BLENDED_LEARNING

Implementation-of-Linear-and-Polynomial-Regression-Models-for-Predicting-Car-Prices

AIM:

To write a program to predict car prices using Linear Regression and Polynomial Regression models.

Equipments Required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Jupyter notebook

Algorithm

1. Data Collection: Import essential libraries like pandas, numpy, sklearn, matplotlib, and seaborn. Load the dataset using pandas.read_csv(). Data Preprocessing:
2. Address any missing values in the dataset. Select key features for training the models. Split the dataset into training and testing sets with train_test_split(). Linear Regression:
3. Initialize the Linear Regression model from sklearn. Train the model on the training data using .fit(). Make predictions on the test data using .predict(). Evaluate model performance with metrics such as Mean Squared Error (MSE) and the R^2 score.
4. Polynomial Regression: Use PolynomialFeatures from sklearn to create polynomial features. Fit a Linear Regression model to the transformed polynomial features. Make predictions and evaluate performance similar to the linear regression model.
5. Visualization: Plot the regression lines for both Linear and Polynomial models. Visualize residuals to assess model performance.

Program:

```
/*
Program to implement Linear and Polynomial Regression models for predicting car prices
Developed by: GANESH PRABHU J
RegisterNumber: 212223220023
*/
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
from sklearn.preprocessing import PolynomialFeatures,StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error,r2_score
import matplotlib.pyplot as plt

df=pd.read_csv("encoded_car_data.csv")
print(df.head())
x=df[['enginesize','horsepower','citympg','highwaympg']]
y=df['price']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
linear_model=Pipeline([
    ('scaler',StandardScaler()),
    ('model',LinearRegression())
])
linear_model.fit(x_train,y_train)
y_pred_linear=linear_model.predict(x_test)

poly_model=Pipeline([
    ('poly',PolynomialFeatures(degree=2)),
    ('scaler',StandardScaler()),
    ('model',LinearRegression())
])
poly_model.fit(x_train,y_train)
y_pred_poly=poly_model.predict(x_test)
print("Linear Regression:")
print(f"MSE: {mean_squared_error(y_test,y_pred_linear):.2f}")
print(f"R Square: {r2_score(y_test,y_pred_linear):.2f}")

print("\nPolynomial Regression:")
print(f"MSE: {mean_squared_error(y_test,y_pred_poly):.2f}")
print(f"R Square: {r2_score(y_test,y_pred_poly):.2f}")

plt.figure(figsize=(10, 5))
plt.scatter(y_test, y_pred_linear, label='Linear', alpha=0.6)
plt.scatter(y_test, y_pred_poly, label='Polynomial (degree=2)', alpha=0.6)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r', label='Perfect Prediction')
plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
plt.title("Linear vs Polynomial Regression")
plt.legend()
plt.show()
```

Output:

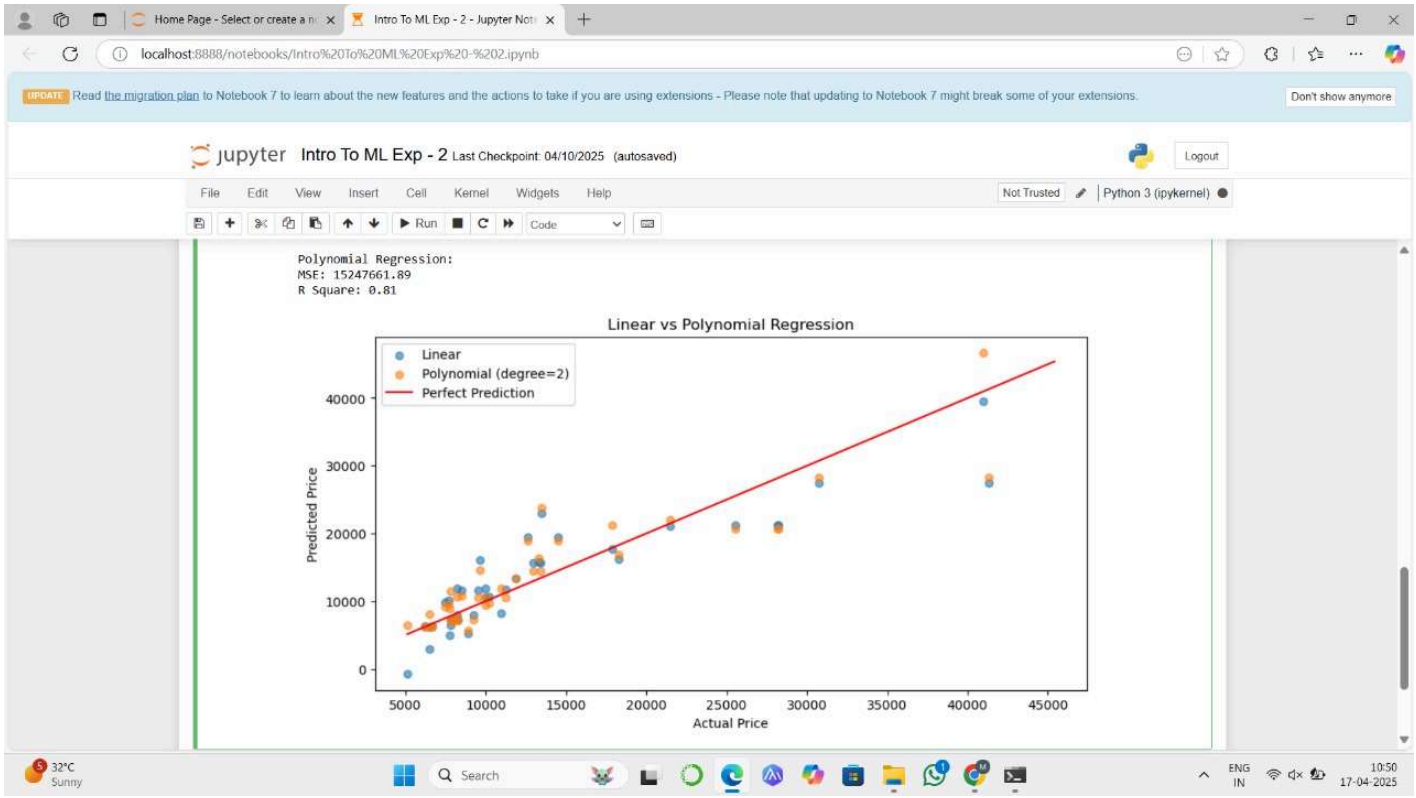
 simple linear regression model for predicting the marks scored

```
   diesel  gas  std  turbo  convertible  hardtop  hatchback  sedan  wagon  \
0     0.0  1.0  1.0    0.0          1.0      0.0        0.0    0.0    0.0
1     0.0  1.0  1.0    0.0          1.0      0.0        0.0    0.0    0.0
2     0.0  1.0  1.0    0.0          0.0      0.0        1.0    0.0    0.0
3     0.0  1.0  1.0    0.0          0.0      0.0        0.0    1.0    0.0
4     0.0  1.0  1.0    0.0          0.0      0.0        0.0    1.0    0.0

   4wd  ...  wheelbase  curbweight  enginesize  boreratio  horsepower  \
0  0.0  ...       88.6      2548.0       130.0       3.47       111.0
1  0.0  ...       88.6      2548.0       130.0       3.47       111.0
2  0.0  ...       94.5      2823.0       152.0       2.68       154.0
3  0.0  ...       99.8      2337.0       109.0       3.19       102.0
4  1.0  ...       99.4      2824.0       136.0       3.19       115.0

   carlength  carwidth  citympg  highwaympg    price
0      168.8      64.1     21.0        27.0  13495.0
1      168.8      64.1     21.0        27.0  16500.0
2      171.2      65.5     19.0        26.0  16500.0
3      176.6      66.2     24.0        30.0  13950.0
4      176.6      66.4     18.0        22.0  17450.0

[5 rows x 36 columns]
Linear Regression:
MSE: 16471505.90
R Square: 0.79
```



Result:

Thus, the program to implement Linear and Polynomial Regression models for predicting car prices was written and verified using Python programming.