**Best Score presentation**

**random Forest predictions transformation**

*#randomforest numtrees = 10*

pred1 = predictions1.select('Id','probability').toDF(\*['Id','vec'])

submit1 = pred1.rdd.map(convert).toDF(features) *#randomforest numtrees = 10*

*#randomforest numtrees = 10*

submit1.coalesce(1).write.csv("rf\_numtree\_10", header = **True**) *#saving results locally*

*Result:*

**

*#randomforest numtrees = 20*

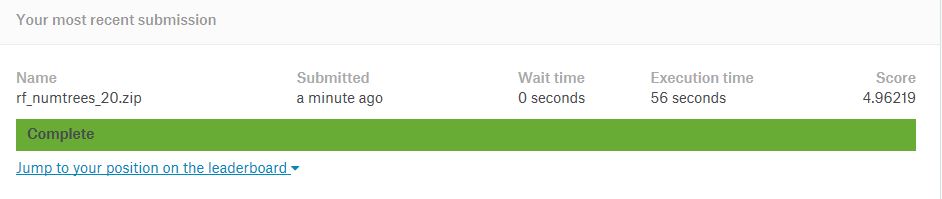
pred2 = predictions2.select('Id','probability').toDF(\*['Id','vec'])

submit2 = pred2.rdd.map(convert).toDF(features) *#randomforest numtrees = 20*

*#randomforest numtrees = 20*

submit2.coalesce(1).write.csv("rf\_numtree\_20", header = **True**) *#saving results locally*

*Result:*



*#randomforest numtrees = 30*

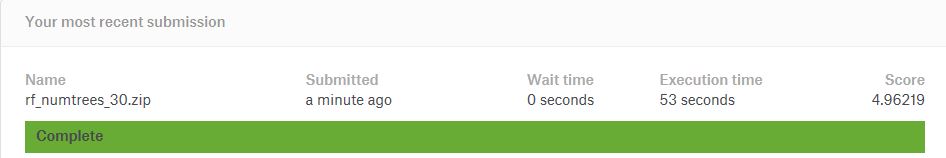
pred3 = predictions3.select('Id','probability').toDF(\*['Id','vec'])

submit3 = pred2.rdd.map(convert).toDF(features) *#randomforest numtrees = 30*

*#randomforest numtrees = 30*

submit3.coalesce(1).write.csv("rf\_numtree\_30", header = **True**) *#saving results locally*

*Result:*



**LogisticRegression predictions transformation**

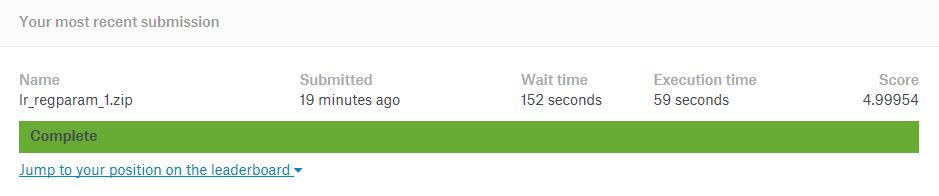
*#Logisticregression regparam = 1*

pred4 = predictions4.select('Id','probability').toDF(\*['Id','vec'])

submit4 = pred4.rdd.map(convert).toDF(features)

submit4.coalesce(1).write.csv("lr\_regparam\_1", header = **True**) *#saving results locally*

*Result:*



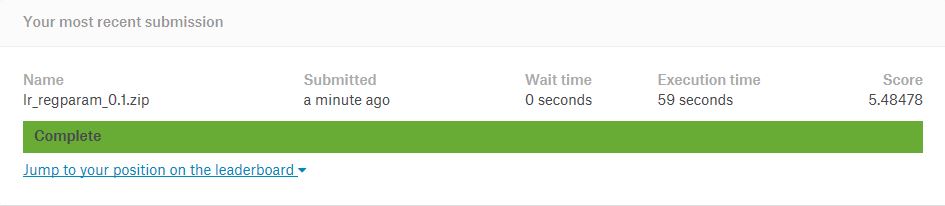
*#Logisticregression regparam = 0.1*

pred5 = predictions5.select('Id','probability').toDF(\*['Id','vec'])

submit5 = pred5.rdd.map(convert).toDF(features)

submit5.coalesce(1).write.csv("lr\_regparam\_0.1", header = **True**) *#saving results locally*

*Result:*



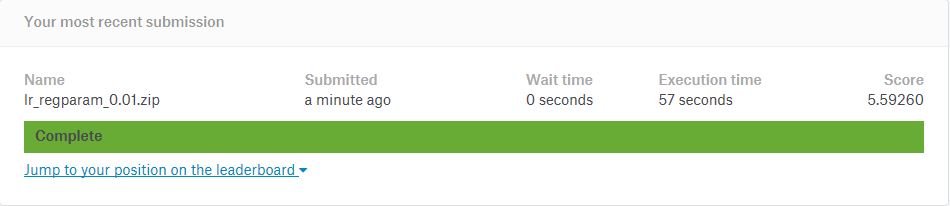
*#Logisticregression regparam = 0.01*

pred6 = predictions6.select('Id','probability').toDF(\*['Id','vec'])

submit6 = pred6.rdd.map(convert).toDF(features)

submit6.coalesce(1).write.csv("lr\_regparam\_0.01", header = **True**) *#saving results locally*

*Result:*



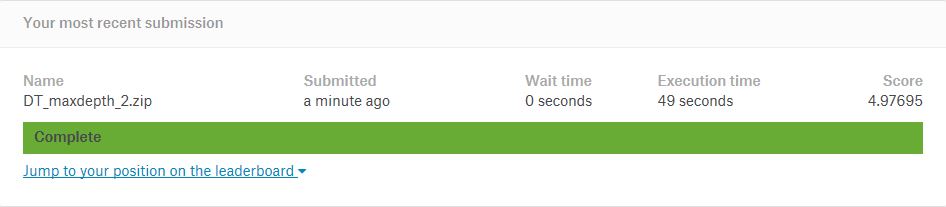
**DecisionTree Predictions Transformation**

*#decisiontree maxdepth = 2*

pred7 = predictions7.select('Id','probability').toDF(\*['Id','vec'])

submit7 = pred7.rdd.map(convert).toDF(features)

submit7.coalesce(1).write.csv("DT\_maxdepth\_2", header = **True**) *#saving results locally*



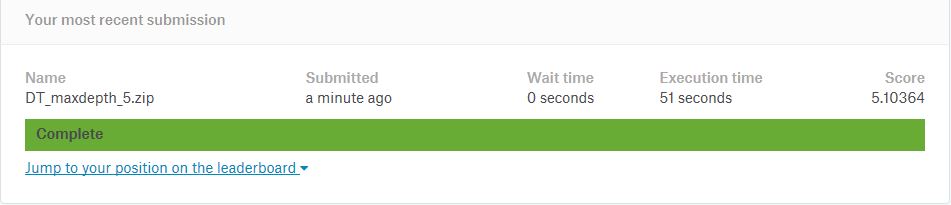
*#decisiontree maxdepth = 5*

pred8 = predictions8.select('Id','probability').toDF(\*['Id','vec'])

submit8 = pred8.rdd.map(convert).toDF(features)

submit8.coalesce(1).write.csv("DT\_maxdepth\_5", header = **True**) *#saving results locally*

*Result:*



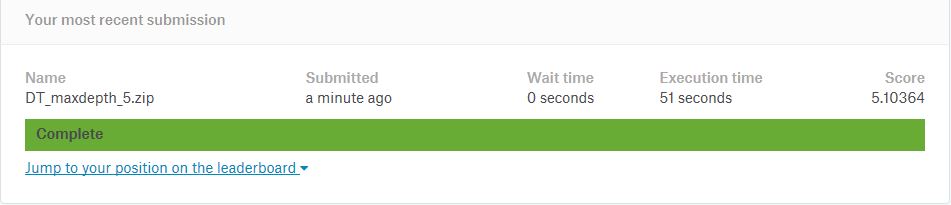
*#decisiontree maxdepth = 10*

pred9 = predictions9.select('Id','probability').toDF(\*['Id','vec'])

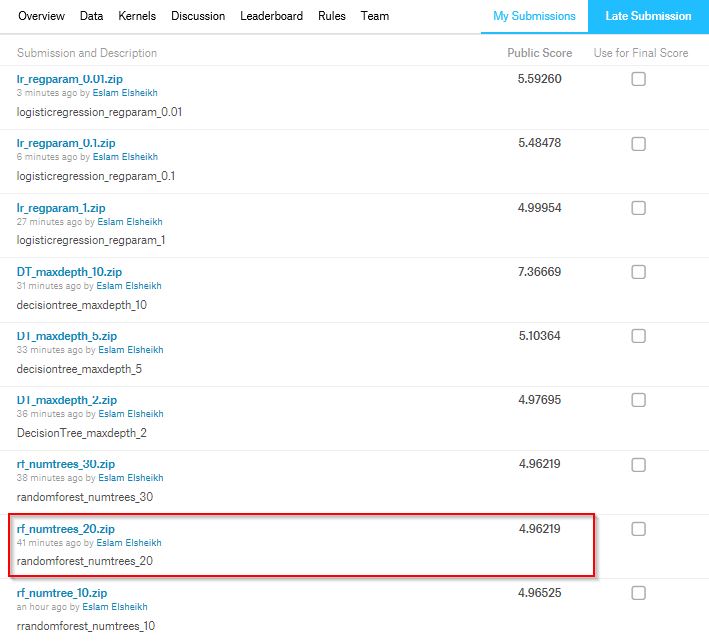
submit9 = pred9.rdd.map(convert).toDF(features)

submit9.coalesce(1).write.csv("DT\_maxdepth\_10", header = **True**) *#saving results locally*

*Result:*



From previus results we can see that best score goes to the Randomforest classifier with hyperparameter (*numtrees = 20)*

**

* ***Why Randomforest is the best?***

Since A random forest is a estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Setting numtrees to 20 with this data size gets best performance

However, the opposite happens in the Decision tree classifier as no sampling done and data is fir to only 1 tree.

Logistic regression should be used more often when you have cleanly and linearly separable classes