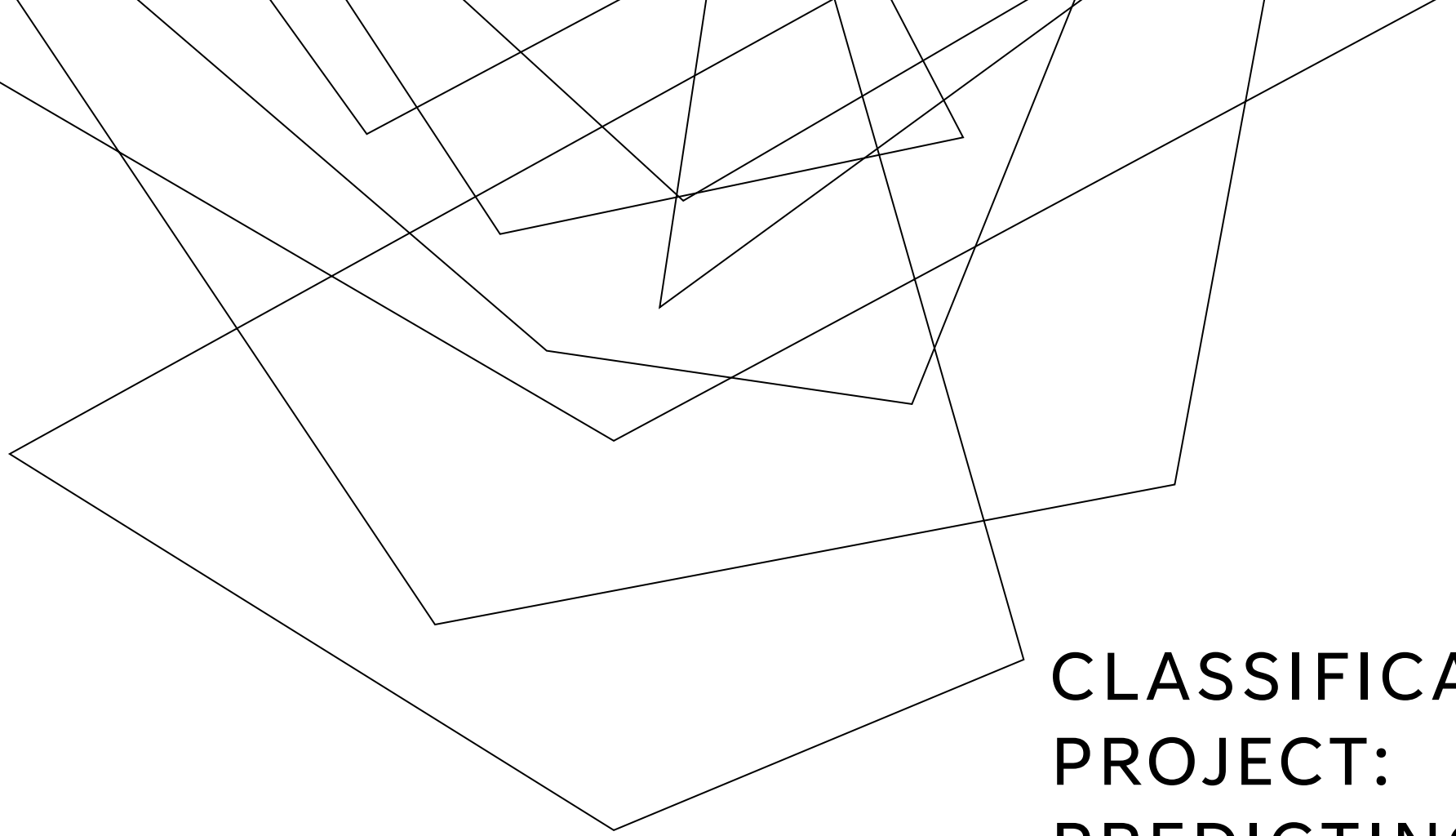


CLASSIFICATION PROJECT: PREDICTING FRAUDULENT CREDIT CARD TRANSACTIONS

Author: Adrian P. Bustamante, Ph.D.
Email: adrianpebus@gmail.com

OUTLINE

Objective

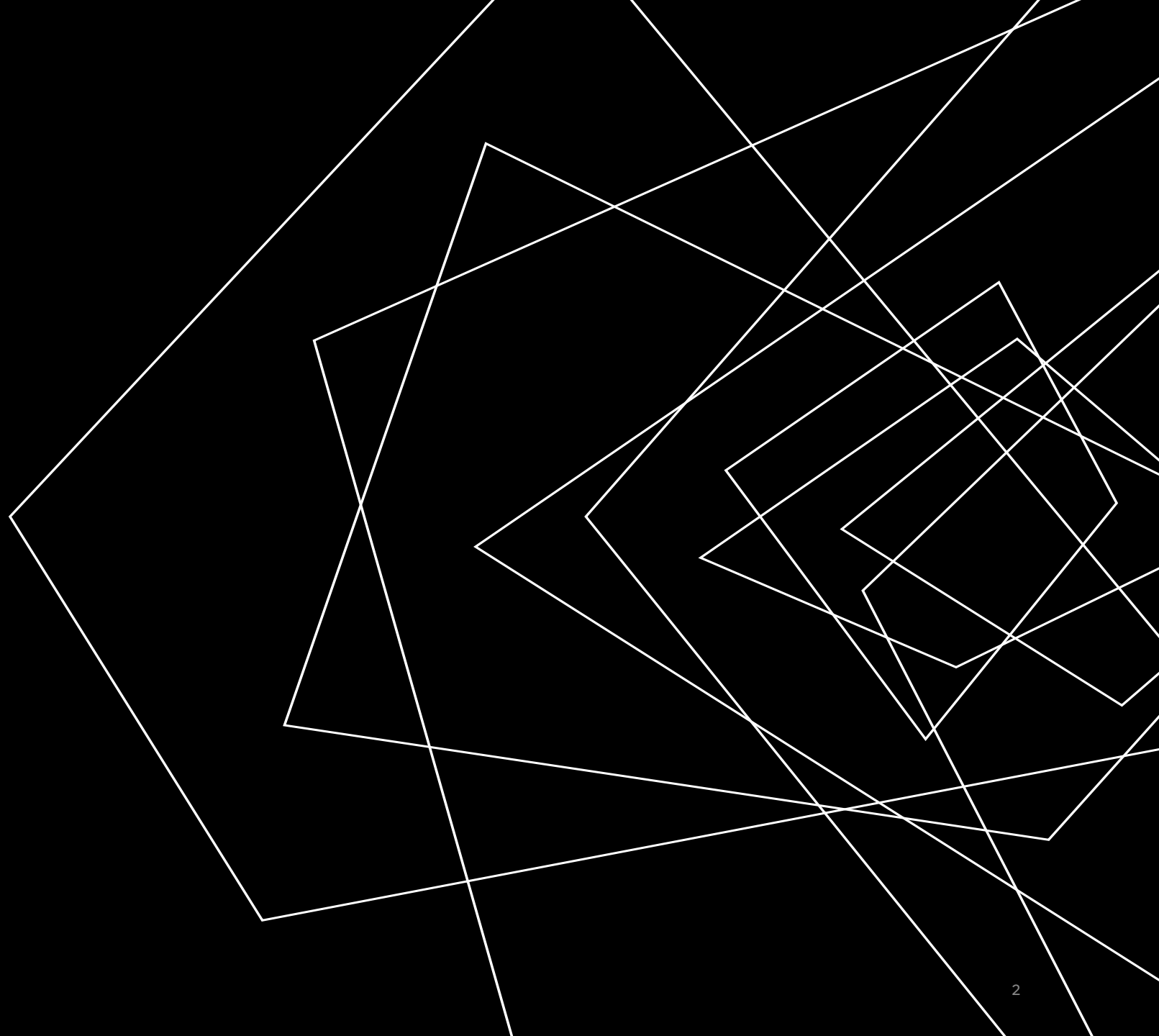
About the data

Feature engineering and EDA

Classification Models

Summary of results and key findings

Conclusion



OBJECTIVE

We study a dataset containing transactions made by credit cards that occurred during two days in 2013. The dataset is highly unbalanced, we have 492 frauds out of 284 807 transactions, roughly the 0.172% of all transactions.

The objective of this study is to train different classification models to predict whether a transaction is fraudulent or not. We want to find a model that performs well on precision, recall and fscore.

Our focus is on predictability but we also include a couple of results on interpretability. To measure the performance of the models considered we will be focusing on precision, recall and f1score.

ABOUT THE DATASET

The datasets contains transactions made by credit cards in September 2013 by European cardholders.

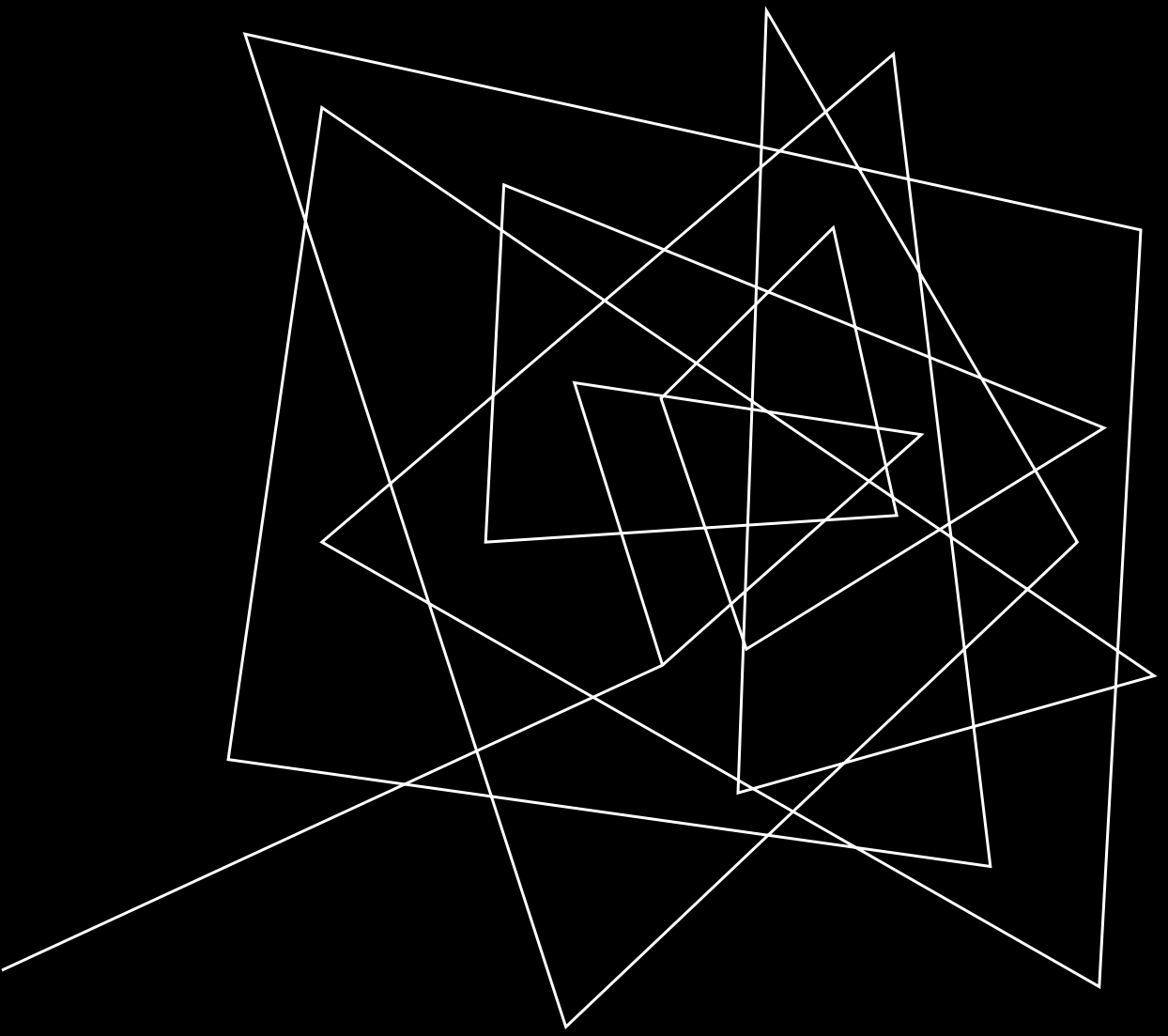
It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, the original features and more background information about the data cannot be provided. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount.

The dataset has been collected and analysed during a research collaboration of Worldline and the Machine Learning Group (mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection. More details on current and past projects on related topics are available on http://mlg.ulb.ac.be/BruFence and http://mlg.ulb.ac.be/ARTML.

The dataset used for this project can be found at https://www.kaggle.com/datasets/qnqfbqfqo/credit-card-fraud-detection-date-25th-of-june-2015

EDA AND FEATURE ENGEENERING

- **The data contains only numerical variables, no need of encoding.**
- **We removed duplicated rows.**
- **Removed feature 'Time' which only order the rows of the data. Ordering is not needed for the models implemented.**
- **We uses stratified spliting for the train and test sets.**
- **Data is highly unbalance, we did some oversampling (SMOTE) and undersampling (Random Undersampling) to train a couple of models.**
- **We use MinMaxScaler when using certain models (Log Regression, KNN, SVC)**



CLASSIFICATION MODELS

MODELS USED WITH OVERSAMPLING AND UNDERSAMPLING

- **Logistic Regression with l2 regulariztions and CrossValidation**

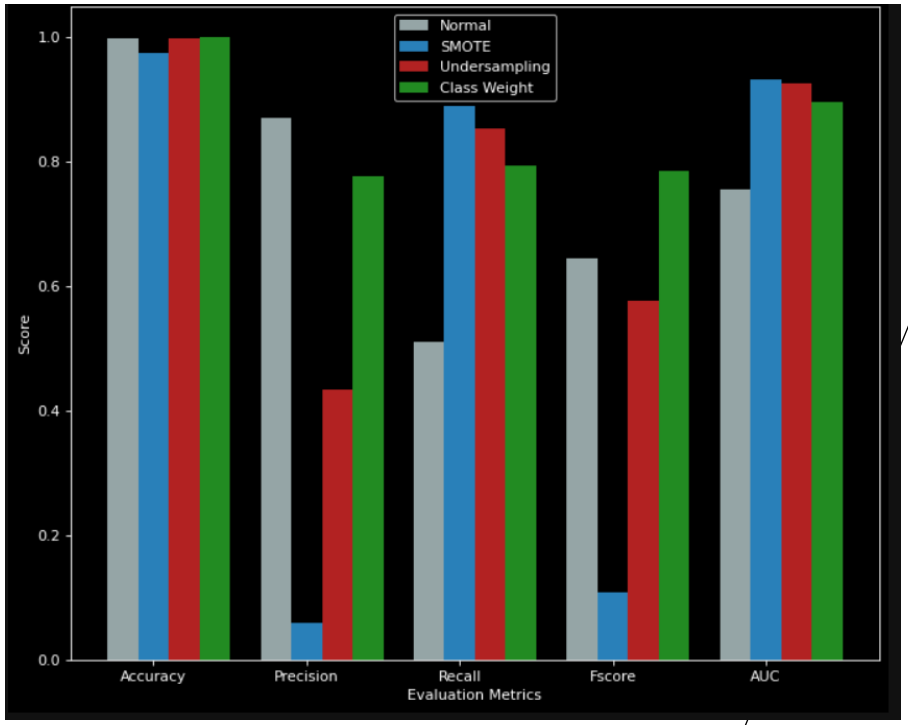
  SMOTE oversampling, Random Undersampling, and class weights. Cross validation to choose class weights.

- **K-Nearest Neighbors (KNN) with CrossValidation.**

  SMOTE oversampling, Randon Undersampling. Cross Validation to choose number of neighbors.

- **Support Vector Classifier (SVC).**

  SMOTE oversampling, Randon Undersampling. Crosss Validation to choose kernel, degree and regularization constant.



Results for Logistic Regression. The results for KNN and SVC are similar, see summary of results below.

ENSEMBLE MODELS

- **Bagging Trees.**

  Cross Validation to choose number of estimator and max_depth of trees. Gini impurity was used.

- **Random Forests.**

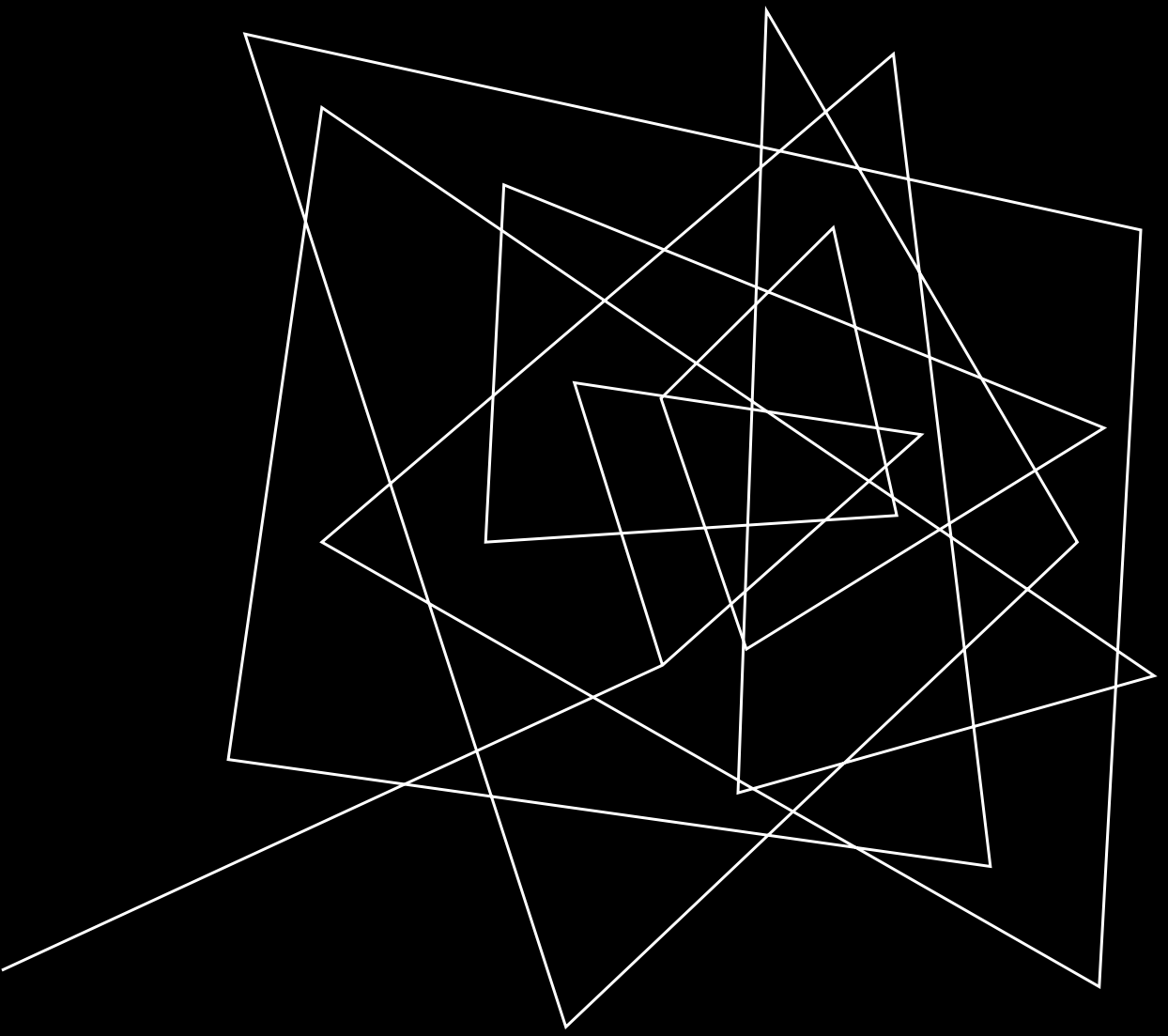
  Cross Validation to choose number of estimator and max_depth of trees. Gini impurity was used.

- **Gradient Boosting (XGBoost)**

  Cross Validation to choose the learning rate and number of estimators. Logistic regression as weak estimators.

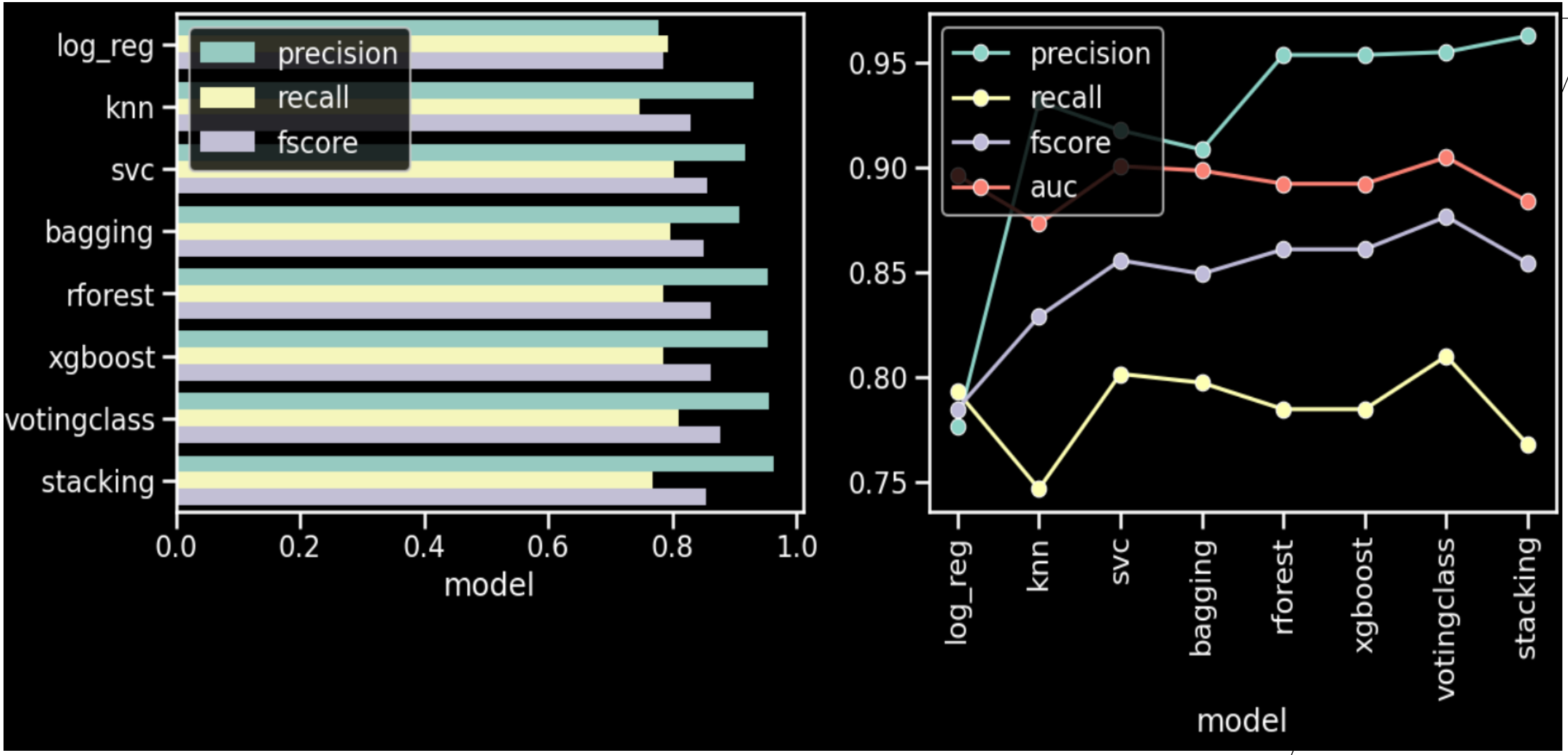
- **Stacking: Using all the previous classification models.**

  - Voting Classifier
  - Stacking Classifier: Log-reg as final estimator.



SUMMARY OF RESULTS AND MODEL INTERPRETABILITY

SUMMARY OF METRICS FOR ALL MODELS CONSIDERED



BEST MODEL:
STACKING – VOTING CLASSIFIER.

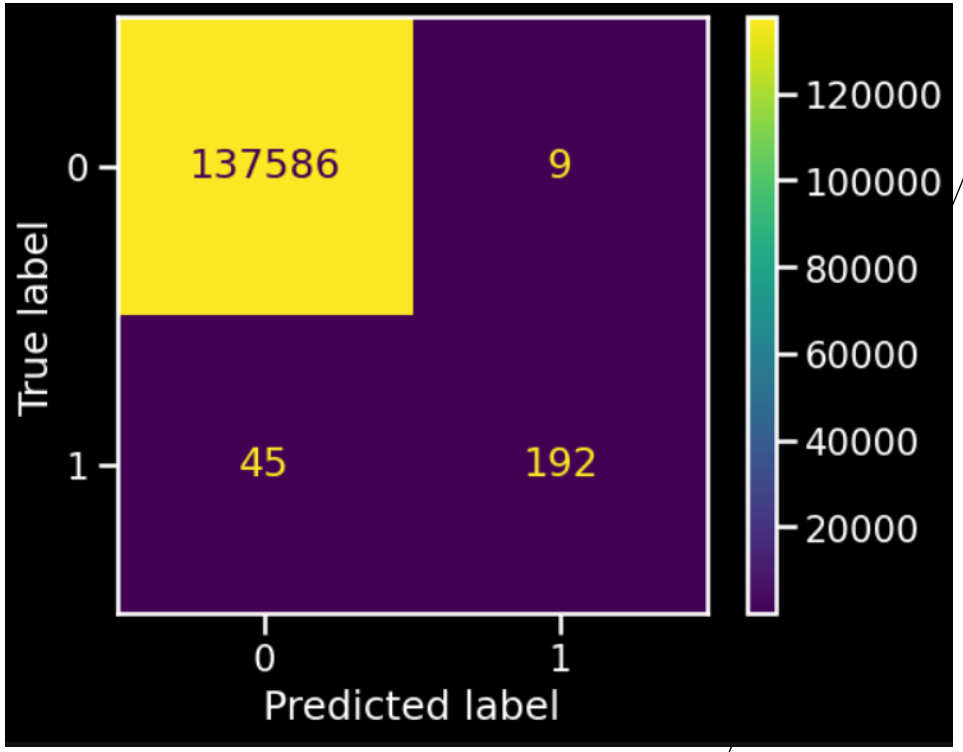
Precision = 0.9552238805970149

Recall = 0.810126582278481

Fscore = 0.8767123287671232

- Among all the classifiers considered the Voting Classifier achieve the higher values for Precision, Recall, and Fscore.

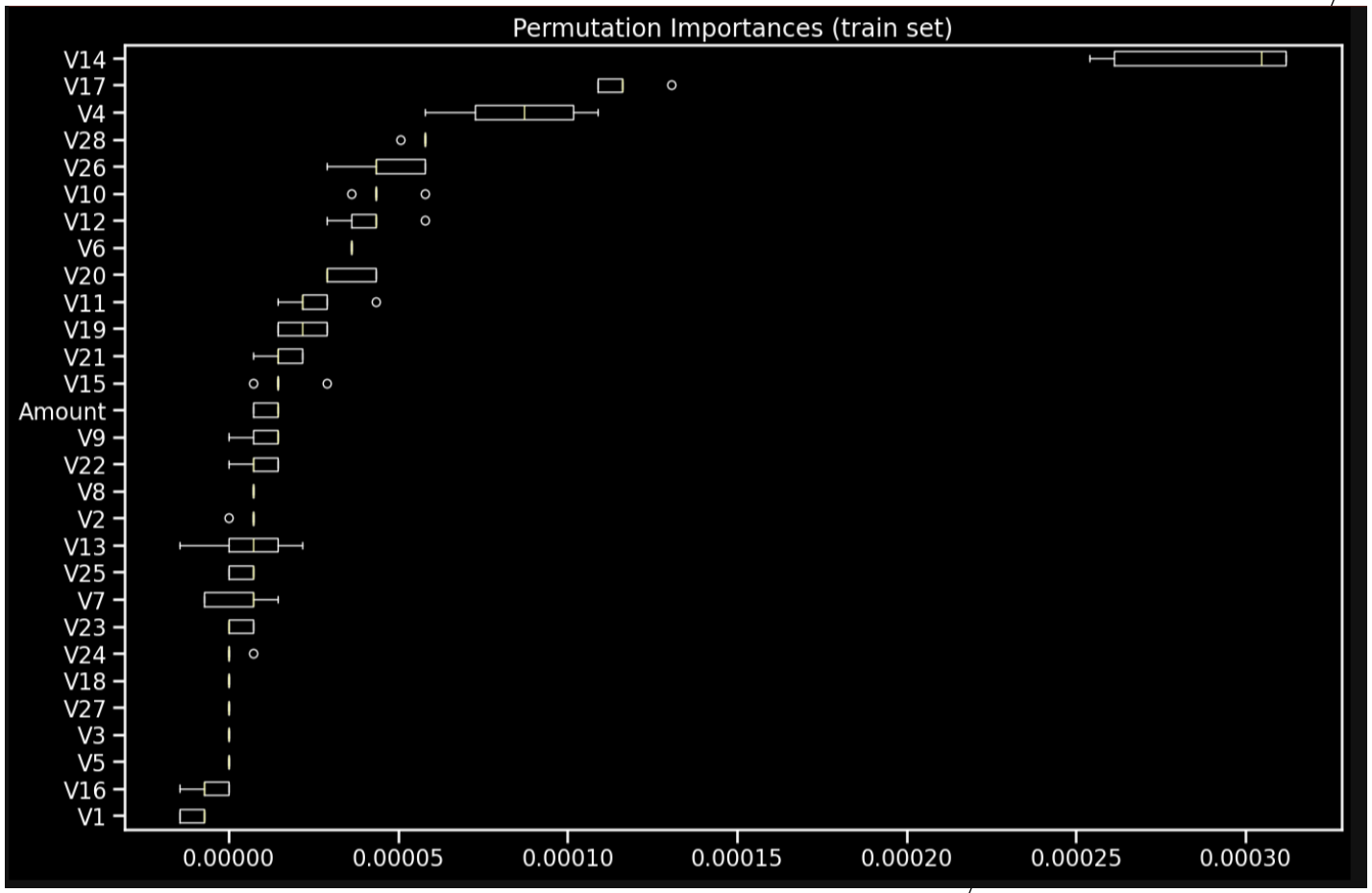
- The Voting Classifier is constructed using the previous trained models: Logistic Regression, KNN, SVC, Bagging Tree, Random Forest, XGBoost.



Confusion matrix computed on the test set for the Voting Classifier considered as best model.

PERMUTATION FEATURE IMPORTANCE

- **Permutation feature importance was performed using 5 as the number of permutation.**
- **Most important feature seems to be 'V14'.**



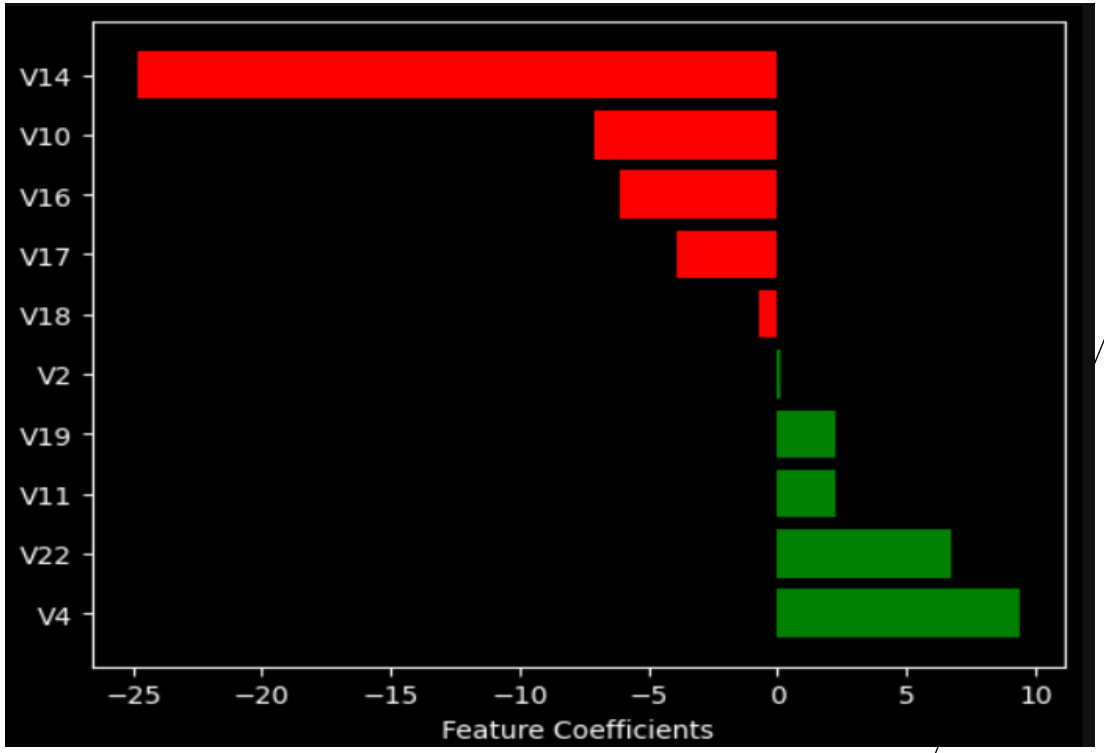
GLOBAL SURROGATE MODEL

- **We used a Logistic Regression as a Global Surrogated Model**
- **Metrics between global surrogated model and Voting Classfier:**

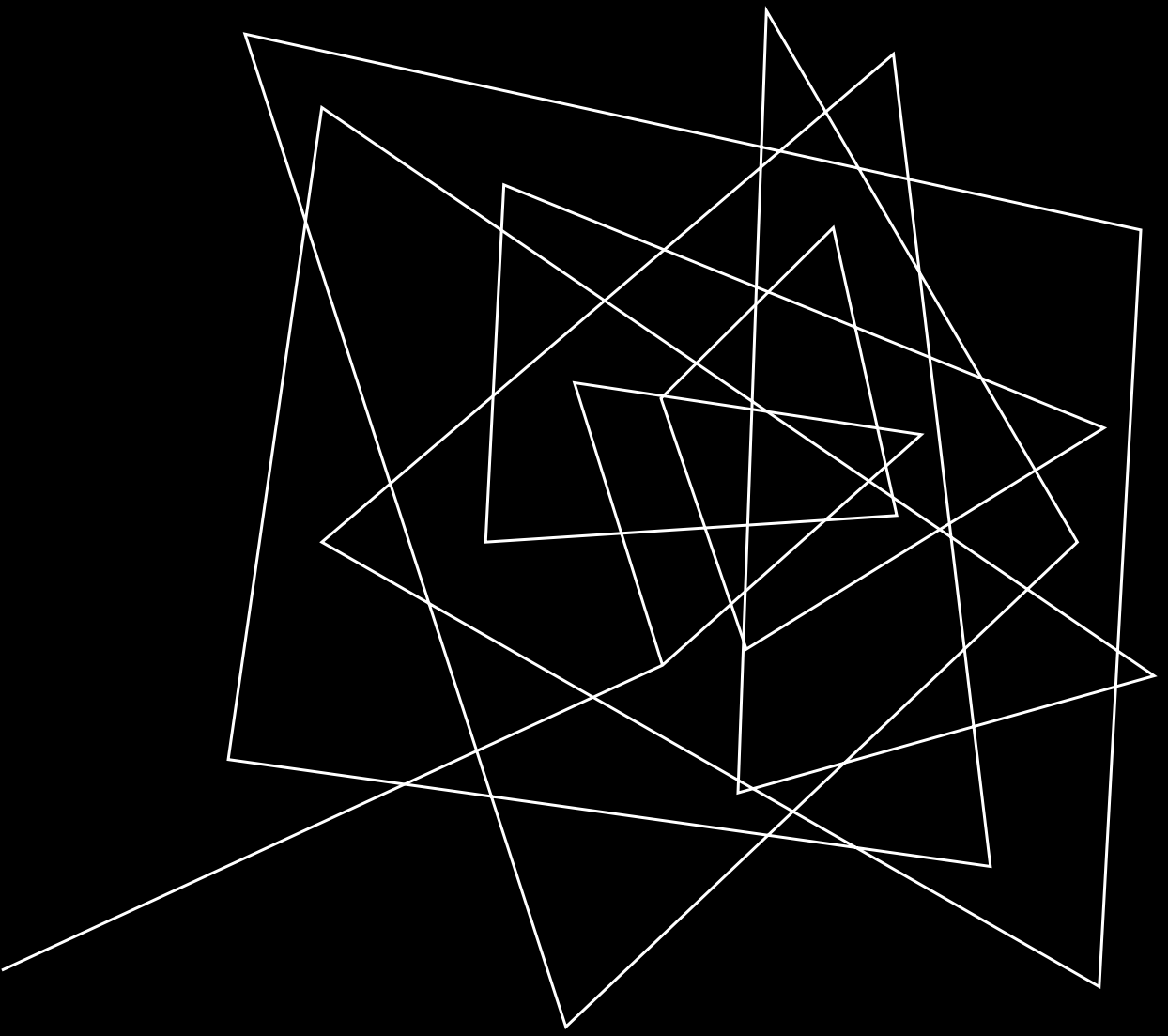
  Precision = 0.9

  Recall = 0.72

  Fscore = 0.8
- **Also here, the most important feature seems to be 'V14'**



Feature Coefficients for the Global Surrogated Model.



CONCLUSIONS AND NEXT STEPS

CONCLUSION

We worked with a dataset containing information about credict card transactions and our objective was to determine whether a credit card transaction was fraudulent or not. That is, our aim was to find a classification model that performs well on precision, recall, and fscore. We focused on these metrics due to the fact that the dataset is highly unbalanced.

The model found to perform best (among the models considered) is a Voting Classifier constructed from an ensemble of models containing the following ones: Log-Regression, KNN, SVC, Bagging Trees, Random Forest, and XGBoost. This Voting Classifier yiels a Precision = 0.95, a Recall = 0.81, and a Fscore = 0.87. Moreover, we performed Permutation Feature Importance (PFI) analysis and use a Logistic Regression as a Global Surrogated Model, from their results it seems that the feature 'V14' is the most important one for the classification.

NEXT STEPS

The main challenge for this study was the lack of computational resources (everything was run in a laptop with 16gp of ram memory and 4 cores). In order to reduce the computational time, to perform the cross validations and PFI, we have decided to use large test sets, about 50% of the data. With better computational resources a more exhaustive search for the best hyperparameters can be achieve and, presumably, better results as well.

Jupyter Notebook with all the computations can be found at https://github.com/adrian-pbustamante/Classification-Project-predicting-fraudulent-credit-card-transactions/blob/main/Classification-fraudulent-transactions.ipynb

THANK YOU

Adrian P. Bustamante, Ph.D.

adrianpebus@gmail.com