Fraud Detection in Mobile Money Transaction

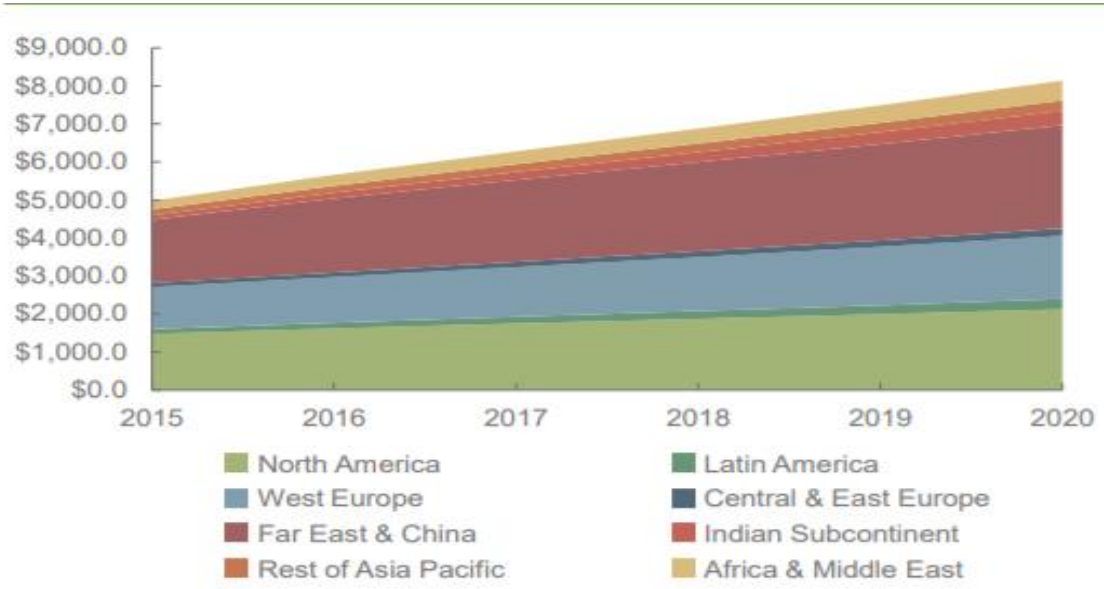
Group P17

Ankit Nanavaty(*ananava*) **Aditya Shah(*ahshah4*)**
Vishwas S P(*vsomase*) **Ashish Pawar(*akpawar*)**

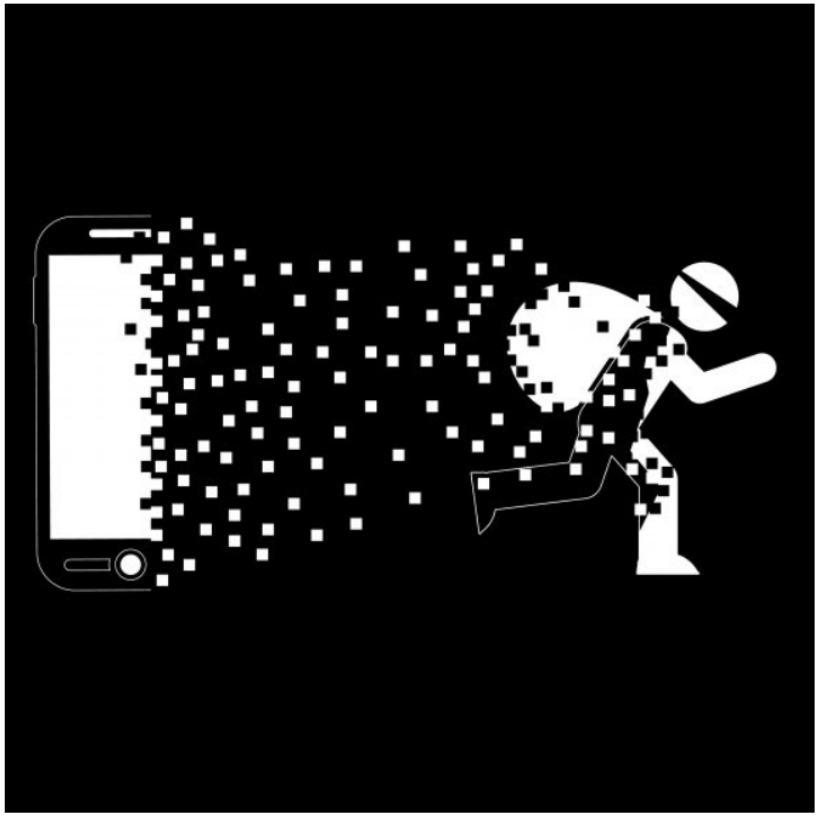
Introduction:

- Mobile transactions have continued to increase in the last 5 years.
- As it increases, online fraud detection continues to become a bigger issue.

Growth in eCommerce Transaction Value ($ billions) by region



Source: Juniper Research



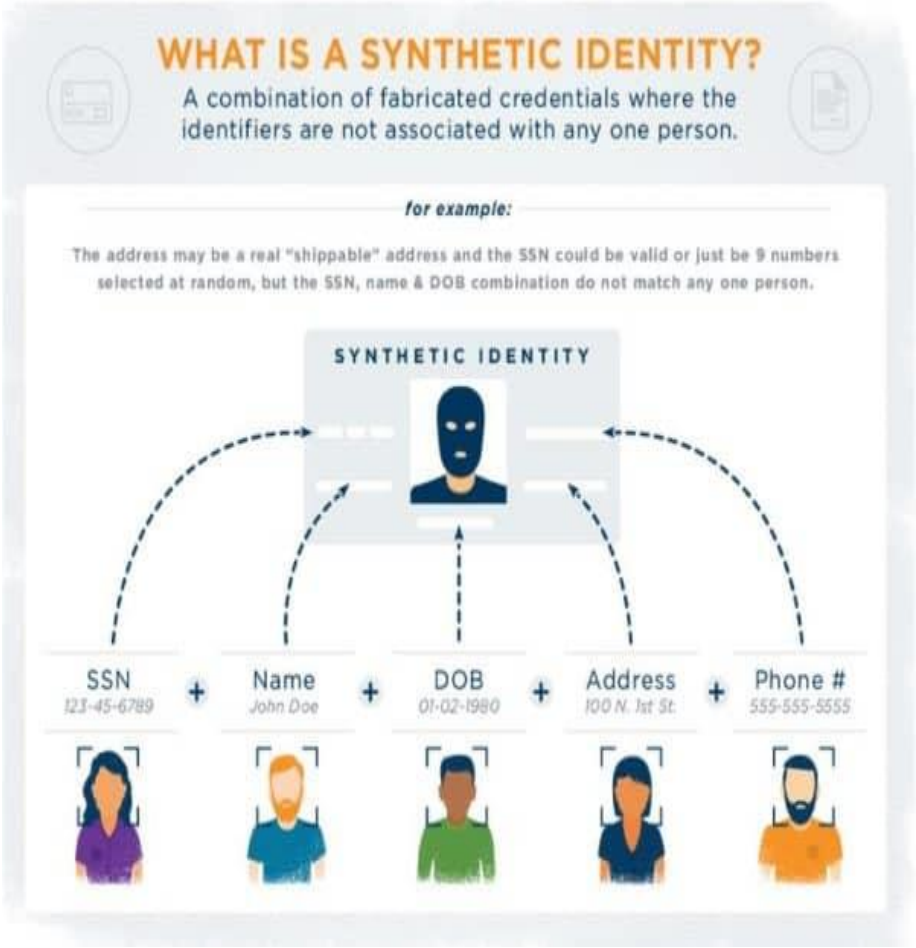
Source: Google Images

How much has it increased?

The 6th Annual Mobile Payments and Fraud report finds that 17% of merchants get more than half of their revenue from mobile, up from 10% in the 2017 survey and a paltry 3% in the 2015 survey.

How are these frauds happening ?

Synthetic Identities: 71 percent of merchants cite this type of fraud as their chief concern.



Loyalty fraud – This can happen when fraudsters intercept loyalty programs or members' accounts for theft and transfer of points. There are also cases in which points are sold and transferred to others for monetary gain.

Friendly fraud – This occurs when legitimate orders are disputed by the consumer, requiring merchants to refund payments (chargebacks). This form of fraud can be unintentional, with the consumer forgetting they placed the order, or one family member using another's payment card without permission

The research in the area of combatting such kind of frauds, motivates us to find a robust system to detect fraudulent transactions.

Problem in developing a model to detect these frauds:

- There is a lack of publicly available datasets on financial services and specially in the emerging mobile money transactions domain.

- So we used a synthetic dataset generated using the simulator called PaySim by E. A. Lopez-Rojas , A. Elmir, and S. Axelsson from Sweden.

- PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The original logs were provided by a multinational company, who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world.

There are 11 features for each transaction and one such transaction entry is shown below:

Features-

(Step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud, isFlaggedFraud)

Sample transaction-

(1, CASH_OUT, 416001.33, C749981943, 0, 0, C667346055, 102, 9291619.62, 1, 0)

Source: Google images

- Data is highly imbalanced
- [1][2] Research suggested that Smote technique had to be used to oversample the minority class, as a way to deal with imbalanced data.
- We can all agree upon the fact that it hurts a company to miss classify something than to correctly classify it in this scenario i.e. If there was a fraudulent transaction and the model did not catch it, that hurts more compared to other correct classification.
- So the goal is to reduce that, hence used Recall and Accuracy to check if the model is good or bad, rather than just Accuracy.
- [2][3] After referring few more papers we expected algorithms Like XGBoost, KNN and Random Forest to do well on this data.

References

[1] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer, “SMOTE : Synthetic Minority Oversampling Technique”, 2002.

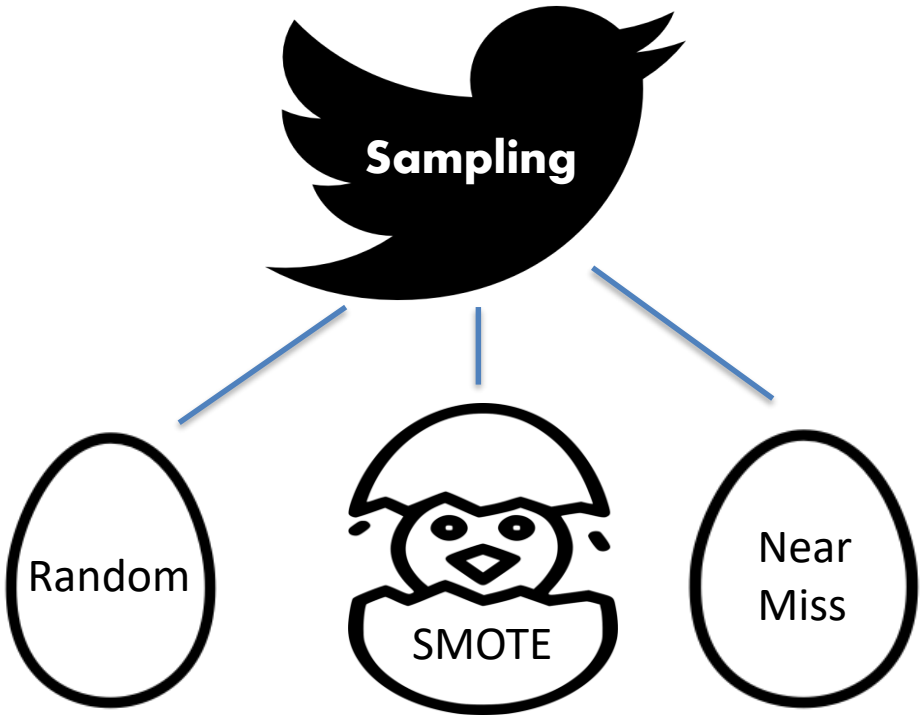
[2] E. A. Garcia and H. He, "Learning from Imbalanced Data," in IEEE Transactions on 225 Knowledge & Data Engineering, vol. 21, no. , pp. 1263-1284, 2008.

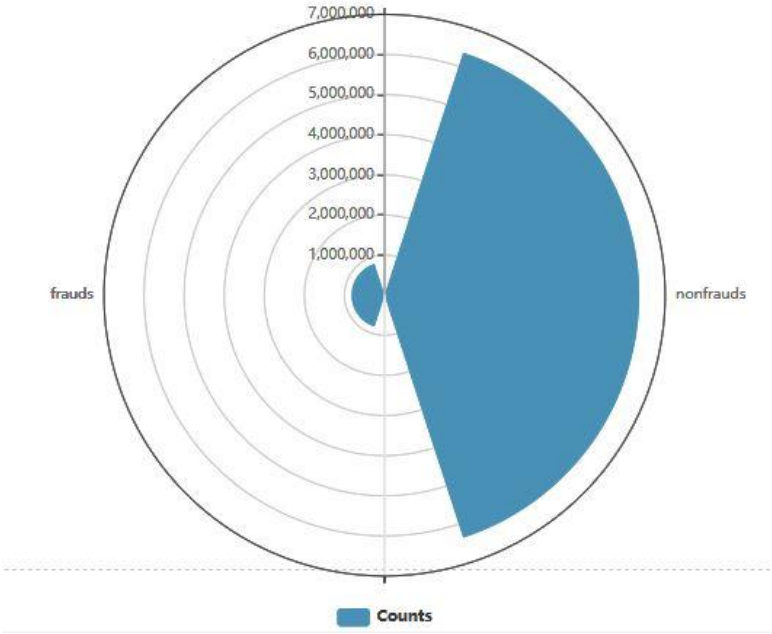
[3] Veni, C.V.. (2018). On the Classification of Imbalanced Data Sets. 10.13140/RG.2.2.14964.24961.

Fraud v/s Non-Fraud

Ratio of Fraudulent to Non-fraudulent transaction ≈ **0.001**

Conclusion – Highly imbalanced data.





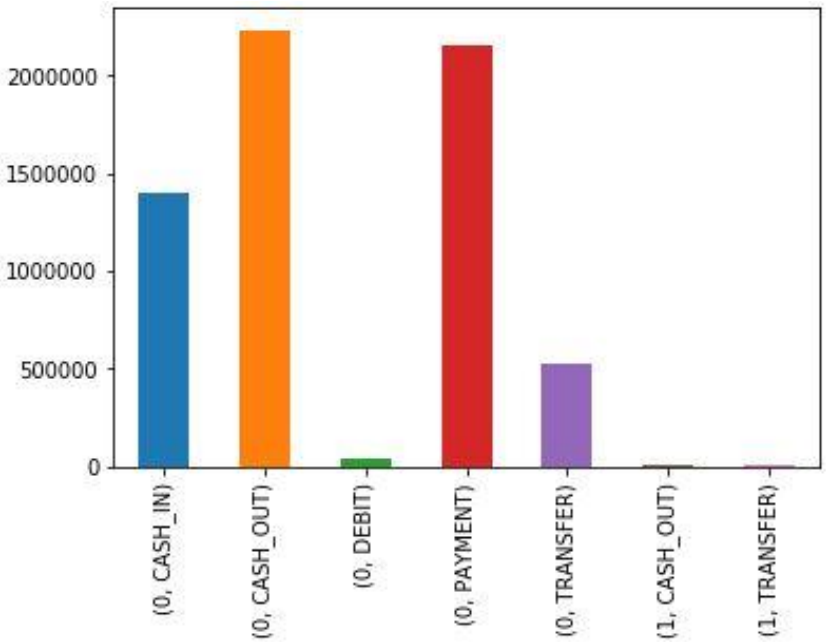
Fraud counts v/s ‘type’

```
# the below data and plot shows the distribution of the fraud transactions in the 'type' of data

# types of fraud transactions
fraud_trans = list(trans_data.loc[trans_data.isFraud==1].type.drop_duplicates().values)
print('Types of transactions that have Frauds: ',fraud_trans)

# plot for the same
fraud_count = trans_data.groupby(['isFraud','type']).size().plot(kind='bar')

# removing the data which do not have fraudalant transactions
trans_data_new = trans_data.loc[(trans_data['type'].isin(['TRANSFER','CASH_OUT'])),:]
```



Prediction

Logistic Regression

It is used when the response variable is categorical in nature.

KNN

A simple implementation of **KNN regression** is to calculate the average of the numerical target of the K nearest neighbors.

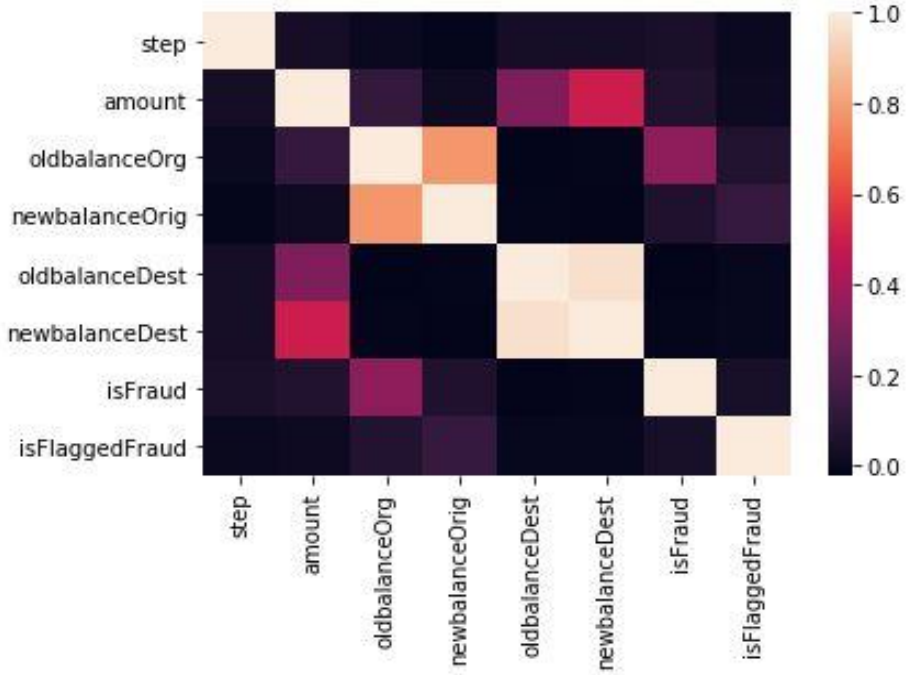
Random Forest

Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features resulting in a wide diversity that gives a better model.

XGBoost

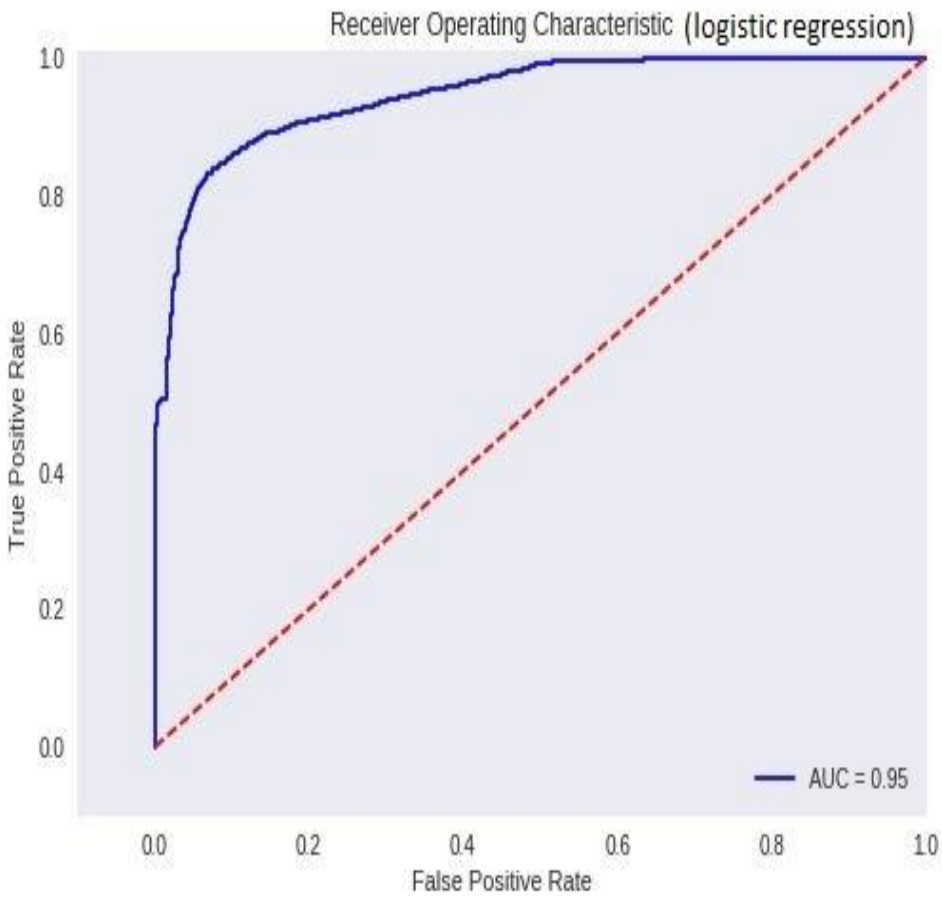
An implementation of the gradient boosted trees algorithm

Correlation Heatmap



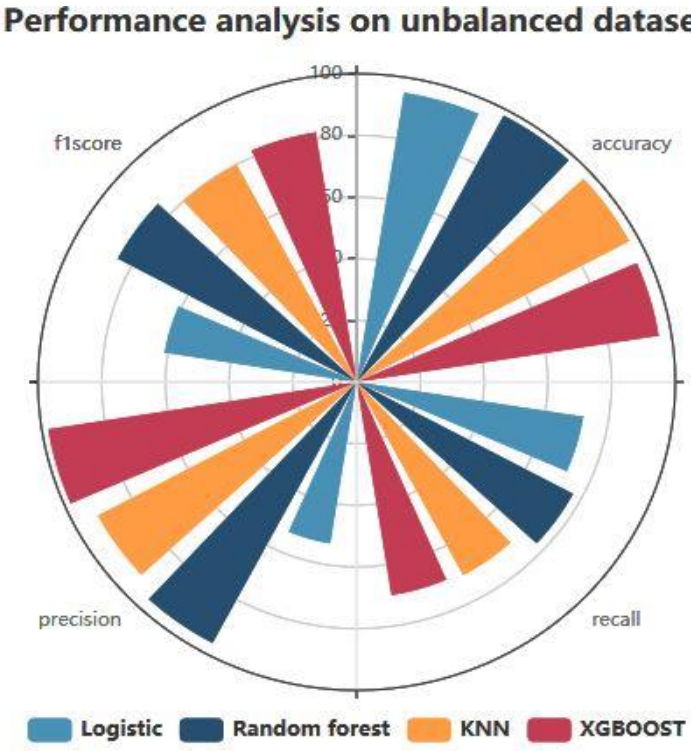
- Correlation among all features.
- Higher the brightness less the correlation
- Step , IsFraud and FlaggedFraud has low correlation.

ROC CURVE



- Trade off between TPR and FPR
- Area under curve(AUC) =0.95
- AUC implies efficacy of binary classification.
- Since AUC is very high model prediction is accurate.

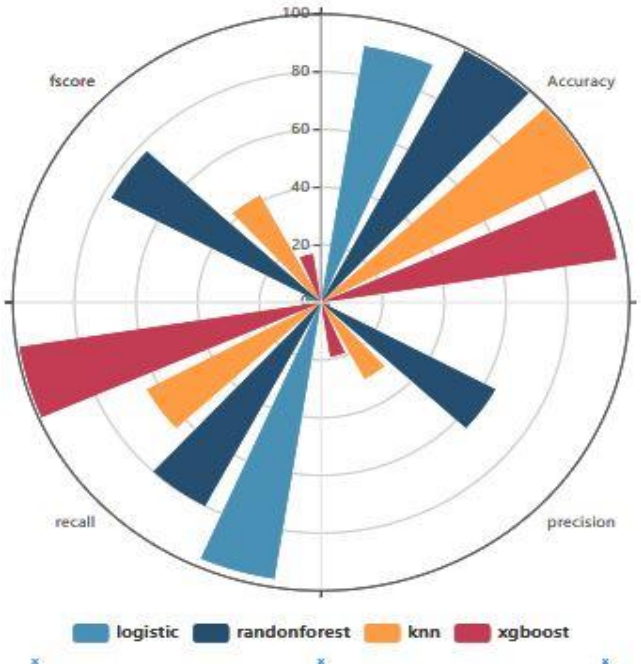
Model Comparison-I



	Accuracy	Precision	Recall	F1 score
Logistic	95.238	53.466	72.861	61.234
Random forest	99.924	96.972	77.138	85.928
KNN	99.897	92.866	71.892	80.892
Xgboost	99.902	98.632	70.986	82.557

Model Comparison-II

Performance analysis on balanced dataset



	Accuracy	Precision	Recall	F1 score
Logistic	89.88	2.705	97.37	5.264
Random forest	99.83	64.67	80.37	77.391
KNN	99.39	30.042	64.52	42.45
Xgboost	97.27	19.38	99.38	17.84

Challenges:

- Tuning of k-value for KNN and n_estimators for Random Forest algorithm, consumed a large chunk of time and processing power.
- Number of features available for classification were limited(11).
- Selecting the appropriate sampling technique for the imbalanced data by performing under-sampling and over-sampling.

Conclusion:
What is the result?

- As per the previous readings it can be conclude that recall and accuracy is the relevant parameter for balanced data and F1 score for imbalanced data.

- Smote leverages the performances of all the algorithms tested by balancing the data and provides a clear distinction which cannot be clearly concluded on the unbalanced data.

- Xgboost performs best in terms of recall (99.38) and accuracy(97.27) on balanced data.

Conclusion:
What does the result mean?

- As per a surge in online mobile transaction , it is the need of the hour to precisely detect fraudulent transactions.
- As most of the real world financial data is unbalanced smote proves to be a significant oversampling technique that can be used to balance the data for clear evaluation.
- Xgboost can lead to significant precise classification on unbalanced dataset in general when used with smote.

Thank You!

Any Question?