Machine Learning (CS 6375.004)

FINAL PROJECT REPORT

TOPIC: Machine Learning from Disaster

Team Members

Ishan Desai (ijd140030)

Silpy Jain (sxj152930)

Milavkumar Shah (mss140730)

Abstract

In this project, we see how to use machine-learning techniques to predict survivors of the Titanic. With a dataset of 891 individuals containing features like sex, age, and class, we attempt to predict the survivors of a small test group of 418. In particular, compare different machine learning techniques like Naïve Bayes, SVM, and decision tree analysis.

Introduction

The sinking of Titanic is one of the most infamous disasters that occurred in history. Although the major reason for such loss of life was inadequate number of lifeboats for passengers and crew, but there were some groups with better chances of survival. The task of this project is to predict whether a given passenger survived the sinking of the Titanic based on various attributes including age, gender, ticket, the fare they paid, and other information using tools of machine learning. We will be doing data pre-processing and use different classifiers to predict survival chances. Solutions are evaluated by comparing the percentage of correct answers on a test dataset.

Problem Definition and Algorithm

Task Definition

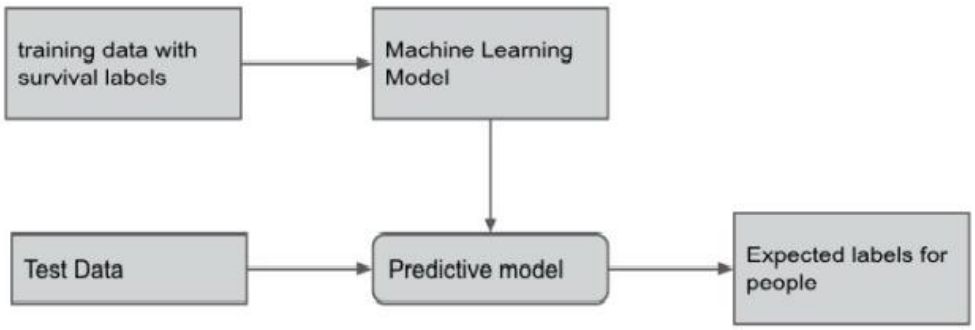
Recently, many studies have been conducted to study this problem in order to compare and contrast the different machine learning techniques. Using data provided by Kaggle, our goal is to apply machine-learning techniques to successfully predict which passengers survived the sinking of the Titanic. We take several approaches to this problem in order to compare and contrast the different machine learning techniques. By looking at the results of each technique we can make some insights about the problem.

Data Analysis

```
> str(train)
'data.frame':   891 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 581 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age        : num  22 38 26 35 35 0 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 62 83 62 57 62 62 131 62 62 62 ...
 $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

The data we used for our project was provided on the Kaggle website. We were given 891 passenger samples for our training set and their associated labels of whether or not the passenger survived. For each passenger, we were given his/her passenger class, name, sex, age, number of siblings/spouses aboard, number of parents/children aboard, ticket number, fare, cabin embarked, and port of embarkation. For the test data, we had 418 samples in the same format. The dataset is not complete for several samples with one or many of fields marked empty or unavailable. However, all sample points contained at least information about gender and passenger class. To normalize the data, we replace missing values with the mean of the remaining data set or other values.

Algorithm Definition



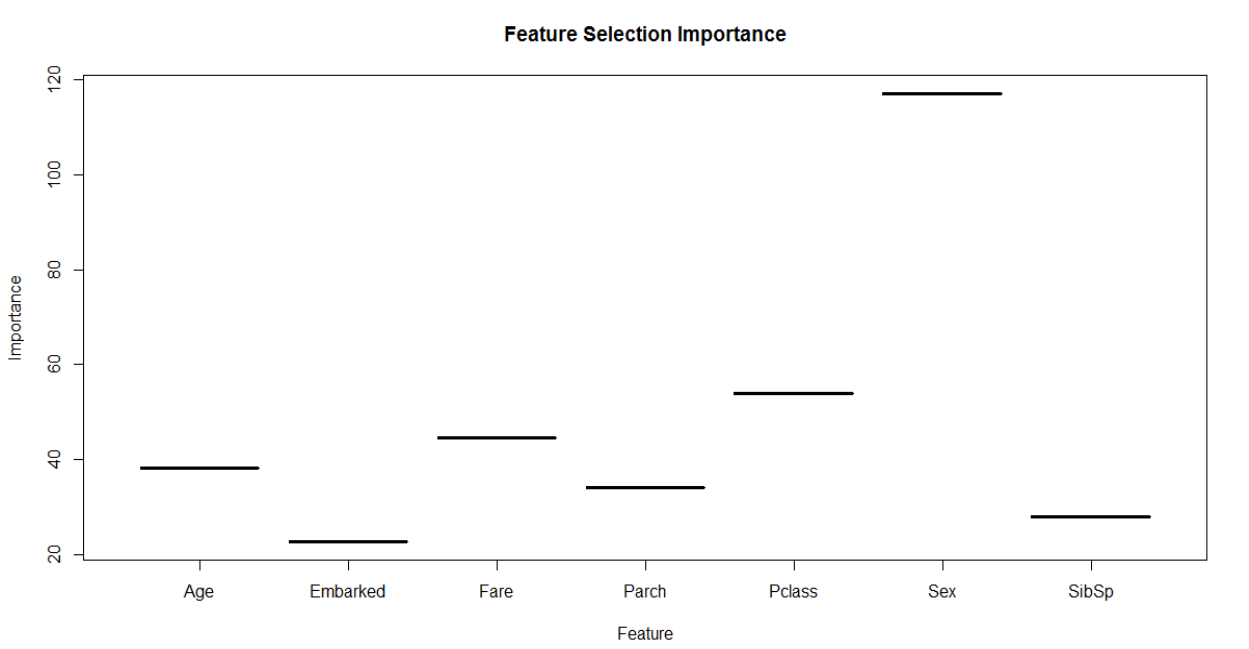
Steps:

1) Pre-processing of dataset: removal of irrelevant columns as Passenger Id and Passenger name and create a subset with the remaining columns.
2) Feature Selection: Extract the important features using random forest.
3) Training model: Create the training model for each of the classifiers.
4) Predict: Using the training model, predict the survival chances for each classifier.
5) Mean Output: Calculate the row mean survival chances for each passenger in test data.
6) Accuracy: Based on the mean output obtained, calculate the accuracy for each classifier and do a comparative analysis.

Experimental Evaluation

Methodology

After a careful analysis of the dataset, we at first replaced all the null values in the dataset with other values. Once we achieved this, then we decided to extract the features relevant for prediction. We used random forest model to evaluate the importance of each attribute.



As a benchmark, we first implemented Decision Tree model based on the important features extracted earlier. For decision tree, the features considered were 1)Sex, 2) Age, 3)Pclass, 4)Fare and 5)Parch. To improve our classification, more sophisticated models like Random Forest, SVM, Neural Net, kNN are applied to the dataset. After this, a comparative analysis for each of the methods used is provided.

Random Forest

A random forest is a classifier consisting of a collection of tree-structured classifiers {h(x, Θk), k = 1, . . .} where the {Θk} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x. A random forest is an ensemble of decision trees, which will output a prediction value, in this case survival. Each decision tree is constructed by using a random subset of the training data. After you have trained your forest, you can then pass each test row through it, in order to output a prediction.

- Approach
  Intuitively, women are more likely getting help when a misery happens. In the Titanic problem, women had much bigger chance to survive than men. From the dataset we got

that classes are not always in direct proportion to fare, and actually first class can be cheaper than third class. For Random Forest , we built the model using randomForest library in R and we used survived column of the dataset as the output factor. The number of trees chosen was limited to 1000 and based on these parameters we predicted the survived column for the test data set.

Support Vector Machine

SVM is a supervised learning model with associated learning algorithms that analyzes data and recognizes patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

- Approach

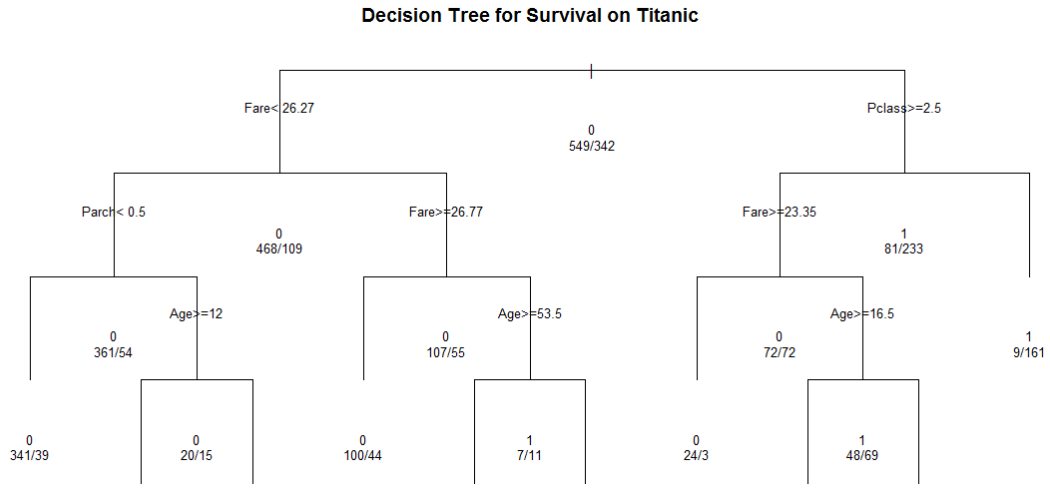
  We considered the following features: 1) passenger class, 2) sex, 3) age, 4) number of siblings, 5) patriarchal status, 6) fare, and 7) place of embarkation. We used a linear classification function as our kernel and set the type to C-classification in the SVM model and set decision values parameter to true in the prediction model to obtain a binary output levels for test data.

Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a test on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represents classification rules. In decision analysis a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values (or expected utility) of competing alternatives are calculated.

- Approach

We built our decision using the following features -- gender, passenger class, age, fare and Parch. We built the model using rpart library with method parameter as class . Using this as the model, we got the following plot for the decision tree.



Neural Net

Artificial neural networks can be applied to approximate any complex functional relationship. Artificial neural networks are applied in many situations. **neuralnet** is built to train multi-layer perceptrons in the context of regression analyses, i.e. to approximate functional relationships between covariates and response variables. Thus, neural networks are used as extensions of generalized linear models. **neuralnet** is a very flexible package. The backpropagation algorithm and three versions of resilient backpropagation are implemented and it provides a custom-choice of activation and error function.

- Approach

We used nnet library to train the neural net model and specified the maximum number of iterations as 1000, the parameter for weight decay factor as 0.001 and switch for tracing optimization as false. Based on these, the training model was built which was then used for classification of test data.

kNN

k nearest neighbor join (kNN join), designed to find k nearest neighbors from a dataset S for every object in another dataset R, is a primitive operation widely adopted by many data mining applications. As a combination of the k nearest neighbor query and the join operation, kNN join is an expensive operation. Given the increasing volume of data, it is difficult to perform a kNN join on a centralized machine efficiently. kNN join is a special type of join that combines each object in a dataset R with the k objects in another dataset S that are closest to it. kNN join typically serves as a primitive operation and is widely used in many data mining and analytic applications, such as the k-means and k-medoids clustering and outlier detection.

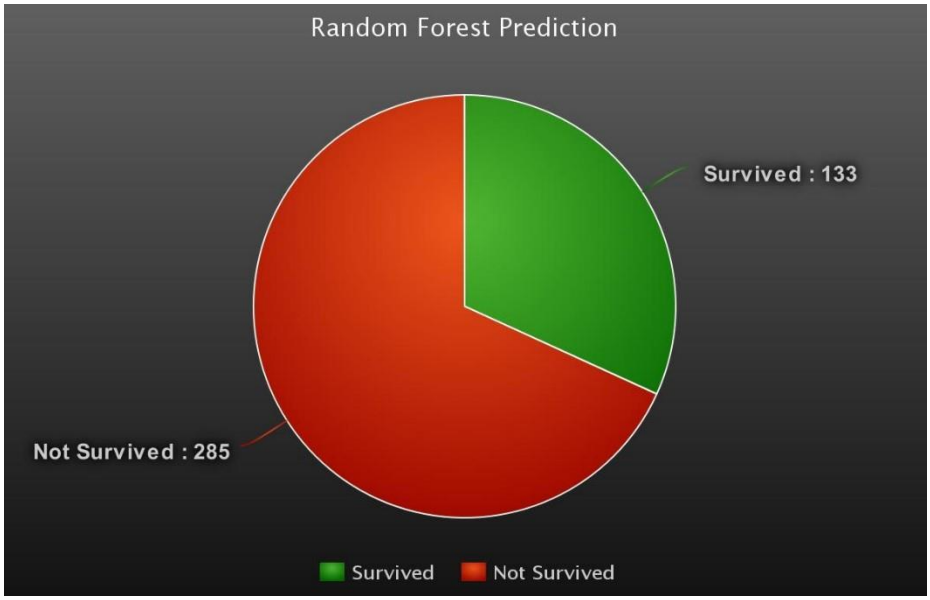
- Approach

We used class library and a 10 fold cross validation to build the kNN training model which is then applied to the test data to predict survival chances.

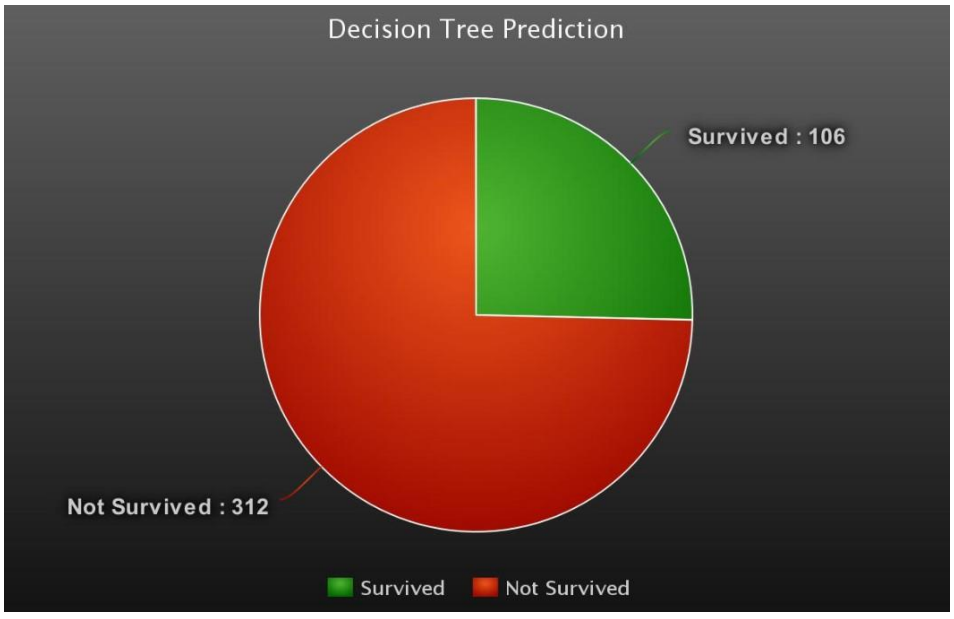
Results

Below are the graphs for the survival chances obtained per classifier and a comparative analysis for each of them.

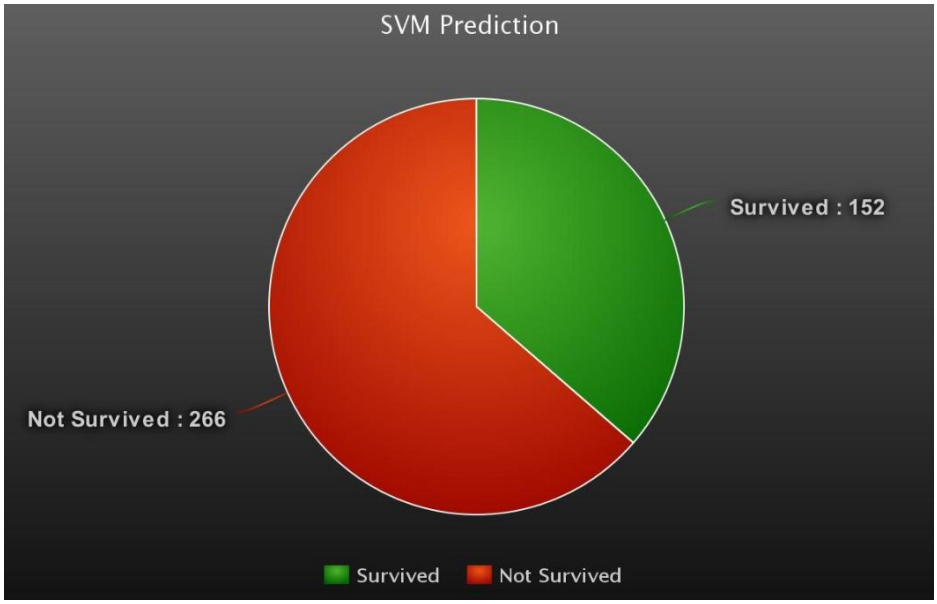
1) Random Forest



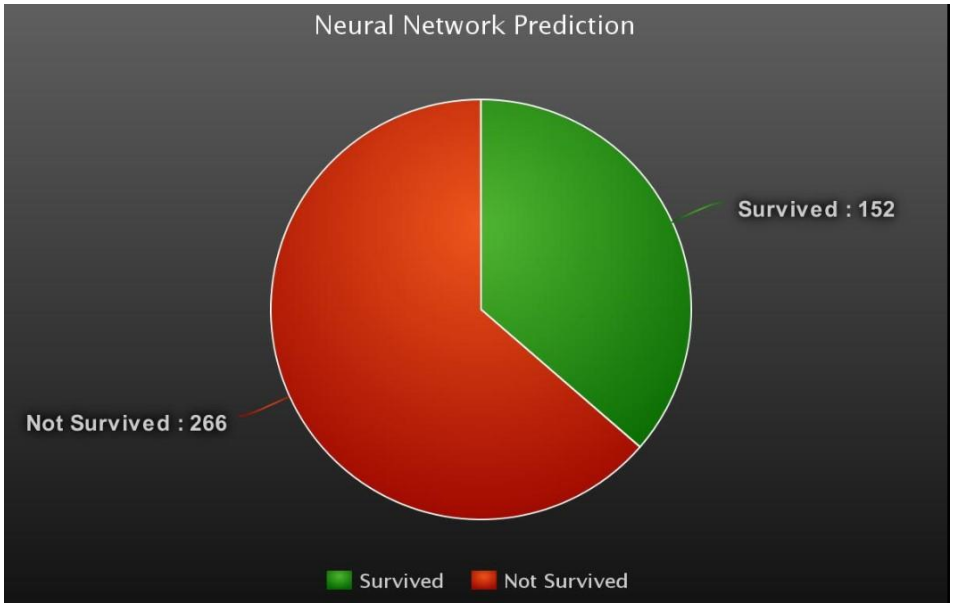
2) Decision tree



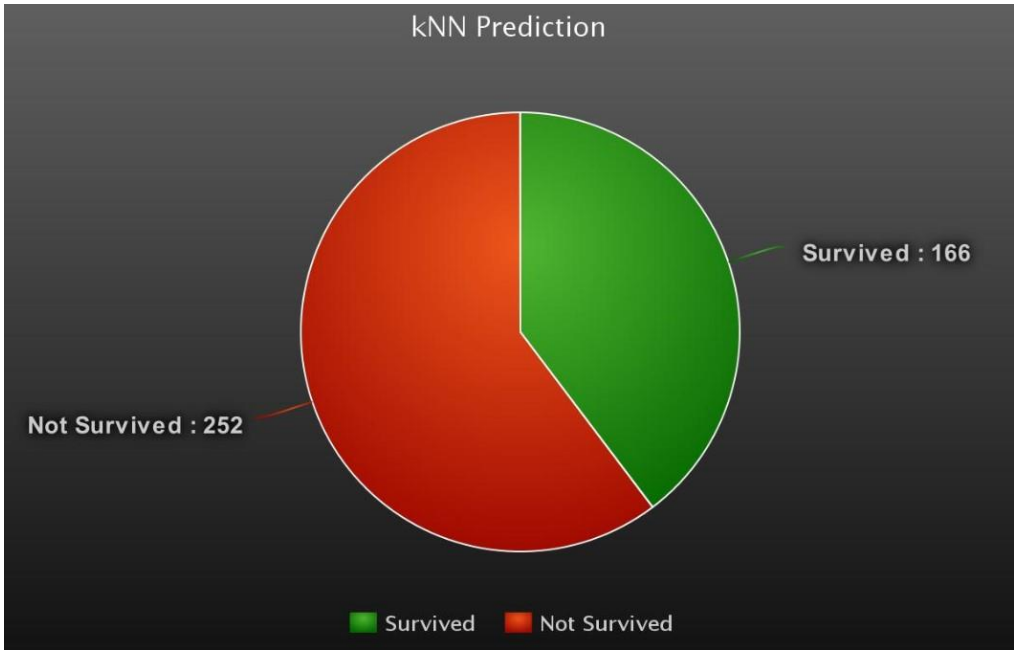
3) SVM



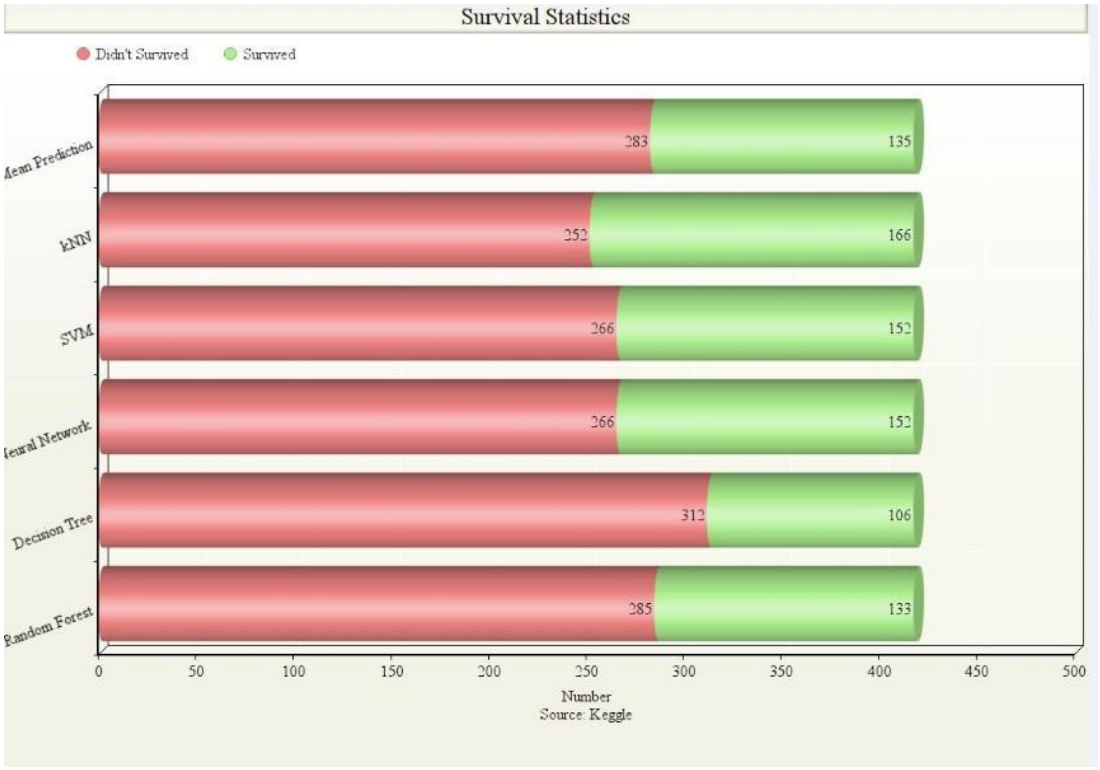
4) Neural Net



5) kNN

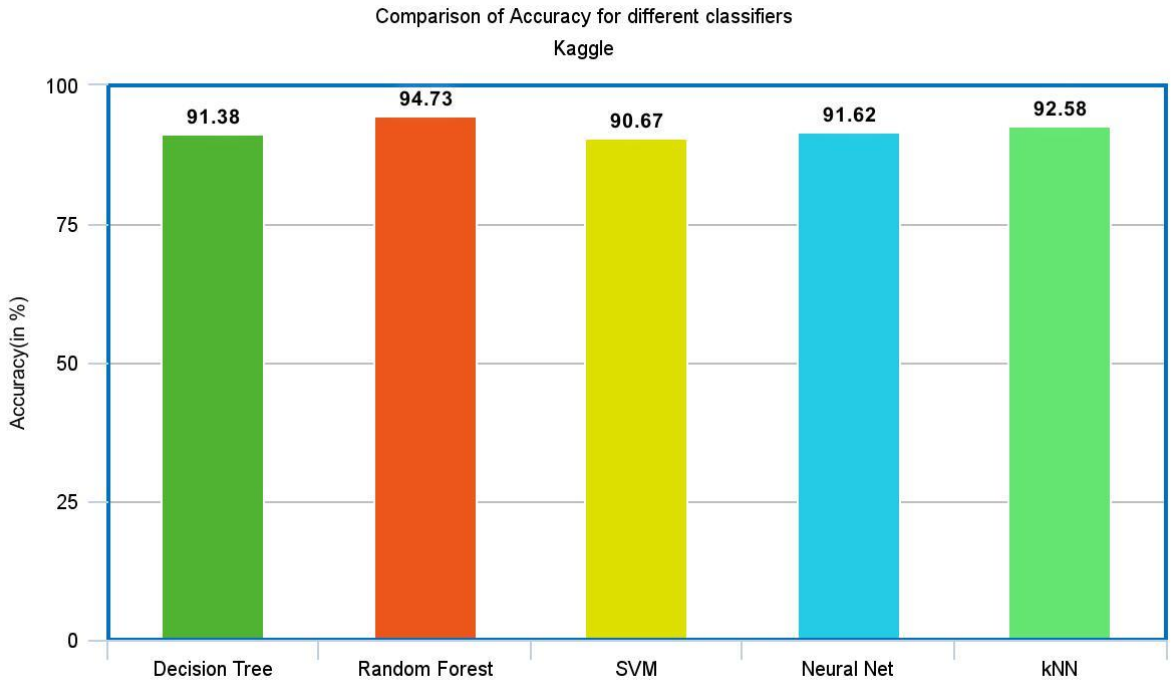


Survival Statistics for all the classifiers and their mean



Accuracy Comparison for all the classifiers based on mean output

S.No.	Classifiers	Accuracy(in %)
1.	Decision Tree	91.38
2.	Random Forest	94.73
3.	State Vector Machine(SVM)	90.67
4.	Neural Net	91.62
5.	K Nearest Neighbours(kNN)	92.58



Conclusion

According to our research, our results can be obtained with a higher accuracy, which improves 2% compared to the results of other models. Besides, during the process of our study, we find more features utilized in the models do not necessarily make better result.

As observed from the analysis, we can conclude that Random Forest gives us better accuracy as compared to other classifiers considered.

References

- https://www.kaggle.com/c/titanic
- http://cs229.stanford.edu/proj2012/LamTangTitanicMachineLearningFromDisaster.pdf
- A. Ng. CS229 Notes. Stanford University, 2012
- Breiman, L. 2001a. Random forests. Machine Learning 45:5-32.
- http://en.wikipedia.org/wiki/Support_vector_machine