Titanic

Team Members:

Tigmanshu Chaudhary(TIC48)
Charu Sreedharan(CHS263)
Varun Nair(VAN17)
Anirban Sen(ANS331)

Problem Statement

- Kaggle competition
- **Objective**: To predict the survival rate of passengers on the Titanic employing machine learning techniques.
- Programming Language used: Python

Dataset

- 891 records in training set.
- 418 records in test set.
- Target variable - Survived
- 11 attributes in total.

Attributes

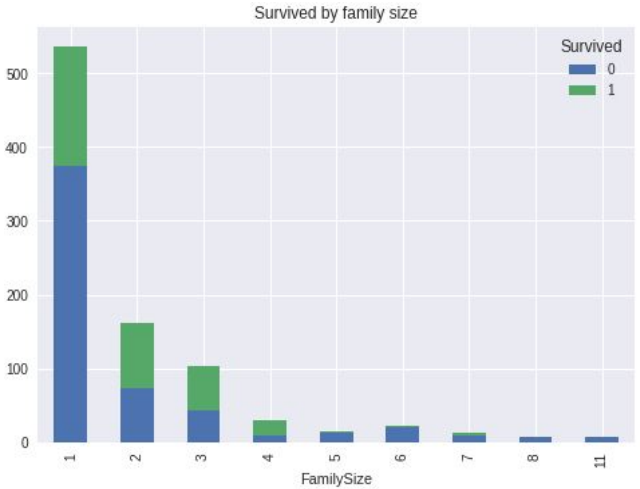
- PassengerId→this is the rowID of the passenger details.
- Pclass→indicates the economic class or status of the passenger(where 1 refers to first class, 2 refers to second class and so on).
- Name→this is the name of the passenger.
- Sex→ this is the gender of the passenger.
- Age→Age of the passenger.
- SibSp→this indicates if there is a sibling or spouse for a passenger.
- Parch→indicates number of parents plus number of children.
- Ticket→this is the serial number of the ticket.
- Fare→indicates the value of the ticket.
- Cabin→indicates the cabin number of the passenger.
- Embarked→indicates the port from which the passenger embarked
  C(Cherbourg),S(Southampton),Q(Queenstown)

Pre-processing

PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

- Columns ‘Age’, ‘Cabin’ & ‘Embarked’ have null values.
- Replaced null values of ‘Age’ with mean of ‘Age’ column.
- Replaced null values of ‘Fare’ (in test set) with mean of ‘Fare’ column.
- Omitted following features for model fitting:
  - Cabin - Cabin is correlated with pclass/fare to have any impact. Also many null values.
  - Embarked - makes no sense of how it could impact survivability.
  - Ticket - makes no sense of how it could impact survivability.
  - Name - felt it is not important in prediction
- Label encode ‘Sex’ into 0 and 1 for males and females.

Feature Engineering



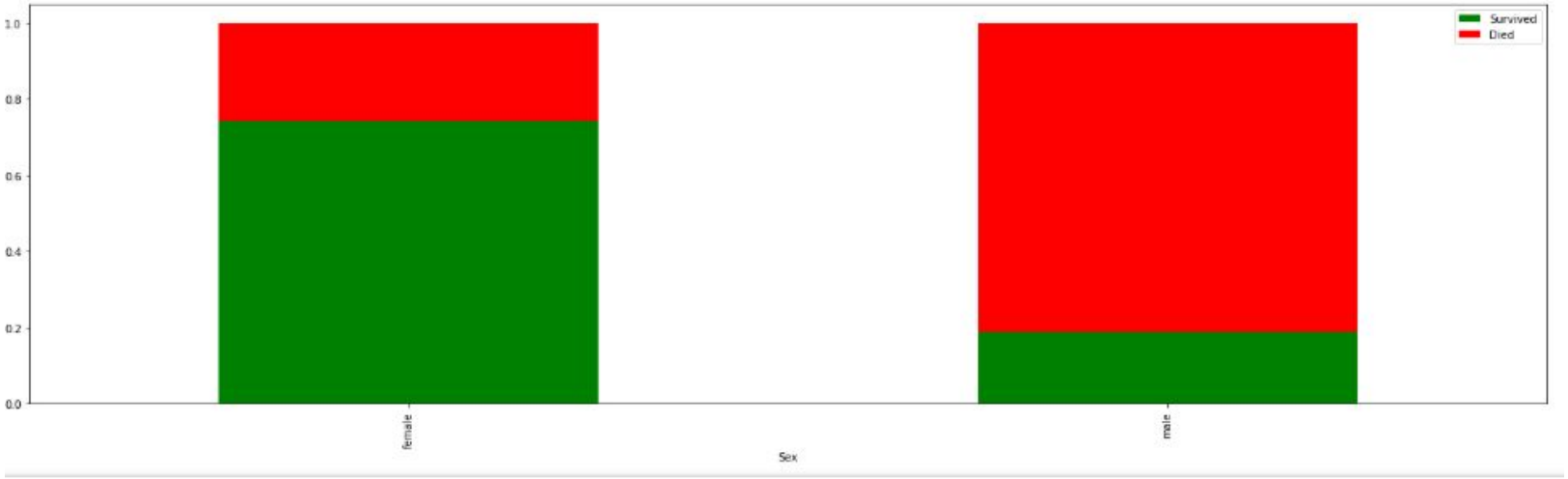
- As can be seen, lower family size(upto 4) leads to higher survival rate.
- Created new feature: ‘FamilySize’

  ‘FamilySize’ = 'SibSp' + 'Parch' + 1

- Created new Feature: ‘Title’. Extracted from name.

  "Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Respectable Titles": 5

Contd.



Sex vs Survival

Contd.

- Features for model fitting: 'Pclass', 'Sex', 'Age', 'SibSp', 'PassengerId','Parch','Fare','FamilySize','Title'

- Split data into training and validation sets.

- Scale training data using Standard Scaler: ensures equal weight for features.

Model Fitting

Model Name	Accuracy on Validation set
KNN Classification	0.758
Logistic Regression	0.815
Linear Discriminant Analysis	0.780
Quadratic Discriminant Analysis	0.741

- Accuracy on Test set : below 0.78 for these models

Contd.

Model Name	Accuracy on Validation set	Accuracy on Test set
Adaptive Boosting	0.79	0.78
SVM	0.816	0.794
Random Forest	0.845	0.799

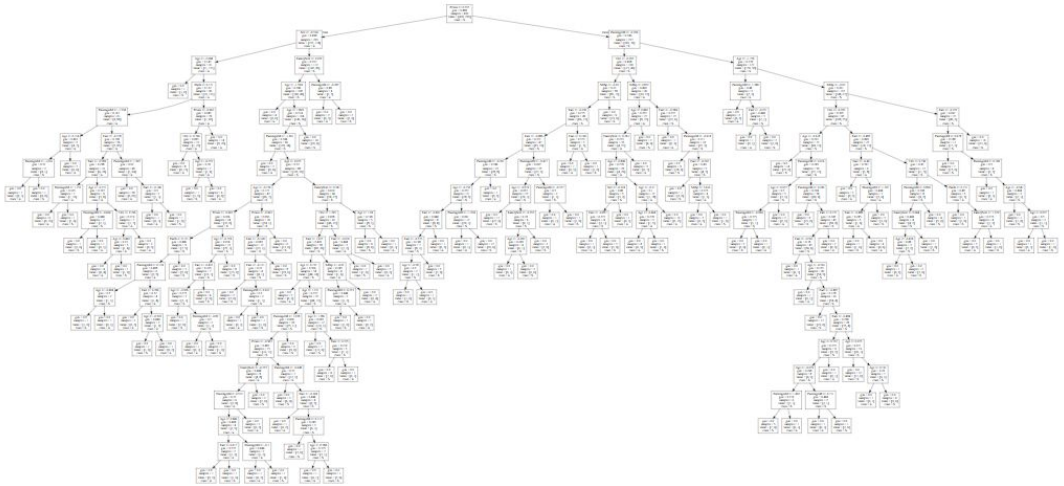
Random Forest Model

```
parameter_grid = {
                  'max_depth' : [4, 6, 8],
                  'n_estimators': [50, 10],
                  'max_features': ['sqrt', 'auto', 'log2'],
                  'min_samples_split': [2, 3, 10],
                  'min_samples_leaf': [1, 3, 10],
                  'bootstrap': [True, False],
                 }
forest = RandomForestClassifier()
cross_validation = StratifiedKFold(n_splits=5)

random_forest = GridSearchCV(forest,
                   scoring='accuracy',
                   param_grid=parameter_grid,
                   cv=cross_validation,
                   verbose=1
                  )


rf = random_forest.fit(X_trainval_transformed, Y_trainval)
Y_pred_rf = random_forest.predict(X_testset_transformed)
```

- Ensemble bagging decision tree algorithm.
  Parameters:
  n_estimators : The number of trees in the forest,
  max_depth : The maximum depth of the tree.
  min_samples_split : The minimum number of samples required to split an internal node.
- Used GridSearchCV for exhaustive search over specified parameter values.
- 5-fold cross-validation.

Contd.

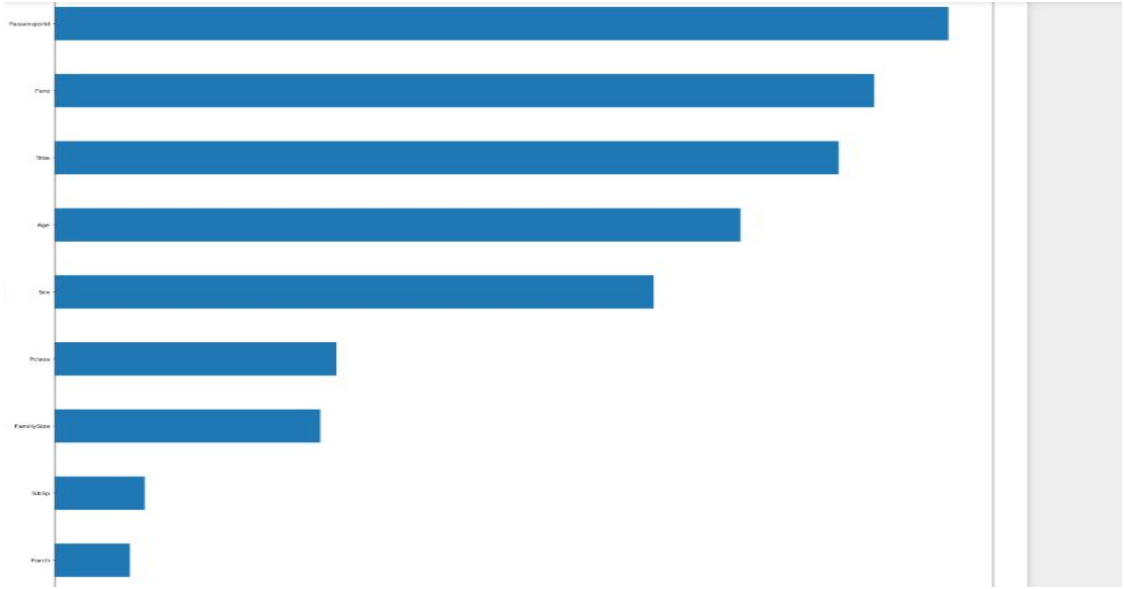
Out[8]:



Random Forest

Decision Tree Visualization

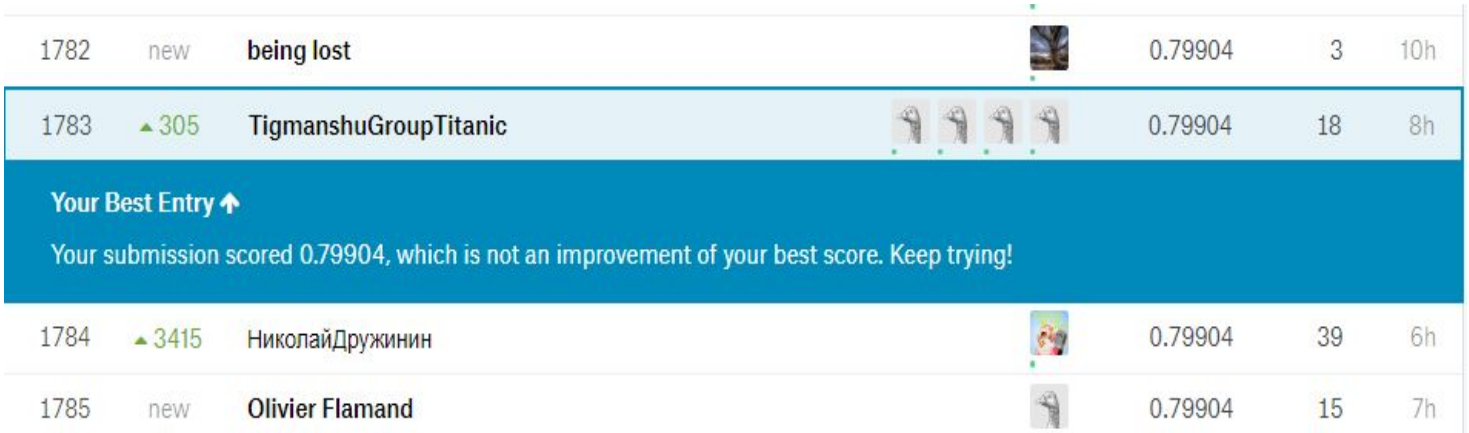
Contd.



Feature importance using Random Forest model

Conclusion and Future Improvements

- Random Forest - most successful model in predicting the survival with an accuracy of 79.9% on the test dataset.



Contd.

- Accuracy maybe low because of multiple null values in Age column.
- Can dig more into data and eventually build new features.
- Try different models like XGBoost or Neural Networks.