Microsoft Elevate

CAPSTONE PROJECT

PROJECT TITLE

PRESENTED BY

STUDENT NAME: SHREERAM MISHRILAL PRAJAPATI

COLLEGE NAME: ST. WILFRED'S COLLEGE OF COMPUTER SCIENCES, MIRA ROAD (E)

DEPARTMENT: MASTER OF COMPUTER APPLICATION

EMAIL ID: SHREERAMPRAJAPATI2@GMAIL.COM

OUTLINE:

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result (Output Image)
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT:

Credit card fraud poses a significant challenge to financial institutions worldwide. Fraudulent transactions are rare compared to legitimate ones, leading to highly imbalanced datasets. The problem requires developing a system that can accurately detect fraudulent transactions while minimizing false positives and false negatives. The goal is to safeguard customers and institutions by identifying suspicious activity early and reducing financial risks.

PROPOSED SOLUTION:

- **The proposed system leverages machine learning models to classify transactions as either fraudulent or legitimate. By analyzing transaction patterns and features, the system aims to provide a reliable fraud detection mechanism:**

- **Data Collection**
- Gather historical credit card transaction data, Ensure the dataset includes both legitimate and fraudulent transactions to train the model effectively.
- Use Azure Blob Storage to securely store raw and processed datasets.
- **Data Preprocessing**
- Clean and preprocess the dataset to handle missing values, outliers, and inconsistencies.
- Apply normalization and scaling techniques to ensure features are comparable.
- **Machine Learning Algorithm**
- Implement **Two-Class Decision Forest** in Azure Machine Learning Designer for model training and evaluation.
- Use **Random Forest Classifier** from the scikit-learn library in Google Colab for experimentation and validation.
- Compare performance metrics across both platforms to ensure robustness and accuracy.
- **Deployment**
- Publish the trained model using Azure Machine Learning Designer, to be deploy for real time use.
- **Evaluation**
- Assess the model’s performance using metrics such as Accuracy, Precision, Recall, F1-Score, and MCC.
- Validate results with confusion matrix visualization to understand classification effectiveness.
- Continuously fine-tune the model based on monitoring feedback and updated transaction data.

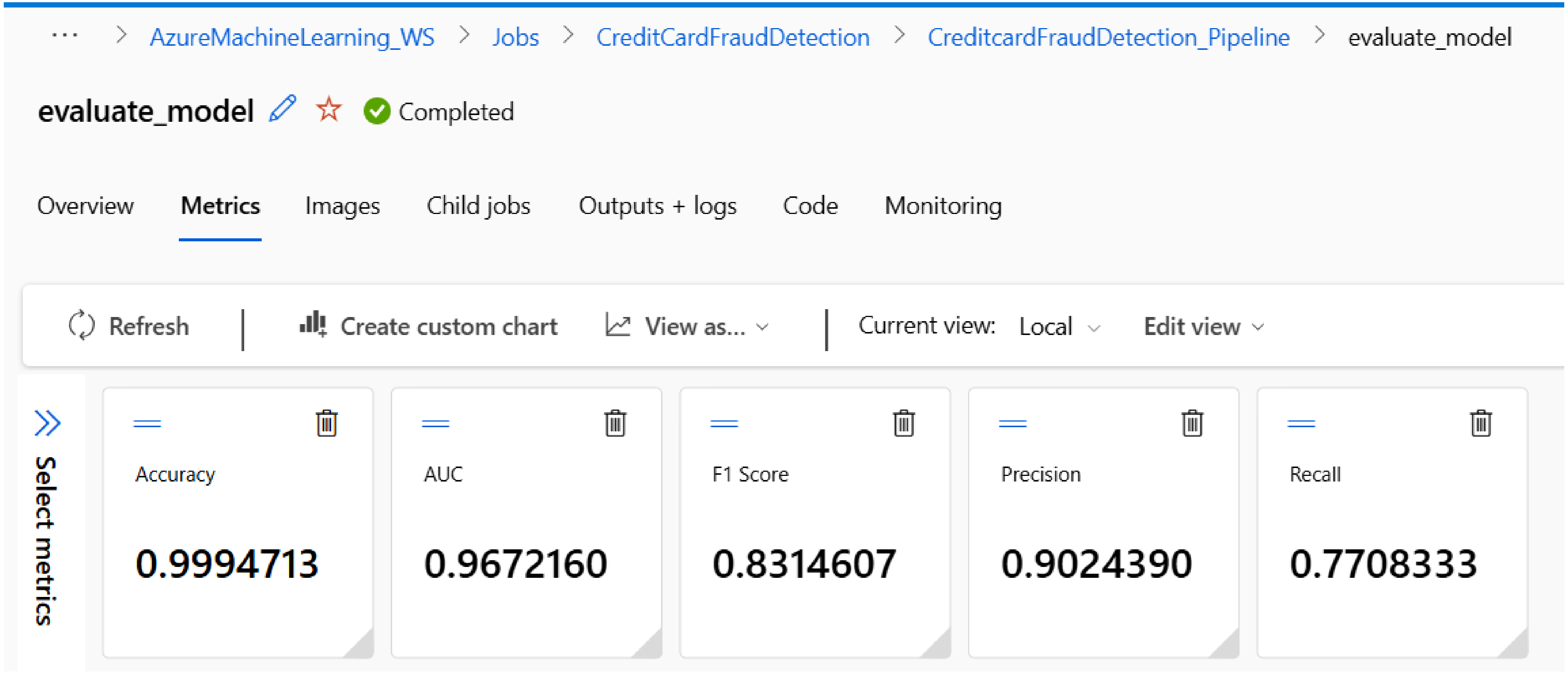
SYSTEM APPROACH:

- **Azure Machine Learning Designer –** Used to design, train, and deploy machine learning models with a drag-and-drop interface.
- **Azure Log Analytics –** Enables monitoring and logging of system performance and fraud detection activities.
- **Azure Blob Storage –** Provides secure and scalable storage for datasets and model outputs.
- **Azure Key Vault –** Ensures secure management of sensitive information such as API keys and credentials.
- **Google Colab –** Used for experimentation, prototyping, and training models with Python libraries (Pandas, Numpy, Matplotib, Seaborn, Scikit - Learn)in a collaborative environment.

ALGORITHM & DEPLOYMENT:

- In the Algorithm section, describe the machine learning algorithm chosen for predicting bike counts. Here's an example structure for this section:
- **Algorithm Selection:**
  - Two-Class Decision Forest in Azure Machine Learning Designer. This ensemble method is chosen for its ability to handle imbalanced datasets and provide robust classification by combining multiple decision trees.
  - Random Forest Classifier from the scikit-learn library in Google Colab. This algorithm is selected due to its proven effectiveness in classification tasks, resistance to overfitting, and interpretability of feature importance.
- **Data Input:**
  - These features are preprocessed and stored securely in **Azure Blob Storage** before being fed into the models..
- **Training Process:**
  - Azure ML Designer: The Two-Class Decision Forest is trained using historical transaction data. Cross-validation is applied to ensure generalization and reduce overfitting. Hyperparameters such as the number of trees and maximum depth are tuned for optimal performance.
  - Google Colab: The Random Forest Classifier is trained using scikit-learn with similar preprocessing steps. Techniques such as stratified sampling are employed to address class imbalance.
- **Prediction Process:**
  - Azure ML Deployment: The trained Two-Class Decision Forest model is deployed as a REST API endpoint. Incoming transactions are scored in real time, with predictions returned as either “fraudulent” or “legitimate.”
  - Google Colab Experimentation: The Random Forest Classifier is used for batch predictions and validation. Predictions are generated by aggregating votes across multiple decision trees, ensuring reliable classification.

RESULT: AZURE MACHINE LEARNING

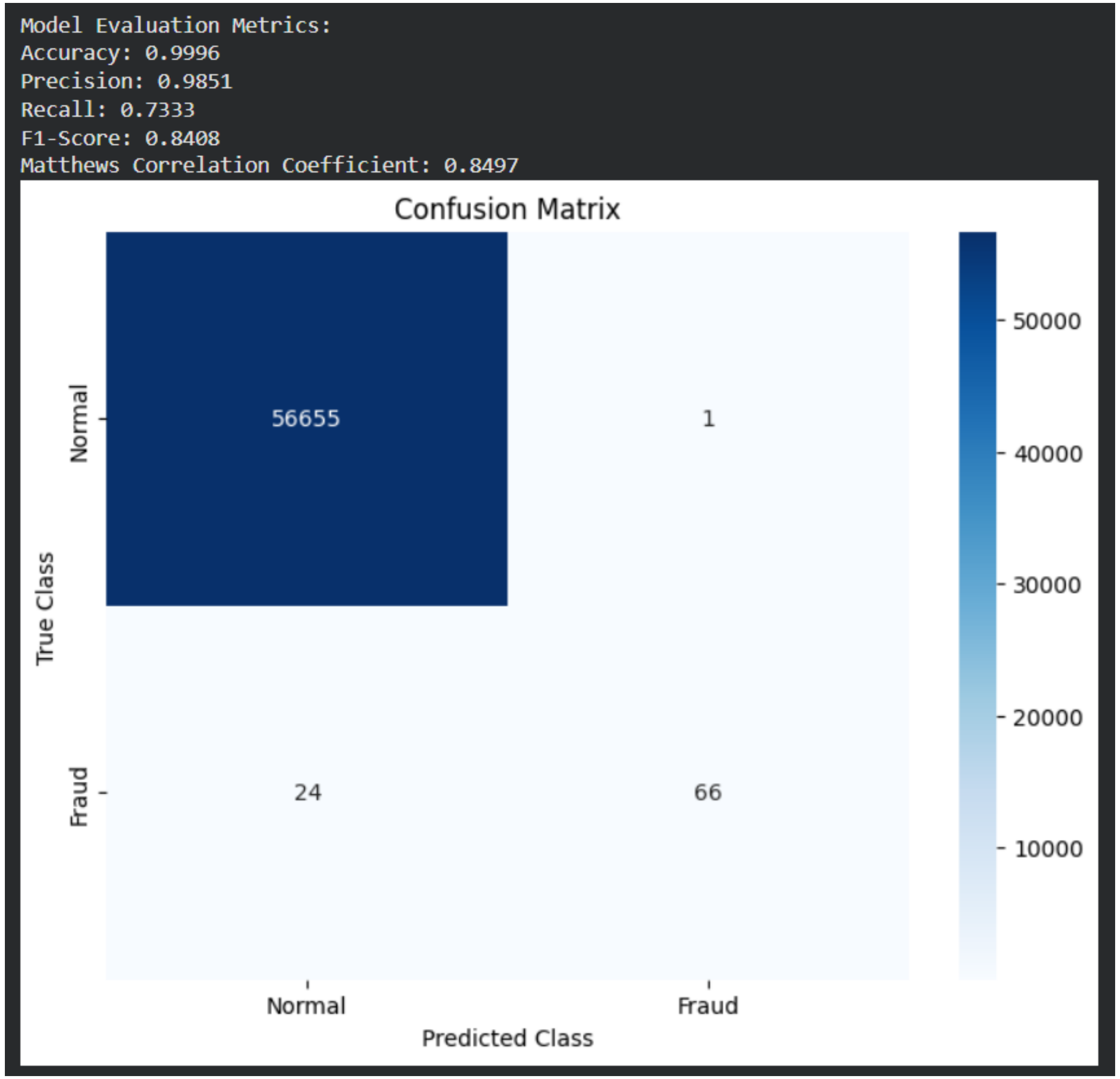


evaluate_model ☆ Completed

Current view: Local

Accuracy	AUC	F1 Score	Precision	Recall
0.9994713	0.9672160	0.8314607	0.9024390	0.7708333

RESULT: GOOGLE COLAB NOTEBOOK



```
Model Evaluation Metrics:
Accuracy: 0.9996
Precision: 0.9851
Recall: 0.7333
F1-Score: 0.8408
Matthews Correlation Coefficient: 0.8497
```

CONCLUSION:

The project successfully demonstrates the application of machine learning for fraud detection. While accuracy is high, the imbalanced dataset highlights the importance of precision and recall in evaluating performance. The integration of Azure services ensures secure, scalable, and monitored deployment, while Google Colab provides flexibility for experimentation.

FUTURE SCOPE:

- **Deployment:** Expanding deployment to production environments with APIs for real-time fraud detection.
- **Consumption:** Financial institutions can consume the model through REST endpoints, integrating it into transaction monitoring systems.
- **Enhancements:** Techniques such as oversampling, undersampling, or advanced algorithms (e.g., XGBoost, deep learning) can be explored to improve recall.
- **Monitoring:** Continuous monitoring with Azure Log Analytics to refine model performance over time.
- **Security:** Enhanced use of Azure Key Vault to safeguard sensitive credentials during deployment.

REFERENCES:

Azure Machine Learning Official Documentation

AICTE Azure Internship Team

Project Github Link

Thank You