# Predicting Mortgage Defaults

## Can we succeed where banks failed?

Lanza G., Gallo S., Florea R., Cialdea G.

Statistical Learning and Large Data 1&2
Academic year 2024/25

Sant'Anna
Scuola Universitaria Superiore Pisa

# Outline

- Dataset description
- Research questions
- Data manipulation
- Analysing models
- Solving the unbalanced dataset problem
- Conclusions

# Research question

- Can we predict insolvency status with a few relevant features?
- How should we handle a **highly skewed** class distribution?
  Solvent customers outnumbered defaulted ones **100:1**
- In a nutshell, can we succeed where banks failed?

# Dataset

Data are collected by Freddie Mac (a federal agency), which securitizes mortgages for US commercial banks.

Snapshot: 14 features and 80k instances

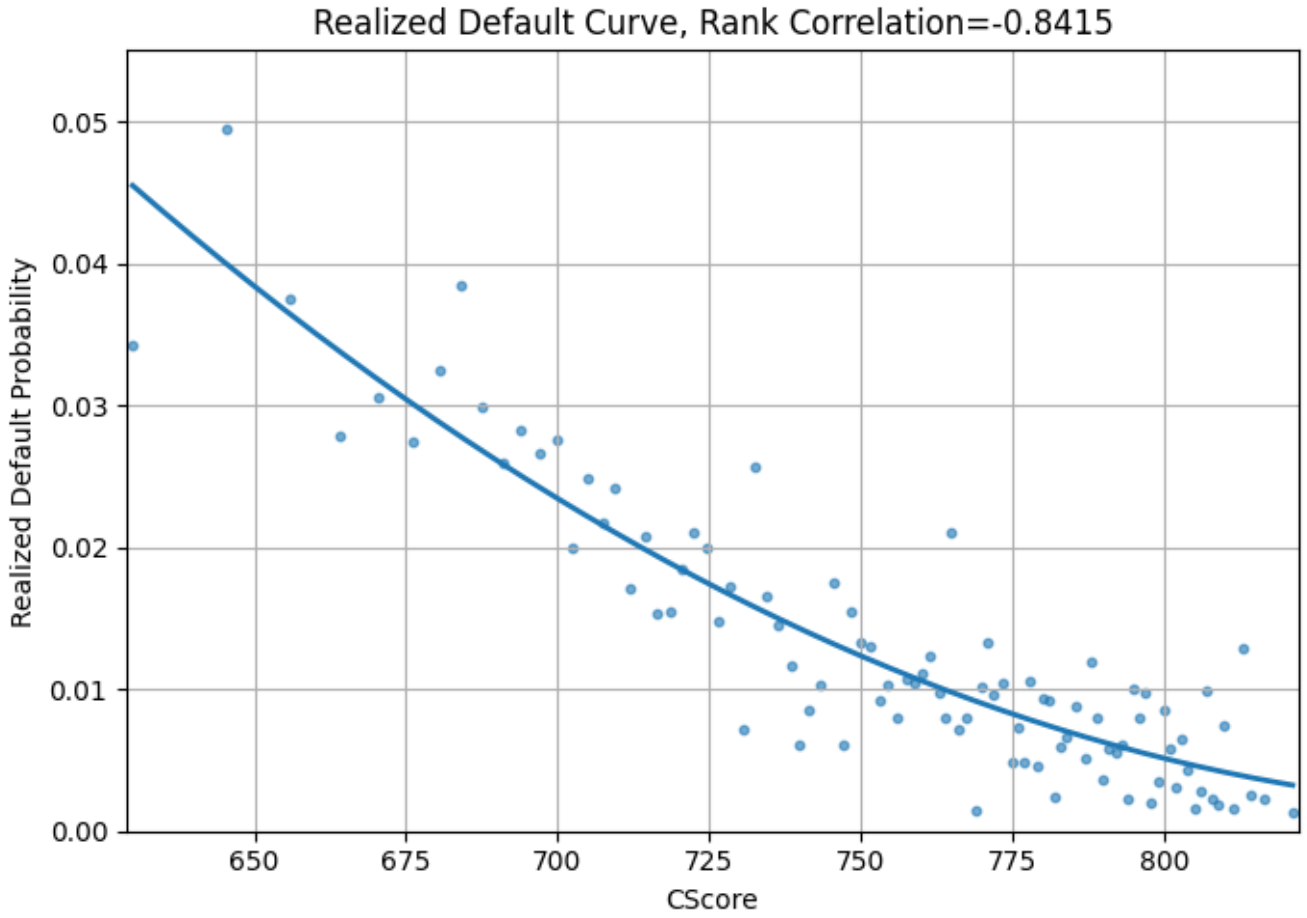
We did some preliminary manipulation: for instance, get dummies from “Applicant race” or get mid values for ‘bands’ variables (ex. Applicant age ‘65-74’ becomes 69.5).

*Features*

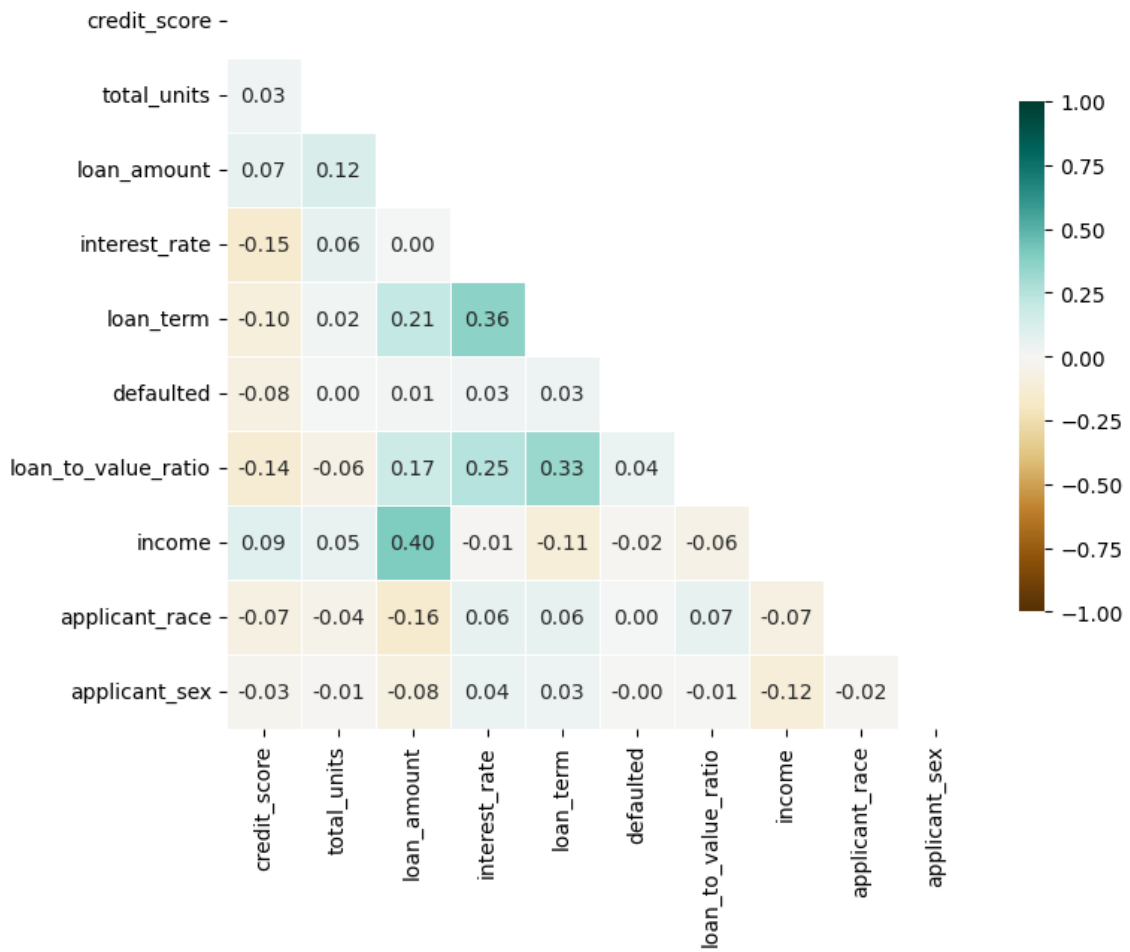
- `credit_score`
- `total_units`
- `loan_amount`
- `interest_rate`
- `loan_term`
- `loan_to_value_ratio`
- `income`
- `applicant_race`
- `applicant_sex`
- `...`
- `DEFAULTED`

# The most predictive feature: CScore

CScore measures an individual's creditworthiness, as assessed by banks based on their historical reliability in repaying borrowed money

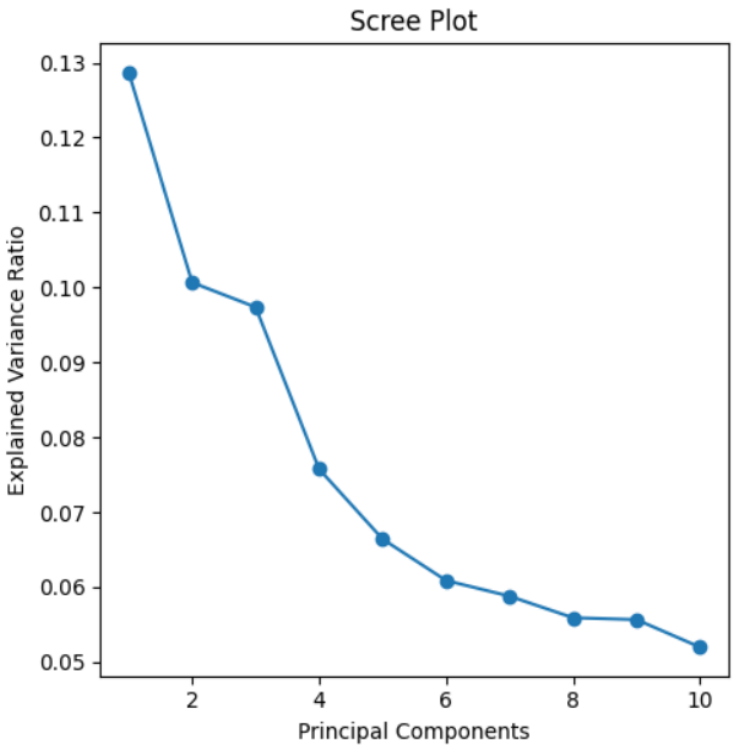


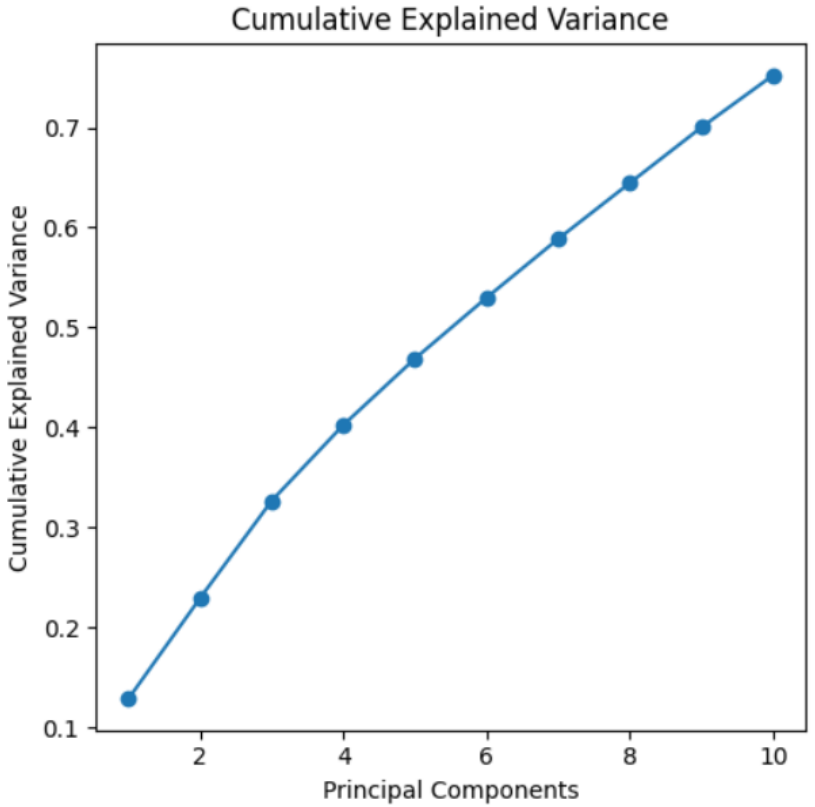
# How much collinearity?



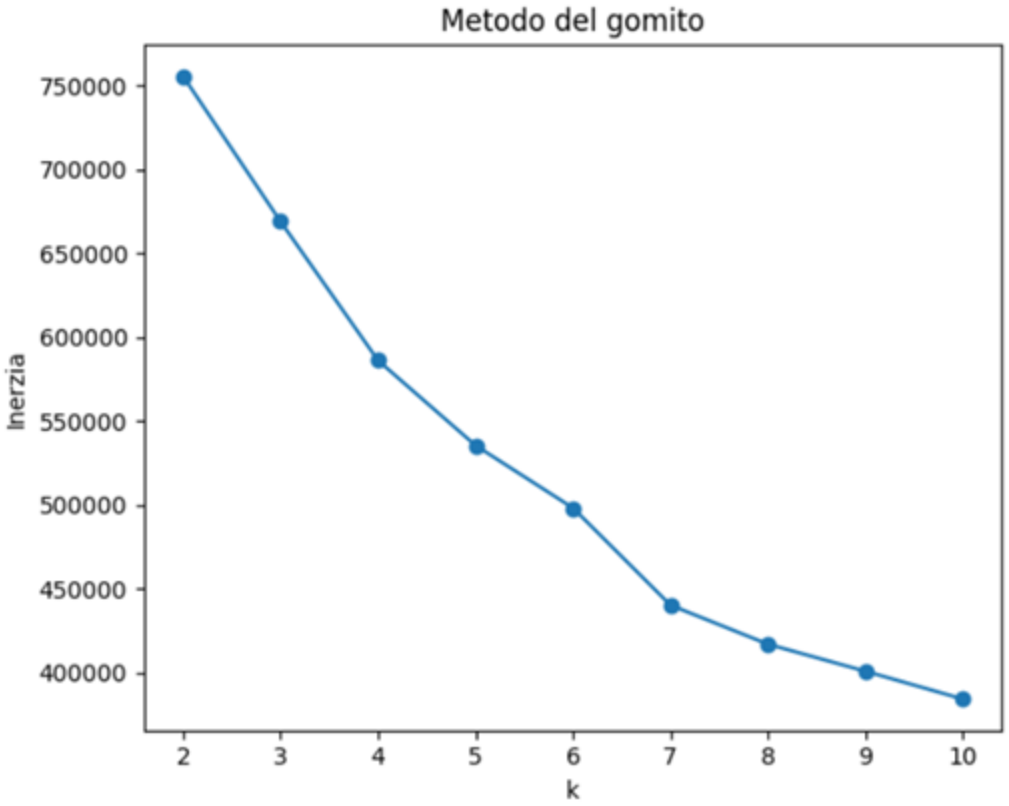

| feature | VIF |
|---|---|
| (const) | (512.23) |
| credit_score | 1.09 |
| total_units | 1.09 |
| occupancy_type | 1.28 |
| debt_to_income_ratio | 1.38 |
| loan_amount | 1.67 |
| interest_rate | 1.36 |
| loan_purpose | 1.37 |
| loan_term | 1.36 |
| defaulted | 1.01 |
| loan_to_value_ratio | 1.56 |
| income | 1.82 |
| applicant_race | 1.06 |
| applicant_sex | 1.02 |
| applicant_age | 1.22 |

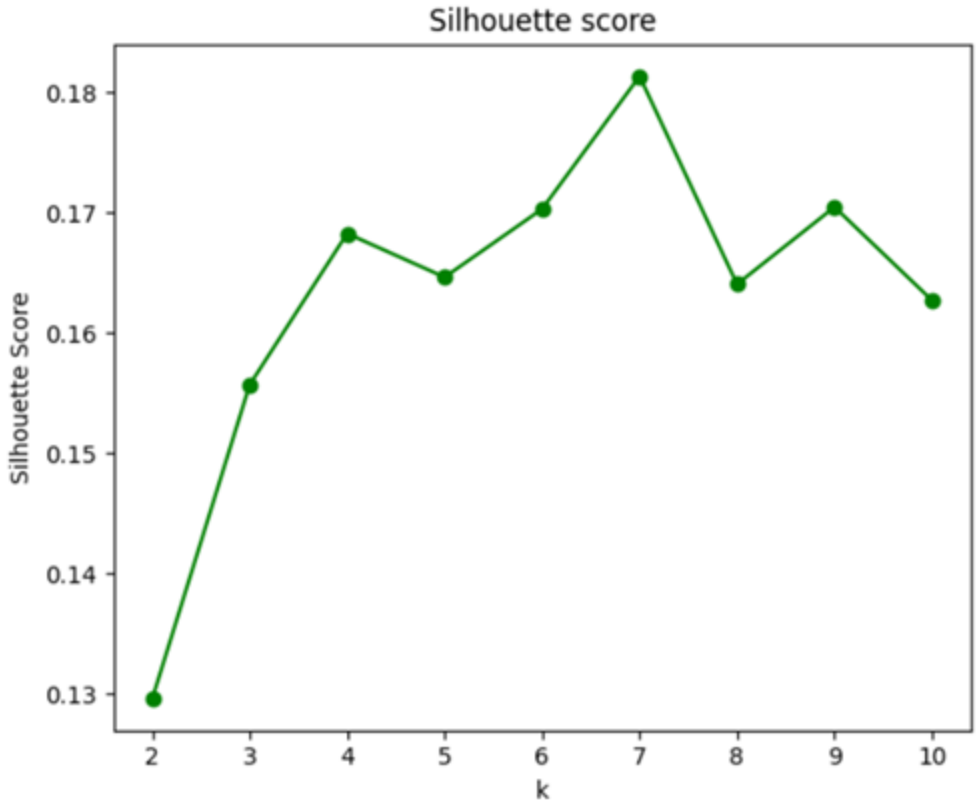
# PCA


Scree Plot
Explained Variance Ratio
0.13
0.12
0.11
0.10
0.09
0.08
0.07
0.06
0.05
2
4
6
8
10
Principal Components

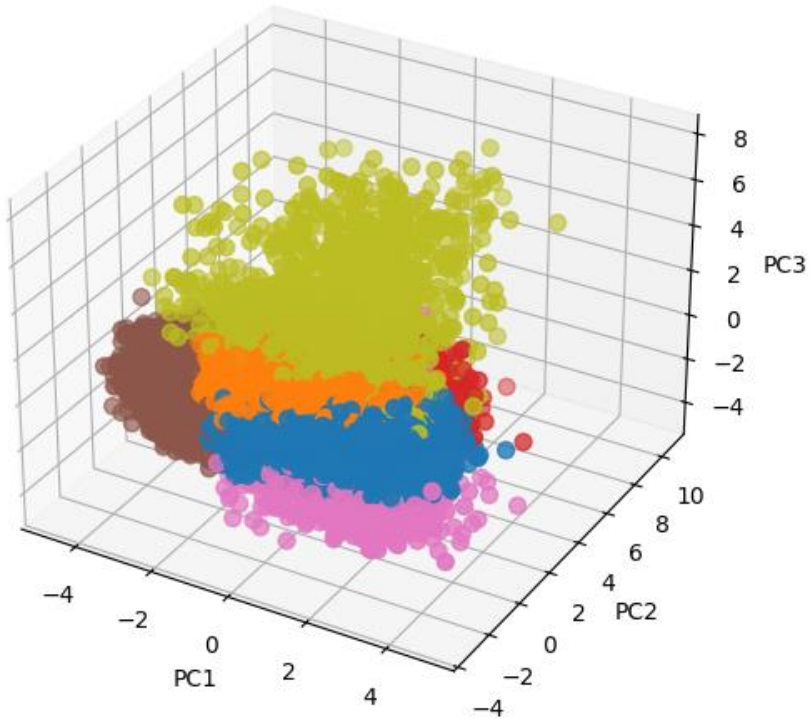

Cumulative Explained Variance
Cumulative Explained Variance
0.7
0.6
0.5
0.4
0.3
0.2
0.1
2
4
6
8
10
Principal Components

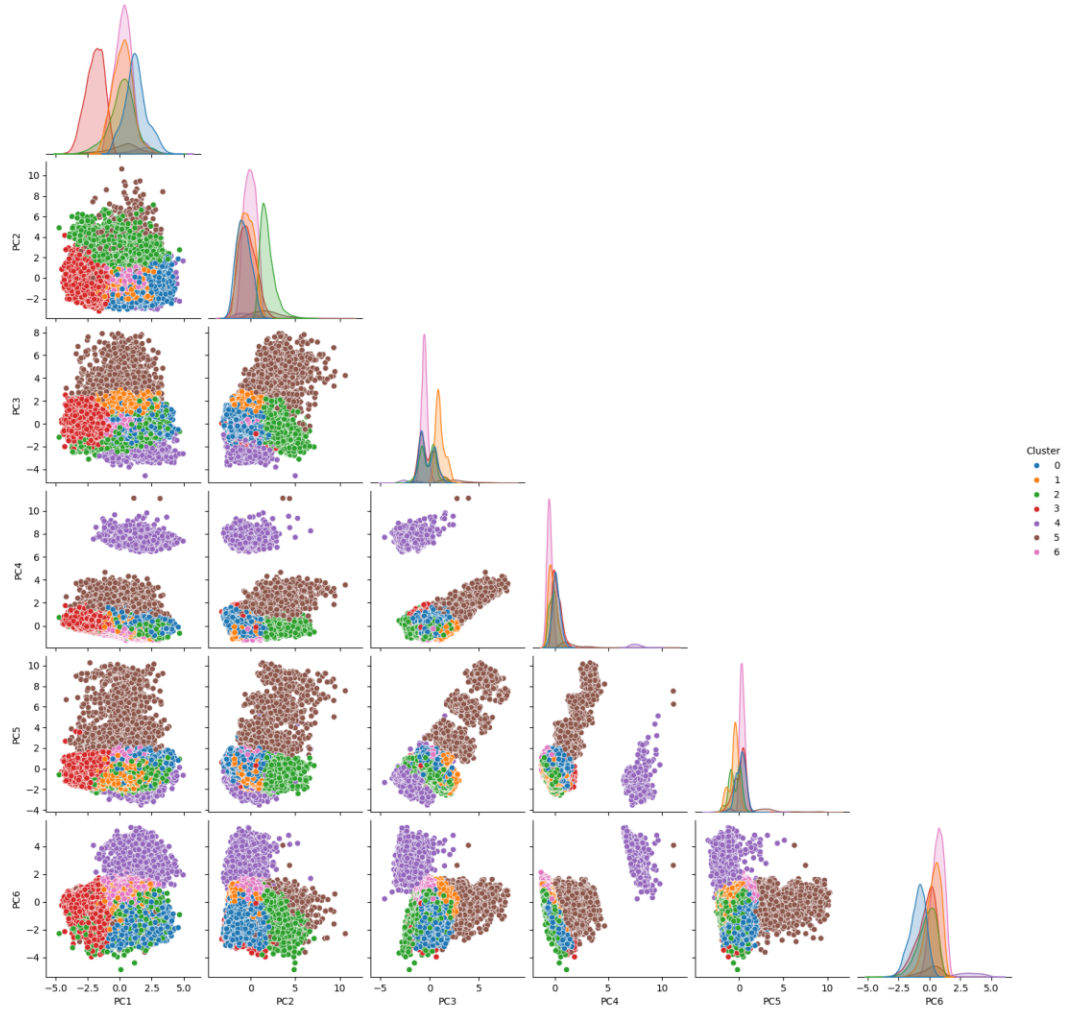
# Clustering – K means



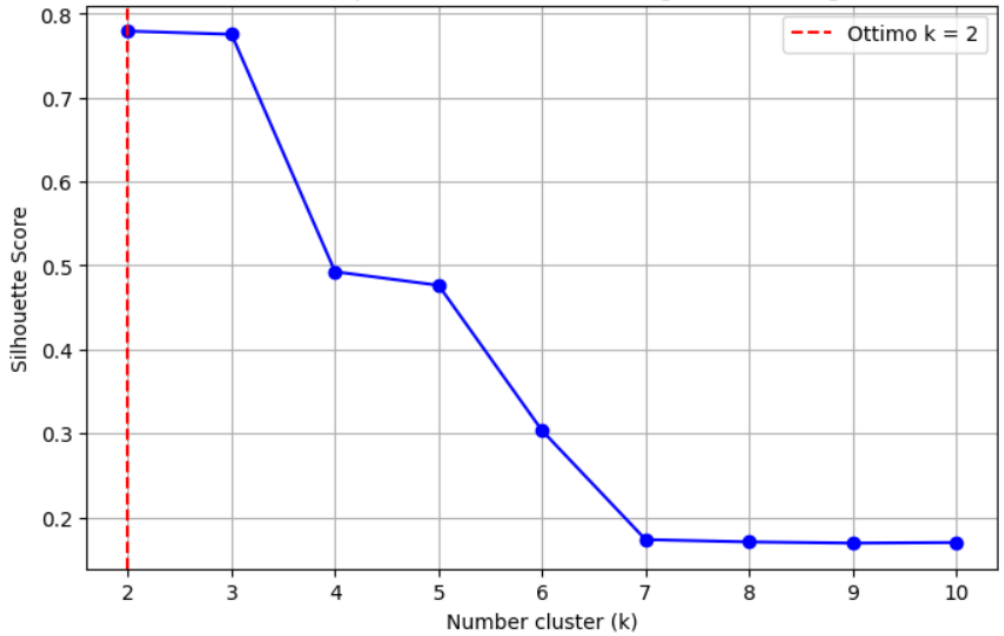
# Clustering – K means

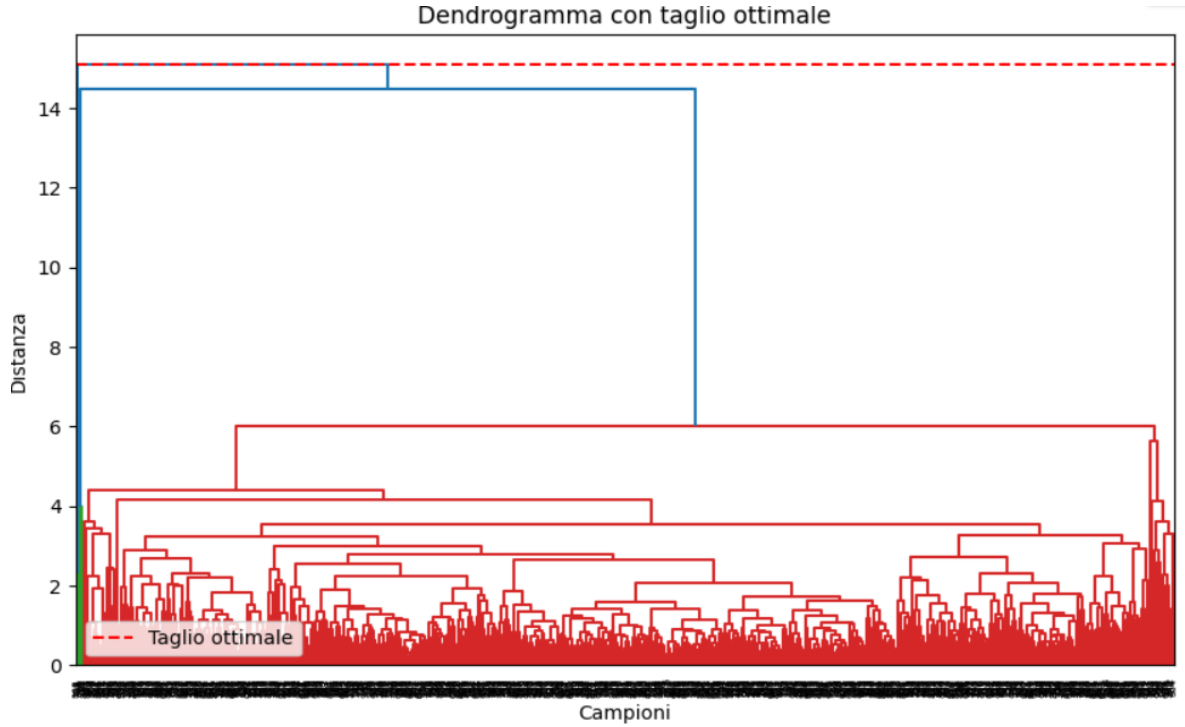


# Clustering – dendrogram

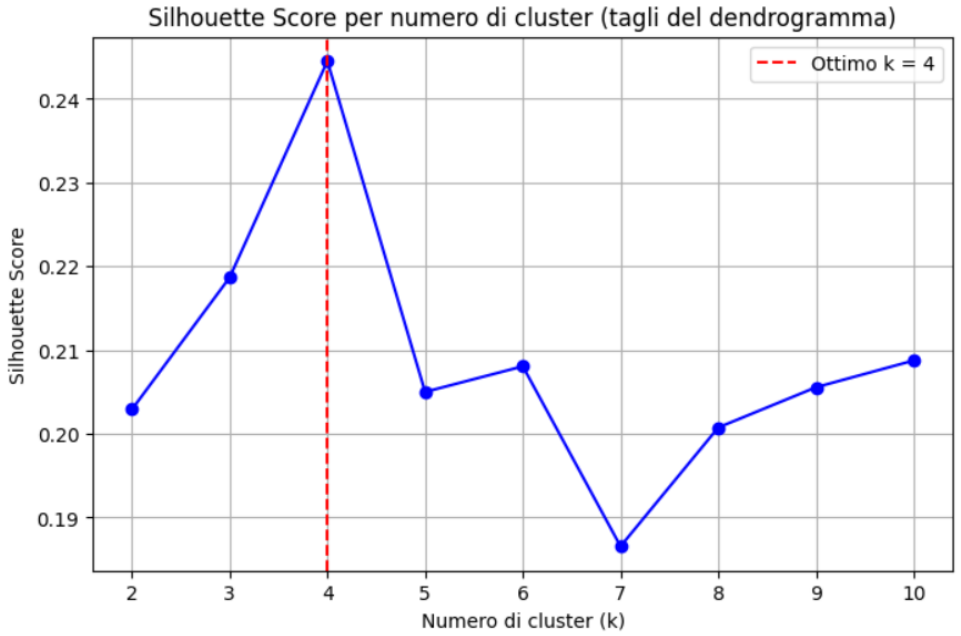
Linkage function: average
Sub-sample: 1000

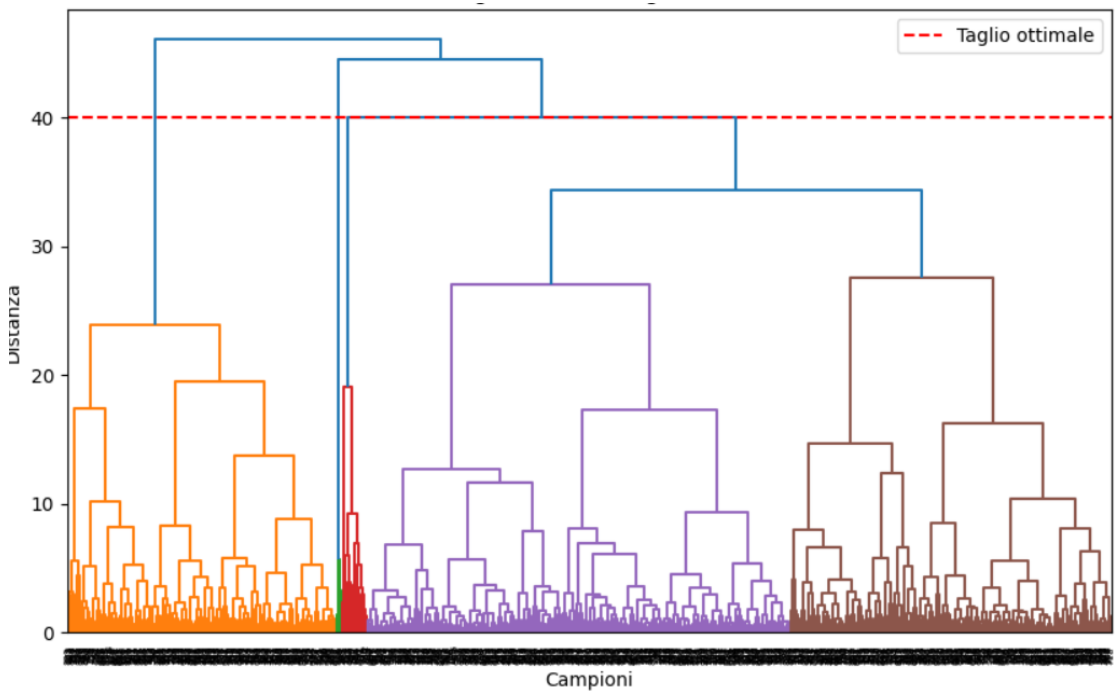


# Clustering – dendrogram

Linkage function: ward, compact
Sub-sample: 1000

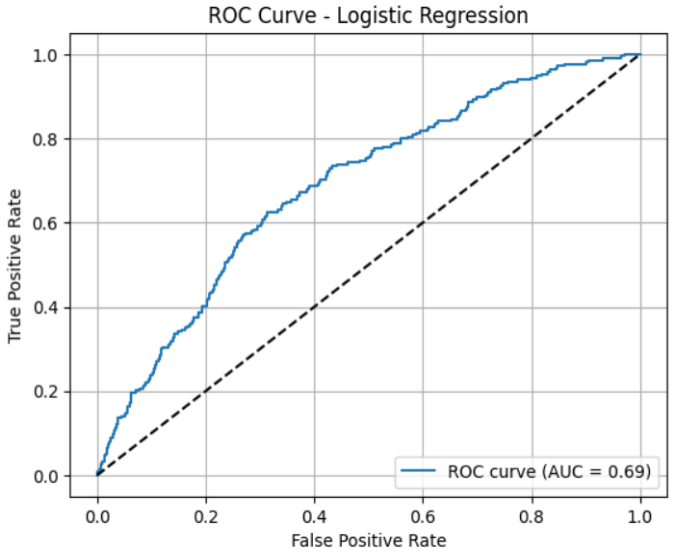


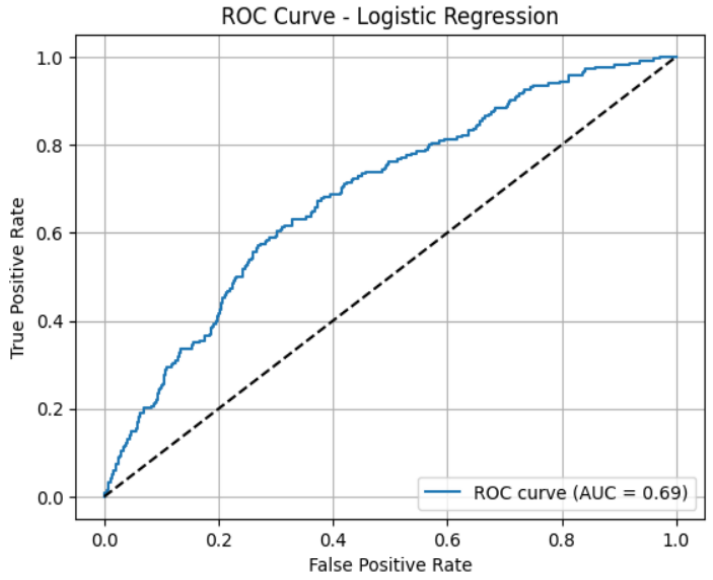



# Logistic PCA

```
Confusion Matrix:
 [[16565     0]
 [  214     0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     16565
           1       0.00      0.00      0.00       214

    accuracy                           0.99     16779
   macro avg       0.49      0.50      0.50     16779
weighted avg       0.97      0.99      0.98     16779
```

```
Confusion Matrix:
 [[9684 6881]
 [  64  150]]
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.58      0.74     16565
           1       0.02      0.70      0.04       214

    accuracy                           0.59     16779
   macro avg       0.51      0.64      0.39     16779
weighted avg       0.98      0.59      0.73     16779
```



# Logistic PCA - 2

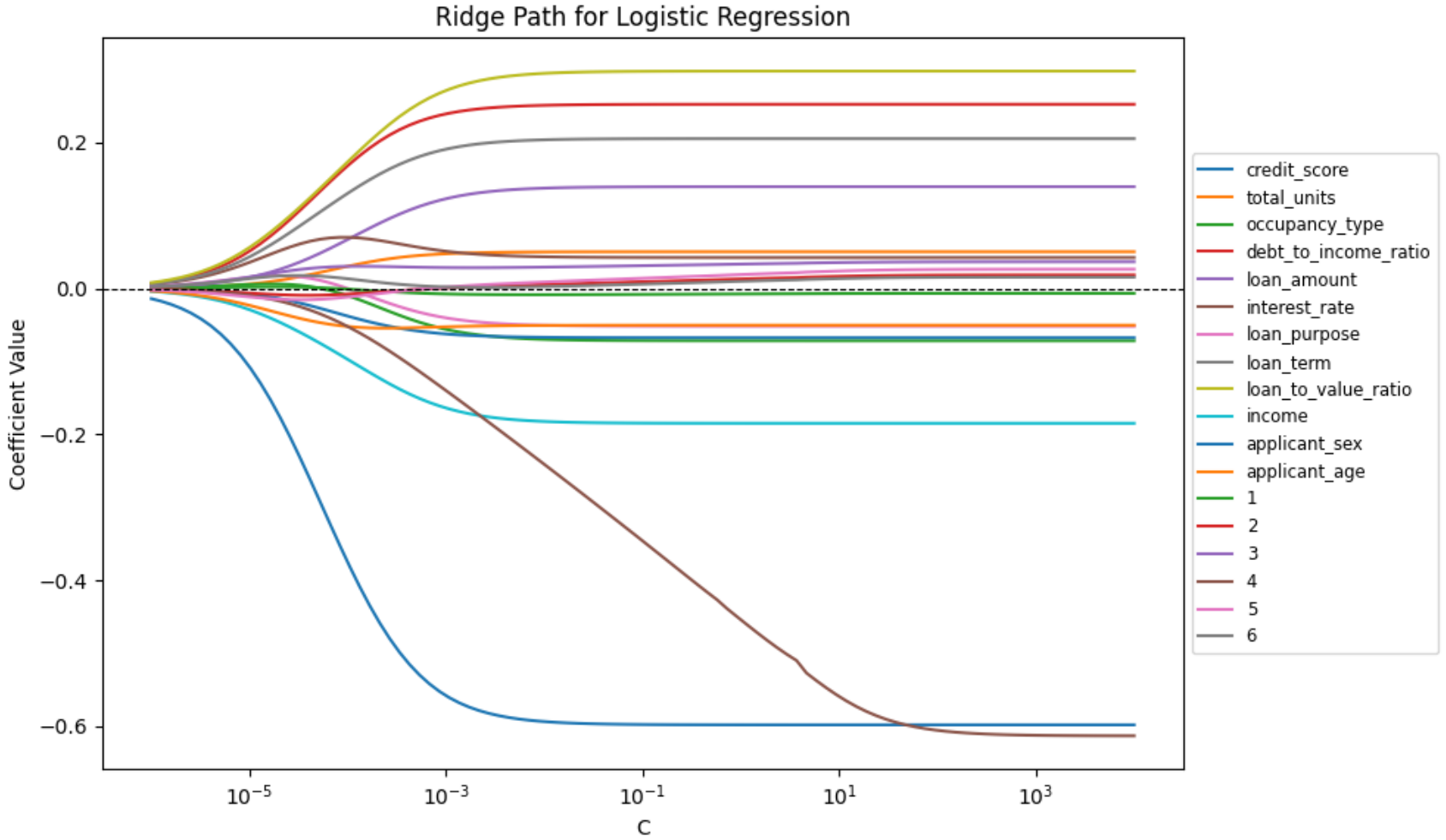
```
credit_score  total_units  occupancy_type  debt_to_income_ratio  loan_amount  interest_rate  loan_purpose  loan_term
loan_to_value_ratio  income  applicant_sex  applicant_age   1   2   3   4   5   6
```

Now, let's take a closer look inside the logistic classifier.

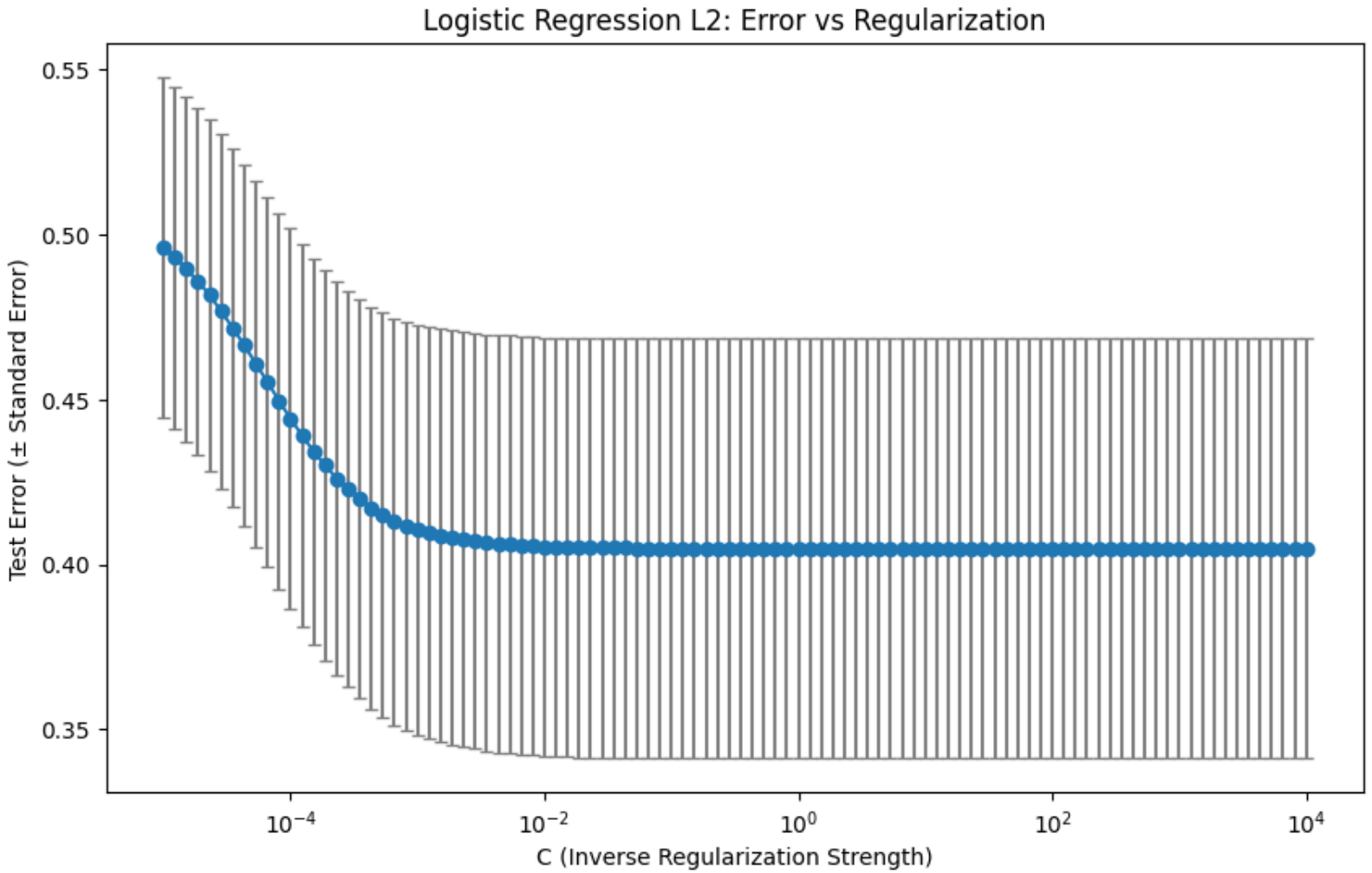
Among all features, the debt-to-income ratio most strongly indicates the likelihood of loan default.

```
[[-0.16363239 -0.04573444  0.1220353   0.31075724  0.05898812  0.18942094
   0.19113466  0.27526974  0.30016855 -0.2518229   0.08009453 -0.15974871
   0.00868074 -0.00174289  0.07159245  0.00964971 -0.10770837  0.09741807]]
```

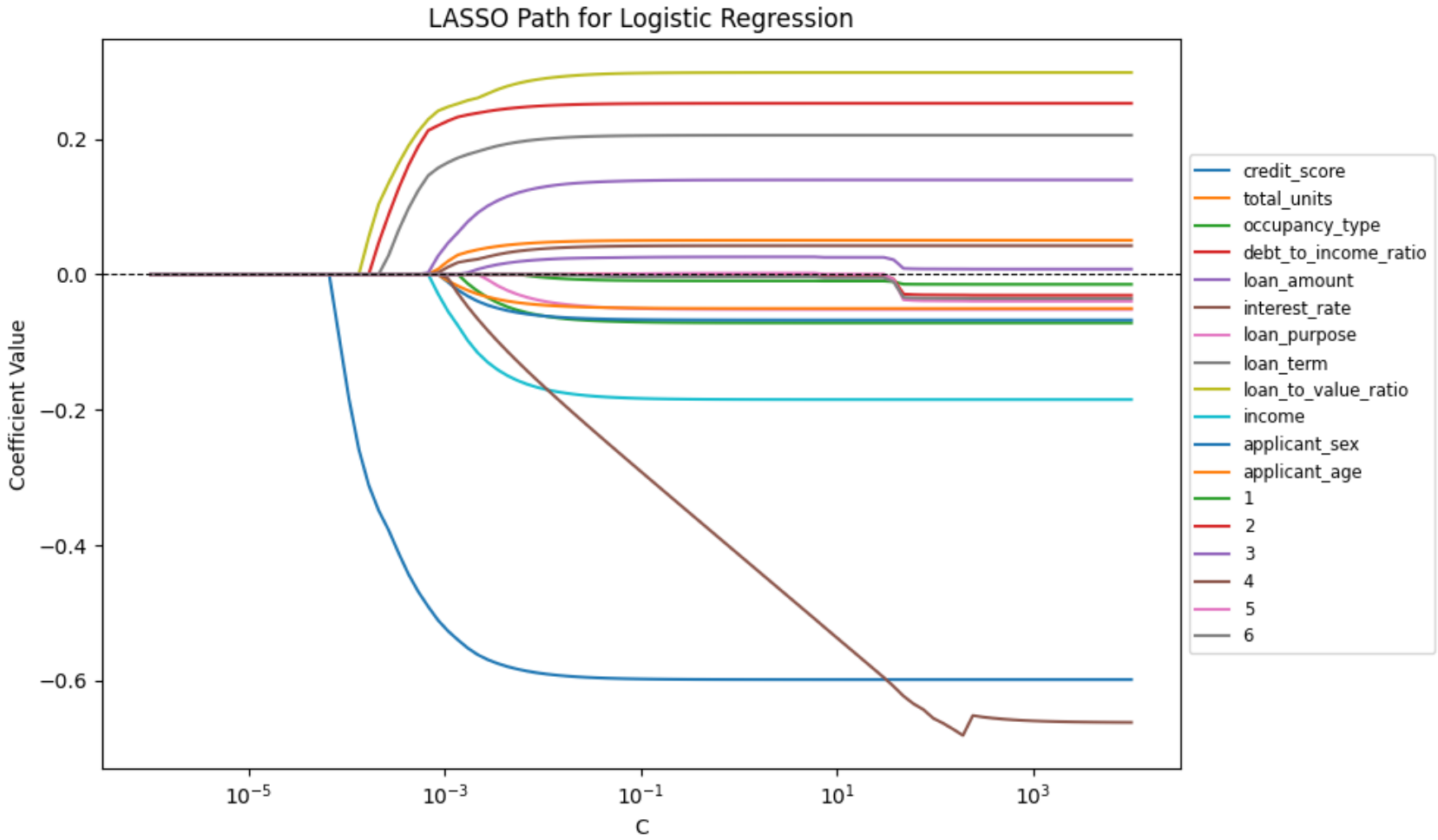
# RIDGE-logistic regression



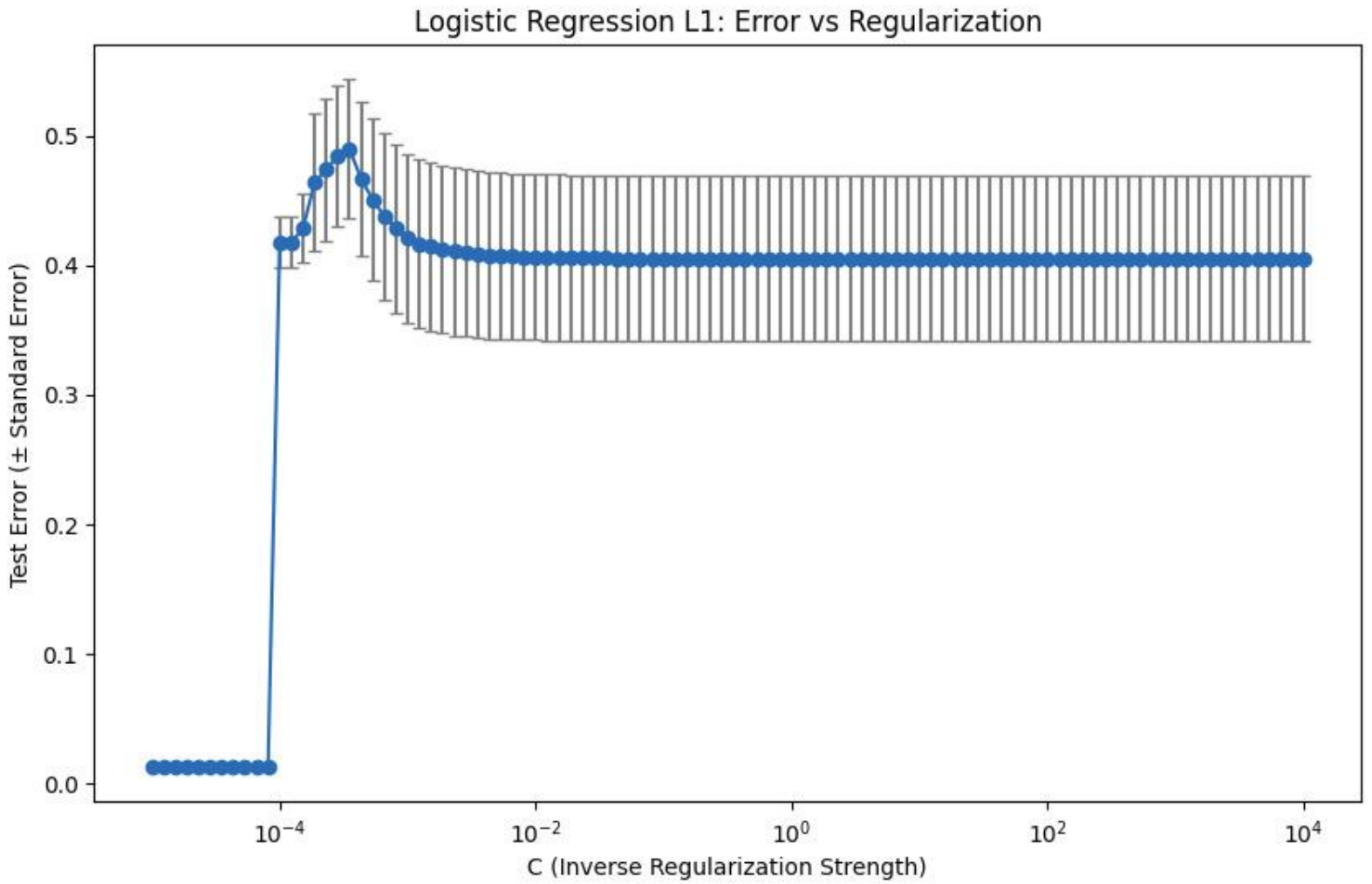
# RIDGE


Logistic Regression L2: Error vs Regularization
Test Error (± Standard Error)
0.55
0.50
0.45
0.40
0.35
$10^{-4}$
$10^{-2}$
$10^{0}$
$10^{2}$
$10^{4}$
C (Inverse Regularization Strength)

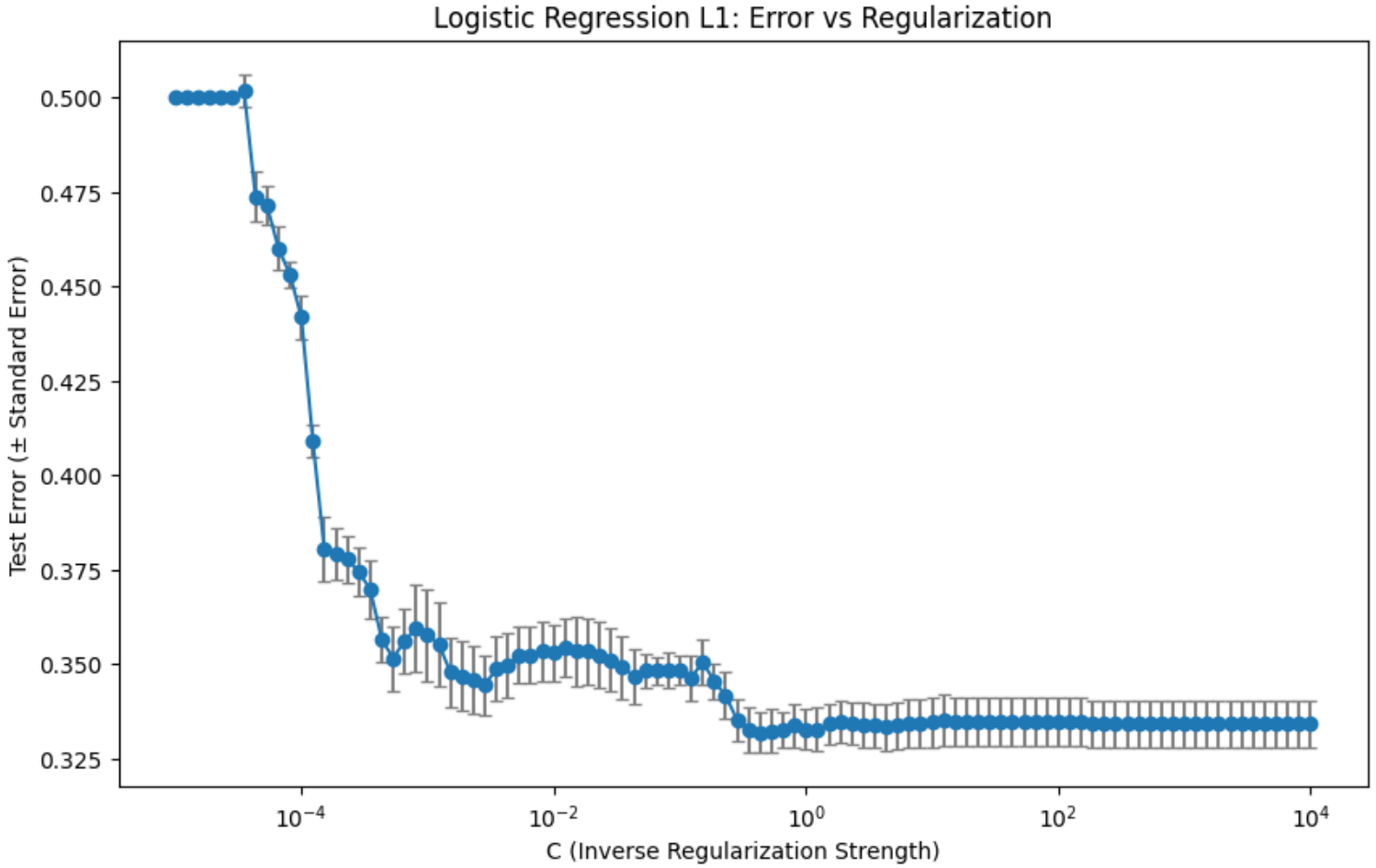
# LASSO-logistic regression



# LASSO


Logistic Regression L1: Error vs Regularization
Test Error (± Standard Error)
0.0
0.1
0.2
0.3
0.4
0.5
$10^{-4}$
$10^{-2}$
$10^{0}$
$10^{2}$
$10^{4}$
C (Inverse Regularization Strength)

# LASSO-undersampling

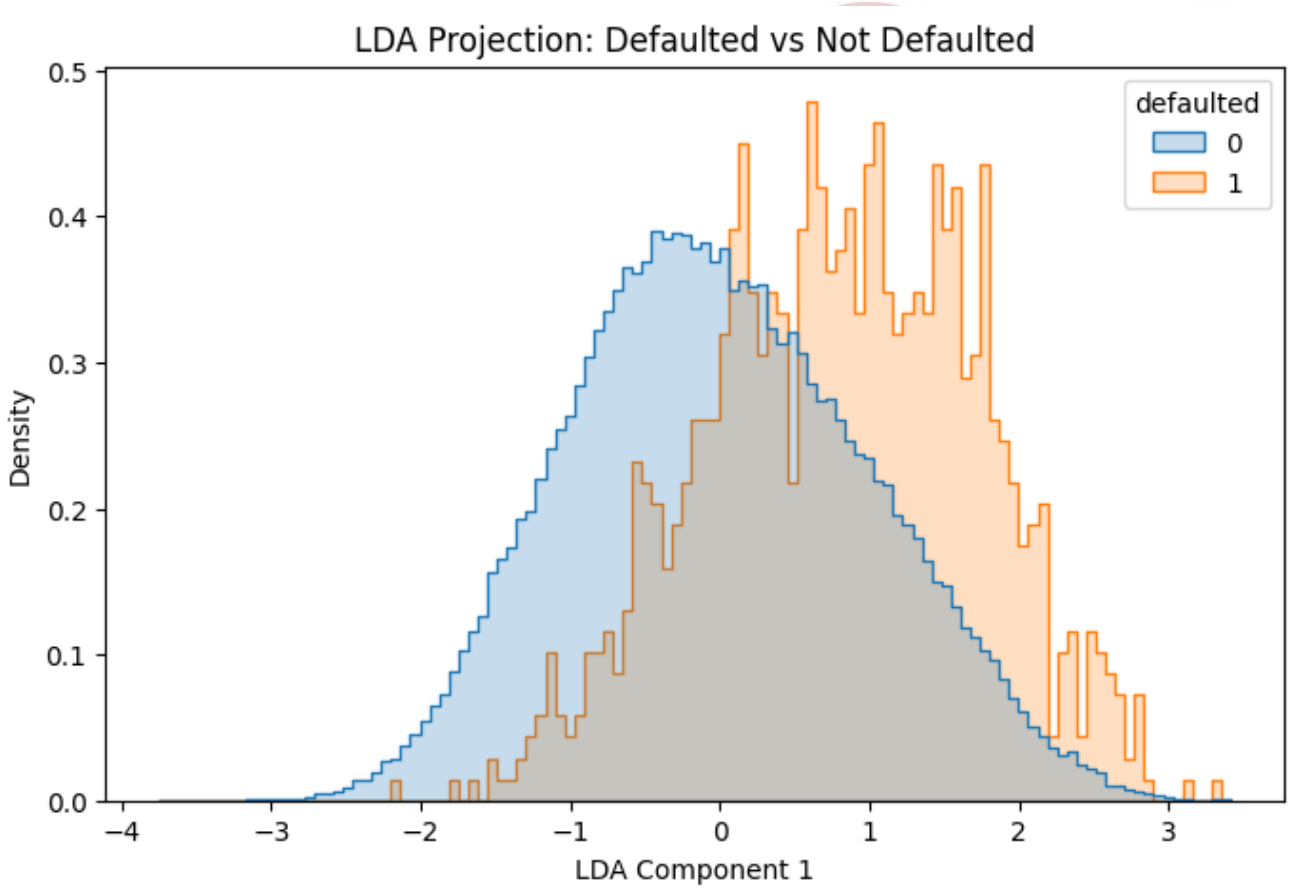


# Linear Discriminant Analysis

Obviously, very poor results:

```
Confusion Matrix:
[[16553     0]
 [  226     0]]
```

But...

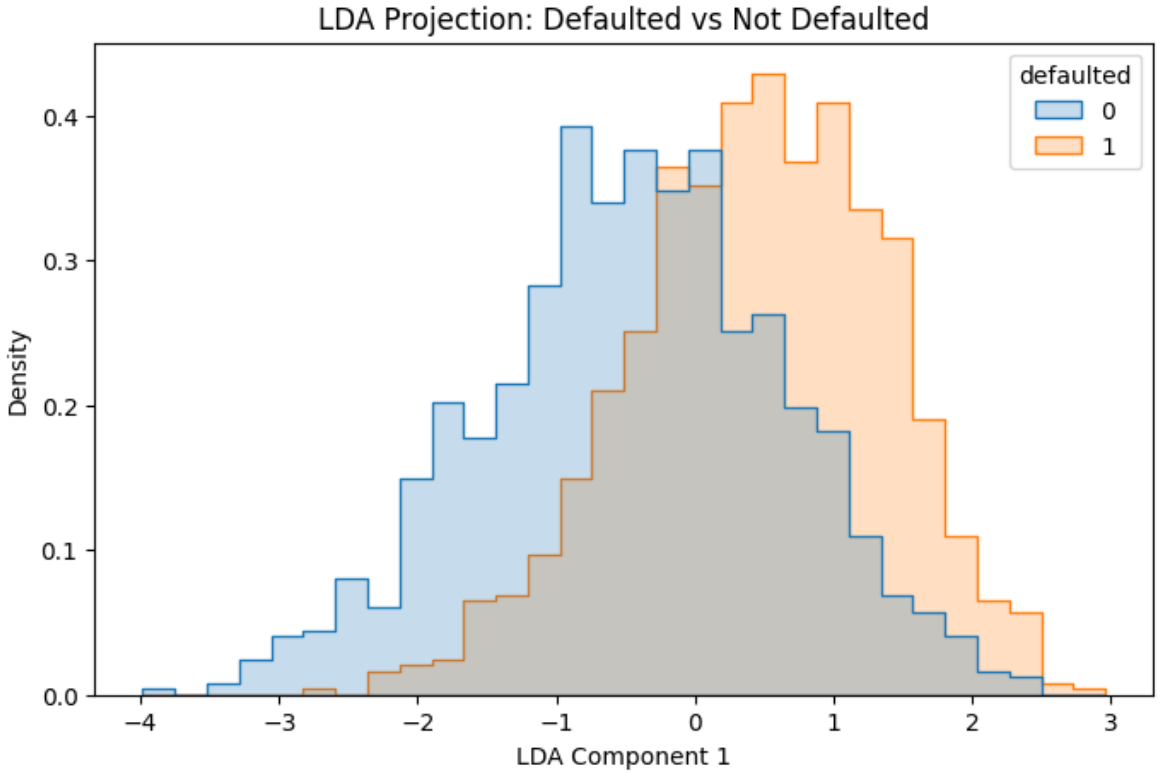


# Linear Discriminant Analysis - undersampling

We are able to extract some information with undersampling:

```
Confusion Matrix:
[[140  64]
 [ 77 147]]
```

| | precision | recall | f1-score |
|---|---|---|---|
| 0 | 0.65 | 0.69 | 0.67 |
| 1 | 0.70 | 0.66 | 0.68 |



# Quadratic Discriminant Analysis

We can do something similar for QDA:

Without undersampling

```
Confusion Matrix:
[[16553     0]
 [  226     0]]
```

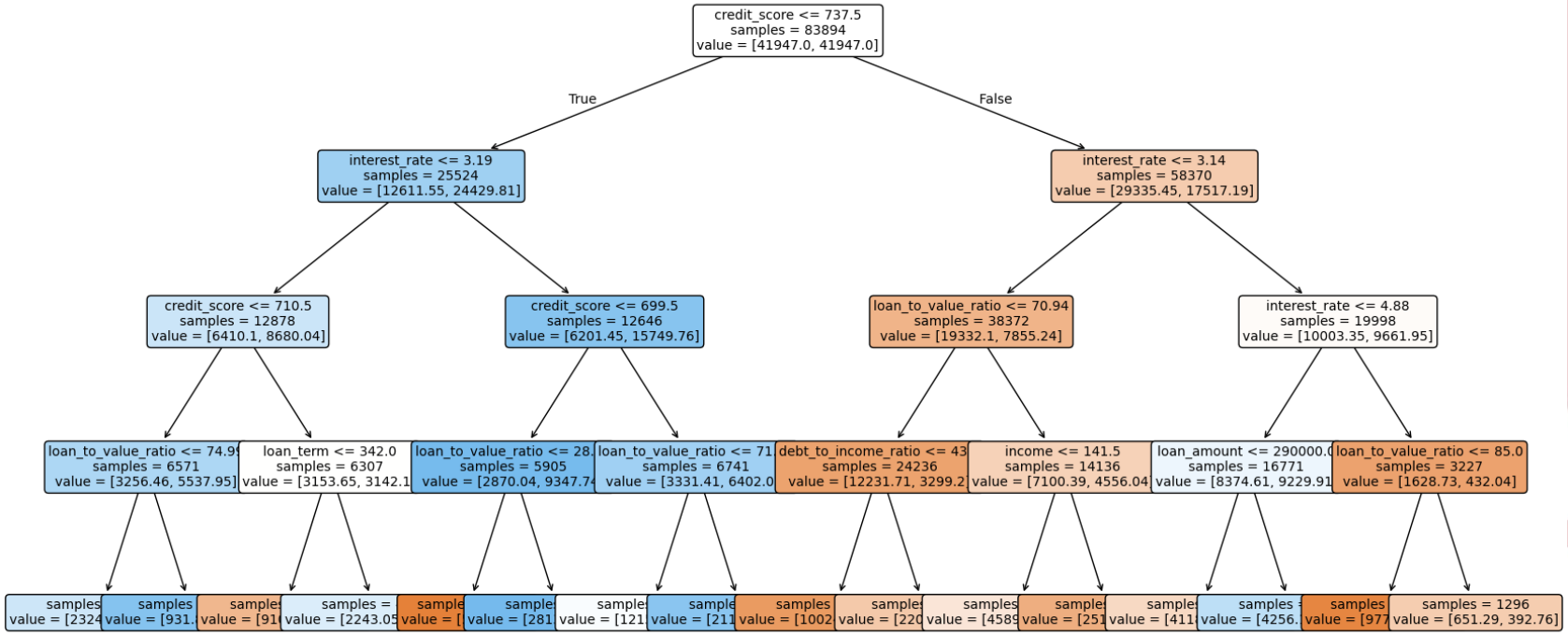
With undersampling

```
Confusion Matrix:
[[ 96 108]
 [ 48 176]]
```

# Decision tree

Decision Trees generate simple rules for making prediction.

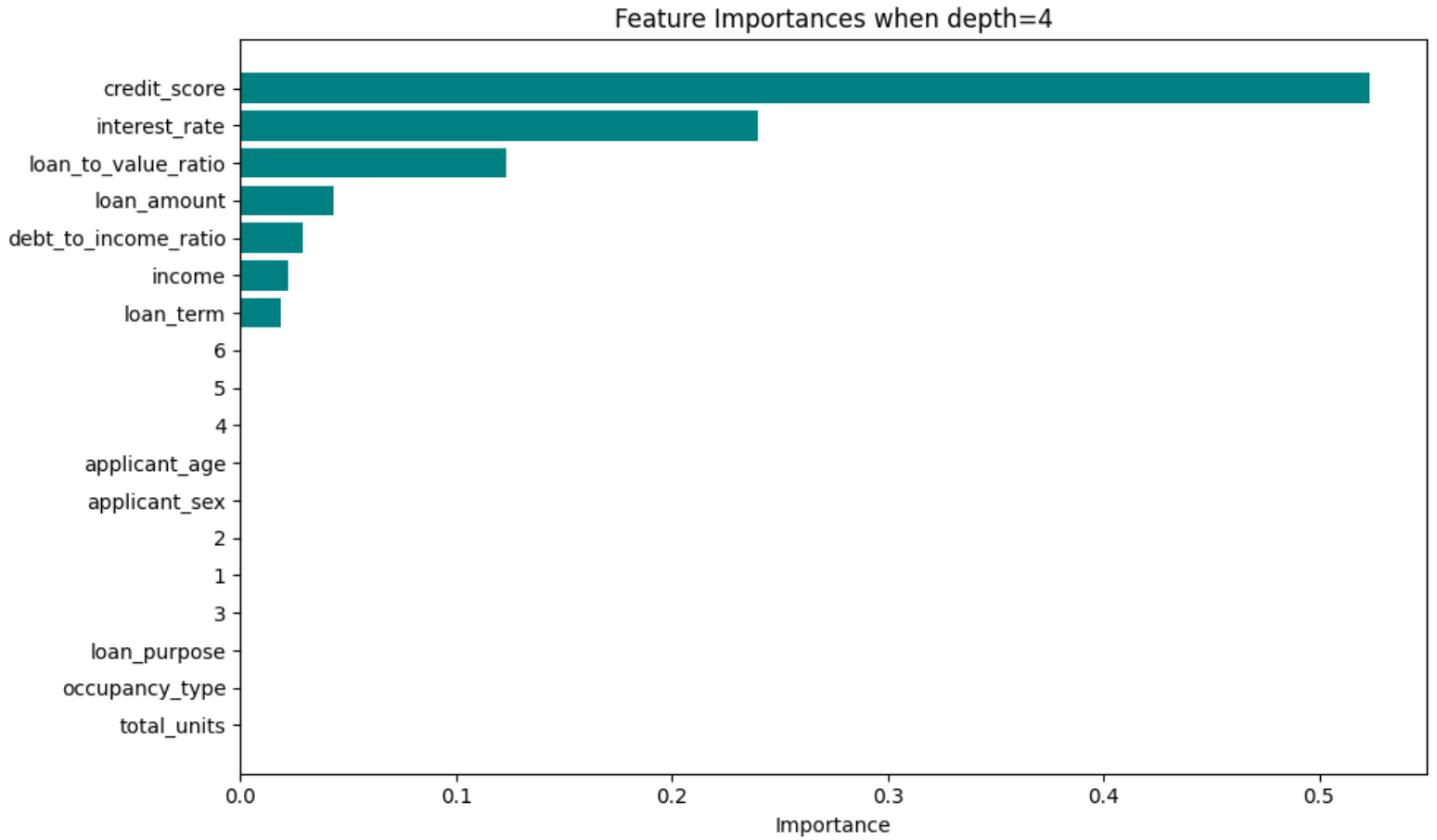
However, the best AUC-PR is 0.2...



# Feature selection with Decision Tree

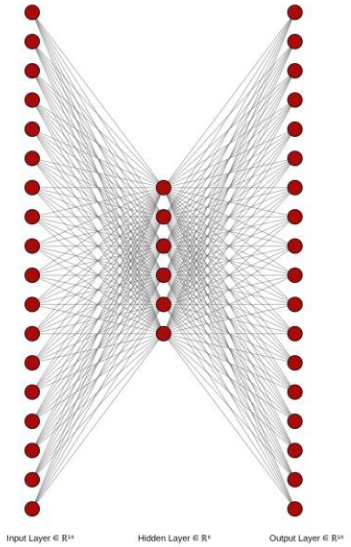
Notice that LASSO, Ridge and Decision Trees seem to agree on the most relevant features

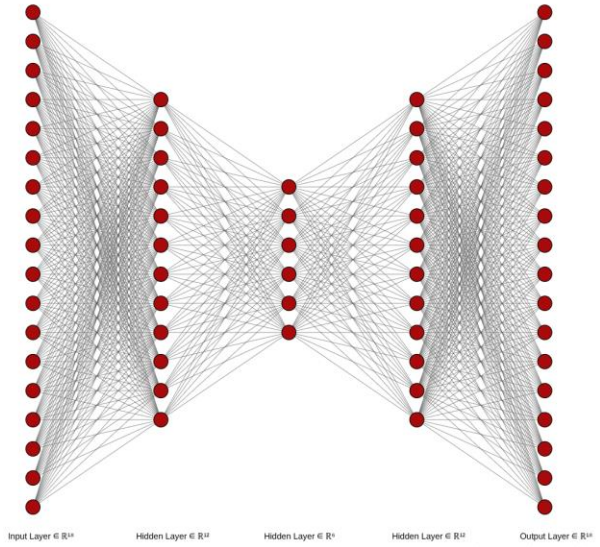
Interestingly, Logistic Regression and Decision Trees with just the first 4 features perform as good as with all features

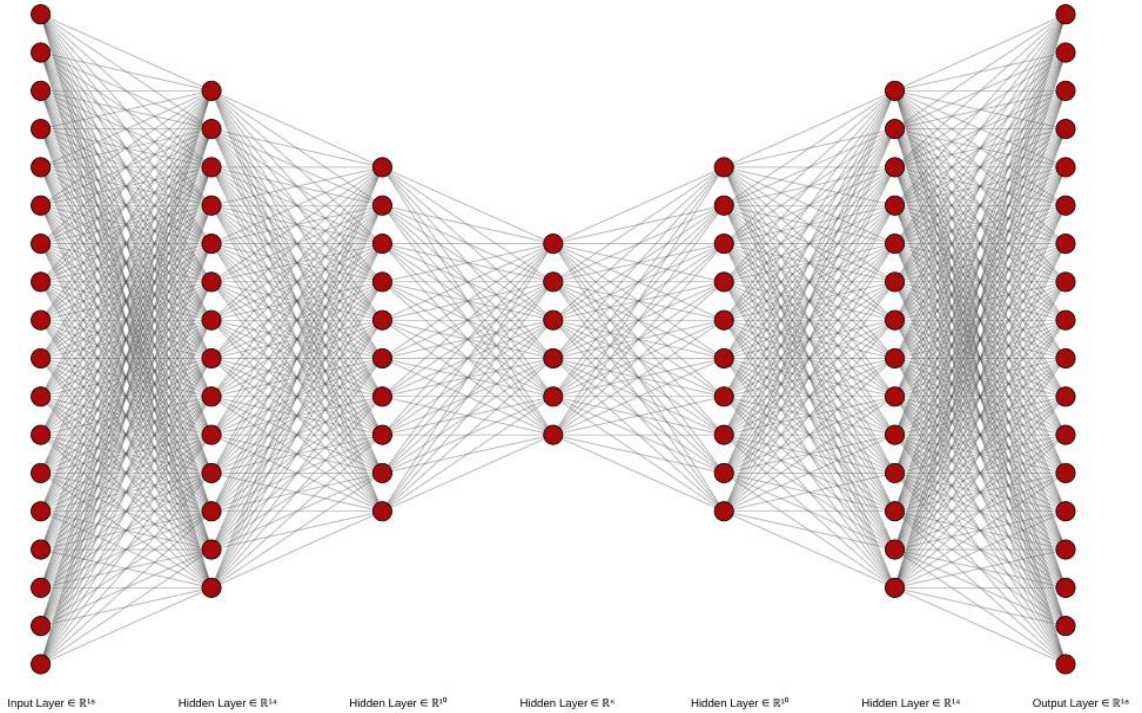


# NN - Logistic

Let's use an autoencoder model for dimensionality reduction, followed by training a logistic regression classifier on the reduced feature space.

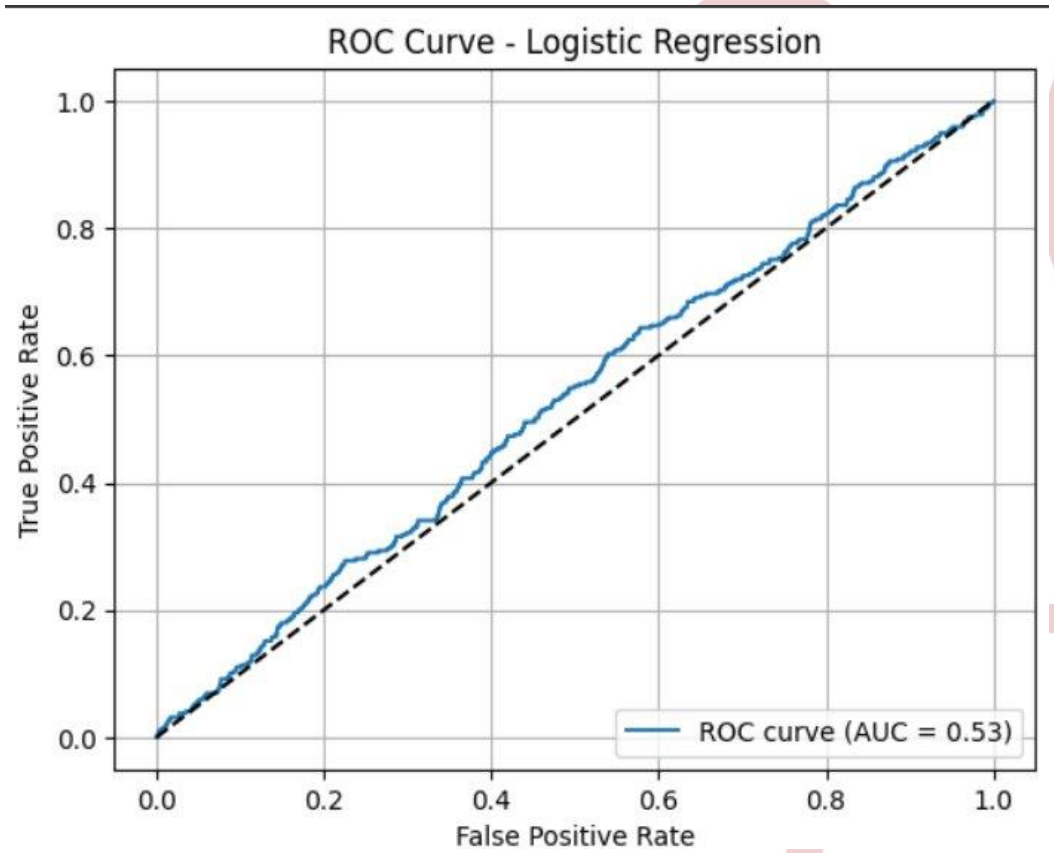


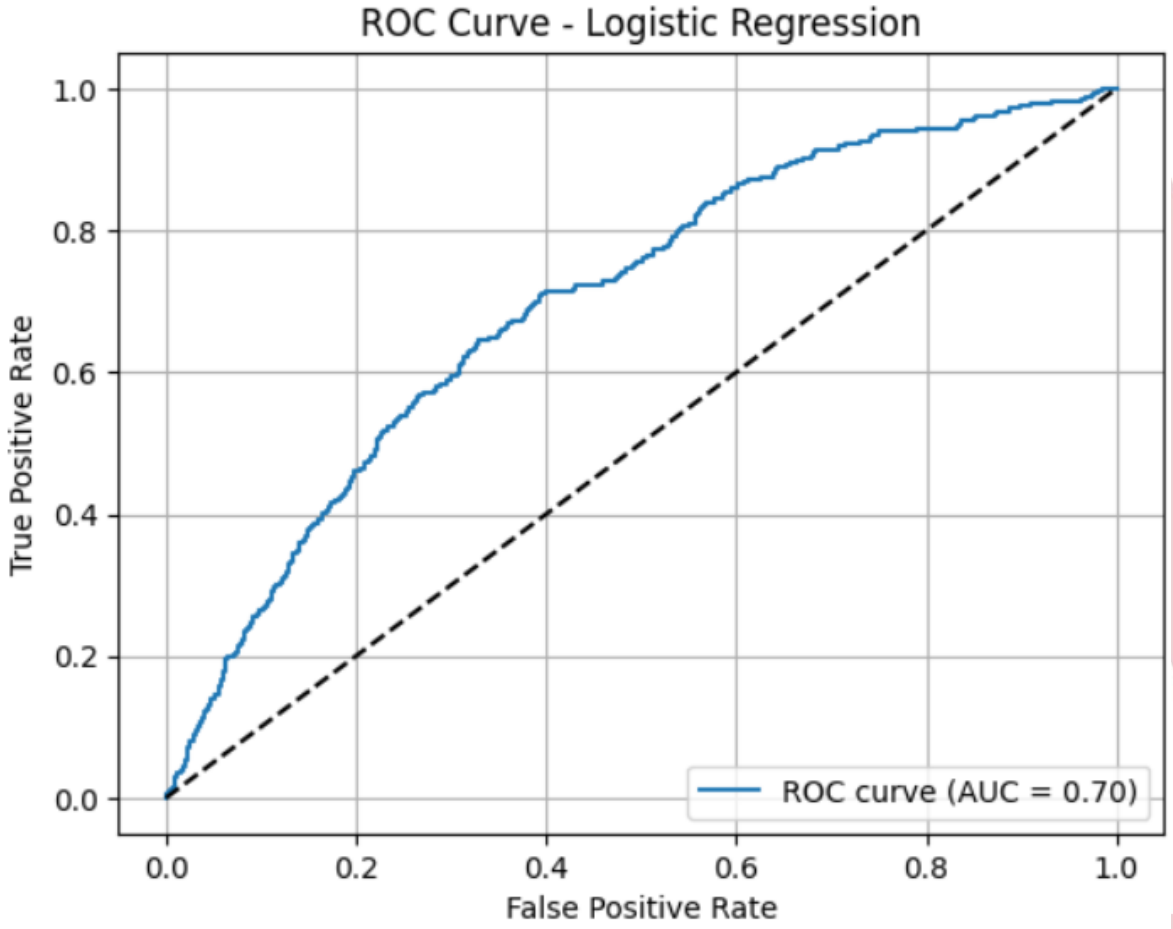
# NN – Logistic 2

```
Confusion Matrix:
 [[7281 9284]
 [  56  158]]
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.44      0.61     16565
           1       0.02      0.74      0.03       214

    accuracy                           0.44     16779
   macro avg       0.50      0.59      0.32     16779
weighted avg       0.98      0.44      0.60     16779
```



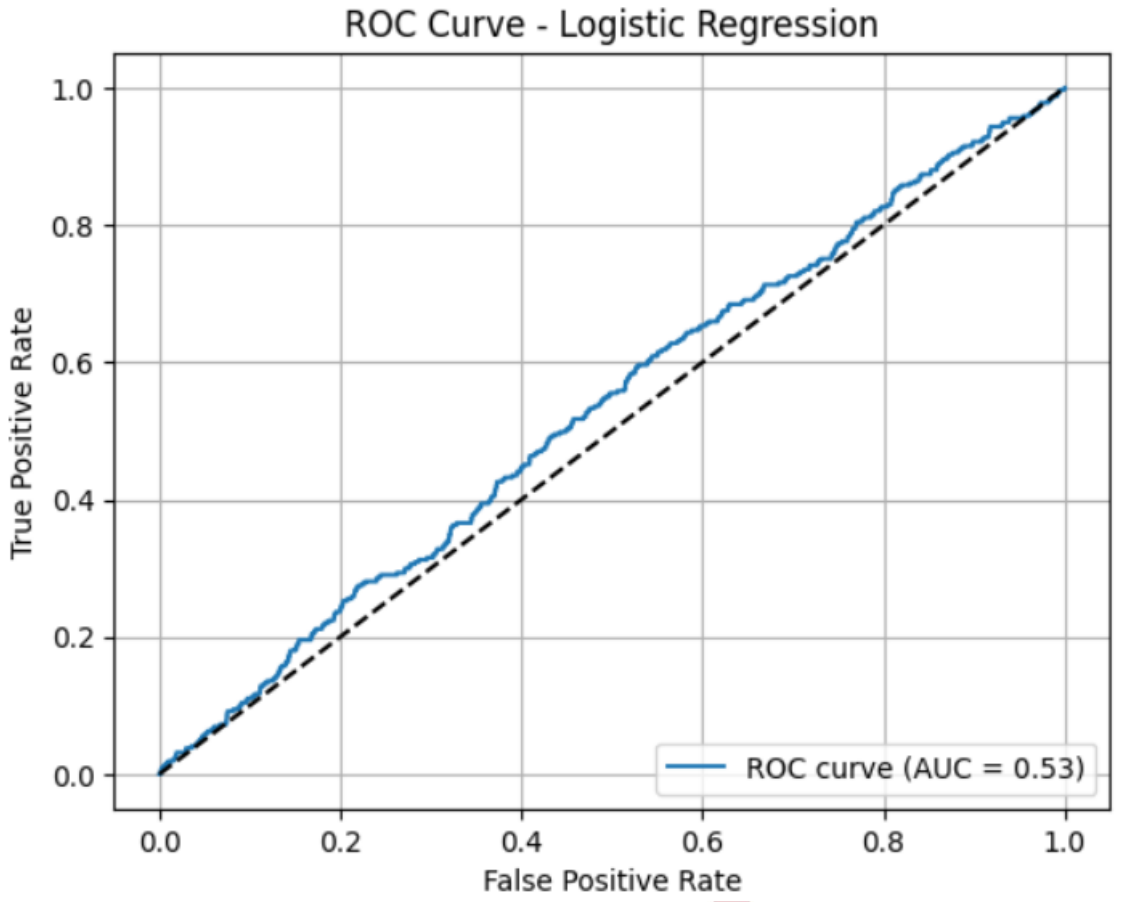
# NN – Logistic 3

```
Confusion Matrix:
 [[14955  9897]
 [   92   225]]
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.60      0.75     24852
           1       0.02      0.71      0.04       317

    accuracy                           0.60     25169
   macro avg       0.51      0.66      0.40     25169
weighted avg       0.98      0.60      0.74     25169
```



# NN – Logistic 4

```
Confusion Matrix:
 [[14308 10544]
 [  167   150]]
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.58      0.73     24852
           1       0.01      0.47      0.03       317

    accuracy                           0.57     25169
   macro avg       0.50      0.52      0.38     25169
weighted avg       0.98      0.57      0.72     25169
```

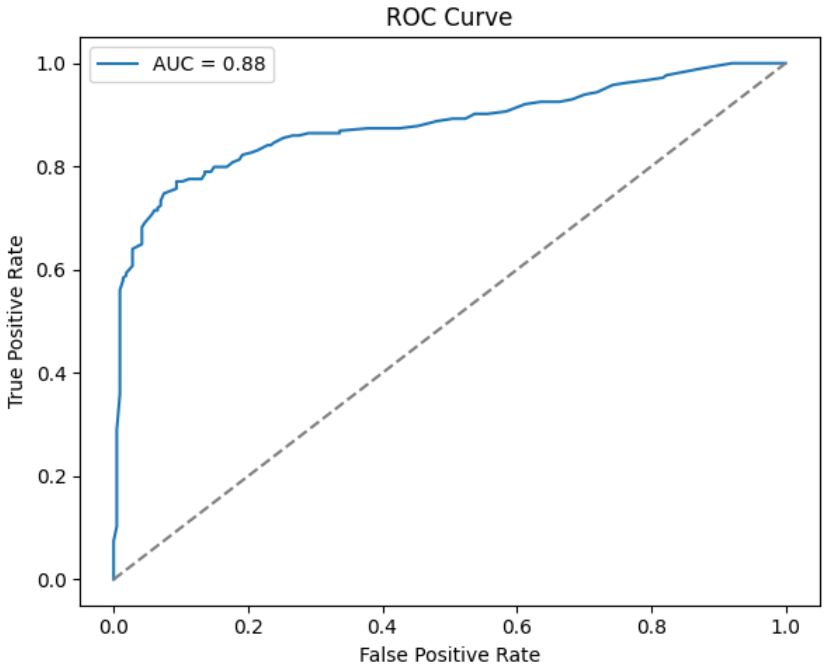


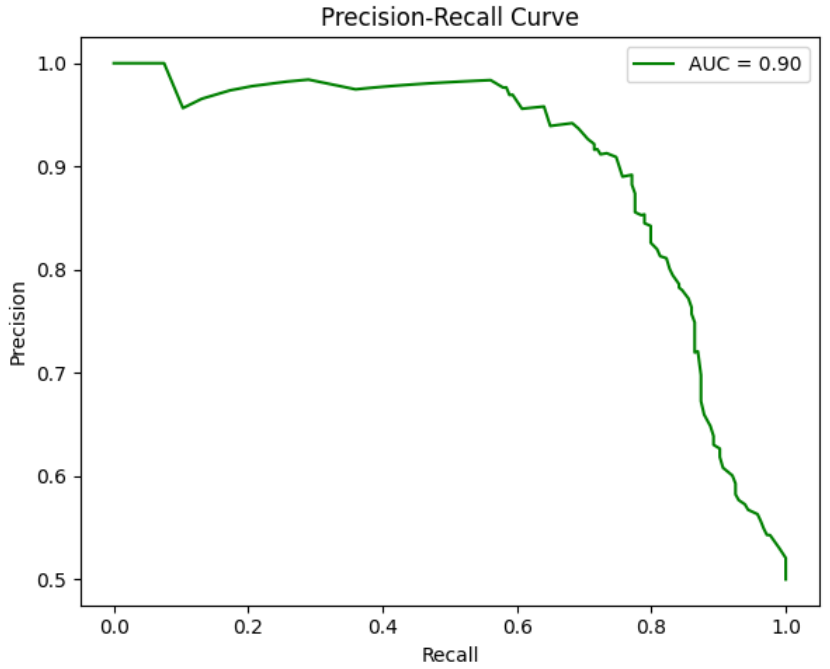
# Undersampling

We experimented with several undersampling methods

The key takeaway is that improvements in recall for the 1s are offset by a dip in recall for the 0s due to information loss

NearMiss2 works like clockwork for the 1s, but has very low specificity



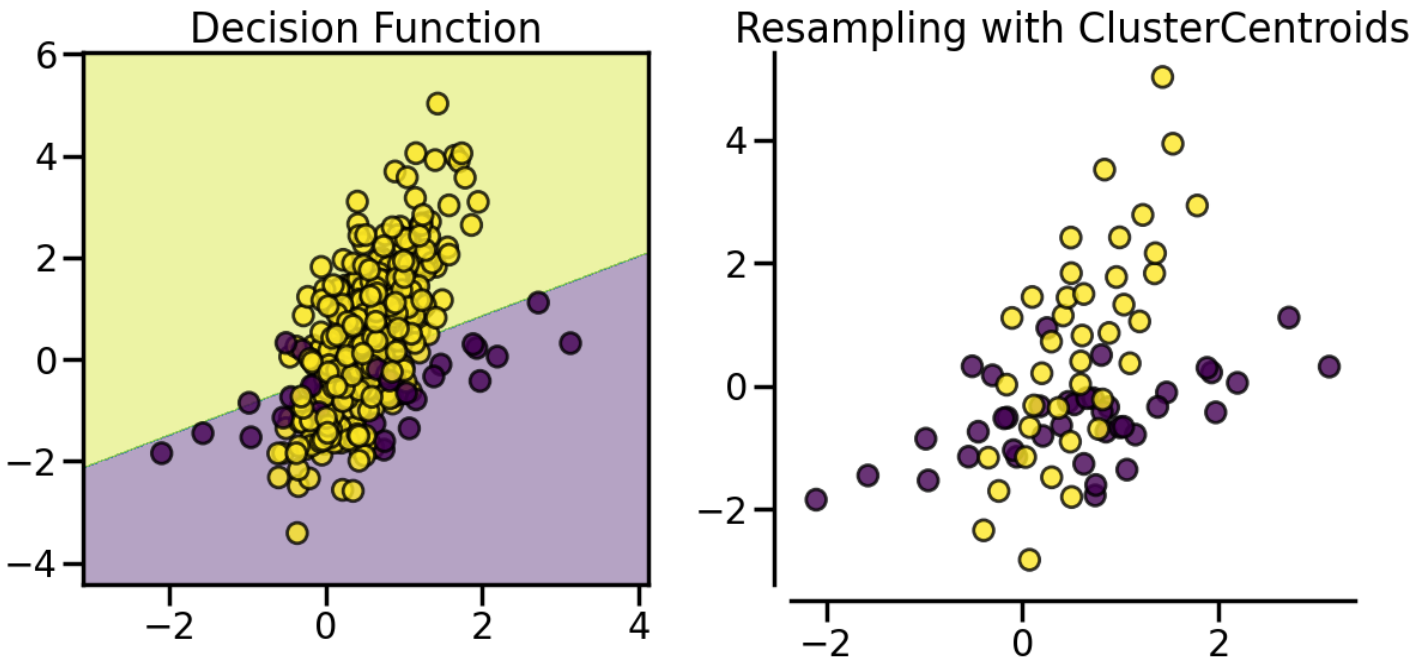

Random Forest with NearMiss2 undersampling

# Our silver bullet: Cluster Centroids

Undersampling with Cluster Centroids yields promising results (Random Forest as classifier):

- AUC=0.99
- AUC-PR=0.99
- For 0s of 86% on the majority class

References: ZHOU (2024), HUANG (2024)

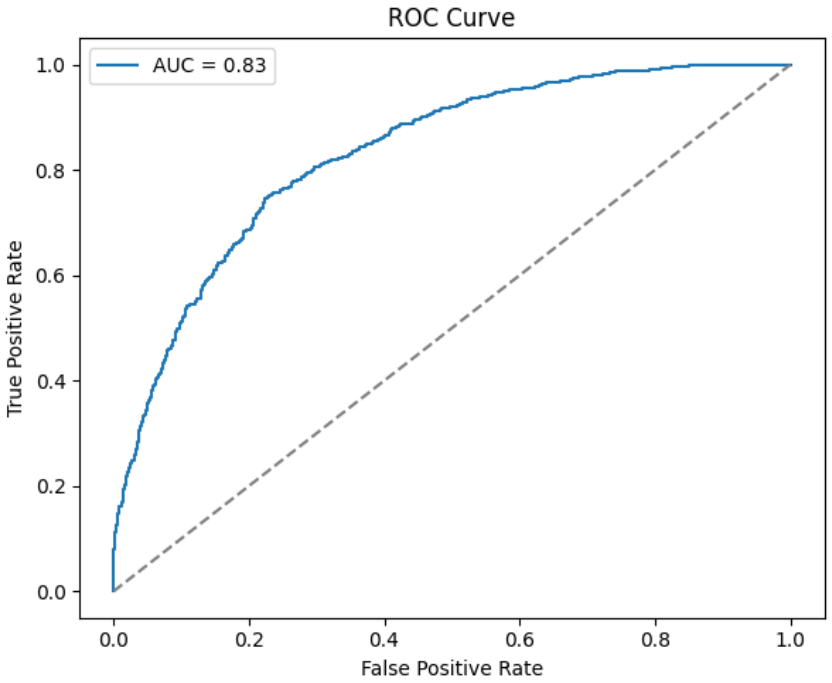


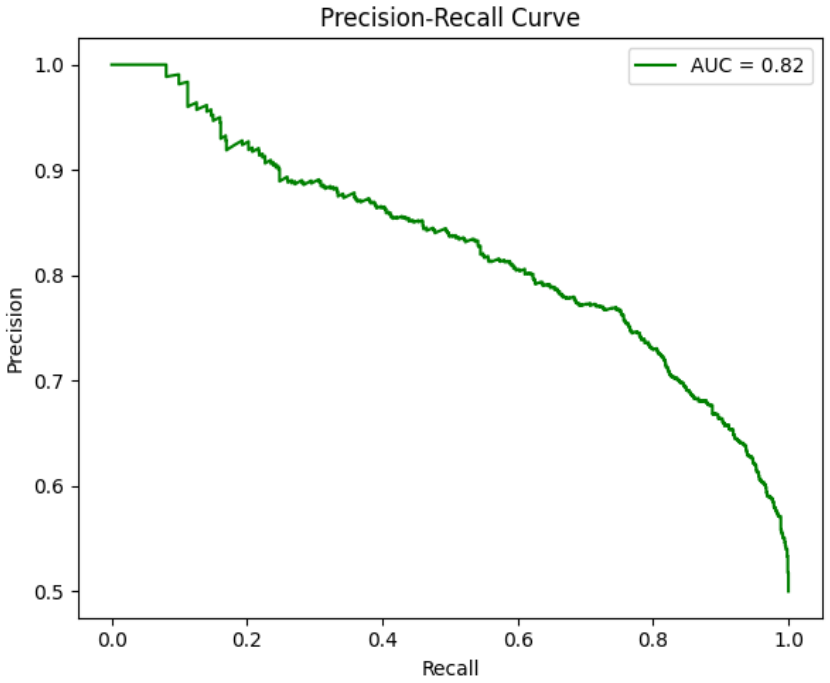

Snapshot of the Cluster Centroid algorithm

# SMOTE

As proposed in CHAWLA (2002), we perform a mix of under- and oversampling

Regarding size, the majority class is shrunk to 5oo% the minority class The latter is then oversampled until classes are perfectly balanced




Random Forest

# Conclusions

- Only a handful of features proved to be informative: CScore, interest rate, loan-to-value ratio and loan amount
- Undersampling generally achieves high recall for defaulted individuals but low specificity
- Nonetheless, undersampling done with Cluster Centroids proved to be outstanding in every measure