**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()

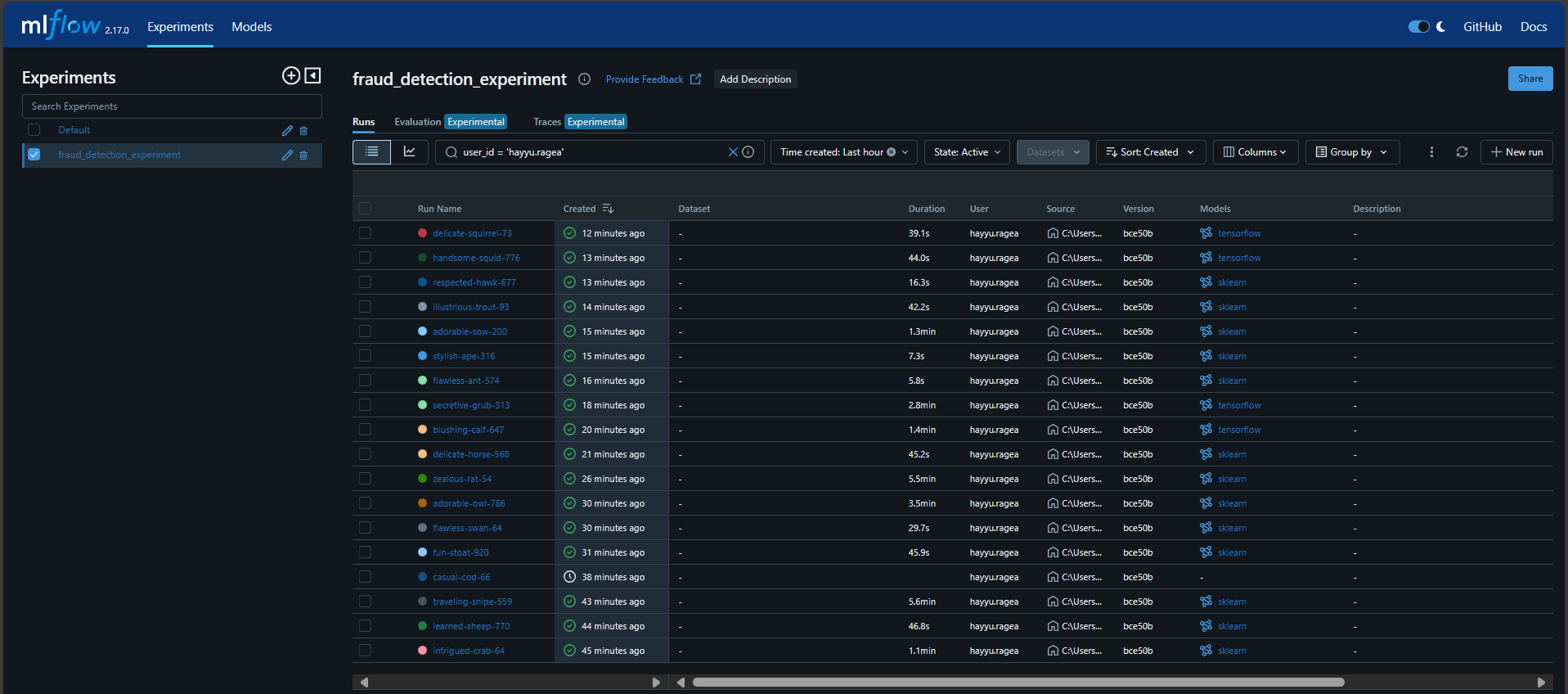
for name, model in models.items():

mlflow.log\_param("model\_name", name)

mlflow.log\_metric("accuracy", results\_fraud[name]['Accuracy'])

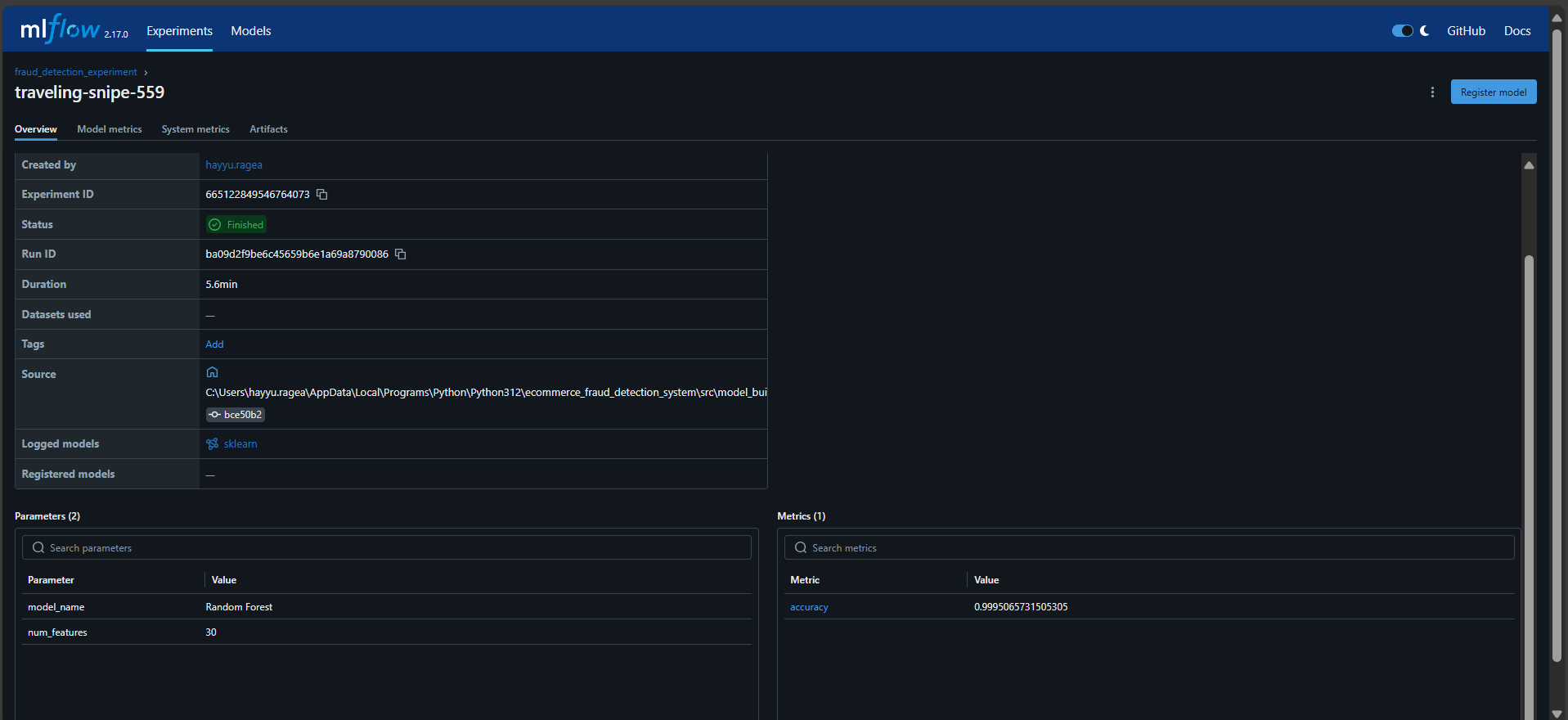
mlflow.sklearn.log\_model(model, name)

mlflow.end\_run()

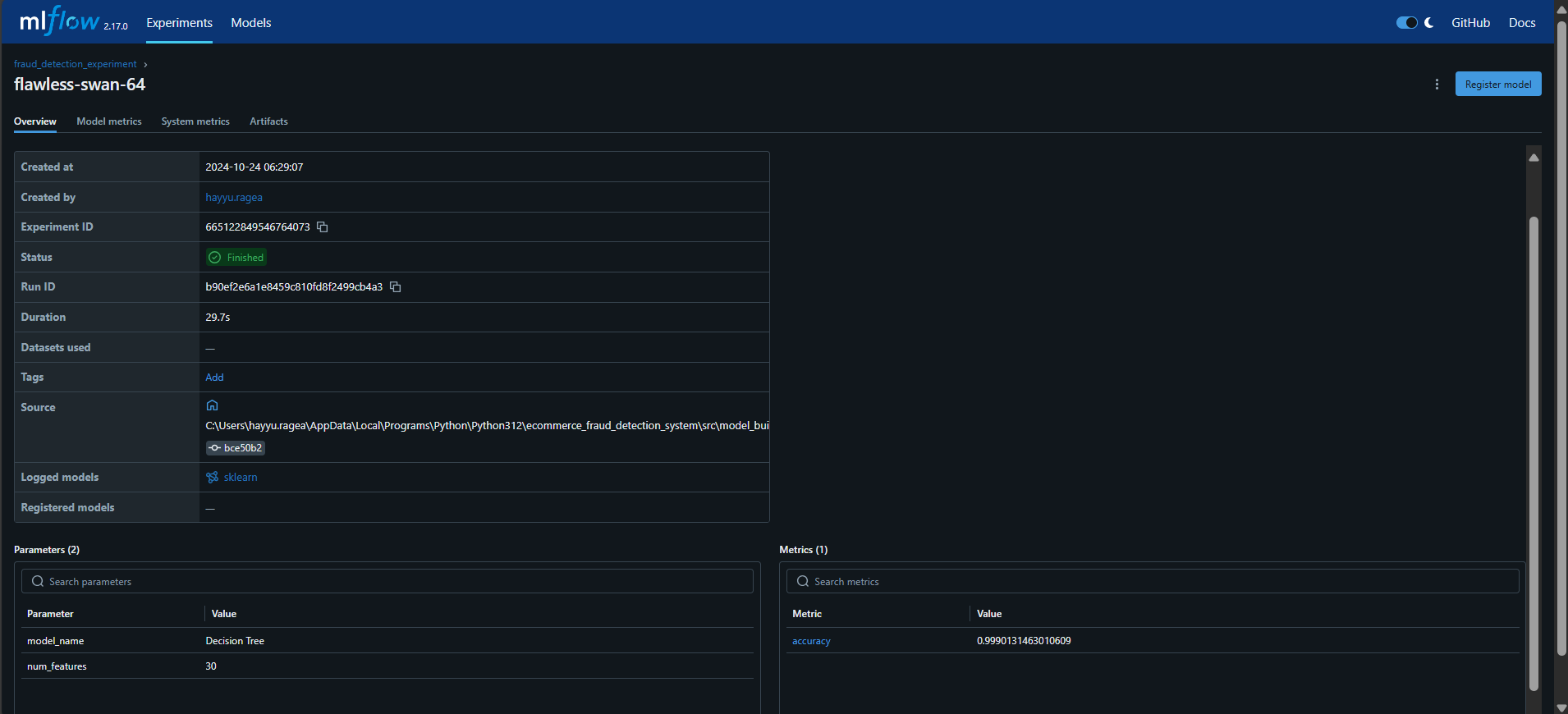


**Evaluating Models for Credit Card Data MLFLOW**

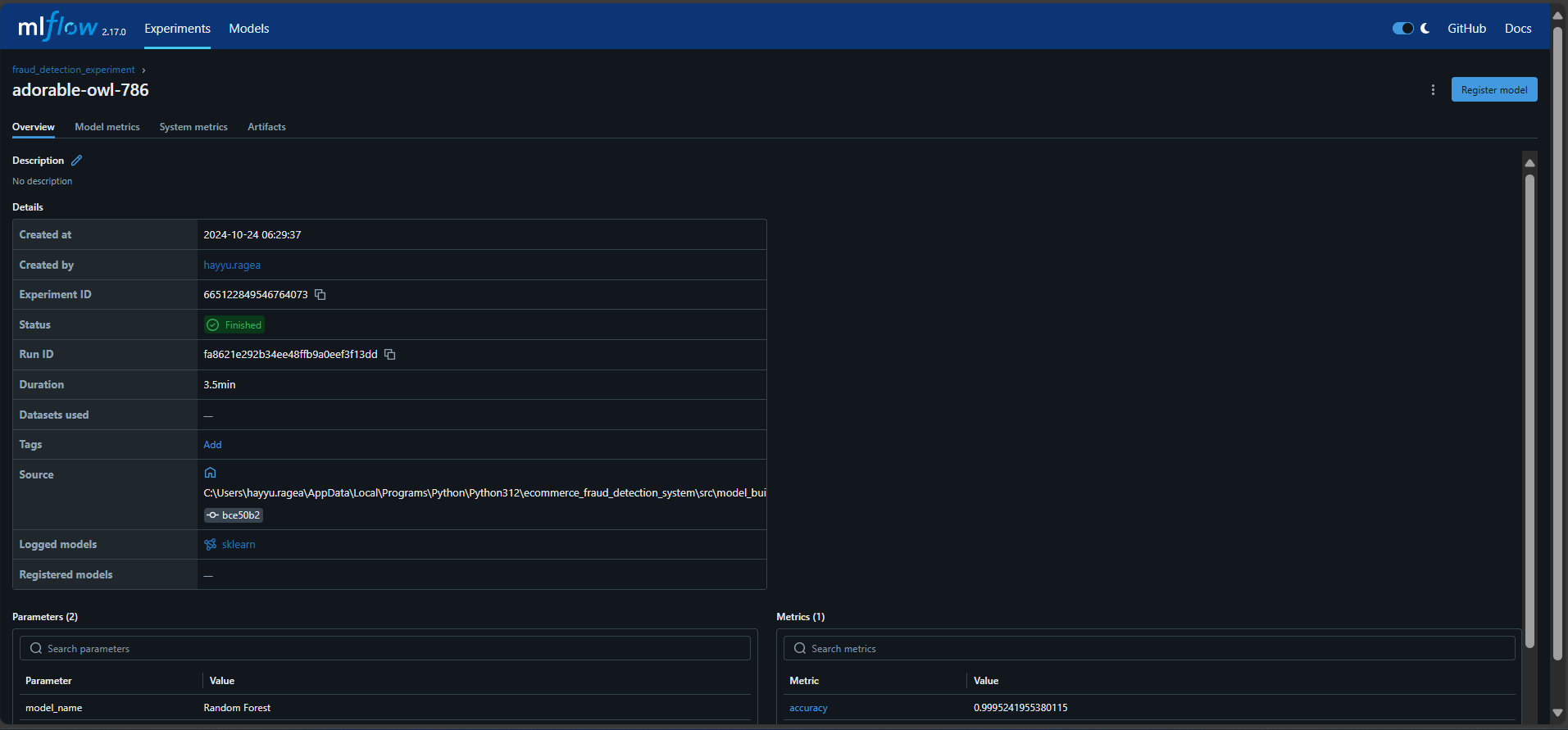
* Logistic regression



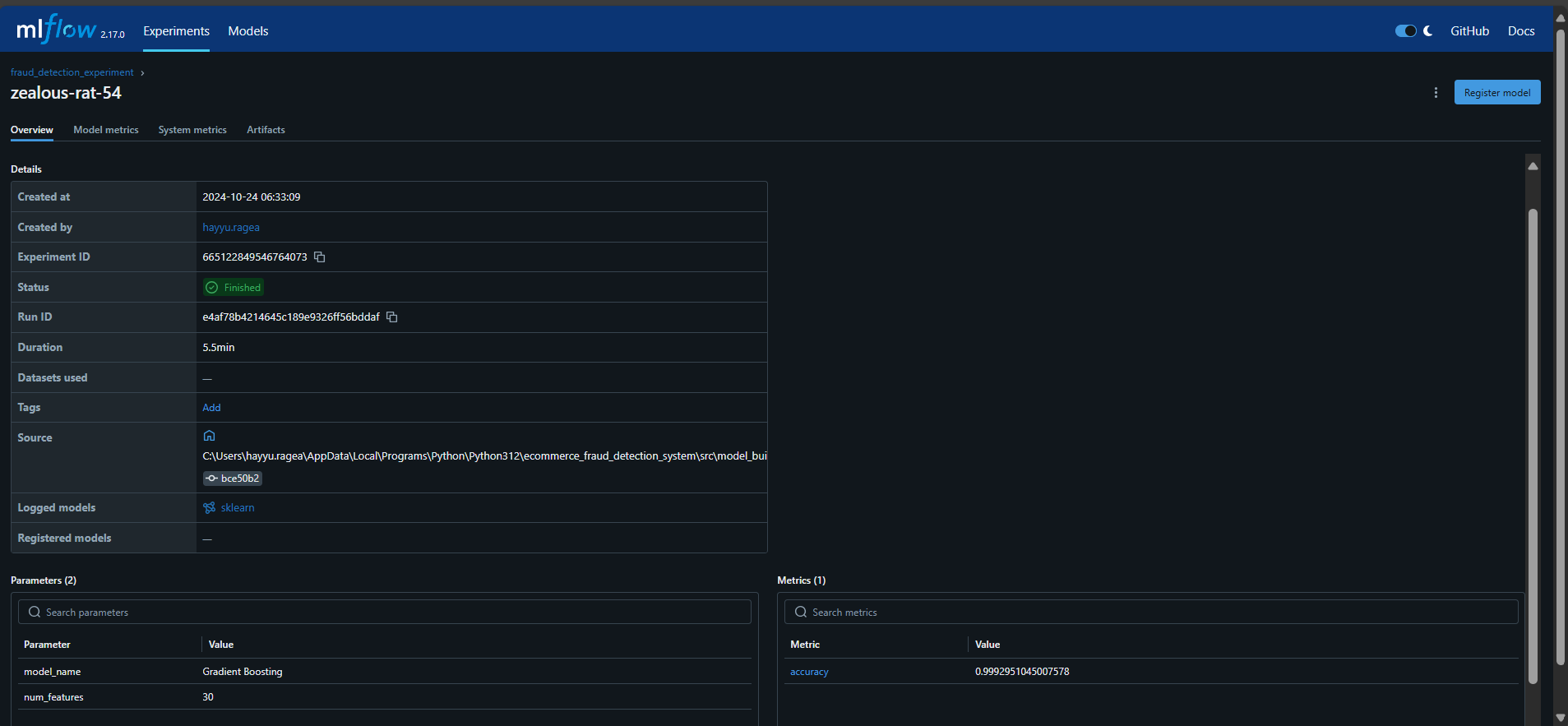
* Decision Tree



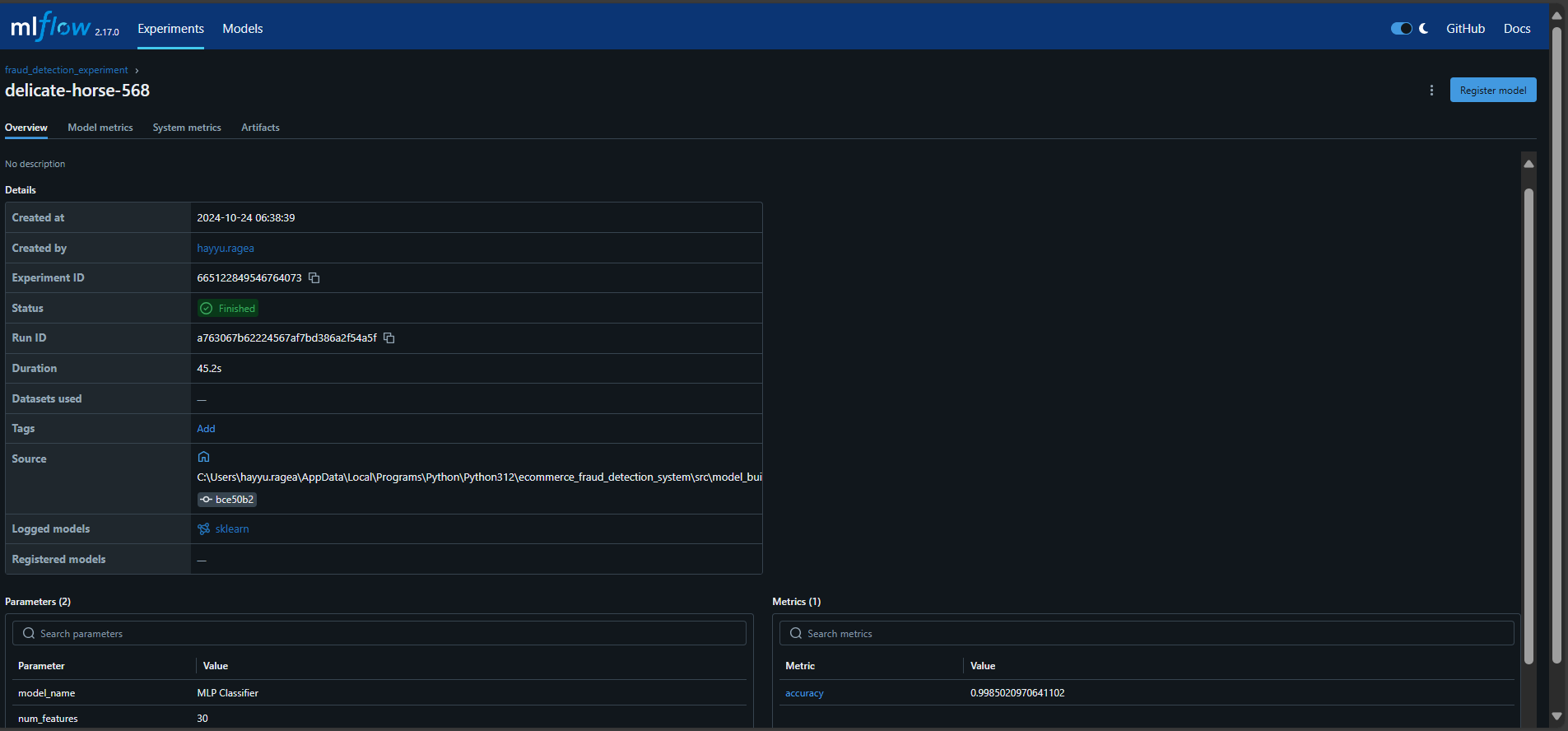
* Random Forest



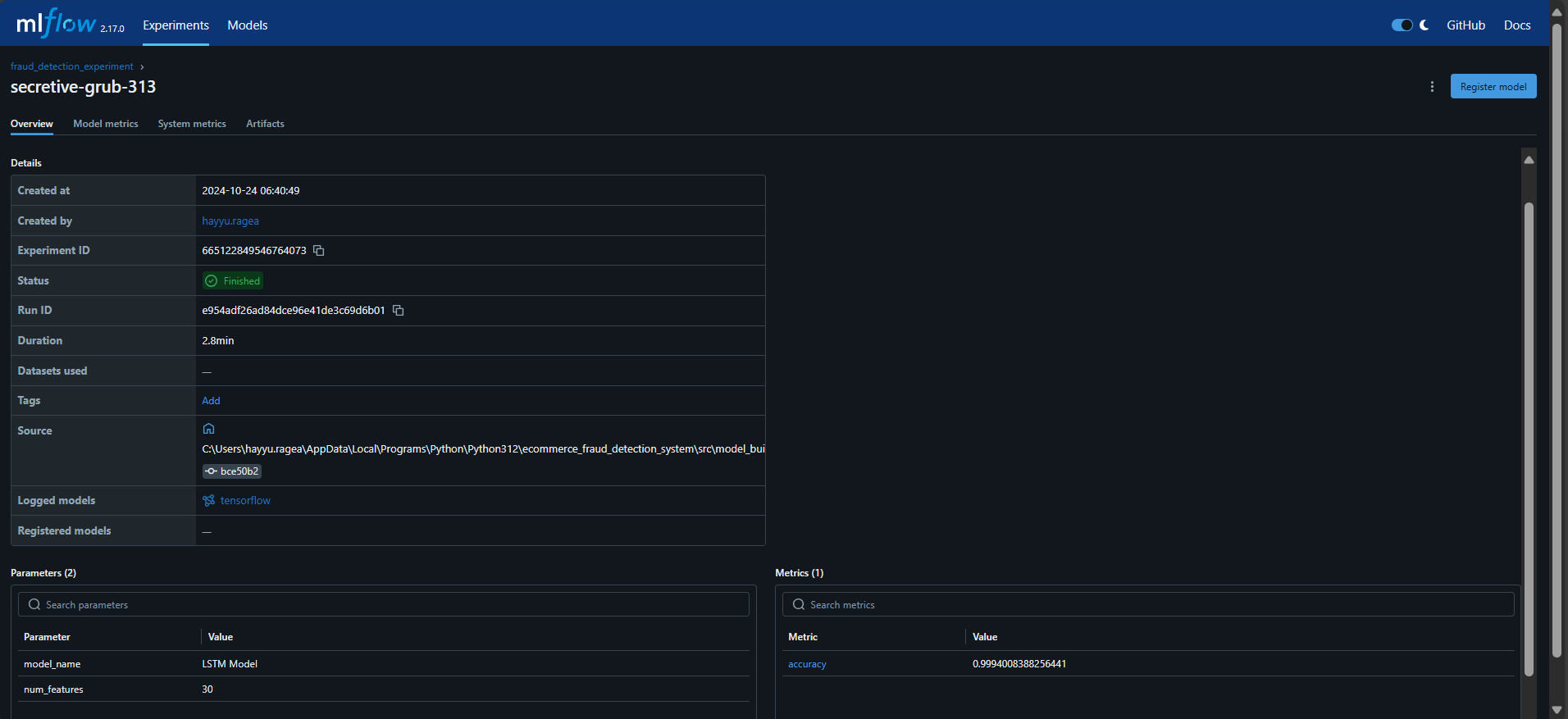
* Gradient Boosting



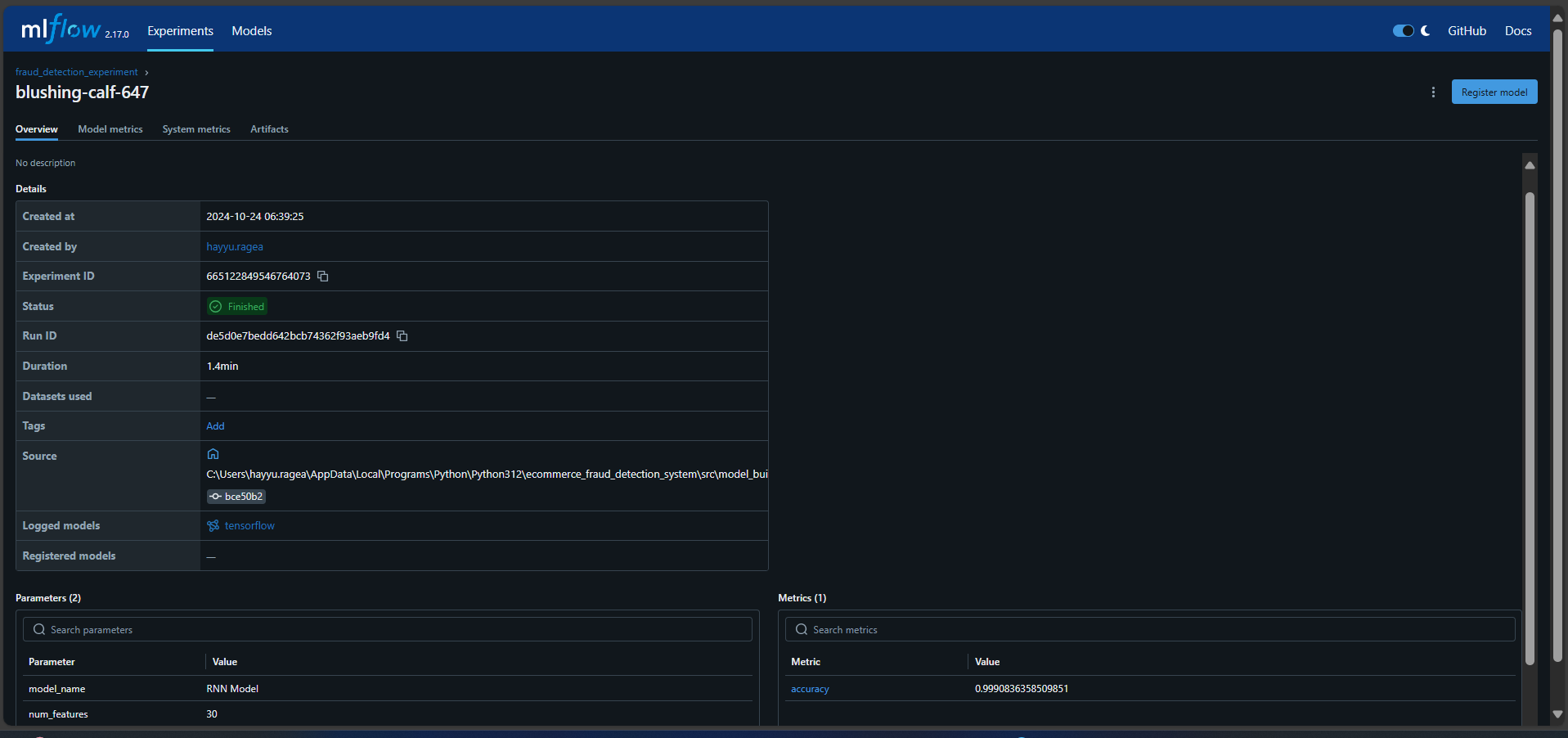
* Multi-Layer Perceptron (MLP)

****

* Long Short-Term Memory (LSTM)

****

* Recurrent Neural Network (RNN) screenshot

****

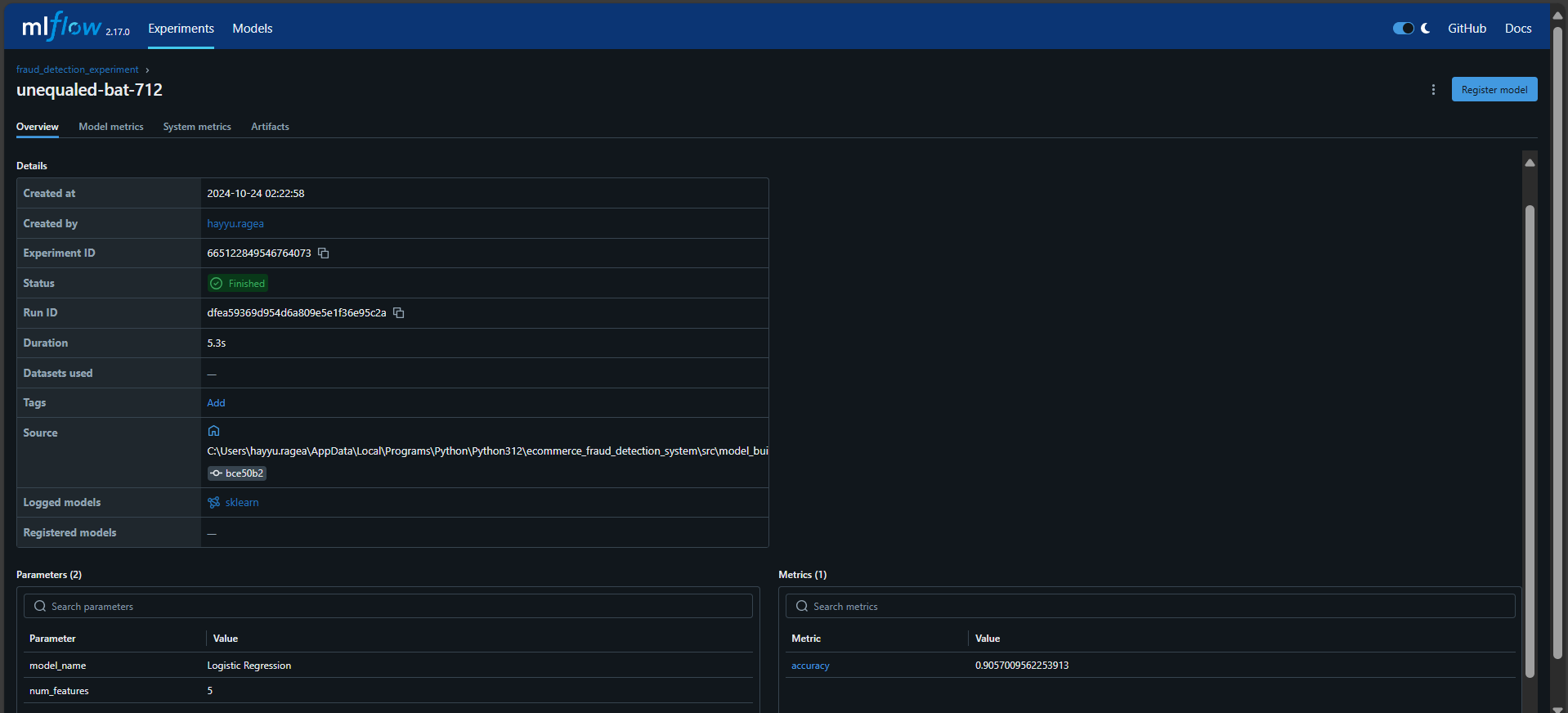
**Model Evaluation Results**

**Evaluating Models for Credit Card Data**

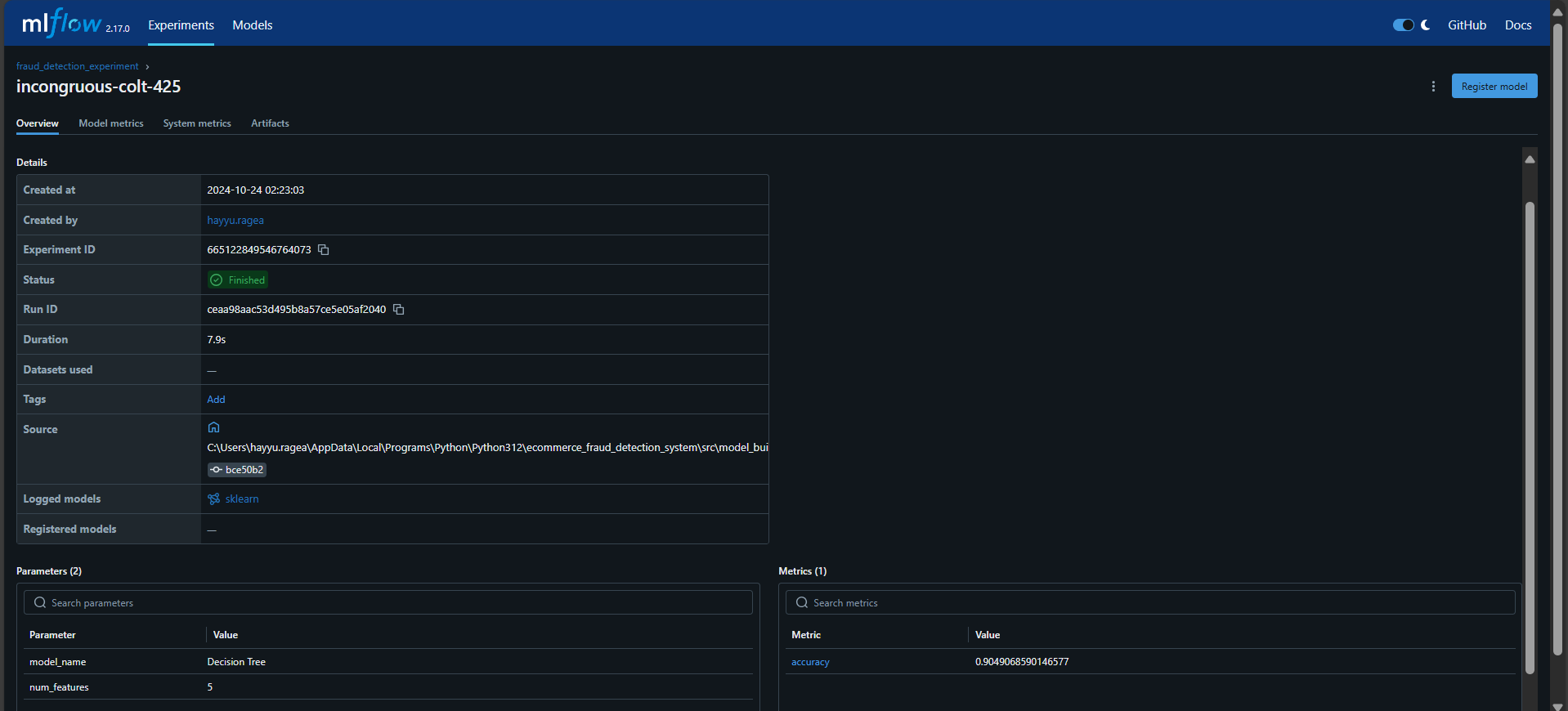
| Model | Accuracy | Precision | Recall | F1-Score | Support |
| --- | --- | --- | --- | --- | --- |
| Logistic Regression | 0.9991 | 1.00 | 0.54 | 0.66 | 90 |
| Decision Tree | 0.9990 | 0.68 | 0.72 | 0.70 | 90 |
| Random Forest | 0.9996 | 0.99 | 0.73 | 0.84 | 90 |
| Gradient Boosting | 0.9993 | 0.89 | 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**

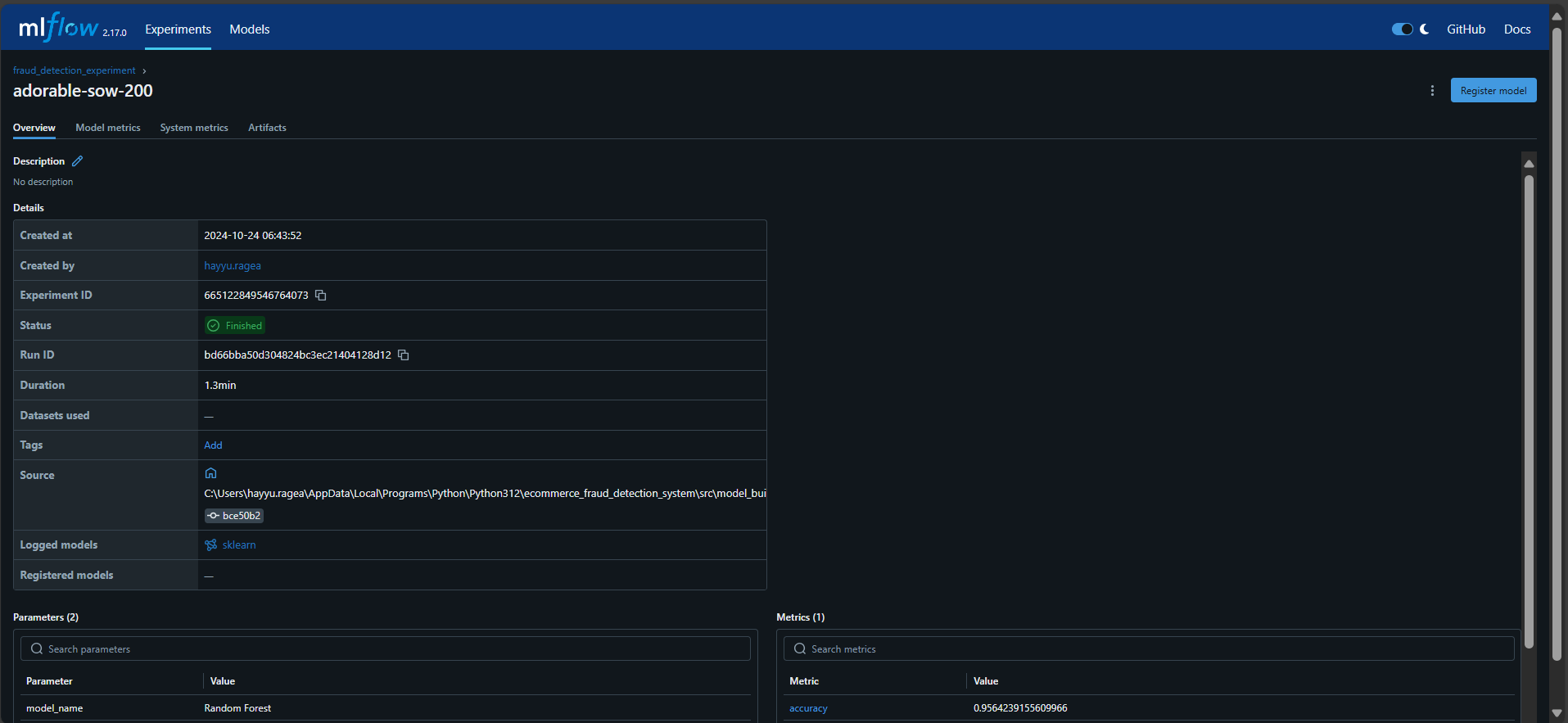
* Logistic Regression



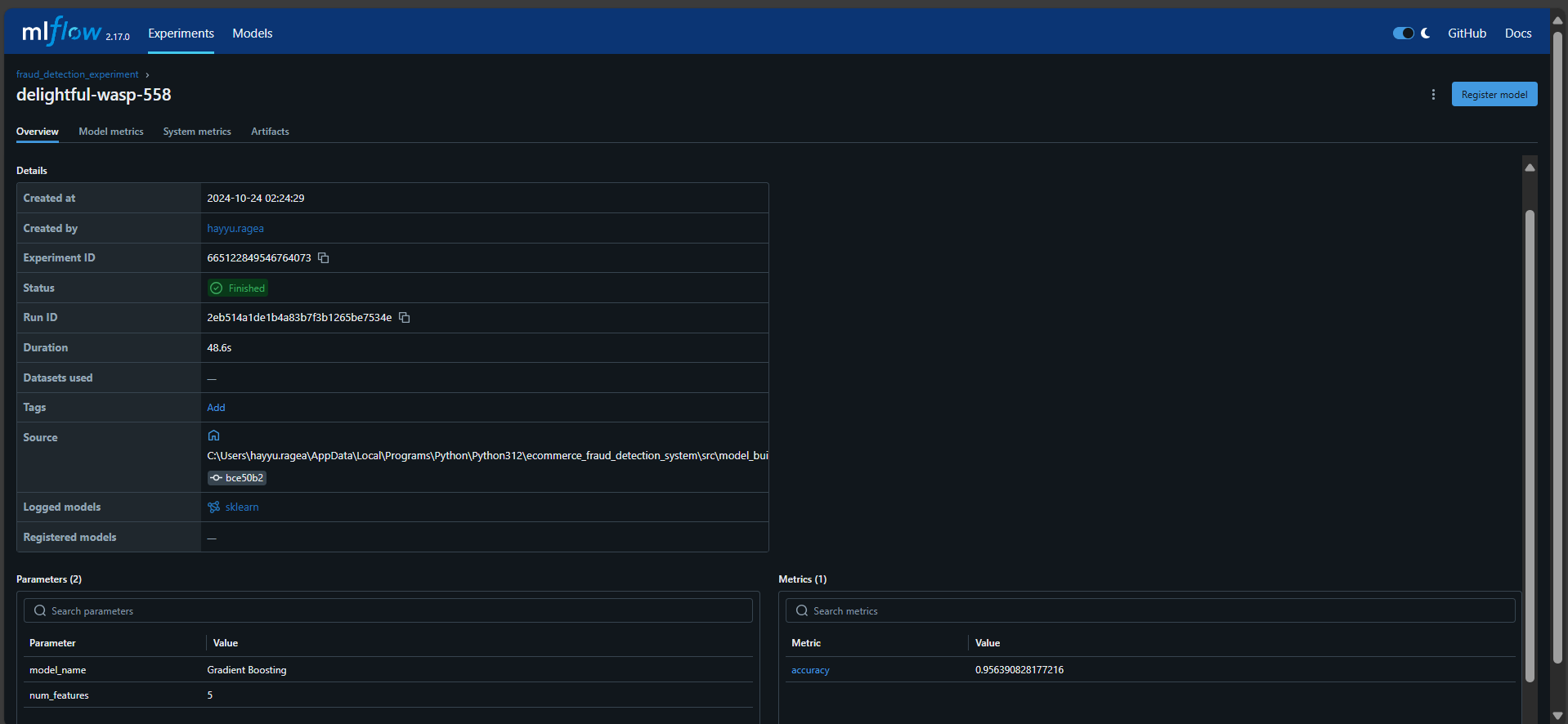
* Decision Tree model

****

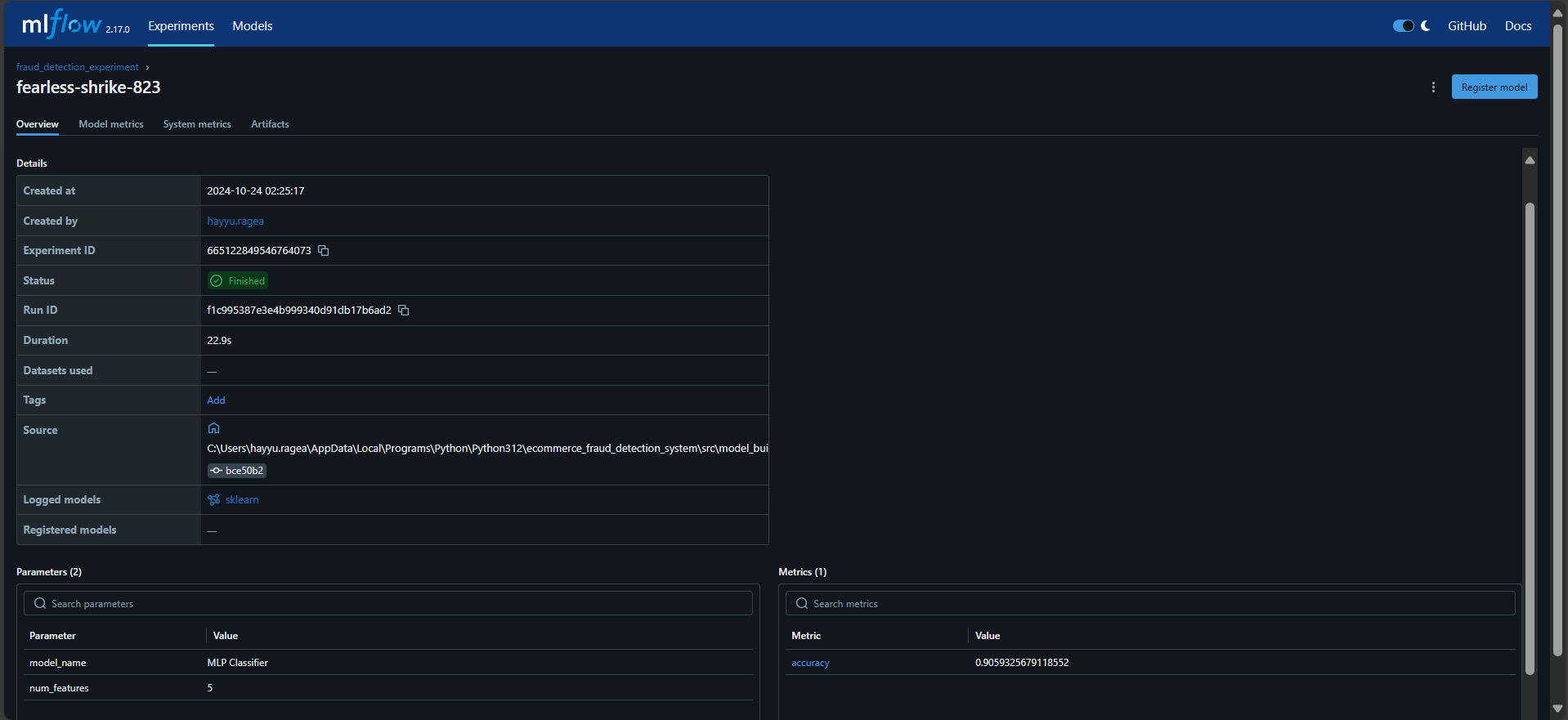
* **Random forest**

****

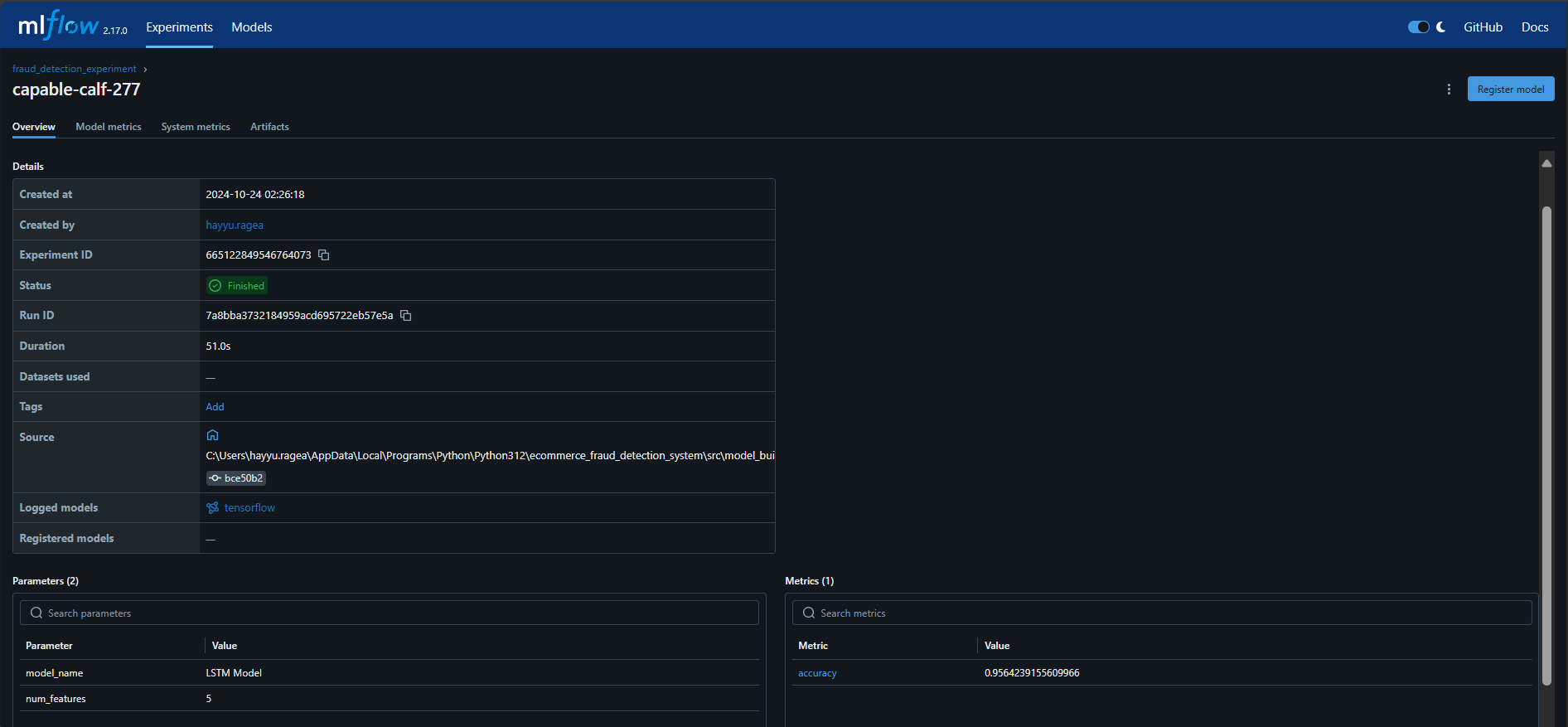
* **Gradient Boosting MLFLOW**



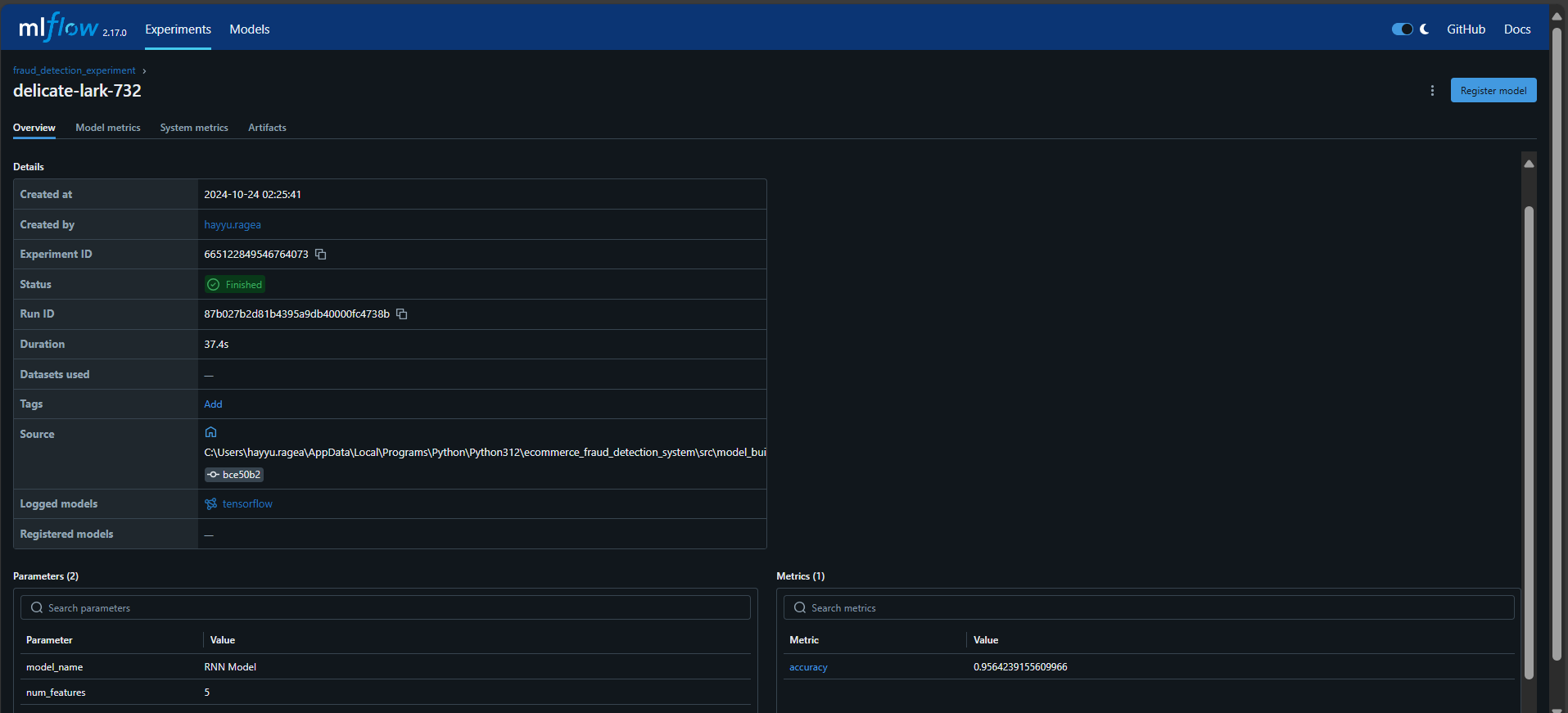
* Multi-Layer Perceptron (MLP)



* 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 |
| 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.