Mystery Data Set

Hieu Nguyen, Nithya Devadoss, Ramesh Simhambhatla, Ramya Mandava Dec 2, 2018

**Abstract**

In this case study, we explore a mystery data set that has about 50 features and 160,000 observations to predict a binary classifier. The business owner provided an open ended data set without much explanation or details about the features. We perform exploratory data analysis and mold the tidy data based on common sense approach. Though the business owner did not provide much details about the data, did provide the impact of wrongful predictions – a False Positive has -$500 and a False Negative has a -$10,000 financial impact. Based on this expectation, ‘Recall’ shall be the model metric to select the best model when there is a high cost associated with False Negative. Recall actually calculates how many of the Actual Positives our model capture through labeling it as Positive (more True Positives, and less False Negatives are better). The approach to machine learning technique in this case study is of 2 stages to determine the best model and parameters using ensemble learning and then stacking. This helps improve machine learning results by combining several models; the predictions of the first models becomes the feature on the second level to train the second level model. This approach allows to produce a better predictive performance compared to a single model. We conclude reporting our results based on our experiment.

1 Introduction

The goal of our project is to build a binary classification model which can predict the class label of the variable y given by our business user into classes of 0 or 1. Our dataset contains 160000 observations with 50 features, most of which are numeric data and few categorical features. We have cleaned our dataset before running through the model which is explained in detail. Our workflow is as follows:

* Import dataset into dataframe
* Exploratory Data Analysis
* Create Tidy Set required for modelling
* Model Architecture - Stacking
* Model building / Hyperparameter optimization
* Evaluate Model performance and Report results.

2 Methods

**2.1 Data Set:**

Our dataset has 160K observations with 50 attributes. It contains some numeric fields and categorical variables like region, month, day of the week, percentage information along with the output variable ‘y’. We have performed data cleanup as follows:

1. After observing the data sample, we have renamed the attributes 'x24', 'x29','x30','x32','x37' to ‘region', ‘month', ‘day', 'rate' and 'PL' (Profit/Loss) respectively.
2. Replaced all nulls to NaN
3. Replaced special characters like ‘%’ and ‘$’ in ‘rate’ and ‘PL’ fields respectively and converted them to float.

After summarizing the data, we found that there are approximately 30 data points per feature that are NaN’s. These missing data points with NaN’s has no relation with other data points and are missing at random. Moreover, the distribution of the predictor variable remains the same with and without the data points containing NaN’s. Hence we have decided to drop the data points and have verified the distribution as below:

|  |  |
| --- | --- |
|  |  |

***Table 1: Histogram of output variable***

**2.2. Exploratory Data Analysis:**

We tried to analyze the distribution of the variables and identify and outliers using box plots before and after normalization which is shown in the Table 2 below.

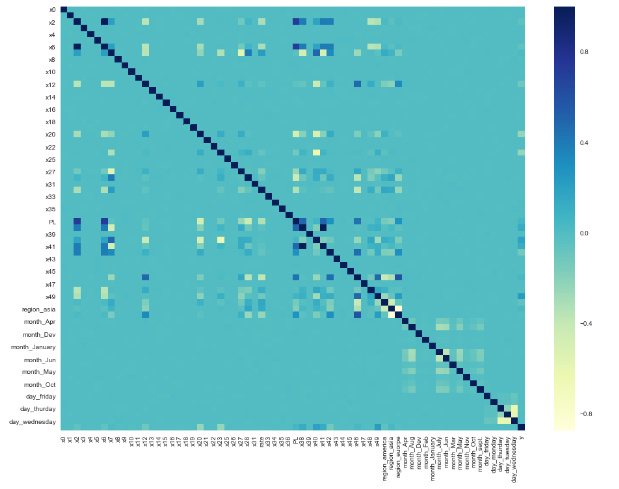
Normalization allowed us to scale the data to one standard and compare the effect of variables on the model without biasing.

|  |  |
| --- | --- |
| Distribution of raw data | Distribution after normalization |
|  |  |

***Table 2: Box plots of the data set***

Coming back to our object datatype variables which are region, day and month, we have performed one hot encoding to find the effect on the model. Now the normalized data set contains 158392 rows and 68 columns

Next we performed a correlation plot to find any associated variables. Based on the plot below in Figure1 we find that variables 'x2' has correlation with ‘x6’ and ‘PL’ whereas 'x41' is highly correlated with ‘x39’. Hence we have dropped both ‘x2’ and ‘x49’.



***Figure 1: Correlation plot***

**2.3. Creating Tidy Data:**

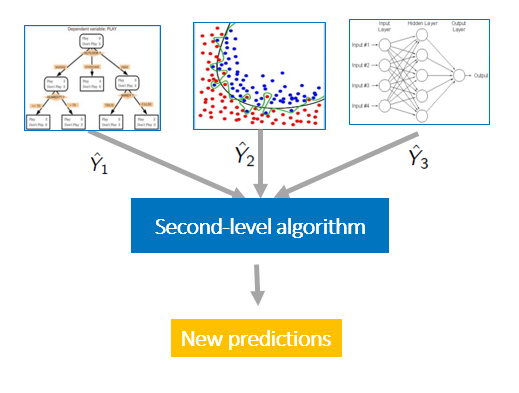
After dropping the NaN’s with count of 1608 observations, the total number of good data in our data frame is around 158392. So the clean data set which is ready to be fed to the model has 158392 observations with 66 attributes including the output variable ‘y’. All the attributes are float data types except for the imputed columns which are uint data type. The output variable is of datatype int64.

**2.4. Model Architecture**

Predictive accuracy of a model can be boosted by combining predictions of multiple machine learning models, generally referred to as an Ensemble method. Traditionally, ‘weak’ learners are combined and ‘Voting classifiers’ are used to predict the output. However, creating an ensemble of well-chosen group of strong and diverse models have proven to be more powerful in the recent times.

The simplest form of ensemble is achieved by averaging the predictions of the models, but a weighted average method yields better results in most cases according to literature. And an even better approach is to estimate the weights more intelligently by using another layer of learning algorithm. This method is referred to as ‘model stacking’.

Model stacking is achieved by feeding the outputs of multiple Machine Learning algorithms to a second layer of learning algorithm as shown in the below Figure2 [2]:



***Figure2: Model Stacking with two layers***

In our case study, Model stacking has been implemented where Random Forest and XGBoost have been employed in the first layer with optimized parameters for improving ‘Recall’ through a Random Search algorithm. These results are further fed to XGBoost in second layer to get to an accuracy of 0.85.

Even though, 5 Machine Learning algorithms(Random Forest, XGBoost, GaussianNB, Logistic Regression) have been tested for layer 1, only RF and XGBoost have been selected for layer 1 implementation as their accuracy is above 90%.

As can be seen from the summary statistics (figure 2.3.2), the range(min:max) of ages has narrowed down from 1999 to 2012 by about 25% (range of 69 years in 1999 to 53years in 2012). We can also notice that the median age of female runners in 1999 is 33 years and it gradually decreased to 30 years by 2007 and it stabilized at that value.

3 Conclusion

In this case study, we acquired women runners' data from Cherry Blossom website, scrubbed the data manually to create a data structure for Age of the runners. We have further analyzed the Women Runners’ Age data using various statistical techniques such as: QQ plots, histogram, Summary statistics, Density curves and Box plots.

From our analysis of age distribution across all years from 1999 to 2012, the general population female race runners display right-skewed distribution with the highest frequency of runners falling into the 25-30 years age bin for Women.

It can be noticed from the quantile-quantile plots that in the earlier years (around 1999) there are several older runners(outliers) with ages above 70 in the top 20% quantile. Stacked density plot with a single curve for each year, further shows a decrease in mean age of female runners from 1999 to 2012. This is further visually supported by the boxplots that shows the median line that decreases from ~35 to ~32 across years. However, the drop in this age has stopped in 2007 and remained stable for the rest of the years.

4 References

[1] Prof Slater’s Jupyter Notebook sample and class material

[2] <https://blogs.sas.com/content/subconsciousmusings/2017/05/18/stacked-ensemble-models-win-data-science-competitions/>

**APPENDIX A**

**CODE:**