Marek Krawczyk

DAT-690

10/21/2022

Programming Revision: Strengthening Your Code

**Creation of the preliminary models:**

In order to test the data thoroughly, Random Forest model was created with two visual Trees, and a breakdown of weight in each independent variable. It was a good start to take the less important variables out of the data set and tackle the ones which truly mattered.

In the first iteration of the Logistic Regression, there was no integration of any kind of scaling, or dummy variables. Just straight data and regression based on it. The model had a hard time understanding the number of categorical variables present in the data.

**Logistic Regression Model:**

The model remained mostly unchanged; the data set did transform a few times to become less cryptic to machine learning algorithm. Data preparation is one of the most important steps. The data was edited based on previous findings, and dummy variables were added in order to improve the accuracy of the model.

*Figure 1: Preparation:*

Graphical user interface, text, application, email

Description automatically generated

From this point, we have the final data set that can be transformed into our training and testing splits. Multiple Linear Regression requires the first split of X and y, our independent variables and our dependent variables respectively. From then on, we must split the data into 4 pieces, a piece to train on the x, y, and test on x,y. We end up with four data frames: X\_train, X\_test, y\_train, y\_test. Our test set is based on 25% of our data size, so in this case 250 cases.

*Figure 2: Train-Test split*Text

Description automatically generated

Next Is the Model creation. Importing a model from a proven platform is a normal occurrence in today’s data science field, these models are highly customizable, and provide amble flexibility without reinventing the wheel. For this task, SKLEARN library was chosen. In order for the model to be useful it must be fitted with data, trained and tested.

*Figure 3: Model Creation and fitment.*  
Text

Description automatically generated

Without predictions finished, next step is to interpret the results. To do this efficiently seaborn plotting library will be used alongside with matplotlib library.

*Figure 3: Results*

Text

Description automatically generated

In order to further drive the understanding of our model we can employ the usage of more metrics and statistics. AUC – Area under the Curve can tell us how effective our algorithm was.

*Figure 4: Results 2*

Chart

Description automatically generated

**Random Forest Model:**   
 In order to baseline our results, two Random Forest Models were drawn up. The data preparation process remained the same, and utilization of the same variables allows us to go directly to augmenting our models.

*Figure 4: Random Forest Model Preparation:*

Text

Description automatically generated

Following the same logic, we are ready to split our data, this time utilizing 30% of our data as testing bed for our predicitons. Fitment of the model is straightforward, since the syntax is shared.

*Figure 5: Train, Test Split and Model Fitment*

Text, letter

Description automatically generated

In order to gather information about our model, it is necessary to break down some important statistics.

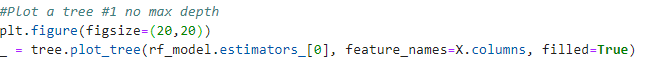
*Figure 6: Importance statistics, along with accuracy and confusion matrix.*

Text

Description automatically generated

Plotting the Tree is also a possibility to see what kind of choices were made.

*Figure 7: Tree #1*



In order to compare this tree to a model that has some boundaries, a random forest with depth of 3 was created. It follows the same steps, with just a simple change in depth.

*Figure 8: Tree #2*

Graphical user interface, text, application, email

Description automatically generated