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. Load the car price dataset, preprocess the data (handle missing values, encode categorical variables), and split it into training and testing sets.
2. Train a Linear Regression model using the training data and predict car prices for the test data.
3. Transform features into polynomial features, train a Polynomial Regression model, and predict prices.
4. Evaluate and compare both models using metrics like MAE, MSE, and R^2 score to select the best model.

Program:

```
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
from sklearn.metrics import mean_absolute_error, r2_score
import matplotlib.pyplot as plt

# Load data
df = pd.read_csv('encoded_car_data (1).csv')
print(df.head())

# Select features & target
X = df.drop('price', axis=1)
y = df['price']
print(df.head())

# Select features & target
X = df[['enginesize', 'horsepower', 'citympg', 'highwaympg']]
y = df['price']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 1. Linear Regression (with scaling)
lr = Pipeline([
    ('scaler', StandardScaler()),
    ('model', LinearRegression())
])
lr.fit(X_train, y_train)
y_pred_linear = lr.predict(X_test)

# 2. Polynomial Regression (degree=2)
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)
# Evaluate models
print('Name: Anise Kinsella A')
print('Reg. No: 212225040021')
print("Linear Regression:")
mse=mean_squared_error(y_test,y_pred_linear)
print('MSE= ',mean_squared_error(y_test,y_pred_linear))
r2score=r2_score(y_test,y_pred_linear)
print('MAE= ',mean_absolute_error(y_test,y_pred_linear))
r2score=r2_score(y_test,y_pred_linear)

print("\nPolynomial Regression:")
print(f"MSE: {mean_squared_error(y_test, y_pred_poly):.2f}")
print(f"R²: {r2_score(y_test, y_pred_poly):.2f}")

# Plot actual vs predicted
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:

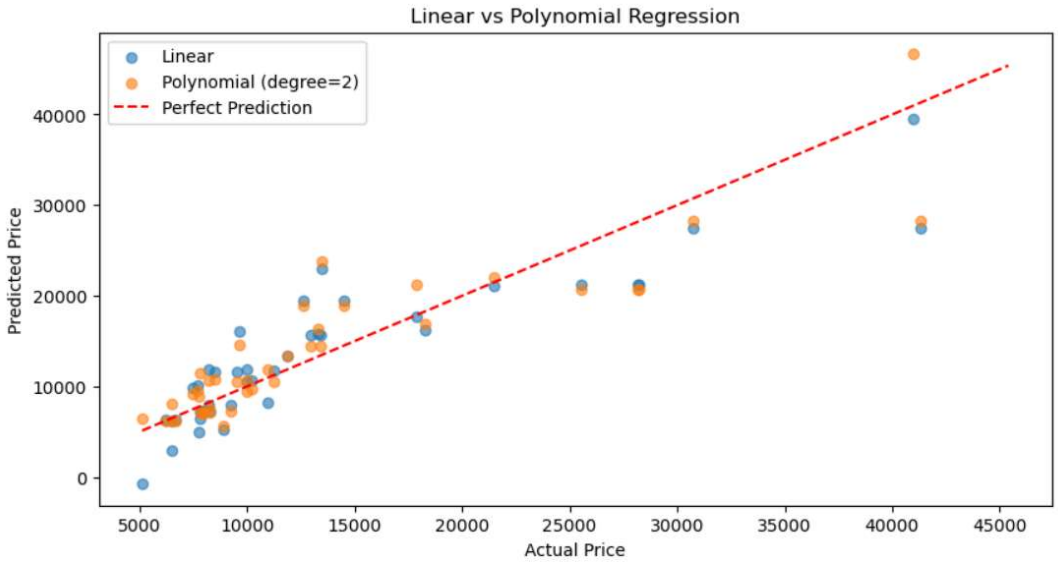
```
   diesel  gas  std  turbo  convertible  hardtop  hatchback  sedan  wagon  \
0     0.0  1.0  1.0    0.0          1.0      0.0        0.0    0.0    0.0
1     0.0  1.0  1.0    0.0          1.0      0.0        0.0    0.0    0.0
2     0.0  1.0  1.0    0.0          0.0      0.0        1.0    0.0    0.0
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]
Name: Anise Kinsella A
Reg. No: 212225040021
Linear Regression:
MSE=  16471505.900042146
MAE=  2892.628134137953

Polynomial Regression:
MSE: 15247661.89
R²: 0.81
```



Result:

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