1. **INTRODUCTION**

Credit card fraud is a prevalent and costly problem in the modern financial landscape. With the increasing prevalence of online transactions and the widespread use of credit and debit cards, criminals have devised various sophisticated methods to defraud individuals and financial institutions. With the increase of people using credit cards in their daily lives, credit card companies should take special care in the security and safety of the customers. According to (Creditcard statistics 2021) the number of people using credit cards around the world was 2.8 billion in 2019, in addition 70% of those users own a single card at least.

To combat this issue, financial institutions and businesses need robust tools and techniques to identify and prevent fraudulent transactions. Machine learning offers a powerful solution to detect credit card fraud effectively. In this project, we will leverage PyCaret to develop a credit card fraud detection model. PyCaret is a Python library that simplifies the end-to-end machine learning workflow, making it easier for data scientists and analysts to experiment with various machine learning algorithms and techniques.

**1.1 MOTIVATION:**

The motivation for credit card fraud detection is multifaceted and driven by several important factors. Firstly, it's crucial for financial protection as credit card fraud presents a direct financial threat to individuals and organizations. Detecting and preventing fraud is essential to safeguard people's financial assets and prevent significant monetary losses. Moreover, it plays a pivotal role in maintaining trust in the financial industry, as customers expect their transactions to be secure and protected. Compliance with regulatory requirements is another motivating factor, as many countries mandate effective fraud detection and prevention measures to avoid legal and financial penalties. Moreover, with the advent of digital banking and fintech innovations, customers have a plethora of options at their disposal, making them more inclined to switch providers in pursuit of enhanced services and experiences. Therefore, understanding the underlying factors contributing to churn is of utmost importance for financial institutions seeking to remain competitive.

Efficient fraud detection systems also enhance operational efficiency by reducing the workload of fraud investigators and customer service representatives. The technological advancements in data science, machine learning, and artificial intelligence provide the tools and techniques necessary for accurate and efficient fraud detection.

**1.2 PROBLEM STATEMENT:**

Credit card fraud presents a multifaceted problem encompassing various dimensions, including severe financial losses for individuals and businesses, identity theft risks, erosion of consumer trust, operational expenses for financial institutions, legal consequences, reputation damage, and concerns about data privacy and security. Moreover, technological advancements continually challenge the effectiveness of fraud prevention. Addressing credit card fraud requires a holistic approach, combining technology, regulations, and public awareness, to safeguard financial systems and protect individuals and organizations from the adverse impacts of fraud.

**1.3 OBJECTIVE OF PROJECT:**

The main aim of this project is the detection of credit card fraudulent transactions, as it’s important to figure out the fraudulent transactions so that customers don’t get charged for the purchase of products that they didn’t buy. The detection of the credit card fraudulent transactions will be performed with multiple ML techniques then a comparison will be made between the outcomes and results of each technique to find the best and most suited model in the detection of the credit card transaction that are fraudulent, graphs and numbers will be provided as well.

1. **ANALYSIS**

**2.1 EXISTING APPROACH:**

Support Vector Machine or SVM and Random Forest are the is one of the most popular Supervised Learning algorithms used for developing the model for detecting the fraudulent credit card transactions.

**Data Collection and Preprocessing:**

Gather historical credit card transaction data, including both legitimate and fraudulent transactions. Preprocess the data by handling missing values, scaling features, and encoding categorical variables. Split the data into training and testing sets to evaluate model performance.

**Feature Engineering:**

Extract relevant features from the transaction data, such as transaction amount, location, time of day, and more. Consider creating additional features that may enhance fraud detection, such as transaction frequency or velocity.

**Model Selection:**

Choose the machine learning models for the project. In this case, you're using SVM and Random Forest.

**Model Training:**

Train the SVM and Random Forest models on the training data. Ensure that you optimize model hyperparameters for the best performance.

**Model Evaluation:**

Assess the performance of the models using various metrics, including accuracy, precision, recall, F1-score, and the ROC-AUC curve. Since credit card fraud detection is often imbalanced, consider using techniques like resampling (oversampling or under sampling) to handle class imbalance.

**Ensemble Methods:**

You can create an ensemble model that combines the predictions of the SVM and Random Forest to improve overall performance. Methods like stacking or voting can be used for this purpose.

**Threshold Tuning**

Adjust the decision threshold to control the trade-off between false positives and false negatives based on the specific requirements of the application. This is crucial to balance fraud detection and customer inconvenience.

**Cross-Validation:**

Implement cross-validation to ensure that the model's performance is reliable and doesn't overfit the training data.

**Real-Time Detection:**

Implement the models in a real-time credit card transaction monitoring system where transactions are analysed as they occur.

**Monitoring and Maintenance:**

Continuously monitor model performance, and retrain the models periodically to adapt to changing fraud patterns. Keep the models up-to-date with the latest data and ensure that they remain effective over time.

**2.2 APPROACH SYSTEM:**

Using the PyCaret library for credit card fraud detection involves a streamlined and efficient approach. PyCaret simplifies the machine learning workflow and can be applied in the following manner:

**Data Collection and Preprocessing**:

Gather historical credit card transaction data, including both legitimate and fraudulent transactions. Preprocess the data, handling missing values, scaling features, encoding categorical variables, and performing any necessary data cleaning.

**Install and Import PyCaret:**

Install the PyCaret library if you haven't already. You can use `pip install pycaret` to install it. Import PyCaret and set up your Python environment.

**Initialize and Setup PyCaret:**

Initialize a PyCaret environment by running ‘from pycaret.classification import \*’ and then using ‘setup()’ to set up the environment for classification tasks. Specify the target variable (fraud or not) and any other relevant configuration options.

**Compare Models:**

Use PyCaret's ‘compare\_models()’ function to compare and evaluate various machine learning models for credit card fraud detection. This step helps you identify the top-performing models that are most suitable for your data.

**Create Model:**

After selecting the best-performing models from the comparison, use the ‘create\_model()’ function to instantiate those models. PyCaret will train them on your dataset, and you can then fine-tune them for better performance.

**Tune Models:**

PyCaret provides built-in hyperparameter tuning capabilities with the ‘tune\_model()’ function. You can use this to optimize the selected models and improve their predictive power.

**Evaluate Models:**

Evaluate the models using PyCaret's built-in functions to assess their performance. This includes metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and more. PyCaret generates visualizations and summary reports for easier model comparison.

**Interpret Models:**

PyCaret's interpretation tools, including feature importance plots and SHAP values, help you understand the factors that influence the model's predictions. This is crucial for explaining why a certain transaction is flagged as fraudulent.

**Model Ensembling:**

Create an ensemble of the best-performing models using PyCaret's ensemble functions. Ensembles can improve overall accuracy and robustness by combining the strengths of individual models.

**Finalize Model:**

After evaluating and interpreting the results, select the best-performing model based on your criteria. Finalize the model using `finalize\_model()` and save it for deployment.

**Deployment:**

Deploy the chosen model in a production environment, where it can analyse credit card transactions in real-time and alert for potentially fraudulent activities.

**Advantages of PyCaret:**

Using PyCaret for credit card fraud detection offers a range of significant advantages. First and foremost, it enables rapid prototyping by simplifying the machine learning workflow, reducing the time needed to set up an initial fraud detection system. It automates many tasks, including data preprocessing, feature engineering, hyperparameter tuning, and model evaluation, enhancing overall efficiency. PyCaret also provides automated exploratory data analysis (EDA) tools and supports model comparison, making it easy to identify the best-performing model for your specific dataset. The library streamlines hyperparameter tuning, offers model interpretation tools, and supports ensemble learning.

1. **REQUIRMENTS SPECIFICATIONS**

**3.1 SOFTWARE REQUIRMENTS:**

OPERATING SYSTEM : WINDOWS, MACOS, AND LINUX.

TECHNOLOGIES : PYCARET, PYTHON , PYTHON LIBRARIES

(NUMPY ,PANDAS ,MATPLOTLIB).

IDE : JUPYTER NOTEBOOK

DOCUMENTATION TOOL : MS-WORD 2021.

**3.2 HARDWARE REQUIRMENTS:**

PROCESSOR : INTEL CORE I5 OR AMD RYZEN 5.

PROCESSOR SPEED : 1.20GHZ

RAM : 8GB

HARD DISK : 256 GB(Min)

MOUSE : PORTONICS

MONITOR : EGI/VGA

KEY BOARD : 101 STANDARDS(MIN.)

**4. IMPLEMENTATION**

**4.1 SAMPLE CODE**

**#importing necessary libraries for the analysis**

import pandas as pd

import numpy as np

**#loading the data**

credit\_card= pd.read\_csv('C:/Users/User/Desktop/Capstone/creditcard.csv')

credit\_card.head()

**#shape which gives total rows and columns in the dataset**

credit\_card.shape

**#Class column has two values '0'(non-fradulent) and '1'(fradulent)**

**#total number of rows having 0 as class value**

len(credit\_card[credit\_card['Class']==0])

**#total number of rows having 1 as class value**

len(credit\_card[credit\_card['Class']==1])

**#284315 entries for class=0 and**

**#492 entires for class=1.**

**# We can clearly see that the data is imbalance.**

**# we will use pycaret library for solving this classfication problem.**

**#install pycaret**

!pip install pycaret

**#import module and intializing setup**

from pycaret.classification import \*

clf = setup(data = credit\_card, target = 'Class')

**#during the setup it will ask if all the variables are as per the dataset, as per your needs.**

**#Type y and enter.**

**#It will also ask you % of data to be kept for training model.**

**# Here we have spit the data 70/30**

**# we will get Accuracy, Recall, Preccision, F1-score..... for different models.**

compare\_models()

**# since it shows CatBoost is giving us highest accuracy let us deep dive into it and create a model.**

catboost= create\_model('catboost')

**#Let us implement hyperpyrameter tuning to tune the model.**

tune\_model('catboost')

interpret\_model(catboost, plot='correlation')

**#Let us predict our values using the predict\_model()**

catboost\_pred = predict\_model(catboost)

**#Let us create another model xgboost as I want to show the evluate\_model() below.**

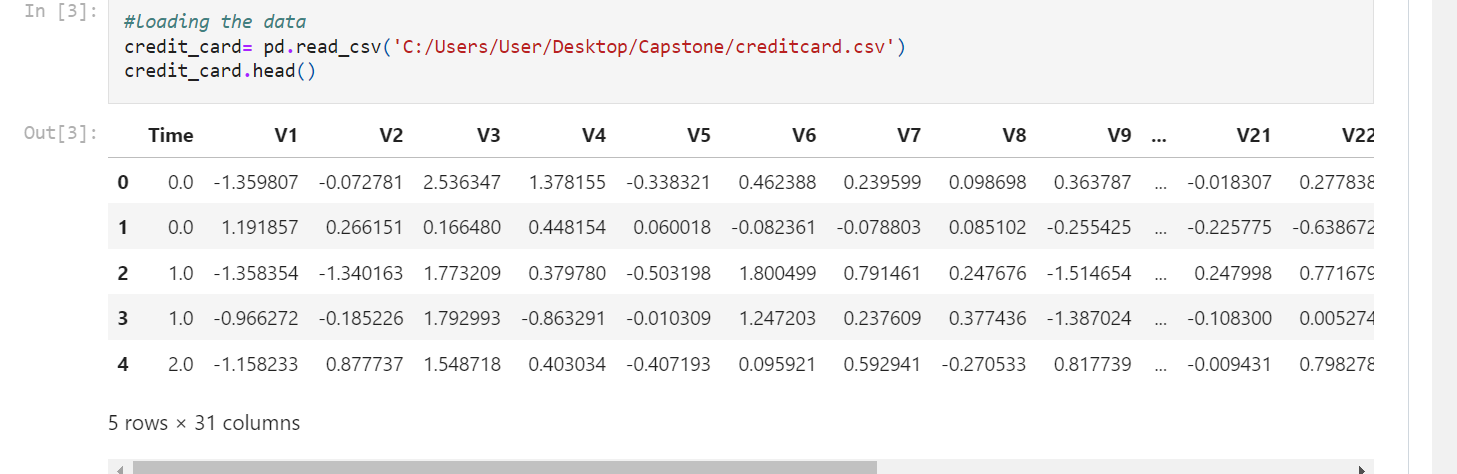
xgboost=create\_model('xgboost')

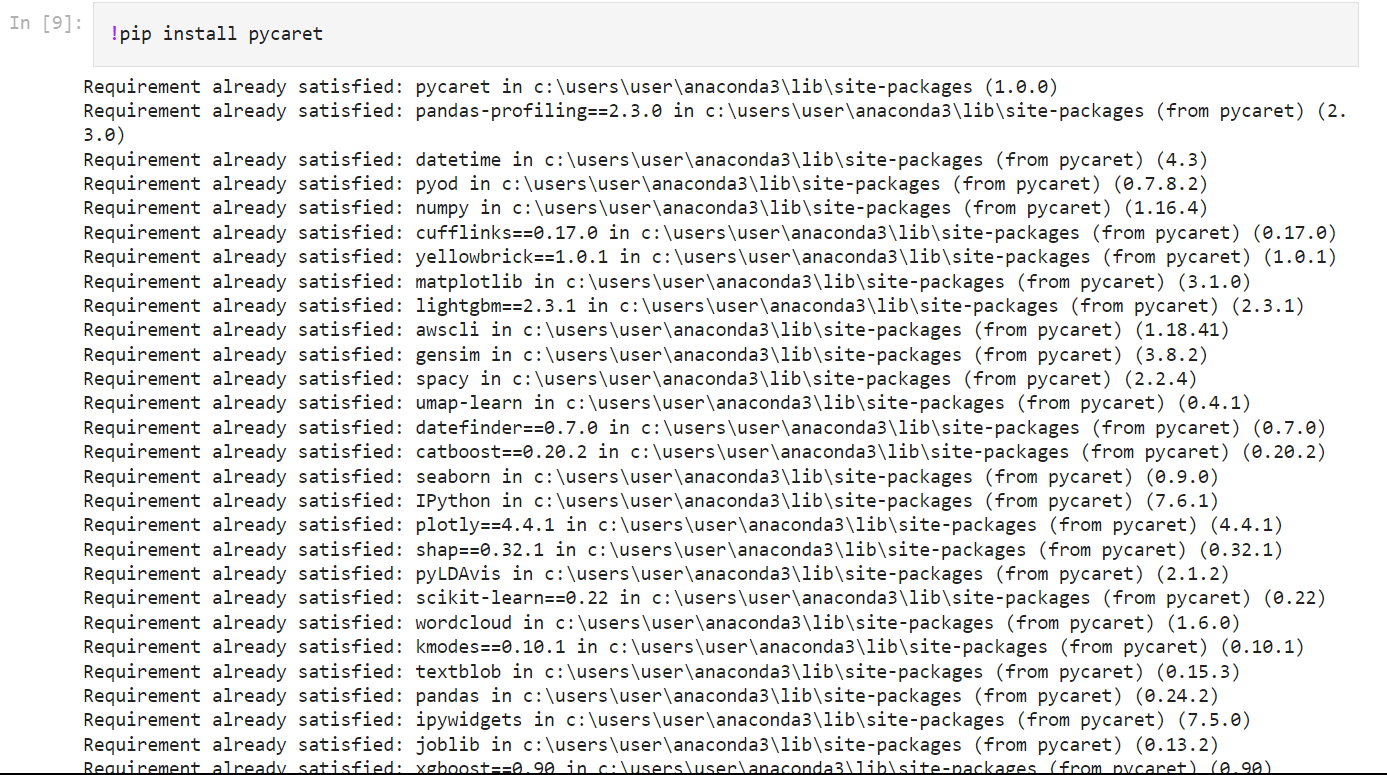
evaluate\_model(xgboost)

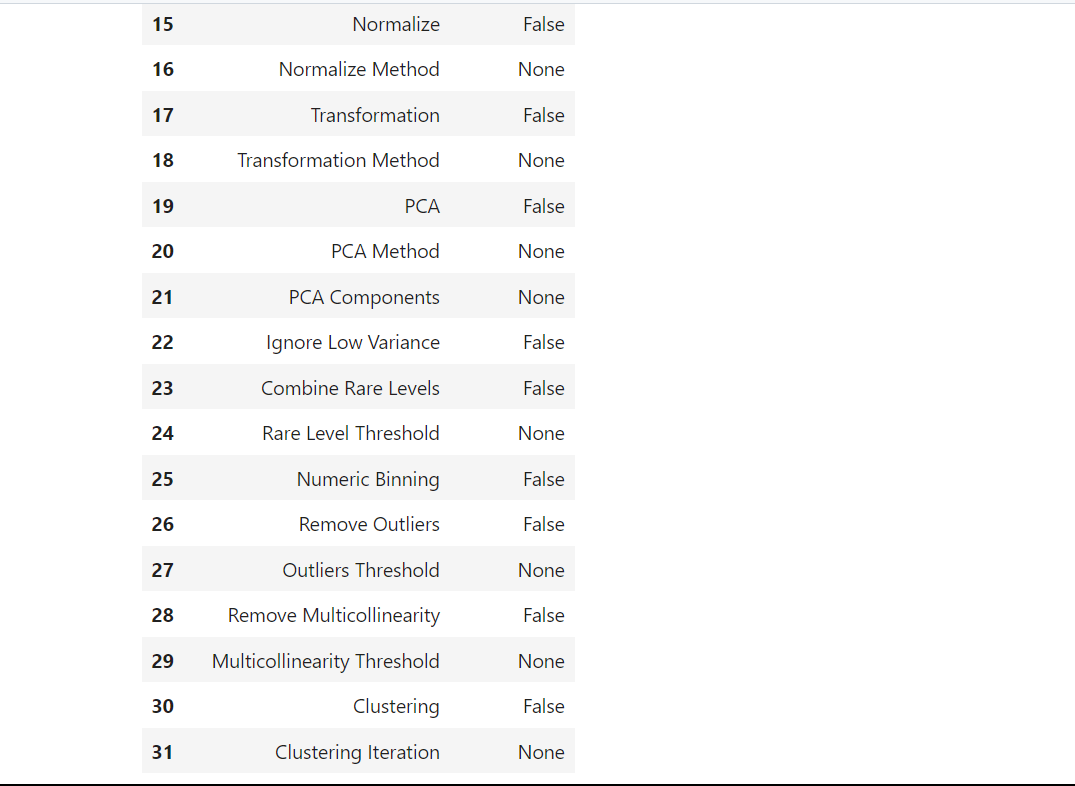
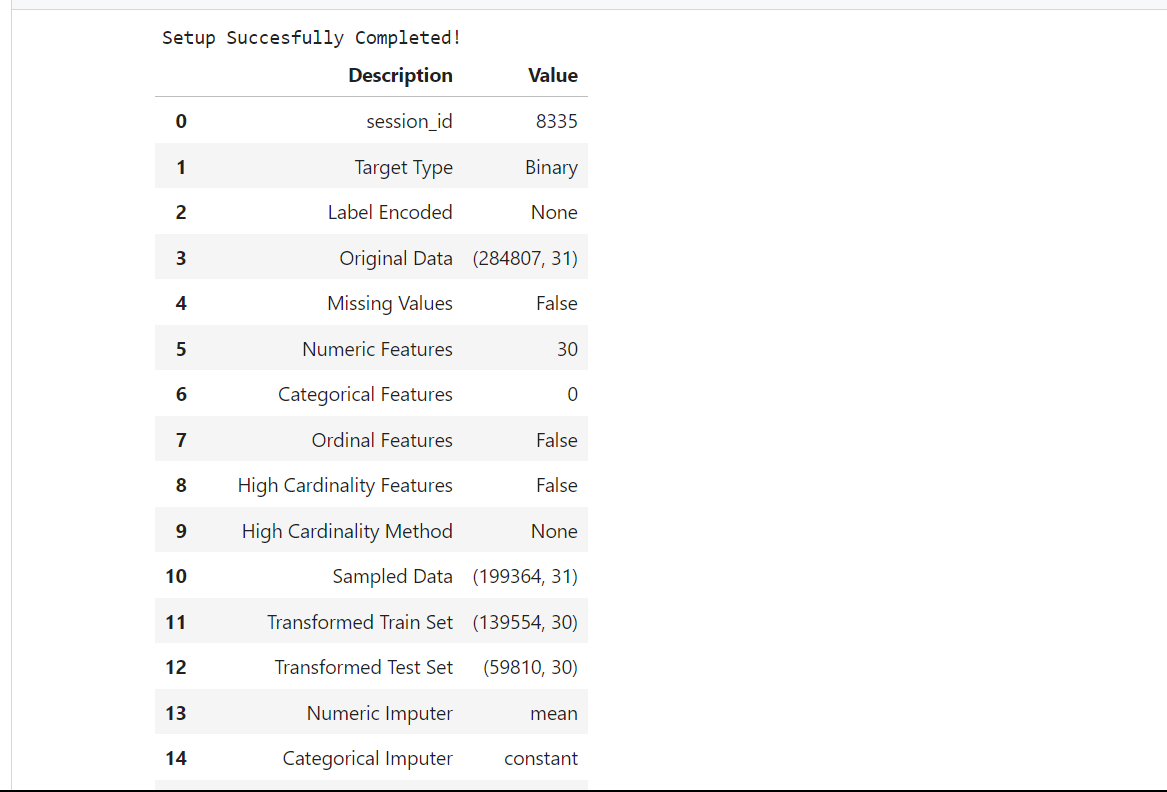
**5. SCREENSHOTS**

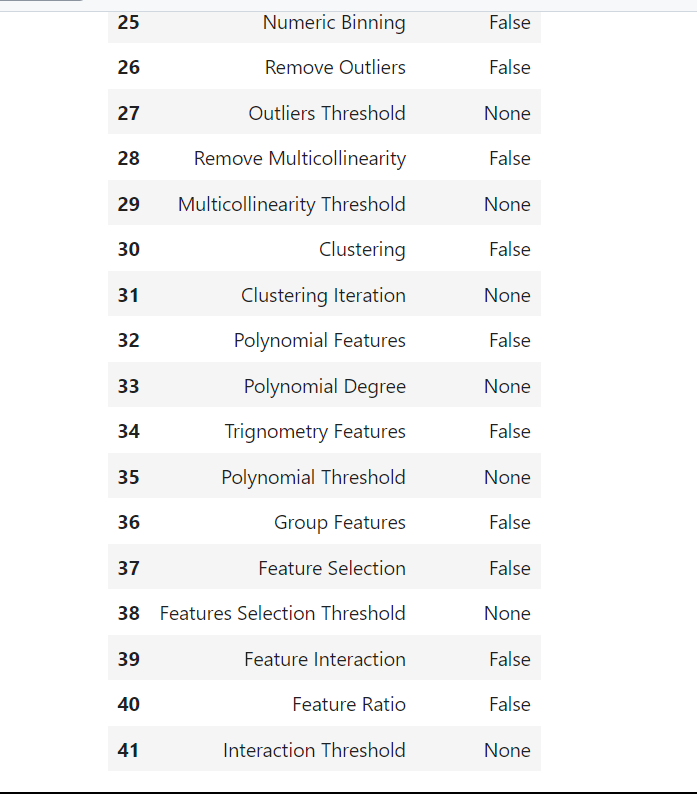


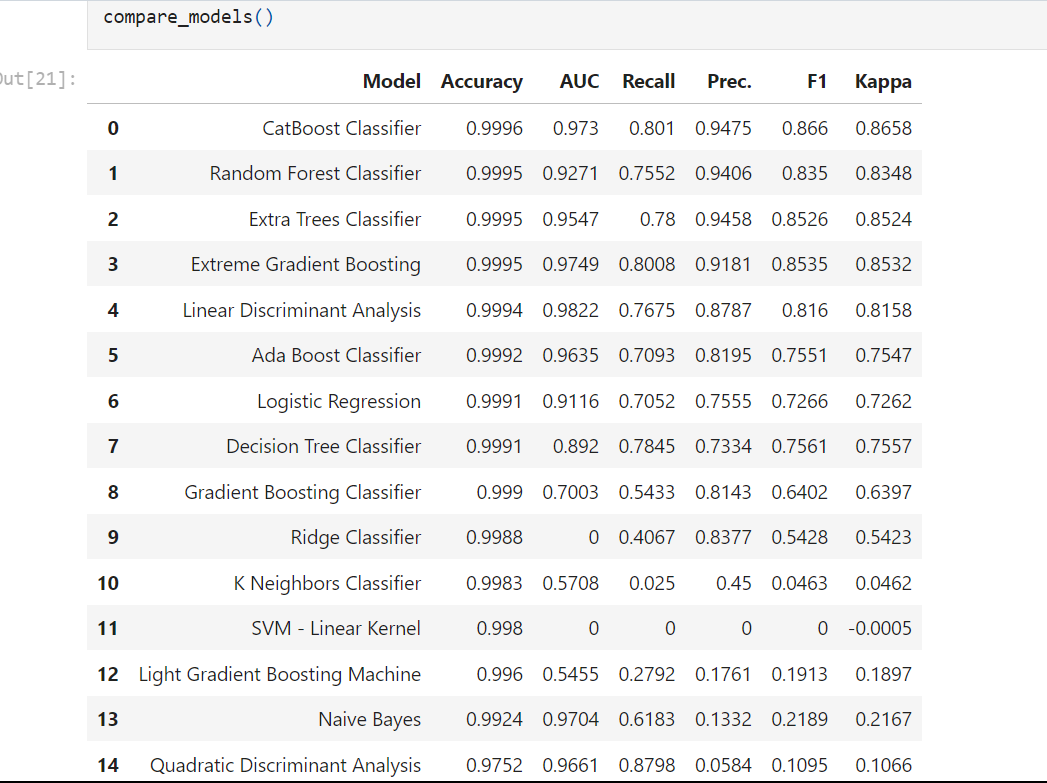


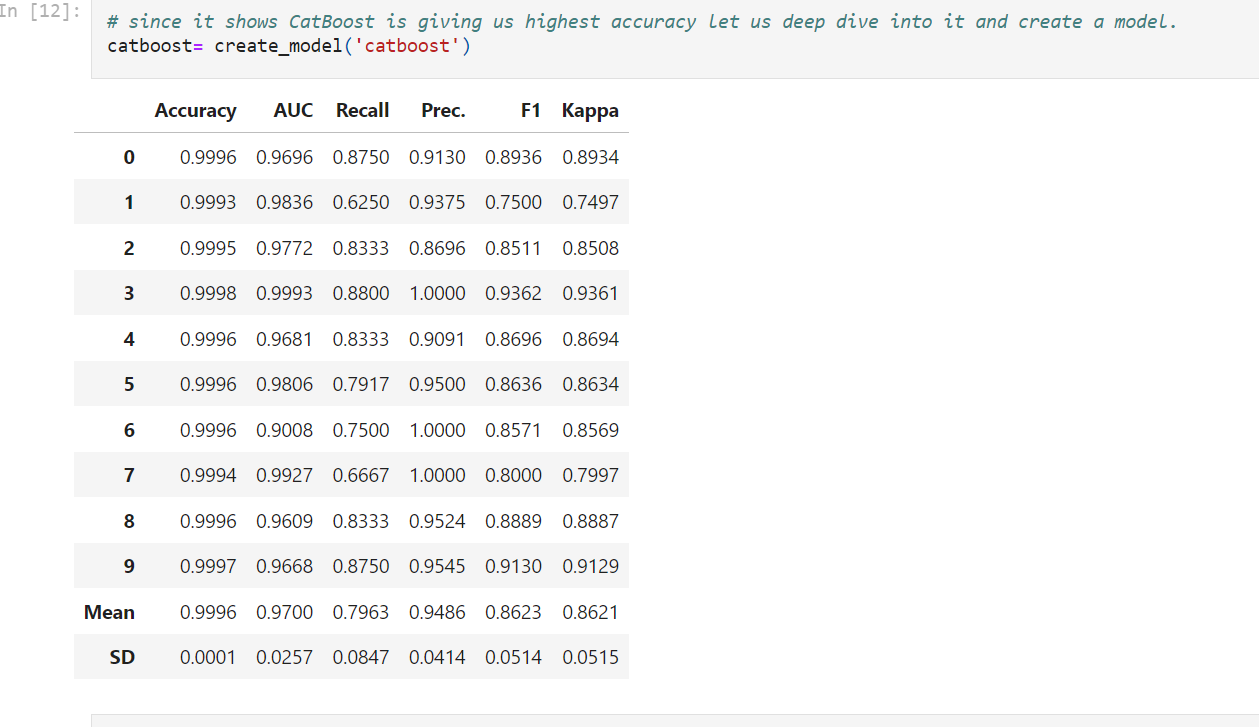
****

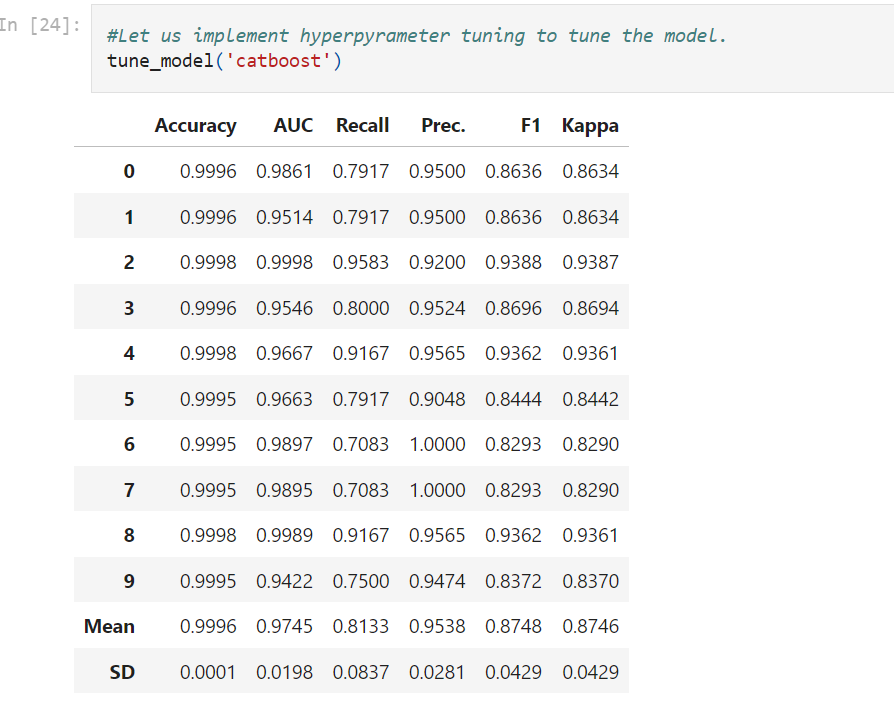
****

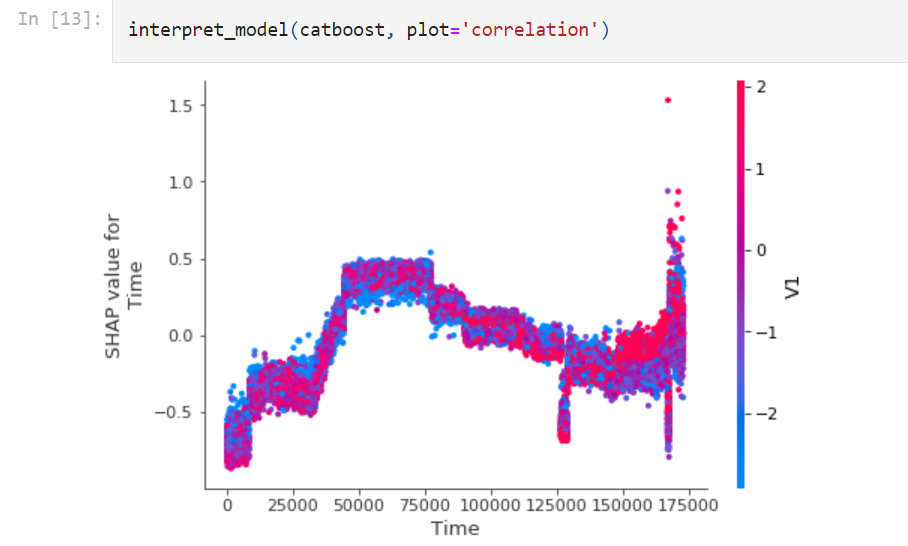
****

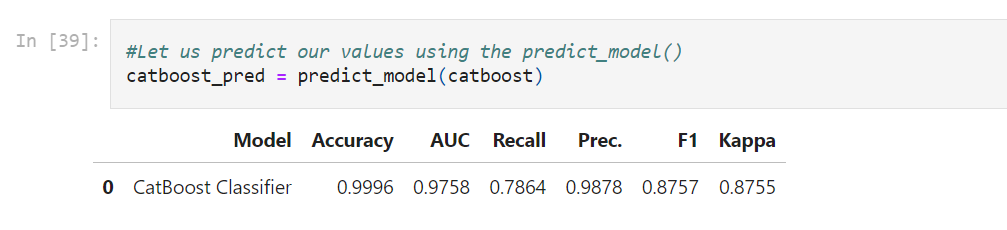
****

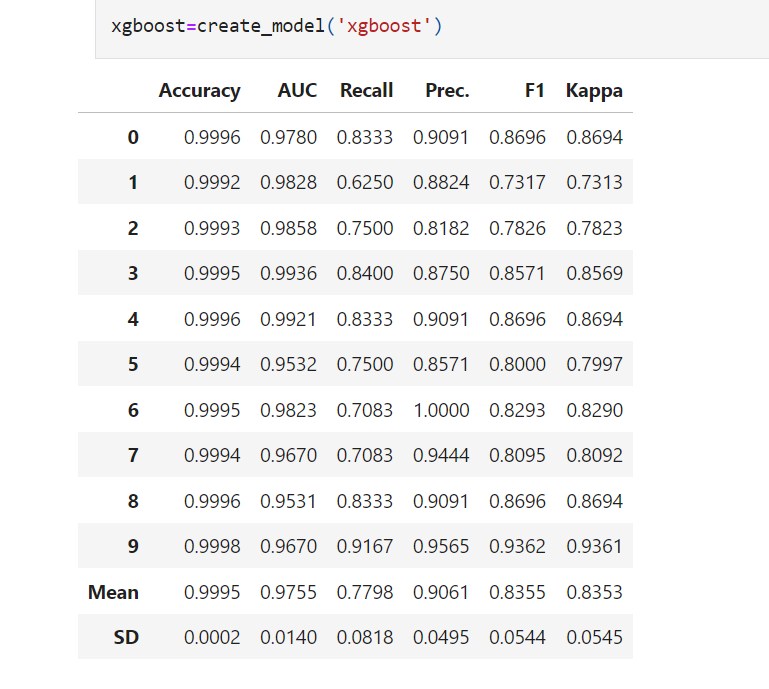
****

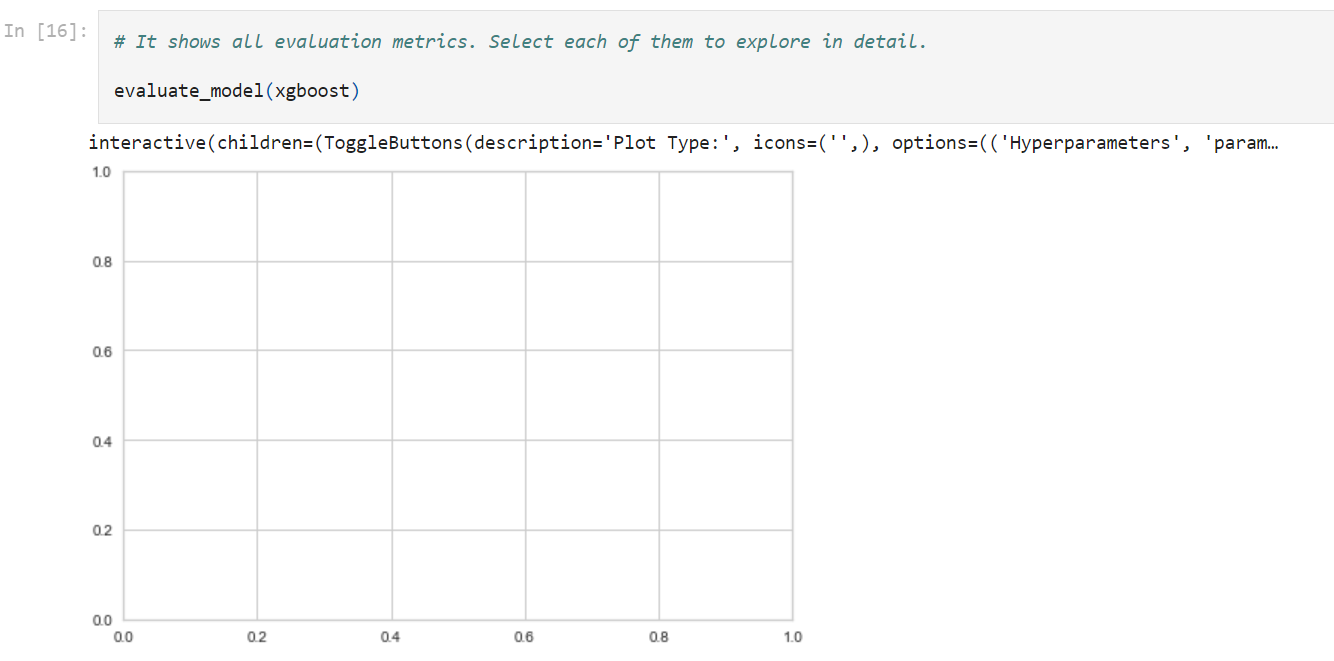
****

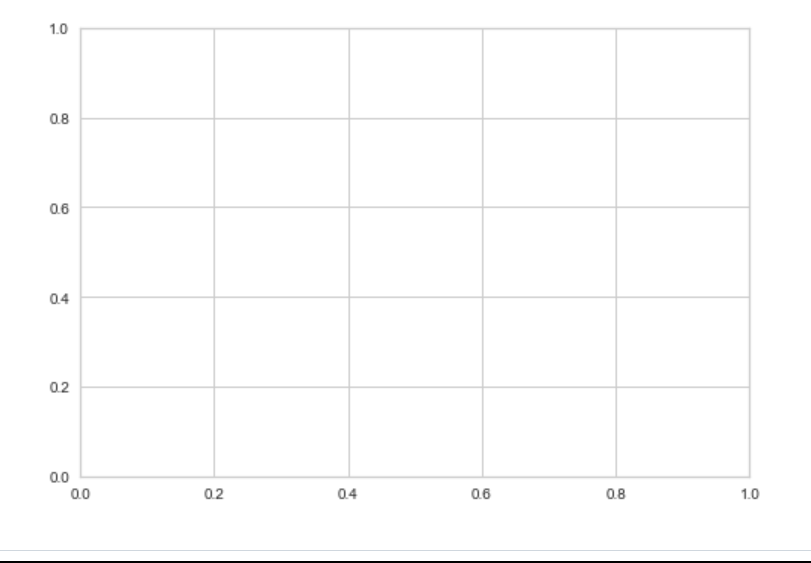
****

****

****

****

****

****

**6. CONCLUSION**

In conclusion, credit card fraud detection using PyCaret is a powerful and efficient approach to tackle the growing problem of fraudulent transactions in the financial industry. By leveraging PyCaret's automated machine learning capabilities, one can build accurate and robust fraud detection models with minimal effort. While PyCaret simplifies the process of credit card fraud detection, it's important to note that the success of any machine learning model depends on the quality and quantity of the data it is trained on. In conclusion, PyCaret is a valuable tool in the fight against credit card fraud, offering a streamlined approach to model development that can save time, improve accuracy, and enhance the security of financial transactions. However, it should be used as part of a comprehensive fraud detection strategy that includes data quality assurance, real-time monitoring, and cooperation with industry experts to stay ahead of evolving fraud schemes.

**7. FUTURE ENHANCEMENT**

In the future, the scope for enhancing credit card fraud detection using PyCaret is vast. Potential developments include advanced feature engineering tailored for fraud detection, real-time monitoring, and improved anomaly detection. Explain ability and interpretability features will be crucial for building trust and compliance. Integrations with data sources, custom model evaluation metrics, scalability for big data, regulatory compliance, and user-friendly interfaces will further bolster its capabilities. Customization and extensibility for advanced users, as well as support for research and collaboration, will be essential. Staying updated with the latest trends and research in machine learning and financial fraud will be key to effectively leveraging PyCaret in this field.

**8. BIBLIOGRAPHY & REFERENCES**

* Adepoju, O., Wosowei, J., lawte, S., & Jaiman, H. (2019). Comparative evaluation ofcredit card fraud detection using machine learning techniques. 2019 Global Conferencefor Advancement in Technology (GCAT).

<https://doi.org/10.1109/gcat47503.2019.8978372>

* Bhanusri, A., Valli, K. R. S., Jyothi, P., Sai, G. V., & Rohith, R. (2020). Credit card fraud detection using Machine learning algorithms. Journal of Research in Humanities and Social Science, 8(2), 04-11.
* Dheepa, V., & Dhanapal, R. (2012). Behavior based credit card fraud detection using support vector machines. ICTACT Journal on Soft Computing, 02(04), 391–397.

<https://doi.org/10.21917/ijsc.2012.0061>

* Dighe, D., Patil, S., & Kokate, S. (2018). Detection of credit card fraud transactions using machine learning algorithms and Neural Networks: A comparative study. 2018 Fourth International Conference on Computing Communication Control and Automation(ICCUBEA). <https://doi.org/10.1109/iccubea.2018.8697799>
* Gupta, A., Lohani, M. C., & Manchanda, M. (2021). Financial fraud detection using naive Bayes algorithm in highly imbalance data set. Journal of Discrete Mathematical Sciences and Cryptography, 24(5), 1559–1572.

<https://doi.org/10.1080/09720529.2021.1969733>

* Kiran, S., Guru, J., Kumar, R., Kumar, N., Katariya, D., & Sharma, M. (2018). Credit card fraud detection using Naïve Bayes model based and KNN classifier. International Journal Of Advance Research, Ideas And Innovations In Technology, 4(3).
* Malini, N., & Pushpa, M. (2017). Analysis on credit card fraud identification techniques based on KNN and outlier detection. 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics(AEEICB).

<https://doi.org/10.1109/aeeicb.2017.7972424>

* Maniraj, S. P., Saini, A., Ahmed, S., & Sarkar, S. D. (2019). Credit card fraud detection using machine learning and Data Science. Credit Card Fraud Detection UsingMachine Learning and Data Science, 08(09).

https://doi.org/10.17577/ijertv8is090031