Predicting Credit Worthiness

Francesca Furia 3078906
Anna Illiano 3185116
Poorva Seth 3170386

2 papers

Paper #1:

R. Turkson; E.Baagyere; G. Wenya *(2009)*

The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients

Paper #2:

Yeh, I.-C., Lien, C.-H. *(2016)*

A Machine Learning Approach for Predicting Bank Credit Worthiness

6 ML techniques

GOAL: Find the model the best estimates the ***probability*** of default

Evaluates the models using 'Sorting Smoothing Method' to estimate the true probability of defaulting

16 ML techniques

GOAL: Find the model the best classifies ***whether or not*** the client defaults

Variable selection to show no difference outperformance for 5 top models on 5 top variables

1 Dataset

- Both papers use the same dataset, collecting information about credit card clients in Taiwan, between April and October 2005.

- **Source:** Paper #1, published in UCI Machine Learning Data Repository.

- **Response Variable:** Binary variable indicating whether or not the client defaulted in October 2005

- **Explanatory Variables:** Set of 23 variables, including both numerical and categorical ones

AGE	Demographics
SEX	
MARRIAGE	
EDUCATION	
LIMIT_BAL	Amount of the given credit, including individual consumer and their family credit
PUNCTUALITY	6 variables indicating history of past payment in the previous 6 months
BILL_AMT	6 variables indicating the amount of the bill statement in the previous 6 months
PAY_AMT	6 variables indicating the amount of payments in the previous 6 months

Our Approach

Data Cleaning & Exploration

Methodologies

Logistic Regression

K Nearest Neighbours

Bagging

Discriminant Analysis

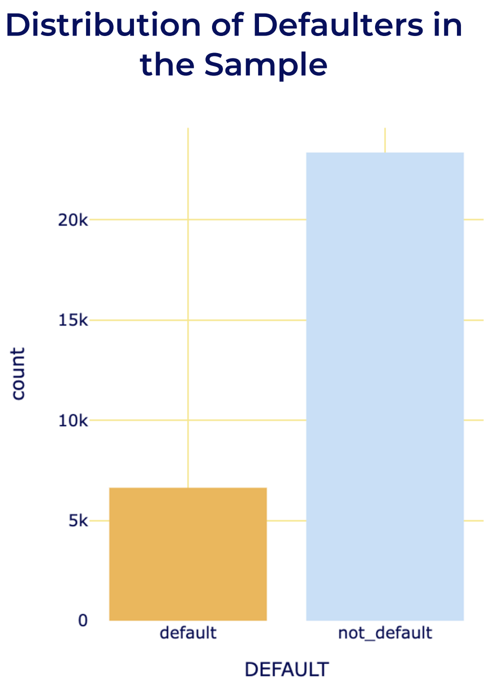
Neural Networks

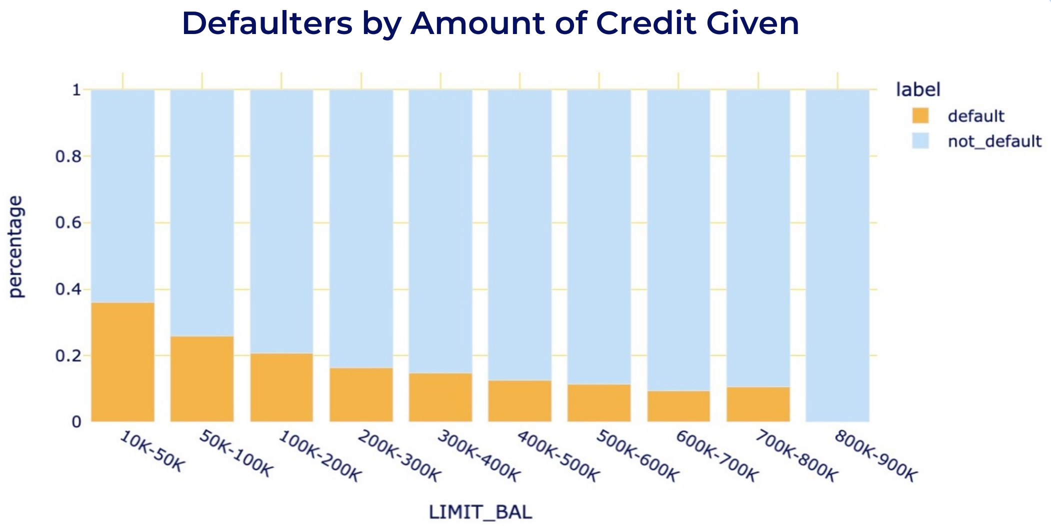
Naive Bayes

VARIABLE SELECTION

Comparison

Data Exploration

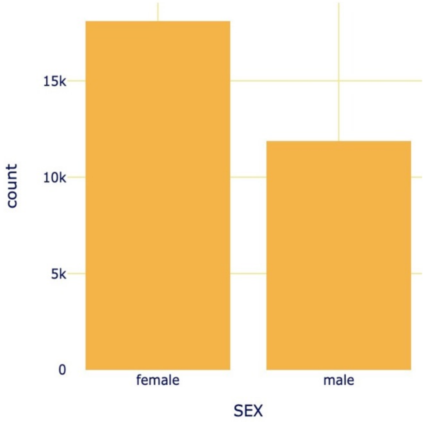




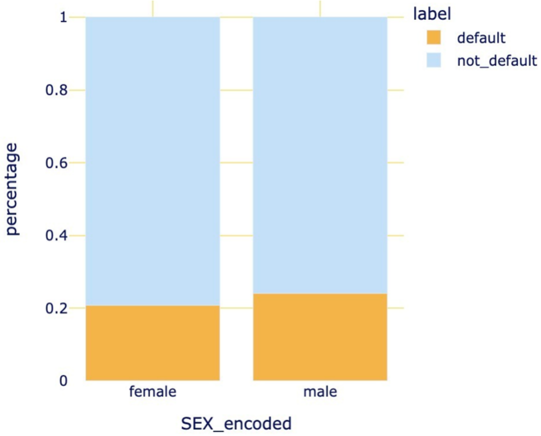
Checking for Biases

- **Gender bias:** 

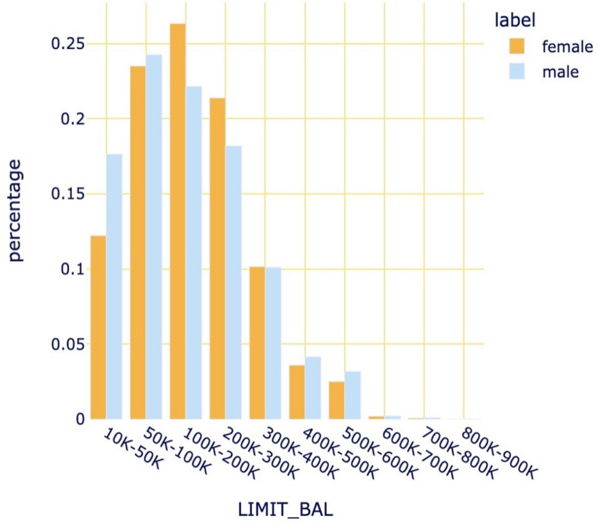
Sex Distribution
in the Sample



Default distribution
by Sex



Limit Balance
by Gender



- **Racial bias:** not applicable in this case

We specifically care about keeping false negatives low,
as it's worse to lend money to people who will actually default,
rather than the other way around.

Hence we primarily use ***recall*** to assess our methods,
as it measures the ability to find all the positive samples.

Recall: $$\frac{True\ Positives}{True\ Positives + False\ Negatives}$$

In order to avoid overfitting on the defaulting class,
we used **macro-averaging**

Sorting Smoothing Method

GOAL: estimate the “true” probability of default

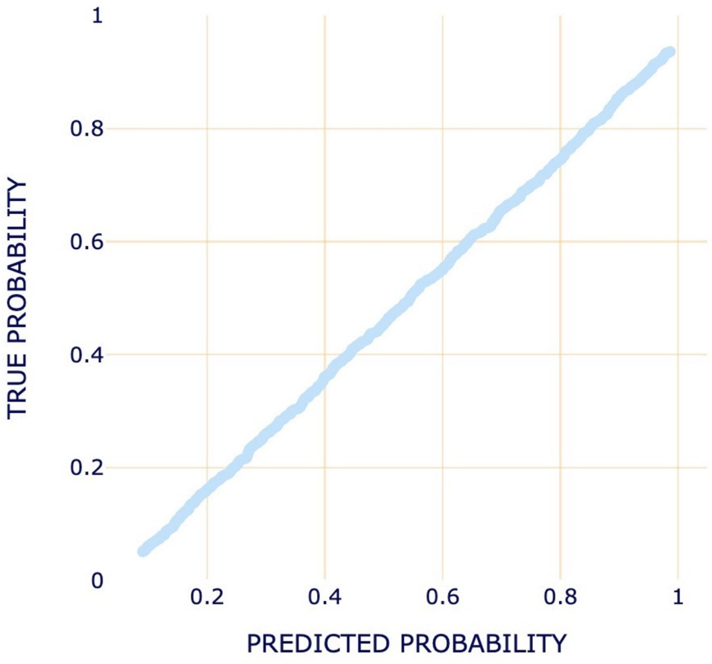
1. Order the predictions by increasing estimated probability of default
2. Compute “true” probability of defaulting as

$$P_i = \frac{Y_{i-n} + \cdots + Y_{i-1} + Y_i + Y_{i+1} + \cdots + Y_{i+n}}{2n+1}$$

   where $Y_i = 1$ if default

3. Evaluate the predicted probabilities from the model:
   - Scatterplot estimated probability VS “true” probability
   - Running a OLS and look at R^2, intercept and slope coefficient

Ideally, we would want...



Sorting Smoothing Method

GOAL: estimate the "true" probability of default

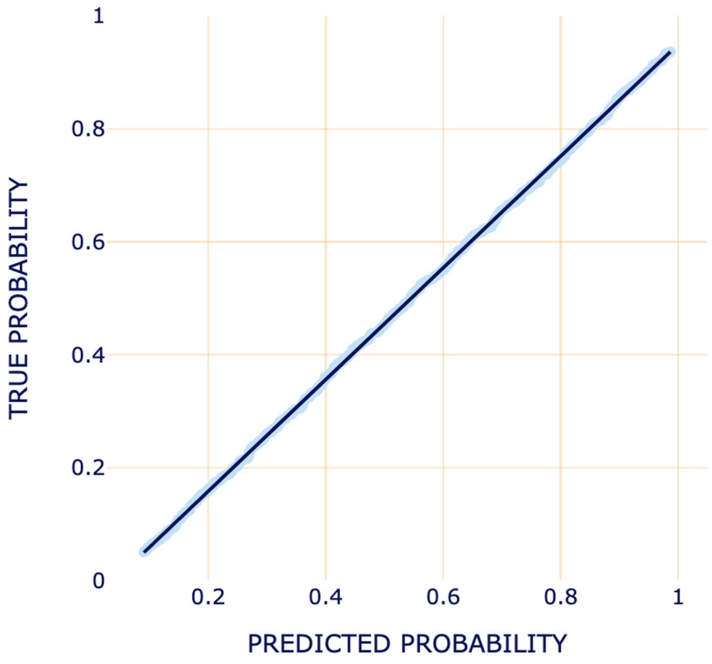
1. Order the predictions by increasing estimated probability of default
2. Compute "true" probability of defaulting as

$$P_i = \frac{Y_{i-n} + \cdots + Y_{i-1} + Y_i + Y_{i+1} + \cdots + Y_{i+n}}{2n+1}$$

   where $Y_i = 1$ if default

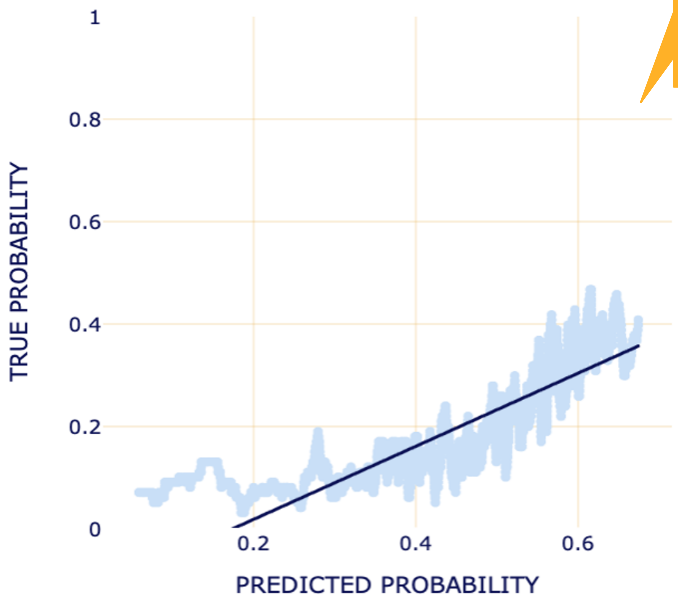
3. Evaluate the predicted probabilities from the model:
   - Scatterplot estimated probability VS "true" probability
   - Running a OLS and look at R^2, intercept and slope coefficient

Ideally, we would want...



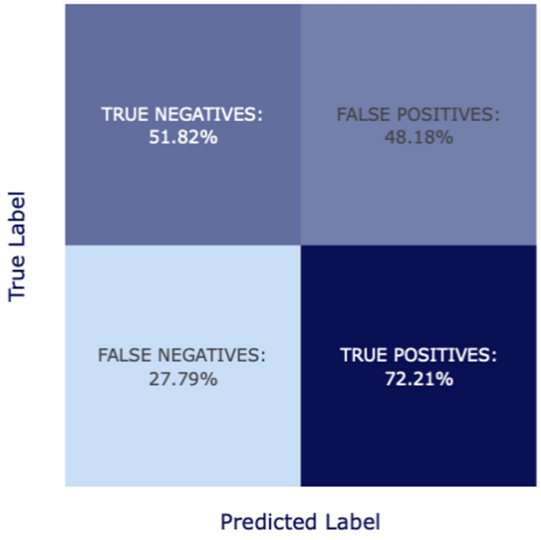
Logistic Regression

SSM Scatter Plot + OLS



R^2: 0.697
Alpha: -0.125
Beta: 0.713

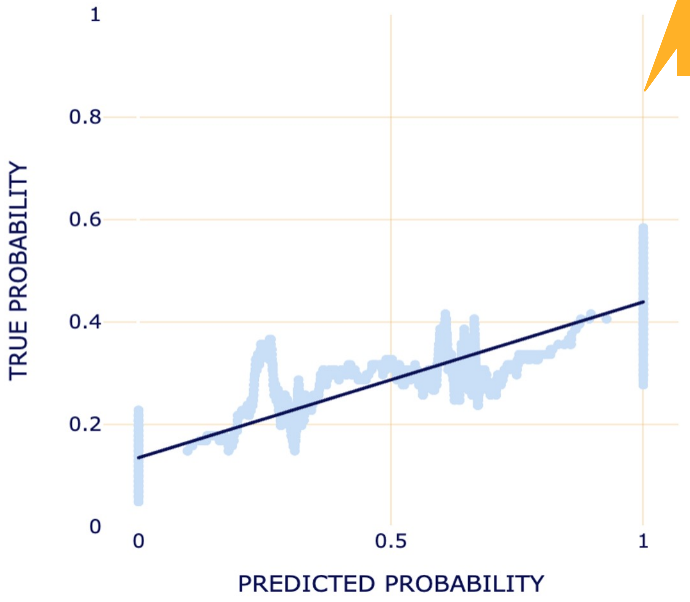
Confusion Matrix
(normalized by true label)



macro-recall: 0.62
macro-precision: 0.58
accuracy: 0.56

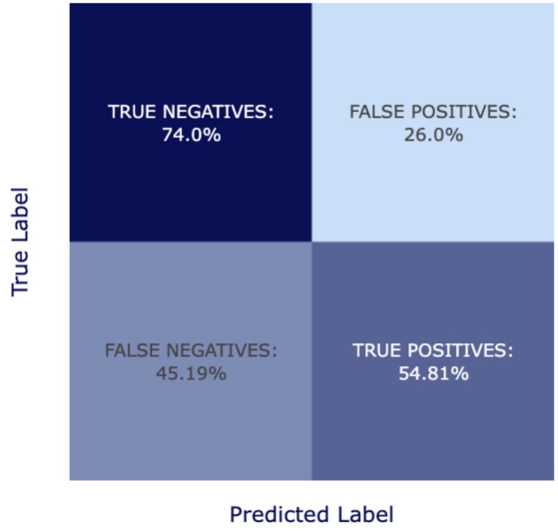
K Nearest Neighbors

SSM Scatter Plot + OLS



R^2: 0.890
Alpha: 0.135
Beta: 0.304

Confusion Matrix
(normalized by true label)

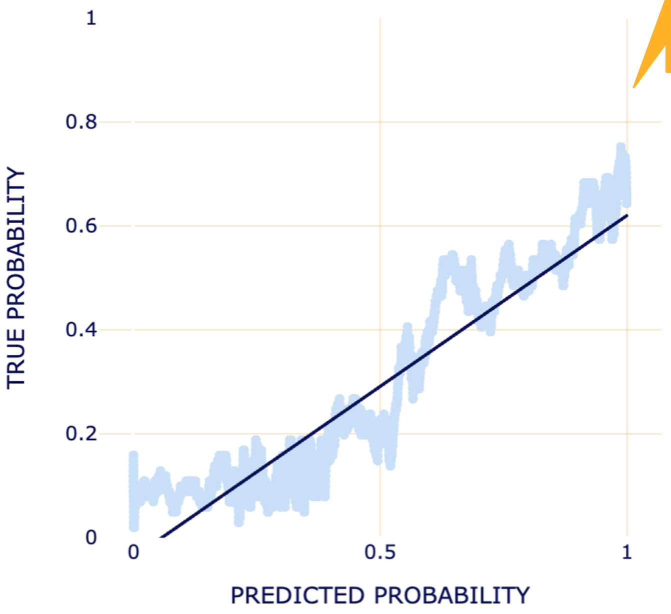


macro-recall: 0.64
macro-precision: 0.62
accuracy: 0.70

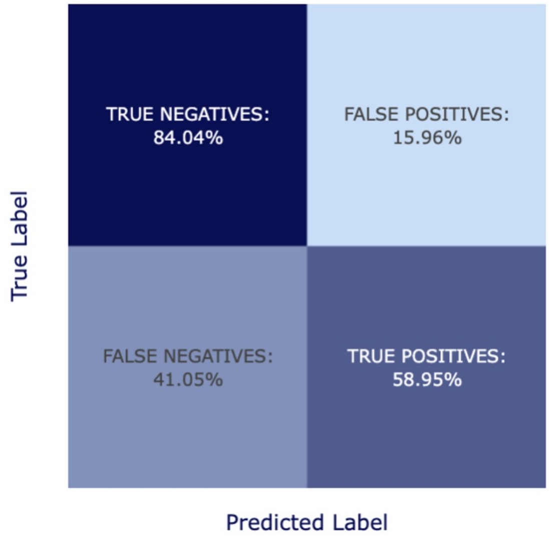
Quadratic Discriminant Analysis

SSM Scatter Plot + OLS

R^2: 0.859
Alpha: -0.038
Beta: 0.658



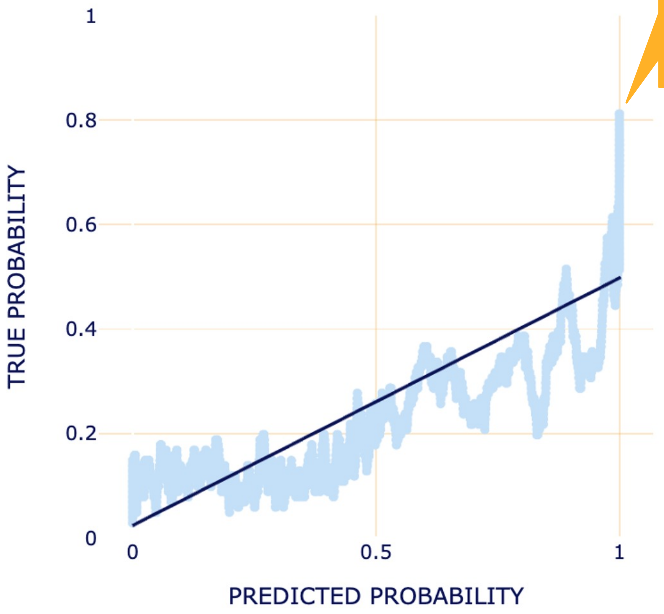
Confusion Matrix
(normalized by true label)



macro-recall: 0.71
macro-precision: 0.70
accuracy: 0.78

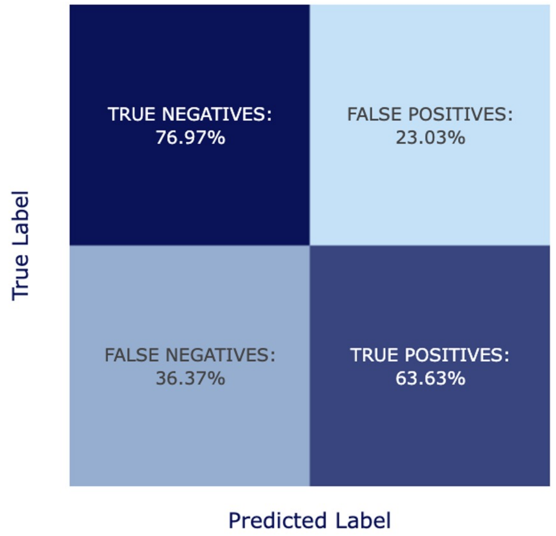
Naive Bayes

SSM Scatter Plot + OLS



R^2: 0.765
Alpha: 0.024
Beta: 0.474

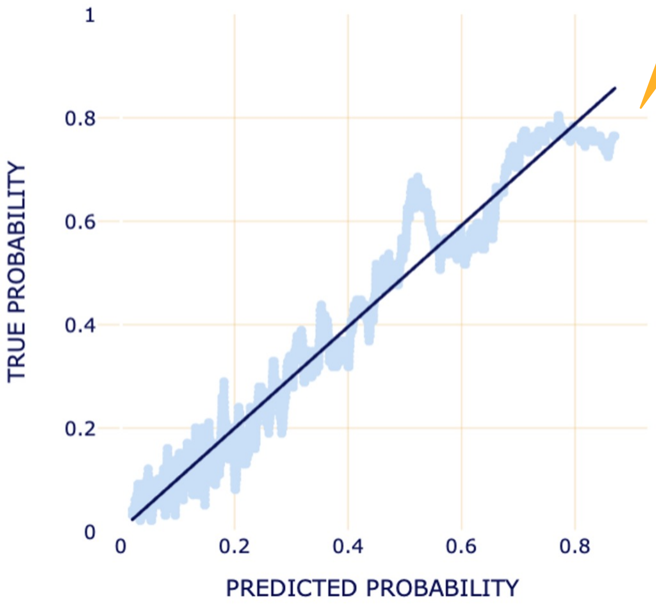
Confusion Matrix
(normalized by true label)



macro-recall: 0.70
macro-precision: 0.66
accuracy: 0.74

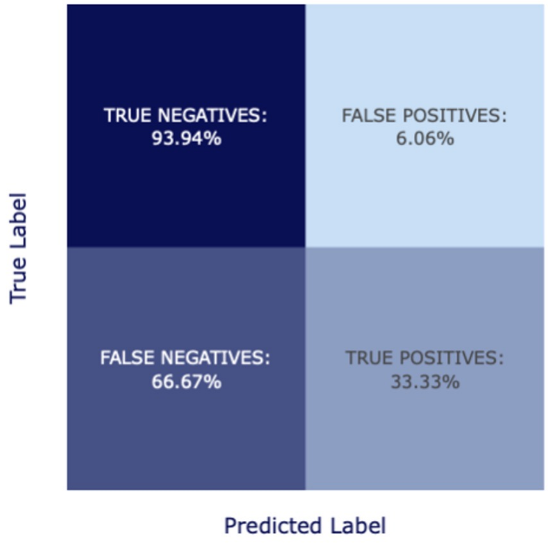
Bagging

SSM Scatter Plot + OLS



R^2: 0.945
Alpha: 0.025
Beta: 0.862

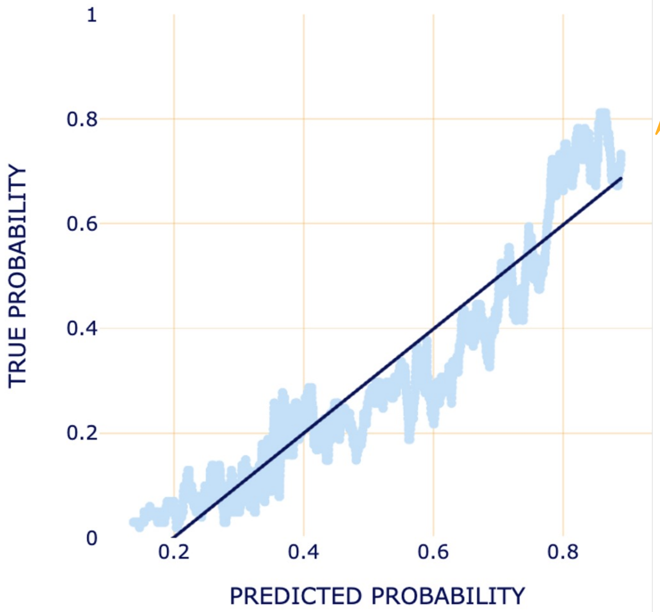
Confusion Matrix
(normalized by true label)



macro-recall: 0.64
macro-precision: 0.72
accuracy: 0.81

Neural Network

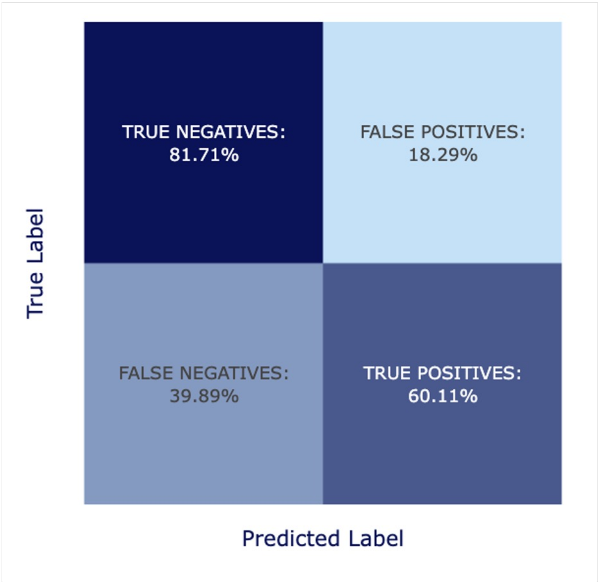
SSM Scatter Plot + OLS



R^2: 0.908
Alpha: -0.198
Beta: 0.9951

Confusion Matrix

(normalized by true label)

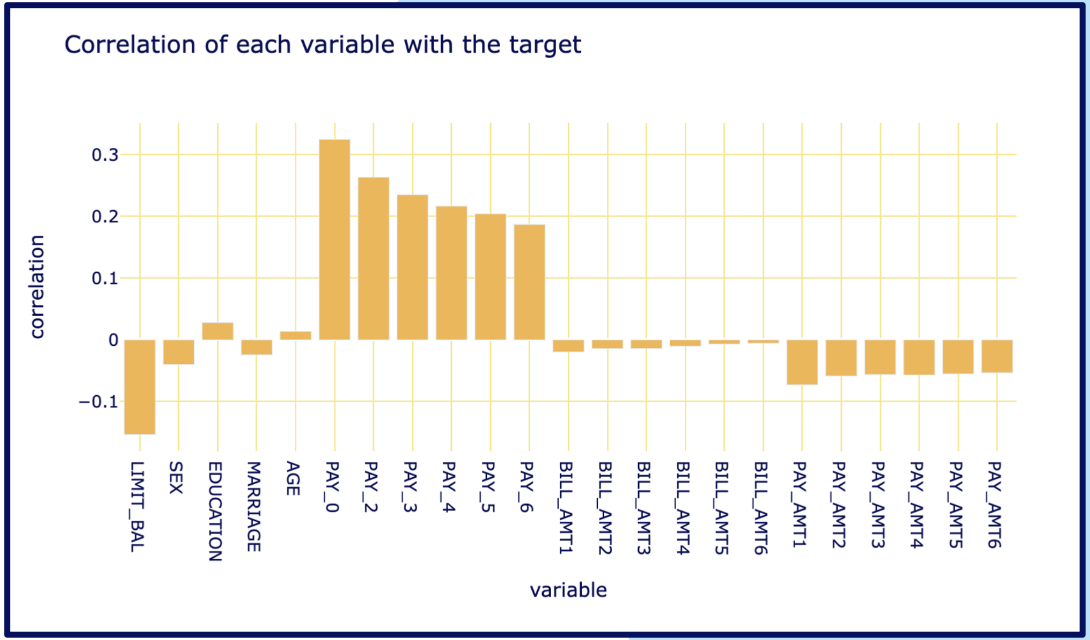


macro-recall: 0.71
macro-precision: 0.68
accuracy: 0.77

Variable Selection

Based on:

- Correlation of each predictor with the response variable
- P-value of FPR test
- SelectKBest algorithm



Chosen Variables:

- LIMIT_BAL
- SEX
- EDUCATION
- PUNCTUALITY_AVG

After Variable Selection

Logistic Regression

⇩ -19% Recall: 0.50
⇧ +39% Accuracy: 0.78
⇩ -54% R^2: 0.32

KNN

⇩ -4% Recall: 0.64
⇧ +4% Accuracy: 0.73
⇩ -15% R^2: 0.744

QDA

⇩ -17% Recall: 0.59
⇧ +3% Accuracy: 0.80
⇧ +4% R^2: 0.813

Naive Bayes

⇩ -9% Recall: 0.61
⇧ +14% Accuracy: 0.80
⇧ +12% R^2: 0.858

Bagging

⇩ -22% Recall: 0.63
= 0% Accuracy: 0.81
⇩ -0.5% R^2: 0.944

Neural Network

⇩ -2% Recall: 0.69
= 0% Accuracy: 0.77
⇩ -2% R^2: 0.887

Best Performing Models

R^2

1.	Bagging	0.945
2.	Neural Network	0.908
3.	KNN	0.890

Macro-Recall

1.	Neural Network & QDA	0.71
2.	Naive Bayes	0.70

Other possible strategies:

- **Changing the evaluation metric** to AUC or accuracy
- **Varying the decision threshold**

Thank you!

Do you have any questions?

Francesca Furia
Anna Illiano
Poorva Seth