

# MLops Assignment Report

By- Tanishq Kankerwal (B21BB032)

## Experiment Tracking with MLflow

**Objective** - My goal is to use the mlflow to see the performance of different models and manage experiment metadata. The steps include model training, performance comparison, and logging the best-performing model in the MLflow Model Registry.

## Dataset used

### Housing Prices Dataset

It contains the following columns.

**crim:** Per capita crime rate by town.

**zn:** Proportion of residential land zoned for lots over 25,000 sq. ft.

**indus:** Proportion of non-retail business acres per town.

**chas:** Charles River dummy variable (1 if tract bounds river; 0 otherwise).

**nox:** Nitric oxides concentration (parts per 10 million).

**rm:** Average number of rooms per dwelling.

**age:** Proportion of owner-occupied units built prior to 1940.

**dis:** Weighted distances to five Boston employment centers.

**rad:** Index of accessibility to radial highways.

**tax:** Full-value property tax rate per \$10,000.

**ptratio:** Pupil-teacher ratio by town.

**b:**  $b = 1000(B_k - 0.63)^2$  where  $B_k$  is the proportion of Black residents by town.

**lstat:** Percentage of lower status of the population.

**medv:** Median value of owner-occupied homes in \$1000s (target variable).

# Models used

## Linear Regression

## Random Forest

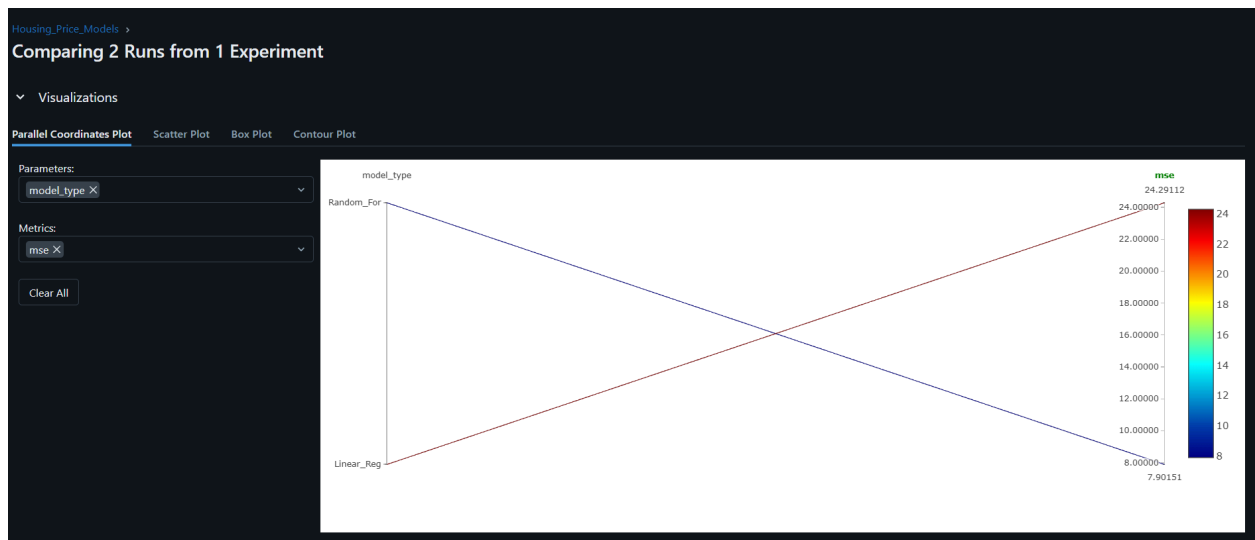
Both models were employed to predict the median value of homes based on various housing features. By comparing their Mean Squared Error (MSE) values in MLflow, you can evaluate which model performs better and under what circumstances. The combination of a simple model (Linear Regression) and a more complex model (Random Forest) provides a comprehensive understanding of the dataset and its predictive capabilities.

# Model training

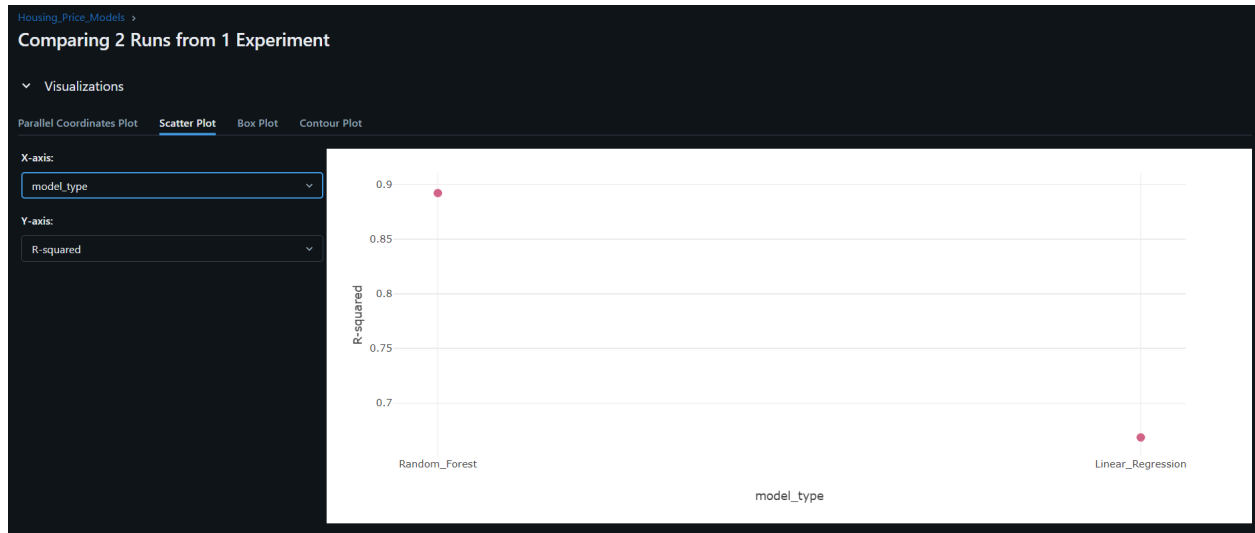
I wrote the code for training the model using 2 approaches, one is linear regression and another is random forest.

# Metrics used for comparison

## Mean squared Error



## R- squared



```
Linear_Regression - MSE: 24.291119474973385
Linear_Regression - R-squared: 0.6687594935356338
2024/10/01 12:01:07 WARNING mlflow.models.model: Mo
Random_Forest - MSE: 7.901513892156864
Random_Forest - R-squared: 0.8922527442109116
```

For Linear Regression:

MSE: 24.291119474973385

R-squared: 0.6687594935356338

For Random Forest:

MSE: 7.901513892156864

R-squared: 0.8922527442109116

## ML Flow for visualization

I used ml flow to see the visualization

The screenshot shows the mlflow Experiments interface. The top navigation bar includes 'mlflow 2.16.2', 'Experiments', and 'Models'. The main header for the experiment is 'Housing\_Price\_Models'. Below this, there are tabs for 'Runs', 'Evaluation', 'Experimental', and 'Traces'. The 'Experimental' tab is active, showing a table of runs. The table has columns for 'Run Name', 'Created', 'Dataset', 'Duration', 'Source', and 'Models'. There are four runs listed: two 'Random\_Forest' runs and two 'Linear\_Regression' runs. The 'Random\_Forest' runs have a duration of 1.6s and 1.5s, while the 'Linear\_Regression' runs have a duration of 1.8s and 2.0s. The 'Random\_Forest' runs are marked as 'v1'.

Run Name	Created	Dataset	Duration	Source	Models
Random_Forest	14 minutes ago	-	1.6s	train.py	sklearn
Linear_Regression	14 minutes ago	-	1.8s	train.py	sklearn
Random_Forest	27 minutes ago	-	1.5s	train.py	Random_forest v1
Linear_Regression	27 minutes ago	-	2.0s	train.py	sklearn

# Registered Models

Compare the model first

Run ID:	22c6377bdb8847d19761f50110b7e90e	b496df42b2c74d91a7ad97b5ec785724
Run Name:	Random_Forest	Linear_Regression
Start Time:	2024-10-01 12:01:06	2024-10-01 12:01:04
End Time:	2024-10-01 12:01:07	2024-10-01 12:01:06
Duration:	1.6s	1.8s
Parameters		
Show diff only		
model_type	Random_Forest	Linear_Regression
Metrics		
Show diff only		
R-squared	0.892	0.669
mse	7.902	24.29

Registered Models						Create Model
Filter registered models by name or tags						
Name	Latest version	Aliased versions	Created by	Last modified	Tags	
Random_forest	Version 1			2024-10-01 11:56:27	—	

Registered the best model which was Random forest

# Conclusion

- **Performance:** The **Random Forest** model significantly outperforms **Linear Regression** in terms of both MSE and R squared. The lower MSE and higher R squared value for Random Forest demonstrate that it provides more accurate predictions and captures more of the data's variance, making it a better model for this dataset.
- **Interpretability:** While Linear Regression is simpler and easier to interpret due to its linear assumptions, it falls short in performance compared to Random Forest, which handles complex and non-linear interactions among features effectively.
- **Recommendation:** Based on these results, **Random Forest** is the recommended model for predicting housing prices due to its superior predictive power and ability to generalize well to unseen data.