthews korrelasjonskoeffisient

### Forvirringsmatrise

Forvirringsmatrisen er en kvadratisk kontingens-tabell. Den brukes for å evaluere prestasjonen til en klassifikasjonsmodell ved å sammenligne de faktiske klassene med de predikerte klassene (Raschka & Mirjalili, 2017). En forvirringsmatrise består av fire ulike celler:

1. Sanne positive (True positive TP) observasjoner der den faktiske klassen og den predikerte klassen er positive

2. Falske negative (False negative FN) observasjoner der den faktiske klassen er positiv, mens den predikerte klassen er negativ
3. Falske positive (False positive FP) observasjoner der den faktiske klassen er negativ, mens den predikerte klassen er positiv
4. Sanne negative (True negative TN) observasjoner der den faktiske klassen og den predikerte klassen er negative

		Predikert klasse	
		Positiv	Negativ
Faktisk klasse	Positiv	TP	FN
	Negativ	FP	TN

FIGUR 4. FIGUREN ILLUSTRERER EN FORVIRRINGSMATRISE DER: TP = SANNE POSITIVE HENDELSER, FN = FALSKE NEGATIVE HENDELSER, FP = FALSKE POSITIVE HENDELSER, TN = SANNE NEGATIVE HENDELSER

En forvirringsmatrise gir en oversikt over hvor godt modellen klassifiserer observasjoner i de forskjellige klassene. Den gir også et innblikk i hvilke typer feil modellen kan gjøre. Forvirringsmatrisen kan brukes til å identifisere spesifikke områder i modellen som har lavere ytelse (Raschka & Mirjalili, 2017). Om modellen har en tendens til å gjennomføre flere falske positive enn falske negative, kan modellen justeres ut ifra denne informasjonen.

## Klassifiseringsfeil og nøyaktighet

Feil i predikasjonene til en klassifiseringsmodell kalles en klassifiseringsfeil. De er to forskjellige klassifiseringsfeil, falsk positiv og falsk negativ. Falsk positiv oppstår når

modellen feilaktig predikerer et positivt resultat, selv om svaret i realiteten er negativt. Ved falsk negativ skjer det motsatte. Modellen predikerer feilaktig et negativt resultat, når svaret egentlig er positivt.

Nøyaktighet er en metrikk som brukes for å evaluere om klassifiseringsmodellen presterer godt. Ved å dele summen av antall sanne positive og sanne negative predikasjoner på totalt antall predikasjoner kan det utregnes en nøyaktighetsverdi.

Formelen blir slik:

$$\text{Nøyaktighet} = \frac{\text{Antall sanne positive predikasjoner (TP)} + \text{Antall sanne negative predikasjoner (TN)}}{\text{Totalt antall predikasjoner}}$$

Generelt sett vil en høy nøyaktighetsverdi og en lav klassifiseringsfeil angi en god modell, men det er viktig å ikke bare se på de to metrikkene når modellen evalueres. Dette gjelder spesielt om det er en ubalanse i klassene, slik det ofte er når myr skal klassifiseres. Andre evalueringer som AOC-ROC, Matthews korrelasjonskoeffisient eller kappa koeffisienten er viktig å bruke for å evaluere modellens fullstendige prestasjon.

## **Klassifiseringsrapporten**

Klassifiseringsrapporten til en klassifiseringsalgoritme oppgir presisjonen (Precision), sensitiviteten (Recall), F1-scoren og støtten (support) til hver klasse som inngår i algoritmen. Presisjonen, sensitiviteten, F1-scoren og støtten til klassene er vanlige evaluasjonsmålinger som kan gi en overflatisk forståelse av klassifiseringsmodellens ytelse. De sier noe om modellens evne til å klassifisere data riktig og oppdage positive tilfeller (Raschka & Mirjalili, 2017). Evaluasjonsmålinger som AUC, MCC eller Kappa-koeffisienten kan gi en dypere innsikt til underliggende informasjon om modellens prestasjon, ettersom de tar hensyn til både korrekte og feilaktige klassifiseringer.

1. I motsetning til nøyaktigheten, så ser presisjonen på hvordan det har blitt utført klassifisering innen hver klasse. Presisjon kan defineres som antall sanne positive hendelser (TP) innen en klasse i forhold til antall predikerte for samme klasse (TP) og (FP) (Raschka & Mirjalili, 2017).
2. Sensiviteten kan også kalles for den sanne positive raten. Sensiviteten er antall sanne positive tilfeller (TP) delt på summen av de sanne positive (TP) og de falske negative (FN) tilfellene. En høy sensitivetsverdi eller recall-verdi

indikerer at modellen er god på å oppdage sanne positive tilfeller og har lav sannsynlighet for å klassifisere falske negative tilfeller (Raschka & Mirjalili, 2017).

3. F1-score er en gjennomsnittsverdi av presisjon og sensitiviteten. F1-score kombinerer nøyaktigheten av positive prediksjoner (presisjon) og evnen til å kjenne igjen virkelige positive tilfeller (sensiviteten) (Raschka & Mirjalili, 2017).
4. Støtte viser til antall eksempler i hver klasse. Støtten kan gi et innblikk i datasettets klassebalanse som kan bidra til å tolke resultatet. Om en klasse er underrepresentert kan dette visualiseres her, slik at en kan balansere modellen om det er nødvendig.

### **Area Under the Curve og Reciver Operating Characteristic (AUC-ROC)**

Reciver operator characteristic (ROC) grafer er nyttige verktøy dersom en ønsker å velge modeller til klassifisering basert på modellens prestasjon i forhold til den falske positive (FP) raten og den sanne positive (TP) raten. Den diagonale linjen i en ROC-graf representerer tilfeldig gjetting. Dersom klassifiseringsmodellen havner under kurven, kan modellens prestasjon sees på som verre enn tilfeldig gjetting. En helt perfekt klassifisering vil da befinne seg i det øverste venstre hjørnet med en sann positiv rate på 1 og en falsk positiv rate på 0. Området under kurven (Area Under the Curve, AUC) finnes under ROC kurven (Raschka & Mirjalili, 2017).

AUC-verdien er et tall mellom 0 og 1, der høyere verdier tilsvarer en bedre ytelse. En AUC-verdi på 0.5 vil tilsvare at modellen ikke predikerer bedre enn ren gjetting, men en AUC-verdi på 1 vil tilsvare en perfekt modell. En verdi på 0.8 kan tyde til at modellen har en god prestasjon, selv om det fortsatt er rom til forbedring (Raschka & Mirjalili, 2017).

TABELL 1. TABELLEN BESKRIVER HVORDAN MAN KAN TOLKE AUC-VERDI I FORHOLD TIL KLASSIFIKASJONSEVALUERING

<b>AUC-verdi</b>	<b>Klassifikasjonsevaluering</b>
<b>0 – 0.5</b>	Dårlig
<b>0.5</b>	Tilfeldig
<b>0.6 – 0.7</b>	Moderat til god

<b>0.7 - 0.8</b>	God
<b>0.8 - 0.9</b>	Svært god
<b>0.9 - 1</b>	Nær perfekt

### Kappa-koeffisienten

Kappa koeffisienten er en evaluasjonsmåling som ofte brukes innen fjernmåling for å evaluere påliteligheten til en klassifiseringsmodell. Evalueringen baserer seg på å justere for en tilfeldig sammenheng mellom observasjoner og prediksjoner (Richards, 2013).

Kappa koeffisienten beregnes med å se på hvor mange ganger modellen og observasjonene samsvarer, også kalt observert samsvarsgrad (observed agreement) (McHugh, 2012). Dette sammenlignes med hvor mange ganger det forventes at de er tilfeldig samsvar, også kalt forventet samsvarsgrad (expected agreement). Hvis den observerte samsvarsgraden er høyere enn den forventede samsvarsgraden, kan dette indikere at modellen presterte bedre enn det som forventes ved tilfeldig gjetting. Den forventede samsvarsgraden trekkes fra den observerte samsvarsgraden og deler på differansen mellom 1 og den forventede samsvarsgraden. Denne metoden kan oppnå en kappa koeffisient mellom -1 og 1. Desto nærmere kappa koeffisienten er 1, desto bedre stemmer modellen overens med virkeligheten. På den andre siden vil en kappa koeffisient mellom 0 og -1 indikere at modellen klassifiserer verre enn tilfeldig gjetting (McHugh, 2012).

TABELL 2. TOLKNINGEN AV KAPPA-KOEFFISIENTEN OG HVORDAN DENNE BESKRIVER SAMMENHENGEN MELLOM OBSERVASJONER OG PREDIKSJONER

<b>Kappa-koeffisient</b>	<b>Beskrivelse av sammenhengen mellom observasjoner og prediksjoner</b>
<b>(-1) - 0</b>	Indikerer at sammenhengen er verre enn tilfeldig gjetting.
<b>0 - 0.2</b>	Svak sammenheng
<b>0.2 - 0.4</b>	Moderat sammenheng
<b>0.4 - 0.6</b>	Moderat til god sammenheng

<b>0.6 - 0.8</b>	God til veldig god sammenheng
<b>0.8 - 1</b>	Veldig god sammenheng
<b>1</b>	Perfekt sammenheng

### Matthews korrelasjonskoeffisient (MCC).

Matthews korrelasjonskoeffisient (MCC) brukes til å vurdere styrken og retningen av sammenhengen til to binære variabler. MCC er en statistisk måling som er passende for å evaluere en klassifiseringsmodells prediktive ytelse. Selve koeffisienten baserer seg på en 2 x 2 kontingenstabell (Chicco & Jurman, 2020) som består av antall observasjoner som er klassifisert riktige eller gale av modellen, sammenlignet med de faktiske klassene. Denne kontingens-tabellen blir også kalt forvirringsmatrisen.

MCC er en måling som brukes til å vurdere korrelasjonen mellom en modells predikerte klasse og den faktiske klassen modellen forsøkte å predikere. MCC tar på seg en verdi mellom -1 og 1, der -1 vil si at alle observasjonene ble feilpredikert og 1 indikerer en perfekt predikering av observasjonene. En MCC verdi på 0 vil tilsa at modellen ikke er bedre enn tilfeldig gjetting. MCC er spesielt bra for ubalanserte datasett der antall observasjoner mellom de to klassene er forskjellige. Dette kommer av at den tar hensyn til både positive og negative predikasjoner, den er mindre følsom for ubalanse i klassene og den tar hensyn til modellens evne til å skille mellom de ulike klassene (Chicco & Jurman, 2020).

TABELL 3. HVORDAN MATTHEWS KORRELASJONSKOEFFISIENT (MCC) KAN TOLKES I FORHOLD TIL KORRELASJONEN MELLOM MODELLENS PREDIKERTE KLASSER OG DEN FAKTISKE KLASSEN

<b>MCC-verdi</b>	<b>Beskrivelse</b>
<b>(-1)</b>	Perfekt motsatt korrelasjon
<b>(-0.7) - (-1)</b>	Sterk motsatt korrelasjon
<b>(-0.5) - (-0.7)</b>	Moderat til sterk motsatt korrelasjon
<b>(-0.3) - (-0.5)</b>	Mild til moderat motsatt korrelasjon
<b>(-0.1) - (-0.3)</b>	Mild motsatt korrelasjon

<b>0</b>	Ingen korrelasjon
<b>0.1 - 0.3</b>	Mild korrelasjon
<b>0.3 - 0.5</b>	Mild til moderat korrelasjon
<b>0.5 - 0.7</b>	Moderat til sterk korrelasjon
<b>0.7 - 1</b>	Sterk korrelasjon
<b>1</b>	Perfekt korrelasjon

# 3. Metode

## 3.1 Datasett

I denne oppgaven ble det benyttet fire ulike datasett. Ett av datasettene var til trening av klassifikasjonsmodellen, et annet ble brukt til å evaluere modellens ytelse og de to siste ble brukt til å preprosessere treningsdatasettet og evalueringsdatasettet.

Datasettene er i det geografiske koordinatsystemet Euref89 UTM32.

### 3.1.1 Arealressurskart AR5

AR5 er hovedsakelig inndelt i arealtyper. Denne inndelingen er basert på kriterier som er satt for vegetasjon og kulturpåvirkning. Tabell 4 viser de forskjellige verdiene til arealtypene (Ahlstrøm et al, 2019).

TABELL 4. ULIKE KODEVERDIER BENYTTET I AR5, DERES EGENSKAPSVERDI, OG DEFINISJONEN AV DISSE.

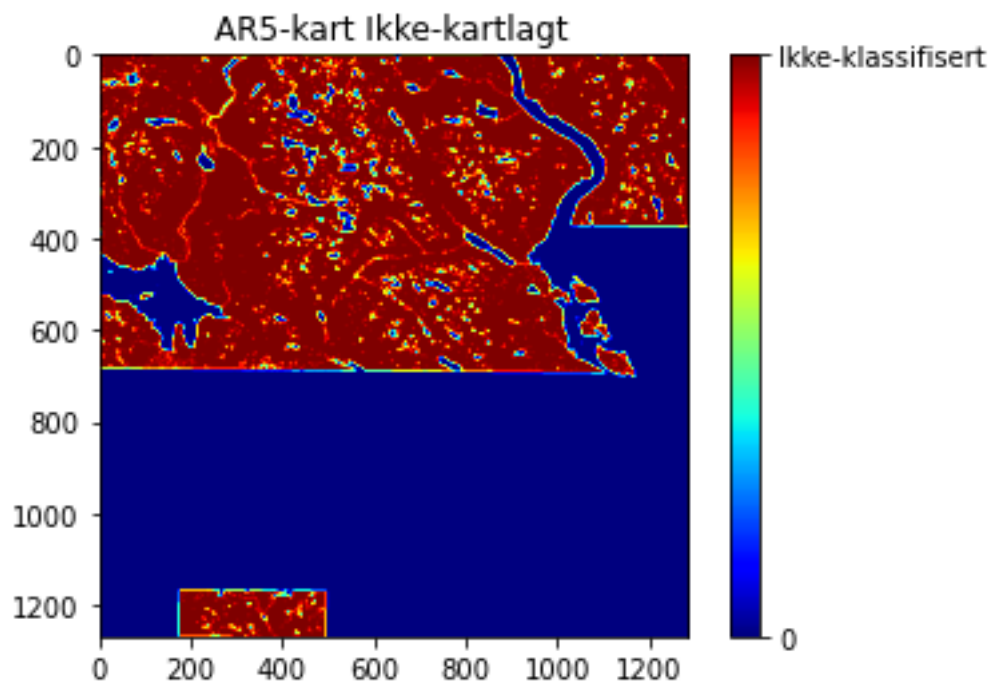
Kodeverdi	Egenskapsverdi	Definisjon
11	Bebygd	Areal som er utbygd eller i betydelig grad opparbeidet, samt tilstøtende arealer som i funksjon er nært knyttet til bebyggelsen.
12	Samferdsel	Areal som brukes til samferdsel.
21	Fulldyrka jord	Jordbruksareal som er dyrka til vanlig pløyedybde, og kan benyttes til åkervekster eller til eng, og som kan fornyes ved pløying.



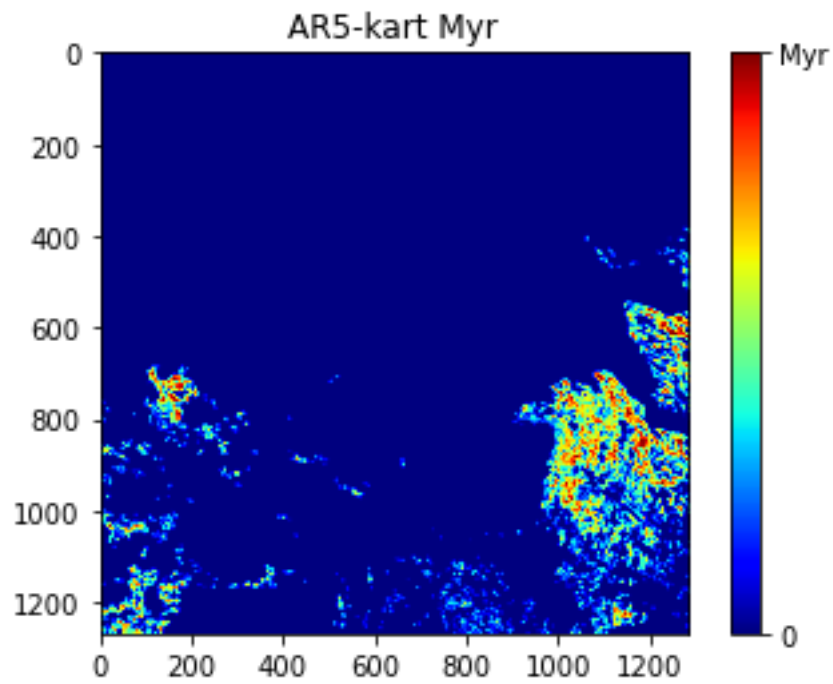
<b>22</b>	Overflatedyrka jord	Jordbruksareal som for det meste er rydda og jevna i overflata, slik at maskinell høsting er mulig.
<b>23</b>	Innmarksbeite	Jordbruksareal som kan benyttes som beite, men som ikke kan høstes maskinelt. Minst 50 % av arealet skal være dekt av godkjente grasarter eller beitetålende urter.
<b>30</b>	Skog	Areal med minst 6 trær per dekar som er eller kan bli 5 meter høye, og disse bør være jevnt fordelt på arealet.
<b>50</b>	Åpen fastmark	Fastmark som ikke er jordbruksareal, skog, bebyggd eller samferdsel.
<b>60</b>	Myr	Areal med myrvegetasjon og minst 30 cm tykt torvlag.
<b>70</b>	Snø/isbre	Blanding av isbre og snø som ikke smelter i løpet av sommeren.
<b>81</b>	Ferskvann	Ferskvann
<b>82</b>	Hav	Hav
<b>99</b>	Ikke kartlagt	Areal som er ukjent eller ubestemt.

Modellene ble trent på AR5 kartet, mer spesifikt på arealtypene 99 (ikke kartlagt) og 60 (myr). Med modellene ønskes det å klassifisere myr og arealtypene som ikke er kartlagt. Arealtypene som ikke er kartlagt, var enkle å klassifisere og kunne brukes til å trene klassifiseringsmodellen på en annen klasse enn myr. Under følger 3 figurer som illustrerer AR5-kartet som et bilde på omtrent 1250x1250 piksler, der X-aksen og Y-

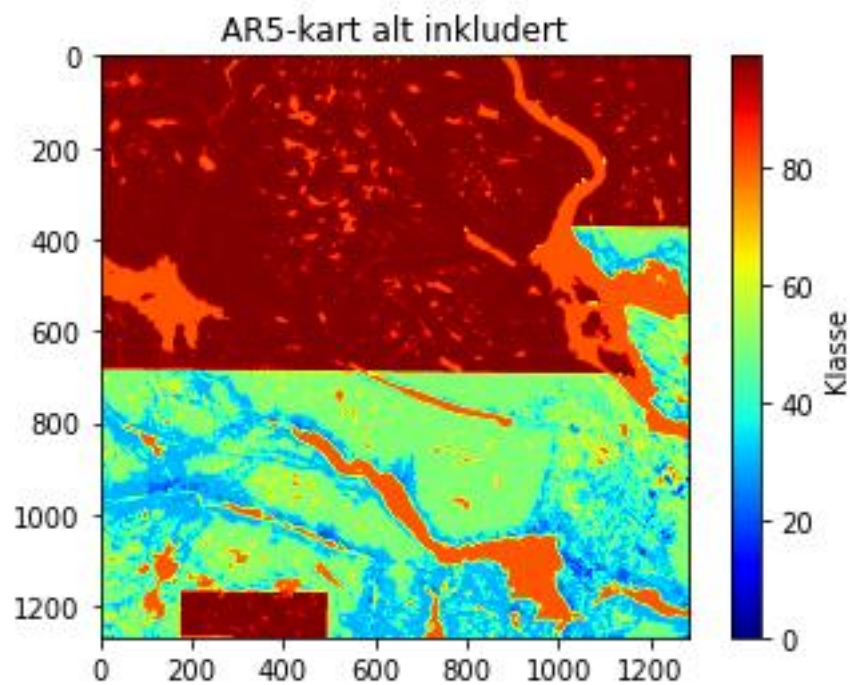
aksen representerer antall piksler i bildet. Figur 7 illustrerer AR5-kartet hvor alle klassene er inkludert, Figur 6 illustrerer AR5-kartet hvor bare myrklassen er inkludert og Figur 5 illustrerer hvor bare den «ikke-klassifiserte» klassen er inkludert.



FIGUR 5 AR5-KART HVOR KUN KLASSE 99- IKKE KLASIFISERT ER INKLUDERT. ALLE PIKSLER MED EN VERDI OVER 0 ANES SOM KLASSE 99



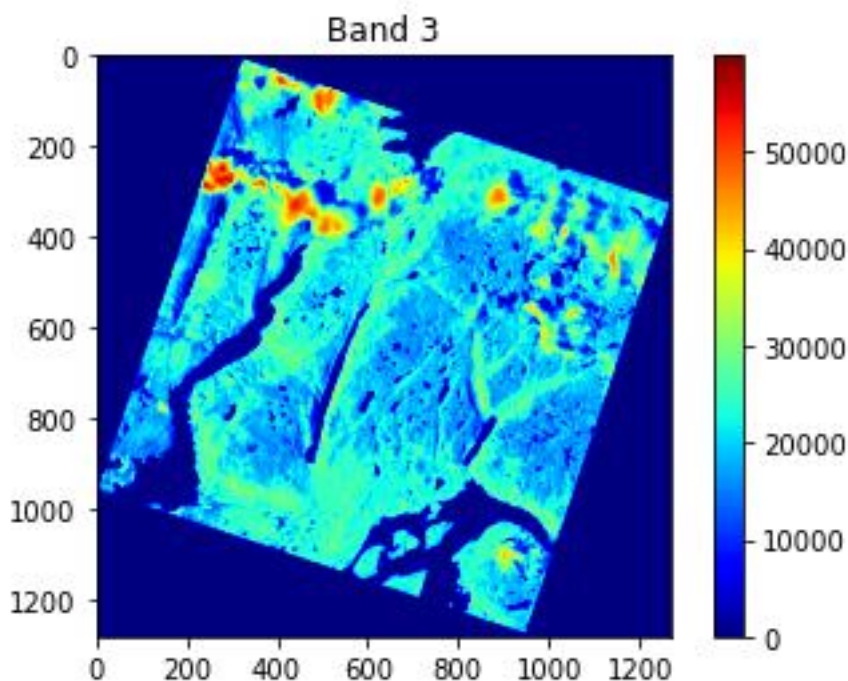
FIGUR 6. AR5-KART HVOR KLASSE 60 - MYR ER INKLUDERT. ALLE PIKSLER MED PIKSELVERDI OVER 0 ANSEES SOM MYR-PIKSLER. PIKSLER SOM SYNES, OG IKKE ER MØRKEBLÅ SOM STORE DELER AV BILDET, HAR PIKSELVERDI OVER 0



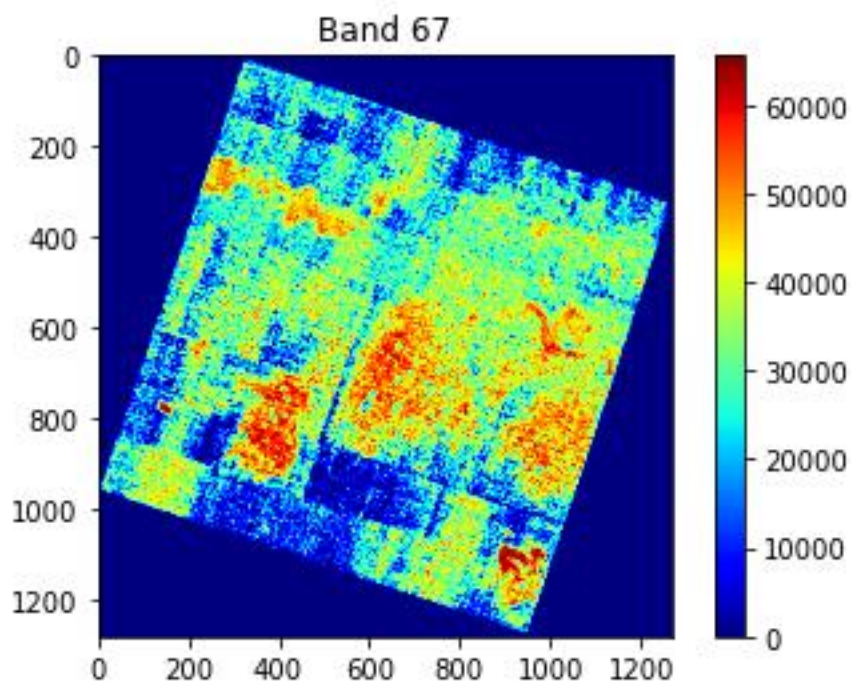
FIGUR 7. AR5-KARTET DER ALLE KLASSE ER INKLUDERT, PIKSELFARGEN REPRESENTERER KLASSENE I AR5-KARTET.

### 3.1.2 HDF5 Hyperspektrale data

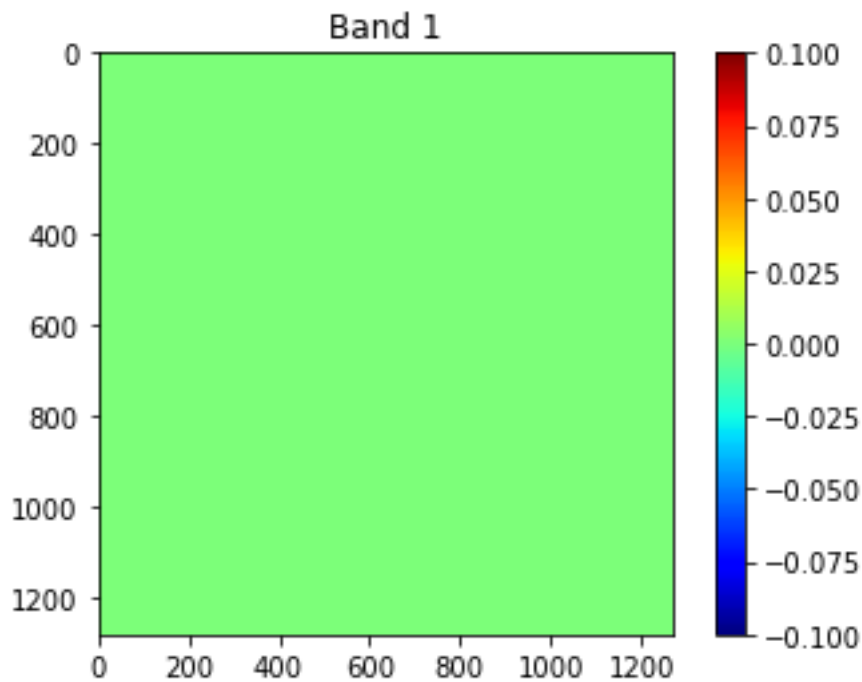
Det hyperspektrale datasettet var hentet fra NIBIO og fanget av en PRISMA satellitt. Datasettet ble lastet ned i HDF5 format der metadataen var printet i python ved bruk av *h5py* pakken. Det var ønskelig å utnytte alle de forskjellige bølgelengdene som befant seg i dataen. For å hente alle 239 spektralbåndene måtte SWIR dataen på 173 bånd og VNIR dataen på 66 bånd sammenslås til ett datasett. I datasettet kom bånd 1, bånd 2 og bånd 238 ut med manglende data (figur 10). Short-wave infrared (SWIR) eller kortbølget infrarødt er en del av det infrarøde spektrumet og strekker seg fra omtrent 1.4 mikrometer til 3 mikrometer bølgelengde. Visible and Near Infrared (VNIR) eller synlig og nær-infrarødt lys er også en del av det elektromagnetiske spekteret som inneholder synlig lys og ligger nært det infrarøde område. VNIR strekker seg fra 0.4 mikrometer til 1 mikrometer. Figur 8 visualiserer det første spektralbåndet fra VNIR-kuben, bånd 1. Figur 9 visualiserer det første spektralbåndet fra SWIR-kuben, bånd 65. På lik måte med figur 5, figur 6 og figur 7, så representerer akse X og Y antall piksler i bildet. Farge-baren på høyresiden av båndet representerer pikselverdiene, eller lysintensiteten registrert når kameraet tok bilde + korreksjonene. Det er 16-bits, som betyr at skalaen går fra 0 til 65535.



FIGUR 8. FIGUREN VISER EN ILLUSTRASJON AV DET FØRSTE BANDET FRA VNIR-KUBEN SOM INNEHOLDT BRUKBARE DATA.



FIGUR 9. FIGUREN VISER EN ILLUSTRASJON AV DET FØRSTE BANDET FRA SWIR-KUBEN SOM INNEHOLDT BRUKBARE DATA.



FIGUR 10. FIGUREN VISER EN ILLUSTRASJON AV BÅND 1 SOM MANGLER DATA, DER ALLE PIKSELVERDIENE ER 0. BAND 2 OG 238, SOM ER DET SISTE BÅNDET, FØLGER SAMME TREND.

Noen av områdene i båndet manglet verdier. Disse områdene befant seg i hjørnene av båndet, der det kunne virke som at det illustrerte området var vridd med klokken. De manglende verdiene var ugyldige og måtte derfor korrigeres. Ved å påføre en maske på datasettet, kunne en fjerne disse ugyldige verdiene.

### **3.1.3 Skydekkemaske**

For å redusere feilkilder som kunne påvirke modellen, ble skydekket fjernet fra datasettet ettersom skyer kunne ha en kraftig negativ påvirkning på modellens predikasjonsevne. Skyer kan blant annet blokkere og reflektere sollys, som igjen kan ha en påvirkning på hvor mye lys som når bakken. Dette kan dermed resultere i undereksponering som kunne gjøre det vanskeligere for modellen å oppdage egenskaper i de undereksponerte områdene (Shu & Freeman, 1990).

I tillegg til skyenes egenskap til å blokkere og reflektere lys, kan de også kaste skygger på overflaten som også fører til delvis eller fullstendige skyggelagte områder. Skyer kan også forårsake atmosfærisk «scattering» eller spredning som vil redusere mengden og kvaliteten på den innhentede dataen (Nishita et al, 1993).

En skydekkemaske ble derfor påført datasettet for å fjerne skyenes påvirkning på datasettet. Sky-pikslene ble maskert, og datasettet ville da bare inneholde de pikslene med korrekte reflektanseverdier. Skydekkemasken er en TIFF fil der pikslene er gitt verdien 0 eller 1. Pikslene med verdi 0 var angitt som skyer og ble maskert. Pikslene med verdi 1 var gyldige og ble inkludert i modellen.

### **3.1.4 AR5-kartmaske**

Ettersom det eksisterte piksler uten verdier i det hyperspektrale datasett (figur 8, figur 9), måtte disse også korrigeres for. Pikslene uten verdier må ikke forveksles med arealtypen 99 – ikke klassifisert. De ugyldige verdiene kunne hatt en negativ påvirkning på klassifiseringsmodellen der de kunne dekke over eller skjule viktig informasjon som

kunne eksistere i datasettet. Disse pikslene kan skyldes feil i datainnsamlingen, feil i sensoren eller andre hindringer som kan ha blokkert datainnsamlingen.

For å sikre at modellene brukte riktig data ble det påført en kartmaske i form av en TIFF-fil. Pikslene i masken var angitt verdien 0 eller 1. Pikslene med verdien 1 i TIFF bildet ble angitt som ugyldige verdier og pikslene med verdi 0 ble angitt som gyldige verdier som måtte inkluderes i modellen.

TABELL 5. DE FORSKJELLIGE MASKETYPENE PÅFØRT DATASETET, TILHØRENDE PIKSELVERDIER OG PIKSELVERDIENS GYLDIGHET FOR MASKETYPEN

<i>Masketype</i>	<i>Pikselverdi</i>	<i>Gyldighet</i>
Skydekkemaske	0	Ugyldig
Skydekkemaske	1	Gyldig
Maske	0	Gyldig
Maske	1	Ugyldig

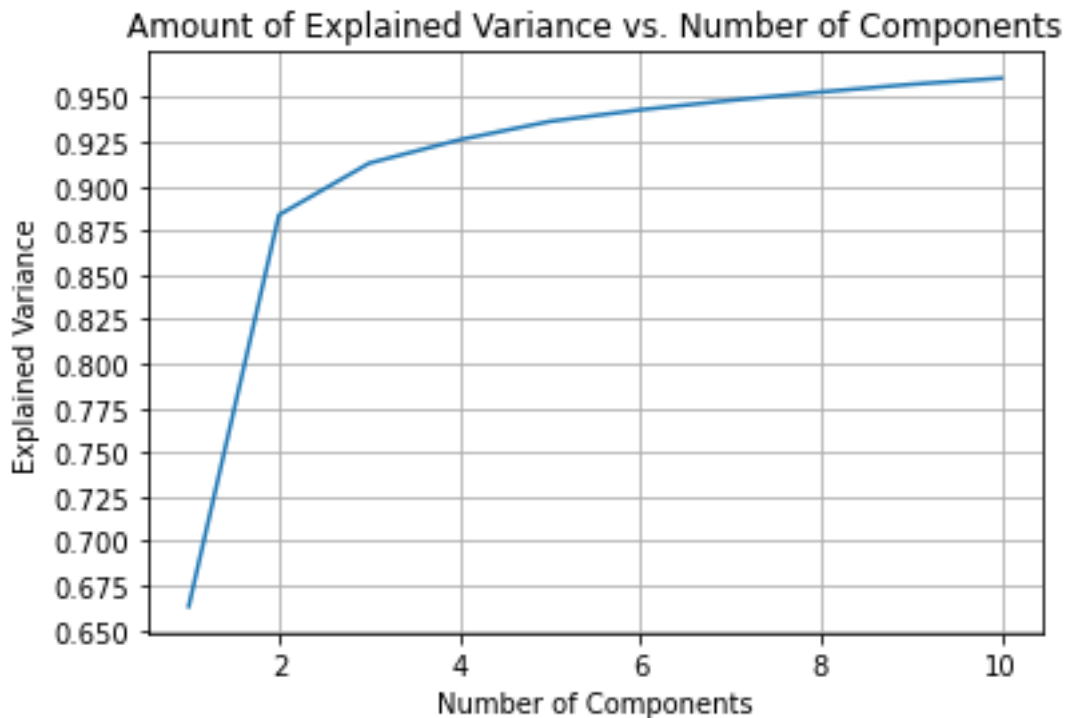
## 3.2 Datasett avledet fra hyperspektrale data

Fra det hyperspektrale datasettet ble det avledet et datasett som fikk sine hovedkomponenter analysert med en PCA (Principal Component Analysis). Datasettet ble pre-prosessert før analysen gjennomføres.

### 3.2.1 Prinsipalkomponent analysert hyperspektrale data

Etter en analyse av prinsipalkomponentene kom det frem at nærmest all variansen ble forklart i de tre første hovedkomponentene. Mengden varians en komponent inneholder sier hvor mye informasjon om datasettet som kommer frem i komponenten. Basert på figur 11, inneholdt de tre første komponentene omtrent 92% av variansen som eksisterer i datasettet. Mer spesifikt så ble omtrent 88% av variansen forklart i de to første komponentene, der den tredje komponenten inneholdt omtrent 4%.

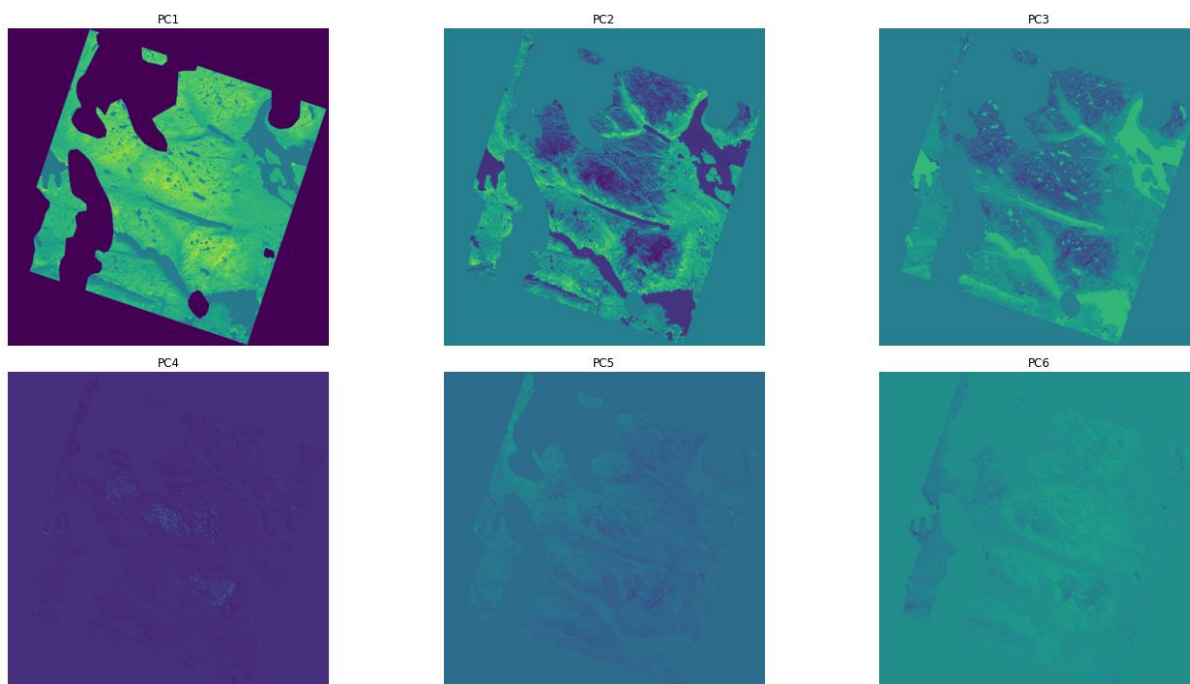
Komponentene etter den tredje komponenten inneholdt mindre varians. Figur 11 visualiserer at 95% av variansen til datasettet ble forklart i 8 komponenter, dette tilsvarer at alle komponentene etter komponent nummer 3 inneholdt 2% av den forklarte variansen til datasettet.



FIGUR 11 X-AKSEN I FIGUREN REPRESENTERER DEN FORKLARTE VARIANSEN OG Y AKSEN I FIGUREN REPRESENTERER ANTALL KOMPONENTER.

En lignende trend kan observeres i figur 12, der det kan observeres hvor mange detaljer som blir «vasket ut» jo høyere komponenttallet var. PC1 og PC2 var illustrert med høyere detalj sammenlignet med komponentene som kom etter. Komponent 3 inneholdt vesentlig mindre varians enn komponent 1 og 2, men en kunne fortsatt tyde flere detaljer fra komponent 3 enn komponent 4, 5 eller 6. Ettersom antall komponenter ble brukerdefinert i Python, var alle 10 komponentene inkludert i komponentanalysen. Selv om det kunne virke unødvendig å inkludere alle 10 komponenter, så økte ikke prosesseringstiden seg betydelig. Det prinsipalkomponentanalyserte datasettet ble utnyttet som et testsett som igjen ble brukt til å evaluere ytelsen til modellen som ble trent på AR5 treningssettet.





FIGUR 12. FIGUREN ILLUSTRERER DE FØRSTE 6 KOMPONENTENE TILPASSET OPPGAVENS DATASETT.

### 3.2.2 Maskert og skydekkemaskert data

Maskene, som ble omtalt i seksjon i 3.1.3 og 3.1.4 ble pålagt de hyperspektrale datasettet før hovedkomponentene ble analysert. Maskene ble også lagt på AR5-kartet som isolerer myr klassen og AR5-kartet som isolerer klassen «ikke-klassifisert.»

### 3.3 Produkter avledet fra arealressurskart

Fra arealressurskartet ble det avledet to kartlag: et maskert og et skydekkemaskert AR5- kart. Det maskerte og skydekkemaskerte AR5- kartet som isolerer myr ble brukt som treningsdata for klassifiseringsmodellen. AR5-kartet som isolerer den «ikke-klassifiserte» arealtypen ble brukt som kartfasong til både testdataene til klassifiseringsmodellen og treningsdataene til den samme modellen.

### 3.4.1 Undersampling

Det eksisterte mange flere piksler i datasettene som representerer den «ikke-klassifiserte» arealtypen enn det gjorde for arealtypen myr. En kraftig ubalanse som dette kunne medføre at modellen presterte dårligere, brukte mer tid på å fullføre klassifiseringen og at modellene kunne gi dårligere klassifiseringsresultater. Ettersom klassene i datasettet var i kraftig ubalanse, måtte en spesifikk preprosesseringsprosess påføres både treningsdataene og testdataene. Ved å påføre en undersampling på både testdataene og treningsdataene med randomundersampler kunne en lik mengde piksler for hver klasse oppnås, dermed balansere datasettet. Dette medførte at modellene ga et mer presist og nøyaktig resultat.

Før undersamplingen så bestod klassen som representerte den «ikke-klassifiserte» arealtypen av 127037 klasse-eksemplarer (støtter), og arealtypen myr bestod av 6890 støtter. En gjennomføring av SVM, uten å påføre en undersampling på datasettet, gjorde at modellen brukte opp mot en halvtime på å fullføre klassifiseringen, samtidig som klassifiseringsresultatet var svært dårlig. Etter en gjennomført undersampling tok det SVM rundt 2 minutter på å gjennomføre klassifiseringen, der resultatene også ble bedre. En observasjon av klassifiseringsrapporten til KNN før og etter en påført undersampling kan observeres i tabell 10. og tabell 11.

Ved å påføre en randomundersampler på datasettet reduseres mengden «ikke-klassifiserte» arealtyper ved tilfeldig valg. Mengden ble redusert helt til antall klasseeksemplere for den «ikke-klassifiserte» arealtypen var lik klasseeksemplene for arealtypen myr. Både myrklassen og den «ikke-klassifiserte» klassen bestod av 6890 støtter etter en påført undersampling.

### 3.4.2 Train-test split

Det balanserte datasettet ble så splittet eller delt inn i to sett, et treningssett og et testsett. Test\_size variabelen ble satt til 0.3, slik at 30% av datasettet ble brukt som et testsett. En test\_size på 30% gjorde det mulig å ha nok data til testing, samtidig som den resterende datamengden på 70% ville være nok data til å trene modellene til å gjøre nøyaktige prediksjoner. Random-state parameteren ble satt til en bestemt verdi, i denne oppgaven var det 28. Ved å sette en bestemt verdi for random-state parameteren kunne

datasettfordelingen gjenskapes for alle modellene, slik at det endelige resultatet var rettferdig.

### **3.4.3 Klassifisering – valg av klassifiseringsmodell**

For å velge korrekt klassifiseringsmodell for datasettene må datasettets begrensninger vurderes. Datasettet var stort og inneholdt mange mange piksler. Oppgaven var komplisert, ettersom den ønsket å klassifisere myr, som kan være utfordrende å klassifisere. Oppgaven ble utført på en rekke klassifiseringsmodeller, og modellen som ga best resultat på de fleste evalueringsmålene viste seg å være en SVM (Tabell 10-14). De andre klassifiseringsmodellene som ble testet var LogReg, RFC og KNN. SVM og RFC egnet seg godt på det store og komplekse datasettet, som kunne komme fra SVM og RFC sine egenskaper til å bearbeide høydimensjonale data.

### **3.4.4 Valg av hyperparametere**

For å optimere ytelsen til klassifiseringsmodellene var valget av riktige hyperparametere for problemstillingen et viktig valg. For denne problemstillingen var det utnyttet en RandomSearchCV for å finne de mest passende hyperparameterne til alle modellene utenom LogReg, ettersom LogReg er en så enkel modell. Et forsøk på å finne hyperparameterne med å bruke GridSearchCV ble utført, men størrelsen på datasettet økte regnekraften til modellen. Den økte regnekraften førte til at tiden det tok for en GridSearchCV på SVM å fullføre kunne strekke seg over flere dager.

Ettersom LogReg og KNN er klassifiseringsmodeller som krever betraktelig mindre regnekraft enn de andre nevnte modellene, ville det være mulig å gjennomføre en GridSearchCV for LogReg og KNN etter en undersampling. KNN tok allikevel betraktelig mer tid enn LogReg. LogReg brukte 2 minutter på å fullføre en GridSearchCV, og KNN brukte 1 time på å fullføre med samme søkeområde. 1 time var en akseptabel prosesseringstid, som la grunnlaget for bruken av GridSearchCV til KNN og LogReg. Ved å bruke GridSearchCV på LogReg og KNN etter en undersampling, istedenfor RandomSearchCV kunne en teste flere hyperparameterkombinasjoner for finne de gunstigste hyperparameterne.

TABELL 6. HYPERPARAMETERE FOR SVM, FUNNET AV RANDOMSEARCHCV

Hyperparameter	Verdi
C	10.097
class_weight	balanced
kernel	rbf

TABELL 7. HYPERPARAMETERE FOR RANDOMFORESTCLASIFIER, FUNNET AV RANDOMSEARCHCV

Hyperparameter	Verdi
n_estimators	143
max_depth	48
max_features	sqrt

TABELL 8. HYPERPARAMETERE FOR LOGISTICREGRESSION, FUNNET AV GRIDSEARCHCV

Hyperparameter	Verdi
C	0.01
penalty	L1

TABELL 9. HYPERPARAMETERE FOR K-NEAREST-NEIGHBOR, FUNNET AV GRIDSEARCHCV

Hyperparameter	Verdi
n_neighbors	10
weights	uniform
p	2

# 4. Resultater

## 4.1 Klassifiseringsrapporten til modellene med fokus på myrklassen

TABELL 10. KLASSIFISERINGSMODELLENE EVALUERINGSMÅL FRA MODELLENS KLASSIFISERINGSRAPPORT FOR MYRKLASSEN. SVM VISES MED FARGEN BLÅ, RANDOMFOREST MED FARGEN GRØNN, LOGISTISK REGRESJON MED FARGEN GRÅ, K-NEAREST NEIGHBORS MED EN LYS BRUN FARGE OG K-NEAREST NEIGHBORS UTEN UNDERSAMPLING VISES MED FARGEN RØD

Klassifiseringsmodell	Precision	Recall	F1-Score
SVM	0.77	0.90	0.83
RandomForest	0.76	0.88	0.82
LogReg	0.70	0.86	0.77
KNN	0.75	0.86	0.80
KNN (Uten randomundersampler)	0.48	0.05	0.10

KNN (uten randomundersampler har en presisjon på 0.48, recall på 0.05, og F1-score på 0.10 (tabell 10). Dette vil si at modellen har middels god presisjon, men har høy sannsynlighet for å klassifisere falske negative tilfeller og lav sannsynlighet for å klassifisere sanne positive tilfeller. Ettersom KNN (uten randomundersampler) presterer dårlig på å klassifisere myr, beregnes det ikke flere evalueringsmål for noen modeller trent på data som ikke er undersamlet. SVM, RandomForest, LogReg, og KNN presterer mye bedre, og blir dermed brukt i videre analyser.

## 4.2 Klassifiseringsrapporten til modellene med fokus på den «ikke-klassifiserte» klassen

TABELL 11. KLASSIFISERINGSMODELLENES EVALUERINGSMÅL FRA MODELLENS KLASSIFISERINGSRAPPORT FOR DEN «IKKE-KLASSIFISERTE» AREALKLASSEN. SVM, RANDOMFOREST, LOGISTISK REGRESJON OG K-NEAREST NEIGHBORS REPRESENTERES AV DE SAMME FARGENE SOM I TABELL 10.

Klassifiseringsmodell	Precision	Recall	F1-Score
SVM	<b>0.88</b>	<b>0.73</b>	<b>0.79</b>
RandomForest	<b>0.86</b>	<b>0.73</b>	<b>0.79</b>
LogReg	<b>0.81</b>	<b>0.64</b>	<b>0.72</b>
KNN	<b>0.84</b>	<b>0.71</b>	<b>0.77</b>
KNN (Uten randomundersamplere)	<b>0.95</b>	<b>1.0</b>	<b>0.97</b>

KNN (uten randomundersamplere) presterer nesten perfekt for klassifiseringen av den «ikke-klassifiserte klassen», med en presisjon på 0.95, recall på 1.0, og F1-score på 0.97 (tabell 11). LogReg presterer dårligst med en presisjon på 0.81, recall på 0.64, og F1-score på 0.72.

## 4.3 Nøyaktighet og klassifiseringsfeil

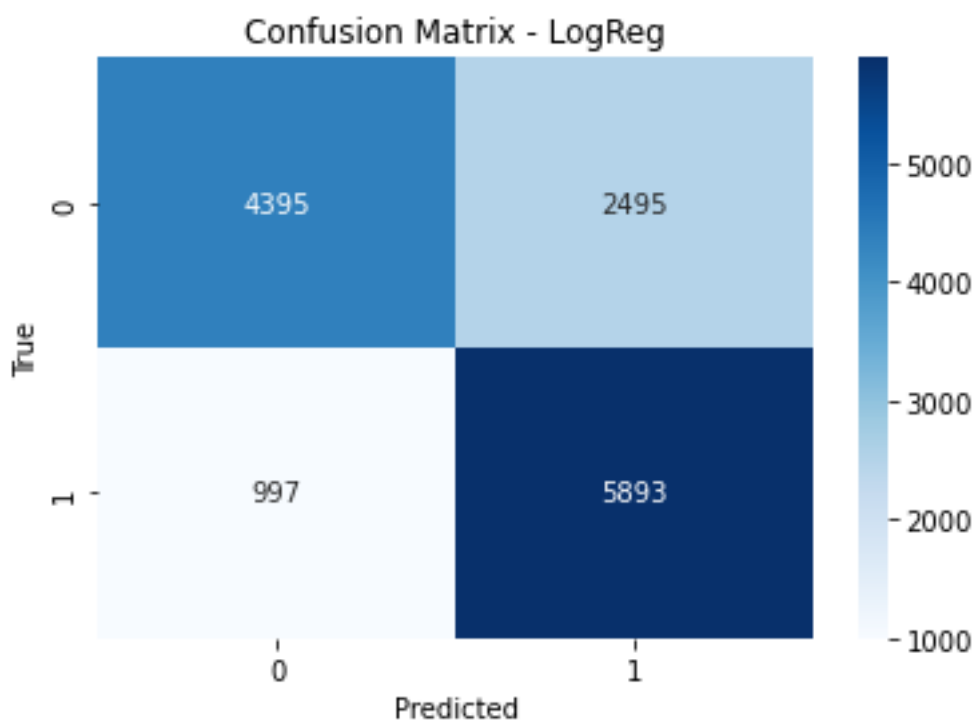
TABELL 12. VISER KLASSIFISERINGSMODELLENES NØYAKTIGHET OG KLASSIFISERINGSFEIL. SVM, RANDOMFOREST, LOGISTISK REGRESJON OG K-NEAREST NEIGHBORS REPRESENTERES AV DE SAMME FARGENE SOM I TABELL 10.

Klassifiseringsmodell	Nøyaktighet	klassifiseringsfeil
SVM	<b>0.811</b>	<b>0.189</b>
RandomForest	<b>0.806</b>	<b>0.194</b>
LogReg	<b>0.747</b>	<b>0.253</b>
KNN	<b>0.788</b>	<b>0.212</b>

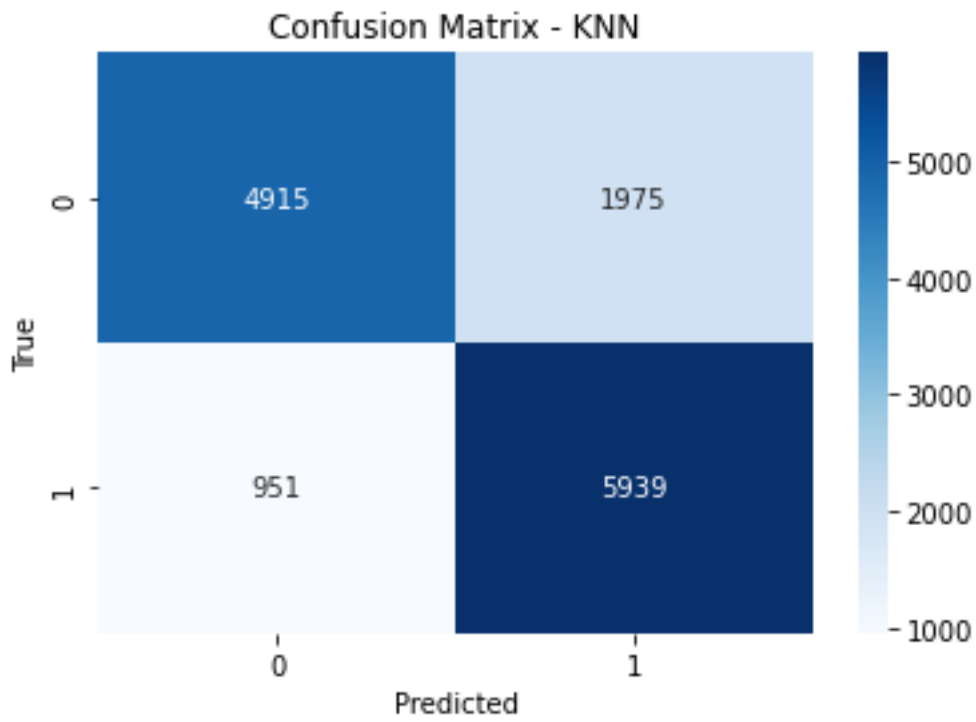
Av modellene brukt i undersøkelsene presterer SVM best, med en nøyaktighet på 0.811 og en klassifiseringsfeil på 0.189 (tabell 12). KNN presterer dårligst med en nøyaktighet på 0.788 og klassifiseringsfeil på 0.212.

## 4.4 Forvirringsmatrisene

Forvirringsmatrisene for de ulike modellene viser at SVM presterer best med tanke på antall sanne positive og sanne negative hendelser (figur 15). Matrisene viser også at RFC presterer best med antall sanne positive hendelser (Figur 16). Videre, presterer LogReg dårligst i forhold til de andre modellene (Figur 13).

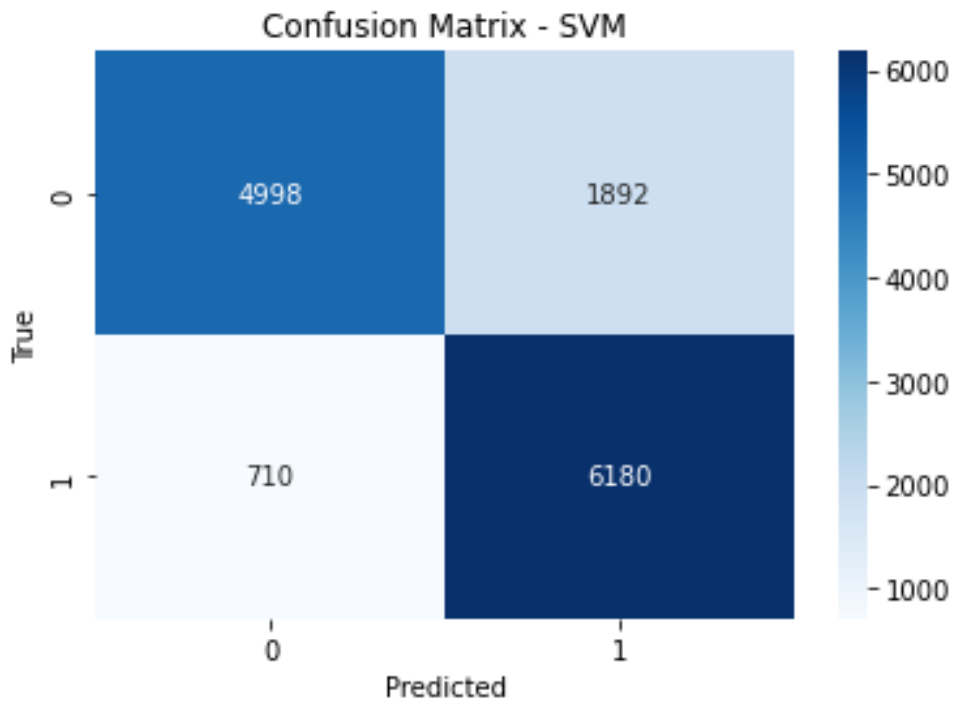


FIGUR 13. FORVIRRINGSMATRISER FOR LOGREG: CELLE 0-0 = SANN POSITIV (SP), CELLE 0-1 = FALSK NEGATIV (FN), CELLE 1-0 = FALSK POSITIV (FP), CELLE 1-1 SANN NEGATIV (SN) (SE FIGUR 4). TALLENE I FIGUREN VISER TIL ANTALL HENDELSER FOR HVER AV RESULTATENE

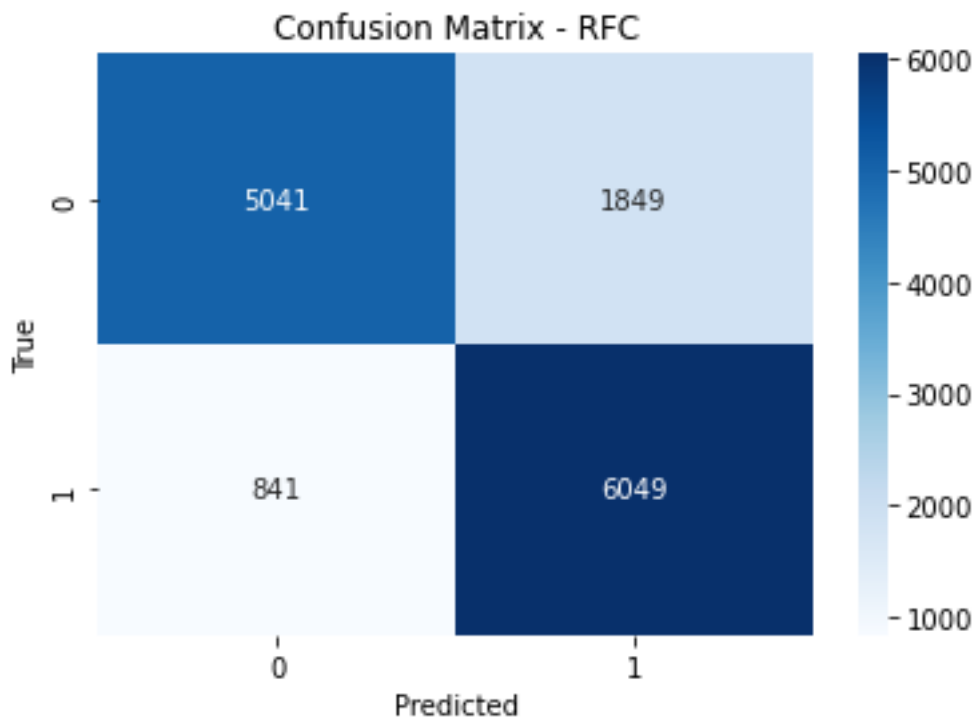


FIGUR 14. FORVIRINGSMATRISE FOR KNN: CELLE 0-0 = SANN POSITIV (SP), CELLE 0-1 = FALSK NEGATIV (FN), CELLE 1-0 = FALSK POSITIV (FP), CELLE 1-1 SANN NEGATIV (SN) (SE FIGUR 4). TALLENE I FIGUREN VISER TIL ANTALL HENDELSER FOR HVER AV RESULTATENE





FIGUR 15. FORVIRINGSMATRISE FOR SVM: CELLE 0-0 = SANN POSITIV (SP), CELLE 0-1 = FALSK NEGATIV (FN), CELLE 1-0 = FALSK POSITIV (FP), CELLE 1-1 SANN NEGATIV (SN) (SE FIGUR 4). TALLENE I FIGUREN VISER TIL ANTALL HENDELSER FOR HVER AV RESULTATENE



FIGUR 16 FORVIRRINGSMATRISE FOR RFC: CELLE 0-0 = SANN POSITIV (SP), CELLE 0-1 = FALSK NEGATIV (FN), CELLE 1-0 = FALSK POSITIV (FP), CELLE 1-1 SANN NEGATIV (SN) (SE FIGUR 4). TALLENE I FIGUREN VISER TIL ANTALL HENDELSER FOR HVER AV RESULTATENE

## 4.5 Kappa koeffisient og MCC-score

SVM oppnår den høyeste verdien for kappa score og MCC-score (Tabell 13.) som den har gjort på de fleste evalueringsmålene. SVM oppnådde en MCC score på 0.632, etterfulgt av RFC med 0.611, KNN med 0.582 og til slutt LogReg på 0.506.

TABELL 13. KLASSIFISERINGSMODELLENES MCC-SCORE OG KAPPA-SCORE. SVM, RANDOMFOREST, LOGISTISK REGRESJON OG K-NEAREST NEIGHBORS REPRESENTERES AV DE SAMME FARGENE SOM I TABELL 10.

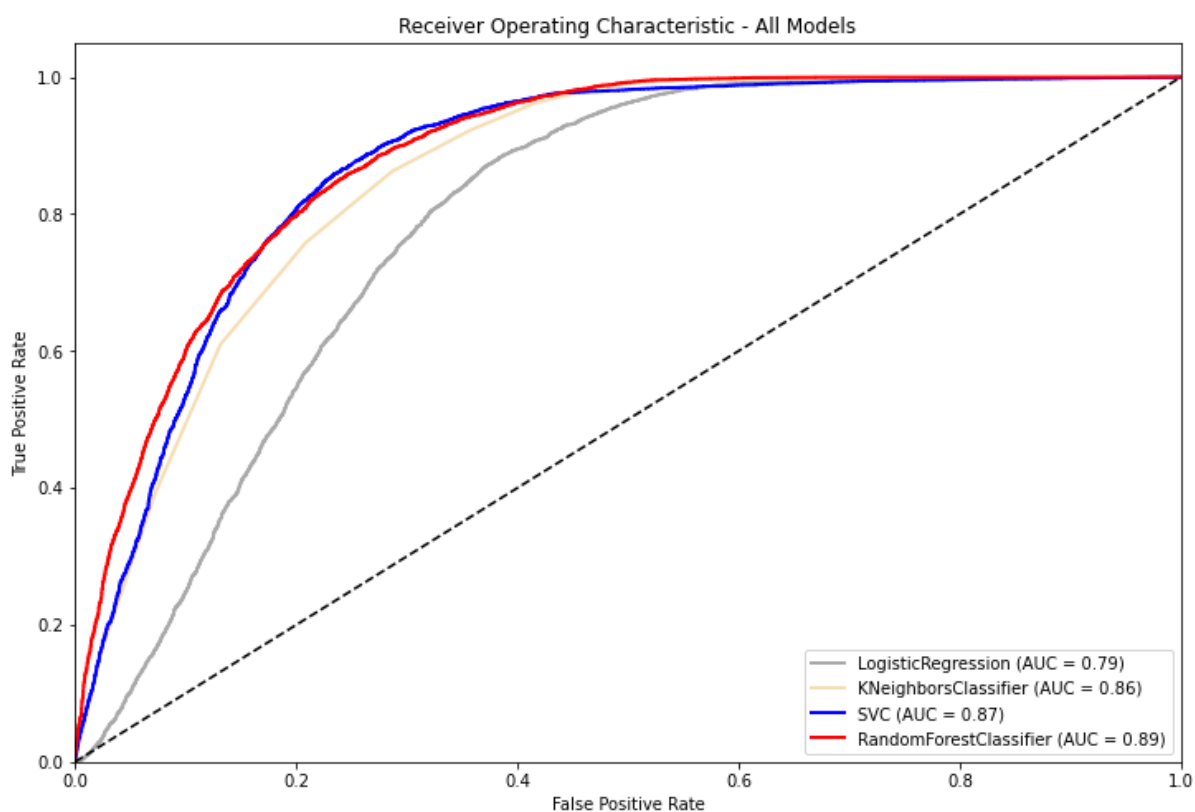
Klassifiseringsmodell	MCC-score	Kappa-score
SVM	<b>0.632</b>	<b>0.622</b>
RandomForest	<b>0.611</b>	<b>0.605</b>
LogReg	<b>0.506</b>	<b>0.494</b>
KNN	<b>0.582</b>	<b>0.575</b>

## 4.6 Area Under the Curve (AUC)

Alle modellene med unntak av LogReg har en AUC-score mellom 0.8-0.9, og evalueres derfor som svært god (tabell 14). RFC har den høyeste AUC-scoren på 0.89. LogReg har den laveste AUC-scoren på 0.79 og evalueres dermed som god. LogReg har dermed færre tilfeller av sanne positive tilfeller enn de andre modellene (figur 17).

TABELL 14. KLASSIFISERINGSMODELLENES AUC-SCORE. SVM, RANDOMFOREST, LOGISTISK REGRESJON OG K-NEAREST NEIGHBORS REPRERENTERES AV DE SAMME FARGENE SOM I TABELL 10.

Klassifiseringsmodell	AUC-score
SVM	<b>0.87</b>
RandomForest	<b>0.89</b>
LogReg	<b>0.79</b>
KNN	<b>0.86</b>



FIGUR 17. MODELLENES ROC-KURVE. X-AKSEN REPRERENTERER DEN FALSKE POSITIVE RATEN. Y-AKSEN REPRERENTERER DEN SANNE POSITIVE RATEN.

# 5. Diskusjon

Av alle klassifiseringsmetodene ga SVM best resultater på de fleste områdene utenom AUC verdien, der RFC presterte best. Årsaken til dette kan trolig vises i forvirringsmatrisene til SVM (figur 15) og RFC (figur 16). Disse matrisene viser at RFC oppnår flere sanne positive hendelser enn SVM. Ettersom AUC verdien baserer seg på sammenhengen mellom modellenes sanne positive grad og modellenes falske positive grad, så vil et høyere antall med sanne positive hendelser lede til en høyere AUC verdi. Årsaken til at SVM presterte bedre enn RFC i andre evalueringsmål, selv om RFC har flere sanne positive hendelser, skyldes trolig av at SVM har høyere antall sanne negative hendelser enn RFC har høyere sanne positive hendelser over SVM. Til sammen så har SVM 11 178 sanne hendelser, sammenlignet med RFC har 11 090 sanne hendelser.

SVM sin største svakhet i forhold til denne oppgaven, var prosesseringstiden det tok for SVM å fullføre. Uten en nødvendig undersampling av datasettet kunne det ta opptil en halvtime for SVM å fullføre. Selv med en undersampling, tar det SVM rundt 3 minutter å fullføre, i motsetning til RFC som bruker under 1 minutt. Svakheten kan muligens håndteres av en sterkere datamaskin, mer spesifikt en kraftigere prosesser og grafikkort. Alle modellene ble prosessert av en 12. generasjon Intel(R) Core™ i5-12600K prosessor og et NVIDIA GeForce RTX 3080 grafikkort.

## 5.1 Valg av hyperparametere

Valget av hyperparametere til modellene baserte seg på om det var mulig å utføre et rutenettsøk med en GridSearch eller et tilfeldig søk med RandomSearch i python. Valget av hyperparametere iverksatte bruken av undersampling, som fikk en positiv retroaktiv effekt på valget av klassifiseringsmetode. Det var gjort et forsøk på å utføre et rutenettsøk på hyperparametere til SVM og RFC, men begge søkene hadde en urealistisk lang prosesseringstid. Den lange prosesseringstiden til SVM og RFC i rutenettsøket, var det som inspirerte forsøket på å påføre en undersampling på datasettet. For LogReg og KNN, som har en betydelig kort prosesseringstid i forhold til SVM og RFC, var parametersøket for de to modellene utført med et rutenettsøk, som

tillater en bedre parameteroptimalisering. For SVM og RFC, var det utført et tilfeldig søk, siden de har en lengre prosesseringstid. Den raske prosesseringstiden til LogReg og KNN sine parametersøk kan gjøre de til et mulig bedre valg for klassifiseringer av mindre sjeldne arealtyper.

## 5.2 AR5 som treningsdata

Kvaliteten på treningsdatasettet kan være med på å påvirke modellenes estimerte nøyaktigheter. Noen steder i AR5 inneholder lavere presisjon, og i andre tilfeller kan det rett og slett være feil. AR5 setter først og fremst fokus på landbruk og skog, som kan føre til at kartet er mindre nøyaktig på sjeldnere arealtyper som myr. For AR5-grensene, som er synlige i terrenget, er det generelle kravet til stedfestningsnøyaktigheten 2 meter. Dette fører til at grensene til polygonene som utgjør AR5 kartet har lav nøyaktighet.

Det er ikke hovedsakelig bare i polygongrensene det kan oppstå feil i AR5 kartet. Det kan også oppstå feilklassifiserte arealtyper. Med feilklassifiserte arealtyper menes arealer som har en spesifikk verdi, men ikke oppfyller kravet for å bli klassifisert som den arealtypen. Dette kan være et problem for arealtypen myr, der myr i AR5 anses som torvlag på 30 centimeter eller mer. Dersom myra befinner seg på fjell, der dybden på torvlaget kan være under 30 cm, klassifiseres den fremdeles som myr, om myra er over 2 dekar ifølge NIBIO (2016). Ettersom det i utgangspunktet er utfordrende å kartlegge myr manuelt, kan det antas at det er en utfordring å opprettholde det spesielle kravet for myr på 2 dekar. Derfor er det viktig å vurdere om all faktisk myr i området er inkludert i AR5 kartet.

## 5.3 Diskusjon av resultater fordelt på evalueringsmål

### 5.3.1 Klassifiseringsrapporten for myrklassen

I oppgaven nådde SVM-modellen det høyeste presisjonsresultatet med 0.77, etterfulgt av RandomForest med 0.76, KNN med 0.75 og til slutt LogReg med 0.70. Disse resultatene viser at SVM-modellen oppnår mest sanne positive tilfeller blant alle modellene som ble evaluert.

Når det gjelder sensitiviteten (recall), som måler modellens evne til å oppdage virkelig positive tilfeller, oppnådde SVM-modellen igjen det høyeste resultatet med en verdi på 0.90. Dette viser til at SVM-modellen er bedre til å identifisere de faktiske positive tilfellene sammenlignet med de andre modellene. RandomForest, KNN og LogReg hadde også relativt høye gjenvinningstall, med henholdsvis 0.88, 0.86 og 0.86.

Gjennomsnittet av presisjonen og sensitiviteten observeres i form av F1-score. SVM-modellen oppnådde den høyeste F1-scoren med 0.83, etterfulgt av RandomForest med 0.82, KNN med 0.80 og LogReg med 0.77. Disse resultatene viser til at SVM-modellen demonstrerer en bedre balanse mellom presisjon og sensitivitet enn de andre modellene med en smal margin.

På den andre siden viser KNN-modellen uten randomundersamplinger betraktelig lavere resultater på alle de tre viktige evalueringsmålene. Med en presisjon på 0.48, sensitivitet på 0.05 og F1-score på 0.10, viser denne modellen at den ikke er i stand til å korrekt klassifisere positive tilfeller eller oppnå en akseptabel balanse mellom presisjon og sensitivitet. Dette kan vise at randomundersamplingsteknikken i form av RUS i Python spiller en viktig rolle på å forbedre modellens ytelse ved å håndtere ubalanserte datamengder. Etter denne undersamplingsteknikken nådde KNN en presisjon på 0.75, en sensitivitet på 0.86 og en F1-score på 0.80. Dette er en betydelig forbedring av klassifiseringsresultatene til KNN.

### 5.3.2 Klassifiseringsrapporten for den «ikke-klassifiserte» arealklassen

Klassifiseringsrapporten for den «ikke-klassifiserte» arealklassen følger den samme trenden som for myr, der SVM oppnådde de høyeste resultatene, men her har også presisjonen til modellen økt, samtidig som sensitiviteten til modellene har avtatt. Det tyder på at modellene oppnår flere antall riktige positive prediksjoner, men modellene er samtidig mindre effektive til å gjenkjenne alle de virkelige positive tilfellene. En årsak til dette kan skyldes at modellene opplever færre falske positive hendelser, som kan forårsake at modellene går glipp av noen faktisk positive tilfeller.

KNN uten undersampling vil oppnå en nesten perfekt klassifisering av den «ikke-klassifiserte» arealklassen. Dette skyldes av at datasettet nå ikke er påført en undersampling, som vil føre til en drastisk ubalanse i klassene. Som nevnt i kapittel 3.4.1 består datasettet uten undersampling av 133 927 støtter, der 6890 av de var myr-støtter som vil tilsvare at 5% av støttene bestod av myr-støtter.

I denne klassifiseringsrapporten kan det også observeres at RandomForest presterte nesten helt likt som SVM, med unntak på et avvik på 0.02 i presisjonen. Dette kan indikere at modellene presterer ganske likt når det gjelder deres evne til å gi sanne positive prediksjoner blant alle de positive prediksjonene.

Som nevnt i kapittel 2.8 er det viktig å se på flere evalueringsmålinger for modellene for å få gi en pålitelig begrunnelse i hvilken modell som presterte best. Basert på disse to klassifiseringsrapportene er det SVM som har prestert best hittil.

### 5.3.3 Klassifiseringsmodellenes nøyaktighet og klassifiseringsfeil

Tabell 12. Gir en oversikt over og illustrerer de forskjellige modellenes evne til å korrekt klassifisere klassene i datasettet. SVM følger den samme trenden som tidligere med å prestere best med en nøyaktighet på 0.811, der RFC også holder samme trend med å være like bak SVM med en nøyaktighet på 0.806. Videre oppnådde KNN en nøyaktighet på 0.788, ikke en like god nøyaktighet som SVM og RFC, men fortsatt et akseptabelt resultat. Til sist kom LogReg med den laveste nøyaktigheten på 0.747.

Modellen vil prestere i lik rangering for klassifiseringsfeilen ettersom feilen er det samme som  $1 - \text{klassifiseringsmodellenes nøyaktighet}$ . For SVM vil klassifiseringsfeilen

være  $1 - 0.811 = 0.189$ . Denne feilen tar for seg antall feilklassifiserte klasser, som SVM har minst av og LogReg har mest av.

### 5.3.4 Klassifiseringsmodellenes forvirringsmatriser

I motsetning til de tidligere evalueringmålene, så skjer det en endring i hva som forventes av modellenes prestasjon i forvirringsmatrisene. Som nevnt tidligere i kapittel 5, kan det observeres at RFC har oppnådd et større antall sanne positive hendelser enn SVM. RFC har oppnådd 5041 sanne positive hendelser, i motsetning til SVM som har oppnådd 4998 sanne positive hendelser. Forskjellen mellom SVM og RFC med tanke på sanne positive hendelser er på 43 sanne positive hendelser. I kapitel 5 ble det også oppdaget at SVM oppnådde flere sanne negative hendelser enn RFC. Forskjellen viser seg å være på  $6180(\text{SVM}) - 6049(\text{RFC}) = 131$  sanne negative hendelser. SVM inneholder da  $131 - 43 = 88$  flere totale sanne hendelser enn RFC, som kan indikere til hvorfor SVM presterer hakket bedre enn RFC på de fleste evalueringmålene. KNN og LogReg presterer henholdsvis med 4915 og 4395 sanne positive hendelser, og 5939 og 5893 sanne negative hendelser. Resultatene til KNN og LogReg kan indikere at de presterer dårligere enn SVM og RFC.

Fra dette kan det vurderes at RFC er bedre til å oppdage selve myrklassene, ettersom den oppnår flere sanne positive hendelser enn SVM. Samtidig kan en argumentere for at en modells prestasjon er avhengig av dens egenskaper til å oppdage sanne positive hendelser i kombinasjon med dens evne til å oppdage sanne negative hendelser.

Videre må det også vurderes om RFC modellens høyere antall sanne positive hendelser kommer fra den «ikke-klassifiserte» klassen. Ved å sammenligne klassifiseringsrapportverdiene til RFC for begge klassene, kan det observeres at RFC avviker fra SVM med  $-0.02$  i sensitiviteten for myrklassen, mens sensitivetsverdien er lik SVM for den «ikke-klassifiserte» klassen. Dette kan indikere at RFC modellens antall sanne positive hendelser kommer fra RFC sin egenskap til å klassifisere den «ikke-klassifiserte» klassen bedre enn myrklassen. Denne indikasjonen kan være viktig å ta til betraktning ved det endelige valget av klassifiseringsmodell.



### 5.3.5 Klassifiseringsmodellenes kappa-score og MCC-score

Tabell 13 viser til modellenes kappa-koeffisient og MCC-score. MCC-score vil indikere en mer informativ og sann modellevaluering for klassifiseringer enn nøyaktighet eller F1-score (Chicco & Jurman, 2020). Videre oppnår SVM også den høyeste verdien for kappa-score på 0.622, etterfulgt av RFC på 0.605, mKNN med 0.575 og LogReg med 0.494

Ved å koble verdiene for MCC som er hentet ut ifra tabell 13 med tabell 3 kan det undersøkes om modellene oppnår god korrelasjon mellom modellenes predikerte klasse og den faktiske klassen. Denne korrelasjonen vil da være et nyttig evalueringsmål å ta med til betraktning ved valg av modell. Ifølge tabell 3, vil alle modellene oppnå en moderat til sterk korrelasjon mellom klassene, der SVM oppnår best korrelasjon.

Dersom kappa-score følger samme prinsipp som MCC-score, vil resultatene ende opp like. Ved å sammenligne verdiene for kappa-score i tabell 13 med tabell 2 kan det undersøkes hvor godt observasjonene samsvarer med modellens predikasjoner basert på de ulike kappa koeffisientene. Her vil SVM og RFC oppnå en god til god sammenheng, og KNN og LogReg vil oppnå en moderat til god sammenheng. De gode resultatene til modellene er trolig en kombinasjon av nødvendige preprosesseringssteg, håndteringen av ubalansen i klassene og den gode kvaliteten på treningsdataene.

### 5.3.6 Klassifiseringsmodellenes AUC score og ROC graf.

På lik måte med forvirringsmatrisen skjer det en endring i evalueringmålene. Her oppnår RFC en høyere AUC score enn SVM. Som diskutert i kapittel 5.5.4 kan årsaken til dette komme fra RFC sitt høyere antall sanne positive hendelser. En høyere AUC-score kan indikere at modellen har en bedre evne til å skille mellom de positive og de negative hendelsene. Dette kan bety at til tross for RFC færre sanne hendelser, kan det være mer effektivt å rangere og klassifisere tilfellene riktig. Der SVM nådde en AUC-score på 0.87, nådde RFC en AUC-score på 0.89. Bak de to kom KNN med en AUC-score på 0.86 og til slutt kom LogReg med en score på 0.79.

Tabell 1, indikerer at SVM, RFC og KNN oppnådde en svært god evaluering basert på AUC-scoren, og LogReg sin AUC-score indikerer at den nådde en god evaluering.

En visualisering av AUC-scoren ser man i form av ROC grafene som er illustrert i figur 17. SVM og RFC krysser hverandre på flere steder og inneholder omtrent det samme område under kurven, men RFC inneholder litt mer. En kan også tyde en svakere predikeringsevne ved LogReg i forhold til de andre modellene.

Det er viktig å ta til betraktning at disse evalueringmålene er innstilt for denne oppgaven og kan være avhengige av forskjellige faktorer som datasettets egenskaper, hyperparametere og klassifiseringsmodell. Derfor bør disse resultatene sees på i sammenheng med konteksten for forskningen og formålet med klassifiseringen. Selv om SVM oppnådde en god klassifiseringsnøyaktighet basert på disse evalueringmålene, er det også viktig veie hvor viktig prosesseringstiden er, opp mot hvor viktig nøyaktigheten er for klassifiseringsformålet.

## 5.4 Valg av klassifiseringsmodell

Som vist i delkapittel 5.3.1 til 5.3.6 er det SVM som har oppnådd de høyeste evalueringverdier på de fleste evalueringmålene, med unntak av AUC-score, der den lå 0.02 bak RFC. RFC har allikevel prestert likt med SVM i alle evalueringmålene og kan være et sterkt alternativ til rett formål. For denne oppgaven er det viktig å se på formålet, nemlig arealklassifisering av myr med bruk av hyperspektrale data. Her er det myr som er ønskelig å klassifisere, og siden det er stor mangel på nøyaktige kartlagte myrer i Norge, er det viktig å være så presise og nøyaktige som mulig under klassifisering. Den negative siden med SVM vil være den lange prosesseringstiden, som blir betraktelig mer negativ når det utføres en hyperparameteroptimalisering. Undersampling bidro til å redusere prosesseringstiden betraktelig, men for fremtidige oppdrag eller formål der datasettet består av flere klasser og mer data som kan gjøre undersampling problematisk kan det være en ide å utnytte RFC eller eventuelt KNN. For denne oppgaven hvor nøyaktigheten er i fokus passer SVM best.

## 6. Konklusjon

Oppgaven undersøkte om det var mulig å gjennomføre en metode som kunne automatisk detektere myr basert på hyperspektrale bilder med AR5 som treningsdata. Modellen som presterte best i oppgaven viste seg å være SVM, der de andre modellenes prediksjoner også når opp til nivået som kan forventes av en brukbar og nyttig klassifiseringsmodell.

Klassifiseringsresultatene til klassifiseringsmodellen SVM endte opp med å være: MCC - score = 0.632, AUC-score = 0.87, Kappa-koeffisient = 0.622 og en klassifiseringsnøyaktighet på 0.811

SVM sin evne til å klassifisere myr er god og kan mulig være nyttig til fremtidig kartlegging av myr. Dersom en kan håndtere SVM sin lange prosesseringstid på store hyperspektrale data, ved en eventuell dimensjonsreduisering eller undersampling, kan SVM være et nyttig verktøy i fremtiden for klassifisering av sjeldne arealer.

Dersom det tas til betraktning hvor viktig det kommer til å være å bevare myra, med tanke på dens karbonlagringsegenskaper, biodiversitet og flomdempende egenskaper, er det nødvendig med mer forskning innen myrkartlegging. Et arealressurskart med en mer nøyaktig kartlegging av myr kan bidra til å hindre fremtidig ødeleggelse av myra. Denne oppgaven endte opp med brukbare og nyttige resultater, men det er fortsatt rom til forbedring. For videre forskning kan andre treningsdata enn arealressurskart utforskes.

# 7. Referanser

Ahlstrøm, A. P., Bjørkelo, K. & Fadnes, K. (2019). *AR5 klassifikasjonssystem*.

*Klassifisering av arealressurser*. NIBIO BOK: Norsk institutt for bioøkonomi.

Bae, S., Lee, J., Jeong, J., Lim, C., Choi, J. (2021). Effective data-balancing

methods for class-imbalanced genotoxicity datasets using machine learning algorithms and molecular fingerprints. *Computational Toxicology*, 20.

<https://doi.org/10.1016/j.comtox.2021.100178>

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization.

*Journal of machine learning research*, 13(2). 281-305

Belgiu, M., & Drăguț, L. (2016). Random forest in remote sensing: A review of

applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24-31. <https://doi.org/10.1016/j.isprsjprs.2016.01.011>

Chang, A. C. (2020). Machine and Deep Learning. *Intelligence-Based Medicine*, 67-140.

<https://doi.org/10.1016/B978-0-12-823337-5.00005-6>

Chicco, D., Jurman, G. (2020) The advantages of the Matthews correlation

coefficient (MCC) over F1 score and accuracy in binary classification evaluation.

*BMC Genomics* 21. <https://doi.org/10.1186/s12864-019-6413-7>

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. *Automated machine*

*learning: Methods, systems, challenges*, 3-33: Springer

Gewali, U. B., Monteiro, S. T., & Saber, E. (2018). Machine learning based hyperspectral

image analysis: a survey. <https://doi.org/10.48550/arXiv.1802.08701>

Guo, G., Wang, H., Bell, D. A., Bi, Y., Greer, K. (2003). KNN Model-Based Approach in

Classification. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, 2888, 986-996. [https://doi.org/10.1007/978-3-540-39964-3\\_62](https://doi.org/10.1007/978-3-540-39964-3_62)

Huang, J., Li, Y., Xie, M. (2015). An empirical analysis of data preprocessing

for machine learning-based software cost estimation. *Information and Software Technology*, 67, 108-127. <https://doi.org/10.1016/j.infsof.2015.07.004>

Kartverket. (2022). *SOSI-standardisert produktspesifikasjon: FKB generell del 5.0*.

Tilgjengelig fra:

[https://register.geonorge.no/data/documents/produktspesifikasjoner\\_fkb-generelldel\\_v5\\_fkb-generell-del-5\\_0\\_.pdf](https://register.geonorge.no/data/documents/produktspesifikasjoner_fkb-generelldel_v5_fkb-generell-del-5_0_.pdf)

Kyrkjeeide, M. O., Bartlett, J., Rusch, G. M., Sandvik, H. & Nordén, J. (2020). Karbonlagring i norske økosystemer. *NINA temahefte*, 76.

Lu, B., Dao, P. D., Liu, J., He, Y. & Shang, J. (2020). Recent Advances of Hyperspectral Imaging Technology and Applications in Agriculture. *Remote sens*, 12(16), 2659. <https://doi.org/10.3390/rs12162659>

Martini, I. (2020). *Estimering av blågrønn faktor i eksisterende bymiljø ved bruk av multispektrale bilder og laserdata. [Masteroppgave]*. Norges Miljø- og biovitenskapelige universitet. Tilgjengelig fra: <https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2673427>

Mchugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia Medica*, 22(3), 276-282. <https://doi.org/10.11613/BM.2012.031>

NIBIO. (2016). *Arealressurskart AR5*

[Brosjyre]. NIBIO informasjonshefte: Norsk institutt for bioøkonomi.

Osmoen, I. A. (2022). *Arealklassifisering av myr ved bruk av ortofoto og laserdata. [Masteroppgave]*. Norges Miljø- og biovitenskapelige universitet. Tilgjengelig fra: <https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/3036715>

Nishita, T., Sirai, T., Tadamura, K., & Nakamae, E. (1993). Display of the earth taking into account atmospheric scattering. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 175-182 <https://doi.org/10.1145/166117.166140>

Ostfeld, A., & Salomons, S. (2005). A hybrid genetic—instance based learning

algorithm for CE-QUAL-W2 calibration. *Journal of Hydrology*, 310(1-4), 122-142.  
<https://doi.org/10.1016/j.jhydrol.2004.12.004>

Raschka, S. & Mirjalili, V. (2017). *Python Machine Learning*. Second EDN.

Birmingham, UK: Packt Publishing.

Richards, J. A. (2013). *Remote sensing Digital image analysis*,

New York: Springer.

Sedano, F., Kempeneers, P., Strobl, P., Kucera, J., Vogt, P., Seebach, L & San-Miguel-Ayanz,

J. (2011). A cloud mask methodology for high resolution remote sensing data combining information from high and medium resolution optical sensors. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(5), 588-596.

<https://doi.org/10.1016/j.isprsjprs.2011.03.005>

Shu, J. S. & Freeman, H. (1990) Cloud shadow removal from aerial photographs. *Pattern*

*Recognition*, 23(6), 647-656. [https://doi.org/10.1016/0031-3203\(90\)90040-R](https://doi.org/10.1016/0031-3203(90)90040-R)

Suthaharan, S. (2016). Support Vector Machine. *Machine Learning Models and*

*Algorithms for Big Data Classification*, 36, 235-297.

[https://doi.org/https://doi.org/10.1007/978-1-4899-7641-3\\_9](https://doi.org/https://doi.org/10.1007/978-1-4899-7641-3_9)

Zheng, H., Sherazi, S. W. A. & Lee, J. Y. (2021). A Stacking Ensemble Prediction

Model for the Occurrences of Major Adverse Cardiovascular Events in Patients with Acute Coronary Syndrome on Imbalanced Data. *IEEE Access*, 9, 113692-

113704. <https://doi.org/10.1109/ACCESS.2021.3099795>



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

Postboks 5003  
NO-1432 Ås  
Norway