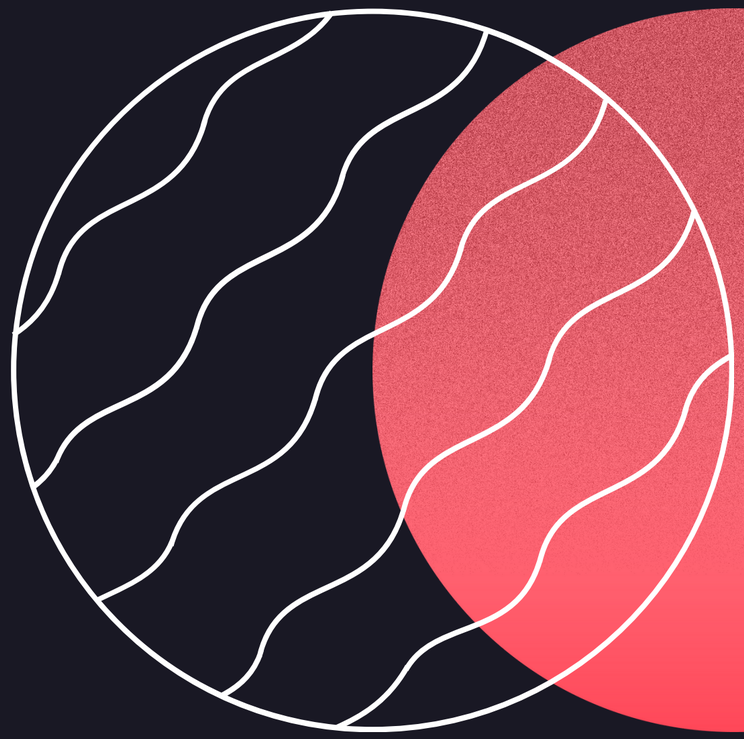


Machine Learning 2 Project

Adam Janczyszyn
Hubert Wojewoda

DATASETS CHOSEN

German Credit Risk

Classification

The original dataset contains 2000 entries with 20 categorial/numerical attributes prepared by Prof. Hofmann. In this dataset, each entry represents a person who takes a credit in a bank. Each person is classified as good or bad credit risks according to the set of attributes.

Red Wine Quality

Regression

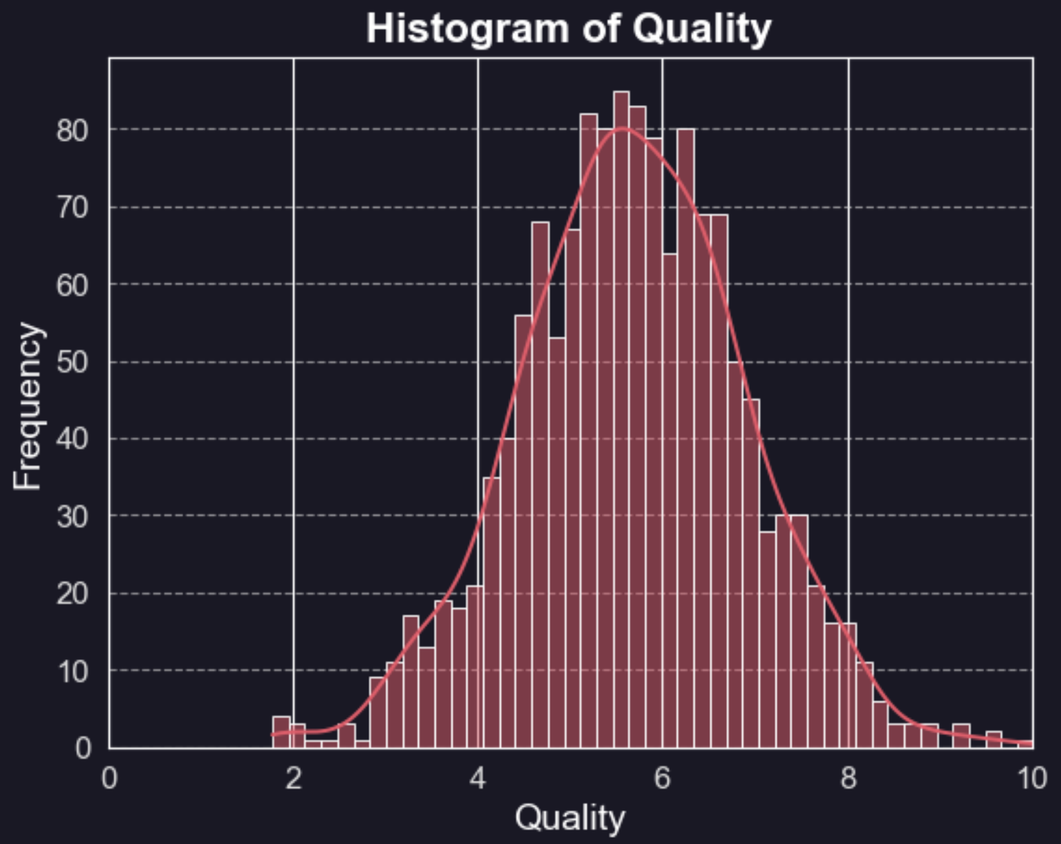
This datasets is related to red variants of the Portuguese "Vinho Verde" wine. For more details, consult the reference [Cortez et al., 2009]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

REGRESSION

VARIABLES

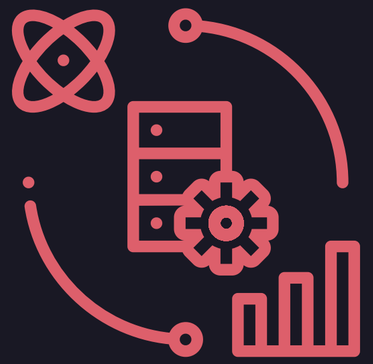
- Fixed Acidity: *Measures tartaric acid levels, affecting wine's taste and color.*
- Volatile Acidity: *Indicates acetic acid amount; high levels can lead to a vinegar taste.*
- Citric Acid: *Added for flavor enhancement and color stability.*
- Residual Sugar: *Determines sweetness; remaining sugar post-fermentation.*
- Chlorides: *Measures salt content in the wine.*
- Free Sulfur Dioxide: *Prevents oxidation and microbial growth.*
- Total Sulfur Dioxide: *Sum of bound and free SO2; affects smell and taste.*
- Density: *Related to alcohol and sugar content, close to water density.*
- pH: *Measures wine's acidity; most wines are between 3-4 on the pH scale.*
- Sulphates: *Additives contributing to antimicrobial and antioxidant properties.*
- Alcohol: *Percentage of alcohol by volume, influencing taste and body.*
- Quality (TARGET): *Sensory score from 0 to 10 by wine experts, indicating overall wine quality.*
- *+ 10 features added artificially*

DATA EXPLORATION



- 1400 observations, 22 features + target
- Target stats (Quality) - Max: 10.55, Min: 1.78, Mean: 5.67
- All variables are floats, no categorical variables
- No missing values in the data
- No duplicated values in the data
- Many features strongly correlated with target (alcohol, volatile acidity, sulphates, etc.)
- Most variables have distributions that resemble normal distribution

DATA PREPARATION



- All variables from the intial dataset scaled using StandardScaler
- All artificial variables scaled using MinMaxScaler
- Used train_test_split, where test is 20% of the data
- Scalers fitted on train split, then transformed both train and test
- All results checked with k-fold Cross-Validation

StandardScaler

Standardizes features by removing the mean and scaling to unit variance. This is achieved by subtracting the mean value from each feature and then dividing by the standard deviation. As a result, the feature will have a mean of 0 and a standard deviation of 1.

MinMaxScaler

Transforms features by scaling each feature to a given range, typically between 0 and 1. It does this by subtracting the minimum value of the feature and then dividing by the range. This scaling preserves the shape of the original distribution without distorting differences in the ranges of values.

MODELLING APPROACH

We checked a variety of different algorithms and chosen the most promising ones

Linear Regression

Offers a straightforward and transparent model that is easy to interpret, making it an excellent choice for understanding the relationship between features and the target. It's the go-to method for establishing a baseline in predictive performance.

Random Forest Regressor

Excels in handling complex datasets with interrelated features. Its ensemble approach, which builds multiple decision trees and merges their outcomes, is effective in reducing overfitting and providing a more generalizable model.

CatBoost Regressor

Has strong robustness to different data distributions. It employs gradient boosting on decision trees and has been engineered to deliver state-of-the-art results with minimal hyperparameter tuning. It is one of the spiritual successors to Random Forest.

LightGBM

High performance gradient boosting framework renowned for its speed and efficiency, particularly with large datasets. It uses advanced techniques to accelerate training and reduce memory usage without compromising accuracy.

Stacking

Ensemble technique that combines multiple regression models to capitalize on their individual predictive power. By learning how to use the strengths of each base model, stacking often achieves better performance than any single model could on its own.

Voting

Incorporates predictions from various models and uses a majority vote for the final prediction. This method is beneficial because it can smooth out individual model anomalies and biases, leading to improved accuracy and reliability in predictive outcomes.

MODELLING RESULTS

For each model, we searched for the best hyperparameters (Random Search)
and used Forward Selection algorithm with Random Forest to get the final features

LightGBM

- Hyperparameters
  - 'subsample': 0.8,
  - 'reg_lambda': 0,
  - 'reg_alpha': 0.1,
  - 'learning_rate': 0.021,
  - 'colsample_bytree': 0.8
- Results
  - MAPE: 0.17
  - MAE: 0.87
  - R2: 0.20

Random Forest

- Hyperparameters
  - 'min_weight_fraction_leaf': 0.0,
  - 'min_samples_split': 2,
  - 'min_samples_leaf': 8,
  - 'max_features': 'auto',
  - 'max_depth': 30
- Results
  - MAPE: 0.17
  - MAE: 0.88
  - R2: 0.20

CatBoost

- Hyperparameters
  - 'learning_rate': 0.046,
  - 'l2_leaf_reg': 0.1,
  - 'iterations':100,
  - 'depth': 6,
  - 'border_count': 254
- Results
  - MAPE: 0.17
  - MAE: 0.88
  - R2: 0.20

Linear Regression

- Results
  - MAPE: 0.17
  - MAE: 0.87
  - R2: 0.21

Stacking

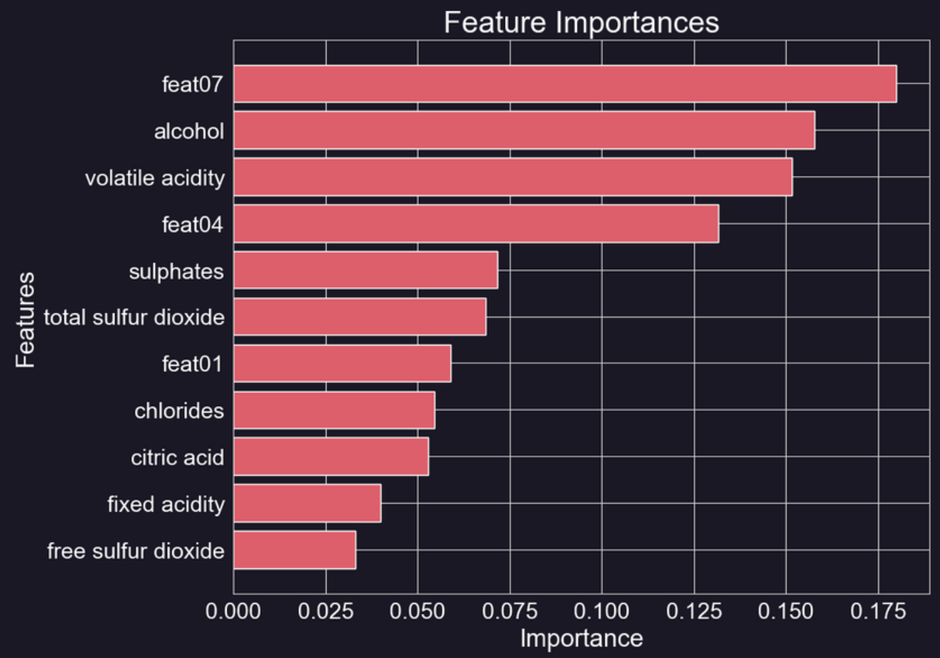
- Results
  - MAPE: 0.17
  - MAE: 0.86
  - R2: 0.22

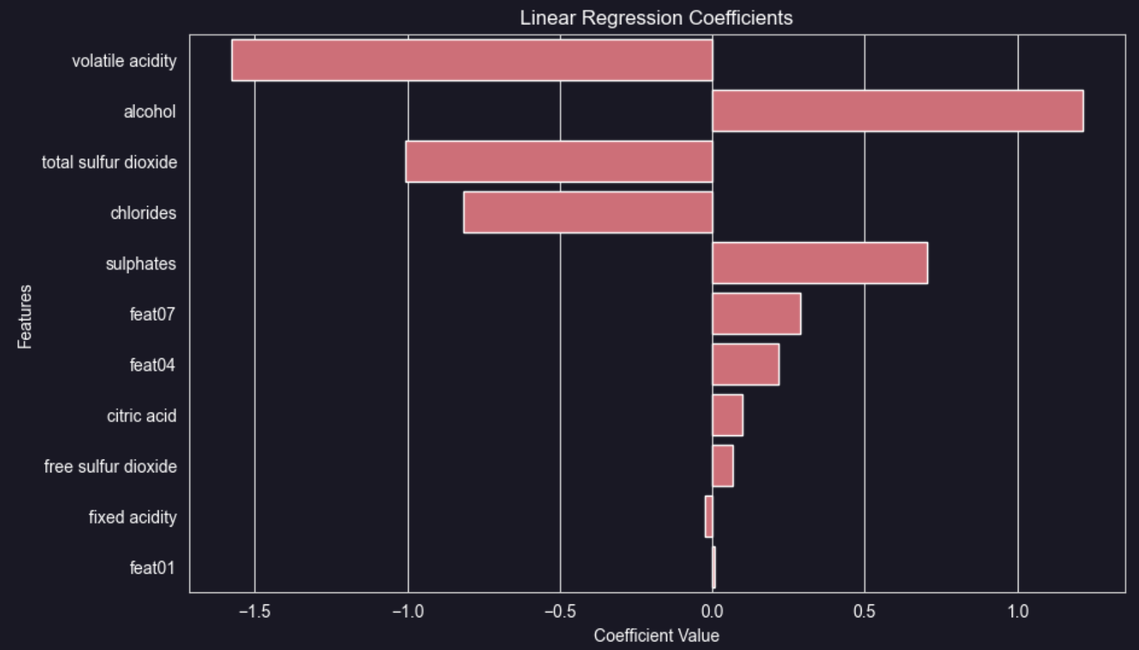
Voting

- Results
  - MAPE: 0.17
  - MAE: 0.86
  - R2: 0.22

EXPLAINABLE AI

To understand the models a bit better, we can analyze the importance for RF and directly interpret and check the coefficients of the linear regression model





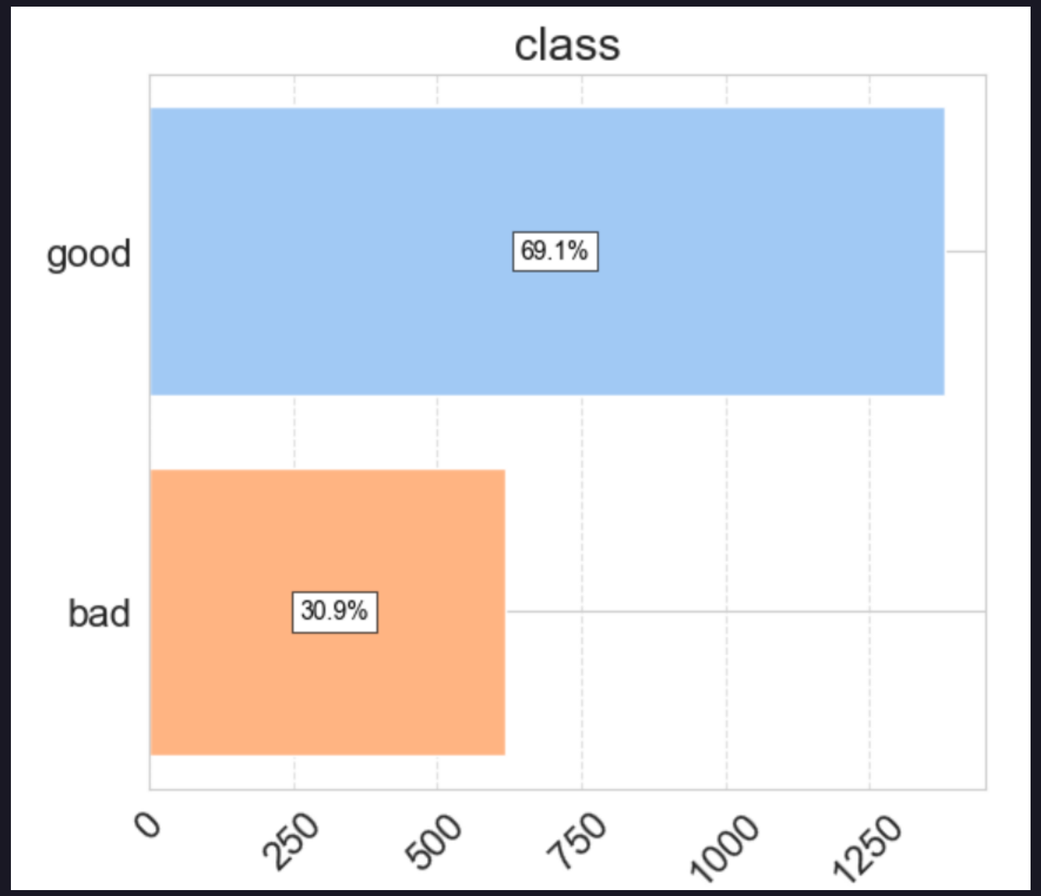
CLASSIFICATION

VARIABLES

- Age: *Age of the individual.*
- Checking Status: *Describes the status of the existing checking account.*
- Credit Amount: *The amount of credit requested.*
- Credit History: *Reflects the credit repayment history.*
- Purpose: *Specifies the reason for seeking credit.*
- Savings Status: *Indicates the status of savings accounts or bonds.*
- Employment: *Duration of present employment.*
- Installment Rate: *The percentage of disposable income dedicated to installments.*
- Personal Status and Sex: *Combines information about personal status and gender.*
- Other Debtors/Guarantors: *Presence of co-applicants or guarantors.*
- Present Residence Since: *Duration of current residence.*
- Property: *Describes the type of property owned or financed.*
- Other Installment Plans: *Indicates if there are other existing installment plans.*
- Housing: *Specifies the housing situation (rent, own, or free).*
- Number of Existing Credits: *The count of existing credits at this bank.*
- Job: *Describes the type of employment or job.*
- Number of People for Maintenance: *The number of dependents.*
- Telephone: *Indicates the presence or absence of a telephone.*
- Foreign Worker: *Specifies whether the individual is a foreign worker or not.*
- Class (TARGET): *Binary variable indicating the creditworthiness of customers good/bad.*
- *+ 10 features added artificially*

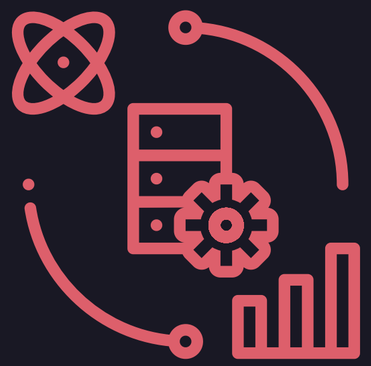
DATA EXPLORATION

Target variable distribution



- 2000 observations, 30 features + target
- Target (Class) - Good=69.1% (1382) / Bad=30.9% (618)
- Continuous varaibles - 13 / Categorical varaibles - 17 / Binary target - 1
- No missing values in the data
- No duplicated values in the data
- Correlated/Associated Variables with the Target (Spearman, Chi2 Test)
- Varied Distributional Characteristics

DATA PREPARATION



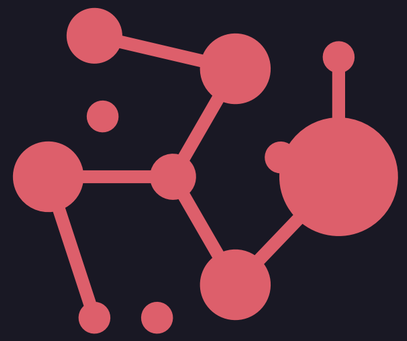
01 Exploratory Data Analysis

- eda using pandas profiling and additonal plots
- visualizations of transformations (e.g. log)
- correlation plots & binning

02 Data Processing

- Mapping categorical variables
- Categorical variables encoded using One Hot Encoding
- Varaibles scaled using StandardScaler and MinMaxScaler accordingly
- Used train_test_split, where test is 20% of the data (Out of sample data)
- Scalers fitted on train split, then transformed both train and test
- All results checked with stratified k-fold Cross-Validation

MODELLING APPROACH



We checked a variety of different algorithms using AutoML and chosen the most promising ones

Random Forest Classifier

A foundational model in predictive modeling, the Random Forest Classifier adopts a bagging (Bootstrap Aggregating) ensemble technique. It constructs multiple decision trees, training each on a subset of the dataset through bootstrapping, and subsequently aggregates their predictions. This methodology enhances robustness and interpretability, serving as an invaluable tool for establishing baseline predictive performance.

Extra Trees Classifier

An ensemble learning method that belongs to the family of decision tree-based models. Similar to Random Forests, Extra Trees builds multiple decision trees during training but with a key distinction — it introduces an additional layer of randomness in the tree-building process. Rather than selecting the optimal split at each node, Extra Trees randomly chooses splits, leading to a higher level of diversity among the individual trees. This technique often results in improved generalization performance and robustness.

XGBoost Classifier

XGBoost, short for eXtreme Gradient Boosting, is a powerful and efficient gradient boosting algorithm designed for classification and regression tasks. It sequentially builds a series of decision trees, each correcting errors from the previous ones. XGBoost introduces regularization techniques and utilizes gradient information for optimal tree construction, making it highly robust and adaptable to different data distributions. Known for its state-of-the-art performance and minimal hyperparameter tuning requirements, XGBoost is a popular choice in machine learning competitions and real-world applications where accuracy and efficiency are paramount.

Stacking

As an advanced ensemble technique, stacking combines predictions from diverse models through a meta-model. By strategically incorporating the outputs of multiple base models, stacking enhances predictive performance beyond the capabilities of individual models. This approach mitigates biases and anomalies, contributing to improved accuracy and reliability in classification scenarios.

CatBoost Classifier

CatBoost, short for Categorical Boosting, is a high-performance gradient boosting framework tailored for classification tasks. Designed to handle categorical features seamlessly, CatBoost employs advanced strategies to accelerate model training without compromising accuracy. With its efficient handling of large datasets and built-in support for categorical variables, CatBoost stands out for its speed and effectiveness. The framework incorporates techniques to reduce memory usage and accelerate convergence, making it particularly well-suited for complex classification challenges.

Voting

The voting ensemble method aggregates predictions from multiple models through a majority decision. This approach is pivotal in alleviating individual model biases and anomalies, fostering a more robust and accurate final prediction in classification tasks.

MODELLING RESULTS

For each model, we searched for the best hyperparameters (Random Search) and used Forward Selection algorithm with Random Forest to get the final features

Random Forest

- Hyperparameters
  - 'n_estimators': 100,
  - 'min_samples_split': 7,
  - 'min_samples_leaf': 4,
  - 'max_features': 'sqrt',
  - 'max_depth': 8
- Results
  - AUC PR: 0.76
  - Balanced Accuracy: 0.71
  - Gini: 0.73

XGBoost

- Hyperparameters
  - 'subsample': 0.7,
  - 'n_estimators': 300,
  - 'max_depth': 9,
  - 'learning_rate': 0.07,
  - 'colsample_bytree': 0.4,
  - colsample_bylevel = 0.4
- Results
  - AUC PR: 0.85
  - Balanced Accuracy: 0.83
  - Gini: 0.83

CatBoost

- Hyperparameters
  - {'subsample': 0.7,
  - 'learning_rate': 0.09,
  - 'iterations': 450,
  - 'depth': 6,
  - 'colsample_bylevel': 0.8,
- Results
  - AUC PR: 0.83
  - Balanced Accuracy: 0.79
  - Gini: 0.79

Extra Trees

- Hyperparameters
  - 'n_estimators': 200,
  - 'min_samples_split': 8,
  - 'min_samples_leaf': 2,
  - 'max_depth': 8,
- Results
  - AUC PR: 0.85
  - Balanced Accuracy: 0.76
  - Gini: 0.85

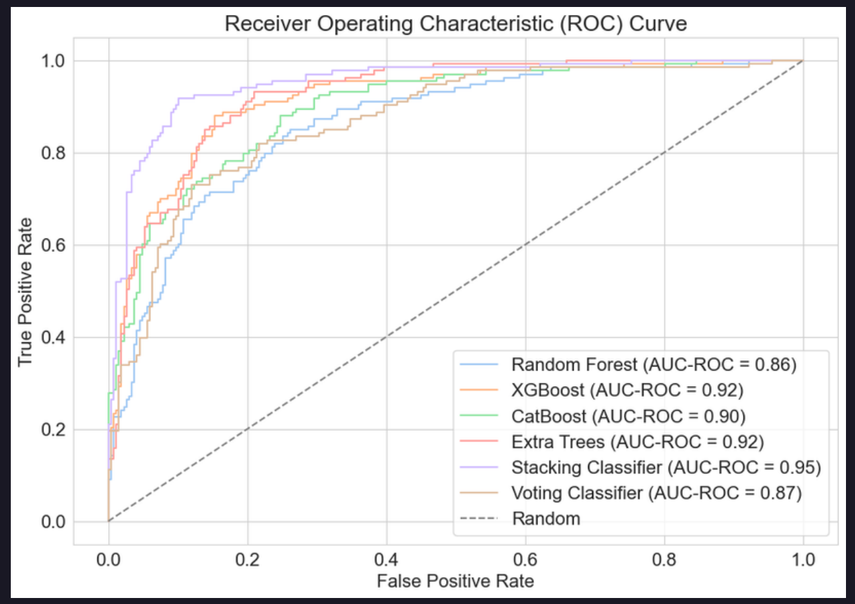
Stacking

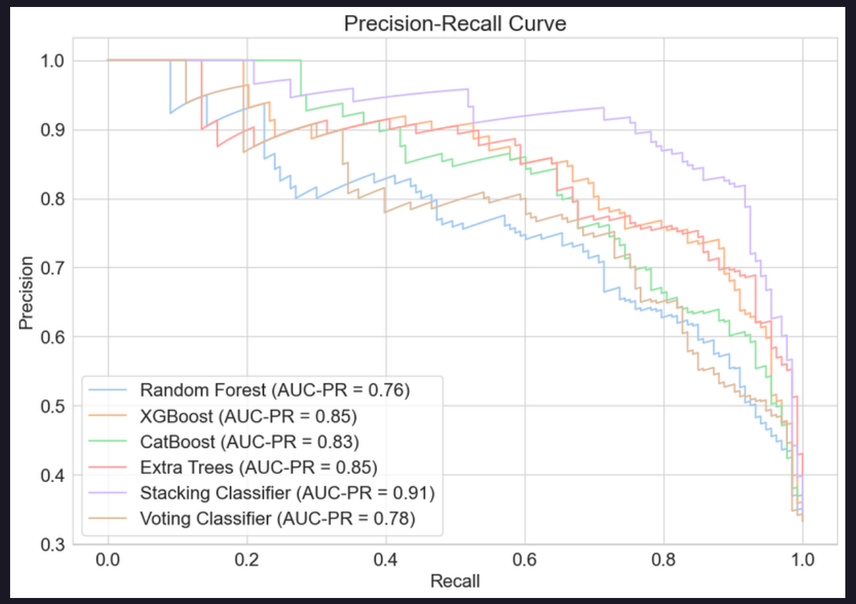
- Hyperparameters
  - 'penalty': 'l2',
  - 'C': 10,
- Results
  - AUC PR: 0.91
  - Balanced Accuracy: 0.87
  - Gini: 0.9

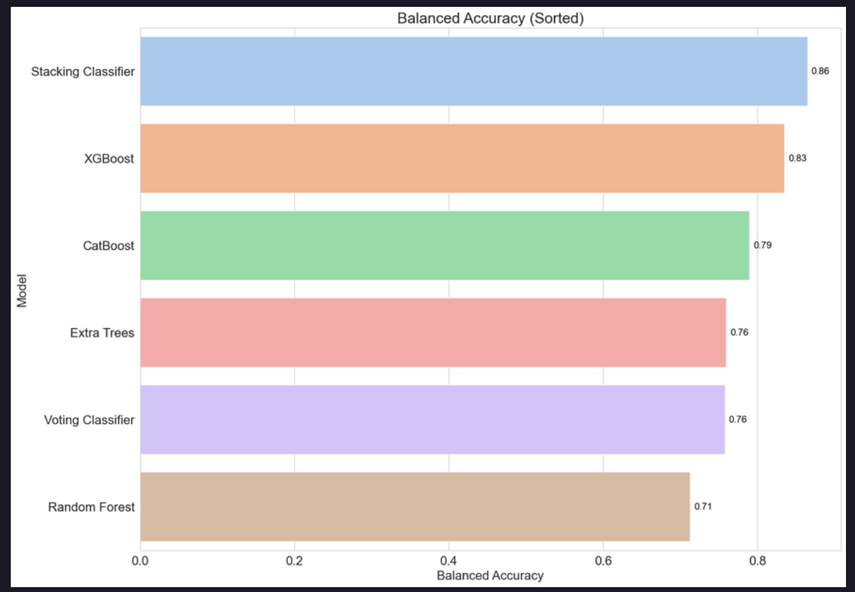
Voting

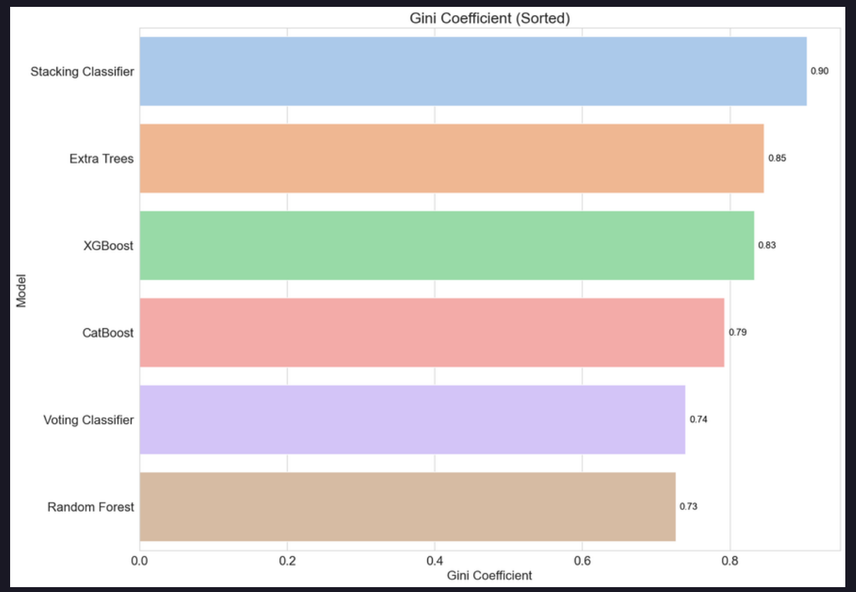
- Hyperparameters
  - 'weights': None,
  - 'voting': 'soft',
  - 'flatten_transform': False,
- Results
  - AUC PR: 0.8
  - Balanced Accuracy: 0.76
  - Gini: 0.76

MODELLING RESULTS







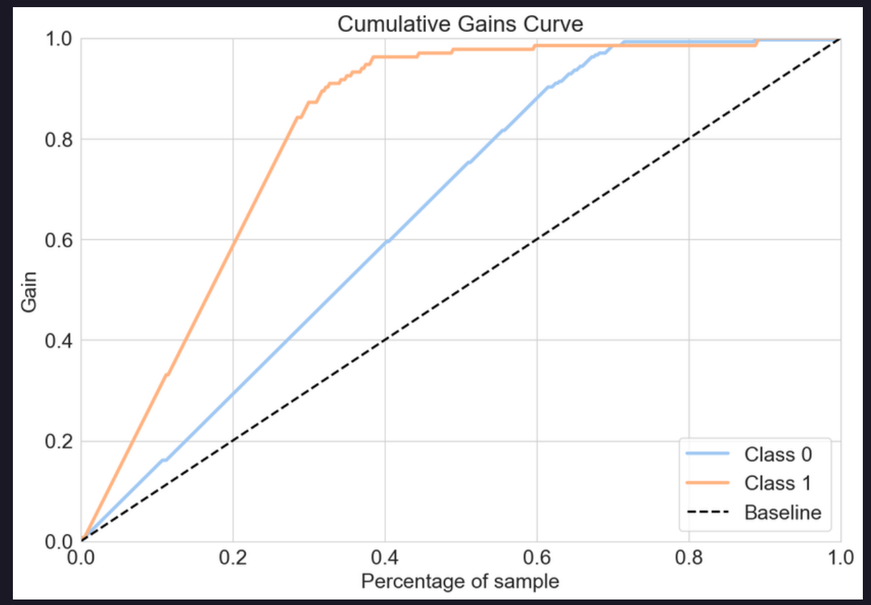


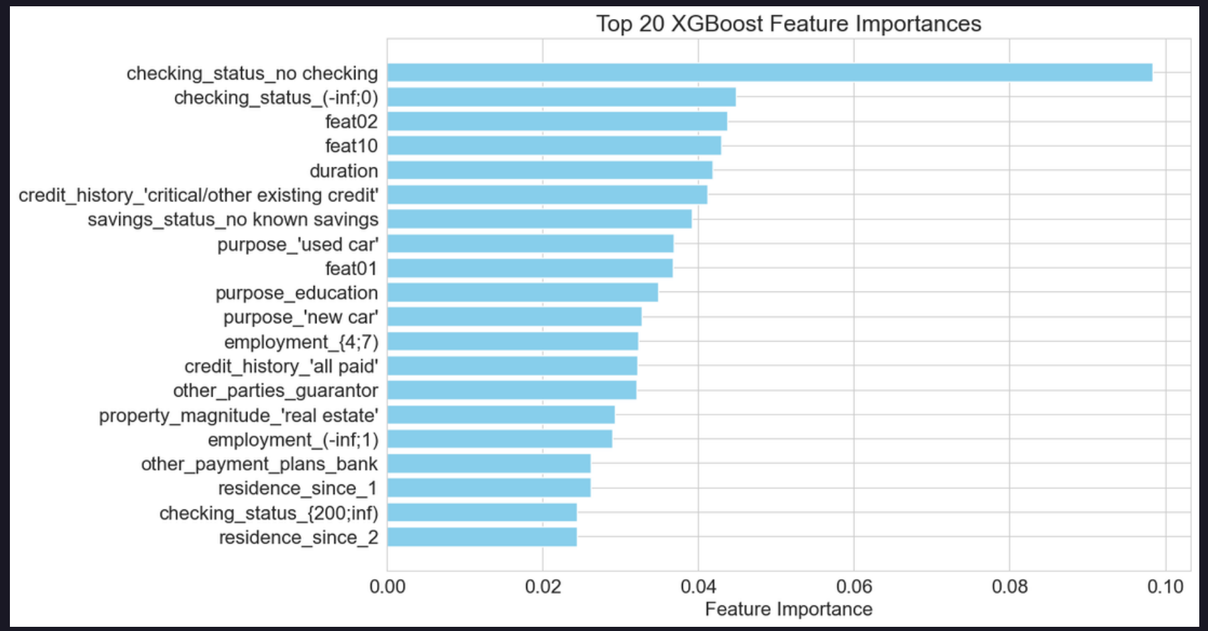
DEEP DIVE - XGBOOST

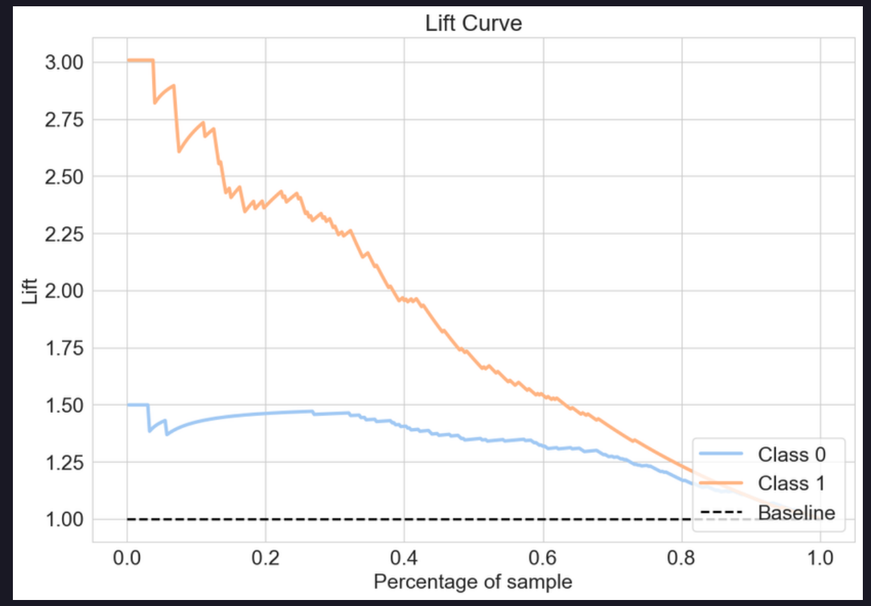
- scale_pos_weight - adressing imbalance classes in target variable
- eval_metric='aucpr' - better for imbalance target

  - Hyperparameters
    - 'subsample': 0.8,
    - 'n_estimators': 250,
    - 'max_depth': 8,
    - 'learning_rate': 0.09,
    - 'colsample_bytree': 0.7,
    - 'colsample_bylevel': 0.9.

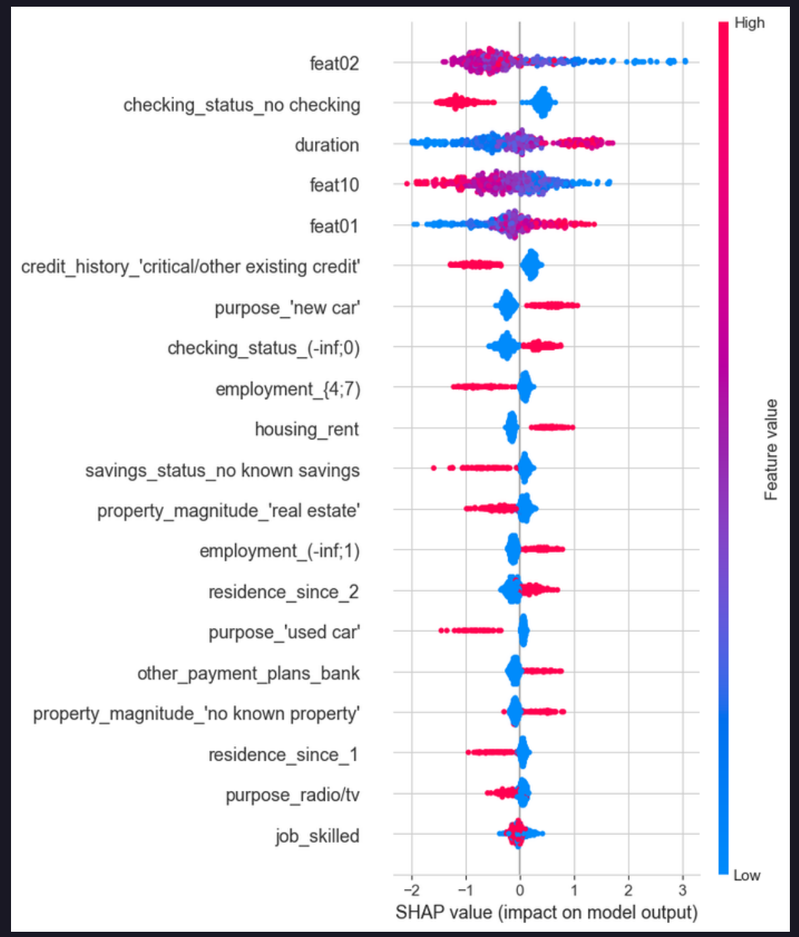
  - Results
    - AUC PR: 0.94
    - Balanced Accuracy: 0.93
    - Gini: 0.94

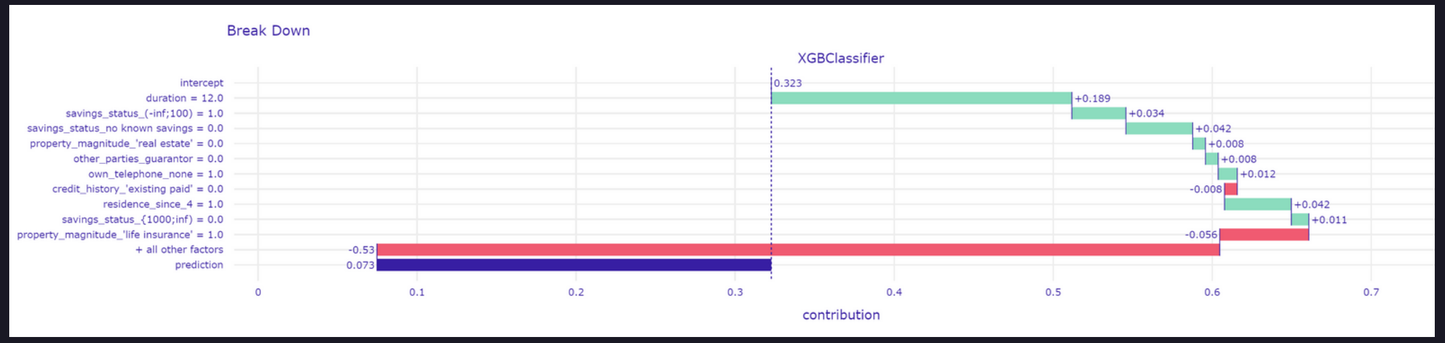


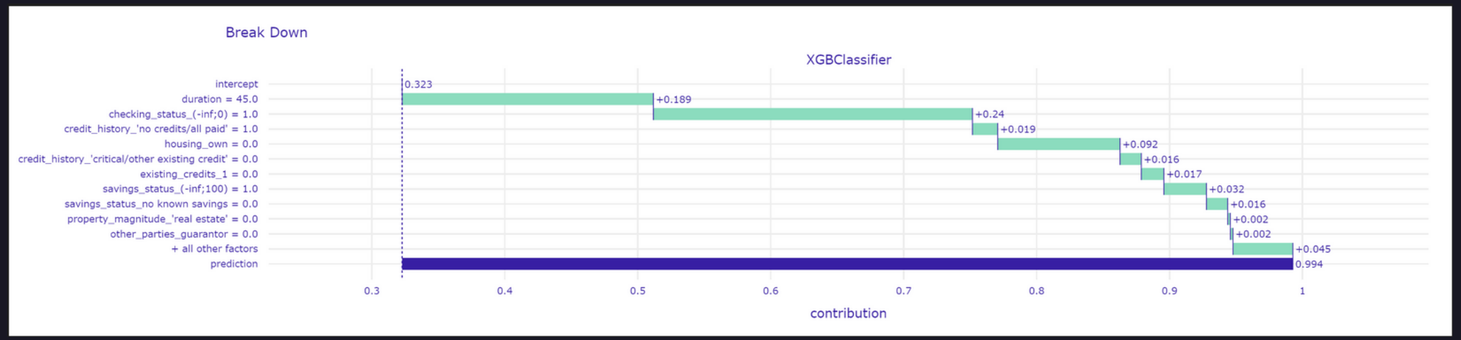




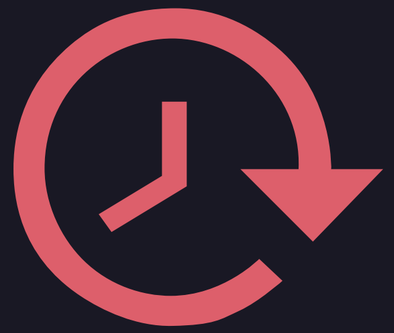
EXPLAINABLE AI







FUTURE ENDEAVORS IN MODEL DEVELOPMENT



REGRESSION

- Choose better hyperparamerets to combat overfitting and generalize the models
- Try to improve the Linear Regression model to find which interactions are beneficial
- Transform the output variable (log etc.) to get better results

CLASSIFICATION

- Explore the integration of Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance
- Mitigating Overfitting with Regularization Techniques
- Exploration of other Hyperparameter Optimization Techniques - Grid Search / Bayesian Optimization

Thank you for attention

Any questions?