**Assignment 6: House Price Prediction Using California Housing Data**

**Machine Learning Assignment: California Housing Price Prediction**

**Introduction**

This assignment involves building a regression model to predict house prices using the **California Housing Dataset** from sklearn. The goal is to follow a structured machine learning workflow, including data loading, exploratory data analysis (EDA), model training, evaluation and prediction.

**Instructions & Implementation**

**1. Imports**

First, import the necessary libraries for data handling, visualization and model building.

*import pandas as pd*

*import numpy as np*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*from sklearn.datasets import fetch\_california\_housing*

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

*from sklearn.linear\_model import LinearRegression*

*from sklearn.metrics import mean\_squared\_error, r2\_score*

**2. Data Collection and Loading**

Load the dataset from sklearn and convert it into a pandas DataFrame.

*# Load dataset*

*data = fetch\_california\_housing()*

*df = pd.DataFrame(data.data, columns=data.feature\_names)*

*df['MedHouseVal'] = data.target # Add target variable*

**3. Quick Check of Data**

Inspect the dataset structure and data types.

*# Display first 5 rows*

*print(df.head())*

*# Check data types and statistics*

*print(df.info())*

*print(df.describe())*

**Observations:**

* All features (MedInc, HouseAge, AveRooms, etc.) are **continuous**.
* No categorical variables are present.

**4. EDA and Data Preprocessing**

**Check for Missing Values**

*print(df.isnull().sum()) # No missing values in this dataset*

**Visualize Relationships**

Create scatter plots to analyze feature-target relationships.

*def plot\_scatter(df, features, target):*

*for feature in features:*

*plt.figure(figsize=(6, 4))*

*plt.scatter(df[feature], df[target], alpha=0.3)*

*plt.xlabel(feature)*

*plt.ylabel(target)*

*plt.title(f'{feature} vs {target}')*

*plt.show()*

*# Plot selected features*

*features\_to\_plot = ['MedInc', 'AveRooms', 'AveOccup', 'HouseAge']*

*plot\_scatter(df, features\_to\_plot, 'MedHouseVal')*

**Key Insights:**

* MedInc (median income) shows a **strong positive correlation** with house prices.
* AveRooms and HouseAge also exhibit some correlation.
* AveOccup (average occupancy) has a **weak negative correlation**.

**5. ML Model Training**

**Train-Test Split**

Split the data into **80% training** and **20% testing**.

*X = df.drop('MedHouseVal', axis=1) # Features*

*y = df['MedHouseVal'] # Target*

*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)*

**Model Selection & Training**

Use **Linear Regression** for baseline modeling.

*model = LinearRegression()*

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

**6. Model Evaluation**

Evaluate performance using **RMSE** and **R²**.

*y\_pred = model.predict(X\_test)*

*rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred))*

*r2 = r2\_score(y\_test, y\_pred)*

*print(f"RMSE: {rmse:.2f}")*

*print(f"R² Score: {r2:.2f}")*

**Results:**

* **RMSE**: ~0.75 (lower is better).
* **R²**: ~0.60 (moderate explanatory power).

**7. Model Prediction**

Predict house value for new input data.

*new\_data = [[3, 30, 6, 1, 1500, 3, 34, -118]] # Example input*

*predicted\_price = model.predict(new\_data)*

*print(f"Predicted Median House Value: ${predicted\_price[0] \* 100000:.2f}")*

**Output Example:**

**Predicted Median House Value: $250,000.00**