**Lab Exercise 16- House Price Prediction Data Processing Pipeline with Metaflow**

Here’s a lab exercise that creates a data processing pipeline for a house price prediction model using Metaflow. The pipeline will simulate data processing steps including data loading, feature engineering, model training, and evaluation.

**Prerequisites**

1. **Metaflow Installed**: Ensure you have Metaflow installed in your Python environment.
2. **Pandas and Scikit-learn**: Install necessary libraries if you haven't already:

pip install pandas scikit-learn

**Step 1: Create a Sample Dataset**

For this exercise, you can create a simple CSV file named **house\_prices.csv** with the following content:

id,area,beds,baths,price

1,1500,3,2,300000

2,2000,4,3,500000

3,1200,2,1,250000

4,1800,3,2,400000

5,1600,3,2,350000

**Step 2: Create the Metaflow Script**

Create a file named house\_price\_pipeline.py with the following code:

from metaflow import FlowSpec, step, batch

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

class HousePricePredictionPipeline(FlowSpec):

@step

def start(self):

"""Load the dataset."""

print("Loading house prices dataset...")

self.data = pd.read\_csv('house\_prices.csv')

print(f"Dataset loaded with {len(self.data)} records.")

self.next(self.feature\_engineering)

@step

def feature\_engineering(self):

"""Perform feature engineering."""

print("Performing feature engineering...")

self.features = self.data[['area', 'beds', 'baths']]

self.labels = self.data['price']

# Split the data into training and test sets

self.X\_train, self.X\_test, self.y\_train, self.y\_test = train\_test\_split(

self.features, self.labels, test\_size=0.2, random\_state=42

)

print("Feature engineering completed.")

self.next(self.train\_model)

@batch

@step

def train\_model(self):

"""Train the regression model."""

print("Training the model...")

self.model = LinearRegression()

self.model.fit(self.X\_train, self.y\_train)

print("Model training completed.")

self.next(self.evaluate\_model)

@step

def evaluate\_model(self):

"""Evaluate the model performance."""

print("Evaluating model performance...")

predictions = self.model.predict(self.X\_test)

mse = mean\_squared\_error(self.y\_test, predictions)

print(f"Mean Squared Error: {mse}")

self.next(self.end)

@step

def end(self):

"""End of the pipeline."""

print("House price prediction pipeline completed.")

if \_\_name\_\_ == '\_\_main\_\_':

HousePricePredictionPipeline()

**Step 3: Run the Metaflow Pipeline**

In your terminal, run the following command to execute the pipeline:

python house\_price\_pipeline.py run