Statistical Learning – Data science - 2020/21 – Exercise 2 - 25/03/2021

Exercise 2: Analysis of Prostate Cancer dataset - linear regression model
Please, execute the following tasks and provide answers to the proposed questions.

1. Open the webpage of the book "The Elements of Statistical Learning", go to the "Data" section and download the info and data files for the dataset called Prostate

- Hint: https://web.stanford.edu/~hastie/ElemStatLearn/

2. Open the file *prostate.info.txt*

- How many predictors are present in the dataset?
- What are those names?
- How many responses are present in the dataset?
- What are their names?
- How did the authors split the dataset in training and test set?
- Hint: please, refer also to Section 3.2.1 (page 49) of the book "The Elements of Statistical Learning" to gather this information

3. Open the file *prostate.data* by a text editor or a spreadsheet and have a quick look at the data

- How many observations are present?
- Which is the symbol used to separate the columns?

4. Open Kaggle, generate a new kernel and give it the name "SL_EX2(L5)_ProstateCancer_Surname"

5. Add the dataset prostate.data to the kernel

- Hint: See the Add Dataset button on the right
- Hint: use import option "Convert tabular files to csv"

6. Run the first cell of the kernel to check if the data file is present in folder ../input

7. Add to the first cell new lines to load the following libraries: seaborn, matplotlib.pyplot, sklearn.linear_model.LinearRegression

8. Add a Markdown cell on top of the notebook, copy and paste in it the text of this exercise and provide in the same cell the answers to the questions that you get step-by-step.

9. Load the Prostate Cancer dataset into a Pandas DataFrame variable called *data*

- How can you say Python to use the right separator between columns?

10. Display the number of rows and columns of variable *data*

11. Show the first 5 rows of the dataset

12. Remove the first column of the dataset which contains observation indices

13. Save column *train* in a new variable called *train* and having type *Series* (the Pandas data structure used to represent DataFrame columns), then drop the column *train* from the *data* DataFrame

14. Save column *lpsa* in a new variable called *lpsa* and having type *Series* (the Pandas data structure used to represent DataFrame columns), then drop the column *lpsa* from the *data* DataFrame and save the result in a new DataFrame called *predictors*

- How many predictors are available?

15. Check the presence of missing values in the data variable

- How many missing values are there? In which columns?
- Which types do the variable have?

16. Show histograms of all variables in a single figure

- Use argument *figsize* to enlarge the figure if needed

17. Show the basic statistics (min, max, mean, quartiles, etc. for each variable) in *data*

18. Generate a new DataFrame called *dataTrain* and containing only the rows of *data* in which the *train* variable has value "T"

- Hint: use the *loc* attribute of DataFrame to access a groups of rows and columns by label(s) or boolean arrays
- How many rows and columns does *dataTrain* have?

19. Generate a new DataFrame called *dataTest* and containing only the rows of *data* in which the *train* variable has value "F"

- How many rows and columns does *dataTest* have?

20. Generate a new Series called *lpsaTrain* and containing only the values of variable *lpsa* in which the *train* variable has value "T"

- How many valuses does lpsaTrain have?

21. Generate a new Series called *lpsaTest* and containing only the values of variable *lpsa* in which the *train* variable has value "F"

- How many valuses does lpsaTest have?

22. Show the correlation matrix among all the variables in *dataTrain*

- Hint: use the correct method in DataFrame
- Hint: check if the values in the matrix correspond to those in Table 3.1 of the book

23. Drop the column *lpsa* from the *dataTrain* DataFrame and save the result in a new DataFrame called *predictorsTrain*

24. Drop the column *lpsa* from the *dataTest* DataFrame and save the result in a new DataFrame called *predictorsTest*

25. Generate a new DataFrame called predictorsTrain_std and containing the standardized variables of DataFrame predictorsTrain

- Hint: compute the mean of each column and save them in variable *predictorsTrainMeans*
- Hint: compute the standard deviation of each column and save them in variable *predictorsTrainStds*
- Hint: compute the standardization of each variable by the formula (predictorsTrain-predictorsTrainMeans)/predictorsTrainStd

26. Show the histogram of each variables of predictorsTrain_std in a single figure

- Use argument *figsize* to enlarge the figure if needed
- Hint: which kind of difference can you see in the histograms?

27. Generate a linear regression model using predictorsTrain_std as dependent variables and *lpsaTrain* as independent variable

- Hint: find a function for linear regression model learning in sklearn (fit)
- How do you set parameter fit_intercept? Why?
- How do you set parameter normalize? Why? Can this parameter be used to simplify the generation of the predictor matrix?

28. Show the parameters of the linear regression model computed above. Compare the parameters with those shown in Table 3.2 of the book (page 50)

29. Compute the *coefficient of determination* of the prediction

30. Compute the standard errors, the Z scores (Student's t statistics) and the related p-values

- Hint: use library *statsmodels* instead of sklearn
- Hint: compare the results with those in Table 3.2 of the book (page 50)