

## Obligatorisk innlevering 2 – Supervised learning

### Del 1: Linear Classifiers

#### Tuning

Jeg lagde en enkel for-loop som testet forskjellige verdier av epochs og eta, og lagret resultatet for best accuracy. Den ligger i koden, men jeg legger ved et bilde for enkelhetsskyld:

```
etas = [0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0] # different learning rates
epochs = [1, 10, 50, 100, 200] # different number of epochs

best_accuracy = 0.0
best_eta = None
best_epochs = None

for eta in etas:
    for epoch in epochs:
        classifier = NumpyLinRegClass()
        classifier.fit(X_train, t2_train, eta=eta, epochs=epoch)
        y_pred = classifier.predict(X_val)
        accuracy_value = accuracy(t2_val, y_pred)

        if accuracy_value > best_accuracy:
            best_accuracy = accuracy_value
            best_eta = eta
            best_epochs = epoch

print("Best accuracy:", best_accuracy)
print("Best eta:", best_eta)
print("Best epochs:", best_epochs)
```

Merkbart ved testing var at jeg ikke kunne sette epochs til å være høyere enn 200, da dette resulterte i en runtime-error. Med et høyere antall epochs, vil modellen ha flere muligheter til å tilpasse seg treningssettet, men for mange epochs kan resultere i overtilpasning og generalisering av nye data. Det er derfor verdt å merke seg at selv om jeg med 200 epochs fikk best resultat for accuracy, da dette ble:

*Best accuracy: 0.586*

*Best eta: 0.01*

*Best epochs: 200*

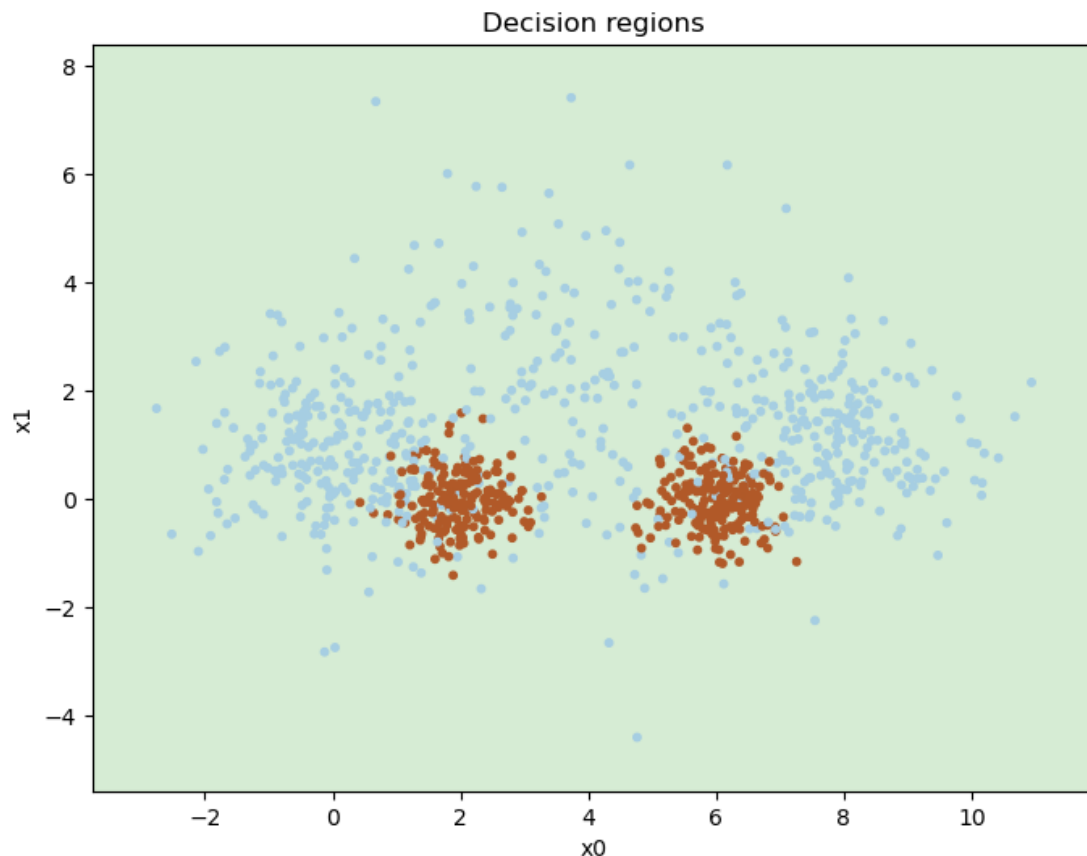
Var det også slik at dersom jeg satte høyeste verdi for epochs til å være 100, ville beste resultat bli da epochs hadde verdi 1:

*Best accuracy: 0.576*

*Best eta: 1e-05*

*Best epochs: 1*

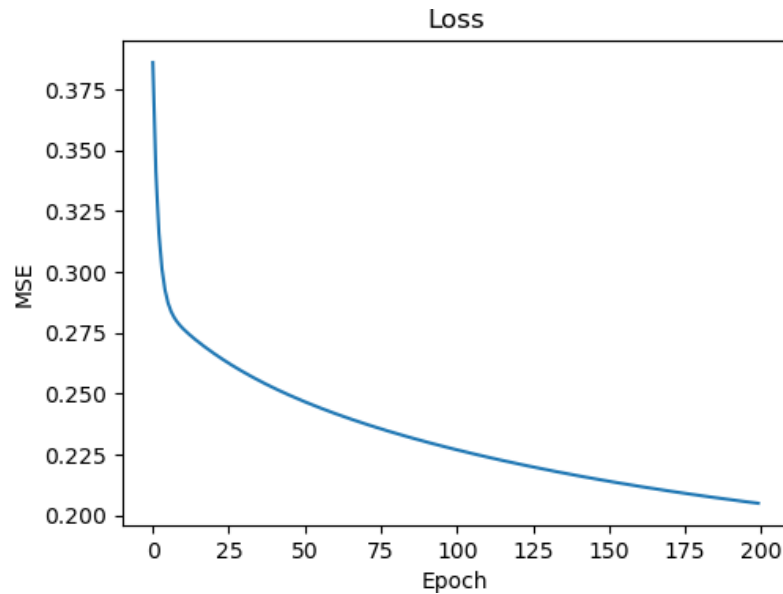
Når jeg da plottet decision-boundaries for resultatet som ga best accuracy, fikk jeg følgende resultat:



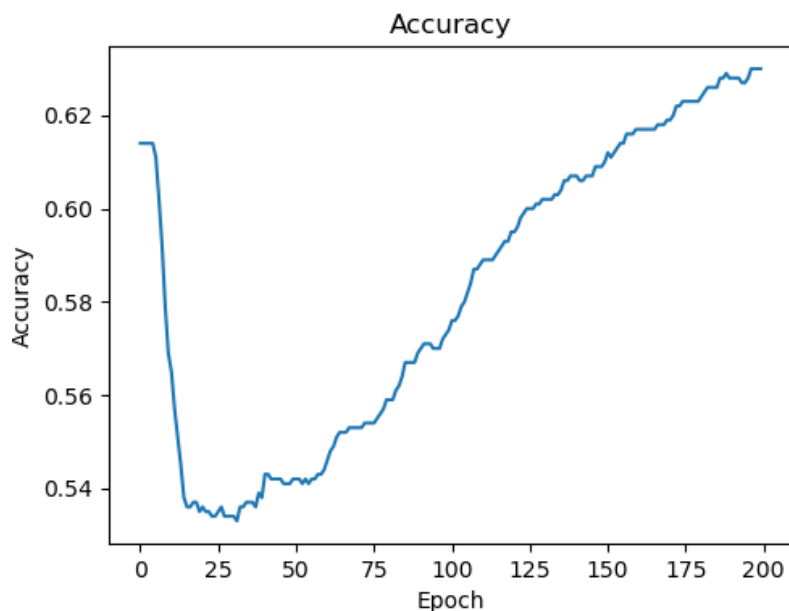
## Loss

Ut fra grafene under kan vi se at grafen for loss er relativt monoton. Da monotont avtagende, fordi funksjonsverdiene er avtar utover i epokene. Vi ser at verdien for loss synker relativt mye i den første delen da vi trener klassifiserer-en vår, tidlig i epokene, og avtar mindre jo lenger ut vi kommer.

Dette er egentlig litt som forventet, da modellen har flere muligheter for å tilpasse seg treningssettet jo høyere verdien vi setter for epochs, hvilket burde resultere i mindre tap.



Når det kommer til accuracy, har grafen kun en liten del som er monoton. Etter dette er det til en viss grad noen svingninger. Likevel ser vi at der svingningene begynner øker accuracy for det meste, selv om det er svingninger. Dette er som forventet, da dersom antall epoker er for lavt, vil ikke modellen ha tilstrekkelig tilpasset dataene. Generelt sett så burde accuracy øke med antall epoker opptil et punkt der det enten stabiliseres eller begynner å avta. Dette ville vi nok sett enda bedre i grafen dersom vi hadde et høyere antall epoker.



## Scaling

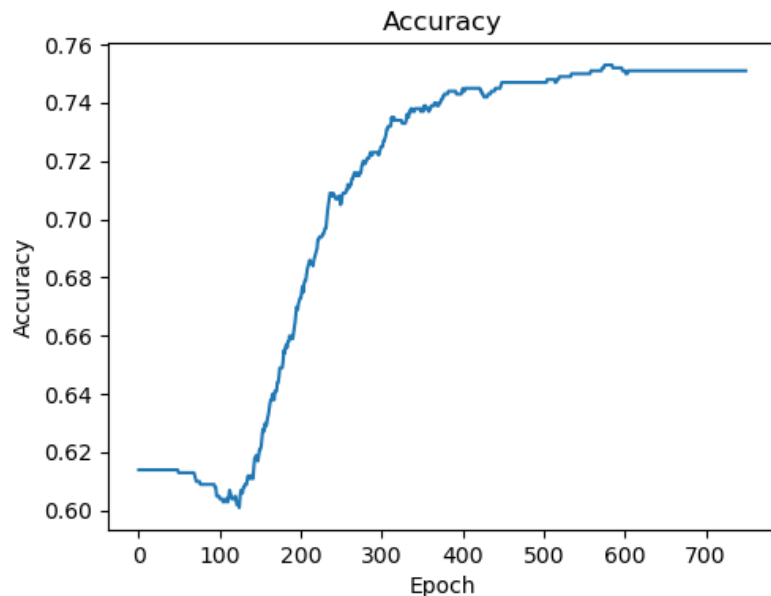
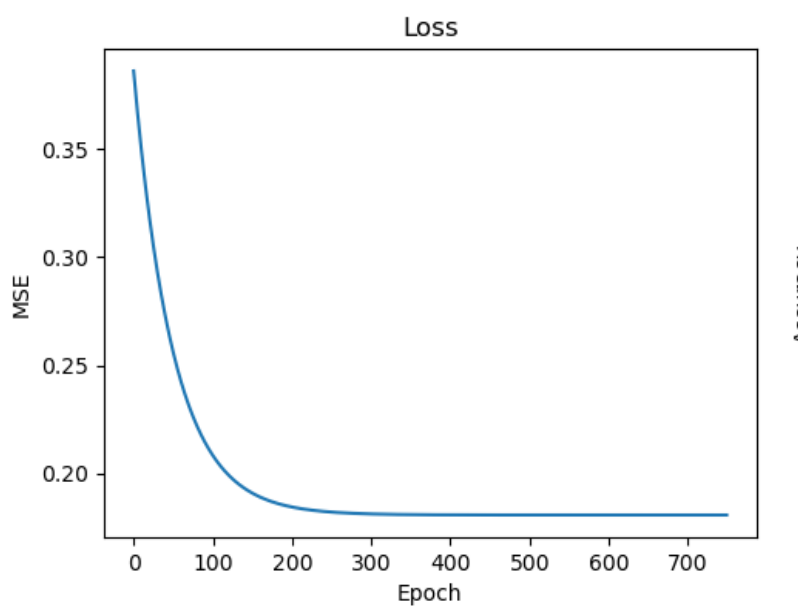
Jeg scalet dataene med en standard scaler, og testet forskjellige hyperparametere på samme måte som i begynnelsen av oppgaven med de nye variablene. Jeg fant da at de beste verdiene ble:

Best accuracy: 0.704

Best eta: 0.01

Best epochs: 750

Dermed ble grafene for loss og accuracy med disse hyperparameterne:



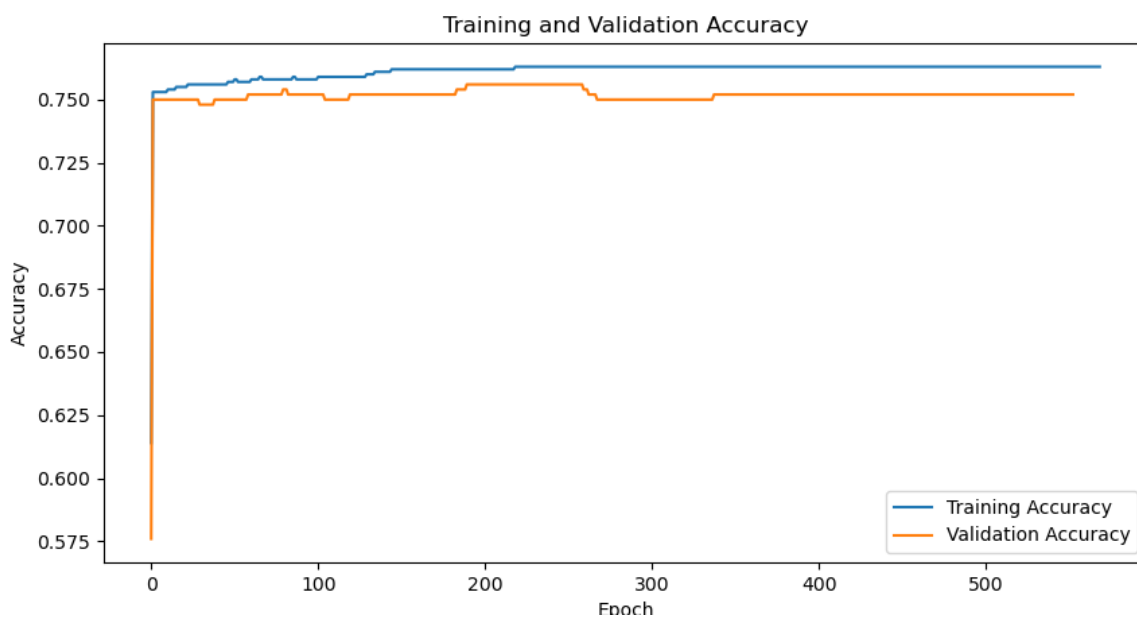
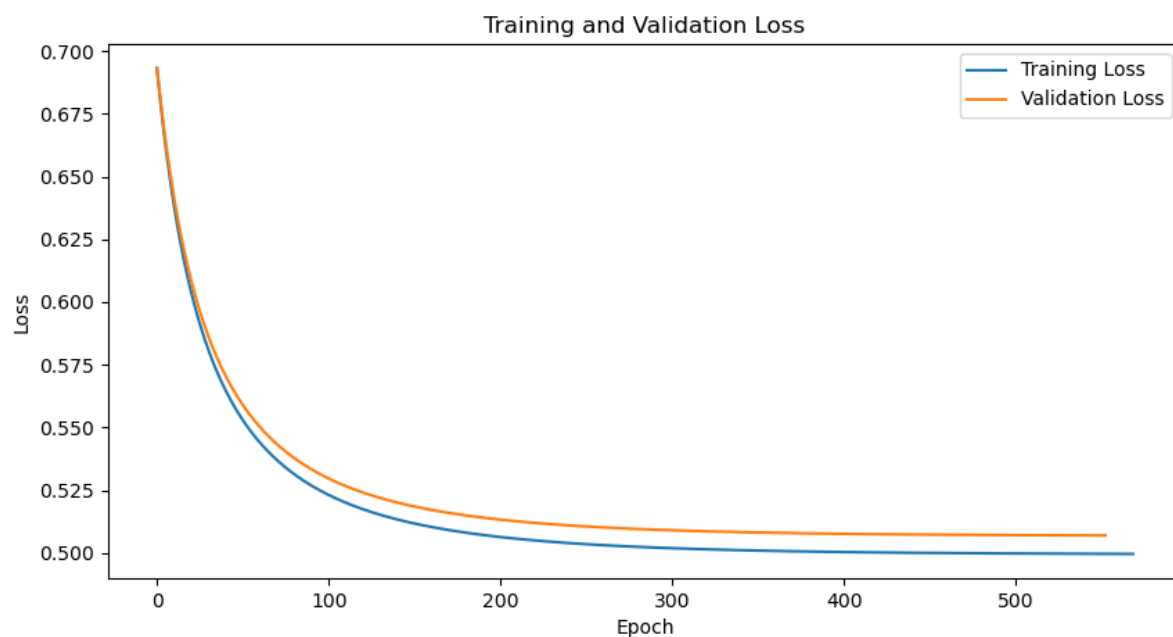
## logistic regression

Jeg trente denne på samme måte som med den lineære og fikk følgende resultat:

Best accuracy: 0.712

Best eta: 0.1

Best tol: 1e-05



Sett fra grafene over, kan vi si at denne har både høyere accuracy og høyere loss enn den lineære.

## multi-class classifiers

### Del 2: Multilayer nerural networks

Her som i de tidligere oppgavene laget jeg en nøstet for loop, for å teste forskjellige verdier for variablene. Dette er noe av resultatene jeg fikk:

Learning rate: 0.01, Epochs: 200, Validation accuracy: 0.576

Learning rate: 0.001, Epochs: 50, Validation accuracy: 0.624

Learning rate: 0.001, Epochs: 100, Validation accuracy: 0.778

Learning rate: 0.001, Epochs: 200, Validation accuracy: 0.804

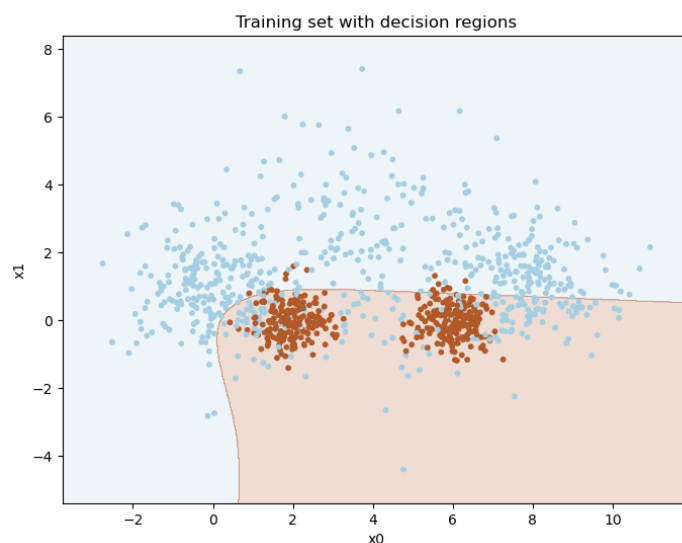
Learning rate: 0.0001, Epochs: 50, Validation accuracy: 0.576

Learning rate: 0.0001, Epochs: 100, Validation accuracy: 0.69

Learning rate: 0.0001, Epochs: 200, Validation accuracy: 0.572

Best hyperparameters: Learning rate: 0.001, Epochs: 200, Validation accuracy: 0.804

Plotter treningssettet sammen med decision regions med disse hyperparameterene:



**\*\*nytt for forsøk 2**

**Kommentar:**

kodefilene mine er stort sett like som de jeg leverte første gang, med unntak av at jeg har en ny `one_vs_rest`, som jeg skrev til oppgaven med samme navn, og at jeg har skrevet litt ekstra kode her og der i de andre, kun for å få kjørt de forskjellige klassifikatorene for å finne accuracy på train, val, test-set, og for å finne precision og recall.

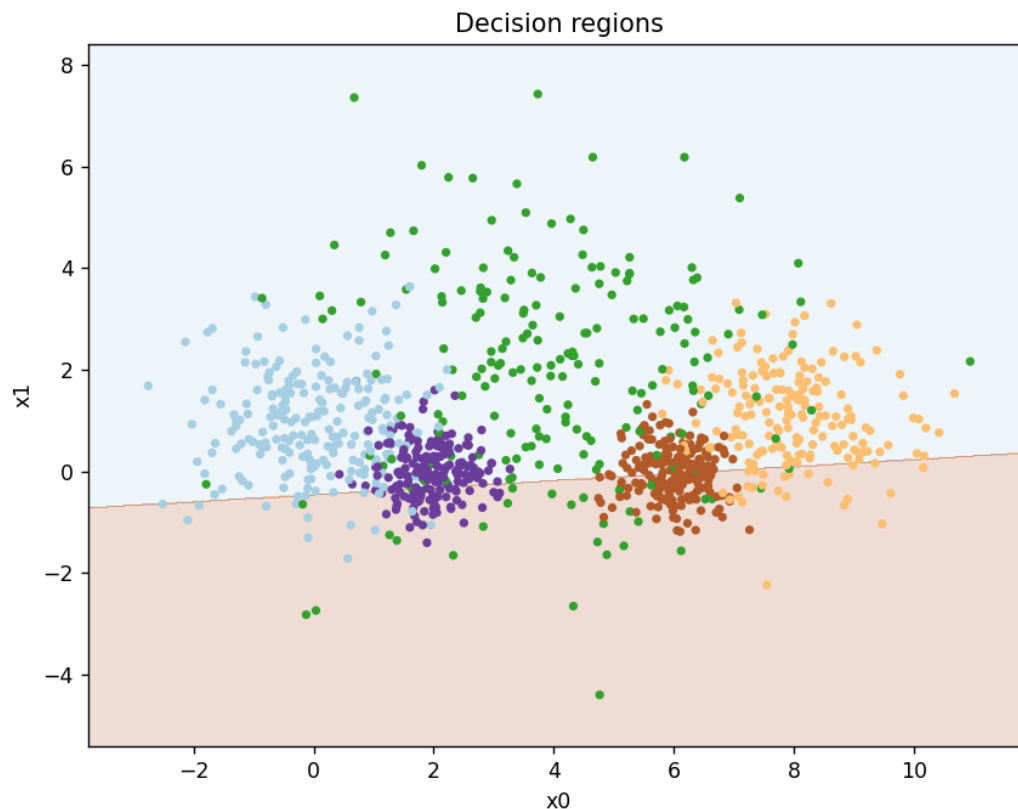
Satser på at dette er nok for andre forsøk, da det var det som sto i tilbakemeldingen 😊

## One-vs-rest

Resultater med multi-class-klassifikator:

Accuracy for validation-set: 0.20

Accuracy for training-set: 0.232



## Binary task

	Test-set	Validation-set	Training-set
Linear classifier	0.596	0.576	0.614
Logistic classifier	0.556	0.542	0.575
MLP-linreg classifier	0.75	0.736	0.77

Generelt sett er det vanlig at modellens resultater på test-settet er litt dårligere enn på valideringssettet. Dette skyldes at test-settet består av helt nye data som modellen ikke tidligere har sett, mens valideringssettet brukes til å justere modellens parametere og hyperparametere. Derfor kan valideringssettet være overtilpasset i forhold til testsettet, noe som kan føre til at modellen ikke genererer like gode resultater på testsettet.

Når vi ser på resultatene i tabellen, ser vi at MLP-linreg-klassifikatoren generelt har de beste resultatene på alle tre settene av data, med høyest nøyaktighet. Linear-klassifikatoren har høyest nøyaktighet på treningssettet, men har dårligere resultater på både validerings- og testsettet. Dette indikerer at modellen kan ha overtilpasset til treningsdataene, noe som gjør at den ikke generaliserer godt til nye data.

For hver av disse klassifikatorene vil vi ønske å se at resultatene på valideringssettet og testsettet er så nærme som mulig resultatene på treningssettet. Dette indikerer at modellen generaliserer godt til nye data og ikke overtilpasser til treningssettet. I tillegg ønsker vi å se at alle modellene har høy nøyaktighet på testsettet, siden det er det settet som vil bli brukt til å evaluere modellens ytelse på nye og uprøvde data.

## Precision and recall

Under er en tabell over recall og precision for de forskjellige klassifikatorene. Jeg var litt usikker på om jeg fikk lov til å bruke ml-biblioteker for denne oppgaven, og valgte derfor å prøve meg uten sklearn.

Jeg brukte samme kode (med forskjellige variabler) for alle klassifikatorene, så litt usikker på om det ble riktig selv om alt kjørte uten problem.



```
classifier = NumpyLinRegClass()
losses, accuracies = classifier.fit(X_train_scaled, t2_train, eta=0.01, epochs=200)

# Predict classes on the test set
y_pred = classifier.predict(X_test_scaled)

# Get the true classes for the first class
y_true = t2_test == 0

# Calculate true positives, false positives, and false negatives for the first class
tp = np.sum((y_pred == 0) & (y_true == 0))
fp = np.sum((y_pred == 0) & (y_true == 1))
fn = np.sum((y_pred == 1) & (y_true == 0))

# Calculate precision and recall for the first class
precision = tp / (tp + fp)
recall = tp / (tp + fn)

print("Precision for the first class:", precision)
print("Recall for the first class:", recall)
```

classifier	percision	recall
Linear classifier	0.35036496350364965	0.7128712871287128
Logistic classifier	0.23974763406940064	0.37623762376237624
MLP-linreg classifier	0.3269689737470167	6782178217821783

Når vi ser på resultatene, ser vi at både linear og MLP-linreg klassifikatorer har høyere recall enn precision. Dette kan tyde på at disse modellene har en tendens til å klassifisere flere ekte positive eksempler riktig, men også at de kan feilaktig identifisere flere negative eksempler som positive. Derimot ser vi at logistic klassifikator har høyere precision enn recall. Dette kan tyde på at denne modellen har en tendens til å begrense feilaktige positive klassifiseringer, men også at den kan overse noen ekte positive eksempler.

Når vi sammenligner precision og recall mellom de tre klassifikatorene, ser vi at MLP-linreg klassifikatoren har høyest recall, men laveste precision. Linear klassifikatoren har en moderat recall og precision, mens logistic klassifikatoren har lavest recall og precision.