Miras M

Regression Analysis Project

According to the ‘distr.csv’ my dataset number is 17, so I worked with ‘data17.txt’. For this project I mainly used jupyter notebook and python libraries like: *pandas, numpy, matplotlib, sklearn* and *statsmodels*.

Before I started, I imported ‘data17.txt’ as pandas dataframe and added the name for every column, from x1 to x13 and y.

* To estimate covariance and correlation between Xi and target feature Y, I used *numpy.cov* and *pandas.DataFrame.corr* functions and produced the output with for loop. The output:



* To calculate absolute value of correlation, I only added *abs* function to the output that I produced before:



* **Answer to Question 1: We can deduce that x2 with absolute correlation 0.8471 and x13 with absolute correlation 0.8476 are relevant for prediction.**
* To calculate the correlation matrix C, I used the function *corr()* for all dataframe without y value and transferred it to the matrix with the *to\_numpy* function:





* **Answer to Question 2: From the table above, I can deduce that there is strong linear relationship between the variables: x3 and x6(0.996283), x7 and x8(0.993901), x4 and x9(0.995679), x5 and x11(0.995269), x1 and x12(0.995722), x2 and x13(0.994414).**
* Python code includes *sklearn* where I trained the data with *fit* function and found the b0 and all other coefficients with similar function names.

So the equation will look like:

Y=9.987224967793953+2.65189441e-03\***x1**-7.99645122\***x2**-1.04091817e-02\***x3**+4.01003511\***x4**-1.14028425e-02\***x5**-4.90312608e-01\***x6**-1.55815206e-02\***x7**+1.13999462e-01\***x8**-1.02390735e-02\***x9**-6.11799055e-04\***x10**-1.99092302e+00\***x11**+9.99191309e-01\***x12**-4.11159056e-03\***x13**

* Coefficients obtained in the 6th step.
* To estimate the variance of noise *sklearn.metrics.mean\_squared\_error* function to the y and y\_pred values obtained from the *predict* method. MSE = 0.04143876243119763
* Here I used the formula provided and *matmul* function to obtain t values:



* Here I compared the t values with 1.968293 and obtained irrelevant x values.
* **Answer to Question 3: Variables that can be discarded, by the previous test, are:**



* Here I used *statsmodels.api* library where it becomes easier to draw the qq plot for my data:



* **Answer to Question 4: Yes, the error is normally distributed based on the plot from step 12.**
* Here I used the *scatter* method to draw a plot from the equation obtained in the step 6:



* **Answer to Question 5: Distributed residuals are even, so we have homoscedasticity, which means that errors don’t depend on the predicted values.**
* *Score* method gave me the result based on the x and y: 0.9999943873917985
* **Answer to Question 6: R^2 is very close to one, which means positive linear relationship and model explains more variability. It means that the model almost perfectly solved the problem.**