

Human V/s Robot

Advance Data Science & Engineering final project

Akash Bangera | 001822445 | [Akash.a@husky.neu.edu](mailto:Akash.a@husky.neu.edu)

Vividh Talesara | 001814460 | [Talesara.v@husky.ney.edu](mailto:Talesara.v@husky.ney.edu)

Harshal Girme | 001278531 | [Girme.h@husky.neu.edu](mailto:Girme.h@husky.neu.edu)

# ABSTRACT

In this project, we will be chasing down robots for an online auction site. Human bidders on the site are becoming increasingly frustrated with their inability to win auctions vs. their software-controlled counterparts. As a result, usage from the site's core customer base is plummeting. In order to rebuild customer happiness, the site owners need to eliminate computer generated bidding from their auctions. Their attempt at building a model to identify these bids using behavioral data, including bid frequency over short periods of time, has proven insufficient.

The goal of this project is to identify online auction bids that are placed by "robots", helping the site owners easily flag these users for removal from their site to prevent unfair auction activity.

There are following Best Features we gave in order to classify humans from bots.

* Used Grid Search to get best C and Gamma Value
* Used Plot in RFC to get Best Max\_depth and n\_estimator Value
* Classification Reports and Confusion Matrix (Did Plotting as well)
* Random Forest Method for result to get 0.94 Accuracy Score

# INTRODUCTION

There are two datasets in this competition. One is a **bidder dataset** that includes a list of bidder information, including their id, payment account, and address. The other is a **bid dataset** that includes 7.6 million bids on different auctions. The bids in this dataset are all made by mobile devices.

The online auction platform has a fixed increment of dollar amount for each bid, so it doesn't include an amount for each bid. You can learn the bidding behavior from the time of the bids, the auction, or the device.

## **Data SETS USED**

* train.csv - the training set from the bidder dataset
* test.csv - the test set from the bidder dataset
* sampleSubmission.csv - a sample submission file in the correct format
* bids.csv - the bid dataset

## **Data fields**

## **For the bidder dataset**

1. bidder\_id – Unique identifier of a bidder.
2. payment\_account – Payment account associated with a bidder. These are obfuscated to protect privacy.
3. address – Mailing address of a bidder. These are obfuscated to protect privacy.
4. outcome – Label of a bidder indicating whether or not it is a robot. Value 1.0 indicates a robot, where value 0.0 indicates human.

* The outcome was half hand labeled, half stats-based. There are two types of "bots" with different levels of proof:

1. Bidders who are identified as bots/fraudulent with clear proof. Their accounts were banned by the auction site.
2. Bidder who may have just started their business/clicks or their stats exceed from system wide average. There are no clear proof that they are bots.

## **For the bid dataset**

1. bid\_id - unique id for this bid
2. bidder\_id – Unique identifier of a bidder (same as the bidder\_id used in train.csv and test.csv)
3. auction – Unique identifier of an auction
4. merchandise –  The category of the auction site campaign, which means the bidder might come to this site by way of searching for "home goods" but ended up bidding for "sporting goods" - and that leads to this field being "home goods". This categorical field could be a search term, or online advertisement.
5. device – Phone model of a visitor
6. time - Time that the bid is made (transformed to protect privacy).
7. country - The country that the IP belongs to
8. ip – IP address of a bidder (obfuscated to protect privacy).
9. url - url where the bidder was referred from (obfuscated to protect privacy).

# METHODS (CODE & DOCUMENTATION)

This Includes Packages/Libraries and Algorithms we used other than Python Libraries. Also we have done plotting for our Classification Reports, Confusion Matrix, ROC curve and few Others. The Final result is predicted from Grid Search, Random Forest Classifier.

Below Presented are all Step wise Code Screenshots and Documentation of Code.

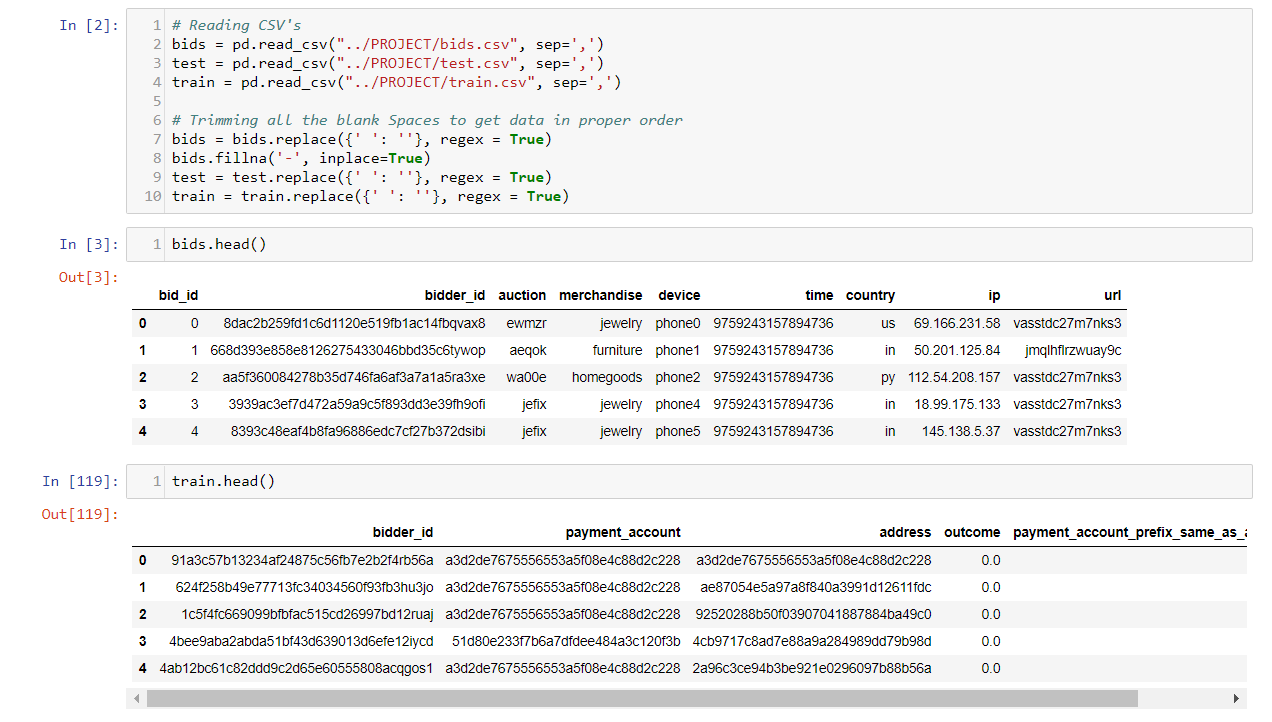
Additional packages needed

* from sklearn\_pandas import DataFrameMapper
* import pickle
* from sklearn.metrics import roc\_curve, auc
* from sklearn.preprocessing import label\_binarize

loading the data

1. Screen Shot of LOading DATA from CSV and Creating Data frame

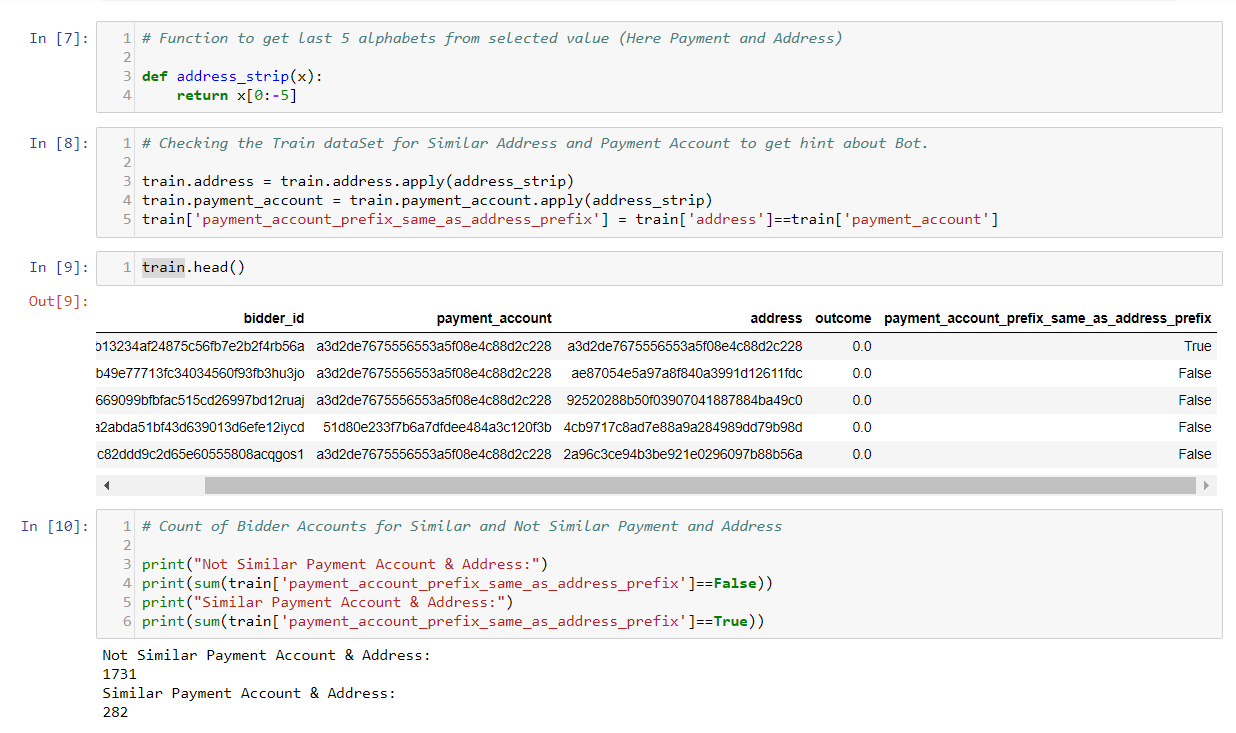
* Below Data includes Reading Csv’s and separating them with separator(‘,’) and then filling all spaces and trimming them.



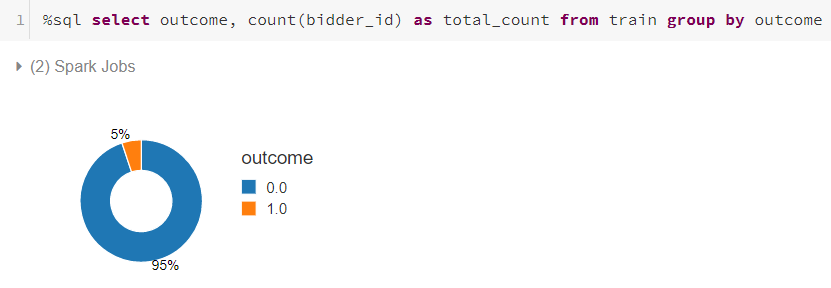
## Basic Queries to understand the data and few counts

1. Screen Shot of Stripping Train data columns and fetching last 6 LETTERS TO find if the payment and address are equal or not.

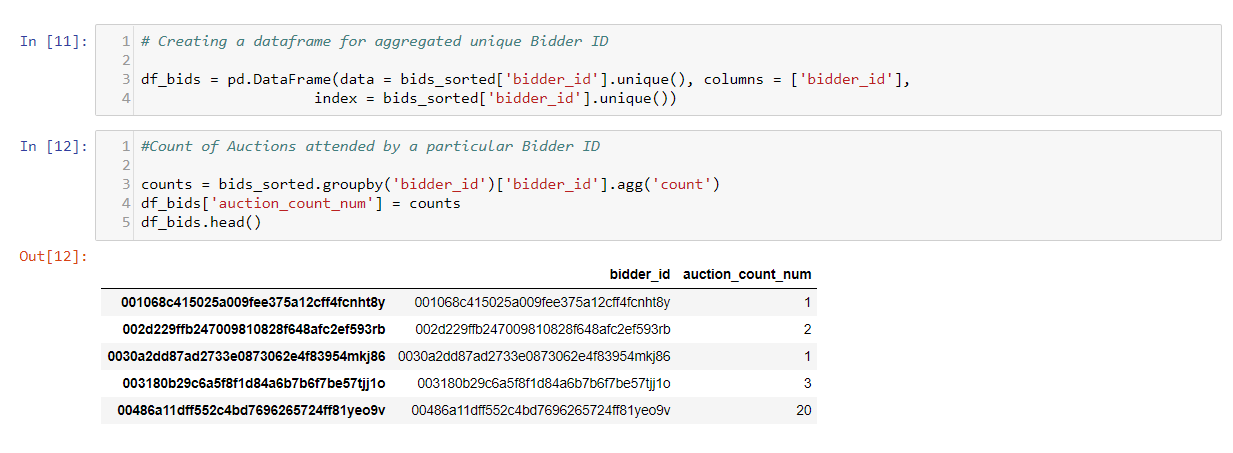
* Created a function to strip Address and Payment column and fetching last 6 alphabets
* Comparing both columns and finding if they belong to human or bot
* Found that **1731** values have different values and **282** were having same Payment and address columns values.
* Also, we did the aggregate unique count of all bidders



* Found the count of bidder ID with number of auctions they attended



* Above graph gives the percentage of Humans(95%) and Bots(5%) in our Data Set



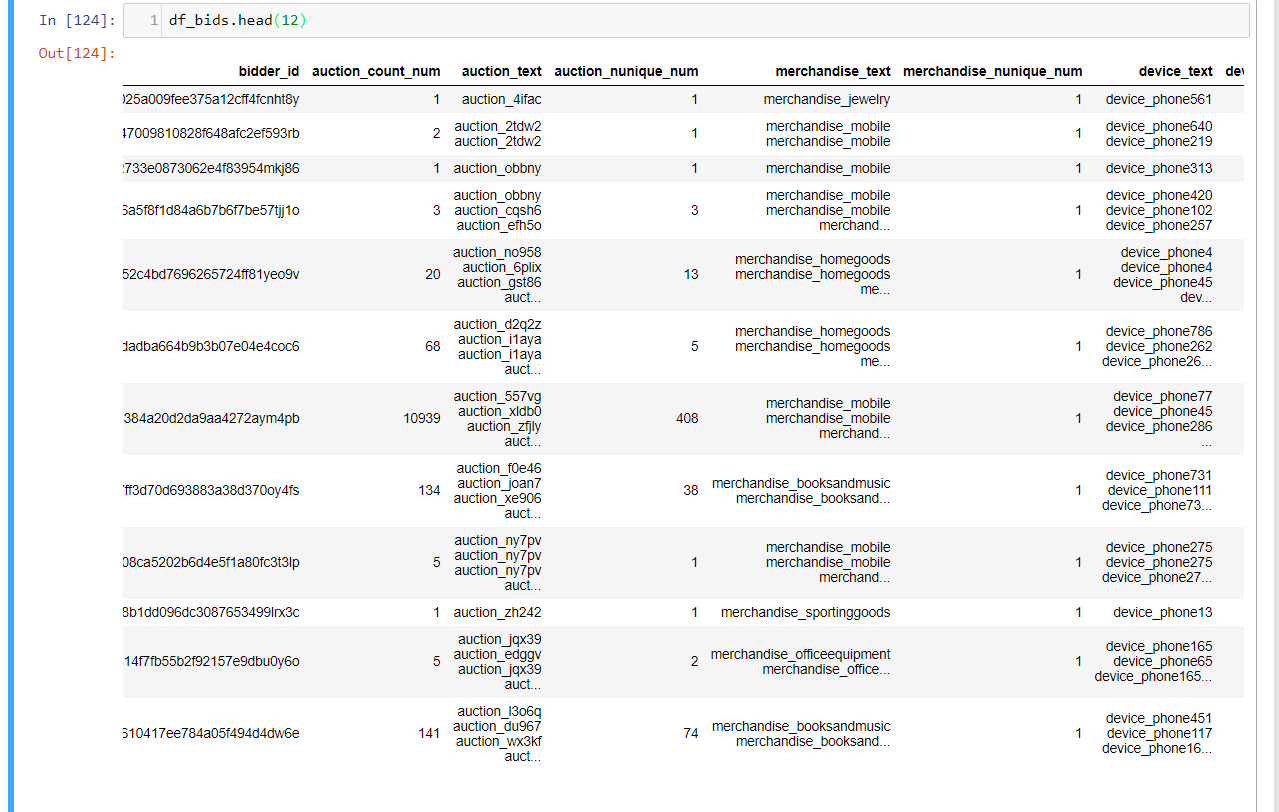
# FEATURE ENGINEERING ON DATASET

1. Screen Shot of Feature Engineering we performed in whole Data set.

* Functions to create time difference and new columns containing categorical columns and count of each categorical column against bidder ID.



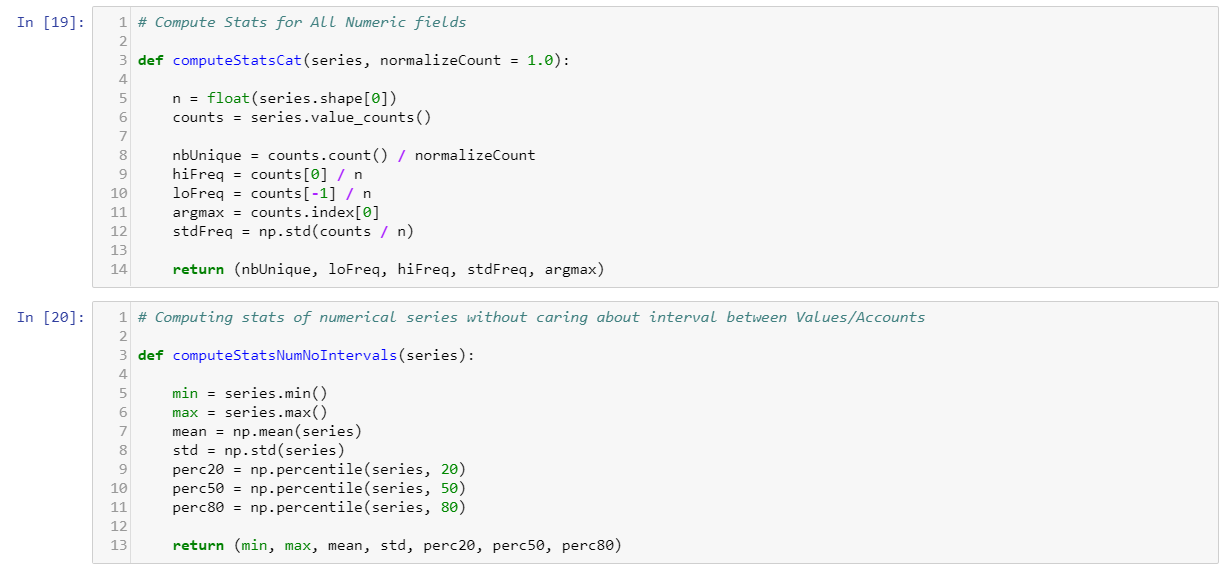
* Below is Screen Shot for count of columns and their names with new columns.



# FUNCTIONS

1. Functions we wrote and referenced from different users to adding features in our model.

* Below is Screen Shot getting count, Hi-Low Frequency, Std Freq. and AgrMax
* Other code is to get Max, min, mean, median and 3 quarter values



1. Function to create new columns based on original values.

* We have created multiple new columns based on previous function we wrote for getting freq. and other values
* Storing those values in a new variable and then creating a new Data frame with all these new features



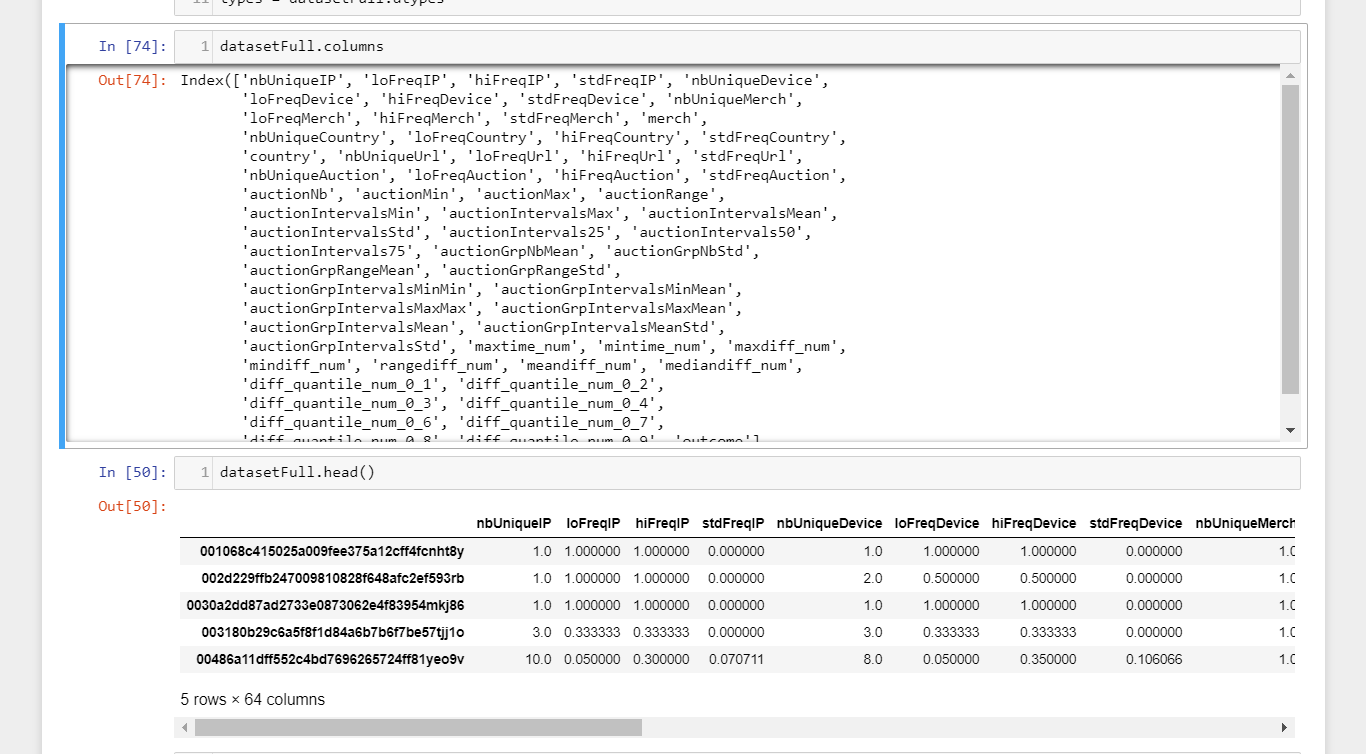
1. Storing data to pickle file.

* We have Stored the new created values in Pickle file to serialize and making optimized use of data.



1. Created new Data frame with new features and displaying below.

* We have Stored the new created values in Pickle file to serialize and making optimized use of data.



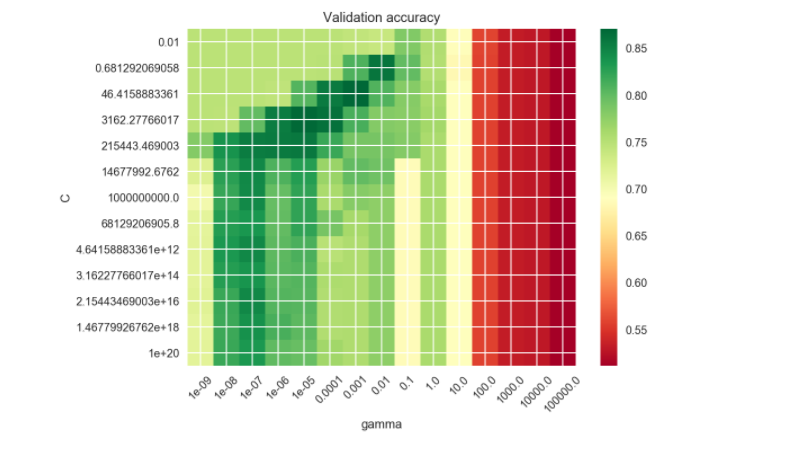
# CLASSIFICATION & RESULT

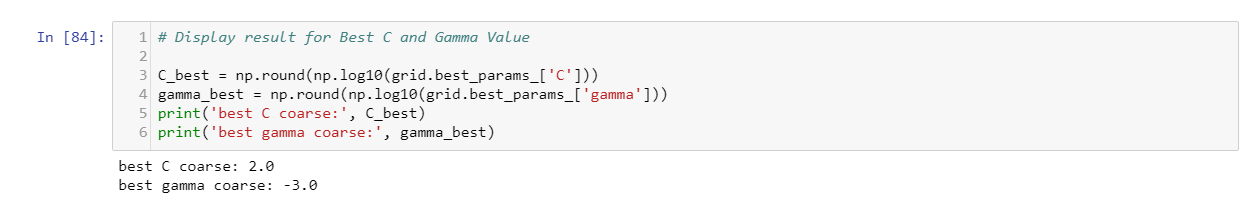
1. Loading pickle file and manupulating in order to get the final dataframe so that can be turned into test and train dataset.

# 

1. GeTting GAMMA and C values best combinations for our model.

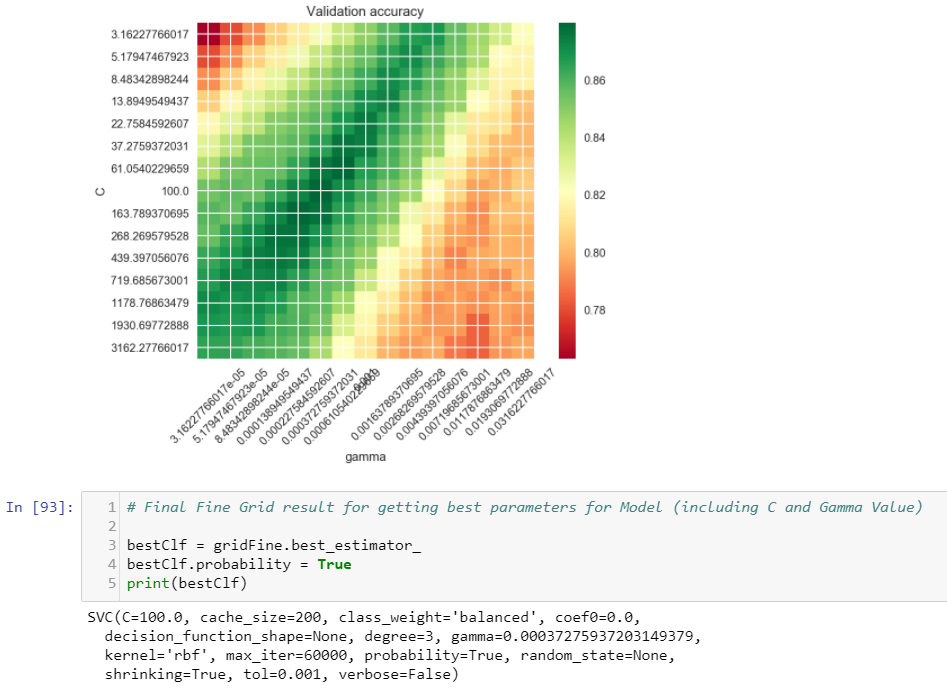
* Getting C and Gamma values for Particular model and finding the best fit for our model
* The Greenest/ Darkest Green color gives the highest possible C and Gamma value combination
* Below given 2nd screenshot gives the best C and Gamma values as of now but still does not give accuracy that was expected





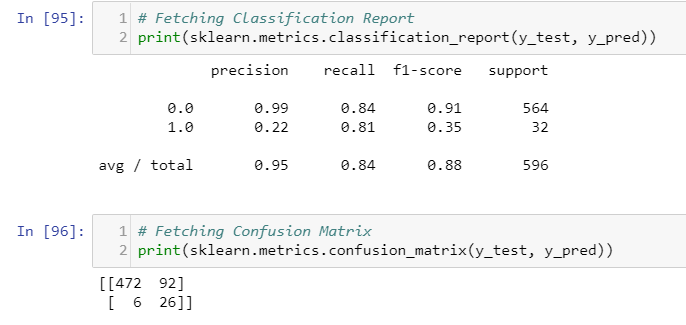
1. GeTting GAMMA and C values best combinations for our model.

* Getting Validation Accuracy from updated Gamma and C values from the below model
* From Fine Grid, we are getting the best estimators values and probabilities for our model.
* Below given C values is : 100 & Below given Gamma value is: 0.0037 \*
* Both of them are still not giving the best classification report.



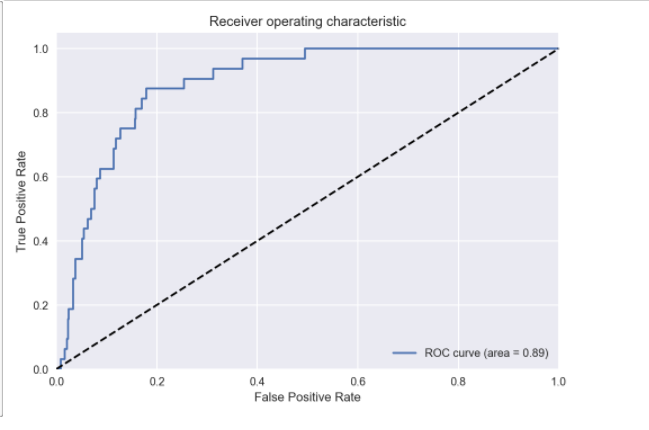
1. Below Screen SHot gives Classification report and confusion matrix for the model.

* Below screen gives us the classification report with precision of 95%
* Also the confusion matrix provide the data that only 92 and 6 are the false results that our prediction gives out of 472 and 26 processed.



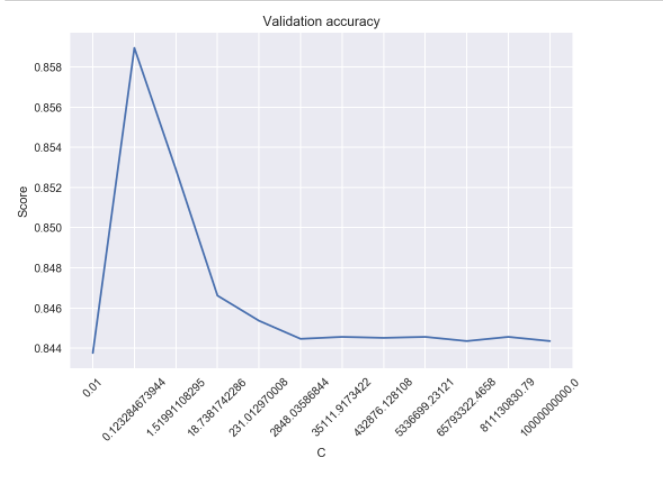
1. Below Screen SHot gives confusion matrix PLOT for the model.

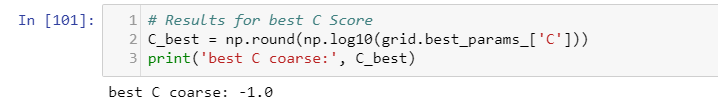
* From below plot we get to know that ROC curve area = 0.89 value.



6. Below Screen SHot gives Best C VALUE PLOT for the model.

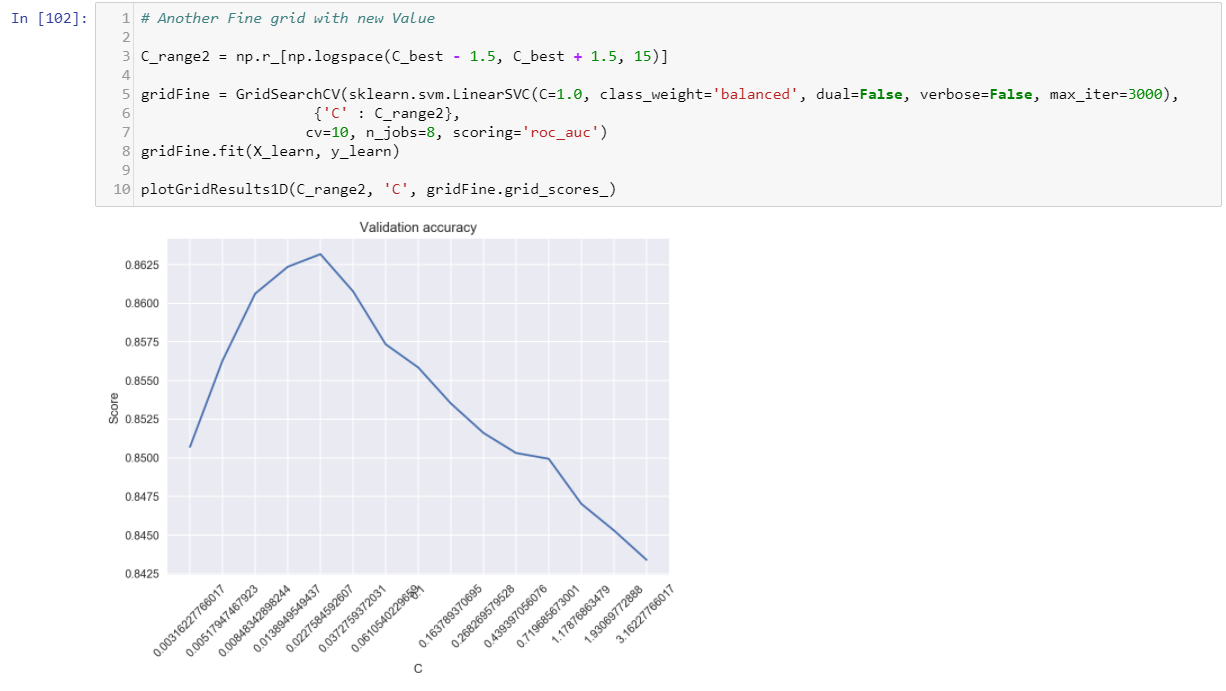
* Below plot gives us the best score compare to respective C value.
* Below plot shows that if C value is similar to 0.1234 around gives the best score
* Below it shows that Best C score for this model is -1.0

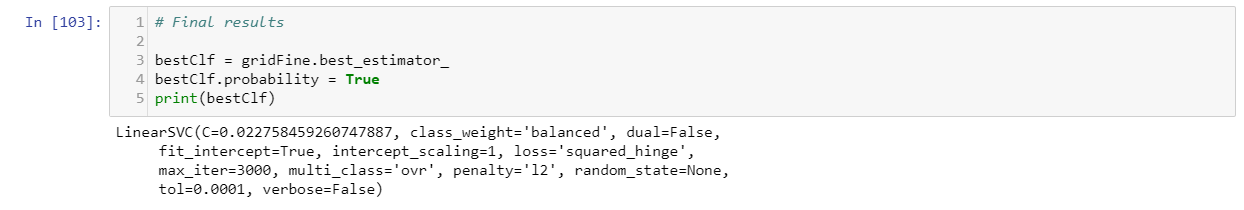




7. Below Screen SHot gives Best C and Gamma Value PLOT for the model.

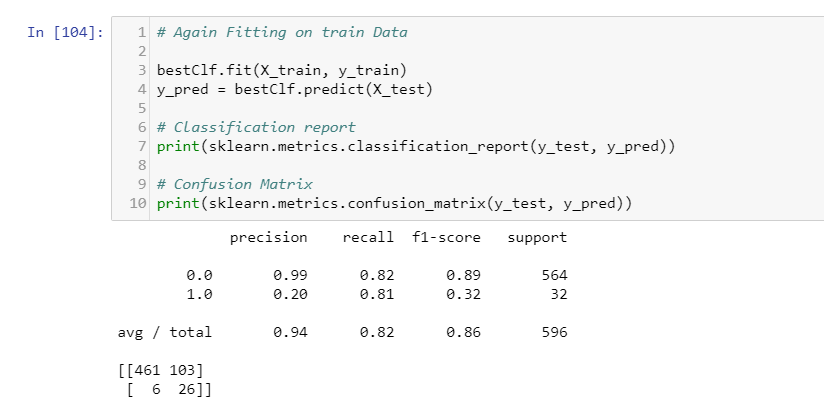
* Below graphs repeat the above steps but with different values
* It gives us the new C value to 0.22, Max iteration = 3000 and total = 0.001





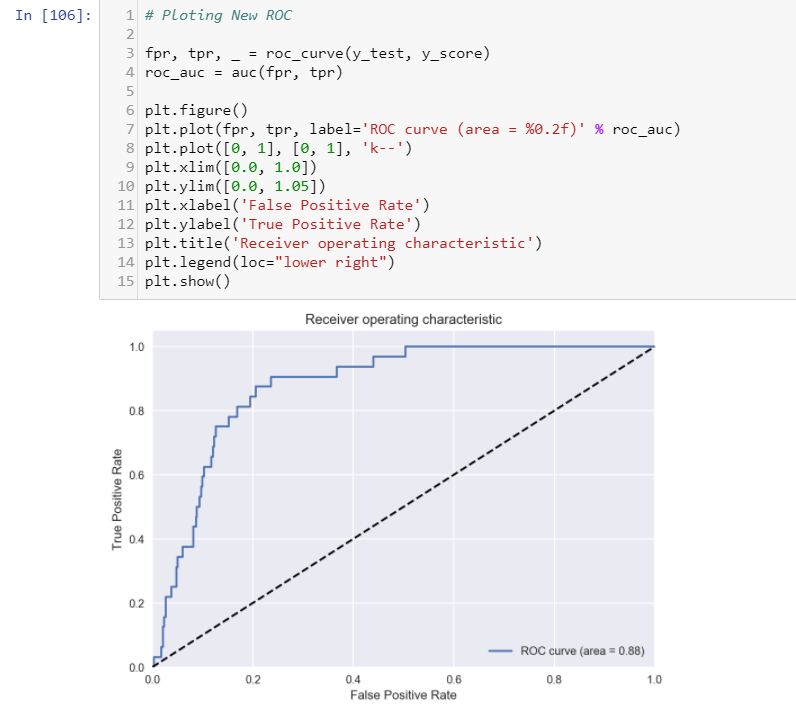
8. Below Screen SHot gives Prediction for the model that includes Classification report and confusion matrix.

* Below given the best precision is 0.94
* Confusion matrix give 103 and 6 error out of 461 and 26 respectively



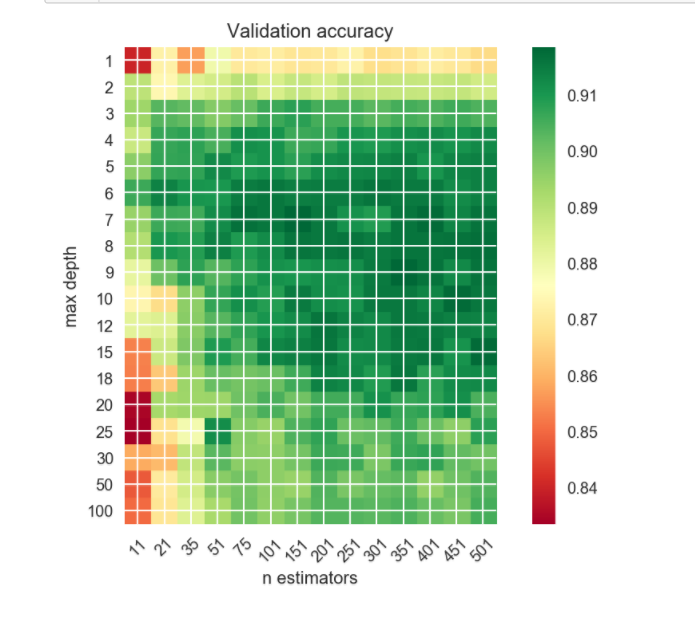
9. Below Screen SHot GIVES the ROC curve.

* Below given roc curve stands for the confusion matric and their accuracy
* Confusion matrix gives the accuracy of 0.88



10 . Below Screen SHot GIVE best values for max depth and n\_estimators.

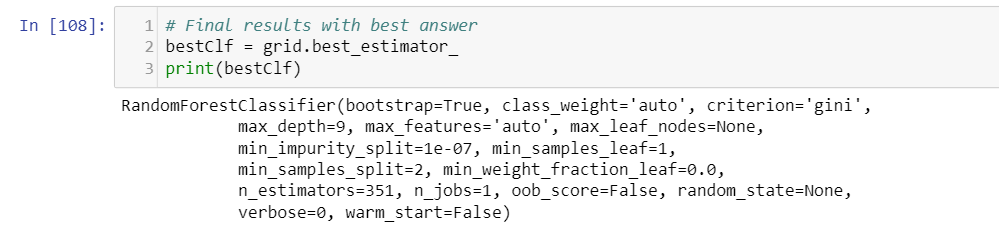
* Plot gives the best estimators and max depth for my random forest method

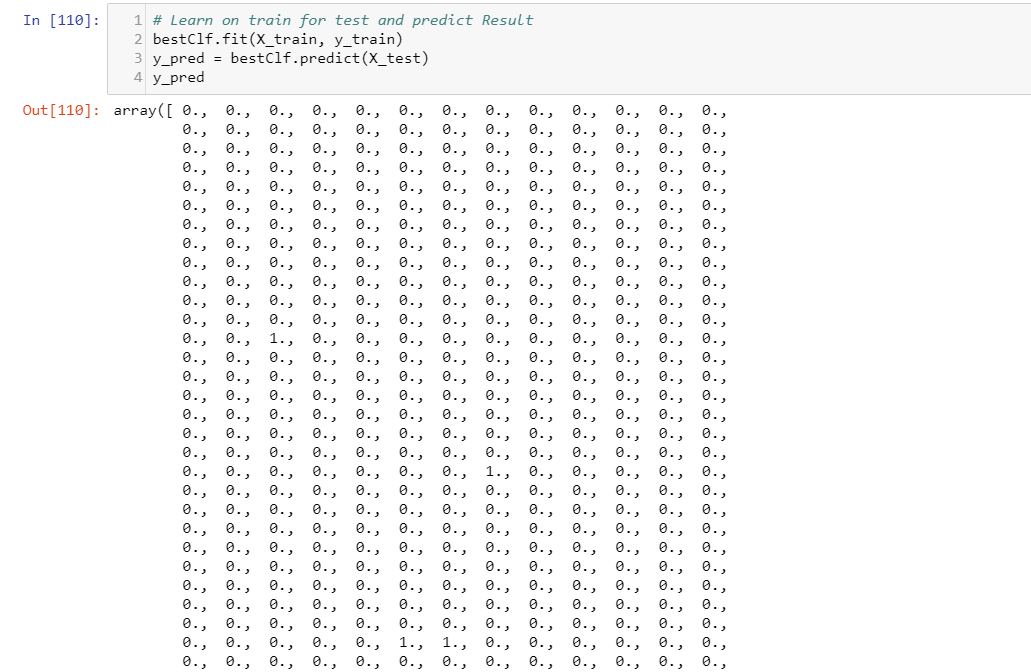


# FINAL MODEL

1. Below Screen SHot GIVE best values for rfc method.

Below screen shot have the values and best predicted results

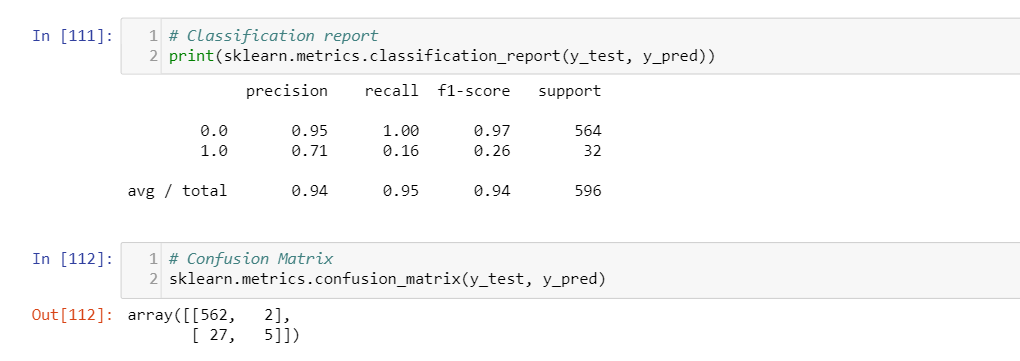




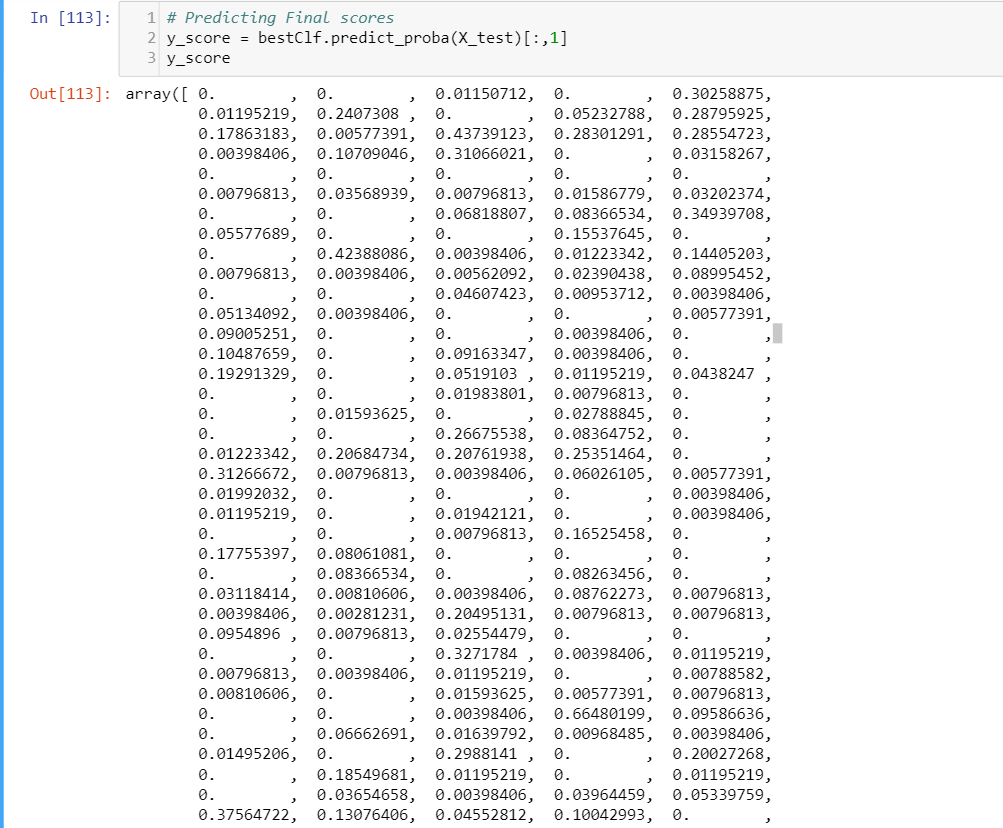
2. Below Screen SHot GIVE Classification report and confusion matrix for method.

- Below screen shot gives the best possible score that 0.94 as classification report

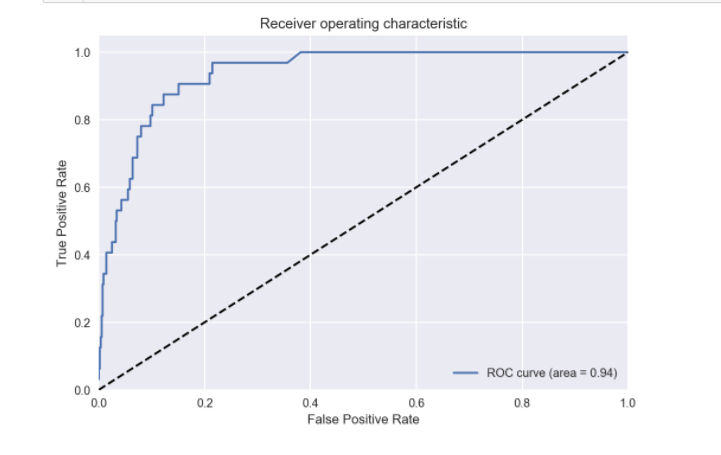
- Confusion matrix having lowest error found this time which is 2 and 27 out of 562 and 5



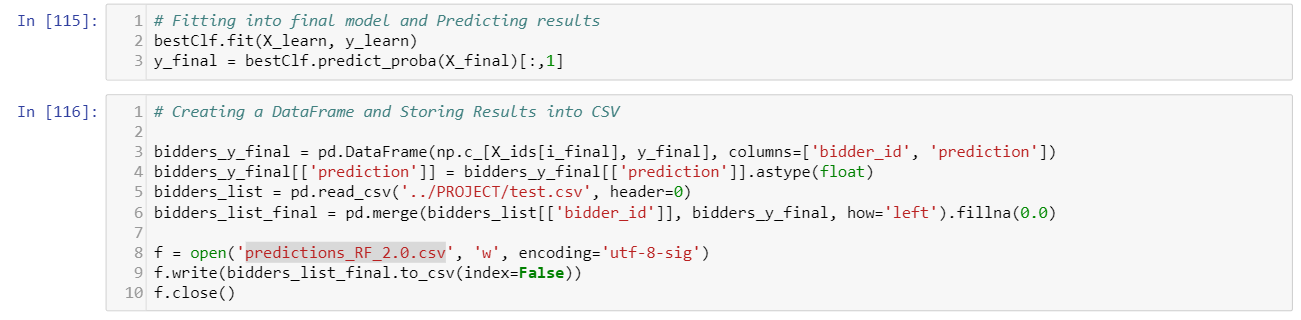
- Below is the result of our prediction



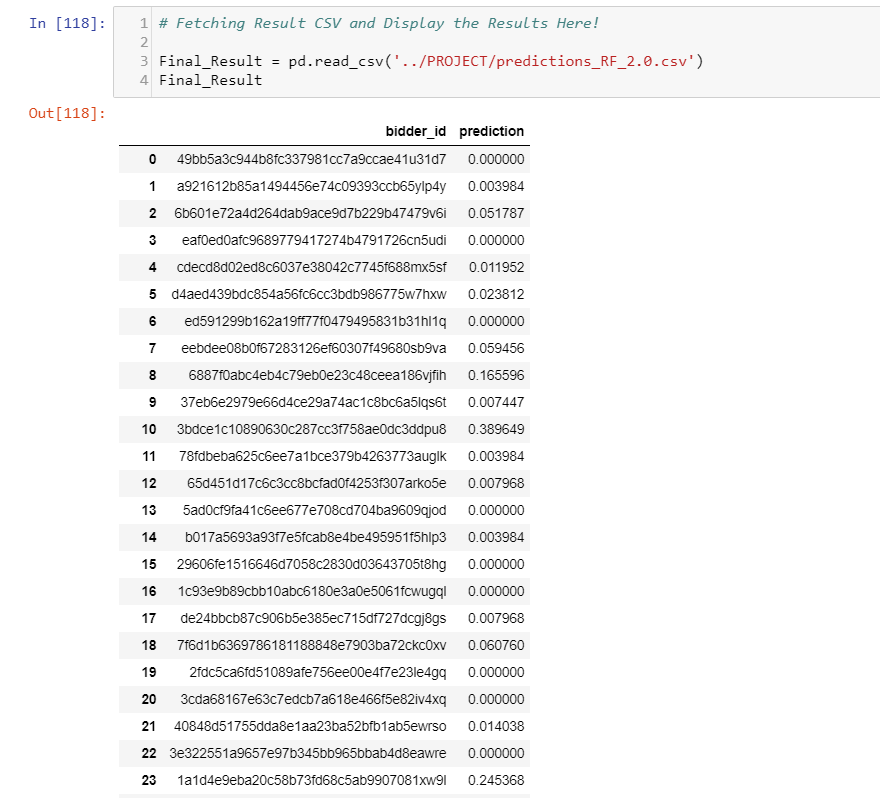
3. Below Screen SHot of ROC curve of final model.



4. Getting final result and storing into a new csv file.



5. Getting final model results by fetching csv.



# REFERENCES

* https://www.kaggle.com/c/facebook-recruiting-iv-human-or-bot
* https://blog.dataiku.com/2015/06/09/kaggle-feature-engineering
* http://small-yellow-duck.github.io/auction.html
* https://github.com/mikegloudemans/kaggleBotOrNot
* https://github.com/rushter/Facebook-Recruiting
* https://rinzewind.org/blog-en/2015/my-take-on-kaggles-facebook-recruiting-iv-human-or-robot.html
* https://github.com/wallinm1/kaggle-facebook-bot/blob/master/facebook\_notebook.ipynb
* https://github.com/small-yellow-duck/facebook\_auction/blob/master/fba-sub9.py
* http://www.thomas-robert.fr/en/kaggles-human-or-robot-competition-feedback/
* http://podlipensky.com/2015/06/kaggle-competition-aftermath/