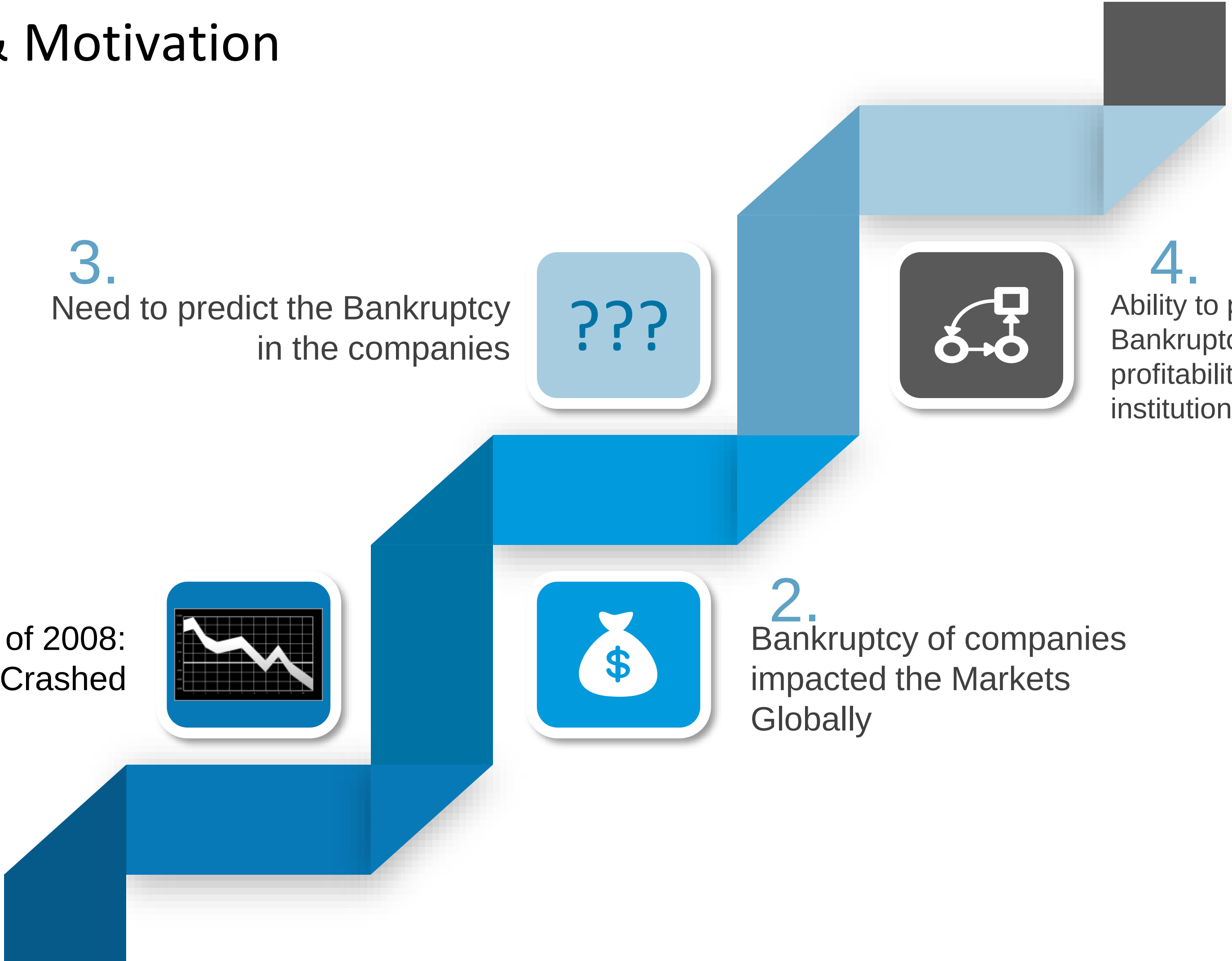
Company Bankruptcy Prediction

IDS 575 Group Project | Fall 2021

Submitted by: Anisha Joshi, Sweta Bansal

Problem & Motivation



1. Financial Crisis of 2008: Market Crashed

2. Bankruptcy of companies impacted the Markets Globally

3. Need to predict the Bankruptcy in the companies

4. Ability to predict Bankruptcy will impact the profitability of Lending institutions

Dataset

The dataset is about bankruptcy prediction of Polish companies. The bankrupt companies were analyzed in the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013.

64 Financial Ratios as Feature Set

10k Training Examples (98% of 0 and 2% of 1)

The training set with both predictors and response variable. Highly imbalanced dataset

5k Test Examples

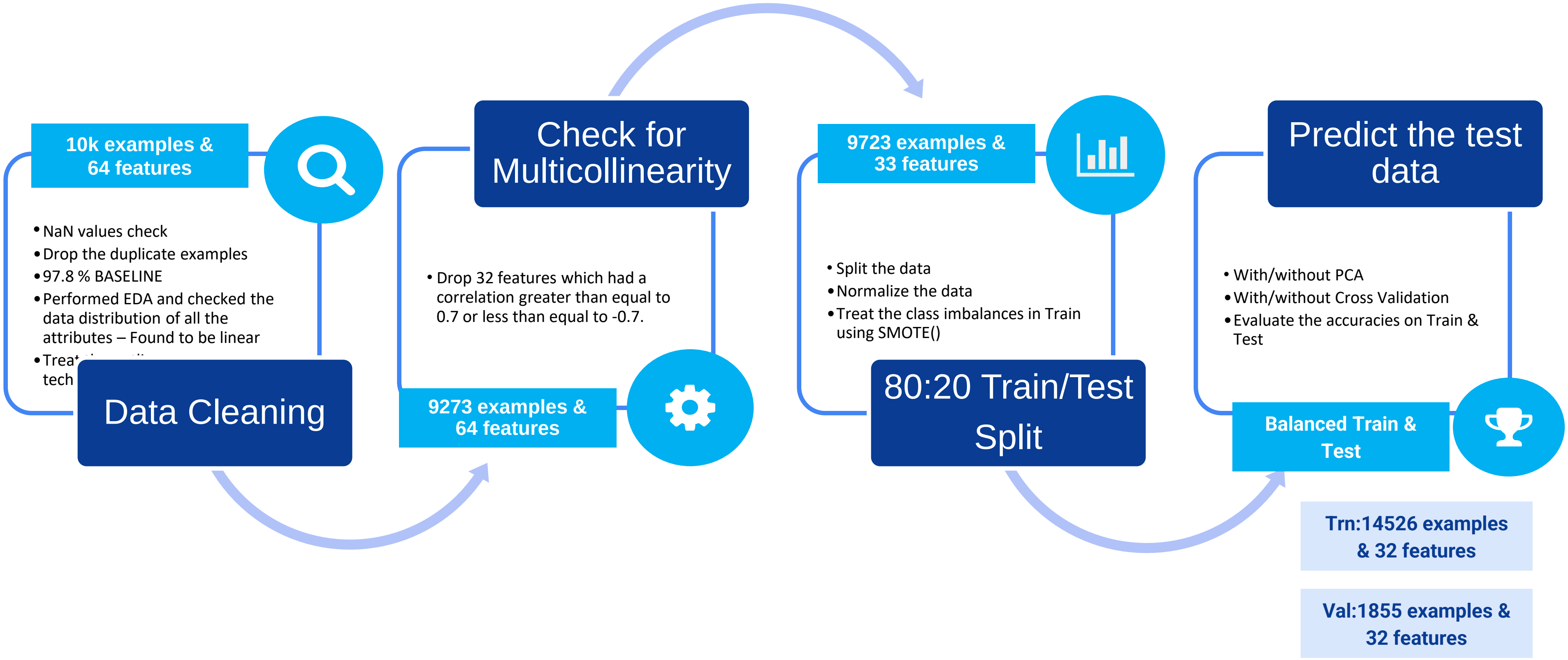
The test set with ID and predictors

0: Not Bankrupt
1: Bankrupt

The Response Variable : Y

0/1

Data Processing Steps



10k examples & 64 features

- NaN values check
- Drop the duplicate examples
- 97.8 % BASELINE
- Performed EDA and checked the data distribution of all the attributes – Found to be linear
- Treat tech

Data Cleaning

Check for Multicollinearity

- Drop 32 features which had a correlation greater than equal to 0.7 or less than equal to -0.7.

9273 examples & 64 features

9723 examples & 33 features

- Split the data
- Normalize the data
- Treat the class imbalances in Train using SMOTE()

80:20 Train/Test Split

Predict the test data

- With/without PCA
- With/without Cross Validation
- Evaluate the accuracies on Train & Test

Balanced Train & Test

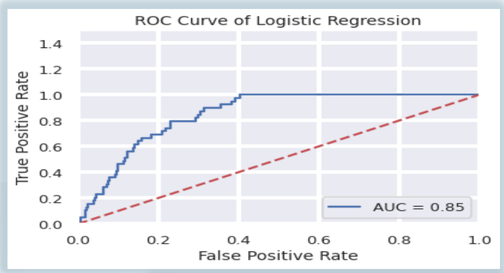
Trn:14526 examples & 32 features

Val:1855 examples & 32 features

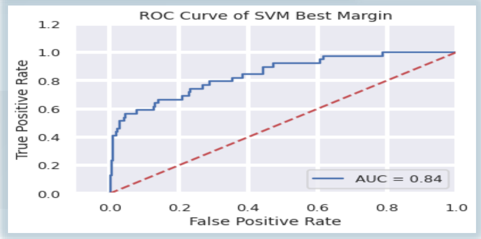
Models

Best AUC of 0.85 achieved with Logistic Regression model

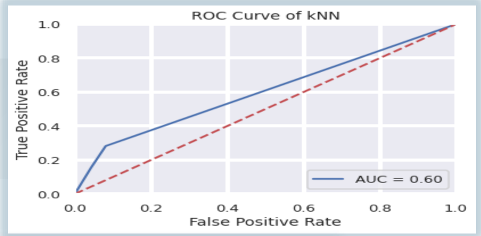
1 Logistic Regression


ROC Curve of Logistic Regression

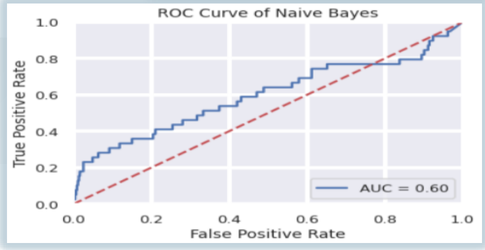
2 SVM


ROC Curve of SVM Best Margin

3 KNN

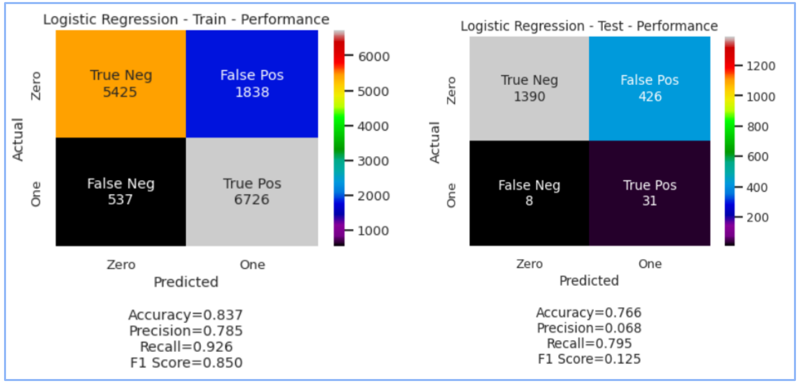

ROC Curve of kNN

4 Naïve Bayes


ROC Curve of Naive Bayes

Results Best Accuracy and F1 Score achieved with SVM and best Recall(0.795) achieved with Logistic Regression

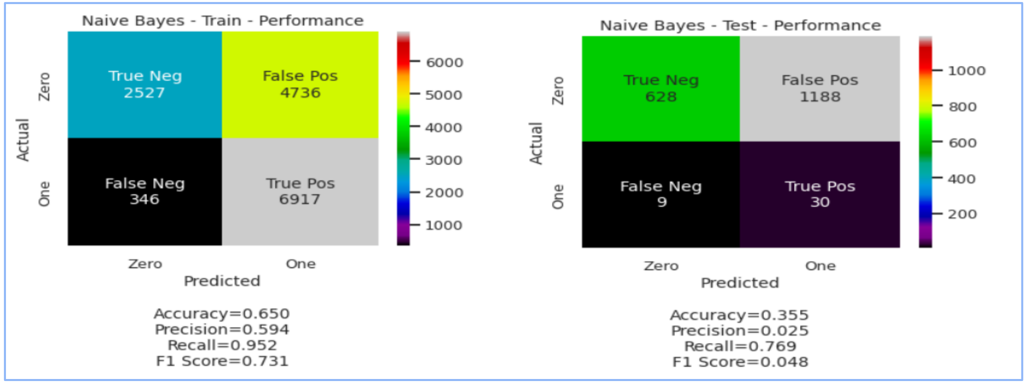
Logistic Regression



Accuracy=0.837
Precision=0.785
Recall=0.926
F1 Score=0.850

Accuracy=0.766
Precision=0.068
Recall=0.795
F1 Score=0.125

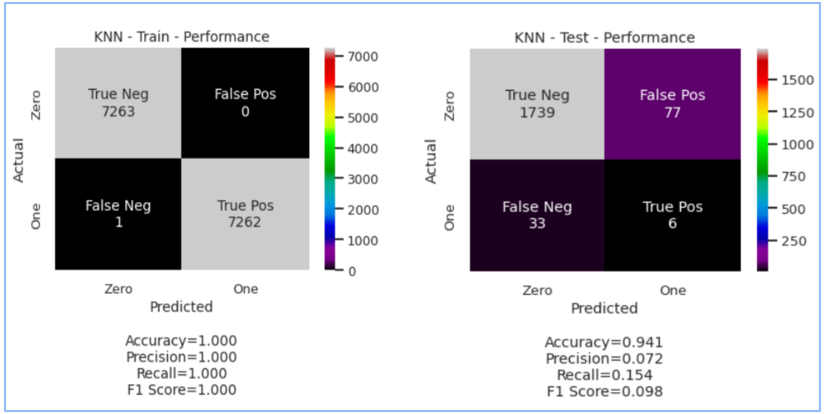
Naïve Bayes



Accuracy=0.650
Precision=0.594
Recall=0.952
F1 Score=0.731

Accuracy=0.355
Precision=0.025
Recall=0.769
F1 Score=0.048

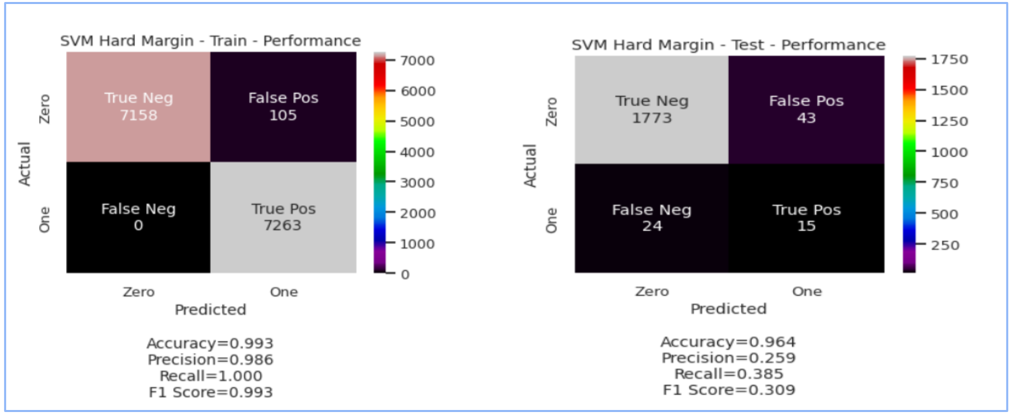
KNN (with K=2)



Accuracy=1.000
Precision=1.000
Recall=1.000
F1 Score=1.000

Accuracy=0.941
Precision=0.072
Recall=0.154
F1 Score=0.098

SVM (with C=50, Gaussian Kernel)



Accuracy=0.993
Precision=0.986
Recall=1.000
F1 Score=0.993

Accuracy=0.964
Precision=0.259
Recall=0.385
F1 Score=0.309

Impact of PCA

Tried on Naïve Bayes & Logistic Regression

Logistic Regression

Without PCA

Training Accuracy : 0.83
Test Accuracy : 0.76

With PCA

Training Accuracy : 0.82
Test Accuracy : 0.76

No improvement with PCA

Naïve Bayes

Without PCA

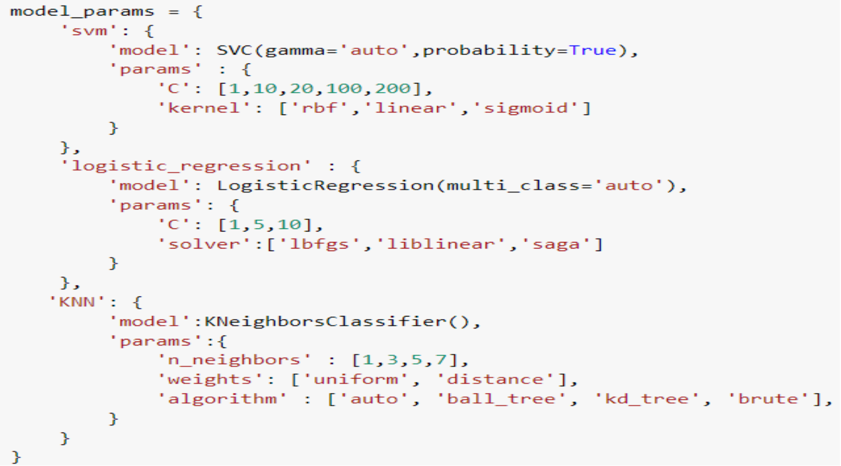
Training Accuracy : 0.65
Test Accuracy : 0.35

With PCA

Training Accuracy : 0.58
Test Accuracy : 0.25

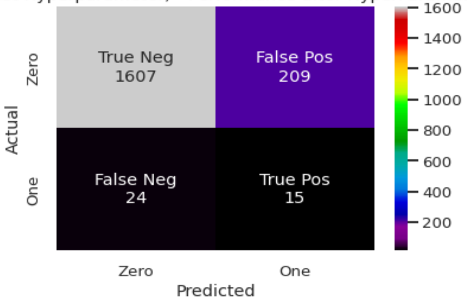
Hyperparameter Tuning – Confusion Matrix

Grid Search Cross Validation with Parameters below:



```
model_params = {
    'svm': {
        'model': SVC(gamma='auto',probability=True),
        'params' : {
            'C': [1,10,20,100,200],
            'kernel': ['rbf','linear','sigmoid']
        }
    },
    'logistic_regression' : {
        'model': LogisticRegression(multi_class='auto'),
        'params': {
            'C': [1,5,10],
            'solver':['lbfgs','liblinear','saga']
        }
    },
   'KNN': {
        'model':KNeighborsClassifier(),
        'params':{
            'n_neighbors' : [1,3,5,7],
            'weights': ['uniform', 'distance'],
            'algorithm' : ['auto', 'ball_tree', 'kd_tree', 'brute'],
        }
    }
}
```

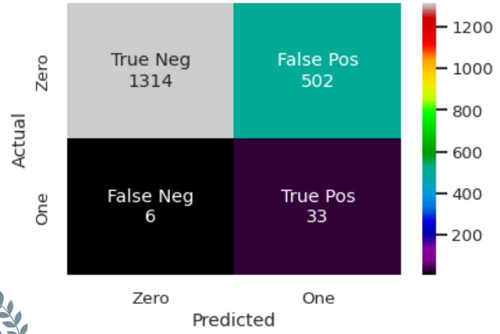
KNN - Test (post Hyperparameter) - Performance after HyperParameter Tuning



KNN

Accuracy=0.874
Precision=0.067
Recall=0.385
F1 Score=0.114

SVM - Test (post Hyperparameter) - Performance after HyperParameter Tuning

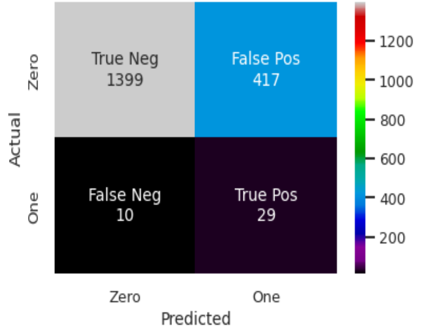


SVM

Accuracy=0.726
Precision=0.062
Recall=0.846
F1 Score=0.115

With C = 10 and kernel 'rbf'

Logistic Regression - Test (post Hyperparameter) - Performance after HyperParameter Tuning



Logistic Regression

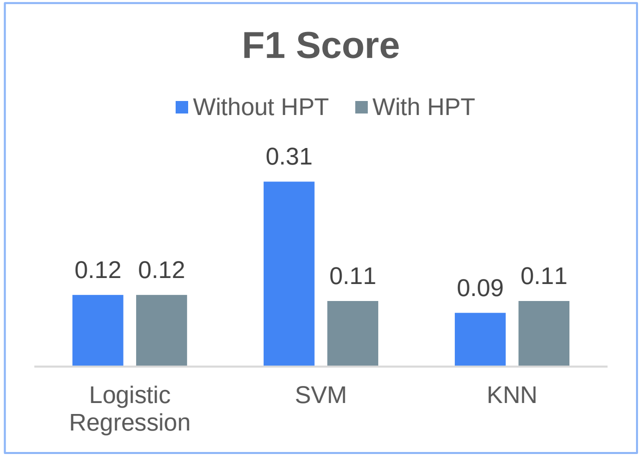
Accuracy=0.770
Precision=0.065
Recall=0.744
F1 Score=0.120

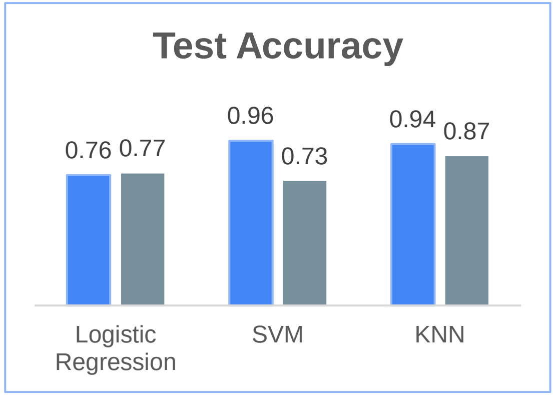
Impact of Hyperparameter Tuning (Performance evaluation metrics)

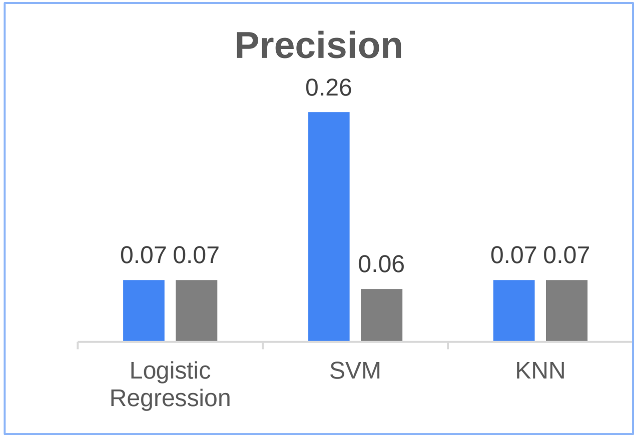
- **Recall = TP/(TP+FN)**
- Recall is the key performance evaluation metric in our case. A good recall value minimize the number of False Negatives case (i.e. a firm has to be predicted bankrupt but is not predicted as bankrupt)

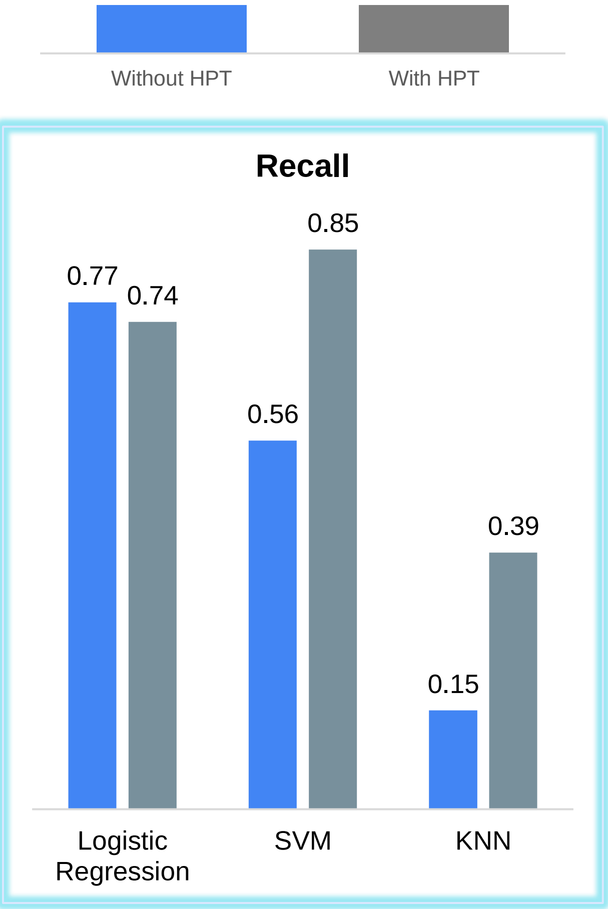
- In this case, it is costlier if a system ignores the bankrupt case

- **SVM is the best model in terms of Recall value**



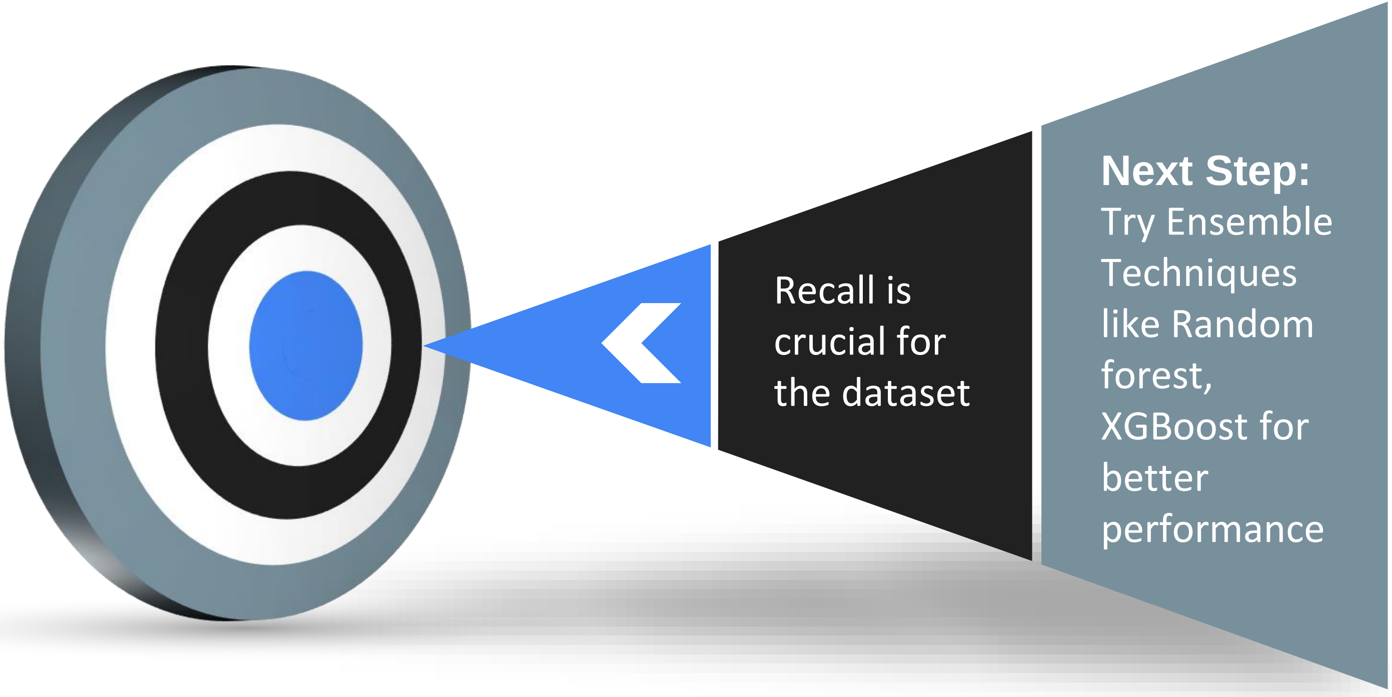






*HPT – Hyperparameter Tuning

Key Takeaway..



Recall is crucial for the dataset

Next Step: Try Ensemble Techniques like Random forest, XGBoost for better performance