AiEnsured

Ensuring comprehensive validation of AI systems

An article on

MLFLOW

On Fuel Efficiency Dataset

Submitted by: Rayapureddi Venkata Sri Sai Subhash

Intern in Machine Learning May 2023-July 2023

(rayapureddi.subhash@testaing.com)

MLflow is an open-source platform designed to manage the end-to-end machine learning lifecycle. It was developed by Databricks and has become widely used in the machine learning community. MLflow provides tools and libraries to track experiments, package code, share models, and manage the deployment of machine learning models.

Key components of MLflow:

- Tracking
- Projects
- Models
- Model Registry

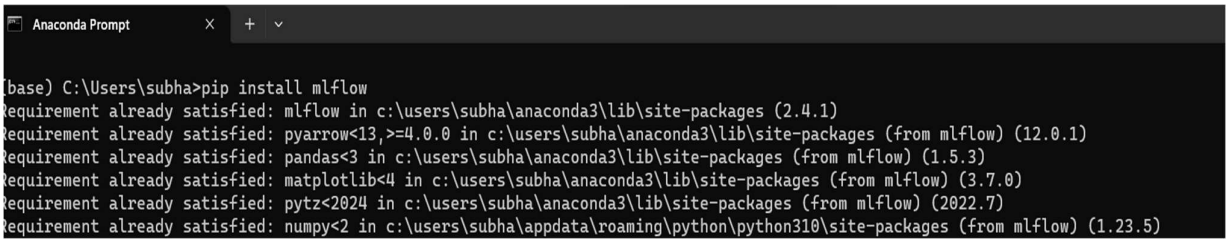
Benefits of using MLflow:

- Reproducibility
- Model Management
- Flexibility

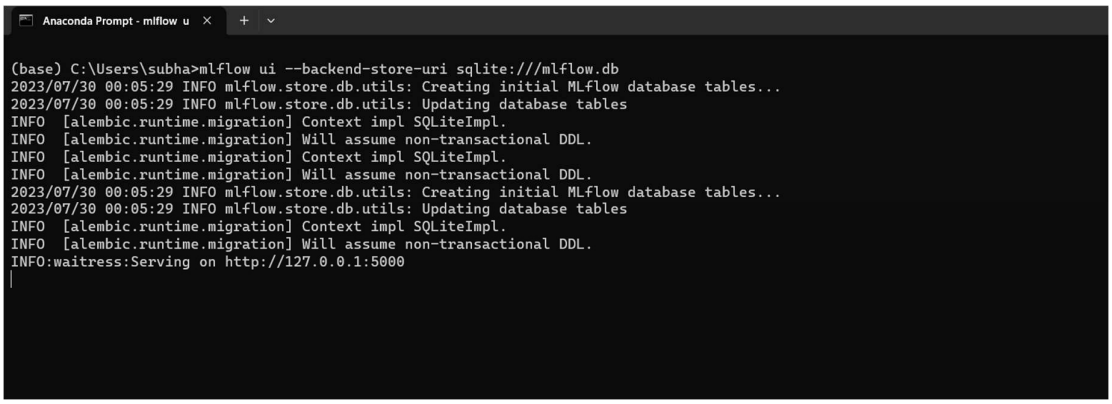
Overall, MLflow provides a comprehensive framework to streamline the machine learning development process, making it easier to track experiments, reproduce results, and deploy models efficiently in real-world applications.

Implementation Of MLFLOW:

First install the mlflow in your local system.



After installation connect it to the backend server to get the url.



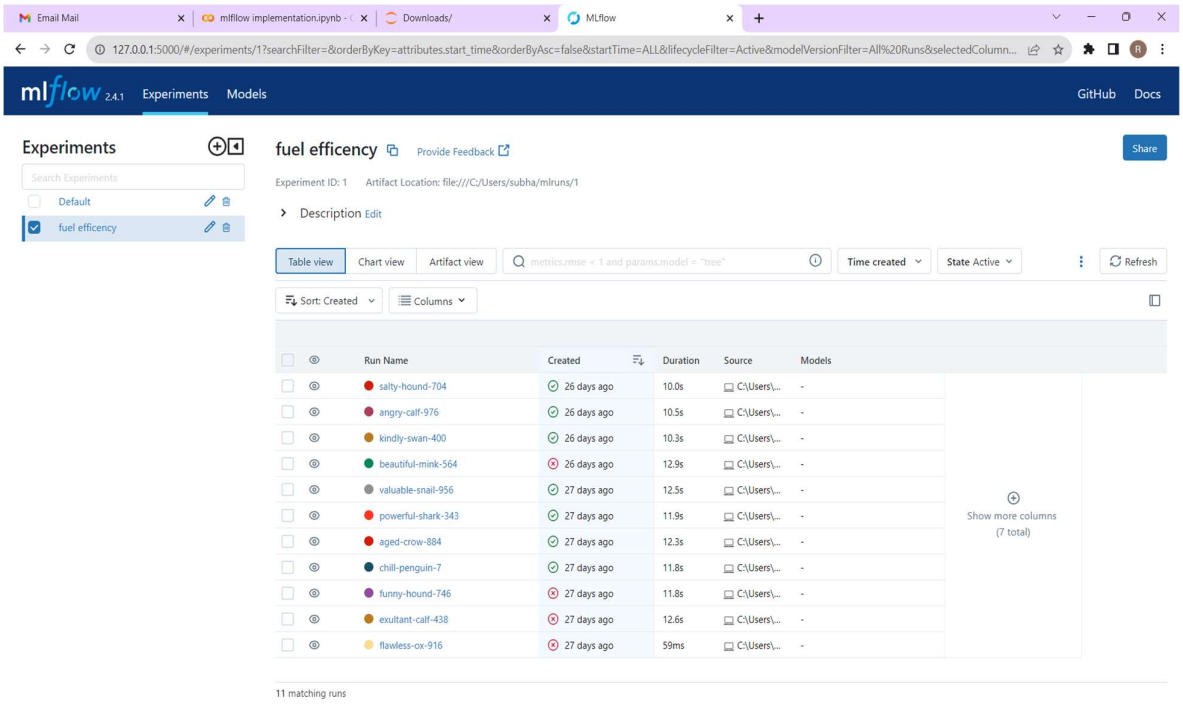
Connect the model to mlflow

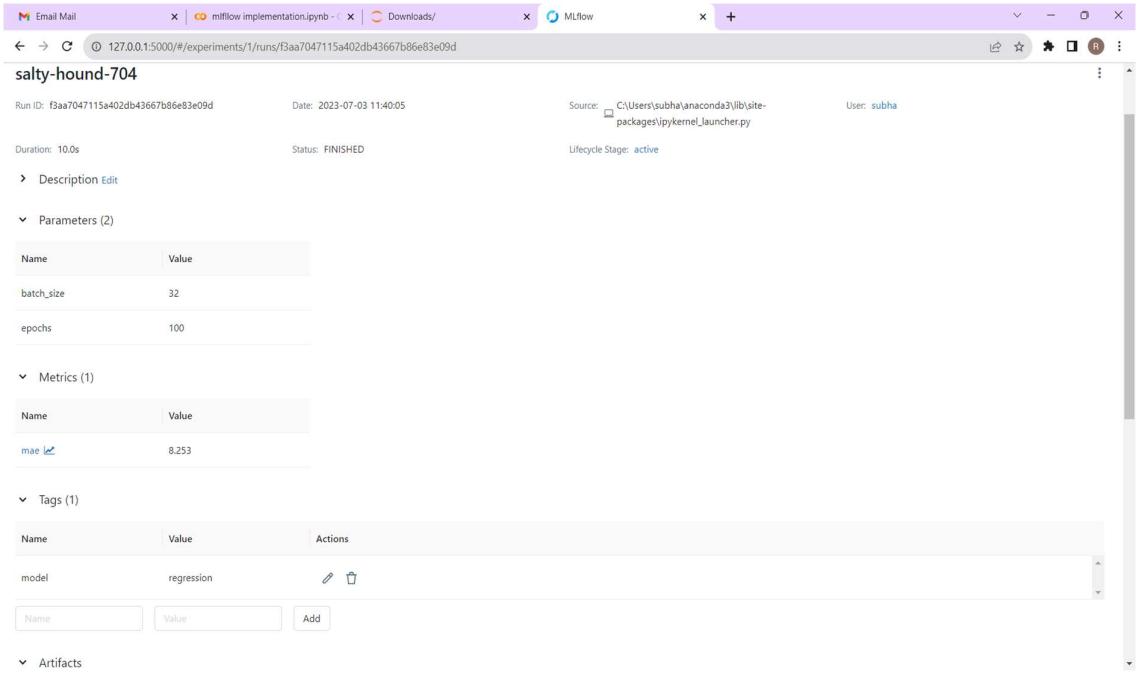
```
mlflow.set_tracking_uri("sqlite:///mlflow.db")
mlflow.set_experiment("fuel efficiency")
```

Log the parameters in the model with mlflow for tracking:

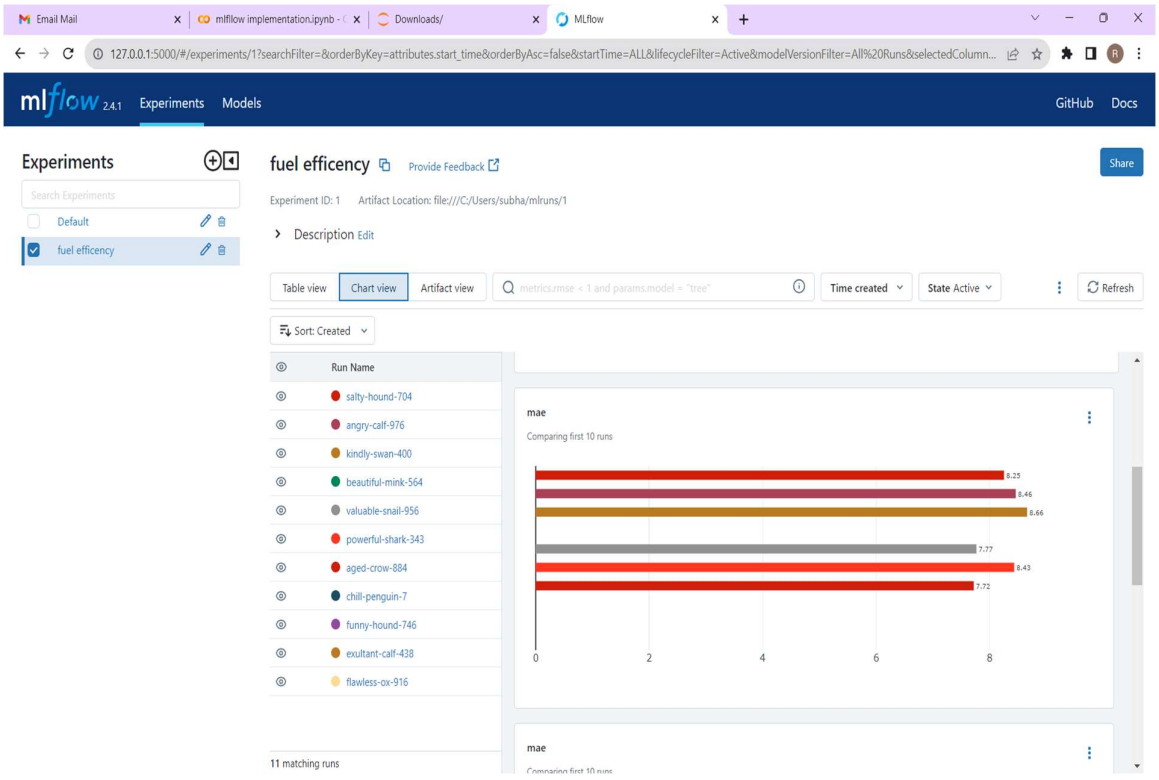
```
with mlflow.start_run():
    history = model.fit(X_train, y_train, batch_size=8, epochs=50, verbose=1, validation_split=0.2)
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['train','test'], loc='upper left')
    plt.show()
    y_pred = model.predict(X_test)
#the hyperparameters changed are bacth_size and epochs and the values changed are (50epochs,16);(100epochs , 32);(100epochs,16)#
    from sklearn import metrics
    mae=metrics.mean_absolute_error(y_test, y_pred)
    mse=metrics.mean_squared_error(y_test, y_pred)
    smse=np.sqrt(metrics.mean_squared_error(y_test, y_pred))
    mlflow.log_param("batch_size",32)
    mlflow.log_param("epochs",100)
    mlflow.set_tag("model","regression")
    mlflow.log_metric("mae",mae)
    print("metric : mae:{mae}")
```

Change the parameters and observe the changes in the mlflow page.





We can get the above page for every run after changing the parameters.



After the some runs of the model, we can easily track the parameters which are need to be change for the improvement of the model.

THE END