STATISTICAL & MACHINE LEARNING

Group Project

Inder Rana, Nguyet Han Nguyen & Aleksandr Shkurin

11th April 2022

OUR TEAM

INDER RANA

MSc in Big Data Analytics for Business

IESEG School of Management

NGUYET HAN NGUYEN

MSc in Big Data Analytics for Business

IESEG School of Management

ALEKSANDR SHKURIN

MSc in Big Data Analytics for Business

IESEG School of Management

CONTENTS

PROJECT OVERVIEW

Problem Description: Credit card delinquency rose from 1.54% to 1.62% in the 4th quarter of 2021*. It is still under 2% of historic low, however the total amount of delinquencies has a major impact on the lending bank.

Project Description: Predict if the client will be able to pay the credit card bill in the upcoming month to help the lending bank take pro-active actions against the client to avoid any losses.

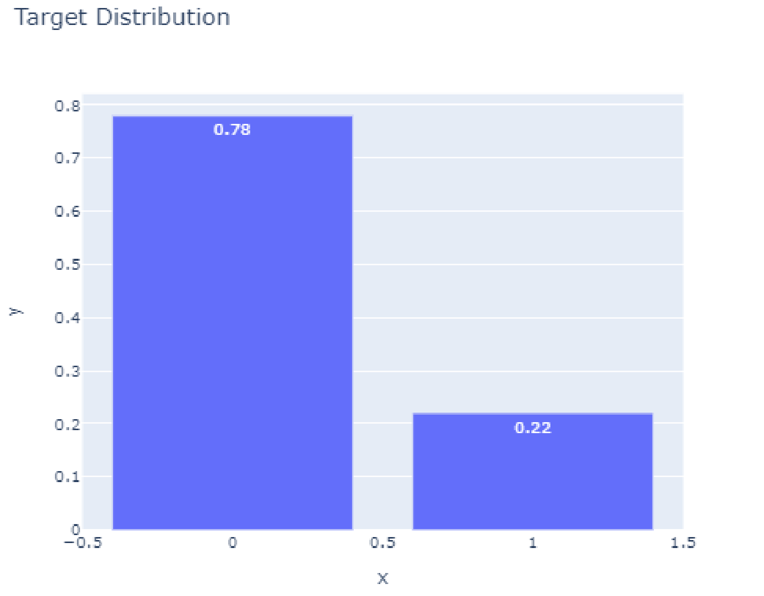
Solution Approach: Use past data from clients' payment and default history and build a ML based solution to predict risky clients who may default in the next month.

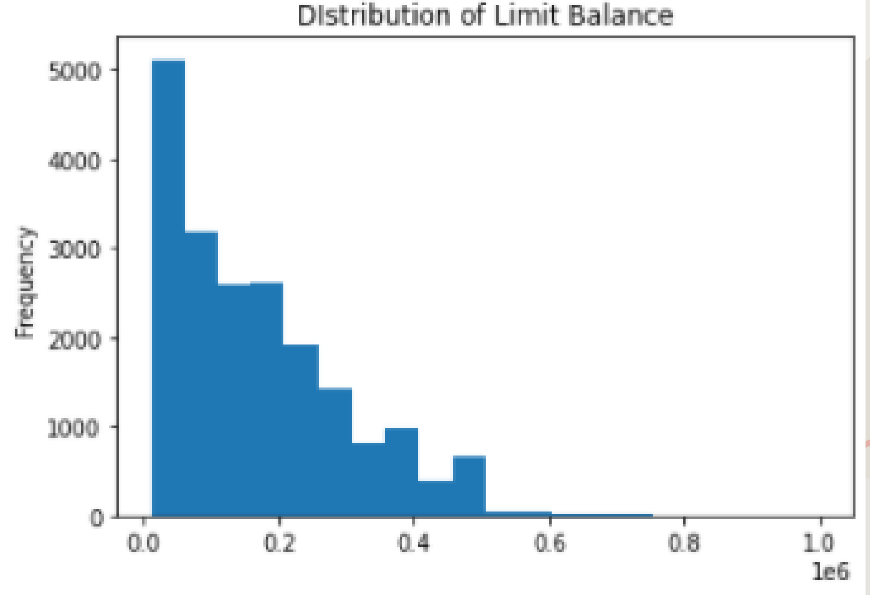
** Source: https://www.federalreserve.gov/releases/chargeoff/delallsa.htm*

Overview

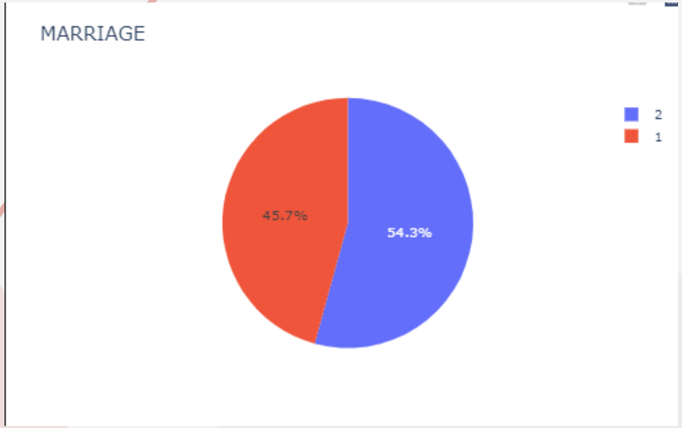
- Predictive modeling pipeline
- Interpretating the results
- Kaggle competition

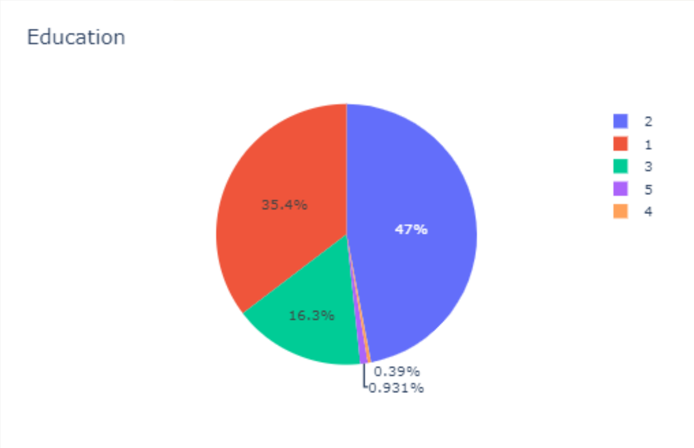
Exploratory Data Analysis

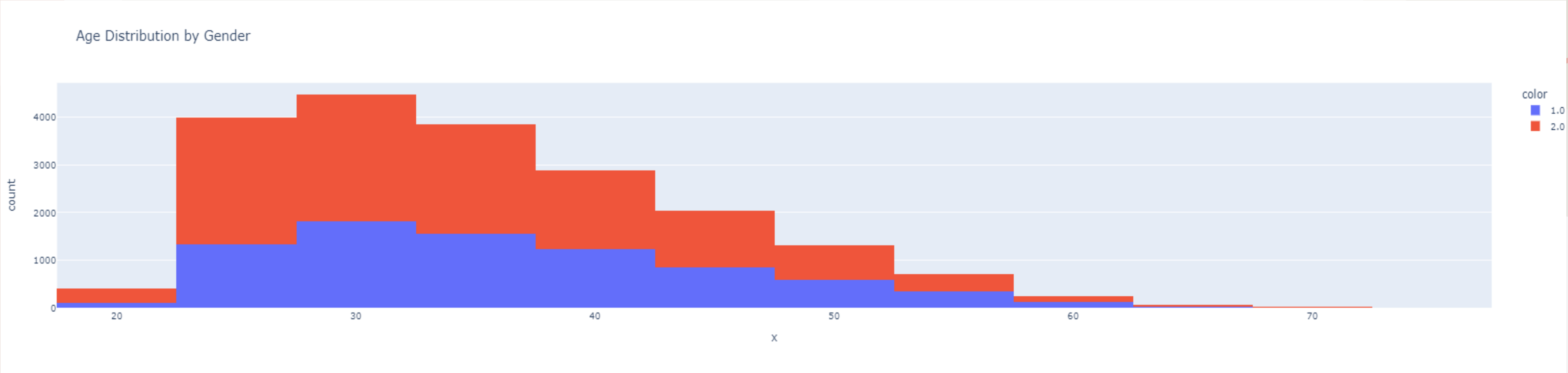




Exploratory Data Analysis







Data Preprocessing

Error Correction

- Constant variables
- Missing values
- Outliers
- Encode categorical

Feature Engineering

- Correlation test
- Mutual information
- Polynomial terms

Value Transformation

- Re-mapping
- Decision tree discretization
- Equal frequency discretization
- Equal width discretization

Value Representation

- Dummy encoding
- Incidence replacement
- WoE conversion

Feature Engineering

01

Correlation Test

02

Mutual Information

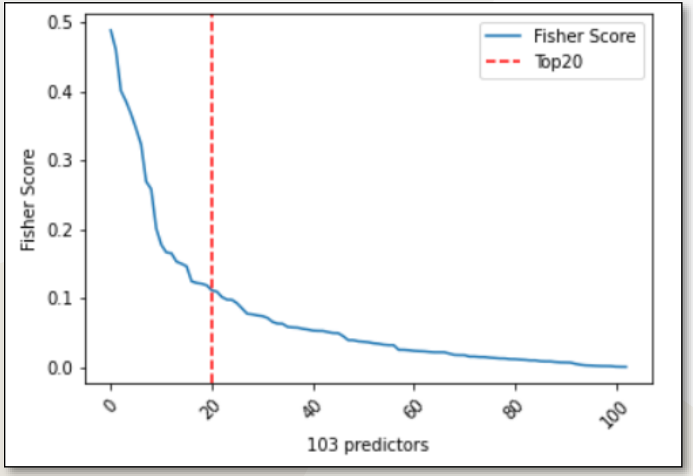
03

Polynomial Terms

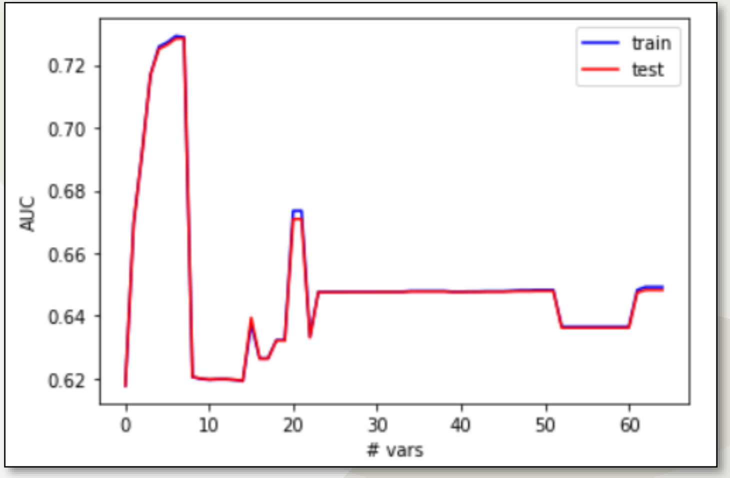
04

Dummy Encoding

Feature Selection



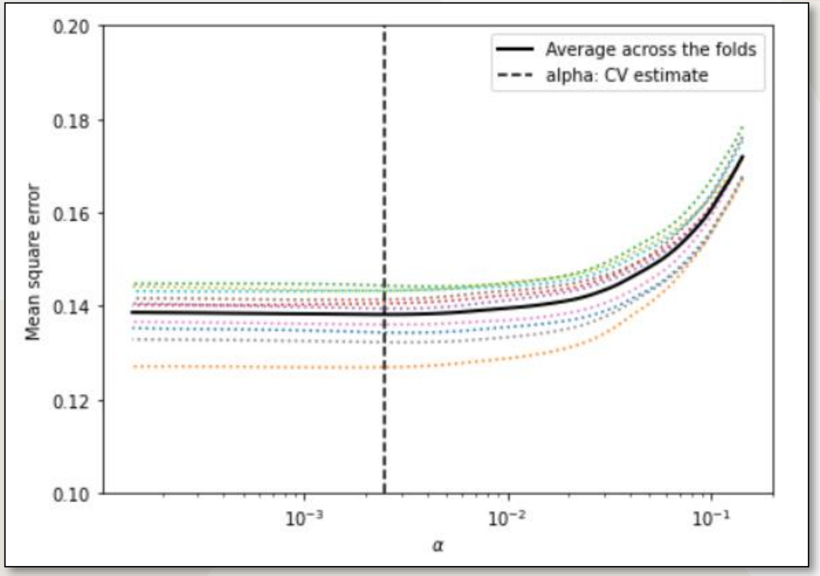
Top 20 Variables based on Fisher's Score



Scores drops and flatlines after 20+ variables, best at ~8 variables

Fisher Score

Feature Selection



Lasso

Alpha = 0.00233

Calculate coefficients and remove features with coefficient equals 0

Feature Selection

Feature Selection Method 1 (Lasso)

Select all features that had coefficients different from 0 to fit in the model

Feature Selection Method 2 (Fisher + Lasso + Manual Stepwise)

1. Select top features based on the Fisher’s Score
2. Add one more feature from lasso method and monitor the change in AUC
3. If AUC improves and there is no sign of overfitting, we keep that feature
4. Add one more feature and repeat steps 1 to 4

Selected Features

For final feature selection, we used a combination of methods that gave us the best results, these methods and the features selected using them are outlined below:

Feature	Description	Selection Method
PAY_0_4.0	Repayment Month	Fisher's Score / Lasso
PAY_2_4.0	Repayment Month	Fisher's Score / Lasso
PAY_3_4.0	Repayment Month	Fisher's Score
PAY_0_2.0	Repayment Month	Fisher's Score / Lasso
PAY_4_4.0	Repayment Month	Fisher's Score / Lasso
PAY_5_3.0	Repayment Month	Fisher's Score / Lasso
PAY_6_3.0	Repayment Month	Fisher's Score / Lasso
PAY_2_2.0	Repayment Month	Fisher's Score
Age	Age	Step wise / Lasso
Age_sq	2^{nd} degree Polynomial of Age	Step wise
Limit_Bal	Scaled Limit Balance	Step wise / Lasso
PAY_0_1.0 PAY_0_2.0 PAY_0_3.0 PAY_0_5.0, PAY_0_6.0, PAY_0_7.0, PAY_0_10.0	Dummy representation of payment months	Lasso
PAY_2_3.0, PAY_2_5.0, PAY_2_7.0, PAY_2_8.0	Dummy representation of payment months	Lasso
PAY_3_5.0,PAY_3_8.0, PAY_3_11.0	Dummy representation of payment months	Lasso

Feature	Description	Selection Method
BILL_AMT2, BILL_AMT3_na		Lasso
PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5, PAY_AMT6, PAY_AMT3_na, PAY_AMT5_na	Payment Amount of previous payment	Lasso
SEX_1.0	Dummy rep of Sex	Lasso
EDUCATION_2.0, EDUCATION_4.0, EDUCATION_5.0	Dummy rep of Education	Lasso
MARRIAGE_1.0, MARRIAGE_3.0	Dummy rep of Marriage	Lasso
AGE_na	Dummy rep - Missing Age	Lasso
PAY_4_1.0, PAY_4_3.0	Dummy rep for payments	Lasso

Model Benchmarking

	Logistic Regression	Cat Boost Classifier	LGBM Classifier	XGB Classifier	Stacked Ensemble (Auto ML)	AdaBoost Classifier	MLP Classifier	Voting + Stacked Ensemble
AUC Train	**76.9%**	**87.2%**	**84.4%**	**83.0%**	**74.6%**	**82.9%**	**84.0%**	**83%**
AUC Validation	**76.3%**	**76.8%**	**77.5%**	**77.3%**	**75.4%**	**77.5%**	**76.7%**	**78%**
AUC Test (Kaggle)	**N/A**	**78.9%**	**79.2%**	**76.07%**	**72.3%**	**N/A**	**77.9%**	**79.2%**

N/A = Not Submitted on Kaggle

** Note AUC on 5 folds cross validation*

Model Selection

Best Scoring Model based on Kaggle Test Set

Mean (Voting Ensemble + AdaBoost)

LightGBM

scikit learn

Random Forest

CatBoost

Logistic Regression

+

Model Selection

Best Scoring Model Parameters

1. LGBMClassifier(boosting_type='gbdt', random_state=1, learning_rate= 0.1, max_depth= 5, num_leaves= 10, objective = 'binary', reg_alpha= 0.8, reg_lambda= 1)

2. RandomForestClassifier(n_estimators=300)

3. CatBoostClassifier(verbose=0, learning_rate=0.009, depth=16)

4. LogisticRegression(C=2, max_iter= 1000, penalty= 'l1', solver= 'liblinear’)

5. AdaBoostClassifier(base_estimator=RandomForestClassifier(), max_depth=4, min_samples_leaf=5, n_estimators=250, learning_rate=0.01)

Hyper-parameter Tuning
Using GRID Search & Cross Validation

Logistic Regression

penalty	l1, l2
C (Inverse of regularization strength)	0.01, 0.1, 0.5, 1, 2

Gradient Boosting Decision Tree

learning rate	0.01, 0.05, 0.1
maximum tree leaves	10, 12, 15, 20
maximum tree depth	5, 8
L1 regularization term on weights	0.8, 1
L2 regularization term on weights	1, 1.2, 1.3, 1.4

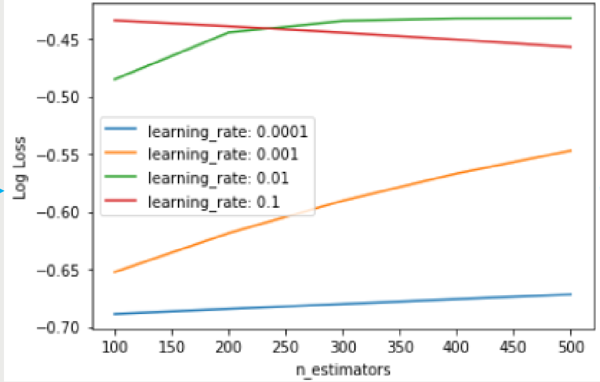
Examples of grid search parameters we used to tune our models

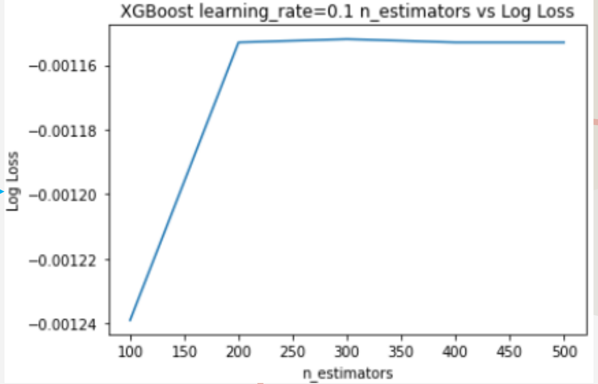
Multiple iterations of Grid Search and Cross Validation were performed to find the best parameters without over or under fitting

XGBoost

Hyper-parameter Tuning
Our Approach

1. Understand impact learning rate has on the performance
2. Monitor the learning rate and number of estimators
3. Re-visit the performance of learning rate vs number of estimators



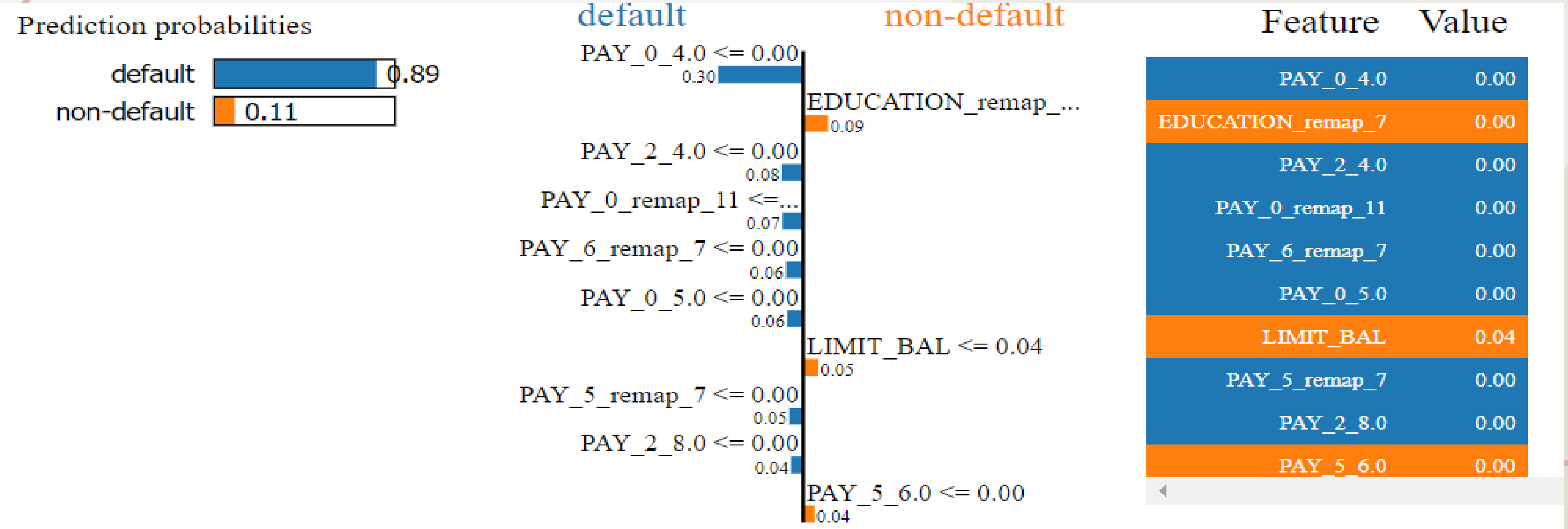


**Our approach was similar for tuning all models: tune 1 parameter → monitor its impact on other parameters → Adjust and tune next parameter*

Model Interpretation - XGBoost

Higher the value of the variables in Blue and lower the value of the variables in Orange will push the target variable towards 1

Model Interpretation - LGBM



Higher the value of the variables in Blue and lower the value of the variables in Orange will push the target variable towards 1

Conclusion

Higher the education of the clients the lower the probability of default

Lower the limit balance, lower the probability of default

Conclusion

Based on the features in our dataset we noticed the following:

- Almost all models (except Logistic Regression) tend to overfit very easily if we use a lot of features.
- Logistic Regression was one of the best models in terms of Train/Test AUC stability of 76% AUC.
- Boosting models performed the best and even better when combined using voting and stacking methods.
- Although an ensemble performed better, we would still recommend using Logistic Regression model as the difference in the performance is only 3% and Logistic Regression model is very fast, simple and easy to explain compared to an ensemble.

Kaggle Competition

42 submissions for HIA

Sort by Select...

All Successful Selected

Submission and Description	Private Score	Public Score	Use for Final Score
thelastone.csv an hour ago by Inder Rana add submission details	0.79252	0.79284	✓

Recommendations

- Banks can adapt the customer’s limit based on default predictions
- Lending institution can send proactive notifications to the clients who are probable to default next month.
- Banks can offer short term loan to the clients with probability of default in order to pay for the credit card and avoid high interest rates.

THANKS!

References

- ML Pipeline template and functions: Minh Phan – IESEG School of Management
- Presentation theme, icons and graphics: Slidesgo, IconsFlaticon, Freepik
- https://scikit-learn.org/stable/auto_examples/linear_model/plot_lasso_model_selection.html
- https://www.geeksforgeeks.org/ensemble-methods-in-python/
- https://en.wikipedia.org/wiki/Mutual_information
- https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/tkdd11.pdf
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html
- https://stackoverflow.com/questions/32210569/using-gridsearchcv-with-adaboost-and-decisiontreeclassifier
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
- https://catboost.ai/en/docs/concepts/python-usages-examples
- https://medium.com/@juniormiranda_23768/ensemble-methods-tuning-a-xgboost-model-with-scikit-learn-54ff669f988a
- https://towardsdatascience.com/lime-how-to-interpret-machine-learning-models-with-python-94b0e7e4432e