Assignment 2

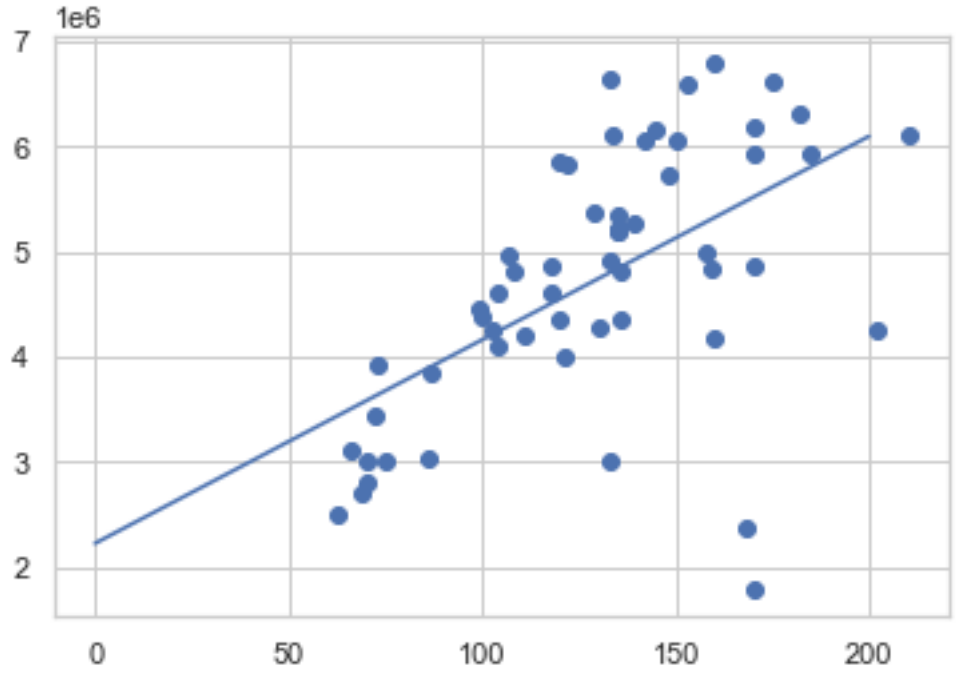
by :

Christoffer Wikner (931012)
Erik Rosvall (960523)

Time spent:
Christoffer - 10 h
Erik - 10 h

Problem 1

a



There was no data cleaning done to the data set. This data set is a well know and tested. Since it's well tested, we didn't find any reason to clean the data.

There are some works with the data, it's reshaped to fit the linear regression model

b

The values we got from the model are:

- intercept: 2220603.243355869
- coefficient: 19370.13854733

c

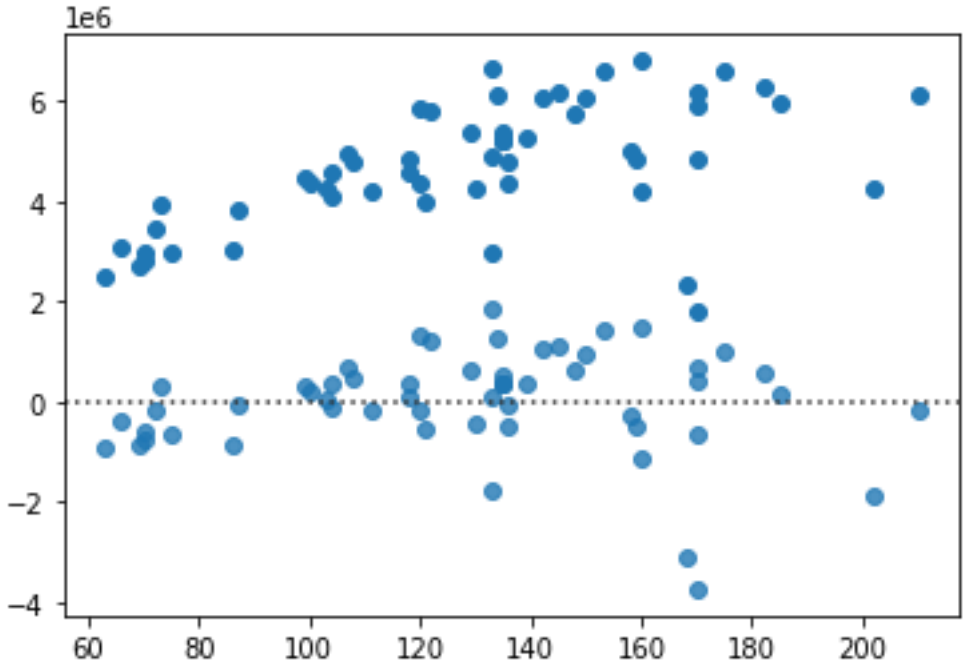
Formula
f(x) = k*x+m

k = intercept
m = coefficient

Our x values to test are 100, 150 and 200. The results are:

- x = 100: 4157617
- x = 150: 5126124
- x = 200: 6094630

d



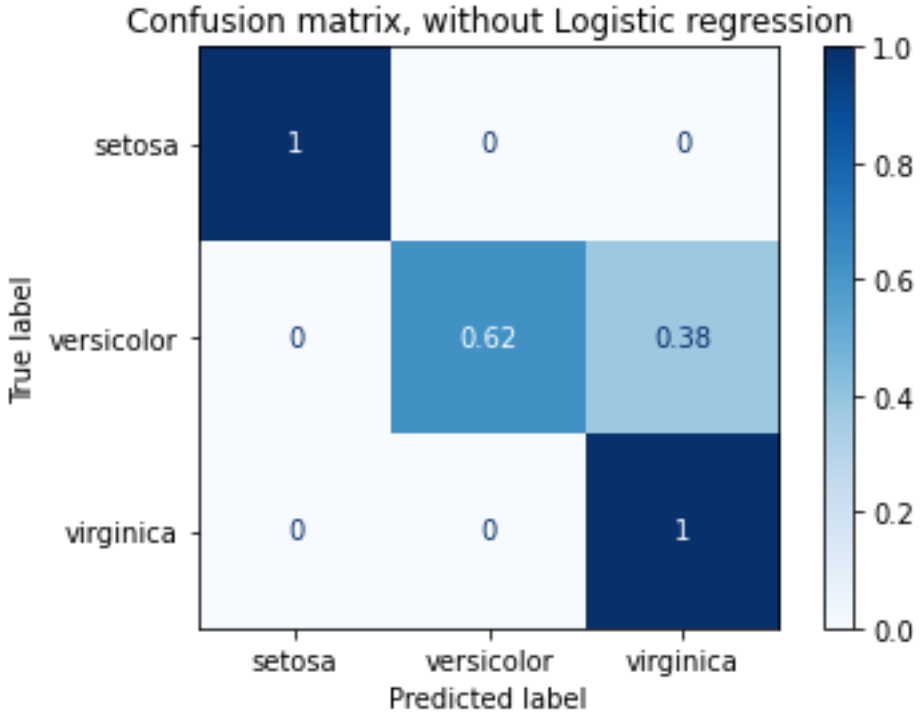
e

The model could be improved by a few steps. One way to improve the model is to modify the scaling and selection of data points. There could also be some transformation, from categorial to numerical format.

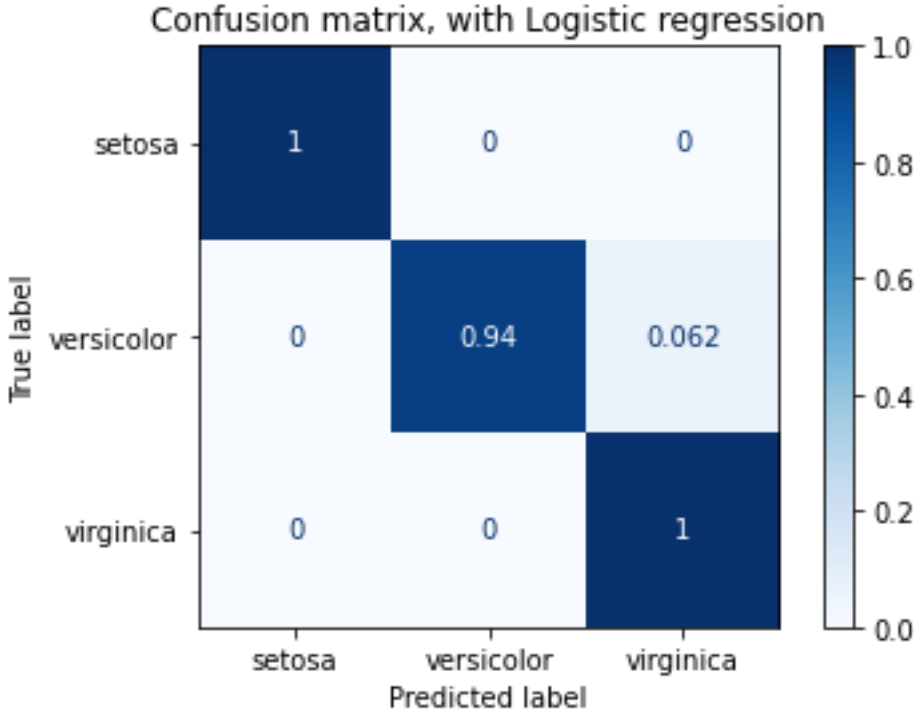
Problem 2

a

Without logic regression



With logic regression



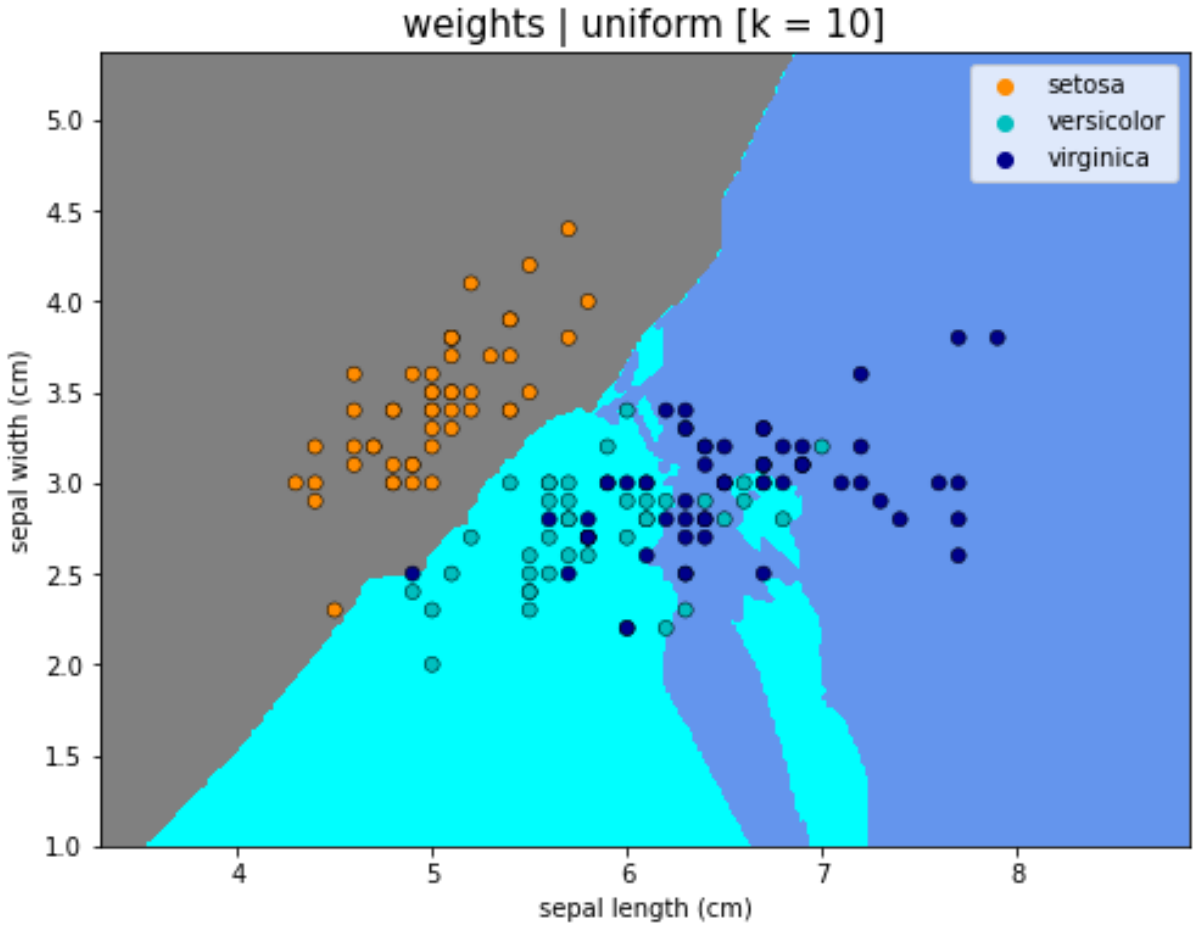
Here are two confusion matrixes, the first one represents the iris data set without logic regression, it's a linear regression training. The other graph represents the same data set, but it has applied the logic regression. Both graphs are in a scale between 0 and 1. In the first graph who only uses linear training, we can see that setosa and virginica is a full 1, but when it ties to predict the versicolor it only has a 0.62 accuracy and it also predicted re remaining 0.38 to be versicolor.

If we now look at the second graph with logic regression applied with lbfgs (Large-scale Bound-constrained Optimization), we can see that it has the same prediction when it comes to setosa and virginica, still a 1. There is a difference to the versicolor, it's now 0.94, which mean that its able to classify almost every versicolor correctly, there is only 0.062 of the remaining who it classifies as a virginica.

By applying logic regression, there is a huge improvement to the model. it's able to correct itself from 0.38 to 0.062.

b

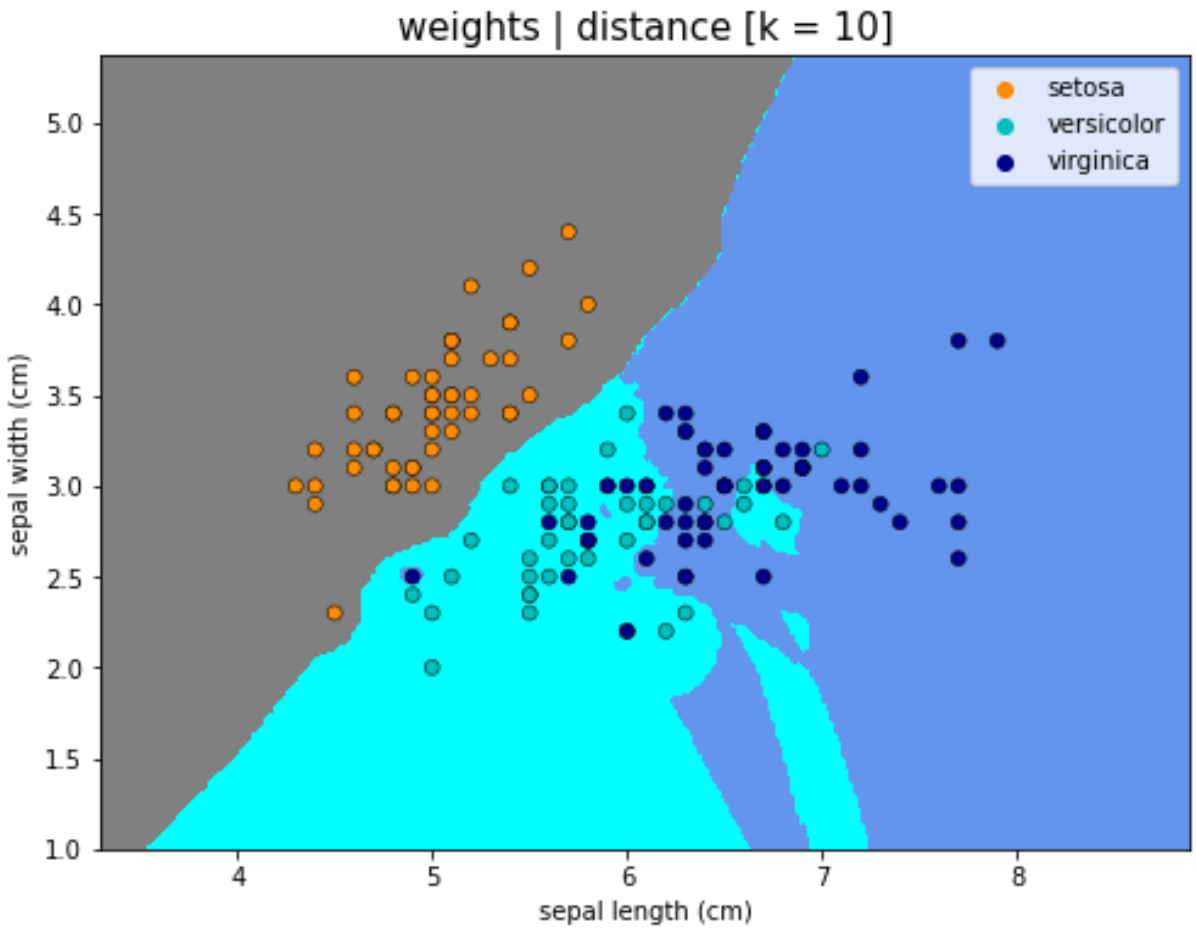
k = 10 (uniform)



we can see all the setosa's are collected in one cluster and there are just one virginica who is placed in the grey area. By looking at the other two there are more confusion. If we take the versicolor, we can see that there are some versicolor who

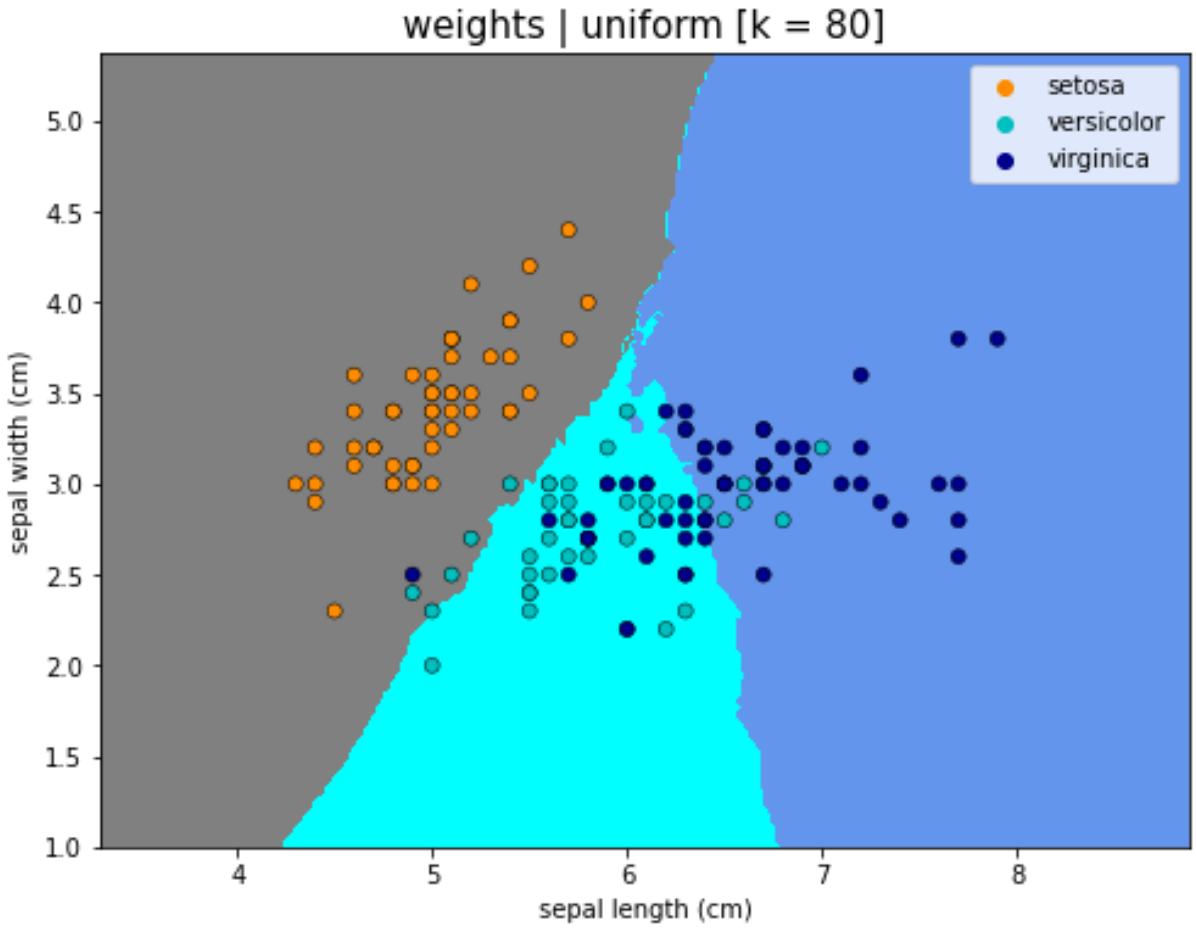
are wrongly placed (around 4). Lastly, we have the virginica. We will start at the grey area where we can find one virginica, and in the cyan area are there 8 virginica who don't belong there and one at the border of the cyan and blue.

k = 10 (distance)



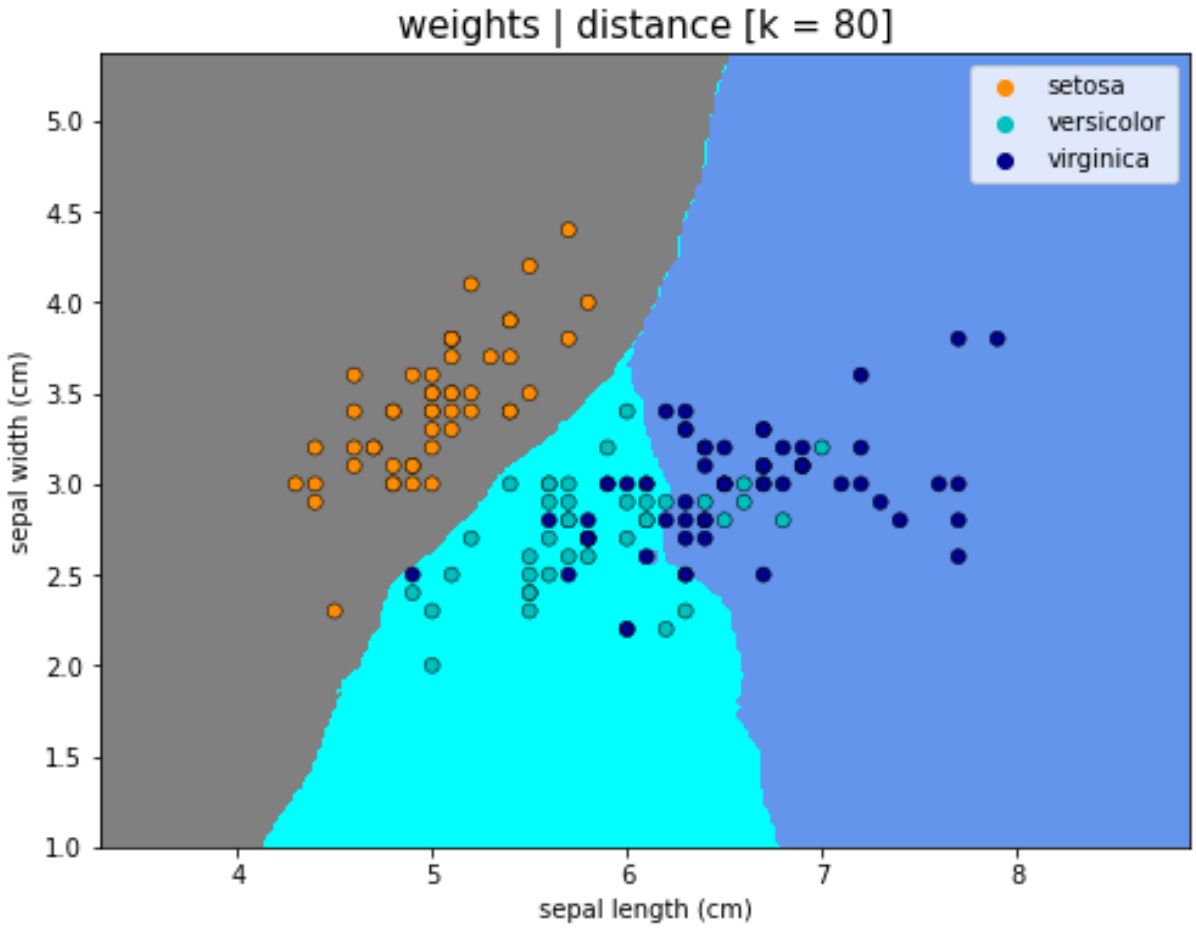
When we are looking at the distance can we see a difference if we compare the two different images when k = 10. If we look to the left in the cyan area can we see one virginica, who have its own blue area, but the remanding of the virginica in the cyan area is incorrect (eight of them). The grey area is the same as in the uniform. There is a difference in the blue area as well, it's only 3 versicolor who is placed wrong instead of four.

k = 80 (uniform)



If we increase the value of k, we can see a difference in the graph. We start with the grey area again, where we can see that the area are including more flowers than it did for a lower value of k. Now it includes one virginica and five versicolor. The same is has happened when it comes to the cyan-area, for a higher value of k there is a lot more of virginica in the versicolor area (around 15, and one on the border). Lastly, we have the blue area where there are six versicolor in the virginica area.

k = 80 (distance)



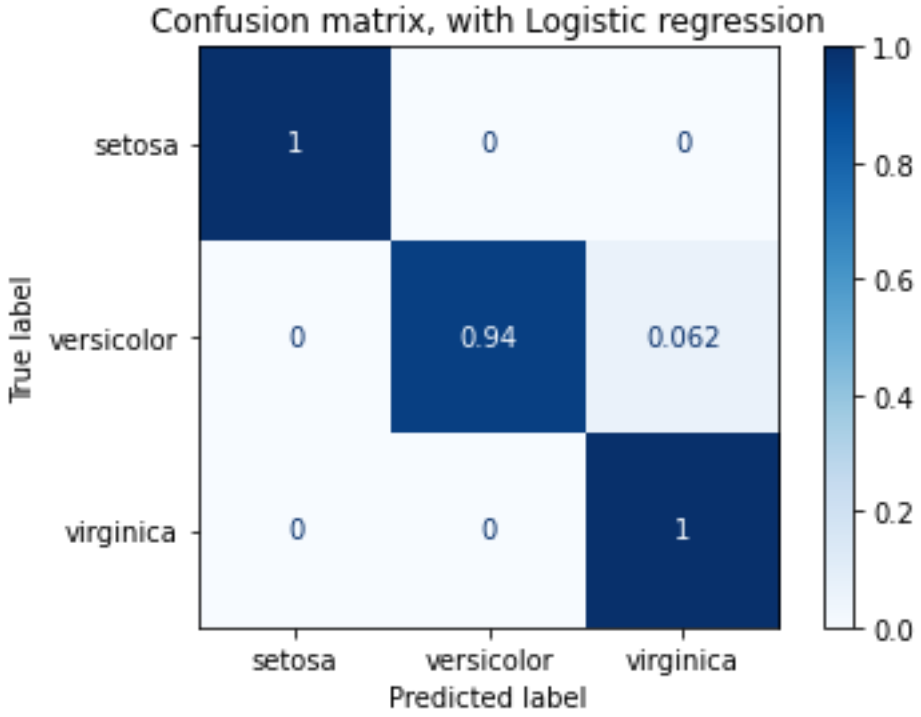
There are some differences between uniform and difference when k = 80. In the grey area when measuring the distance there is some versicolor who are in their right area compared looking at the uniform where they are in the grey area. In the cyan area are there only 10 misplaced virginica and there is one who has its own blue area in the cyan. The versicolor in the blue area are the same as in the uniform graph for k = 80.

The bigger value of k the less flexible the classification gets. If k is smaller value, it's more flexible which we can see in the graph k=10 (distance) where is separated out a single viginica in the versicolor area. There is a a risk of higher miss classification for higher values. There is a risk with to low value of k as well. We can get a perfect classification, with separated areas. With a lower value of k there is a problem with the prediction of new data. the same problem occurs when k is to high (classification is inflexible). To solve this, we should find a sweet point between the extreme values.

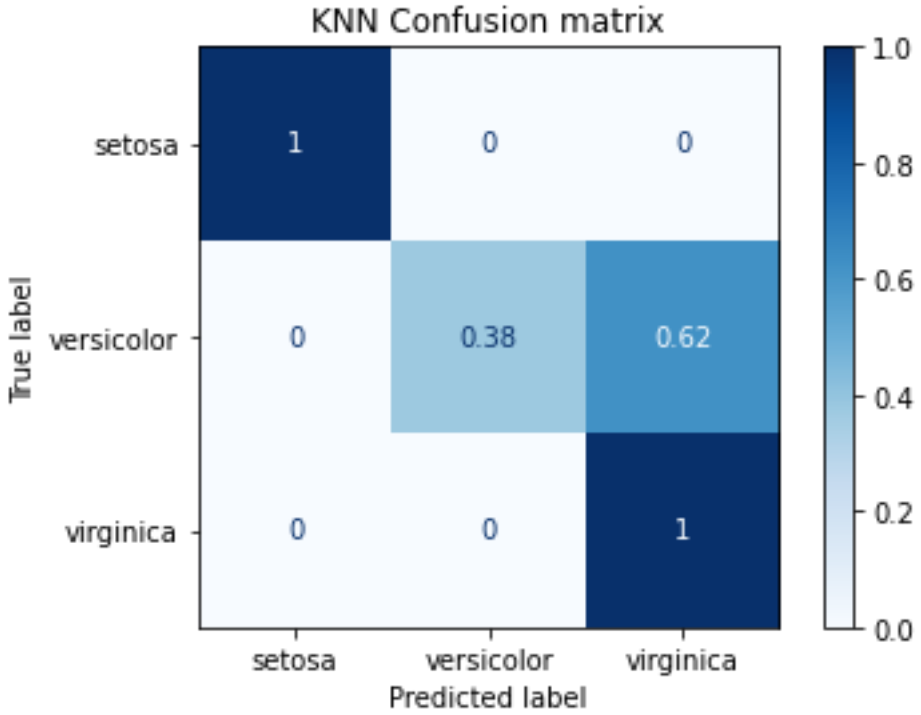
c

Compare the classification models for the iris data set that are generated by k-nearest neighbours (for the different settings from question 2b) and by logistic regression. Calculate confusion matrices for these models and discuss the performance of the various models.

Logic regression



KNN Confusion matrix



The first graph are explained before, look at problem 2-a. The other graph represents the KNN problem which is discussed in 2-b. If we look at the last graph, we can see that its different than the first one. If we start at the upper left corner, we can see that both models did a perfect prediction setosa. If we now continue to the second row of

the matrix, we can see that in the upper graph, there were a prediction accuracy of the versicolor 0.94, and it miss predicted 0.062 as virginica. In the lower graph the prediction was not as accurate, it did only a prediction 0.38 of all versicolor correct and had a miss of 0.62. This is a 0.56 difference. This means the logic regression is more accurate to predict versicolor. If we now look at the virginica, in the upper graph we got a perfect 1 for the virginica, and the same result in the lower graph.

To summarise, the logic regression model did a much more accurate prediction than the KNN and linear regression.

Problem 3

With a data set we need to split it into training/testing pieces. One way to split the data is cross validation (you split the data into equal block sizes) and select one of the parts to be the test block. when you have trained the data and tested is, you can swap the test block with one of the training blocks and repeat this process until every training block has been used as a testing block. It is important to clear out the model when switching testing block because there is a risk that the model can "over think". When using the data, it’s good to be as random as possible, in other words we want the training data to represent the full model as much as possible, the same applies to the testing.

The difference between testing and training data is that with the testing data, we what to check if the model we trained can do accurate prediction with similar data the model have not seen yet. The same applies to the validation set, but a validation set is more used to fine tune the the model in its final stages.