BRAIN STATION 23

Project Summary On,

“ Credit Card Fraud Detection App”

Submitted By:

Sudipta Das
Student ID:2008049
Dept. of ETE,CUET

1. Project Overview

This project involved developing a machine learning-powered web application that predicts the probability of credit card fraud based on transaction details. The app is built using **Streamlit**, and it utilizes a **Random Forest Classifier** trained on a highly imbalanced dataset using **undersampling** and **hyperparameter tuning** techniques.

The final product allows users to manually input transaction features and get real-time fraud risk predictions with visual feedback.

2. Objectives

- Detect fraudulent transactions based on anonymized features.
- Provide a user-friendly web interface for prediction.
- Demonstrate the effectiveness of a tuned Random Forest model on an imbalanced dataset.

3. What I Built

- **Machine Learning Pipeline**: Trained a Random Forest classifier on a processed and undersampled dataset with proper preprocessing.
- **Model Tuning**: Performed grid/random search to optimize parameters.
- **Web Application**: Built using Streamlit, featuring input forms for transaction data, model loading via joblib, and fraud risk output.
- **Integration**: Ensured seamless integration of the trained model with the web interface using proper input shaping and validation.

4. What I Learned

- **Dealing with Imbalanced Data**: Implemented undersampling techniques to handle class imbalance and learned how such methods affect model performance.
- **Model Evaluation**: Gained experience in using ROC-AUC and precision-recall curves to assess classification performance beyond accuracy.
- **Web Deployment**: Learned how to convert a machine learning model into a deployable, interactive web app using Streamlit.
- **Serialization**: Learned how to serialize and load trained models using joblib.

5. Challenges Faced

- **Data Imbalance**: The dataset had extremely few fraudulent transactions, requiring careful handling to prevent the model from being biased toward the majority class.

- **Feature Interpretation**: The anonymized features (V1–V28) made it difficult to understand their individual impact, requiring the model to be treated as a black-box.

- **Model Size**: The trained model (fraud_rf_model.pkl) needed to be optimized for compatibility with the lightweight app interface.

- **User Input Validation**: Ensuring that the interface gracefully handles missing or incorrect values from users was another key challenge.

6. Future Improvements

- **Model Comparison**: Provide results from multiple models (e.g., XGBoost, SVM) to compare outcomes.

- **Visualization**: Integrate more charts (like feature importance or SHAP plots) for transparency.

- **Deploy Online**: Deploy the app using Streamlit Cloud or other platforms for broader access.

7. Conclusion

This project offered a hands-on experience in bridging data science and web development. From handling imbalanced data to building a real-time prediction app, I learned the full workflow of an applied machine learning project. It deepened my understanding of model evaluation, serialization, and app deployment, while reinforcing the importance of user experience in ML tools.