Tabular Data Science Project submission

By Ron Eliav and Ariel Vetzler

February 2, 2022

1 Abstract

Over the past years, Tabular Data Science has contributed to dramatic advances in the performance of machine learning. One exciting application is feature engineering. Feature engineering is typically formulated by going over the given data sets, and categorizing each column and the operations which can be applied upon. As our experience grow while reading more and more articles about the subject, we started to see similar patterns.

Every sensible person can see after diving into the Feature Engineering field that this technique is the building block of every machine learning project. So combined with the knowledge we learned in class that Feature Engineering is essential to enhancing accuracy and leads to faster model convergence, why isn't it embedded in every project?

The problem that we encountered was the understanding of the data. For feature engineering to work the researcher has to understand what type of data he has and which operations he can apply. Hence, we decided to solve the problem by creating a generic program, that at the flick of a finger adjusts any Tabular data set. Our program includes applying various oneiric operations on numeric columns. After applying the operations on our data set, we attached the results to the Tabular data set as new columns, just as we learned in class. We tested the results with four different data sets with numerical columns, using a Random Forest classifier. On each and every data set the accuracy of the model increased, from 3 percent and all the way to 20 percent.

2 Problem description

The DS pipeline element that we chose to improve, as you can already guess is Feature Engineering. The main problem, as we discussed in the Abstract section is that it takes time to research the specific Tabular data, the project receives as a training set. The research is required even tho eventually the techniques which will be deployed will be nearly identical to other projects. In other words, the creation of features requires some understanding of the data which not everyone posses, and the creation of that features are usually already implemented in other works. Our vision is to improve that element and save time and energy on redundant research by creating a one-time generic program that will do basic operations on the data which will end up by improving the model accuracy.

3 Solution overview

The solution is rather simple. We conducted the simple operations which we saw in class. We separated the categorical and numeric columns and focused on the numeric columns. It's important to stress out that we only use the important features, we choose them by sklearn feature importance function [Sl21].

For each numeric column, we created a new column that we attached to the data set, which will eventually be fed to the classifier. The new columns are a result of oneiric and binary operations that we applied on each column and between the columns. The binary operations included power function and quantile function. For example, for each value in the numeric columns we raised by the power of 2, our tests showed that the power of 2 had the highest throughput than any other const. Additionally, we also quantile each value to 5 buckets between zero and one. The justification to do so is that high values don't add any further value to the classifier, as we discussed in class therefore we bounded the

values. In concern to the binary functions, we used to multiply and divide operations between columns, which means for every two columns we divide and multiply column by column. After calculating the results of the operations, we created a new tabular data set. The data set was created using the old columns and attached to them as new columns, results which were created by our calculations.

Regrading researcher experience, we allow the researcher to diverge from our methods, and choose specific columns he would like to operate on.

Furthermore, we wanted to point out, that as we learned in class, we tried to use only the columns which are normally distributed. Unfortunately, the results were inconclusive so we decided not to further explore that matter.

In order to see our repository, click on the following link: GitHub Link.

4 Related Work

[Hea20] [UK17]

5 Experimental evaluation

In this section, we will try to establish two points. The first point is that feature engineering can increase substantially the classifier accuracy. The second point is that we created generic software for generating features that will help the classifier. We tested our software on various data sets, and we will plot in front of you four important ones. We tested our software on Wine, Cancer, Irises, and Diabetes. The results were remarkable, the software enhances the accuracy of the models and leads to faster convergence in several data sets. The fact that we could improve our classifier in 20 percentages only by changing the data is a game changer for us. It separates between false and right Cancer and Diabetes detection, which can rescue human life. In the following figure, you can see the effect that our program had on the data sets mentioned above. 1 2

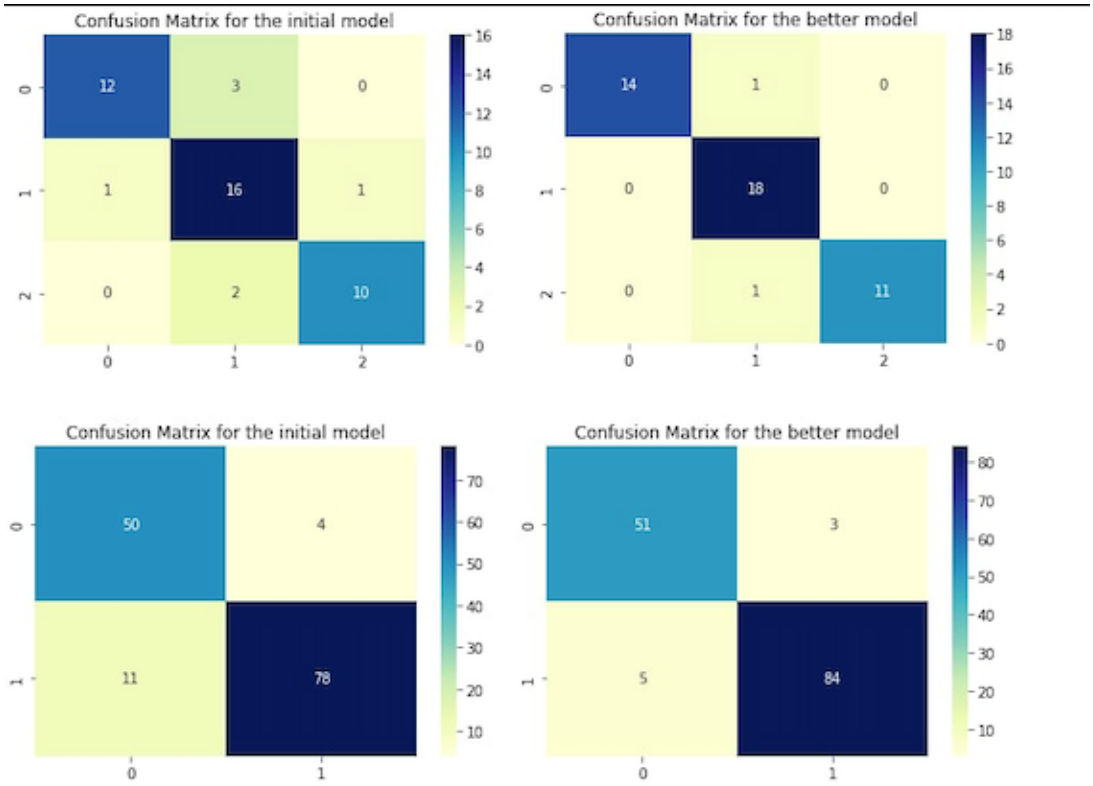

Figure 1: Random Forest Classifier before and after adjusting his data. Data sets: Wine and Cancer respectably. On the second matrix we see better false positive and true negative results.

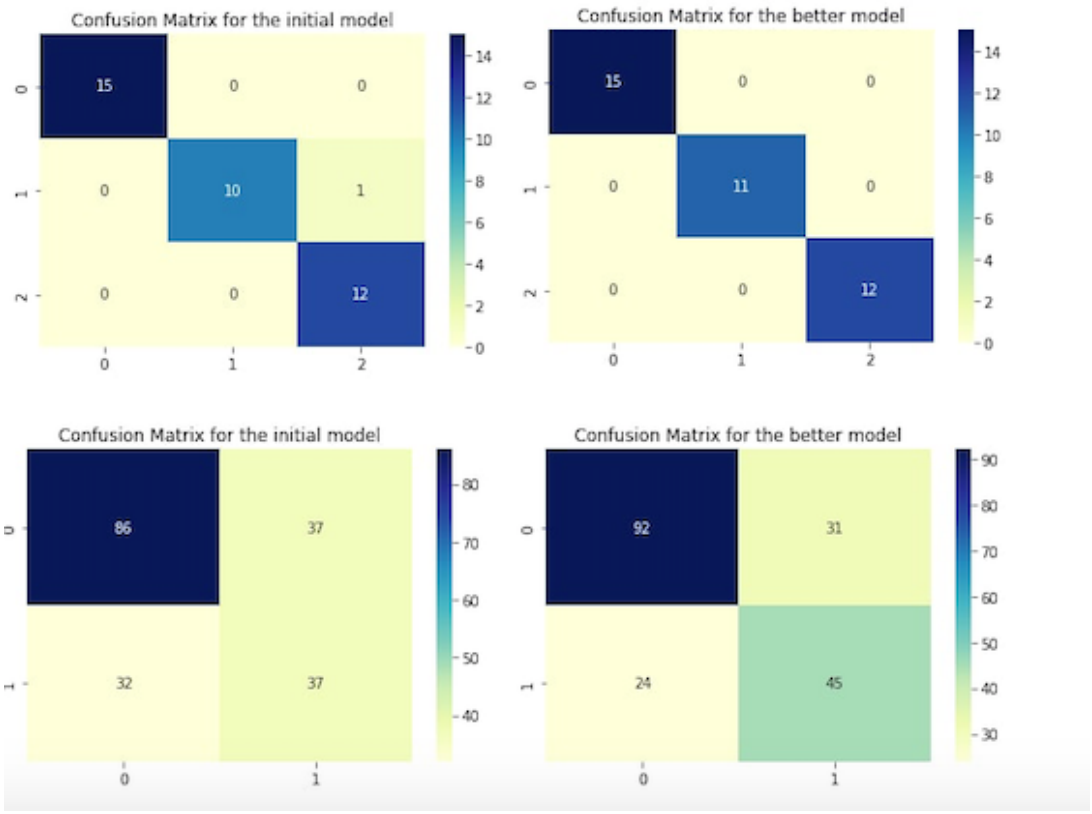

Figure 2: Random Forest Classifier before and after adjusting his data. Data sets: Diabetes and Irises respectably. On the first matrix we see that our program eliminate all false results.

6 Conclusions

After weeks of working on this project and reading articles regarding our domain, we were amazed by the value that each manipulation of the data brought. As data science researchers we spend lots of time and energy trying to improve our model accuracy. We do so by searching for the right classification model and tweaking the hyperparameters. And Most of the time it only resolves by increasing the accuracy only by a few. When we looked at the graphs that we plotted in the experimental evaluation section the penny dropped, we saw a huge increase in our model (Random Forest classification) accuracy, only by using feature engineering techniques. We saw an accuracy of 99 percent on the iris data set, also 5, 11, 20 percentages increase on the iris, breast cancer, wine data sets respectively. After we saw this kind of result, we could not put aside anymore the importance of feature engineering in machine learning research. The fact that a classification model can predict eleven percentages better using our project is astonishing, the difference between 84 percentages to 95 percentages accuracy prediction on cancer treatments, should not be taken light-headed.

We encourage you to use our project to advance your work and get better results.

References

[Hea20] Jeff Heaton. An empirical analysis of feature engineering for predictive modeling. *Washington University in St. Louis*, 2, 2020.

[Sl21] Scikit-learn. Feature importances with a forest of trees. *Scikit*, 1, 2021.

[UK17] Deepak Turaga Udayan Khurana, Horst Samulowitz. Feature engineering for predictive modeling using reinforcement learning. *IBM*, 1, 2017.