PYTHON ASSIGNMENT

CHINTAPENT SAI SRINIVAS-126004058

GUBULA NIKHIL- 126004089

ASKSHAY REDDY- 126004087

Q.) Real-time application using Python

Title: ***Predicting Housing Prices Using Linear Regression***

This project aims to develop a predictive model for housing prices using linear regression techniques. The dataset utilized for this analysis is a comprehensive housing dataset, typically featuring various attributes that influence property values, such as the number of rooms, location, and other socioeconomic factors. This project is a practical application of supervised learning where the model learns from labeled data (features and target prices) and applies that knowledge to predict house prices for unseen data.

Key Components:

 Dataset: The dataset consists of various attributes like average number of rooms (RM), percentage of lower status population (LSTAT), and pupil-teacher ratio (PTRATIO), among others.

 Linear Regression: A linear regression model is used to predict housing prices based on these features.

 Evaluation: The performance of the model is evaluated using metrics like Mean Squared Error (MSE) and R-squared (R^2), which help determine how well the model fits the data.

 Visualization: The results are visualized by plotting actual vs predicted housing prices, providing a visual insight into the model's performance.

PROGRAM:

```
import numpy as np
 import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Loading the external housing dataset
data = pd.read_csv('housing.csv')
# Display the first few rows of the dataset
print(data.head())
# Define features and target (modify based on your CSV structure)
X = data.drop(columns=['MEDV'])  # Assuming 'MEDV' is the target
y = data['MEDV']
# Splitting the dataset into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Initialize the Linear Regression model

model = LinearRegression()

# Train the model

model.fit(X_train, y_train)

# Make predictions

y_pred_train = model.predict(X_train)

y_pred_test = model.predict(X_test)

# Evaluate the model

mse_train = mean_squared_error(y_train, y_pred_train)

r2_train = r2_score(y_train, y_pred_train)

mse_test = mean_squared_error(y_test, y_pred_test)

r2_test = r2_score(y_test, y_pred_test)

# Print evaluation metrics

print(f"Training MSE: {mse_train}")

print(f"Training R-squared: {r2_train}")

print(f"Testing MSE: {mse_test}")

print(f"Testing R-squared: {r2_test}")

# Plotting Actual vs Predicted Prices

plt.scatter(y_test, y_pred_test, color='blue')

plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],
color='red', linewidth=2)

plt.xlabel('Actual Prices')
```

plt.ylabel('Predicted Prices')

plt.title('Actual vs Predicted Housing Prices')

plt.show()