|  |  |  |
| --- | --- | --- |
|  |  |  |
| Ramses Meza  April 2021  <https://github.com/lordtable/UTDSGC_C3T1> |  | Data Analytics Course 3.1: Predicting Customer Preferences |

Business Acumen

Given the enormous business impact of Data Science, it is necessary for Blackwell Electronics to expand its Data Science capabilities beyond Python, acknowledging that the programming language R is also a valid, open-source tool worth incorporating into the company toolset. In order to develop the internal expertise on the language, I have been commissioned with the execution of a hands-on R/RStudio tutorial in order to gain familiarity with the tool in preparation for imminent data science assignments.

Overall Tutorial Description

Installing R and its IDE, RStudio, was very simple and straightforward. RStudio provides a “feel” quite similar to that of MATLAB, allowing in one view to have the code and the acting variables as well. The tutorial was useful, especially on guiding on the most basic Data Science functionalities. I strongly recommend any newcomer to first try this tutorial before going into more elaborate aspects of R.

Tutorial Data Exercise #1

The first tutorial exercise was based on a provided dataset that was loaded into RStudio, and consisted of 50 observations and three (3) features:

* ***Name.of.car***: brand of the car
* ***Speed.of.car***: Observed car speed
* ***Distance.of.car***: Distance traveled by the car.

The exercise consisted on completing basic steps in order to gain familiarity with the data, identify issues and correct them when found. While screening the features, it was noticed that inspection of the ***name.of.car*** feature via histograms is not possible, since R only performs histograms on strictly numeric data. It was also noticed that the other two variables were highly correlated when the observation were displayed on a scatter plot. QQ plots for the ***speed*** and ***distance*** variables were built to assess whether each is normally distributed, finding out that ***speed*** seems to be much closer to be normally distributed than ***distance***, inference that could be made from their histograms.

Names on headers were edited for ease of reading and shorter length, making them less error-prone. No missing values were found on the dataset, nor repeated instances.

A linear model was created in order to predict ***distance*** using ***speed*** as predictor. It was trained using 70% of the samples, while 30% were kept apart for blind testing. Once built, the model yielded the statistics shown on Table 1:

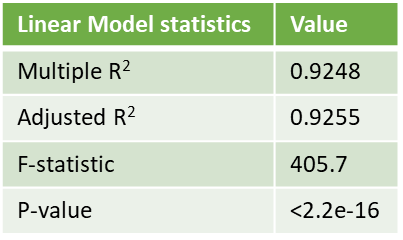


Table 1: Summarized statistics of the trained linear regression model targeting the feature ***distance*** from the feature ***speed***.

R2 measures the strength of the linear relationship between the predictor variable(s) and the dependent variable, which in this case it is strong since it is very close to 1. The F-statistic or F-test, interpreted in combination with the p-value, measure whether or not the linear model provides a better fit to a dataset than a model with no predictor variables, therefore providing a sense of statistical significance. It is said than a F-statistic much greater than 1 and a p-value less than the selected significance level (commonly either 0.01, 0.05 or 0.1) are indicative that a model is statistically significant, therefore its outcomes are not a product of chance. For this exercise, we can attest that the model is statistically significant.

Finally, ***distance*** values were predicted using the calibrated model and the ***speed*** values on the test dataset, and the predicted and actual ***distance*** values were cross plotted as shown on Figure 1, a reference perfect-fit line (blue) was added to the plot to visually assess how good the prediction is, which in this case is very good.

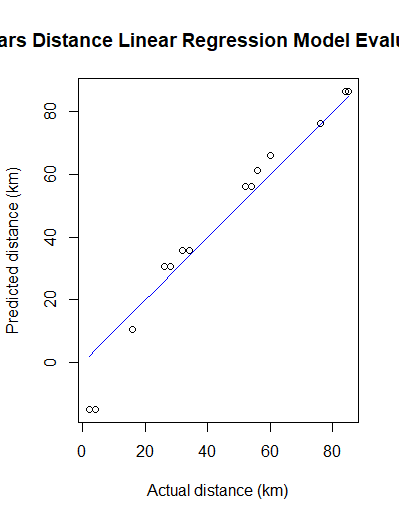


Figure 1: Cross-plot of predicted vs actual distance (hollow circles), with a perfect-fit straight line (blue).

Tutorial Exercise #2

The second tutorial exercise was based on a provided dataset of the famous *Iris* dataset, that was loaded into RStudio, and consisted of 150 observations and six (6) features:

* ***X***: sequential number (?)
* Sepal.Length
* ***Sepal.Width***
* ***Petal.Length***
* ***Petal.width***
* ***Species***

The exercise also consisted on completing basic steps in order to gain familiarity with the data, identify issues and correct them if found. Based on the script instructions provided on this tutorial, many errors were due to misspelling and/or incorrect syntaxis (i.e., missing quotation marks when reading a .csv file, or not-needed quotation marks when loading a library).

Histogram of ***Species*** could only be executed when the feature data type was converted to *factor*. Plotting ***Sepal.Length*** yielded a plot of the values of this feature on the y-axis and each sample index on the x-axis.

Converting the ***Species*** data type from *factor* to *numeric* provided discrete integers 1,2 and 3, each corresponding to a former species name: finding an automated way to have R yield a sort of equivalence table is pending.

No missing values found on the dataset, nor repeated instances.

A linear model was created in order to predict ***Petal\_Length*** using ***Petal\_Width*** as predictor. It was first trained using 20% of the samples, while 80% were kept apart for blind testing, as instructed by the tutorial. However, predictions could not be made since an error was persisting, stating that there was a dataset dimension mistmatch between the testing set predictor and the model. It was initially thought that the small portion of samples chosen to be on the training set (20%) was the culprit, so the partition was flipped to no avail. After several attempts on data partitioning, inspecting data types, creating dataset copies, etc., I renamed the features in order to make them simpler and avoid special characters. I could successfully calibrate the model using a dataset with such edits and 80% of the samples as training set. Once built, the model yielded the statistics shown on Table 2:

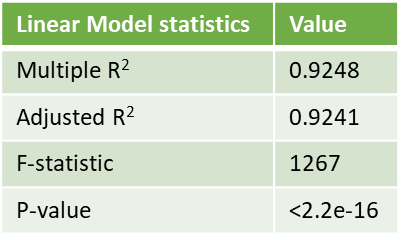


Table 2: Summarized statistics of the trained linear regression model targeting the feature ***Petal\_Length*** from the feature ***Petal\_Width***.

As discussed on exercise #1, based on exercise #2 summary statistics for the linear regression model, we can conclude that there is a very strong correlation between predictor and target feature, and the model results are statistically significant. The quality of the prediction can be visually assessed based on the scatter-plot on Figure 2, with a red line representing a perfect-fit line, and the hollow scatter points are the actual & predicted ***Petal\_Length*** points, which tend to lie close to the perfect-fit line.

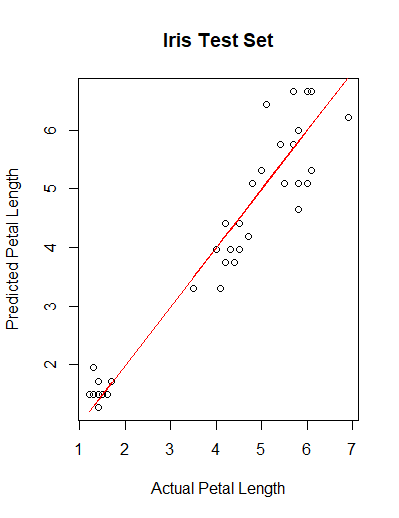


Figure 2: Cross-plot of predicted vs actual ***Petal\_Length*** (hollow circles), with a perfect-fit straight line (red).

Lessons Learnt

* When building a linear model, care needs to be taken in the order the *x* and *y* variables are entered: *y* goes first.
* Changing the seed for the random-numbers generator after the data has been already split doesn’t change that dataset.
* To produce *qqnorm* plots, not only the dataset but the feature of interest must be specified.
* By design, it is harder to keep track of the workflows and errors on RStudio compared to Jupyter notebook, therefore it is highly recommended to keep notes & screen captures continuously, especially to track errors.