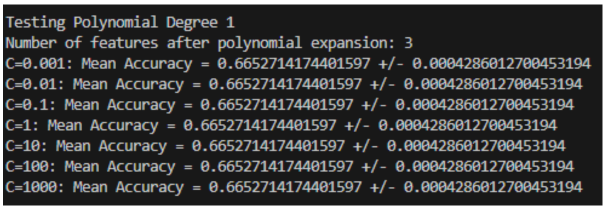
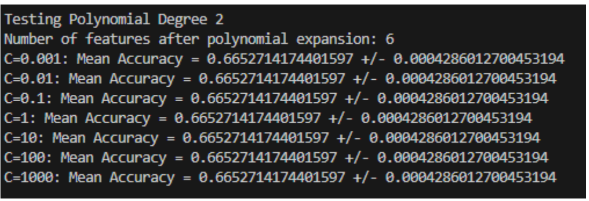
(i) Dataset 1 # id:11--22-11-1

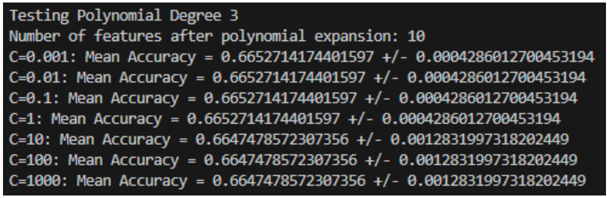
a.



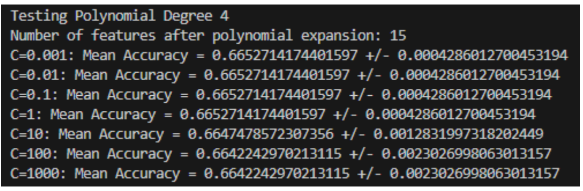
```
Testing Polynomial Degree 1
Number of features after polynomial expansion: 3
C=0.001: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.01: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.1: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=1: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=10: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=100: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=1000: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
```



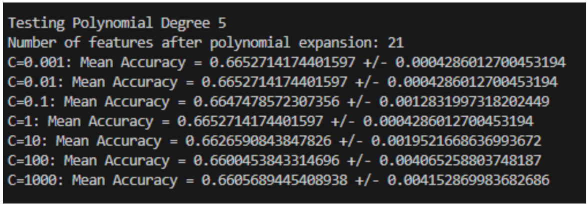
```
Testing Polynomial Degree 2
Number of features after polynomial expansion: 6
C=0.001: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.01: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.1: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=1: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=10: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=100: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=1000: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
```



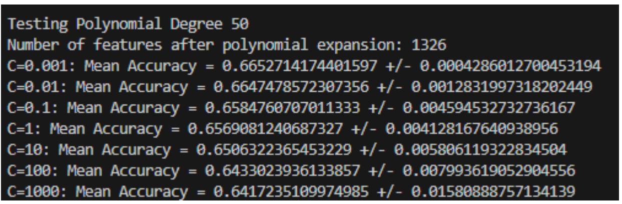
```
Testing Polynomial Degree 3
Number of features after polynomial expansion: 10
C=0.001: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.01: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.1: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=1: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=10: Mean Accuracy = 0.6647478572307356 +/- 0.0012831997318202449
C=100: Mean Accuracy = 0.6647478572307356 +/- 0.0012831997318202449
C=1000: Mean Accuracy = 0.6647478572307356 +/- 0.0012831997318202449
```



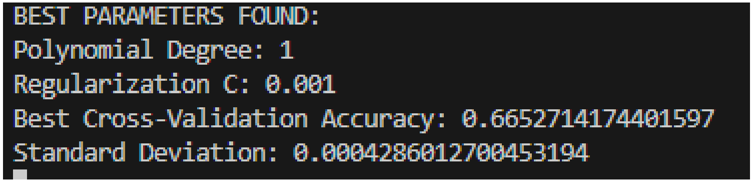
```
Testing Polynomial Degree 4
Number of features after polynomial expansion: 15
C=0.001: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.01: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.1: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=1: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=10: Mean Accuracy = 0.6647478572307356 +/- 0.0012831997318202449
C=100: Mean Accuracy = 0.6642242970213115 +/- 0.0023026998063013157
C=1000: Mean Accuracy = 0.6642242970213115 +/- 0.0023026998063013157
```



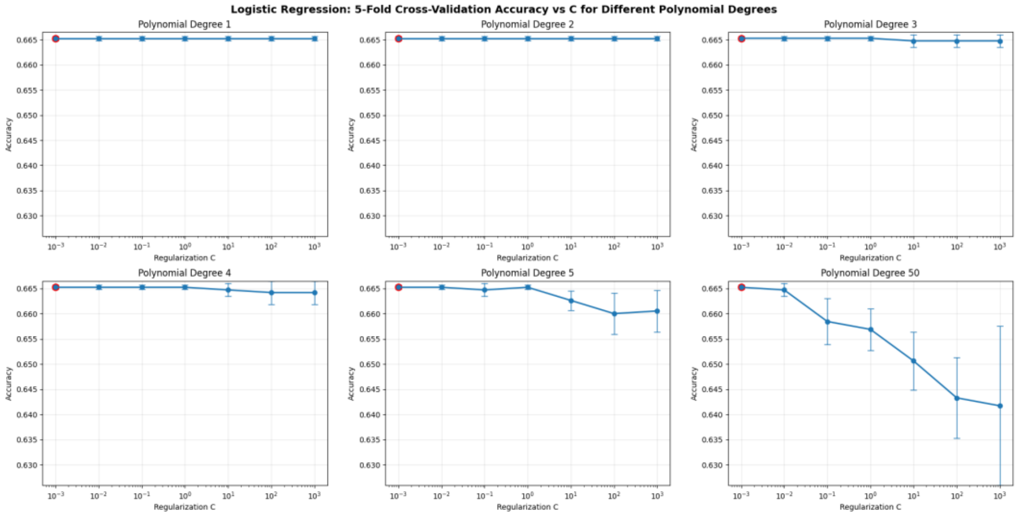
```
Testing Polynomial Degree 5
Number of features after polynomial expansion: 21
C=0.001: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.01: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.1: Mean Accuracy = 0.6647478572307356 +/- 0.0012831997318202449
C=1: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=10: Mean Accuracy = 0.6626590843847826 +/- 0.0019521668636993672
C=100: Mean Accuracy = 0.6600453843314696 +/- 0.004065258803748187
C=1000: Mean Accuracy = 0.6605689445408938 +/- 0.004152869983682686
```

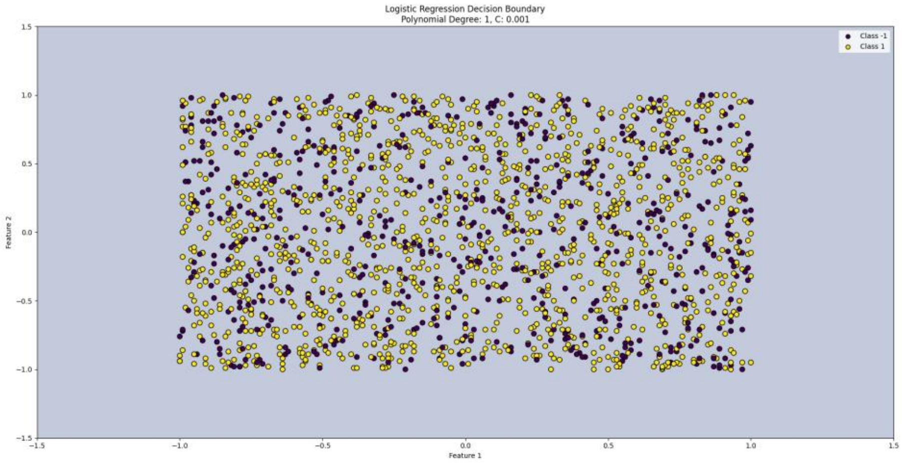


```
Testing Polynomial Degree 50
Number of features after polynomial expansion: 1326
C=0.001: Mean Accuracy = 0.6652714174401597 +/- 0.0004286012700453194
C=0.01: Mean Accuracy = 0.6647478572307356 +/- 0.0012831997318202449
C=0.1: Mean Accuracy = 0.6584760707011333 +/- 0.004594532732736167
C=1: Mean Accuracy = 0.6569081240687327 +/- 0.004128167640938956
C=10: Mean Accuracy = 0.6506322365453229 +/- 0.005806119322834504
C=100: Mean Accuracy = 0.6433023936133857 +/- 0.007993619052904556
C=1000: Mean Accuracy = 0.6417235109974985 +/- 0.01580888757134139
```

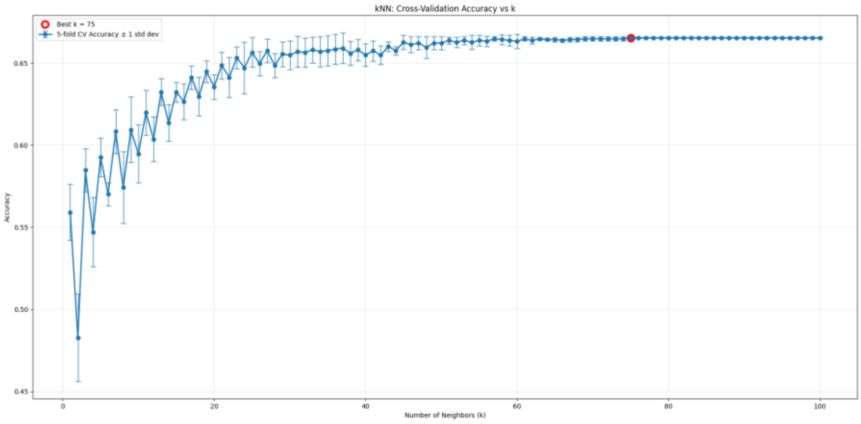


```
BEST PARAMETERS FOUND:
Polynomial Degree: 1
Regularization C: 0.001
Best Cross-Validation Accuracy: 0.6652714174401597
Standard Deviation: 0.0004286012700453194
```



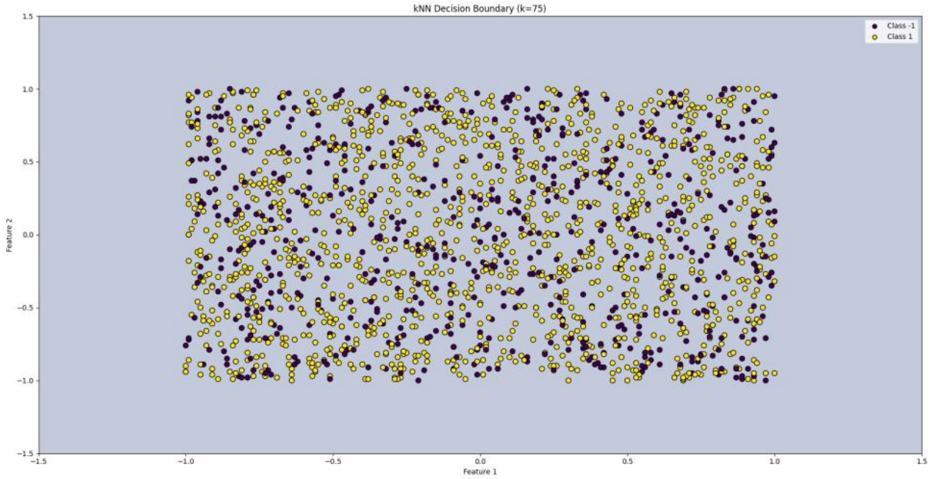


This model didn’t work well, the best classifier found was the first, as they all have equal or worse performance. Looking at the data we can tell that no possible decision boundary could do any better than a “most frequent” classifier

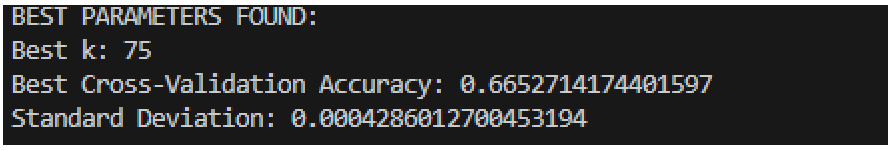


b.

The kNN classifier was again evaluated with 100 values for k, while the performance did improve for a while it plateaud at k=75

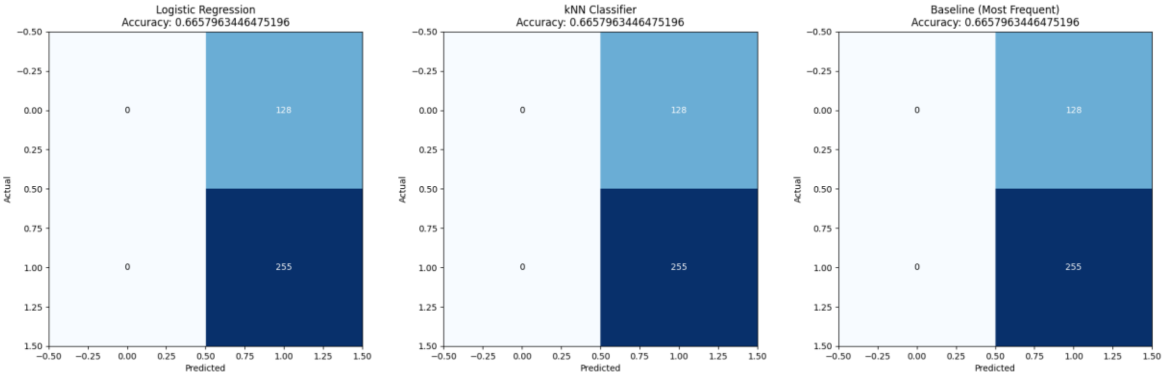


We can again see that no possible decision boundary could resolve this



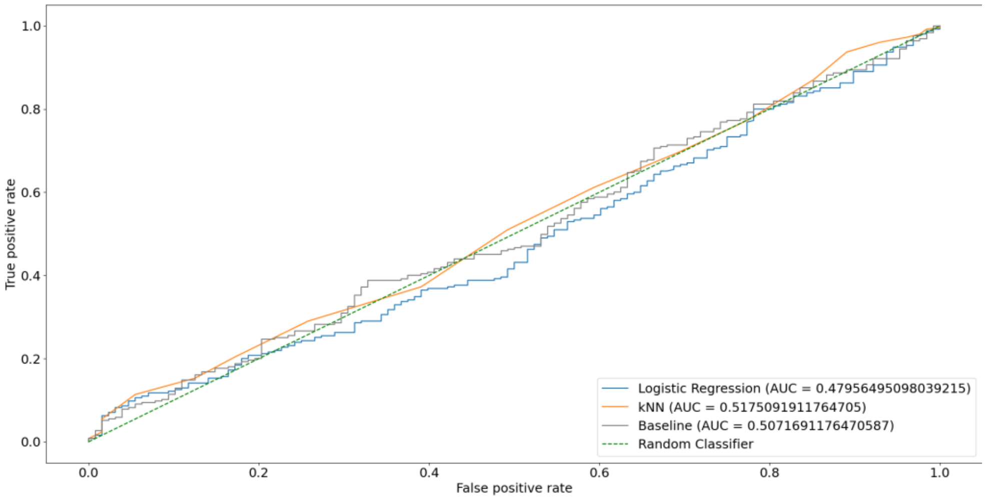
```
BEST PARAMETERS FOUND:
Best k: 75
Best Cross-Validation Accuracy: 0.6652714174401597
Standard Deviation: 0.0004286012700453194
```

The accuracy and standard deviation of the model being identical to the logistic regression model proves that both models have moved their decision boundary far from the table



c.

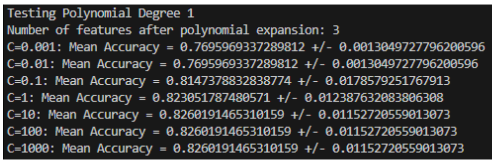
Here we can see that all 3 classifiers return identical results, with each of them predicting that all entries will be positive, allowing them to correctly guess 255/383 points



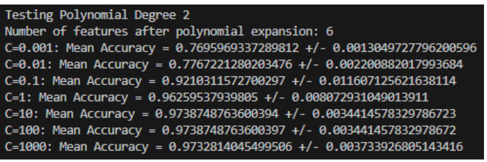
d. Again all 3 models return very similar performance on the ROC curves
   While the logistic regression slightly underperformed, the kNN and baseline slightly overperformed the random classifier, but this is subject to chance, as these models are so close to the line that their random guessing seems vary their curve enough to sway the score either way.
   I don't believe that the slight difference in performance here is indicative of anything.
e. Altogether I would recommend the "most frequent" classifier, as all the models perform so similarly, but the most frequent classifier is the easiest to implement and has the highest efficiency.

(ii) Dataset 2 # id:11--22-11-1

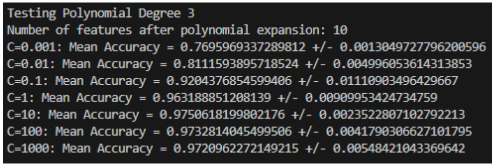
a.



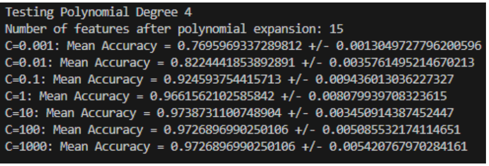
```
Testing Polynomial Degree 1
Number of features after polynomial expansion: 3
C=0.001: Mean Accuracy = 0.7695969337289812 +/- 0.0013049727796200596
C=0.01: Mean Accuracy = 0.7695969337289812 +/- 0.0013049727796200596
C=0.1: Mean Accuracy = 0.8147378832838774 +/- 0.017857925176791З
C=1: Mean Accuracy = 0.823051787480571 +/- 0.012387632083806308
C=10: Mean Accuracy = 0.8260191465310159 +/- 0.01152720559013073
C=100: Mean Accuracy = 0.8260191465310159 +/- 0.01152720559013073
C=1000: Mean Accuracy = 0.8260191465310159 +/- 0.01152720559013073
```



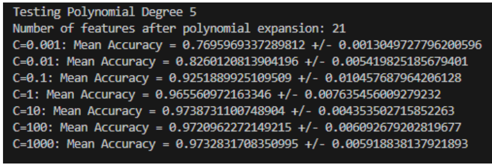
```
Testing Polynomial Degree 2
Number of features after polynomial expansion: 6
C=0.001: Mean Accuracy = 0.7695969337289812 +/- 0.0013049727796200596
C=0.01: Mean Accuracy = 0.7767221280203476 +/- 0.002200882017993684
C=0.1: Mean Accuracy = 0.9210311572700297 +/- 0.011607125621638114
C=1: Mean Accuracy = 0.96259537939805 +/- 0.008072931049013911
C=10: Mean Accuracy = 0.9738748763600394 +/- 0.003441457832978672З
C=100: Mean Accuracy = 0.9738748763600397 +/- 0.003441457832978672
C=1000: Mean Accuracy = 0.9732814045499506 +/- 0.003733926805143416
```



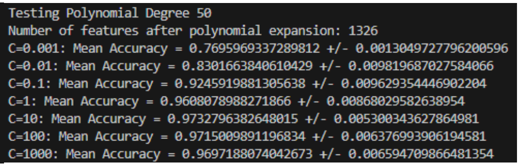
```
Testing Polynomial Degree 3
Number of features after polynomial expansion: 10
C=0.001: Mean Accuracy = 0.7695969337289812 +/- 0.0013049727796200596
C=0.01: Mean Accuracy = 0.8111593895718524 +/- 0.004996053614313853
C=0.1: Mean Accuracy = 0.9204376854599406 +/- 0.011109034964296 67
C=1: Mean Accuracy = 0.963188851208139 +/- 0.00909953424734759
C=10: Mean Accuracy = 0.9750618199802176 +/- 0.0023522807102792213
C=100: Mean Accuracy = 0.9732814045499506 +/- 0.004179030662710179 5
C=1000: Mean Accuracy = 0.9720962272149215 +/- 0.005484210433696 42
```



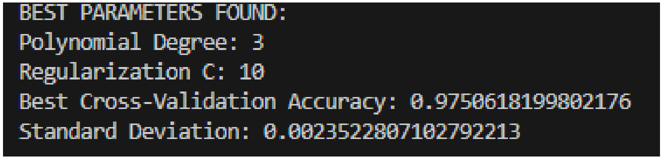
```
Testing Polynomial Degree 4
Number of features after polynomial expansion: 15
C=0.001: Mean Accuracy = 0.7695969337289812 +/- 0.0013049727796200596
C=0.01: Mean Accuracy = 0.8224441853892891 +/- 0.0035761495214670213
C=0.1: Mean Accuracy = 0.924593754415713 +/- 0.009436013036227327
C=1: Mean Accuracy = 0.9661562102585842 +/- 0.008079939708323615
C=10: Mean Accuracy = 0.9738731100748904 +/- 0.003450914387452447
C=100: Mean Accuracy = 0.9726896990250106 +/- 0.005085532174114651
C=1000: Mean Accuracy = 0.9726896990250106 +/- 0.005420767970284161
```



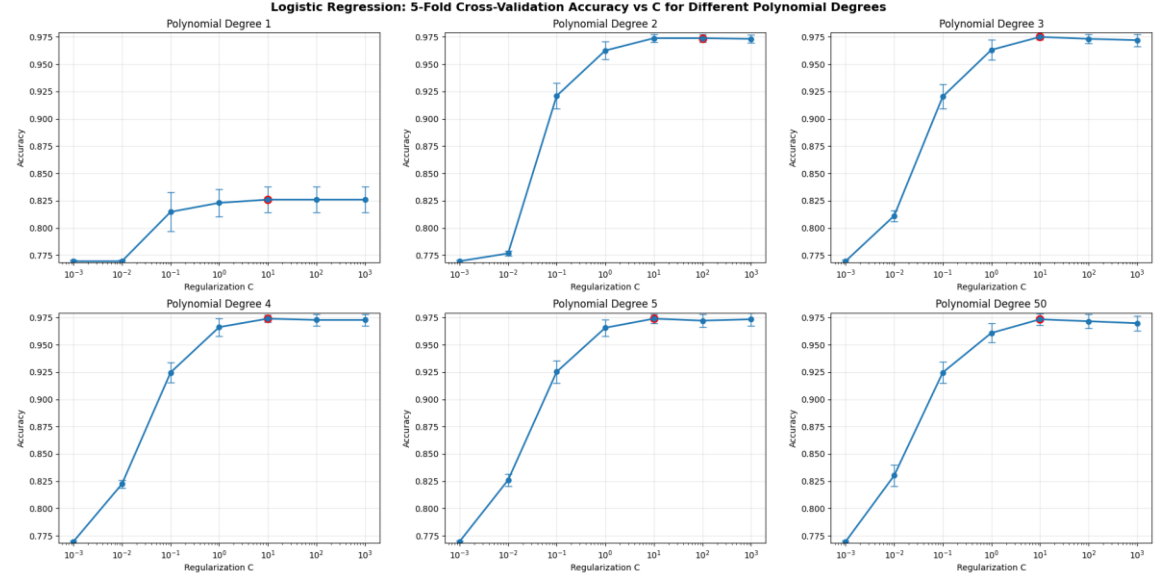
```
Testing Polynomial Degree 5
Number of features after polynomial expansion: 21
C=0.001: Mean Accuracy = 0.7695969337289812 +/- 0.0013049727796200596
C=0.01: Mean Accuracy = 0.8260120813904196 +/- 0.005419825185679401
C=0.1: Mean Accuracy = 0.9251889925109509 +/- 0.010457687964206128
C=1: Mean Accuracy = 0.965560972163346 +/- 0.007635456009279232
C=10: Mean Accuracy = 0.9738731100748904 +/- 0.004353502715852263
C=100: Mean Accuracy = 0.9720962272149215 +/- 0.006092679202819677
C=1000: Mean Accuracy = 0.9732831708350995 +/- 0.005918838137921893
```



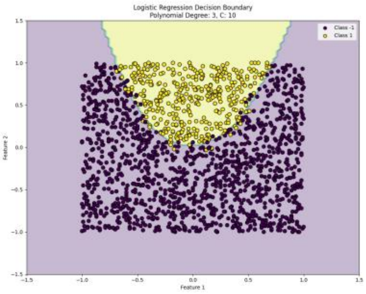
```
Testing Polynomial Degree 50
Number of features after polynomial expansion: 1326
C=0.001: Mean Accuracy = 0.7695969337289812 +/- 0.0013049727796200596
C=0.01: Mean Accuracy = 0.8301663840610429 +/- 0.009819687027584066
C=0.1: Mean Accuracy = 0.9245919881305638 +/- 0.009629354446902204
C=1: Mean Accuracy = 0.9608078988271866 +/- 0.008680295826389 54
C=10: Mean Accuracy = 0.9732796382648015 +/- 0.005300343627864981
C=100: Mean Accuracy = 0.9715009891196834 +/- 0.006376993906194581
C=1000: Mean Accuracy = 0.9697188074042673 +/- 0.006594709866481354
```



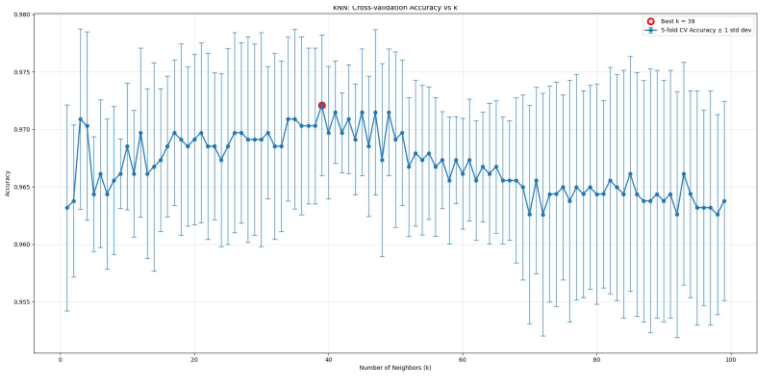
```
BEST PARAMETERS FOUND:
Polynomial Degree: 3
Regularization C: 10
Best Cross-Validation Accuracy: 0.9750618199802176
Standard Deviation: 0.0023522807102792213
```



By creating a model for each combination of polynomial degree and C-value I was able to create 42 separate models, of which the polynomial degree of 3 with C value of 10 created the best model, with an accuracy of 0.975 and a standard deviation of 0.00235.

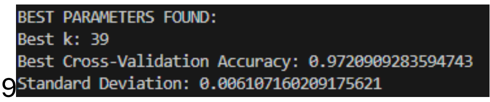


We can see that the decision boundary seems well suited with no signs of overfitting



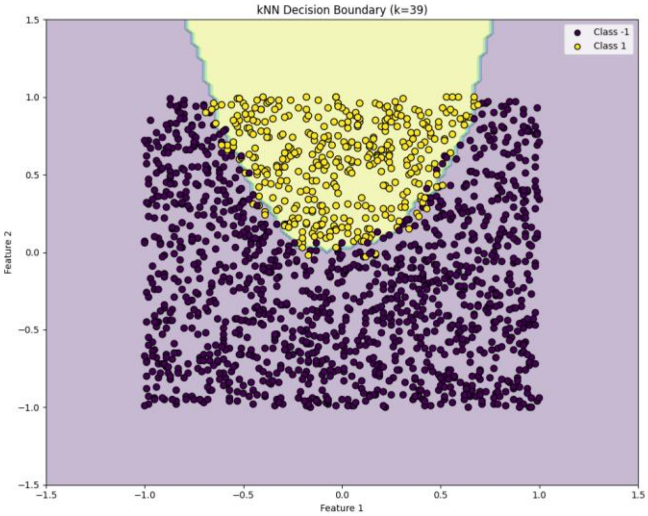
b.

The kNN classifier was evaluated with k values from 1 to 100, we can see from the graph that the best performing k-value was 39 With an accuracy of 0.972 and a standard deviation of 0.0061.

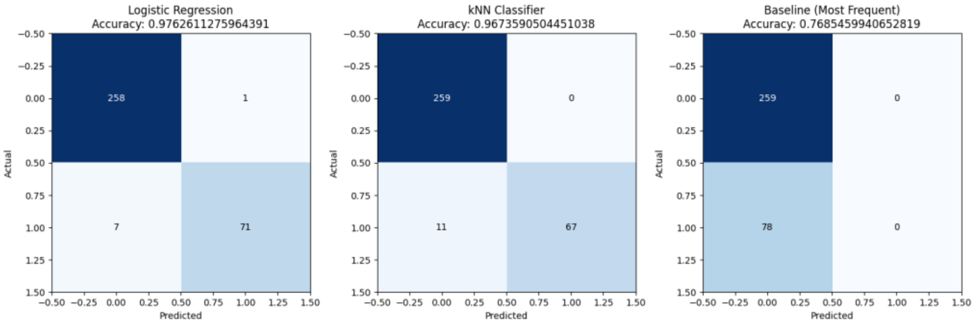


```
BEST PARAMETERS FOUND:
Best k: 39
Best Cross-Validation Accuracy: 0.9720909283594743
Standard Deviation: 0.006107160209175621
```

This model slightly underperforms compared to the the logistic regression model.



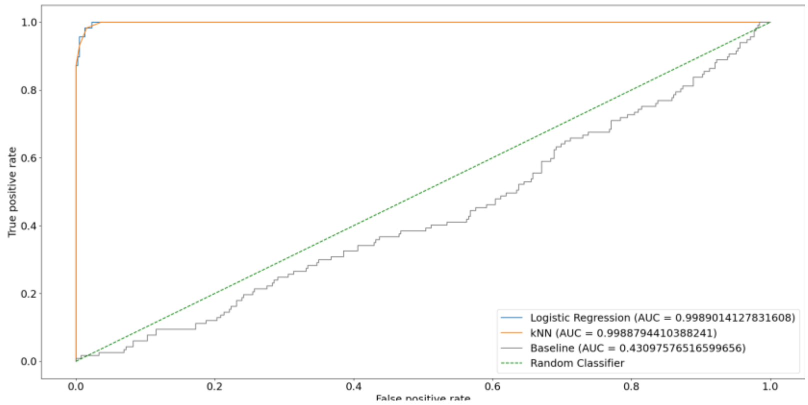
The decision boundary again seems very well suited with no signs of overfitting



c.

The 2 models were compared against each other and a baseline(most frequest) model.
I used a 20% test split from full data set.
 Logistic regression can be seen outperforming KNn again, we can see that both models significantly outperform the baseline model, making 8/11 mistakes as opposed to 78

d.



The Logistic rRegression and the kNN plot performed incredibly well with an AUC of 0.9989
The baseline classifier performed horribly with an AUC of 0.43

e. Altogether I would recommend the Logistic regression model as it has slightly higher accuracy and more understandable weights.