MLOPS LAB 5 REPORT

Name: Ratnesh Dubey

Roll Number: M24CSA027

Objective 1

Data Preparation:

- Load the Boston Housing dataset
- Identify and extract the following features
- Define the target variable ‘PRICE’

Outcome

1. Imported all the necessary libraries
2. Loaded the dataset

Objective 2

Linear Regression Model:

- Train a Linear Regression model using the numerical features to predict target variable.
- Evaluate the model using Mean Squared Error (MSE) and R-squared score.
- Provide an analysis of the model's performance.

Outcome

Implementing Linear Regression Model and Random Forest Model

1. Splitting the dataset in training and test set
2. Creating a basic Linear regression and Random Forest Model

Objective 3

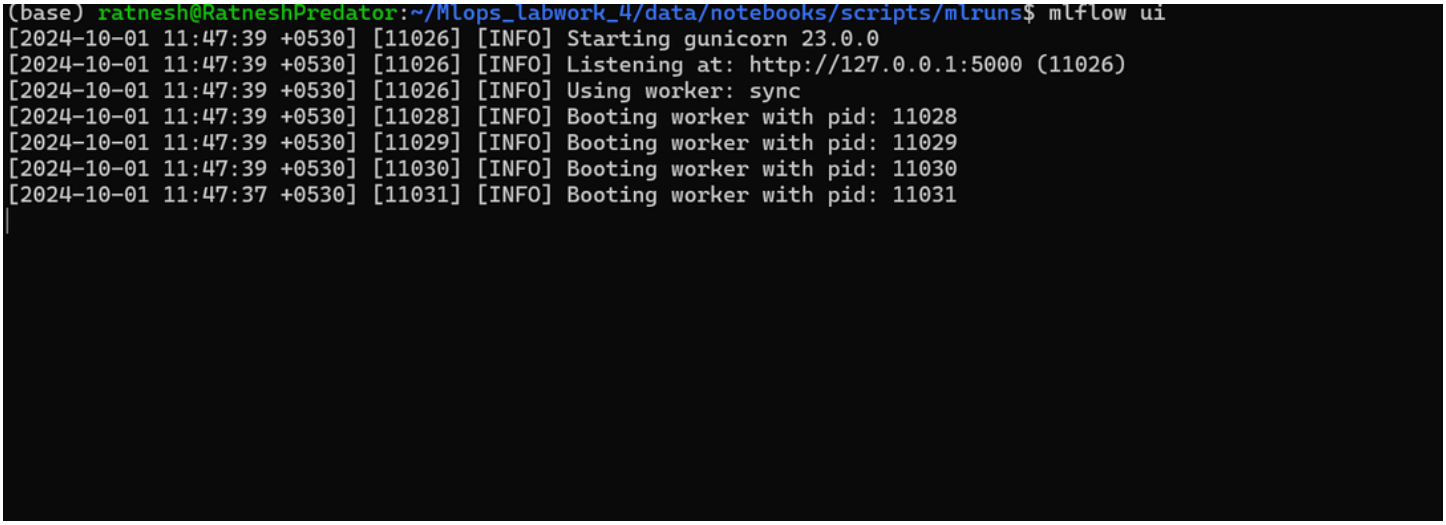
Run Experiments and Track with MLflow: Launch the MLflow UI in your local environment

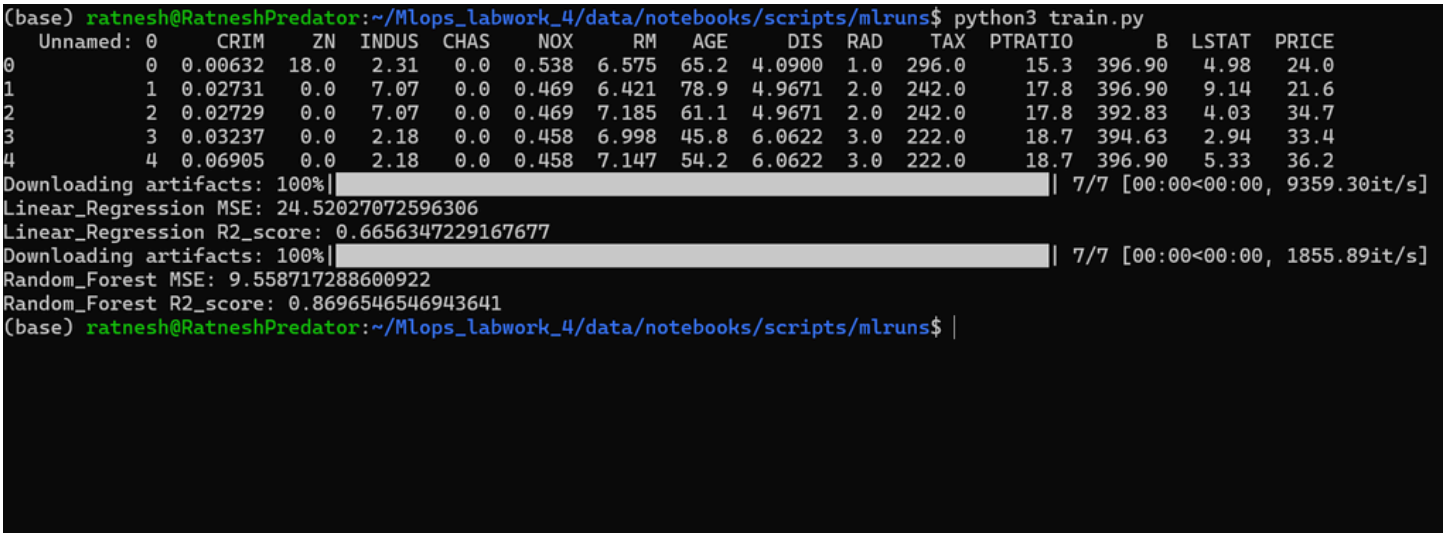
and run train.py file using python3 train.py

Outcome

Running successfully

Screenshots





Objective 4

Compare Models: Use the MLflow UI to compare the mse (Mean Squared Error) of the

Linear_Regression and Random_Forest models.

Outcome

The two models—Linear Regression and Random Forest—have provided the following results on the housing dataset:

- **Linear Regression:**
  - MSE: 24.52
  - R-squared: 0.6656

- **Random Forest:**
  - MSE: 9.56
  - R-squared: 0.8697

Comparison:

1. **Mean Squared Error (MSE):**
   - **Lower is better** for MSE, as it represents the average squared difference between the actual and predicted values. The Random Forest model has a significantly lower MSE (9.56) compared to Linear Regression (24.52). This suggests that Random Forest is much more accurate in predicting the housing prices and has lower errors in its predictions.
2. **R-squared (R^2):**
   - **Higher is better** for R^2, as it represents the proportion of variance in the dependent variable (housing price) that is predictable from the independent variables. Random Forest has a higher R-squared value (0.8697) compared to Linear Regression (0.6656). This means that the Random Forest model explains a larger portion of the variance in the data, making it a better fit.

Conclusion:

- **Performance of Random Forest vs Linear Regression:**
  - The Random Forest model **outperforms Linear Regression** in both metrics, with a lower MSE and a higher R-squared value. This indicates that the Random Forest model captures the relationship between the features and the housing prices much better than the Linear Regression model.
- **Why Random Forest is Better:**
  - **Random Forest** is a more complex model that can capture nonlinear relationships and interactions between features, which is likely why it performs better on the housing dataset. Housing price prediction is often influenced by complex factors that linear models like Linear Regression may not capture effectively.
- **Linear Regression** tends to work well when the relationship between features and the target variable is approximately linear. In this case, the relatively high MSE and lower R-squared indicate that the linear assumption may not fully capture the complexity of the housing price data.

Screenshots

mlflow 2.16.2 Experiments Models GitHub

MLflow Quickstart ›

chill-dolphin-141

Register m

Overview Model metrics System metrics Artifacts

Created by	ratnesh
Experiment ID	136812734081471738
Status	Finished
Run ID	325ffb68b5e64eb2abd95fde22b3425a
Duration	2.8s
Datasets used	—
Tags	Training Info: Basic Linear_Regression model for di...
Source	train.py
Logged models	sklearn
Registered models	—

Parameters (2)

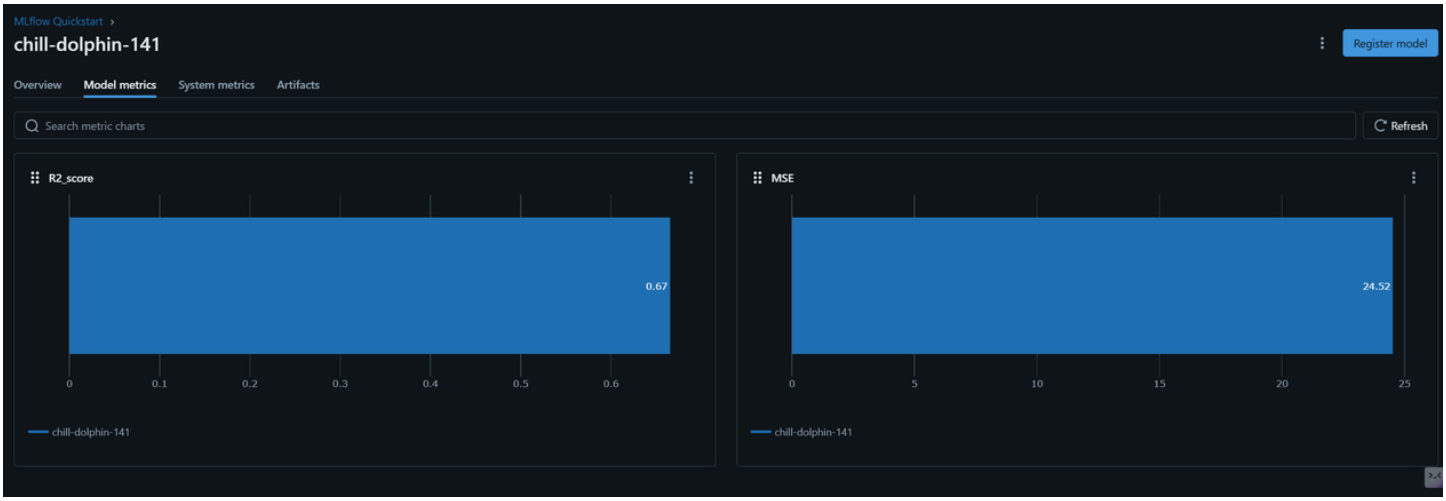
Search parameters

Parameter	Value
fit_intercept	True
normalize	False

Metrics (2)

Search metrics

Metric	Value
R2_score	0.6656347229167677
MSE	24.52027072596306



mlflow 2.16.2 Experiments Models GitHub

MLflow Quickstart ›

righteous-duck-305

Register r

Overview Model metrics System metrics Artifacts

Created at	2024-10-01 11:41:55
Created by	ratnesh
Experiment ID	136812734081471738
Status	Finished
Run ID	5ff38345e7c94f419e208caffacd420c
Duration	1.0s
Datasets used	—
Tags	Training Info: Basic Random_Forest model for dia...
Source	train.py
Logged models	sklearn
Registered models	—

Parameters (3)

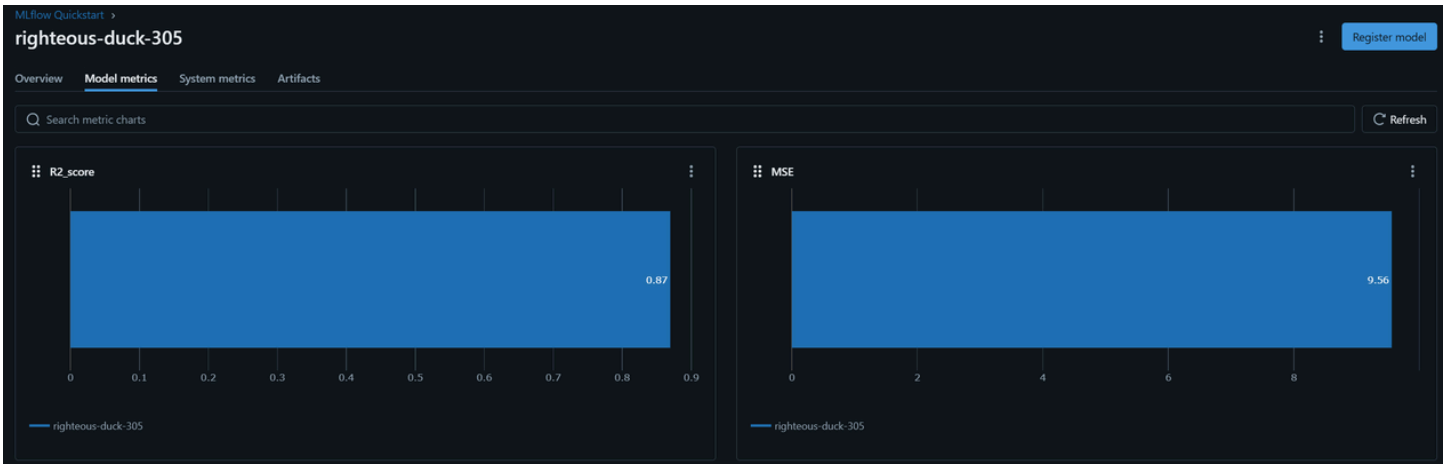
Search parameters

Parameter	Value
random_state	42
max_depth	5
n_estimators	100

Metrics (2)

Search metrics

Metric	Value
R2_score	0.8696546546943641
MSE	9.558717288600922



Objective 5

Save Model: The best-performing model is saved and logged in MLflow's Model

Outcome

Best Model is Random Forest in this case with MSE = 9.55871 and logged in MLflow

Screenshots

```
Best model: Random_Forest with MSE: 9.558717288600922
Successfully registered model 'Best_Housing_Model: Random_Forest'.
Created version '1' of model 'Best_Housing_Model: Random_Forest'.
Random_Forest registered in the Model Registry.
(base) ratnesh@RatneshPredator:~/Mlops_labwork_4/data/notebooks/scripts/mlruns$
```