PROJECT REPORT

on

Credit Card Fraud Detection System

(CSE V Semester Mini project PCS-504)

2021-2022

Submitted to:

Ms. Preeti Chaudhary

(CC-CSE-D-V-Sem)

Guided by:

Mr. Prateek Srivastava

(Resource Person)

Submitted by:

Mr. Chinmay Tiwari

Roll. No.: 1918306

CSE-D-V-Sem

Session: 2021-2022

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

CERTIFICATE

Certified that Mr. Chinmay Tiwari (Roll No.- 1918306) has developed mini project on "Credit Card Fraud Detection" for the CS V Semester Mini Project Lab (PCS-504) in Graphic Era Hill University, Dehradun. The project carried out by Students is their own work as best of my knowledge.

Date:16/12/2021

(Ms. Preeti Chaudhary)	(Mr. Prateek Srivastava)
Class Co-ordinator	**Project Guide**
CSE-D-V-Sem	Resource Person
(CSE Department)	(CSE Department)
GEHU Dehradun	GEHU Dehradun

ACKNOWLEDGMENT

We would like to express our gratitude to The Almighty Shiva Baba, the most Beneficent and the most Merciful, for completion of project.

We wish to thank our parents for their continuing support and encouragement. We also wish to thank them for providing us with the opportunity to reach this far in our studies.

We would like to thank particularly our project Co-ordinator Ms. Preeti Chaudhary and our Project Guide Mr. Prateek Srivastava for his patience, support, and encouragement throughout the completion of this project and having faith in us.

At last but not the least we greatly indebted to all other persons who directly or indirectly helped us during this work.

Mr. Chinmay Tiwari

Roll No.- 1918306

CSE-D-V-Sem

Session: 2020-2021

GEHU, Dehradun

Table Of Contents

ABSTRACT:

The goal of this project is to build a Credit Card Fraud Detection System with the help of machine learning.

Credit Card Fraud Detection System is a process of data investigation by a Data Science and the development of a model that will provide the best results in revealing and preventing fraudulent transactions. This is achieved through bringing together all meaningful features of card users' transactions, such as Date, User Zone, Product Category, Amount, Provider, Client's Behavioural Patterns, etc. The information is then run through a subtly trained model that finds patterns and rules so that it can classify whether a transaction is fraudulent or is legitimate. For implementing this we used creditcard.csv dataset available on Kaggle which consisted of all the required data mentioned above.

As we are building a recommendation system based on their similarity index, it is a part of unsupervised machine learning.

Unsupervised learning refers to the use of artificial intelligence algorithms to identify patterns in data sets containing data points that are neither classified nor labelled.

The algorithms are thus allowed to classify, label and/or group the data points contained within the data sets without having any external guidance in performing that task.

In other words, unsupervised learning allows the system to identify pattern within dataset on its own

MOTIVATION:

In this project, I have designed, implemented, and analysed a Credit Card Fraud Detection System using Machine Learning. By Building this project I got to learn about various ways in which we can build a detection system and built implemented one here.

SOFTWARE REQUIREMENTS:

•Jupyter Notebook

• Libraries (Pandas, NumPy, Matplotlib, Seaborn, Streamlit)

HARDWARE REQUIREMENTS:

• 2 GHz Intel or high processor

• Minimum of 180 GB HDD

• At least should have 2 GB RAM

LANGUAGE USED:

• Python

LIBRARY USED:

Pandas – It is a library used in Python Programming Language for the manipulation of the data and its analysis. It lets us perform various operations like selecting, reshaping, data cleaning, merginf.

Numpy – NumPy is a Python Programming Language library that is used to provide us a simple yet powerful data structure and is also used to perform a number of mathematical operations on arrays.

Matplotlib.pyplot – It is a plotting library used in Python Programming Language and it it is used to provide an object oriented apifor displaying bar plots.

Sklearn – Scikit-learn is a Python Programming Language library that provides us a lot of supervised and unsupervised machine learning algorithms. It is a very essential library for ML in Python. It consists of classification, regression, clustering etc.

Seaborn – Seaborn is a Python library used mainly for data visualization and data exploration analysis. It is based on matplotlib. It provides us a high-level interface required for drawing attractive and informative statistical graphics.

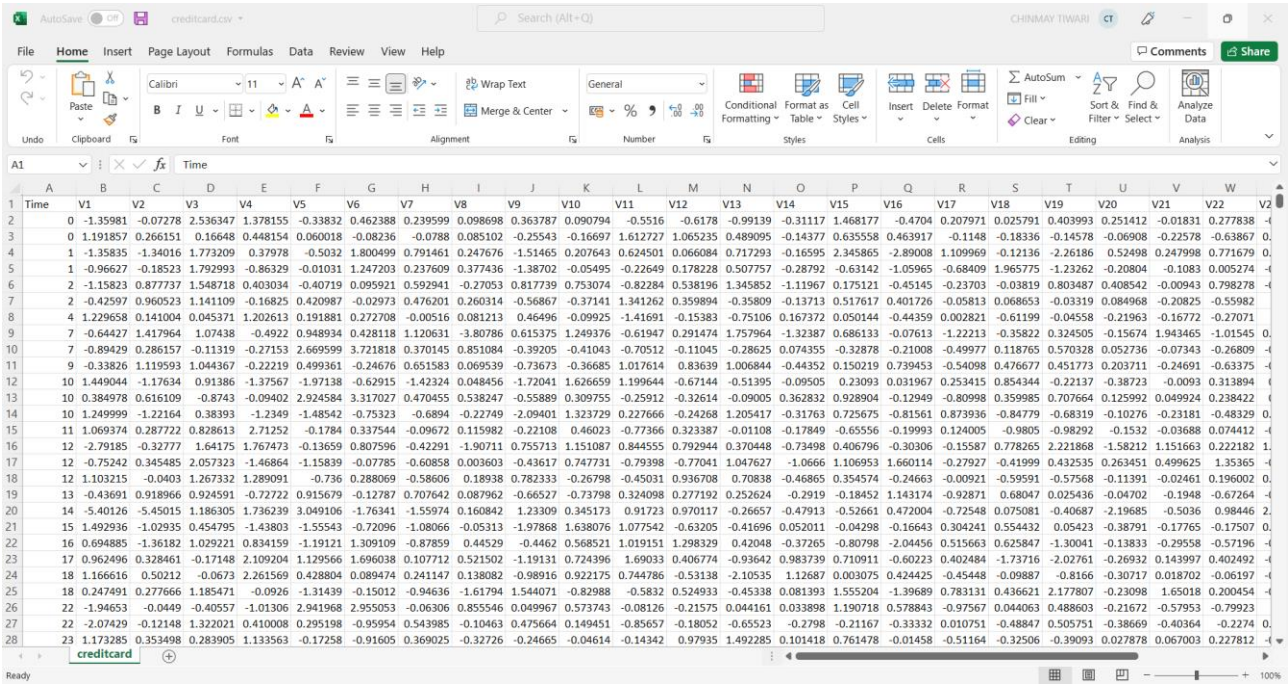
Importing the required Python libraries

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

%matplotlib inline
sns.set()
warnings.simplefilter('ignore')
```

DATASET:

I have used creditcard.csv dataset available on Kaggle website for my project. It consists of 284808 rows and 31 columns.



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22
2	0	-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.403993	0.251412	-0.01831	0.277838
3	0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867
4	1	-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-2.26186	0.52498	0.247998	0.771679
5	1	-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.23262	-0.20804	-0.1083	0.005274
6	2	-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.803487	0.408542	-0.00943	0.798278
7	2	-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.03319	0.084968	-0.20825	-0.55982
8	4	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071
9	7	-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.324505	-0.15674	1.943465	-1.01545
10	7	-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.570328	0.052736	-0.07343	-0.26809
11	9	-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.451773	0.203711	-0.24691	-0.63375
12	10	1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.22137	-0.38723	-0.0093	0.313894
13	10	0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.707664	0.125992	0.049924	0.238422
14	10	1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329
15	11	1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.98292	-0.1532	-0.03688	0.074412
16	12	-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.221868	-1.58212	1.151663	0.222182
17	12	-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.432535	0.263451	0.499625	1.35365
18	12	1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196002
19	13	-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264
20	14	-5.40126	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.55974	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446
21	15	1.492936	-1.02935	0.454795	-1.43803	-1.55543	-0.72096	-1.08066	-0.05313	-1.97868	1.638076	1.077542	-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	0.05423	-0.38791	-0.17765	-0.17507
22	16	0.694885	-1.36182	1.029221	0.834159	-1.19121	1.309109	-0.87859	0.44529	-0.4462	0.568521	1.019151	1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847	-1.30041	-0.13833	-0.29558	-0.57196
23	17	0.962496	0.328461	-0.17148	2.109204	1.129566	1.696038	0.107712	0.521502	-1.19131	0.724396	1.69033	0.406774	-0.93642	0.983739	0.710911	-0.60223	0.402484	-1.73716	-2.02761	-0.26932	0.143997	0.402492
24	18	1.166616	0.50212	-0.0673	2.261569	0.428804	0.089474	0.241147	0.138082	-0.98916	0.922175	0.744786	-0.53138	-2.10535	1.12687	0.003075	0.424425	-0.45448	-0.09887	-0.8166	-0.30717	0.018702	-0.06197
25	18	0.247491	0.277666	1.185471	-0.0926	-1.31439	-0.15012	-0.94636	-1.61794	1.544071	-0.82988	-0.5832	0.524933	-0.45338	0.081393	1.555204	-1.39689	0.783131	0.436621	2.177807	-0.23098	1.65018	0.200454
26	22	-1.94653	-0.0449	-0.40557	-1.01306	2.941968	2.955053	-0.06306	0.855546	0.049967	0.573743	-0.08126	-0.21575	0.044161	0.033898	1.190718	0.578843	-0.97567	0.044063	0.488603	-0.21672	-0.57953	-0.79923
27	22	-2.07429	-0.12148	1.322021	0.410008	0.295198	-0.95954	0.543985	-0.10463	0.475664	0.149451	-0.85657	-0.18052	-0.65523	-0.2798	-0.21167	-0.33332	0.010751	-0.48847	0.505751	-0.38669	-0.40364	-0.2274
28	23	1.173285	0.353498	0.283905	1.133563	-0.17258	-0.91605	0.369025	-0.32726	-0.24665	-0.04614	-0.14342	0.97935	1.492285	0.101418	0.761478	-0.01458	-0.51164	-0.32506	-0.39093	0.027878	0.067003	0.227812

DATA CLEANING:

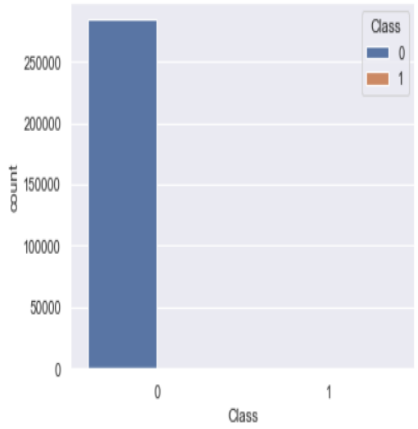
In the data cleaning, every features percentage missing of missing values, number of unique values, and percentage of biggest category were considered.

I have stored all the columns after the 11th column in another variable cols as these were the important features which would be used for clustering.

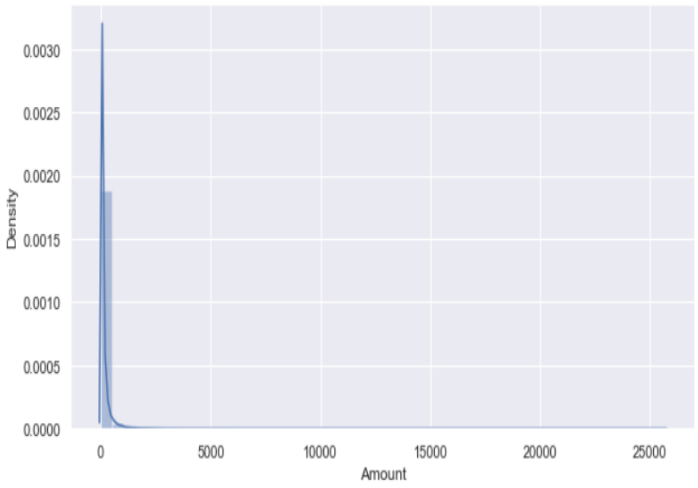
Coulmn 11 (i.e unnamed) was deleted as this playing no role in the recommendation system.

DATA VISUALIZATION:

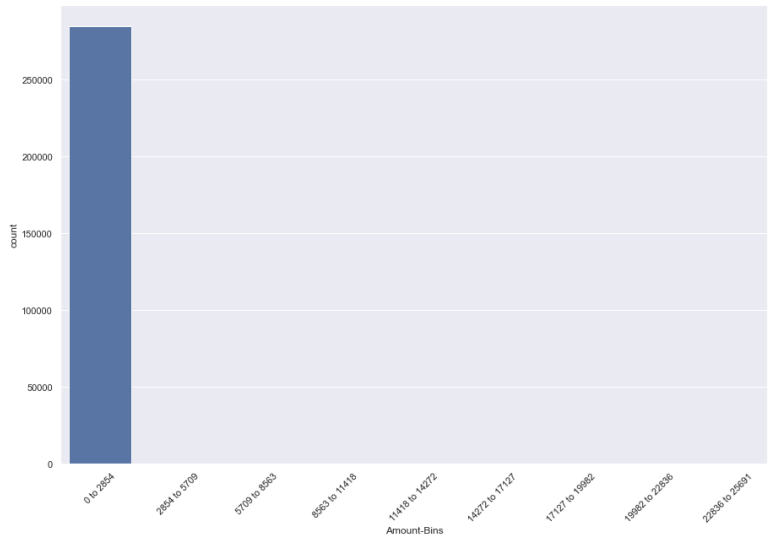
Data visualization is the discipline of trying to understand data by placing it in a visual context so that patterns, trends and correlations that might not otherwise be detected can be exposed. I have used python libraries like seaborn and matplotlib for data visualization.



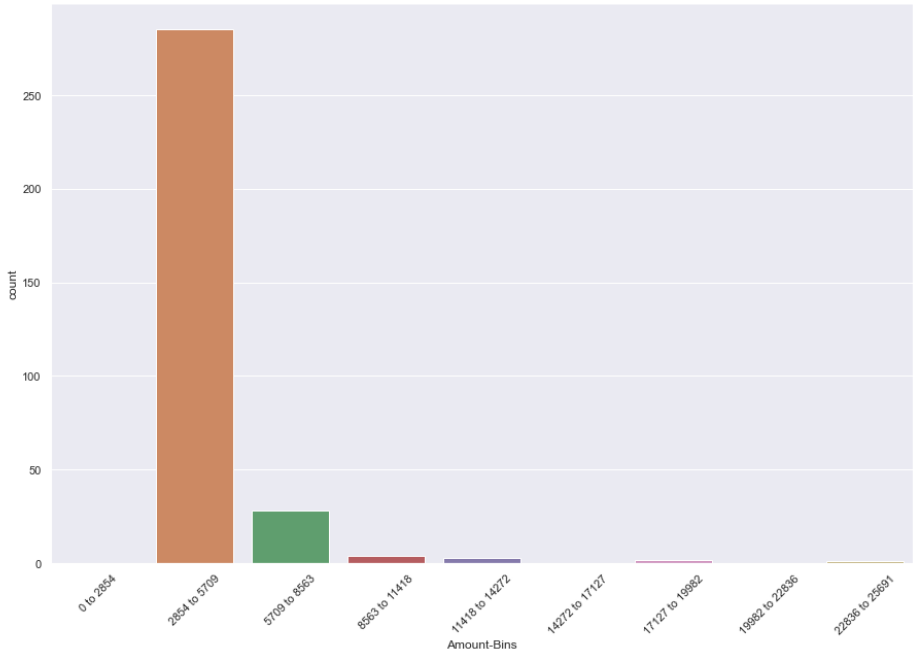
By looking at the above statistics, we can see that the data is highly imbalanced. Only 492 out of 284807 are fraud.



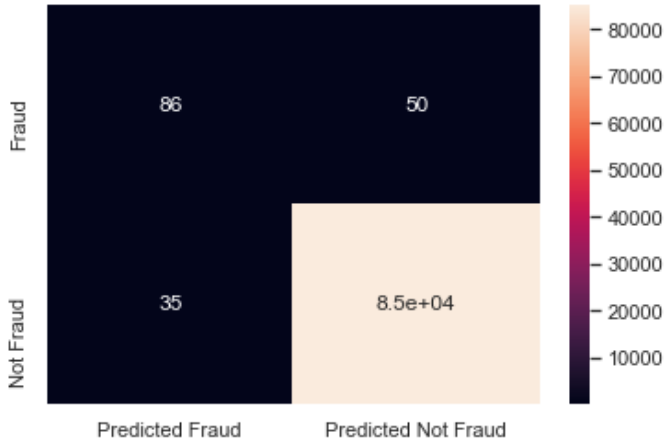
Since, it is a little difficult to see. Let's engineer a new feature of bins.



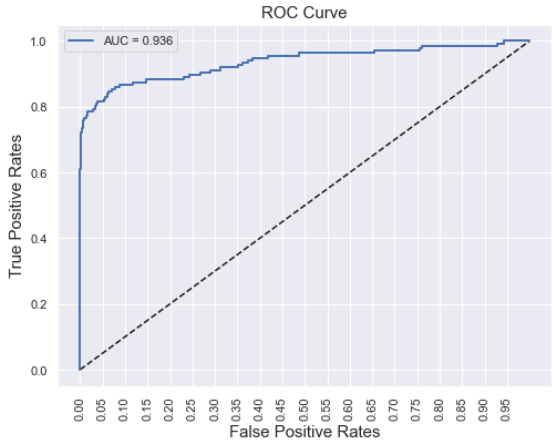
Since, count of values of Bins other than '0 to 2854' are difficult to view. Let's not insert the first one.



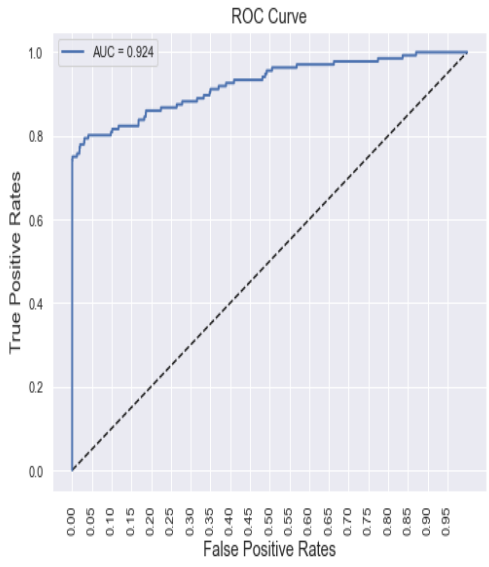
We can see that mostly the amount is between 0 and 2854 euros.



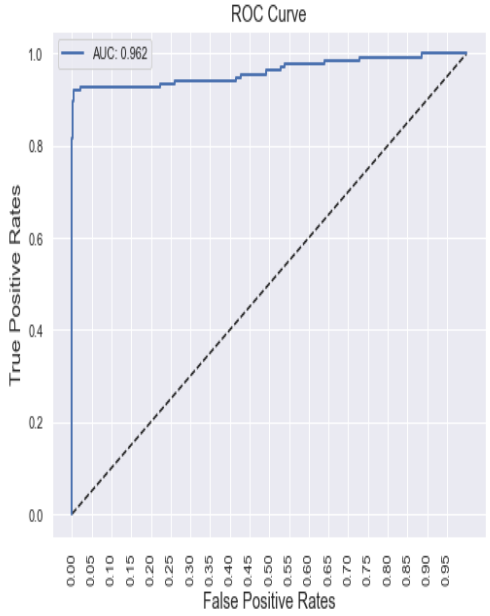
Heatmap also suggests that the data is highly imbalanced.



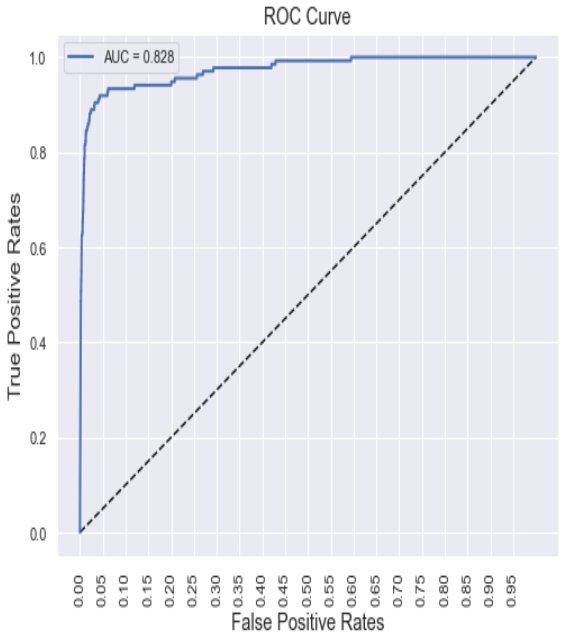
AUC is quite good. i.e. 0.965. Based on the data being highly imbalanced, we'll only check the AUC metric in later algorithms.



First degree is better in Logistic Regression case which gives 0.965 AUC Score.



The score AUC Score SVC gave is also pretty great. But it's still less than Logistic Regression Model. But the Recall increased significantly.



Modelling:

Now we will be explaining the different algorithms we used for Prediction:

1. k-nearest neighbors (KNN):

KNN or k-nearest neighbours is an algorithm which is used for classification and regression problems. But, it is widely used for classification. This algorithm is fair amongst all the parameters. It is mainly used because its very easy for prediction and has a low calculation time as compared to other algorithms. The classification mainly depends on the majority of votes received. The "K" in KNN states for the nearest neighbour whose vote we want to take for the classification. When the value of K is 1, object is allocated to the nearest.

KNN can be implemented by using following steps:

1. Importing the dataset.
2. Initializing the value of K
3. The value of K is chosen on the basis of following factors
    - The most important task in KNN algorithm is determining the value of k. If the value of k is less, it states that our model is overfitted but if its value is large, the logic behind the algorithm is lost. Thus it is defined by k = sqrt(n).
    - We can also use cross validation for optimizing the results.
    - We can also run each value of k for different instances and check for the optimal result.

Distance between test data and each row of training data can be calculated. We have used Euclidean distance as it is the most popular one but other metrics like Manhattan , Chebyshev and Cosine can be used.

2. Random Forest Classifier (RF):

Random Forest is a supervised machine learning algorithm. It is used both for classification and for regression. It is very easy to use. Unlike a forest is comprised of trees, and it is said that a forest which has more number of trees, is more potent.

Random forest creates a number of decision trees on randomly selected data, samples it, gets the prediction from each of the tree and provides the best solution by a method of voting. It also provides a nice indicator of the feature.

It works in four simple steps:

1. Firstly it selects random samples from the given dataset.

2. Then, it constructs a decision tree for each sample and produces a prediction result from each of the decision tree.
3. Next, it performs a vote for each of the predicted result.
4. Lastly, it selects the prediction result which is having the most votes as the final prediction.

3. Logistic Regression (LR):

Logistic Regression is a supervised Machine Learning Algorithm used to model the probability of certain events. It is used to predict Binary Outcome (when there are only two possible outcomes for a problem i.e. 1 or 0). There are two types of Logistic Regression Binary and Multi Linear Function. It is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability .The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. It is used to map predictions to probabilities. The function maps any real value into another value between 0 and 1 .t is a technique to analyse a data-set which has a dependent variable and one or more independent variables to predict the outcome in a binary variable, meaning it will have only two outcomes.

For this model we have only two possibility of outcome i.e. flood occurred or not which can be mapped to 1 or 0. Based on the fact of probability, it is used to predict certain events further used mostly in predictive analysis. It uses a logistic sigmoid function to predict the value. The dependent variable used in logistic model is categorical in nature and are known as target variables and the independent variables are known as predictors.

We use sigmoid function to predict the categorical values and the threshold decides the outcome. Logistic Regression equation: $\boldsymbol{p = 1 / 1 + e^{-(\beta 0 + \beta 1 X1 + \beta 2 X2 .... + \beta Xn)}}$.

4. SVM:

Support Vector Machine" (SVM) is a supervised Machine Learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number

of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

5.EXTRA TREES:

Extra Trees is an ensemble machine learning algorithm that combines the predictions from many decision trees. ... It can often achieve as-good or better performance than the random forest algorithm, although it uses a simpler algorithm to construct the decision trees used as members of the ensemble.

ACCURACY:

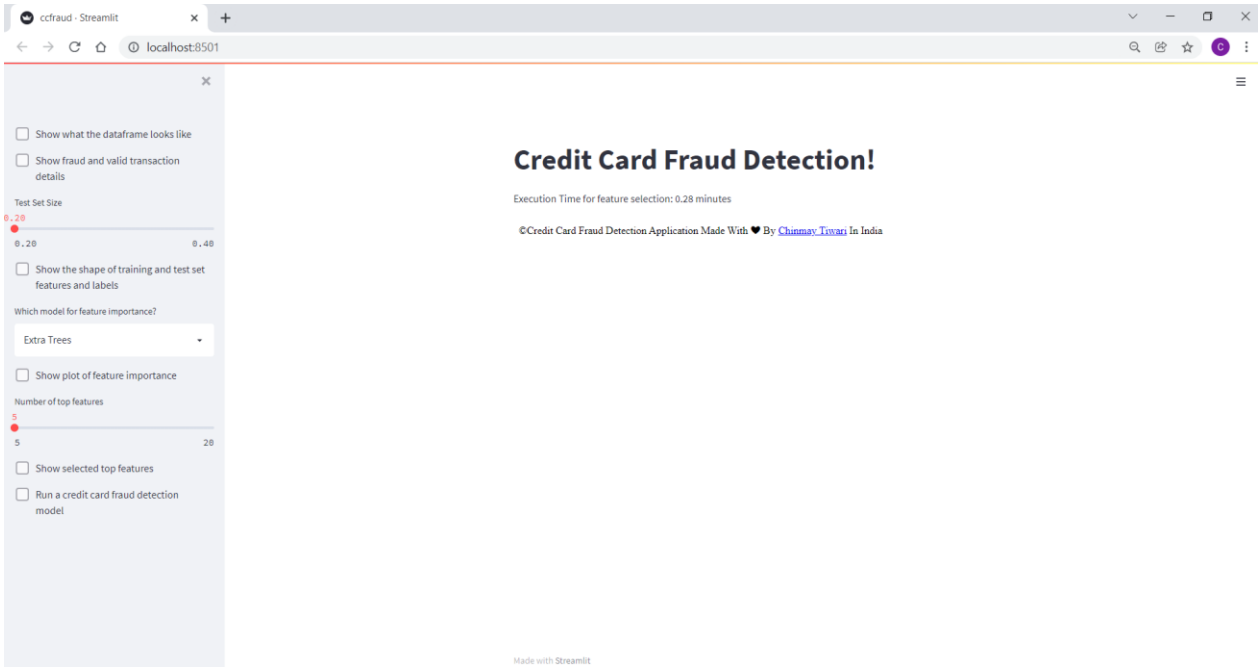
Algorithms	Accuracy	Recall	ROC Score
KNN	0.9994864932417329	0.78	0.86
Random Forest	0.9994777153059523	0.78	0.85
SVM	0.9994382145949395	0.73	0.84
Logistic Regression	0.9991485439277344	0.56	0.67
Extra Trees	0.9994864930105796	0.77	0.85

We gained highest accuracy of 0.9994864932417329 from KNN.

FRONT-END USING STREAMLIT:

Streamlit is an open-source app framework for Machine Learning and Data Science. Trusted by over 50% of Fortune 50 companies.

I have used it to give UI to my python script.



FUTURE ENHANCEMENTS:

While I couldn't reach out goal of 100% accuracy in fraud detection, we did end up creating a system that can, with enough time and data, get very close to that goal. As with any such project, there is some room for improvement here.

REFERENCES:

• Research Paper on Credit Card Fraud Detection using Machine Learning and Data Science published by www.ijert.org. LinkToResearchPaper

CONCLUSION:

There are many different approaches to this problem, and we get to know some algorithms in detail and especially the three models for calculating similarity that are being explained above. The maximum accuracy which we achieved was 99.94% using K-Nearest Neighbor.