Mike Milad Nourian

4.1)a)

- Feature Pclass: We see that the rate of surviving is extremely less when Pclass = 3.0 which indicates that the it is less likely for the passengers from the low class to survive. Also the survival rate of Pclass = 1 is higher.

- Feature Sex: For feature sex, it is more likely for Sex=0 (female) to survive than male, and Sex=1 death rate is higher.

- Feature Age: We see that children are proportionally more likely to survive. People in 20s and 30s have a high death rate. Most of the population also are in their 20s and 30s.

- Feature SibSp:
  For Sibpy= 0, the death rate is higher than others.
  For Sibpy=1, the survival and the death rate are very close. Very small proportion of population are in SibSp >= 2.

- Feature Parch:
  Death rate of Parch = 0 is high. Death rate and survival rate for Parch = 1 and Parch ==2 are almost equal. Almost all the population fall within Parch <= 2.

- Feature Fare:
  Death rate of fare = 0 (low fair/cheap) was high while the survival rate for other fares (More than 0) was higher, meaning that most of people who payed more fare survived. Most of the population fell into fare = 0.

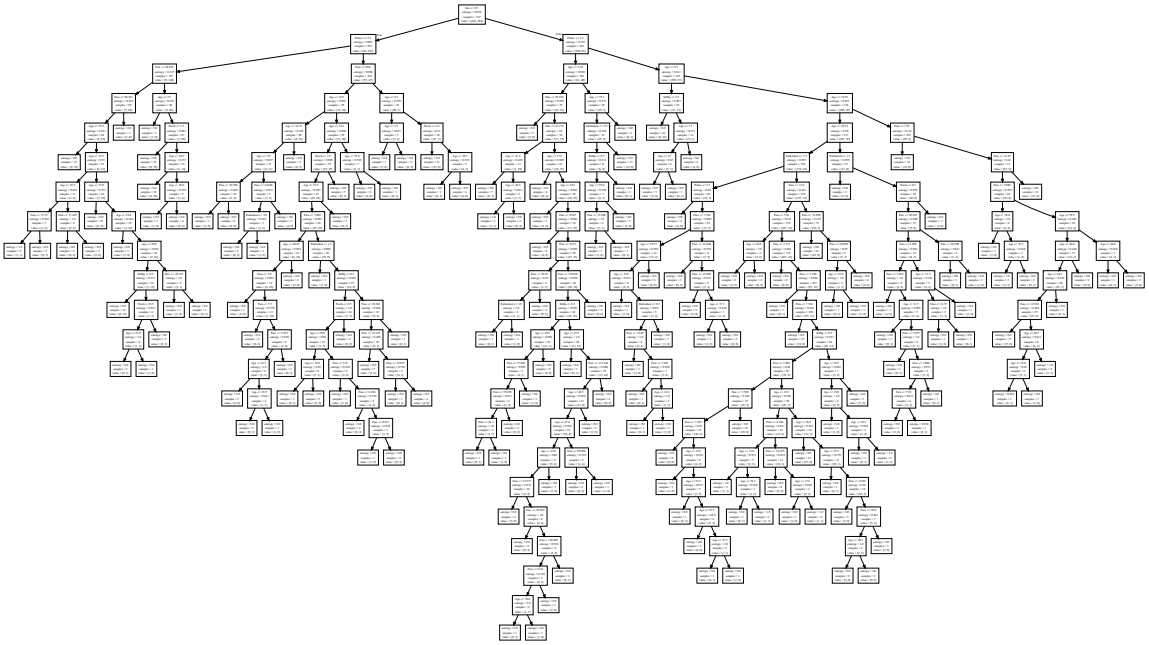
- Feature Embarked:
  Relative Survival rate for Embarked = 0 is high but the death rate for Embarked = 2.0 is very high. Also, most of the population fell into Embarked = 0.0 category.

4.2)b)

- The error obtained was 0.485. I used the numpy random library functions such as np.random.choice() to obtain the results.

4.2)c)

- training error for the DecisionTreeClassifier is: 0.014
  The graph for the decision tree is built in the pdf.



4.2)d)

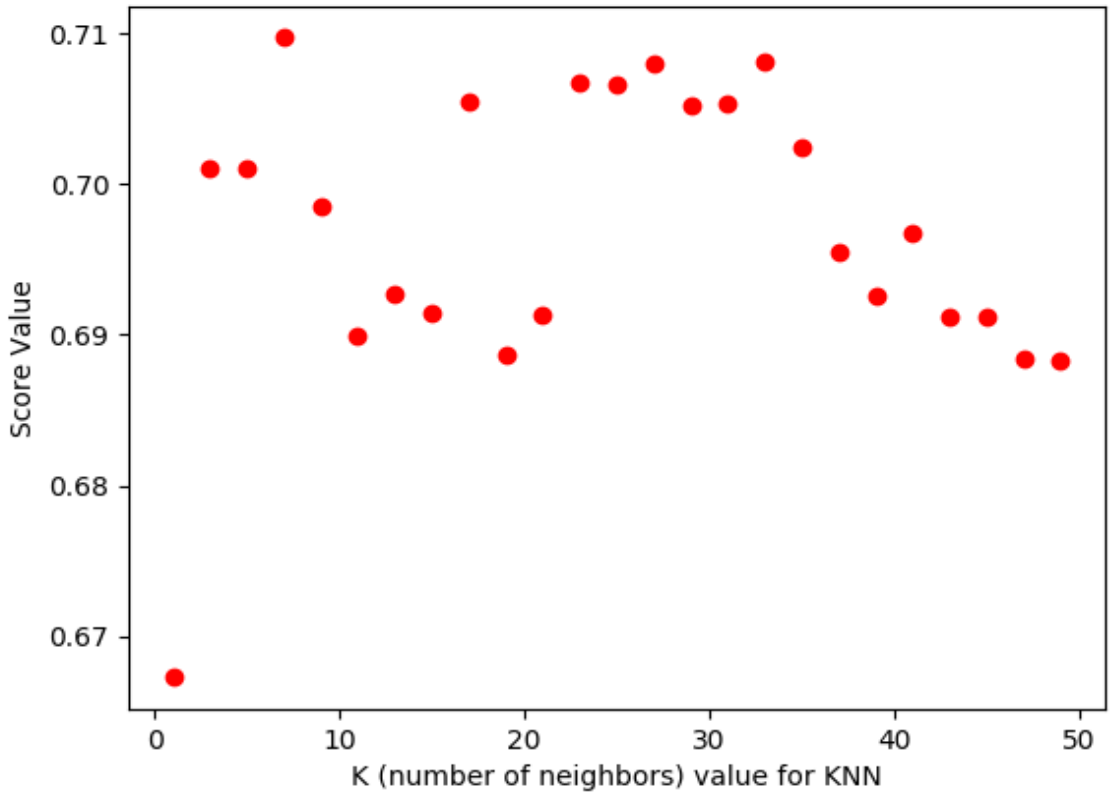
- training error for the KNeigbborsClassifier for n_neighbors = 3 is: 0.167
- training error for the KNeigbborsClassifier for n_neighbors = 5 is: 0.201
- training error for the KNeigbborsClassifier for n_neighbors = 7 is: 0.240

4.2)e)

- Majority: train_err: 0.071, test_err: 0.286
- Random: train_err: 0.086, test_err: 0.342
- DecisionTree: train_err: 0.002, test_err: 0.173
- KNN: train_err: 0.037, test_err: 0.221

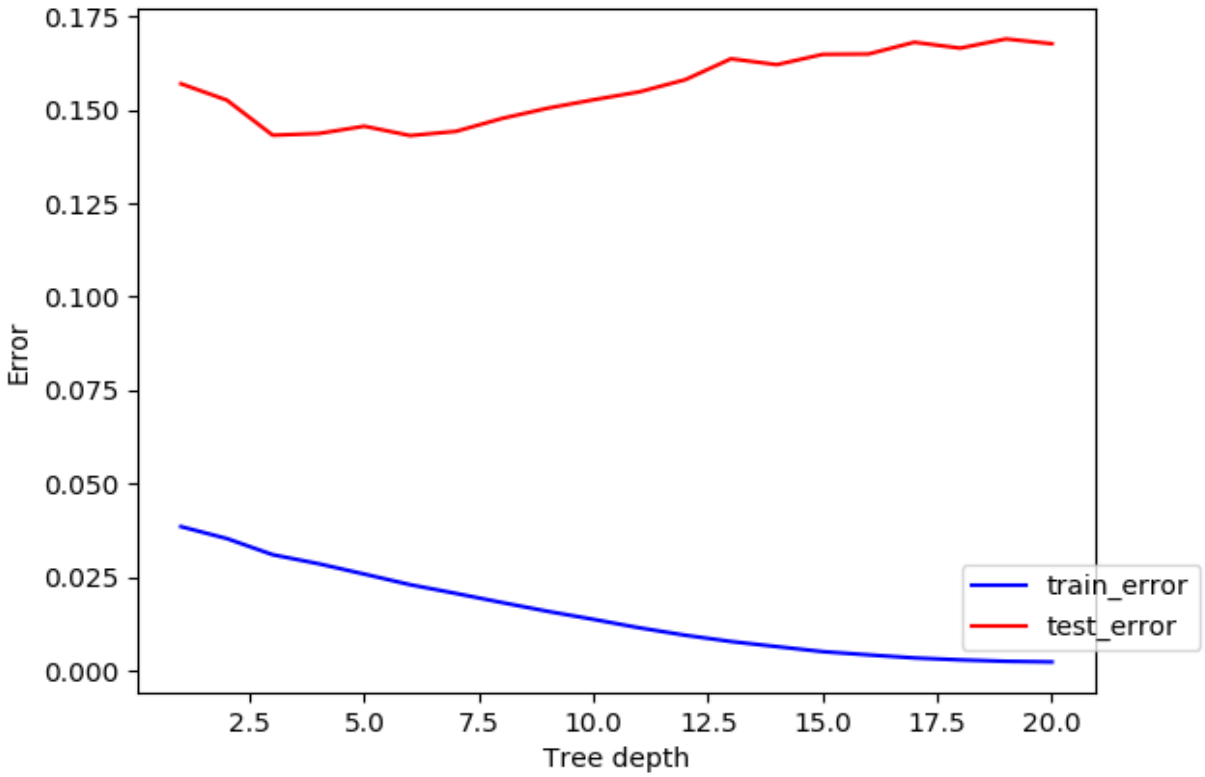
4.2)f)

- As shown on the graph as data points, the best value for the k parameter is k=5 (ie. looking at the 5 nearest neighbors). We obtain this value by looking at the accuracy (score) which indicates how many times our prediction has been correct.
- The general trend is as such:
  for small k values, the model is inaccurate, and it improves until k = 7, and after that point it oscillates for a few k values, and for k~30, the trend shows the score (accuracy) decreases, indicating that increasing k actually results in less accuracy after a certain threshold as we observe the negative effect of overfitting.



4.2)g)

- Looking at the figure for testing overfitting of the graphs, we see that by increasing the depth of the tree, one can see that the smallest error is found when the depth of the tree is depth == 3 or depth == 6.
- In addition, we can observe the pattern of overfitting in the graph. As we increase the depth of the tree, we see that the error on the training sample becomes smaller and smaller, however, more complexity results in larger error on the testing sample. Therefore, overfitting is seen in the plot, specially when we increase the depth (complexity) of the tree.



4.2)h)

- From the graph, we can see that by increasing the size of the dataset where we train our model,
- the error in our prediction decreases significantly. One can see that this trend is more obvious for the DecisionTree model, as KNN is less sensitive to the changes to the training data size.