Titanic: Machine Learning from Disaster

HARJONO, Natasha Valerie (Natasha)

TSUI, Ying Tsz (Yvonne)

WAHYU, Zoya Estella (Zoya)

Data Overview

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

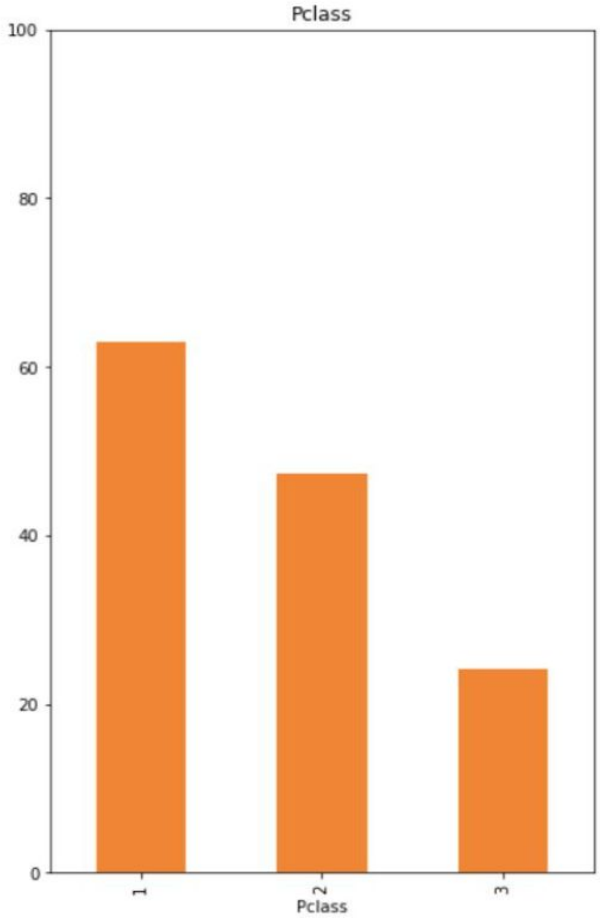
TRAIN

891 people
12 columns
Target Variable

TEST

481 people
11 columns
Non Target Variable

Data Pre-Processing

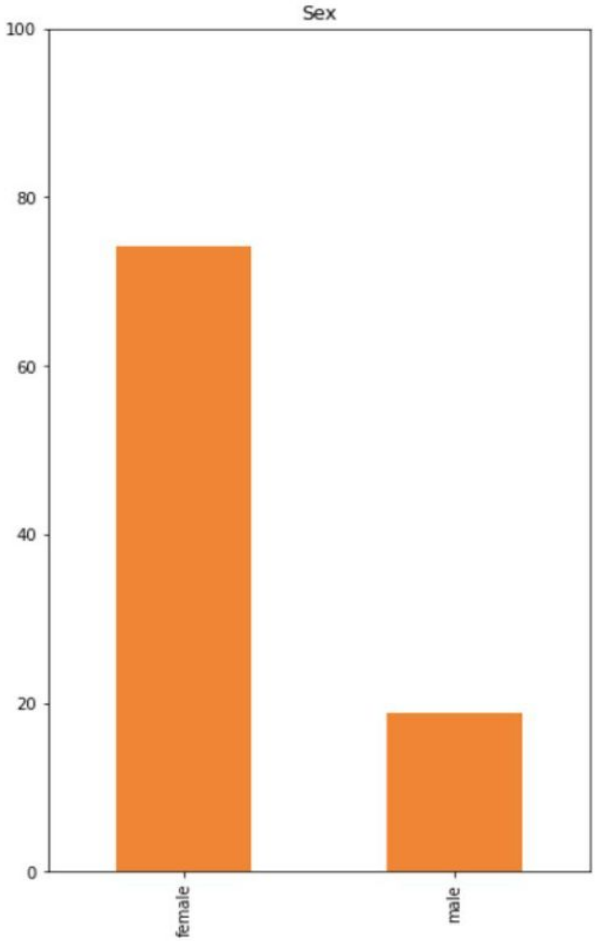


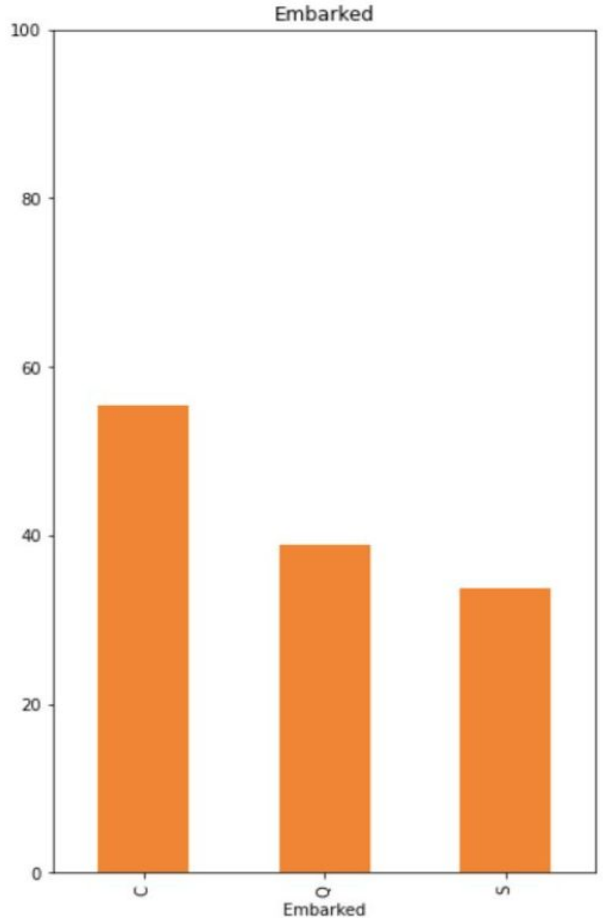
P-CLASS

CLASS 1 > CLASS 2 > CLASS 3

SEX

FEMALE > MALE

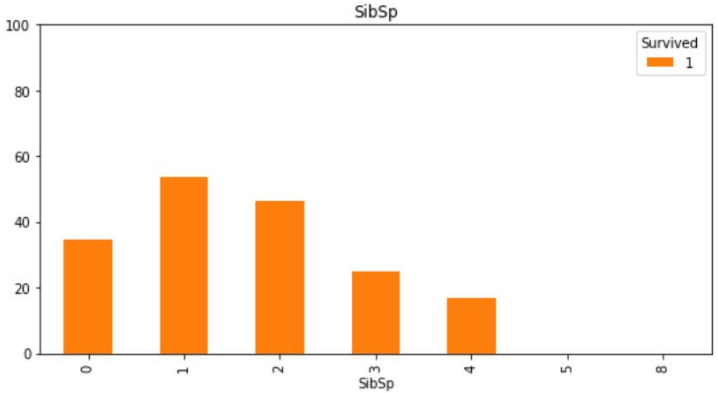


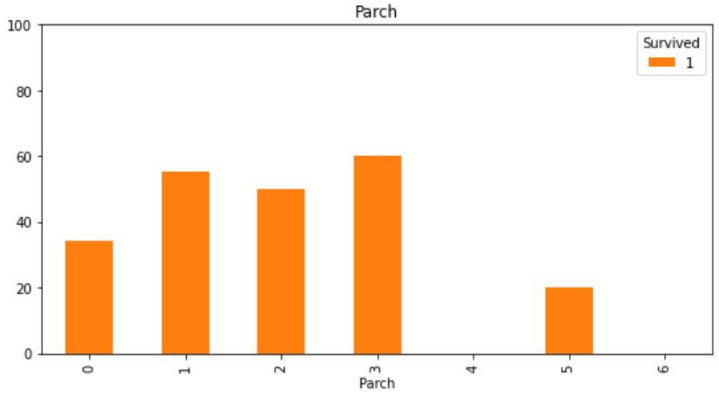


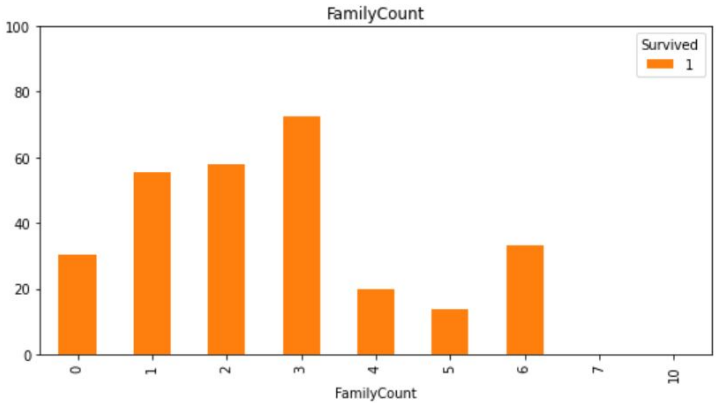
EMBARKED

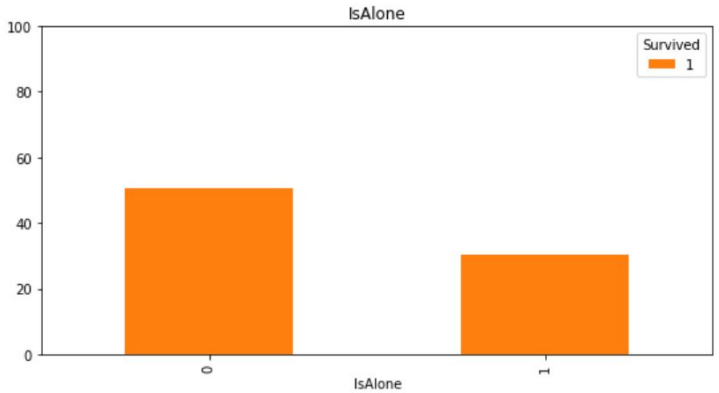
C > Q > S

Number of Group Members

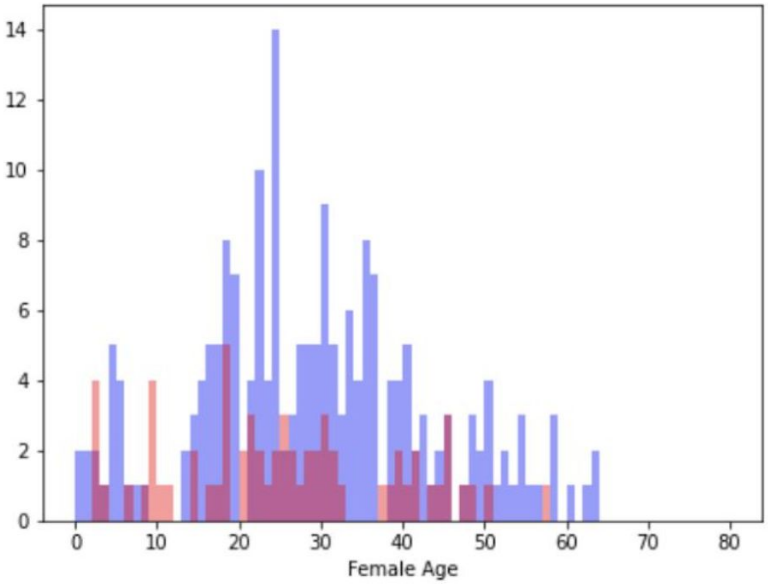


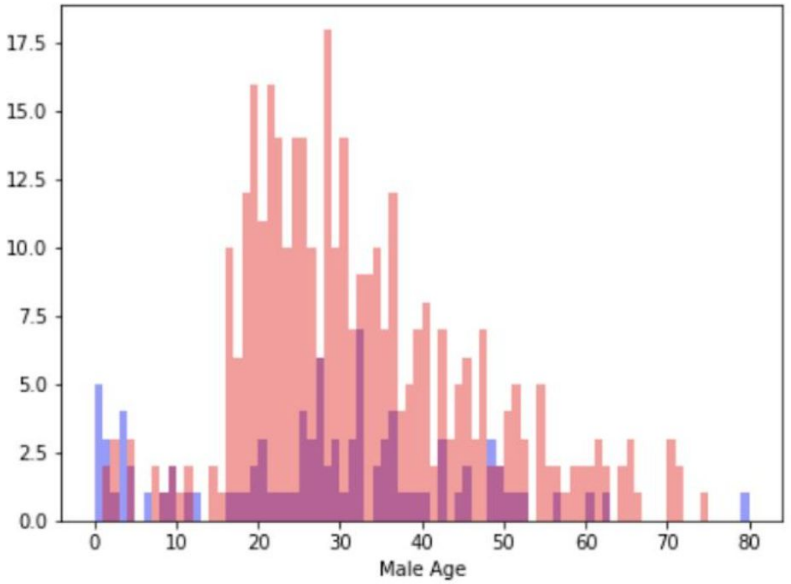






SEX AND AGE


14
12
10
8
6
4
2
0
0
10
20
30
40
50
60
70
80
Female Age


17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
10
20
30
40
50
60
70
80
Male Age

MISSING VALUES

PassengerId	0
Survived	418
Pclass	0
Name	0
Sex	0
Age	263
SibSp	0
Parch	0
Ticket	0
Fare	1
Cabin	1014
Embarked	2

AGE

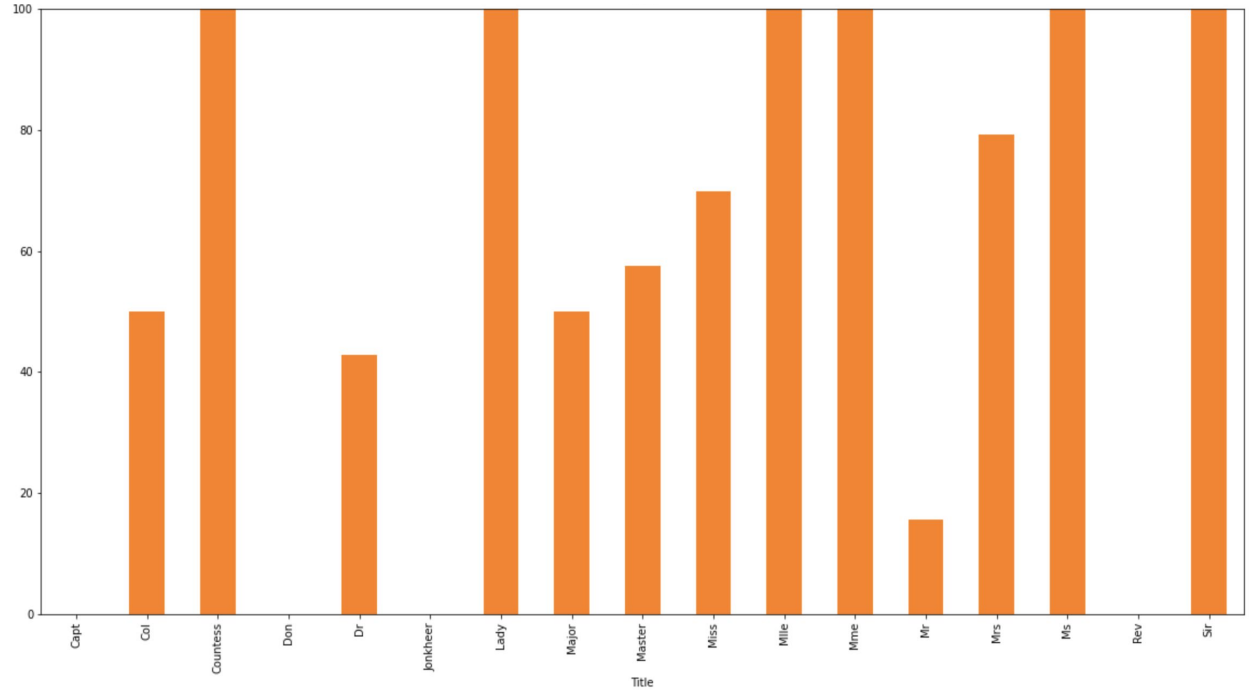
```
for df in [train_num, test_num]:
    # get all the existed features
    age_df = df[['Age','Fare', 'Parch', 'SibSp', 'Pclass', 'Title']]
    # divide the passengers to known and unknown
    known_age = age_df[age_df.Age.notnull()].values
    unknown_age = age_df[age_df.Age.isnull()].values
    # y is tha target age
    y = known_age[:, 0]
    # X is the feature value
    X = known_age[:, 1:]
    # fit
    rfr = RandomForestRegressor(random_state=15, n_estimators=2000, n_jobs=-1)
    rfr.fit(X, y)
    # use the model to predict
    age_pred = rfr.predict(unknown_age[:, 1::])
    df.loc[ (df['Age'].isnull()), 'Age' ] = age_pred

train['Age'] = train_num['Age']
test['Age'] = test_num['Age']
del train_num
del test_num
```

Feature Engineering

TITLE

Capt	1
Col	4
Countess	1
Don	1
Dona	1
Dr	8
Jonkheer	1
Lady	1
Major	2
Master	61
Miss	260
Mlle	2
Mme	1
Mr	757
Mrs	197
Ms	2
Rev	8
Sir	1



DECK

C90 → C , B191 → B

FARE PER PERSON

Fare / No. of Group Members

FAMILY COUNT

Sibs + Parch

IS ALONE

0 / 1

Data Finalization

Age	0
Deck	0
Died	418
Embarked	0
FamilyCount	0
Fare	0
FarePerPerson	0
IsAlone	0
Parch	0
PassengerId	0
Pclass	0
Sex	0
SibSp	0
Survived	418
Title	0

Transform the categorical features to numerical features.

get_dummies

Deck
X
C
X
C
X

Deck_A	Deck_B	Deck_C	Deck_D	Deck_E	Deck_F	Deck_G	Deck_T	Deck_X
0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1

Data Finalization

1. Drop "**Passenger ID**" . (We didn't really drop it, but when being fed into model, it wasn't included in the train set)

2. Drop "**Ticket**" column

3. Drop "**Cabin**" column

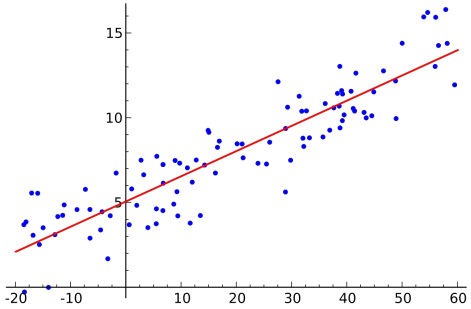
4.
```
x = train.iloc[:, 2:].values
y = train.iloc[:, 1].values
x_test = test.iloc[:, 2:].values
```

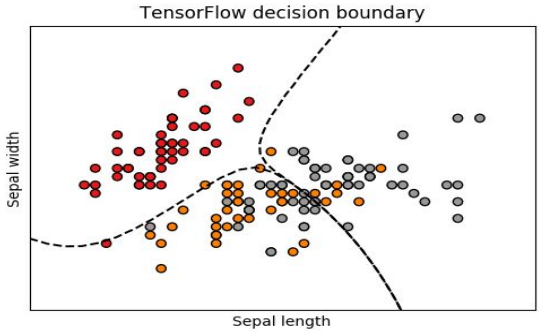
x: 891 rows, 30 columns
y: 891 rows, 1 column
x_test:418 rows ,30 columns

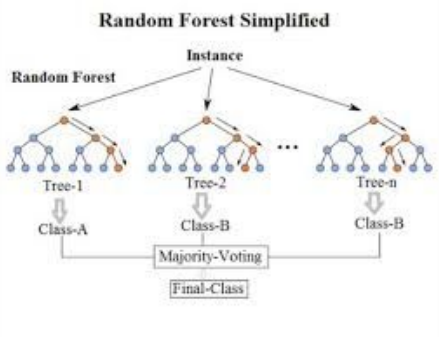
Data Finalization

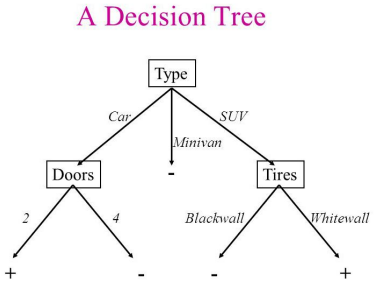
	PassengerId	Survived	Sex	Age	SibSp	Parch	Fare	Died	FamilyCount	IsAlone	...	Embarked_S	Deck_A	Deck_B	Deck_C	Deck_D	Deck_E	Deck_F
0	1	0	1	2	1	0	0	1	1	0	...	1	0	0	0	0	0	0
1	2	1	0	3	1	0	3	0	1	0	...	0	0	0	1	0	0	0
2	3	1	0	2	0	0	1	0	0	1	...	1	0	0	0	0	0	0
3	4	1	0	3	1	0	3	0	1	0	...	1	0	0	1	0	0	0
4	5	0	1	3	0	0	1	1	0	1	...	1	0	0	0	0	0	0
5	6	0	1	2	0	0	1	1	0	1	...	0	0	0	0	0	0	0
6	7	0	1	5	0	0	3	1	0	1	...	1	0	0	0	0	1	0
7	8	0	1	0	3	1	2	1	4	0	...	1	0	0	0	0	0	0
8	9	1	0	2	0	2	1	0	2	0	...	1	0	0	0	0	0	0
9	10	1	0	1	1	0	2	0	1	0	...	0	0	0	0	0	0	0

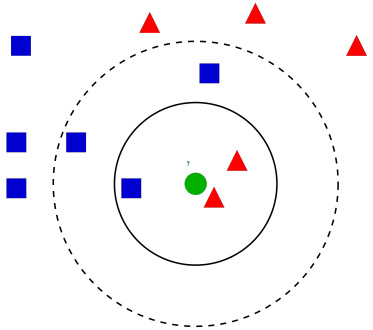
Model Selection

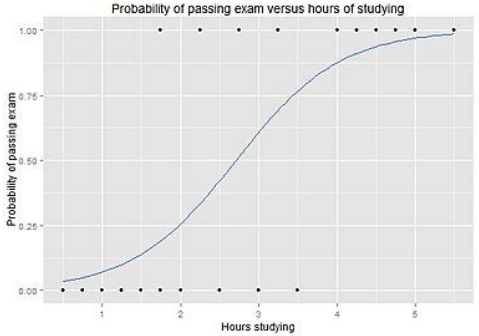










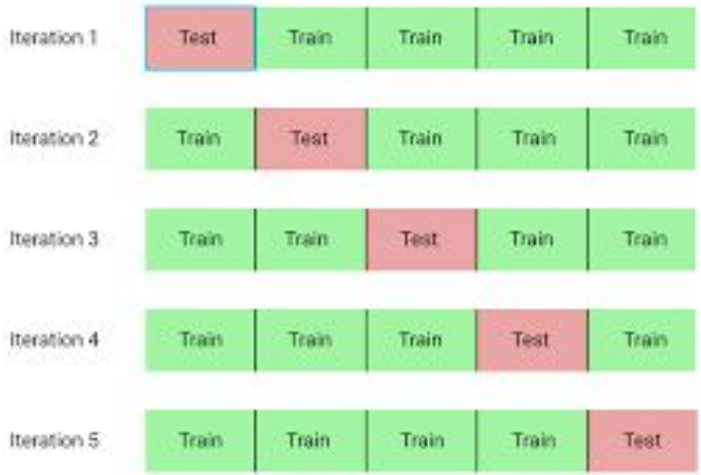


Model Selection

```
Linear Regression model: 45.73%
Naive Bayes Classifier model: 80.13%
Random Forest model: 98.77%
Decision Tree model: 98.77%
KNN model: 81.71%
Logistic Regression model: 84.29%
```

A roughly estimation with default parameters.
Compare the score from different models and finally conclude that RF and Decision Tree model have a better performance.

Model Selection



K-fold cross validation (K=5)

More precise
More stable
All data could be used

```
          Score Mean +/- Std
Random Forest: 80.69 +/- 2.97 %
Decision Tree: 77.67 +/- 2.74 %
```

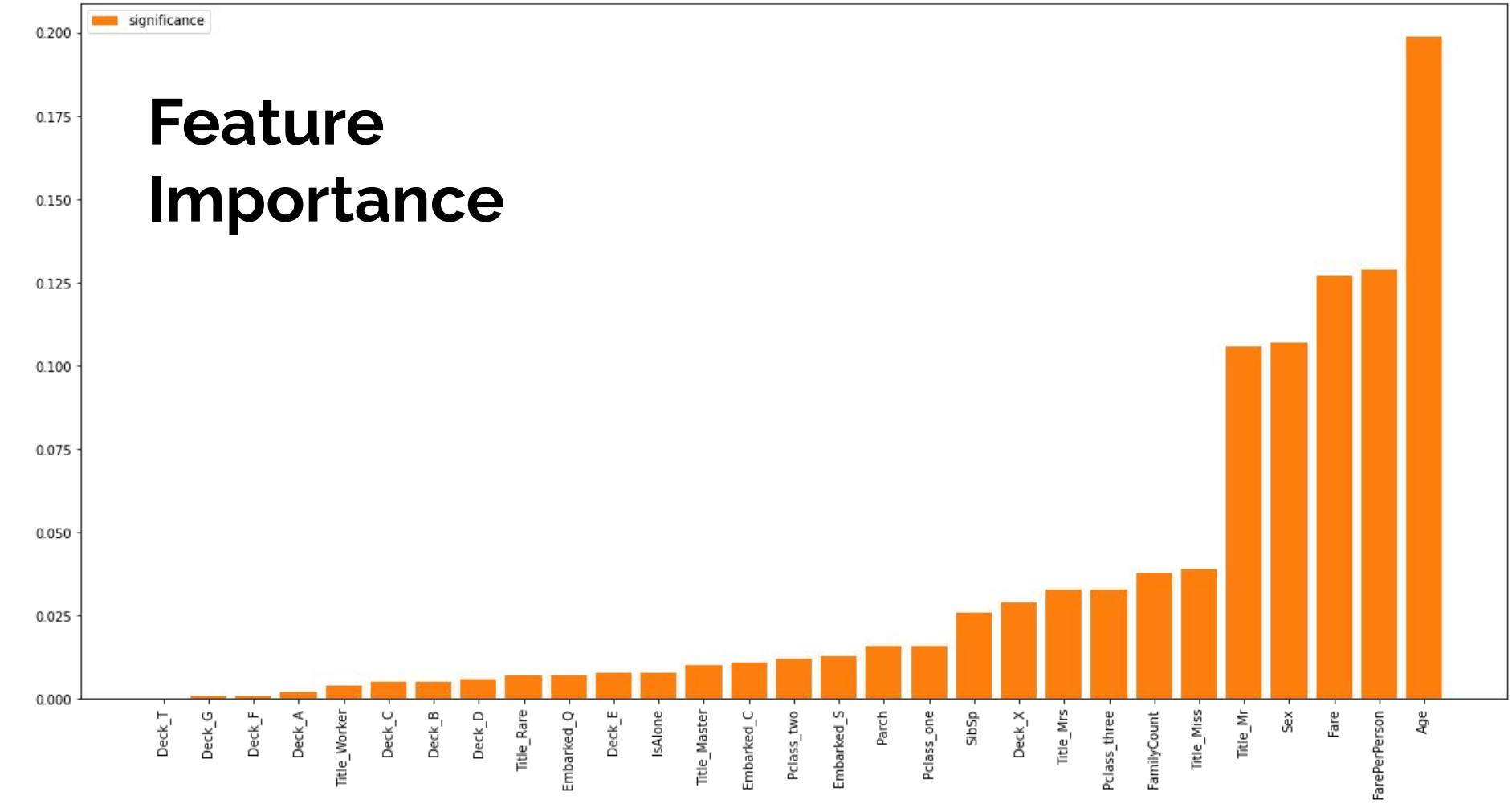
Random Forest

Hyperparameter Tuning

```
random_forest = RandomForestClassifier(criterion='gini',
                                       n_estimators=1750,
                                       max_depth=7,
                                       min_samples_split=6,
                                       min_samples_leaf=6,
                                       max_features='auto',
                                       oob_score=True,
                                       random_state=42,
                                       n_jobs=-1,
                                       verbose=1)
```

GridsearchCV

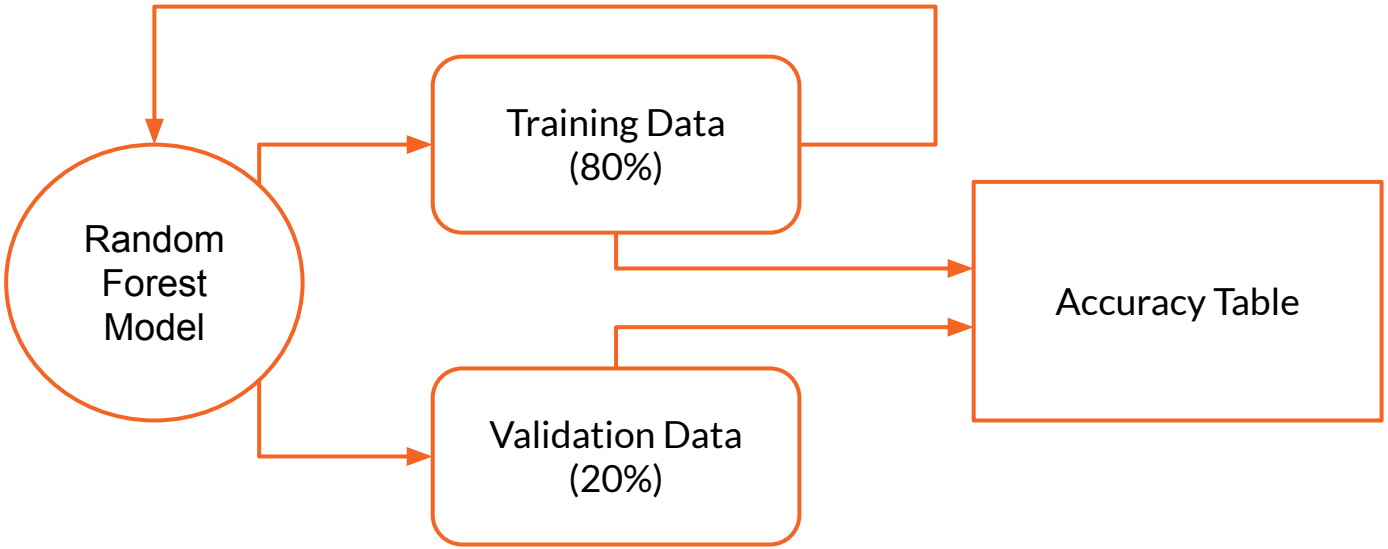
- n_estimators
- max_depth
- min_sample_split
- min_sample_leaf
- min_sample_split
- max_features

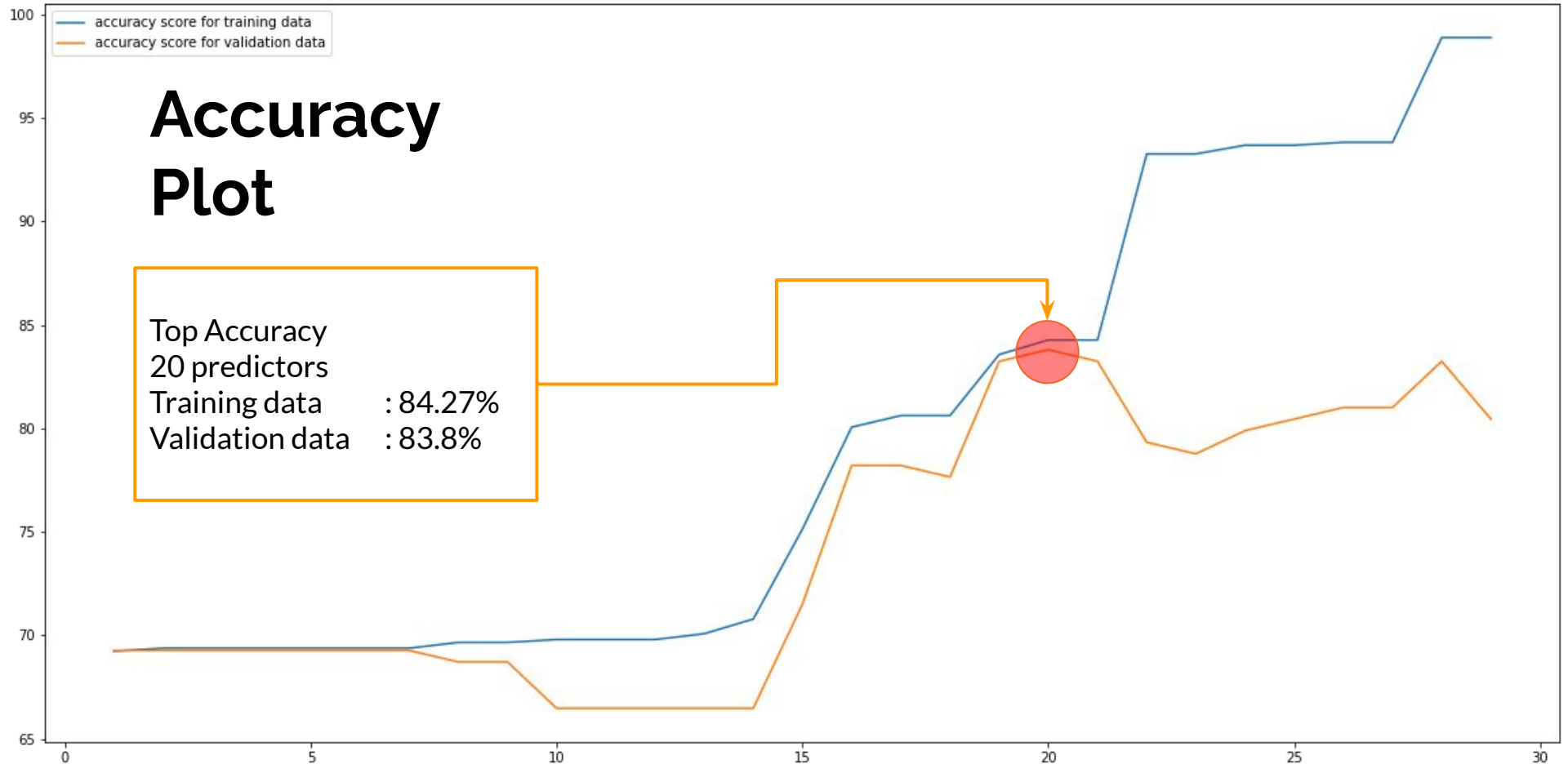


Feature Importance

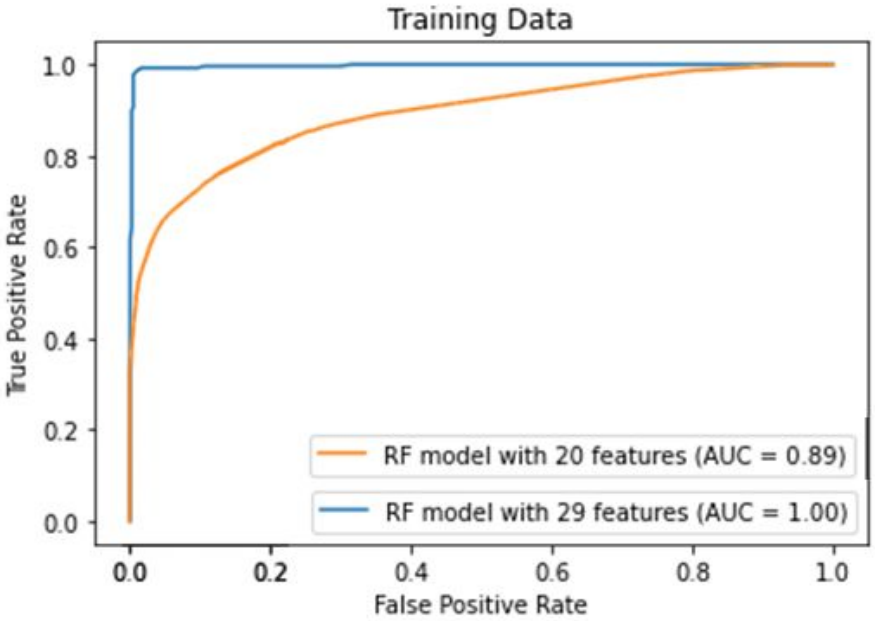
Feature Selection

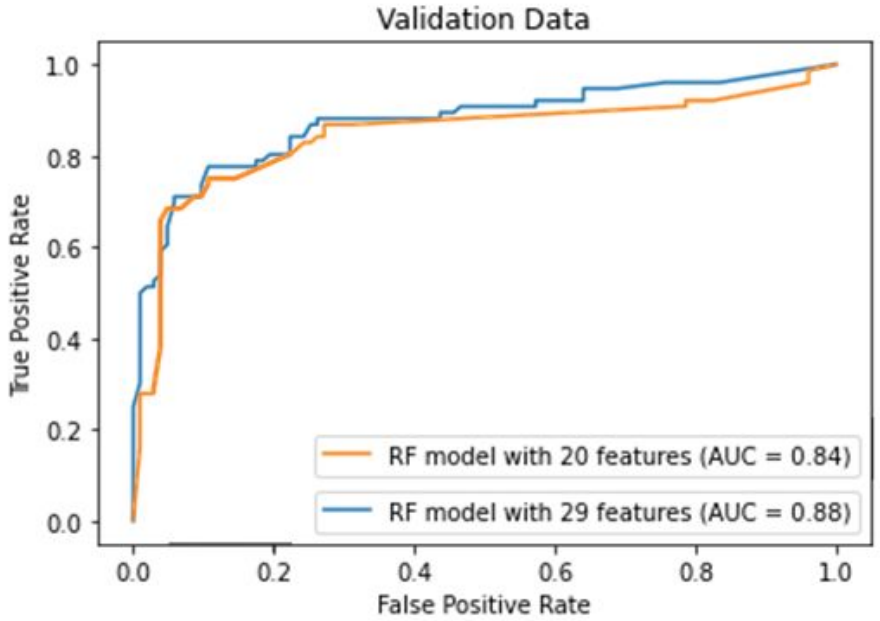
Subtract the least significant predictor of the model
\+ **Re-fit** the remaining predictors to the model

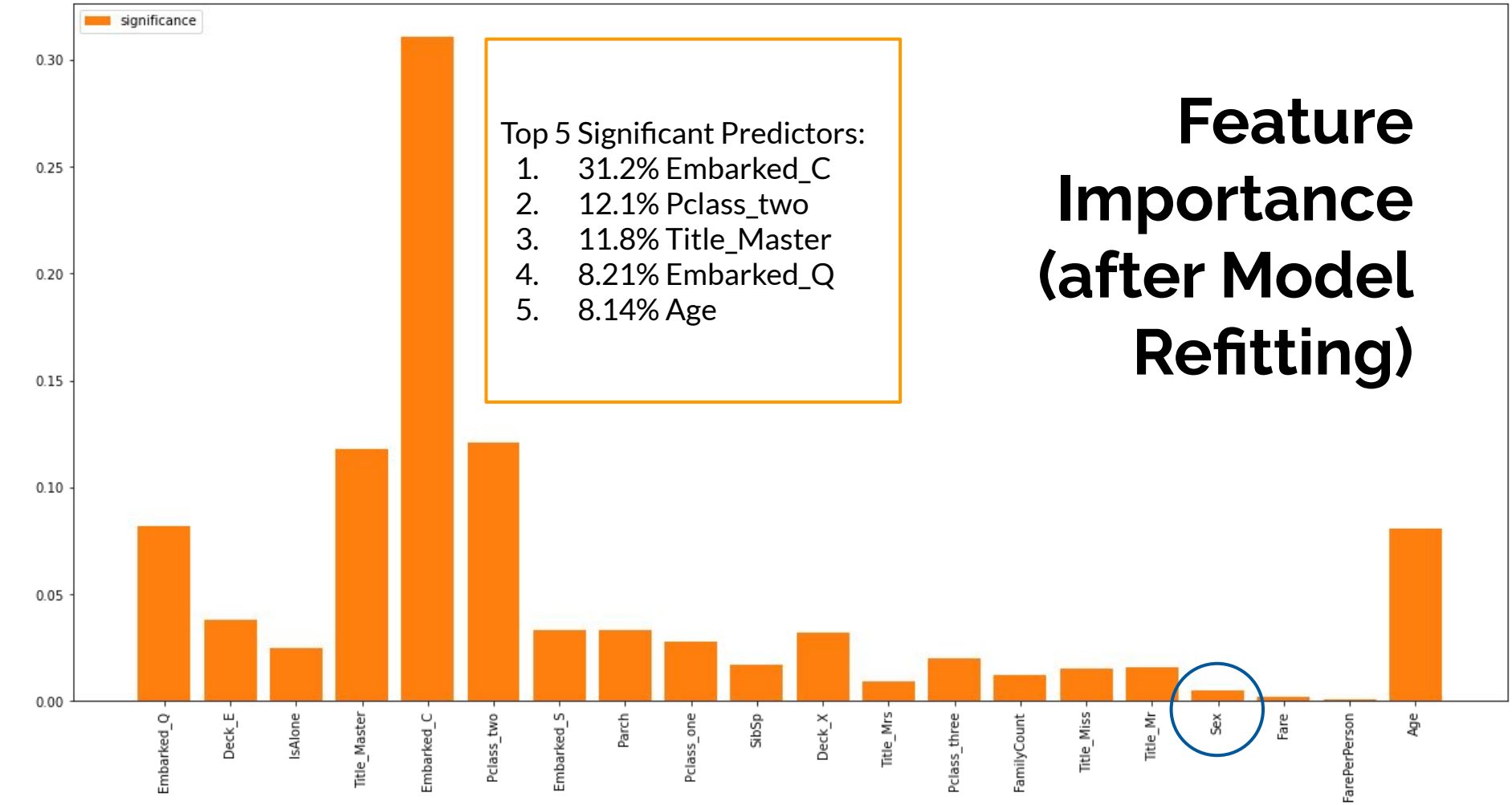



accuracy score for training data
accuracy score for validation data
Accuracy Plot
Top Accuracy
20 predictors
Training data : 84.27%
Validation data : 83.8%
100
95
90
85
80
75
70
65
0
5
10
15
20
25
30

ROC Curve


Training Data
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
RF model with 20 features (AUC = 0.89)
RF model with 29 features (AUC = 1.00)


Validation Data
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
RF model with 20 features (AUC = 0.84)
RF model with 29 features (AUC = 0.88)



Top 5 Significant Predictors:

1. 31.2% Embarked_C
2. 12.1% Pclass_two
3. 11.8% Title_Master
4. 8.21% Embarked_Q
5. 8.14% Age

Feature Importance (after Model Refitting)

Confusion Matrix

Cross-Validation Data
(cv=5)

TN: 412/712	FP: 34/712
FN: 95/712	TP: 171/712

Overall Error rate: 18.12%
Precision: 83.41%
Recall : 64.29%
F1 Score : 72.61%

Test Data
(Kaggle Accuracy: 0.78947)

TN: 232/418	FP: 28/418
FN: 60/418	TP: 98/418

Overall Error rate: 21.05%
Precision: 77.78%
Recall : 62.03%
F1 Score : 69.01%

Problems and Mistakes

- Filling in missing values on “Age”
  - Transform categorical data into numerical data before feeding into Random Forest Regressor
- Did not remove outliers
  - “Fare” column spread: [0, 512], mean: 32.2, std: 49.7
- Accuracy score stuck
  - Only tried to optimize the Random Forest Model
  - Model Assessment only after Model Selection

Problems and Mistakes

- Filling in missing values on “Age”
  - Transform categorical data into ~~numerical data~~ indicator data before feeding into Random Forest Regressor
- Did not remove outliers
  - “Fare” column spread: [0, 512], mean: 32.2, std: 49.7

Mean + 10 × Std

- Accuracy score stuck
  - Only tried to optimize the ~~Random Forest Model~~ all models
  - Model Assessment only after Model Selection

before and

Q and A