**PHASE-5**

**TEAM NO 11**

**Credit card fraud Detection**

**DOCUMENTATION**

Problem Definition

The problem is to develop a machine learning-based system for real-time credit card fraud detection. The goal is to create a solution that can accurately identify fraudulent transactions while minimizing false positives. This project involves data pre-processing, feature engineering, model selection, training, and evaluation to create a robust fraud detection system.

Design Thinking

Data Sourcing:

Data sourcing is a critical step in credit card fraud detection. It involves gathering historical transaction data from various sources, such as financial institutions, payment processors, and e-commerce platforms. This data typically includes information on transaction amounts, timestamps, merchant details, and customer profiles. Additionally, dataset provided in the website of Kaggle is taken in hand to carry forward with the processes.

Dataset link: [**https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud**](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)

Data Pre-processing:

Data pre-processing is a pivotal phase in credit card fraud detection that focuses on cleaning and preparing the gathered data for analysis. This step involves tasks like handling missing values, removing duplicates, and standardizing data formats. Given the imbalanced nature of fraud detection datasets, where legitimate transactions far outnumber fraudulent ones, techniques like oversampling or undersampling may be employed to balance the classes. Furthermore, feature scaling and normalization help ensure that features contribute equally to the model, preventing any bias. Data pre-processing ensures that the dataset is in a suitable form for subsequent analysis and model development.

Feature Engineering:

Feature engineering is a creative and crucial step in credit card fraud detection. It involves selecting, creating, or transforming features to extract meaningful information from the raw data. Features like transaction frequency, spending patterns, and geographical information can be engineered to improve the model's ability to discriminate between legitimate and fraudulent transactions. Additionally, dimensionality reduction techniques, such as principal component analysis (PCA), can be applied to reduce the computational complexity of the model while preserving essential information. Effective feature engineering plays a vital role in enhancing the model's predictive power.

Model Selection:

Model selection is a pivotal step in credit card fraud detection that involves choosing the most appropriate machine learning algorithms for the task. Commonly used models include logistic regression, decision trees, random forests, support vector machines (SVMs), and neural networks. The choice of model depends on factors like the dataset size, complexity, and interpretability requirements. Ensemble methods, which combine multiple models, are often favored for their ability to capture complex patterns in the data. Model selection is a critical decision as it directly impacts the accuracy and efficiency of fraud detection.

Model Training:

Model training is the process of teaching the selected machine learning model to recognize patterns and make predictions based on the prepared dataset. During training, the model learns to differentiate between legitimate and fraudulent transactions by adjusting its internal parameters. It's essential to split the dataset into training and validation sets to evaluate the model's performance and prevent overfitting. Hyperparameter tuning and cross-validation are used to optimize model performance. The model is trained iteratively until it achieves the desired level of accuracy and generalization.

Model Evaluation:

Model evaluation is the final stage in credit card fraud detection, where the performance of the trained model is assessed using various metrics like precision, recall, F1-score, and the area under the receiver operating characteristic curve (AUC-ROC). Precision measures the model's ability to correctly identify fraud cases, while recall quantifies its ability to find all fraudulent transactions. The F1-score combines these metrics to provide a balanced assessment of model performance. AUC-ROC evaluates the model's overall ability to discriminate between fraud and non-fraud cases. Regular model re-evaluation is crucial to ensure continued effectiveness in detecting evolving fraud patterns, and adjustments

Overview of the complete process:

Split the dataset into training, validation, *and* test sets. The training set is used to train the model, the validation set is used for hyper parameter tuning, and the test set is reserved for final evaluation.

Creating an innovative credit card fraud detection system involves a comprehensive process with several key steps:

**1. Data Collection:** Gather historical credit card transaction data, including both legitimate and fraudulent transactions, along with relevant features like transaction amount, location, and time.

**2. Data Preprocessing:** Clean the data by handling missing values and outliers, normalize or scale numeric features, and encode categorical variables into numerical representations.

**3. Feature Engineering:** Create new features to enhance fraud detection, such as aggregating transaction history and identifying anomalies in transaction patterns.

**4. Data Splitting:** Divide the dataset into training, validation, and test sets for model development and evaluation.

**5. Model Selection:** Choose an appropriate machine learning or deep learning model, considering options like logistic regression, decision trees, random forests, support vector machines, or neural networks.

**6. Model Training:** Train the selected model using the training data while addressing class imbalance using techniques like oversampling, under sampling, or synthetic data generation.

**7. Hyper parameter Tuning:** Optimize the model's hyper parameters using the validation dataset to improve performance.

**8. Model Evaluation:** Assess the model's performance on the test dataset using metrics like precision, recall, F1-score, and ROC-AUC.

**9. Threshold Setting:** Determine a classification threshold that balances false positives and false negatives based on business requirements.

**10. Deployment:** Deploy the trained model in the production environment, integrating it with credit card processing systems.

**11. Monitoring and Maintenance:** Continuously monitor model performance, retraining it periodically to adapt to changing fraud patterns.

**12. Alerts and Reporting**: Implement alerting systems for flagged transactions, investigating them manually or automatically.

**13. Feedback Loop:** Establish a feedback loop between the fraud detection system and the fraud investigation team to improve accuracy.

**14. Documentation and Compliance**: Thoroughly document the process, ensuring compliance with data privacy regulations and security standards.

**15. Scalability**: Design the system to scale with transaction volume growth and evolving fraud tactics.

The Credit card fraud detection dataset extracted from kaggle platform accurately consists of 31 columns with 2 lakhs+ entries.

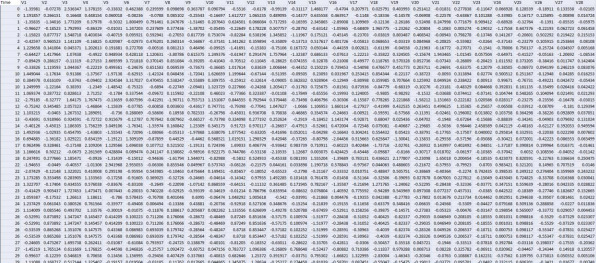
COLUMN NAMES

>Time

>V1-V28

>Amount

>Class

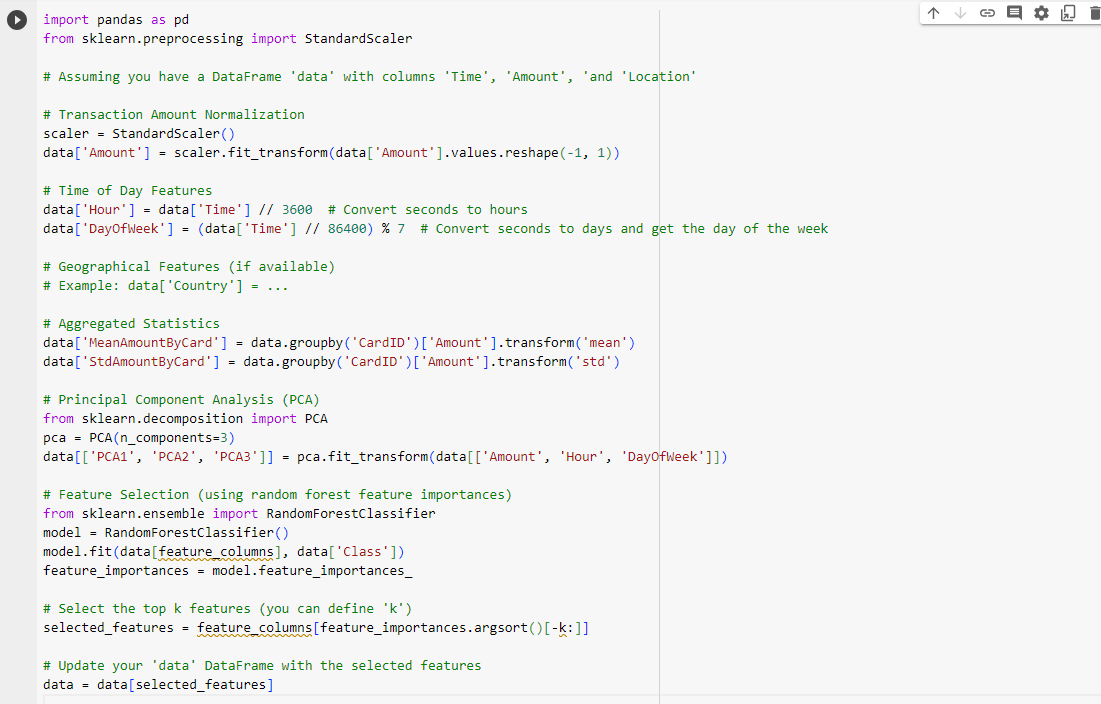


**NOTE:** The initial columns from the dataset is displayed above for you perusal**.**

**MODEL DEVELOPMENT**

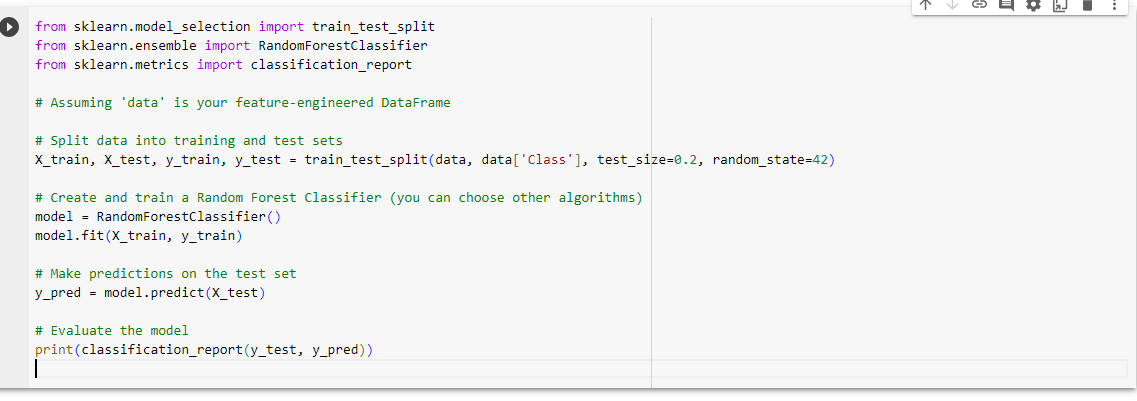
**Feature Engineering:**

Feature engineering is a crucial step in creating a successful fraud detection model. It involves selecting, transforming, and creating relevant features that help the model distinguish between genuine and fraudulent transactions.



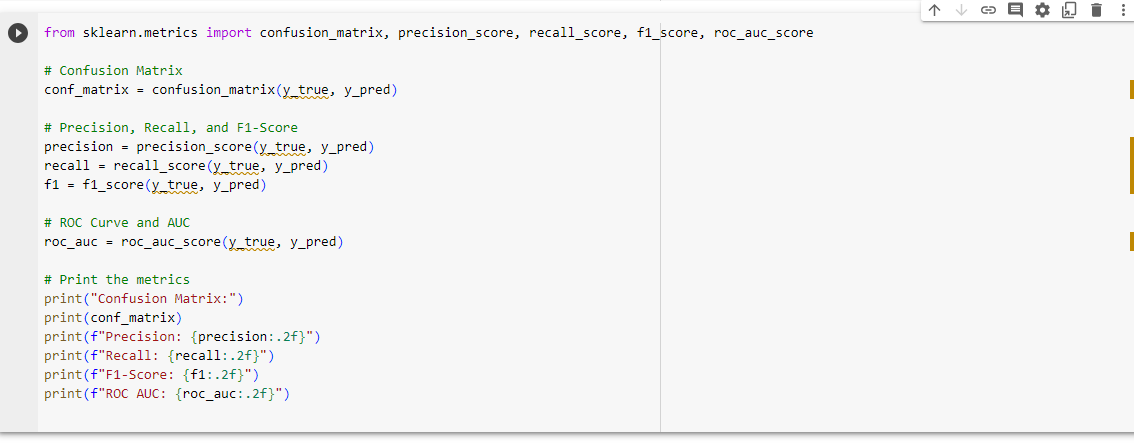
**Model Training:**

Here we can use various machine learning algorithms for this, but ensemble methods and deep learning models tend to work well for fraud detection tasks.



**Evaluation:**

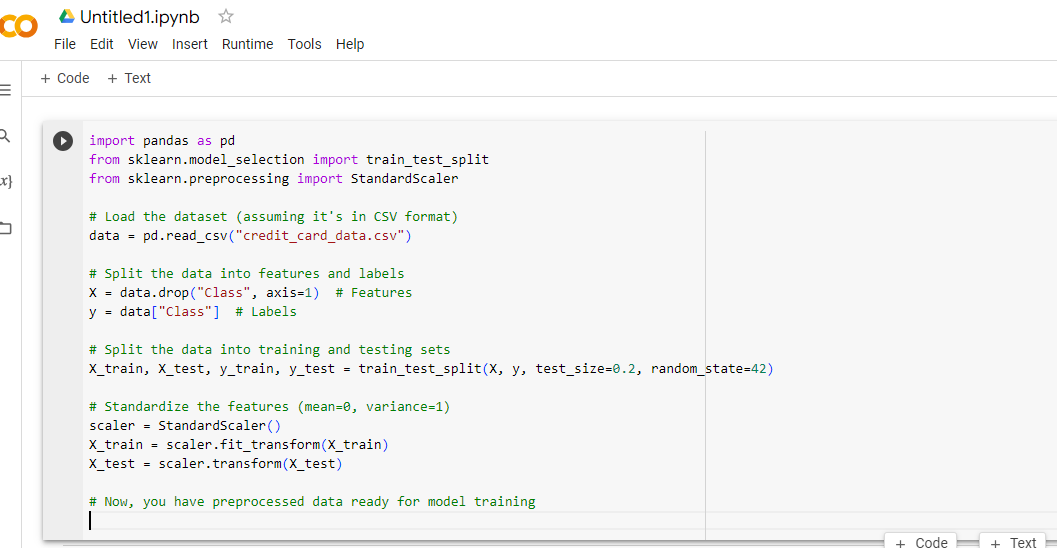
After training your models, it's essential to evaluate their performance. Since fraud detection is an imbalanced classification problem, standard accuracy is not the best metric.



SUBMISSION

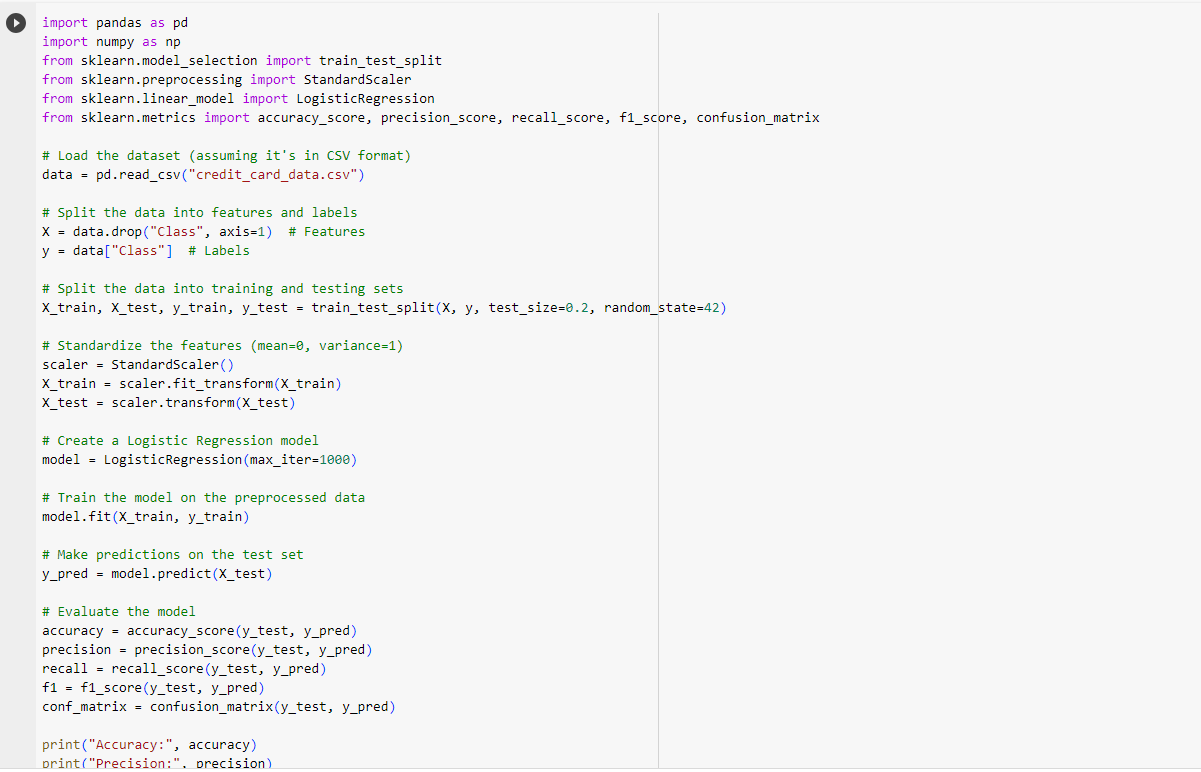
Data Pre-processing

Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.



Model Training

Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.



Evaluation Metrics

Evaluation metrics are used to measure the quality of the statistical or machine learning model. Evaluating machine learning models or algorithms is essential for any project. There are many different types of evaluation metrics available to test a model.

