Practical 4: Data Analytics I

Data Analytics I

Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/datasets/altavish/boston-housing-dataset). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset. The objective is to predict the value of prices of the house using the given features.

The **Boston Housing Dataset** contains information about various houses in Boston, with each column representing a specific feature or attribute. Here's a detailed explanation of each column:

1. CRIM

- **Description**: Per capita crime rate by town.
- **Explanation**: This column represents the crime rate in the area where the house is located. Higher values indicate higher crime rates.

2. ZN

- **Description**: Proportion of residential land zoned for lots over 25,000 sq. ft.
- **Explanation**: This column indicates the proportion of land in the area that is zoned for large residential lots. Higher values mean more land is allocated for large residential properties.

3. INDUS

- **Description**: Proportion of non-retail business acres per town.
- **Explanation**: This column represents the proportion of land in the area used for non-retail businesses (e.g., industrial or commercial). Higher values indicate more industrial or commercial land use.

4. CHAS

- **Description**: Charles River dummy variable.
- **Explanation**: This is a binary variable:
  - 1 if the property is adjacent to the Charles River.
  - 0 if the property is not adjacent to the Charles River.

5. NOX

- **Description**: Nitrogen oxides concentration (parts per 10 million).
- **Explanation**: This column represents the concentration of nitrogen oxides in the air, which is a measure of air pollution. Higher values indicate higher pollution levels.

6. RM

- **Description**: Average number of rooms per residence.
- **Explanation**: This column indicates the average number of rooms in houses in the area. Higher values mean larger houses with more rooms.

7. AGE

- **Description**: Proportion of owner-occupied units built prior to 1940.
- **Explanation**: This column represents the proportion of houses in the area that were built before 1940. Higher values indicate older housing stock.

8. DIS

- **Description**: Weighted distances to five Boston employment centers.
- **Explanation**: This column measures the weighted distance of the property to major employment centers in Boston. Higher values mean the property is farther from employment centers.

9. RAD

- **Description**: Index of accessibility to radial highways.
- **Explanation**: This column represents the accessibility of the property to radial highways. Higher values indicate better access to highways.

10. TAX

- **Description**: Full-value property tax rate per $10,000.
- **Explanation**: This column indicates the property tax rate for the area. Higher values mean higher property taxes.

11. PTRATIO

- **Description**: Pupil-teacher ratio by town.
- **Explanation**: This column represents the ratio of students to teachers in schools in the area. Higher values mean fewer teachers per student.

12. BLACK

- **Description**: Proportion of Black residents by town, scaled by the formula: $1000(Bk - 0.63)^2$, where Bk is the proportion of Black residents.
- **Explanation**: This column is a transformed measure of the proportion of Black residents in the area. The formula scales the values to emphasize differences.

13. LSTAT

- **Description**: Lower status of the population (percent).

- **Explanation**: This column represents the percentage of the population in the area that is considered lower status (e.g., lower income or education levels). Higher values indicate a higher proportion of lower-status residents.

14. MEDV

- **Description**: Median value of owner-occupied homes in $1000s.
- **Explanation**: This is the **target variable** (dependent variable) in the dataset. It represents the median price of owner-occupied homes in the area, measured in thousands of dollars. For example, a value of 30 means the median home price is $30,000.

Sample Example:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
# Sample data
X = [[1], [2], [3], [4]]
y = [1, 2, 3, 4]
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
# Train a model
model = LinearRegression()
model.fit(X_train, y_train)
# Predict
print("X_train:", X_train, "X_test:", X_test)
print(y_train, y_test)
y_pred=model.predict(X_test)
print(y_test, y_pred)
```

Step 1: Import Libraries

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

Step 2: Load the Dataset

Download the dataset from Kaggle (https://www.kaggle.com/datasets/altavish/boston-housing-dataset) and load it into a pandas DataFrame:

```
# Load the dataset
# Load the dataset
file_path = "C:/Users/Talha Ahmed/Desktop/My Practicals/Practical 4/HousingData.csv"  # Replace with your file path
data = pd.read_csv(file_path)
data

# Summary statistics of the dataset
data.describe()

data.isnull().sum()

# Drop rows or columns with missing values (if necessary)
data = data.dropna()
data

data.isnull().sum()

# Correlation matrix to understand relationships, Heatmap
plt.figure(figsize=(12, 12))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')#
plt.title("Correlation Heatmap")
plt.show()

#data.corr(): The correlation matrix to visualize.
#annot=True: Displays the correlation coefficients as text in each cell of the heatmap.
#cmap='coolwarm': Specifies the color map for the heatmap. coolwarm uses a gradient from cool colors (e.g., blue) for negative values to warm colors (e.g., red) for positive values.
#Red: Strong positive correlation.
#Blue: Strong negative correlation.
White: Weak or no correlation.


data.corr()
```

```
# Target variable (House prices)
y = data["MEDV"]  # Update 'MEDV' to the target variable name in your dataset because it is the price of house

# Feature variables
X = data.drop(["MEDV"], axis=1)  # Remove the target variable from the features

# Split data into train and test sets (67% train, 33% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)

# Create the model
model = LinearRegression()
```

Step 3: Train the Model

```
# Train the model
regr=model.fit(X_train, y_train)
print ('Coefficients: ', regr.coef_)
```

Step 4: Calculate predicted price of house

```
# Predict on the test set
y_pred = model.predict(X_test)

# Calculate Mean Squared Error and R-squared
mse = mean_squared_error(y_test, y_pred)
mse
# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)

# Calculate R-squared, should be higher than 0.6
r2 = r2_score(y_test, y_pred)
r2

comparison_df = pd.DataFrame({
    "Actual": y_test,
    "Predicted": y_pred
})
comparison_df

# Save the DataFrame to a CSV file
comparison_df.to_csv("actual_vs_predicted.csv")
```