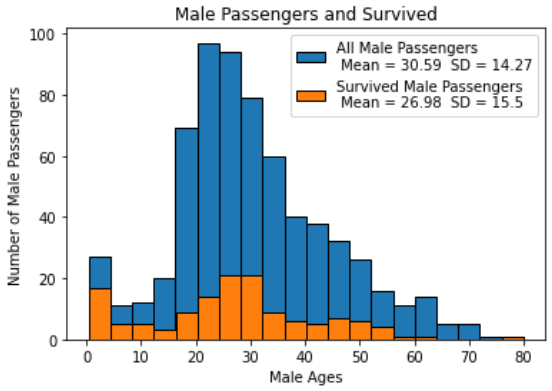
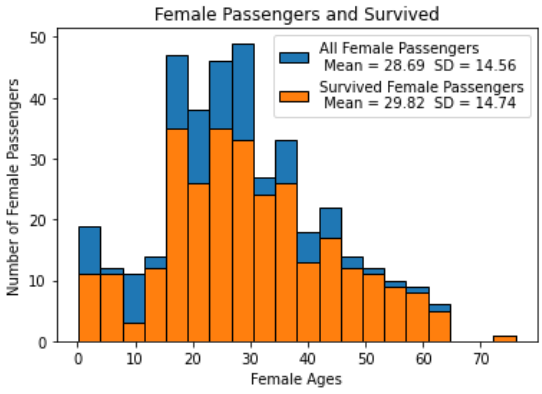
On the morning of April 15, 1912, the RMS Titanic hit an iceberg and sank in the North Atlantic. Of the roughly 1,300 passengers on board, 832 perished in the disaster. There were many factors contributing to the disaster, including navigational error, inadequate lifeboats, and the slow response of a nearby ship. Whether or not individual passengers survived had an element of randomness, but was far from completely random. In fact, it is possible to make a reasonably good model for predicting survival using information from the ship's passenger manifest.

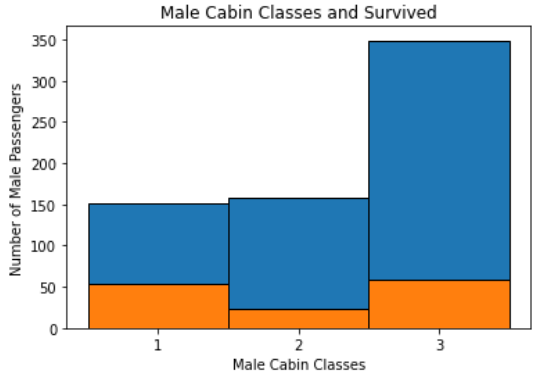
This project is to build prediction models from the data set ("TitanicPassengers.txt") containing information for 1046 passengers. Each line of the file contains information about a single passenger: cabin class (1st, 2nd, or 3rd), age, gender (1,0), whether the passenger survived (1,0) the disaster, and the passenger's name.
Please build models using **logistic regression** and the **k-nearest neighbors**. The logistic regression and the k-NN are the most commonly used classification methods. By examining the weights produced by logistic regression and the confusion matrix by k-NN to gain some insight into why some passengers were more likely to have survived than others.

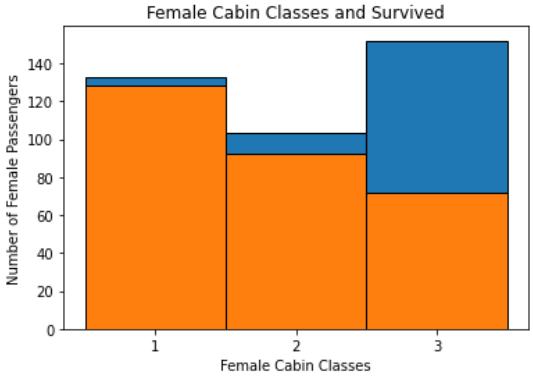
What you have to do:

1. First, read in the file and built examples of passengers with proper feature vector for features: cabin class (1st, 2nd, or 3rd), age, gender, whether the passenger survived. The feature survived is for the prediction label (hint: use 1,0,0 for the first-class passengers and etc.). Separate the examples into male and female examples respectively and find the statistics of the number of passenger in each cabin class and the number of passengers survived.
2. Gain insight into the passenger details by plotting the following figures



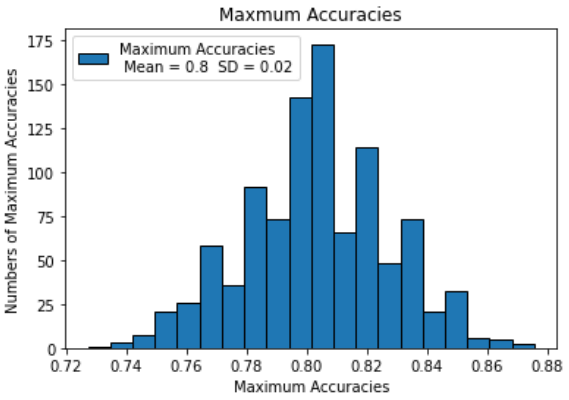


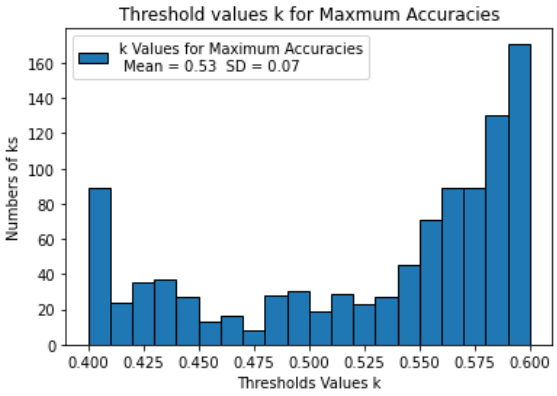


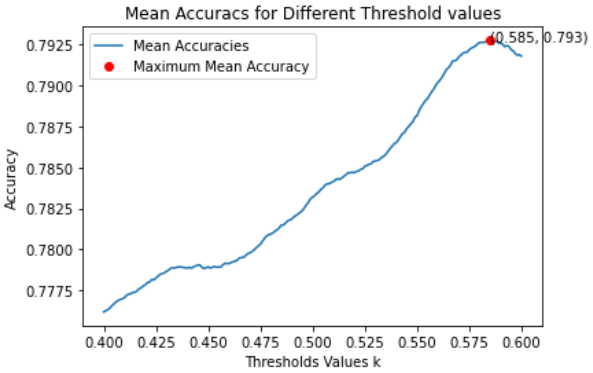


3. With the passenger examples, build a logistic regression model (refer to using the similar code used to build a model of the Boston Marathon data 24.15). Because the data set has a relatively small number of examples (1046 only), to avoid of getting an unrepresentative 80-20 split of the data, and then generate misleading results, **a.)** repeatedly creating 1000 different 80-20 splits for training-set and test-set (each split is created using the divide80_20 function defined in Figure 24.5), building and evaluate a classifier model using threshold probability k=0.5 for each split, and then reporting mean values of weights for each feature and 95% confidence intervals. **b.)** For each split, after finding the model, use the model to find the threshold probability k value that yields the maximum prediction accuracy. Collect these 1000 threshold values ks and their associated maximum prediction accuracies and generate bar charts to demo their mean and standard deviations. Also generate the plot that shows the mean accuracies vs the threshold values k. with a mark showing the threshold value k that yields maximum accuracy. **c.)** For each split, calculate the **auroc** of the roc curve by using the accuracy, sensitivity specificity, and pos. pred. val. Output the mean **auroc** for the 1000 tries too. The result should be like:

   Logistic Regression:
   Averages for all examples 1000 trials with k=0.5
   Mean weight of C1 = 1.138, 95% confidence interval = 0.12
   Mean weight of C2 = -0.081, 95% confidence interval = 0.102
   Mean weight of C3 = -1.057, 95% confidence interval = 0.111
   Mean weight of age = -0.033, 95% confidence interval = 0.006
   Mean weight of male gender = -2.407, 95% confidence interval = 0.148
   Mean accuracy = 0.783, 95% confidence interval = 0.05
   Mean sensitivity = 0.704, 95% confidence interval = 0.095
   Mean specificity = 0.783, 95% confidence interval = 0.05
   Mean pos. pred. val. = 0.704, 95% confidence interval = 0.095
   Mean AUROC = 0.84, 95% confidence interval = 0.053





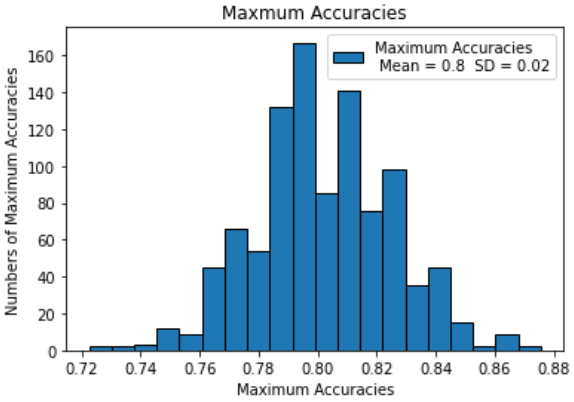


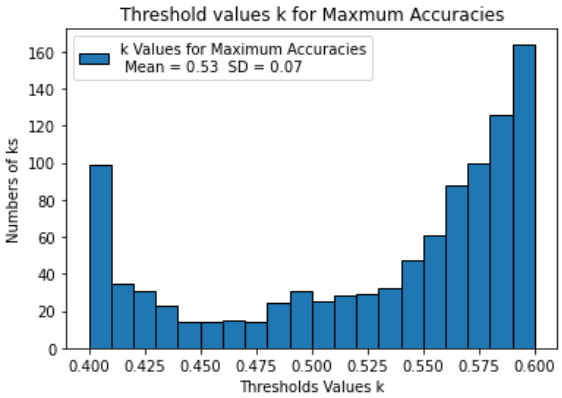
4. Concerning the value of age feature is much greater than other features, try to use **zScaling** and **iScaling** for the features of the examples and do step 3 again using scaled examples. The result should look like this:

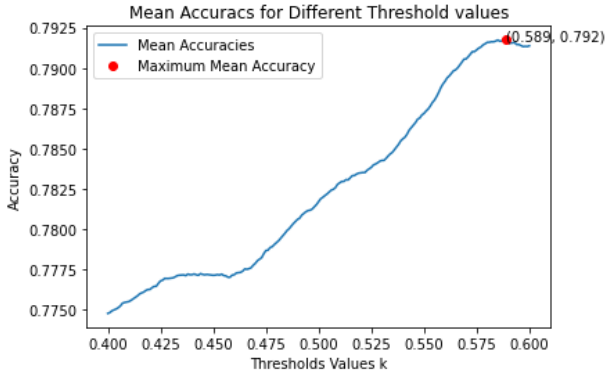
Logistic Regression with zScaling:

Averages for all examples 1000 trials with k=0.5

Mean weight of C1 = 1.14, 95% confidence interval = 0.115
Mean weight of C2 = -0.083, 95% confidence interval = 0.1
Mean weight of C3 = -1.057, 95% confidence interval = 0.113
Mean weight of age = -0.475, 95% confidence interval = 0.09
Mean weight of male gender = -2.408, 95% confidence interval = 0.146
Mean accuracy = 0.782, 95% confidence interval = 0.048
Mean sensitivity = 0.7, 95% confidence interval = 0.091
Mean specificity = 0.782, 95% confidence interval = 0.048
Mean pos. pred. val. = 0.7, 95% confidence interval = 0.091
Mean AUROC = 0.838, 95% confidence interval = 0.051



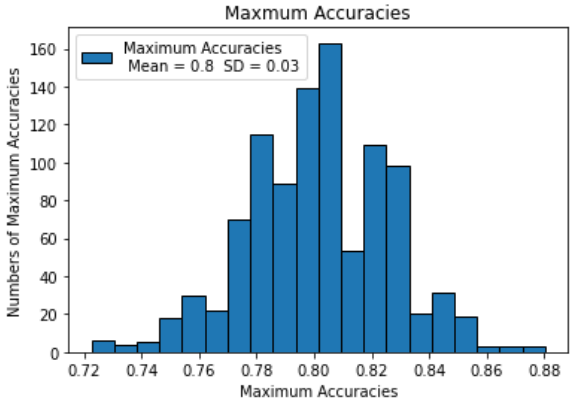


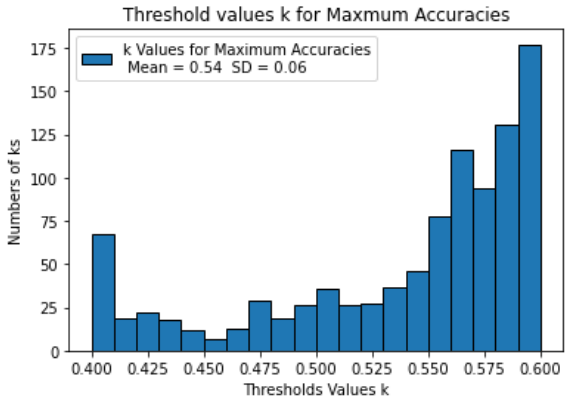


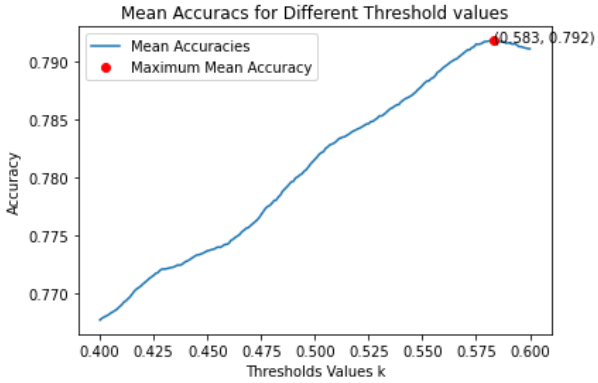
Logistic Regression with **iScaling**:

Averages for all examples 1000 trials with k=0.5

Mean weight of C1 = 1.069, 95% confidence interval = 0.112
Mean weight of C2 = -0.066, 95% confidence interval = 0.098
Mean weight of C3 = -1.002, 95% confidence interval = 0.112
Mean weight of age = -2.04, 95% confidence interval = 0.366
Mean weight of male gender = -2.403, 95% confidence interval = 0.152
Mean accuracy = 0.782, 95% confidence interval = 0.052
Mean sensitivity = 0.698, 95% confidence interval = 0.095
Mean specificity = 0.782, 95% confidence interval = 0.052
Mean pos. pred. val. = 0.698, 95% confidence interval = 0.095
Mean AUROC = 0.837, 95% confidence interval = 0.054





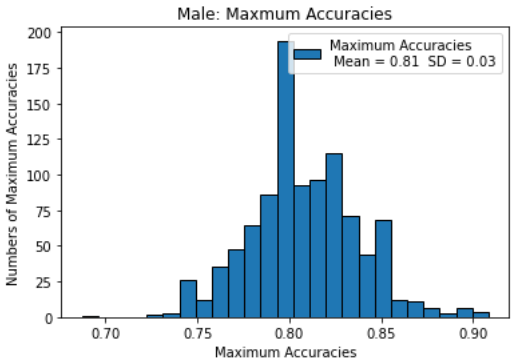


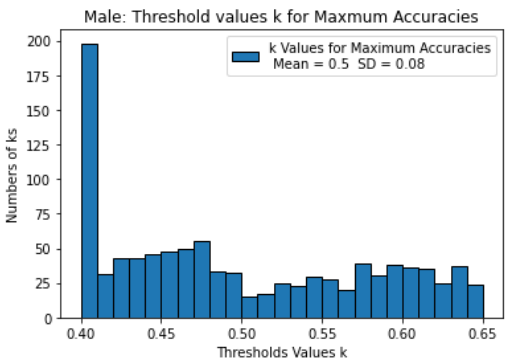
5. A bizarre idea is to predict male examples and female examples separately and combine their statistics. First, try to separate male passenger examples and female passenger examples from the whole examples. Then preform the same work from step 3 to step 5 and output the similar results and figures.

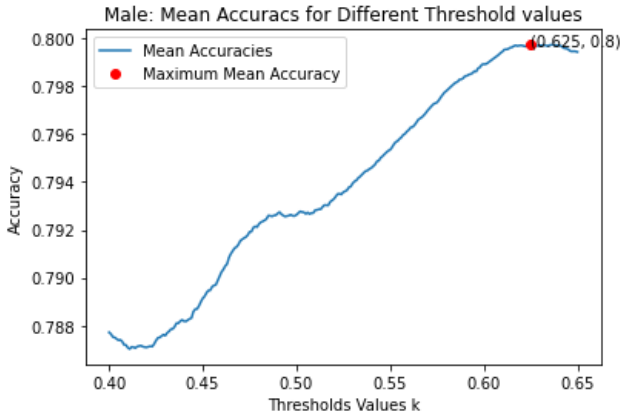
   Logistic Regression with Male and Female Separated:

   Averages for Male Examples 1000 trials with k=0.5

   Mean weight of C1 = 1.101, 95% confidence interval = 0.164
   Mean weight of C2 = -0.533, 95% confidence interval = 0.153
   Mean weight of C3 = -0.558, 95% confidence interval = 0.137
   Mean weight of age = -0.047, 95% confidence interval = 0.009
   Mean weight of male gender = 0.01, 95% confidence interval = 0.053
   Mean accuracy = 0.793, 95% confidence interval = 0.062
   Mean sensitivity = 0.081, 95% confidence interval = 0.103
   Mean specificity = 0.793, 95% confidence interval = 0.062
   Mean pos. pred. val. = 0.081, 95% confidence interval = 0.103
   Mean AUROC = 0.686, 95% confidence interval = 0.107

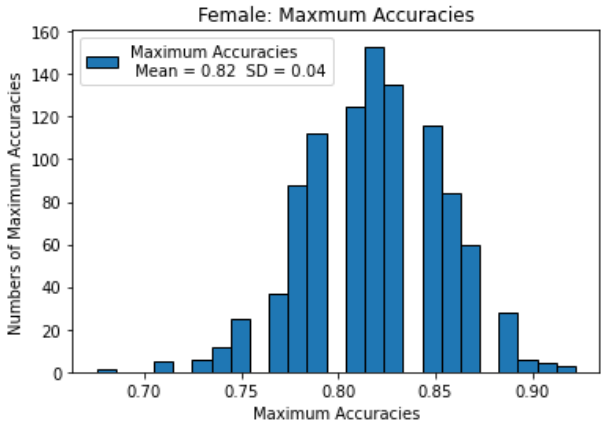


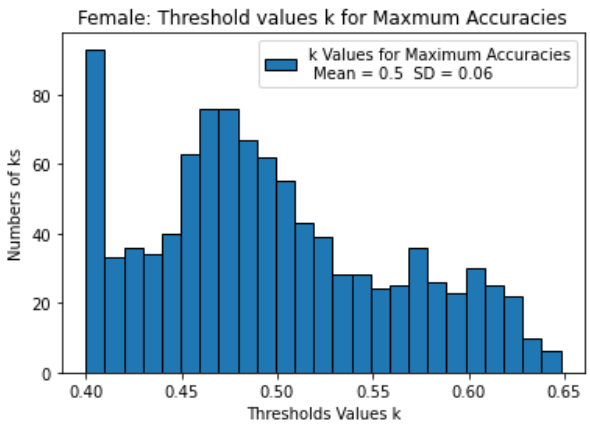


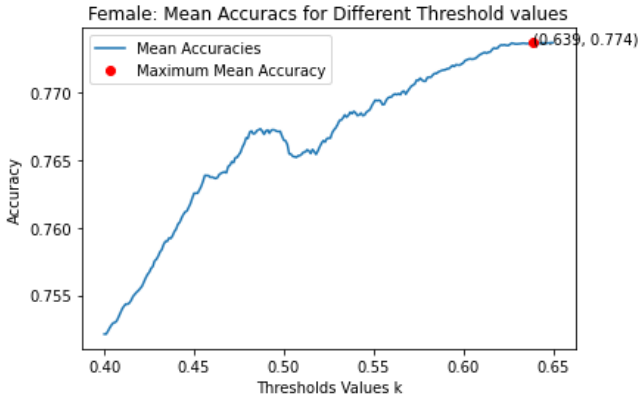


Averages for Female Examples 1000 trials with k=0.5

Mean weight of C1 = 1.415, 95% confidence interval = 0.259
Mean weight of C2 = 0.41, 95% confidence interval = 0.214
Mean weight of C3 = -1.824, 95% confidence interval = 0.193
Mean weight of age = -0.016, 95% confidence interval = 0.011
Mean weight of male gender = 0.0, 95% confidence interval = 0.0
Mean accuracy = 0.766, 95% confidence interval = 0.084
Mean sensitivity = 0.857, 95% confidence interval = 0.142
Mean specificity = 0.766, 95% confidence interval = 0.084
Mean pos. pred. val. = 0.857, 95% confidence interval = 0.142
Mean AUROC = 0.827, 95% confidence interval = 0.093







And the results for the zScaling and iScaling of both Male and Female examples……

6. For the same data examples, use **k-nearest neighbors** (**k-NN**) classifier to predict the labels of the test-set from the training-set and generate the confusion matrix for the predictions. First use k=3 to predict and generate the statistics. Then use **n-fold cross validation** to find the

proper k value for maximum accuracy. Use this k value to predict the labels of the test-set, generate the statistics of the prediction, and compare it to the result of predictions by using k=3. The results should look like:

k-NN Prediction for Survive with k=3:

TP,FP,TN,FN = 62 23 95 29

	TP	FP
Confusion Matrix is:	62	23
	95	29
	TN	FN

Accuracy = 0.751
Sensitivity = 0.681
Specificity = 0.805
Pos. Pred. Val. = 0.729

Using n-fold cross validation to find proper k for k-NN Prediction

K for Maximum Accuracy is: 5

TP, FP, TN, FN = 57 15 103 34

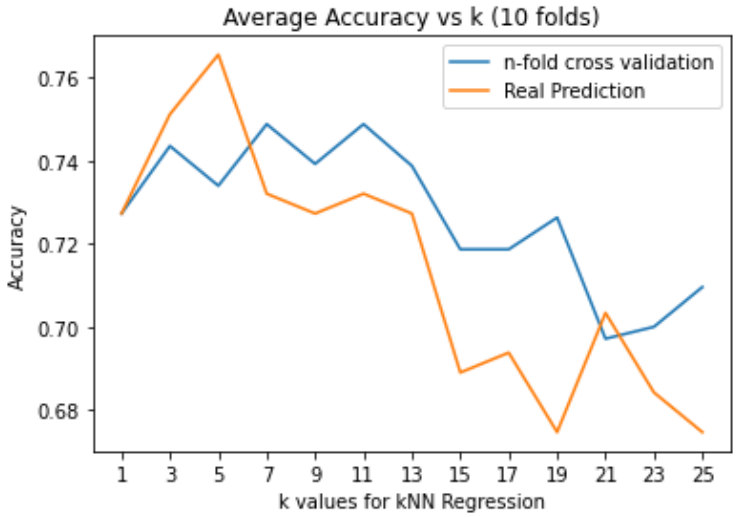
	TP	FP
Confusion Matrix is:	57	15
	103	34
	TN	FN

Accuracy = 0.766
Sensitivity = 0.626
Specificity = 0.873
Pos. Pred. Val. = 0.792

Predictions with maximum accuracy k: 5

Cross Validation Accuracies is: 0.7578947368421052

Predicted Accuracies is: 0.7655502392344498



7. The bizarre idea to predict male examples and female examples separately and combine their statistics is applied to the k-NN classifier too. Repeat the same work of step 6 without n-fold cross validation. Just use k=3 and output the similar results.

Try to predict male and female separately and combined with k=3:

For Male:

TP, FP, TN, FN = 7 7 100 17

Confusion Matrix is:	TP	FP
	7	7
	100	17
	TN	FN

Accuracy = 0.817
Sensitivity = 0.292
Specificity = 0.935
Pos. Pred. Val. = 0.5

For Female:

TP, FP, TN, FN = 47 13 9 8

Confusion Matrix is:	TP	FP
	47	13
	9	8
	TN	FN

Accuracy = 0.727
Sensitivity = 0.855
Specificity = 0.409
Pos. Pred. Val. = 0.783

Combined Predictions Statistics:

TP,FP,TN,FN = 54 20 109 25

Confusion Matrix is:	TP	FP
	54	20
	109	25
	TN	FN

Accuracy = 0.784
Sensitivity = 0.684
Specificity = 0.845
Pos. Pred. Val. = 0.73