ALY6020- Second Quarter, Term B, Dr. Tammy Wang

Module 6, Week Six

Collaborative Group Project

Manvinder Kaur

Assignment Completion Date: 03/30/19

Silicon Valley, Northeastern University, CA.

Abstract

This report cites the result of Five supervised learning models on a single data-set and comparing each of the model for their accuracy and prediction. Three of the implemented models are classification models- KNN, decision tree and random forest. Other two are the regression models – Logistic regression and linear regression. The problem addressed from the dataset is “Forecasting Rainfall using Historical Data”. Introducing Dataset: This dataset contains daily weather observations from numerous Australian weather stations. It covers data of 49 countries with few categorical and rest continuous numerical fields. Total records are 142,194 and number of features are 23. This dataset contains the information for 10 years from year 2007 to 2017. The target variable is Rain Tomorrow meaning: Did it rain the next day? Yes or No. As the data-set is of classification type it is best to use the classification algorithms to make the predictions, but the logistic regression is also a very efficient in this case as the response variable is binary. The implementation of Linear regression in this project has been performed to see the correlation between various features and validating the results with other models. The major challenge with the dataset was the cleaning process. There was more than 70% data which had N/A in one of the fields so removing these fields wouldn’t have been the right approach also replacing with zero would also have impacted the model a lot, so I have used the replacement with mean approach to handle the null values. The models are built and run in python and the results and graphs are plotted using various in-build library function in python. The most important functions/ logics used in the algorithms are “KNeighborsClassifier”, “DecisionTreeClassifier”, “RandomForestClassifier”, “LinearRegression” and “LogisticRegression” which are covered in the report. To conclude, the type of dataset is very important in determining the type of model to be used. The important parameters to be keep in mind before finalizing the model is the research question that we are addressing. If we are interested in prediction or inferences is the very first question. Few models work well for predictions like KNN, Decision tree and random forest and the accuracy of these models is high where if we are interested in making inferences than complex models like XgBoost. Other important factor is the size of the data set. If we keep these things in mind, we will easy know which model would be best suited for the dataset. And then we can fine tune the model using various functions.

Keywords: KNN – K-Nearest neighbor

The data-set used is taken from Kaggle and has Rainfall record of 49 Australian countries with 23 features impacting the prediction of rainfall. Firstly, the data is extracted, and few steps are taken to clean the data.

1. **Data set and extraction**

   - Dataset is taken from https://www.kaggle.com/jsphyg/weather-dataset-rattle-package/version/2#weatherAUS.csv
   - Data extraction is by downloading and reading the excel file.

2. **Data cleanup and organizing existing data**

   - Column Dropped - 'Date', 'Location','RainToday', 'RISK_MM' These columns are not used.
   - All the null values 'NaN' is replaced by mean of the column as this will not have wrong effect on our model for calculations.

3. **Training the model**

   - The model has been trained on 80% of data and tested on 20% of data.

4. **For Data visualization**

   - The critical dimensions(columns) of the data are Rainfall, Humidity, Evaporation, Pressure, Sunshine
   - The correlation of rainfall with Evaporation, Humidity and WindSpeed has been plotted.

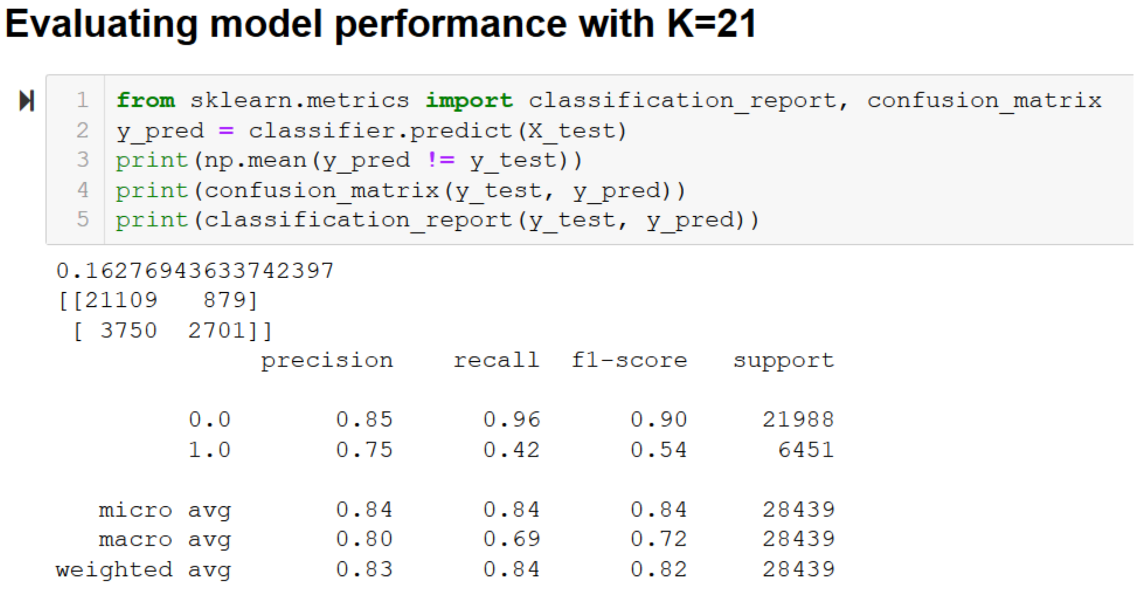
5. **Predictive statistics**

   - Target - It will Rain Tomorrow? Yes or No

Models and their results:

K Nearest Neighbors (KNN)

The K Nearest Neighbors classification model has been ran for the value of K=21, I ran the model for different values of k and this the best result I have got so far. I couldn’t find out the optimum value of K by running through all the records set as the number of records in my data-set is huge. And is out of the capacity of my hardware. Let’s look closely into the results received, the Accuracy of the model is 84 % which is not bad. But we have to look into the confusion matrix as well for better interpretations. We can see the precision and recall for the prediction of ‘0’ is 0.85 and 0.96 which is good but the recall for ‘1’ is 0.42 which is not that good as for predicting the rainfall correctly atleast we expect the model to have recall greater than 0.50. Higher the recall better the accuracy of the model is considered.



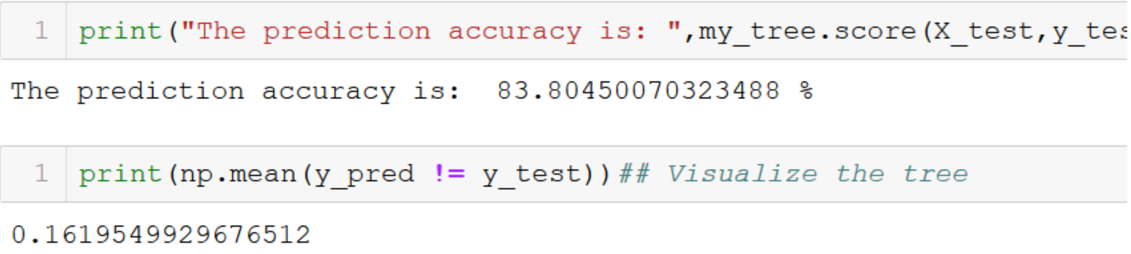
Evaluating model performance with K=21

```
from sklearn.metrics import classification_report, confusion_matrix
y_pred = classifier.predict(X_test)
print(np.mean(y_pred != y_test))
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
0.16276943633742397
[[21109   879]
 [ 3750  2701]]
              precision    recall  f1-score   support

         0.0       0.85      0.96      0.90     21988
         1.0       0.75      0.42      0.54      6451

   micro avg       0.84      0.84      0.84     28439
   macro avg       0.80      0.69      0.72     28439
weighted avg       0.83      0.84      0.82     28439
```

Decision Tree Classifier

For Decision tree classifier, the accuracy of the model came out to be 83.8% which is less than what we achieved in KNN model. Let’s dive deeper in this and understand the confusion matrix of this one. The precision and recall for ‘1’ has slightly improved which will make this model look better. The tree has been created with the node as 5. And the below tree is formed. Also, this model helps us in identifying the importance features of the predicting model. The critical dimensions(columns) of the data are Rainfall, Humidity, Evaporation, Pressure, Sunshine. We can also run the model for better performance by removing the features which are least significant. For now, will are proceeding with all the features for the comparison purpose.

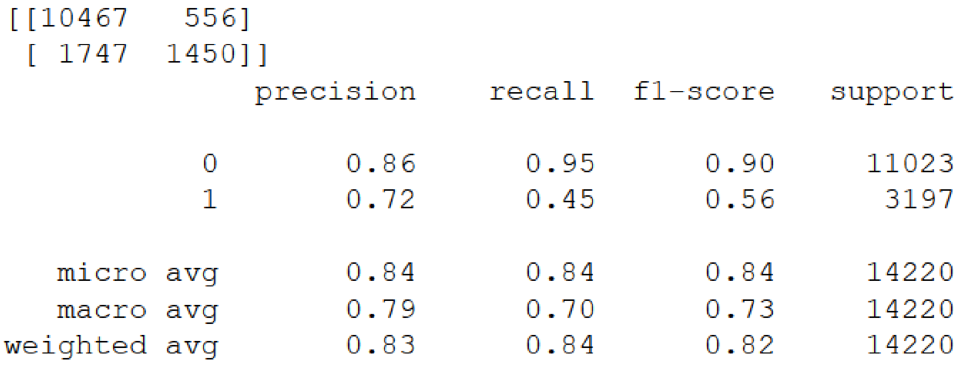


```
print("The prediction accuracy is: ",my_tree.score(X_test,y_tes
```

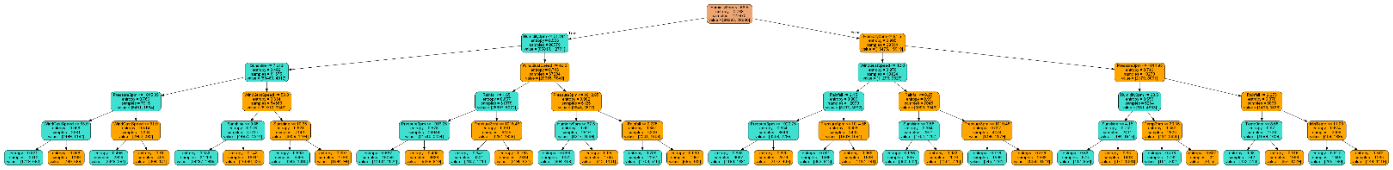
```
The prediction accuracy is:  83.80450070323488 %
```

```
print(np.mean(y_pred != y_test))## Visualize the tree
```

```
0.1619549929676512
```

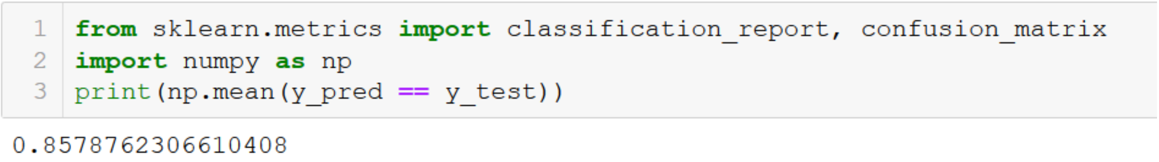


```
[[10467   556]
 [ 1747  1450]]
              precision    recall  f1-score   support

           0       0.86      0.95      0.90     11023
           1       0.72      0.45      0.56      3197

   micro avg       0.84      0.84      0.84     14220
   macro avg       0.79      0.70      0.73     14220
weighted avg       0.83      0.84      0.82     14220
```



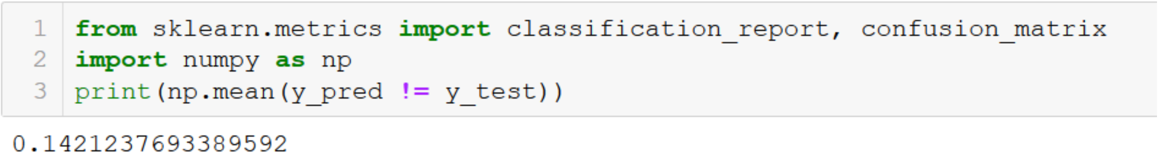
Random Forest Classifier

Now moving to the next model which is Random Forest. The results are impressive as compared to the other models. As the accuracy of the model has been increased which is 85.78%. This is the best model as far as the accuracy is concerned. Now we will see the confusion matrix for the same. While looking into the confusion matrix we see the precision and recall for ‘0’ is 0.87 and .96 respectively which is very good and for ‘1’ the values are 0.77 and 0.52 respectively, which is the best so far. So, we can say that this is the best model for predicting the Rainfall. Of course, we will have to do a lot tuning to the model further.



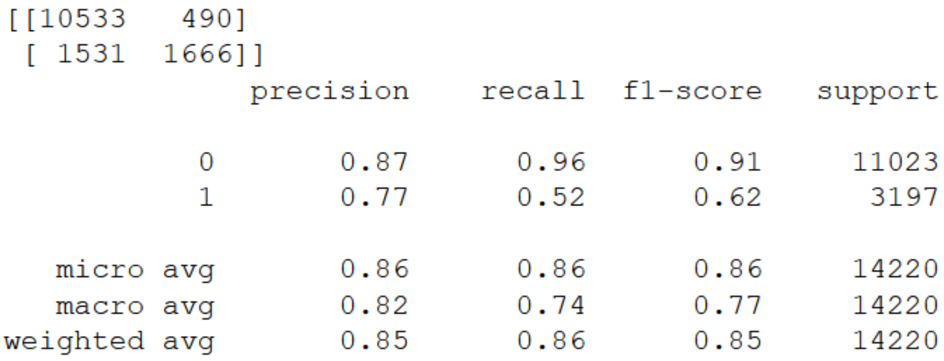
```
from sklearn.metrics import classification_report, confusion_matrix
import numpy as np
print(np.mean(y_pred == y_test))
```

```
0.8578762306610408
```



```
from sklearn.metrics import classification_report, confusion_matrix
import numpy as np
print(np.mean(y_pred != y_test))
```

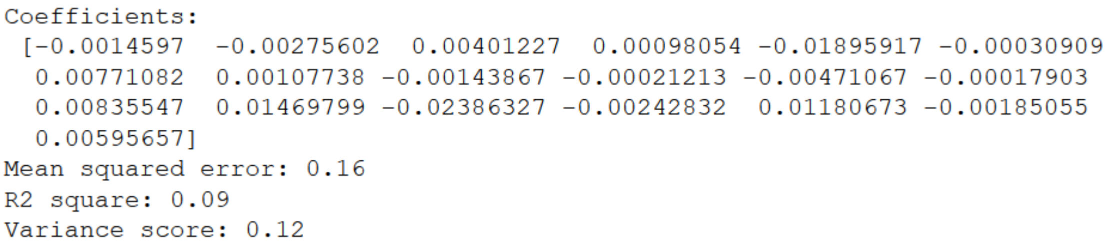
```
0.1421237693389592
```



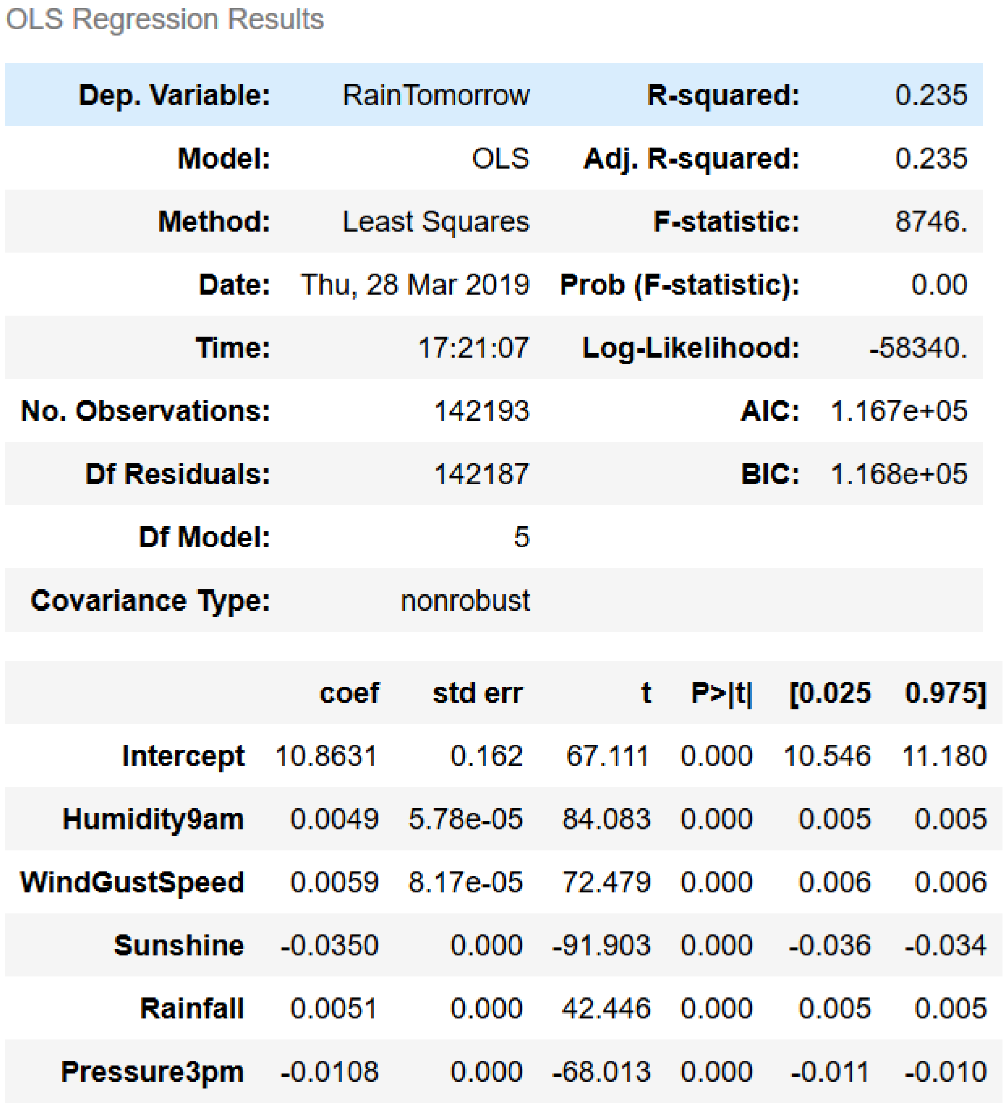
```
[[10533   490]
 [ 1531  1666]]
              precision    recall  f1-score   support

           0       0.87      0.96      0.91     11023
           1       0.77      0.52      0.62      3197

   micro avg       0.86      0.86      0.86     14220
   macro avg       0.82      0.74      0.77     14220
weighted avg       0.85      0.86      0.85     14220
```

Linear Regression

This model is used when the response variable is continuous buy here our response variable has binary classes. So, this model will not be the right choice for this dataset. But still try running the model and see the results and we can look for the coefficient factors of the features and we will know which all variables are negatively impacting the rainfall. And we can see that Sunshine and pressure has negative effect on rainfall.



```
Coefficients:
 [-0.0014597  -0.00275602  0.00401227  0.00098054 -0.01895917 -0.00030909
  0.00771082  0.00107738 -0.00143867 -0.00021213 -0.00471067 -0.00017903
  0.00835547  0.01469799 -0.02386327 -0.00242832  0.01180673 -0.00185055
  0.00595657]
Mean squared error: 0.16
R2 square: 0.09
Variance score: 0.12
```



OLS Regression Results

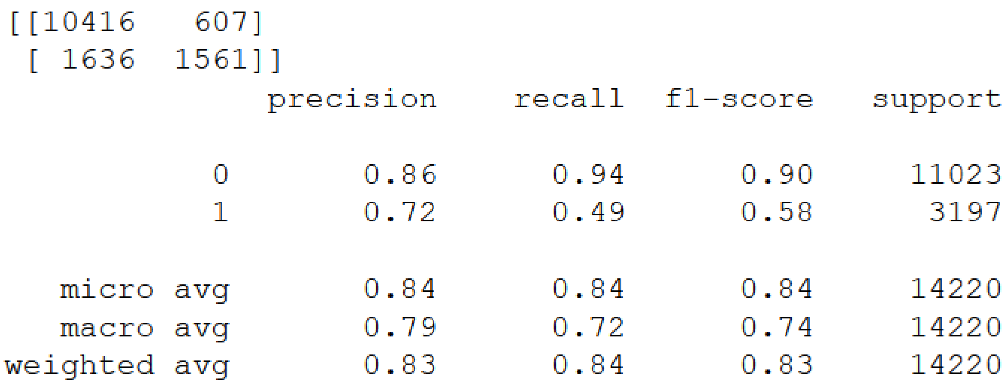
Dep. Variable:	RainTomorrow	**R-squared:**	0.235
Model:	OLS	**Adj. R-squared:**	0.235
Method:	Least Squares	**F-statistic:**	8746.
Date:	Thu, 28 Mar 2019	**Prob (F-statistic):**	0.00
Time:	17:21:07	**Log-Likelihood:**	-58340.
No. Observations:	142193	**AIC:**	1.167e+05
Df Residuals:	142187	**BIC:**	1.168e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	10.8631	0.162	67.111	0.000	10.546	11.180
Humidity9am	0.0049	5.78e-05	84.083	0.000	0.005	0.005
WindGustSpeed	0.0059	8.17e-05	72.479	0.000	0.006	0.006
Sunshine	-0.0350	0.000	-91.903	0.000	-0.036	-0.034
Rainfall	0.0051	0.000	42.446	0.000	0.005	0.005
Pressure3pm	-0.0108	0.000	-68.013	0.000	-0.011	-0.010

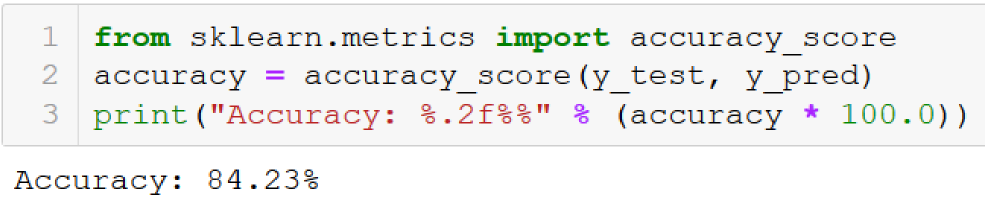
Logistic regression

This is the right regression model for this data-set as the response variable is binomial. The accuracy of the model is 84.32% which not bad and as per the confusion matrix the precision and recall for both ‘0’ and ‘1’ is fine. As the recall of one of the class is just reaching 50%. So, this model is better than few of the models discussed but not the best one. Instead of Mean Squared Error, logistic regression uses cost function called Cross-Entropy, also known as Log Loss. Cross-entropy loss has two parts: one for y=1 and one for y=0.

Important thing to note is the cost function penalizes wrong predictions more than it rewards and right predictions. The corollary is increasing prediction accuracy (closer to 0 or 1) has diminishing returns on reducing cost due to the logistic nature of our cost function. The larger the loss, the worse the estimate is according to the loss function. And for our dataset the loss is 5.45 which is quite significant and it makes this model not so good for prediction.

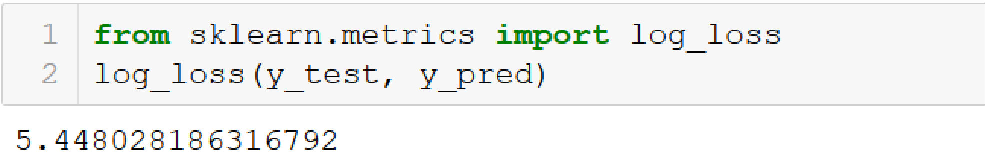


```
[[10416   607]
 [ 1636  1561]]
              precision    recall  f1-score   support

           0       0.86      0.94      0.90     11023
           1       0.72      0.49      0.58      3197

   micro avg       0.84      0.84      0.84     14220
   macro avg       0.79      0.72      0.74     14220
weighted avg       0.83      0.84      0.83     14220
```



```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

```
Accuracy: 84.23%
```



```
from sklearn.metrics import log_loss
log_loss(y_test, y_pred)
```

```
5.448028186316792
```

In conclusion, we can say that the best way to find the right model for a dataset is to understand our data-set and the problem statement that we are trying to address. Once we know this we will have to dive deeper to understand the dataset by doing some visualization. When you know the data-set you will be able to choose the right model. We have seen the results of all the model that we run and looked closely into the accuracy and confusion matrix. For the prediction of rainfall random forest classifier is the best suitable model as the data-set has response variable as categorical and random forest creates multiple random trees using different number of features and different sub-set of data. And selects the most optimal record by polling which makes it better. The accuracy of this model gave us the results as 85.75% with the best confusion matrix among all the other models that we run on the same data-set. We can further use the techniques to enhance this model by using XgBoost and we can also reduce the features which don’t have a very significant effect. These are few of the improving techniques which can further be implemented.

References

1. UCI Machine Learning, (2017, Jan). *Glass Classification Data Set.* Retrieved from https://www.kaggle.com/uciml/glass
2. Eng. Juan Gabriel Colonna, (2017, Feb). *Anuran Calls (MFCCs) Data Set,* Retrieved from https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+%28MFCCs%29#
3. J. Strickland, (2014). *Predictive Analytics Using R.* Retrieved from https://www.slideshare.net/JeffreyStricklandPhD/predictiveanalyticsusingrredc