

Practical No: 8

Least Square Regression Algorithm | Logistic Regression algorithm

AIM: For a given set of training data examples stored in a CSV. File implement Least Square Regression Algorithm.

Description:

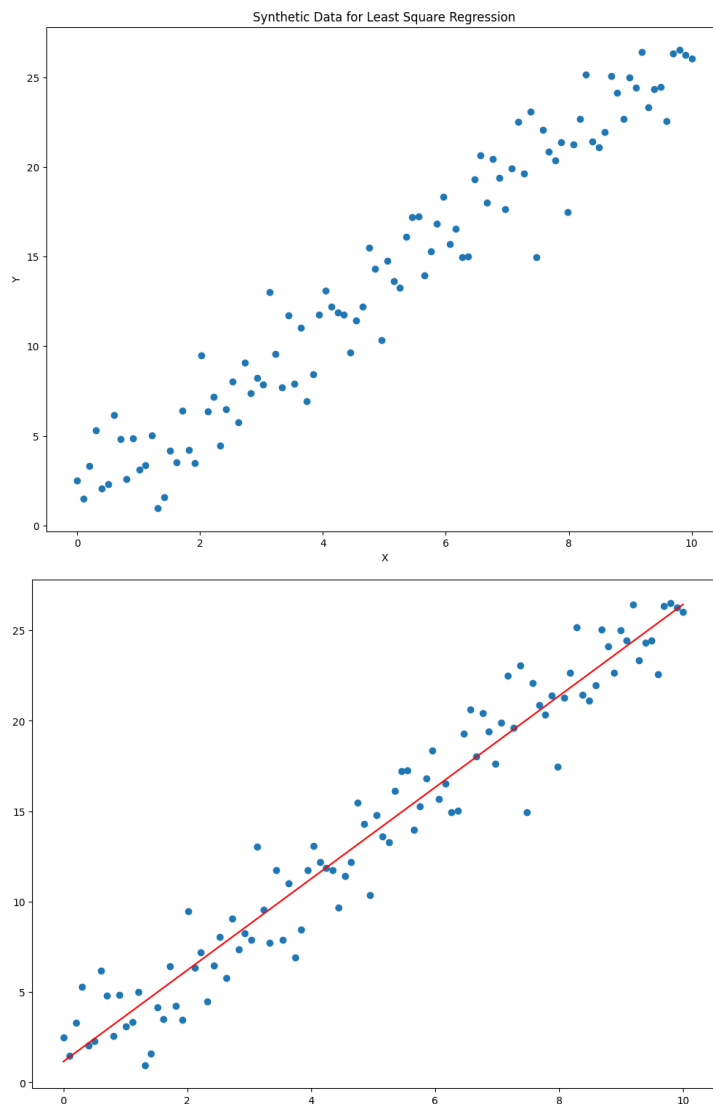
Least Square Regression is a linear regression algorithm used to find the best-fitting line through data points by minimizing the sum of squared differences between observed and predicted values. It calculates the slope and intercept that minimize the vertical distances of data points from the regression line, providing a straightforward method for modeling linear relationships.

Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (12.0, 9.0)

np.random.seed(42)
X = np.linspace(0, 10, 100)
Y = 2.5 * X + 1.5 + np.random.normal(0, 2, 100)
data = pd.DataFrame({"X": X, "Y": Y})
plt.scatter(X, Y)
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Synthetic Data for Least Square Regression")
plt.show()

import numpy as np
# Assuming X and Y are your data arrays
X_mean = np.mean(X)
Y_mean = np.mean(Y)
num = 0
den = 0
for i in range(len(X)):
    num += (X[i] - X_mean) * (Y[i] - Y_mean)
    den += (X[i] - X_mean) ** 2
m = num / den
c = Y_mean - m * X_mean
print(m, c)
# Making predictions
Y_pred = m * X + c
plt.scatter(X, Y) # actual
plt.plot([min(X), max(X)], [min(Y_pred), max(Y_pred)], color="red") # prediction
plt.show()
```

OUTPUT:**Learnings**

The provided Python script aims to implement the Least Square Regression algorithm for a given set of training data stored in a CSV file. The synthetic data is generated, and a scatter plot is visualized to showcase the relationship between the independent variable (X) and the dependent variable (Y). The Least Square Regression is then applied manually, calculating the slope (m) and intercept (c) to define the best-fit line. Finally, predictions are made using these parameters, and both the actual data points and the regression line are plotted for visualization.

B) For a given set of training data examples stored in a .CSV file implement Logistic Regression algorithm

Code with output

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('https://raw.githubusercontent.com/mk-gurucharan/Classification/master/DMVWrittenTests.csv')
X = dataset.iloc[:, [0, 1]].values
Y = dataset.iloc[:, 2].values
dataset.head(5)
```

[1] ✓ 1.8s

...	DMV_Test_1	DMV_Test_2	Results
0	34.623660	78.024693	0
1	30.286711	43.894998	0
2	35.847409	72.902198	0
3	60.182599	86.308552	1
4	79.032736	75.344376	1

```
# Splitting the dataset into the training set and test set.
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=0)
```

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
# Training the logistic regression model on the training set
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier.fit(X_train, Y_train)
```

```
# Predicting the test set results.
```

```
y_pred = classifier.predict(X_test)
```

```
y_pred
```

```
[3] ✓ 0.0s
```

```
... array([1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0,
          1, 1, 0], dtype=int64)
```

```
# Confusion Matrix and Accuracy.
```

```
from sklearn.metrics import confusion_matrix
```

```
cm = confusion_matrix(Y_test, y_pred)
```

```
from sklearn.metrics import accuracy_score
```

```
print("Accuracy:", accuracy_score(Y_test, y_pred))
```

```
cm
```

```
[4] ✓ 0.0s
```

```
... Accuracy: 0.88
```

```
... array([[11, 0],
          [ 3, 11]], dtype=int64)
```

Learnings

This code implements a logistic regression model for classifying DMV written test results. It utilizes the scikit-learn library to preprocess the dataset, splitting it into training and testing sets. Feature scaling is applied for standardization, and the logistic regression model is trained on the training set. The predictions are then evaluated using a confusion matrix, providing insights into the model's performance. The accuracy score is calculated, representing the proportion of correctly classified instances. This code demonstrates a basic yet effective application of logistic regression for binary classification, essential in scenarios like DMV written test outcome prediction based on specific features.