DEFAULT OF CREDIT CARD CLIENTS

AMIT ROVSHITZ AND MATAN ADAR

Our Team

MATAN ADAR

AMIT ROVSHITZ

PROCESS DESCRIPTION

We chose to predict whether a client will default on their credit card payment by a machine learning tool

We chose to use 5 machine learning algorithms:

- Knn
- Logistic Regression
- PCA
- Random Forest
- Adaboost

Our dataset is the file call “default of credit card clients.csv” (the link to download the dataset can be found in the file call “description”)

OUR PROBLEM

- PREDICT WHETHER A CLIENT WILL DEFAULT ON THEIR CREDIT CARD PAYMENT
- FEATURES OF CREDIT CARD CLIENT
- MITIGATE POTENTIAL LOSSES
- EVALUATION: F1,ACCURACY,PRECISION,RECALL

DATA DESCRIPTION

- 30000 INSTANCES 25 FEATURES

- PERSONAL DATA:
  AGE,GENDER,EDUCATION, MARITAL STATUS

- FINANCIAL DATA:
  HISTORY OF PAST PAYMENT/DEBT
  LIMIT BALANCE

DATA DESCRIPTION

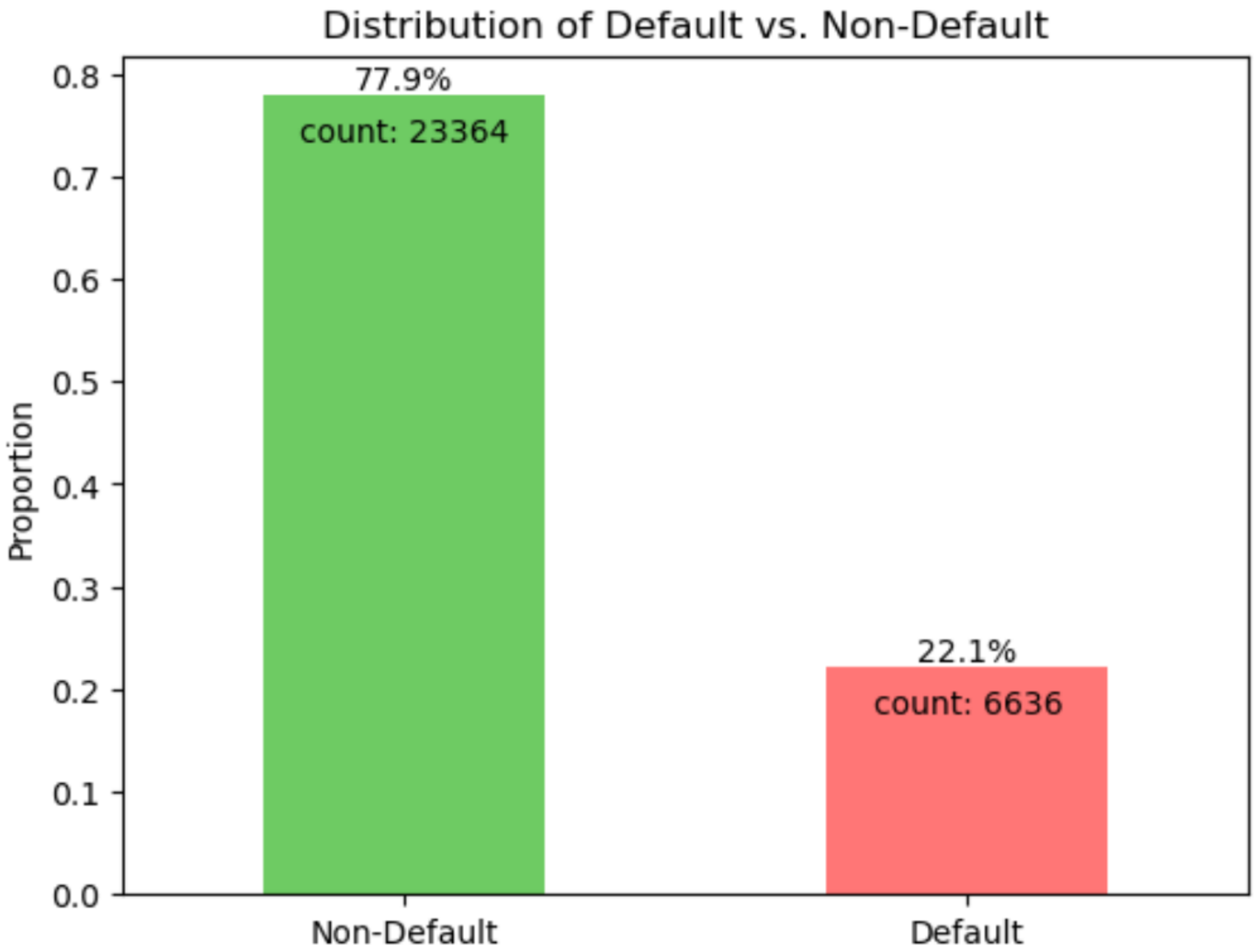
	אפריל	מאי	יוני	יולי	אוגוסט	ספטמבר
Bill_Amt	100	200	1000	400	500	1000
Pay_Amt	0	100	200	1000	0	0
Pay_status	-1	-1	-1	1	2	3

DATA DESCRIPTION

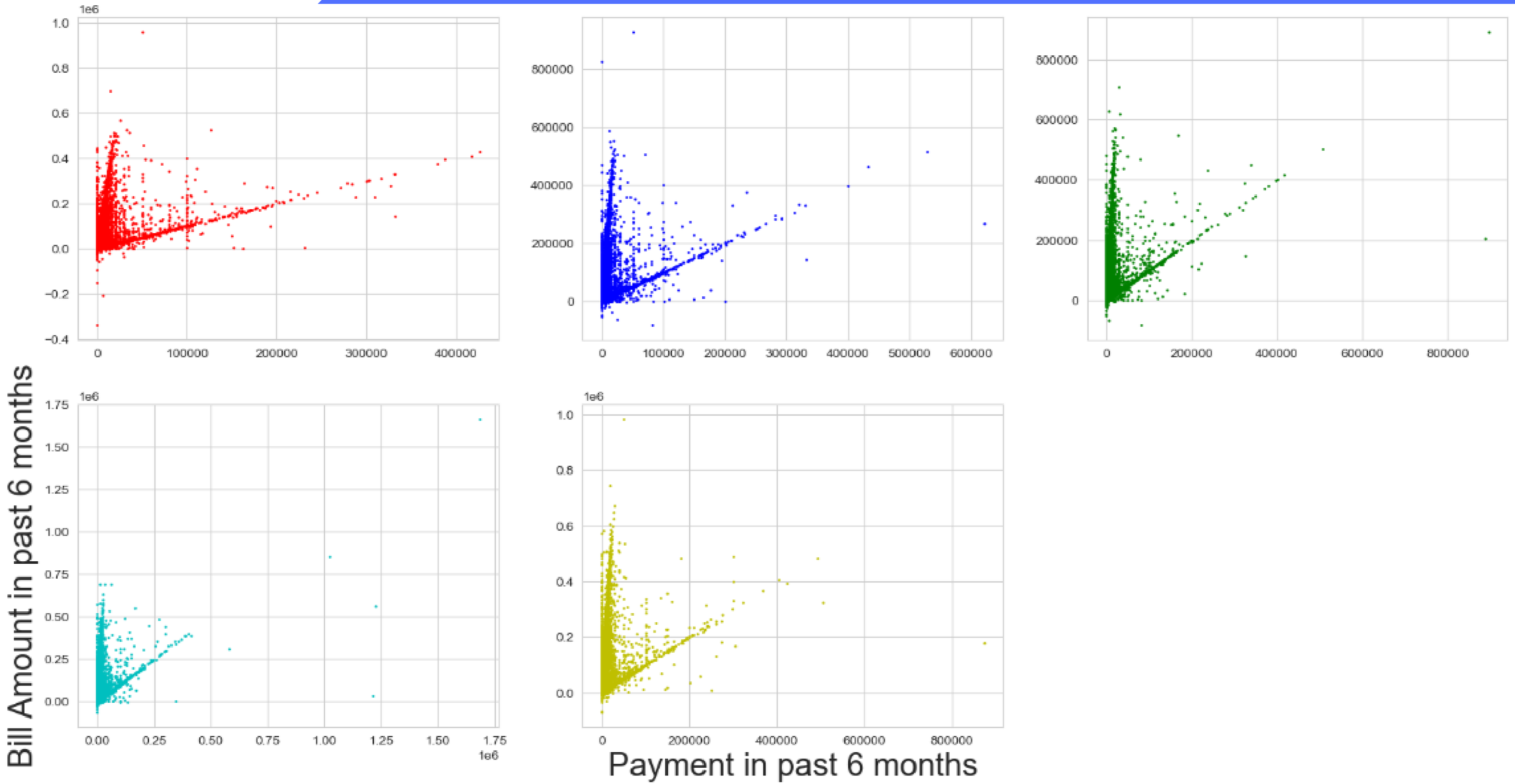
- NA'S: MARRIAGE- 0.18% NA, EDUCATION - 0.05% NA

- 30% TEST /70% TRAIN

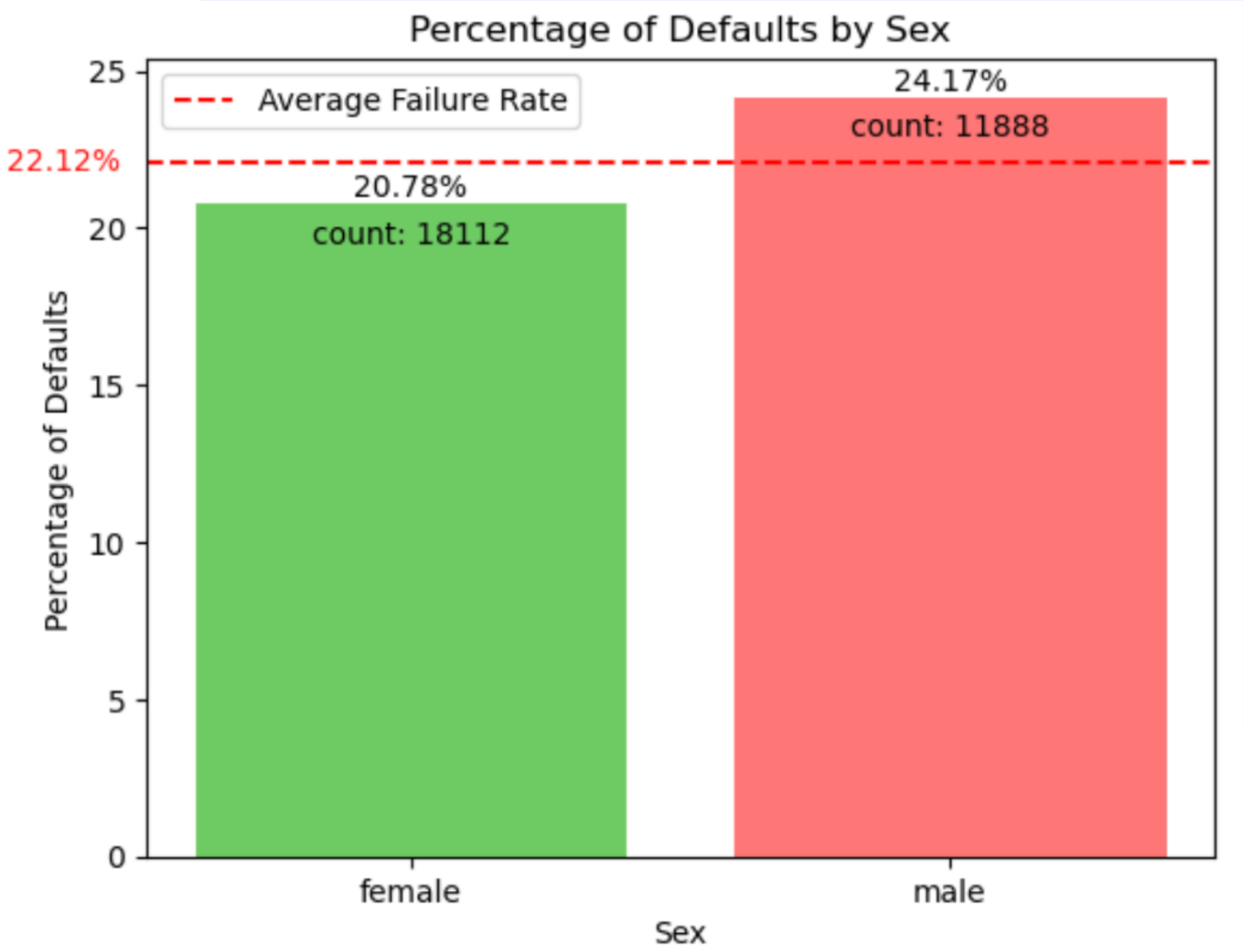
DATA DISTRIBUTION



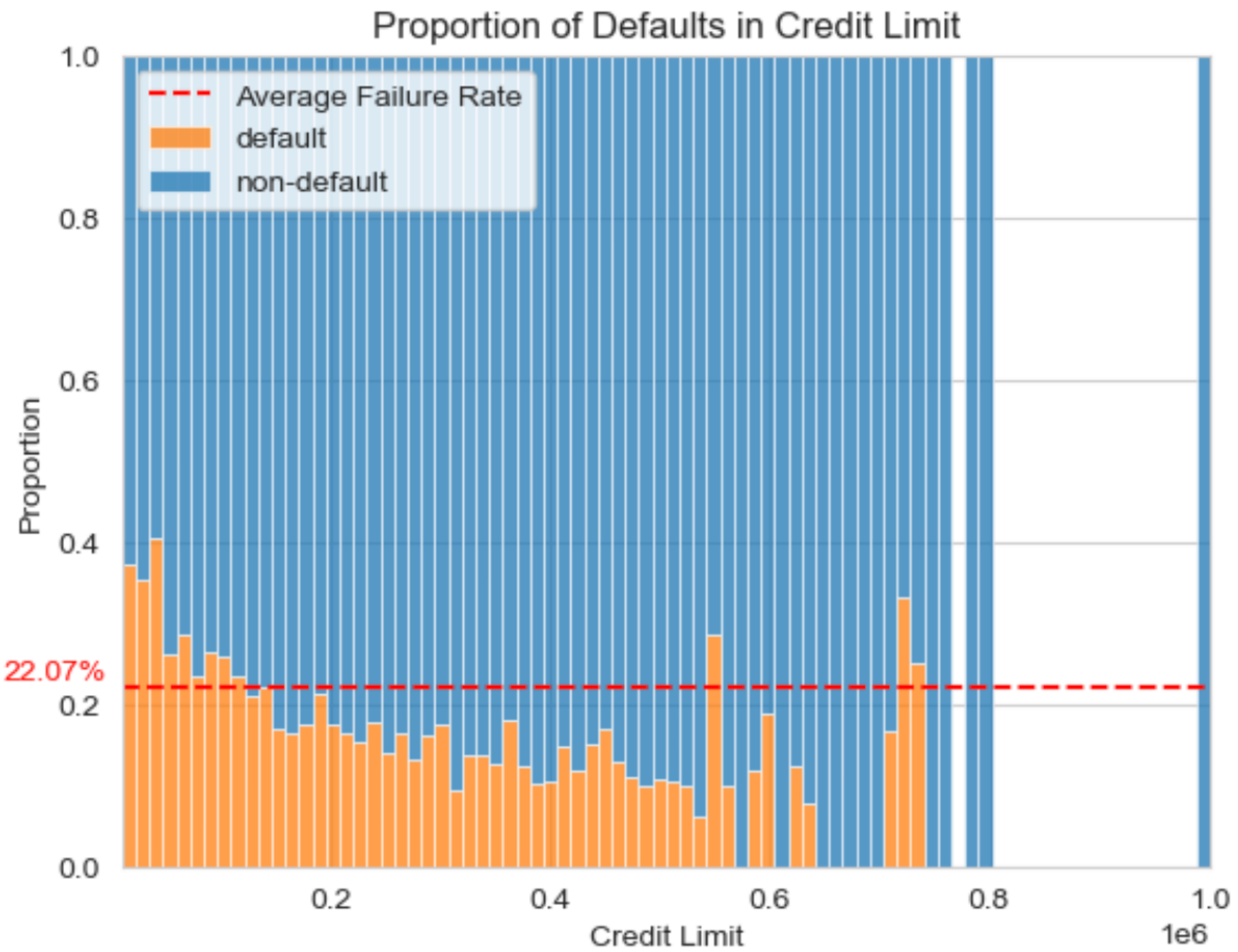
BILL vs. PAY


Bill Amount in past 6 months
Payment in past 6 months
1e6
1e6
1e6
1e6

DEFAULTERS BY GENDER



CREDIT LIMIT DEFAULTERS



DATA ENGINEERING

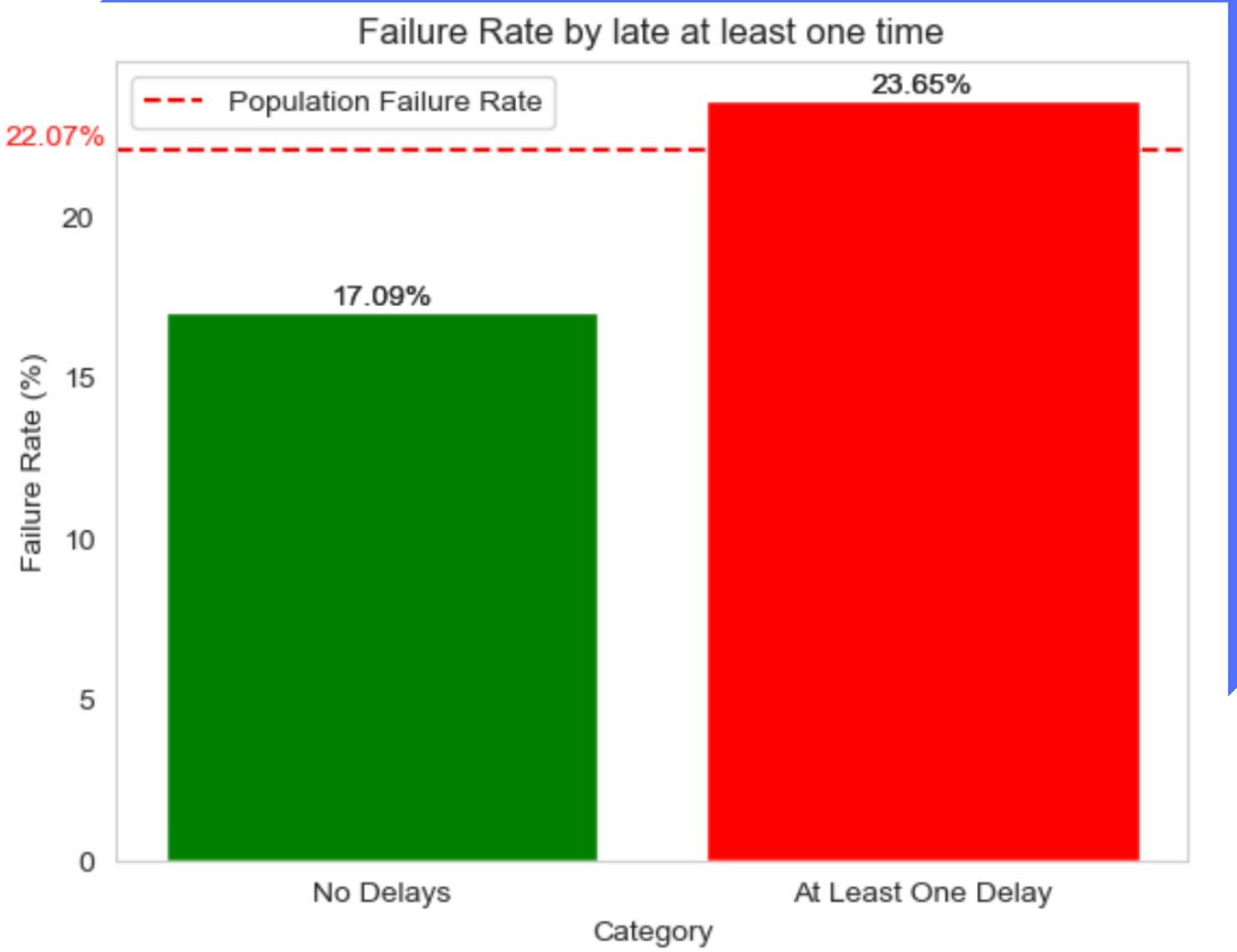
MISSING VALUES:

- NA's: Marriage- 0.18% NA,
  Education - 0.05% NA
- 3 Different Metrics: Remove, Knn, Binning

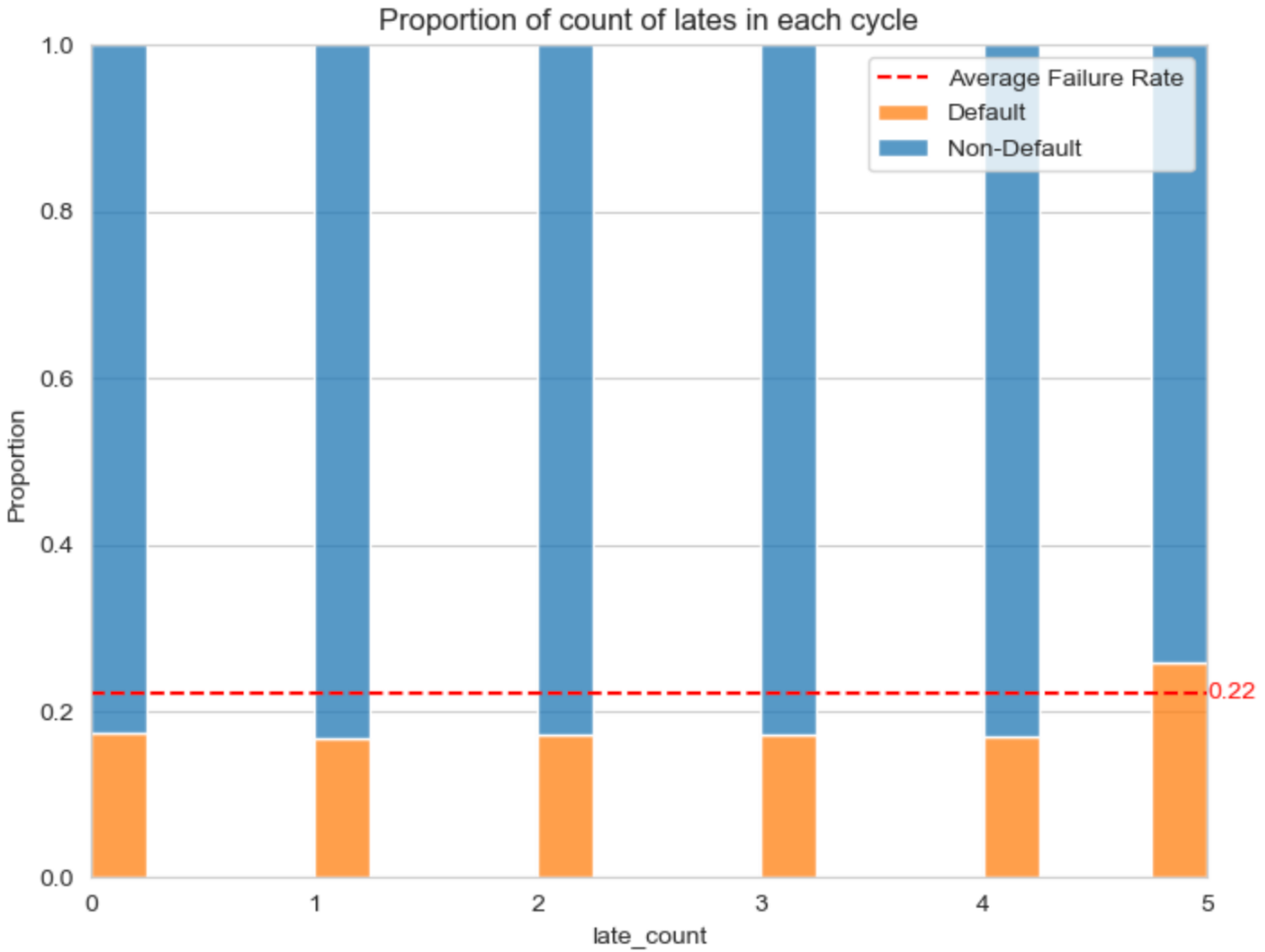
FEATURE ENGINEERING:

- 8 new features
- Late payment (yes/no, count)
- Std of bill/payment

DEFAULT RATE FOR LATE CUSTOMERS



PROPORTION OF LATE COUNT



BASELINE MODEL

- **BASELINE MODEL**
- **MAJORITY OF NON DEFAULT INSTANCES:**
  - Training Accuracy: 0.77
  - Test Accuracy: 0.78

MODELING

- Knn
- Logistic Regression
- Random Forest
- Ada-Boost
- SVM
- PCA

MODELING

Knn

Assigns a class or value to a new data point based on the majority class or average of its nearest neighbors in feature space, using a predefined number of neighbors (k).

MODELING

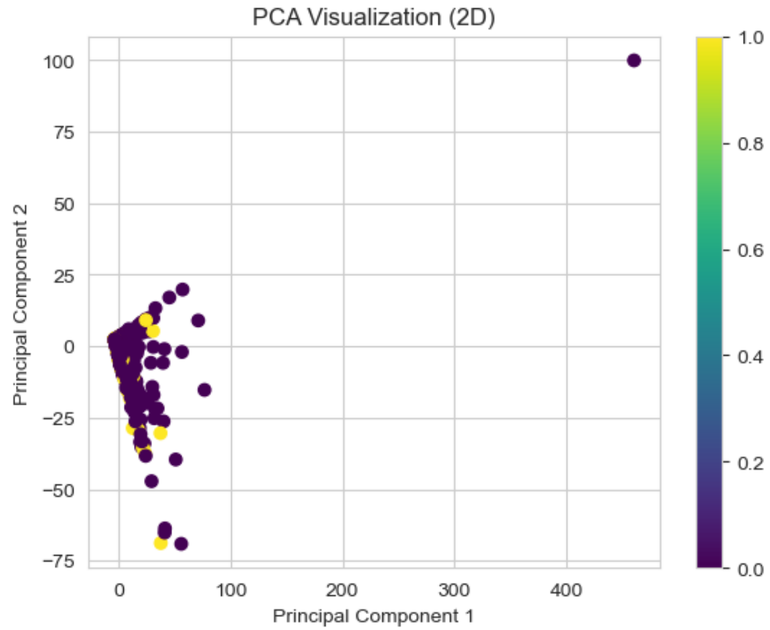
Logistic regression

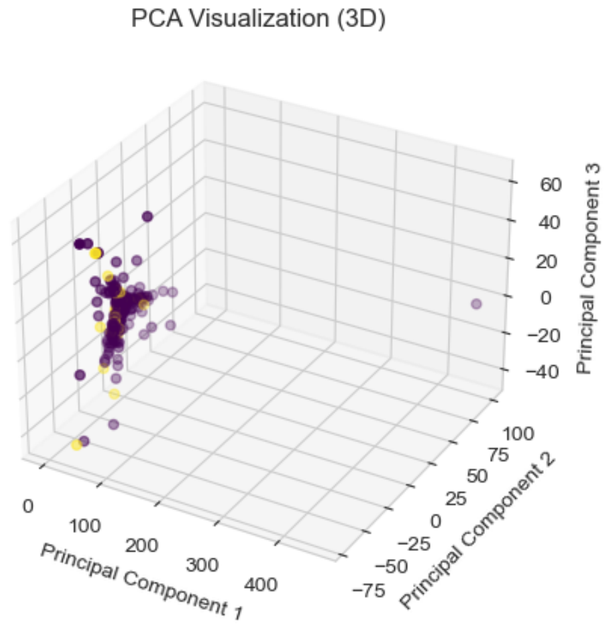
A statistical method used for binary classification tasks, which predicts the probability that an instance belongs to a certain class.

MODELING

PCA

Dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while preserving the variance in the data, aiming to identify and represent patterns in the data efficiently.





MODELING

Random Forest

Group decision-making process where many decision trees (like different experts) vote on a prediction for each input, and the most popular choice wins. This helps to make accurate predictions while reducing the risk of making decisions based on just one tree's biases or errors.

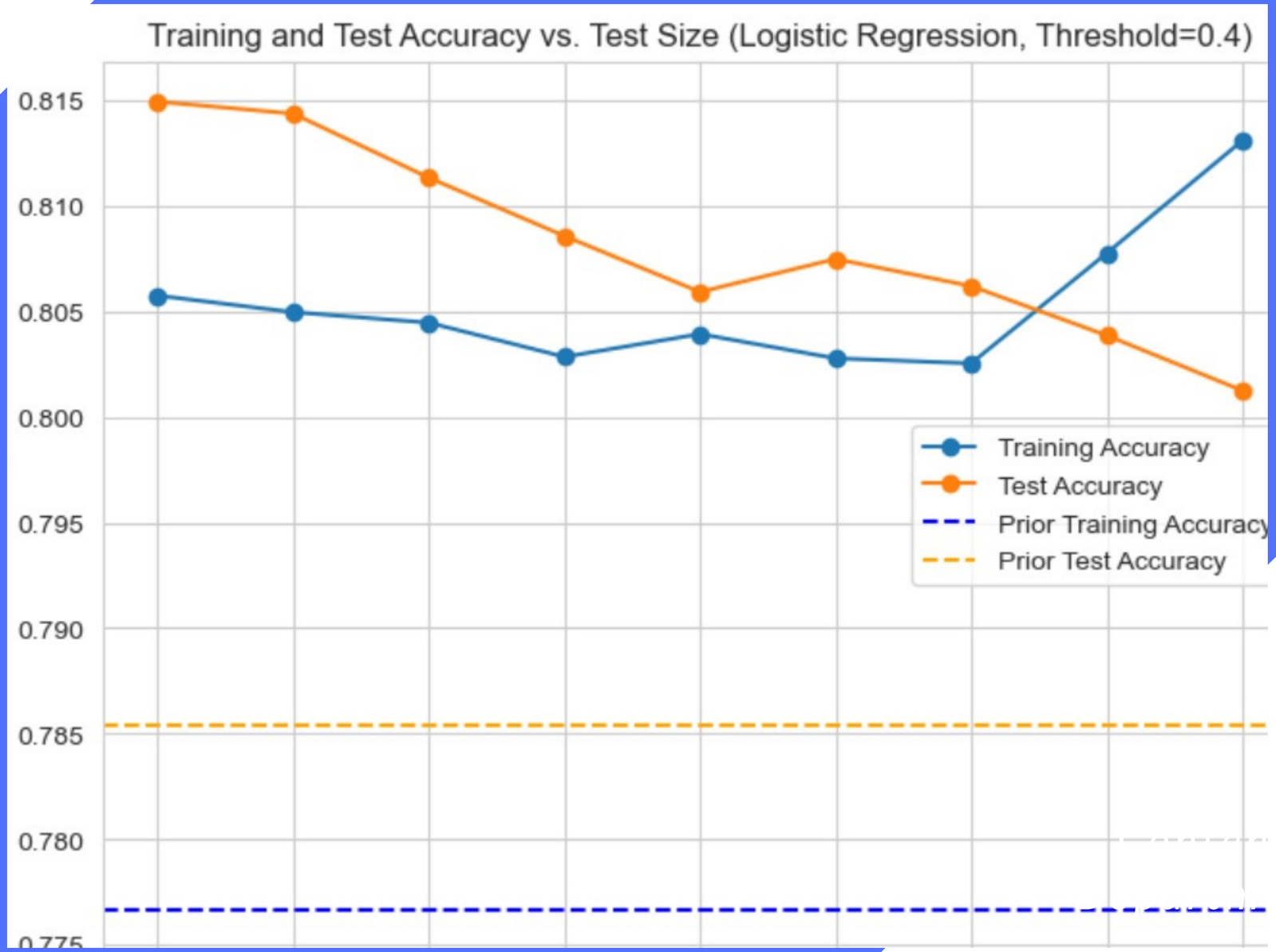
MODELING

Adaboost

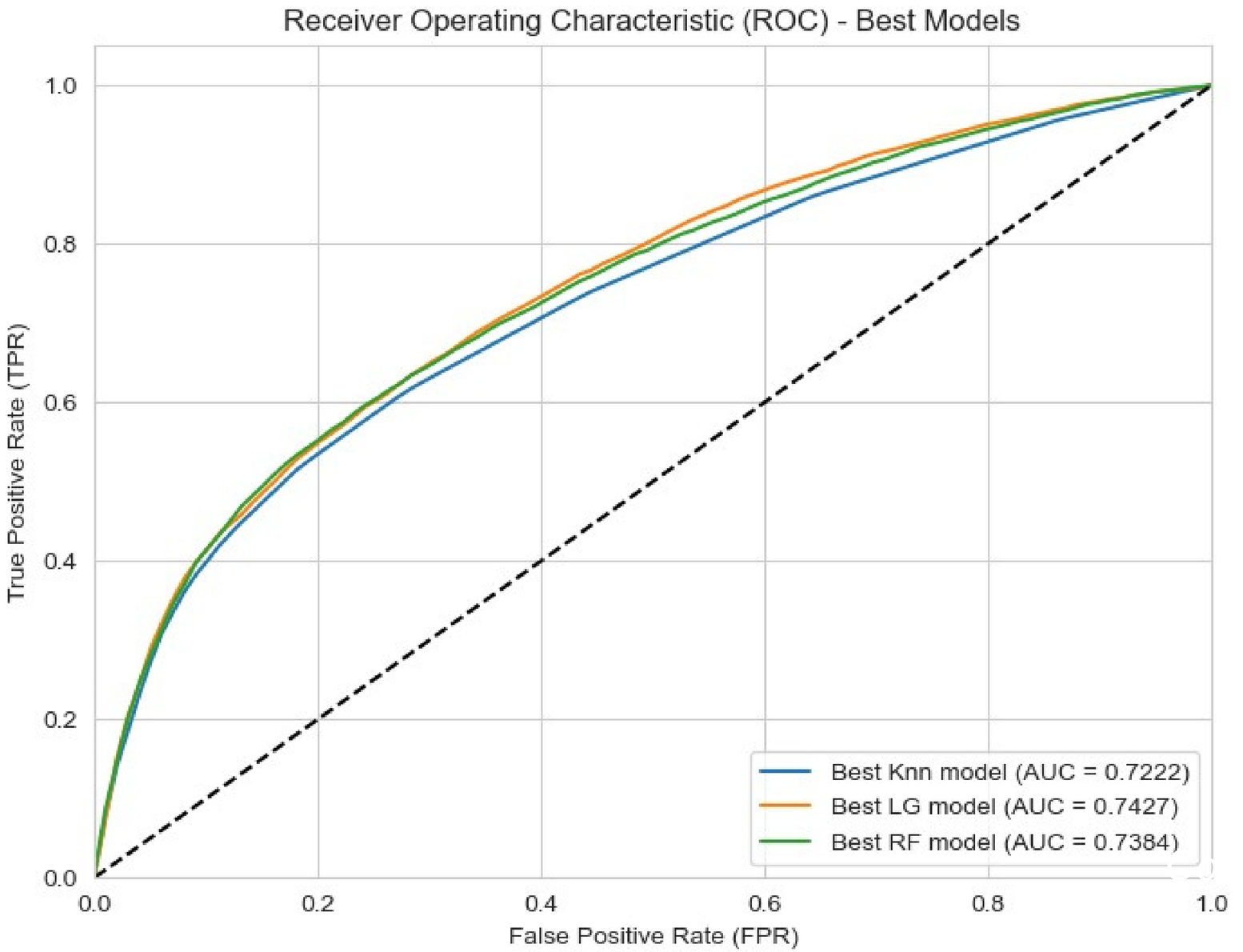
Adaboost combines multiple weak classifiers into a strong one by iteratively giving more weight to misclassified data, improving overall classification accuracy.

RESULTS

BEST MODEL vs BASELINE MODEL



ROC CURVE


Receiver Operating Characteristic (ROC) - Best Models
True Positive Rate (TPR)
False Positive Rate (FPR)
0.0
0.2
0.4
0.6
0.8
1.0
Best Knn model (AUC = 0.7222)
Best LG model (AUC = 0.7427)
Best RF model (AUC = 0.7384)

TIME IMPACT ANALYSIS

4 Months

Metric	Value
F1	0.783383
recall	0.808244
precision	0.785415
accuracy	0.808244

6 Months

Metric	Value
F1	0.776789
recall	0.802819
precision	0.779792
accuracy	0.802819

SVM

INITIALLY, WE ATTEMPTED TO RUN AN SVM MODEL- HOWEVER, DUE TO THE LARGE SIZE OF OUR DATASET, THE PROCESS DID NOT COMPLETE EVEN AFTER RUNNING FOR 5 HOURS.

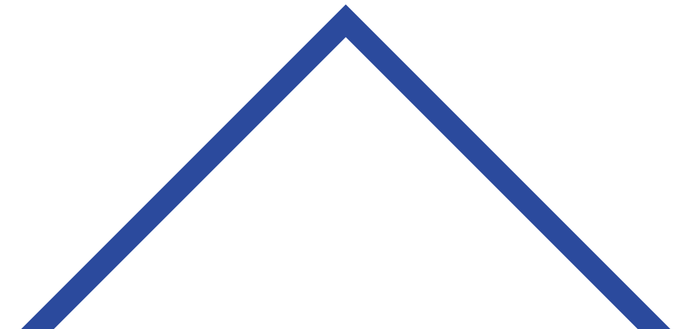
CONSEQUENTLY, WE DECIDED TO APPLY THE PCA ALGORITHM TO REDUCE THE DIMENSIONALITY OF THE DATA.

UNFORTUNATELY, EVEN WITH PCA, THE MODEL FAILED TO FINISH RUNNING AFTER APPROXIMATELY 4 HOURS, LEADING US TO TERMINATE THE ATTEMPT.

WE DID IT BECAUSE CONSIDERED EXPLORING OTHER MODEL WITH DIFFERENT ASSUMPTIONS.

PROBLEMS AND CHALLENGES WE FACED

- SVM took alot of time and didnt finish after 4 hours
- We lack certain qualities in our data
- We had difficulty decoding the data and turning it into numbers
- Data modeling took a long time
- Learning to use new tool such as matplotlib to display the results



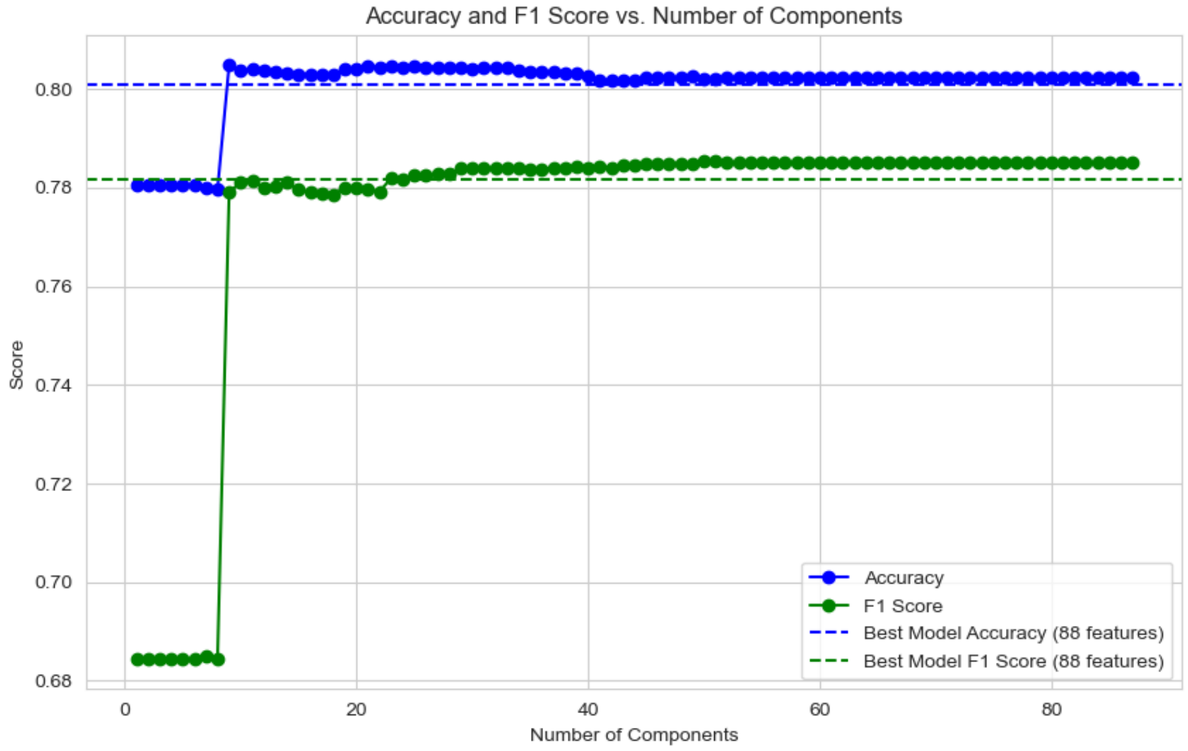
PCA ON OUR MODAL

It can be seen that for n_components smaller than 12 the accuracy and f1 are lower than our model now.

But as n_components increase, it can be seen that the indices improve slightly from our model.

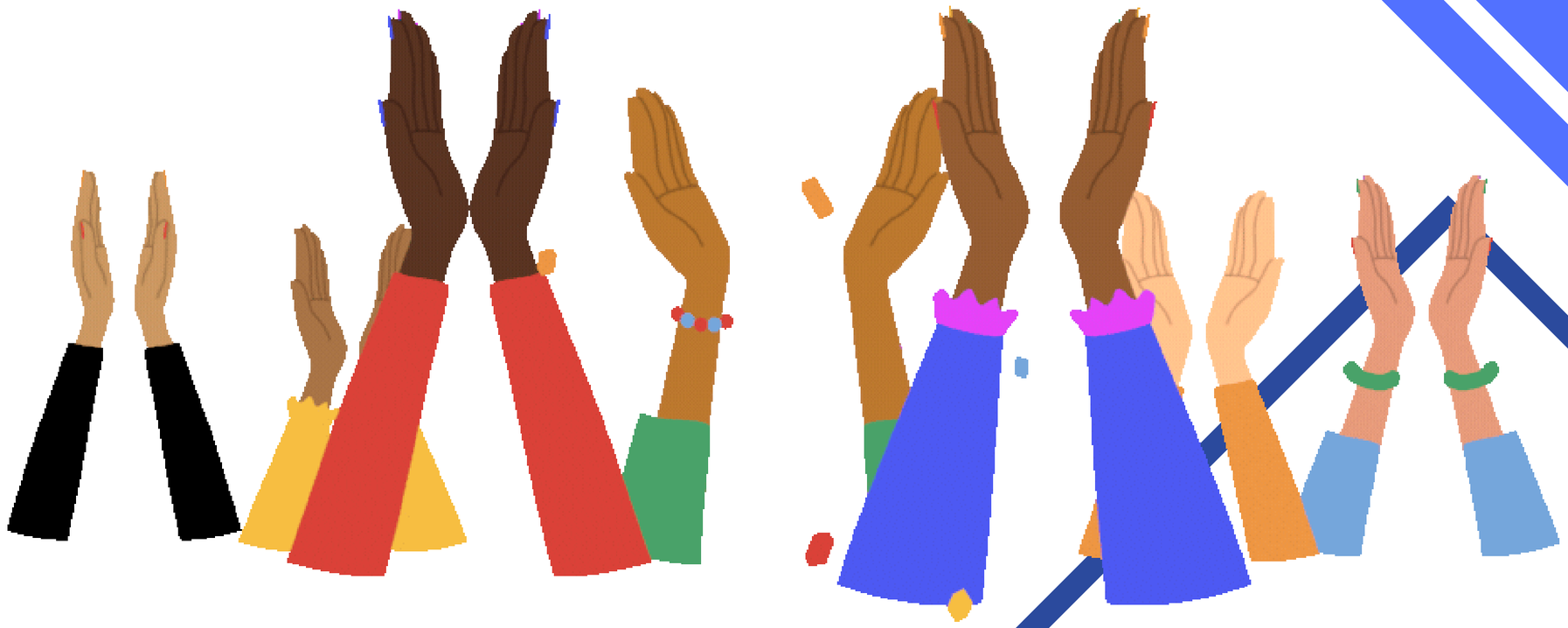
Since we did not implement the code together, we will leave the model as it is at the moment.

```
----------------------------------------
n_components: 8
Test F1 Score: 0.6844821935
Test Accuracy: 0.7797299431
----------------------------------------
n_components: 12
Test F1 Score: 0.7800191970
Test Accuracy: 0.8038139608
----------------------------------------
n_components: 17
Test F1 Score: 0.7788441577
Test Accuracy: 0.8029453569
----------------------------------------
n_components: 20
Test F1 Score: 0.7800930421
Test Accuracy: 0.8040903348
----------------------------------------
n_components: 30
Test F1 Score: 0.7838009553
Test Accuracy: 0.8042087808
----------------------------------------
n_components: 40
Test F1 Score: 0.7838808795
Test Accuracy: 0.8025900190
----------------------------------------
...
n_components: 70
Test F1 Score: 0.7850308059
Test Accuracy: 0.8023531270
----------------------------------------
```



CODE

GitHub link to our code:

https://github.com/MatanAdar/default-of-credit-card-clients

THANK YOU