HW1- Data622 Qn3

The result/output are mentioned here. 1. Accuracy, confusion Matrix for all values of k value. Plot accuracy v/s Accuracy.

Rpub link : https://rpubs.com/charlsjoseph/Data622_HWQn3

Below is the test data distribution. There are 53 negative classes and 7 positive classes. From this, we can infer that the dataset is imbalanced and accuracy is not one stop metric to determine the goodness of the classifier.

```
104 ```{r}
105 table(test.df$STA)
106 ```
```

```
 0  1
53  7
```

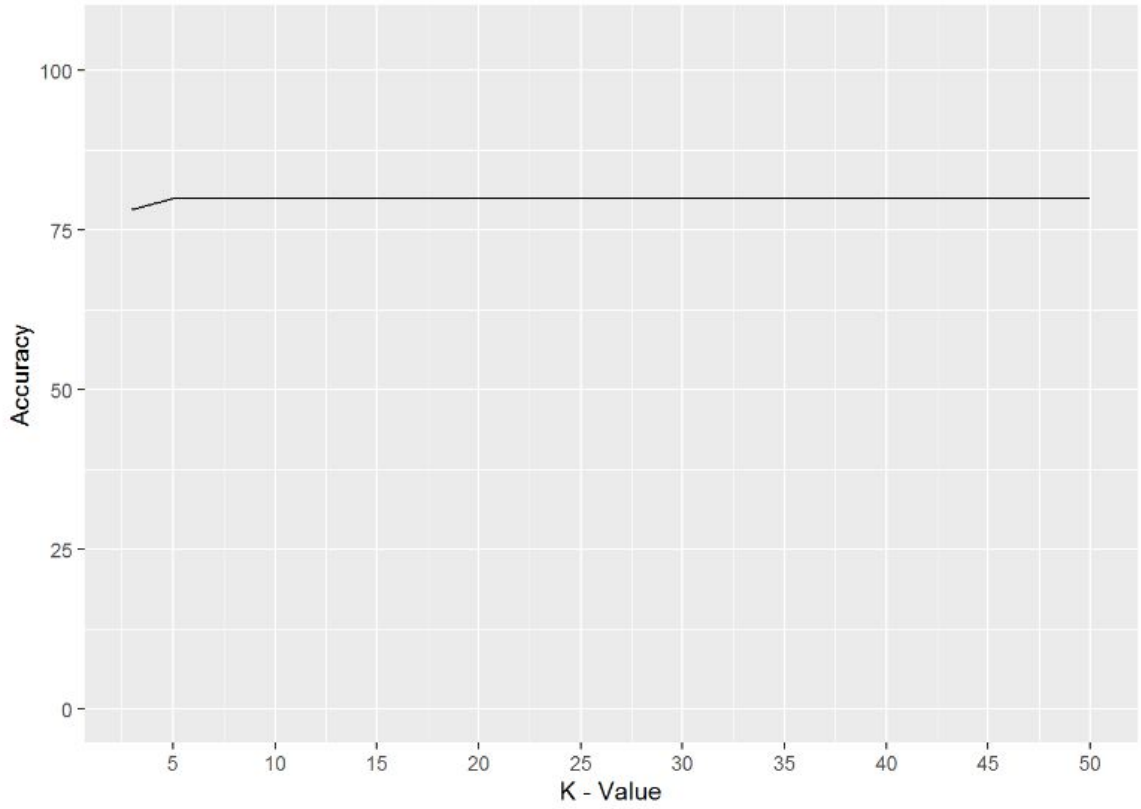
Let's look at the accuracy and the confusion matrix for all the K values.

```
kable(knnMetrics)
```

k	accuracy
3	78.33333
5	80.00000
7	80.00000
15	80.00000
25	80.00000
50	80.00000

```
## [1] 78.33333
## [1] "The confusion metric when k =  3"
##
##      0  1
##   0 44  9
##   1  4  3
## [1] 80
## [1] "The confusion metric when k =  5"
##
##      0  1
##   0 47 11
##   1  1  1
## [1] 80
## [1] "The confusion metric when k =  7"
##
##      0  1
##   0 48 12
## [1] 80
## [1] "The confusion metric when k =  15"
##
##      0  1
##   0 48 12
## [1] 80
## [1] "The confusion metric when k =  25"
##
##      0  1
##   0 48 12
## [1] 80
## [1] "The confusion metric when k =  50"
##
##      0  1
##   0 48 12
```

Accuracy v/s K-value



Even though accuracy is increasing when the k increases, we need to look for the false positive(type 1) and false negative(type 2 error). Let's find the precision and sensitivity of this model.

Accuracy = 78.33

Precision = TP/(TP+FP)

= 3/(3+4) = **0.42**

Recall = TP/(TP+FN)

= 3/(3+9) = **0.25**

Let's also do the same exercise for the model(k=5) to compare the performance.

Accuracy = 80

Precision = TP/(TP+FP)

= 1/(1+1) = **0.5**

Recall = TP/(TP+FN)

= 1/(1+11) = **0.07**

When K= 7, 15 an so on

Accuracy = 80

Precision = TP/(TP+FP) = 0

Recall = TP/(TP+FN) = 0

We see different models with different precision, recall with almost the same accuracy. Since the dataset is imbalanced, we can't rely on Accuracy. Based on the business priorities, we have to decide on which performance metrics we will use to choose the models.

If we decide to reduce the false positive error(Type1 error), the model with k=5 is better. Because this gives a comparatively good precision(i.e Less Type 1 Errors)

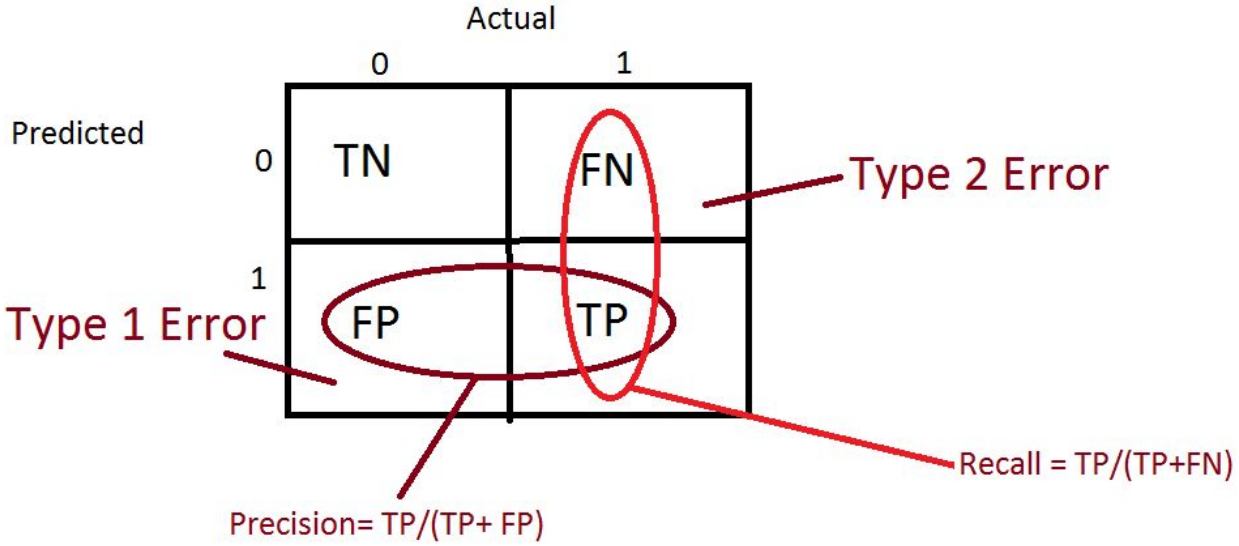
If we decide to reduce the false negative error(Type 2), the model with k=3 is better. Because this gives a comparatively good recall(i.e Less Type 2 Errors)

Conclusion- Since this looks to be a classifier used in the medical field, I assume this would be a classifier focussing on reducing the Type 2 errors. I will reach out to the business stakeholders to get a better picture on the requirement explaining on these terms.

Type1 error(false positive error) : Predicting as positive when it is actually negative.
E.g : predicting spam when it is not spam. Impact is that customers will miss the email.

Type2 error(false negative error) : Predicting as negative when it is actually positive.
E.g : diagnosed as not having cancer when the patient has cancer. Impact is that the patient will die.

However the overall goal is to reduce both Type-1 and Type-2 errors. But in most cases, it is not possible, we would end up compromising either of the performance metrics if the requirement is restricted to a specific metric.



When you have a situation where you need good precision and recall both, we can go for a f1 score to compare the different models.