Project Title: **Credit Card Fraud Detection System**

Information Security Management
CSE3502
J Component
Review - 3

Submitted by:

Hrithik Ahuja (18BCE2154)
Ayush Rana (18BCE2305)
Rakshith Sachdev (18BCI0109)
Rohan Allen (18BCI0247)

Under the Guidance of:
Prof. AMUTHA PRABHAKAR M.

TITLE: CREDIT CARD FRAUD DETECTION SYSTEM

DEMO VIDEO LINK:

https://drive.google.com/file/d/1cI3vY6sR9snj_7c0qqLlhyjkivy0Pm3P/view

TABLE OF CONTENTS:

3. BRIEF ON ISOLATION FOREST ALGORITHM

Abstract:

The main purpose of this project is understand and implement the distinct approach of Isolation forest algorithm and LOF algorithm to identify fraudulent transactions in a database instead of using generic Random Forest approach. The model will be able to identify the transactions with greater accuracy and by comparing both the approaches we will be working towards a more optimal solution. The Credit Card Fraud Detection Problem includes modelling past credit card transactions with the knowledge of the ones that turned out to be fraud. This model is then used to identify whether a new transaction is fraudulent or not. The aim of the project here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications.

INTRODUCTION:

The following are reasons why we need to develop a robust system to detect fraudulent transactions. The following challenges have to be overcome to make sure that the final product is resilient.

The challenge is to recognize fraudulent credit card transactions so that the customers of credit card companies are not charged for items that they did not purchase. Main challenges involved in credit card fraud detection are:

- Enormous Data is processed every day and the model build must be fast enough to respond to the scam in time.
- Imbalanced Data i.e. most of the transactions (99.8%) are not fraudulent which makes it really hard for detecting the fraudulent ones Data availability as the data is mostly private.
- Misclassified Data can be another major issue, as not every fraudulent transaction is caught and reported.
- Adaptive techniques used against the model by the scammers.

BRIEF ON ISOLATION FOREST ALGORITHM:

- Isolation forest is an unsupervised learning algorithm for anomaly detection that works on the principle of isolating anomalies, instead of the most common techniques of profiling normal points.
- In statistics, an anomaly (a.k.a. outlier) is an observation or event that deviates so much from other events to arouse suspicion it was generated by a different mean.
- Anomalies in a big dataset may follow very complicated patterns, which are difficult to detect "by eye" in the great majority of cases. This is the reason why the field of anomaly detection is well suited for the application of Machine Learning techniques.
- The most common techniques employed for anomaly detection are based on the construction of a profile of what is "normal": anomalies are reported as those instances in the dataset that do not conform to the normal profile.
- Isolation Forest uses a different approach: instead of trying to build a model of normal instances, it explicitly isolates anomalous points in the dataset. The main advantage of this approach is the possibility of exploiting sampling techniques to an extent that is not allowed to the profile-based methods, creating a very fast algorithm with a low memory demand

BRIEF ON LOF (Local Outlier Factor Algorithm) ALGORITHM:

The Local Outlier Factor (LOF) algorithm is an unsupervised anomaly detection method which computes the local density deviation of a given data point with respect to its neighbours. It considers as outliers the samples that have a substantially lower density than their neighbours.

The number of neighbors considered (parameter n_neighbors) is typically greater than the minimum number of samples a cluster has to contain, so that other samples can be local outliers relative to this cluster, and 2) smaller than the maximum number of close by samples that can potentially be local outliers. In practice, such informations are generally not available, and taking n_neighbors=20 appears to work well in general.

ALGORITHMS USED TO SOLVE THE PROBLEM OF ANAMOLY DETECTION UNTIL NOW:

1. Distributed Data Mining
2. Artificial Neural Networks
3. Neural Data mining
4. Hidden Markov Models
5. Bayesian neural networks
6. Computational Intelligence
7. Dempster–Shafer theory and Bayesian learning
8. Parallel granular neural networks
9. Phone usage patterns
10. Transaction aggregation methodology
11. Feature engineering

ALGORITHMS USED TO SOLVE THIS PROBLEM IN THIS PROJECT:

1. Isolation forest algorithm
2. Local Outlier Factor

LITERATURE SURVEY:

1. **Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection.** ***IEEE Intelligent Systems and Their Applications*****,** ***14*****(6)**

   Large scale data-mining techniques can improve the state of the art in commercial practice. Scalable techniques to analyze massive amounts of transaction data that efficiently compute fraud detectors in a timely manner is an important problem, especially for e-commerce. Besides scalability and efficiency, the fraud-detection task exhibits technical problems that include skewed distributions of training data and non- uniform cost per error, both of which have not been widely studied in the knowledge- discovery and data mining community. The proposed methods of combining multiple learned fraud detectors under a "cost model" are general and demonstrably useful; the empirical results demonstrate that we can significantly reduce loss due to fraud through distributed data mining of fraud models.

2. **Ghosh, S., & Reilly, D. L. (1994, January). Credit card fraud detection with a neural- network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on* (Vol. 3, pp. 621-630). IEEE**

   Using data from a credit card issuer, a neural network based fraud detection system was trained on a large sample of labelled credit card account transactions and tested on a holdout data set that consisted of all account activity over a subsequent two-month period of time. The neural network was trained on examples of fraud due to lost cards, stolen cards, application fraud, counterfeit fraud, mail-order fraud and NRI (non- received issue) fraud. The network detected significantly more fraud accounts (an order of magnitude more) with significantly fewer false positives (reduced by a factor of 20) over rule-based fraud detection procedures. We discuss the performance of the network on this data set in terms of detection accuracy and earliness of fraud detection.

3. **Raj, S. B. E., & Portia, A. A. (2011, March). Analysis on credit card fraud detection methods. In *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)* (pp. 152-156). IEEE**

   The most commonly used fraud detection methods are Neural Network (NN), rule-induction techniques, fuzzy system, decision trees, Support Vector Machines (SVM), Artificial Immune System (AIS), genetic algorithms, K-Nearest Neighbour algorithms. These techniques can be used alone or in collaboration using ensemble or meta-learning techniques to build classifiers. This paper presents a survey of various techniques used in credit card fraud detection and evaluates each methodology based on certain design criteria.

4. **Brause, R., Langsdorf, T., & Hepp, M. (1999, November). Neural data mining for credit card fraud detection. In *Proceedings 11th International Conference on Tools with Artificial Intelligence* (pp. 103-106)**

   The prevention of credit card fraud is an important application for prediction techniques. One major obstacle for using neural network training techniques is the high necessary diagnostic quality: Since only one financial transaction of a thousand is invalid no prediction success less than 99.9% is acceptable. Due to these credit card transaction proportions complete new concepts had to be developed and tested on real credit card data. This paper shows how advanced data mining techniques and neural network algorithm can be combined successfully to obtain a high fraud coverage combined with a low false alarm rate.

5. **Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. (2008). Credit card fraud detection using hidden Markov model. *IEEE Transactions on dependable and secure computing*, *5*(1)**

Due to a rapid advancement in the electronic commerce technology, the use of credit cards has dramatically increased. As credit card becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. In this paper, the sequence of operations in credit card transaction processing are modelled using a hidden Markov model (HMM) and show how it can be used for the detection of frauds. An HMM is initially trained with the normal behaviour of a cardholder. If an incoming credit card transaction is not accepted by the trained HMM with sufficiently high probability, it is considered to be fraudulent.

6. **Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002, January). Credit card fraud detection using Bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies***

This paper discusses automated credit card fraud detection by means of machine learning. In an era of digitalization, credit card fraud detection is of great importance to financial institutions. Two machine learning techniques suited for reasoning under uncertainty: artificial neural networks and Bayesian belief networks to the problem and show their significant results on real world financial data. Finally, future directions are indicated to improve both techniques and results.

7. **Aleskerov, E., Freisleben, B., & Rao, B. (1997, March). Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr)* (pp. 220- 226). IEEE**

CARDWATCH, a database mining system used for credit card fraud detection, is presented. The system is based on a neural network learning module, provides an interface to a variety of commercial databases and has a comfortable graphical user interface. Test results obtained for synthetically generated credit card data and an auto associative neural network model show very successful fraud detection rates.

8. **Quah, J. T., & Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. *Expert systems with applications*, *35*(4)**

   Online banking and e-commerce have been experiencing rapid growth over the past few years and show tremendous promise of growth even in the future. This has made it easier for fraudsters to indulge in new and abstruse ways of committing credit card fraud over the Internet. This paper focuses on real-time fraud detection and presents a new and innovative approach in understanding spending patterns to decipher potential fraud cases. It makes use of self-organization map to decipher, filter and analyze customer behaviour for detection of fraud.

9. **Ogwueleka, F. N. (2011). Data mining application in credit card fraud detection system. *Journal of Engineering Science and Technology*, *6*(3)**

   Data mining is popularly used to combat frauds because of its effectiveness. It is a well- defined procedure that takes data as input and produces models or patterns as output. Neural network, a data mining technique was used in this study. The design of the neural network (NN) architecture for the credit card detection system was based on unsupervised method, which was applied to the transactions data to generate four clusters of low, high, risky and high-risk clusters. The self-organizing map neural network (SOMNN) technique was used for solving the problem of carrying out optimal classification of each transaction into its associated group, since a prior output is unknown. The receiver-operating curve (ROC) for credit card fraud (CCF) detection watch detected over 95% of fraud cases without causing false alarms unlike other statistical models and the two-stage clusters. This shows that the performance of CCF detection watch is in agreement with other detection software, but performs better.

10. **Ogwueleka, F. N. (2011). Data mining application in credit card fraud detection system. *Journal of Engineering Science and Technology***

   The fraud detection system (FDS) consists of four components, namely, rule-based filter, Dempster–Shafer adder, transaction history database and Bayesian learner. In the rule- based component, we determine the suspicion level of each incoming transaction based on the extent of its deviation from good pattern. Dempster–Shafer's theory is used to combine multiple such evidences and an initial belief is computed. The transaction is classified as normal, abnormal or suspicious depending on this initial belief. Once a transaction is found to be suspicious, belief is further strengthened or weakened according to its similarity with fraudulent or genuine transaction history using Bayesian learning. Extensive simulation with stochastic models shows that fusion of different evidences has a very high positive impact on the performance of a credit card fraud detection system as compared to other methods.

11. **Syeda, M., Zhang, Y. Q., & Pan, Y. (2002, May). Parallel granular neural networks for fast credit card fraud detection. In *2002 IEEE World Congress on Computational Intelligence***

This paper proposes an intelligent credit card fraud detection model for detecting fraud from highly imbalanced and anonymous credit card transaction datasets. The class imbalance problem is handled by finding legal as well as fraud transaction patterns for each customer by using frequent item set mining. A matching algorithm is also proposed to find to which pattern (legal or fraud) the incoming transaction of a particular customer is closer and a decision is made accordingly. In order to handle the anonymous nature of the data, no preference is given to any of the attributes and each attribute is considered equally for finding the patterns.

12. **Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications***

Billions of dollars of loss are caused every year due to fraudulent credit card transactions. The design of efficient fraud detection algorithms is key for reducing these losses, and more and more algorithms rely on advanced machine learning techniques to assist fraud investigators. The design of fraud detection algorithms is however particularly challenging due to non stationary distribution of the data, highly imbalanced classes distributions and continuous streams of transactions. At the same time public data are scarcely available for confidentiality issues, leaving unanswered many questions about which is the best strategy to deal with them. In this paper they provided some answers from the practitioner's perspective by focusing on three crucial issues: non-balancedness, non-stationary and assessment. The analysis is made possible by a real credit card dataset provided by our industrial partner.

13. Stolfo, S., Fan, D. W., Lee, W., Prodromidis, A., & Chan, P. (1997, July). Credit card fraud detection using meta-learning: Issues and initial results.

The paper describes initial experiments using meta-learning techniques to learn models of fraudulent credit card transactions. The experiments reported here are the first step towards a better understanding of the advantages and limitations of current meta- learning strategies on real-world data. They argue that, for the fraud detection domain, fraud catching rate (True Positive rate) and false alarm rate (False Positive rate) are better metrics than the overall accuracy when evaluating the learned fraud classifiers. They showed that given a skewed distribution in the original data, artificially more balanced training data leads to better classifiers. They demonstrated how meta-learning can be used to combine different classifiers and maintain, and in some cases, improve the performance of the best classifier.

14. **Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications***

Due to the growing volume of electronic payments, the monetary strain of credit-card fraud is turning into a substantial challenge for financial institutions and service providers, thus forcing them to continuously improve their fraud detection systems. However, modern data-driven and learning-based methods, despite their popularity in other domains, only slowly find their way into business applications. In this paper, we phrase the fraud detection problem as a sequence classification task and employ Long Short-Term Memory (LSTM) networks to incorporate transaction sequences. We also integrate state-of-the-art feature aggregation strategies and report our results by means of traditional retrieval metrics. A comparison to a baseline random forest (RF) classifier showed that the LSTM improves detection accuracy on offline transactions where the card-holder is physically present at a merchant. Both the sequential and non-sequential learning approaches benefit strongly from manual feature aggregation strategies. A subsequent analysis of true positives revealed that both approaches tend to detect different frauds, which suggests a combination of the two. We conclude our study with a discussion on both practical and scientific challenges that remain unsolved.

15. **Carneiro, N., Figueira, G., & Costa, M. (2017). A data mining based system for credit- card fraud detection in e-tail. *Decision Support Systems***

Credit-card fraud leads to billions of dollars in losses for online merchants. With the development of machine learning algorithms, researchers have been finding increasingly sophisticated ways to detect fraud, but practical implementations are rarely reported. We describe the development and deployment of a fraud detection system in a large e-tail merchant. The paper explores the combination of manual and automatic classification, gives insights into the complete development process and compares different machine learning methods. The paper can thus help researchers and practitioners to design and implement data mining based systems for fraud detection or similar problems. This project has contributed not only with an automatic system, but also with insights to the fraud analysts for improving their manual revision process, which resulted in an overall superior performance.

16. **Zhu, Kevin. "Phone usage pattern as credit card fraud detection trigger.**

    A credit card fraud reduction system is disclosed. The system comprises a computer system and an application. The application, when executed on the computer system, applies increased credit card fraud prevention procedures to use of an electronic credit card application in a mobile electronic device, based on a changed communication usage pattern of the mobile electronic device.

17. **Shen, A., Tong, R., & Deng, Y. (2007, June). Application of classification models on credit card fraud detection. In *2007 International conference on service systems and service management***

    Along with the great increase in credit card transactions, credit card fraud has become increasingly rampant in recent years. This study investigates the efficacy of applying classification models to credit card fraud detection problems. Three different classification methods, i.e. decision tree, neural networks and logistic regression are tested for their applicability in fraud detections. This paper provides a useful framework to choose the best model to recognize the credit card fraud risk.

18. **Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data mining and knowledge discovery***

    The problem of pre-processing transaction data for supervised fraud classification is considered. It is impractical to present an entire series of transactions to a fraud detection system, partly because of the very high dimensionality of such data but also because of the heterogeneity of the transactions. Hence, a framework for transaction aggregation is considered and its effectiveness is evaluated against transaction-level detection, using a variety of classification methods and a realistic cost-based performance measure. These methods are applied in two case studies using real data. Transaction aggregation is found to be advantageous in many but not all circumstances. Also, the length of the aggregation period has a large impact upon performance. Aggregation seems particularly effective when a random forest is used for classification. Moreover, random forests were found to perform better than other classification methods, including SVMs, logistic regression and KNN. Aggregation also has the advantage of not requiring precisely labelled data and may be more robust to the effects of population drift.

19. **Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications***

In this paper they expanded the transaction aggregation strategy, and proposed to create a new set of features based on analyzing the periodic behaviour of the time of a transaction using the von Mises distribution. Then, using a real credit card fraud dataset provided by a large European card processing company, we compare state-of-the-art credit card fraud detection models, and evaluate how the different sets of features have an impact on the results. By including the proposed periodic features into the methods, the results show an average increase in savings of 13%.

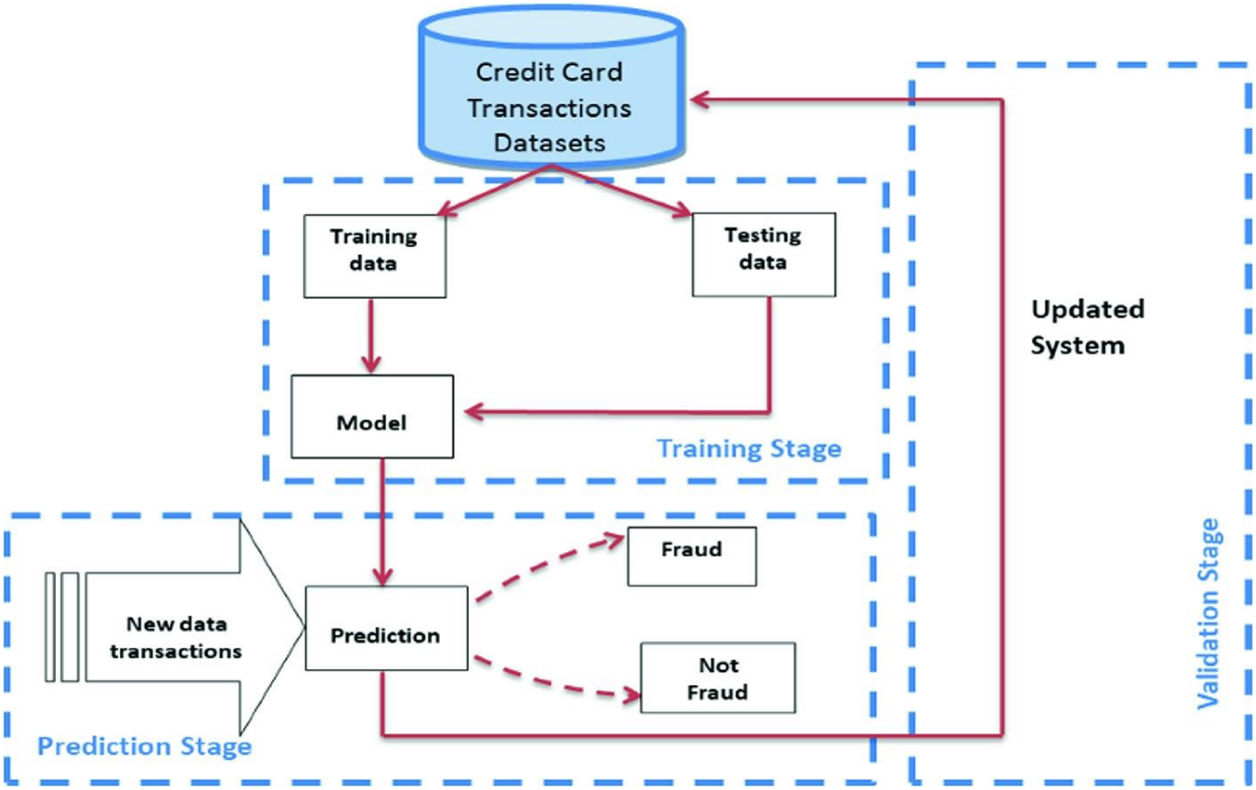
20. Patidar, R., & Sharma, L. (2011). Credit card fraud detection using neural network.

In this paper they tried to detect fraudulent transaction through the neural network along with the genetic algorithm. As we will see that artificial neural network when trained properly can work as a human brain, though it is impossible for the artificial neural network to imitate the human brain to the extent at which brain work, yet neural network and brain, depend for there working on the neurons, which is the small functional unit in brain as well as ANN. Genetic algorithm are used for making the decision about the network topology, number of hidden layers, number of nodes that will be used in the design of neural network for our problem of credit card fraud detection. For the learning purpose of artificial neural network we will use supervised learning feed forward back propagation algorithm.

EXISITING METHODOLODY

1. Heavy emphasis on neural networks
2. Tiny emphasis on pre-processing phase
3. Direct use of dataset without any cleansing
4. No use of feature scaling to more existing features more computational.

PROPOSED METHODOLOGY



NOVEL APPROACH TO BE FOLLOWED

1. Problem solving without neural networks. Though neural networks can prove effective in this domain, yet they can never solve the problem of time constraints. By using a simplified approach we will be able to produce better throughput within limited time frame.
2. Using algorithms which have not been used to solve problems in this domain until 2020, like(LOF and Isolation Forest)
3. Feature scaling to adjust features in accordance with other features.
4. Dimensionality reduction using PCA. PCA technique will be used to reduce the dimensions of the existing data which will help yielding better results.

ISOLATION FOREST METHODOLOGY

- Isolation forest is an unsupervised learning algorithm for anomaly detection that works on the principle of isolating anomalies, instead of the most common techniques of profiling normal points.
- In statistics, an anomaly (a.k.a. outlier) is an observation or event that deviates so much from other events to arouse suspicion it was generated by a different mean.
- Anomalies in a big dataset may follow very complicated patterns, which are difficult to detect "by eye" in the great majority of cases. This is the reason why the field of anomaly detection is well suited for the application of Machine Learning techniques.
- The most common techniques employed for anomaly detection are based on the construction of a profile of what is "normal": anomalies are reported as those instances in the dataset that do not conform to the normal profile.
- Isolation Forest uses a different approach: instead of trying to build a model of normal instances, it explicitly isolates anomalous points in the dataset. The main advantage of this approach is the possibility of exploiting sampling techniques to an extent that is not allowed to the profile-based methods, creating a very fast algorithm with a low memory demand

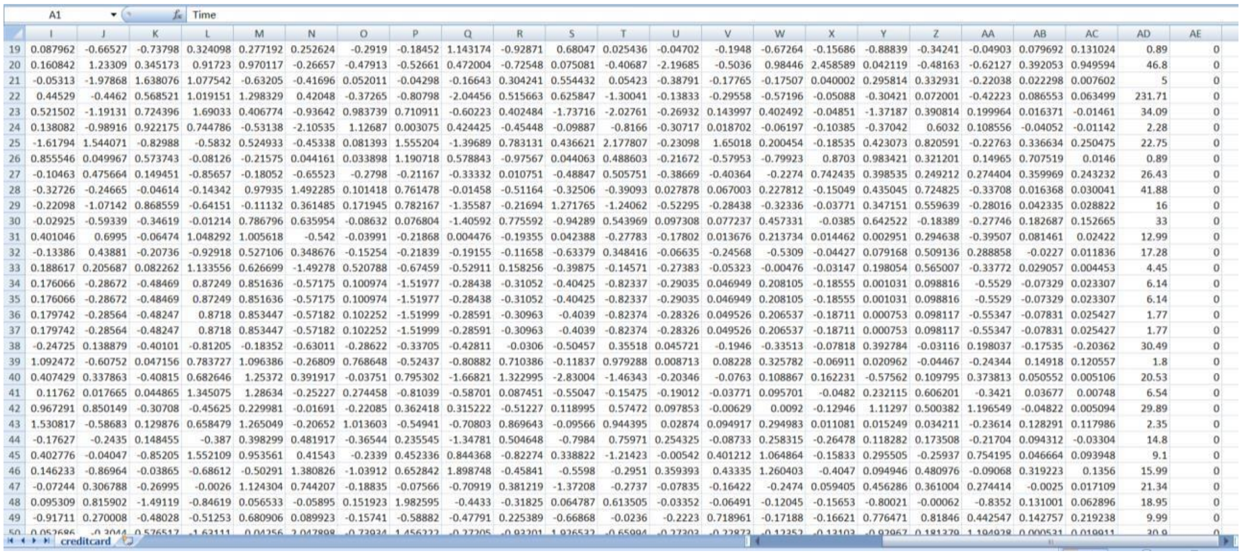
LOCAL OUTLIER FACTOR METHODOLOGY

- The local outlier factor is based on a concept of a local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density.
- By comparing the local density of an object to the local densities of its neighbors, one can identify regions of similar density, and points that have a substantially lower density than their neighbors. These are considered to be outliers.
- The local density is estimated by the typical distance at which a point can be "reached" from its neighbors.

- The definition of "reachability distance" used in LOF is an additional measure to produce more stable results within clusters.
- The "reachability distance" used by LOF has some subtle details that are often found incorrect in secondary sources.

IMPLEMENTATION:

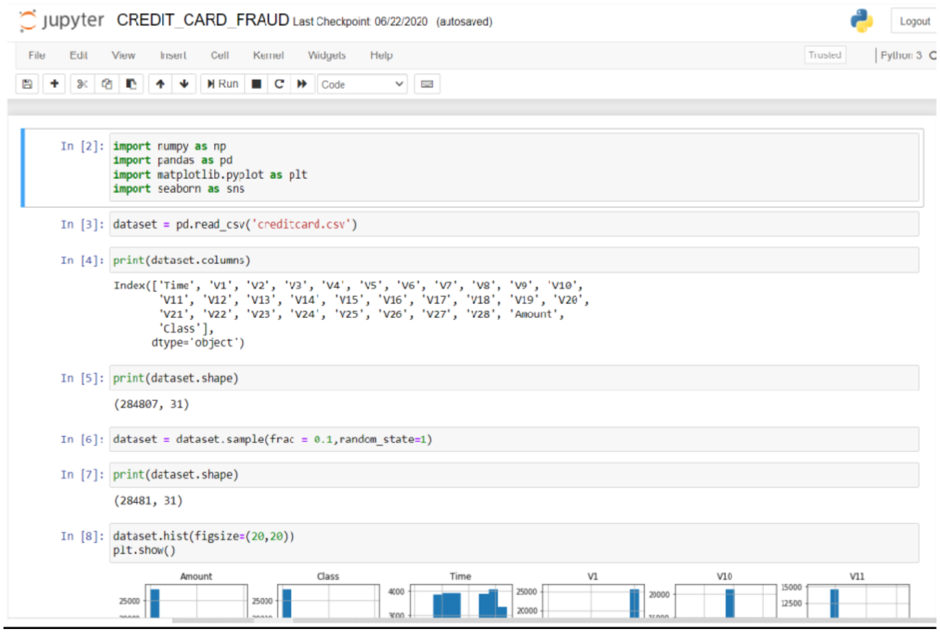
Dataset after pre-processing, feature scaling and PCA:



A1 Time

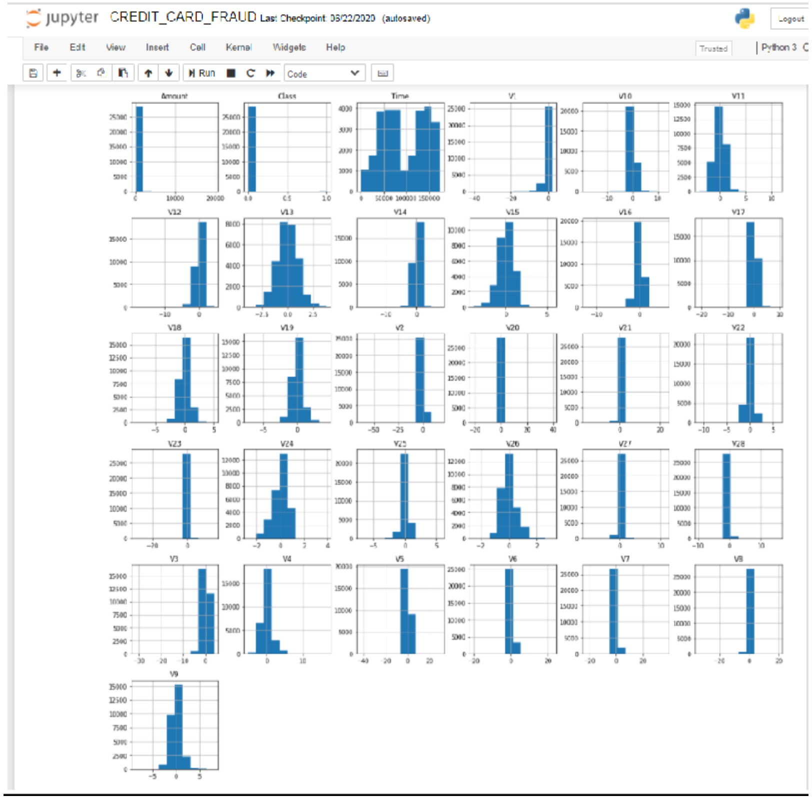
	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
19	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264	-0.15686	-0.88839	-0.34241	-0.04903	0.079692	0.131024	0.89	0
20	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446	2.458589	0.042119	-0.48163	-0.62127	0.392053	0.949594	46.8	0
21	-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	0.040002	0.295814	0.332931	-0.22038	0.022298	0.007602	5	0
22	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	-0.05088	-0.30421	0.072001	-0.42223	0.086553	0.063499	231.71	0
23	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	-0.04851	-1.37187	0.390814	0.199964	0.016371	-0.01461	34.09	0
24	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	-0.10385	-0.37042	0.6032	0.108556	-0.04052	-0.01142	2.28	0
25	-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	-0.18535	0.423073	0.820591	-0.22763	0.336634	0.250475	22.75	0
26	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	0.8703	0.983421	0.321201	0.14965	0.707519	0.0146	0.89	0
27	-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	0.742435	0.398535	0.249212	0.274404	0.359969	0.243232	26.43	0
28	-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	-0.15049	0.435045	0.724825	-0.33708	0.016368	0.030041	41.88	0
29	-0.22098	-1.07142	0.868559	-0.64151	-0.11132	0.361485	0.171945	0.782167	-1.35587	-0.21694	1.271765	-1.24062	-0.52295	-0.28438	-0.32336	-0.03771	0.347151	0.559639	-0.28016	0.042335	0.028822	16	0
30	-0.02925	-0.59339	-0.34619	-0.01214	0.786796	0.635954	-0.08632	0.076804	-1.40592	0.775592	-0.94289	0.543969	0.097308	0.077237	0.457331	-0.0385	0.642522	-0.18389	-0.27746	0.182687	0.152665	33	0
31	0.401046	0.6995	-0.06474	1.048292	1.005618	-0.542	-0.03991	-0.21868	0.004476	-0.19355	0.042388	-0.27783	-0.17802	0.013676	0.213734	0.014462	0.002951	0.294638	-0.39507	0.081461	0.02422	12.99	0
32	-0.13386	0.43881	-0.20736	-0.92918	0.527106	0.348676	-0.15254	-0.21839	-0.19155	-0.11658	-0.63379	0.348416	-0.06635	-0.24568	-0.5309	-0.04427	0.079168	0.509136	0.288858	-0.0227	0.011836	17.28	0
33	0.188617	0.205687	0.082262	1.133556	0.626699	-1.49278	0.520788	-0.67459	-0.52911	0.158256	-0.39875	-0.14571	-0.27383	-0.05323	-0.00476	-0.03147	0.198054	0.565007	-0.33772	0.029057	0.004453	4.45	0
34	0.176066	-0.28672	-0.48469	0.87249	0.851636	-0.57175	0.100974	-1.51977	-0.28438	-0.31052	-0.40425	-0.82337	-0.29035	0.046949	0.208105	-0.18555	0.001031	0.098816	-0.5529	-0.07329	0.023307	6.14	0
35	0.176066	-0.28672	-0.48469	0.87249	0.851636	-0.57175	0.100974	-1.51977	-0.28438	-0.31052	-0.40425	-0.82337	-0.29035	0.046949	0.208105	-0.18555	0.001031	0.098816	-0.5529	-0.07329	0.023307	6.14	0
36	0.179742	-0.28564	-0.48247	0.8718	0.853447	-0.57182	0.102252	-1.51999	-0.28591	-0.30963	-0.4039	-0.82374	-0.28326	0.049526	0.206537	-0.18711	0.000753	0.098117	-0.55347	-0.07831	0.025427	1.77	0
37	0.179742	-0.28564	-0.48247	0.8718	0.853447	-0.57182	0.102252	-1.51999	-0.28591	-0.30963	-0.4039	-0.82374	-0.28326	0.049526	0.206537	-0.18711	0.000753	0.098117	-0.55347	-0.07831	0.025427	1.77	0
38	-0.24725	0.138879	-0.40101	-0.81205	-0.18352	-0.63011	-0.28622	-0.33705	-0.42811	-0.0306	-0.50457	0.35518	0.045721	-0.1946	-0.33513	-0.07818	0.392784	-0.03116	0.198037	-0.17535	-0.20362	30.49	0
39	1.092472	-0.60752	0.047156	0.783727	1.096386	-0.26809	0.768648	-0.52437	-0.80882	0.710386	-0.11837	0.979288	0.008713	0.08228	0.325782	-0.06911	0.020962	-0.04467	-0.24344	0.14918	0.120557	1.8	0
40	0.407429	0.337863	-0.40815	0.682646	1.25372	0.391917	-0.03751	0.795302	-1.66821	1.322995	-2.83004	-1.46343	-0.20346	-0.0763	0.108867	0.162231	-0.57562	0.109795	0.373813	0.050552	0.005106	20.53	0
41	0.11762	0.017665	0.044865	1.345075	1.28634	-0.25227	0.274458	-0.81039	-0.58701	0.087451	-0.55047	-0.15475	-0.19012	-0.03771	0.095701	-0.0482	0.232115	0.606201	-0.3421	0.03677	0.00748	6.54	0
42	0.967291	0.850149	-0.30708	-0.45625	0.229981	-0.01691	-0.22085	0.362418	0.315222	-0.51227	0.118995	0.57472	0.097853	-0.00629	0.0092	-0.12946	1.11297	0.500382	1.196549	-0.04822	0.005094	29.89	0
43	1.530817	-0.58683	0.129876	0.658479	1.265049	-0.20652	1.013603	-0.54941	-0.70803	0.869643	-0.09566	0.944395	0.02874	0.094917	0.294983	0.011081	0.015249	0.034211	-0.23614	0.128291	0.117986	2.35	0
44	-0.17627	-0.2435	0.148455	-0.387	0.398299	0.481917	-0.36544	0.235545	-1.34781	0.504648	-0.7984	0.75971	0.254325	-0.08733	0.258315	-0.26478	0.118282	0.173508	-0.21704	0.094312	-0.03304	14.8	0
45	0.402776	-0.04047	-0.85205	1.552109	0.953561	0.41543	-0.2339	0.452336	0.844368	-0.82274	0.338822	-1.21423	-0.00542	0.401212	1.064864	-0.15833	0.295505	-0.25937	0.754195	0.046664	0.093948	9.1	0
46	0.146233	-0.86964	-0.03865	-0.68612	-0.50291	1.380826	-1.03912	0.652842	1.898748	-0.45841	-0.5598	-0.2951	0.359393	0.43335	1.260403	-0.4047	0.094946	0.480976	-0.09068	0.319223	0.1356	15.99	0
47	-0.07244	0.306788	-0.26995	-0.0026	1.124304	0.744207	-0.18835	-0.07566	-0.70919	0.381219	-1.37208	-0.2737	-0.07835	-0.16422	-0.2474	0.059405	0.456286	0.361004	0.274414	-0.0025	0.017109	21.34	0
48	0.095309	0.815902	-1.49119	-0.84619	0.056533	-0.05895	0.151923	1.982595	-0.4433	-0.31825	0.064787	0.613505	-0.03352	-0.06491	-0.12045	-0.15653	-0.80021	-0.00062	-0.8352	0.131001	0.062896	18.95	0
49	-0.91711	0.270008	-0.48028	-0.51253	0.680906	0.089923	-0.15741	-0.58882	-0.47791	0.225389	-0.66868	-0.0236	-0.2223	0.718961	-0.17188	-0.16621	0.776471	0.81846	0.442547	0.142757	0.219238	9.99	0
50	0.052686	-0.3044	0.576517	-1.63111	0.04256	2.047898	-0.73934	1.456222	-0.27205	-0.93201	1.926532	-0.65994	-0.27303	-0.22873	-0.12352	-0.13103	-0.92967	0.181379	1.194928	0.000531	0.019911	30.9	0

CODE :

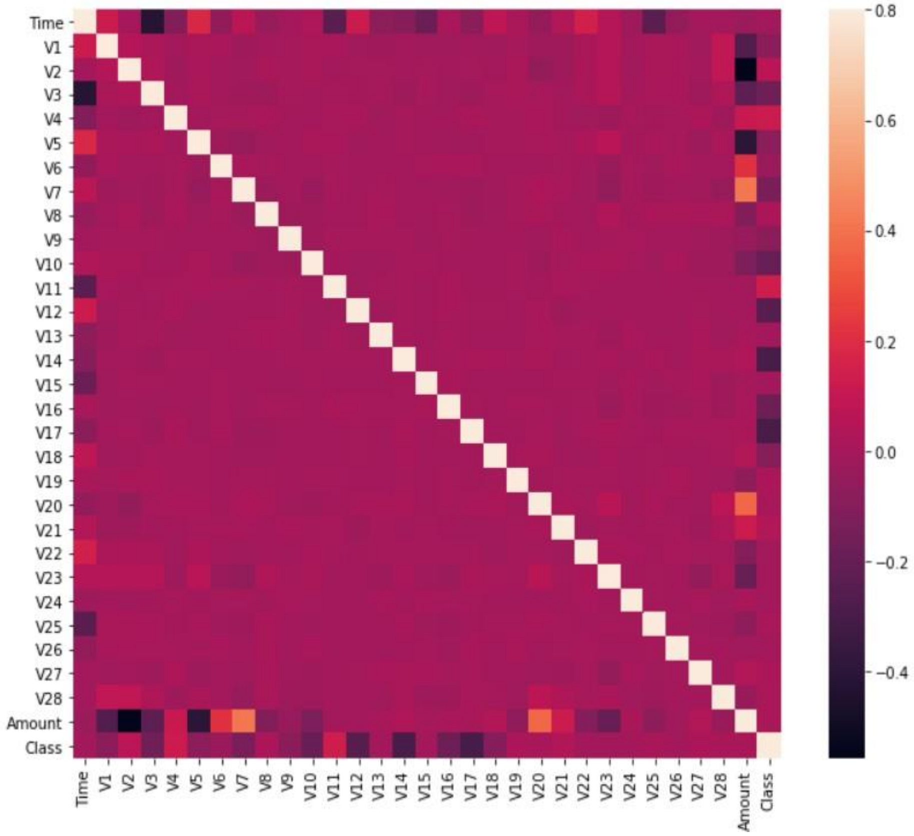


```
In [2]: import rumpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns

In [3]: dataset = pd.read_csv('creditcard.csv')

In [4]: print(dataset.columns)

        Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
               'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
               'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
               'Class'],
              dtype='object')

In [5]: print(dataset.shape)

        (284807, 31)

In [6]: dataset = dataset.sample(frac = 0.1,random_state=1)

In [7]: print(dataset.shape)

        (28481, 31)

In [8]: dataset.hist(figsize=(20,20))
        plt.show()
```

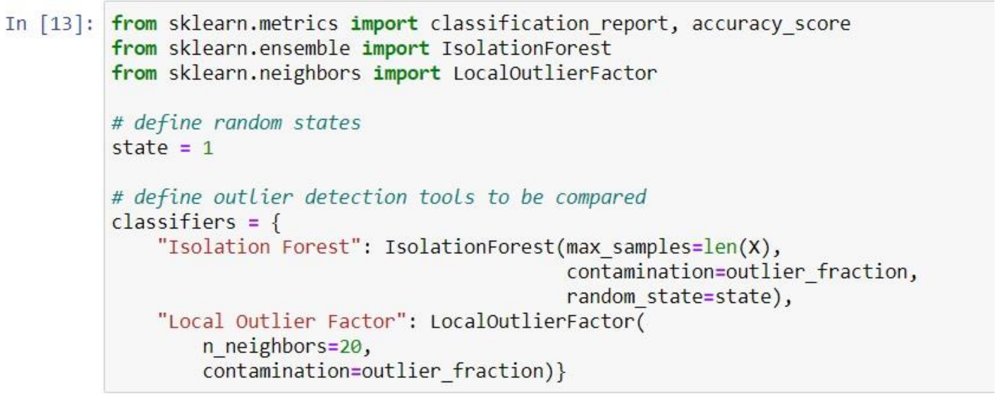
ANALYSIS OF INDEPENDENT VARIABLES:



Generating HeatMap to determine correlation between different columns in the dataset:

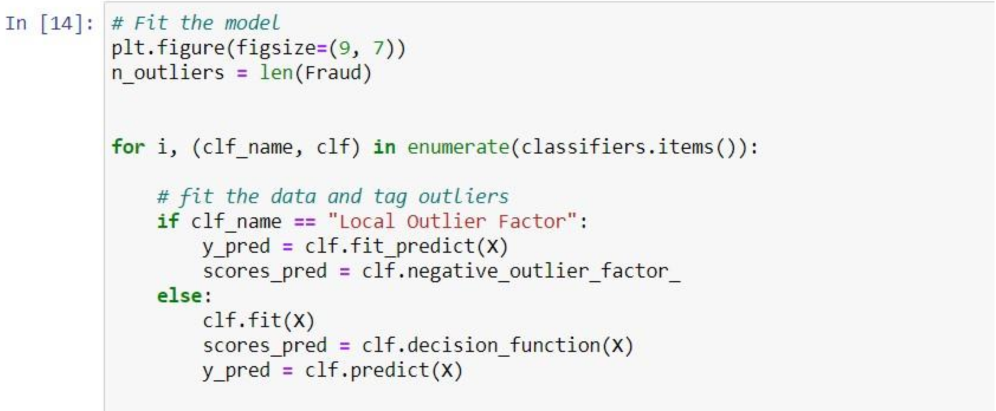

Time
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23
V24
V25
V26
V27
V28
Amount
Class
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

CORE – IMPLEMENTATION:



In [13]:

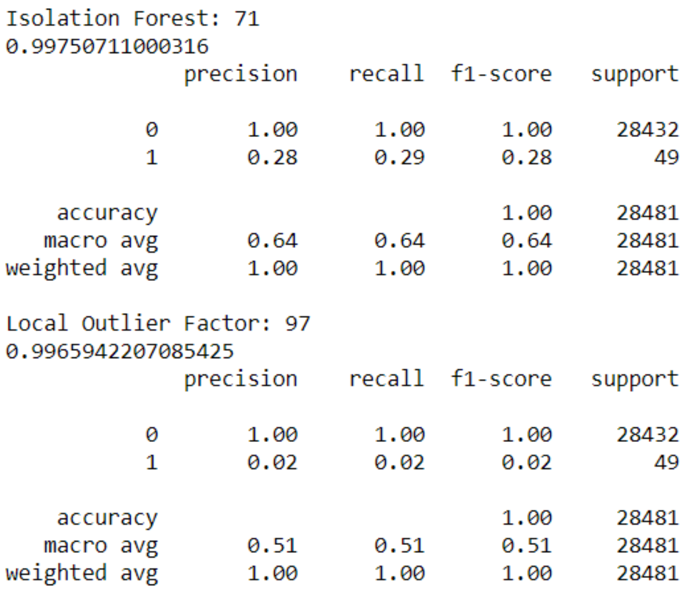
```
from sklearn.metrics import classification_report, accuracy_score
from sklearn.ensemble import IsolationForest
from sklearn.neighbors import LocalOutlierFactor

# define random states
state = 1

# define outlier detection tools to be compared
classifiers = {
    "Isolation Forest": IsolationForest(max_samples=len(X),
                                        contamination=outlier_fraction,
                                        random_state=state),
    "Local Outlier Factor": LocalOutlierFactor(
        n_neighbors=20,
        contamination=outlier_fraction)}
```



In [14]:

```
# Fit the model
plt.figure(figsize=(9, 7))
n_outliers = len(Fraud)


for i, (clf_name, clf) in enumerate(classifiers.items()):

    # fit the data and tag outliers
    if clf_name == "Local Outlier Factor":
        y_pred = clf.fit_predict(X)
        scores_pred = clf.negative_outlier_factor_
    else:
        clf.fit(X)
        scores_pred = clf.decision_function(X)
        y_pred = clf.predict(X)
```

EVALUATION METRICS:



```
Isolation Forest: 71
0.99750711000316
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     28432
           1       0.28      0.29      0.28        49

    accuracy                           1.00     28481
   macro avg       0.64      0.64      0.64     28481
weighted avg       1.00      1.00      1.00     28481

Local Outlier Factor: 97
0.9965942207085425
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     28432
           1       0.02      0.02      0.02        49

    accuracy                           1.00     28481
   macro avg       0.51      0.51      0.51     28481
weighted avg       1.00      1.00      1.00     28481
```

PROJECT OUTCOME:

- The model is simple and fast enough to detect the anomaly and classify it as a fraudulent transaction as quickly as possible.
- Imbalance can be dealt with by properly using some Isolation forest algorithm and Local Outlier factor (LOF).
- For protecting the privacy of the user the dimensionality of the data will be reduced. A more trustworthy source will be taken which double-checks the data, at least for training the model.
- The model is simple and interpretable so that when the scammer adapts to it with just some tweaks, we can have a new model up and running to deploy.