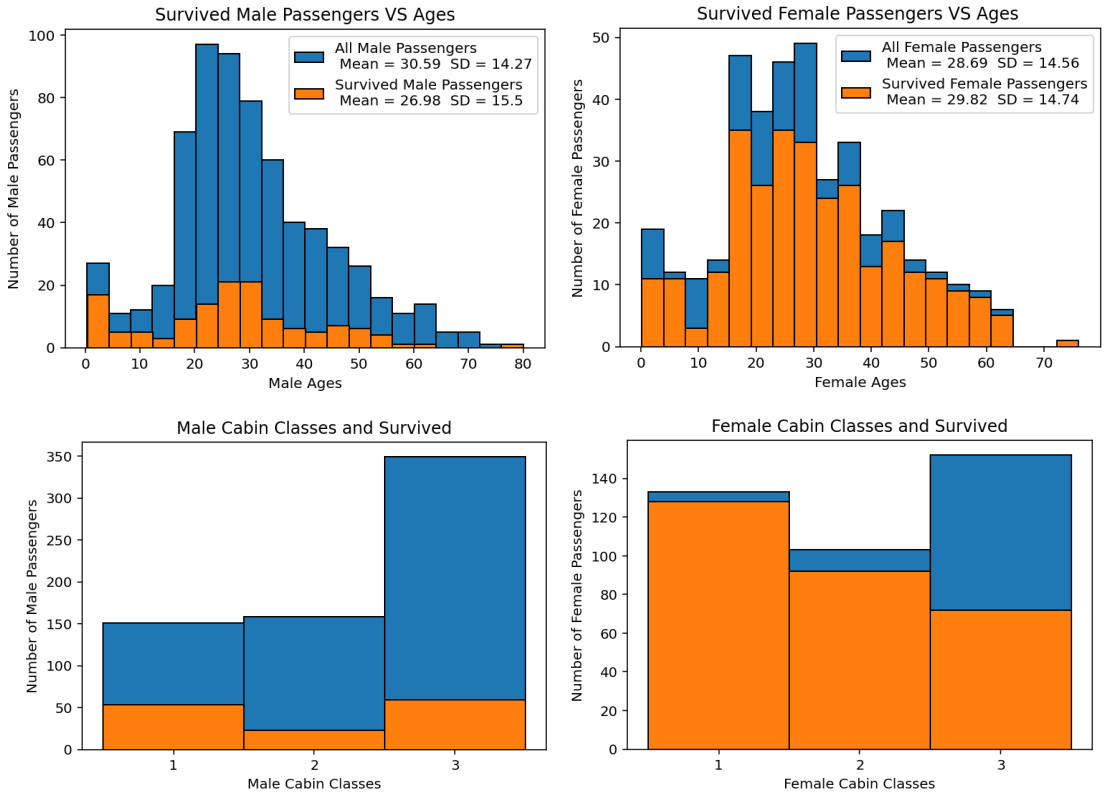
On the morning of April 15, 1912, the RMS Titanic hit an iceberg and sank in the North Atlantic. Of the roughly 1,300 passengers on board, 832 perished in the disaster. There were many factors contributing to the disaster, including navigational error, inadequate lifeboats, and the slow response of a 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 both **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 in the final exercise:

1. First, read in the file and built examples of passengers with proper feature vector for features: cabin class (1st, 2nd, or 3rd hint: use 1,0,0 for the first-class passengers and etc.), age, gender, whether the passenger survived. The feature survived is for the prediction label (use 1 for survived).
2. Separate the examples into male and female examples respectively and find the statistics of the number of passengers in each cabin class and the number of passengers survived. 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 chap2415.py). 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 chap2415.py), building and evaluate a classifier model using threshold probability k=0.5 for each split, and then collecting and reporting mean values of weights for each feature, mean value of intercepts (returned by model.intercept_[0]), and 95% confidence intervals. **b.)** For each split, after finding the model, use the same model to find the threshold probability k value (hint: between 0.5-0.65) that yields the maximum prediction accuracy on the test-set. 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 results of this step should look like:

Logistic Regression:
Averages for all examples 1000 trials with threshold k=0.5
Mean weight of C1 = 1.143, 95% confidence interval = 0.114
Mean weight of C2 = -0.084, 95% confidence interval = 0.1
Mean weight of C3 = -1.059, 95% confidence interval = 0.109
Mean weight of age = -0.033, 95% confidence interval = 0.006
Mean weight of gender = -2.412, 95% confidence interval = 0.151
Mean intercept of fitted model = 2.242 , 95% confidence interval = 0.239
Mean accuracy = 0.781, 95% confidence interval = 0.05
Mean sensitivity = 0.702, 95% confidence interval = 0.092
Mean specificity = 0.781, 95% confidence interval = 0.05
Mean pos. pred. val. = 0.702, 95% confidence interval = 0.092
Mean AUROC = 0.838, 95% confidence interval = 0.051

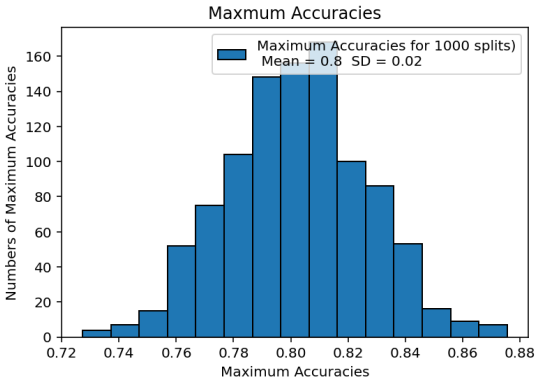
################ For your information ##
Understanding model.coef_ (feature weights) and model.intercept_ values in relation to testSet prediction:
The coef_, along with the intercept_ (bias term) values, define the linear combination that is passed through the logistic (sigmoid) function to produce the predicted probability. Specifically, for a given sample X, the log-odds of the positive class are calculated as:

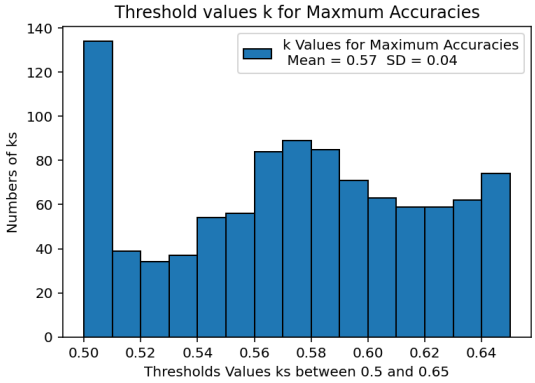
```
log_odds = intercept_ + (coef_[0] * x[0]) + (coef_[1] * x[1]) + ... \
   + (coef_[n_features-1] * x[n_features-1])
```

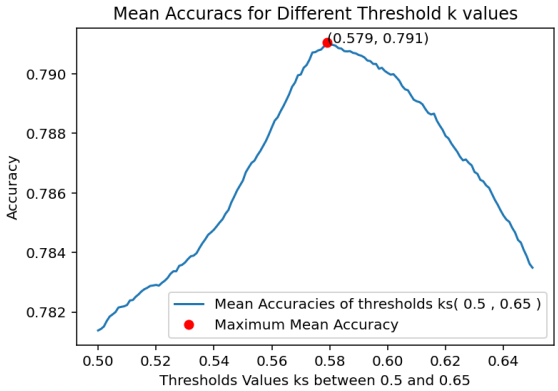
Then, the probability of the positive class is calculated as:

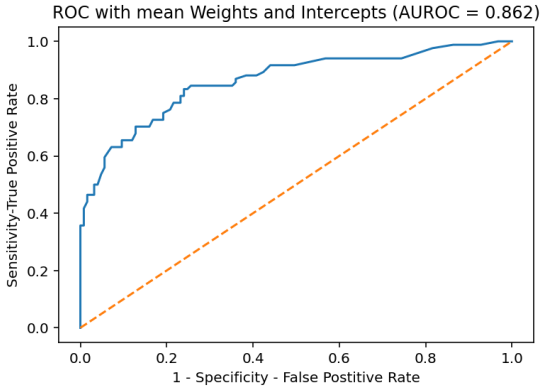
```
probability = 1 / (1 + exp(-log_odds))
```

While you can manually perform these calculations using coef_ and intercept_, it is generally recommended to use model.predict() or model.predict_proba() as they handle these computations efficiently and consistently.
But for this final exercise, manual prediction is required for the ROC chart by using mean feature coefficients and the mean intercept values
##









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 repeat step 3 again using both scaled examples. The result should look like this:

   Logistic Regression with zScaling:

   Averages for all examples (zScaling) 1000 trials with threshold k=0.5

   Mean weight of C1 = 1.138, 95% confidence interval = 0.118

   Mean weight of C2 = -0.082, 95% confidence interval = 0.098

   Mean weight of C3 = -1.059, 95% confidence interval = 0.118

   Mean weight of age = -0.475, 95% confidence interval = 0.086

   Mean weight of gender = -2.409, 95% confidence interval = 0.152

   Mean intercept of fitted model = 1.246 , 95% confidence interval = 0.111

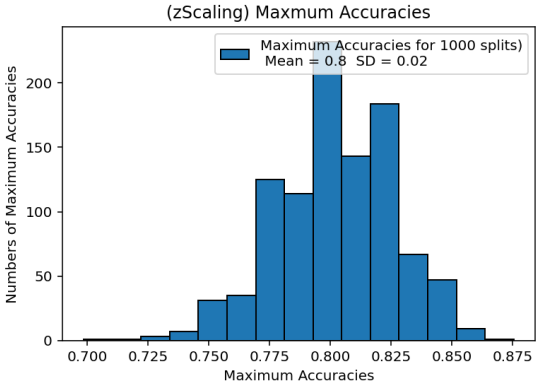
   Mean accuracy = 0.782, 95% confidence interval = 0.052

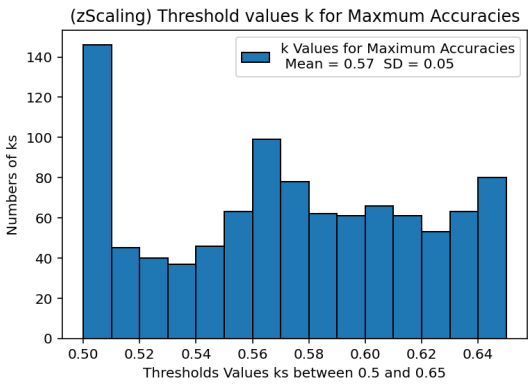
   Mean sensitivity = 0.703, 95% confidence interval = 0.091

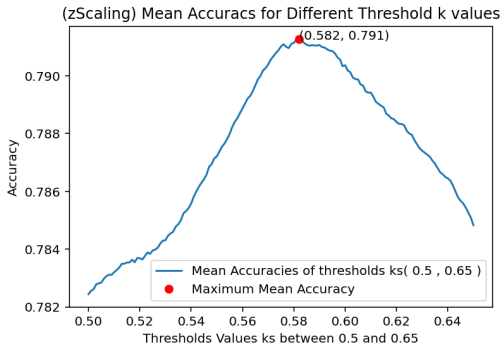
   Mean specificity = 0.782, 95% confidence interval = 0.052

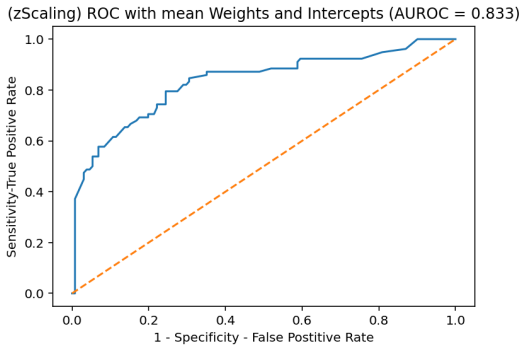
   Mean pos. pred. val. = 0.703, 95% confidence interval = 0.091

   Mean AUROC = 0.838, 95% confidence interval = 0.052









Logistic Regression with **iScaling**:

Averages for all examples (iScaling) 1000 trials with threshold k=0.5

Mean weight of C1 = 1.07, 95% confidence interval = 0.112

Mean weight of C2 = -0.07, 95% confidence interval = 0.099

Mean weight of C3 = -1.002, 95% confidence interval = 0.108

Mean weight of age = -2.044, 95% confidence interval = 0.388

Mean weight of gender = -2.402, 95% confidence interval = 0.146

Mean intercept of fitted model = 1.997 , 95% confidence interval = 0.192

Mean accuracy = 0.782, 95% confidence interval = 0.051

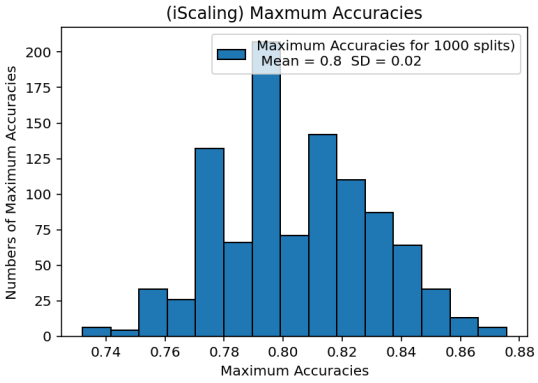
Mean sensitivity = 0.698, 95% confidence interval = 0.096

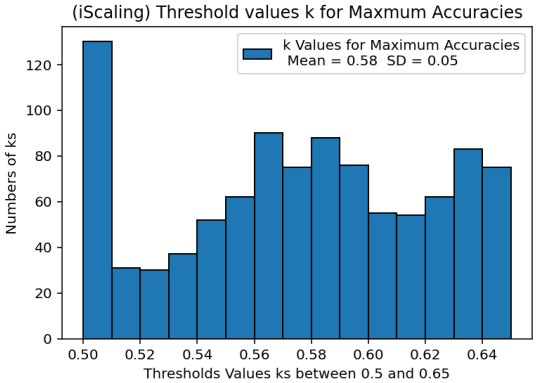
Mean specificity = 0.782, 95% confidence interval = 0.051

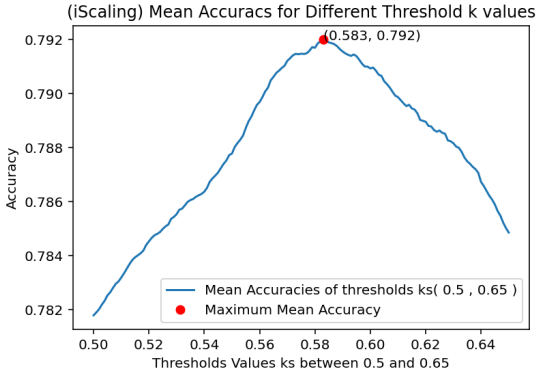
Mean pos. pred. val. = 0.698, 95% confidence interval = 0.096

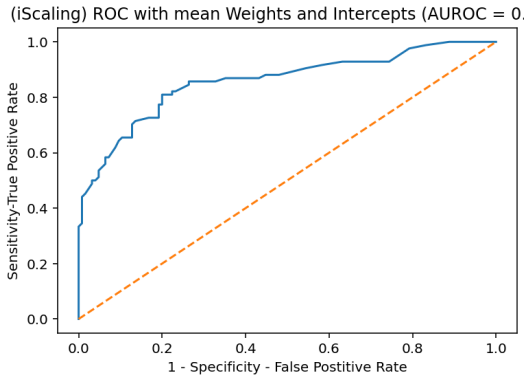
Mean AUROC = 0.837, 95% confidence interval = 0.053

(iScaling) statistics for mean maximum threshold k= 0.583



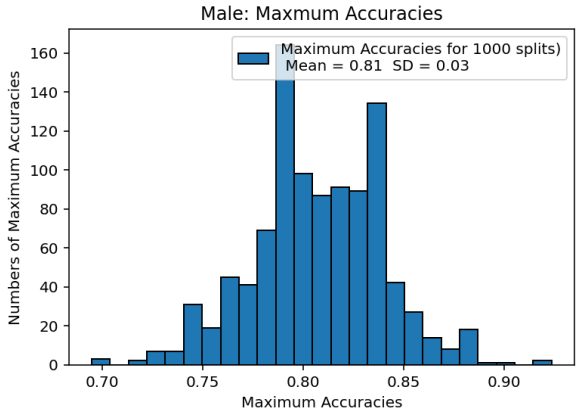


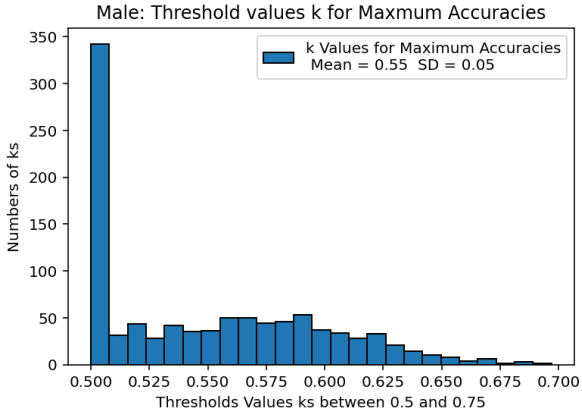


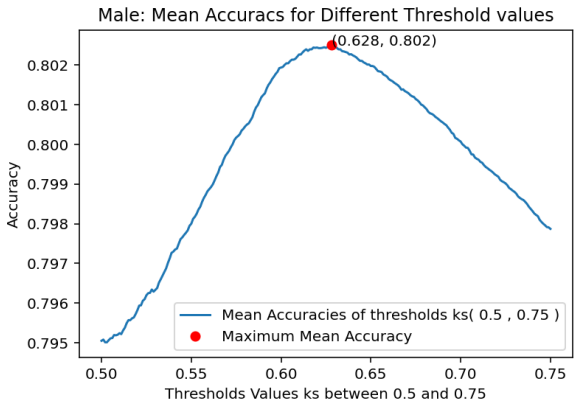


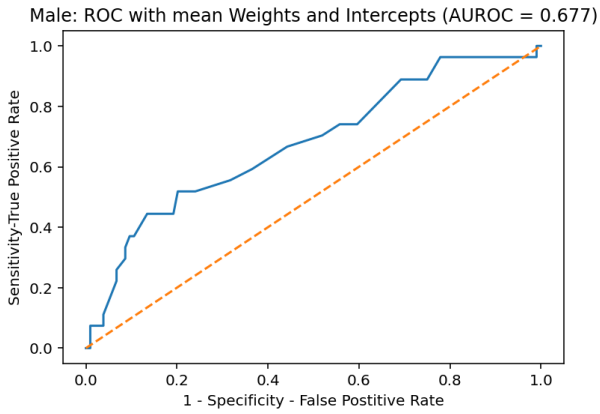
5. A bizarre idea is to predict male examples and female examples separately and combine their statistics. First, try to separate male passenger examples and female passenger examples from the whole examples. Then preform the same work from step 3 to step 4 and output the similar results and figures. But in this step, use k values between 0.5 and 0.75.

Logistic Regression With Male And Female Separated:
Averages for Male Examples 1000 trials with threshold k=0.5
Mean weight of C1 = 1.104, 95% confidence interval = 0.159
Mean weight of C2 = -0.527, 95% confidence interval = 0.144
Mean weight of C3 = -0.555, 95% confidence interval = 0.142
Mean weight of age = -0.047, 95% confidence interval = 0.009
Mean weight of gender = 0.022, 95% confidence interval = 0.07
Mean intercept of fitted model = 0.082 , 95% confidence interval = 0.251
Mean accuracy = 0.795, 95% confidence interval = 0.063
Mean sensitivity = 0.081, 95% confidence interval = 0.098
Mean specificity = 0.795, 95% confidence interval = 0.063
Mean pos. pred. val. = 0.081, 95% confidence interval = 0.098
Mean AUROC = 0.688, 95% confidence interval = 0.107



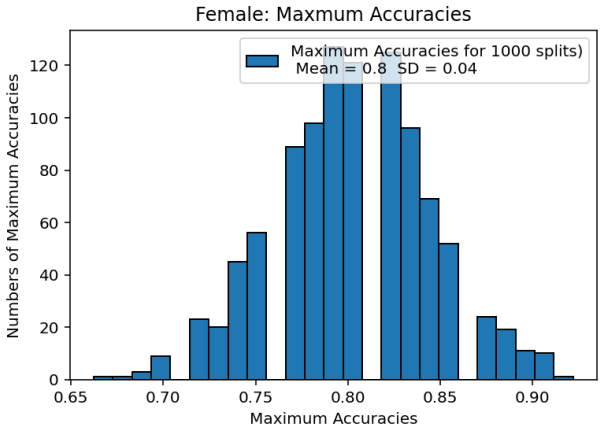


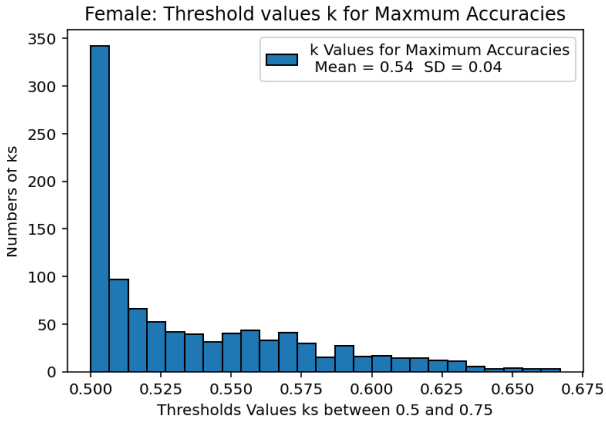


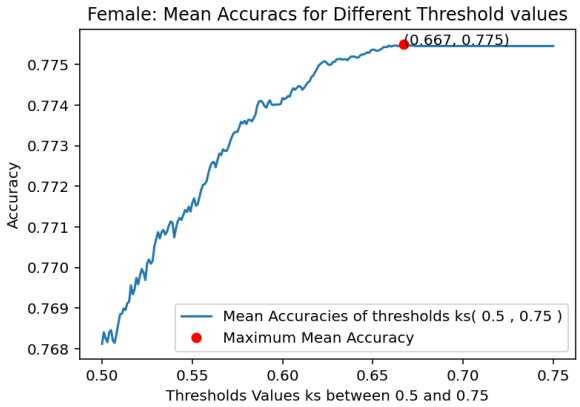


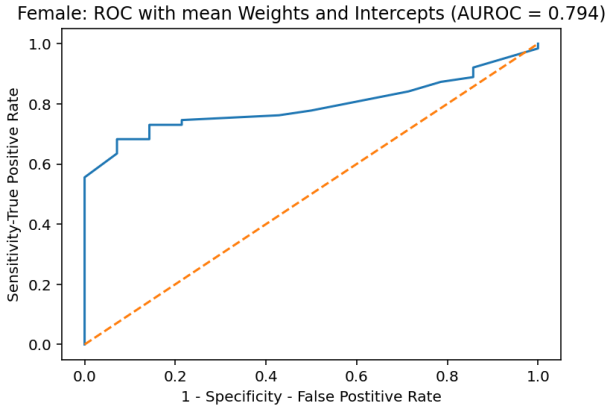
Averages for Female Examples 1000 trials with threshold k=0.5
Mean weight of C1 = 1.413, 95% confidence interval = 0.242
Mean weight of C2 = 0.403, 95% confidence interval = 0.207
Mean weight of C3 = -1.815, 95% confidence interval = 0.189
Mean weight of age = -0.015, 95% confidence interval = 0.011
Mean weight of gender = 0.0, 95% confidence interval = 0.0
Mean intercept of fitted model = 2.109 , 95% confidence interval = 0.361

Mean accuracy = 0.768, 95% confidence interval = 0.086
Mean sensitivity = 0.857, 95% confidence interval = 0.143
Mean specificity = 0.768, 95% confidence interval = 0.086
Mean pos. pred. val. = 0.857, 95% confidence interval = 0.143
Mean AUROC = 0.83, 95% confidence interval = 0.091









And the results for the zScaling and iScaling of both seperated 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 (between 3 and 25) 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 :
Cross Validation Accuracies is: [0.7602870813397129]
Predicted Accuracies is: [0.7464114832535885]
TP,FP,TN,FN = 63 31 93 22

	TP	FP
Confusion Matrix is:	63	31
	93	22
	TN	FN

Accuracy = 0.746

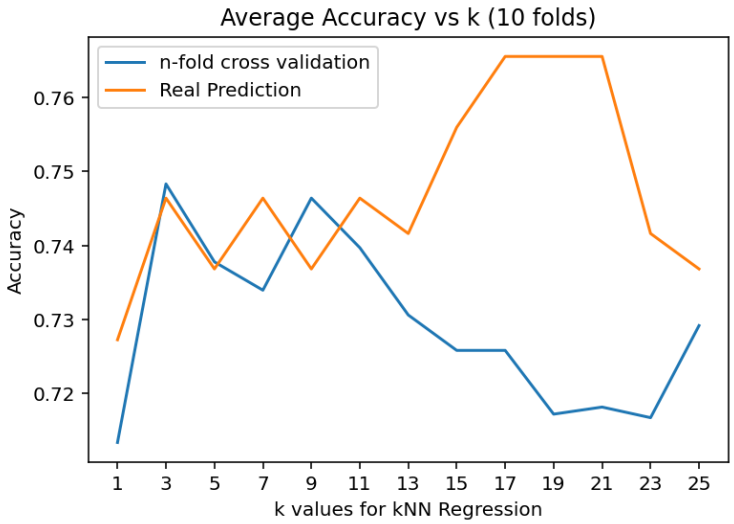
Sensitivity = 0.741
Specificity = 0.75
Pos. Pred. Val. = 0.67

Using n-fold cross validation to find proper k for k-NN Prediction
K for Maximum Accuracy is: 17 (for example only, you may get a value other than this!)
TP,FP,TN,FN = 51 15 109 34

	TP	FP
Confusion Matrix is:	51	15
	109	34
	TN	FN

Accuracy = 0.766
Sensitivity = 0.6
Specificity = 0.879
Pos. Pred. Val. = 0.773

Predictions with maximum accuracy k: 17
Cross Validation Accuracies is: [0.7349282296650717]
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 for both male and female passengers without using n-fold cross validation. Just use k=3 and output the similar results like:

Try to predict male and female separately and combine

For Male:
Cross Validation Accuracies is: [0.7587786259541984]
Predicted Accuracies is: [0.7938931297709924]
TP,FP,TN,FN = 6 5 98 22

	TP	FP
Confusion Matrix is:	6	5
	98	22
	TN	FN

Accuracy = 0.794
Sensitivity = 0.214
Specificity = 0.951
Pos. Pred. Val. = 0.545

For Female:
Cross Validation Accuracies is: [0.7688311688311689]
Predicted Accuracies is: [0.7012987012987013]
TP,FP,TN,FN = 42 14 12 9

	TP	FP
Confusion Matrix is:	42	14
	12	9
	TN	FN

Accuracy = 0.701
Sensitivity = 0.824
Specificity = 0.462
Pos. Pred. Val. = 0.75

Combined Predictions Statistics:
TP,FP,TN,FN = 48 19 110 31

	TP	FP
Confusion Matrix is:	48	19
	110	31
	TN	FN

Accuracy = 0.76
Sensitivity = 0.608
Specificity = 0.853
Pos. Pred. Val. = 0.716