Kaiburr Assessment

Task 6

Implement a sample machine learning program for a problem statement of your choice.

Chosen Problem Statement:

Credit Fraud Detection System Using ML and DL Models

Name: Venkata Raghu Ram Raavi

Reg No: 19BCE2561

University: Vellore Institute of Technology

Email Id: venkataraghu.ramraavi2019@vitstudent.ac.in

Problem Statement:

We have much research that are going in this field and we many more existing systems in the same idea with good models and with high accuracies. In this System I have included some economic parameters like Margins, Chargeback, Lost (False Positives), True Positives such as no lost customers. Considering these parameters in the Net Gain for creating a new formula with the existing features in the dataset and these economical parameters and to choose the model with all these factors to select a model which is economically feasible for both the credit card company and other companies who are using the algorithms.

Dataset Link

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

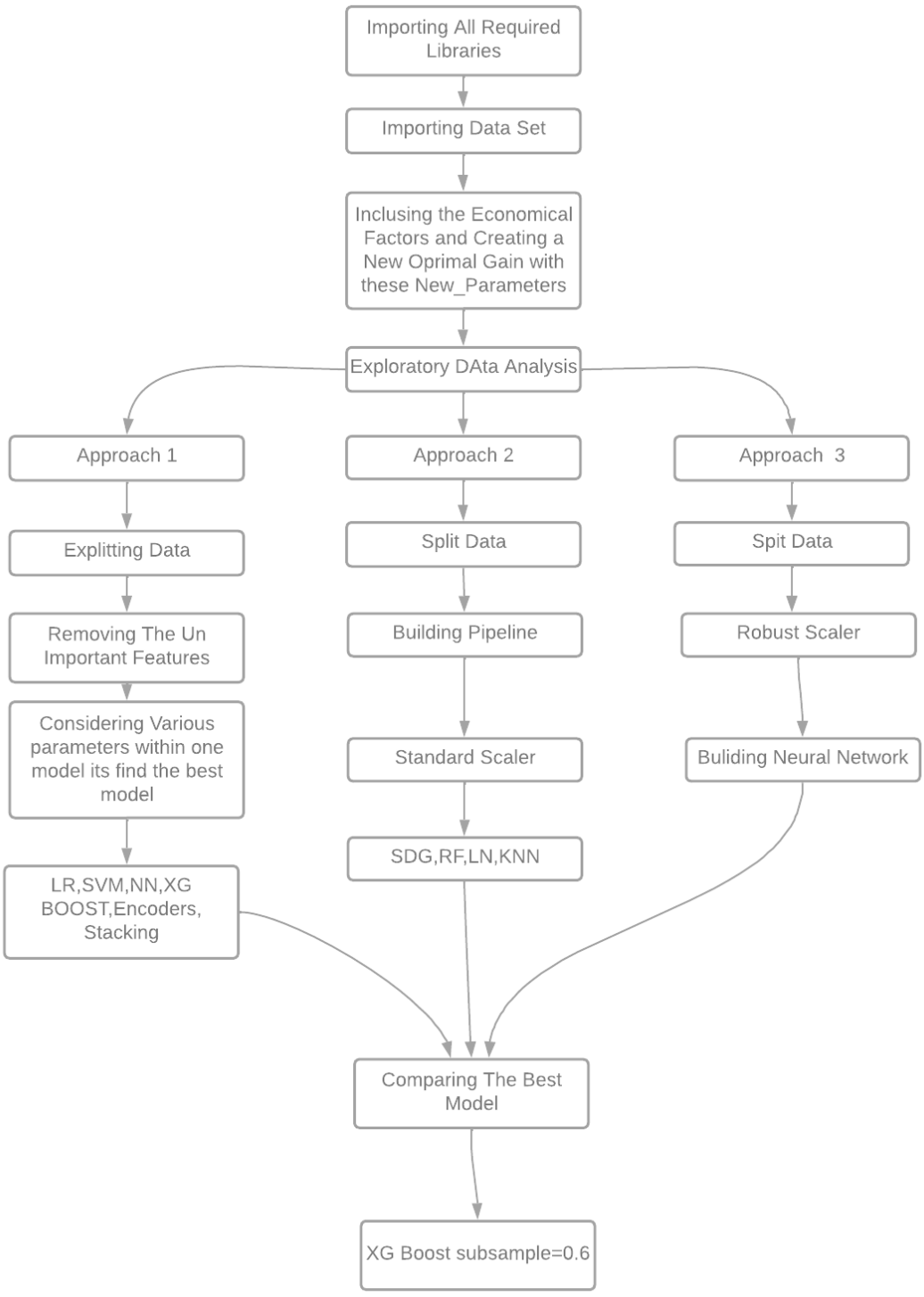
About Dataset

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

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

transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise.

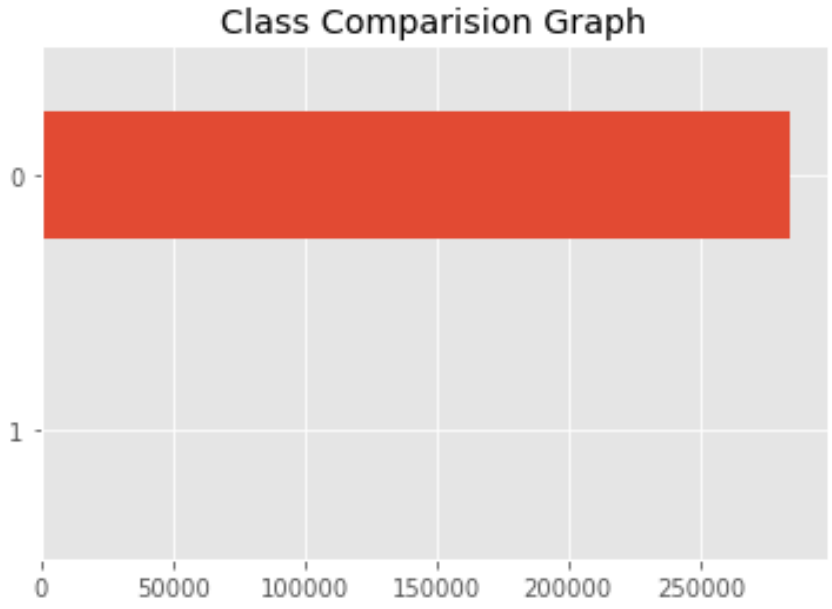
Process Flow:



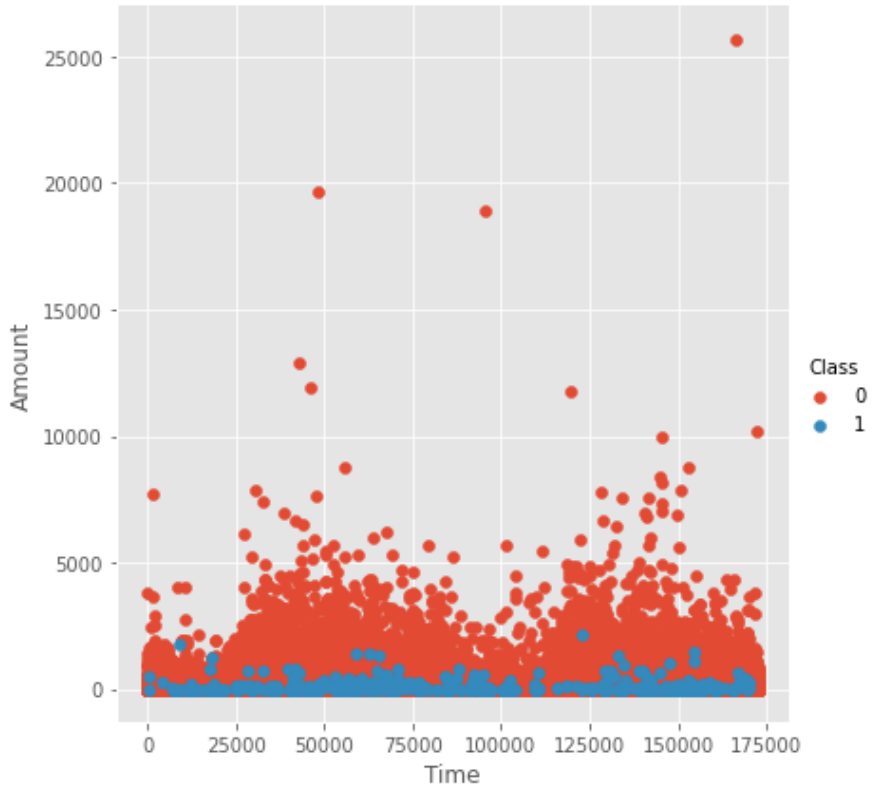
EDA

Histogram Plots for the Data set:

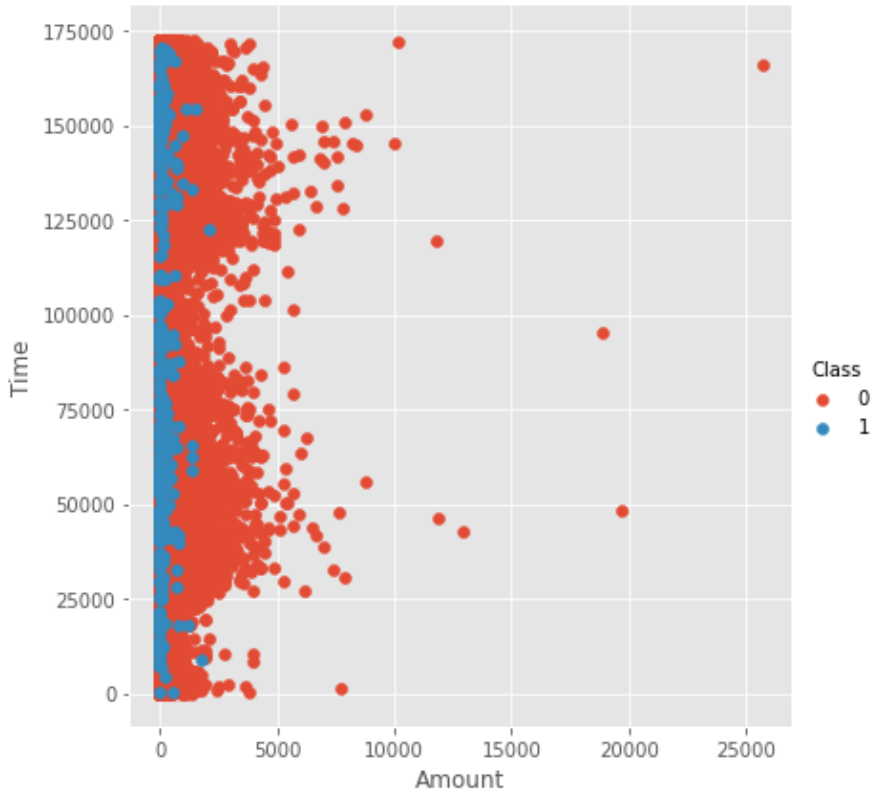
Class Comparison Graph



Pair plot Graph for Amount Vs Time on Class

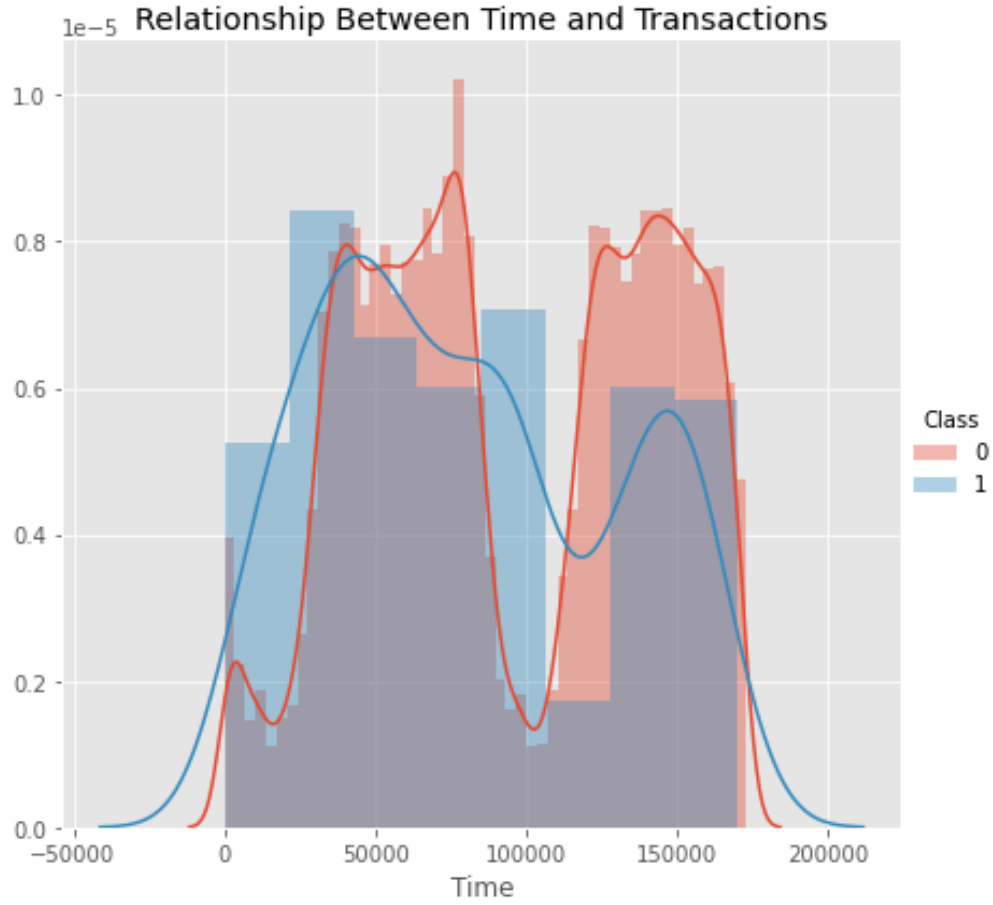


Pair plot Graph for Time Vs Amount on Class

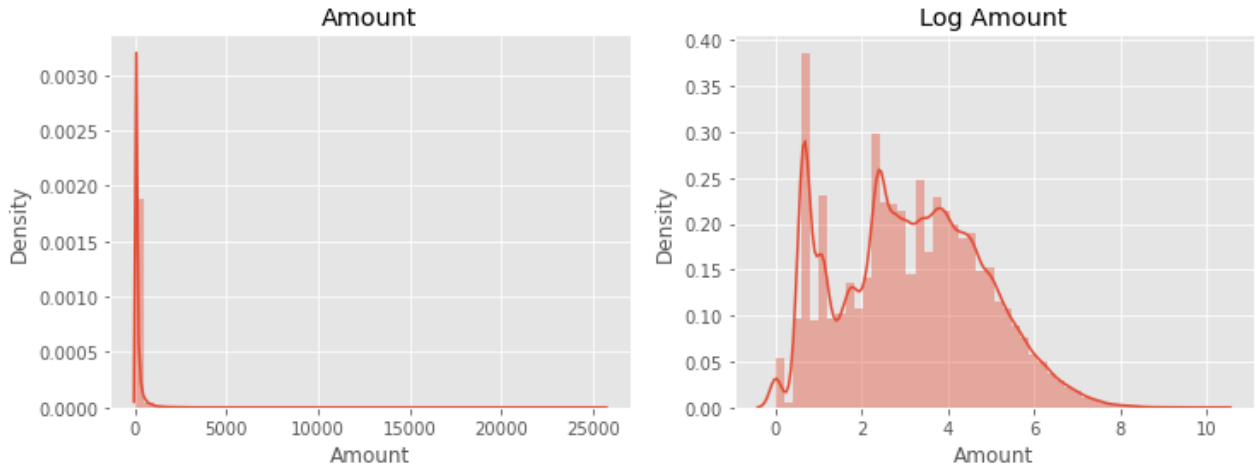


Pair Plots Between the Features Time, Amount, Class

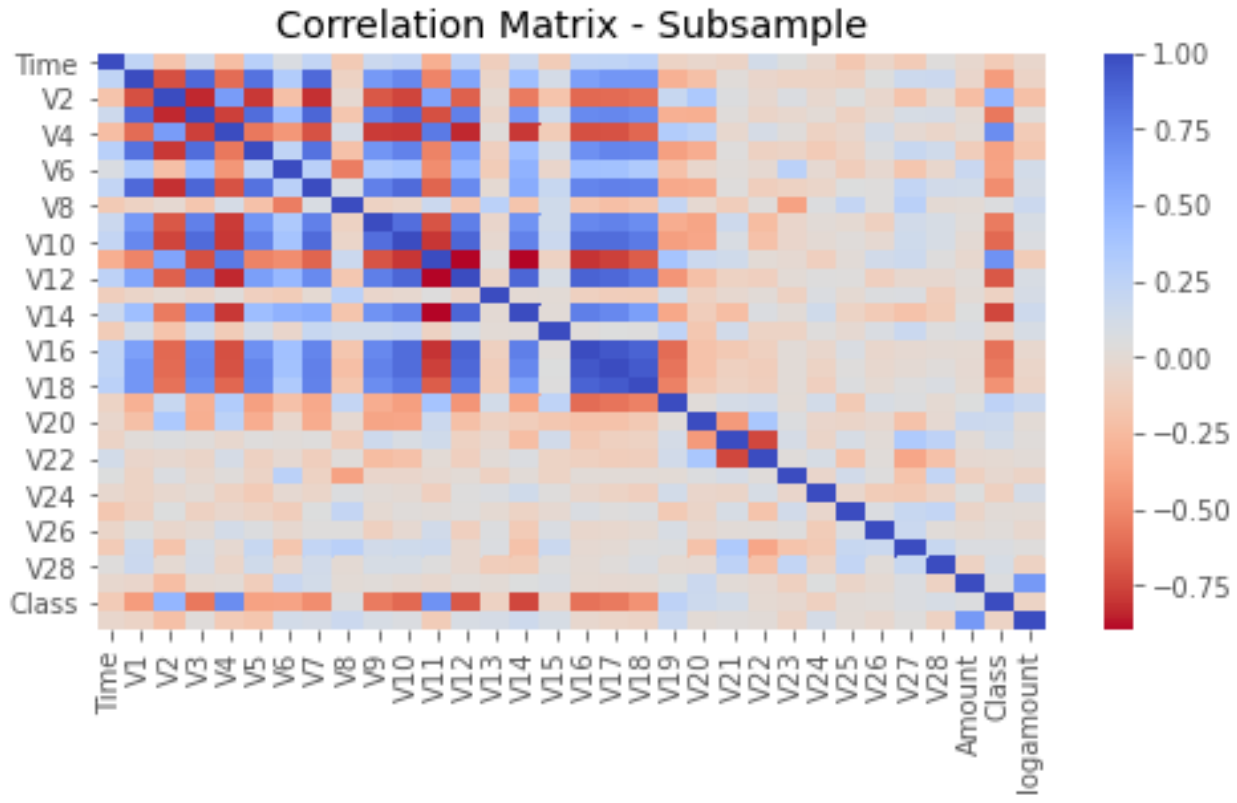
Relationship Between Time and Transactions

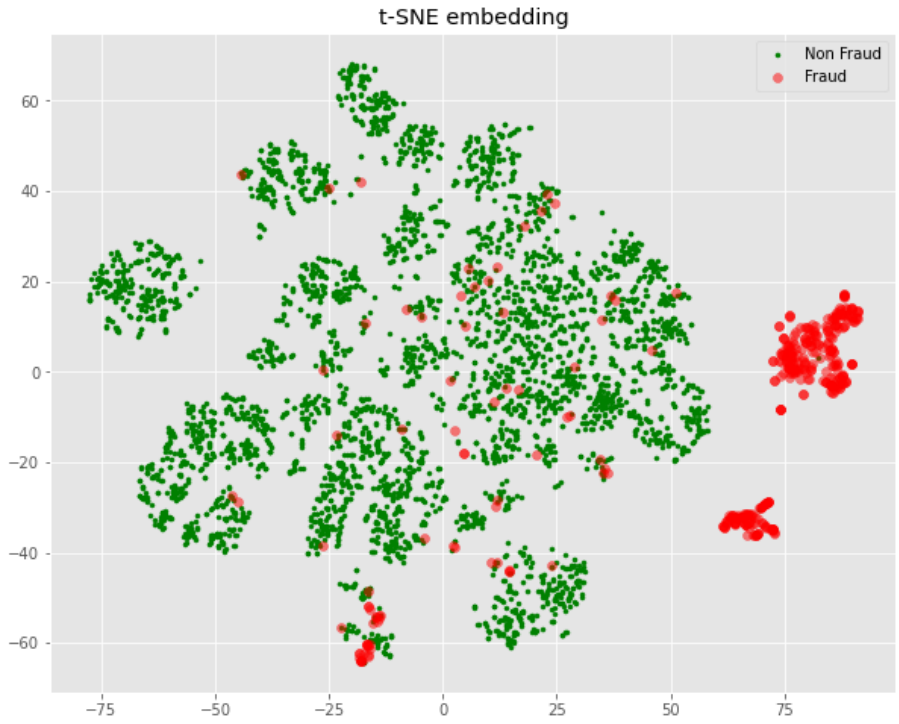


Comparison between the Amount and Log Amount:



Correlation Matrix for the Dataset:

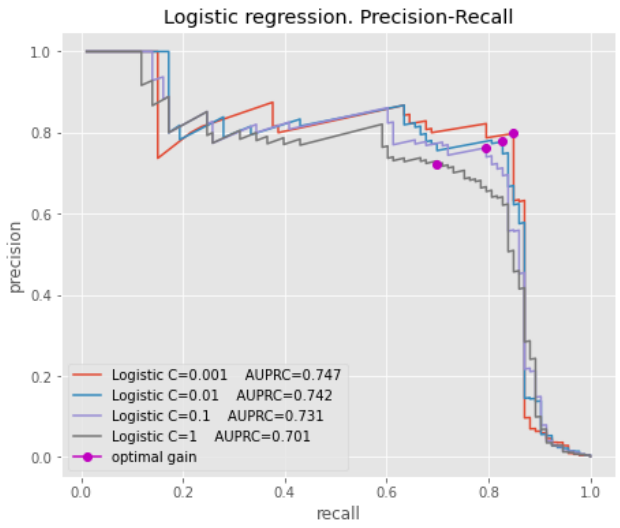


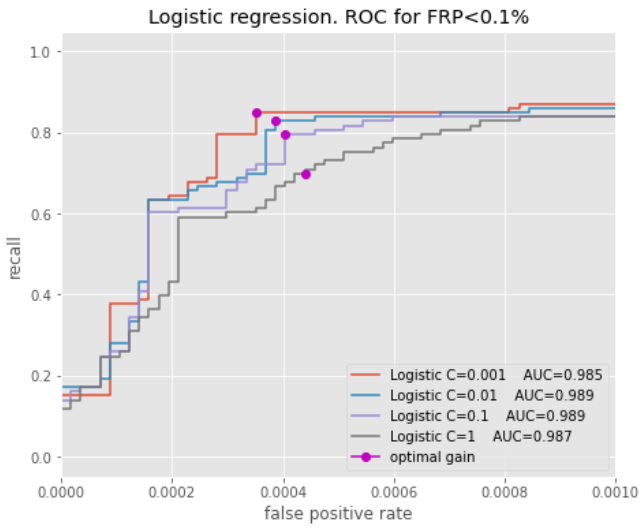


Models Performance

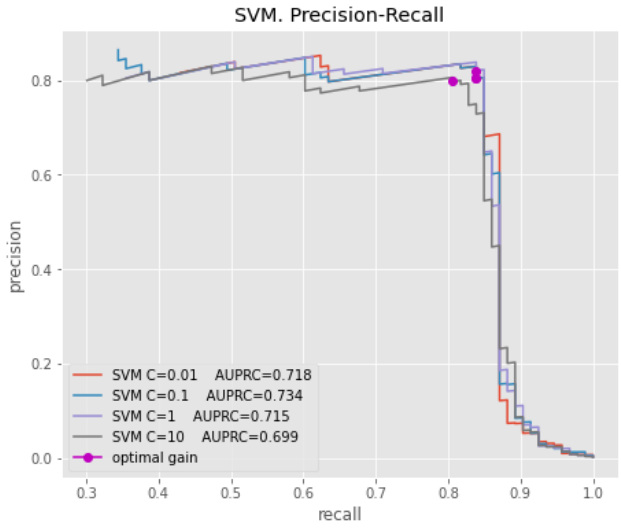
Approach – 1 Models Performance

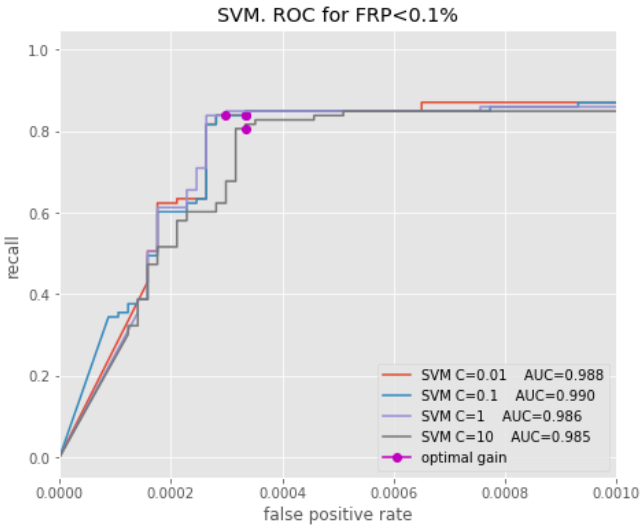
Linear Regression



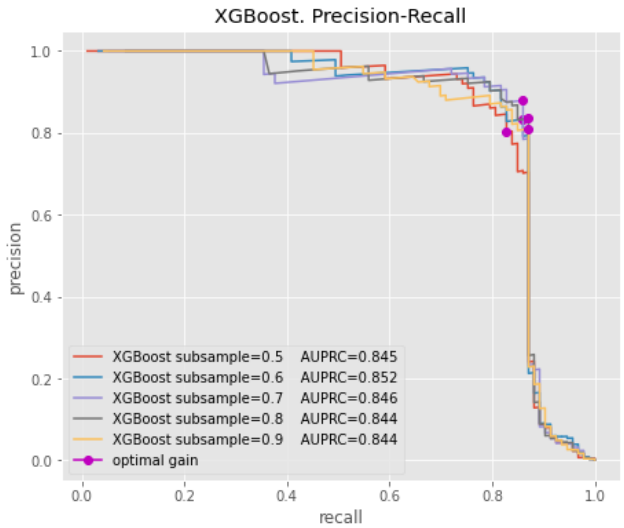


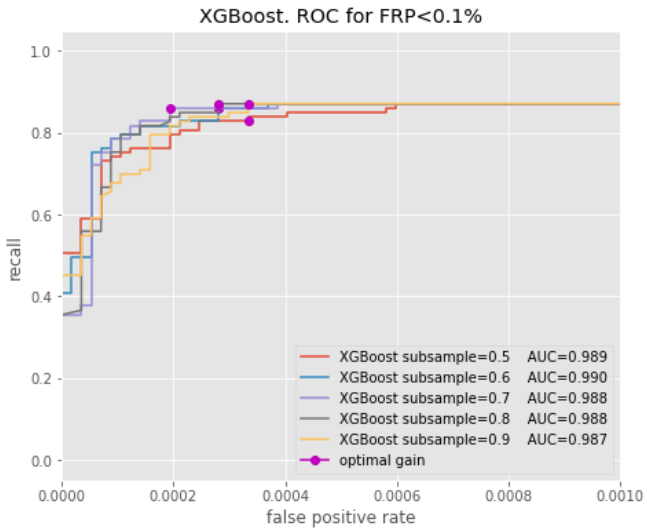
SVM Model Performance

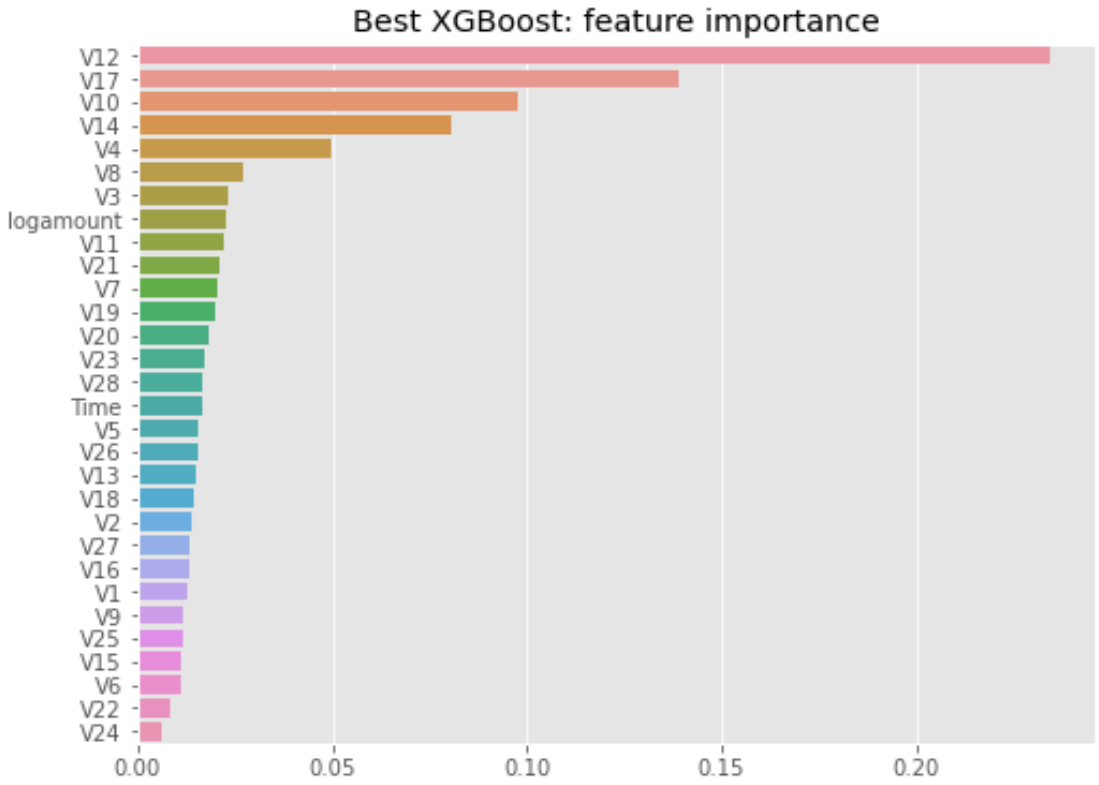




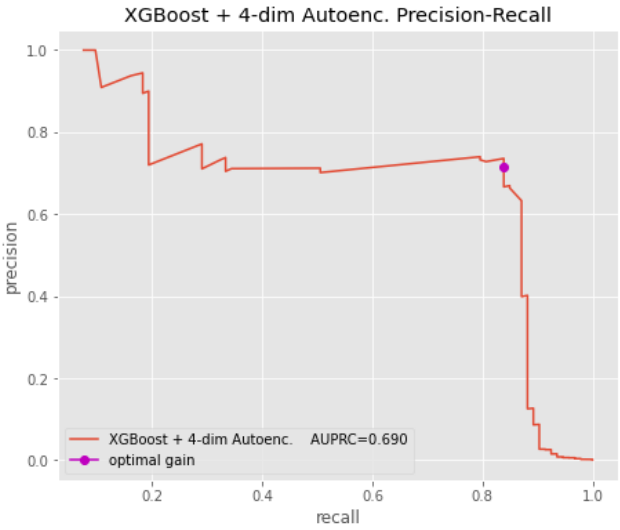
XGBoost Model Performance

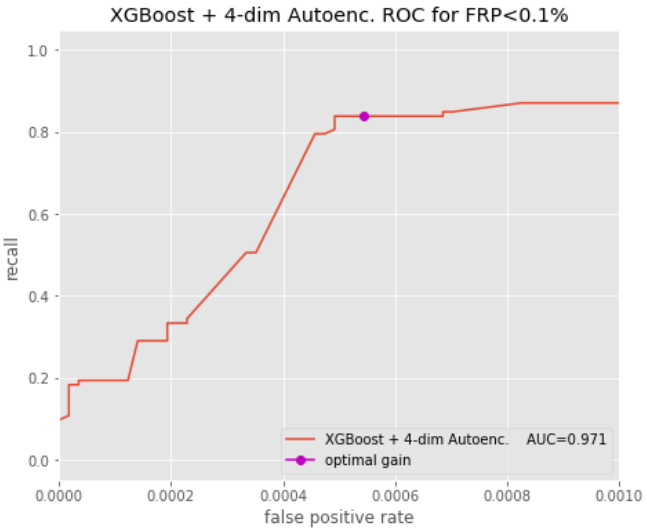






XGBoost + 4-dim Auto Encoder Model Performance

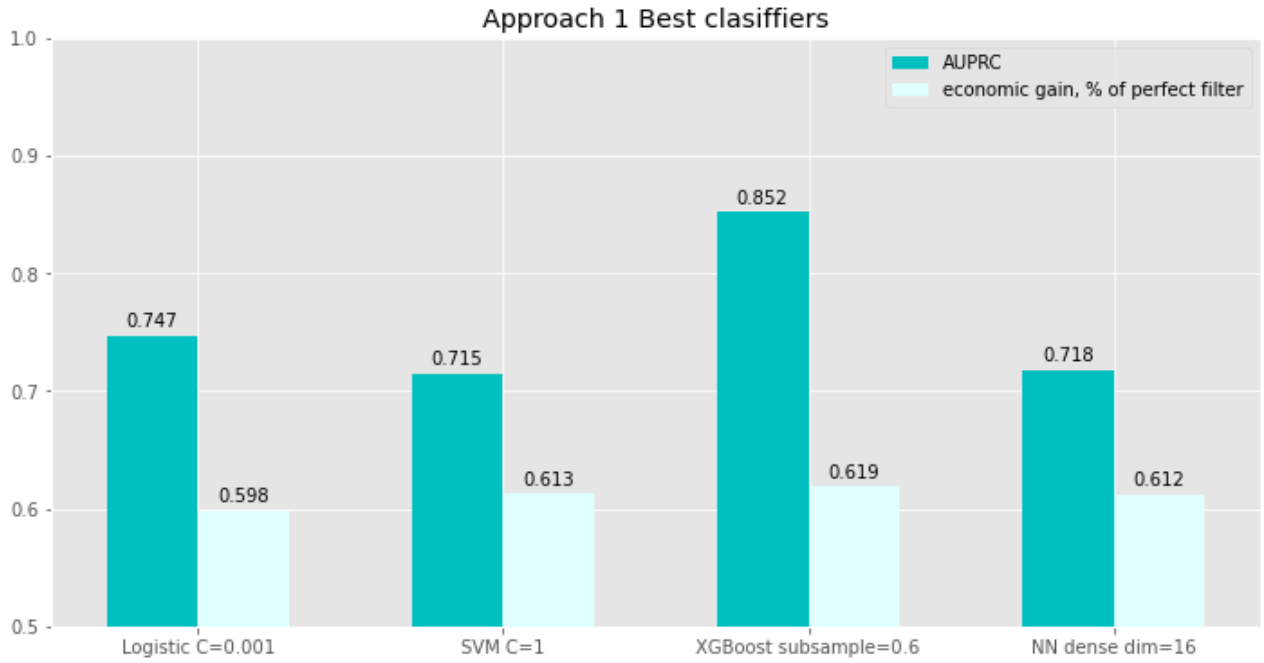




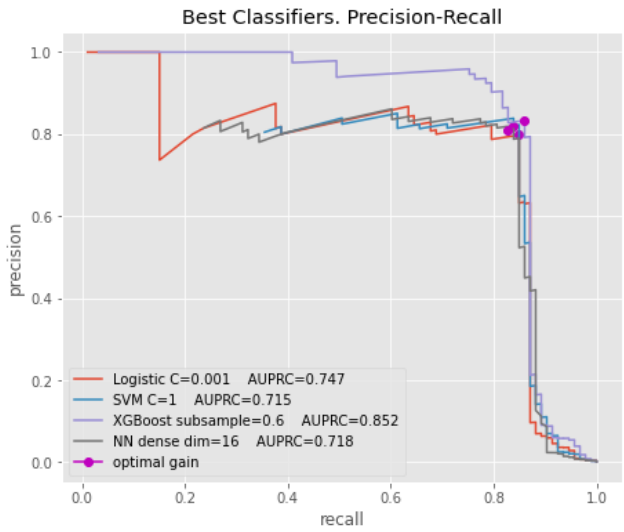
Approach 1 -- Comparision of Model Performance

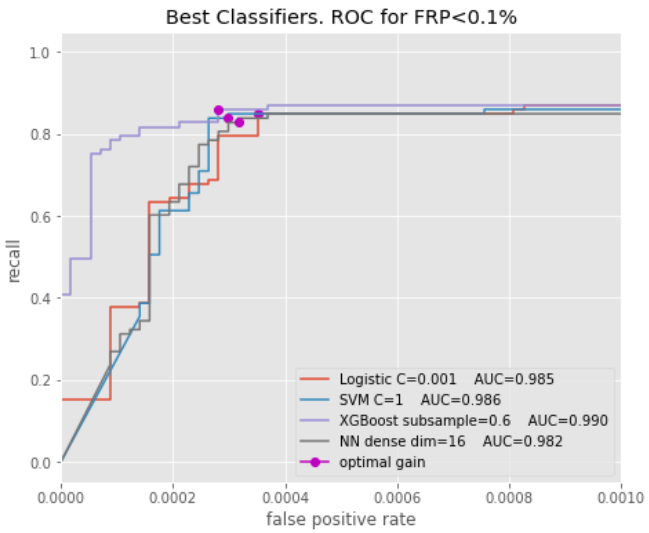
	classifier	TN	FP	FN	TP	AUC	AUPRC	max_gain	precision	recall	fpr
0	Logistic C=0.001	56849	20	15	78	0.985	0.747	0.598	0.798	0.849	0.000352
1	Logistic C=0.01	56847	22	17	76	0.989	0.742	0.567	0.778	0.828	0.000387
2	Logistic C=0.1	56846	23	20	73	0.989	0.731	0.538	0.763	0.796	0.000404
3	Logistic C=1	56845	24	28	65	0.987	0.701	0.441	0.722	0.699	0.000440
4	SVM C=0.01	56851	18	15	78	0.988	0.718	0.612	0.804	0.839	0.000334
5	SVM C=0.1	56851	18	15	78	0.990	0.734	0.612	0.804	0.839	0.000334
6	SVM C=1	56853	16	15	78	0.986	0.715	0.613	0.821	0.839	0.000299
7	SVM C=10	56851	18	18	75	0.985	0.699	0.567	0.798	0.806	0.000334
8	XGBoost subsample=0.5	56851	18	16	77	0.989	0.845	0.613	0.802	0.828	0.000334
9	XGBoost subsample=0.6	56853	16	14	79	0.990	0.852	0.619	0.833	0.860	0.000281
10	XGBoost subsample=0.7	56858	11	14	79	0.988	0.846	0.602	0.879	0.860	0.000193
11	XGBoost subsample=0.8	56853	16	13	80	0.988	0.844	0.619	0.835	0.871	0.000281
12	XGBoost subsample=0.9	56850	19	13	80	0.987	0.844	0.619	0.810	0.871	0.000334
13	NN dense dim=16	56852	17	16	77	0.982	0.718	0.612	0.811	0.828	0.000317
14	NN dense dim=32	56847	22	15	78	0.981	0.716	0.592	0.772	0.839	0.000404
15	NN dense dim=64	56856	13	21	72	0.984	0.722	0.564	0.837	0.774	0.000246
16	NN dense dim=128	56853	16	18	75	0.985	0.724	0.587	0.826	0.817	0.000281
17	XGBoost + 4-dim Autoenc	56839	30	15	78	0.971	0.690	0.580	0.716	0.839	0.000545
0	Stacking	56837	32	14	79	0.977	0.732	0.609	0.705	0.849	0.000580

	classifier	TN	FP	FN	TP	AUC	AUPRC	max_gain	precision	recall	fpr
0	Logistic C=0.001	56849	20	15	78	0.985	0.747	0.598	0.798	0.849	0.000352
6	SVM C=1	56853	16	15	78	0.986	0.715	0.613	0.821	0.839	0.000299
9	XGBoost subsample=0.6	56853	16	14	79	0.990	0.852	0.619	0.833	0.860	0.000281
13	NN dense dim=16	56852	17	16	77	0.982	0.718	0.612	0.811	0.828	0.000317



Approach 1 Best Classifiers Precision – Recall and Roc Comparisons





Approach – 2 Models Performance

SGD Model Performance

```
# Evaluation of SGD
evaluation(y_test, grid_sgd, X_test)
```

```
CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0       1.00      0.99      1.00     85295
           1       0.14      0.91      0.25       148

    accuracy                           0.99     85443
   macro avg       0.57      0.95      0.62     85443
weighted avg       1.00      0.99      0.99     85443

AUC-ROC
0.9479720619851928
F1-Score
0.2460973370064279
Accuracy
0.990391254988706
```

RF Model Performance

```
# Evaluation of Grid Random Forest
evaluation(y_test, grid_rf, X_test)
```

```
CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85295
           1       1.00      0.23      0.37       148

    accuracy                           1.00     85443
   macro avg       1.00      0.61      0.69     85443
weighted avg       1.00      1.00      1.00     85443

AUC-ROC
0.6148648648648649
F1-Score
0.37362637362637363
Accuracy
0.9986657771847899
```

LR Model Performance

```
# Evaluation of Grid Linear Regression
evaluation(y_test, grid_lr, X_test)
```

```
CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85295
           1       0.58      0.61      0.59       148

    accuracy                           1.00     85443
   macro avg       0.79      0.80      0.80     85443
weighted avg       1.00      1.00      1.00     85443

AUC-ROC
0.8036730235129906
F1-Score
0.594059405940594
Accuracy
0.9985604438046417
```

KNN Model Performance

```
#Evaluation of Grid KNN(K-Nearest Neighbour)
evaluation(y_test, grid_knn, X_test)
```

```
CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85295
           1       0.22      0.09      0.13       148

    accuracy                           1.00     85443
   macro avg       0.61      0.55      0.57     85443
weighted avg       1.00      1.00      1.00     85443

AUC-ROC
0.5469983348727706
F1-Score
0.13145539906103285
Accuracy
0.9978348138525098
```

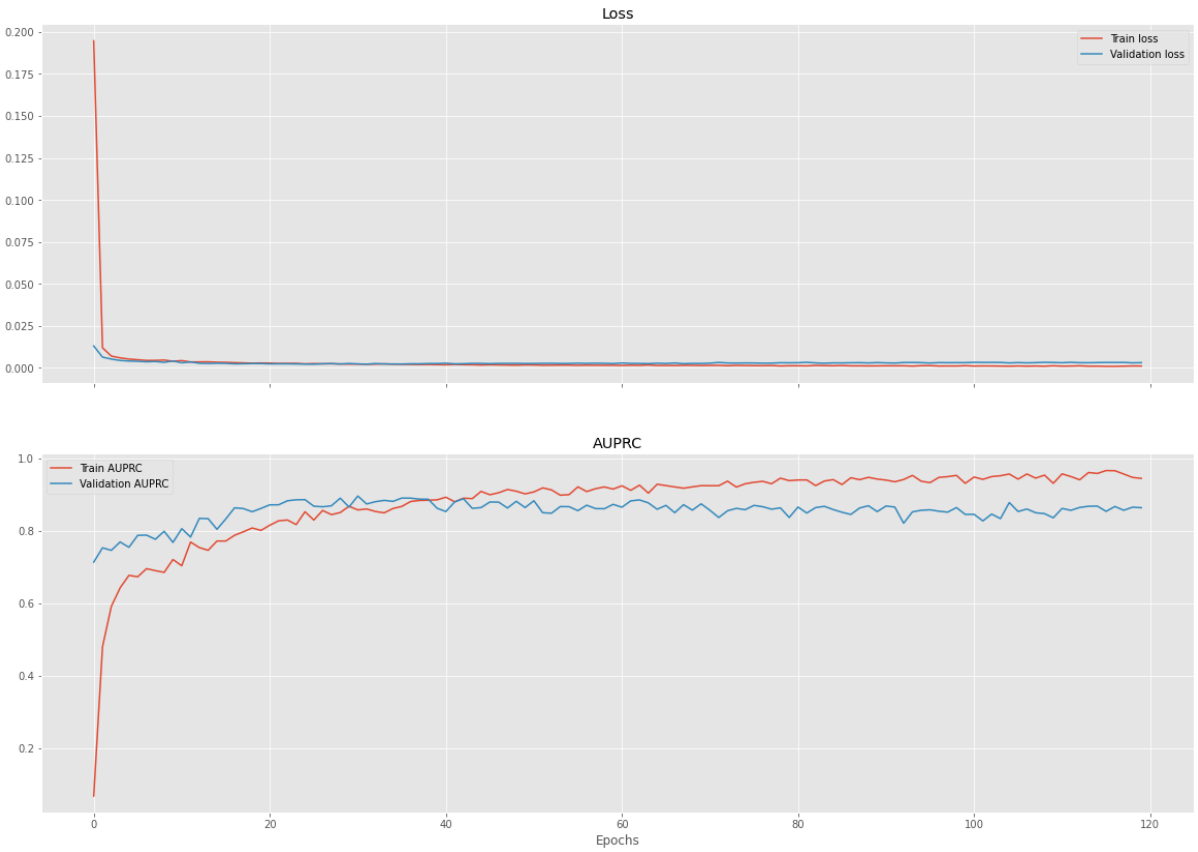
ANN – Four Layers (Approach 3)

```
y_pred = model.predict(test_dataset)
auprc = tf.keras.metrics.AUC(curve='PR')
auprc.update_state(test_labels, y_pred)
TF_Model_AUPRC = auprc.result().numpy()
TF_Model_AUPRC
```

```
0.7890331
```

```
TF_Model_metrics = compute_metrics(test_labels, y_pred)
print_metrics(TF_Model_metrics)
```

```
Fraudulent Transactions Detected (True Positives): 77
Fraudulent Transactions Missed (False Negatives): 21
Legitimate Transactions Incorrectly Detected (False Positives): 6
Legitimate Transactions Detected (True Negatives): 56858
```



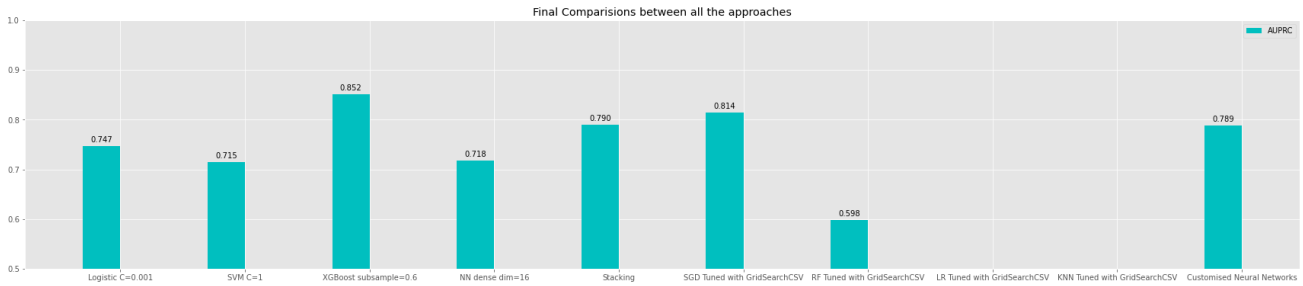
Comparision of Model Performance in Three Approaches

	classifier_name	AUC	AUPRC
0	Logistic C=0.001	0.985	0.747000
1	Logistic C=0.01	0.989	0.742000
2	Logistic C=0.1	0.989	0.731000
3	Logistic C=1	0.987	0.701000
4	SVM C=0.01	0.988	0.718000
5	SVM C=0.1	0.990	0.734000
6	SVM C=1	0.986	0.715000
7	SVM C=10	0.985	0.699000
8	XGBoost subsample=0.5	0.989	0.845000
9	XGBoost subsample=0.6	0.990	0.852000
10	XGBoost subsample=0.7	0.988	0.846000
11	XGBoost subsample=0.8	0.988	0.844000
12	XGBoost subsample=0.9	0.987	0.844000
13	NN dense dim=16	0.982	0.718000
14	NN dense dim=32	0.981	0.716000
15	NN dense dim=64	0.984	0.722000
16	NN dense dim=128	0.985	0.724000
17	XGBoost + 4-dim Autoenc	0.971	0.690000
18	Stacking	0.984	0.790000
19	SGD Tuned with GridSearchCSV	0.990	0.814430
20	RF Tuned with GridSearchCSV	0.998	0.597997
21	LR Tuned with GridSearchCSV	0.990	0.491725
22	KNN Tuned with GridSearchCSV	0.998	0.108052

Best Models Over all the Approaches:

	classifier_name	AUC	AUPRC
0	Logistic C=0.001	0.985	0.747000
6	SVM C=1	0.986	0.715000
9	XGBoost subsample=0.6	0.990	0.852000
13	NN dense dim=16	0.982	0.718000
18	Stacking	0.984	0.790000
19	SGD Tuned with GridSearchCSV	0.990	0.814430
20	RF Tuned with GridSearchCSV	0.998	0.597997
21	LR Tuned with GridSearchCSV	0.990	0.491725
22	KNN Tuned with GridSearchCSV	0.998	0.108052
23	Customised Neural Networks	0.988	0.789033

Best Models Comparison Over the three Approaches:



Conclusion:

Based on the Average Precision, Recall Score XG Boost has the highest score hence can be considered as the best model of all the other models over all three approaches