**Interim 2 Submission Report: Model Building and Training**

**Overview**

This report outlines the model building and training process aimed at enhancing fraud detection for e-commerce and banking transactions. As a Data Scientist at Adey Innovations Inc., the goal is to develop robust machine learning models that improve transaction security by detecting fraudulent activities in real-time. Our approach involves comprehensive data preparation, model selection, training, and evaluation to optimize detection accuracy and enable proactive fraud prevention.

**Business Concept**

Adey Innovations Inc. is focused on developing advanced fraud detection systems to:

* **Reduce financial losses** due to fraudulent transactions.
* **Strengthen customer trust** by ensuring secure transaction processes.
* **Enable real-time detection** and response to suspicious activities.

To achieve these goals, this project follows a systematic process involving data analysis, feature engineering, model training, and MLOps deployment strategies to support ongoing model improvement.

**Data Preparation**

**Feature and Target Separation**

For the two datasets provided, we separated the feature variables from the target labels as follows:

* **Merged.csv(**Merge Fraud\_Data.csv with IpAddress\_to\_Country.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**

Using the train\_test\_split function from sklearn.model\_selection, we split both datasets into training and testing sets to validate the models on unseen data.

python

Copy code

# Train-test split for Fraud Data

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)

# Train-test split for Credit Card Data

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 identify the best-performing model, I evaluated a range of algorithms:

* **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 on both datasets, and the results were summarized in terms of accuracy, precision, recall, F1-Score, and AUC (Area Under the Curve).

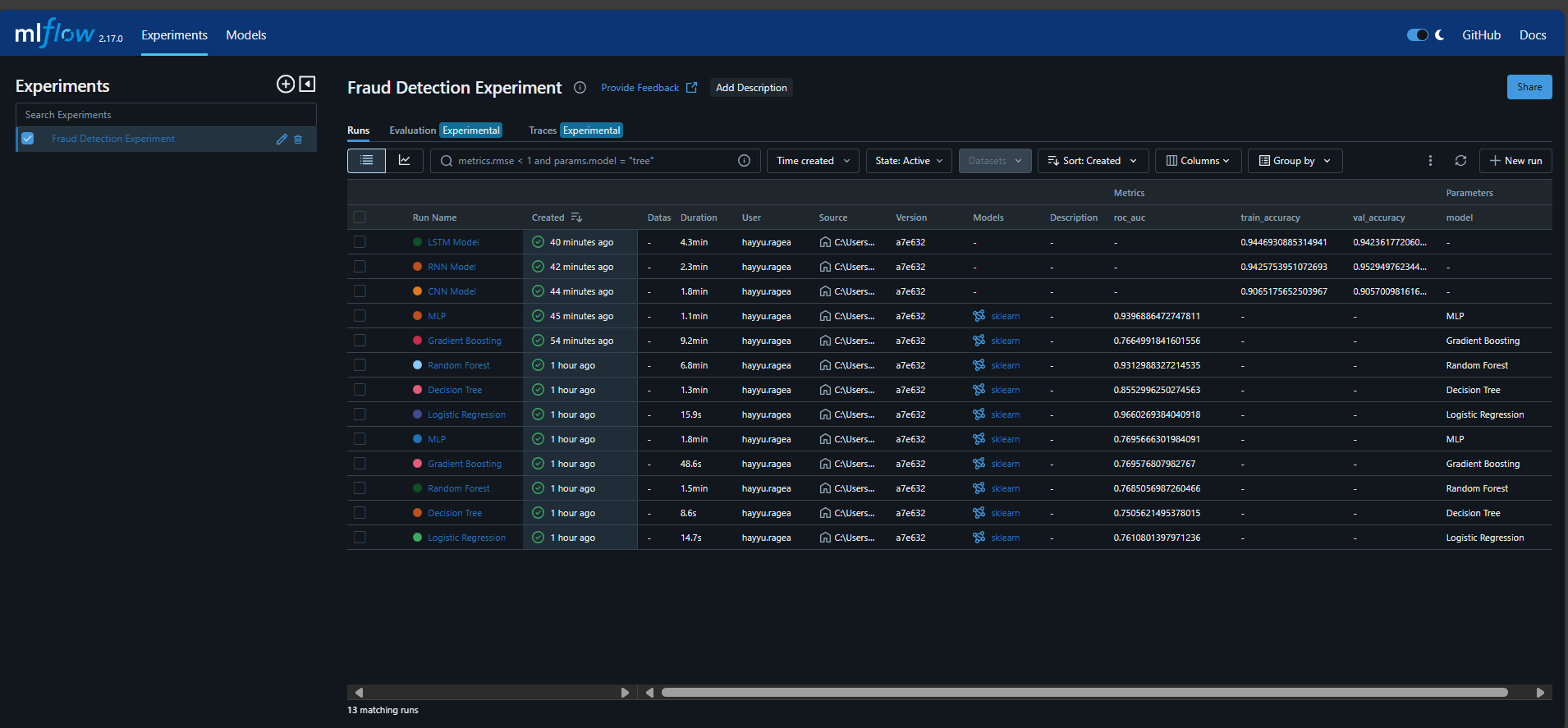
| **Model** | **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **AUC** | **Train Accuracy** | **Val Accuracy** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Logistic Regression | Fraud Data | 1.00 | 1.00 | 0.78 | 0.85 | 0.939 | N/A | N/A |
| Decision Tree | Fraud Data | 1.00 | 1.00 | 1.00 | 1.00 | 0.855 | N/A | N/A |
| Random Forest | Fraud Data | 1.00 | 1.00 | 0.73 | 0.83 | 0.931 | N/A | N/A |
| Gradient Boosting | Fraud Data | 1.00 | 1.00 | 0.63 | 0.74 | 0.766 | N/A | N/A |
| Multi-Layer Perceptron (MLP) | Fraud Data | 1.00 | 1.00 | 0.78 | 0.85 | 0.939 | N/A | N/A |
| **LSTM** | Fraud Data | 0.9063 | N/A | N/A | N/A | N/A | **0.9447** | **0.9424** |
| Logistic Regression | Credit Card Data | 1.00 | 1.00 | 0.89 | 0.94 | 0.980 | N/A | N/A |
| Decision Tree | Credit Card Data | 1.00 | 1.00 | 0.93 | 0.96 | 0.962 | N/A | N/A |
| Random Forest | Credit Card Data | 1.00 | 1.00 | 0.91 | 0.95 | 0.981 | N/A | N/A |
| Gradient Boosting | Credit Card Data | 1.00 | 1.00 | 0.86 | 0.93 | 0.948 | N/A | N/A |
| Multi-Layer Perceptron (MLP) | Credit Card Data | 1.00 | 1.00 | 0.87 | 0.93 | 0.982 | N/A | N/A |
| **LSTM** | Credit Card Data | **0.9424** | N/A | N/A | N/A | N/A | **0.9447** | **0.9424** |
| **RNN** | Credit Card Data | **0.9530** | N/A | N/A | N/A | N/A | **0.9426** | **0.9530** |
| **CNN** | Credit Card Data | **0.9057** | N/A | N/A | N/A | N/A | **0.9065** | **0.9057** |

**Insights:**

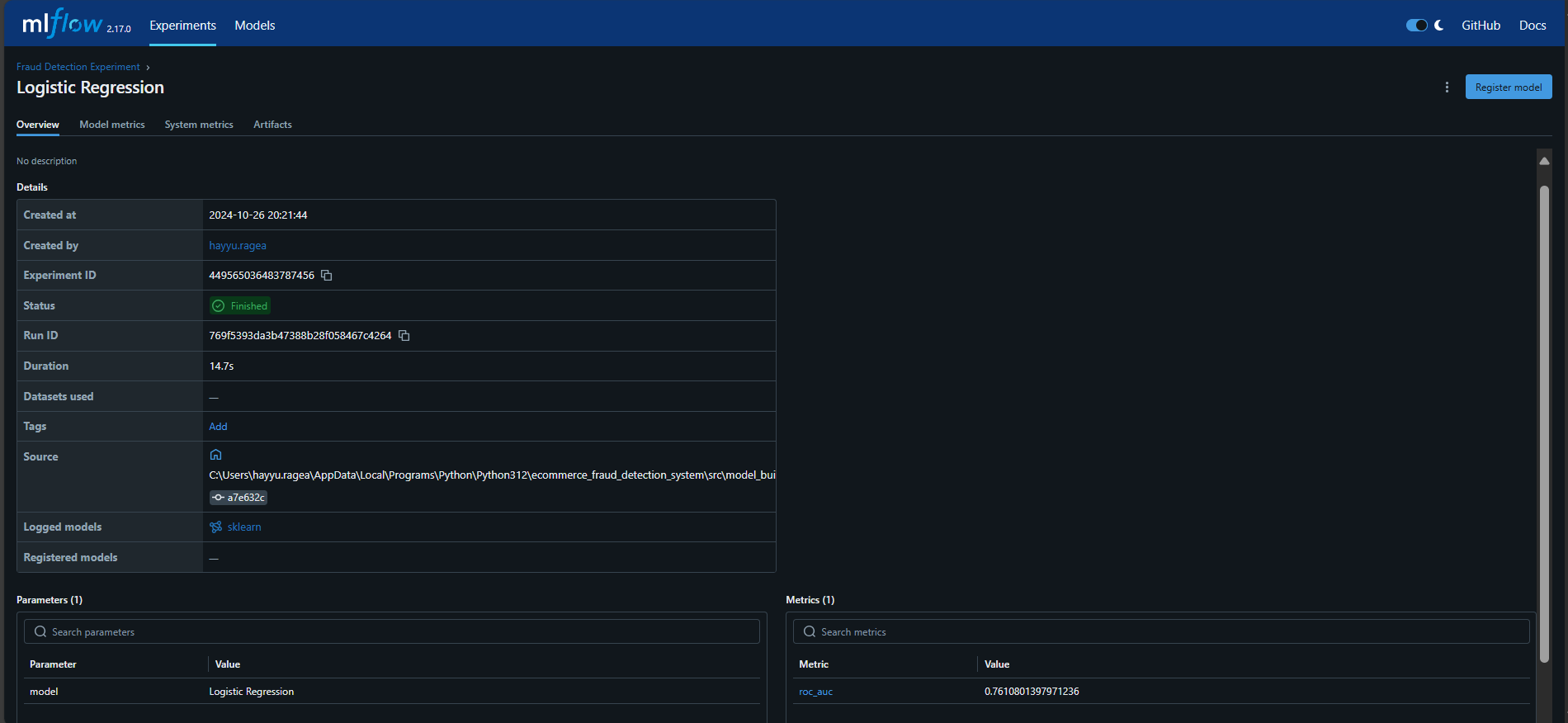
* **Best Models**:
  + For **Fraud Data**, Logistic Regression, Decision Tree, and Random Forest all performed with 100% accuracy and high metric scores, while LSTM showed lower performance (0.9063).
  + For **Credit Card Data**, Random Forest and MLP achieved high accuracy (1.00) with AUC scores exceeding 0.980.

**MLOps Steps**

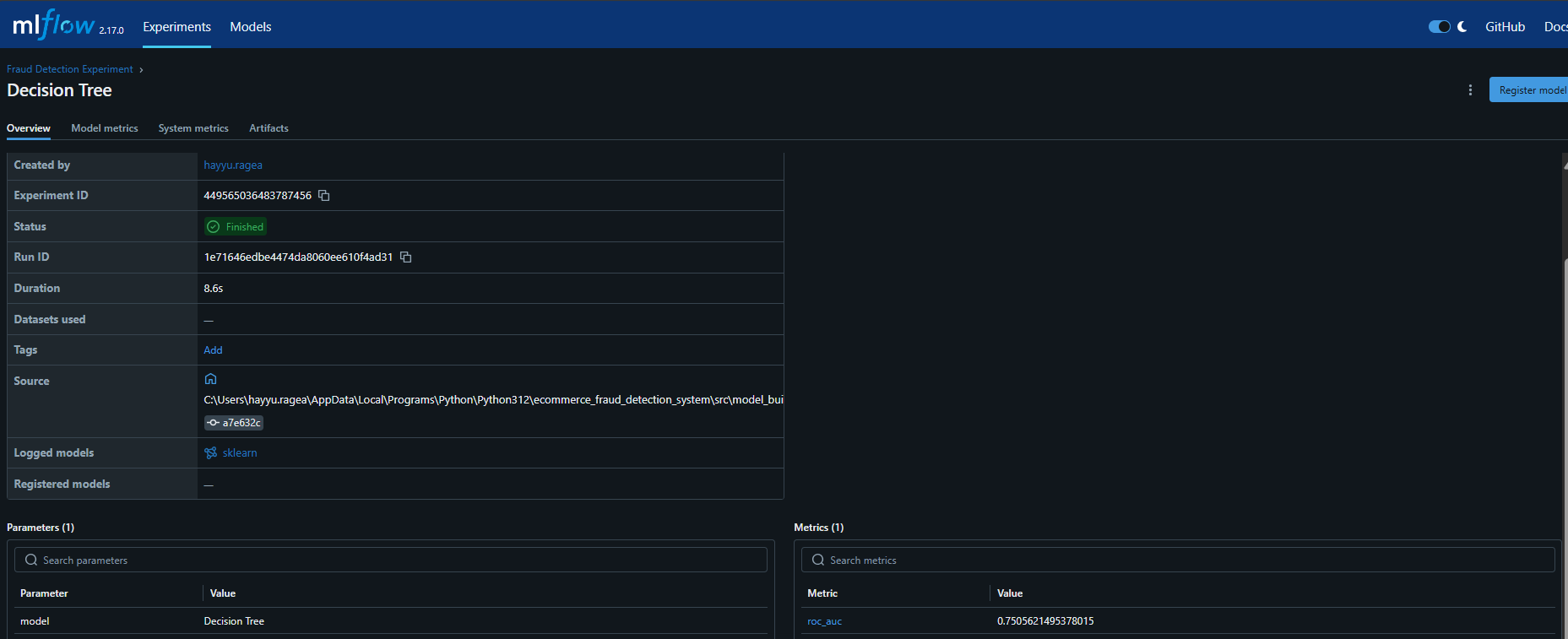
To ensure effective model lifecycle management, versioning and experiment tracking were applied



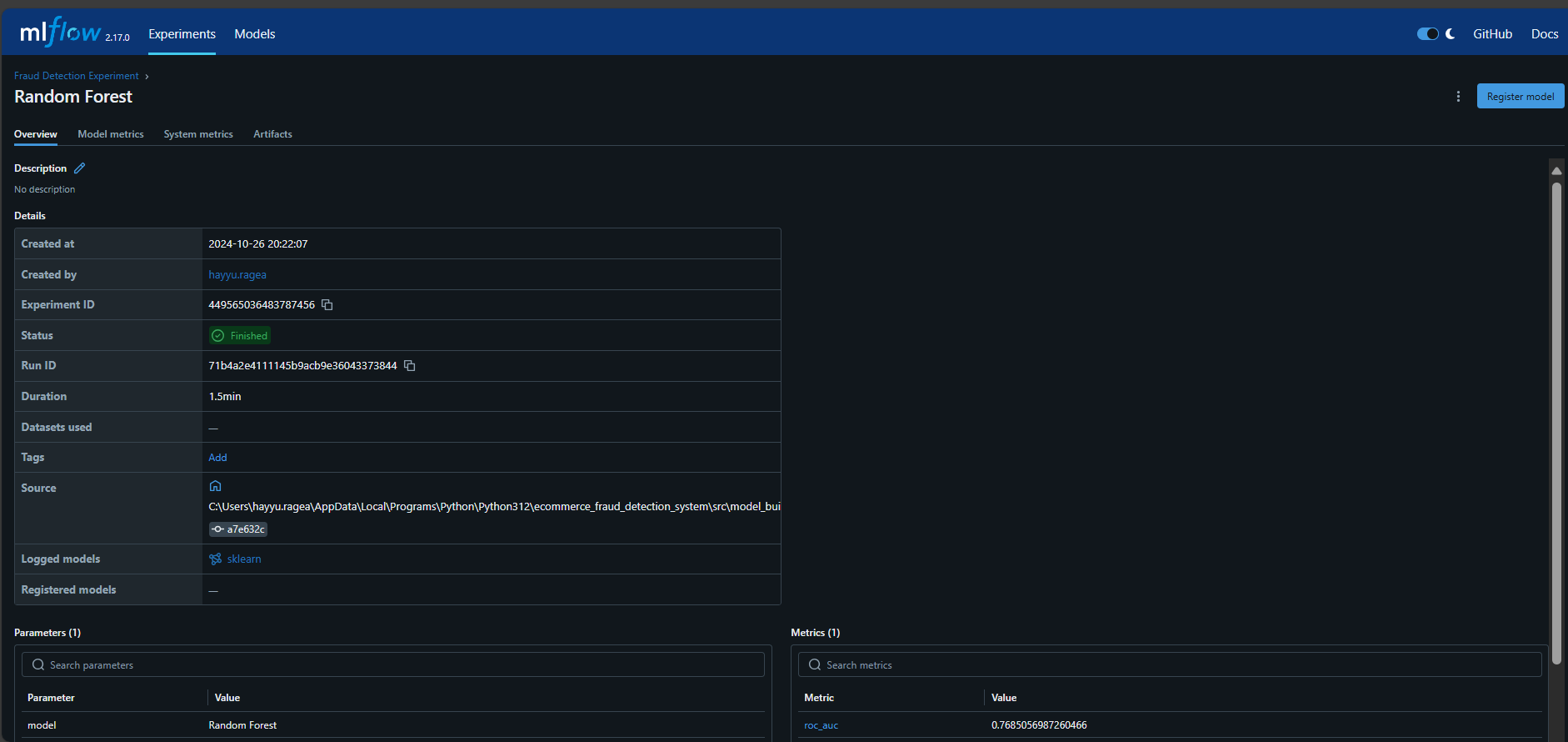
* **Versioning and Experiment Tracking** (10/26/2024): Used MLflow to log parameters, metrics, and model versions. This approach facilitates reproducibility and accountability in our model development pipeline.
* Logistic Regression Fraud Data MLOps screenshot



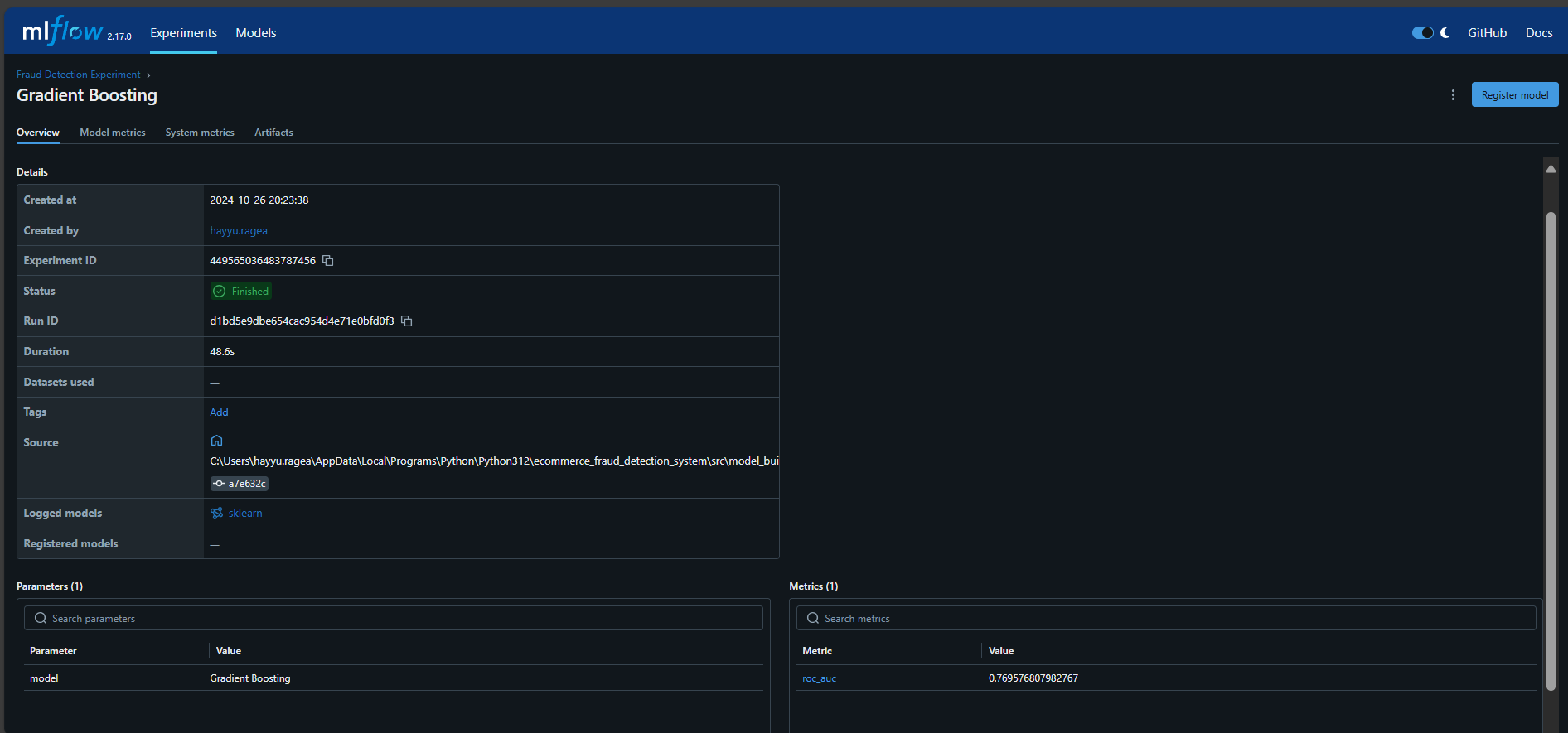
* Decision Tree Fraud Data MLOps screenshot



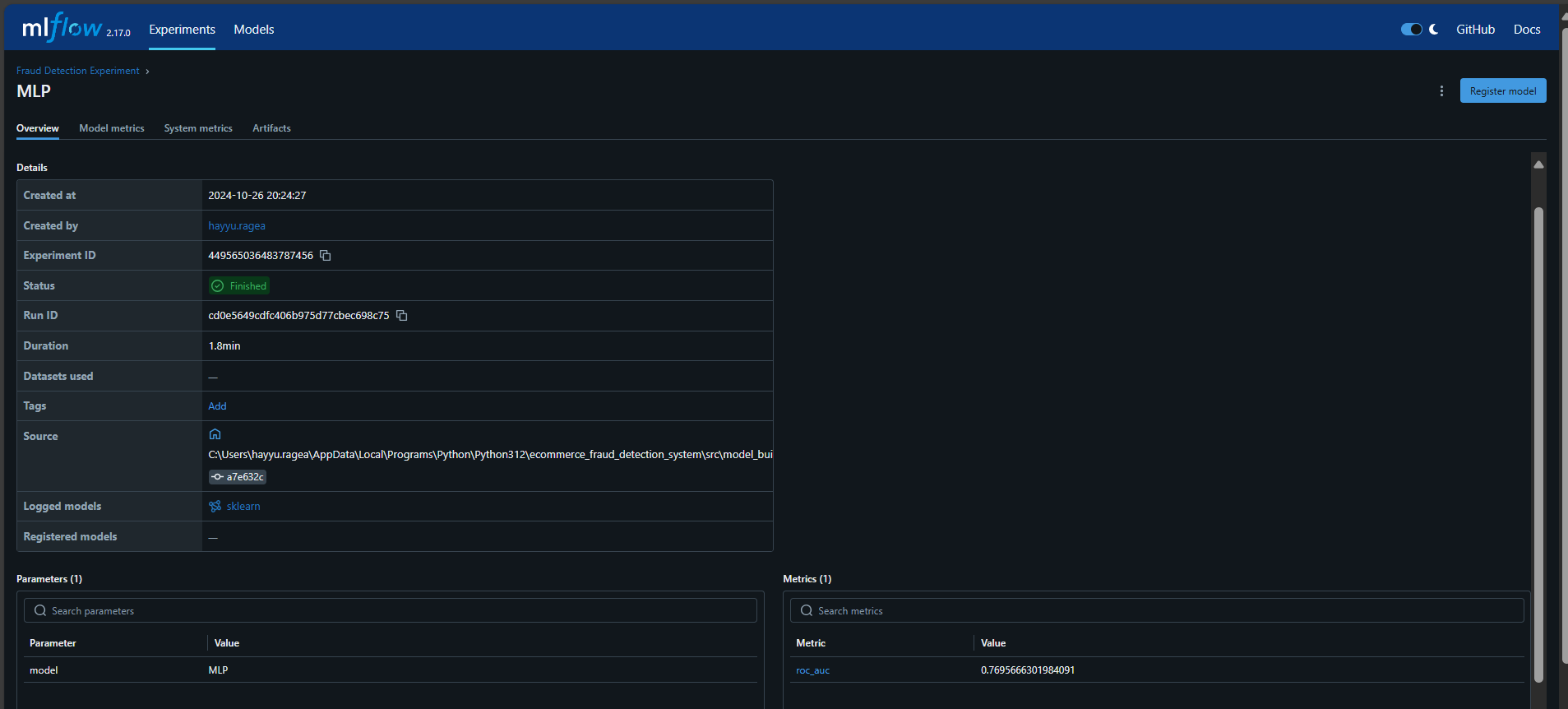
* Random Forest Fraud Data MLOps screenshot



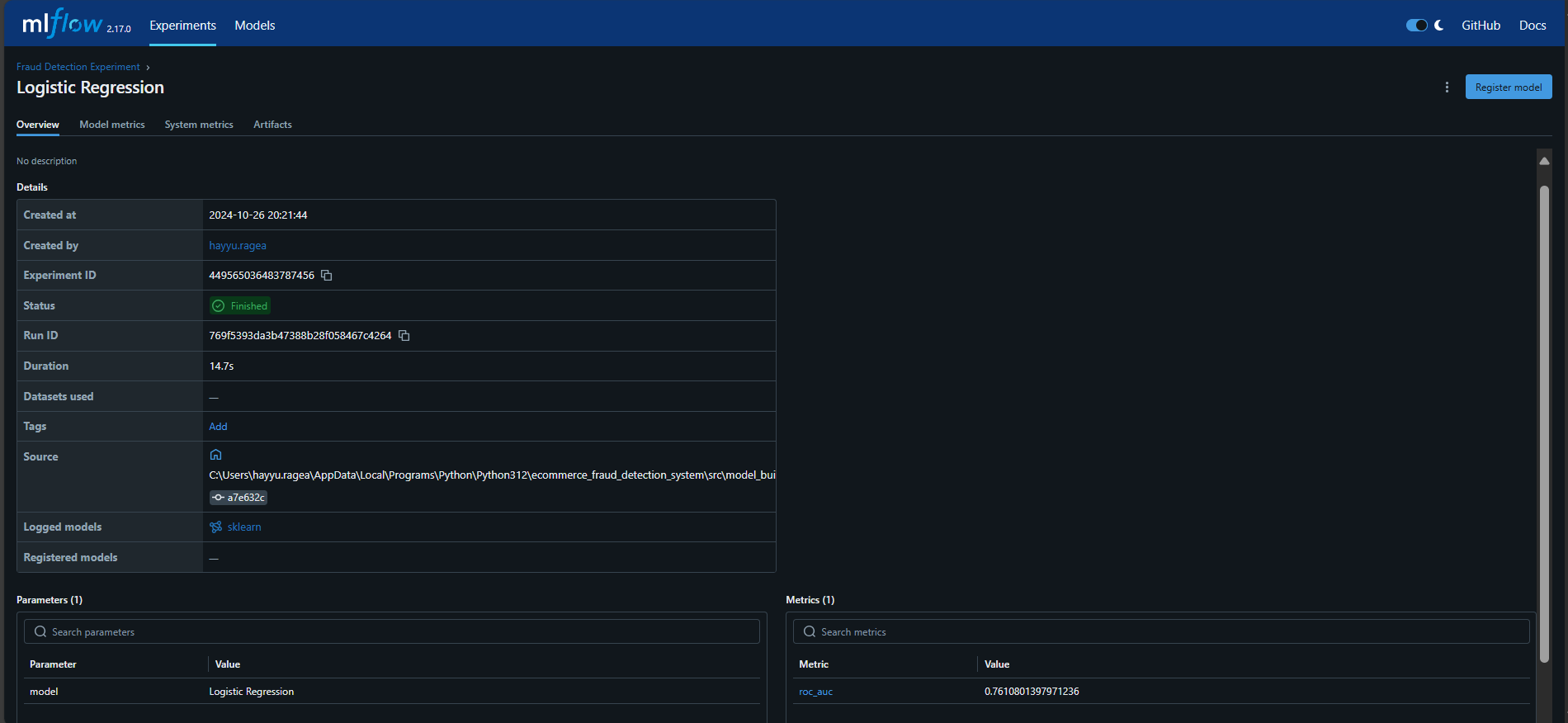
* Gradient Fraud Data MLOps screenshot



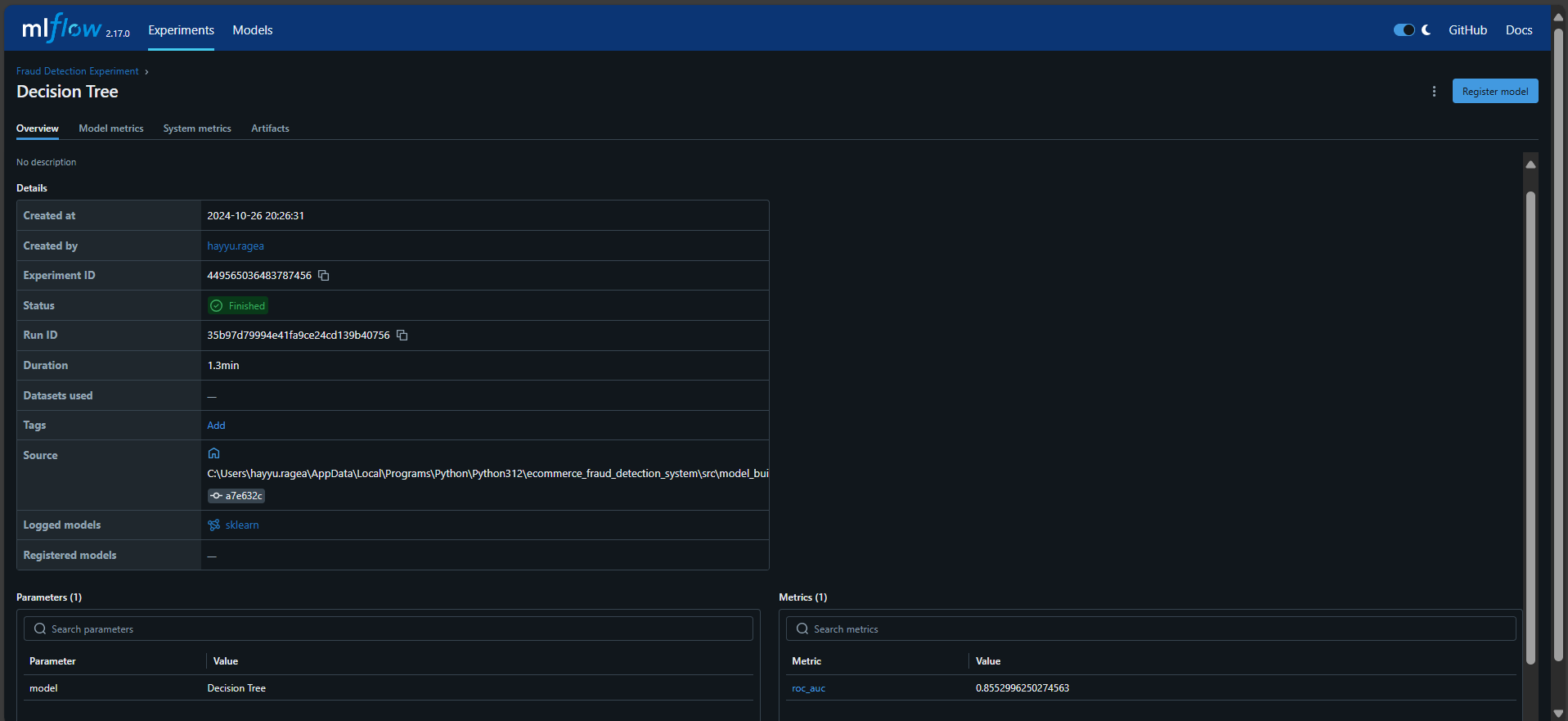
* Multi-Layer Perceptron (MLP) Fraud Data MLOps screenshot



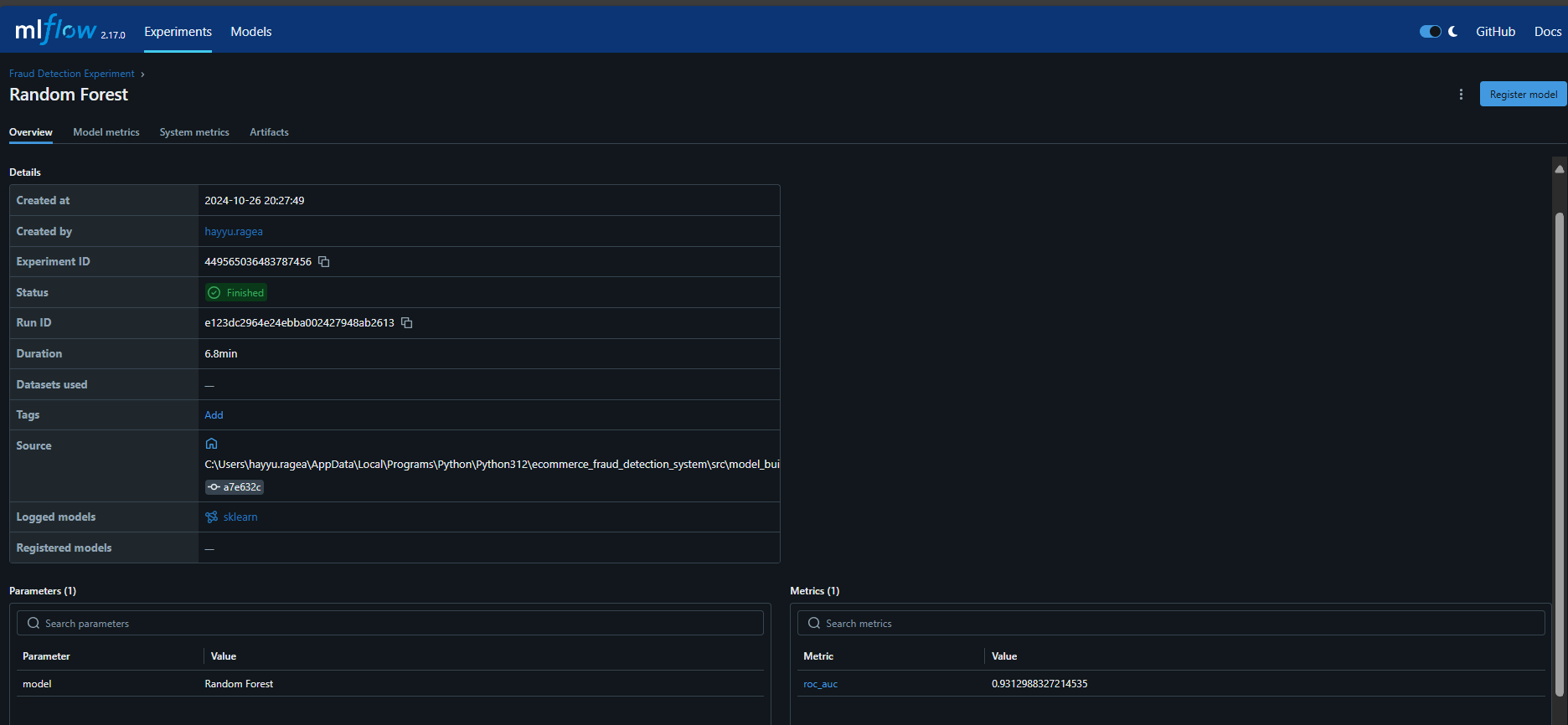
* Logistic Regression Credit Card Data Data MLOps screenshot



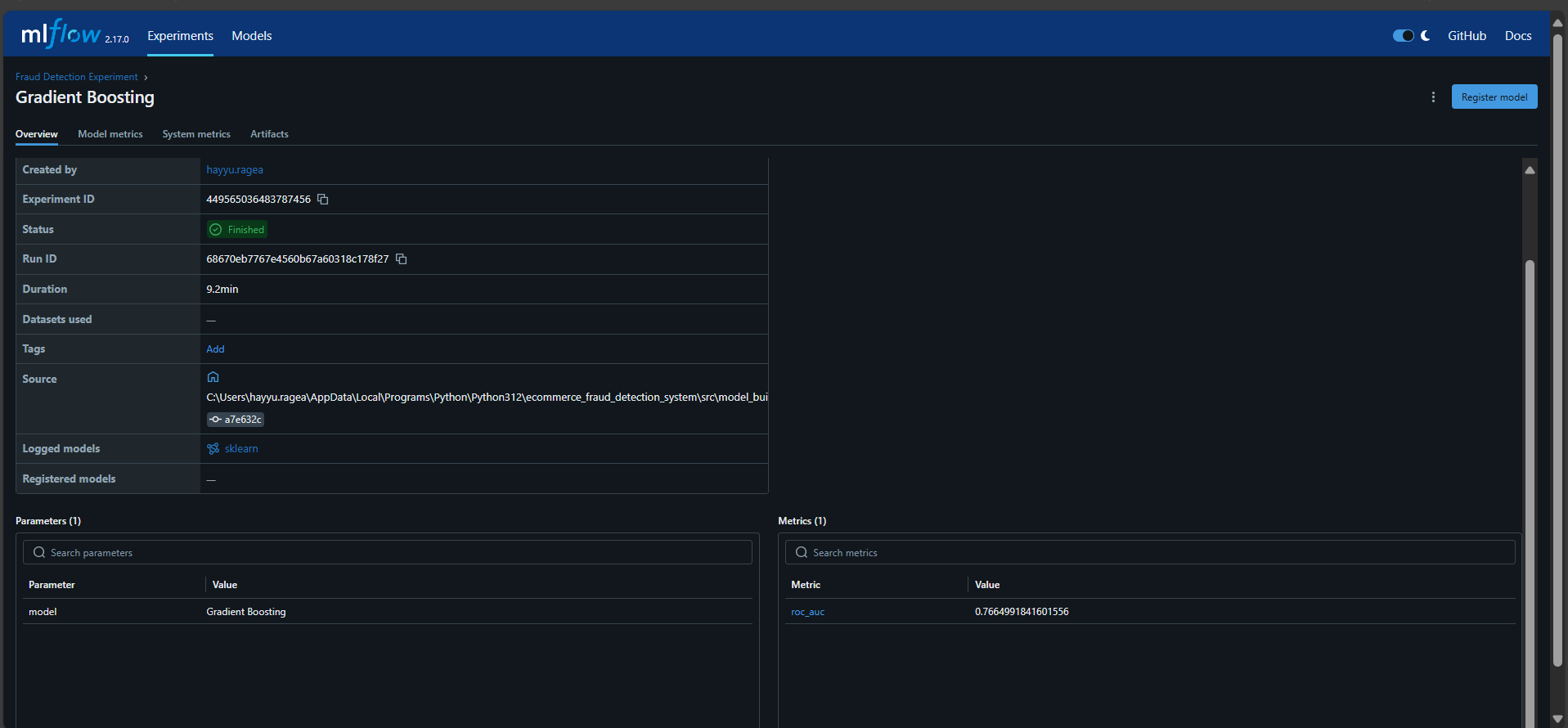
* Decision Tree Credit Card Data Data MLOps screenshot



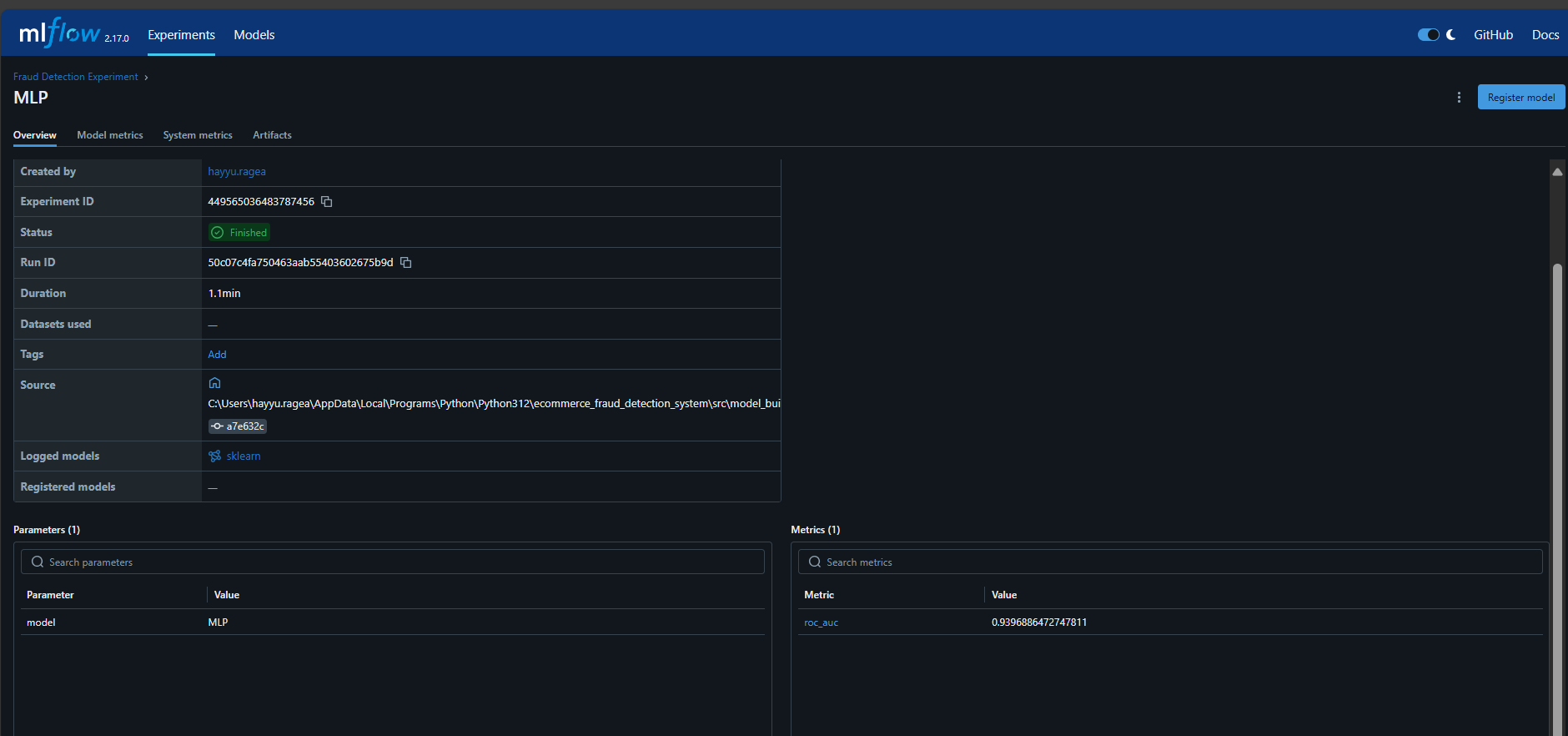
* Logistic Regression Credit Card Data Data MLOps screenshot



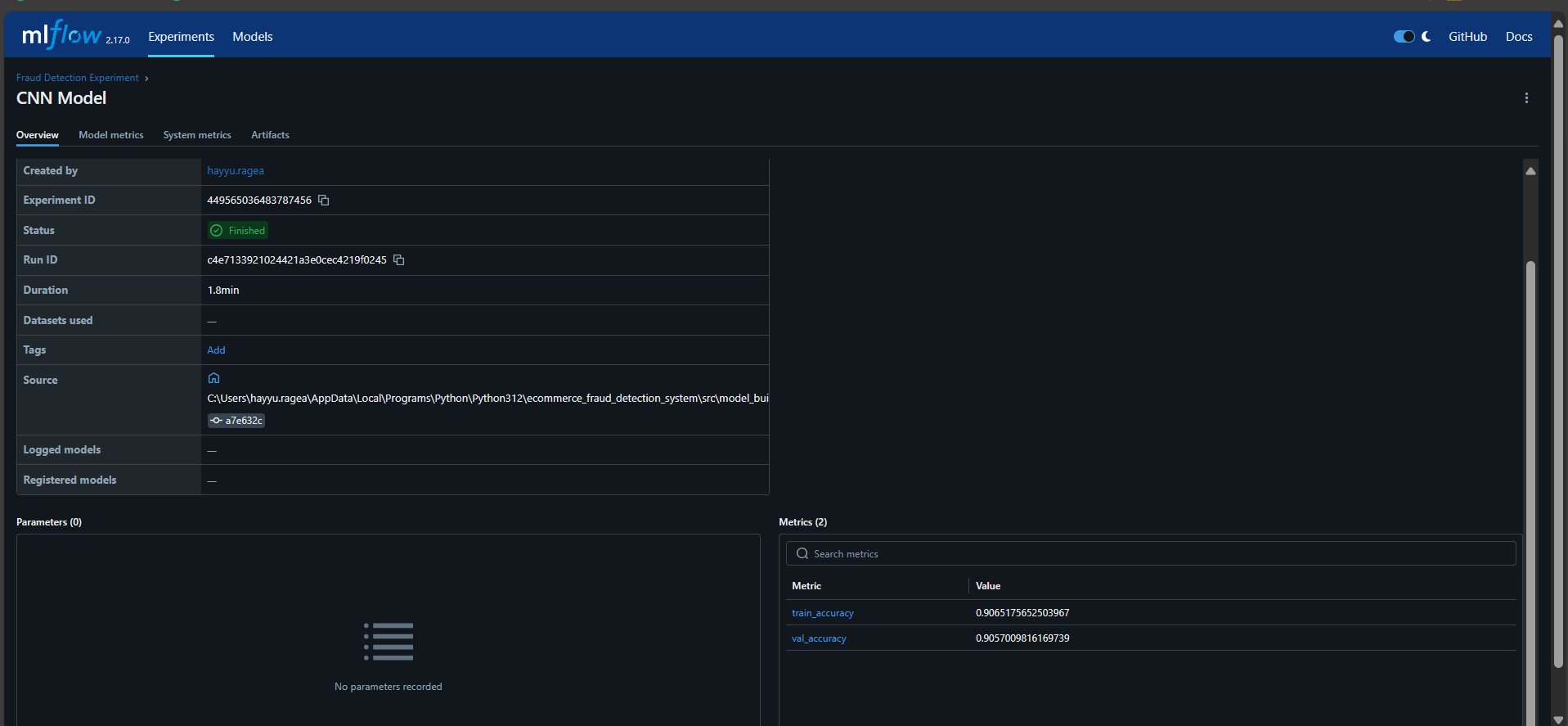
* Gradient Credit Card Data Data MLOps screenshot



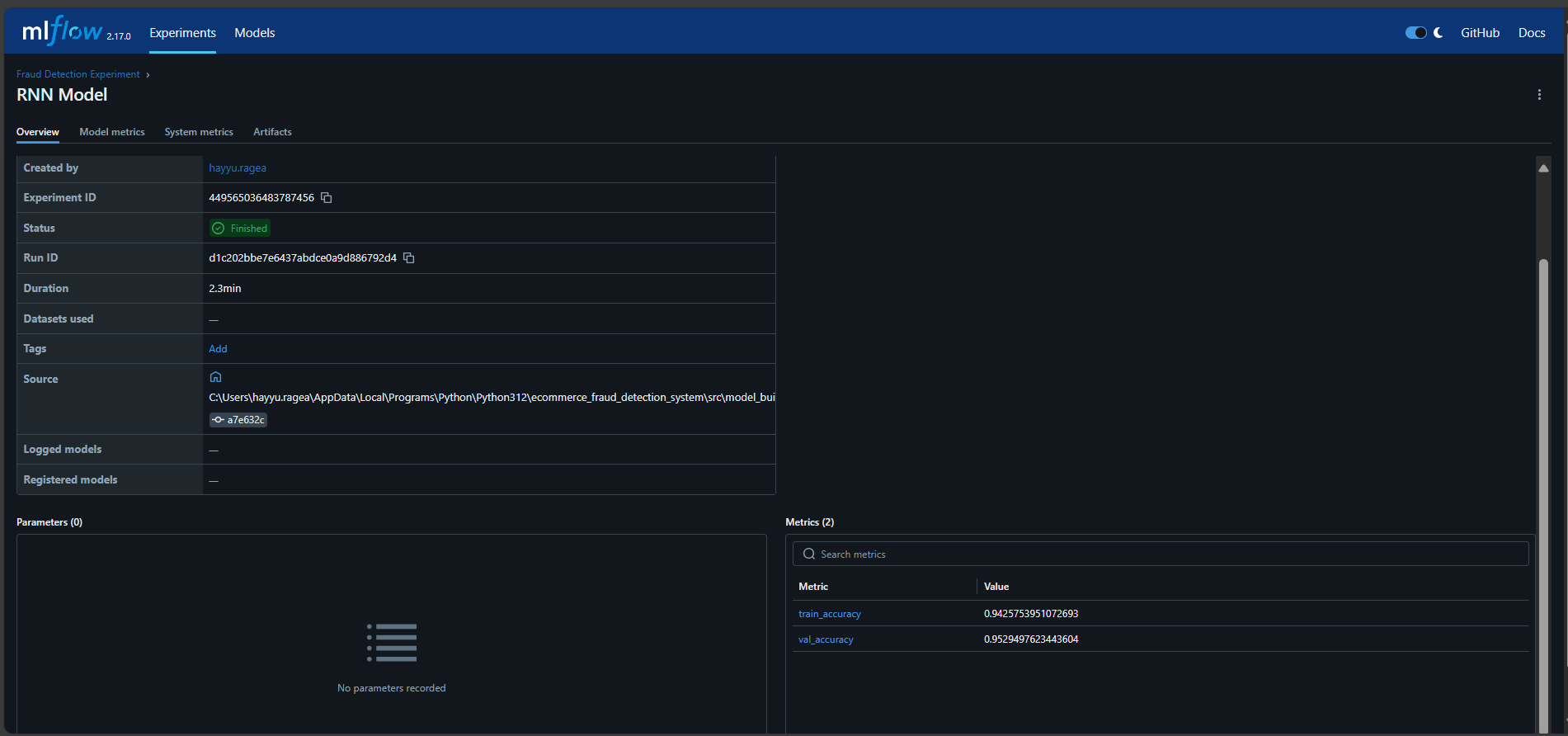
* Multi-Layer Perceptron (MLP) Credit Card Data Data MLOps screenshot



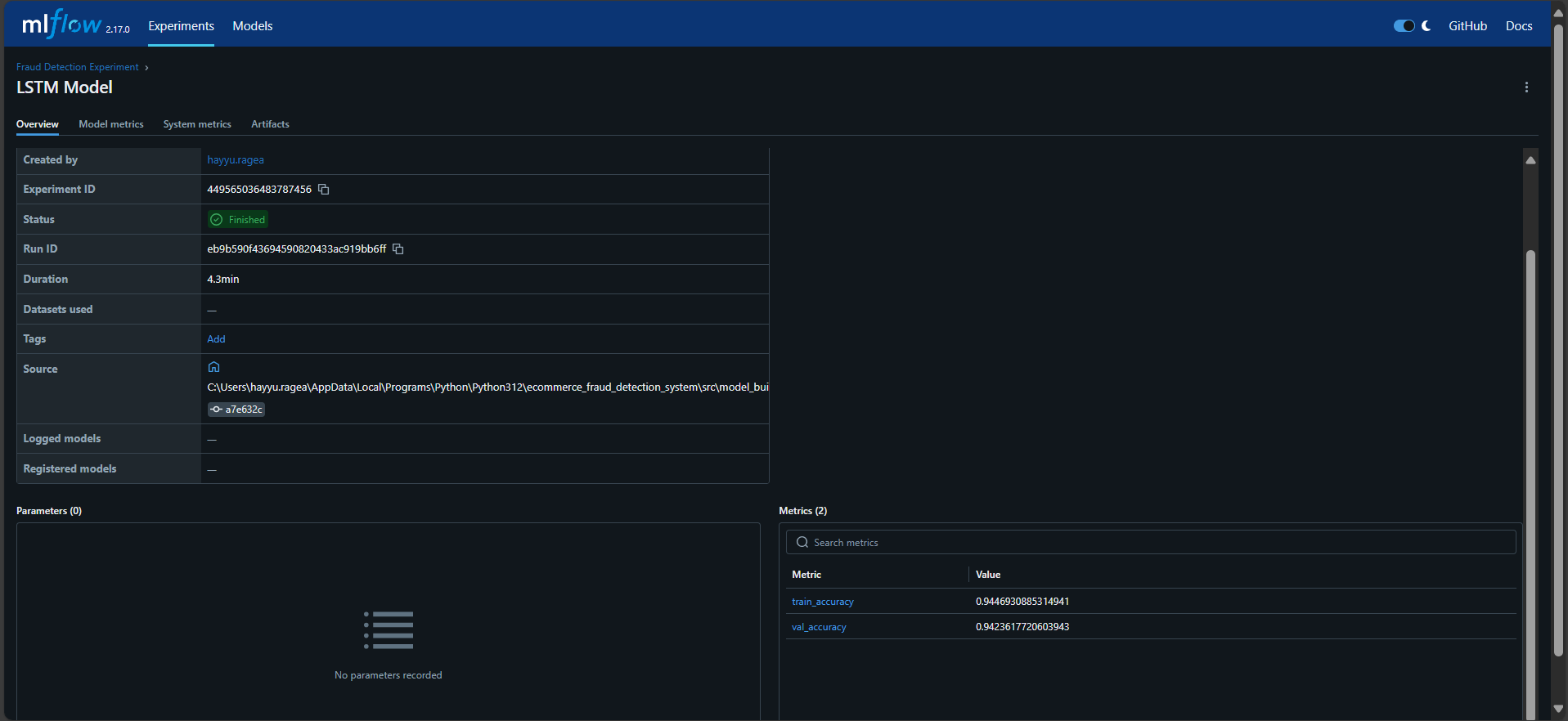
* Convolutional Neural Network (CNN) Credit Card Data Data MLOps screenshot



* Recurrent Neural Network (RNN) Credit Card Data Data MLOps screenshot



* Long Short-Term Memory (LSTM)Credit Card Data Data MLOps screenshot



**Project Impact**

The model building and training process have positioned Adey Innovations Inc. to proactively secure transactions and foster customer trust by implementing reliable, real-time fraud detection solutions.

**GitHub Link**:[**https://github.com/HaYyu-Ra/ecommerce\_fraud\_detection\_analysis/blob/master/notebooks/model\_biulding.ipynb**](https://github.com/HaYyu-Ra/ecommerce_fraud_detection_analysis/blob/master/notebooks/model_biulding.ipynb)

**Conclusion**:

The Random Forest model demonstrated consistent accuracy and robustness across both datasets, making it a suitable choice for deployment. Future steps include optimizing model parameters, managing any convergence warnings, and fine-tuning to improve overall performance.