Anomaly Detection Challenge

CHALLENGE 3:

Finding Needle in a HayStack

Team: Abracadata
Ishmeet Kaur(03677735)
Mustika Rizki Fitriyanti(03667399)

1.
Introduction about the Data

1. Introduction about the Data

Aim: Classify the network traces in the test set as normal or anomalous with the help of a highly imbalanced training dataset.

Problem Type:Binary Classification

Dataset

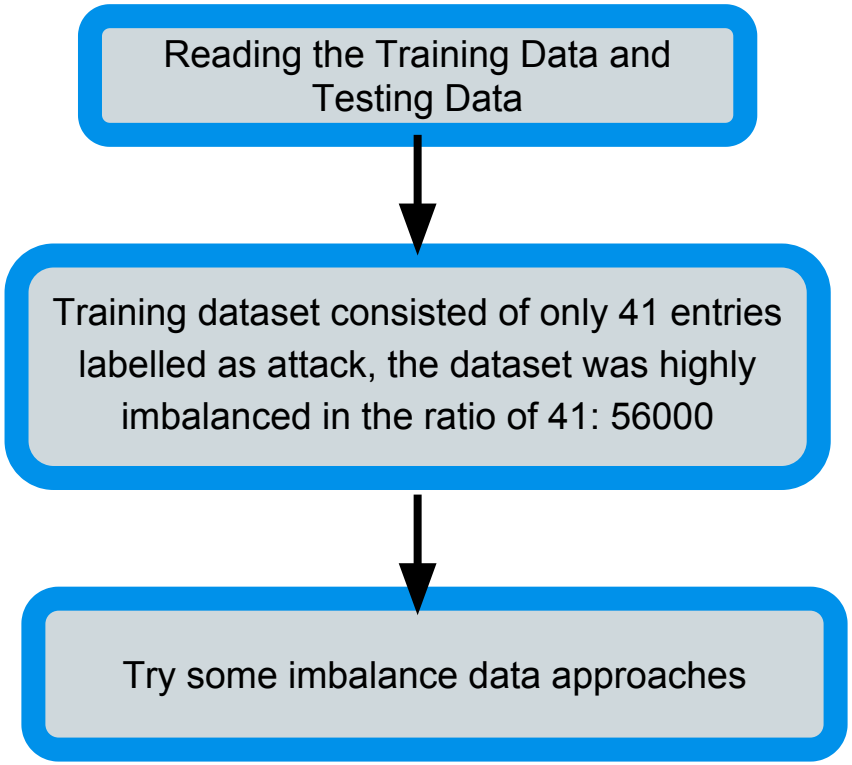
- Training Set: 56041 (only 41 entries were labelled as 1)
- Test Set: 82332

Number of Features: 43 (+ attack category)

2.
Data Preprocessing and Data Analysis

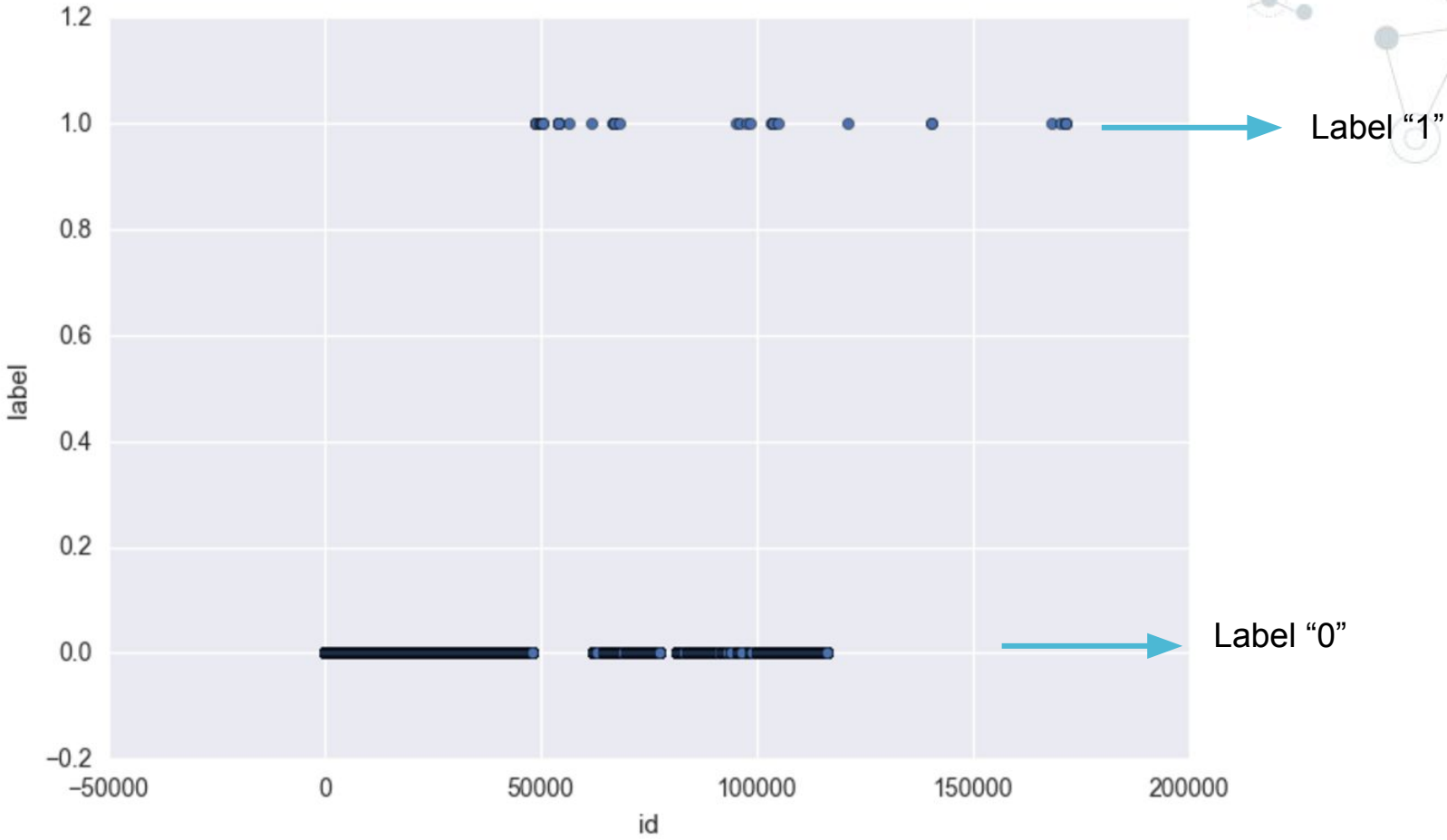
2.1
Data Preprocessing

2.1 Data Preprocessing



2.2
Handling Imbalanced Data

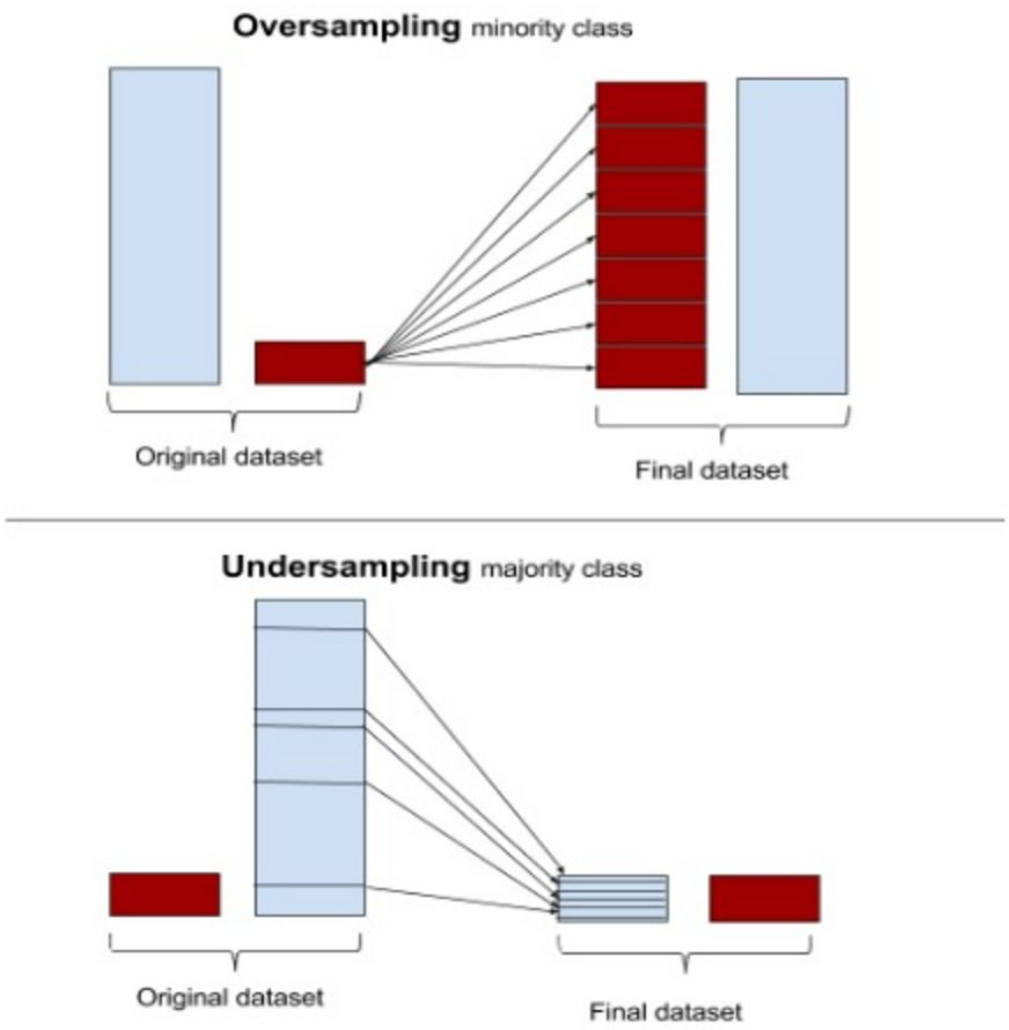
2.2 Why?


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
label
−50000
0
50000
100000
150000
200000
id
Label “1”
Label “0”

2.2 Imbalance Data Approaches

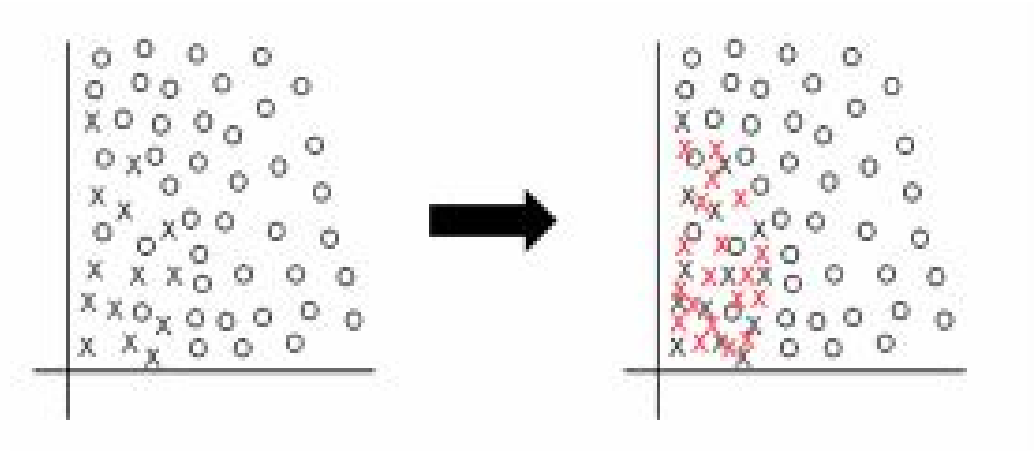
- ★ Entire Dataset(No modification):
- ★ Manual Random UnderSampling
- ★ Sampling Techniques using imbalanced-learn package
- ★ Anomaly Detection(One Class Learning)
- ★ Class Weights
- ★ Increased Metrics for Evaluation

2.2.1 Sampling Techniques using imbalanced-learn package



2.2.1 Sampling Techniques using imbalanced-learn package

Example of Oversampling approach:

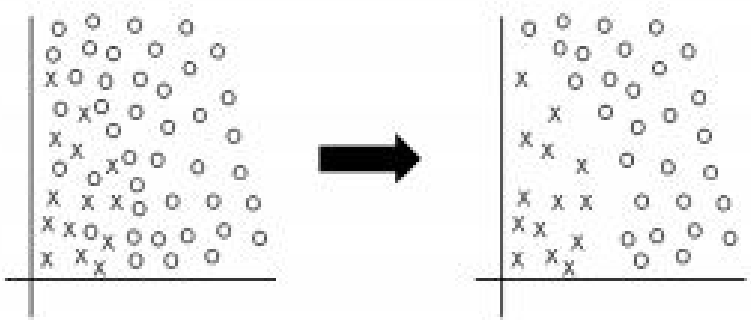


SMOTE

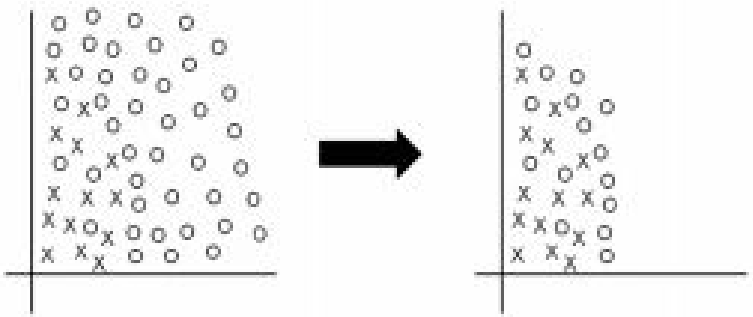
- Random minority over-sampling with replacement
- SVM SMOTE - Support Vectors SMOTE

2.2.1 Sampling Techniques using imbalanced-learn package

Example of Undersampling approaches:



Tomek Link

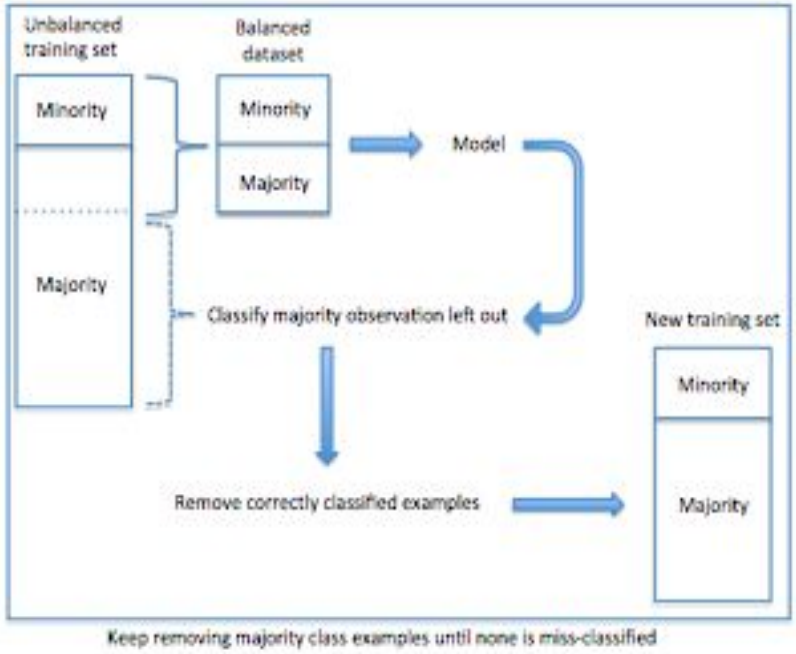


Condensed Nearest Neighbor

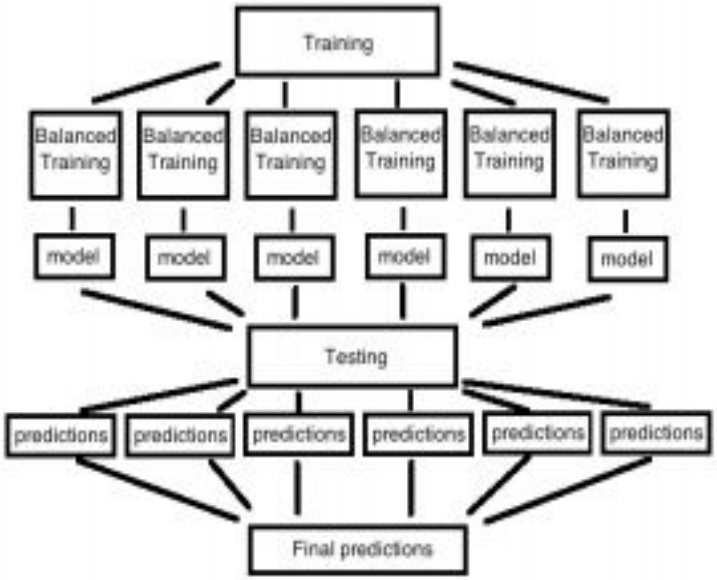
- ➔ Random Undersampling
- ➔ Tomek Links
- ➔ CNN(Condensed Nearest Neighbours)
- ➔ OSS(One Sided Selection)
- ➔ Under-sampling with Cluster Centroids
- ➔ Near miss methods
- ➔ Neighbourhood Cleaning Rule.

2.2.1 Sampling Techniques using imbalanced-learn package

Ensemble Sampling approaches:



Balance Cascade



Easy Ensemble

Anomaly Detection and Weighting:

- One Class SVM for Anomaly Detection.
- **Class Weighting:**

- SVC and SGD Classifier
- Random Forest Classifier
- Extra Tree Classifier
- Decision Tree
- Logistic Regression
- Adaboost Classifier

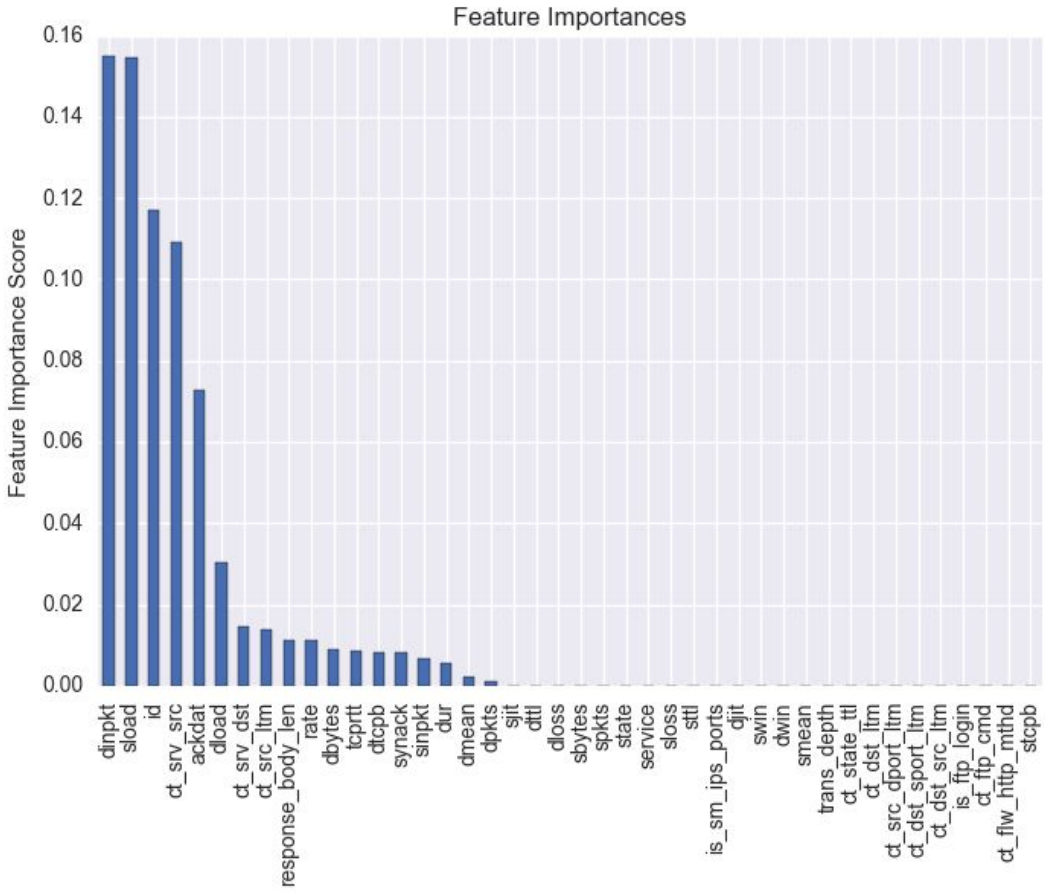
Set the class_weight in the classifier i

1. "Auto"
2. Set Manually, ex: {0:.1, 1:.9}

3.
Feature Engineering

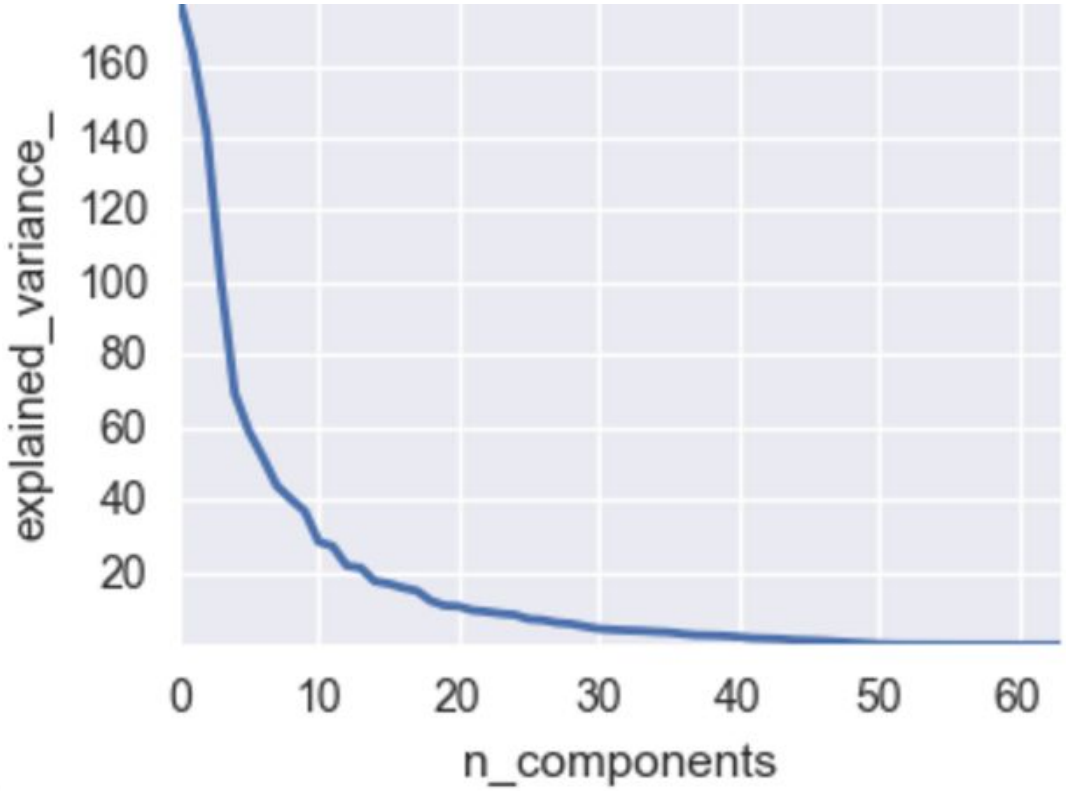
3.1 Feature Engineering

The features importances were analysed using simple brute force analysis and Gradient Boosting Classifier.



3.2 Dimensionality Reduction by PCA

- All the features were tried initially with and without sampling along with normalization .
- The reduced features by dimensionality reduction of PCA were also tried but did not give any good results.

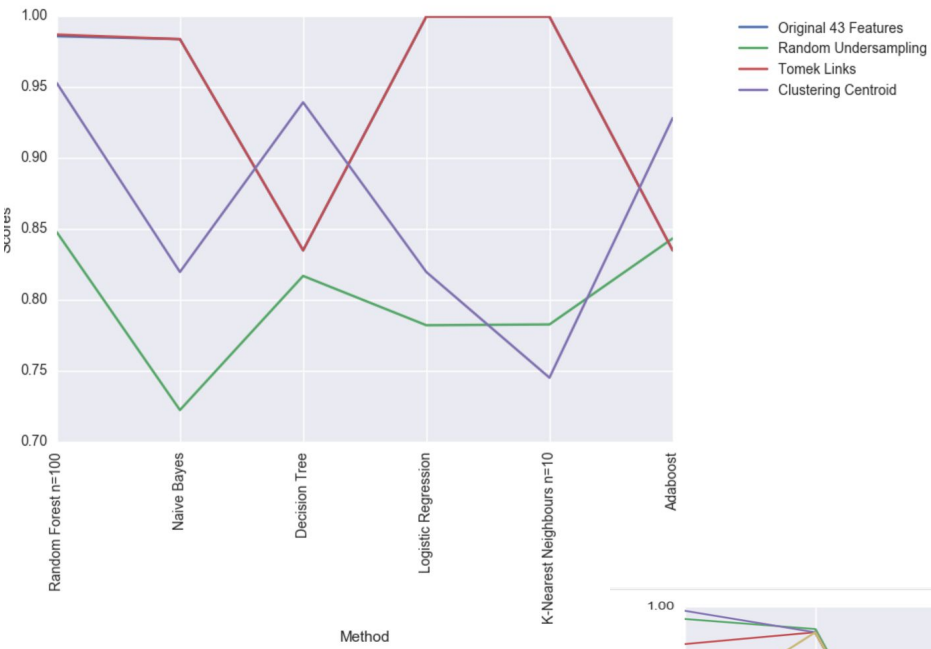


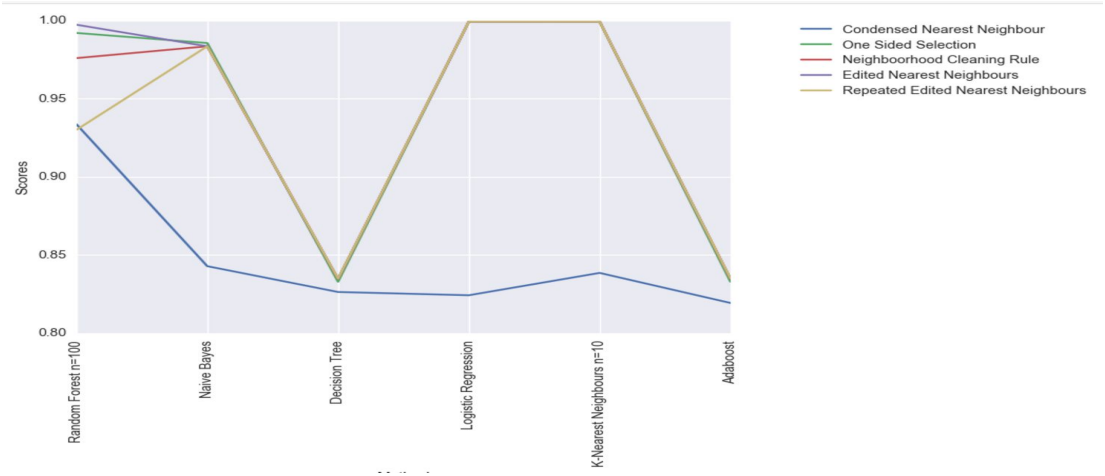
4.

Creation and Evaluation of the Models

- Naive Bayes Classifier
- Decision Tree Classifier
- K- nearest Neighbors
- Random Forest
- Logistic Regression
- AdaboostClassifier
- Support Vector Machine
- SGD
- Adagrad
- One Class SVM
- Gradient Boosting Classifier
- Extra Trees Classifier
- OneVs RestClassifier

4.1 Accuracy Score





4.2 Kaggle Score

Method	Public Score
Extra Trees + Undersampling	0.84612
Extra Trees + Undersampling + Manual Weights	0.79649

5. Learning

- Third party packages like lightning (for Adagrad Classifier) and Imbalanced -learn(for intelligent sampling) highly increased the accuracy.

- Accuracy (unlike previous challenges) standalone could not give an estimate of the better model.

- Ensemble Models (Bagging and Boosting) improved the accuracy.

- One Class SVM -high complexity ,SGD was used which is highly sensitive to learning rate alpha.(normalized dataset)

- Adagrad kept the learning rate constant, lead to overfitting

Thank you for your attention