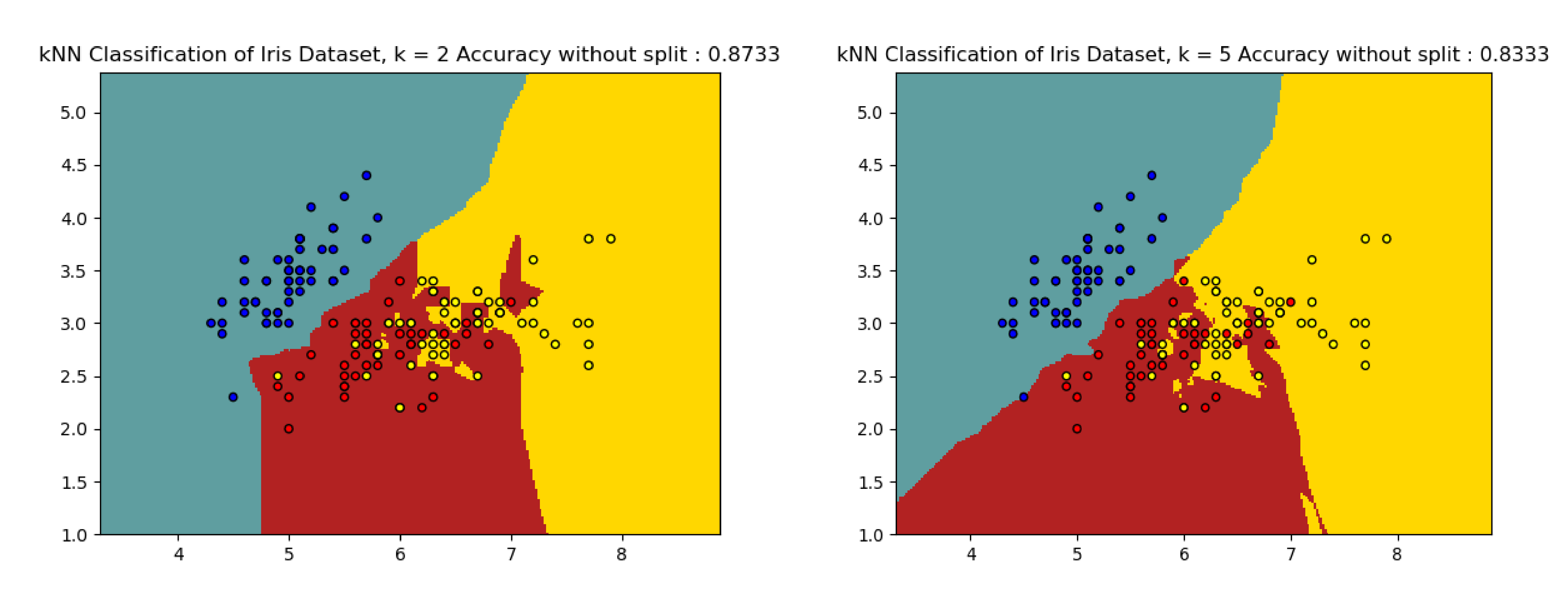
Determining Classifier Strengths

Methods

- For this project I used two of the available datasets in the scikit-learn library of python.
- The first dataset I analyzed is the famous iris dataset which is very suitable for a classification task. It has three classes and thus the task of classification here would be multi-class classification.
- The second dataset is the digits dataset from scikit-learn. This dataset is made up of 1797 8x8 images. Each image is of a hand-written digit. This dataset is highly collinear and thus the entries are correlated.
- I use 4 different kinds of algorithms: kNN, nearest centroids, QDA and logistic regression and compare their performances using the accuracy metric. I use hold-out cross-validation to obtain the results, unless otherwise stated.

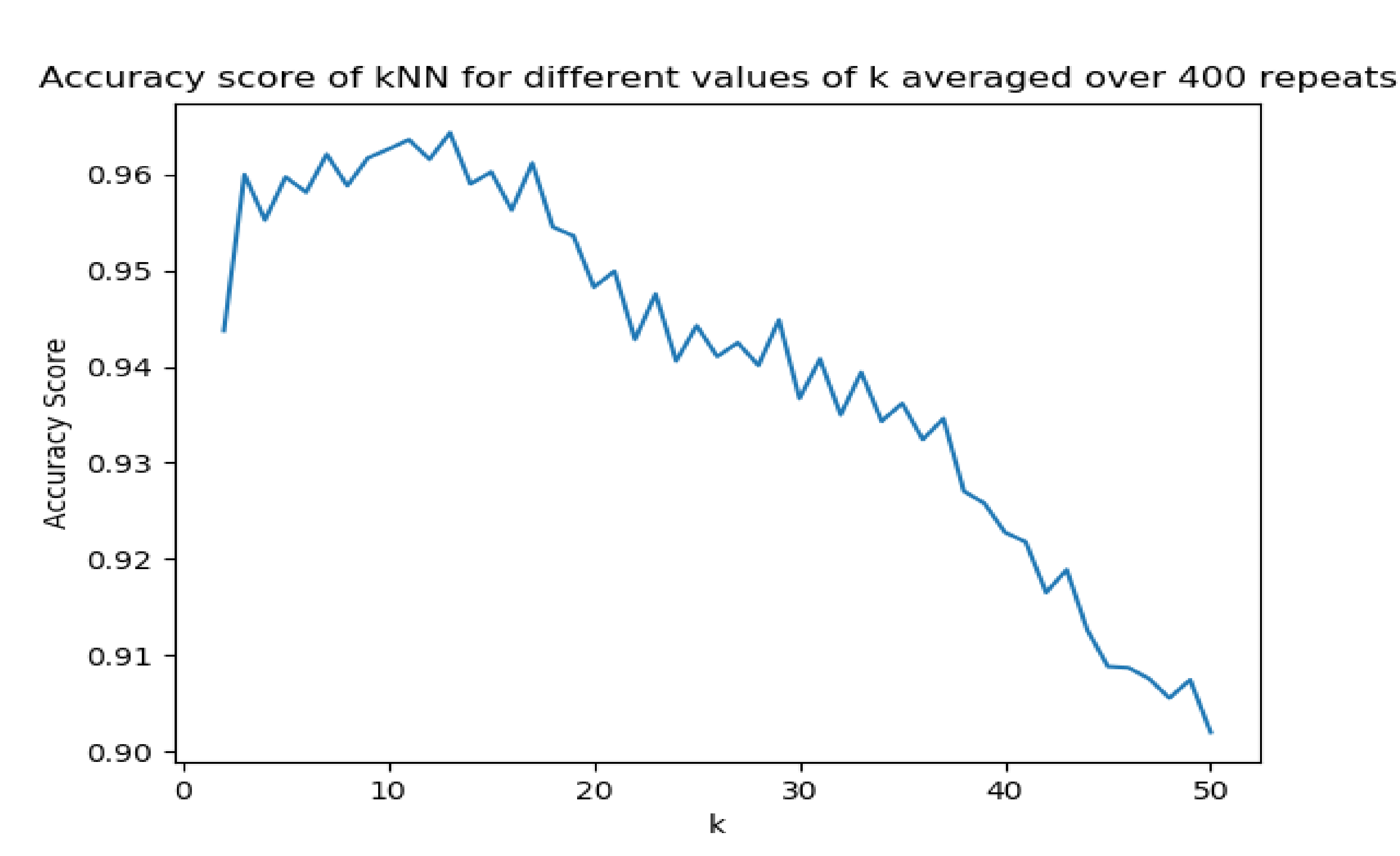
Results



kNN Classification of Iris Dataset, k = 2 Accuracy without split : 0.8733

kNN Classification of Iris Dataset, k = 5 Accuracy without split : 0.8333

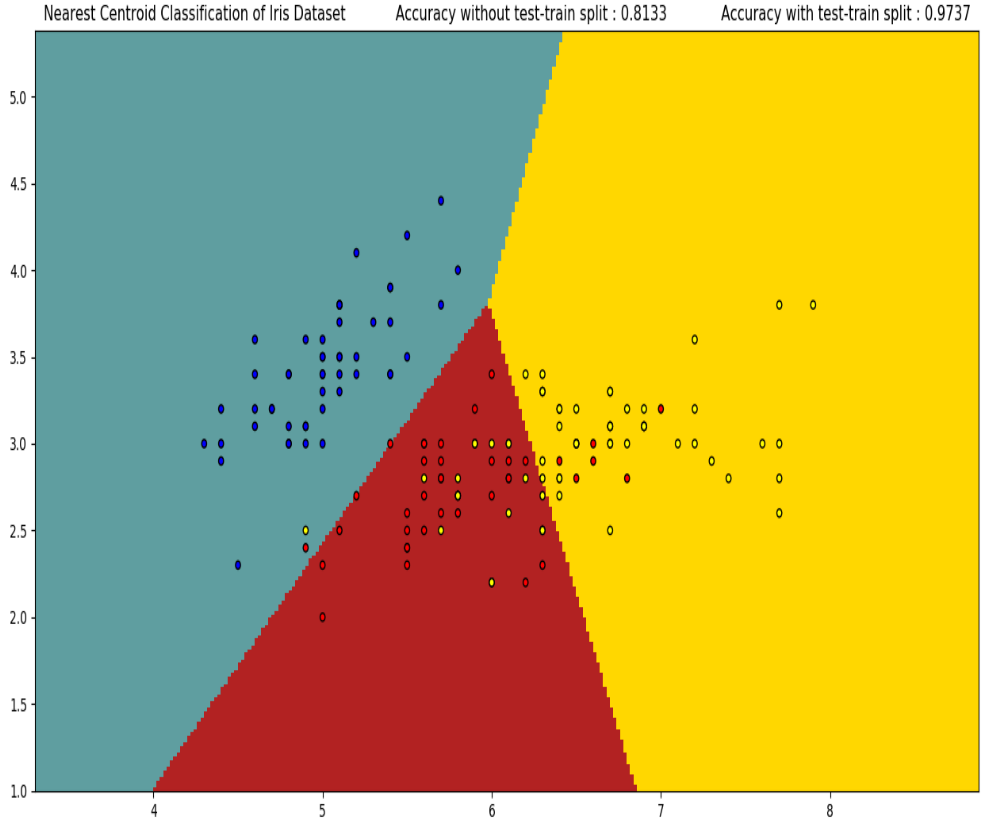
The plots here are not intended for showing accuracy or performance. They show how the kNN Classifier categorizes data into different sets (here 3 as the iris dataset was used). In order to visualize Only the first two features (petal length and width) were used for plotting the figures. The accuracies Written above the figures were obtained without cross-validating the data hence the low performance.



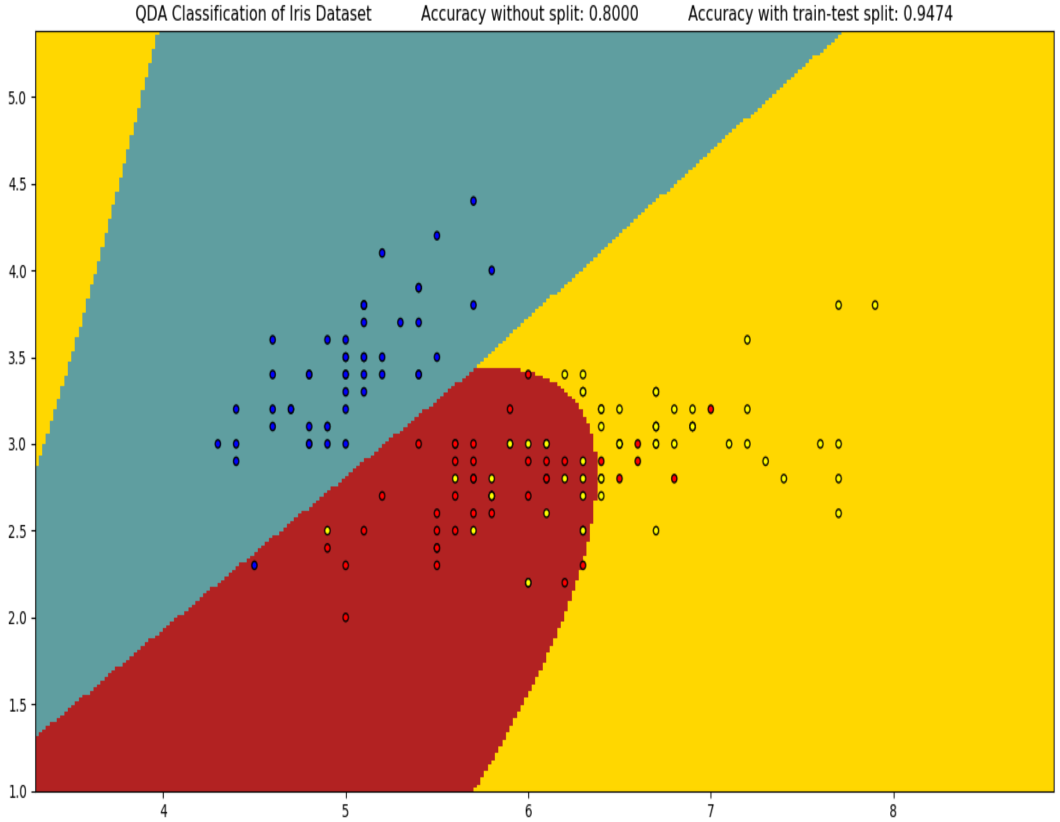
The maximal performance of the kNN classifier is achieved at k = 13 based on the graph and after K = 13 the performance starts to worsen albeit non-monotonically as k is increased.

Performance of kNN classifier with two different cross-validation methods:
KNN (k=13) Accuracy (train-test split : 75 % - 25%) (Iris Dataset): 0.9736842105263158

c-fold (c = 4) Cross-validation score is:
Cross-validation score is [1. , 0.97368421, 0.94594595, 0.97297297] : Mean CV is 0.9731507823613088

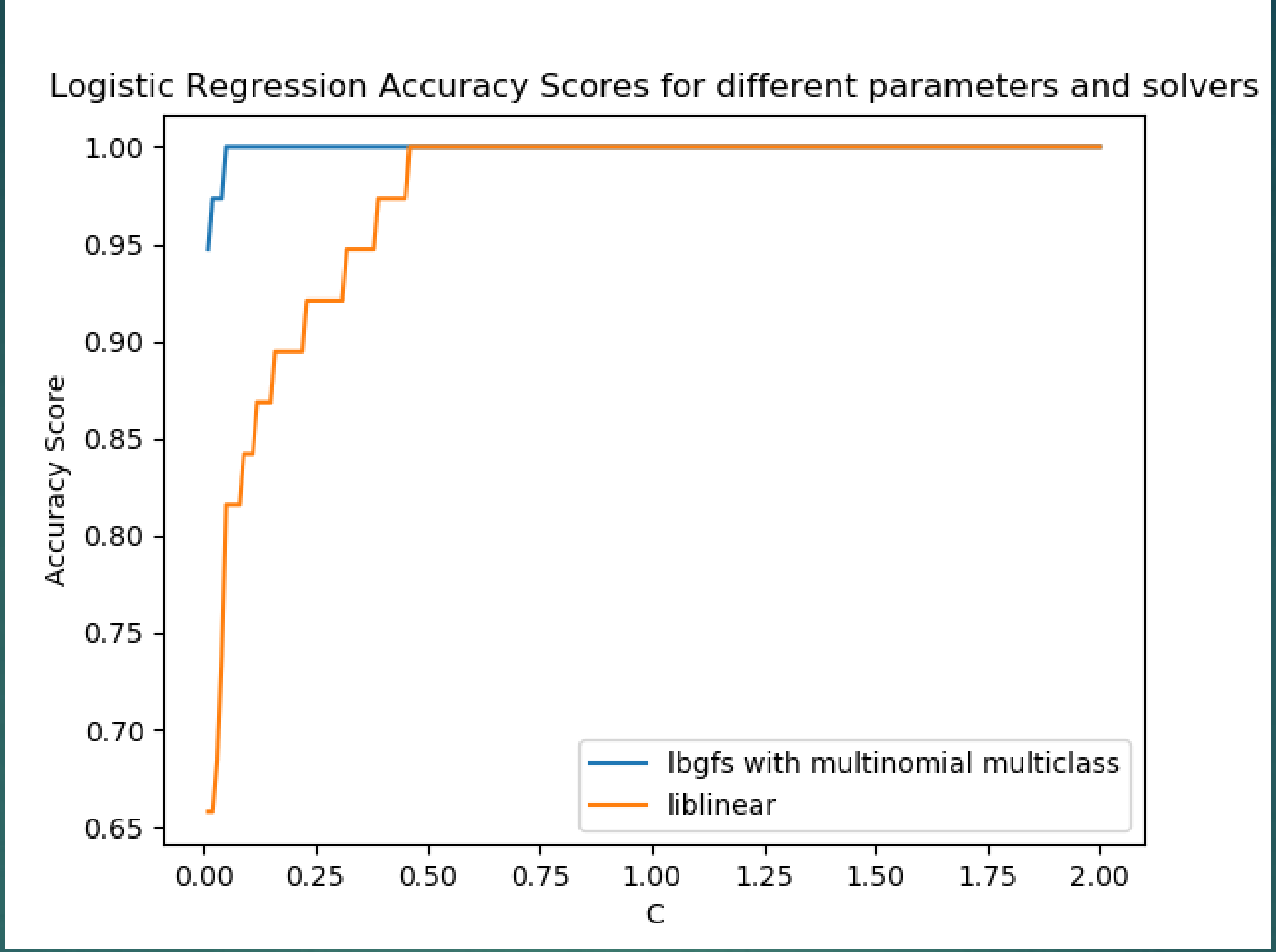


Nearest Centroid Classification of Iris Dataset Accuracy without test-train split : 0.8133 Accuracy with test-train split : 0.9737



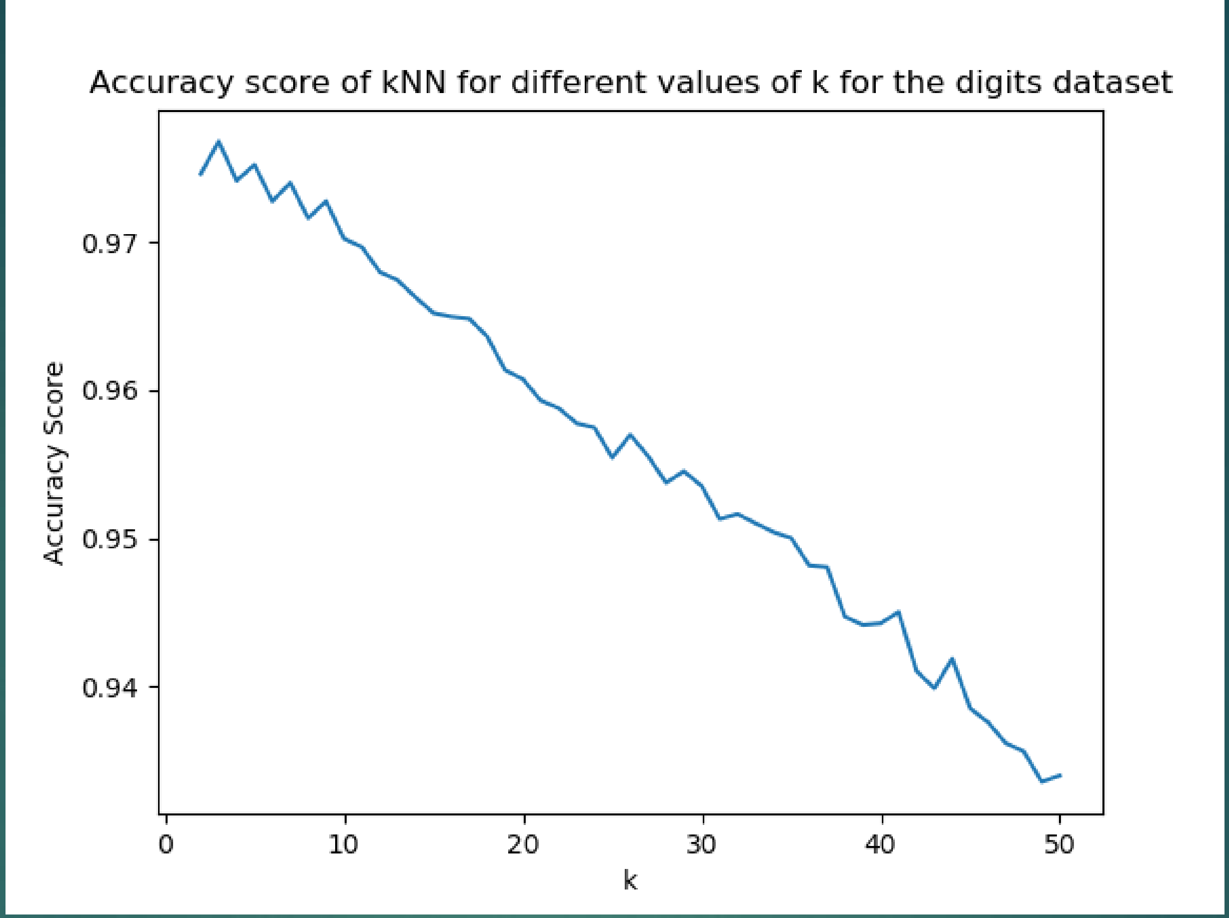
QDA Classification of Iris Dataset Accuracy without split: 0.8000 Accuracy with train-test split: 0.9474

The nearest centroid algorithm is able to find linearly separable boundaries and performs better than the QDA algorithm suggesting that the boundaries tend to be linear between The different classes of iris documented in this study. Not surprisingly, using a split cross-validation Significantly increases the performance of the classifiers as noted above the figures.

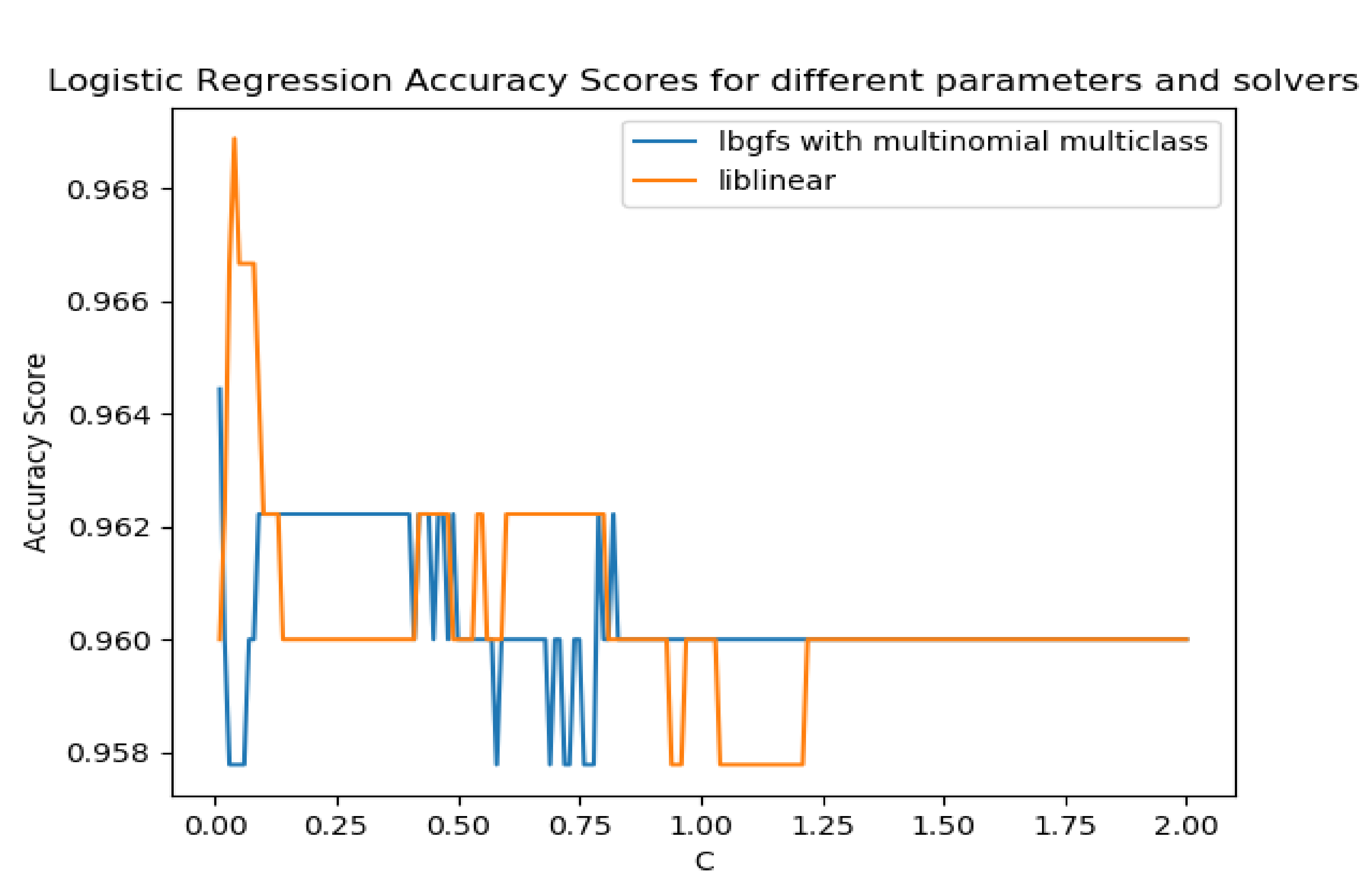


Here I am using penalized regression with L2 regularization. C here is the inverse of the regularization strength; smaller values means stronger regularization. Regularization is used in general to avoid overfitting. Some of the coefficients in the regression model may be assigned very high coefficients which will cause the model to learn the training data too ‘perfectly’. By doing regularization we remove these large coefficients to get a model that captures the data with less overfitting. Here, the regularization factor is multiplied by the squared coefficients and added to the cost function of the model. This causes the learned coefficients to draw closer to zero if they are too large. The larger the regularization factor the more the coefficients are set to zero.

Comparing the different methods we can see that logistic regression performs better on the iris dataset. Especially with the lbgfs solver (which is default in scikit-learn) without much parameter tuning. I just choose these two solvers to see if using different solvers will cause a significant change in the results.



In the digits dataset increasing the value of k more coherently reduces performance compared
To the iris dataset. This shows that the digits dataset is more sensitive to tuning k compared to the iris
Dataset. The optimal k (3) results in a very good performance on this dataset.



Compared to the iris dataset the logistic regression performs worse here over a range of Parameter values. And logistic regression performs worse than kNN for the digits Dataset.

dataset/classifer	kNN	Nearest centroid	QDA	Logistic regression
Iris	0.9736	0.9737	0.9474	1.0
Digits	0.9911	0.9111	0.8288	0.969

The accuracy scores shown are the best ones obtained over a range of parameters. Interestingly, QDA has the lowest performance in both cases.

Discussion

- We saw that as k is increased beyond a threshold the performance of the kNN algorithm starts a non-monotonous decrease. The reason for this decrease in performance could be the tendency for higher k values to undermine the boundaries between classes, especially if the classes have a lot of overlap (since we are using the Euclidean distance). The fact that the performance decreases non-monotonously is due to the local distribution of the data around each point and a general conclusion cannot be derived. In general, one can agree that a smaller k is more suitable as seen from the performance curves over the two different datasets. However, kNN performs well on classification tasks overall, and is a good choice to use on most datasets as it is not affected by correlations in the data, for example. The liblinear solver is better for high dimensional data which is the case in the digits dataset.
- Interestingly, the nearest centroid classifier which is essentially a parameter free classifier performs good on the iris Dataset. It is evidently not as good as kNN or logistic regression since it is a linear model. This maybe due to the observation that the iris dataset is more or less linearly separable. As the classification becomes more complicated, either due to nonlinear separability of the classes or as the number of classes increases, the nearest centroid algorithm’s performance decreases.

Discussion (continued)

- The logistic regression with L2 regularization is able to perfectly classify the data in the iris dataset. The model does so in the absence of strong regularization which suggests that this task could be done by OLS regression without the need to regularize. Logistic regression (or in this case ridge regression, although regularization helps in this manner) performs not as good when the data are correlated as in the case of the digits dataset. Multicollinearity reduces the precision of the estimate coefficients, which weakens the statistical power of the regression model. However, the regression model perform pretty well in this case as well (near 97%) which suggests that the multi-collinearity in the data is moderate and not a huge obstacle. Nevertheless, the kNN has a better performance here both due to the high number of data points and due to its oblivion towards collinearity.
- QDA assumes that the data are sampled from a normal distribution and this could be aptly assumed for the iris dataset, however, not for the digits dataset. This might be one reason why QDA performs significantly worse in the latter case. Also another assumption of QDA is that each class has its own covariance matrix. As we have mentioned the digits data set is multi-collinear and thus the covariance matrices of each class is not independent of other classes which further worsens QDA's performance here.