Titanic: Machine Learning from Disaster

A Project Report

Submitted in Partial Fulfillment of the

Requirements for The Course

Cs 6375.004 Machine Learning

By

Jaya Padma Sri Maddi (jxm166230)

Jeevan Hunsur Eswara (jhe140030)

Lakshmi Priyanka Parimi (lxp160730)

Sriharshareddy Munjuluru (sxm153630)

April 30, 2017

Introduction and Project Description:

This project aims for a complete analysis of the titanic dataset and predict what sort of passenger 's survived the sinking of the Titanic ship. We used different classifiers for best performing classification through Experimental analysis. We are considering strong classifiers like Bagging Technique, Random Forest, SVM with non-linear kernel and K-NN to classify the dataset and compare the results to determine the best classifier for the Titanic Data Set.

In our project, we considered different techniques to standardize the data set having null values to find out which would be the best standardization for this data set. Some of the techniques considered are replacing the null values in the data set with the median of that column, mean, frequently used values in that column. We performed an in-depth analysis of the data set classification by verifying different parameters before concluding best classifier. Accuracy, F-measure, Precision, Recall are the techniques used to evaluate each classification technique.

The experimental analysis, relationships between different columns, visualizations and the evaluation techniques to determine the best classifiers can be found in the following sections.

Dataset Description:

Name: Titanic: Machine Learning from Disaster

Link: https://www.kaggle.com/c/titanic

Number of instances: 891

Number of features: 11

List of Attributes:

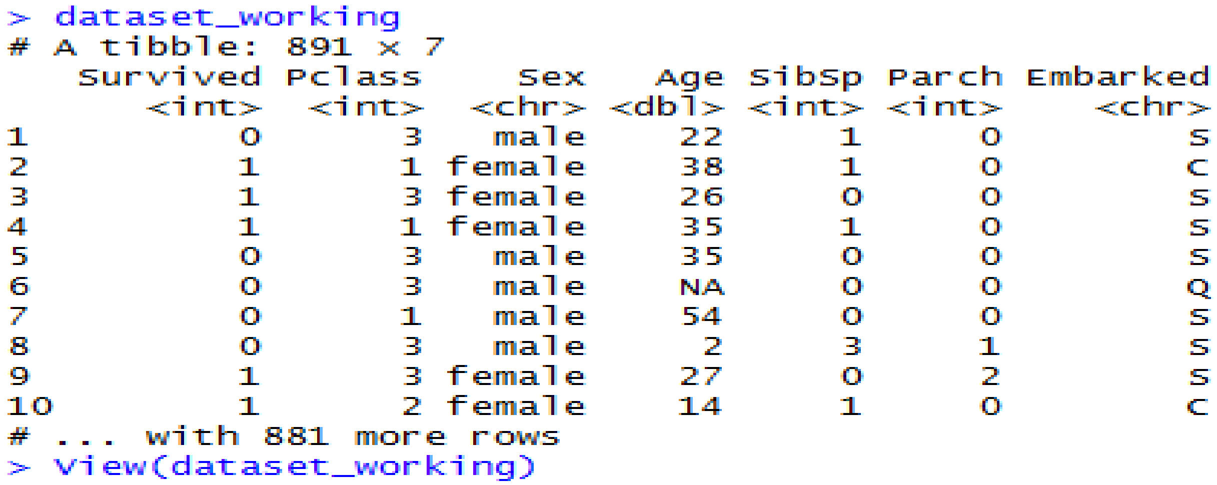
- Survival – 0 if not survived and 1 if survived
- Pclass – Passenger Class (1 - Upper, 2 - Middle and 3 – Lower class)
- Name – Name of the passenger
- Sex – Gender of the passenger
- Age – Age of the passenger
- Sibsp – Number of siblings/spouses aboard the titanic
- Parch – Number of Parents/Children aboard the titanic

  Parent = mother, father

  Child = daughter, son, stepdaughter, stepson

  Some children travelled only with a nanny, therefore parch=0 for them.
- Ticket – Ticket Number
- Fare – Ticket fare
- Cabin – Cabin number
- Embarked – Port of Embarkation (C, Q and S)

Snapshot of data:



```
> dataset_working
# A tibble: 891 x 7
    Survived Pclass    Sex   Age SibSp Parch Embarked
       <int>  <int>  <chr> <dbl> <int> <int>    <chr>
1          0      3   male    22     1     0        S
2          1      1 female    38     1     0        C
3          1      3 female    26     0     0        S
4          1      1 female    35     1     0        S
5          0      3   male    35     0     0        S
6          0      3   male    NA     0     0        Q
7          0      1   male    54     0     0        S
8          0      3   male     2     3     1        S
9          1      3 female    27     0     2        S
10         1      2 female    14     1     0        C
# ... with 881 more rows
> View(dataset_working)
```

Experimental Methodology:

Our approach for implementing this project is as follows:

1. Preprocessing the dataset:
    - This step involves removing null and missing values.
    - Converting categorical data to numeric data.
2. On the dataset:
    - Implement all the above-mentioned classifiers on data.
    - Also, find the best set of parameters for which the techniques performed best.
3. Evaluation the techniques:
    - The techniques are evaluated using Accuracy and F-measure metrics.
4. Plotting results for evaluation using ROC curve.

Dataset Preprocessing:

Real world data are mostly [1] Incomplete, Noisy and inconsistent. To get cleaned and consistent data we formed following data cleaning and transformation steps:

1. Age feature: This feature had quite more NAN/missing values. We handled this by replacing missing values with most frequent of rest of ages. For experiments, we also tested with replacing missing values by mean or median all occurring ages.
2. Embarked feature: There was relative small amount say 2 NAN values. So, went with dropping values of that row. This feature had categorical variables with values C, Q and S. Transformed the values to 0, 1 and 2 respectively.
3. Sex feature: This feature had categories values: male and female. The values are transformed into values 0 and 1.

4. Dropping features: The following features: Cabin, Name, Ticket and fair did not have any co relation in determining survival (class label). So, these columns were dropped.

By end of preprocessing steps, we rectified missing values and extracted only relevant data. So, final set of features used for building classifier model are as follows: Pclass, Sex, Age, Sibsp, Parch, Embarked and Survival (class label).

Data Visualization:

Data visualization done the preprocessed data. Graphical plots give visual understanding of attributes correlation and data distribution. Following python components are used to plot the same: seaborn, matplotlib and biokit.viz

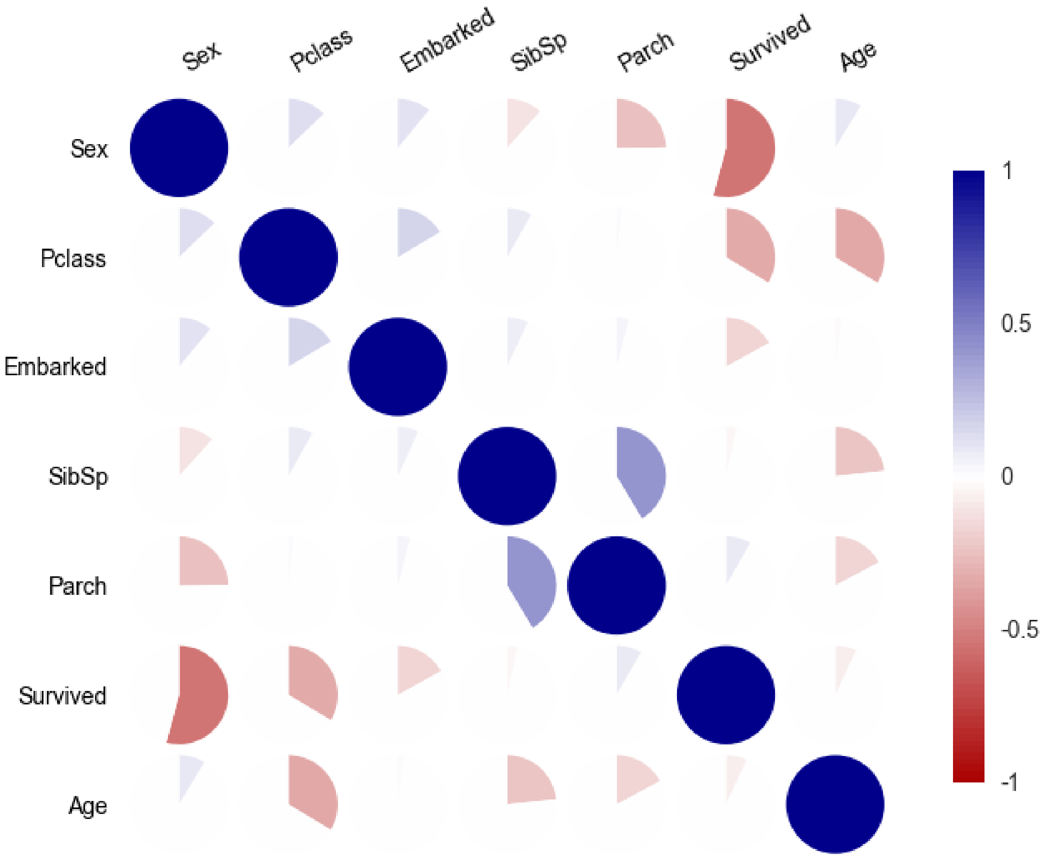


Fig: 1 Correlation between the features and class label

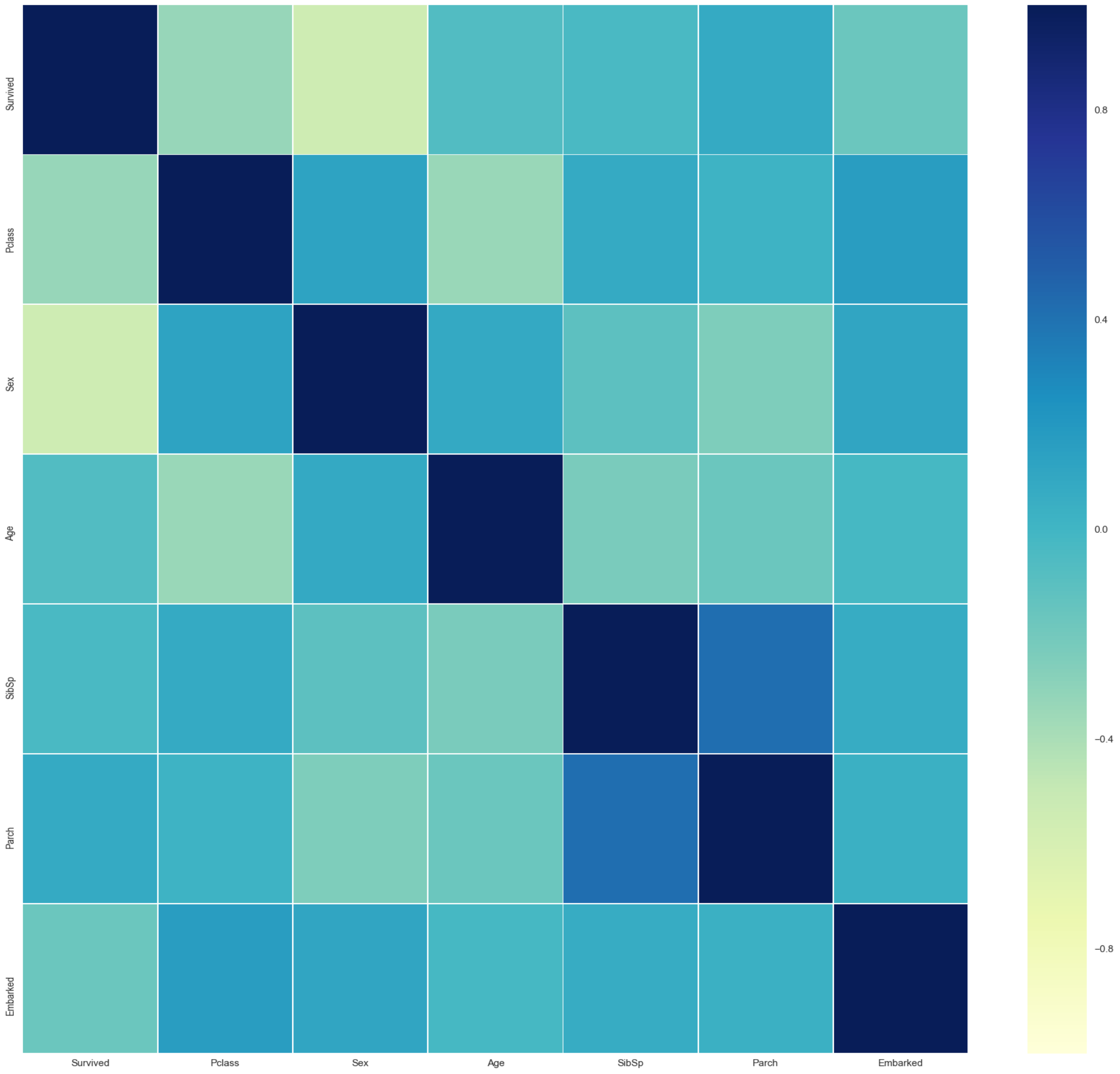


Fig: 2 Correlation Heatmap

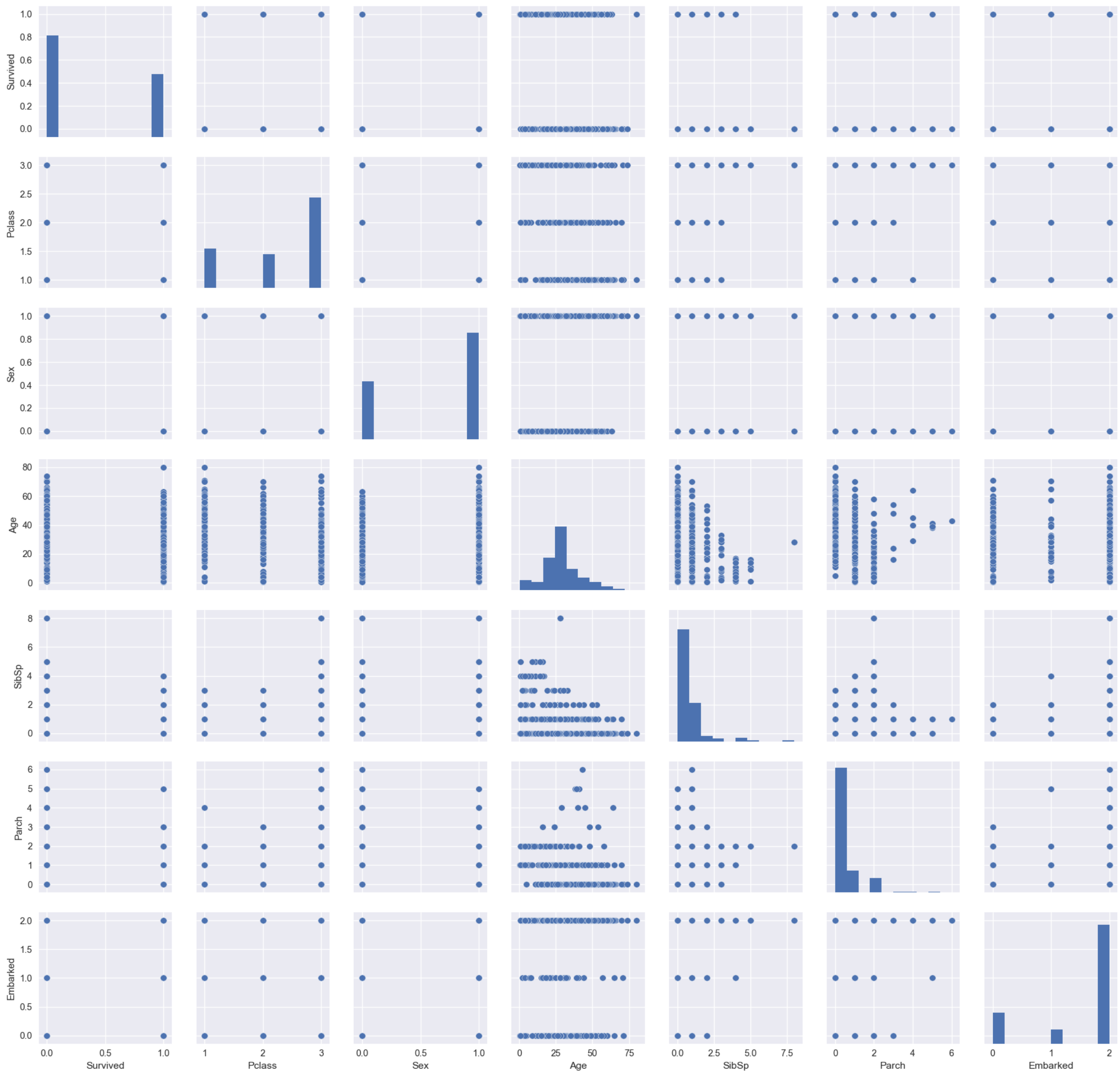


Fig: 3 Data distribution

Proposed solution and methods:

In this project, we started with preprocessing out data set with different techniques to standardize the columns having null data. We ran all the classifiers and all the proposed data sets and then found the best method to train. In the project, we considered Bagging, Random Forest, SVM non-linear kernel, K-NN to find the best classifier on this data set. K-Fold cross validation is used to find the best model on the training data set. Evaluation metrics like Accuracy, F- Measure, Precision and Recall are considered.

Our project concludes by determine the best model with experimental analysis on different parameters and attributes and some of the best accuracies are included as part of the report to understand our conclusion on the classifiers.

Experimental results and analysis:

Bagging:

In our project as mentioned in the Pre-processing section above we considered the following datasets. We tried to apply different attributes and conclude which are the best set of attributes.

DataSets	Parameter1	Parameter2	Parameter3	Parameter 4
preprocessed_data_age_median.csv	78.9413	64.9663	68.56	79.05979
preprocessed_data_age_frequent_fair.csv	78.08449	61.9556	66.5932	77.13746
preprocessed_data_age_frequent.csv	72.1779	53.58	62.99854	72.0843
preprocessed_data_age_mean.csv	76.52956	62.4632	65.73254	77.25943

Legend:	
Parameter1	Age+Sex+Pclass
Parameter2	Age+Parch+SibSp
Parameter3	Age+Embarked+Pclass
Parameter4	Age+Embarked+Sex

We also used 4 attributes to classify and got the following results:

1) bagging(Survived~ Age+Sex+Pclass+Parch, trainData, mfinal= 3, boos = TRUE,rpart.control(maxdepth=10,minsplit=15));

Accuracy of Bagging: 77.14821

2) Age+Parch+Sex+SibSp

Accuracy of Bagging: 77.28

Since they are significantly less we considered to settle with 3 attributes.

By the above step we could identify the data set which replaces the missed values of sex with the median value (preprocessed_data_age_median.csv) is the best data set to consider for further evaluation.

We decided to use Age+ PClass+ Sex as our attributes for further experiments as it has provided consistent results in all the 4 data sets as observed above. Our experimental results after considering the best data set is as follows:

Parameters Considered	**Accuracy**	**Precision**	**Recall**	**F-measure**
bagging(Survived~ Age+Sex+Pclass, trainData, mfinal= 3, boos = TRUE,rpart.control(maxdepth=7)):	76.5246	0.952381	0.2903226	0.6779661
bagging(Survived~ Age+Sex+Pclass, trainData, mfinal= 3, boos = FALSE,rpart.control(maxdepth=7));	76.57894	0.733333	0.1875	0.7213115
bagging(Survived~ Age+Sex+Pclass, trainData, mfinal= 3, boos = FALSE,rpart.control(maxdepth=10));	78.04499	0.903226	0.3023256	0.7777778
bagging(Survived~ Age+Sex+Pclass, trainData, mfinal= 3, boos = TRUE,rpart.control(maxdepth=10,minsplit=15));	**79.39623**	0.88	0.2368421	0.7857143
bagging(Survived~ Age+Sex+Pclass, trainData, mfinal= 3, boos = TRUE,rpart.control(maxdepth=10,minsplit=10));	77.97335	0.894737	0.244898	0.7083333
bagging(Survived~ Age+Sex+Pclass, trainData, mfinal= 3, boos = TRUE,rpart.control(maxdepth=10,minsplit=20));	78.08574	0.888889	0.16	0.7619048
bagging(Survived~ Age+Sex+Pclass, trainData, mfinal= 10, boos = TRUE,rpart.control(maxdepth=10,minsplit=20));	77.52788	0.689655	0.1206897	0.7142857
Increased the Cross Validation to 20	76.19785	0.909091	0.3461538	0.6666667

Analysis:

1) Increasing the cross validation from 10 to 20 has no significant effect over finding the accuracy.
2) Increasing the number of iterations has not increased the accuracy significantly.

3) Also the accuracy of the bagging for the test data set with the best assumed attributes on different parameters is between 75 to 79.39. So the accuracy seems to be consistent across different parameters.
4) We can observe that by increasing the maximum depth we could see increase in accuracy. So, as the number of classifiers increases parallel the accuracy has some positive effect.
5) Defining max depth and min split has increased our accuracy and stands out to be the best parameters among our experiments.

Also, since we are trying to find out classification on data set which has most frequent value in the age experiment results over that data set is as follows:

Iteration#	No of folds (Cross validation)	Parameter considered	Accuracy	Precision	Recall	F-Measure
1	10	Survived~ Age, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	55.08	0.542	0.42	0.53
2	10	Survived~ Sex, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	71.97259	0.718	0.66	0.65
3	10	Survived~ Sex+Age, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	72.64626	0.724	0.72	0.6545
4	10	Survived~ Pclass+Sex, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	74.82	0.734	0.8823	0.6
5	10	Survived~ Pclass+Age, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	62.73	0.6256	0.56	0.44
6	10	Survived~ Embarked+Sex, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	72.167	0.713	0.75	0.688
7	10	Survived~ Pclass+Age+Sex, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	73.07	0.7221	0.92	0.8214
8	10	Survived~ Pclass+Age+Sex, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	75.108	0.738	0.888	0.653

9	**15**	**Survived~ Pclass+Embarked+Sex, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15**	**75.585**	**0.7405**	**0.9**	**0.6206**
10	15	Survived~ Pclass+Embarked+Sex, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10	74.839	0.739	0.888	0.666
11	15	Survived~ Pclass+Embarked+Age+Sex, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	73.86	0.728	0.846	0.687

Analysis:

We are able to determine that Pclass+Embarked+Sex gives the maximum accuracy for the frequent dataset considered in bagging model.

```
Predicted.Class Observed.Class Freq
1       0       0       23
2       1       0       2
3       0       1       8
4       1       1       11
```

The above is the frequency of how many are correctly classified and how many are miss-classified from the test data set. Out of all the test data set results we can see that 10 are miss classified and 34 are correctly classified.

Random Forest Classifier:

Definition:

Random Forest is one of the ensemble methods. The basic idea behind this classifier is to not only create new datasets by varying instances but also attributes. We build Decision tree models from Bootstrap Samples and draw a random sample of attributes to use for splitting criteria at a node in each decision tree.

We used this model to analyze which category of people had higher chances of survival during titanic ship wreck. The given training set has missing values of age. Instead of removing the entire instance from our dataset, we used three different approaches to replace the missing values. In this way, we had many instances to work with, which in turn has higher chances to improve the accuracy of the model. We replaced missing value with median, mean, mode.

Accuracy Analysis by tuning features and replacing missing values of age with mean, median and mode is as follows:

Using Mode Value:

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex : 82.24227

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex+Parch: 81.59069

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex+Parch+SibSp : 81.45393

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass: 70.74089

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex+SibSp: 82.97368

Using Median Value:

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex : 82.1432

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex+Parch : 81.92803

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex+Parch+SibSp : 82.52872

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass: 70.60172

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex+SibSp: 82.7449

Using Mean Value:

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex: 82.14411

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex+Parch : 82.02574

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex+Parch+SibSp : 81.82621

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass: 70.15839

Accuracy of Random Forest with Survived~ Age+Embarked+Pclass+Sex+SibSp: 82.24689

According to the above-mentioned results, we have replaced missing values of age with the most occurring value of age in the dataset.

Parameter tuning:

The following are the parameters that are varied during the experimental analysis to increase accuracy:

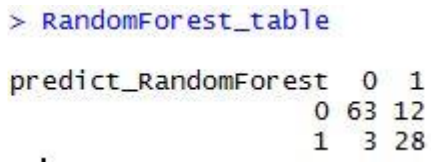
- ***ntree***: The number of trees to grow during classification. This value shouldn't either be too small or too big.
- ***mtry***: number of sampled predictors for splitting at each node.
- ***Importance***: This parameter can either be true or false. If true, the value of predictors is assessed and considered.

- ***Replace***: This parameter can either be true or false. If true, sampling of cases is done with replacement.
- ***Sampsize***: Size of sample to draw.

Results obtained by tuning different parameters are as follows:

Iteration#	Importance	Ntree	mtry	Samp Size	replace	Accuracy	Precision	Recall	FMeasure
1	TRUE	400	4	600	FALSE	81.014	0.727	0.771	0.748
2	TRUE	400	4	600	TRUE	81.20	0.721	0.783	0.742
3	FALSE	400	4	600	TRUE	81.010	0.713	0.769	0.740
4	FALSE	400	4	600	FALSE	80.479	0.733	0.746	0.739
5	**TRUE**	**600**	**4**	**600**	**TRUE**	**82.216**	**0.739**	**0.785**	**0.761**
6	TRUE	900	4	600	TRUE	81.661	0.722	0.785	0.752
7	TRUE	300	4	600	TRUE	82.09	0.714	0.801	0.755
8	TRUE	100	4	600	TRUE	81.009	0.714	0.772	0.742
9	TRUE	600	3	600	TRUE	80.97	0.683	0.787	0.73
10	TRUE	600	5	600	TRUE	80.132	0.733	0.752	0.743
11	TRUE	600	4	700	TRUE	80.747	0.732	0.761	0.7464
12	TRUE	600	4	500	TRUE	81.599	0.706	0.797	0.748
13	FALSE	600	5	400	FALSE	82.045	0.751	0.766	0.759
14	TRUE	700	4	300	TRUE	81.64	0.699	0.799	0.743
15	TRUE	600	1	100	TRUE	81.227	0.615	0.846	0.712

We took five parameters into consideration. We tried to keep four parameters constant initially and vary the fifth one to know how it's value effects the accuracy. We found out the best value for that parameter by looking at accuracies. For that value, we tried to change another parameter and see the effect. By tuning parameter values in this way, we concluded that the best accuracy is obtained when importance is set to TRUE, ntree set to 600, mtry set to 4, sampsize set to 600 and replace set to TRUE.



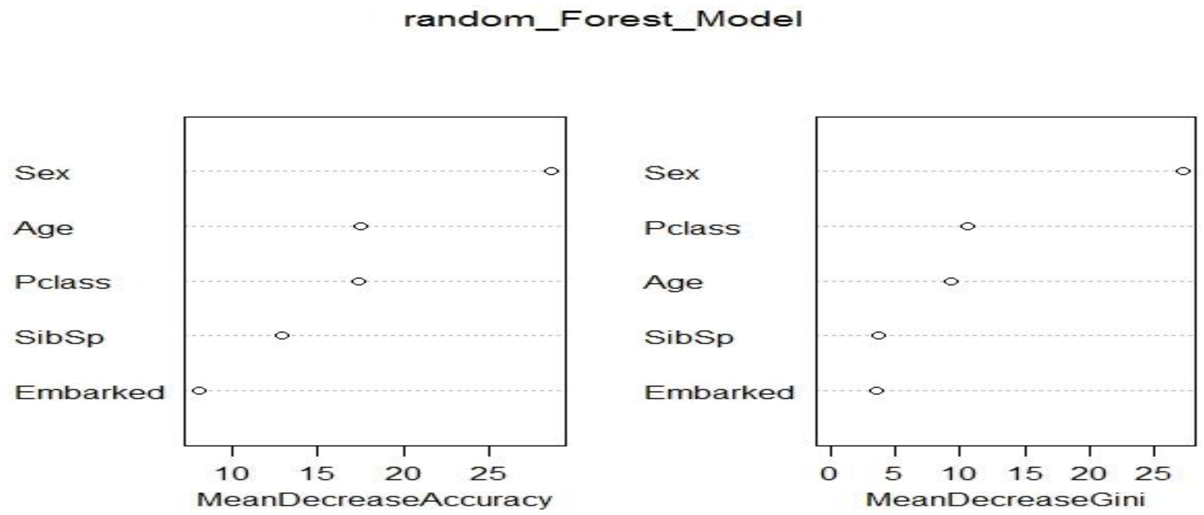
```
> RandomForest_table

predict_RandomForest  0  1
                   0 63 12
                   1  3 28
```

We now analyze the predicted values of classifier on test dataset. We can see that 28 people from the given test sample survived the titanic ship wreck.

Importance of a feature:

>varImpPlot(random_Forest_Model)



Order of Importance of attributes per the above plot is as follows:

- Sex
- Pclass
- Age
- Sibsp
- Embarked

KNN Classifier:

In KNN Classifier, we considered the K-nearest training instances to classify people who survived the titanic ship wreck. We tried to find out the accuracies by varying the parameters. Also various standardization techniques for missing data values are considered.

These are the various formats for which we extracted the values. The accuracies of all the models is as below:

Using Mode Value:

Accuracy of KNearest Neighbor with Survived~ Pclass : 67.59

Accuracy of KNearest Neighbor with Survived~ Sex+Pclass: 77.544

Accuracy of KNearest Neighbor with Survived~ Pclass+Sex+Age: 80.287

Accuracy of KNearest Neighbor with Survived~ Pclass+Sex+Age+SibSp: 81.356

Accuracy of KNearest Neighbor with Survived~ Age+Parch+SibSp+Pclass+Sex : 80.381

Accuracy of KNearest Neighbor with Survived~ Age+Embarked+Pclass+Sex+Parch+SibSp: 80.468

Using median value:

Accuracy of KNearest Neighbor with Survived~ Pclass : 68.197

Accuracy of KNearest Neighbor with Survived~ Sex+Pclass: 76.058

Accuracy of KNearest Neighbor with Survived~ Pclass+Sex+Age: 78.786

Accuracy of KNearest Neighbor with Survived~ Pclass+Sex+Age+SibSp: 81.043

Accuracy of KNearest Neighbor with Survived~ Age+Parch+SibSp+Pclass+Sex : 79.986

Accuracy of KNearest Neighbor with Survived~ Age+Embarked+Pclass+Sex+Parch+SibSp: 79.284

Using Mean Value:

Accuracy of KNearest Neighbor with Survived~ Pclass : 67.813

Accuracy of KNearest Neighbor with Survived~ Sex+Pclass: 77.530

Accuracy of KNearest Neighbor with Survived~ Pclass+Sex+Age: 80.351

Accuracy of KNearest Neighbor with Survived~ Pclass+Sex+Age+SibSp: 80.598

Accuracy of KNearest Neighbor with Survived~ Age+Parch+SibSp+Pclass+Sex : 80.1469

Accuracy of KNearest Neighbor with Survived~ Age+Embarked+Pclass+Sex+Parch+SibSp: 78.35

Based on the above model we can clearly see that preprocessed frequent age is giving the highest accuracies with the following attributes:

Pclass+Sex+Age+SibSp: 81.356

We considered various parameters for the above derived attributes. The various parameters considered and the output of them is as follows:

Iteration#	No of folds (Cross validation)	K Value	Accuracy	Precision	Recall	F-Measure
1	10	5	81.494	0.727	0.776	0.751
2	10	10	81.235	0.697	0.796	0.744
3	10	8	81.145	0.697	0.784	0.738
4	10	20	80.209	0.634	0.804	0.709

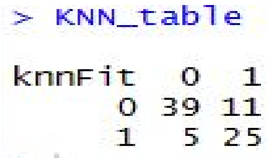
5	10	25	81.191	0.658	0.822	0.731
6	**10**	**3**	**82.618**	**0.734**	**0.790**	**0.761**
7	15	3	81.217	0.727	0.770	0.748
8	15	5	81.204	0.729	0.773	0.750
9	20	5	81.125	0.726	0.774	0.749
10	20	20	79.117	0.623	0.777	0.691

Analysis:

1. Increasing the number of folds from 10 – 20 did not have a positive effect on the overall accuracies.

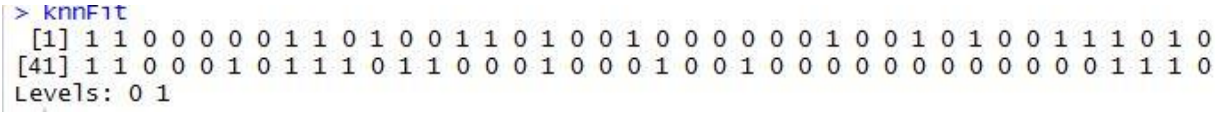
2. We can observe that increase in the K-Value did not have a positive effect on the overall accuracies.

3. We can obtain best accuracies when a k-value of 3 and maximum fold validation of 10 is taken.



```
> KNN_table

knnFit  0  1
     0 39 11
     1  5 25
```



```
> knnFit
 [1] 1 1 0 0 0 0 0 1 1 0 1 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 1 1 0 1 0
[41] 1 1 0 0 0 1 0 1 1 1 0 1 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0
Levels: 0 1
```

By the above we can say that the K-NN classifier is able to classify 25 people correctly who got survived. It was also able to identify that 39 people did not survive correctly.

SVM:

SVM (Support Vector Machines) is one of the powerful classifiers. It works by increasing the data dimensionality using its kernels. For the analysis of survival of the titanic data set one of the classifiers used was SVM.

The given training set has missing values of age. Instead of removing the entire instance from our dataset, we used three different approaches to replace the missing values. In this way, we had many instances to work with, which in turn has higher chances to improve the accuracy of the model. We replaced missing value with median, mean, mode.

Replaced With	**Accuracy**
Most frequent	75.621
Mean	73.287
Median	74.342

So, we have decided to proceed by replacing the missing age values with the most frequent values of the column.

Parameter Tuning:

The parameters used for the analysis were type of the kernel, gamma value and the cost (of regularization).

Kernel - The kernel that is used for training and predicting.

Gamma – It is the inverse of data dimensionality.

Cost – Cost for constraint violation. It is the cost of regularization.

Below is the tabulated result of the analysis:

Kernel	Gamma	Accuracy	Precision	Recall	F-Measure	Cost
Linear	NA	70.305	0.639	0.691	0.664	100
Linear	NA	70.611	0.655	0.704	0.679	70
Linear	NA	70.813	0.652	0.702	0.676	60
Polynomial	0.5	72.234	0.613	0.745	0.673	100
Polynomial	0.33	71.661	0.594	0.737	0.658	100
Polynomial	0.25	72.462	0.609	0.755	0.674	100
Polynomial	0.2	70.214	0.598	0.726	0.656	100
Polynomial	0.1667	71.206	0.604	0.735	0.664	100
Polynomial	0.1428	73.035	0.615	0.765	0.684	100
Polynomial	0.125	73.657	0.612	0.779	0.685	100

Polynomial	0.111	72.329	0.594	0.764	0.668	100
Polynomial	0.1	72.746	0.635	0.768	0.695	100
Sigmoid	0.5	61.025	0.631	0.582	0.605	100
Sigmoid	0.33	62.196	0.588	0.609	0.598	100
Sigmoid	0.25	61.459	0.579	0.589	0.584	100
Sigmoid	0.2	62.880	0.587	0.618	0.602	100
Sigmoid	0.1667	63.624	0.604	0.613	0.608	100
Sigmoid	0.1428	64.357	0.626	0.634	0.630	100
Sigmoid	0.125	64.516	0.615	0.622	0.618	100
Sigmoid	0.111	64.854	0.633	0.635	0.634	100
Sigmoid	0.1	68.456	0.662	0.661	0.662	100
Radial	0.5	70.324	0.589	0.723	0.649	100
Radial	0.33	71.717	0.614	0.736	0.669	100
Radial	0.25	70.480	0.579	0.733	0.647	100
Radial	0.2	73.434	0.623	0.783	0.694	100
Radial	0.1667	72.045	0.606	0.752	0.671	100
Radial	0.1428	73.590	0.626	0.768	0.690	100
Radial	0.125	74.539	0.642	0.783	0.706	100
Radial	0.111	74.322	0.623	0.782	0.693	100
Radial	0.1	74.092	0.635	0.771	0.696	100
Polynomial	0.5	73.188	0.613	0.768	0.688	60
Polynomial	0.25	72.234	0.596	0.759	0.667	70
Polynomial	0.25	73.054	0.602	0.763	0.673	60

Polynomial	0.2	72.800	0.615	0.765	0.682	60
Polynomial	0.111	72.847	0.612	0.764	0.680	60
Polynomial	0.1	73.111	0.632	0.749	0.686	60
Polynomial	0.125	74.185	0.635	0.790	0.704	60
Radial	0.125	74.458	0.649	0.774	0.706	70
Radial	**0.125**	**75.621**	**0.641**	**0.791**	**0.708**	**60**
Radial	0.2	73.095	0.624	0.772	0.690	60
Radial	0.1	74.052	0.627	0.787	0.698	60
Radial	0.111	74.472	0.629	0.796	0.703	60

First the analysis was carried on which kernel to use. Linear kernel was used first and all the variations of cost had given almost same values of the accuracy stating that linear kernel is not the one best suited for classification. Then keeping the gamma values and the cost constant, polynomial sigmoid and radial kernels were tested. From this we arrived at the conclusion that radial kernels are better for classification. After deciding the kernel, keeping the cost constant, the best data dimensionality was arrived. Gamma value of 0.125 came out as the best one. Then the various values of cost were decided in which the value of 60 had come out the best. The best set of parameters and the best metrics obtained have been highlighted in the table.

```
>SVM_table

prediction_SVM      0  1
                  0 24 10
                  1  4 20
```

It has predicted that 20 people have survived the wreck.

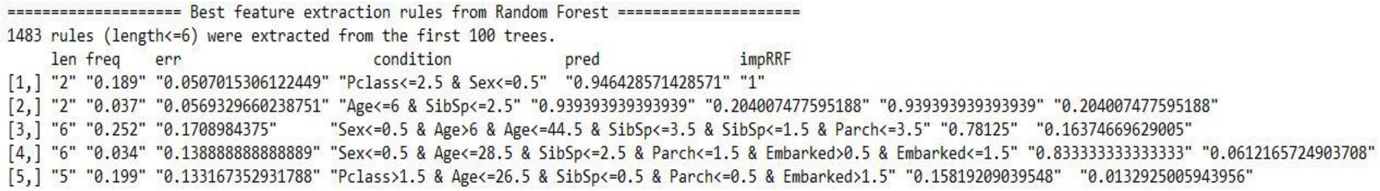
Conclusions:

The best set of parameters and their accuracies for all the considered classifiers are:

Classifier	Parameters	Accuracy	Precision	Recall	F-Measure
Bagging	Survived~ Pclass Embarked+Sex, mfinal = 3, boos= TRUE, rpart.control : maxdepth =10, minsplit = 15	75.585	0.7405	0.9	0.6206
Random Forest	importance=TRUE,Ntree=400,mtry=4,Samp size=600	82.216	0.739	0.785	0.761
K-NN	K=3	82.618	0.734	0.790	0.761
SVM	Kernel=radial, gamma=0.125, cost=60	75.621	0.641	0.791	0.708

First, we have worked on selecting the best classifier that fits the data set. By looking at all the results, it was found that K-NN and random forest have accounted similar accuracies on their best parameters. K-NN is susceptible to noise and so we have concluded to proceed with Random Forest.

After that we have tried it over different rule engines. The best rule over we got the highest accuracy is "Pclass<=2.5 and Sex<=0.5". This rule had 95% accuracy. This rule implies that females of Upper and Middle class are more likely to survive than compared to other passengers.



```
==================== Best feature extraction rules from Random Forest =====================
1483 rules (length<=6) were extracted from the first 100 trees.
     len freq    err                       condition                pred                 impRRF
[1,] "2" "0.189" "0.0507015306122449" "Pclass<=2.5 & Sex<=0.5"  "0.946428571428571" "1"
[2,] "2" "0.037" "0.0569329660238751" "Age<=6 & SibSp<=2.5" "0.939393939393939" "0.204007477595188" "0.939393939393939" "0.204007477595188"
[3,] "6" "0.252" "0.1708984375"      "Sex<=0.5 & Age>6 & Age<=44.5 & SibSp<=3.5 & SibSp<=1.5 & Parch<=3.5" "0.78125"  "0.16374669629005"
[4,] "6" "0.034" "0.138888888888889" "Sex<=0.5 & Age<=28.5 & SibSp<=2.5 & Parch<=1.5 & Embarked>0.5 & Embarked<=1.5" "0.833333333333333" "0.0612165724903708"
[5,] "5" "0.199" "0.133167352931788" "Pclass>1.5 & Age<=26.5 & SibSp<=0.5 & Parch<=0.5 & Embarked>1.5" "0.15819209039548"  "0.0132925005943956"
```

Contribution of Team Members:

Jeevan Hunsur Eswara – Data Pre-processing, data visualization and report writing.

SriHarshaReddy Munjuluru – More report writing, experimental analysis of Bagging techniques.

Jaya Padma Sri Maddi – Experimental analysis of SVM – non-linear kernel, conclusion analysis.

Lakshmi Priyanka Parimi – Experimental analysis of Random Forest and K-NN.

References:

[1] http://www.cs.ccsu.edu/~markov/ccsu_courses/datamining-3.html
[2] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Imputer.html
[3] http://scikit-learn.org/stable/modules/preprocessing.html
[4] http://seaborn.pydata.org/generated/seaborn.heatmap.html
[5] https://pythonhosted.org/biokit/references.html#biokit.viz.corrplot.Corrplot
[6] http://stackoverflow.com/questions/14996619/random-forest-output-interpretation