BLENDED_LERNING

Implementation-of-Multiple-Linear-Regression-Model-with-Cross-Validation-for-Predicting-Car-Prices

AIM:

To write a program to predict the price of cars using a multiple linear regression model and evaluate the model performance using cross-validation.

Equipments Required:

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

Algorithm

1.Import Libraries: Bring in the necessary libraries.

2.Load the Dataset: Load the dataset into your environment.

3.Data Preprocessing: Handle any missing data and encode categorical variables as needed.

4.Define Features and Target: Split the dataset into features (X) and the target variable (y).

5.Split Data: Divide the dataset into training and testing sets.

6.Build Multiple Linear Regression Model: Initialize and create a multiple linear regression model.

7.Train the Model: Fit the model to the training data.

8.Evaluate Performance: Assess the model's performance using cross-validation.

9.Display Model Parameters: Output the model’s coefficients and intercept.

10.Make Predictions & Compare: Predict outcomes and compare them to the actual values.

Program:

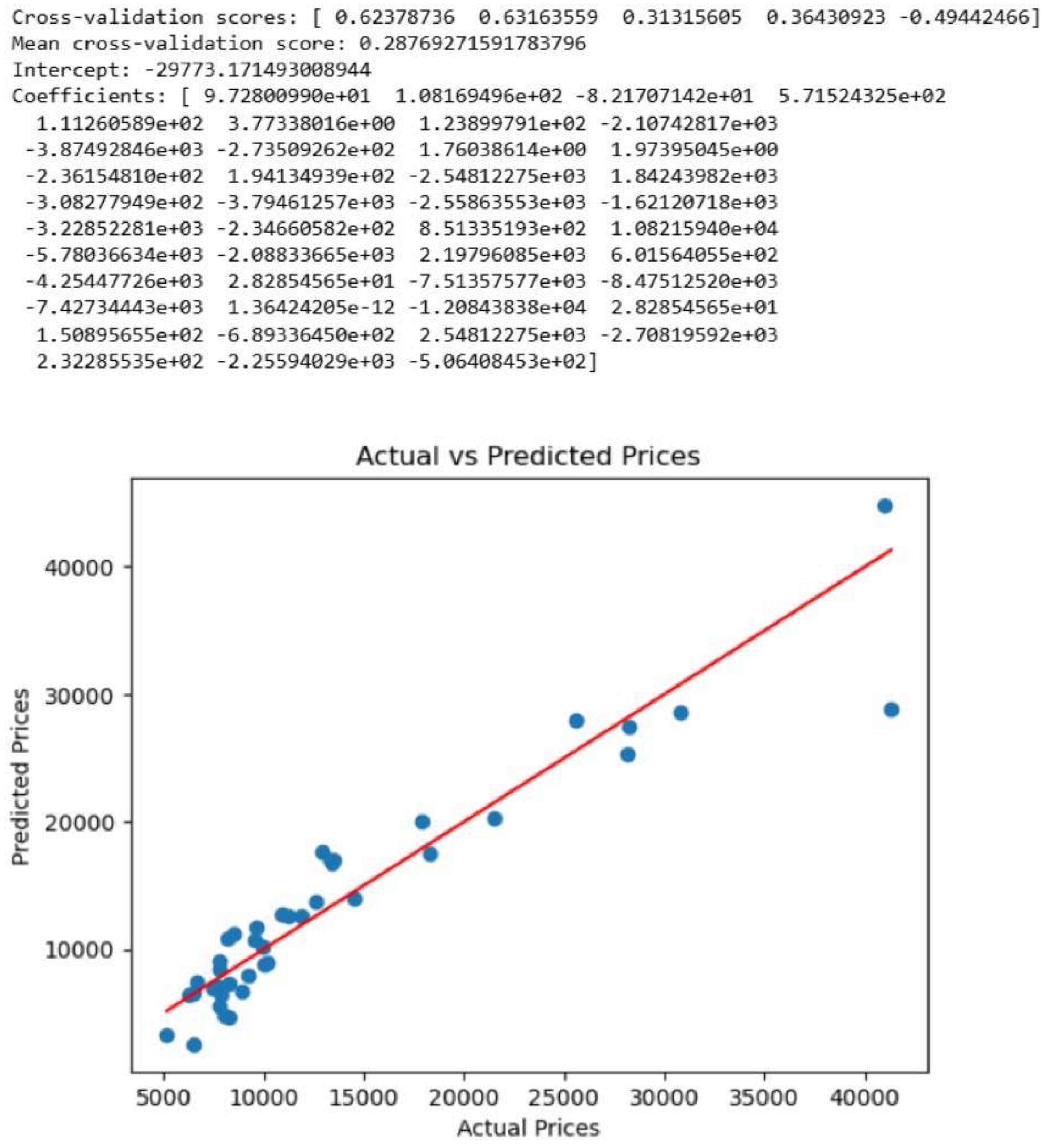
```
/*
Program to implement the multiple linear regression model for predicting car prices wi
Developed by: GANESH PRABHU J
```

```
RegisterNumber: 212223220023
*/
```

```
# Importing necessary libraries
import pandas as pd
import numpy as np
import statsmodels.api as sm
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud

# Data preprocessing
# Dropping unnecessary columns and handling categorical variables
data = data.drop(['CarName', 'car_ID'], axis=1)
data = pd.get_dummies(data, drop_first=True)

# Splitting the data into features and target variable
X = data.drop('price', axis=1)
y = data['price']

# 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=

# Creating the model
model = LinearRegression()

# Fitting the model on the training data
model.fit(X_train, y_train)

# Evaluating model performance using cross-validation
cv_scores = cross_val_score(model, X, y, cv=5)

# Printing cross-validation scores
print("Cross-validation scores:", cv_scores)
print("Mean cross-validation score:", cv_scores.mean())

# Print model coefficients
print("Intercept:", model.intercept_)
print("Coefficients:", model.coef_)

# Make predictions
predictions = model.predict(X_test)

# Visualizing actual vs predicted prices
plt.scatter(y_test, predictions)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted Prices")
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red')  # Perfe
plt.show()
```

Output:

simple linear regression model for predicting the marks scored



```
Cross-validation scores: [ 0.62378736  0.63163559  0.31315605  0.36430923 -0.49442466]
Mean cross-validation score: 0.28769271591783796
Intercept: -29773.171493008944
Coefficients: [ 9.72800990e+01  1.08169496e+02 -8.21707142e+01  5.71524325e+02
  1.11260589e+02  3.77338016e+00  1.23899791e+02 -2.10742817e+03
 -3.87492846e+03 -2.73509262e+02  1.76038614e+00  1.97395045e+00
 -2.36154810e+02  1.94134939e+02 -2.54812275e+03  1.84243982e+03
 -3.08277949e+02 -3.79461257e+03 -2.55863553e+03 -1.62120718e+03
 -3.22852281e+03 -2.34660582e+02  8.51335193e+02  1.08215940e+04
 -5.78036634e+03 -2.08833665e+03  2.19796085e+03  6.01564055e+02
 -4.25447726e+03  2.82854565e+01 -7.51357577e+03 -8.47512520e+03
 -7.42734443e+03  1.36424205e-12 -1.20843838e+04  2.82854565e+01
  1.50895655e+02 -6.89336450e+02  2.54812275e+03 -2.70819592e+03
  2.32285535e+02 -2.25594029e+03 -5.06408453e+02]
```

Result:

Thus, the program to implement the multiple linear regression model with cross-validation for predicting car prices is written and verified using Python programming.