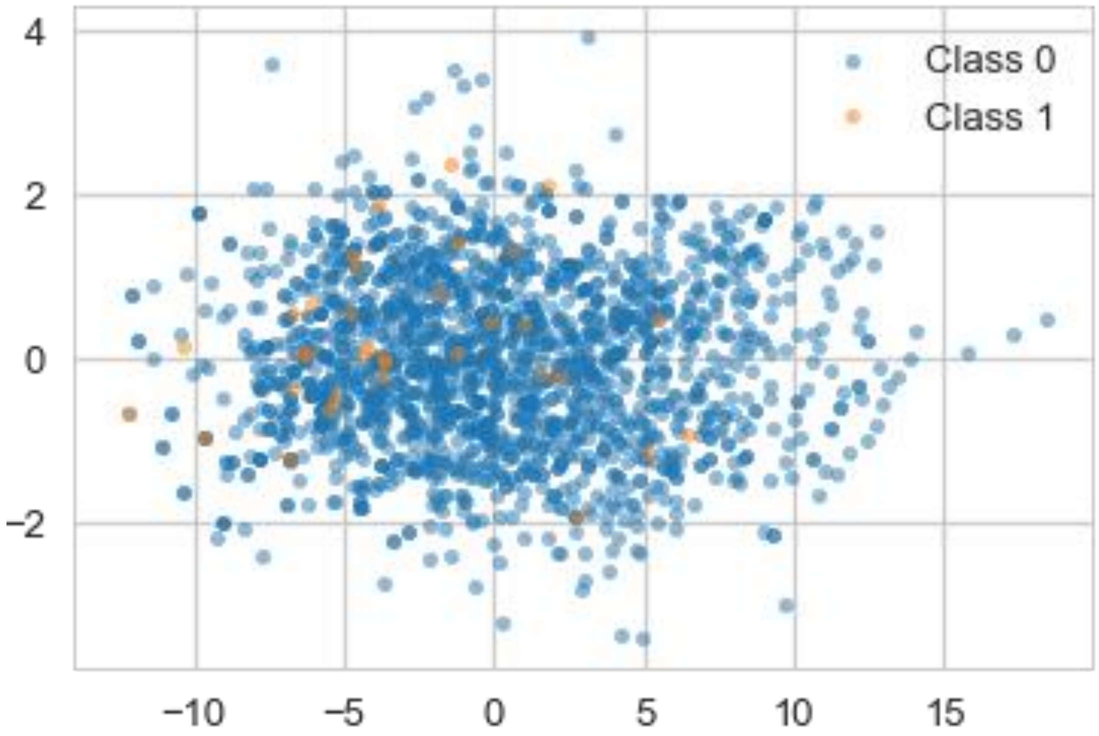
Study of the Class Imbalance Problem

Team : **F-Society**

- Sushant Kumar
- Raman Ranjan Shukla
- Mohit Manwani

Class Imbalance:

- The training set for one class (**majority**) far surpasses the training set of the other class (**minority**), in which, the minority class is often the more interesting class.



In this Project we..

- Review the **issues** that come with learning from imbalanced class data sets and various problems in class imbalance classification.
- Study the **existing approaches** for handling classification with imbalanced datasets.

Understanding the Problem Statement

- Many real-world classification problems have an imbalanced class distribution, such as rare disease identification, fraud detection, spam detection, churn prediction, electricity theft & pilferage etc.
- **Imbalanced Classification**: A classification predictive modeling problem where the distribution of examples across the classes is not equal.
- These types of problems are also known as **rare event prediction** or **extreme event prediction**.

- Suppose class A is 90% of our data-set and class B is the other 10%, but we are most interested in identifying instances of class B.
- We can reach an accuracy of 90% by simply predicting class A every time, but this provides a **useless** classifier for our intended use case.
- Instead, a properly calibrated method may achieve a **lower accuracy**, but would have a substantially **higher true positive rate** (or recall), which is really the metric we want to optimize for.
- Generally, this problem deals with the **trade-off** between **recall** (percent of truly positive instances that were classified as such) and **precision** (percent of positive classifications that are truly positive).

Challenges Identified

Most standard algorithms are accuracy driven, however, in a class imbalance dataset, accuracy tells very little about the minority class.

Lack of information caused by small sample size.

Dataset Description

- We have chosen a dataset that 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 imbalanced**, 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 (due to confidentiality issues).
- Feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise.

Data Level Approaches (external techniques)

Employ a pre-processing step to rebalance the class distribution

Either under sampling or over sampling to reduce the imbalance in training data

SMOTE (Synthetic Minority Over-sampling Technique)

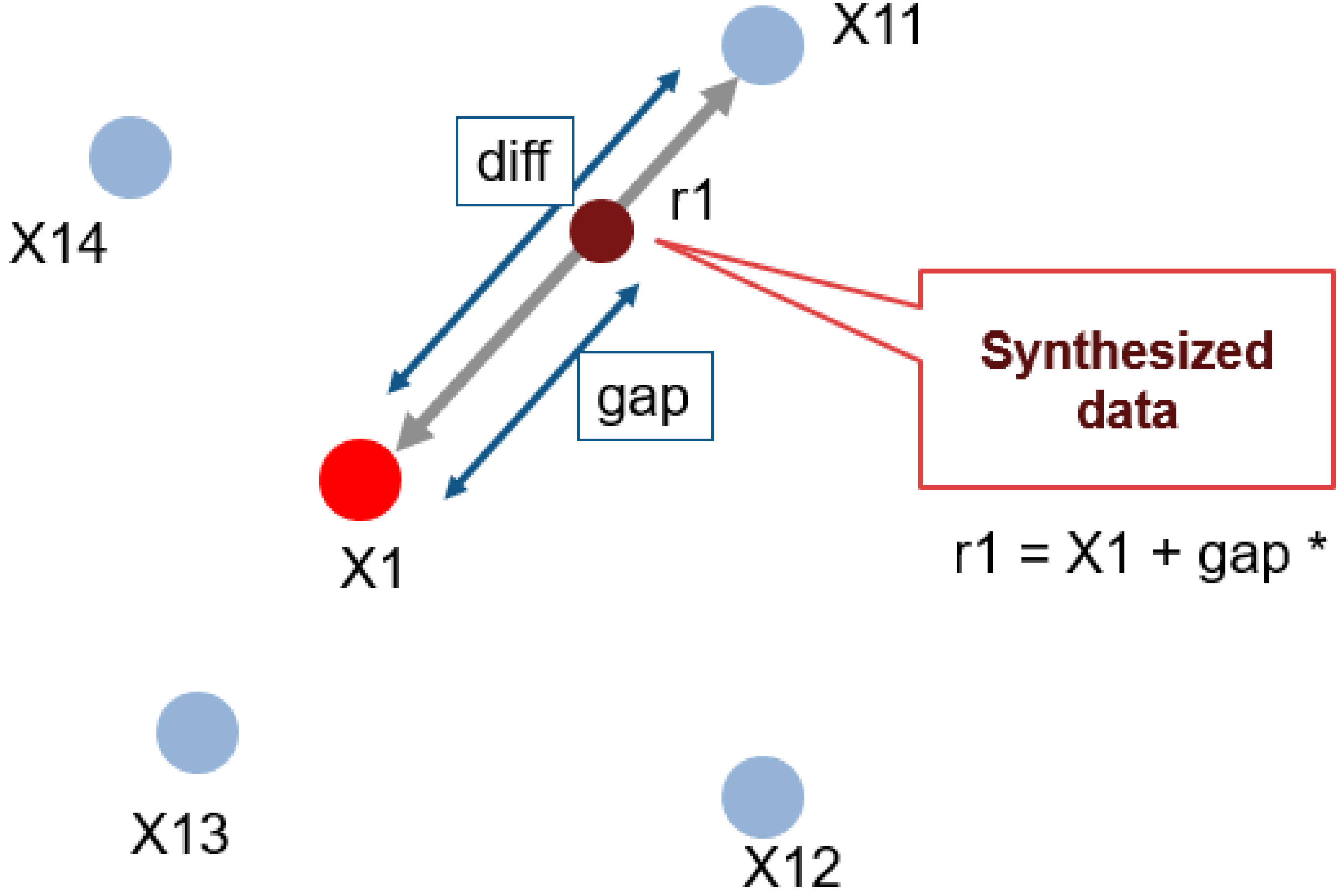
- Over sampling the minority class
- Adding new examples to minority class by computing a probability distribution to model the smaller class
- thus, making the decision boundary larger in order to capture adjacent minority class examples
- It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together

Steps involved in SMOTE

- Select a minority class instance *a* at random
- Find its k nearest minority class neighbors
- Choose one of the k nearest neighbors *b* at random
- Connect *a* and *b* to form a line segment in the feature space
- A synthetic example is created at a randomly selected point between the two examples in feature space

● Minority class

Example case with k = 4



$r1 = X1 + gap * diff$

Results of SMOTE on Logistic Regression

Without SMOTE

Accuracy: ~1.0

Recall of minor class: 0.55

With SMOTE:

Accuracy: 0.98

Recall of minor class: 0.90

Pros:

- The approach is effective because new synthetic examples from the minority class are created that are plausible, that is, are relatively **close** in feature space to existing examples from the minority class.
- Overcomes the overfitting problem posed by random oversampling

Cons:

- A general downside of the approach is that synthetic examples are created without considering the majority class, possibly resulting in ambiguous examples if there is a **strong overlap** for the classes.

RUS vs SMOTe

In general, both sampling methods have their drawbacks, such as the increased model training time for any method that generates additional data, and loss of valuable data for any undersampling technique. Also, when approaching any classification problem with class imbalance, or any machine learning problem altogether, proper data preparation is critical.

For our dataset, SMOTe outperforms RUS.

The Recall of RUS is very high, but other parameters are surprisingly low and therefore we use SMOTe for processing our dataset.

Algorithm Level Approaches

- Cost Sensitive Learning
- Ensemble Method
- One class Learning
- Hybrid Approaches

Cost Sensitive Learning

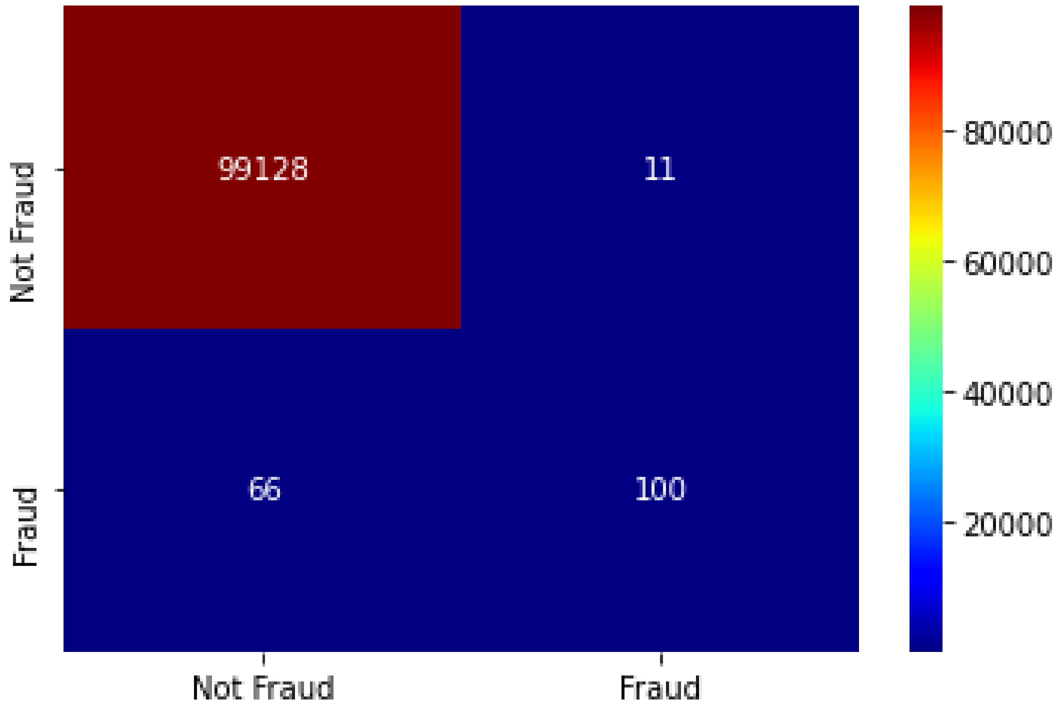
- **Problem Identified** : misclassification cost being regarded as equal for both classes by many traditional learning algorithm.
- **Cost-sensitive learning** approaches are designed with the idea that an expensive cost is imposed on a classifier when a misclassification of the minority class happens.
- For example, a classifier assigns larger cost to false negatives compared to false positives thus emphasizing any correct classification or misclassification regarding the positive class (the minority class).

Cost Sensitive Logistic Regression

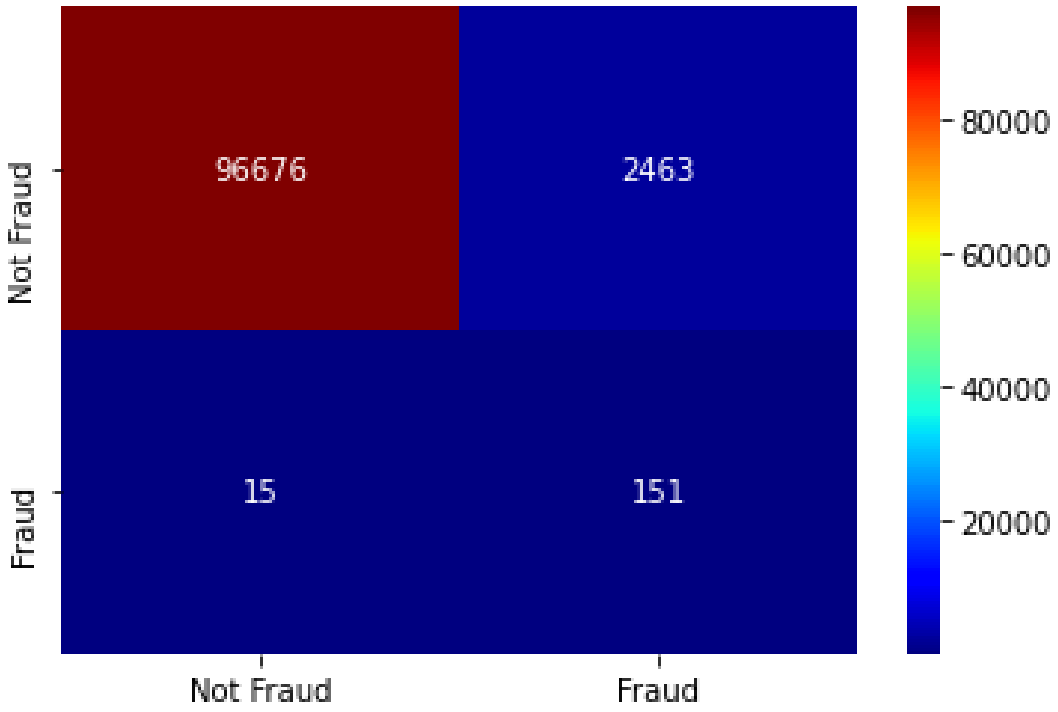
- The minority class was assigned a higher weight than the majority class.
- Keeping the weight of majority cost constant, weight of the minority class was varied, and results were plotted.
- There was a slight drop in accuracy, which is acceptable because the more important metrics read good values.

Confusion Matrix

- Cost sensitive logistic regression model yields better true positive numbers at the cost of increased false positives (I.e, decreased precision).

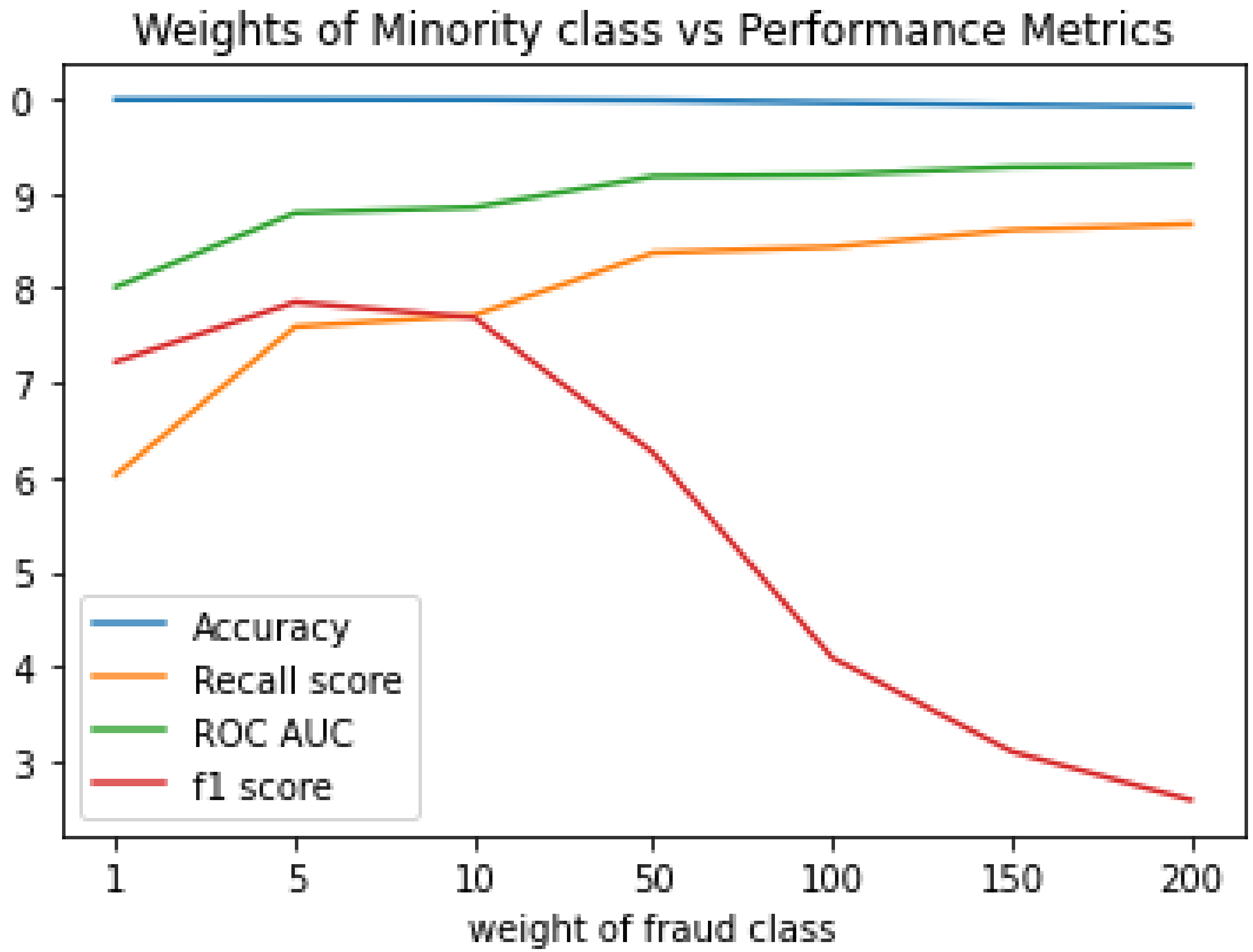


Vanila logictic regression



Weighted logistic regression

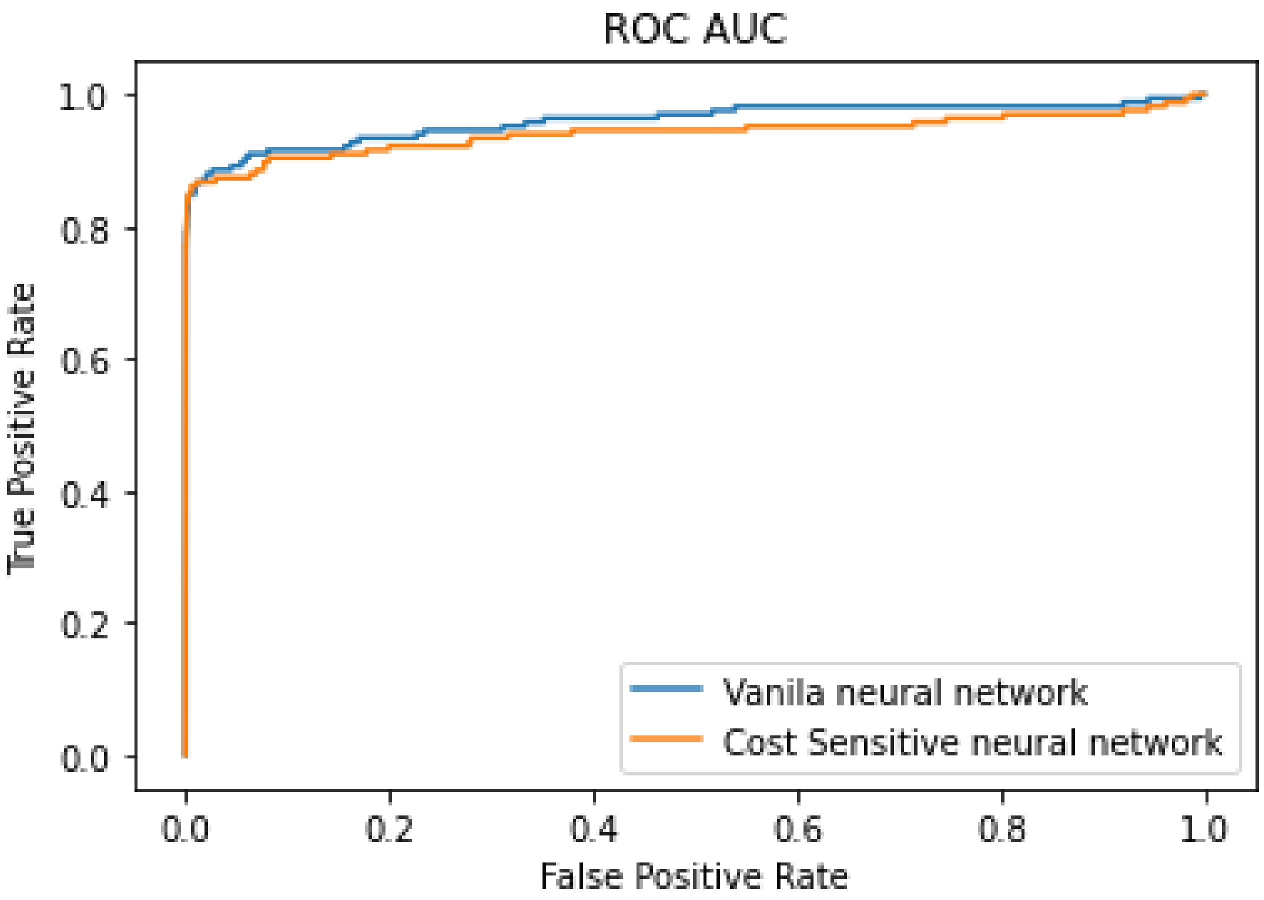
- As the weight assigned to minority class increased, there was an increment in true positives (positive class being the minority), hence there was a good increment in the recall score and ROC AUC score.
- After a certain point, the precision and f1 score start to drop because of the increase in false positives.



Cost Sensitive Neural Network

- The backpropagation algorithm can be updated to **weigh misclassification errors in proportion to the importance of the class**, referred to as weighted neural networks or cost-sensitive neural networks.
- This has the effect of allowing the model to pay more attention to examples from the minority class than the majority class in datasets with a severely skewed class distribution.
- Instead of minimizing the squared error, the backpropagation learning procedure should **minimize the misclassification costs**.

Performance of Cost Sensitive Neural Network



Vanilla Neural Network

- ROC AUC: 0.94

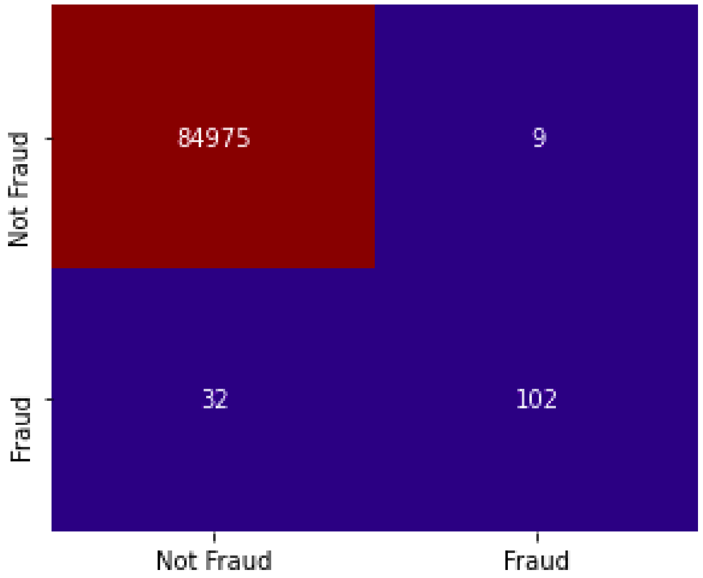
Cost Sensitive Neural Network

- ROC AUC: 0.97

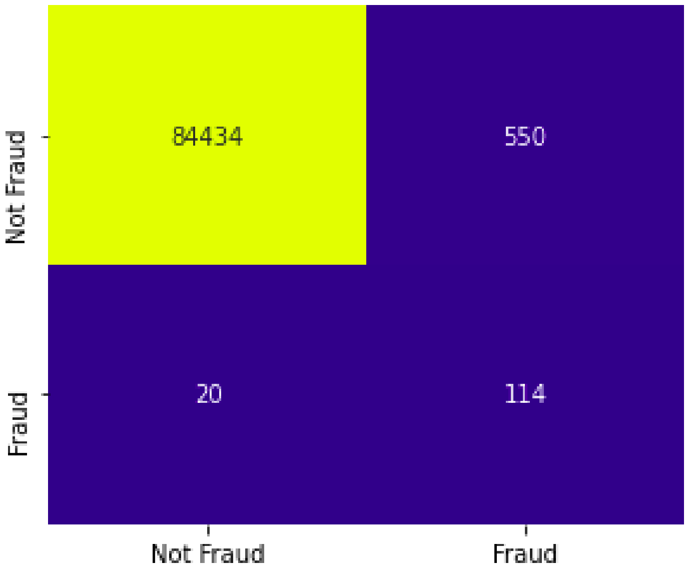
Ensemble Methods

Experimentation Done with-

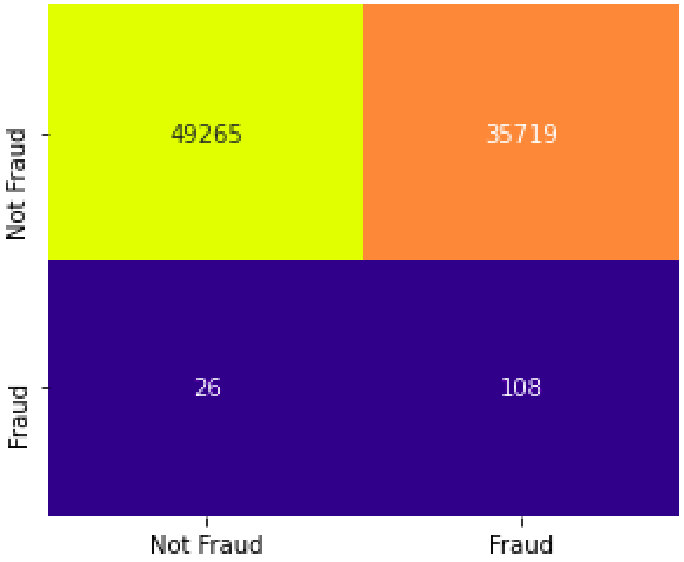
- Vanilla AdaBoost
- SMOTe + AdaBoost
- RUS + AdaBoost
- RUSboost
- SMOTeBoost



Vanilla AdaBoost



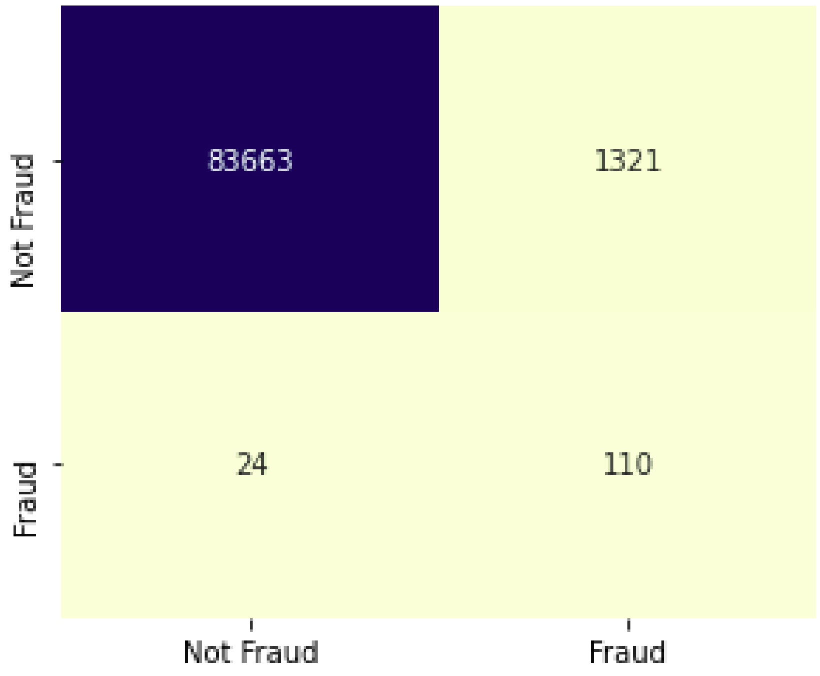
SMOTe + AdaBoost



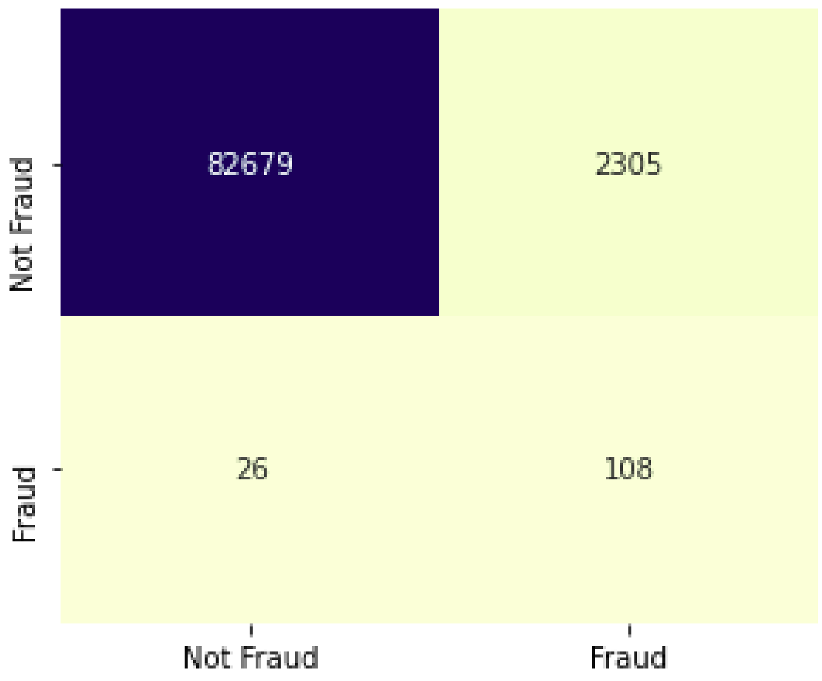
RUS + AdaBoost

Confusion Matrices for different methods

Confusion Matrices for different methods



SMOTeBoost



RUSboost

Analysis

- After examining the confusion matrices,F1-Scores and Recall for different methodologies, we can say that for this dataset SMOTe + AdaBoost gives the best results ,although we have to compromise a bit with the accuracy but still that does not matter much in this case.
- Here complexity grows with the use of more classifiers.
- More number of classifiers does not ensure greater accuracy.
- Diversity concept is difficult to achieve.

One Class Learning

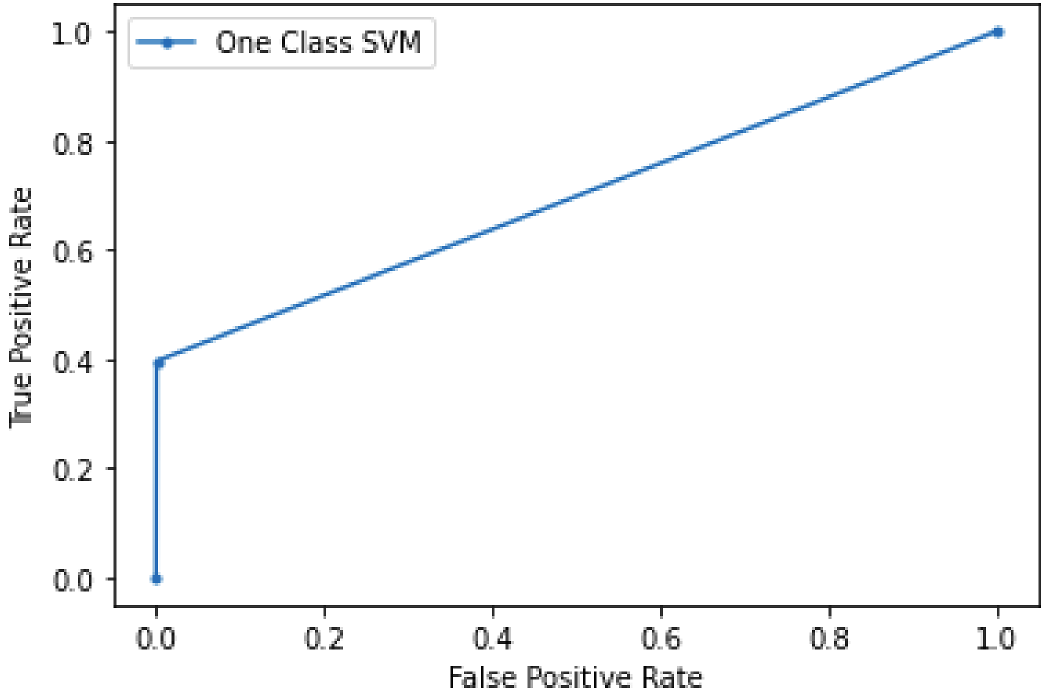
- **Problem Identified** : More samples of one class than the other, makes the classifier more biased towards the majority class.
- **One-Class learning** approaches are designed with the idea of identification of only one class and the rest of entities are treated as outliers/anomaly.
- We train the model in identifying only one class only anything which does not fit in that one class is regarded as anomaly.

One Class Learning

- In the context of outlier detection, the outliers/anomalies cannot form a dense cluster as available estimators assume that the outliers/anomalies are located in low density regions.
- On the contrary, in the context of novelty detection, novelties/anomalies can form a dense cluster as long as they are in a low density region of the training data, considered as normal in this context.

One Class SVM

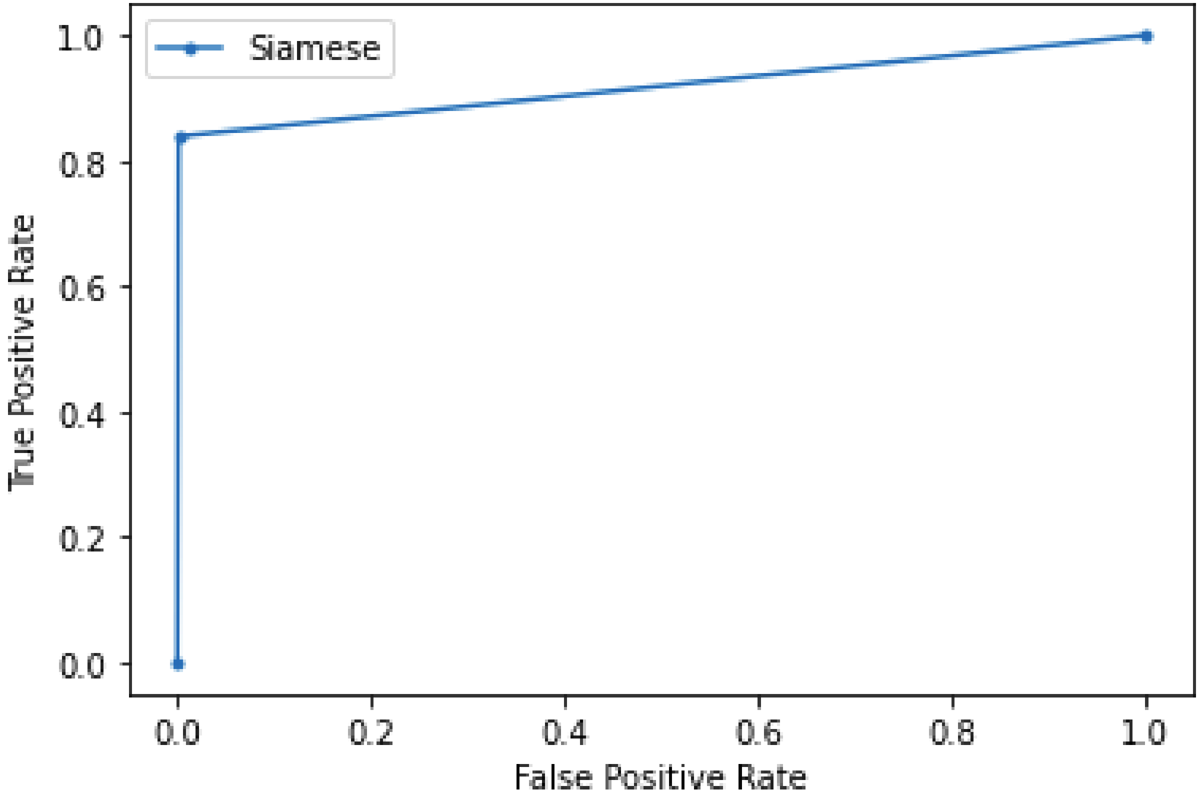
- ROC AUC: 0.69
- Minority Recall: 0.4
- Minority Precision:0.49



Siamese One Class Classification

- Unlike normal Siamese here we will train the model where only majority class are given as pair for similarity finding.
- And The other type of sample is where we are finding dissimilarities between majority sample and the minority class.
- There is a parameter k which decides the pairing of a sample with how many other samples.
- 2x k pairs where majority class pairs are used in one half of the pair and for the rest we have made pairs with the majority-minority class.

Siamese Classification



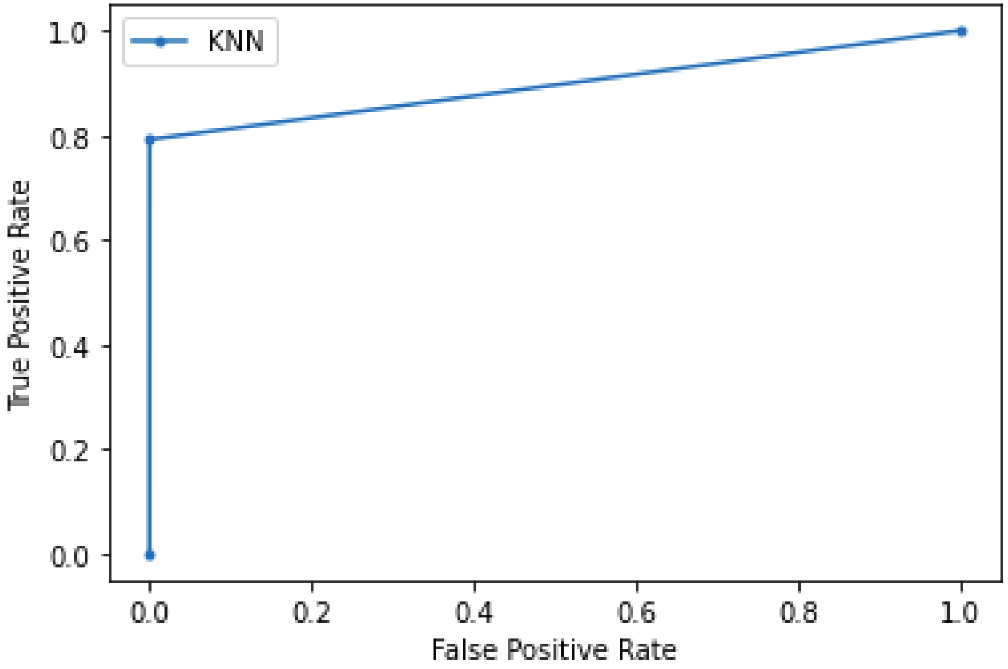
Siamese Classification

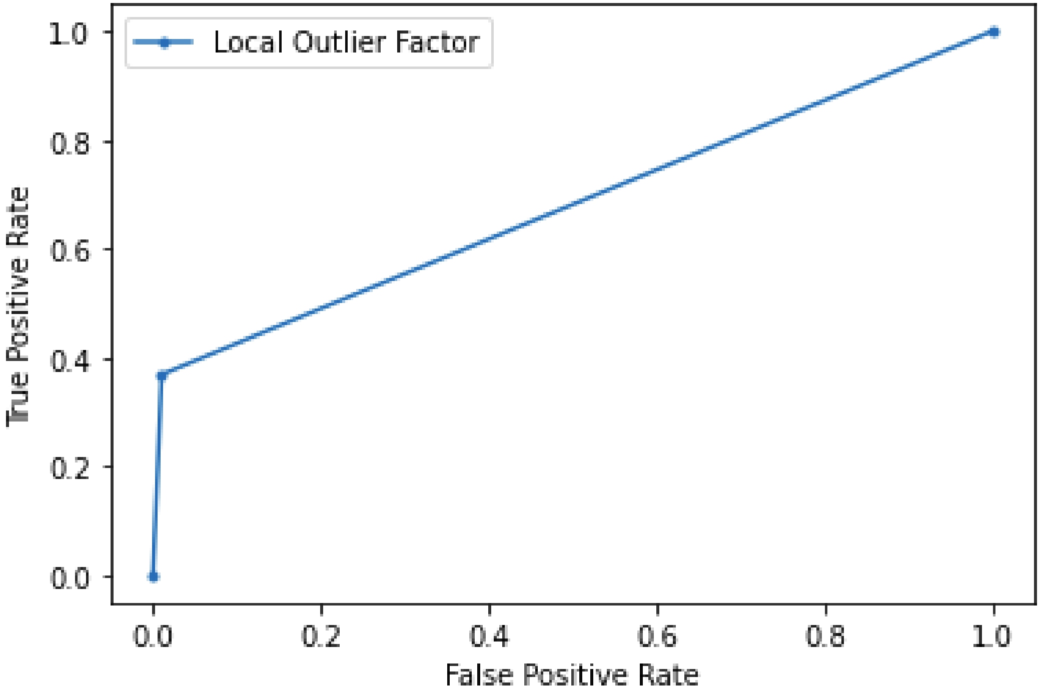
- ROC AUC: 0.92
- Minority Recall: 0.84
- Minority Precision:0.53

Some Other Methods:

KNN with weighted distances.
More weights to the closer points.

Local Outlier Factor





KNN weighted distances

- ROC AUC: 0.89
- Minority Recall: 0.79
- Minority Precision:0.94

Local Outler Factor

- ROC AUC: 0.68
- Minority Recall: 0.37
- Minority Precision:0.06

Hybrid Approach

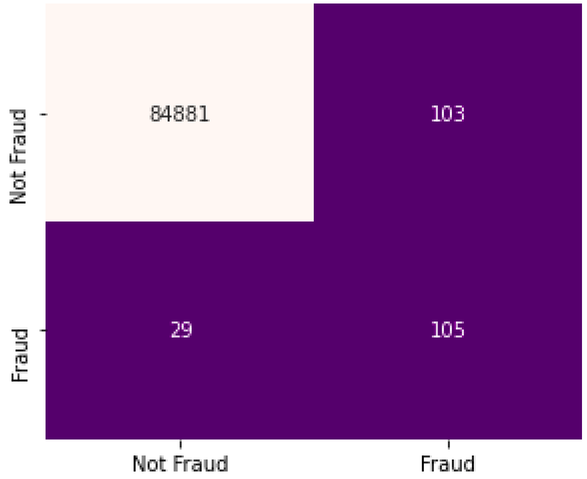
- They employ more than one machine learning algorithms to improve the classification quality, often through the hybridization with other learning algorithms to achieve better results.
- It is designed with the idea to alleviate the problem in sampling, feature subset selection, cost matrix optimization and fine-tuning the classical learning algorithms.

Some Hybrid Approach techniques

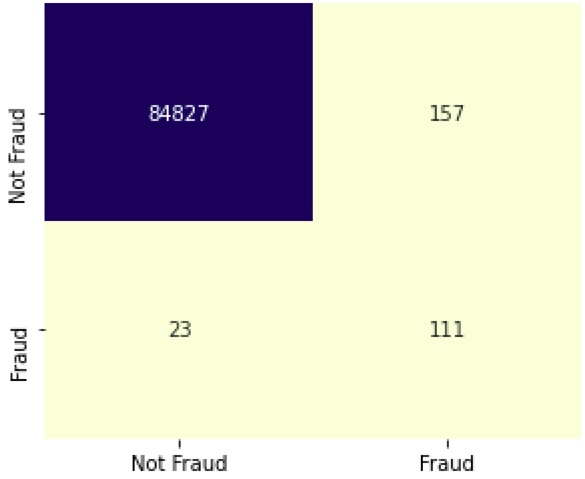
- Neural networks with SMOTE
- Exploiting cost sensitive in tree
- Fuzzy rule extraction with GA
- Undersampling and GA for SVM

Comparison

- Regular neural network

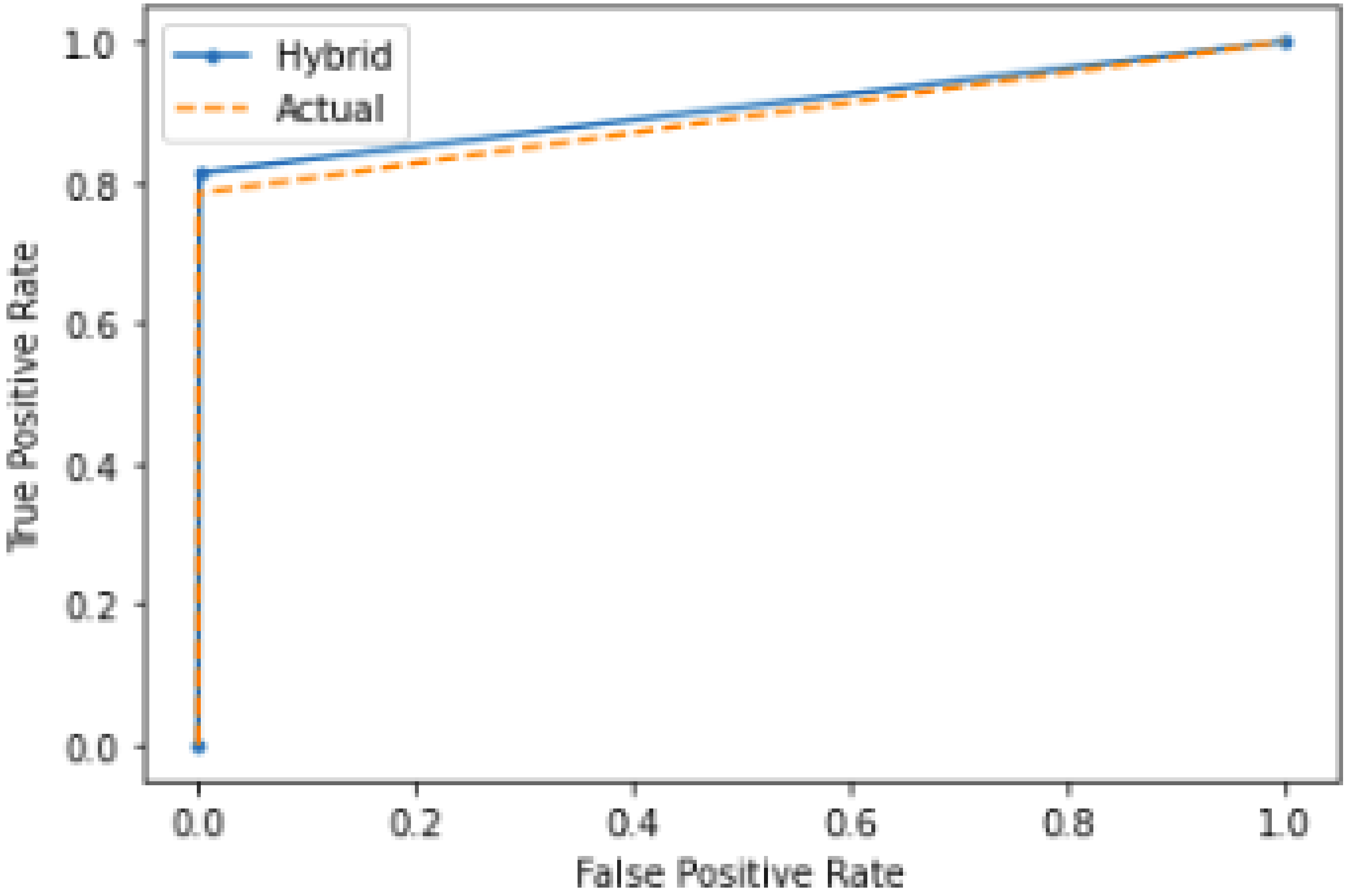


- Neural Network with SMOTe



Cost-Sensitive Neural Networks do a very good job in the binary classification,but we can go a step further to increase the performance even more.We can see here that with SMOTe the true positives experience a good rise though the accuracy is compromised a bit.

ROC_AUC plot for NN with and without SMOTe



The small elevation of the hybrid plot is evident that our hybrid model gives a better recall.

Efficiency comparison between different approaches for this dataset

Parameters used

- Recall of the minority class
- ROC_AUC score
- F1-Score
- Precision

Approach	Recall of Minority Class	ROC_AUC Score	F1-Score	Precision
SMOTE(Over Logistic Regression)	0.67	0.83570322	0.74	0.82
Ensemble Methods	0.76	0.85326172	0.83	0.92
Hybrid Approach	0.85	0.90596921	0.84	0.94
Cost Sensitive Logistic Regression	0.83	0.91212608	0.68	0.57
One Class SVM	0.4	0.69732	0.44	0.49
Siamese Classification	0.84	0.92	0.65	0.53

We have to make a trade-off between recall and precision and other parameters , after comparing all the approaches we went for SMOTe + Cost Sensitive Neural Network

Weekly Plan

Week 1	**Fine tuning our dataset with Data-Level approaches including-** **1.Some sampling Techniques** **2.Some Feature Selection Techniques**
Week 2	Testing different performance measures with different Algorithm-Level approaches including some from each category- 1.Improved Algorithm 2.One-class Learning
Week 3	3.Cost-sensitive Learning 4.Ensemble Method 5.Hybrid Approach Comparing and plotting results from different techniques.

Contribution from each Team-Member

Team Member	Techniques Implemented
Raman Shukla	• One-class Learning • Ensemble Method • Testing and result gathering
Sushant Kumar	• Sampling techniques • Cost-sensitive Learning • Testing and result gathering
Mohit Manwani	• Data-Level Approach • Hybrid Approach • Testing and result gathering

Conclusive Statements

On extensive analysis we conclude that there are some inevitable challenges that come with class imbalance problem.

After experimenting with various strategies it is evident that there is no sureshot method to eradicate the class imbalance problem.

The best method for different datasets are different.

We were able to minimize the ill-effects of the problem to a good extent with a cost-sensitive neural network along with smote for our dataset.