KNN

Subhadeep Majumder

07-09-2021

Creating A KNN Model On The Titanic Dataset

Figure 1: Sinking of the Titanic

Importing The Required Libraries

Well, first we import all the libraries required to develop the model:

```
library(gt)
library(class)
library(caret)
```

```
library(GGally)
```

Importing the given Dataset

Then we import the Given Dataset "titanic.csv" into the project:

```
titanic_ds<-read.csv('titanic_ds.csv',stringsAsFactors = FALSE)
```

Pre-processing The Data to Increase its Quality

In this stage of Data Analysis, we transform the structure and type of the data to make it suitable for the analysis that is to follow.

First, we change the categorical data of Sex(Male,Female) to a Numerical Form Sex(1,0):

```
titanic_ds$Sex<-ifelse(titanic_ds$Sex== "male" ,1,0)
```

Second, we change the Categorical data of Embarked(Q,S,C) to Embarked(0,1,2):

```
 titanic_ds$Embarked[titanic_ds$Embarked=="Q"]<-0
 titanic_ds$Embarked[titanic_ds$Embarked=="S"]<-1
 titanic_ds$Embarked[titanic_ds$Embarked=="C"]<-2
```

Now let's take a look at our imported dataset:

```
gt_preview(titanic_ds)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
1	892	0	3	Kelly, Mr. James	1	34.5	0	0
2	893	1	3	Wilkes, Mrs. James (Ellen Needs)	0	47.0	1	0
3	894	0	2	Myles, Mr. Thomas Francis	1	62.0	0	0
4	895	0	3	Wirz, Mr. Albert	1	27.0	0	0
5	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	22.0	1	1
6..417								
418	1309	0	3	Peter, Master. Michael J	1	NA	1	1

We noticed that there are some columns that won't contribute to building the KNN model like the PassengerID and Name column. So we remove them and Preview the clean data:

```
titanic_clean<-titanic_ds[,c(2,3,5,6,7,8,10,12)]
gt_preview(titanic_clean)
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
1	0	3	1	34.5	0	0	7.8292	0
2	1	3	0	47.0	1	0	7.0000	1
3	0	2	1	62.0	0	0	9.6875	0
4	0	3	1	27.0	0	0	8.6625	1
5	1	3	0	22.0	1	1	12.2875	1
6..417								
418	0	3	1	NA	1	1	22.3583	2

In the next step, we check if there are any missing values in the dataset:

```
sum(is.na(titanic_clean))
```

```
## [1] 87
```

As the attribute of interest based on which we need to classify the data is "Survived" where (Survived,Non-Survived):(1,0), we change the type of the Survival Data as Factors:

```
titanic_clean$Survived<-as.factor(titanic_clean$Survived)
str(titanic_clean)
```

```
## 'data.frame':    418 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 1 1 2 1 2 1 2 1 ...
##  $ Pclass  : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex     : num  1 0 1 1 0 1 0 1 0 1 ...
##  $ Age     : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp   : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch   : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Fare    : num  7.83 7 9.69 8.66 12.29 ...
##  $ Embarked: chr  "0" "1" "0" "1" ...
```

Before we change the type of the data, let's first locate the missing values and impute them:

```
sum(is.na(titanic_clean$Age)) #86 missing values in Age column
```

```
## [1] 86
```

```
sum(is.na(titanic_clean$Pclass))
```

```
## [1] 0
```

```
sum(is.na(titanic_clean$Sex))
```

```
## [1] 0
```

```
sum(is.na(titanic_clean$SibSp))
```

```
## [1] 0
```

```
sum(is.na(titanic_clean$Fare))# 1 missing value in fare column
```

```
## [1] 1
```

We impute the missing values in Age column with Median and that of Fare column with the Mode:

```
getmode <- function(mode_fare) {
   uniqv <- unique(mode_fare)
   uniqv[which.max(tabulate(match(mode_fare, uniqv)))]
}
mode_fare<-titanic_clean$Fare

titanic_clean$Age[is.na(titanic_clean$Age)]<-median(titanic_clean$Age,na.rm = TRUE)
titanic_clean$Fare[is.na(titanic_clean$Fare)]<-getmode(mode_fare)
getmode(mode_fare)
```

```
## [1] 7.75
```

Then we coerce all the columns in the dataset into numeric data type:

```
titanic_clean$Embarked<-as.numeric(titanic_clean$Embarked)
titanic_clean$Pclass<-as.numeric(titanic_clean$Pclass)
titanic_clean$SibSp<-as.numeric(titanic_clean$SibSp)
titanic_clean$Parch<-as.numeric(titanic_clean$Parch)
titanic_clean$Fare<-as.numeric(titanic_clean$Fare)
str(titanic_clean)
```

```
## 'data.frame':    418 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 1 1 2 1 2 1 2 1 ...
##  $ Pclass  : num  3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex     : num  1 0 1 1 0 1 0 1 0 1 ...
##  $ Age     : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp   : num  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch   : num  0 0 0 0 1 0 0 1 0 0 ...
##  $ Fare    : num  7.83 7 9.69 8.66 12.29 ...
##  $ Embarked: num  0 1 0 1 1 1 0 1 2 1 ...
```

We preview the final processed dataset again:

```
gt_preview(titanic_clean)
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
1	0	3	1	34.5	0	0	7.8292	0
2	1	3	0	47.0	1	0	7.0000	1
3	0	2	1	62.0	0	0	9.6875	0
4	0	3	1	27.0	0	0	8.6625	1
5	1	3	0	22.0	1	1	12.2875	1
6..417								
418	0	3	1	27.0	1	1	22.3583	2

Now we Start the Analysis Stage.

Well first we normalize the data set using Z-scores so that there are no biases in the data due to difference in location and scale:

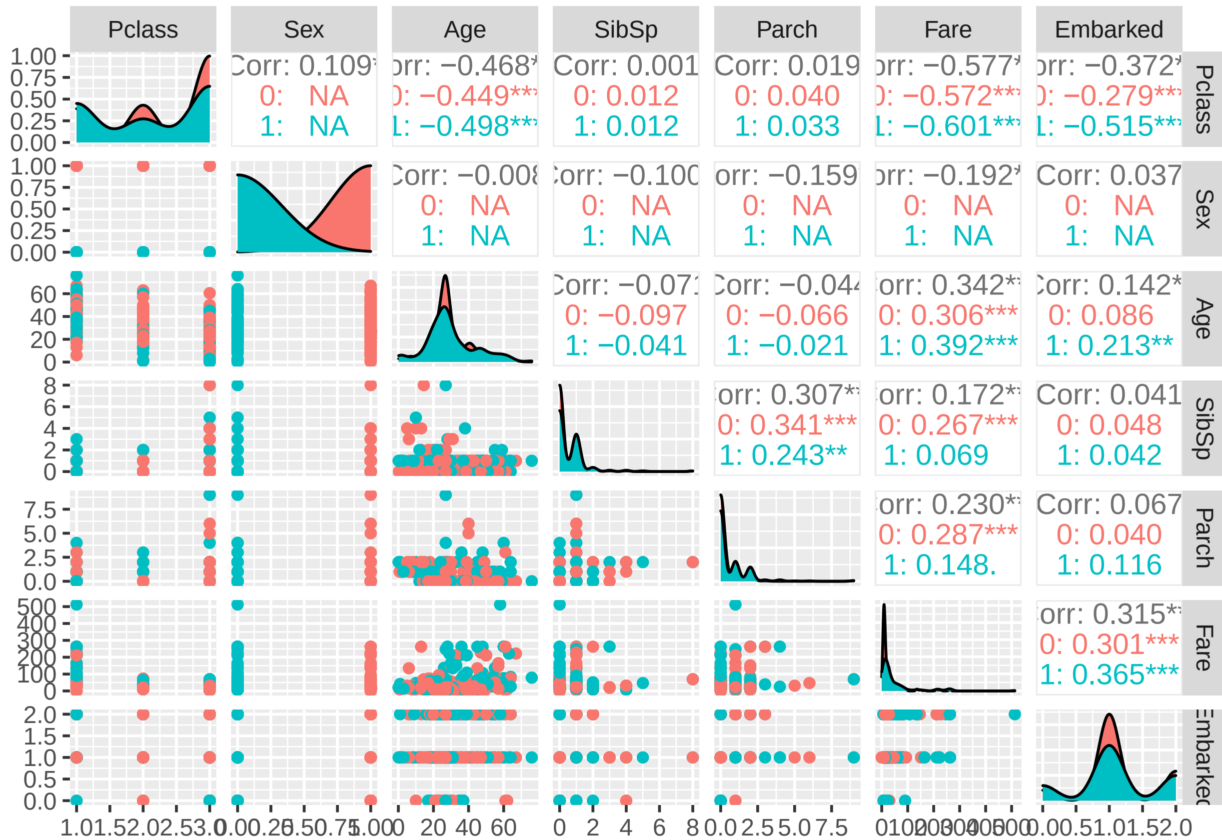
```
normlz <-function(x) {return(x-as.numeric(mean(x)))/as.numeric(sd(x)) }
titanic_norm<-as.data.frame(lapply(titanic_clean[,2:8], normlz))
summary(titanic_norm)
```

```
##      Pclass             Sex                Age              SibSp
##  Min.   :-1.2656   Min.   :-0.6364   Min.   :-29.429   Min.   :-0.4474
##  1st Qu.:-1.2656   1st Qu.:-0.6364   1st Qu.: -6.599   1st Qu.:-0.4474
##  Median : 0.7344   Median : 0.3636   Median : -2.599   Median :-0.4474
##  Mean   : 0.0000   Mean   : 0.0000   Mean   :  0.000   Mean   : 0.0000
##  3rd Qu.: 0.7344   3rd Qu.: 0.3636   3rd Qu.:  6.151   3rd Qu.: 0.5526
##  Max.   : 0.7344   Max.   : 0.3636   Max.   : 46.401   Max.   : 7.5526
##      Parch             Fare             Embarked
##  Min.   :-0.3923   Min.   :-35.560   Min.   :-1.134
##  1st Qu.:-0.3923   1st Qu.:-27.665   1st Qu.:-0.134
```

```
## Median :-0.3923   Median :-21.106   Median :-0.134
## Mean   : 0.0000   Mean   :  0.000   Mean   : 0.000
## 3rd Qu.:-0.3923   3rd Qu.: -4.089   3rd Qu.:-0.134
## Max.   : 8.6077   Max.   :476.769   Max.   : 0.866
```

This plot observes the correlation between the different attributes:

```
ggpairs(titanic_clean,columns=2:8,mapping =aes(color=Survived))
```



Then we split the Data frame into the Training Dataset and Testing Dataset:

```
titanic_train<-titanic_norm[1:293,1:7]
titanic_test<-titanic_norm[294:418,1:7]
titanic_train_labels<-as.array(titanic_clean[1:293,1])
titanic_test_labels<-as.array(titanic_clean[294:418,1])
```

Finding the Optimal Number Of Neighbours

Here we create a loop to find the Percentage of Accuracy for each k from 1 to 25:

```
i=1
k.optm=1
for (i in 1:25){
 knn.pred <- knn(train=titanic_train, test=titanic_test, cl=titanic_train_labels, k=i)
 k.optm[i] <- 100 * sum(titanic_test_labels == knn.pred)/NROW(titanic_test_labels)
```

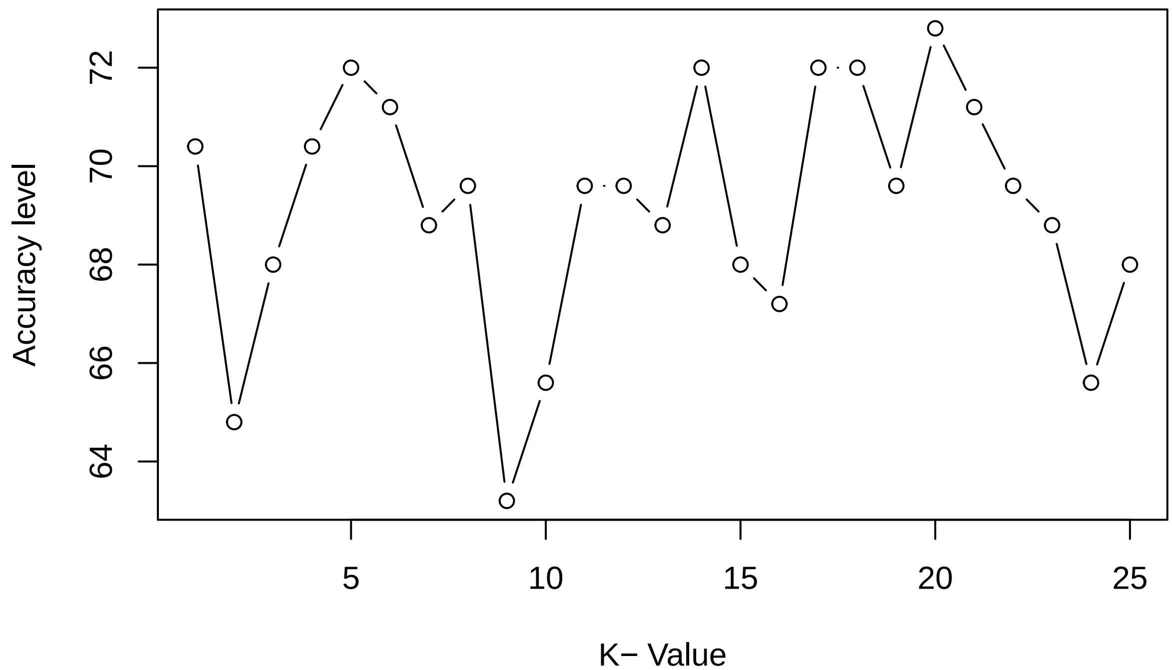
```
 k=i
 cat(k,'=',k.optm[i],'
')

}
```

```
## 1 = 70.4
## 2 = 64.8
## 3 = 68
## 4 = 70.4
## 5 = 72
## 6 = 71.2
## 7 = 68.8
## 8 = 69.6
## 9 = 63.2
## 10 = 65.6
## 11 = 69.6
## 12 = 69.6
## 13 = 68.8
## 14 = 72
## 15 = 68
## 16 = 67.2
## 17 = 72
## 18 = 72
## 19 = 69.6
## 20 = 72.8
## 21 = 71.2
## 22 = 69.6
## 23 = 68.8
## 24 = 65.6
## 25 = 68
```

Then we draw the accuracy plot and determine the value of k for which we have the highest Accuracy:

```
acc_plot<-plot(k.optm, type="b", xlab="K- Value",ylab="Accuracy level")
```



Thus, from the graph we see that the model has the best accuracy of 72.8% for K=20 neighbours.
