Miniproject Machine Learning

Ian Enderli and Bettina Weber

Introduction

The goal of this project was to accurately predict if it would rain the next day in Pully based on a data set with various measurements across Switzerland on the preceding day. The variable we wanted to predict is precipitation_nextday which is a boolean that tells us whether or not it rained the day after. To compare our models we used 10-fold cross validation and averaged the AUC of each fold. This is referred to as AUC in the rest of the report. In general when tuning we started with a large range of values and honing in on the best values one step at a time.

Visualization and Cleaning Steps

When looking at the data we did not observe any severe outliers. Furthermore, we noticed that there are slightly less data points that correspond to no rain on the next day. To begin cleaning the data we compared the input variables of the test and training sets and discovered that they are identical so we didn't remove any columns during this step.
In addition to this we used both PCA and tSNE to see if there are visible clusters relating to whether it rained the next day in Pully in the data. Unfortunately both methods did not display any clear clusters that could be exploited.
As for missing data, the easiest and safest way to deal with it is to simply drop all of the lines that are missing any entries. However, when doing this one reduces the original amount of data that we can train on by about half which is a massive reduction. We instead used FillImputer which fills the missing values with the median value which is less influenced by outliers than the mean value.
We then calculated the variance of each column and directly removed all columns with a standard deviation of zero as they will not influence our prediction. Additionally, we decided to remove all columns with a variance of less than 0.5 as when testing our predictions with and without these columns the result was the same. As all of the concerned columns were of the sunshine type, we checked that other data sets of the same type had a higher variance to make sure the issue was not with the units. The removed columns are: ALT_sunshine_4, ABO_sunshine_4, SAM_sunshine_4, ZER_sunshine_1 and CHU_sunshine_4.
Since we used methods that are sensitive to standardization we decided to standardize our data set. This worked well as we had removed the columns with the lowest variance which can cause problems when computing the standardization. The methods all use this standardized and cleaned data as it produced either the same or better results when testing.

Linear Methods

Logistic Classification

After completing the data visualization and cleaning processes we decided to start with the simplest fitting model: Logistic Classification. This would allow us to establish a baseline for our evaluation of the tuning of the following functions. Changing the values of lambda between 0 and 1 had no impact, higher values between 5 and 150 worked much better. We tested both L1 and L2 regularization to see which one worked better with our data. For the Logistic Classifier we received an AUC of 0.92595 with L2 regularization. Our highest AUC of **0.92934** was achieved with the Logistic Classifier when using L1 regularization.

Non-linear Methods

K Nearest Neighbors:

The second method we chose to explore was KNN as it is a simple method that can produce surprisingly good results at the cost of taking a long time for predictions. For this method we had the model tune itself by using TunedModel(). As KNN is very sensitive to standardization we made sure to use the standardized data for this method. In the end our best observed model produced an AUC of **0.91119**. This score is lower than our other models which is not surprising, however, as it is such a simple model is it still impressive that it can achieve a rather high accuracy.

Random Forest Classifier

We decided to start a Random Forest classifier as this method produces good results without too much tuning. This method should also give us a good baseline for the tuning of our Neural Network as we should expect to achieve a better result with that method. The random seed was fixed in order for thee result to be reproducible. After testing different values for the number of trees, maximum depth and number of subfeatures we settled on 322 trees, 300 maximum depth and 60 subfeatures as this produced the best results. Our highest AUC for the Random Forest Classifier was **0.92406**.

Neural Network

We chose to implement a Neural Network as it has the ability to produce highly accurate predictions. We started with a smaller network that takes less time to run in order to get a feeling for the influence of each of the hyperparameters. For the hyperparameter tuning we used RandomSearch() adapting the search parameters to smaller ranges as the tuning got better. We then expanded to a larger network as with reasonable regularization this should produce better results as it is much less likely to get stuck in a local minimum. After some experimentation regarding the build of the network we noticed that enlarging the network further no longer had a positive effect on our predictions despite adaptions of the regularization. We decided to use a smaller network as the larger ones are much more computationally expensive and, in our case, did not produce significantly better results. We also explored different activation functions, but eventually settled for relu as it produced the best results. All in all, our best model used a combination of L1 and L2 regularization as well as a dropout layer achieving an AUC of **0.92896**. For the dropout layer we used AlphaDropout() which makes sure mean and variance of the activations remain the same as this tended to work better when testing with the standardized data.

Conclusion

In conclusion, we arrived at the highest AUC of 0.92934 with our Logistic Classifier based solely on our training set. The second highest was the Neural Network an AUC of 0.92896. We therefore chose those two submissions to be evaluated in the Kaggle competition. In the future it would be interesting to spend more time on visualization in order to find interesting features for the Logistic Classifier as it has the potential to vastly improve the method.