Master
Project Management
and Data Science

htw

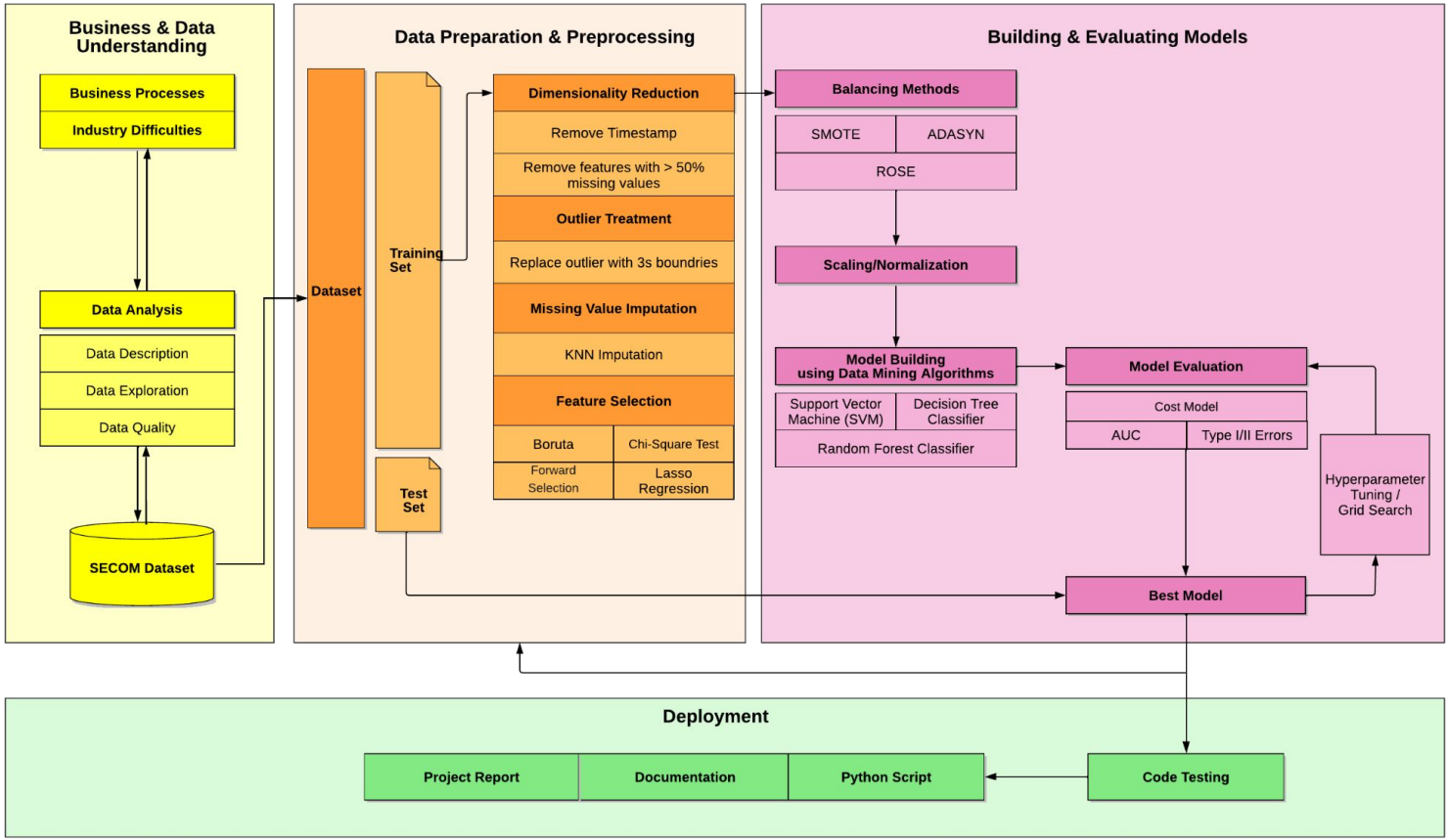
SECOM ANALYTICS

TEAM 3

Gupta, Himansha
Pomay Polat, Ekin
Dsouza, Rashmi Carol
Pham, Quynh Dinh Hai

TABLE OF CONTENTS

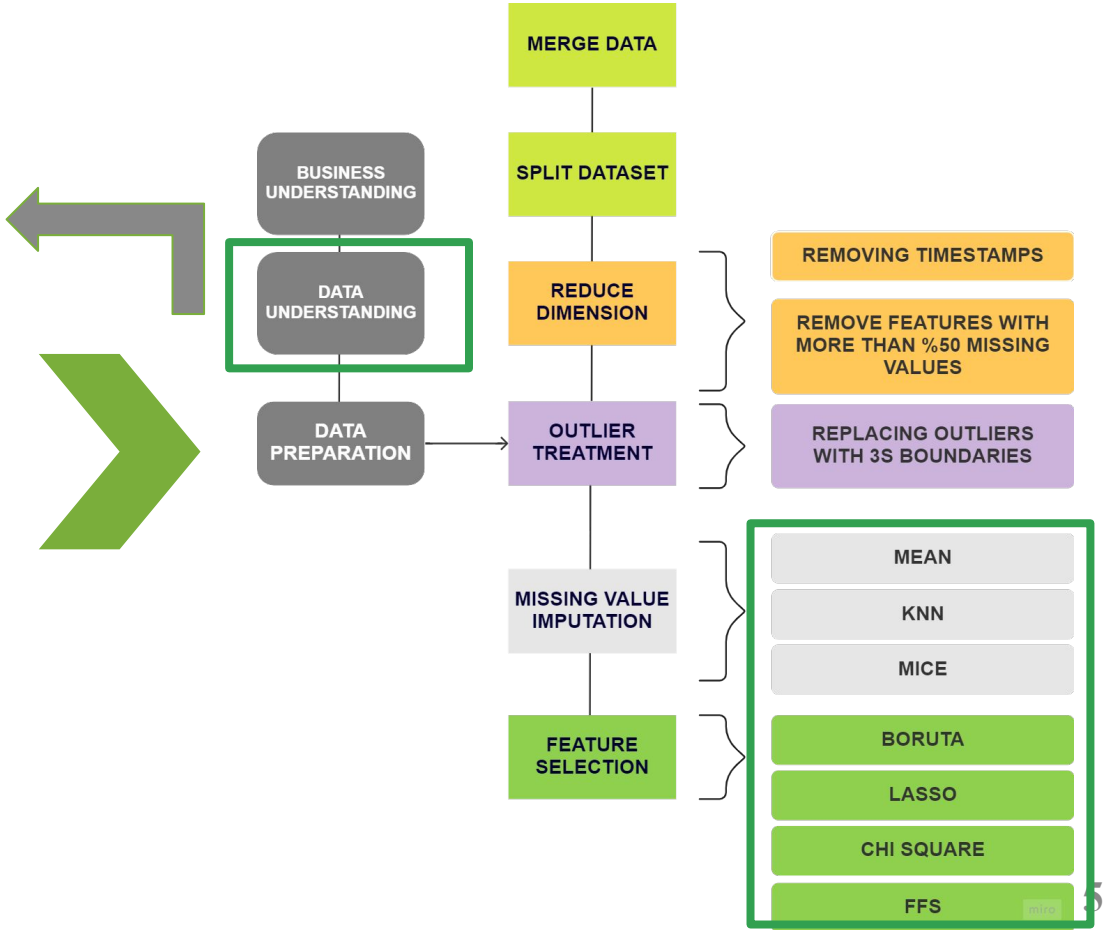
WORKFLOW


Business & Data Understanding
Business Processes
Industry Difficulties
Data Analysis
Data Description
Data Exploration
Data Quality
SECOM Dataset
Data Preparation & Preprocessing
Dataset
Training Set
Test Set
Dimensionality Reduction
Remove Timestamp
Remove features with > 50% missing values
Outlier Treatment
Replace outlier with 3s boundries
Missing Value Imputation
KNN Imputation
Feature Selection
Boruta
Chi-Square Test
Forward Selection
Lasso Regression
Building & Evaluating Models
Balancing Methods
SMOTE
ADASYN
ROSE
Scaling/Normalization
Model Building using Data Mining Algorithms
Support Vector Machine (SVM)
Decision Tree Classifier
Random Forest Classifier
Model Evaluation
Cost Model
AUC
Type I/II Errors
Hyperparameter Tuning / Grid Search
Best Model
Deployment
Project Report
Documentation
Python Script
Code Testing

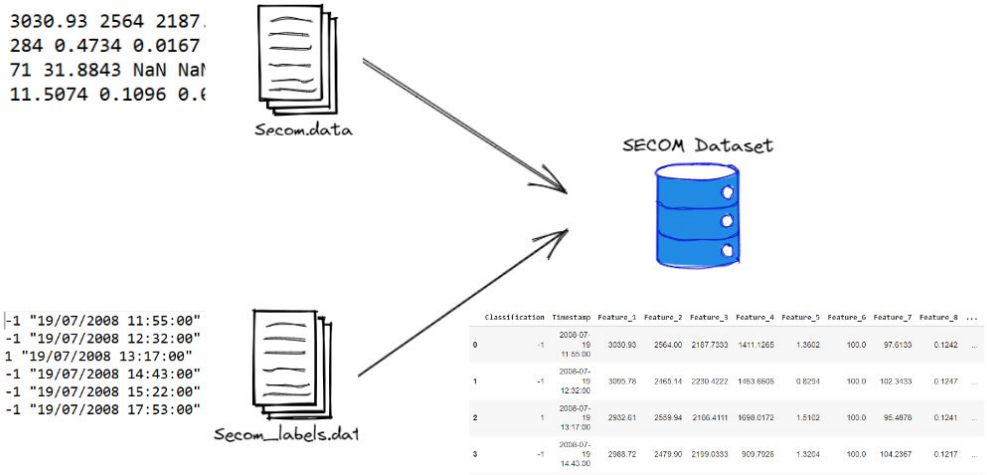
DATA UNDERSTANDING & PREPROCESSING

DATA PREPROCESSING

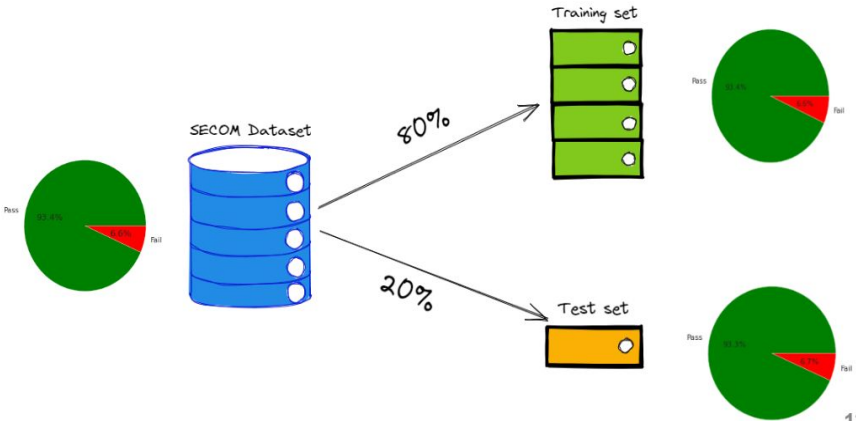
Problems with the dataset
Imbalance data between pass & fail test results
Missing values
Timestamp does not yield meaningful insight to build model
Many highly correlated features
Outliers



DATA PREPARATION



MERGING DATA



SEPARATE TRAINING & TEST DATA

DATA PREPARATION

REDUCING DIMENSIONALITY

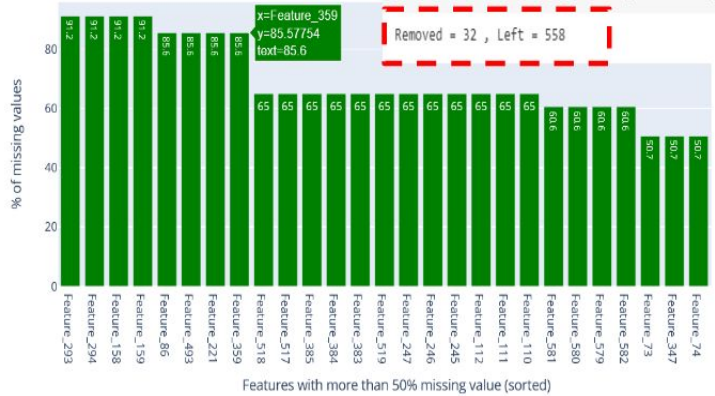
1. Remove the Timestamp feature
2. Remove features with missing values

Threshold:
Features with more than 50% missing value

```
def percent(dataframe, threshold):
    columns = dataframe.columns[(dataframe.isna().sum()/dataframe.shape[1])>threshold]
    return columns.tolist()

na_columns = percent(X_train, 0.5)
X_train_na = X_train.drop(na_columns, axis=1)
X_test_na = X_test.drop(na_columns, axis=1)
n_features1 = X_train_na.shape[1]
print(f'Removed = {len(na_columns)} , Left = {n_features1}')
```

```
Removed = 32 , Left = 558
```



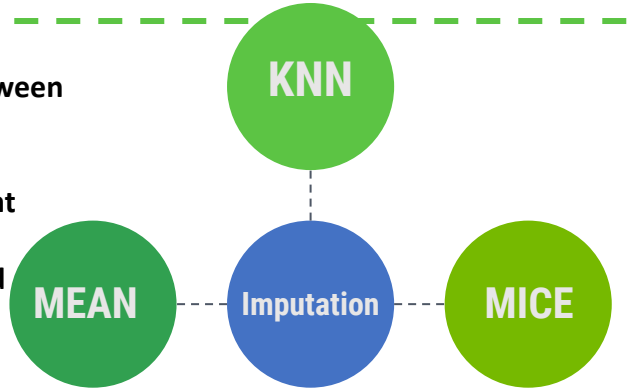
OUTLIER TREATMENT

Replace outliers with 3s boundaries

MISSING VALUE IMPUTATION

Which Imputation Method did we choose?

- Replacing missing values with actual occurring values or very similar values while calculating the average.
- Better imputer for highly dimensional dataset
- non-parametric approach, less prone to misspecifications.



- Does not preserve relationships between variables such as correlations.
- Takes only single feature into account
- Reduces the variance of the imputed variables.

- Time Consuming
- Performance depends on the size of the dataset

FEATURE SELECTION

29 Features left

CHI SQUARE

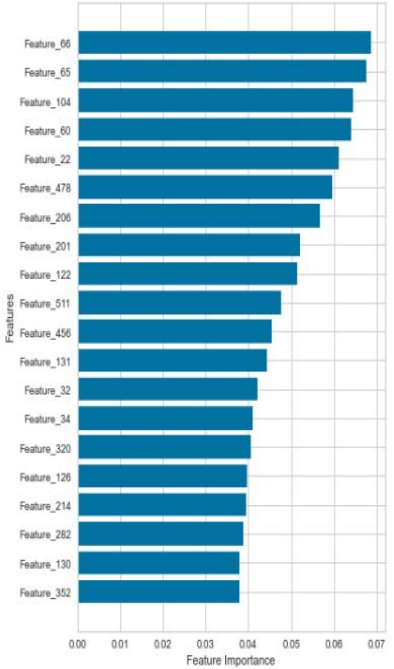
29 Features left — LASSO — Feature Selection — BORUTA — **20** Features left

FFS

25 Features left

We compared AUC and ROC

Random Forest Feature Importance



TRIAL AND ERROR PROCESS

SELECTION AND ELIMINATION

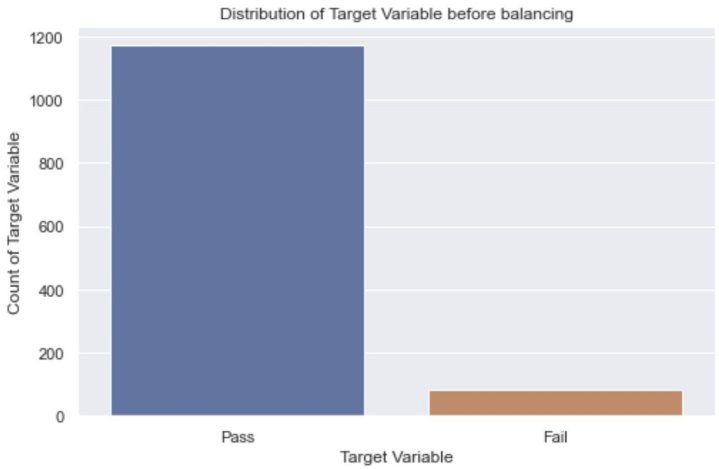
	Type I error	Type II error	F1 score	AUC	Cost	Remaining features
KNN - Boruta - SMOTE	11	74	72	70	$ 98.75	12
KNN - Boruta - SMOTE-ENN	10	88	69	73	$ 110.50	12
KNN - Boruta - SMOTE-TOMEK	11	67	75	68	$ 91.75	12
KNN - Boruta - ADASYN	12	79	71	63	$ 106.00	12
KNN - Boruta - ROSE	10	53	79	72	$ 75.50	12
KNN - Chi-Square Test - SMOTE	13	43	82	67	$ 72.25	29
KNN - Chi-Square Test - SMOTE-ENN	11	61	77	69	$ 85.75	29
KNN - Chi-Square Test - SMOTE-TOMEK	14	41	82	68	$ 72.50	29
KNN - Chi-Square Test - ADASYN	12	42	82	68	$ 69.00	29
KNN - Chi-Square Test - ROSE					$ -	29
KNN - Lasso Regression - SMOTE	14	25	87	73	$ 56.50	29
KNN - Lasso Regression – SMOTE-ENN	11	62	76	72	$ 86.75	29
KNN - Lasso Regression – SMOTE-TOMEK	16	20	87	72	$ 56.00	29
KNN - Lasso Regression - ADASYN	14	34	84	71	$ 65.50	29
KNN - Lasso Regression - ROSE					$ -	29
KNN - Boruta SHAP - SMOTE					$ -	
KNN - Forward Selection - SMOTE	13	30	86	70	$ 59.25	15
KNN - Forward Selection - SMOTE-ENN	9	49	81	73	$ 69.25	15
KNN - Forward Selection - SMOTE-TOMEK	13	30	86	72	$ 59.25	15
KNN - Forward Selection - ADASYN	14	30	85	73	$ 61.50	15
KNN - Forward Selection - ROSE	11	28	87	75	$ 52.75	15
KNN - RFE - SMOTE	14	45	81	73	$ 76.50	15
KNN - RFE - SMOTE-ENN	11	77	71	70	$ 101.75	15

	Type I error	Type II error	F1 score	AUC	Cost	Remaining Features
MICE - Boruta - SMOTE	11	76	68	72	$100.75	12
MICE - Boruta - SMOTE-ENN	9	84	74	70	$104.25	12
MICE - Boruta - SMOTE-TOMEK	11	78	71	69	$102.75	12
MICE - Boruta - ADASYN	11	82	70	63	$106.75	12
MICE - Boruta - ROSE	9	53	80	71	$ 73.25	12
MICE - Chi-Square Test - SMOTE	14	43	81	65	$ 74.50	29
MICE - Chi-Square Test - SMOTE-ENN	13	61	76	67	$ 90.25	29
MICE - Chi-Square Test - SMOTE-TOMEK	15	46	80	66	$ 79.75	29
MICE - Chi-Square Test - ADASYN	11	42	83	67	$ 66.75	29
MICE - Chi-Square Test - ROSE					$ -	29
MICE - Lasso Regression - SMOTE	14	26	87	74	$ 57.50	29
MICE - Lasso Regression - SMOTE-ENN	10	64	76	71	$ 86.50	29
MICE - Lasso Regression - SMOTE-TOMEK	14	26	87	74	$ 57.50	29
MICE - Lasso Regression - ADASYN	15	36	83	74	$ 69.75	29
MICE - Lasso Regression - ROSE					$ -	29
MICE - Boruta SHAP - SMOTE					$ -	6
MICE - Forward Selection - SMOTE	12	29	86	73	$ 56.00	
MICE - Forward Selection - SMOTE-ENN	10	46	82	70	$ 68.50	
MICE - Forward Selection - SMOTE-TOMEK	14	27	86	71	$ 58.50	
MICE - Forward Selection - ADASYN	12	32	85	72	$ 59.00	
MICE - Forward Selection - ROSE	10	35	85	76	$ 57.50	

There is ***no best feature selection method***. Instead, we have discovered what works best for the specific problems related to this Dataset using careful systematic experimentation/trial and error method. We tried a range of different models fit on different subsets of features chosen via different statistical measures and ***discovered the ones works best***.

BALANCING METHODS

BALANCING METHODS



Problem with Imbalance Dataset

Standard learners are often biased towards the majority class. The reason is because these classifiers attempt to reduce overall quantities such as error rate, not taking the data distribution into consideration, therefore, tend to bias performance towards the majority class.

Solutions:

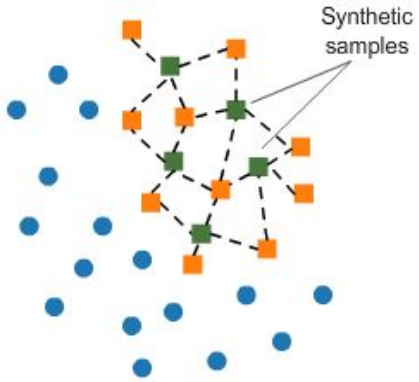
Re-sampling provides a simple way of biasing the generalization process. It can do so by:

- Generating synthetic samples accordingly biased
- Controlling the amount and placement of the new samples

Methods:

- Synthetic Minority Oversampling Technique (SMOTE)
- Adaptive Synthetic (ADASYN)
- Random Over-Sampling Examples (ROSE)

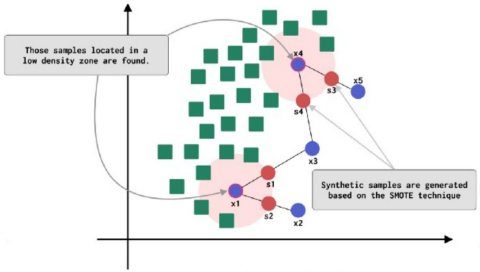
BALANCING METHODS



SMOTE

Synthetic Minority Oversampling Technique (SMOTE)

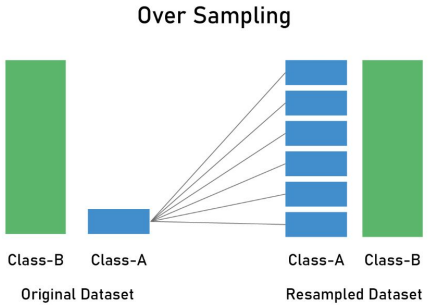
It combines Informed Oversampling of the minority class with Random Undersampling of the majority class.



ADASYN

Adaptive Synthetic (ADASYN)

It uses a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn

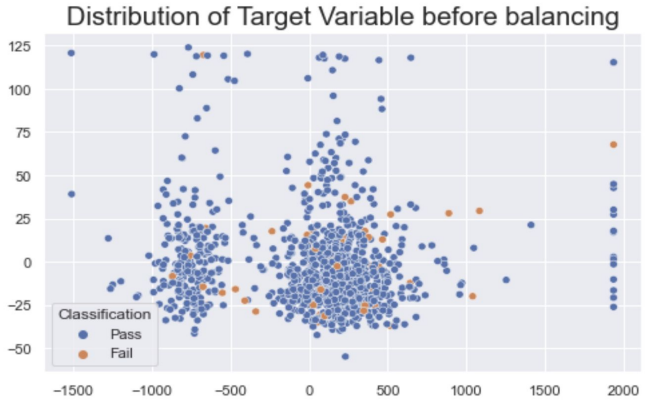


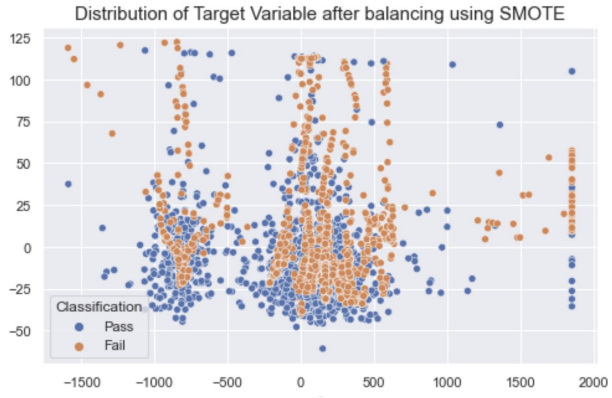
ROSE

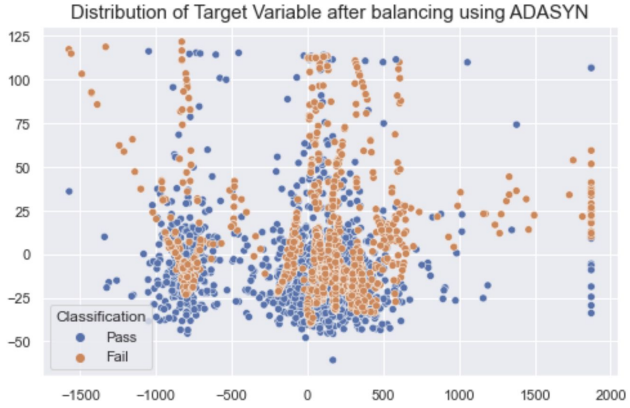
Random Over-Sampling Examples (ROSE)

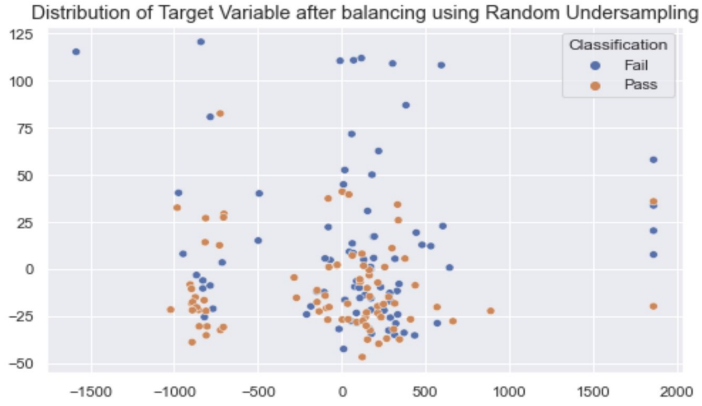
It is a bootstrap-based technique which aids the task of binary classification in the presence of rare classes.

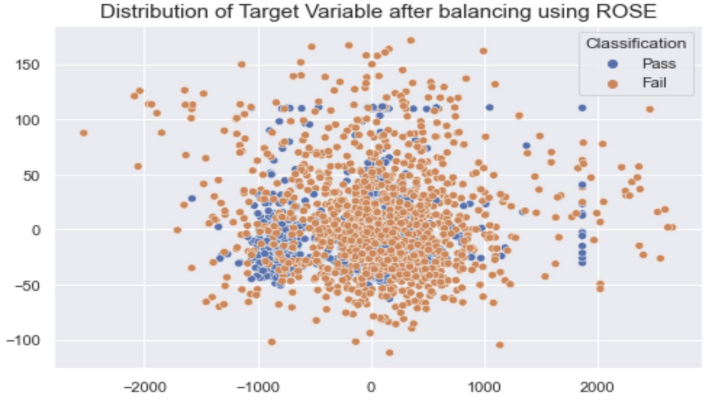
BALANCING METHODS





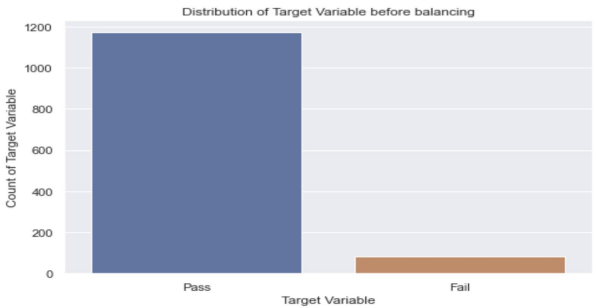




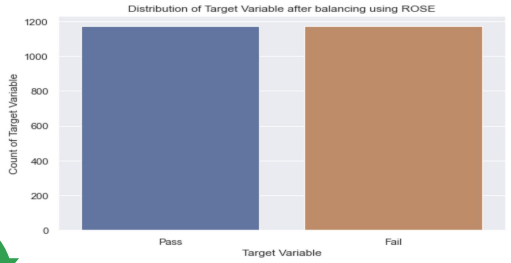


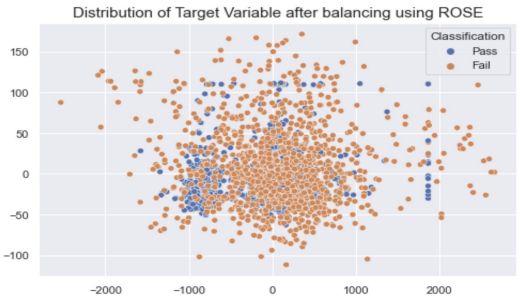
BALANCING METHOD

Why we chose a certain balancing method?



ROSE





SCALING

SCALING

We are almost ready to fit the model, but…

Features have different Dimensions…

We applied scaling in KNN. Scaling is recommended

KNN

Since the approach relying on measuring distances

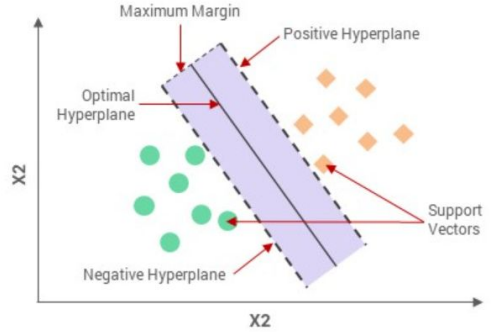
SCALING needs RESCALING

To evaluate the results on the same scale, the final model is applied to generate the predictions.

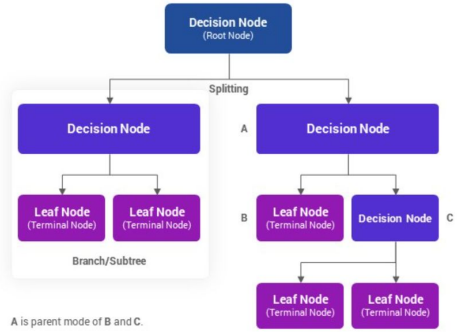
TEMPORARY SCALING

MACHINE LEARNING MODELS

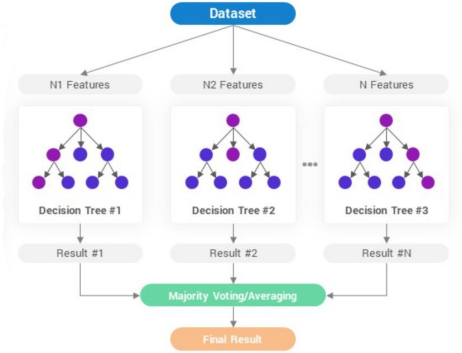
MACHINE LEARNING MODELS



Support Vector Machine (SVM)



Decision Tree Classifier



Random Forest Classifier

Support Vector Machine (SVM)

It creates the best line or decision boundary (hyperplane) that can segregate n-dimensional space into classes so that new data point can be easily put in correct category in the future.

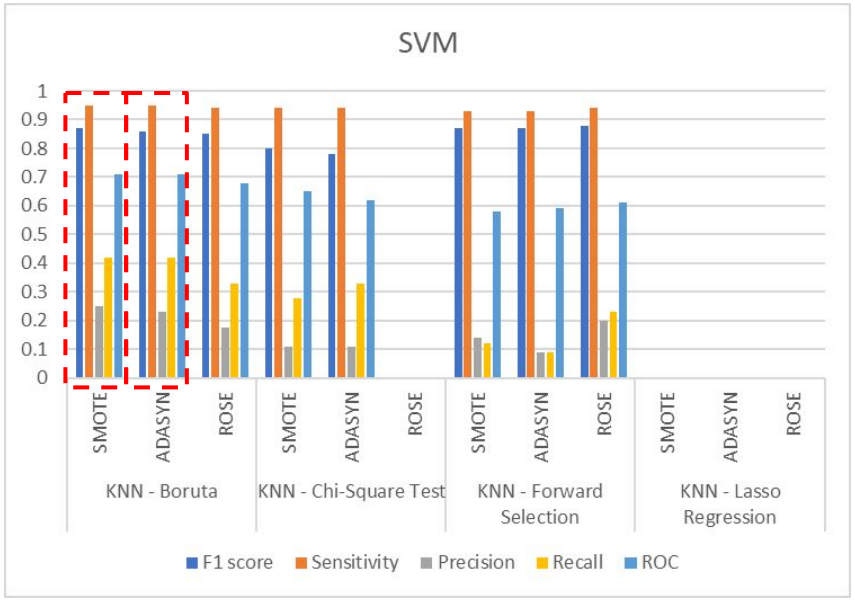
Decision Tree Classifier

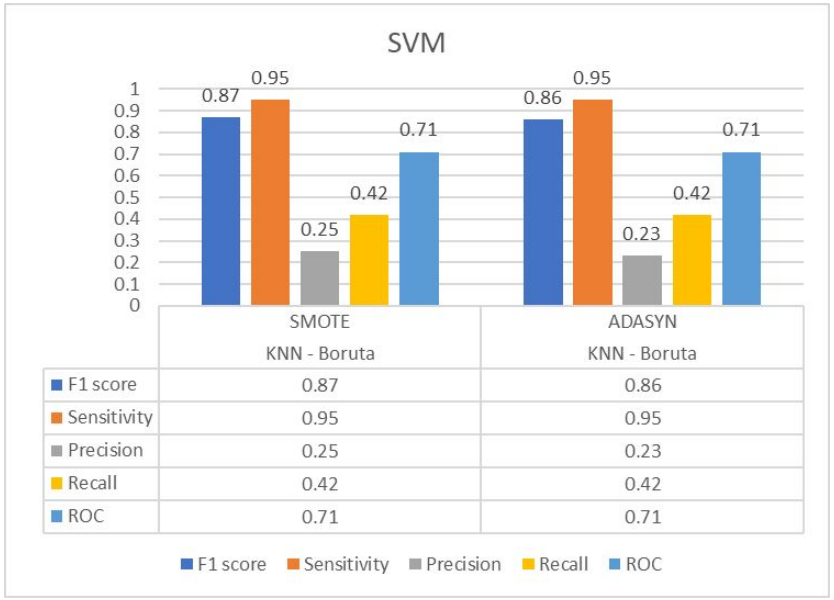
It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

Random Forest Classifier

It combines hundreds of decision trees and trains each decision tree on a different sample of the observations. The final predictions are made by averaging the predictions of each individual tree.

RESULTS WITH SVM

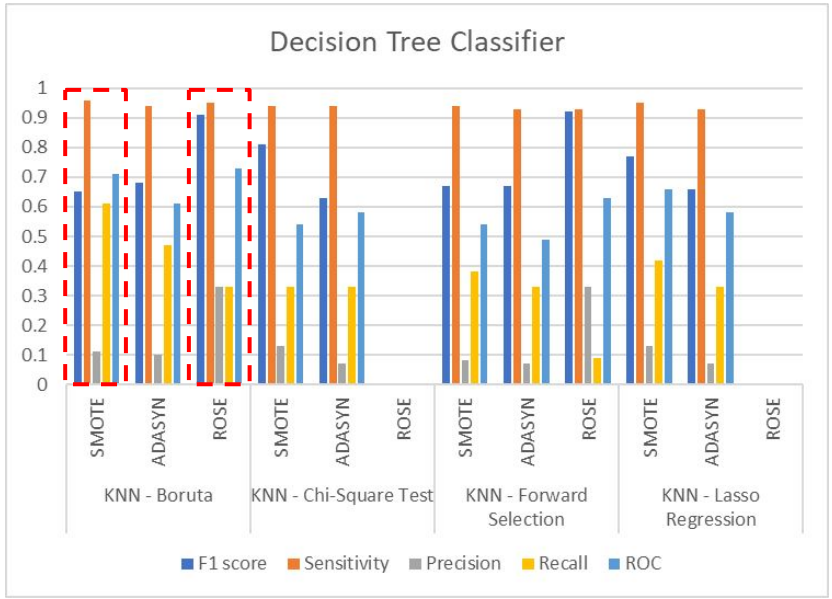


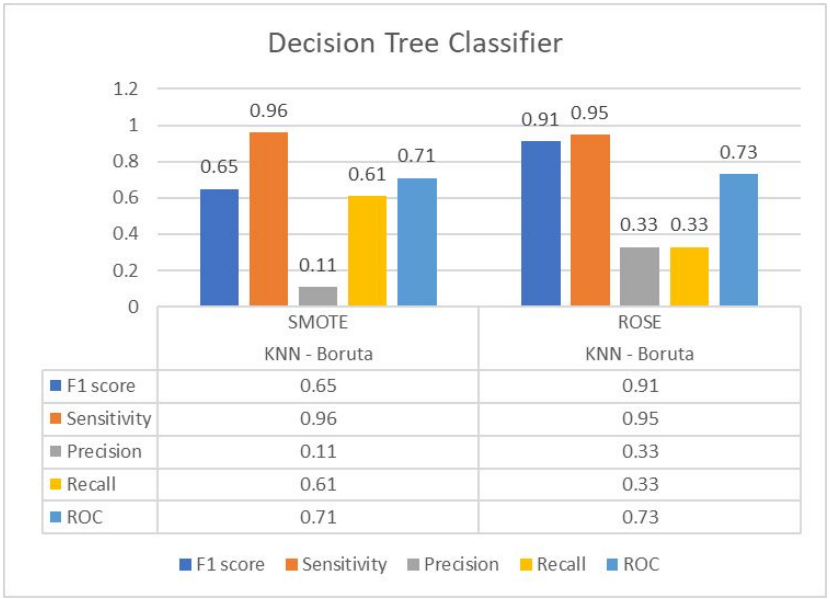


Best combinations in terms of accuracy:

- KNN-Boruta-SMOTE
- KNN-Boruta-ADASYN

RESULTS WITH DECISION TREE CLASSIFIER

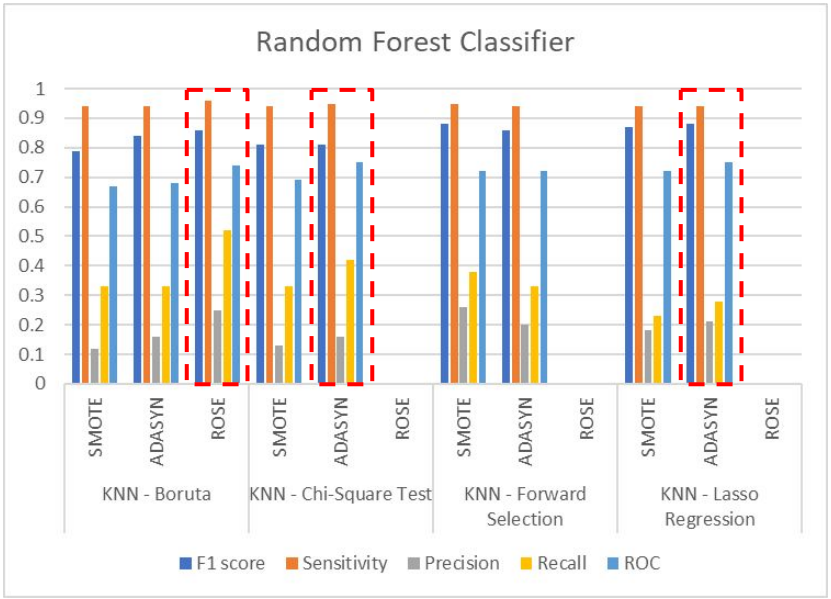


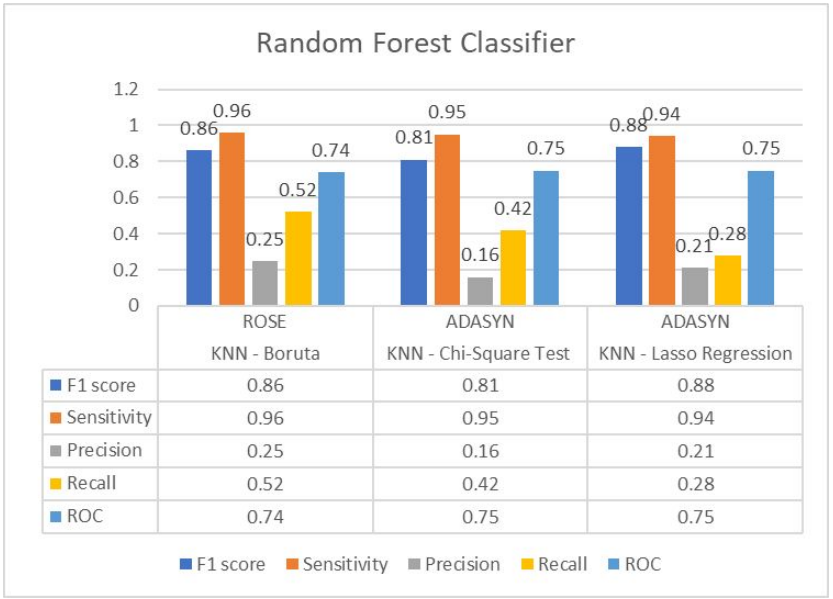


Best combinations in terms of accuracy:

- KNN-Boruta-SMOTE
- KNN-Boruta-ROSE

RESULTS WITH RANDOM FOREST CLASSIFIER





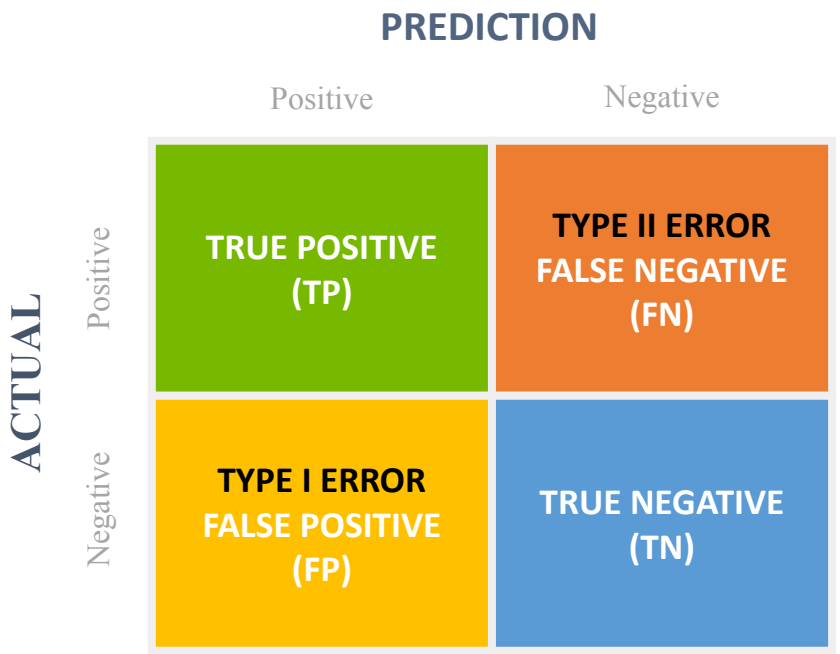
	ROSE KNN - Boruta	ADASYN KNN - Chi-Square Test	ADASYN KNN - Lasso Regression
F1 score	0.86	0.81	0.88
Sensitivity	0.96	0.95	0.94
Precision	0.25	0.16	0.21
Recall	0.52	0.42	0.28
ROC	0.74	0.75	0.75

Best combinations in terms of accuracy:

- KNN-Boruta-SMOTE
- KNN-Chi-Square Test-ADASYN
- KNN-Lasso Regression-ADASYN

EVALUATION PROCESS

CONFUSION MATRIX



CONFUSION MATRIX

Positive = **Pass** classification
Negative = **Fail** classification

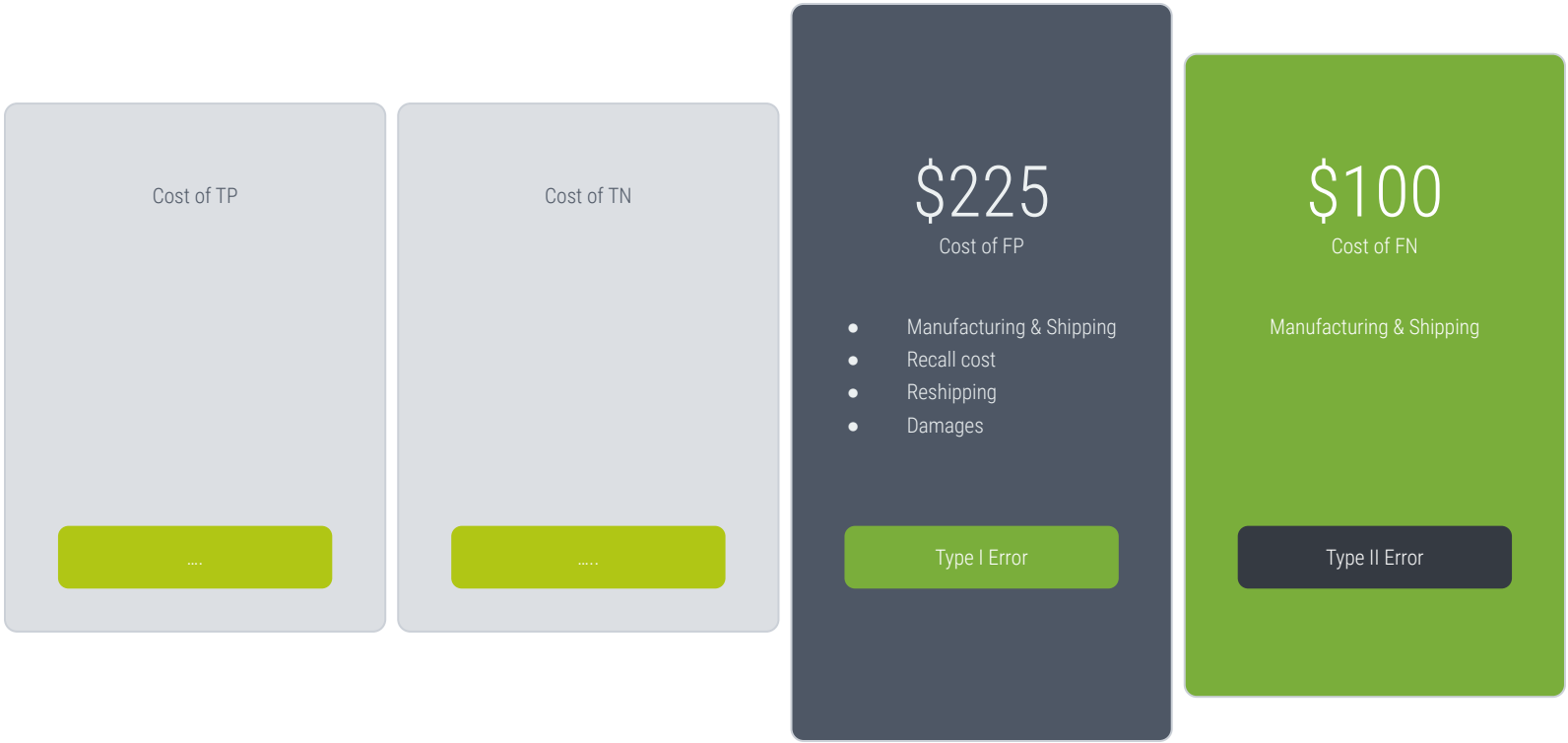
True Positive: Pass classification correctly identified as Pass

True Negative: Fail classification correctly identified as Fail

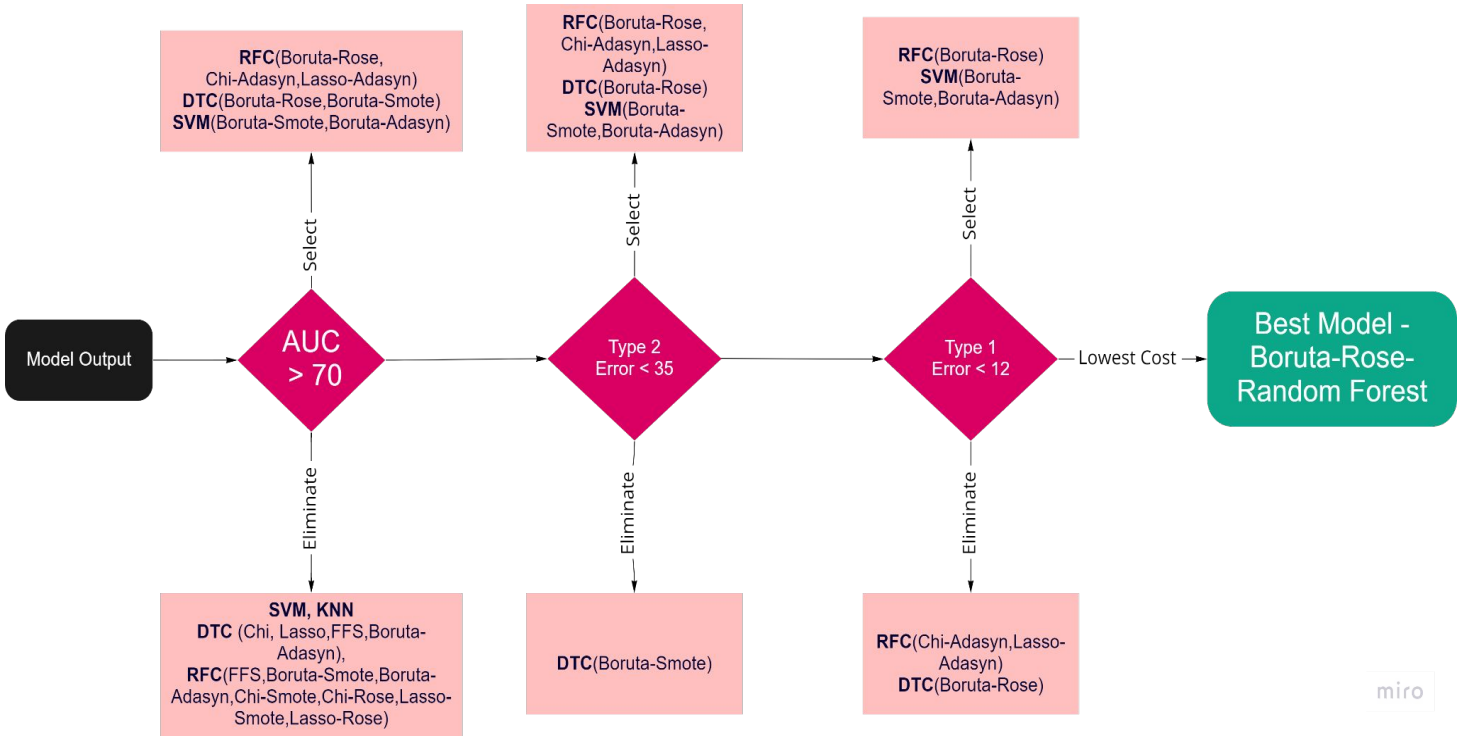
False Positive: Pass classification identified as Fail

False Negative: Fail classification identified as Pass

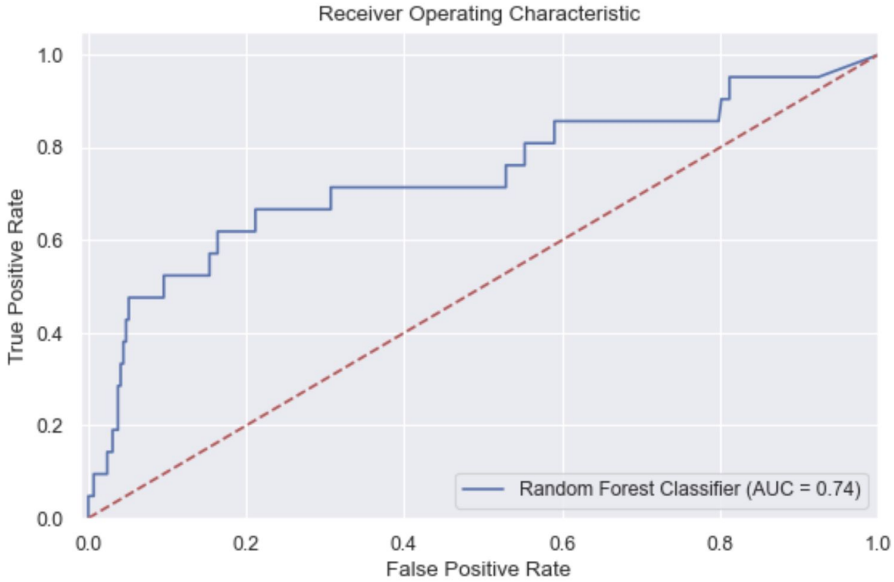
COST MODEL


Cost of TP
....
Cost of TN
.....
$225
Cost of FP
Manufacturing & Shipping
Recall cost
Reshipping
Damages
Type I Error
$100
Cost of FN
Manufacturing & Shipping
Type II Error

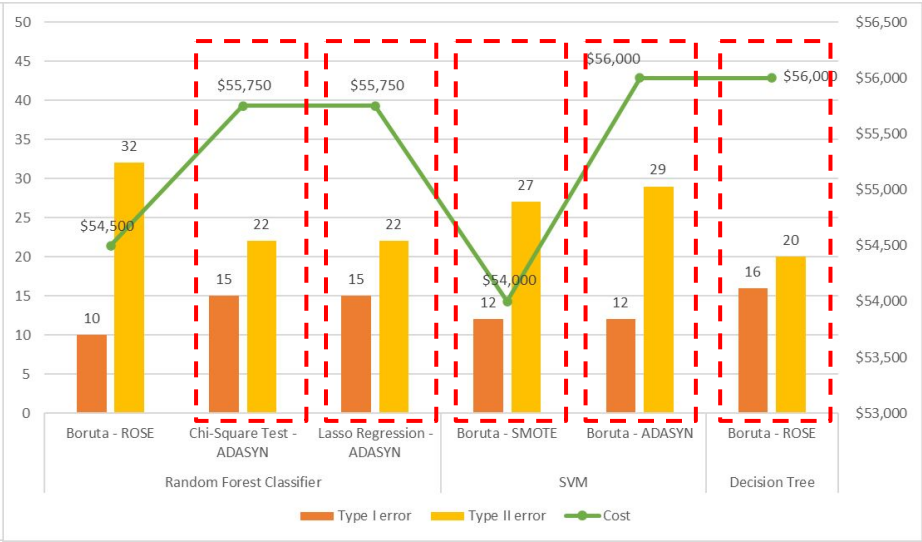
ELIMINATION OF MODELS



RANDOM FOREST CLASSIFIER



RESULTS WITH COST MODEL



Type 2 Error < 35

Lowest Type 1 Error

Best Model: KNN-Boruta-ROSE using Random Forest Classifier

- Type I error: 10
- Type II error: 32
- Cost: $54,500

MODEL OPTIMISATION

MODEL OPTIMISATION

MODEL EVALUATION

Error Function

An error function serves to evaluate the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model

Decision Process

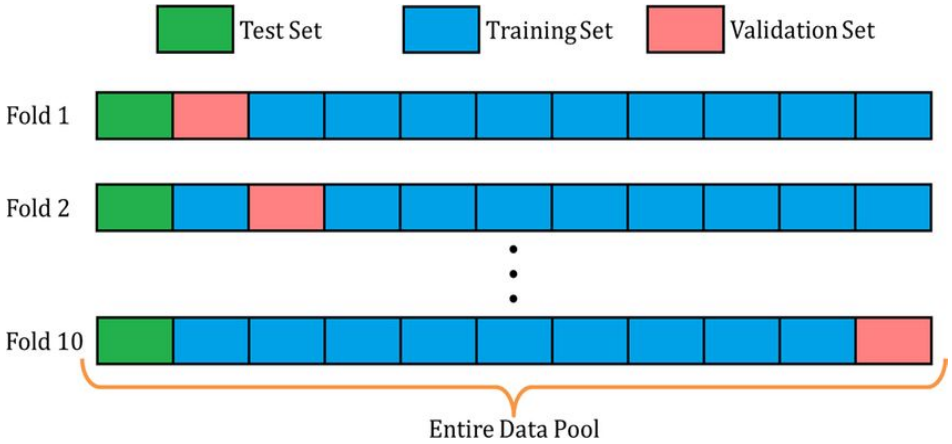
Machine Learning algorithms are used to make a prediction or classification. Based on the input data, the algorithm will produce an estimate about a pattern in the data

Model Optimisation Process

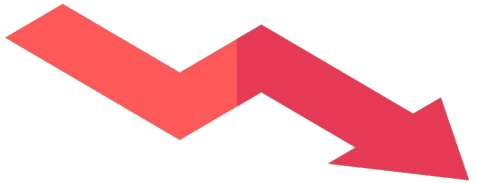
If the model can fit better to the data points in the training set, weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this evaluate and optimize process, updating weights autonomously until a threshold of accuracy has been met

CROSS VALIDATION

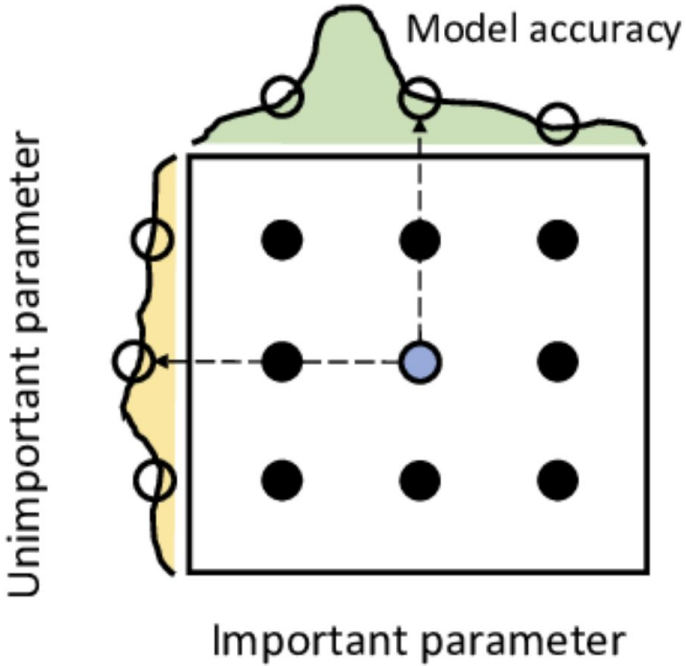
For Cross Validation we have used a **10-fold** approach in which the data is split randomly in 10 subsets that have the same number of samples. The steps described in the next subsections are repeated 10 times and each time the testing data will be one distinct fold from the set of the 10 folds and the training data will consist of the remaining 9 folds.



REDUCES OVERFITTING PROBLEM



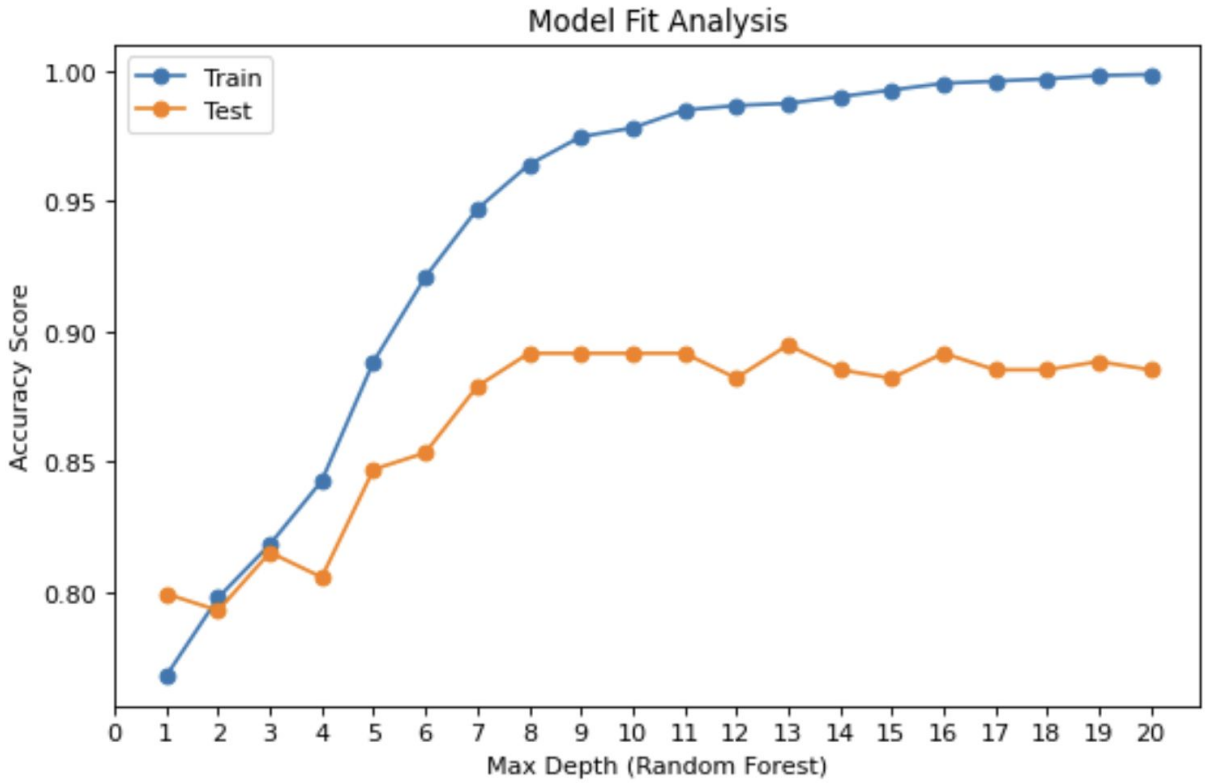
HYPERPARAMETER TUNING WITH GRID SEARCH CV



GridSearchCV

```
GridSearchCV(cv=10,
             estimator=RandomForestClassifier(n_jobs=-1, random_state=42),
             param_grid={'criterion': ['gini', 'entropy', 'log_loss'],
                         'max_depth': [4, 5, 6, 7, 8, 9, 10],
                         'max_features': ['sqrt', 'log2', 'auto', None]},
             scoring='roc_auc', verbose=1)
```

HYPERTUNING FOR MAXIMUM DEPTH



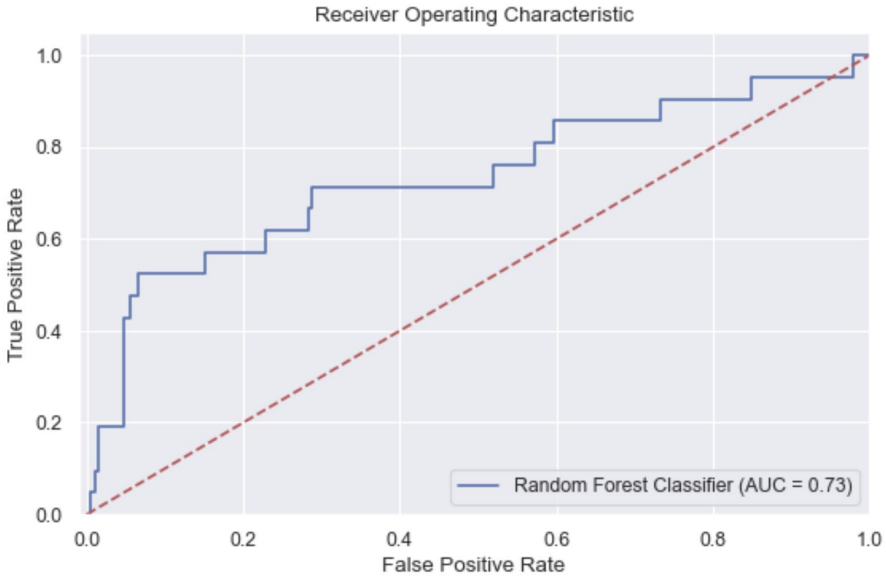
RESULTS

Master
Project Management
and Data Science

FINAL RESULT (KNN - Boruta - ROSE with Random Forest Classifier)

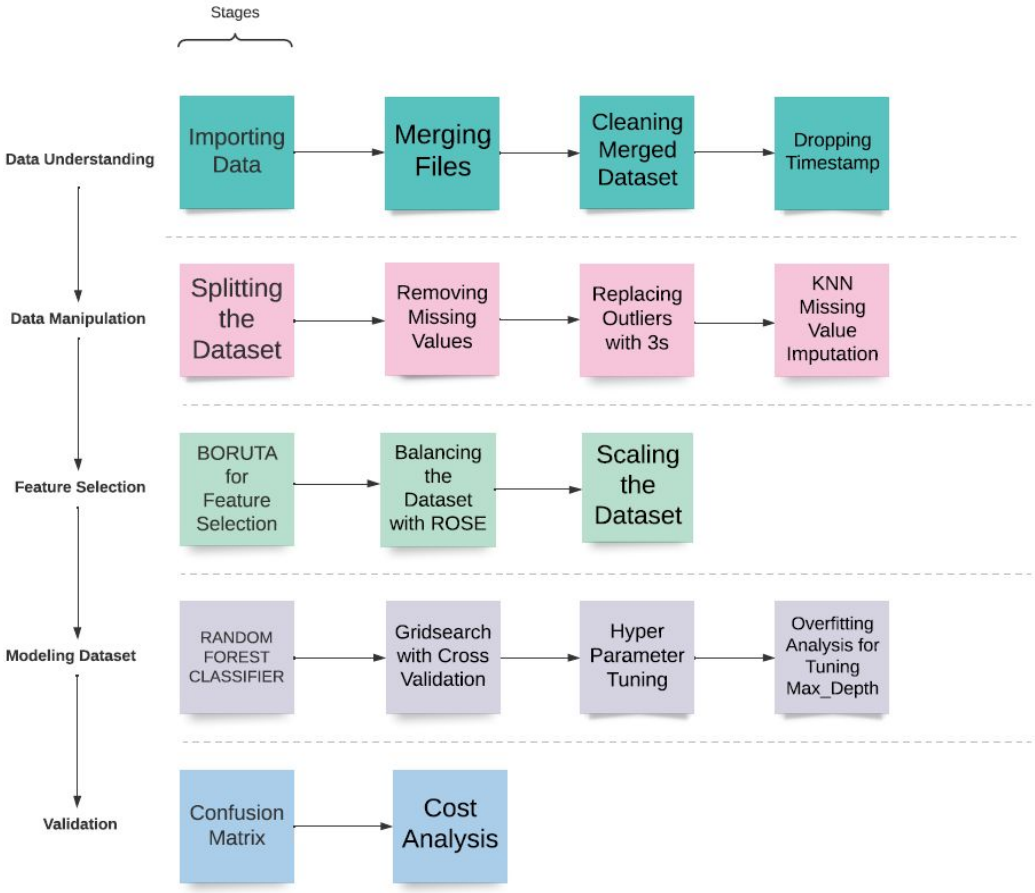
CONFUSION MATRIX

AUC



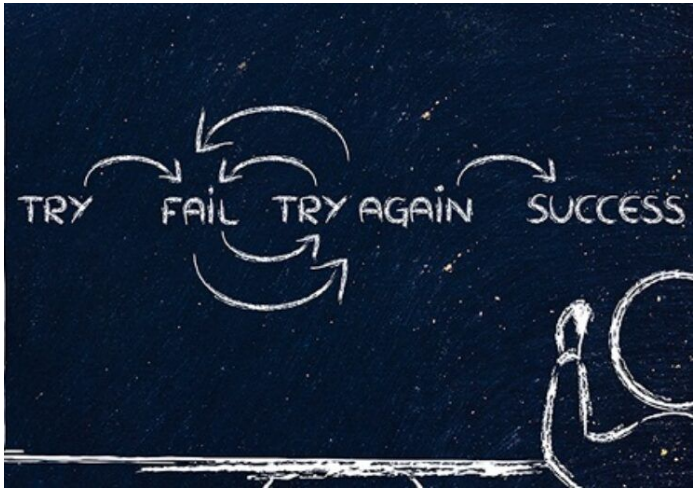
From $ 54500 ↓ **$ 46500**

SUMMARY


Stages
Data Understanding
Importing Data
Merging Files
Cleaning Merged Dataset
Dropping Timestamp
Data Manipulation
Splitting the Dataset
Removing Missing Values
Replacing Outliers with 3s
KNN Missing Value Imputation
Feature Selection
BORUTA for Feature Selection
Balancing the Dataset with ROSE
Scaling the Dataset
Modeling Dataset
RANDOM FOREST CLASSIFIER
Gridsearch with Cross Validation
Hyper Parameter Tuning
Overfitting Analysis for Tuning Max_Depth
Validation
Confusion Matrix
Cost Analysis

Lesson Learned and Best Practices

1. Using CRISP DM
2. Complete Data
3. Outlier treatment and imputation of data
4. Balancing the data
5. Scaling the data
6. Business Understanding and defining model
7. Model Quality



Master
Project Management
and Data Science

Vielen Dank!

Gupta, Himansha
Himansha.Gupta@student.htw-berlin.de

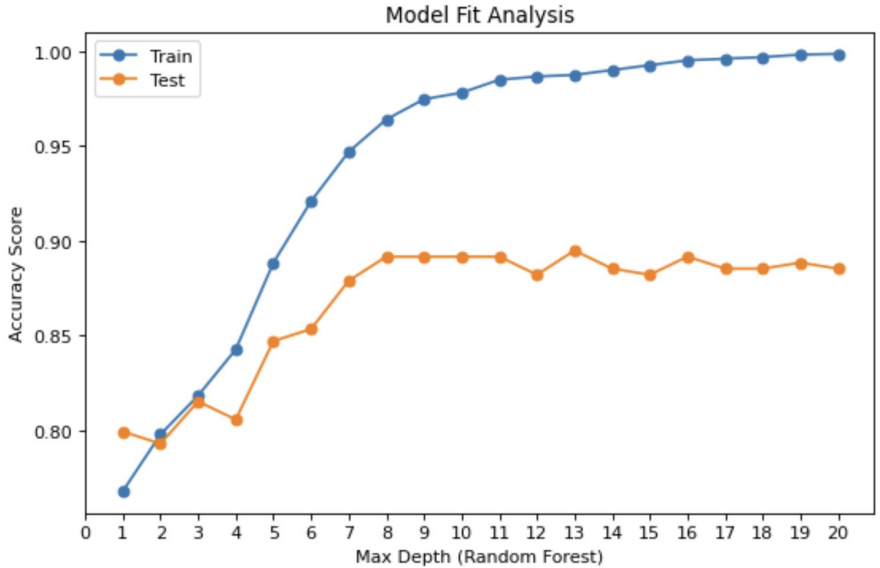
Pomay Polat, Ekin
Ekin.PomayPolat@student.htw-berlin.de

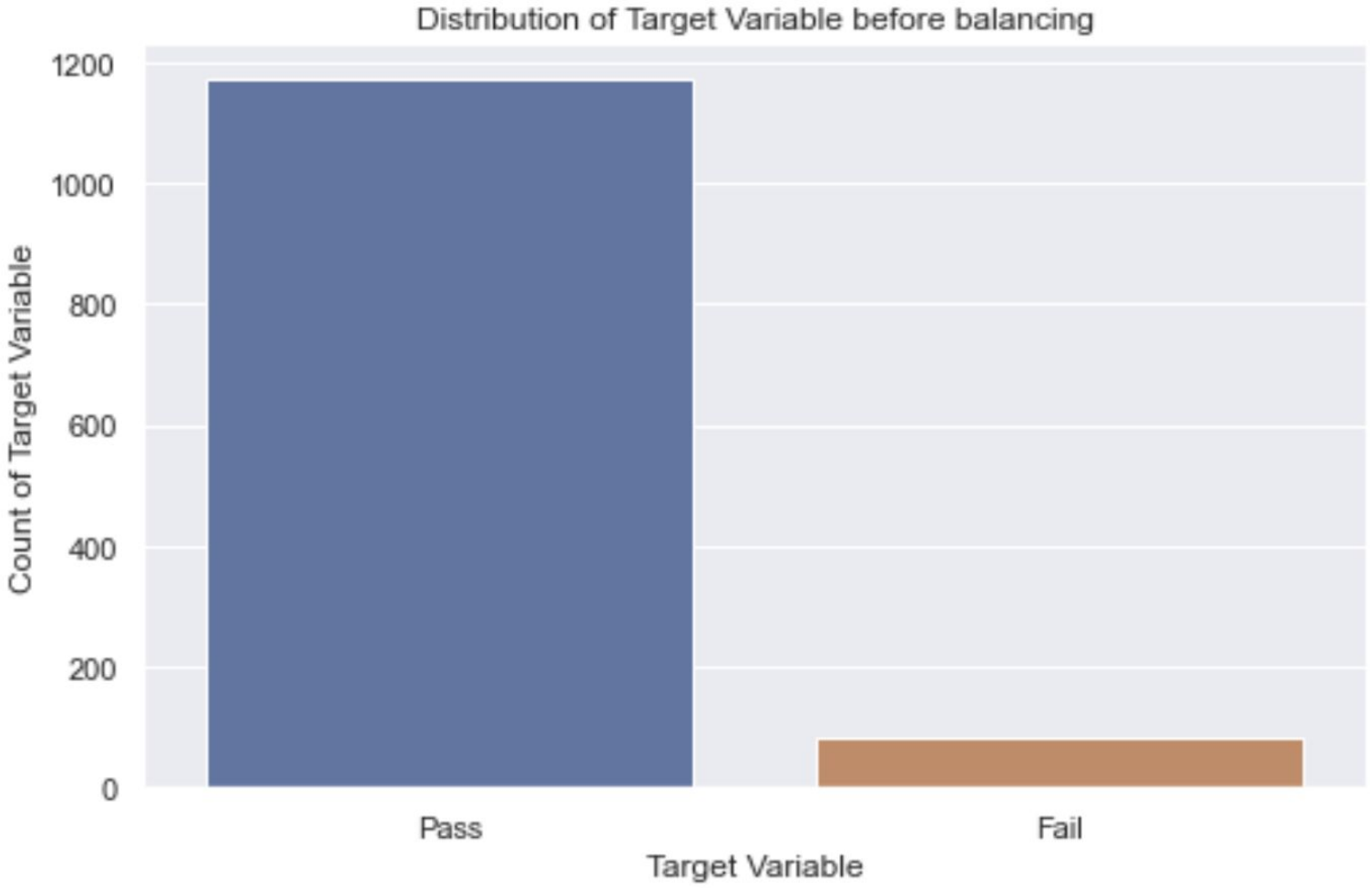
Dsouza, Rashmi Carol
Rashmi.Dsouza@Student.HTW-Berlin.de

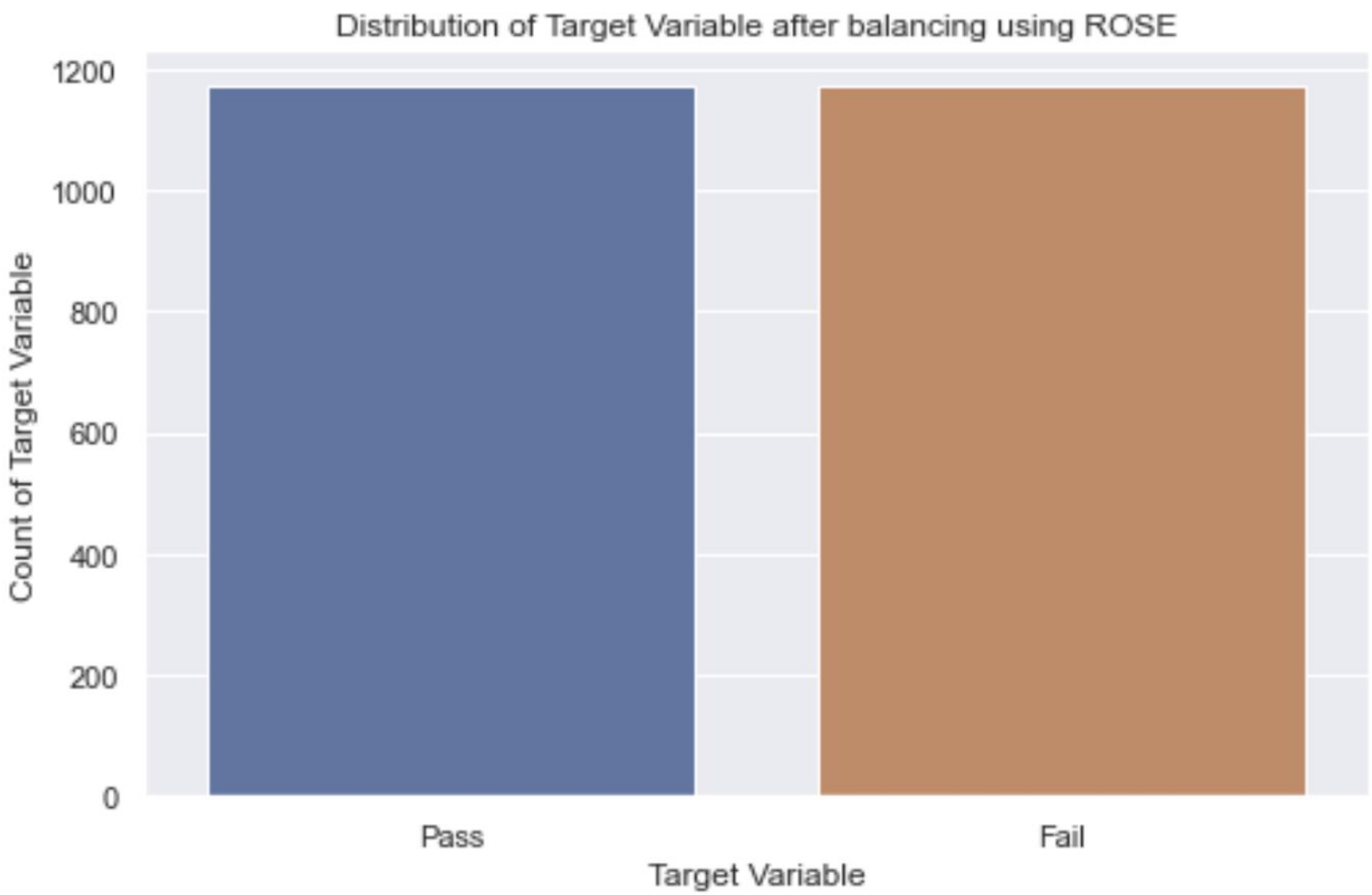
Pham, Quynh Dinh Hai
Quynh.Pham@Student.HTW-Berlin.de

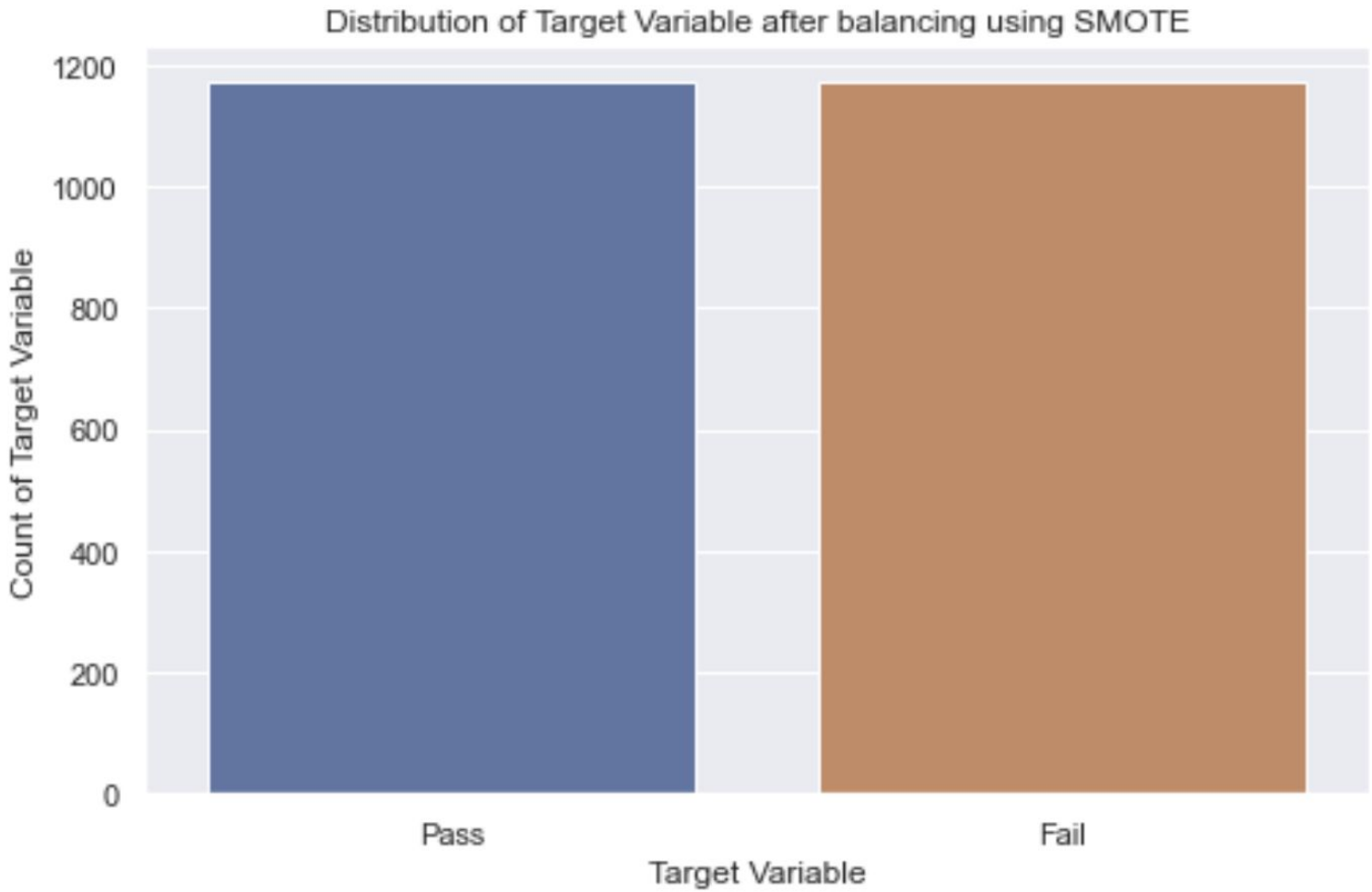
www.mpmd.htw-berlin.de

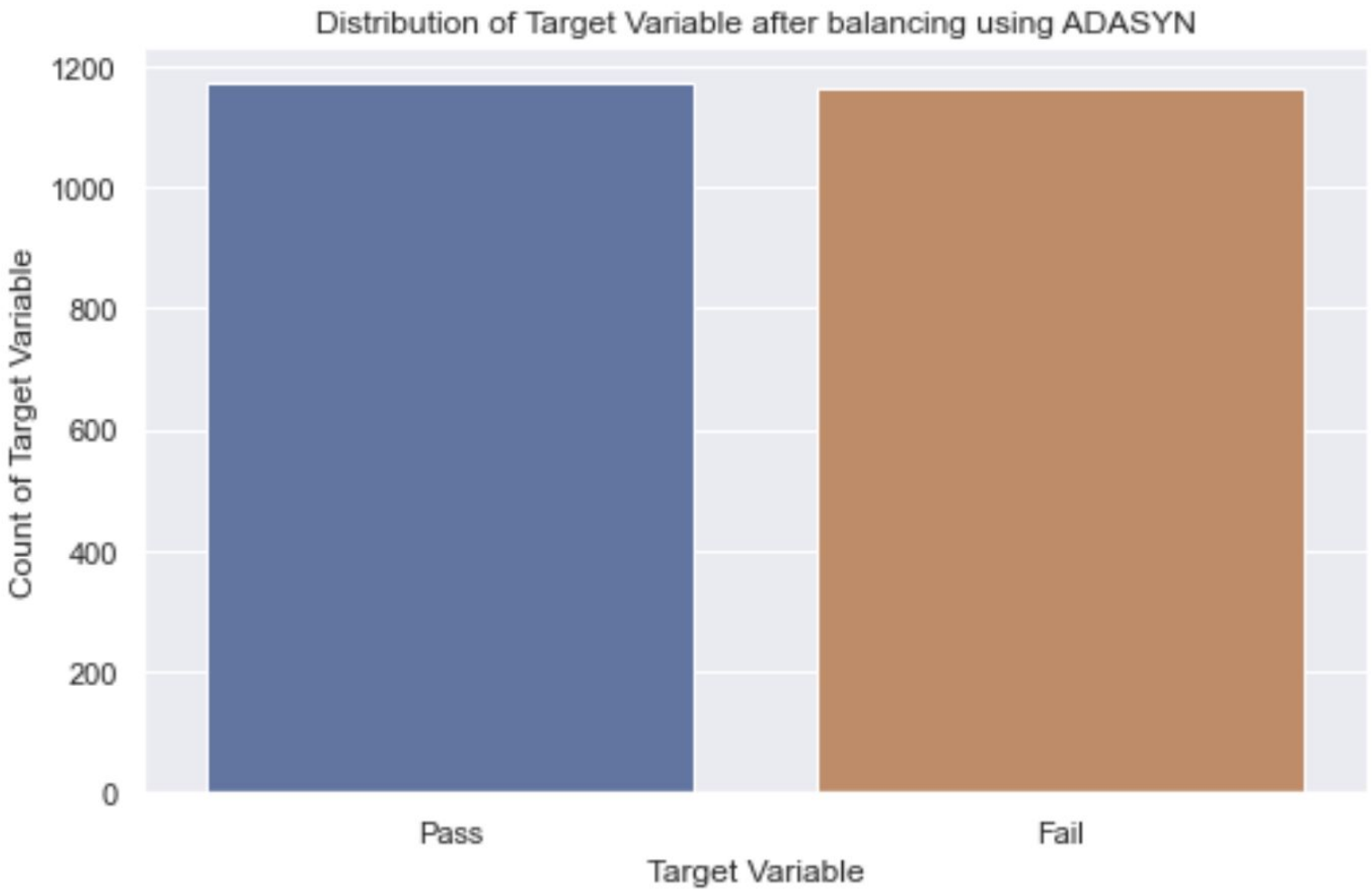
htw


Model Fit Analysis
Train
Test
Accuracy Score
1.00
0.95
0.90
0.85
0.80
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Max Depth (Random Forest)

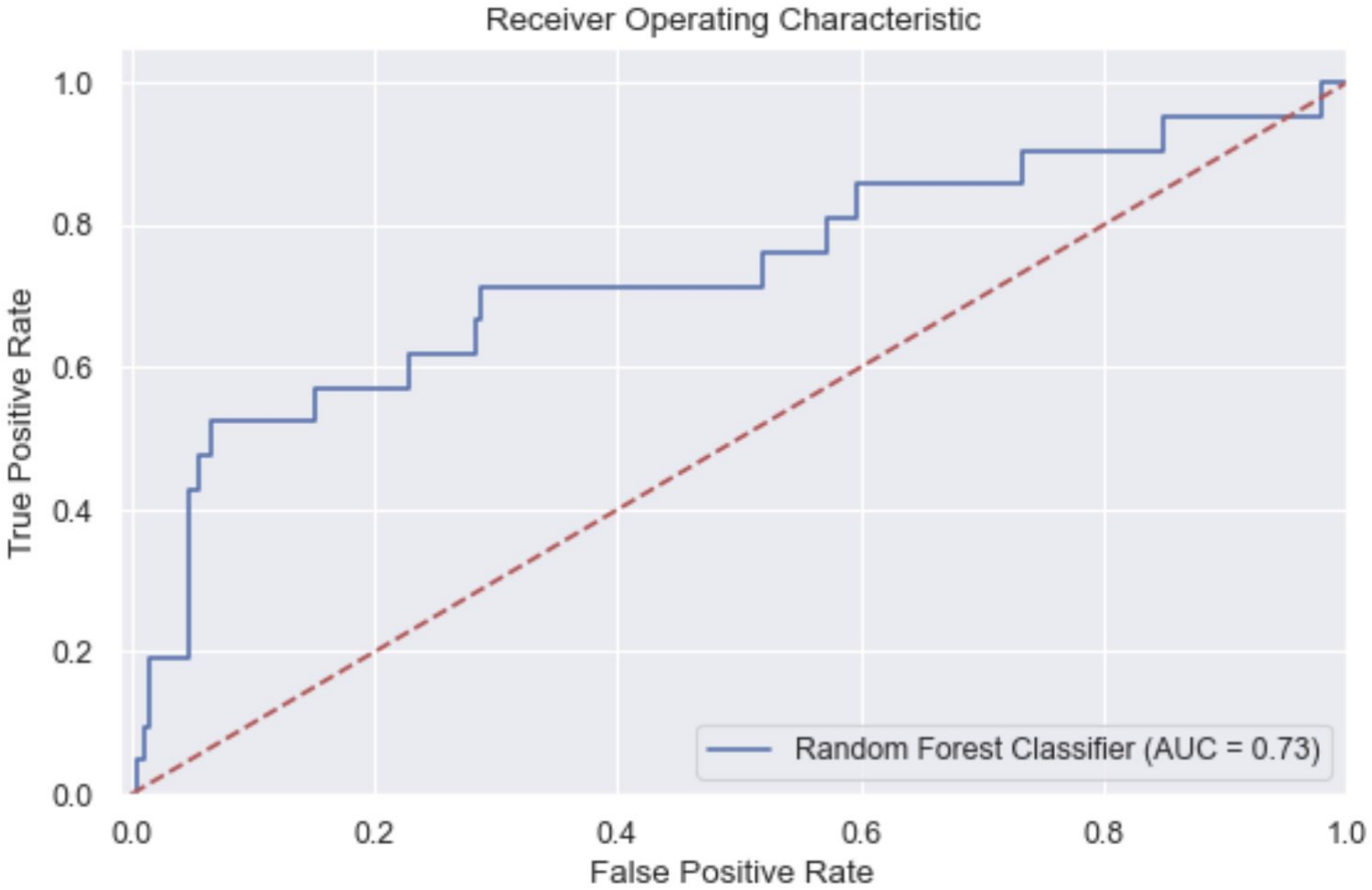

Distribution of Target Variable before balancing
1200
1000
800
600
400
200
0
Count of Target Variable
Pass
Fail
Target Variable


Distribution of Target Variable after balancing using ROSE
1200
1000
800
600
400
200
0
Count of Target Variable
Pass
Fail
Target Variable


Distribution of Target Variable after balancing using SMOTE
1200
1000
800
600
400
200
0
Count of Target Variable
Pass
Fail
Target Variable


Distribution of Target Variable after balancing using ADASYN
1200
1000
800
600
400
200
0
Count of Target Variable
Pass
Fail
Target Variable


Act. Pass
Act. Fail
269
24
10
11
Pred. Pass
Pred. Fail
250
200
150
100
50


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Random Forest Classifier (AUC = 0.73)

How to Choose Machine Learning Algorithm

1.Type of Problem
What type of problem you are dealing with and what kind of algorithm works best for each type of problem.

2. Size of the Training Data
It is usually recommended to gather a good amount of data to get reliable predictions. For small training data, choose algorithms with high bias/low variances like Linear regression, Naïve Bayes, or Linear SVM. For large training data, one can go for low bias/high variance algorithms like KNN, Decision trees, or kernel SVM.

3. Accuracy and/or Interpretability of the Output
Accuracy in machine learning measures the effectiveness of a model as the proportion of true results to total cases.

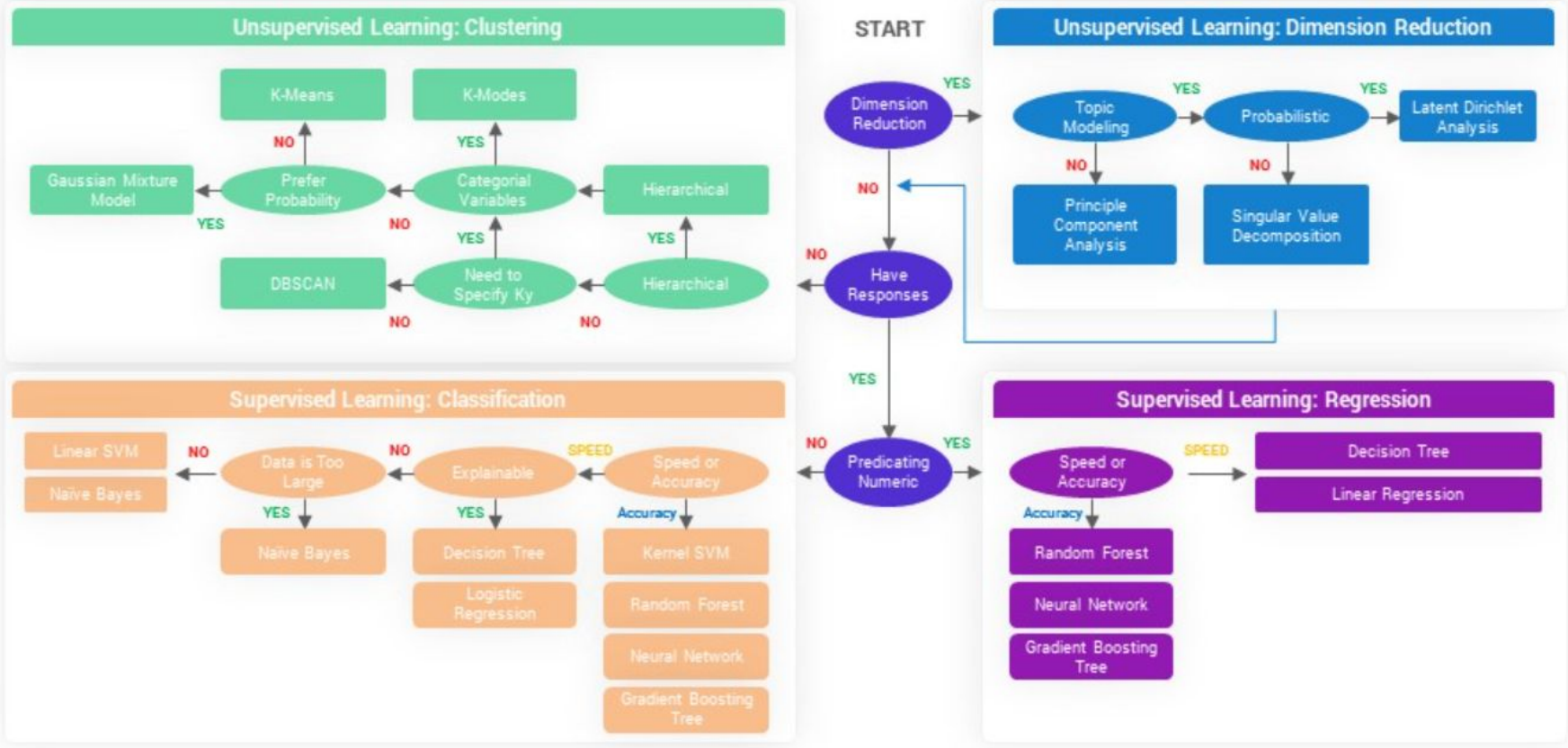
4. Speed or Training Time
The number of minutes or hours necessary to train a model varies a great deal between algorithms. Training time is often closely tied to accuracy; one typically accompanies the other.

5. Linearity
Linearity in statistics and machine learning means that there is a linear relationship between a variable and a constant in your dataset.

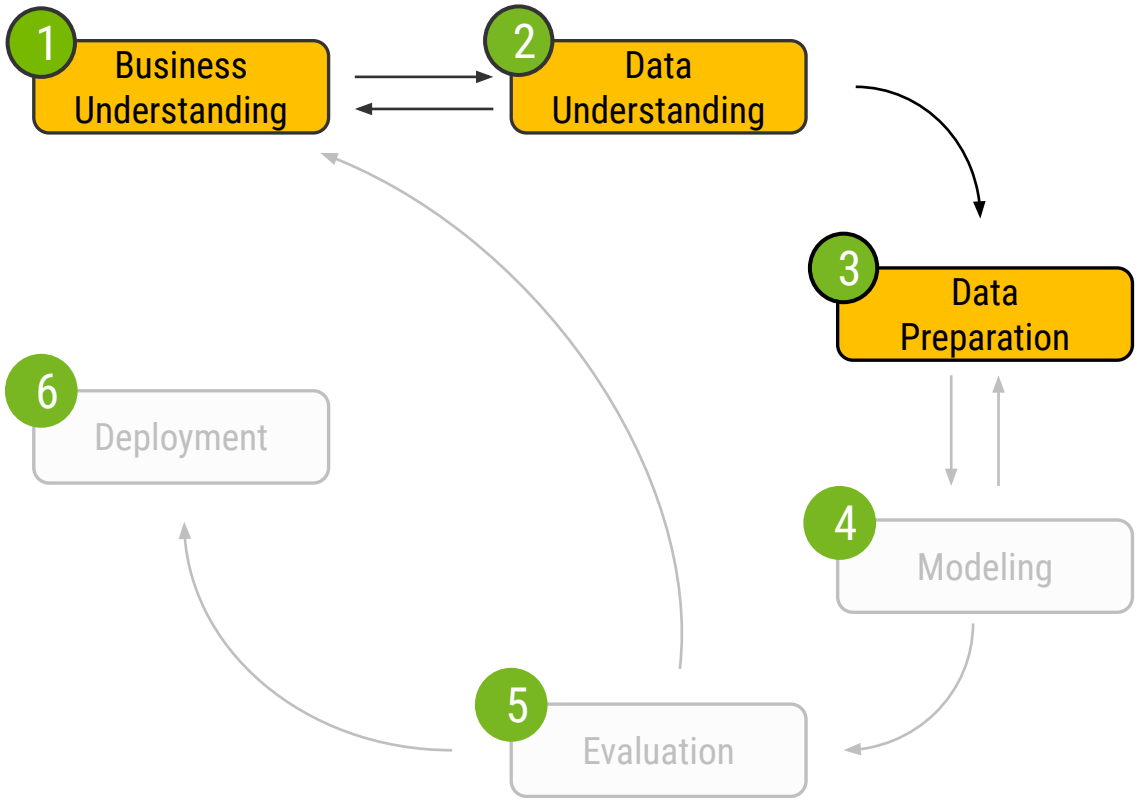
6.Number of Parameters
They are numbers that affect the algorithm's behavior, such as error tolerance or the number of iterations, or options between variants of how the algorithm behaves.

7. Number of Features
In machine learning, a feature is a quantifiable variable of the phenomenon you are trying to analyze. For certain types of data, the number of features can be very large compared to the number of data points. This is often the case with genetics or textual data.

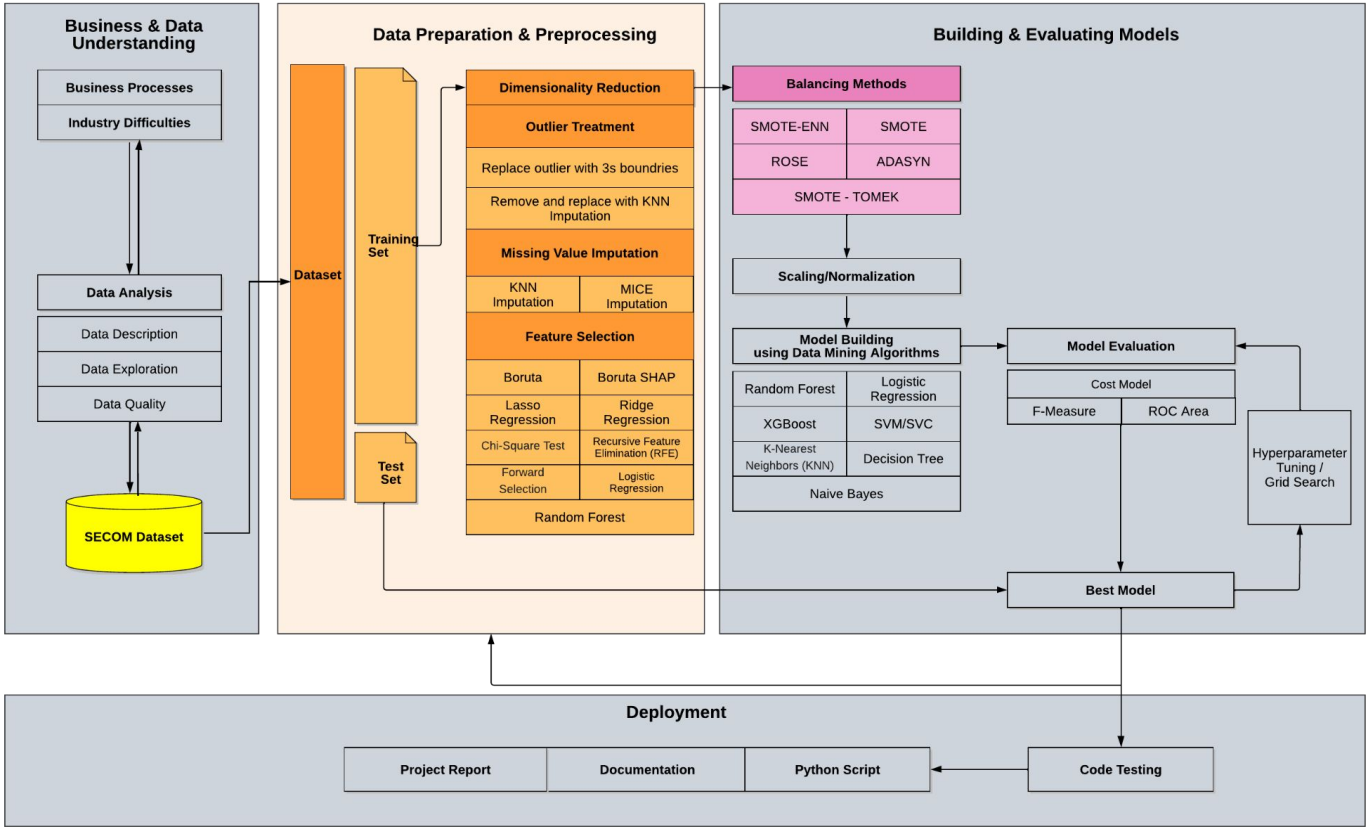
The Machine Learning Algorithm Cheat Sheet



01
DATA PREPARATION

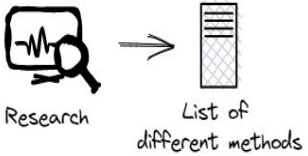


OVERVIEW


Business & Data Understanding
Business Processes
Industry Difficulties
Data Analysis
Data Description
Data Exploration
Data Quality
SECOM Dataset
Dataset
Data Preparation & Preprocessing
Training Set
Test Set
Dimensionality Reduction
Outlier Treatment
Replace outlier with 3s boundries
Remove and replace with KNN Imputation
Missing Value Imputation
KNN Imputation
MICE Imputation
Feature Selection
Boruta
Boruta SHAP
Lasso Regression
Ridge Regression
Chi-Square Test
Recursive Feature Elimination (RFE)
Forward Selection
Logistic Regression
Random Forest
Building & Evaluating Models
Balancing Methods
SMOTE-ENN
SMOTE
ROSE
ADASYN
SMOTE - TOMEK
Scaling/Normalization
Model Building using Data Mining Algorithms
Random Forest
Logistic Regression
XGBoost
SVM/SVC
K-Nearest Neighbors (KNN)
Decision Tree
Naive Bayes
Model Evaluation
Cost Model
F-Measure
ROC Area
Hyperparameter Tuning / Grid Search
Best Model
Deployment
Project Report
Documentation
Python Script
Code Testing

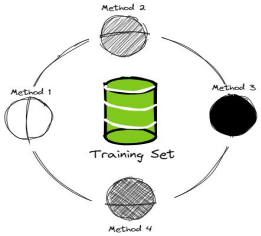
DATA PREPARATION APPROACH

01



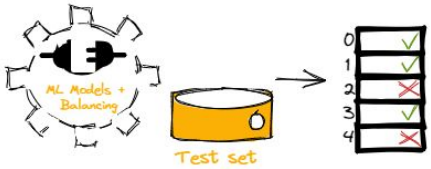
Research different methods for each step of data preparation

02



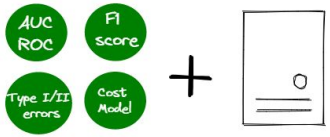
Use training set to train the machine with different method

03



Use different machine learning models, balancing methods & test set to test each method

04

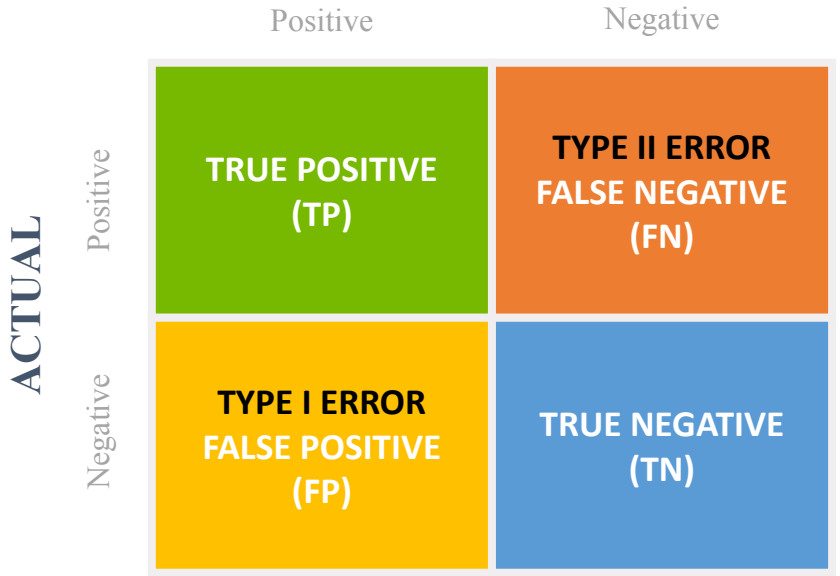


Use AUC, ROC, type I/II errors, F1 score & cost model as measures for decision making & document our findings

RESULT MEASUREMENTS

RESULT MEASURES

PREDICTION



CONFUSION MATRIX

Positive = **Pass** classification
Negative = **Fail** classification

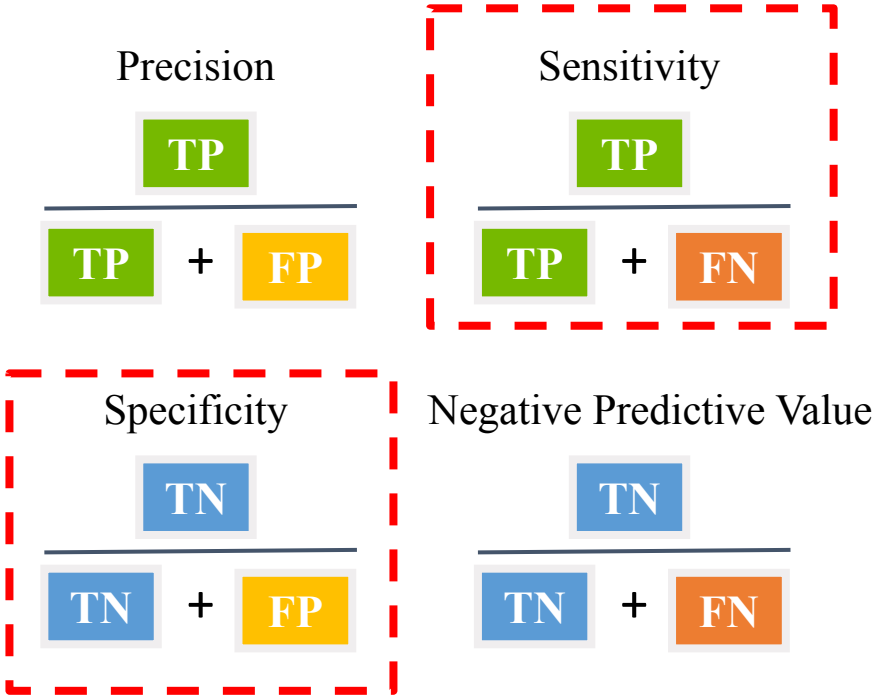
True Positive: Pass classification correctly identified as Pass

True Negative: Fail classification correctly identified as Fail

False Positive: Pass classification identified as Fail

False Negative: Fail classification identified as Pass

RESULT MEASURES



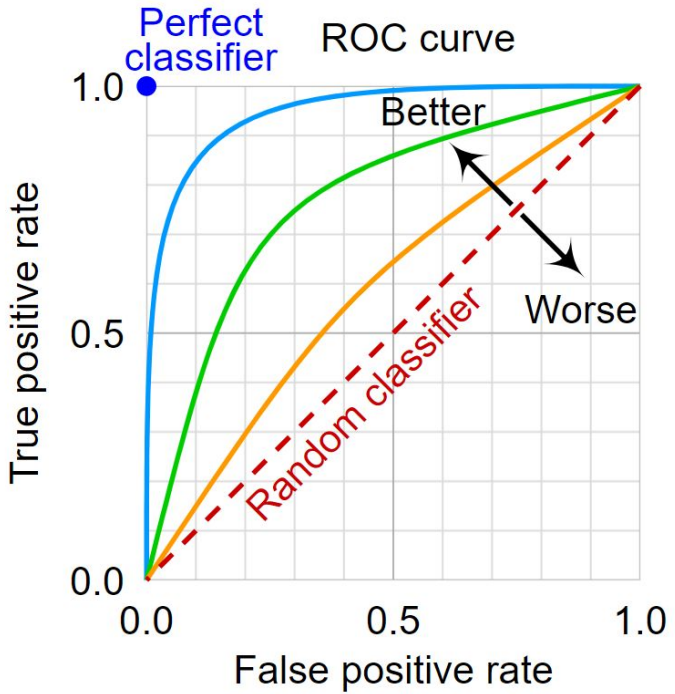
- **Sensitivity/Recall**: the ratio between how much were correctly identified as positive to how much were actually positive.
- **Specificity:** the ratio between how much were correctly classified as negative to how much was actually negative.
- **Precision:**How much were correctly classified as positive out of all positives.
- **NPV**:How much were correctly classified as negative out of all negatives

F1-Score

$$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 Score: combines precision and recall into one metric by calculating the harmonic mean between those two. It is primarily used to compare the performance of two classifiers. Classifier A has a higher recall, and classifier B has higher precision. In this case, the F1-scores for both the classifiers are used to determine which one produces better results.

ROC CURVE



- ROC curve is used to visualize the trade off between Sensitivity and Specificity.
- AUC is the area under the curve
- **Sensitivity** on the **Y-axis**
- **Specificity** on the **X-axis**

COST MODEL

Because of trade-off between Sensitivity & Specificity, we think in terms of business and what decision to make that is good for business, which is to **minimize the cost** as much as possible → **build a cost model** for method comparison & decision making

01
Cost of TP

02
Cost of TN

03
Type I Error
Cost of FP = $225

- Manufacturing & Shipping
- Recall cost
- Reshipping
- Damages

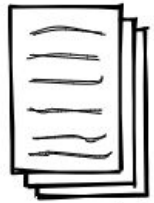
04
Type II Error
Cost of FN = $100

- Manufacturing & Shipping

DATA PREPARATION

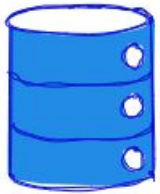
MERGING DATA

```
3030.93 2564 2187.
284 0.4734 0.0167
71 31.8843 NaN NaN
11.5074 0.1096 0.0
```

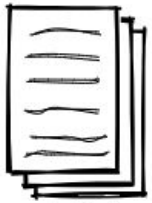


Secom.data

SECOM Dataset



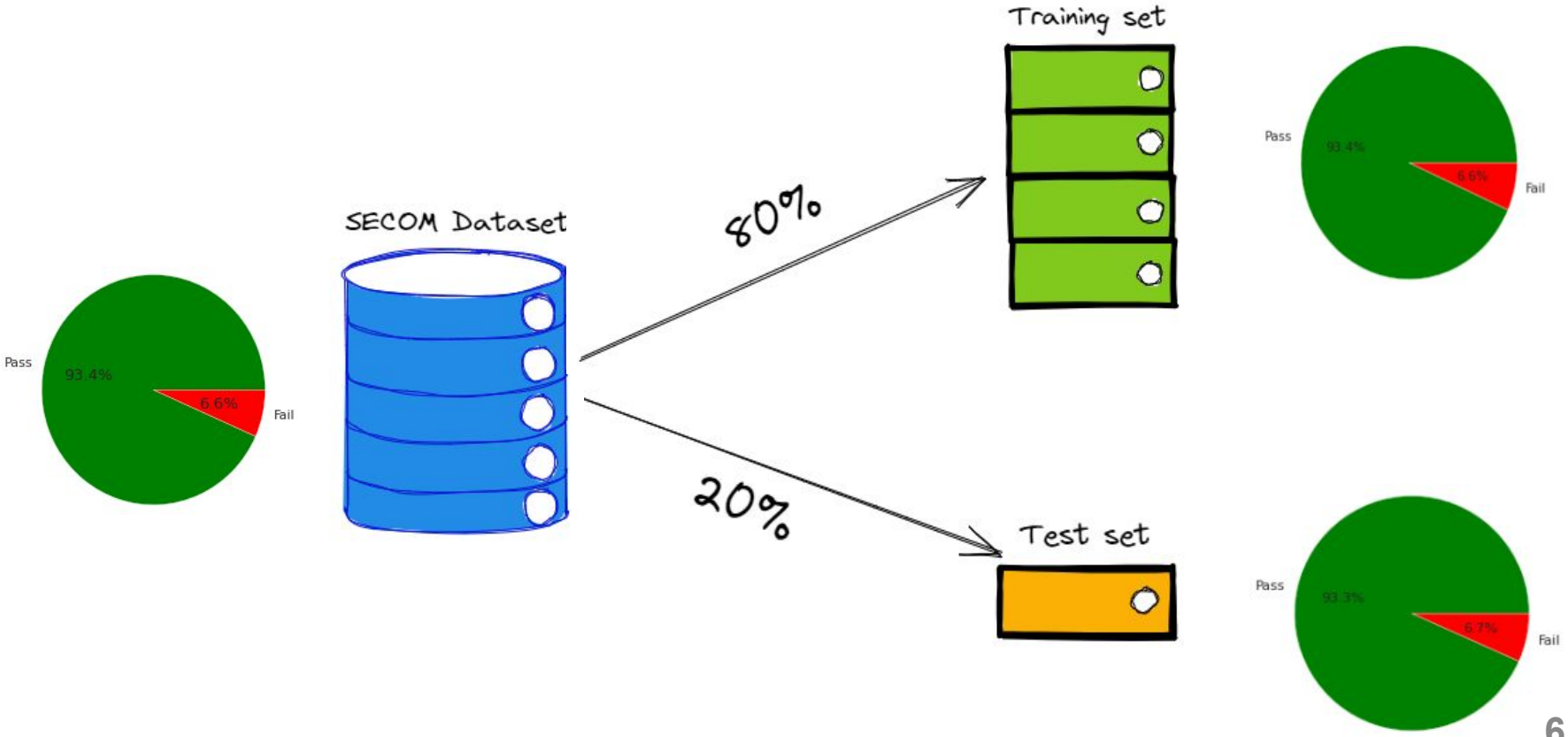
```
-1 "19/07/2008 11:55:00"
-1 "19/07/2008 12:32:00"
1 "19/07/2008 13:17:00"
-1 "19/07/2008 14:43:00"
-1 "19/07/2008 15:22:00"
-1 "19/07/2008 17:53:00"
```



Secom_labels.dat

	Classification	Timestamp	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Feature_8	...
0	-1	2008-07-19 11:55:00	3030.93	2564.00	2187.7333	1411.1265	1.3602	100.0	97.6133	0.1242	...
1	-1	2008-07-19 12:32:00	3095.78	2465.14	2230.4222	1463.6606	0.8294	100.0	102.3433	0.1247	...
2	1	2008-07-19 13:17:00	2932.61	2559.94	2186.4111	1698.0172	1.5102	100.0	95.4878	0.1241	...
3	-1	2008-07-19 14:43:00	2988.72	2479.90	2199.0333	909.7926	1.3204	100.0	104.2367	0.1217	...

SEPARATE TRAINING & TEST DATA



DIMENSIONALITY REDUCTION

1. Remove the Timestamp feature
2. Remove features with missing values

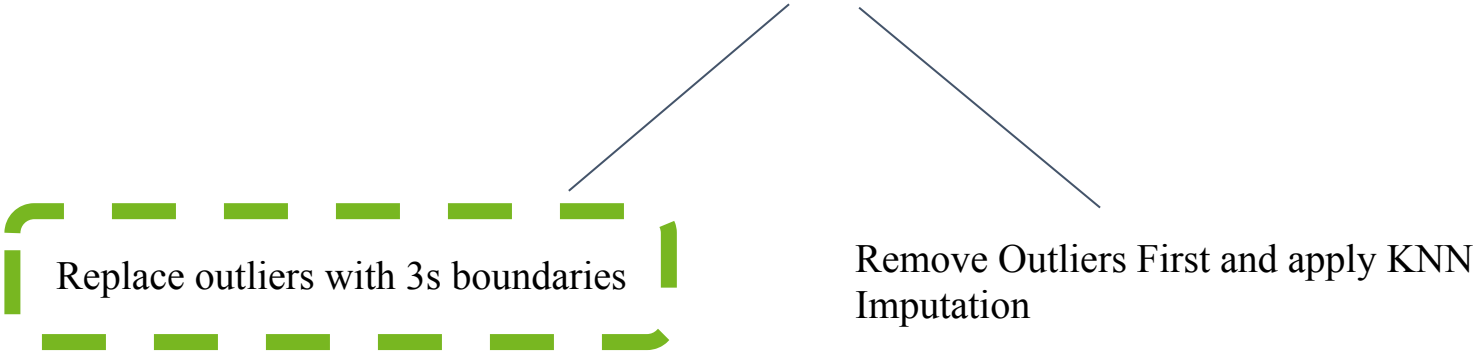
Threshold:
Features with more than 50% missing value

```
] def percent(dataframe, threshold):
      columns = dataframe.columns[(dataframe.isna().sum()/dataframe.shape[1])>threshold]
      return columns.tolist()

  na_columns = percent(X_train, 0.5)
  X_train_na = X_train.drop(na_columns, axis=1)
  X_test_na = X_test.drop(na_columns, axis=1)
  n_features1 = X_train_na.shape[1]
  print(f'Removed = {len(na_columns)} , Left = {n_features1}')

  Removed = 32 , Left = 558
```

OUTLIER TREATMENT



We tried both methods and we get a better result with replacing outliers with 3s boundaries => **we chose to replace the outliers with 3s boundaries**

FEATURE EXPLORATION

SELECTION AND ELIMINATION

	Type I error	Type II error	F1 score	AUC	Cost	Remaining features
KNN - Boruta - SMOTE	11	74	72	70	$ 98.75	12
KNN - Boruta - SMOTE-ENN	10	88	69	73	$ 110.50	12
KNN - Boruta - SMOTE-TOMEK	11	67	75	68	$ 91.75	12
KNN - Boruta - ADASYN	12	79	71	63	$ 106.00	12
KNN - Boruta - ROSE	10	53	79	72	$ 75.50	12
KNN - Chi-Square Test - SMOTE	13	43	82	67	$ 72.25	29
KNN - Chi-Square Test - SMOTE-ENN	11	61	77	69	$ 85.75	29
KNN - Chi-Square Test - SMOTE-TOMEK	14	41	82	68	$ 72.50	29
KNN - Chi-Square Test - ADASYN	12	42	82	68	$ 69.00	29
KNN - Chi-Square Test - ROSE					$ -	29
KNN - Lasso Regression - SMOTE	14	25	87	73	$ 56.50	29
KNN - Lasso Regression – SMOTE-ENN	11	62	76	72	$ 86.75	29
KNN - Lasso Regression – SMOTE-TOMEK	16	20	87	72	$ 56.00	29
KNN - Lasso Regression - ADASYN	14	34	84	71	$ 65.50	29
KNN - Lasso Regression - ROSE					$ -	29
KNN - Boruta SHAP - SMOTE					$ -	
KNN - Forward Selection - SMOTE	13	30	86	70	$ 59.25	15
KNN - Forward Selection - SMOTE-ENN	9	49	81	73	$ 69.25	15
KNN - Forward Selection - SMOTE-TOMEK	13	30	86	72	$ 59.25	15
KNN - Forward Selection - ADASYN	14	30	85	73	$ 61.50	15
KNN - Forward Selection - ROSE	11	28	87	75	$ 52.75	15
KNN - RFE - SMOTE	14	45	81	73	$ 76.50	15
KNN - RFE - SMOTE-ENN	11	77	71	70	$ 101.75	15

	Type I error	Type II error	F1 score	AUC	Cost	Remaining Features
MICE - Boruta - SMOTE	11	76	68	72	$100.75	12
MICE - Boruta - SMOTE-ENN	9	84	74	70	$104.25	12
MICE - Boruta - SMOTE-TOMEK	11	78	71	69	$102.75	12
MICE - Boruta - ADASYN	11	82	70	63	$106.75	12
MICE - Boruta - ROSE	9	53	80	71	$ 73.25	12
MICE - Chi-Square Test - SMOTE	14	43	81	65	$ 74.50	29
MICE - Chi-Square Test - SMOTE-ENN	13	61	76	67	$ 90.25	29
MICE - Chi-Square Test - SMOTE-TOMEK	15	46	80	66	$ 79.75	29
MICE - Chi-Square Test - ADASYN	11	42	83	67	$ 66.75	29
MICE - Chi-Square Test - ROSE					$ -	29
MICE - Lasso Regression - SMOTE	14	26	87	74	$ 57.50	29
MICE - Lasso Regression - SMOTE-ENN	10	64	76	71	$ 86.50	29
MICE - Lasso Regression - SMOTE-TOMEK	14	26	87	74	$ 57.50	29
MICE - Lasso Regression - ADASYN	15	36	83	74	$ 69.75	29
MICE - Lasso Regression - ROSE					$ -	29
MICE - Boruta SHAP - SMOTE					$ -	6
MICE - Forward Selection - SMOTE	12	29	86	73	$ 56.00	
MICE - Forward Selection - SMOTE-ENN	10	46	82	70	$ 68.50	
MICE - Forward Selection - SMOTE-TOMEK	14	27	86	71	$ 58.50	
MICE - Forward Selection - ADASYN	12	32	85	72	$ 59.00	
MICE - Forward Selection - ROSE	10	35	85	76	$ 57.50	

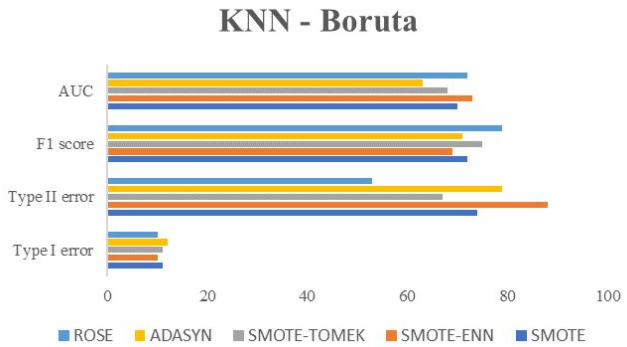
There is *no best feature selection method*. Instead, we have discovered what works best for the specific problems related to this Dataset using careful systematic experimentation/trial and error method. We tried a range of different models fit on different subsets of features chosen via different statistical measures and *discovered the ones works best*.

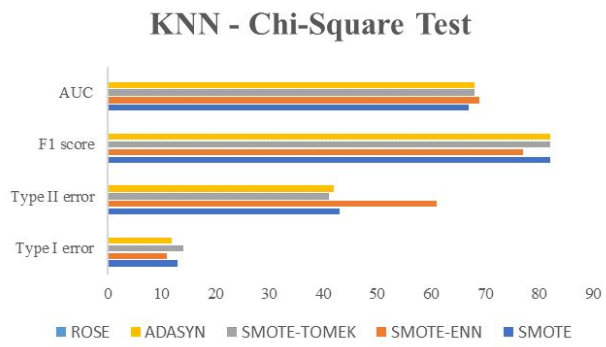
COMBINATIONS

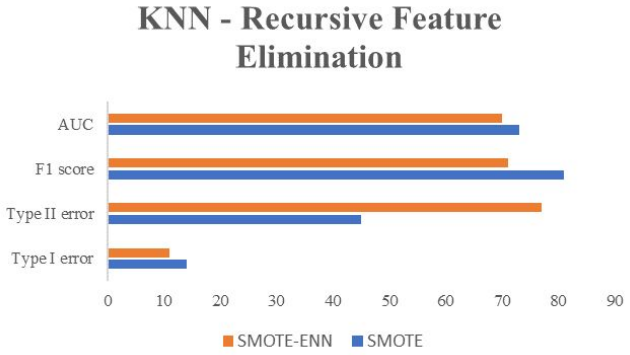
Missing Value Imputation	Feature Selection	Balancing
KNN imputation	Boruta	SMOTE-ENN
MICE imputation	Boruta SHAP	SMOTE
	Lasso Regression	ROSE
	Ridge Regression	ADASYN
	Chi-Square Test	SMOTE-TOMEK
	Recursive Feature Elimination (RFE)	
	Forward Selection	
	Logistic Regression	
	Random Forest	

1. Missing Value Imputation
2. Feature Selection
3. Balancing

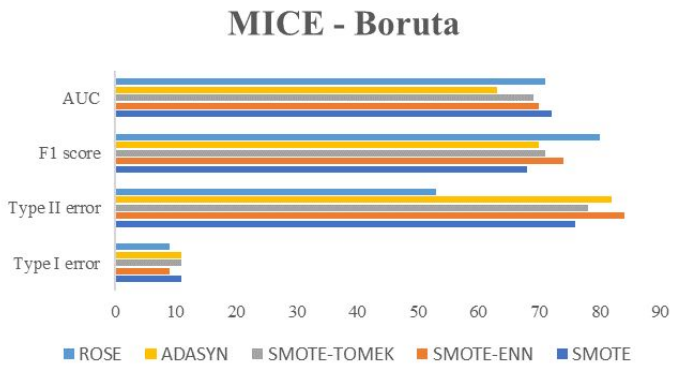
EVALUATION METRICS (KNN)

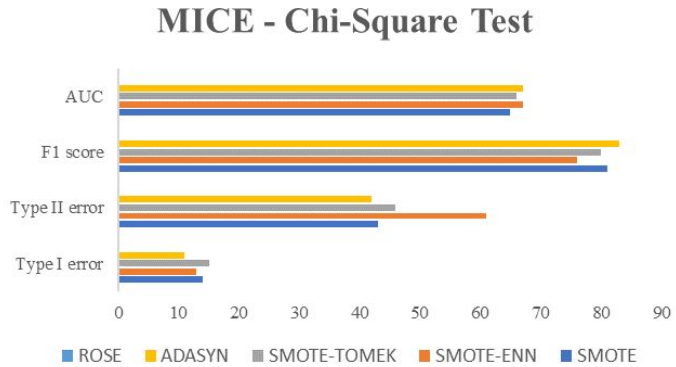


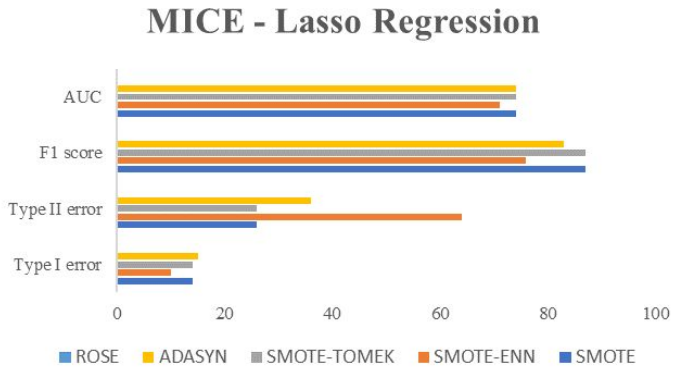


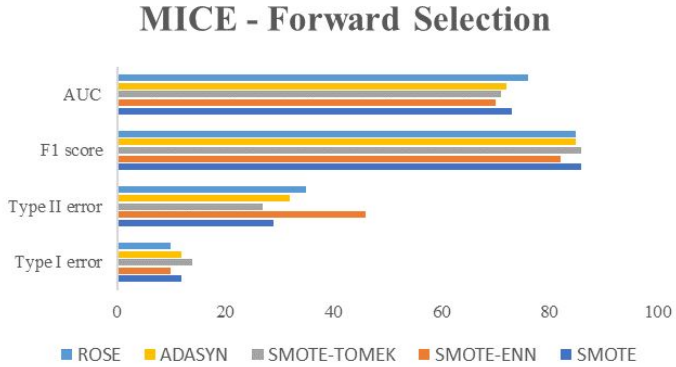


EVALUATION METRICS (MICE)





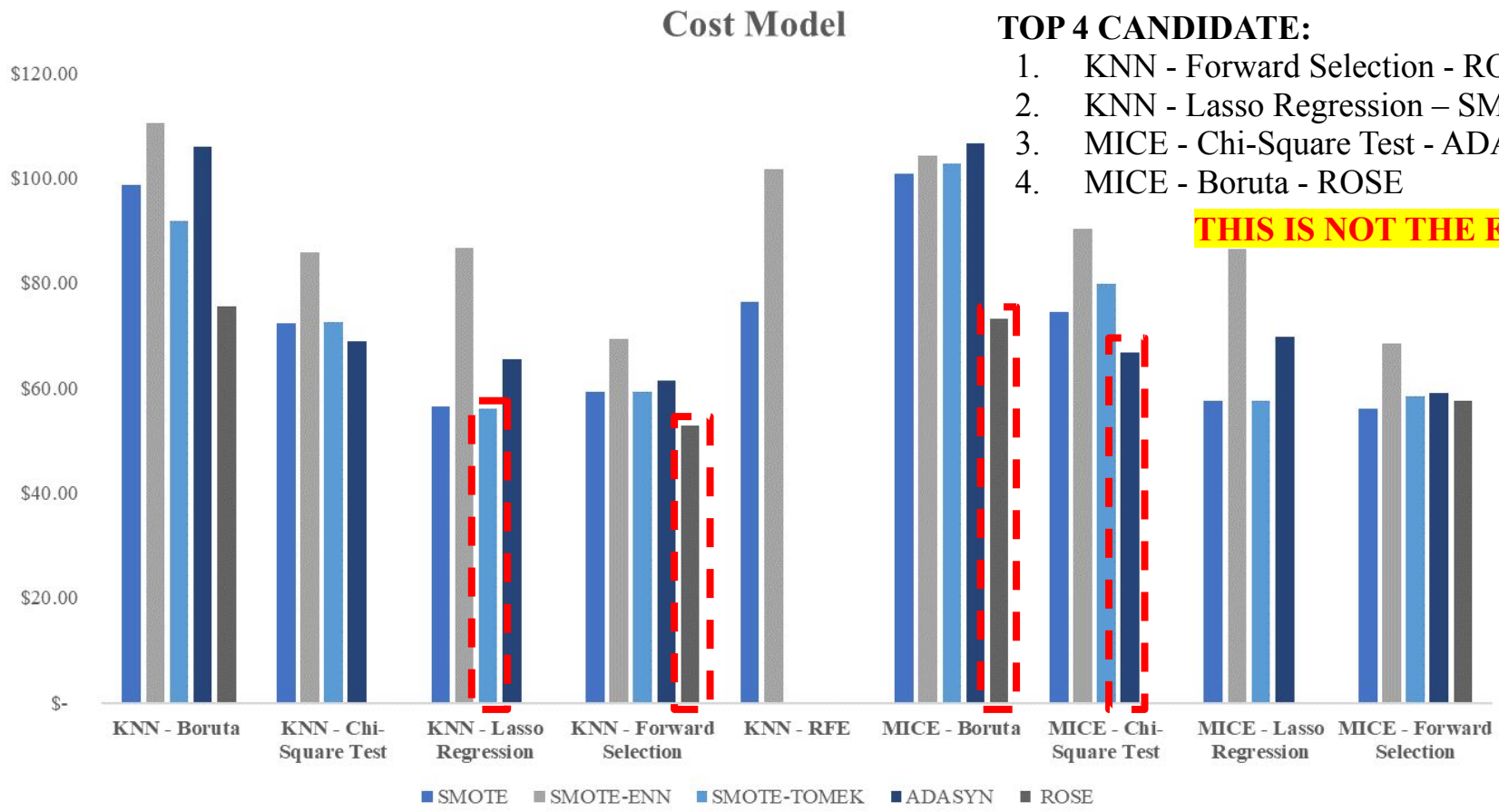




TRIAL RESULTS
Cost Model

	SMOTE	SMOTE-ENN	SMOTE-TOMEK	ADASYN	ROSE
KNN - Boruta	$98.75	$110.50	$91.75	$106.00	$75.50
KNN - Chi-Square Test	$72.25	$85.75	$72.50	$69.00	$0.00
KNN - Lasso Regression	$56.50	$86.75	$56.00	$65.50	$0.00
KNN - Forward Selection	$59.25	$69.25	$59.25	$61.50	$52.75
KNN - RFE	$76.50	$101.75			
MICE - Boruta	$100.75	$104.25	$102.75	$106.75	$73.25
MICE - Chi-Square Test	$74.50	$90.25	$79.75	$66.75	$0.00
MICE - Lasso Regression	$57.50	$86.50	$57.50	$69.75	$0.00
MICE - Forward Selection	$56.00	$68.50	$58.50	$59.00	$57.50

TRIAL RESULTS



TOP 4 CANDIDATE:

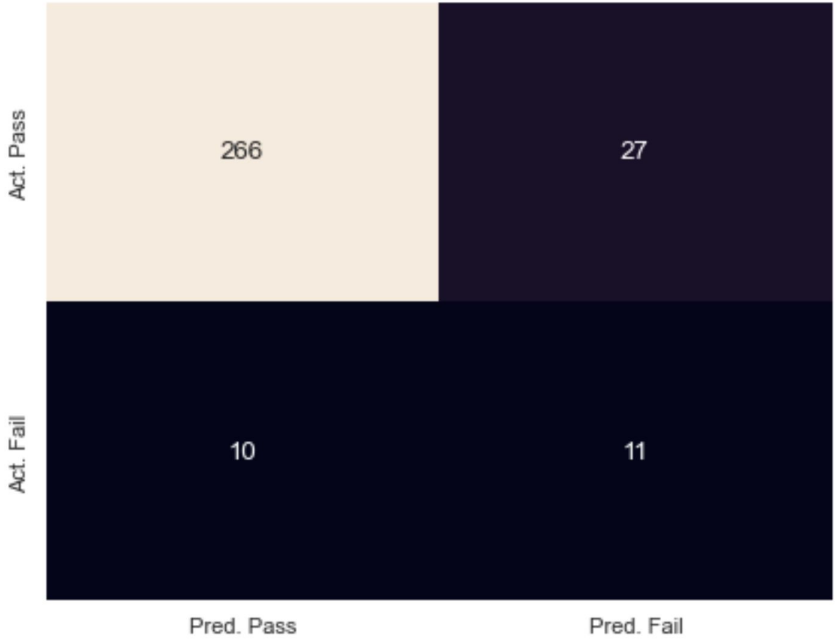
1. KNN - Forward Selection - ROSE
2. KNN - Lasso Regression – SMOTE-TOMEK
3. MICE - Chi-Square Test - ADASYN
4. MICE - Boruta - ROSE

THIS IS NOT THE END RESULTS

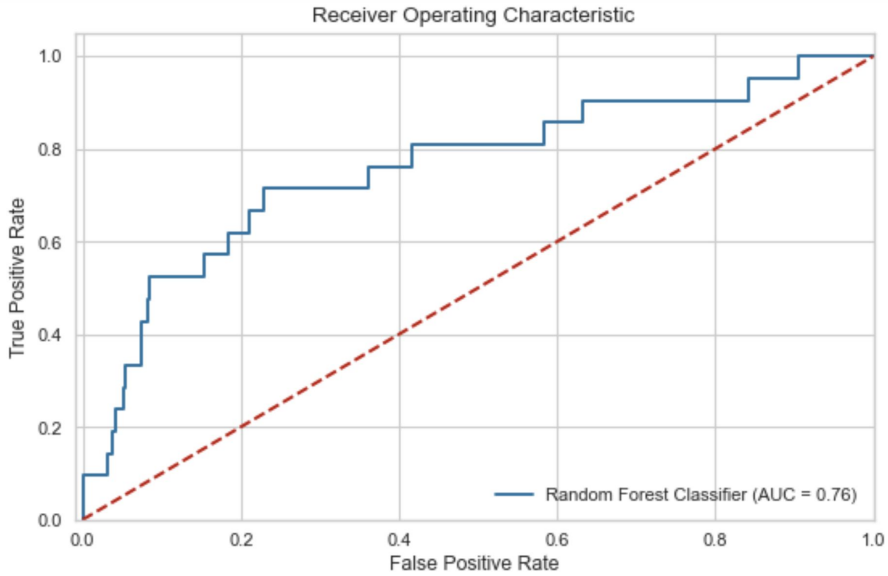
KNN + Forward Selection + ROSE

KNN - Forward Selection - ROSE

Confusion Matrix



AUC/ROC

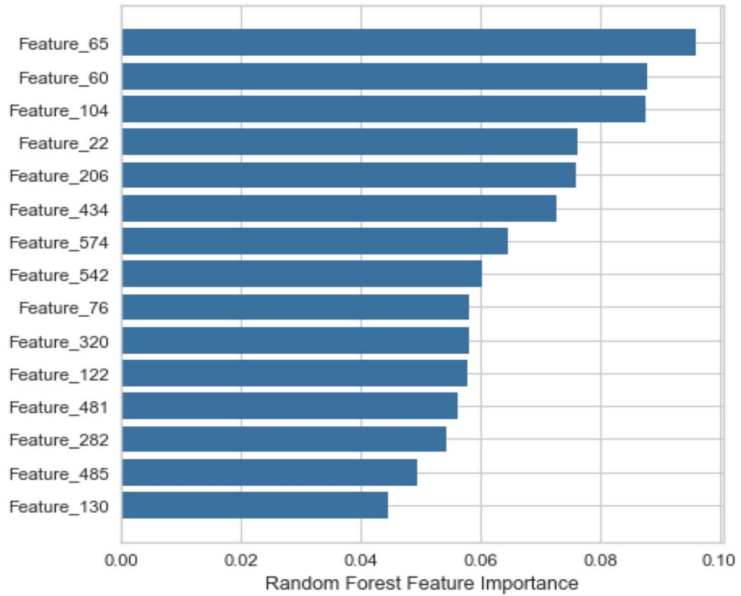


Total cost: $ 49,500

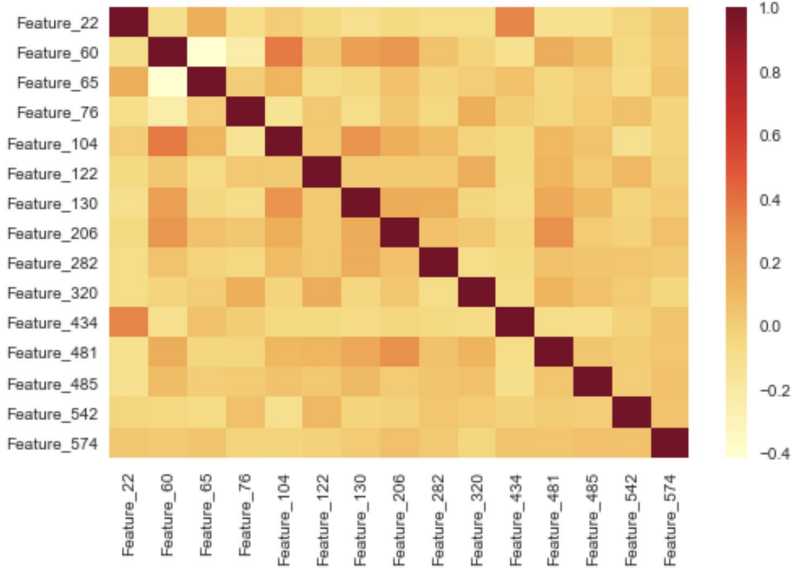
Number of remaining features:15

KNN - Forward Selection - ROSE

Feature Importance



Correlation between remaining features

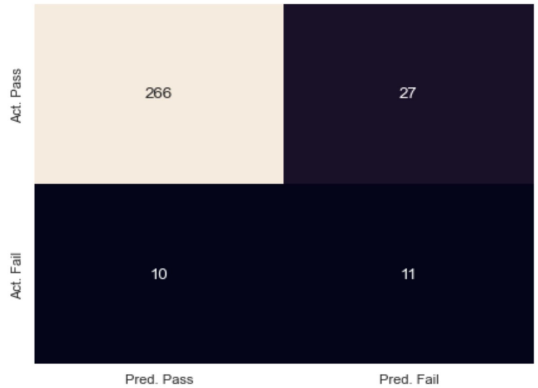


COMPARATIVE ANALYSIS

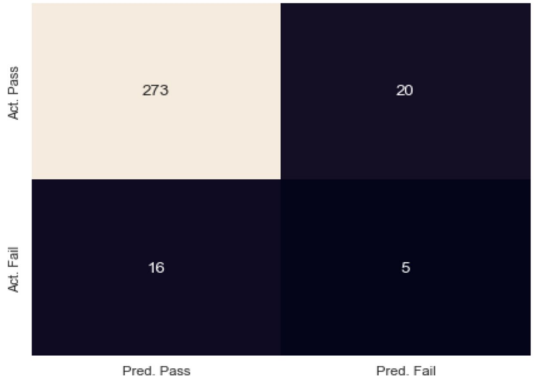
Master
Project Management
and Data Science

COMPARISON

KNN - Forward Selection - ROSE
$49,000; 15 Features



KNN - Lasso Regression – SMOTE-TOMEK
$56,000; 29 Features

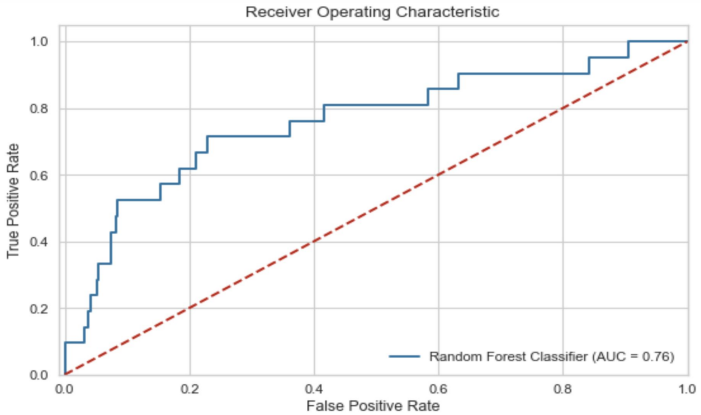


MICE - Boruta - ROSE
$72,500; 12 Features

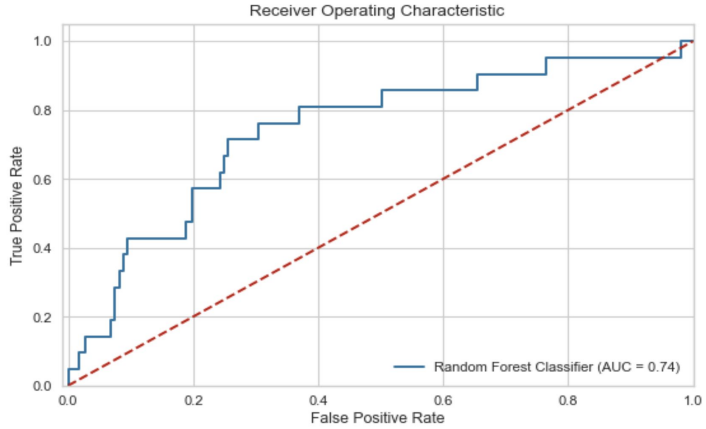
MICE - Chi-Square Test - ADASYN
$67,000; 29 Features

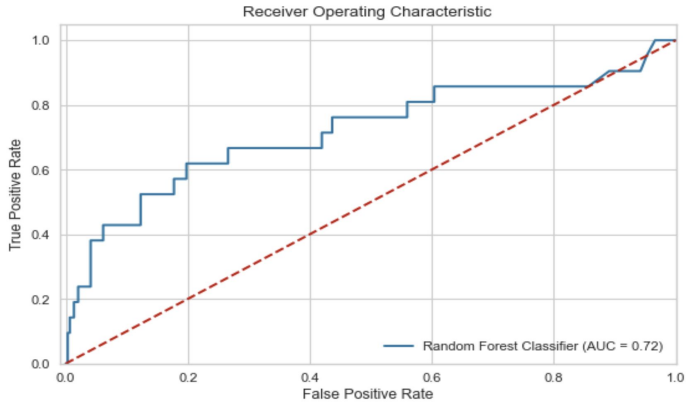
AUC/ROC

KNN - Forward Selection - ROSE

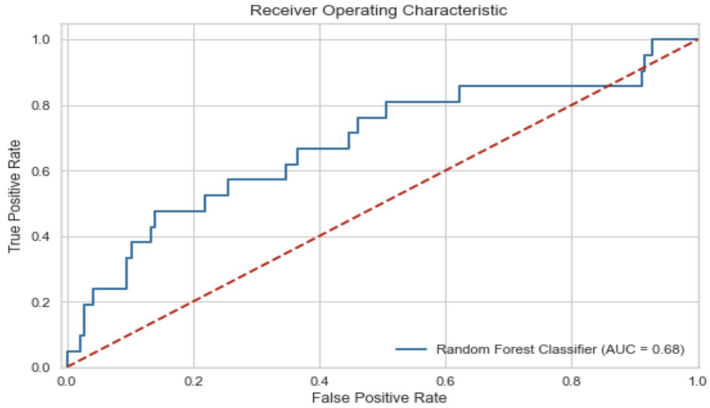


KNN - Lasso Regression – SMOTE-TOMEK





MICE - Boruta - ROSE

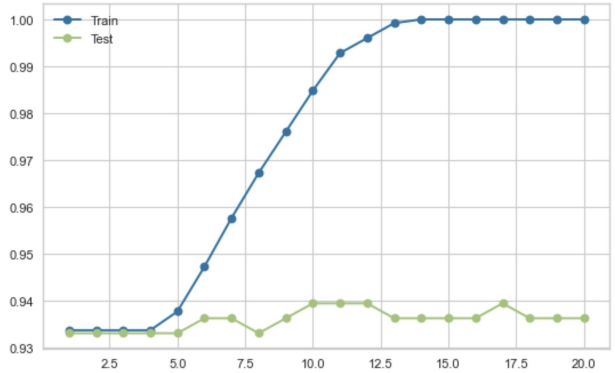


MICE - Chi-Square Test - ADASYN

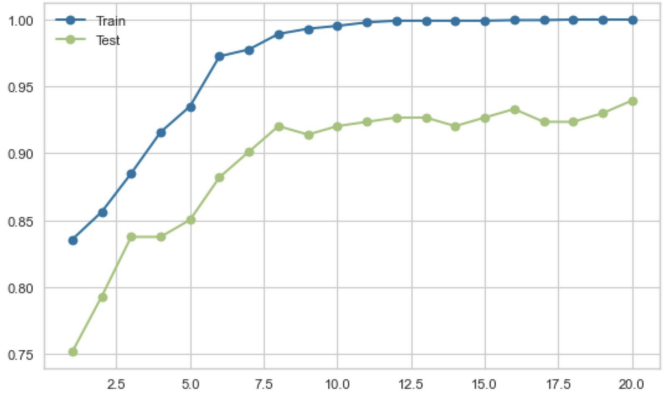
UNDERSTANDING MODELLING

OVERFITTING ANALYSIS

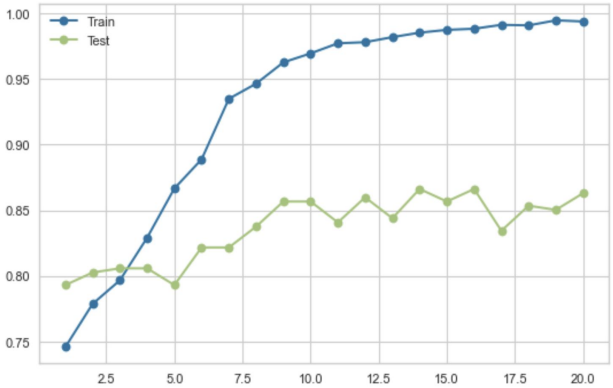
KNN - Forward Selection - ROSE



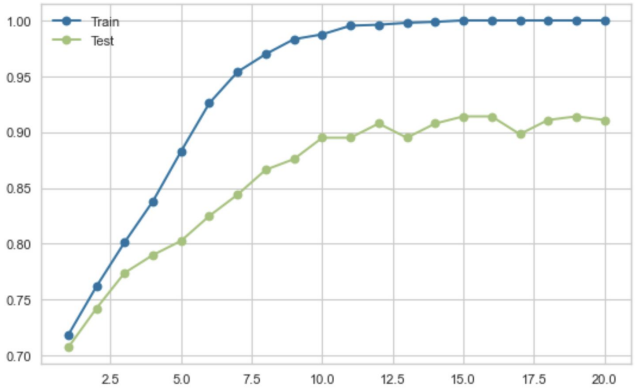
KNN - Lasso Regression – SMOTE-TOMEK



MICE - Boruta - ROSE



MICE - Chi-Square Test - ADASYN

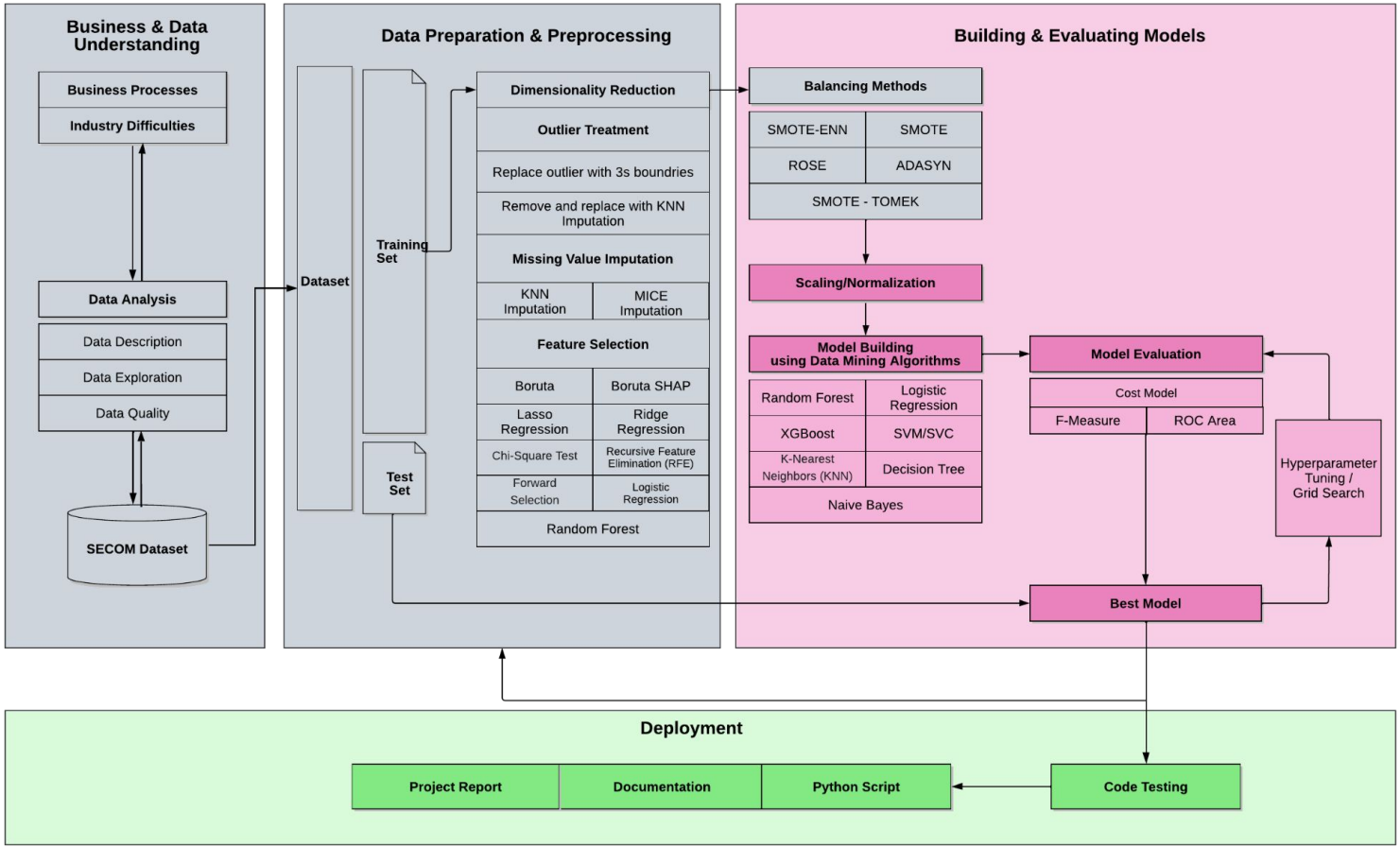


02

Next Steps

- Modeling
- Evaluation
- Deployment

Upcoming…


Business & Data Understanding
Business Processes
Industry Difficulties
Data Analysis
Data Description
Data Exploration
Data Quality
SECOM Dataset
Data Preparation & Preprocessing
Dataset
Training Set
Test Set
Dimensionality Reduction
Outlier Treatment
Replace outlier with 3s boundries
Remove and replace with KNN Imputation
Missing Value Imputation
KNN Imputation
MICE Imputation
Feature Selection
Boruta
Boruta SHAP
Lasso Regression
Ridge Regression
Chi-Square Test
Recursive Feature Elimination (RFE)
Forward Selection
Logistic Regression
Random Forest
Building & Evaluating Models
Balancing Methods
SMOTE-ENN
SMOTE
ROSE
ADASYN
SMOTE - TOMEK
Scaling/Normalization
Model Building using Data Mining Algorithms
Random Forest
Logistic Regression
XGBoost
SVM/SVC
K-Nearest Neighbors (KNN)
Decision Tree
Naive Bayes
Model Evaluation
Cost Model
F-Measure
ROC Area
Hyperparameter Tuning / Grid Search
Best Model
Deployment
Project Report
Documentation
Python Script
Code Testing

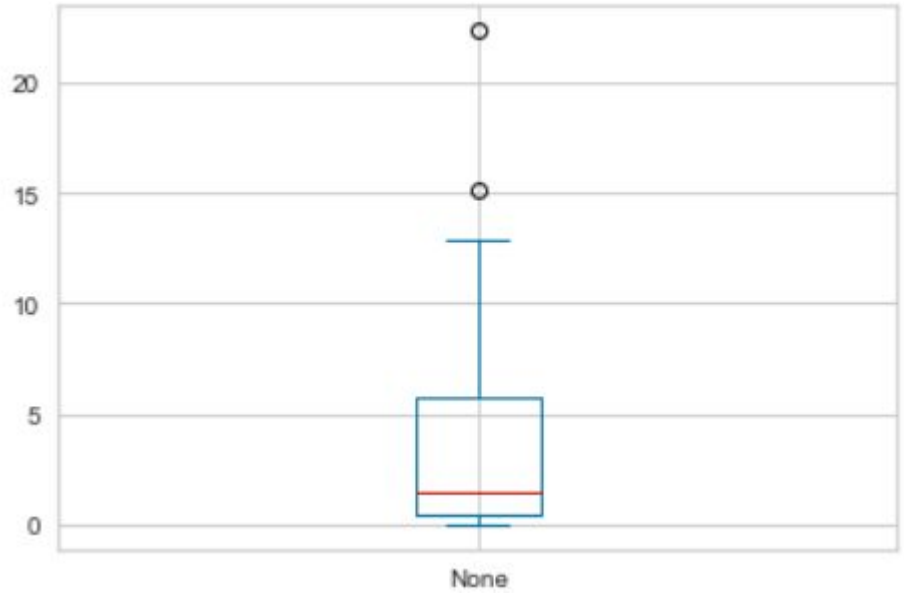
Outlier Treatment

No.	Outlier Treatment Methods	AUC	ROC	Number of features removed	Number of remaining features	Decision
1	Replace outlier & replace with 3s boundaries					
2	Remove outlier and replace with KNN imputation					
3	Remove outlier and replace with MICE imputation					

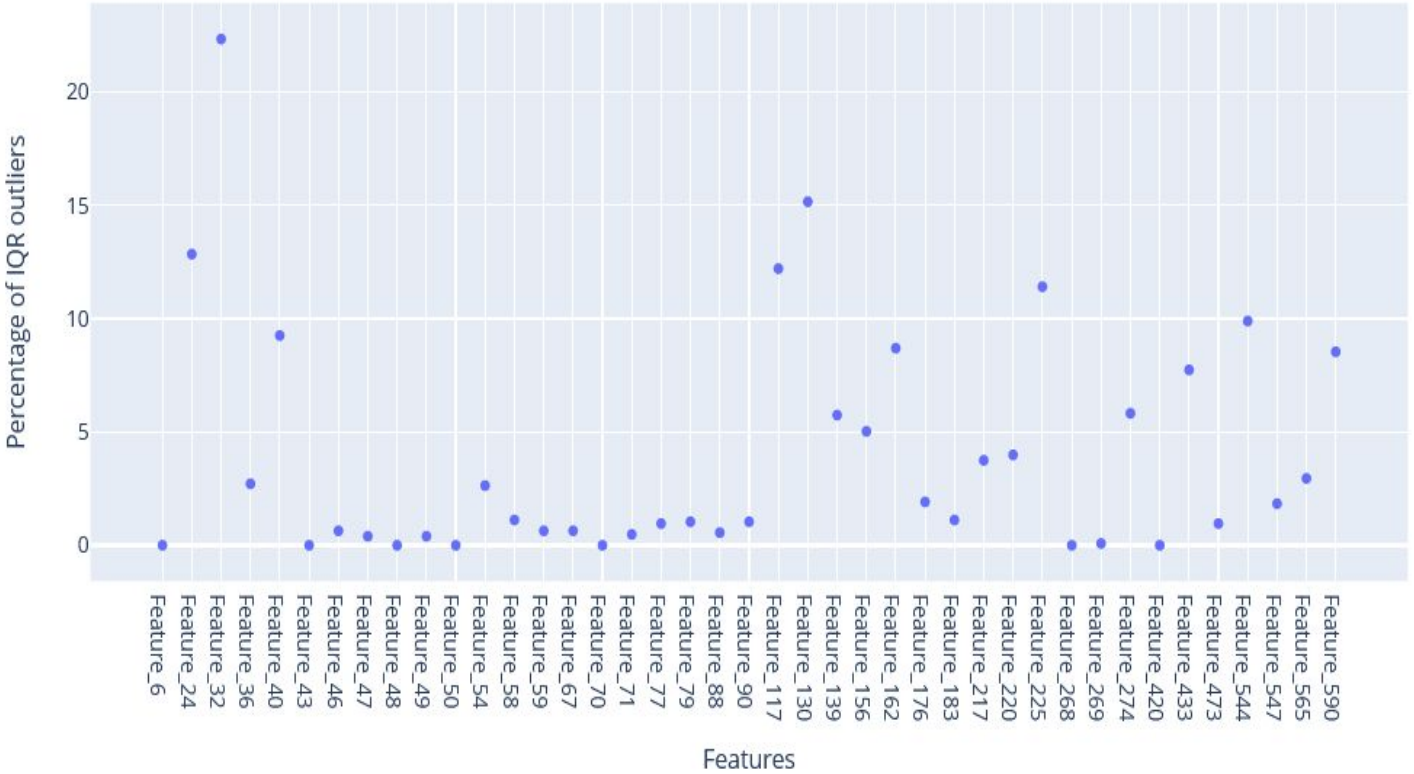
Outlier Analysis

```
def IQR_outliers(data,limit=1.5):
    numColumns = data.select_dtypes(include=np.number).columns.tolist();
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3-Q1;
    outliers=((data[numColumns] < (Q1 - limit*IQR)) | (data[numColumns] > (Q3 + limit*IQR))).sum()*100/data.shape[0]
    return outliers
```

```
fig3 = plt.figure("Figure 3")
outliers_train = IQR_outliers(X_train_corw)
outliers_train.plot(kind = 'box')
plt.show()
```



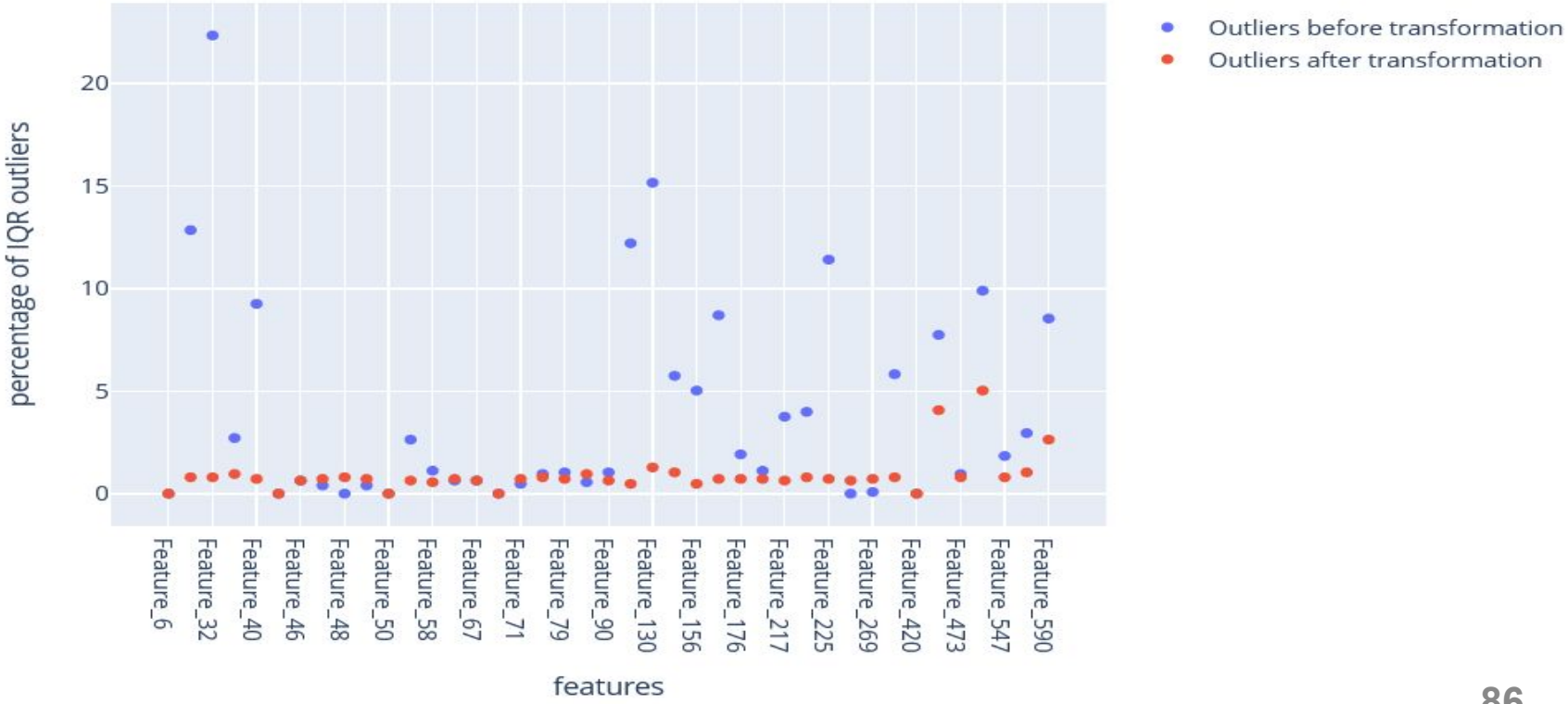
Outlier Analysis



Outlier Transformation

```
from sklearn.preprocessing import QuantileTransformer
quantile_transformer = QuantileTransformer(output_distribution='normal', random_state=42)
df1 = pd.DataFrame(quantile_transformer.fit_transform(X_train_corw),columns=X_train_corw.columns)
outliers = IQR_outliers(X_train_corw)
outliers1=IQR_outliers(df1)
```

Outlier Transformation



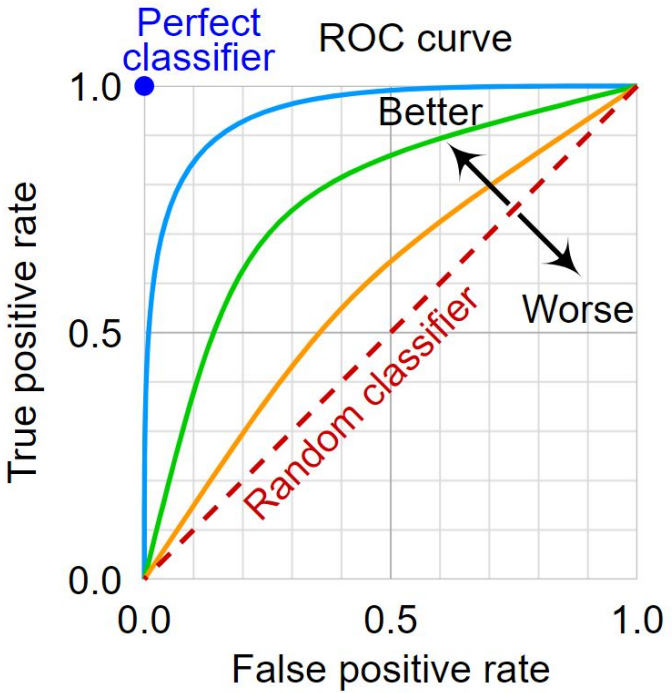
Result Measures

AUC/ ROC

$$\frac{TP + TN}{TP + FP + TN + FN}$$

F1-Score

$$\frac{2 \times Precision \times Recall}{Precision + Recall}$$



ıde off

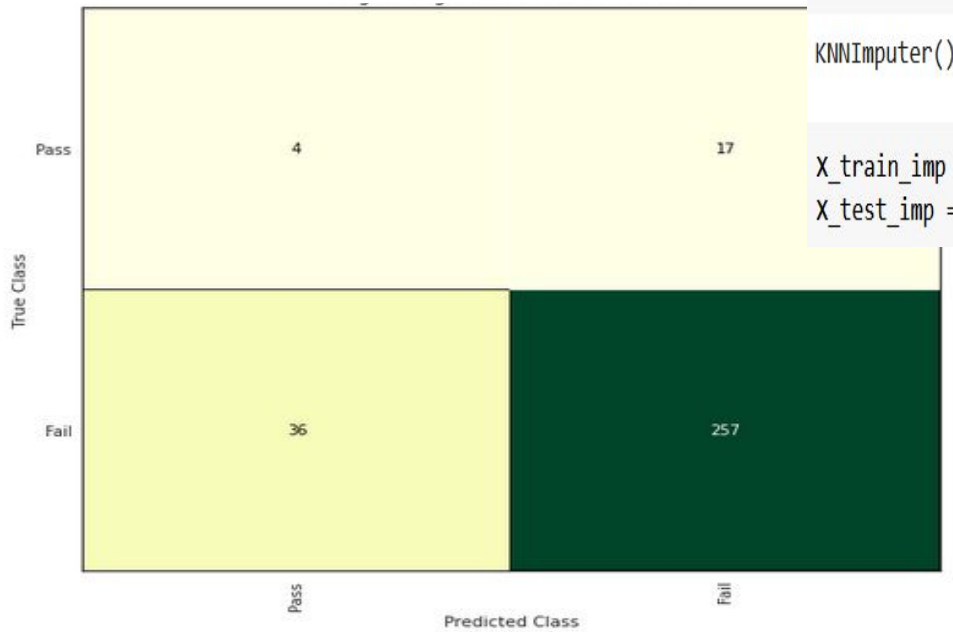
Feature Selection

2. KNN Imputer

```
imputer = KNNImputer()
imputer.fit(X_train_na)
```

```
KNNImputer()
```

```
X_train_imp = pd.DataFrame(imputer.transform(X_train_na), columns = X_train_na.columns)
X_test_imp = pd.DataFrame(imputer.transform(X_test_na), columns = X_test_na.columns)
```




```
F1 Score: 0.8312101910828026
 MCC Score: 0.05065759831945425
```

Feature Selection

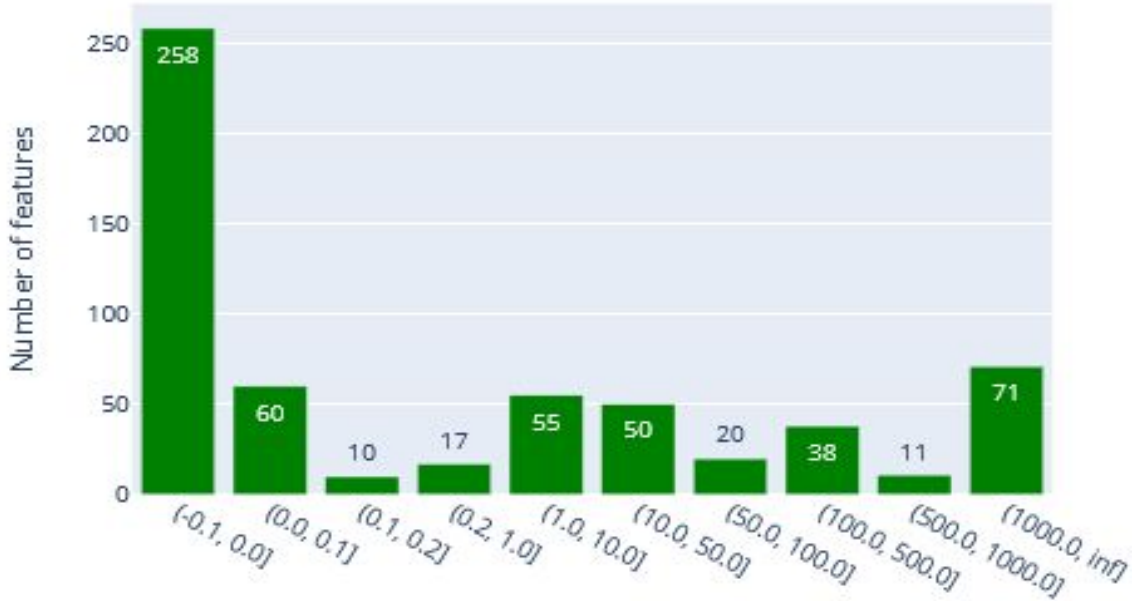
3. Normalization and Variance Threshold

```
normalizer = Normalizer()
normalizer.fit(X_train_imp)
```

```
Normalizer()
```

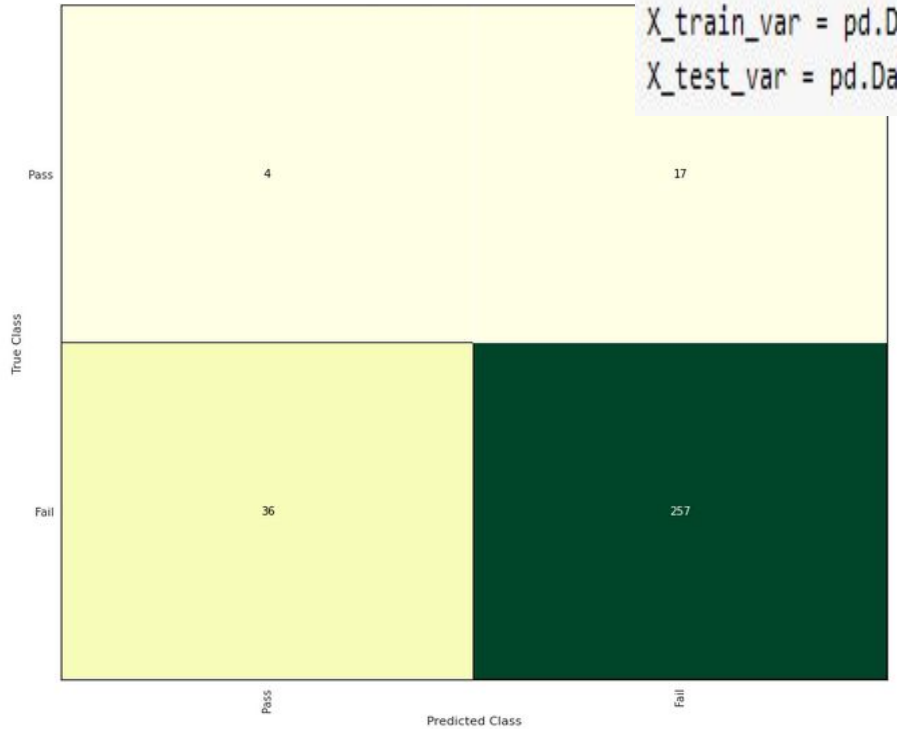
```
vr = VarianceThreshold()
vr.fit(X_train_nrm)
```

```
VarianceThreshold()
```



3. Variance Threshold

```
X_train_var = pd.DataFrame(vr.transform(X_train_imp), columns = selected_cols)
X_test_var = pd.DataFrame(vr.transform(X_test_imp), columns = selected_cols)
```




```
F1 Score: 0.8312101910828026
MCC Score: 0.05065759831945425
```

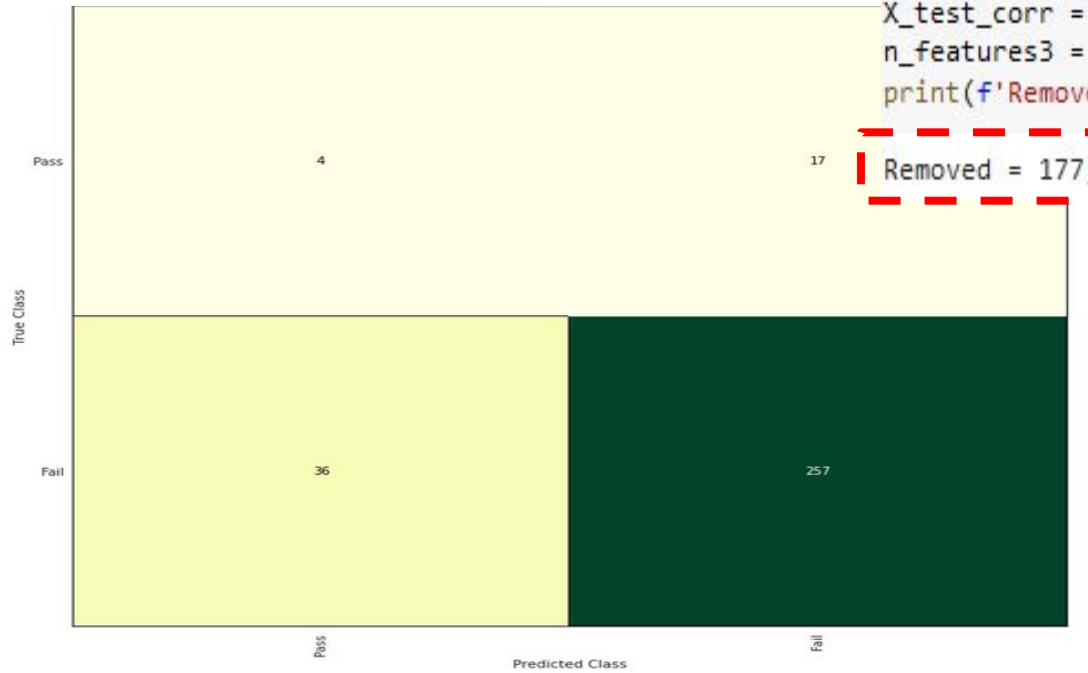
```
remaining features: 446
```

Feature Selection

4. Correlation pairs

```
corr_features = correlation(X_train_var, 0.95)
X_train_corr = X_train_var.drop(corr_features, axis=1)
X_test_corr = X_test_var.drop(corr_features, axis=1)
n_features3 = X_train_corr.shape[1]
print(f'Removed = {len(corr_features)}, Left = {n_features3}')
```

```
Removed = 177, Left = 269
```



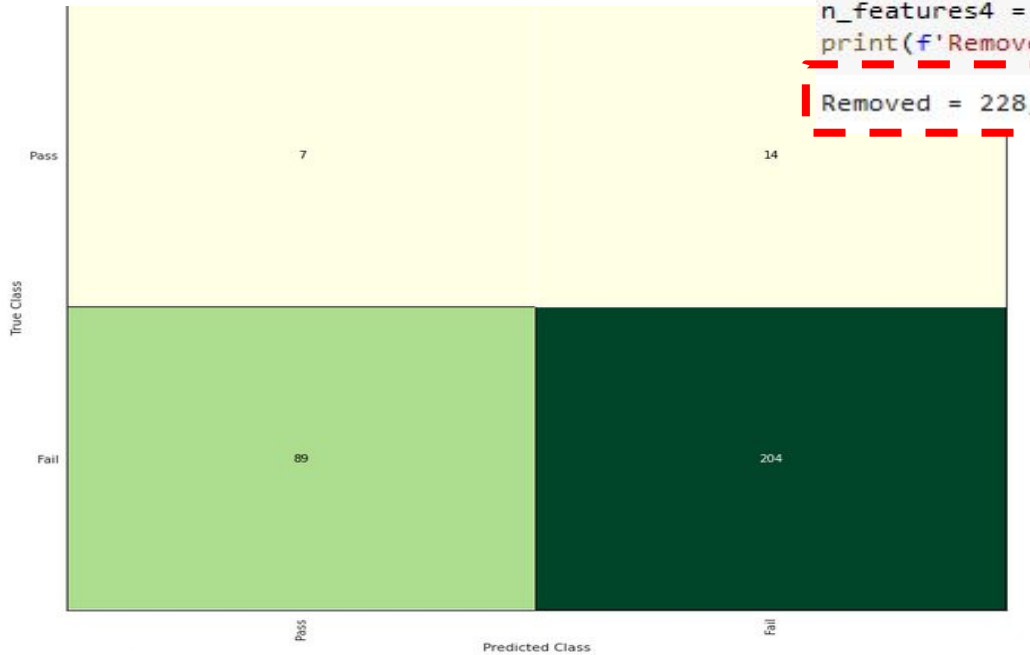

```
F1 Score: 0.8312101910828026
MCC Score: 0.05065759831945425
```

Feature Selection

5. Correlation with Target

```
corrwith_cols = corrwith_target(dummy_train, 'target', 0.05)
X_train_corw = X_train_corr.drop(corrwith_cols, axis=1)
X_test_corw = X_test_corr.drop(corrwith_cols, axis=1)
n_features4 = X_train_corw.shape[1]
print(f'Removed = {len(corrwith_cols)}, Left = {n_features4}')
```

```
Removed = 228, Left = 41
```




```
F1 Score: 0.6719745222929936
MCC Score: 0.016038511767519862
```

Result measurements

CONFUSION MATRIX

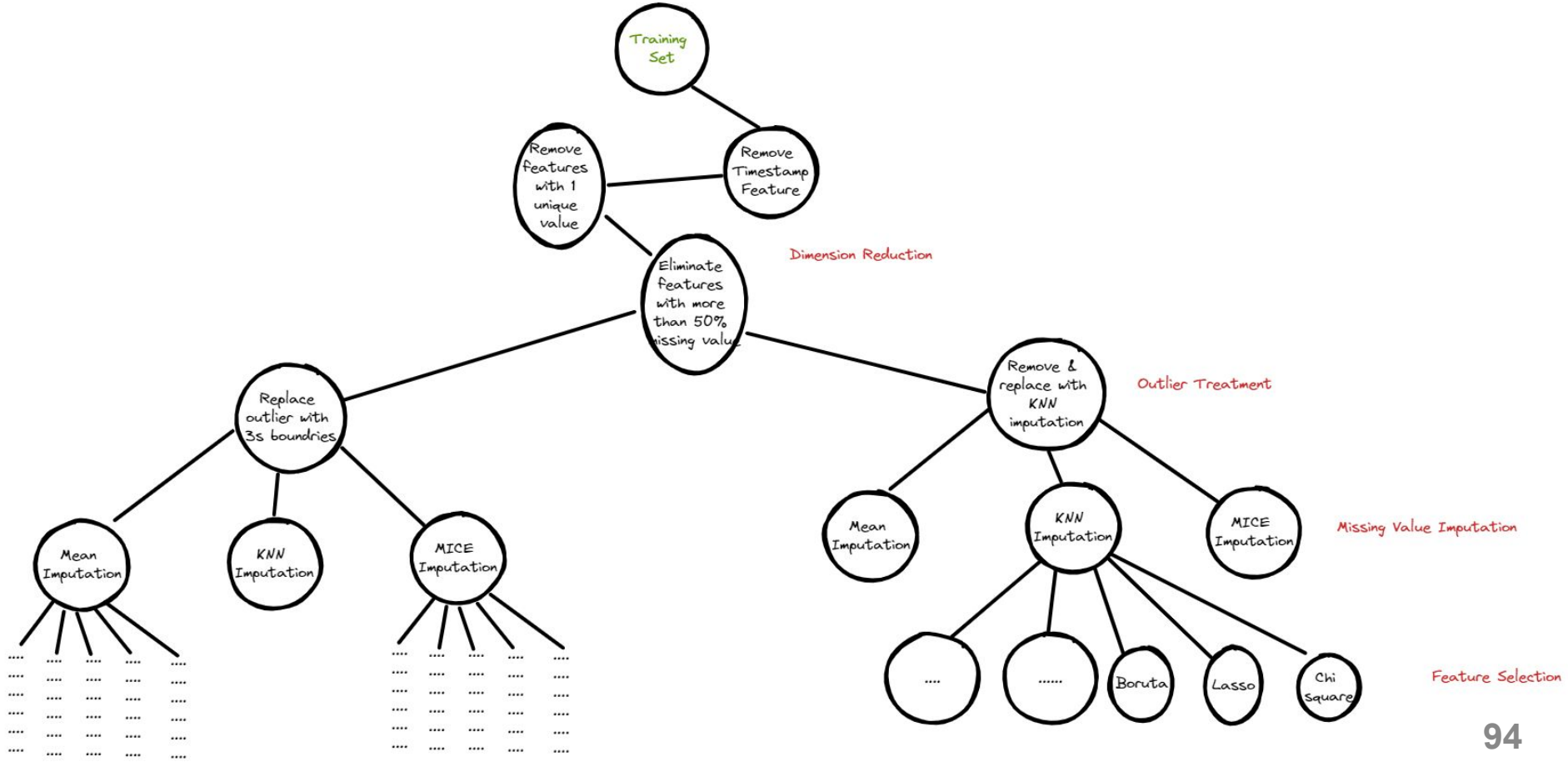
A confusion matrix allows visualisation of the performance of an algorithm. Each column of the Matrix represents the instances in predicted class while each row represents actual class or vice versa.

The most informative metric to evaluate a confusion metric is the Matthews correlation coefficient (MCC).

MCC is a correlation coefficient between observed and predicted binary classifications, returns a value between 1 and -1.
A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and -1 indicates total disagreement between prediction and observation.

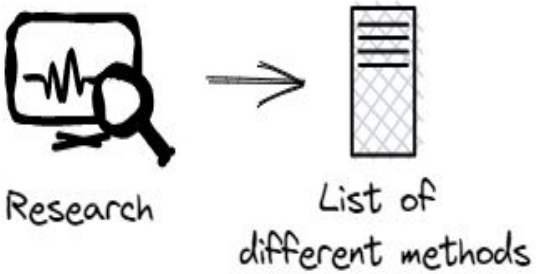
F score test's accuracy of binary classification. It does not take negative values in account

Our Approach for Data Preparation



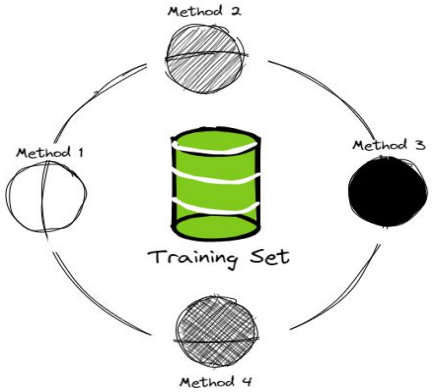
DATA PREPARATION APPROACH

01



Research different methods for each step of data preparation

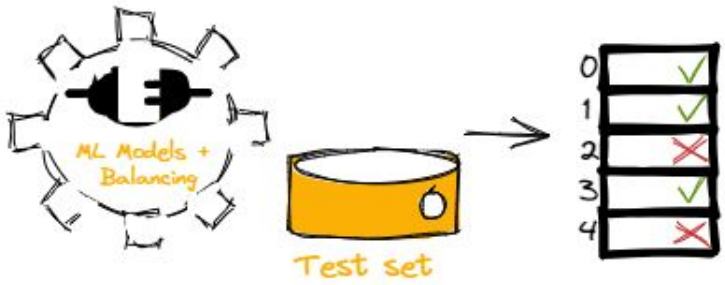
02



Use training set to train the machine with different method

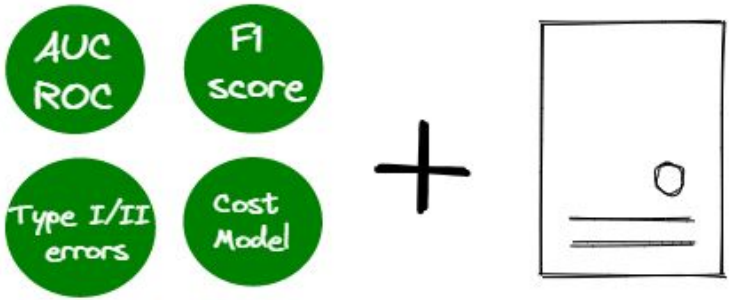
DATA PREPARATION APPROACH

03



Use different machine learning models, balancing methods & test set to test each method

04



Use AUC, ROC, type I/II errors, F1 score & cost model as measures for decision making & document our findings

01
Lorem ipsum dolor sit amet, Consetetur ester Sadipscing magna

Lorem ipsum dolor sit amet, Consetetur ester magnat

Lorem ipsum dolor sit amet, consectetuer adipiscing elit et altera dominus estis.

Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, At vero eos et accusam et justo duo dolores et easit amet commodo magna eros quis quod plusquam erat demonstrandum.

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa.

01

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

02

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

03

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

01

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Et accusam et justo duo dolores et ea rebum sine fins erat demonstradum

02

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Et accusam et justo duo dolores et ea rebum sine fins erat demonstradum

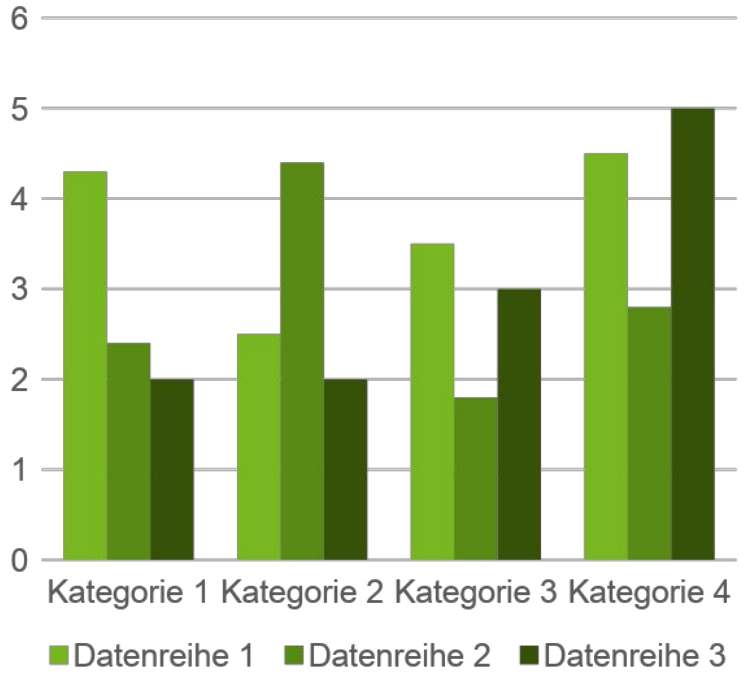
01
Lorem ipsum dolor sit amet, Consetetur ester Sadipscing magna

Lorem ipsum dolor sit

Lorem ipsum dolor sit amet, consectetuer adipiscing elit et eternam etipisis.

Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, At vero eos et accusam et justo duo dolores et easit amet commodo magna eros quis urna.

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa



Lorem ipsum dolor sit

Lorem ipsum dolor sit amet, consectetuer adipiscing elit et eternam etipisis.

Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, At vero eos et accusam et justo duo dolores et easit amet commodo magna eros quis magna sed pulvinar ultricies, purus lectus malesuada urna est.


C
11


C
11

Lorem ipsum dolor sit amet, Consetetur ester

Fusce posuere, At vero eos et accusam et justo duo magna sed pulvinar ultricies, purus lectus malesuada libero, At vero eos et accusam et justo duo dolores et easit amet commodo magna eros quis

Fusce posuere, At vero eos et accusam et justo duo magna sed pulvinar ultricies, purus lectus malesuada libero, At vero eos et accusam et justo duo dolores et easit amet commodo magna eros quis

01

Lorem ipsum dolor sit amet, Consetetur ester Sadipscing magna

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Et accusam et justo duo dolores et ea rebum sine fins erat demonstradum

Master
Project Management
and Data Science

Vielen Dank!

Name Referent*in
Kontakt und weitere Informationen

www.mpmd.htw-berlin.de

Master
Project Management
and Data Science

Vielen Dank!

Name Referent*in
Kontakt und weitere Informationen

www.mpmd.htw-berlin.de

Master
Project Management
and Data Science

Vielen Dank!

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

www.mpmd.htw-berlin.de

Master
Project Management
and Data Science

Vielen Dank!

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

www.mpmd.htw-berlin.de

Master
Project Management
and Data Science

Vielen Dank!

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

Name Referent*in
Kontakt und weitere Informationen

www.mpmd.htw-berlin.de

Problem with Imbalance Dataset

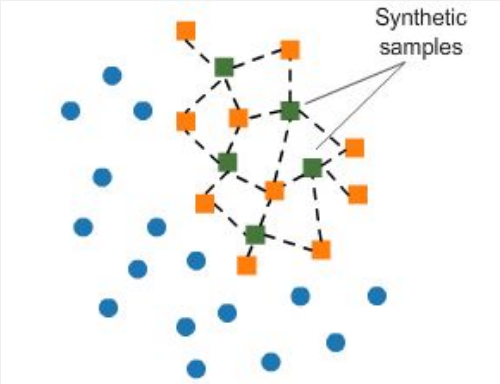
Imbalance Dataset Problem:

1. Standard learners are often biased towards the majority class
2. reason is because these classifiers attempt to reduce overall quantities such as error rate, not taking the data distribution into consideration
3. => tend to bias performance towards the majority class

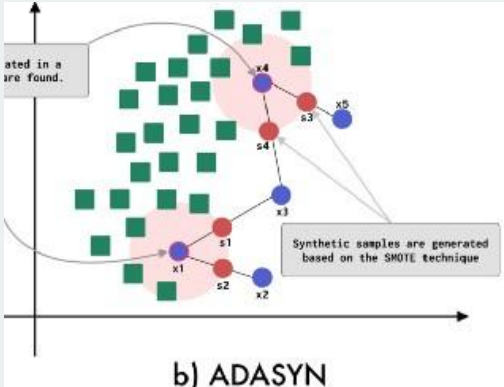
Solutions:

- Re-sampling provides a simple way of biasing the generalization process.
- It can do so by:
    - Generating synthetic samples accordingly biased
    - Controlling the amount and placement of the new samples
- Note: this type of control can also be achieved by smoothing the classifier's probabilistic estimate (e.g., Zadrozny & Elkan, 2001), but that type of control cannot be as localized as the one achieved with re-sampling techniques.
- Methods we use:
    - SMOTE
    - ADASYN
    - ROSE

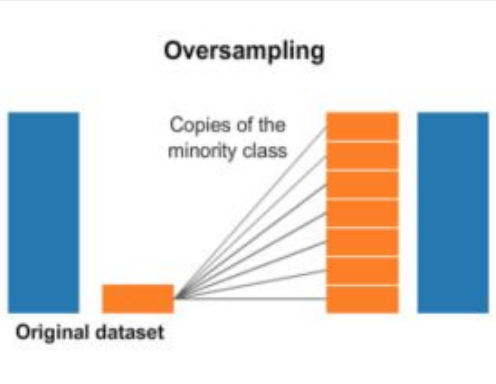
Balancing Methods



SMOTE



ADASYN



ROSE

Synthetic Minority Oversampling Technique (SMOTE)

It combines Informed Oversampling of the minority class with Random Undersampling of the majority class.

Adaptive Synthetic (ADASYN)

It uses a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn

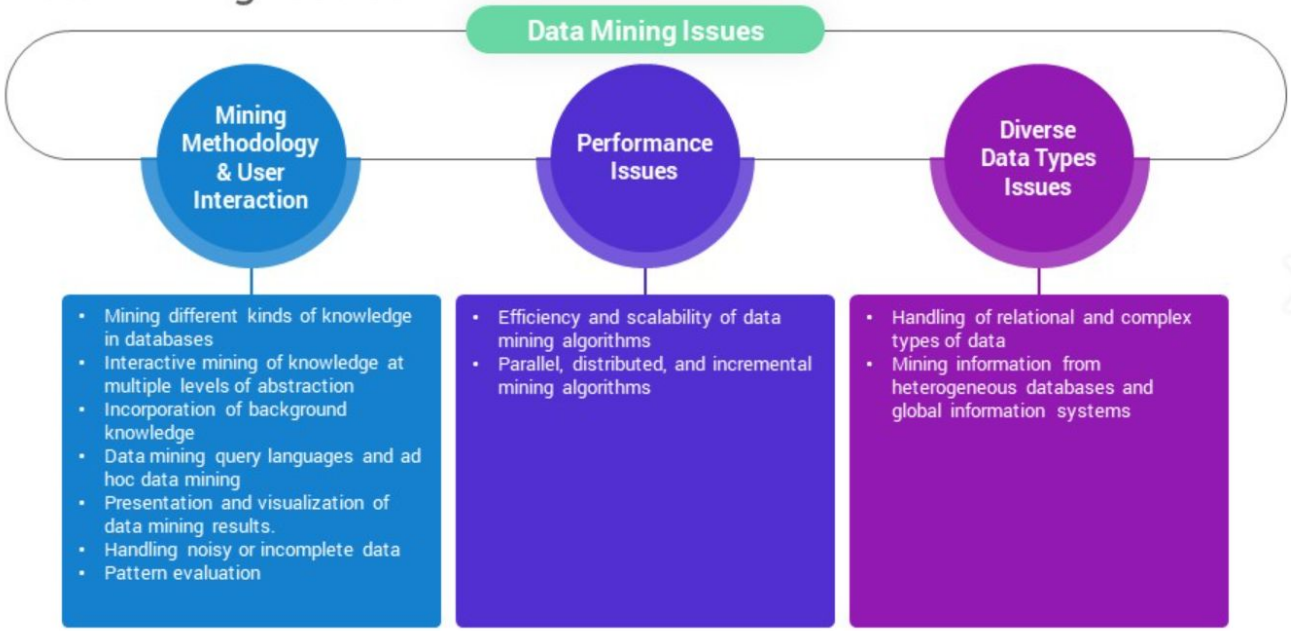
Random Over-Sampling Examples (ROSE)

It is a bootstrap-based technique which aids the task of binary classification in the presence of rare classes.

Overfitting Models

The risk of overfitting increases with a rise in volume and variety. The result is the model begins to show natural errors in a sample instead of displaying underlying trends. Lowering the number of variables results in an irrelevant model, while adding too many restrict the model. The challenge is finding the right moderation of the variables used and their balance in predictive accuracy.

Data Mining Issues



Synthetic Minority Oversampling Technique (SMOTE)

Description:
SMOTE stands for Synthetic Minority Oversampling Technique.
It is a technique designed by Chawla, Hall, & Kegelmeyer in 2002.
It combines Informed Oversampling of the minority class with Random Undersampling of the majority class.
SMOTE currently yields the best results as far as re-sampling and modifying the probabilistic estimate techniques go (Chawla, 2003).
For each minority Sample

- Find its k-nearest minority neighbours
- Randomly select j of these neighbours
- Randomly generate synthetic samples along the lines joining the minority sample and its j selected neighbours

(j depends on the amount of oversampling desired)

Advantage:
SMOTE's informed oversampling generalizes the decision region for the minority class. As a result, larger and less specific regions are learned, thus, paying attention to minority class samples without causing overfitting.

Disadvantage:
1. Overgeneralization:

- SMOTE's procedure is inherently dangerous since it blindly generalizes the minority area without regard to the majority class.
- This strategy is particularly problematic in the case of highly skewed class distributions since, in such cases, the minority class is very sparse with respect to the majority class, thus resulting in a greater chance of class mixture.

2. Lack of Flexibility

- The number of synthetic samples generated by SMOTE is fixed in advance, thus not allowing for any flexibility in the re-balancing rate.

Adaptive Synthetic (ADASYN)

a. **Description**:
The essential idea of ADASYN is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn. As a result, the ADASYN approach improves learning with respect to the data distributions in two ways: (1) reducing the bias introduced by the class imbalance, and (2) adaptively shifting the classification decision boundary toward the difficultexamples.

b. **Advantage**:
The biggest advantages of ADASYN are it's adaptive nature of creating more data for "harder-to-learn" examples and allowing you to sample more negative data for your model.

c. **Disadvantage**:

There are two major weaknesses of ADASYN:

- For minority examples that are sparsely distributed, each neighbourhood may only contain 1 minority example.
- Precision of ADASYN may suffer due to adaptability nature.

SPE

Random Forest

a. Description

Random forest is one of the most popular tree-based supervised learning algorithms. It is also the most flexible and easy to use.

The algorithm can be used to solved both classification and regression problems. Random forest tends to combine hundreds of decision trees and then trains each decision tree on a different sample of the observations

The final predictions of the random forest are made by averaging the predictions of each individual tree

b. Advantage:

The benefits of random forests are numerous. The individual decision trees tend to overfit the training data but the random forest can mitigate that issue by averaging the prediction results from different trees. This gives random forests a higher predictive accuracy than a single decision tree

c. Disadvantage:

…

MACHINE LEARNING MODELS

What and how we applied these models in SECOM? Why we chose certain scaling over the other?(show code or reasons…)

Explain the steps of the modelling process, the implementation, the evaluation and the details of your preferred model.

QUYNH-ML models in 3 slides DONE

EVALUATION PROCESS

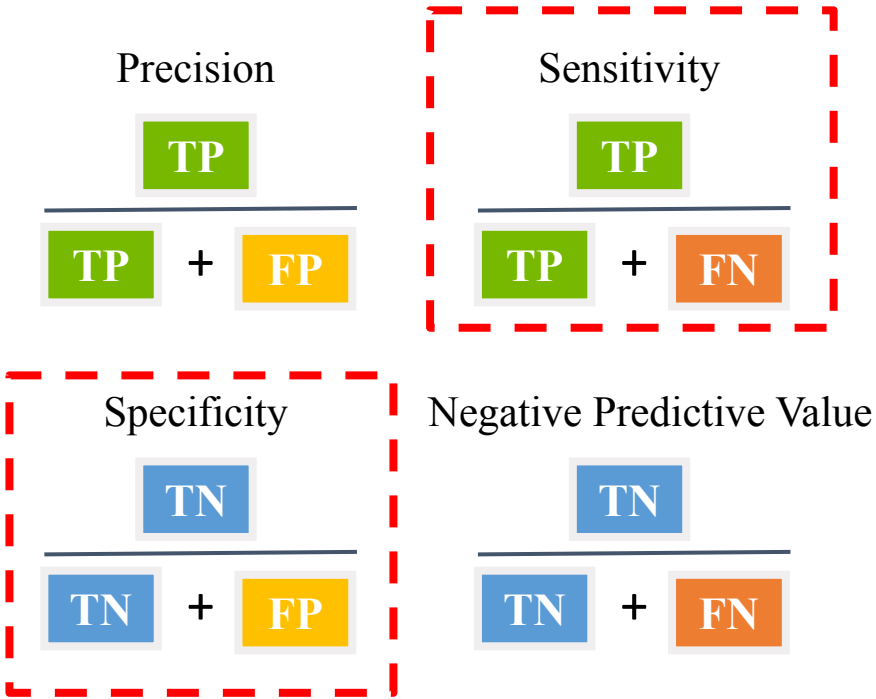
1. list of all combination (imputation, feature selection, balancing)
2. choose ML model => train the machine
3. use test set to test the result
4. assess the confusion matrix, roc, accuracy and recall
5. fine tune with grid search
6. arrive at the best result for the combination and ML model => rec
7. Repeat step 2-5 with another ML model
8. pick the 3 best results

RASHMI- ALL COMBINATIONS of how we evaluate the models based on

DECISION MAKING

HIMANSHA TUNING

RESULT MEASURES



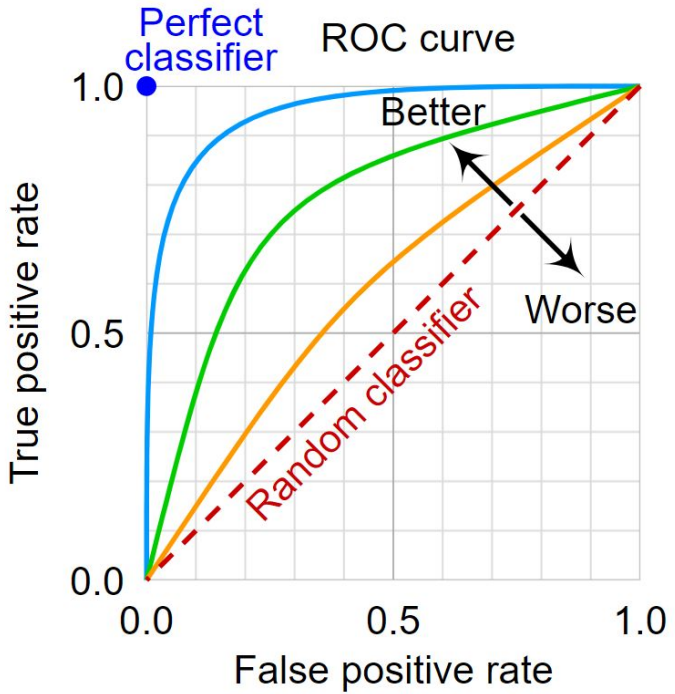
- **Sensitivity/Recall**: the ratio between how much were correctly identified as positive to how much were actually positive.
- **Specificity:** the ratio between how much were correctly classified as negative to how much was actually negative.
- **Precision:**How much were correctly classified as positive out of all positives.
- **NPV**:How much were correctly classified as negative out of all negatives

F1-Score

$$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 Score: combines precision and recall into one metric by calculating the harmonic mean between those two. It is primarily used to compare the performance of two classifiers. Classifier A has a higher recall, and classifier B has higher precision. In this case, the F1-scores for both the classifiers are used to determine which one produces better results.

ROC CURVE



- ROC curve is used to visualize the trade off between Sensitivity and Specificity.
- AUC is the area under the curve
- **Sensitivity** on the **Y-axis**
- **Specificity** on the **X-axis**

CONFUSION MATRIX VS. COST MODEL

1. Describe how we choose the best model (how we decide to between accuracy vs cost)

Final Result

	F1 score	Sensitivity	Precision	Recall	ROC	Type I error	Type II error	Cost
Boruta - ROSE - Random Forest Classifier	0.87	0.95	0.25	0.42	0.71	12	27	$ 54,500
Forward Selection - SMOTE - Random Forest Classifier	0.78	0.94	0.11	0.33	0.62	21	6	$ 51,250
Boruta - SMOTE - SVM	0.87	0.95	0.25	0.42	0.71	12	27	$ 54,000
Boruta - ROSE - Decision Tree	0.91	0.95	0.33	0.33	0.73	14	14	$ 45,500
	0.91	0.95	0.33	0.42	0.73	12	6	45,500
	max	max	max					min

RASHMI

Best Model (our pick): KNN-Boruta-ROSE with Decision Tree algorithm

Support Vector Machine (SVM)

a. Description

RESULTS OF acc score roc

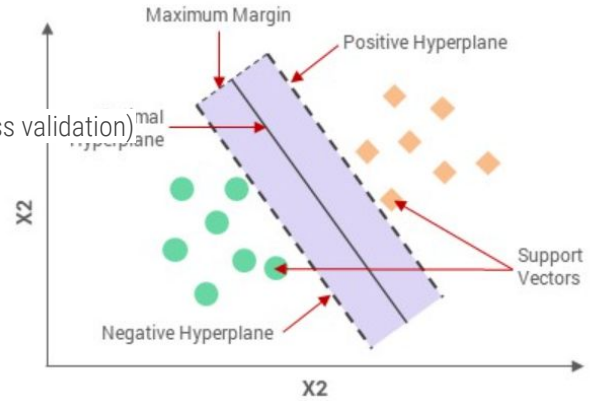
ssifier which takes in training data and output an optimal hyperplane which categorizes new

or decision boundary that can segregate n-dimensional space into classes so that we can
the future. This best decision boundary is called a hyperplane.

s is greater than numbers of samples
unction (called support vectors) => memory effective

he required training time is higher

se are calculated using an expensive 5-fold cross validation)

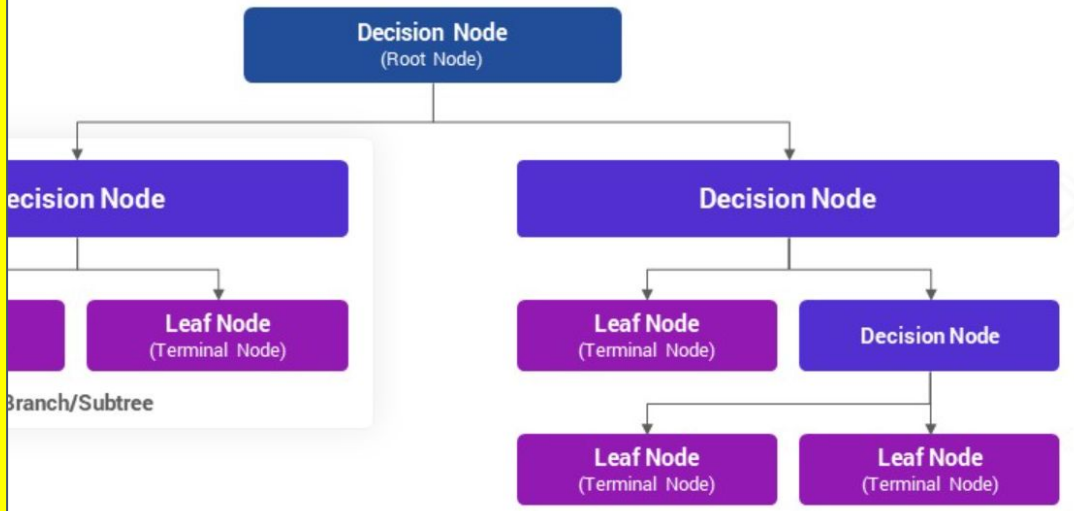


Decision Tree

a. Description

RESULTS OF acc score roc

ation and regression problems but mostly it is preferred for solving classification problems

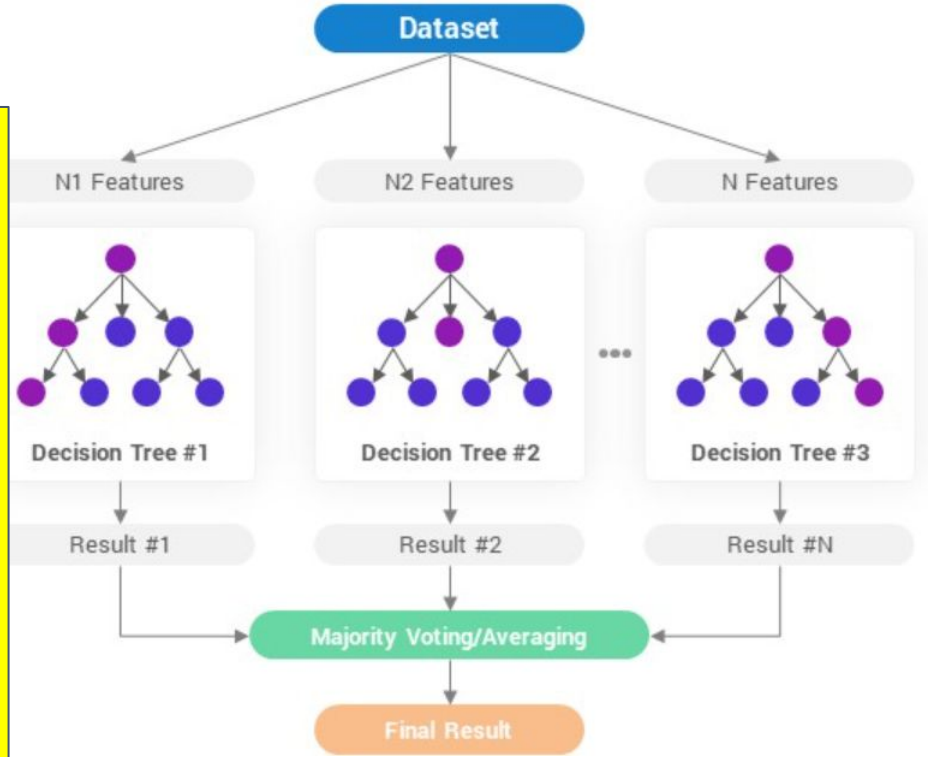


Random Forest

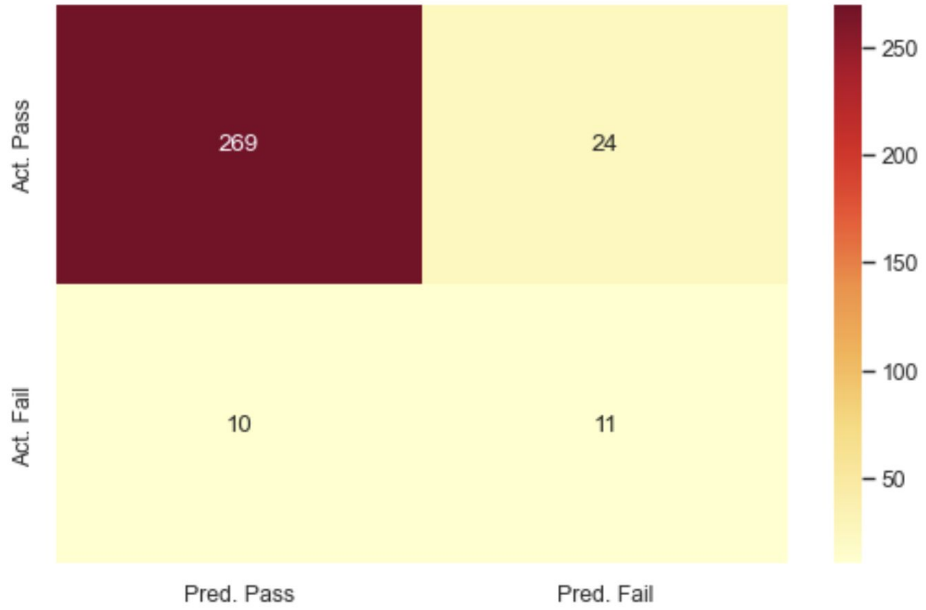
how does algorithm work?

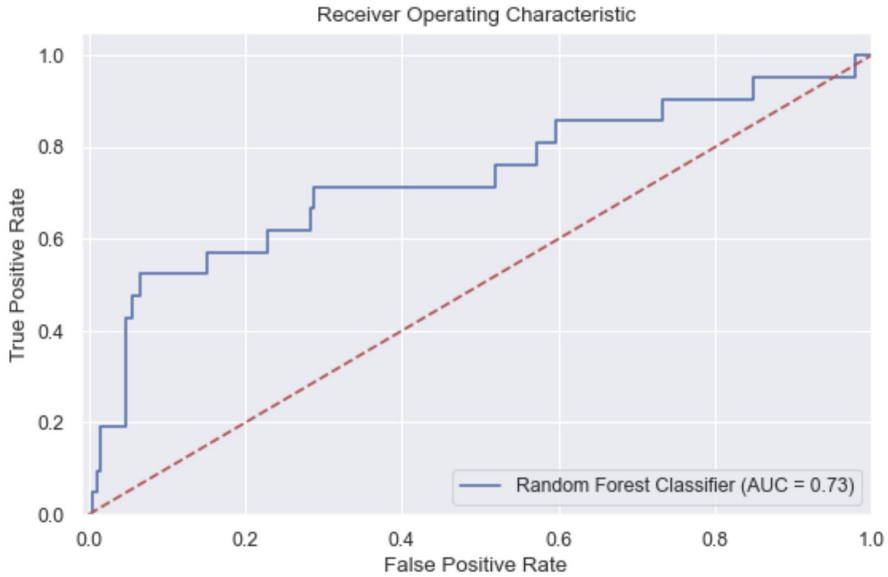
1. the algorithm selects random samples from the dataset provided
2. the algorithm will create a decision tree for each sample selected.
   then
3. votin
   class
   it wil
4. and
   resu

Random Forest RESULTS OF top 3 based on acc score, roc

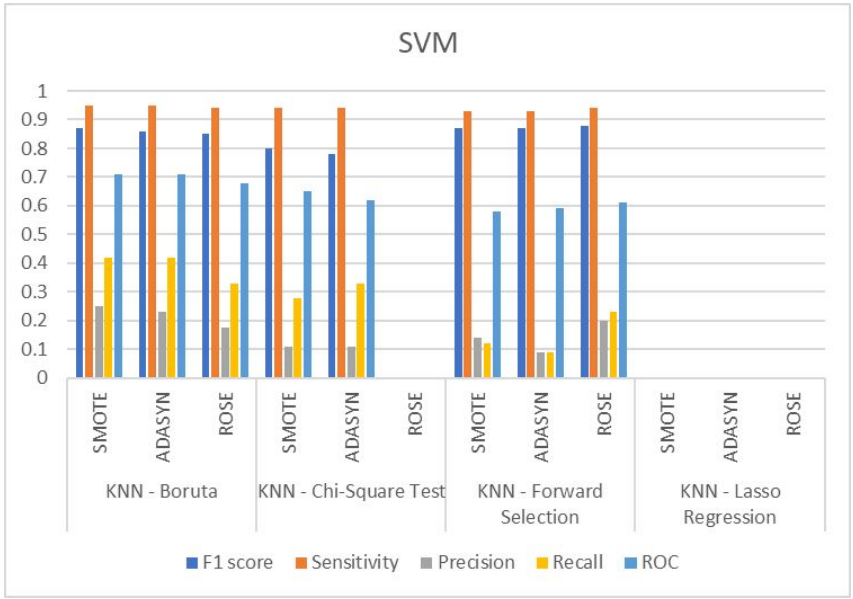


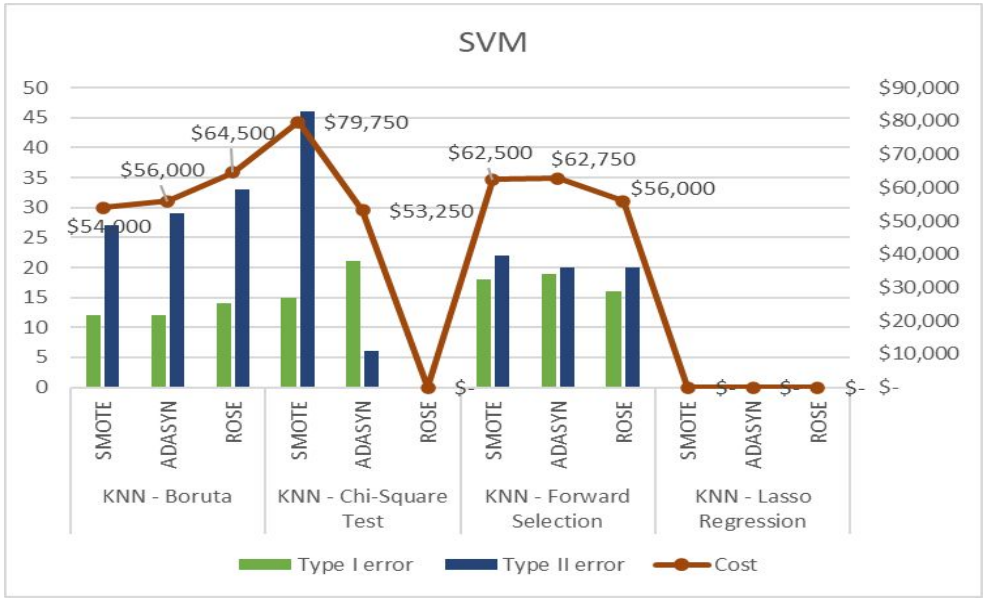
Final Result





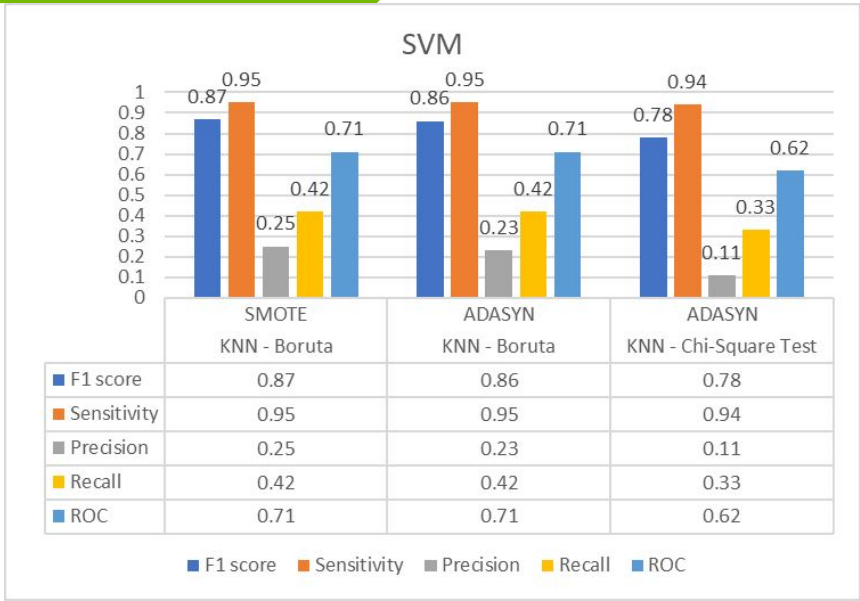
SVM Results



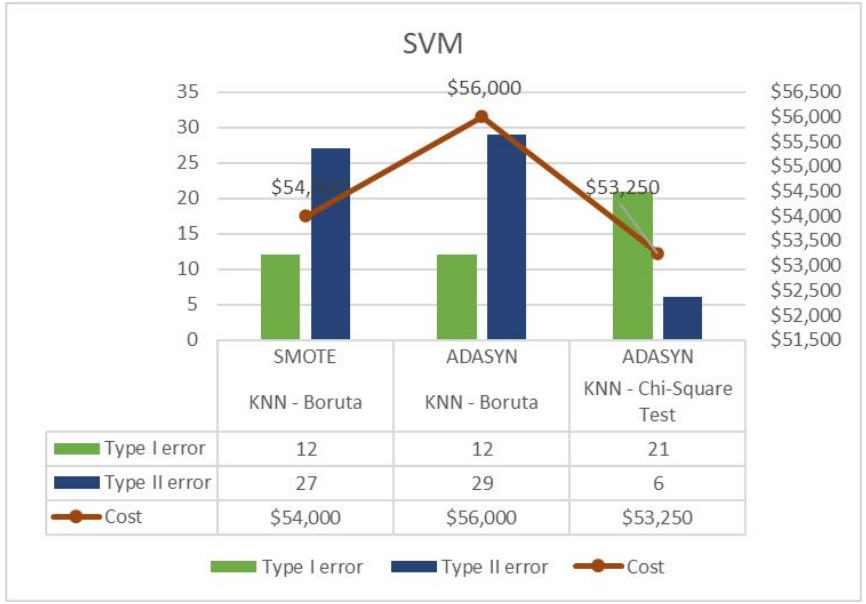


Best Model according to Precision/Recall: KNN-Boruta-SMOTE
Best Model according to Cost: KNN-Chi-ADASYN
=> our pick: KNN-Boruta-SMOTE

SVM Results



	SMOTE KNN - Boruta	ADASYN KNN - Boruta	ADASYN KNN - Chi-Square Test
F1 score	0.87	0.86	0.78
Sensitivity	0.95	0.95	0.94
Precision	0.25	0.23	0.11
Recall	0.42	0.42	0.33
ROC	0.71	0.71	0.62



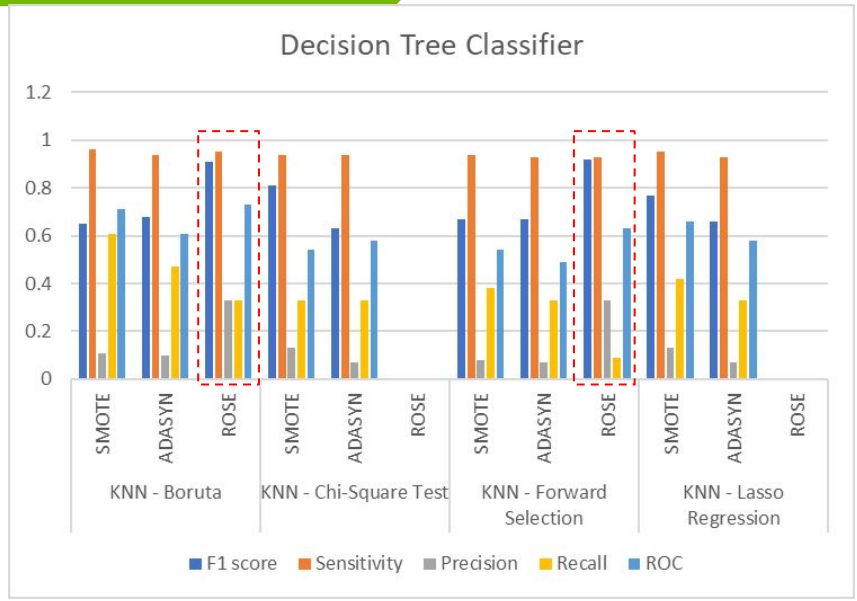
	SMOTE KNN - Boruta	ADASYN KNN - Boruta	ADASYN KNN - Chi-Square Test
Type I error	12	12	21
Type II error	27	29	6
Cost	$54,000	$56,000	$53,250

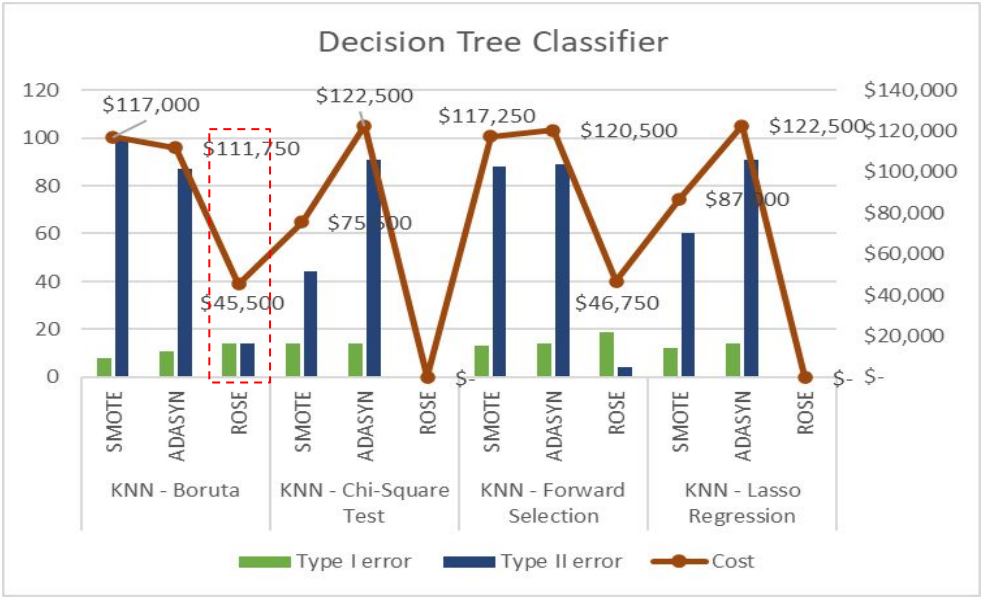
Best Model according to Precision/Recall: KNN-Boruta-SMOTE
Best Model according to Cost: KNN-Chisquare-ADASYN

- KNN-Boruta-SMOTE has better performance overall but the cost is a bit higher
- Accuracy or Cost is more important? Our decision: Higher cost is worth higher accuracy

=> our pick: KNN-Boruta-SMOTE

Decision Tree Results

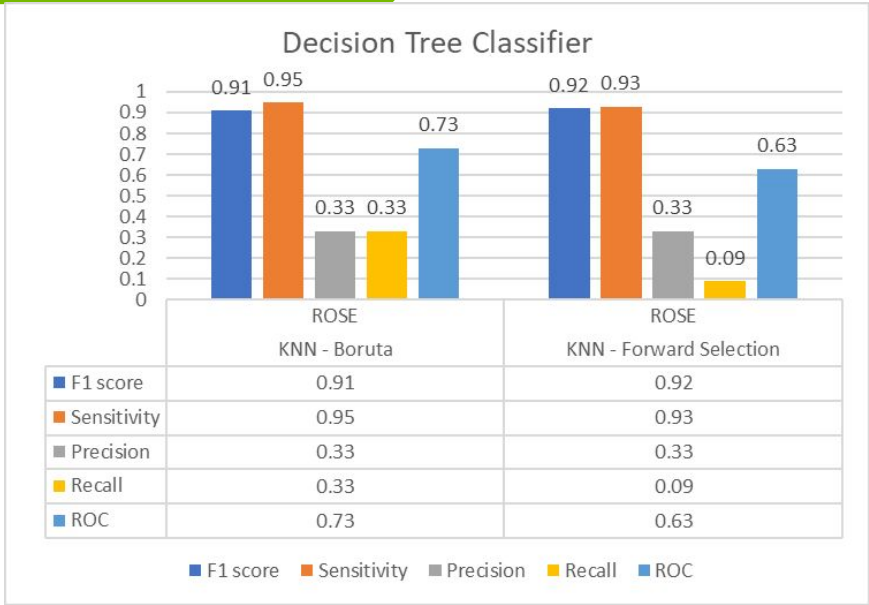


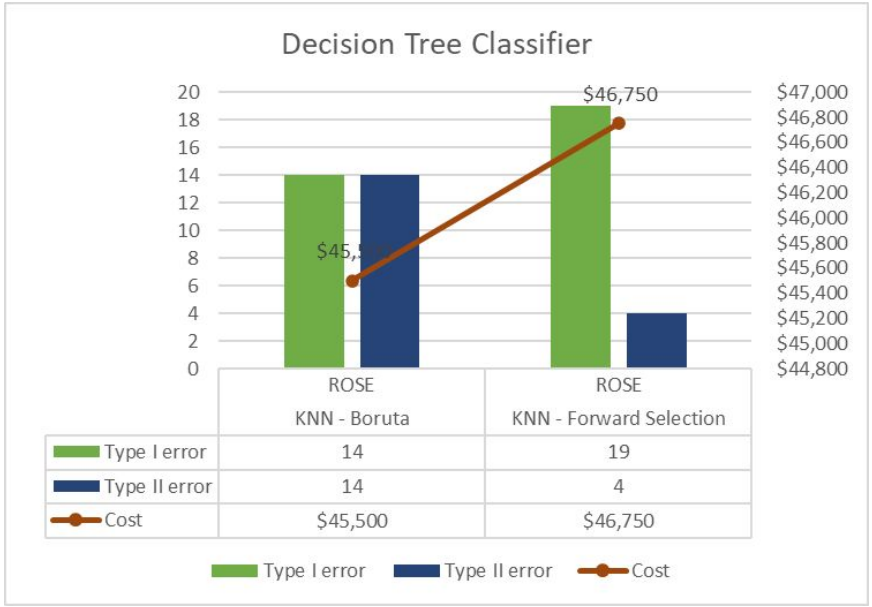


Best Model according to Precision/Recall:
KNN-Boruta-ROSE, KNN - Forward Selection - ROSE
Best Model according to Cost: KNN-Boruta-ROSE
=> our pick: KNN-Boruta-ROSE

Decision Tree Classifier	F1 score	Sensitivity	Precision	Recall	ROC	Type I error	Type II error	Cost
KNN - Boruta - SMOTE	0.65	0.96	0.11	0.61	0.71	8	99	$ 117,000
KNN - Boruta - ADASYN	0.68	0.94	0.1	0.47	0.61	11	87	$ 111,750
KNN - Boruta - ROSE	0.91	0.95	0.33	0.33	0.73	14	14	$ 45,500
KNN - Chi-Square Test - SMOTE	0.81	0.94	0.13	0.33	0.54	14	44	$ 75,500
KNN - Chi-Square Test - ADASYN	0.63	0.94	0.07	0.33	0.58	14	91	$ 122,500
KNN - Chi-Square Test - ROSE	Didnt work							
KNN - Forward Selection - SMOTE	0.67	0.94	0.08	0.38	0.54	13	88	$ 117,250
KNN - Forward Selection - ADASYN	0.67	0.93	0.07	0.33	0.49	14	89	$ 120,500
KNN - Forward Selection - ROSE	0.92	0.93	0.33	0.09	0.63	19	4	$ 46,750
KNN - Lasso Regression - SMOTE	0.77	0.95	0.13	0.42	0.66	12	60	$ 87,000
KNN - Lasso Regression - ADASYN	0.66	0.93	0.07	0.33	0.58	14	91	$ 122,500
KNN - Lasso Regression - ROSE	Didnt work							
	0.92	0.96	0.33	0.61	0.73	8	4	45,500
	max	max	max	max	max	min	min	min

Decision Tree Results





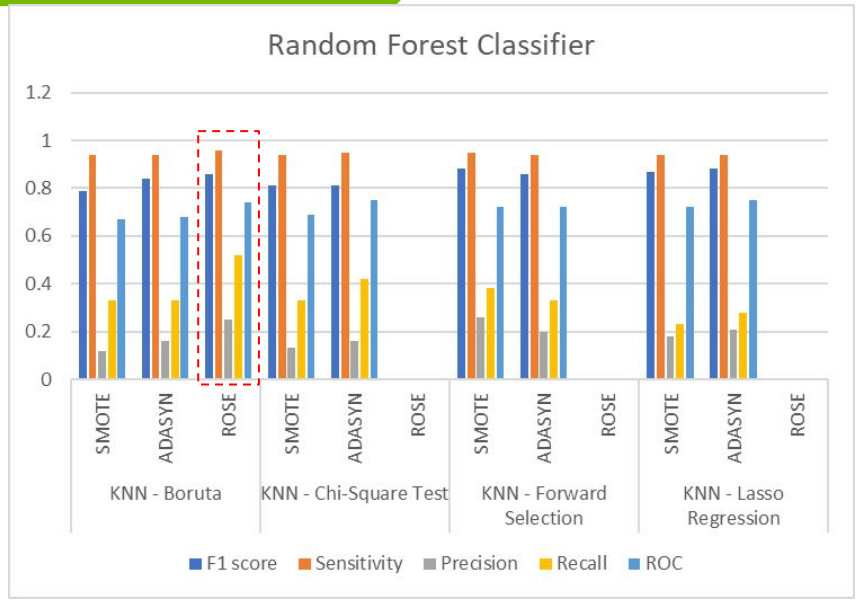
Best Model according to Precision/Recall: KNN-Boruta-ROSE, KNN - Forward Selection - ROSE

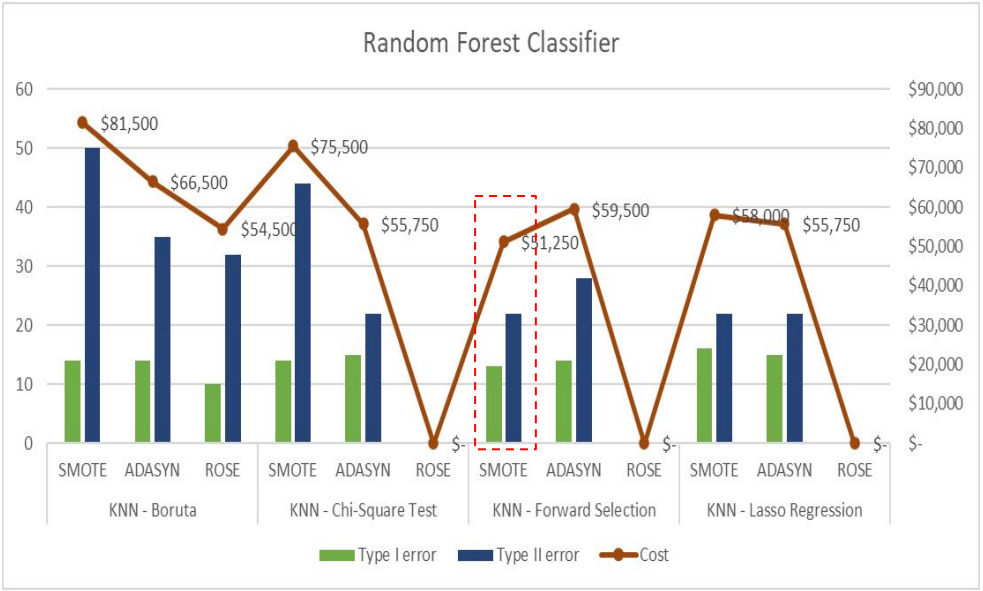
Best Model according to Cost: KNN-Boruta-ROSE

- KNN-Boruta-SMOTE has better performance overall and lower cost

=> our pick: KNN-Boruta-ROSE

Random Forest Classifier Results

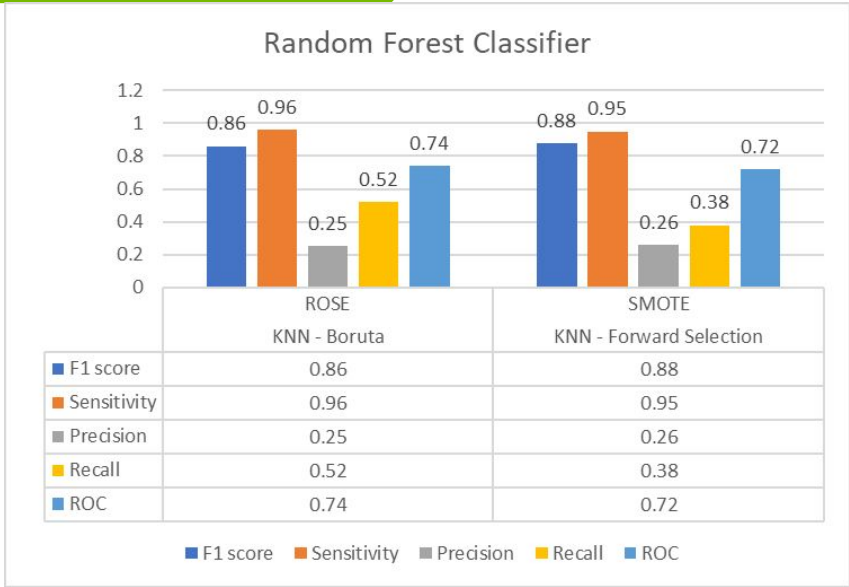


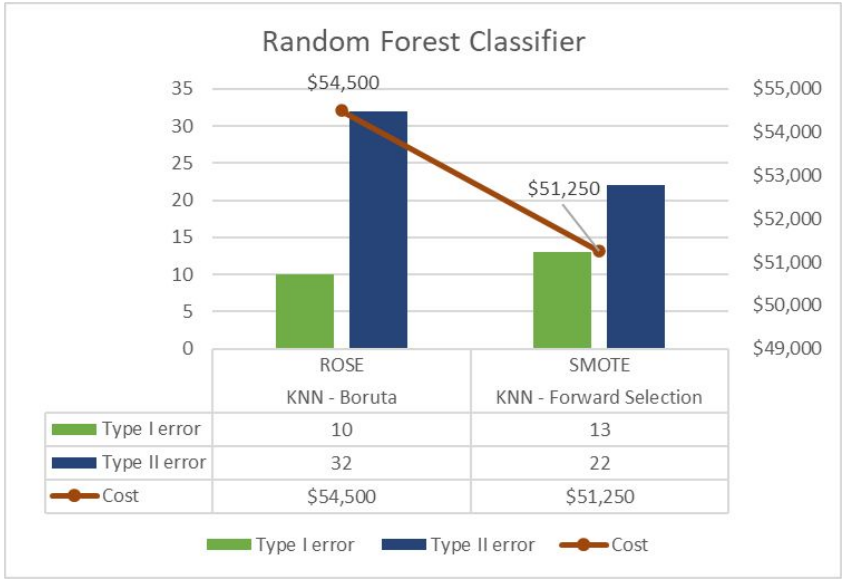


Best Model according to Precision/Recall:
KNN-Boruta-ROSE
Best Model according to Cost: KNN-FFS-SMOTE

Random Forest Classifier	F1 score	Sensitivity	Precision	Recall	ROC	Type I error	Type II error	Cost
KNN - Boruta - SMOTE	0.79	0.94	0.12	0.33	0.67	14	50	$ 81,500
KNN - Boruta - ADASYN	0.84	0.94	0.16	0.33	0.68	14	35	$ 66,500
KNN - Boruta - ROSE	0.86	0.96	0.25	0.52	0.74	10	32	$ 54,500
KNN - Chi-Square Test - SMOTE	0.81	0.94	0.13	0.33	0.69	14	44	$ 75,500
KNN - Chi-Square Test - ADASYN	0.81	0.95	0.16	0.42	0.75	15	22	$ 55,750
KNN - Chi-Square Test - ROSE	Didnt work							
KNN - Forward Selection - SMOTE	0.88	0.95	0.26	0.38	0.72	13	22	$ 51,250
KNN - Forward Selection - ADASYN	0.86	0.94	0.2	0.33	0.72	14	28	$ 59,500
KNN - Forward Selection - ROSE	Didnt work							
KNN - Lasso Regression - SMOTE	0.87	0.94	0.18	0.23	0.72	16	22	$ 58,000
KNN - Lasso Regression - ADASYN	0.88	0.94	0.21	0.28	0.75	15	22	$ 55,750
KNN - Lasso Regression - ROSE	Didnt work							
	0.88	0.96	0.26	0.52	0.75	10	22	51,250
	max	max	max	max	max	min	min	min

Random Forest Classifier Results





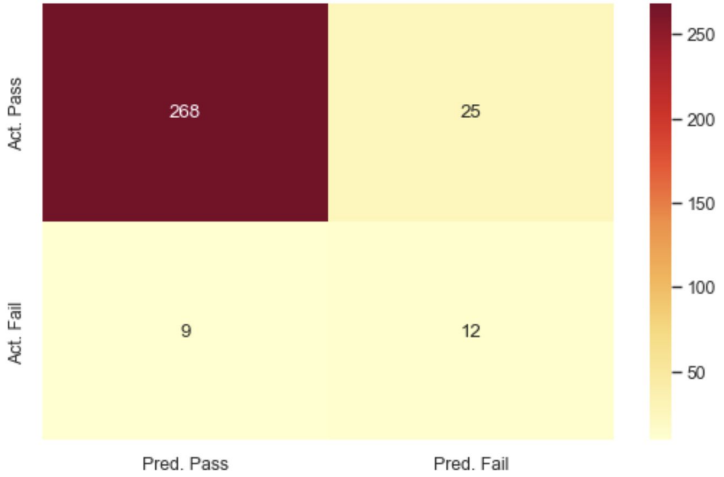
Best Model according to Precision/Recall: KNN-Boruta-ROSE
Best Model according to Cost: KNN-FFS-SMOTE

- KNN-Boruta-ROSE has better recall rate, lower type I error but the cost is a bit higher
- Accuracy or Cost is more important? Our decision: Higher cost is worth higher accuracy

=> our pick: KNN-Boruta-ROSE

Master
Project Management
and Data Science

FINAL RESULT

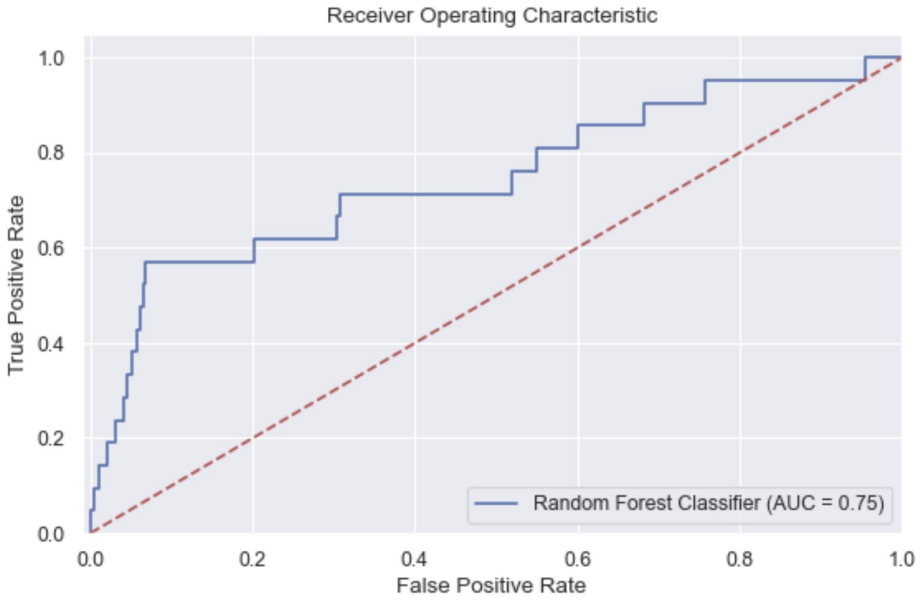


CONFUSION MATRIX

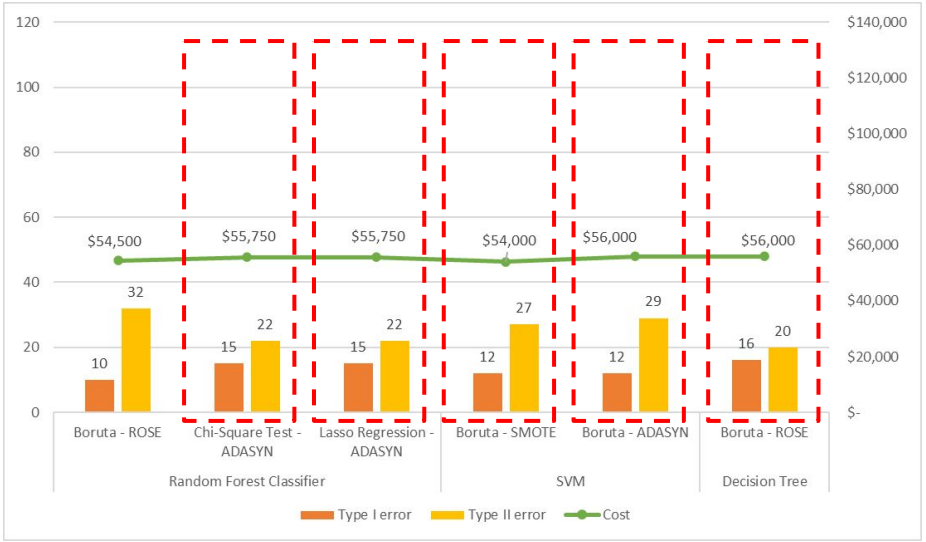
Model.....

AUC

Model.....



RESULTS WITH COST MODEL



Type 2 Error < 35

Lowest Type 1 Error

Best Model: KNN-Boruta-ROSE using Random Forest Classifier

- Type I error: 10
- Type II error: 32
- Cost: $54,500