**Data Preparation:**

* Separated Nick’s family\_survival.csv file into family\_survival\_test.csv and family\_survival\_train.csv as test.csv and train.csv are similarly two separate files. I then create dataframes of these two files and join them to the existing test and train dataframes on PassengerId.
* Ran cleanData.r and wrote test and train CSV files from dataNewTest and dataNewTrain dataframes respectively. I then removed all but AgeBucket, Imputed\_Age, and Imputed\_Binned\_Age, and PassengerId
* Turned Imputed\_Binned\_Age to numeric using the following conversion:
  + Child -> 1
  + Adolescent -> 2
  + Young Adult -> 3
  + Adult -> 4
  + Old -> 5
* The reason was that some of the models were failing to run unless this feature were numeric

**Information, Charts, and Plots Created by Python Notebook**

* Number of passengers traveling alone vs. with family (1 = alone, 0 = not alone) using SibSp and Parch

1 537

0 354

Name: alone, dtype: int64

* Description of each feature in dataframes

Data columns (total 16 columns):

PassengerId 891 non-null int64

Survived 891 non-null int64

Pclass 891 non-null int64

Name 891 non-null object

Sex 891 non-null object

SibSp 891 non-null int64

Parch 891 non-null int64

Ticket 891 non-null object

Fare 891 non-null float64

Cabin 204 non-null object

Embarked 889 non-null object

familyid 891 non-null int64

Family\_member\_survival 891 non-null int64

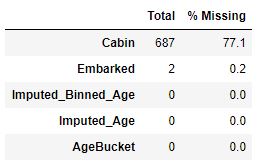
AgeBucket 891 non-null int64

Imputed\_Age 891 non-null float64

Imputed\_Binned\_Age 891 non-null int64

dtypes: float64(2), int64(9), object(5)

* Fields with Missing Data



* Get the most common value of “Embarked” in order to fill in missing values with it, since there are only 2 values missing

count 889

unique 3

top S

freq 644

Name: Embarked, dtype: object

* Get descriptions of our features again, now we have 17 of them

Data columns (total 17 columns):

Survived 891 non-null int64

Pclass 891 non-null int64

Name 891 non-null object

Sex 891 non-null object

SibSp 891 non-null int64

Parch 891 non-null int64

Ticket 891 non-null object

Fare 891 non-null float64

Embarked 891 non-null object

familyid 891 non-null int64

Family\_member\_survival 891 non-null int64

AgeBucket 891 non-null int64

Imputed\_Age 891 non-null float64

Imputed\_Binned\_Age 891 non-null int64

relatives 891 non-null int64

alone 891 non-null int32

Deck 891 non-null int32

dtypes: float64(2), int32(2), int64(9), object(4)

* Get a count of unique values in “Ticket” feature. We see that we have 681 unique values, so this feature is probably not particularly useful, so we drop it

count 891

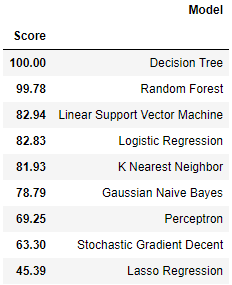
unique 681

top CA. 2343

freq 7

Name: Ticket, dtype: object

* Get the scores of each model using the training data, as we don’t have labels for the testing data. Though sometimes Decision Tree has a higher score than Random Forest, the difference is negligible, and the scores from cross validation are higher for Random Forest, so we used Random Forest



* The following are cross validation scores, means, and standard deviations of both Random Forest and Decision Tree

Scores - Random Forest: [0.78888889 0.78888889 0.73033708 0.84269663 0.85393258 0.83146067

0.82022472 0.78651685 0.86516854 0.84090909]

Mean - Random Forest: 0.8149023947338554

Standard Deviation - Random Forest: 0.03887396177531784

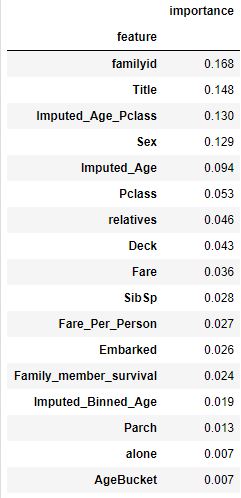
Scores - Decision Tree: [0.76666667 0.8 0.74157303 0.78651685 0.75280899 0.76404494

0.79775281 0.7752809 0.73033708 0.76136364]

Mean - Decision Tree: 0.7676344909771877

Standard Deviation - Decision Tree: 0.02170918613402234

* The following are the “importances” of the features in our training set.



* The following is our prediction score, followed by our “out of bag” scores pre and post hyperparameter tuning, followed by our precision and recall scores

100.0 %

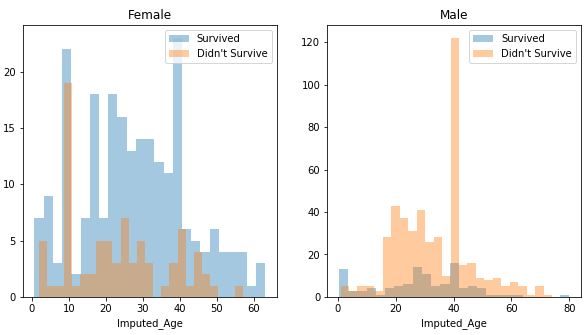
oob score: 81.93 %

oob score: 81.71000000000001 %

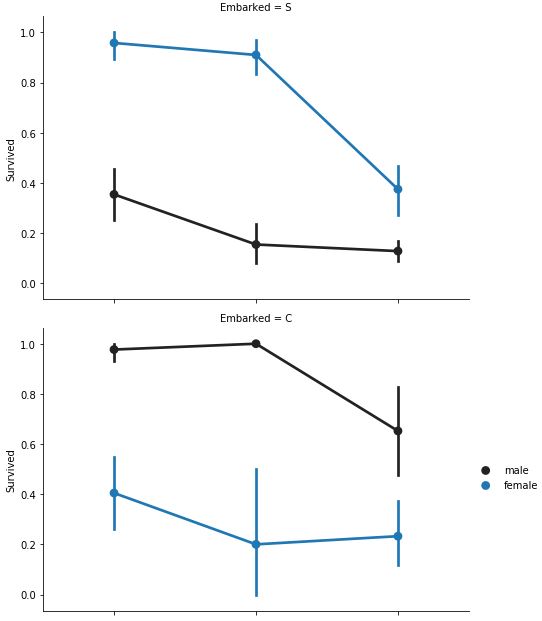
Precision: 0.8080808080808081

Recall: 0.7017543859649122

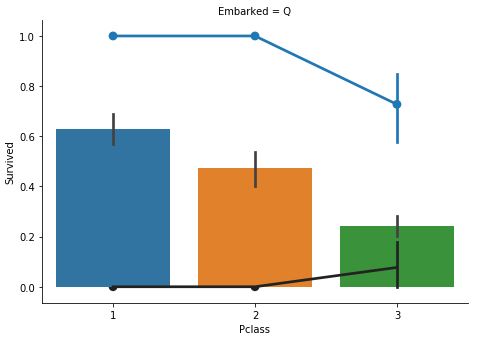
* The following is a comparison of gender by age (imputed age) survivability



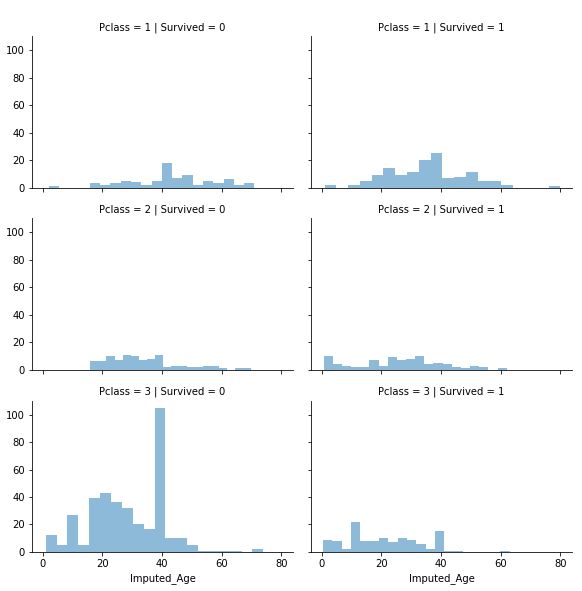
* Facetgrid or survivability per “Embarked” class



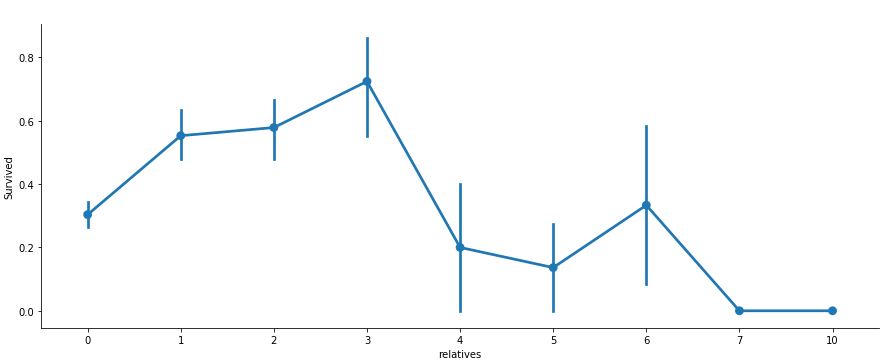
* Facetgrid of survivability with “Embarked” classes and PClass barcharts



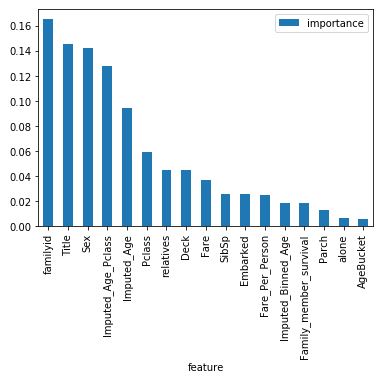
* Age and PClass compared to survivability

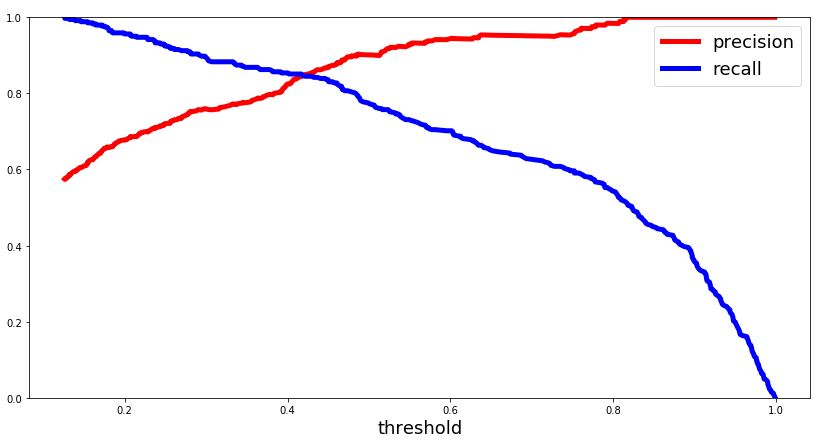


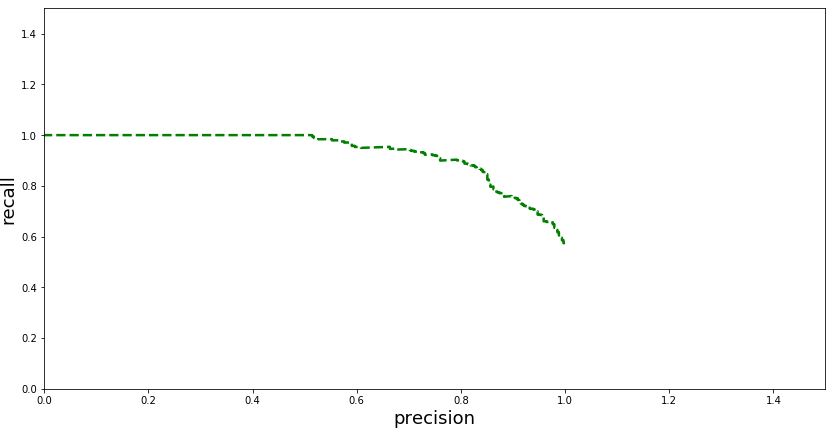
* Factorplot of number of relatives vs. survival

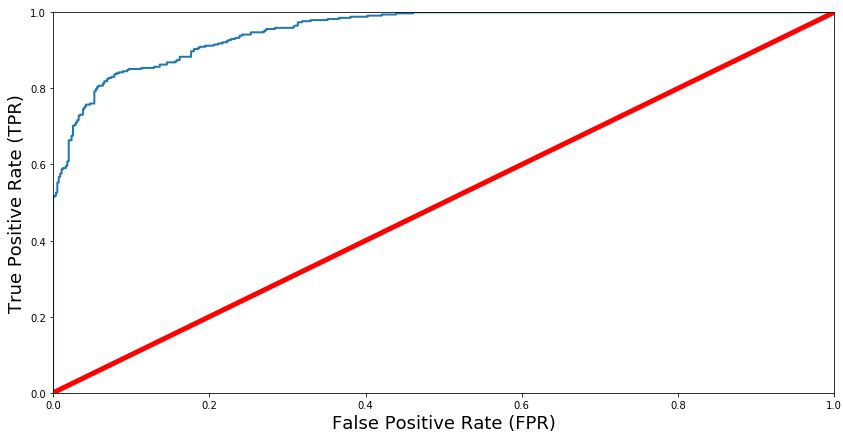


* Importances in bar chart form









Finally, we output the variable yPred, which our are test data predictions using Random Forest, to a file named Titanic\_Mortality\_Predictions.csv, containing features PassengerId and Survived, which are then submitted to Kaggle.