Interim 2 submission Report: Model Building and Training

Overview

This report outlines the process of building and training machine learning models to enhance fraud detection in e-commerce and banking transactions. As a data scientist at Adey Innovations Inc., the goal is to develop accurate fraud detection models by leveraging machine learning techniques and performing comprehensive data analysis.

Business concept

Adey Innovations Inc. aims to improve transaction security by developing advanced fraud detection systems to

- Reduce financial losses due to fraudulent transactions.
- Strengthen trust with customers and financial institutions.
- Enable real-time fraud detection and rapid response.

The project involves multiple steps, including data analysis, feature engineering, model training, and deployment to ensure continuous improvements.

Data Preparation

Feature and Target Separation

For both datasets, we separated the features and target variables:

Fraud_Data.csv

Features: user_id, signup_time, purchase_time, purchase_value, device_id, source, browser, sex, age, ip_address

Target Variable: class

Creditcard.csv

Features: Time, V1 to V28, Amount

Target Variable: Class

Train-Test Split

We used the train_test_split function from sklearn.model_selection to create training and testing datasets for both data sources. This ensures the model is validated on unseen data.

python

Copy code

Train-test split

X_fraud, y_fraud = fraud_data.drop('class', axis=1), fraud_data['class']

X_train_fraud, X_test_fraud, y_train_fraud, y_test_fraud = train_test_split(X_fraud, y_fraud, test_size=0.2, random_state=42)

X_creditcard, y_creditcard = creditcard_data.drop('Class', axis=1), creditcard_data['Class']

X_train_creditcard, X_test_creditcard, y_train_creditcard, y_test_creditcard = train_test_split(X_creditcard, y_creditcard, test_size=0.2, random_state=42)

Model Selection

To explore model performance, the following algorithms were selected for comparison

- Logistic Regression
- Decision Tree
- Random Forest
- Gradient Boosting
- Multi-Layer Perceptron (MLP)
- Recurrent Neural Network (RNN)
- Long Short-Term Memory (LSTM)

Model Training and Evaluation

Training Models

Each model was trained using both datasets. Here is a summary of the process:

python

Copy code

```
# Model training and evaluation
models = {
    'Logistic Regression': LogisticRegression(),
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),
    'Gradient Boosting': GradientBoostingClassifier(),
    'MLP': MLPClassifier(),
}

results_fraud = {}
results_creditcard = {}

for name, model in models.items():
    model.fit(X_train_fraud, y_train_fraud)
    y_pred_fraud = model.predict(X_test_fraud)
    results_fraud[name] = {
        'Accuracy': accuracy_sc-ore(y_test_fraud, y_pred_fraud),
        'Report': classification_report(y_test_fraud, y_pred_fraud)
    }
```

```
    model.fit(X_train_creditcard, y_train_creditcard)

    y_pred_creditcard = model.predict(X_test_creditcard)

    results_creditcard[name] = {

        'Accuracy': accuracy_score(y_test_creditcard, y_pred_creditcard),

        'Report': classification_report(y_test_creditcard, y_pred_creditcard)

    }
```

Evaluation Metrics

Accuracy scores and classification reports were generated for each model to facilitate performance comparisons.

MLOps Steps

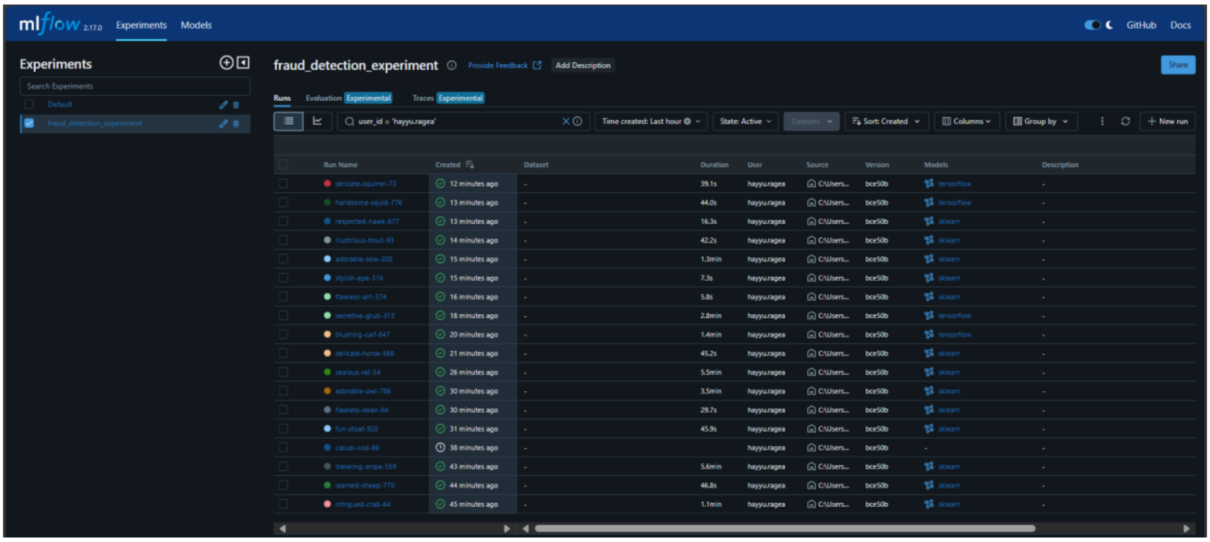
Versioning and Experiment Tracking

To track model performance, parameters, and metrics, we utilized MLflow, which enabled seamless versioning and experiment management.

python

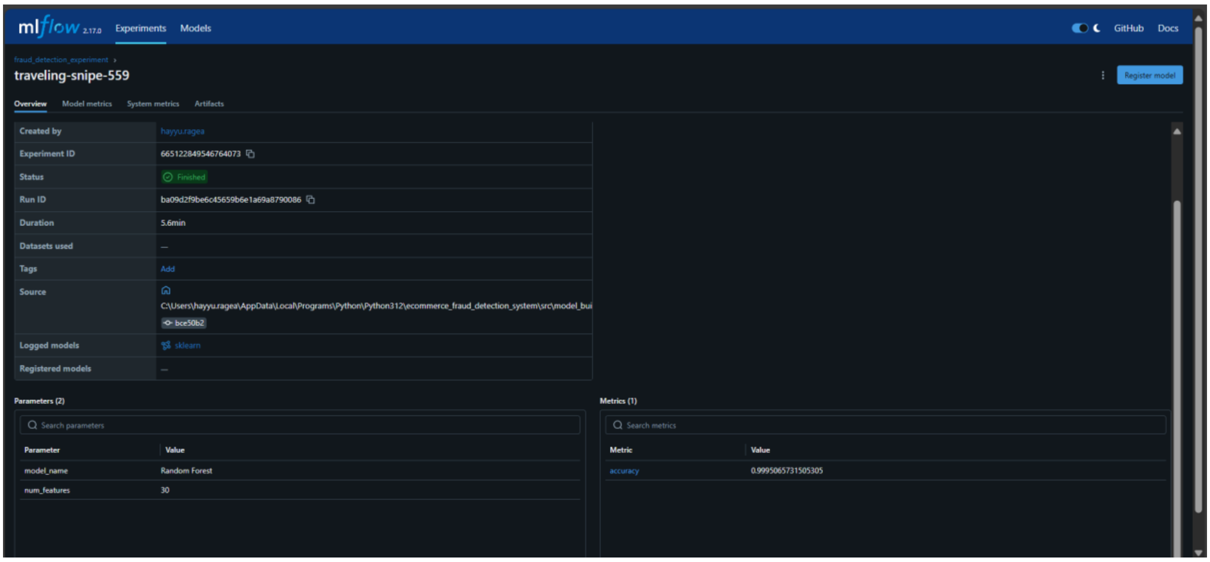
Copy code

```
# MLflow tracking setup

mlflow.start_run()

for name, model in models.items():

    mlflow.log_param("model_name", name)

    mlflow.log_metric("accuracy", results_fraud[name]['Accuracy'])

    mlflow.sklearn.log_model(model, name)

mlflow.end_run()
```

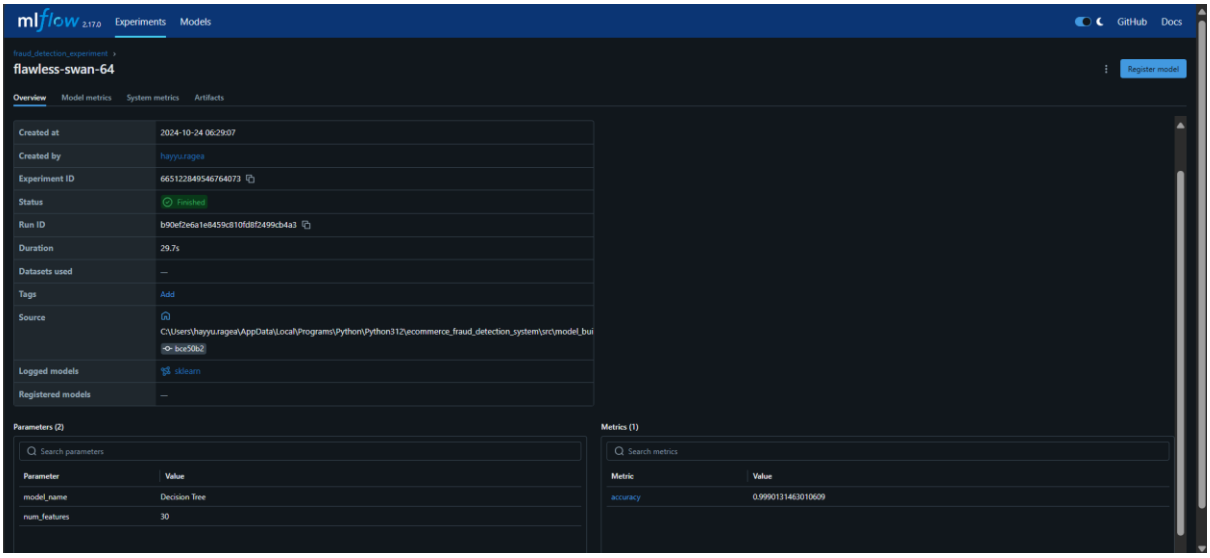


Evaluating Models for Credit Card Data MLFLOW

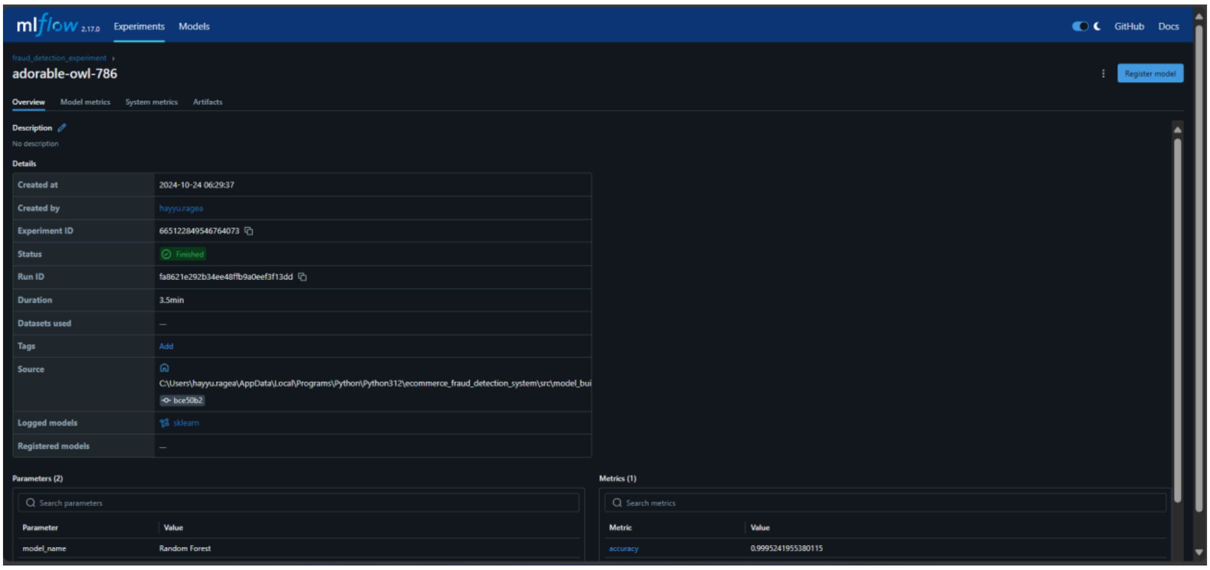
- Logistic regression



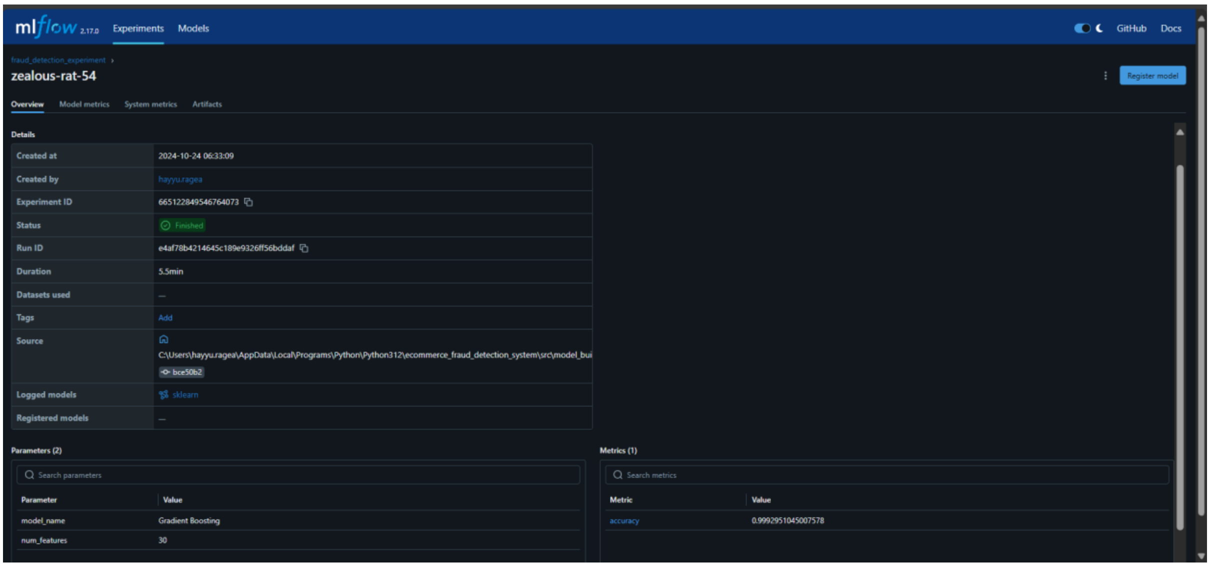
- Decision Tree



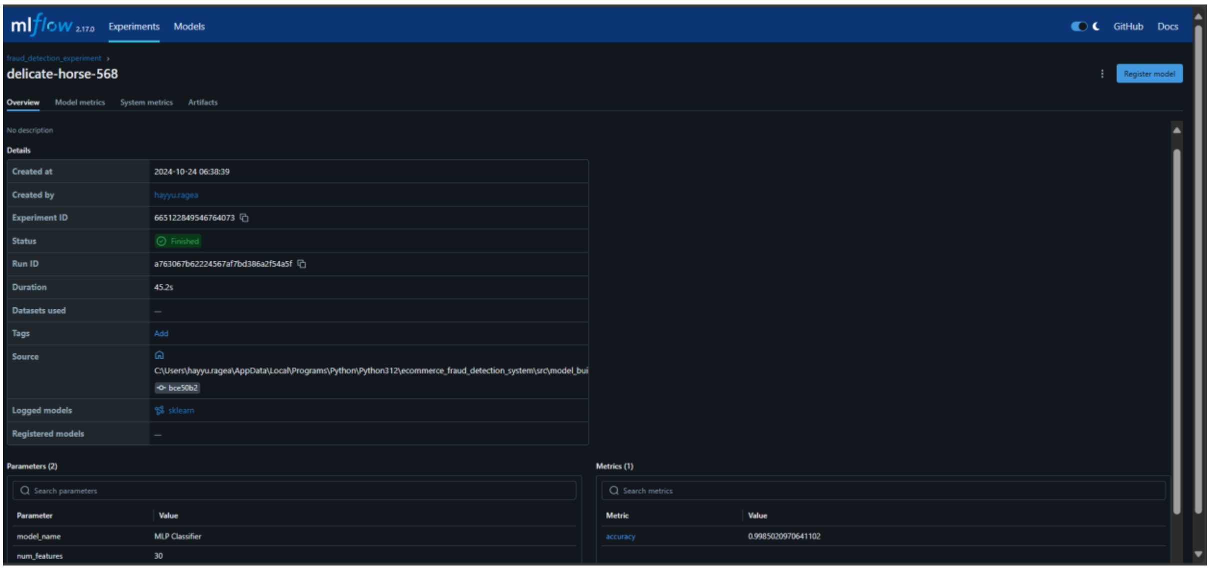
- Random Forest



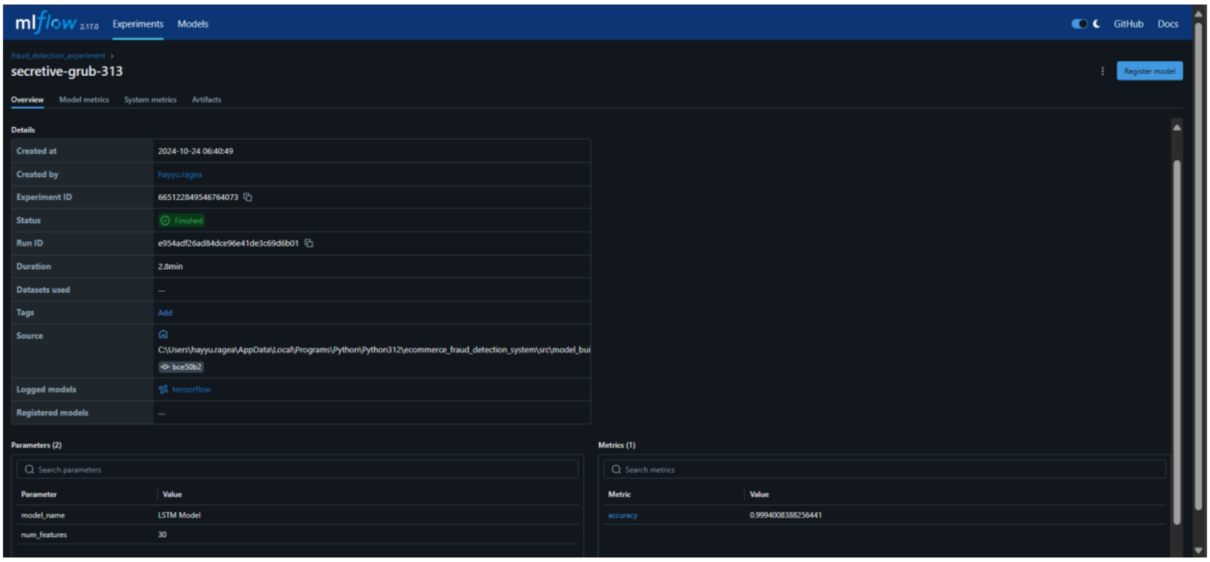
- Gradient Boosting



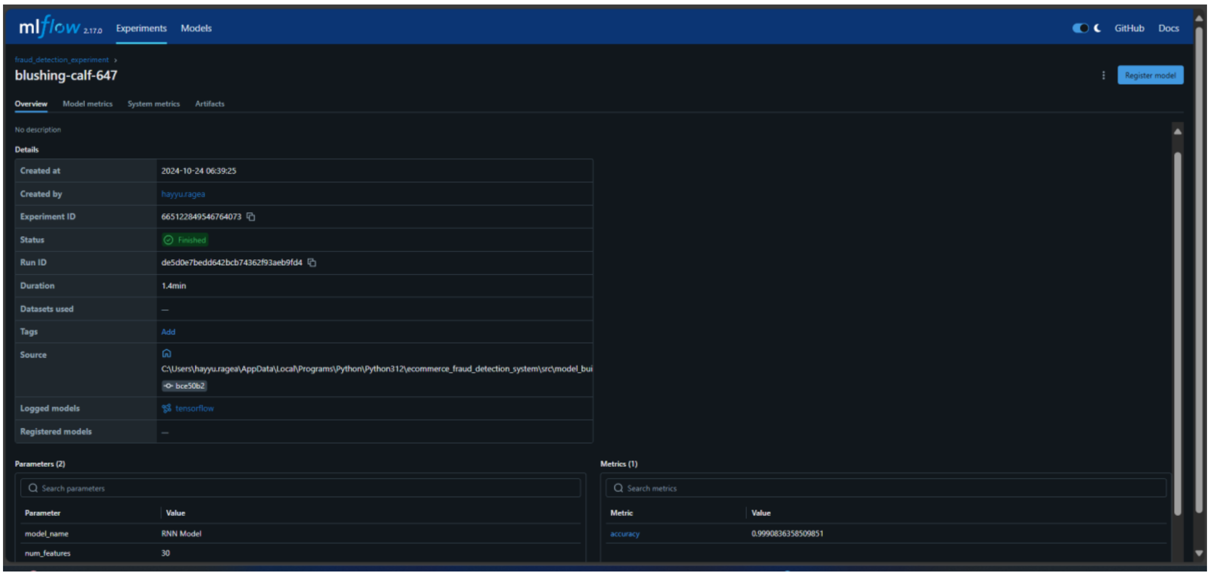
- Multi-Layer Perceptron (MLP)



- Long Short-Term Memory (LSTM)



- Recurrent Neural Network (RNN) screenshot



Model Evaluation Results

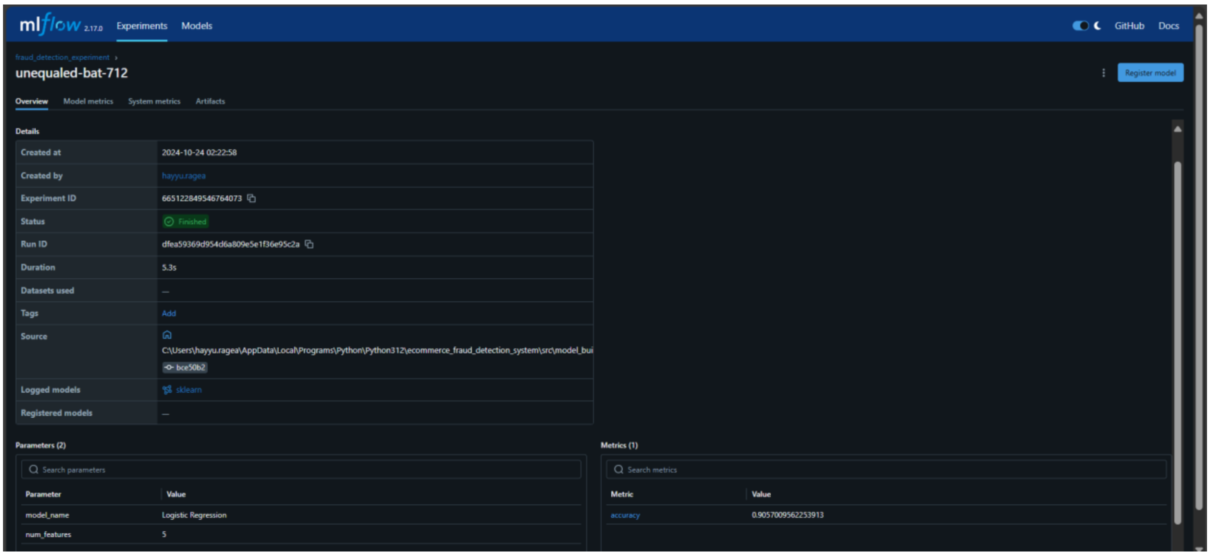
Evaluating Models for Credit Card Data

Model	Accuracy	Precision	Recall	F1-Score	Support
Logistic Regression	0.9991	1.00	0.54	0.66	90

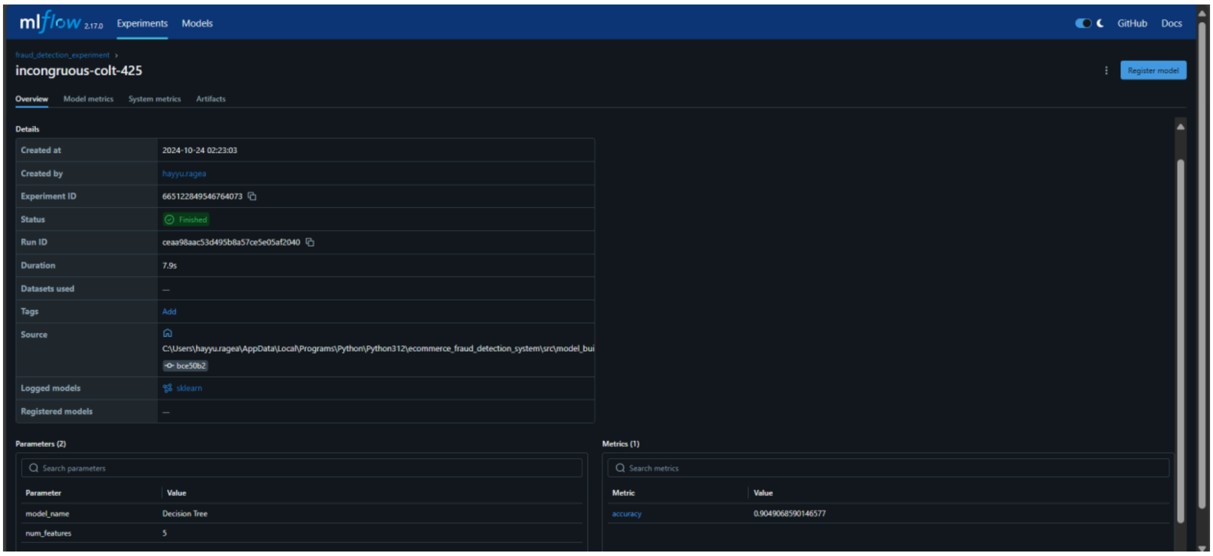
Model	Accuracy	Precision	Recall	F1-Score	Support
Decision Tree	0.9990	0.68	0.72	0.70	90
Random Forest	0.9996	0.99	0.73	0.84	90
Gradient Boosting	0.9993	0.89	0.63	0.74	90
MLP Classifier	0.9982	0.47	0.78	0.58	90
RNN	0.9986	0.91	0.11	0.20	90
LSTM	0.]9992	0.76	0.77	0.76	90

Evaluating Models for Fraud Data MLFLOW

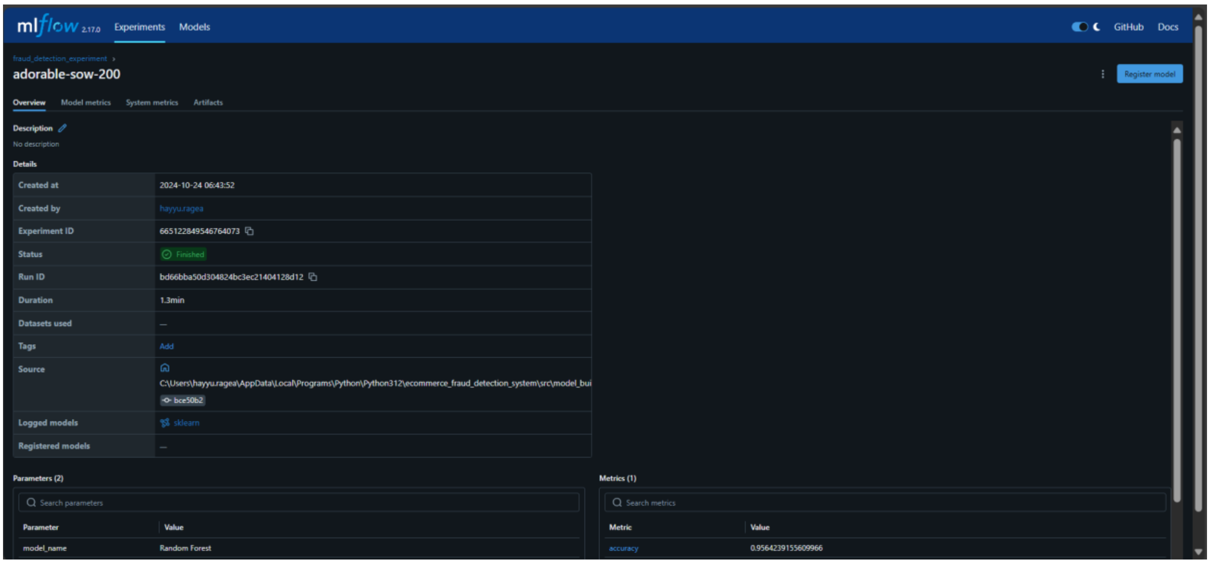
- Logistic Regression



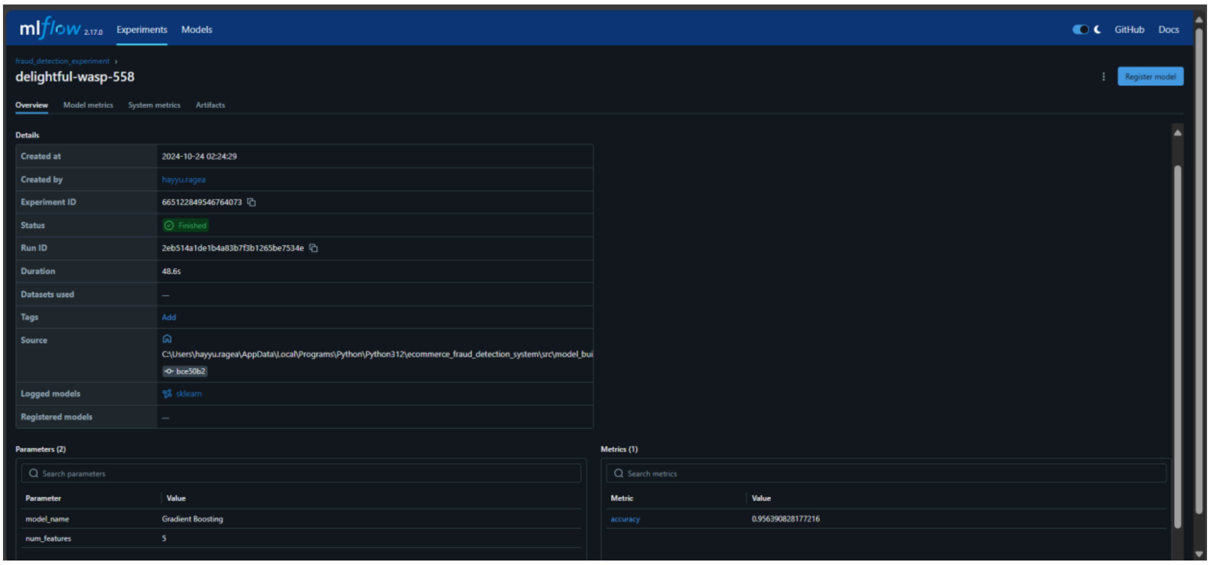
- Decision Tree model



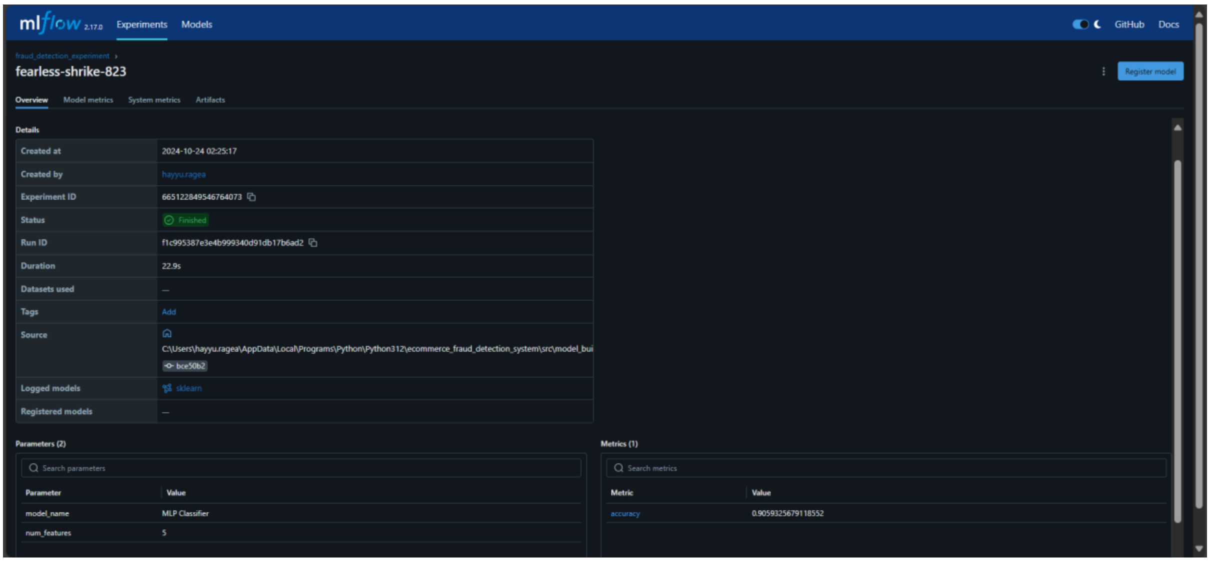
- Random forest



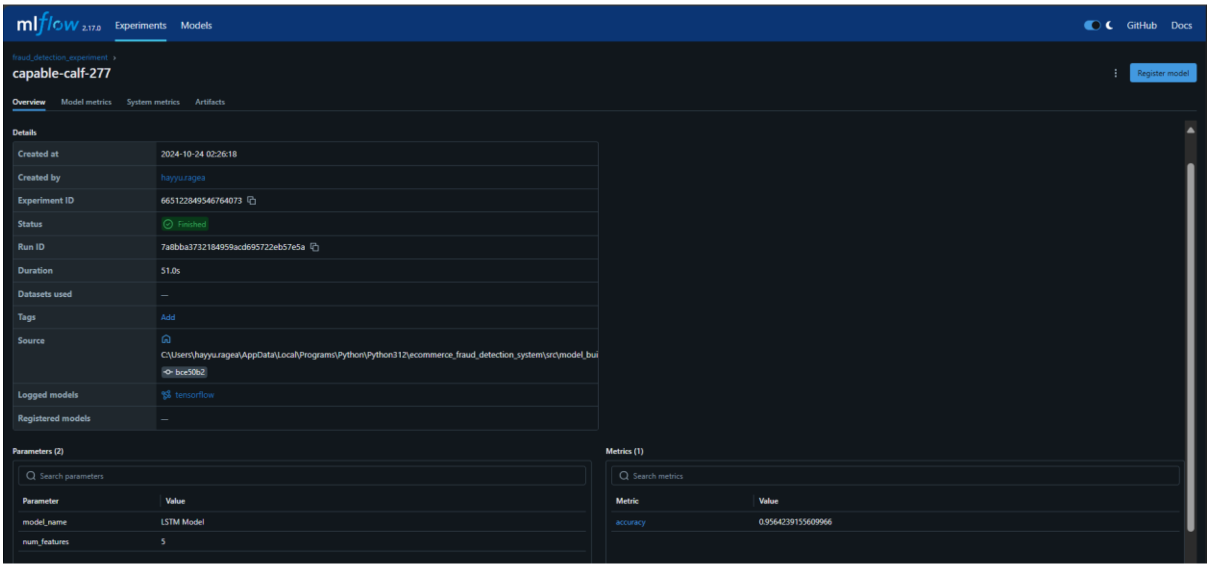
- Gradient Boosting MLFLOW



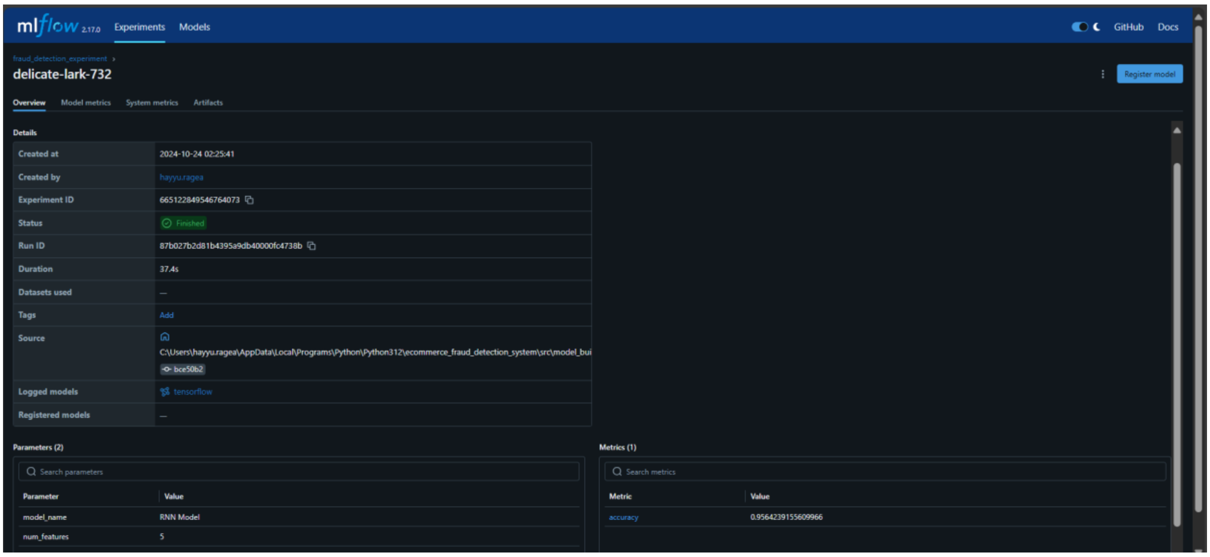
- Multi-Layer Perceptron (MLP)



- Long Short-Term Memory (LSTM)



- Recurrent Neural Network (RNN)



Evaluating Models for Fraud Data

Model	Accuracy	Precision	Recall	F1-Score	Support
Logistic Regression	0.9057	0.00	0.00	0.00	2850
Decision Tree	0.9063	0.50	0.56	0.53	2850

Model	Accuracy	Precision	Recall	F1-Score	Support
Random Forest	0.9564	1.00	0.54	0.70	2850
Gradient Boosting	0.9564	1.00	0.54	0.70	2850
MLP Classifier	0.6179	0.16	0.71	0.26	2850
RNN	0.9564	1.00	0.54	0.70	2850

GitHublink:https://github.com/HaYyu-Ra/ecommerce_fraud_detection_analysis/blob/master/notebooks/model_biulding.ipynb

Conclusion

The models were trained and evaluated on both datasets, revealing key insights for fraud detection in e-commerce and credit card transactions. Random Forest, Gradient Boosting, and Logistic Regression delivered strong performances. Continued refinement is recommended, including addressing convergence warnings and model optimization.

This project has positioned Adey Innovations Inc. to enhance transaction security and build customer trust through advanced fraud detection technologies.