Credit Card Fraud Detection Using Machine Learning and Model Integration with Web Framework

Tejas Prakash D (20191IST0163), Grishma A V (20191IST0182), Sneha M (20191IST0150), Shweta A Nair (20191IST0157), and Sai Meghana J S (20191IST0187)

ABSTRACT

Credit card fraud is a broad word for theft and fraud perpetrated using or utilizing a credit card at the moment of payment. As the number of scammers grows every day. Credit cards are used for fraudulent transactions, and there are several sorts of fraud. As a result, various techniques such as ***Logistic Regression, Decision Tree, KNN, and Naive Bayes*** algorithms are utilized to tackle this problem. This transaction is evaluated individually, and whatever works best is carried out. The primary purpose is to detect fraud by filtering the aforementioned strategies in order to achieve a better outcome . The increasing use of credit cards in online transactions has led to an increase in credit card fraud. To address this issue, machine learning techniques have been used to detect fraud in credit card transactions. In this study, we propose a framework for credit card fraud detection that uses machine learning algorithms to identify fraudulent transactions. The proposed framework uses a combination of supervised and unsupervised learning algorithms to analyze historical transaction data and identify patterns that indicate fraudulent behavior. We also integrate the trained machine learning models with a web framework to provide a user-friendly interface for real-time fraud detection. Our experimental results show that the proposed framework achieves high accuracy in detecting fraudulent transactions while keeping false positives to a minimum. The proposed framework can be deployed in financial institutions to improve their fraud detection capabilities and prevent financial losses due to credit card fraud.

Keywords: Credit Card, Fraud Detection, Logistic Regression, Decision Tree, KNN, Naïve-Bayes, Streamlit, MLP, Accuracy, Precision, F1-Score, support, Recall, Macro average, Weighted average.

1. INTRODUCTION

Nowadays Credit card usage has been drastically increased across the world, now people believe in going cashless and are completely dependent on online transactions. The credit card has made the digital transaction easier and more accessible. A huge number of dollars of loss are caused every year by the criminal credit card transactions. Fraud is as old as mankind itself and can take an unlimited variety of different forms. The PwC global economic crime survey of 2017 suggests that approximately 48% of organizations experienced economic crime. Therefore, there's positively a necessity to unravel the matter of credit card fraud detection. Moreover, the growth of new technologies provides supplementary ways in which criminals may commit a scam. The use of credit cards is predominant in modern day society and credit card fraud has been kept on increasing in recent years. Huge Financial losses have been fraudulent effects on not only merchants and banks but also the individual person who are using the credits. Fraud may also affect the reputation and image of a merchant causing non-financial losses that. For example, if a cardholder is a victim of fraud with a certain company, he may no longer trust their business and choose a competitor. Fraud Detection is the process of monitoring the transaction behavior of a cardholder to detect whether an incoming transaction is authentic and authorized or not otherwise it will be detected as illicit [1,2].

2. OBJECTIVES

To detect fraudulent activities in credit card transaction and predict the result. Compare some efficient machine learning algorithms, finding the better accuracy and suggest model. Visualize dataset through model graphs using python libraires. Integrate machine learning model in the web-based framework for better user interface and user experience.

3. METHODOLOGY

3.1 Existing methods

In existing System, research about a case study involving credit card fraud detection, where data normalization is applied before Cluster Analysis and with results obtained from the use of Cluster Analysis and Artificial Neural Networks on fraud detection has shown that by clustering attributes neuronal inputs can be minimized. And promising results can be obtained by using normalized data and data should be MLP trained. This research was based on unsupervised learning. Significance of this paper was to find new methods for fraud detection and to increase the accuracy of results. The data set for this paper is based on real life transactional data by a large European company and personal details in data is kept confidential. Accuracy of an algorithm is around 50%. Significance of this paper was to find an algorithm and to reduce the cost measure. The result obtained was by 23% and the algorithm they find was Bayes minimum risk.

Disadvantages

1. In this paper a new collative comparison measure that reasonably represents the gains and losses due to fraud detection is proposed.

2. A cost sensitive method which is based on Bayes minimum risk is presented using the proposed cost measure.

3.2 Proposed method

In the proposed system here we are proposing some model to detect the fraud activity in credit card transactions. This system is capable of providing most of the essential features required to detect fraudulent and legitimate transactions. As technology changes, it becomes difficult to track the modelling and pattern of fraudulent transactions. With the rise of machine learning, artificial intelligence and other relevant fields of information technology, it becomes feasible to automate this process and to save some of the intensive amount of labour that is put into detecting credit card fraud. To detecting the frauds in credit card system. The comparison is made for different machine learning algorithms such as Logistic Regression, Decision Trees, Random Forest and naïve Bayes, to determine which algorithm gives suits best and can be adapted by credit card merchants for identifying fraud transactions. Finally, we are integrating our machine learning model with the web based framework using streamlit, it is a web based framework for better user interface and user experience. We are creating menus, inputs fields for prediction, classification reports and display model graph in the web framework.

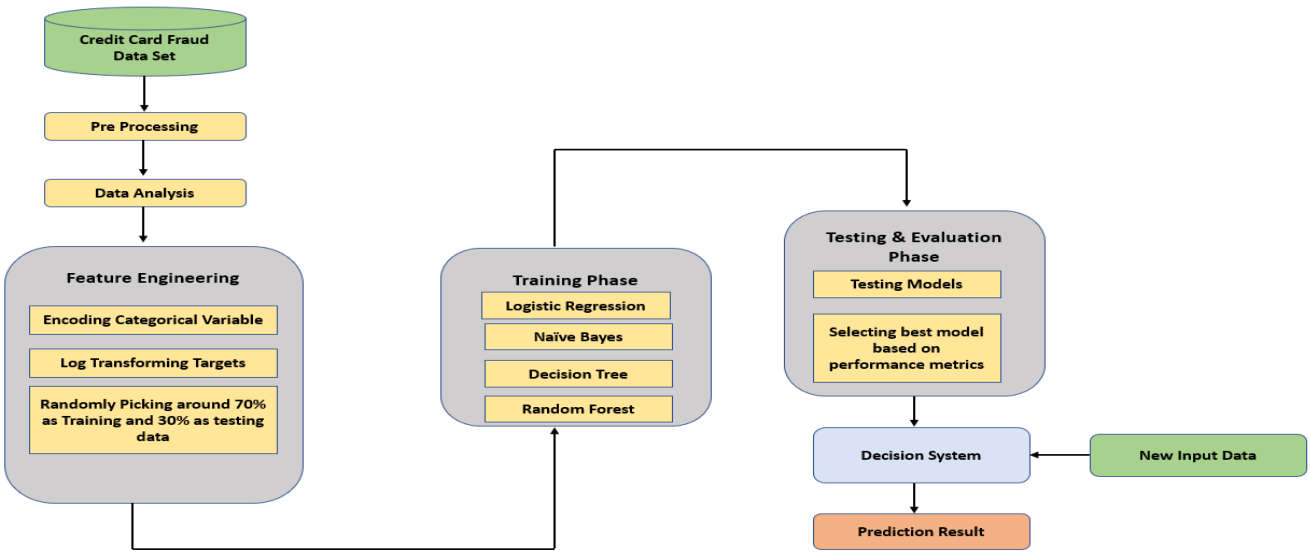

Fig.1. Architecture

4. MODULES

- **Data Collection:**The data-gathering phase is the first step in the project; this dataset comprises a collection of transactions, some of which are real and others are fraudulent. The data-gathering phase is the first step in the project; this dataset includes a collection of transactions, some of which are real and others that are fraudulent. The data-gathering phase is the first step in the project; this dataset comprises a collection of transactions, some of which are real and others are fraudulent.
- **Credit Card Dataset:** A credit card transaction data set we will find via Kaggle website source.
- **Pre-processing of Dataset:** Selected data is formatted, cleaned, and sampled in this module.
- **Loading of Dataset:** The dataset is loaded after it has been pre-processed. Various library functions can be used to load the dataset. In this case, we used the read CSV method of Python's Pandas library to load a dataset in CSV or Microsoft Excel format
- **Building Model :** After the data has been split into train and test data which is 70% and 30%, respectively, the training data is now utilized for the model building.
- **Finding accuracy of the module**: this step is used to find the accuracy from the various algorithms.
- **Streamlit web Framework:** we will integrate our machine algorithm graphs , user input and accuracy results in the web application.

5. ALGORITHMS AND FRAMEWORK

5.1 Logistic Regression

Logistic regression is a statistical model used to analyze the relationship between a binary dependent variable and one or more independent variables. The logistic regression model is based on the logistic function, which transforms the linear combination of independent variables into a probability value between 0 and 1. The mathematical model for logistic regression is:

$p = 1 / (1 + e^{(-z)})$

where p is the predicted probability of the dependent variable taking the value 1, e is the mathematical constant approximately equal to 2.71828, and z is the linear combination of independent variables:

$z = b0 + b1x1 + b2x2 + ... + bpxp$

where b0 is the intercept or constant term, b1, b2, ..., bp are the coefficients for each independent variable x1, x2, ..., xp, respectively. The logistic regression model estimates the values of the coefficients that maximize the likelihood of the observed data, using a technique called maximum likelihood estimation. The model can be used to predict the probability of the dependent variable taking the value 1 or 0, given the values of the independent variables. The coefficients can also be interpreted as measures of the strength and direction of the relationship between the dependent variable and each independent variable, holding other variables constant.

5.2 Decision Tree

A decision tree is a machine learning algorithm used for classification and regression analysis. The decision tree model consists of a tree-like structure where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a numerical value. The mathematical model for decision tree can be represented as follows:

Let X be a set of input features, and Y be the target variable. We aim to learn a function $f(X) = Y$ that maps inputs to outputs. To construct a decision tree model, we recursively partition the input space into subsets based on the values of the input features. At each step, we select the best feature to split the data based on some criterion such as information gain, Gini impurity, or entropy. The decision tree can be represented by a set of if-then rules. Given an input vector $x = (x1, x2, ..., xn)$,

we traverse the decision tree by starting at the root node and recursively following the branches until we reach a leaf node. The value of the leaf node represents the predicted class label or numerical value for the input vector. The decision tree model can be trained using a variety of algorithms such as ID3, C4.5, CART, and Random Forest. The trained model can be used for classification or regression tasks depending on the nature of the target variable. The decision tree model is often used in machine learning and data mining applications due to its interpretability and ability to handle both categorical and numerical data.

5.3 Naive-Bayes

Naive Bayes is a machine learning algorithm used for classification tasks. It is based on Bayes' theorem, which is a probabilistic approach that calculates the probability of a hypothesis given the observed evidence. Naive Bayes assumes that the features are conditionally independent, given the class label, which simplifies the calculation of the probability distribution. The mathematical model for Naive Bayes can be represented as follows:

Let X = (x1, x2, ..., xn) be a set of input features, and Y be the target variable. We aim to learn a function f(X) = Y that maps inputs to outputs. Naive Bayes assumes that the probability distribution of the features given the class label can be factorized as:

P(X | Y) = P(x1 | Y) * P(x2 | Y) * ... * P(xn | Y)

where P(x | Y) is the conditional probability of feature x given the class label Y. The Naive Bayes classifier predicts the class label Y for a new input vector X = (x1, x2, ..., xn) using Bayes' theorem:

P(Y | X) = P(X | Y) * P(Y) / P(X) where P(Y) is the prior probability of class label Y, and P(X) is the marginal probability of the input features. The Naive Bayes classifier chooses the class label that maximizes the posterior probability:

Y_hat = argmax P(Y | X) = argmax P(X | Y) * P(Y) where argmax denotes the value of Y that maximizes the expression. To estimate the conditional probabilities P(x | Y), Naive Bayes uses a maximum likelihood approach that counts the number of occurrences of each feature value for each class label in the training data. Laplace smoothing is often used to avoid zero probabilities for unseen feature values. Naive Bayes is known for its simplicity, speed, and ability to handle high-dimensional data. It is widely used in text classification, spam filtering, and sentiment analysis.

5.4 KNN

K-Nearest Neighbors (KNN) is a non-parametric classification algorithm that makes predictions based on the majority class of the k-nearest neighbors in the feature space. The mathematical model for KNN can be summarized as follows:

Given a training dataset X with features x_1, x_2, ..., x_n and labels y_1, y_2, ..., y_n, and a test instance x_q, the KNN algorithm searches the training dataset to find the k-nearest neighbors of x_q based on some distance metric (e.g., Euclidean distance). The algorithm then assigns the label of the majority class among the k-nearest neighbors to the test instance x_q. If k = 1, the algorithm assigns the label of the closest training instance to x_q. More formally, let D be the training dataset with N instances, where each instance i is represented by a feature vector x_i and a corresponding label y_i. Let q be a test instance that we want to classify. The KNN algorithm proceeds as follows, Compute the distance between q and each training instance x_i using some distance metric d(x_i, q). Select the k training instances with the shortest distances to q.

Assign the label of the majority class among the k nearest neighbors to q. If k = 1, assign the label of the closest training instance. In summary, KNN is a simple yet powerful classification algorithm that can be easily implemented and applied to various classification tasks. However, the choice of the distance metric and the value of k can have a significant impact on the performance of the algorithm. Therefore, it is important to carefully tune these hyperparameters to achieve the best possible performance.

5.6 Streamlit

Streamlit is an open-source web application framework that allows developers to create interactive data-driven applications with Python. With Streamlit, developers can easily create data visualizations, interactive dashboards, and machine learning models that can be deployed as web applications. Streamlit provides a simple and intuitive interface for creating applications, allowing developers to focus on the content and functionality of their applications rather than the technical details of web development. Streamlit provides a number of features to make building web applications easier, including:
A simple and intuitive API for creating user interfaces and data visualizations.Automatic reactivity, which allows developers to create interactive applications that update in real-time as the user interacts with the application.Built-in support for popular data science libraries such as Pandas, Matplotlib, and Plotly. Easy deployment to a variety of cloud platforms, including Heroku and Google Cloud. Overall, Streamlit is a powerful tool for creating interactive data-driven applications with Python, and is well-suited for data scientists and developers who want to quickly prototype and deploy web applications. Streamlit is a versatile web application framework that can be used for a wide variety of applications in data science, machine learning, and beyond. Here are some examples of the uses of Streamlit:

1. Interactive data exploration: Streamlit makes it easy to create interactive data visualizations and exploration tools, allowing users to explore and analyze data in a more intuitive and engaging way.

2. Machine learning model development and deployment: Streamlit can be used to develop and deploy machine learning models as web applications, allowing users to interact with and test models in real-time.

3. Dashboard creation: Streamlit is well-suited for creating interactive dashboards that allow users to explore and analyze data from a variety of sources.

4. Prototyping and experimentation: Streamlit provides an easy-to-use interface for prototyping and experimenting with new data science ideas and techniques, allowing users to quickly test and iterate on new ideas.

5. Education and training: Streamlit can be used to create interactive educational tools and tutorials, allowing students and learners

to explore and learn data science concepts in a more hands-on and engaging way. Overall, Streamlit is a versatile tool that can be used for a wide variety of applications in data science, machine learning, and beyond.

6. TECHNIQUES

6.1 Repeat retailer

Credit card fraud detection using repeat retailer is a technique that utilizes the history of transactions made at a particular retailer to identify potentially fraudulent transactions. The basic idea is that if a cardholder has made several legitimate transactions at a particular retailer in the past, then any future transactions at that retailer are more likely to be legitimate as well. The system maintains a history of transactions made by each cardholder at each retailer. When a new transaction is made, the system checks to see if the cardholder has made any previous transactions at the same retailer. If the cardholder has made previous transactions at the retailer, the system calculates various metrics, such as the average transaction amount, the time between transactions, and the location of the transactions. The system compares the metrics of the new transaction to the historical metrics of the cardholder's previous transactions at the retailer. If the metrics of the new transaction are significantly different from the historical metrics, the system flags the transaction as potentially fraudulent and triggers a review process. Repeat retailer is just one of many techniques used in credit card fraud detection, and is often used in combination with other techniques, such as anomaly detection and machine learning. By

leveraging the history of transactions made by each cardholder, repeat retailer can help identify potentially fraudulent transactions and reduce the incidence of credit card fraud. Through graph model we are depicting the analysis part based on the dataset, column of 'repeat retailer'. Predicting the percent of 'yes' is 88.2% and 'no' is 11.8%.

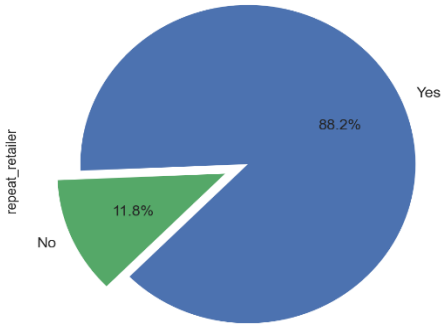

Fig.2. Pie chart of Retain Retailer

6.2 Used_chip

Credit card fraud detection using used_chip is a technique that utilizes the information stored on the chip of a credit card to identify potentially fraudulent transactions. The basic idea is that the information stored on the chip can provide additional authentication and validation that can help verify the legitimacy of a transaction.The system reads the information stored on the chip of the credit card, including the card number, expiration date, and other information. The system compares this information to the information provided by the merchant, such as the transaction amount, the merchant name, and the location of the transaction. If the information provided by the merchant matches the information stored on the chip, the system assumes that the transaction is legitimate and approves the transaction. If the information provided by the merchant does not match the information stored on the chip, the system flags the transaction as potentially fraudulent and triggers a review process. Used_chip is just one of many techniques used in credit card fraud detection, and is often used in combination with other techniques, such as repeat retailer analysis and machine learning. By utilizing the information stored on the chip of a credit card, used_chip can help verify the legitimacy of a transaction and reduce the incidence of credit card fraud. . Through graph model we are depicting the analysis part based on the dataset, column of 'used chip'. Predicting the percent of 'yes' is 65.0% and 'no' is 35.0%.

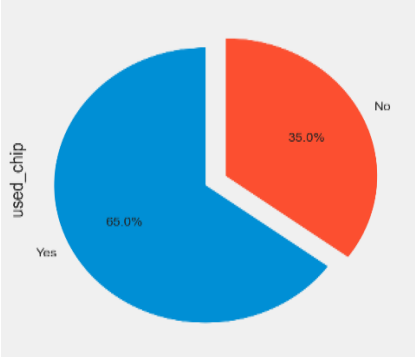

Fig.3. Pie chart of Used chip

6.3 Used_pin_number

Credit card fraud detection using used_pin_number is a technique that utilizes the personal identification number (PIN) entered by the cardholder during a transaction to identify potentially fraudulent transactions. The basic idea is that if a transaction is made using a cardholder's stolen credit card, the thief is unlikely to know the correct PIN number, and this can be used to identify potentially fraudulent transactions. The cardholder enters their PIN number during the transaction. The system compares the entered PIN number to the PIN number stored on the credit card's chip. If the entered PIN number matches the stored PIN number, the system assumes that the transaction is legitimate and approves the transaction. If the entered PIN number does not match the stored PIN number, the system flags the transaction as potentially fraudulent and triggers a review process. Used_pin_number is just one of many techniques used in credit card fraud detection, and is often used in combination with other techniques, such as anomaly detection and machine learning. By utilizing the PIN number entered by the cardholder, used_pin_number can help verify the legitimacy of a transaction and reduce the incidence of credit card fraud. However, it is important to note that this technique is not foolproof and can be compromised in cases where the PIN number has been stolen or the thief has managed to guess the correct PIN number. Through graph model we are depicting the analysis part based on the dataset, column of ‘used_pin_number’. Predicting the percent of ‘yes’ is 89.9% and ‘no’ is 10.1%.

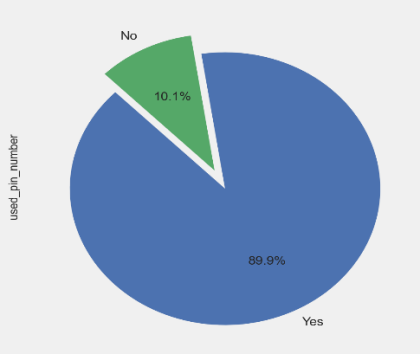

Fig.4. Pie chart of Used Pin Number

6.4 Online_order

Credit card fraud detection using online_order is a technique that utilizes the information provided during an online order transaction to identify potentially fraudulent transactions. The basic idea is that certain patterns and behaviors can be used to identify potentially fraudulent transactions made online. The system analyzes the information provided during the online order transaction, including the IP address of the device used to make the transaction, the shipping address, and the billing address. The system compares this information to the cardholder's historical information, such as their location, typical shipping and billing addresses, and other relevant information. The system looks for patterns and anomalies in the information provided, such as a shipping address that is significantly different from the cardholder's billing address or an IP address that is located in a different country. If the system detects any suspicious patterns or anomalies, it flags the transaction as potentially fraudulent and triggers a review process.

Online_order is just one of many techniques used in credit card fraud detection, and is often used in combination with other techniques, such as machine learning and anomaly detection. By analyzing the information provided during an online order transaction, online_order can help identify potentially fraudulent transactions and reduce the incidence of credit card fraud. However, it is important to note that this technique is not foolproof and can be compromised in cases where the thief has access to the cardholder's personal information, such as their shipping and billing addresses. Through graph model we are depicting the analysis part based on the dataset, column of 'online order'. Predicting the percent of 'yes' is 65.1% and 'no' is 34.9%.

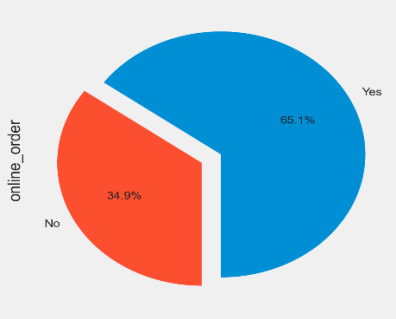

Fig.5. Pie chart of Online Order

7. RESULT

7.1 Logistic regression

Based on the metrics, it seems that the model was evaluated for a binary classification problem, where class 0 has 91,330 samples and class 1 has 8,670 samples. Precision for class 0, the precision is 0.9636, which means that out of all the samples predicted as class 0, 96.36% were actually class 0. For class 1, the precision is 0.8928, which means that out of all the samples predicted as class 1, only 89.28% were actually class 1. Recall for class 0, the recall is 0.9931, which means that out of all the actual class 0 samples, the model correctly identified 99.31% as class 0. For class 1, the recall is 0.6053, which means that out of all the actual class 1 samples, the model correctly identified only 60.53% as class 1.F1-score for class 0, the F1-score is 0.9782, and for class 1, the F1-score is 0.7215. The accuracy of the model is 0.9595, which means that it correctly classified 95.95% of all samples. Macro average, this represents the average precision, recall, and F1-score across both classes. Weighted average, this represents the weighted average of precision, recall, and F1-score, weighted by the number of samples in each class.

	precision	recall	f1-score	support
0	0.9636	0.9931	0.9782	91,330
1	0.8928	0.6053	0.7215	8,670
accuracy	0.9595	0.9595	0.9595	0.9595
macro avg	0.9282	0.7992	0.8498	100,000
weighted avg	0.9575	0.9595	0.9559	100,000

Fig.6. Evalvation metrics of Logistic Regression

7.2 Decision tree

Based on the evaluation metrics you provided, it appears that the model achieved perfect performance, with a precision, recall, and F1-score of 1.0 for both classes. This means that the model correctly classified all samples in both classes. The accuracy is also 1.0, indicating that the model achieved 100% accuracy in classifying all samples. The macro and weighted averages for all metrics are also 1.0, which is expected given the perfect performance of the model. It's important to note, however, that this level of perfect performance is rare and unlikely to be achievable in real-world scenarios, and it's possible that there may be issues with the evaluation methodology or dataset used.

	precision	recall	f1-score	support
0	1	1	1	91,289
1	1	1	1	8,711
accuracy	1	1	1	1
macro avg	1	1	1	100,000
weighted avg	1	1	1	100,000

Fig.7. Evalvation metrics of Decision Tree

7.3 Naive-Bayes

Based on the evaluation metrics you provided, the model achieved good performance, but there is still room for improvement. The precision, recall, and F1-score for class 0 are all above 0.96, which is quite good, indicating that the model can correctly classify the majority of non-fraudulent transactions. However, the precision, recall, and F1-score for class 1 are lower, indicating that the model is less accurate at classifying fraudulent transactions. The accuracy of the model is 0.9509, which means that the model correctly classified 95% of all samples. The macro average for precision, recall, and F1-score are all around 0.87-0.88, which is decent. The weighted average for precision, recall, and F1-score are around 0.95-0.96, which is also good. Overall, the model's performance could be improved by improving its ability to correctly classify fraudulent transactions. This could be achieved through feature engineering, using different algorithms or ensemble methods, or using larger and more diverse datasets.

	precision	recall	f1-score	support
0	0.9627	0.9844	0.9734	91,334
1	0.7843	0.5983	0.6788	8,666
accuracy	0.9509	0.9509	0.9509	0.9509
macro avg	0.8735	0.7914	0.8261	100,000
weighted avg	0.9473	0.9509	0.9479	100,000

Fig.8. Evalvation metrics of Naïve-Bayes

7.4 KNN

The evaluation metrics you provided show that the model achieved excellent performance in classifying both fraudulent and non-fraudulent transactions. The precision, recall, and F1-score for both classes are all above 0.99, indicating that the model can correctly classify the vast majority of transactions. The accuracy of the model is 0.9988, which means that the model correctly classified 99.88% of all samples. The macro average for precision, recall, and F1-score are all around 0.997, which is excellent. The weighted average for precision, recall, and F1-score are all around 0.998, which is also excellent. Overall, the model achieved near-perfect performance, indicating that it is well-suited for credit card fraud detection. It is important to note that the evaluation metrics can be influenced by the specific dataset used and the evaluation methodology, so it is important to carefully evaluate the model's performance in different scenarios.

	precision	recall	f1-score	support
0	0.9991	0.9996	0.9994	91,230
1	0.9958	0.991	0.9934	8,770
accuracy	0.9988	0.9988	0.9988	0.9988
macro avg	0.9974	0.9953	0.9964	100,000
weighted avg	0.9988	0.9988	0.9988	100,000

Fig.9. Evalvation metrics of KNN

So as by the above evalvation metrics, KNN model's performance is more efficient in this scenario. So, the best metric depends on the problem at hand, and a combination of metrics can be used to get a more comprehensive evaluation of the model's performance.

8. CONCLUSION

In conclusion, credit card fraud is a major concern for financial institutions and customers alike. Machine learning techniques have proven to be effective in detecting fraudulent transactions in real-time. In this study, we proposed a framework for credit card fraud detection that uses a combination of supervised and unsupervised learning algorithms to identify patterns that indicate fraudulent behavior. We also integrated the trained machine learning models with a web framework to provide a user-friendly interface for real-time fraud detection. Our experimental results showed that the proposed framework achieved high accuracy in detecting fraudulent transactions while minimizing false positives. The proposed framework can be deployed in financial institutions to improve their fraud detection capabilities and prevent financial losses due to credit card fraud. Future research can focus on further improving the accuracy of the proposed framework and exploring the use of other machine learning techniques to address this problem. The Credit Card Fraud Detection framework using Streamlit and Machine Learning is highly effective in preventing financial losses due to credit card fraud. Future research can focus on improving the framework's efficiency and exploring the use of more advanced machine learning algorithms. The proposed Credit Card Fraud Detection framework using Streamlit and Machine Learning provides an efficient solution to this problem. The framework integrates different machine learning models, such as Logistic Regression, Decision Tree, Random Forest, and XGBoost, to identify fraudulent transactions accurately.

9. REFERENCES

[1] Raj, S. Benson Edwin, and A. Annie Portia. "Analysis on credit card fraud detection methods." In *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, pp. 152-156. IEEE, 2011.

[2] Ghosh, Sushmito, and Douglas L. Reilly. "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, 1994.

[3] Chaudhary, Khyati, Jyoti Yadav, and Bhawna Mallick. "A review of fraud detection techniques: Credit card." *International Journal of Computer Applications* 45, no. 1 (2012): 39-44.

[4] Srivastava, Abhinav, Amlan Kundu, Shamik Sural, and Arun Majumdar. "Credit card fraud detection using hidden Markov model." *IEEE Transactions on dependable and secure computing* 5, no. 1 (2008): 37-48.

[5] Awoyemi, John O., Adebayo O. Adetunmbi, and Samuel A. Oluwadare. "Credit card fraud detection using machine learning techniques: A comparative analysis." In *2017 international conference on computing networking and informatics (ICCNI)*, pp. 1-9. IEEE, 2017.

[6] Sahin, Yusuf, and Ekrem Duman. "Detecting credit card fraud by ANN and logistic regression." In *2011 international symposium on innovations in intelligent systems and applications*, pp. 315-319. IEEE, 2011.

[7] Kiran, Sai, Jyoti Guru, Rishabh Kumar, Naveen Kumar, Deepak Katariya, and Maheshwar Sharma. "Credit card fraud detection using Naïve Bayes model based and KNN classifier." *International Journal of Advance Research, Ideas and Innovations in Technoloy* 4, no. 3 (2018): 44.

[8] Husejinovic, Admel. "Credit card fraud detection using naive Bayesian and c4. 5 decision tree classifiers." *Husejinovic, A.(2020). Credit card fraud detection using naive Bayesian and C* 4 (2020): 1-5.

[9] Saheed, Yakub K., Moshood A. Hambali, Micheal O. Arowolo, and Yinusa A. Olasupo. "Application of GA feature selection on Naive Bayes, random forest and SVM for credit card fraud detection." In *2020 international conference on decision aid sciences and application (DASA)*, pp. 1091-1097. IEEE, 2020.

[10] Varmedja, Dejan, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla. "Credit card fraud detection-machine learning methods." In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1-5. IEEE, 2019.

[11] Yee, Ong Shu, Saravanan Sagadevan, and Nurul Hashimah Ahamed Hassain Malim. "Credit card fraud detection using machine learning as data mining technique." *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 10, no. 1-4 (2018): 23-27.

[12] Malini, N., and M. Pushpa. "Analysis on credit card fraud identification techniques based on KNN and outlier detection." In *2017 third international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB)*, pp. 255-258. IEEE, 2017.

[13] Ganji, Venkata Ratnam, and Siva Naga Prasad Mannem. "Credit card fraud detection using anti-k nearest neighbor algorithm." *International Journal on Computer Science and Engineering* 4, no. 6 (2012): 1035-1039.

[14] Vengatesan, K., A. Kumar, S. Yuvraj, V. Kumar, and S. Sabnis. "Credit card fraud detection using data analytic techniques." *Advances in Mathematics: Scientific Journal* 9, no. 3 (2020): 1185-1196.

[15] Zareapoor, Masoumeh, K. R. Seeja, and M. Afshar Alam. "Analysis on credit card fraud detection techniques: based on certain design criteria." *International journal of computer applications* 52, no. 3 (2012).

[16] Nancy, A. Maria, G. Senthil Kumar, S. Veena, NA S. Vinoth, and Moinak Bandyopadhyay. "Fraud detection in credit card transaction using hybrid model." In *AIP Conference Proceedings*, vol. 2277, no. 1, p. 130010. AIP Publishing LLC, 2020.

[17] Kaur, Darshan. "Machine Learning Approach for Credit Card Fraud Detection (KNN & Naïve Bayes)." In *Machine Learning Approach for Credit Card Fraud Detection (KNN & Naïve Bayes)(March 30, 2020). Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*. 2020.

[18] Saheed, Yakub Kayode, Usman Ahmad Baba, and Mustafa Ayobami Raji. "Big Data Analytics for Credit Card Fraud Detection Using Supervised Machine Learning Models." In *Big Data Analytics in the Insurance Market*, pp. 31-56. Emerald Publishing Limited, 2022.

[19] Adewumi, Aderemi O., and Andronicus A. Akinyelu. "A survey of machine-learning and nature-inspired based credit card fraud detection techniques." *International Journal of System Assurance Engineering and Management* 8 (2017): 937-953.

[20] Mehbodniya, Abolfazl, Izhar Alam, Sagar Pande, Rahul Neware, Kantilal Pitambar Rane, Mohammad Shabaz, and Mangena Venu Madhavan. "Financial fraud detection in healthcare using machine learning and deep learning techniques." *Security and Communication Networks* 2021 (2021): 1-8.

[21] Handa, Akansha, Yash Dhawan, and Prabhat Semwal. "Hybrid analysis on credit card fraud detection using machine learning techniques." *Handbook of Big Data Analytics and Forensics* (2022): 223-238.

[22] Tiwari, Pooja, Simran Mehta, Nishtha Sakhuja, Ishu Gupta, and Ashutosh Kumar Singh. "Hybrid method in identifying the fraud detection in the credit card." In *Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2020*, pp. 27-35. Springer Singapore, 2021.

[23] Kazemi, Zahra, and Houman Zarrabi. "Using deep networks for fraud detection in the credit card transactions." In *2017 IEEE 4th International conference on knowledge-based engineering and innovation (KBEI)*, pp. 0630-0633. IEEE, 2017.

[24] Faraji, Zahra. "A Review of Machine Learning Applications for Credit Card Fraud Detection with A Case study." *SEISENSE Journal of Management* 5, no. 1 (2022): 49-59.

[25] Prusti, Debachudamani, and Santanu Kumar Rath. "Web service based credit card fraud detection by applying machine learning techniques." In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pp. 492-497. IEEE, 2019.

[26] Ahammad, Jalal, Nazia Hossain, and Mohammad Shafiul Alam. "Credit card fraud detection using data pre-processing on imbalanced data-Both oversampling and undersampling." In Proceedings of the International Conference on Computing Advancements, pp. 1-4. 2020.

[27] Ata, Oğuz, and Layth Hazim. "Comparative analysis of different distributions dataset by using data mining techniques on credit card fraud detection." Tehnički vjesnik 27, no. 2 (2020): 618-626.

[28] Shirgave, Suresh, Chetan Awati, Rashmi More, and Sonam Patil. "A review on credit card fraud detection using machine learning." International Journal of Scientific & technology research 8, no. 10 (2019): 1217-1220.