Experiment No.3

Title: Predicting missing data using regression modelling

Batch: 1 Roll No.: 16010420133 Experiment No. 3

Aim: Predict missing data using regression modelling.

Resources needed: Any free and Open online statistical analysis tools

---

Results:

Code:

Linear Regression:

```
import pandas as pd

x_column = []
y_column = []
csv_file = pd.read_csv('/Users/Soumen/Downloads/SEM_4
College_Stuff/HON/FDS/runs.csv')
csv_file_x = csv_file[csv_file['position_sec1'].notna()]
csv_file_y = csv_file[csv_file['time1'].notna()]

for i in range(0,10):
    x_column.append(float(csv_file_x['position_sec1'][i]))
    y_column.append(float(csv_file_y['time1'][i]))

x_sum = 0
y_sum = 0
for i in x_column:
    x_sum += i
for i in y_column:
    y_sum += i

x_mean = x_sum / len(x_column)
y_mean = y_sum / len(y_column)

num = 0
denum = 0
for i in range(len(x_column)):
    num += (x_column[i] - x_mean) * (y_column[i] - y_mean)
for i in range(len(y_column)):
    denum += (x_column[i] - x_mean) ** 2

w1 = num / denum
w0 = y_mean - (w1 * x_mean)
```

```
x = float(input("Enter the position of the horse: "))
y = w0 + (w1 * x)
print("The predicted value that horse must have taken ",y,"seconds to be at
",int(x),"position")

y_index = 0
difference_in_y_sum = 0
for i in x_column:
    predicted_y = w0 + (w1 * i)
    difference_in_y = abs(y_column[y_index] - predicted_y)
    y_index += 1
    difference_in_y_sum += difference_in_y

error = difference_in_y_sum / 10
print("The mean absolute error in the predicted value is ",error)
```

Output:

```
Enter the position of the horse: 6
The predicted value that horse must have taken  13.967809855649577 seconds to be
 at  6 position
The mean absolute error in the predicted value is  0.10760378297660492
```

Multiple Linear Regression:

```
import pandas as pd
import numpy as np

x_column2 = []
x_column3 = []
x_column4 = []
y_column = []
csv_file = pd.read_csv('/Users/Soumen/Downloads/SEM_4
College_Stuff/HON/FDS/runs.csv')
csv_file_x2 = csv_file[csv_file['position_sec1'].notna()]
csv_file_x3 = csv_file[csv_file['position_sec2'].notna()]
csv_file_x4 = csv_file[csv_file['position_sec3'].notna()]
csv_file_y = csv_file[csv_file['time1'].notna()]

for i in range(0,len(csv_file_x4)):
    x_column2.append(csv_file_x2['position_sec1'][i])
    x_column3.append(csv_file_x3['position_sec2'][i])
    x_column4.append(csv_file_x4['position_sec3'][i])
    y_column.append(csv_file_y['time1'][i])
x_column1 = [1] * len(csv_file_x2)
```

```
x_column1 = np.array(x_column1).reshape(len(csv_file_x3),1)
x_column2 = np.array(x_column2).reshape(len(csv_file_x2),1)
x_column3 = np.array(x_column3).reshape(len(csv_file_x3),1)
x_column4 = np.array(x_column4).reshape(len(csv_file_x4),1)

x_column = np.hstack([x_column1,x_column2,x_column3,x_column4])
y_column = np.array(y_column).reshape(len(csv_file_y),1)

x_transpose = np.transpose(x_column)
x_transpose_x__inverse = np.linalg.inv(np.dot(x_transpose, x_column))
x_transpose_x__inverse_x_transpose = np.dot(x_transpose_x__inverse, x_transpose)
x_transpose_x__inverse_x_transpose_y = np.dot(x_transpose_x__inverse_x_transpose,
y_column)

xi1 = input("Enter a predictor variable: ")
xi2 = input("Enter a predictor variable: ")
xi3 = input("Enter a predictor variable: ")
y = x_transpose_x__inverse_x_transpose_y[0][0] +
((x_transpose_x__inverse_x_transpose_y[1][0]) * float(xi1)) +
((x_transpose_x__inverse_x_transpose_y[2][0]) * float(xi2)) +
((x_transpose_x__inverse_x_transpose_y[3][0]) * float(xi3))
print("The predicted value is: " , y)

difference_y_sum = 0
for i in range(0, len(csv_file_y)):
    x1_value = x_column2[i]
    x2_value = x_column3[i]
    x3_value = x_column4[i]
    y_value = y_column[i]
    predicted_y = x_transpose_x__inverse_x_transpose_y[0][0] +
((x_transpose_x__inverse_x_transpose_y[1][0]) * float(x1_value)) +
((x_transpose_x__inverse_x_transpose_y[2][0]) * float(x2_value)) +
((x_transpose_x__inverse_x_transpose_y[3][0]) * float(x3_value))
    difference_in_y = abs(y_value - predicted_y)
    difference_y_sum += difference_in_y

error = difference_y_sum / len(csv_file_y)
print("The mean absolute error in the predicted value is ",float(error))
```

Output:

```
Enter a predictor variable: 6
Enter a predictor variable: 4
Enter a predictor variable: 6
The predicted value is:  21.076068343858957
The mean absolute error in the predicted value is  5.693148783370081
```

Questions:

1. How will you choose between linear regression and non-linear regression?

A)

Linear models must follow one very particular form:
Dependent variable = constant + parameter * IV + … + parameter * IV
The form is linear in the parameters because all terms are either the constant or a parameter multiplied by an independent variable (IV). A linear regression equation simply sums the terms. While the model must be linear in the parameters, you can raise an independent variable by an exponent to fit a curve. For instance, you can include a squared or cubed term.
Nonlinear regression models are anything that doesn't follow this one form. While both types of models can fit curvature, nonlinear regression is much more flexible in the shapes of the curves that it can fit.

2. Explain the nature or characteristics of a dataset where we can apply regression imputation.

A)

Regression imputation fits a statistical model on a variable with missing values.
Predictions of this regression model are used to substitute the missing values in this variable.
There should be a good number of records in the dataset so that we can get a good accuracy with more data and at least 2 attributes should be related to each other to perform regression on it.
Regression imputation consists of two subsequent steps:
A linear regression model is estimated on the basis of observed values in the target variable Y and some explanatory variables X.
The model is used to predict values for the missing cases in Y. Missing values of Y are then replaced on the basis of these predictions.

Outcomes:

CO2: Comprehend descriptive and proximity measures of data

Conclusion:

We were able to understand the concept of linear regression model and multiple linear regression model. In our dataset for a particular column

or columns we found the linear regression equation and multiple linear regression equation and also calculated the error for the same

---

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition