Machine Learning Assignment 3

(1st Data set: # id:5-10-5-0)
(2nd Data set: # id:5--10-5-0)

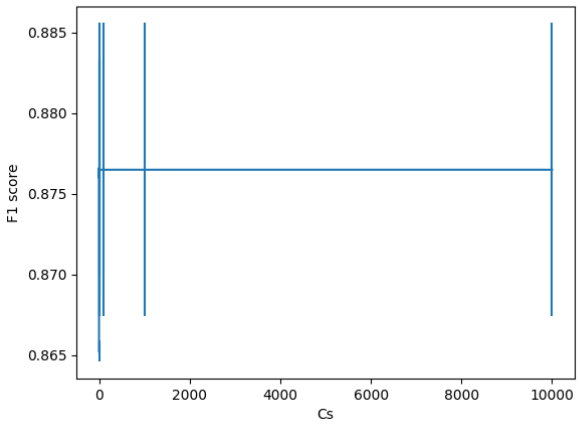
(i) (a)

Firstly, I will make the assumption that the best weight (C) for each set of polynomial features and corresponding highest order i.e degree is different so my approach will involve using cross validation to determine the weight for each degree and then using cross validation again to compare the degrees and its best weight.
The “best” in this case is the model with the highest average f1 score after cross validation. I am using f1 score because it's a good metric for comparing the performance of multiple classifiers which is what I will have to do for part (e).

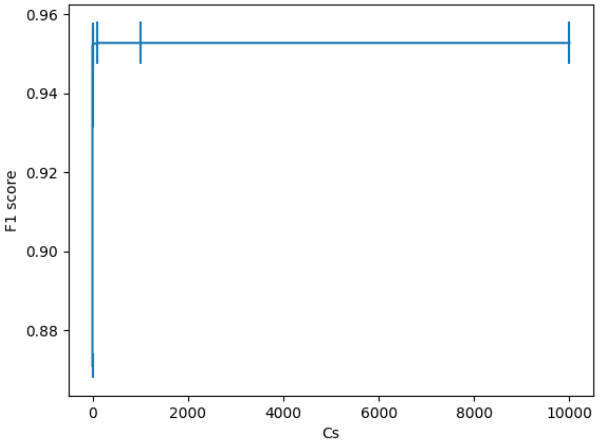
Justification for range of C values:
The first and last values are chosen because I want to see the performance of the model when the weight has very high influence(0.01) and very low influence (10000)
The values in between are factors of 10 of the previous value because it is conventional to do so and this convention is to avoid computing too many values for C.

Justification for degree values:
The input data is values less than 1 so the higher the polynomial, the lower the feature will be and the more floating point inaccuracy I introduce from doing so and the more computationally heavy my algorithm will be. From testing, the maximum value will be 6 because the model doesn’t perform any better after 5 and the starting degree will be 1 (no feature engineering) and I will increment by 1.

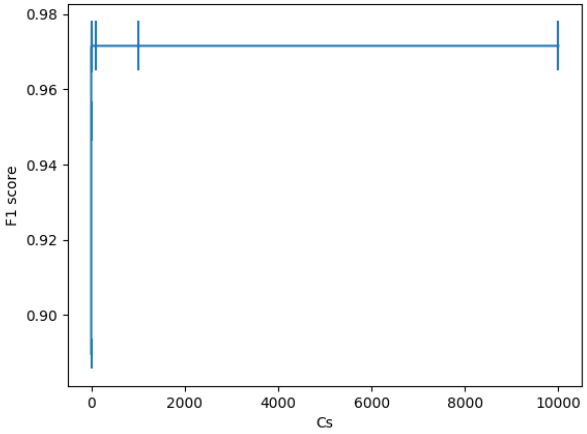
After implementing my approach, these are the cross validation plots, comparing C for each degree.



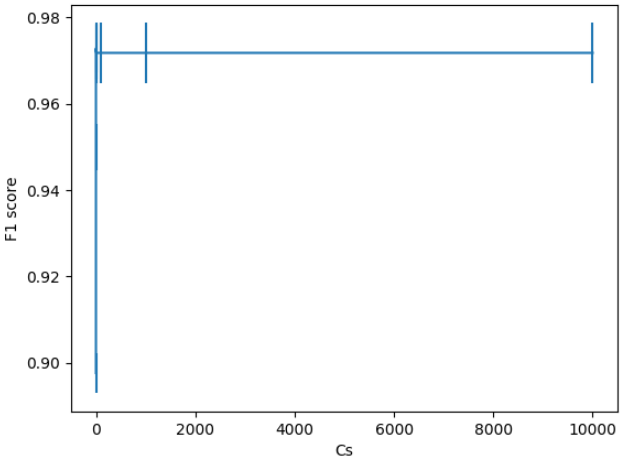
Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 1.
Mean: 0.8750



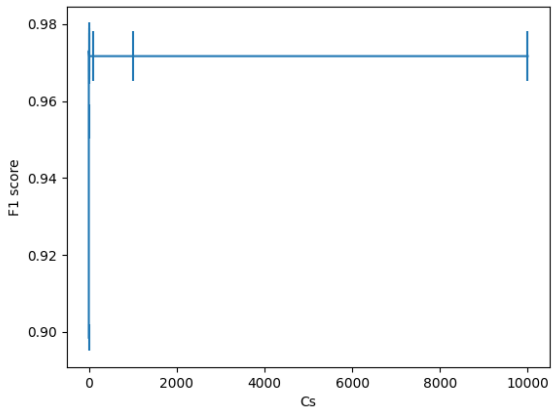
Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 2.
Mean: 0.9404



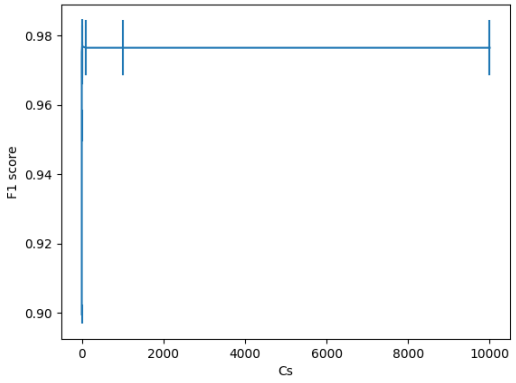
Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 3.
Mean: 0.9586



Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 4.
Mean: 0.9600



Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 5.
Mean: 0.9607

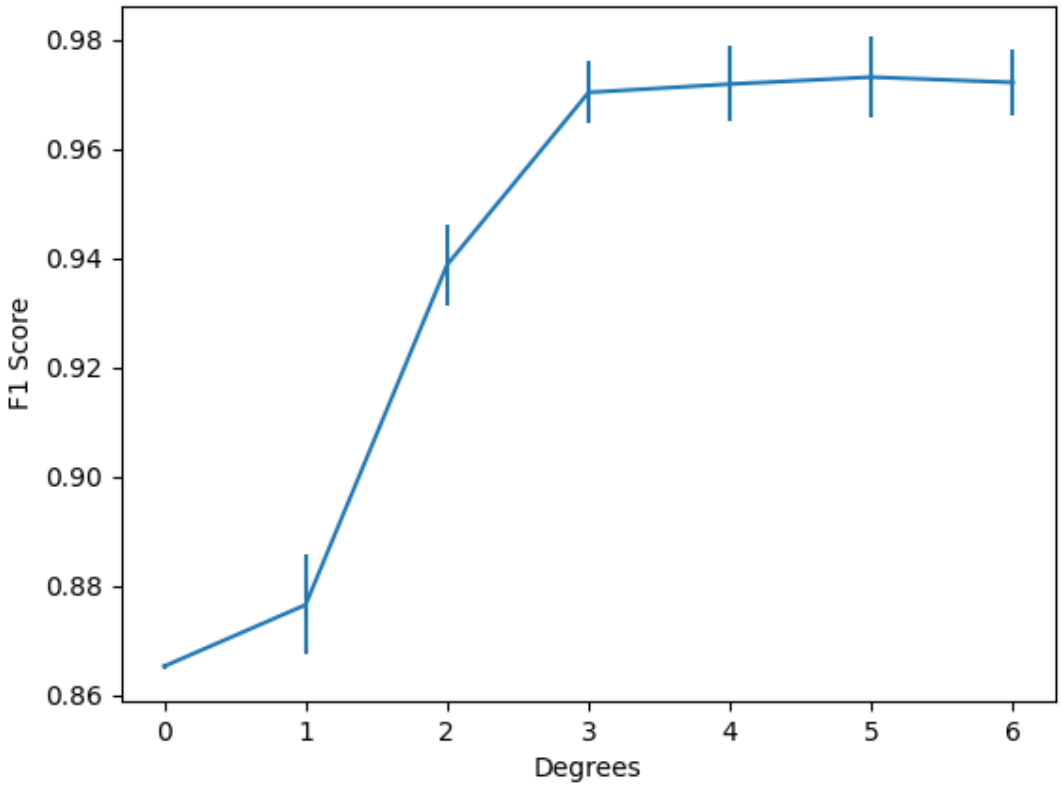


Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 6.
Mean: 0.9635

The mean given to each plot is the mean of the mean of the f1 score for each model after cross validation. The mean indicates that the performance increases as the degree increases.

From the graphs, we can see that for some degree, their performance either increases or decreases which confirms my assumption that the best weight will be different for each degree but the change in change in performance is very slight. After a certain point, the performance doesn't get any better so it is safe to choose the lower values of C to avoid overengineering.

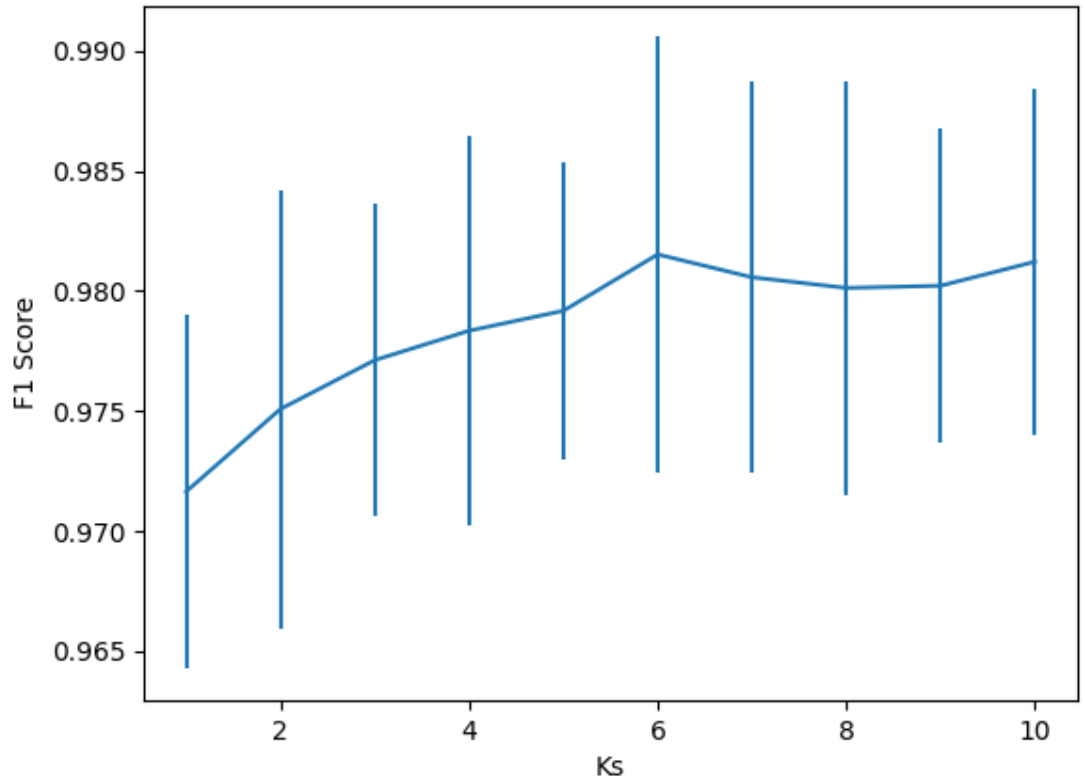
After choosing the best value of C for each degree, I then use cross validation to compare the performance of each degree.



Each degree is a model trained on a set of polynomial features with its highest order set to that degree and the F1 score is the mean of the F1 score for each model after performing cross validation.
From the graph we can see that the best performing degree is 5. It is not shown in the graph, but its weight is 1 which indicates that the model performs better with a large L_2 penalty.

(b)

The range of K values that I choose are to get the full extent of how these can models can perform so the range is from the lowest possible value up to a point where the performance begins to decline so from 1 to to 10.



Comparing the average F1 score for each model and its corresponding value for K.

From the graph we can see that the performance steadily increases up to the value of 6 and then declines but increases again but it doesn't perform better than 6. Anyways, choosing a higher value might make the model more susceptible to overengineering.

(c)

Logistic Regression

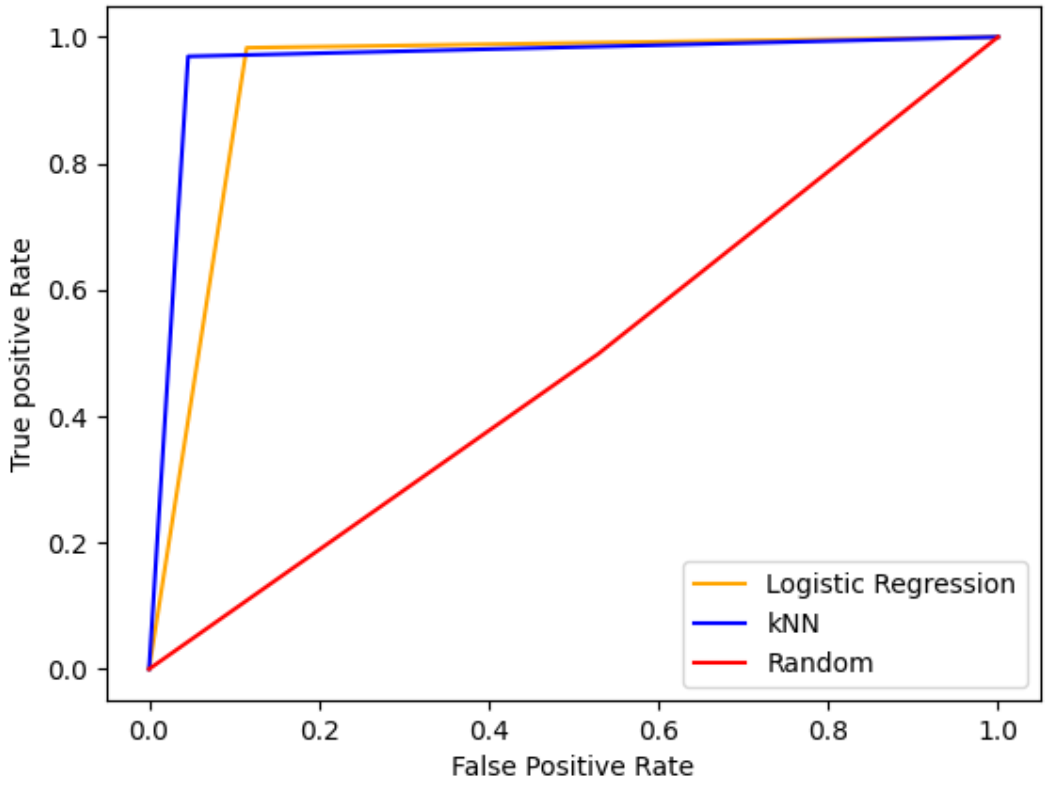
		Actual	
		True	False
Predicted	True	288	10
	False	5	77

kNN

		Actual	
		True	False
Predicted	True	284	4
	False	9	83

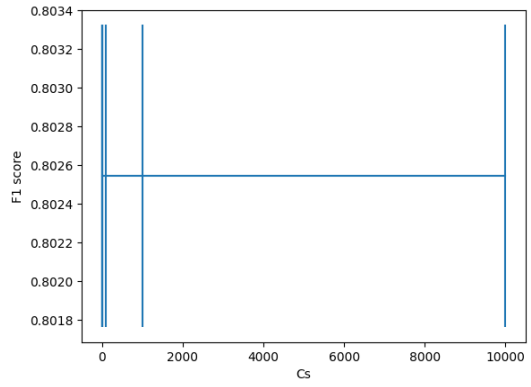
Random

		Actual	
		True	False
Predicted	True	146	46
	False	147	41

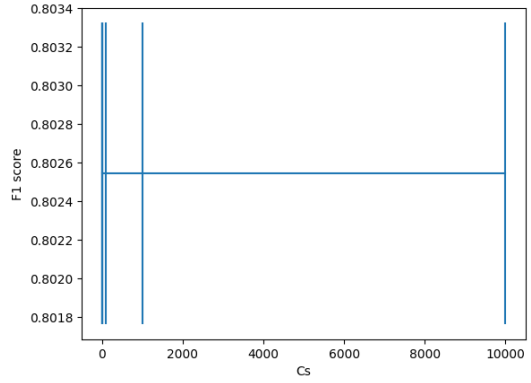


The performance of kNN is slightly better than logistic regression because it has a larger area under the ROC curve. The models perform better than random. We know this because the baseline classifier is random and it performs the worst. I would choose the kNN model over the Logistic regression model because it was both easier to compute, and more accurate in the end.

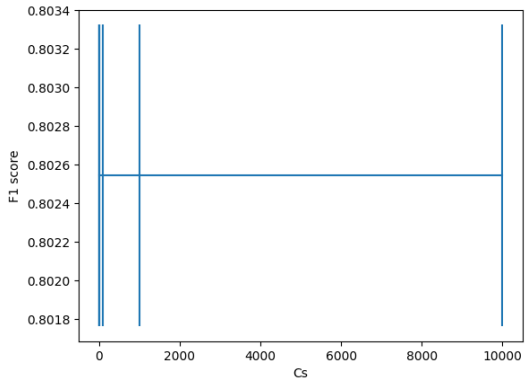
(ii) (a)



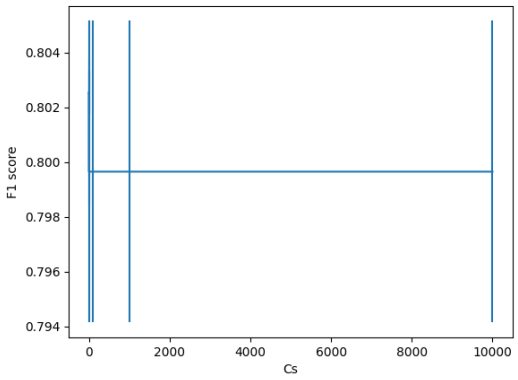
Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 1.
Mean: 0.8025



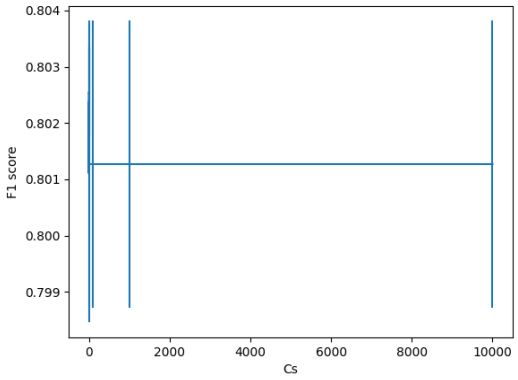
Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 2.
Mean: 0.8025



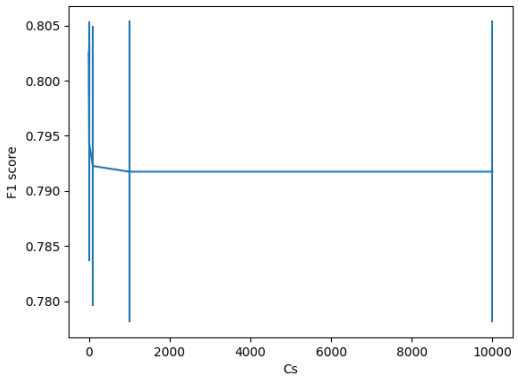
Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 3.
Mean: 0.8025



Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 4.
Mean: 0.8006

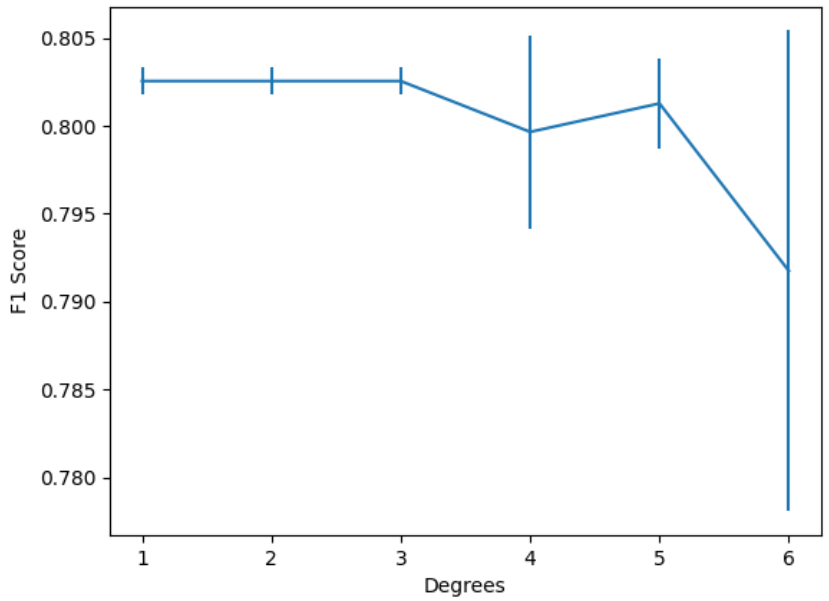


Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 5.
Mean: 0.8017



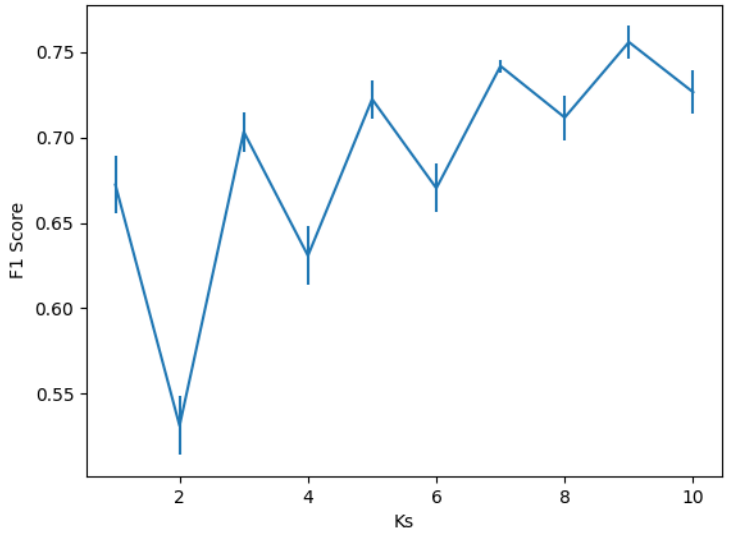
Comparing the f1 score for each value of C for a set of engineered features with its highest order set to 6.
Mean: 0.7971

The mean given to each plot is the mean of the mean of the f1 score for each model after cross validation. The mean indicates that the performance decreases as the degree increases.



Each degree is a model trained on a set of polynomial features with its highest order set to that degree and the F1 score is the mean of the F1 score for each model after performing cross validation.
From the graph we can see that the best performing degree is 1. It is not shown in the graph, but its weight is 0.001 which indicates that the model performs better with a large L_2 penalty.

(ii) (b)



Comparing the average F1 score for each model and its corresponding value for K.

From the graph we can see that the performance is chaotic but has an upward trend as k increases, the best performing value for k is 9. After that, the performance decreases.

(c)
Logistic Regression

		Actual	
		True	False
Predicted	True	203	99
	False	0	0

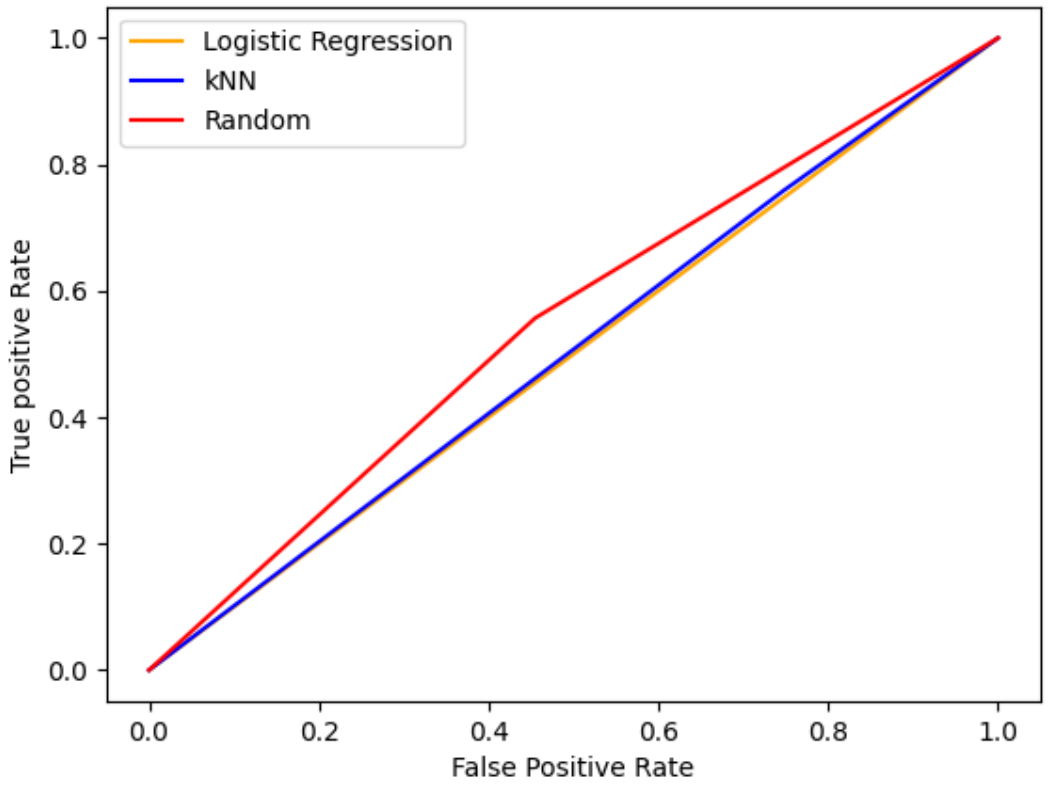
kNN

		Actual	
		True	False
Predicted	True	176	85
	False	27	14

Random

		Actual	
		True	False
Predicted	True	113	45
	False	90	54

(d)



(e)

The performance of the model is worse than the baseline classifier and the baseline classifier predicts random values of +1 and -1 which indicates that the data must be random. When it comes to choosing, the random classifier would be the best because it outperformed the other algorithms and if computation time is a concern, then the random classifier has the best performance to accuracy ratio, making it an even better choice.