Building a Random Forest Model and Prediction of Instances

Arvind Ramkumar – Data Analyst

Graduate Student – Industrial Engineering

Table of Contents

INTRODUCTION

There are a given training set containing 66 Variables and 2500 instances to train the data and 1600 instances that has to be predicted. Without any knowledge about the model, to validate the model, splitting the available data into Train and Test, to validate the model built from the train.

PREPROCESSING

Handling Categorical Data

There are two ways that will help in handling the Categorical Variables.

1. Label Encoding
2. Get Dummies

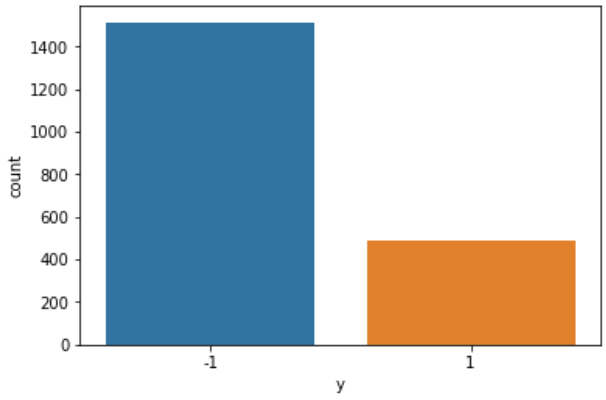
I preferred Label Encoding over Get Dummies because, get dummies will assign an extra column for each of the category, and there his high chances of forming a sparse matrix. Also, the number of values in each category in the train set doesn't equal to the number of values in each category of the test set. So, it is a manual work of adding a null column to the categories that aren't available in the test set.

Also using get dummies, doesn't help in expanding the model beyond the scope of the test and the training set. So, preferring Label Encoder over Get Dummies will make utmost sense.

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	...	x58	x59	x60	x61	x62	x63	x64	x65	x66	y
0	27	1	1	1	0	18	3	1	28	119.9	...	1	0	0	0	0	1	3	0	1	-1
1	30	0	1	1	1	18	13	3	19	86.7	...	1	0	0	0	0	0	2	1	1	-1
2	37	0	1	1	0	1	3	14	33	174.0	...	1	0	0	0	0	0	3	0	0	1
3	29	0	1	1	2	14	9	3	29	8.8	...	0	0	0	0	0	0	3	1	0	-1
4	33	1	1	0	4	2	15	12	39	55.0	...	1	0	0	0	0	0	2	1	1	-1

Class Imbalance

There is a heavy class imbalance in the model. Out of the 2500 data points with the binary output, there are 1891 instances with output as -1, and 691 instances with output as +1.



If the class imbalance is not taken care off, then there is a high possibility that almost every of the instances that are tested will give us the y value equal to -1.

There are a couple of ways to handle the class imbalance.

1. Up Sampling
2. Down Sampling

In general, Down Sampling is preferred, because using Up Sampling will cause repetitions. But in our given train data set, we see that there are only 2500 instances and if we are down sampling, we will lose almost 50% of the data. This might cause a huge problem. So, we up sample the data to avoid missing any of the vital instances.

Although there could be redundant instances, it is much better to have a model with more than half of its instances lost.

Handling Outliers

There are few outliers that are present in the model. Although their effect might be very minimal, to handle those effects, Scaling will come in hand to hand. Here I have used the MinMax Scaler. MinMax Scaler rescales all the features to lie within [0, 1]. So, the effect of the outliers will also be much minimal relatively.

MODELLING

The model that I choose to build was Random Forest. There are a few reasons for choosing the Random Forest Model, which is listed below.

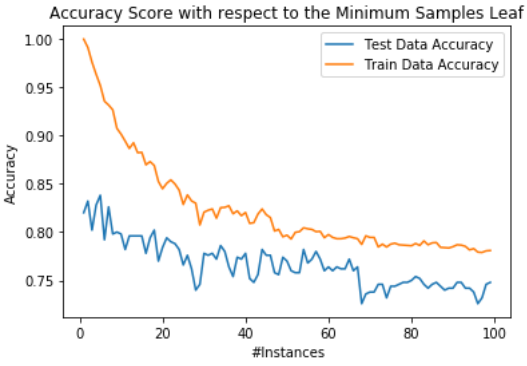
- They can handle the data of mixed type
- They can handle missing values (Although there aren't any missing values, there are two columns that have all the values to be null, and RF handles this pretty well)
- They are robust to outliers (Although Scaling has been done to overcome the effect of outlier, still RF performs better on this)
- They are insensitive to Monotone transformations of input.

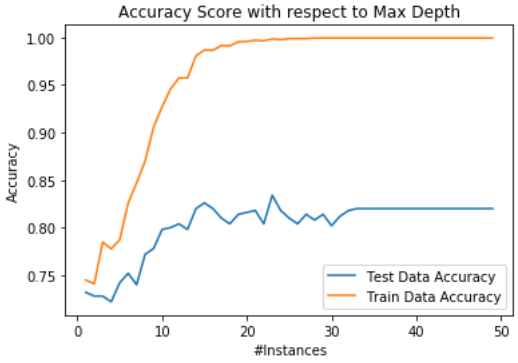
- Ability to deal with irrelevant inputs (Here, there are few categorical values found in train is not found in test set, RF handles that pretty well)

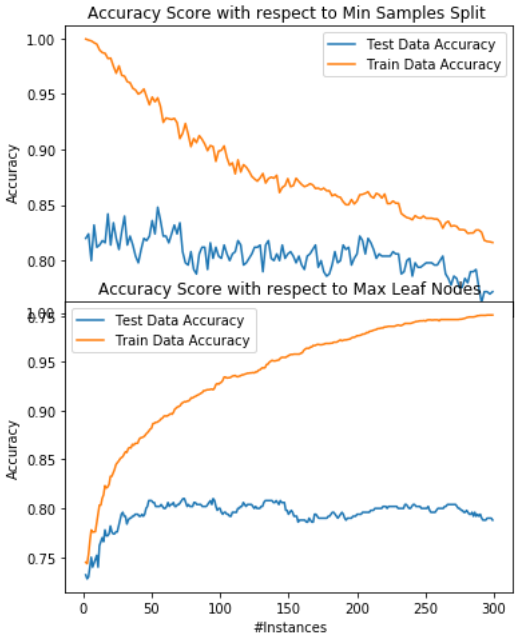
In developing a RF model there are various parameters that are to be studied and fine-tuned for obtaining a better solution. The parameters that are taken for study are

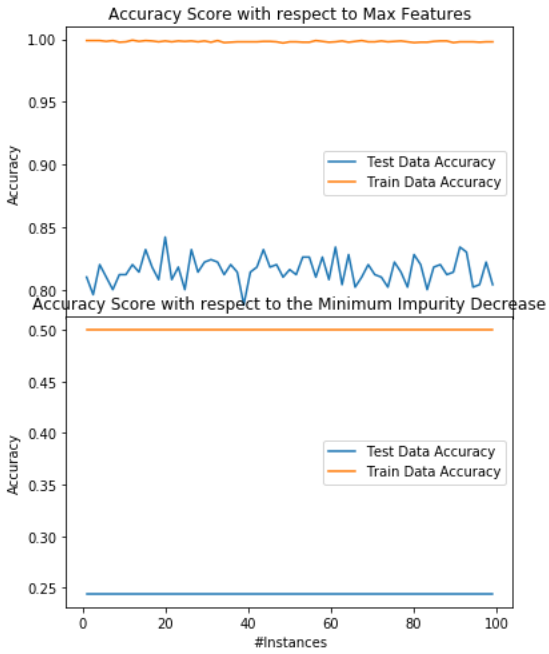
1. Min Samples Split
2. Min Samples Leaf
3. Max Depth
4. Max Leaf Nodes
5. N_estimators
6. Max Features
7. Min Impurity Decrease
8. N_jobs

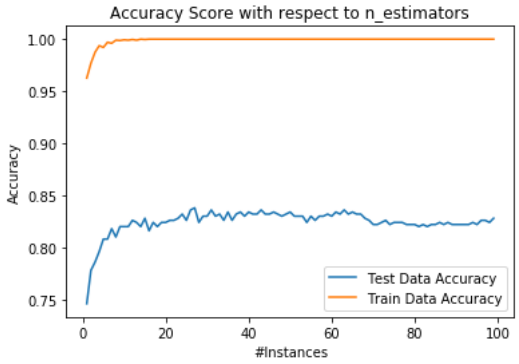
Choosing a range of values for the grid search is quite insensible. So, developing a graph to look at the change of the Test and Train Accuracy as we change these parameters will be helpful. The graph is given below.











The range to be selected for the parameters is given below.

Parameters	Range
Min Samples Split	2 – 25
Min Samples Leaf	1 – 20
Max Depth	5 – 25
Max Features	10 – 64
Max Leaf Nodes	50 – 150
Min Impurity Decrease	No effect on the accuracy
N_estimators	5 – 40
N_Jobs	No effect on the accuracy

The initial grid search was performed and based on the result, further tuning is done, and the result is summarized below

	Base Gini	Grid Search Gini	Base Entropy	Grid Search Entropy 1	Grid Search Entropy 2
Train Accuracy	99.93	94.28	96.001	99.63	99.27
Test Accuracy	82	82.19	82.8	83.8	85
Balanced Error	30.5	21.76	21.92	24.59	20.97

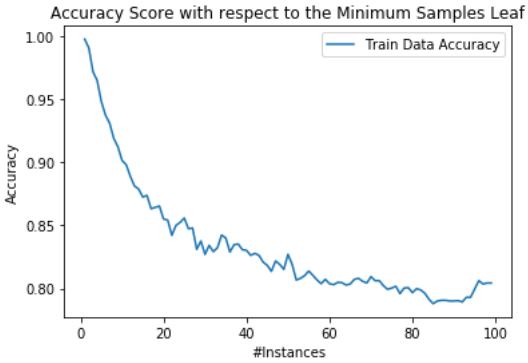
The Parameters for the obtained final model are as follows

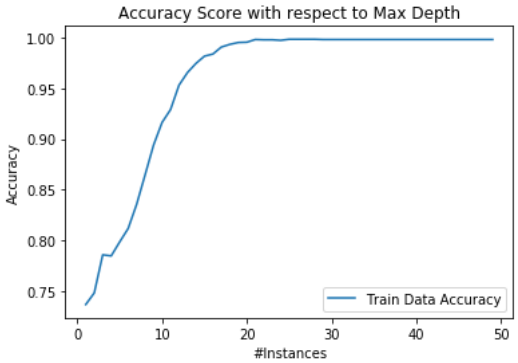
- Max Depth – 23
- Max Features – 25

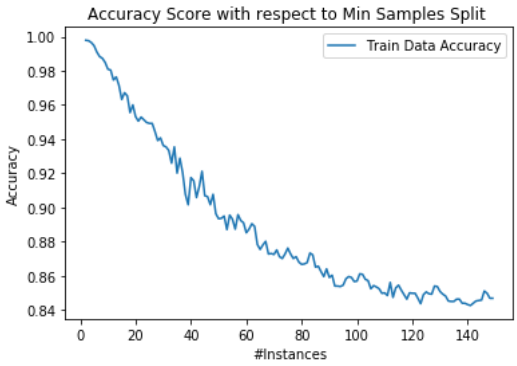
- Max Leaf Nodes – 145
- N_estimators – 37

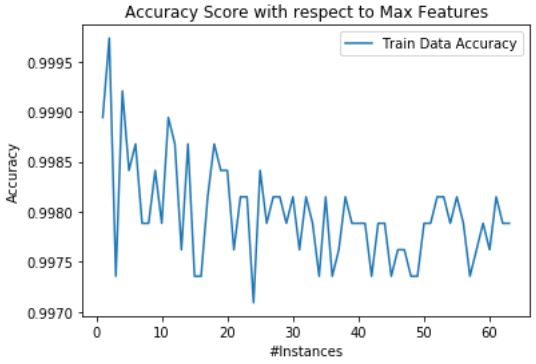
POST PROCESSING

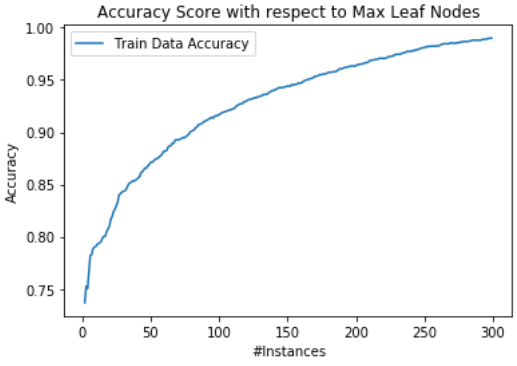
Now, the obtained model is extrapolated to the full train data to predict the instances given. Same steps are followed as mentioned above. The Range selection for the feature is carried out similarly and it is explained below.

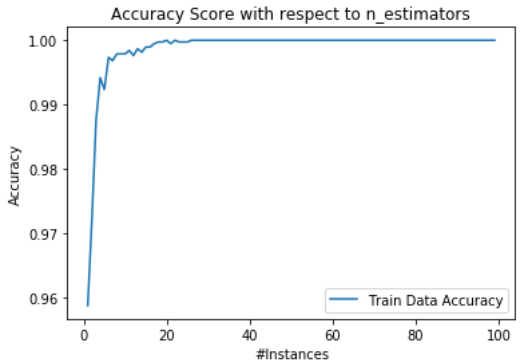












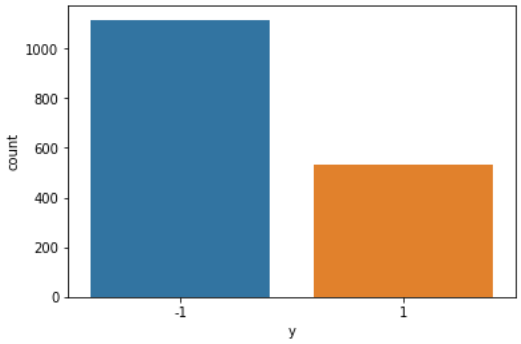
Parameters	Range
Min Samples Split	2 – 40
Min Samples Leaf	1 – 15
Max Depth	5 – 15
Max Features	3 - 10
Max Leaf Nodes	50 – 150
N_estimators	5 – 20

The final obtained model to fit the full training data is a Random Forest with the parameters

- Max Features – 9
- N_Estimators – 7
- Min Samples Leaf – 2
- Min_Samples Split – 3
- Max Depth – 30
- Max Leaf Nodes – 175

REMARKS

The class distribution of the test instances is given below



Number of -1 in Train = 1891
Number of +1 in Train = 609

Proportion in Train = 3.105

Number of -1 in Test = 1116
Number of +1 in Test = 531

Proportion in Test = 2.101

```
-1    1116
 1     531
Name: y, dtype: int64
```

The Model is considered to be reasonable.