Fraud Detection System Report

2015B4A70594G
Kshitij R Dani

Do we need data processing ?

Yes, we need data pre-processing as a many columns and tuples have non-float values such as " ; , . [ ] { } () / ' | ". And so we need to remove them.

Do we need Data normalisation?

Data normalisation not required as most of the columns have already undergone PCA and so had to be standardised beforehand itself. Also, even though the 'Amount' column is on a different scale than the other column values, there isn't much effect on the clusters being generated.

Closest Clustering Algorithm:

DBScan was the closest algorithm that got us the result with a very good accuracy. This is probably because it groups clusters on the basis of density and true transactions happen to be closer to each other the fake transactions. It also had the least Root Mean Square Error among the algorithms analysed.

We first took the data into a data frame and cleaned it. After realising that normalisation was not required, we proceeded with further reducing the data using PCA till it was just 2 features. After this we ran clustering algorithms on both the cleaned data as well as the data obtained after further PCA.

Insights and Inferences:

The data necessarily had to be cleaned before any procedure could take place. Before we ran any clustering algorithm on the data, we first plotted the 'Elbow Plot' for k-means just to verify the numbers of clusters that K-means would ideally need.

We ran K-means and DBScan on just the cleaned data and K-means and Birch on the data set after PCA was done to reduce its dimensions into just 2 features.

1) **K-means (with just cleaned data) :** accuracy = 53.7 % ; Root Mean Square Error = 0.681 ; Correlation = -0.01

2) **K-means (with further reduced data) :** accuracy = 44.9% ; Root Mean Square Error = 0.741 ; Correlation = ~0.01

3) **DBScan (with eps = 3000 and min_samples = 2) :** accuracy = 99.8% ; Root Mean Square Error = 0.10 ; Correlation = -0.00036

4) **DBScam (with eps=1000 and min_sample = 2) :** accuracy = 99.8% ; Root Mean Square Error = 0.041 ; Correlation = 0.0001

5) **Birch (with further reduced data) :** accuracy = 0.31% ; Root Mean Square Error = 0.99 ; Correlation = 0.0015

Result :

From our analysis, it appears that DBScan is able to quite accurately classifying the transactions. This make sense as we can expect all valid transactions to be closer to each other, and therefore more dense. K-means and Birch aren't ideal to apply to this situation as we aren't considered with finding clusters with centroid, but rather a continuous cluster that holds a majority of points except a few. Therefore, our current model should be able to tell whether an incoming transaction is valid/invalid