Project 3

Submitted in requirement for CS 6923

Machine learning online

Introduction

The project aims to accomplish the prediction of survivors on the titanic. The dataset used is from the kaggle competition Titanic: Machine learning from disaster. The loss of life on the titanic was particularly due to the lack of lifeboats. Some of the groups were more likely to survive like children, women and upper class. The challenge is to find which passengers did survive the tragedy. I will try two methods, Decision Trees and K-Nearest Neighbour to predict survival rate

Data Set

The dataset is available as the train.csv file at https://www.kaggle.com/c/titanic/data.

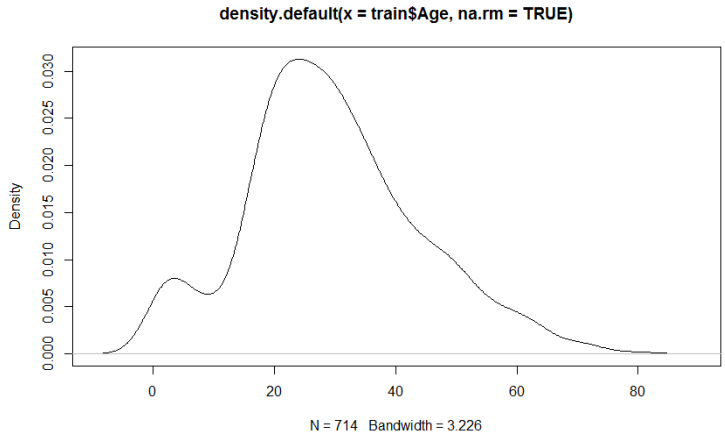
	A	B	C	D	E	F	G	H	I	J	K	L
1	Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, M	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, M	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen,	female	26	0	0	STON/O2.	7.925		S
5	4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. '	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy,	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, M	male	2	3	1	349909	21.075		S
10	9	1	3	Johnson, M	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, Mr	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, M	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saunderco	male	20	0	0	A/5. 2151	8.05		S
15	14	0	3	Andersson	male	39	1	5	347082	31.275		S
16	15	0	3	Vestrom, M	female	14	0	0	350406	7.8542		S
17	16	1	2	Hewlett, M	female	55	0	0	248706	16		S

We will be using Sex,Age, Pclass,Sibsp,Parch,Embarked to find survival

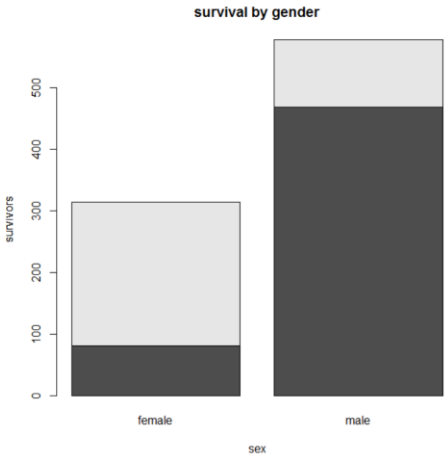
To better the algorithms used in project 1 I have chosen more variables to supply to the model

Cleaning the dataset

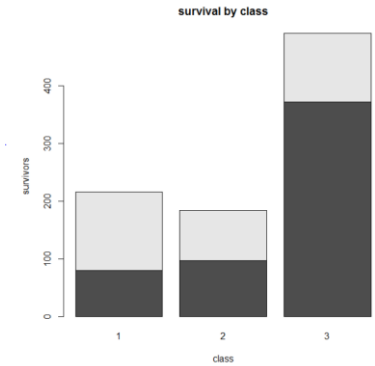
We check for the plots of sex,age ,pclass,SibSp and parch to get an idea of how they will affect the decision tree



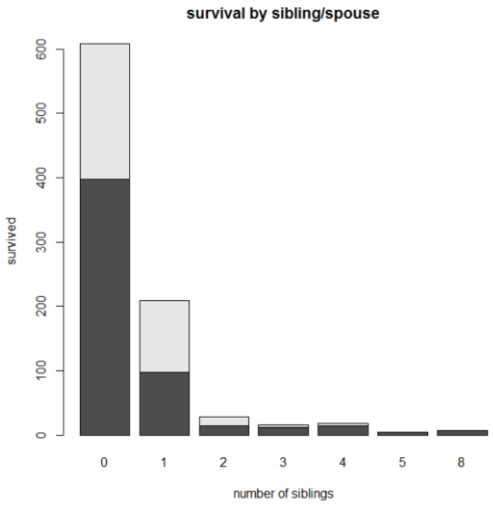
We see that the max density is near 27 years of age



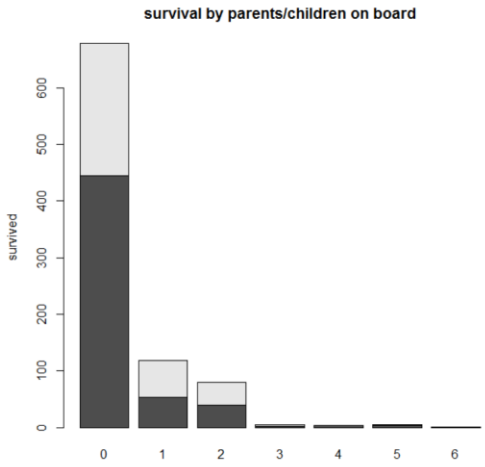
The lighter areas indicate survival. As we had expected the survival rate for women is much higher than men, which will be a key node in the decision tree



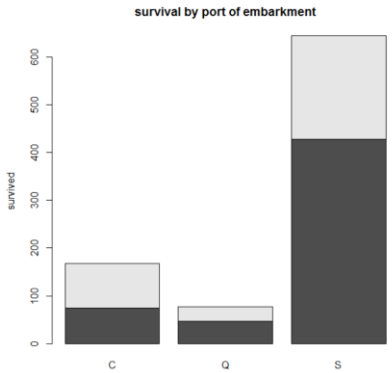
The survival for class 2 and 3 is much more indicating that higher classes were given preference.



We see that for higher number of siblings the survival rate is very low. Assuming they had waited for their siblings to arrive and were not able to get on the lifeboat



We se that the same is true for parents/children on board. As the number of family members goes up survival rate goes down.



We can see that the survival rate is highest for port followed by port q and then port s

We check for any missing values in the dataset we find that there are 177 na values in age and 2 na values in embarked column

We set the two embarked values to “s” as it is the most frequent variable.

We find that there are 177 missing values in the age column

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
1	1	0	3	Braund, Mr. Owen Harris	male	22.00	1	0
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.00	1	0
3	3	1	3	Heikkinen, Miss. Laina	female	26.00	0	0
4	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	1	0
5	5	0	3	Allen, Mr. William Henry	male	35.00	0	0
6	6	0	3	Moran, Mr. James	male	NA	0	0

Missing age value at no.6

To fill out the age values we use the titles like mr ,mrs, master, miss, dr etc given to the each individual in the name column. We use the mean of the values with a given title to substitute for the missing value.

eg- for row 6 in the above figure the age value will be replaced by the mean of the age value for people with Mr as their title

we check for the titles contained in the dataset consisting of missing values

we find that all the missing age values have either Mr, Mrs, Dr, Master or Miss as their title

the mean values for the respective titles is given by

```
# get mean value of vectors
master_mean=round(mean(train[master_vec,]$Age,na.rm=TRUE))
miss_mean=round(mean(train[miss_vec,]$Age,na.rm=TRUE))
mr_mean=round(mean(train[mr_vec,]$Age,na.rm=TRUE))
mrs_mean=round(mean(train[mrs_vec,]$Age,na.rm=TRUE))
dr_mean=round(mean(train[dr_vec,]$Age,na.rm=TRUE))
```

The values are

Master_mean=5

Mr_mean=32

Mrs_mean=36

Dr_mean=42

Miss_mean=22

After substituting the values the dataset looks like this

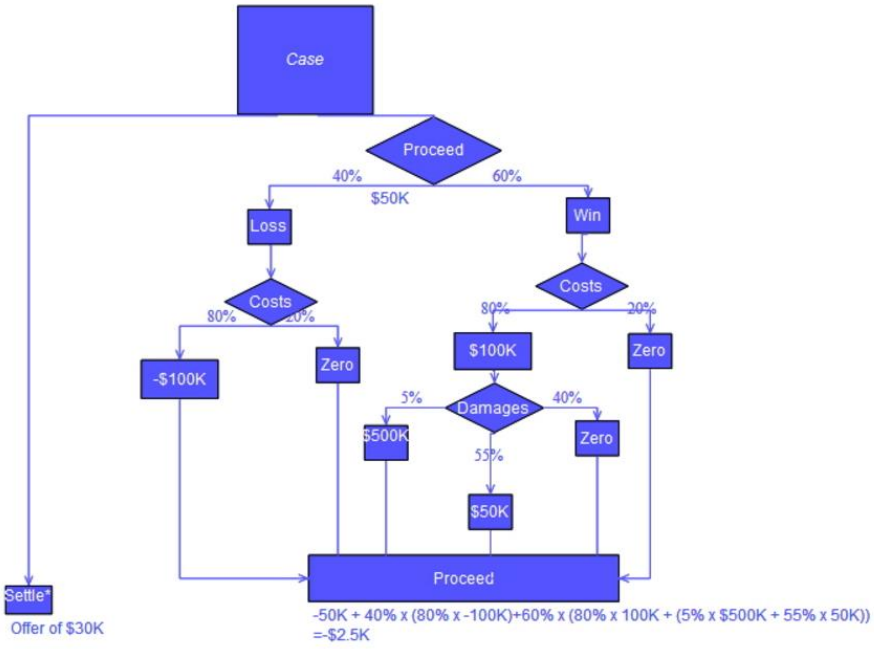
	Survived	Pclass	Sex	Age
1	0	1	male	32.00
2	0	3	male	23.00
3	0	3	female	41.00
4	1	1	male	35.00
5	1	3	male	32.00
6	0	3	male	32.00
7	1	2	female	18.00

We can see that the missing age value in row 6 has been replaced by 32 which is mean for title mr

Decision Trees

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute. 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.



For our decision tree the nodes will have class, age,sex,sibsp,parch,embark as deciding factors

Confusion Matrix

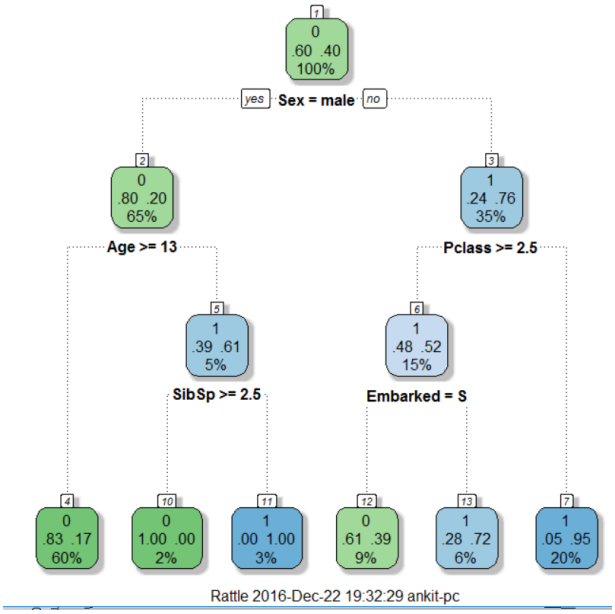
It measures the prediction rate of the classifier

		Predicted condition	
	Total population	Predicted Condition positive	Predicted Condition negative
True condition	condition positive	**True positive**	**False Negative** (Type II error)
	condition negative	**False Positive** (Type I error)	**True negative**

The accuracy of the model is given by

$$\text{Accuracy (ACC)} = \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$$

Interpreting the decision tree



We can see that only **20% men survived** while **76% women survived** confirming that **women were given preference** over men in the lifeboats. **61% of people under 13 years** were alive compared to **only 17% for the age above 13** implying **that children were given priority**. We see that looking at the SibSp node that 100% of children under 13 yrs having less than 2.5 siblings/spouse survived and 100% of them having more than 2.5sibs/spouse died, confirming our plot that **people with less siblings have high survival rate**. Lastly for women who were in the lowest class and embarked on port q,c had a survival rate of 72% while those who got on at port S had survival rate of 39%. This is in support of our port barplot which says that **more people from the s port died as compared to q,c ports**.

The confusion matrix

	pred_dectree	
	0	1
0	111	8
1	21	38

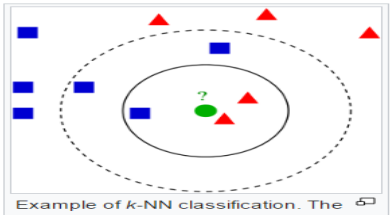
The accuracy is 0.837078 which is higher as compared to one computed in first project which was 0.7977

Thus the model has improved after adding some more variables to the tree

K-Nearest Neighbour

In pattern recognition, the ***k*-Nearest Neighbors algorithm** (or ***k*-NN** for short) is a non-parametric method used for classification and regression.[1] In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

In *k-NN classification*, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.



Example of *k*-NN classification. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If *k* = *3* (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If *k* = *5* (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

We first convert the strings like “female”, “male” in the Sex column to numbers so that distance can be attributed to those vectors

Also, the scale of the values affects the algorithm as it is a measure of distance. For example the age values are very large compared to the numbers 1,0 so the mismatch in scale will cause problems in the actual distance calculated.

Thus we scale all the parameters age, class,sibsp and parch and

```
# scaling and normalizing data
#scaling pclass
min_class <- min(train_knn$Pclass)
max_class <- max(train_knn$Pclass)
train_knn$Pclass <- (train_knn$Pclass - min_class) / (max_class - min_class)
test_knn$Pclass <- (test_knn$Pclass - min_class) / (max_class - min_class)

# scaling Age
min_age <- min(train_knn$Age)
max_age <- max(train_knn$Age)
train_knn$Age <- (train_knn$Age - min_age) / (max_age - min_age)
test_knn$Age <- (test_knn$Age - min_age) / (max_age - min_age)

# scaling sibsp
min_sibsp <- min(train_knn$SibSp)
max_sibsp <- max(train_knn$SibSp)
train_knn$SibSp<- (train_knn$SibSp - min_sibsp) / (max_sibsp - min_sibsp)
test_knn$SibSp <- (test_knn$SibSp - min_sibsp) / (max_sibsp - min_sibsp)

# scaling parch
min_parch <- min(train_knn$Parch)
max_parch <- max(train_knn$Parch)
train_knn$Parch <- (train_knn$Parch - min_parch) / (max_parch - min_parch)
test_knn$Parch <- (test_knn$Parch - min_parch) / (max_parch - min_parch)
```

We also set values to sex,embarkment variables as we will need to get the distance while using knn

```
# changing class male , female to values 0,1
train_knn$Sex<-gsub("female",0,train_knn$Sex)
train_knn$Sex<-gsub("^male",1,train_knn$Sex)

test_knn$Sex<-gsub("^male",1,test_knn$Sex)
test_knn$Sex<-gsub("female",0,test_knn$Sex)

#changing values S,Q,C to 0,1,2
train_knn$Embarked<-gsub("S",1,train_knn$Embarked)
train_knn$Embarked<-gsub("^Q",0,train_knn$Embarked)
train_knn$Embarked<-gsub("^C",2,train_knn$Embarked)

test_knn$Embarked<-gsub("S",1,test_knn$Embarked)
test_knn$Embarked<-gsub("^Q",0,test_knn$Embarked)
test_knn$Embarked<-gsub("^C",2,test_knn$Embarked)
```

We set “female”,”male” to 0,1

And “s”,”C”,”Q” to 0,1,2

Optimizing the value of the number of neighbours

I have written the code so that k will vary from 1 to 0.2*no of observations in training data

We get the accuracy from each of these and then find the k value for the accuracy is maximum

The code for this is

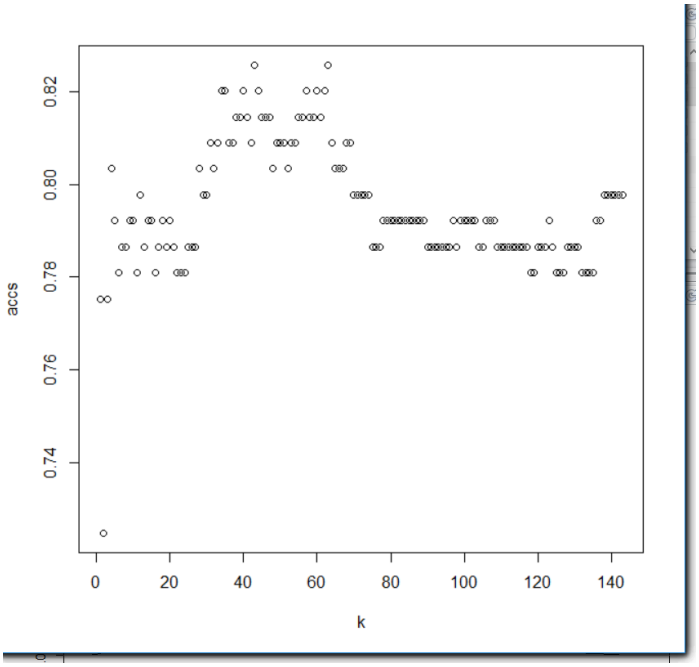
```
range <- 1:round(0.2 * nrow(train_knn))
accs <- rep(0, length(range))

for (k in range) {
  #make prediction using k neighbours
  pred_knn <- knn(train_knn, test_knn,train_labels, k = k)

  #  construct the confusion matrix: conf
  conf_knn <- table(test_labels,pred_knn)
  # calculate the accuracy and store it in accs[k]
  accs[k] <- sum(diag(conf_knn))/sum(conf_knn)
}

# Plot the accuracies. Title of x-axis is "k".
plot(range, accs, xlab = "k")
x11()
which.max(accs)

acc_kn<-accs[43]
```

We get the k =43 as the value with max accuracy

The accuracy is given by acc_kn=0.8258426

Which is better than we got in project 1 which was 0.7921

I have plotted the points to see which gives us the highest accuracy



we can see that at k=43 accuracy is max

Cross validation

Cross-validation, sometimes called rotation estimation, is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of *known data* on which training is run (*training dataset*), and a dataset of *unknown data* (or *first seen* data) against which the model is tested (*testing dataset*). The goal of cross validation is to define a dataset to "test" the model in the training phase (i.e., the *validation dataset*), in order to limit problems like overfitting, give an insight on how the model will generalize to an independent dataset.

One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the *training set*), and validating the analysis on the other subset (called the *validation set* or *testing set*). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. [5]

Method -3

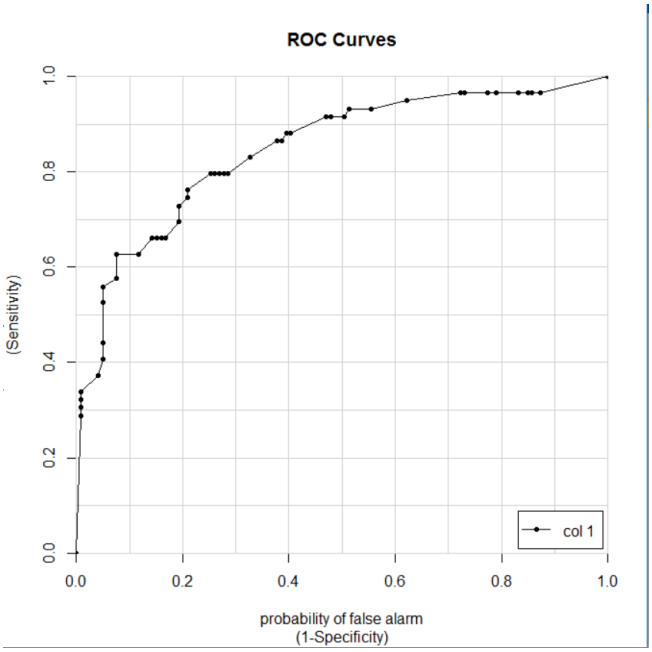
I have used repeated-cross validation on knn using the train function in the caret package to tarin the model

In this model we end up getting the probabilities of survival.

We can optimize this by finding the probability separation which will classify these values for highest accuracy using the roc

ROC

In statistics, a **receiver operating characteristic** (**ROC**), or **ROC curve**, is a **graphical plot** that illustrates the performance of a **binary classifier** system as its discrimination threshold is varied. The curve is created by plotting the **true positive rate** (TPR) against the **false positive rate** (FPR) at various threshold settings



We get the probability for separation into classes that maximizes the accuracy =0.8458

It is given by

```
0 vs. 1 0.8458909
```

We classify the predicted values using this threshold and calculate the accuracy

Acc_n=0.8088988

This is better than project 1 but not as good as one we got by optimizing k

Conclusion

By adding more variables to the decision tree we have improved its accuracy

By optimizing the value of k we have increased the accuracy of the model

By plotting the roc curve and using cross validation we have increased the accuracy of the model

The models ranked by accuracy are

Decision tree(accuracy=0.837078), knn with optimized k(accuracy=0..82584), cross validated knn (0.809)

References

https://en.wikipedia.org/wiki/Decision_tree[1]

https://en.wikipedia.org/wiki/Confusion_matrix[[2]

https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm[3]

https://en.wikipedia.org/wiki/Cross-validation_(statistics) [4]