Report Introduction to Data Science – Assignment 5

Exercise 1 - Linear regression

Linear regression is a model that assumes a linear relationship between the input variables (*physicochemical properties* of wine in this example) and the single output variable (*quality score*). Running the model on the first features of the data set resulted in these weights: [5.20 0.05].
The first value is the offset parameter w_0 and the second value w_1 is the coefficient found for the first feature from the data set. This value tells us that the feature *fixed acidity* has a scale factor of 0.05 in the correlation with the *quality score* of the wines.
When the algorithm is run on all features from the data set the weights found are: [5.16e+01 1.95e-02 -1.06e+00 2.58e-02 5.02e-02 -2.75e+00 5.65e-03 -3.80e-03 -4.72e+01 -4.26e-01 8.50e-01 2.37e-01]
The first value, w_0, is almost equal to the one found in the previous run.
The second value w_1 is different, this is explained because each coefficient represents the additional effect of adding another variable to the model, in this case all the other features are added to the model.
The highest value is 8.50e-01 corresponding to the correlation factor between *sulfates* and the *quality score*; whereas the smallest is -4.72e+01 corresponding to the *densitiy* feature. When a coefficient is closer to zero this could mean that the influence of the feature on the *quality score* is low. However, in this analysis if the value of the feature multiplied by the coefficient has to be taken into account. This means that in the case where the coefficient is close to zero and the value of the feature is high, the result of the multiplication could still be higher than cases where the coefficient value is higher but the value of the feature is low.

Exercise 2

The weights computed in the previous exercise allows us to predict the output variables and to use these predictions to compute the root mean squared error measure. This is done by: given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together.
The RMSE for one feature test set is: 0.78 and for all features test set is: 0.64.
This final result tells that using all features from the data set gives a RMSE value smaller than using only one feature. The linear model seeks to minimize this quantity.

Random forest

Normalization doesn't affect the random forest classifier because features are never compared between themselves, so the ranges don't matter. At each stage the range on one feature only is split.

Exercise 3&4

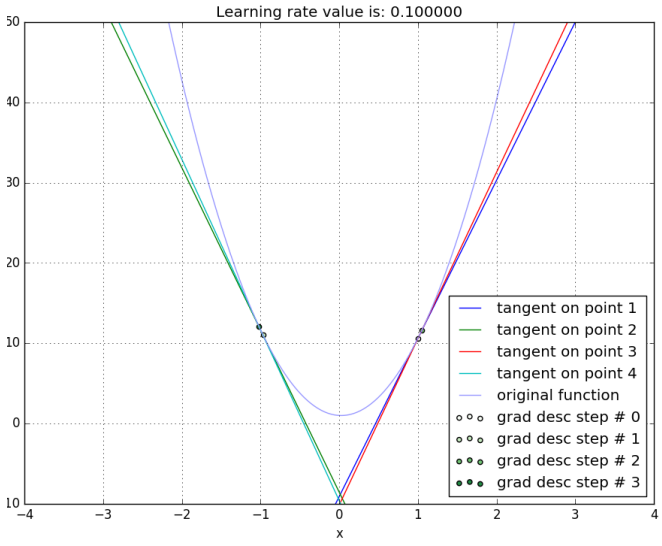
In the manual implementation of the 1NN-classifier the accuracy score of the on the test set is 0.947 and the accuracy score of 3NN-classifier on the test set is 0.949

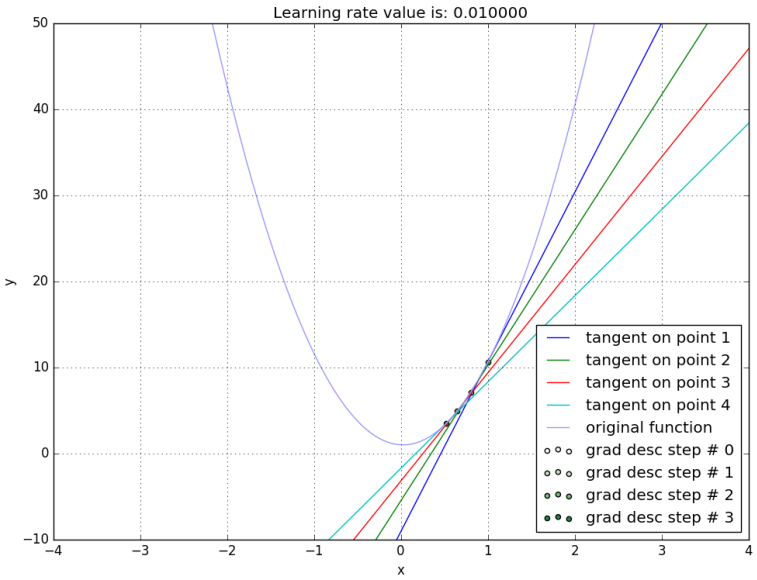
The random forest classifier is applied to the weed and crop data and the accuracy score for the test set is: 0.968. So it does performs better than the NN classifier.

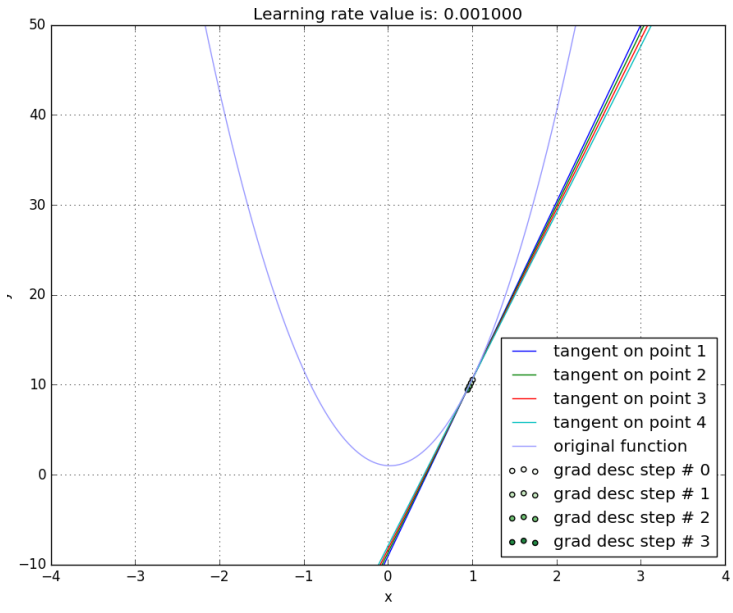
Exercise 5 - Gradient descent

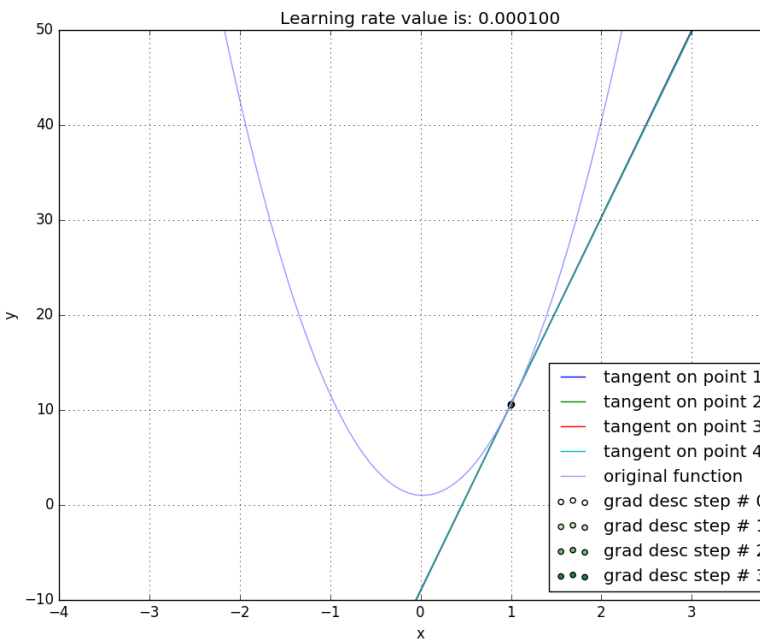
Gradient descent algorithm has been applied to find the minimum of the function $f(x) = e^{-x/2} + 10x^2$.

b) Visualization of the tangent lines and gradient descent steps for the first three iterations are presented in the plots below. The gradient descent has been applied with learning rates η = 0.1, 0.01, 0.001, 0.0001.

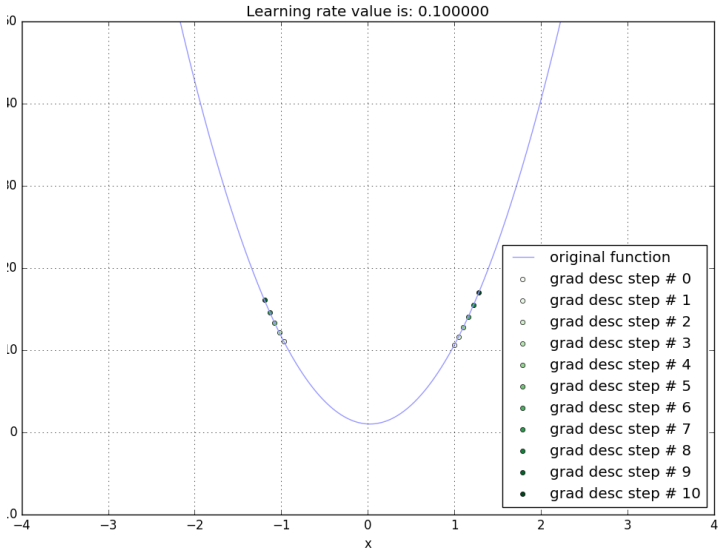


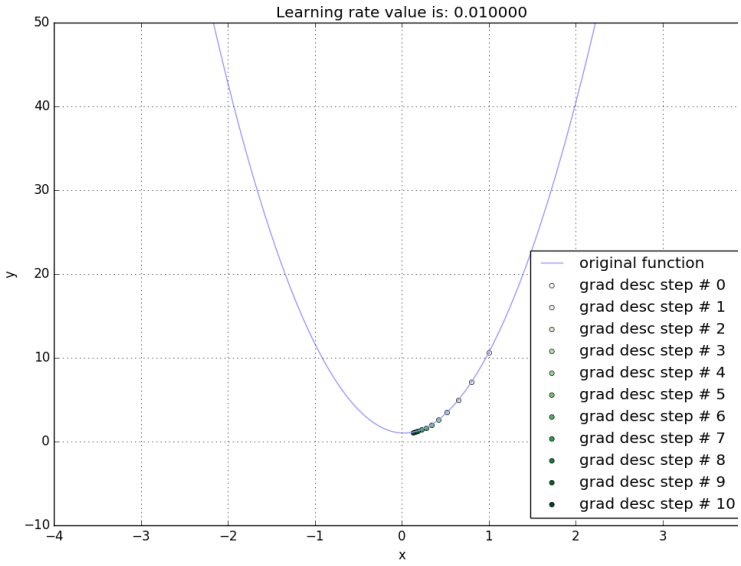


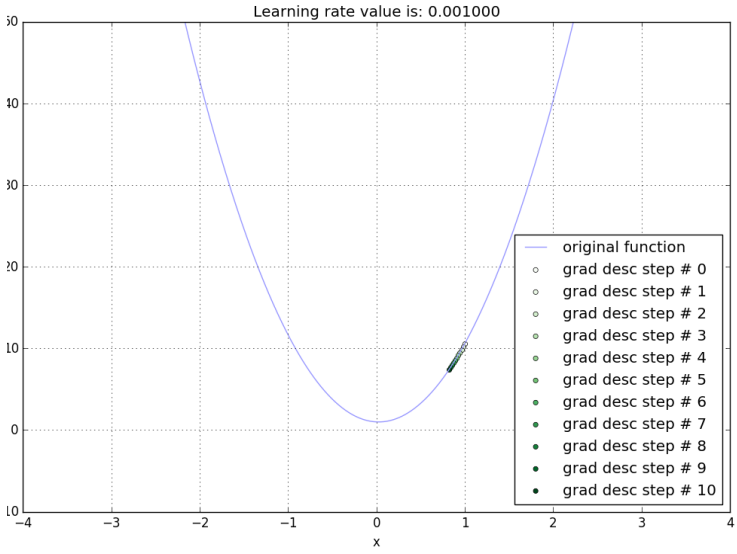


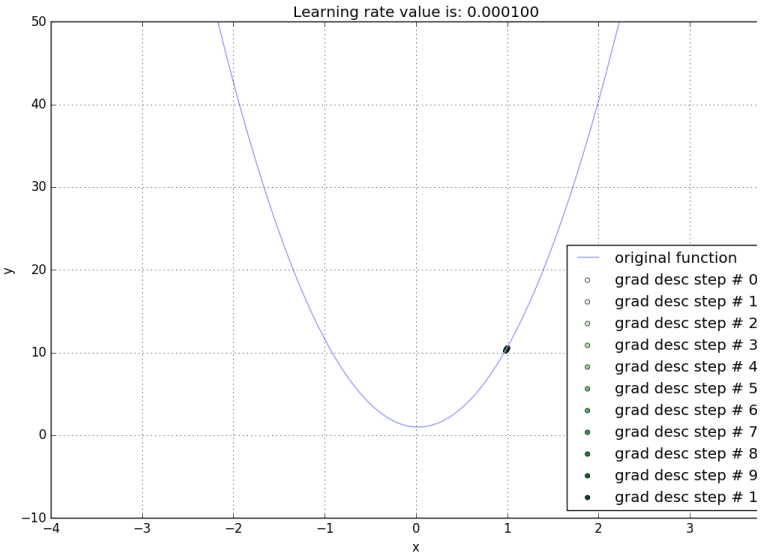


c) Visualization of the gradient descent steps for the first **ten** iterations are presented in the plots below. As in point b) of this exercise, the gradient descent has been applied with learning rates η = 0.1, 0.01, 0.001, 0.0001.









d) The algorithm has been run with the number of maximum iterations set to 10.000. The run for learning rate 0.1 return the error below:

File "<string>", line 1, in <lambda> OverflowError: math range error

This error is reported because the function used to compute the gradients has been turned from 'sympy' function to 'lambda function' using the built-in function **lambdify**() in line 15 of **gradient_descent.py** file. When this function encounter a string then the overflow error is reported. This means the gradients couldn't be read as numbers.

If the learning rate 0.1 is not used in the computation process then the algorithm reports the following results:

With learning rate: 0.01

the function value in the last iteration is: 0.02469323271

the final iteration is iteration number: 95

With learning rate: 0.001

the function value in the last iteration is: 0.0246932372

the final iteration is iteration number: 934

With learning rate: 0.0001

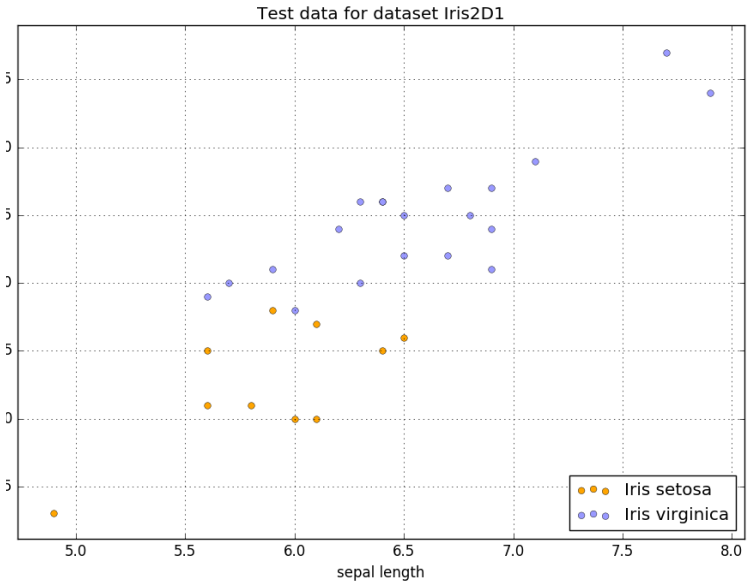
the function value in the last iteration is: 0.0246932816

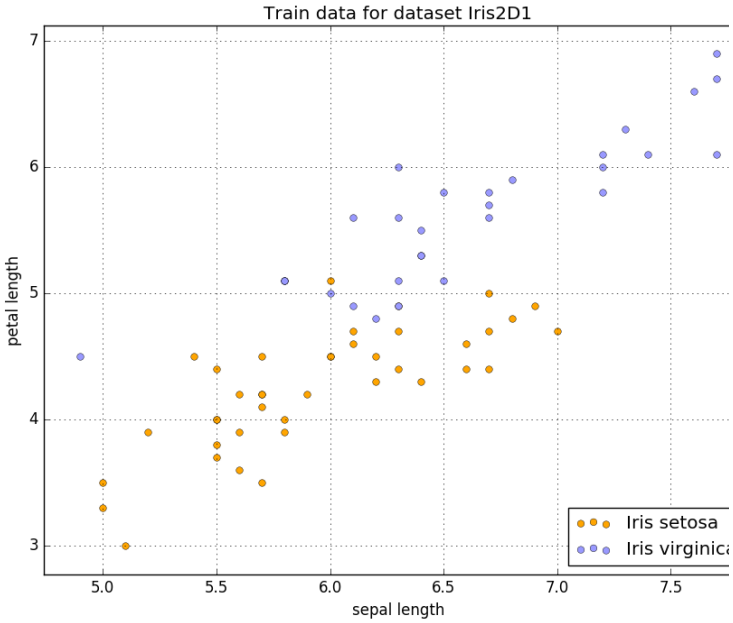
the final iteration is iteration number: 8290

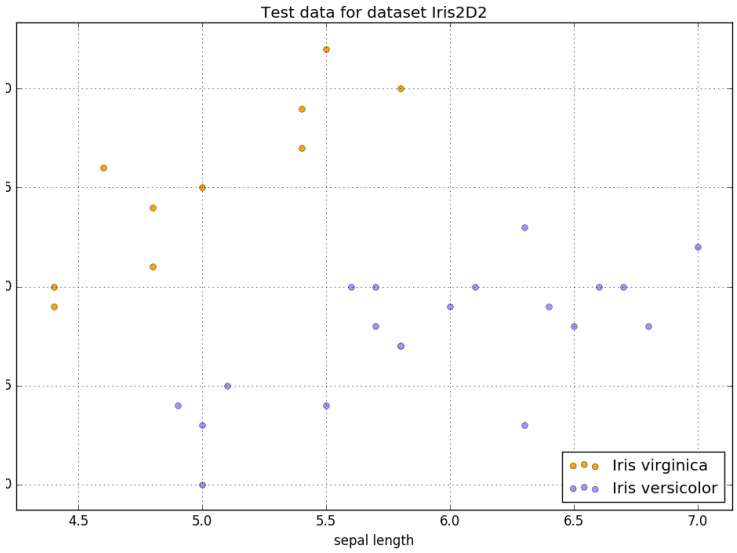
Logistic Regression

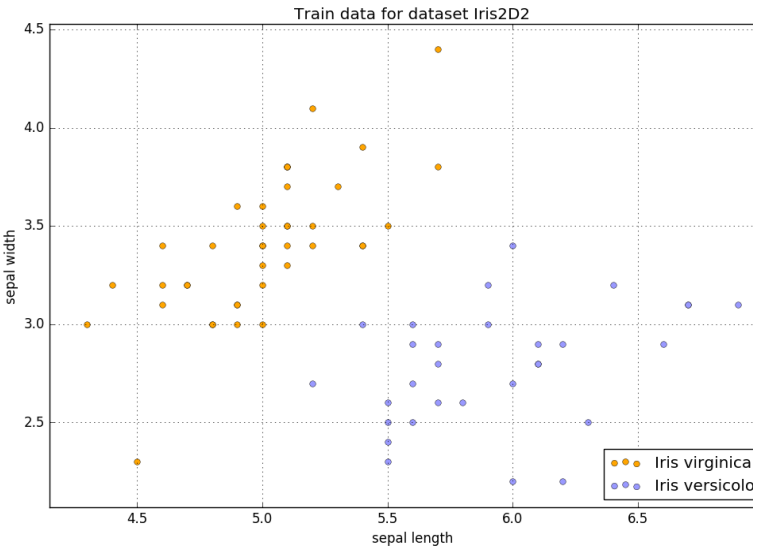
Exercise 6

1) Scatterplots of each data set (divided by test set and train set) are reported below:





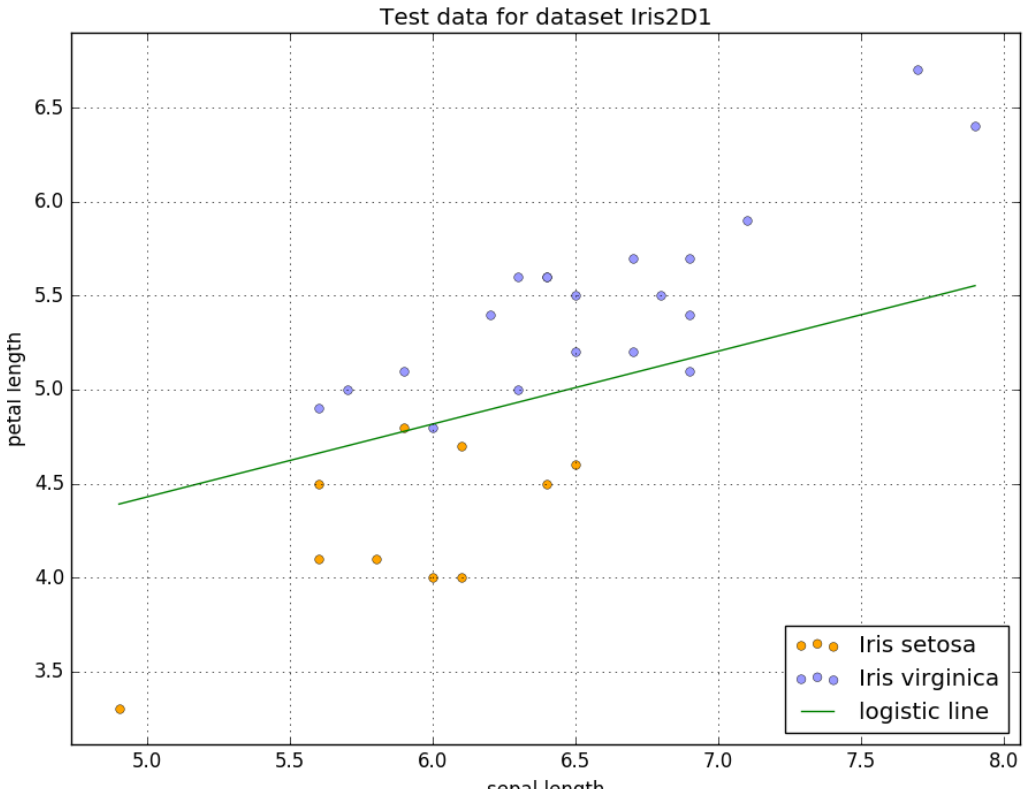


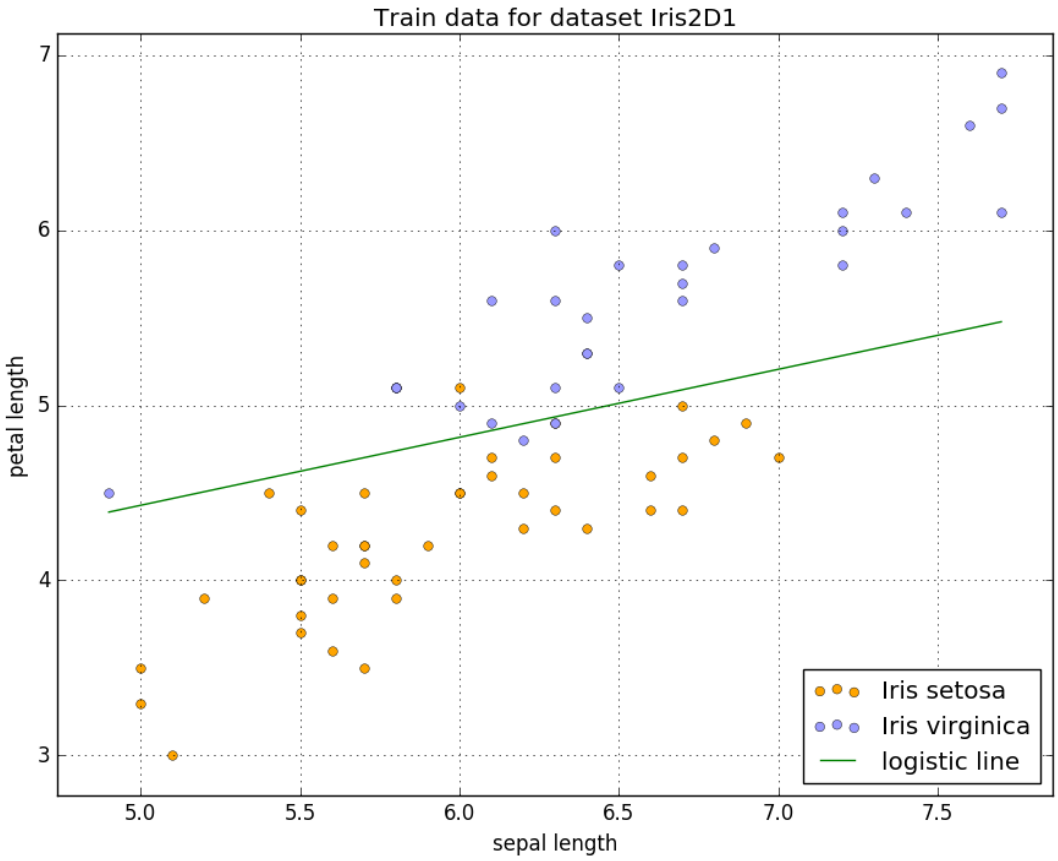


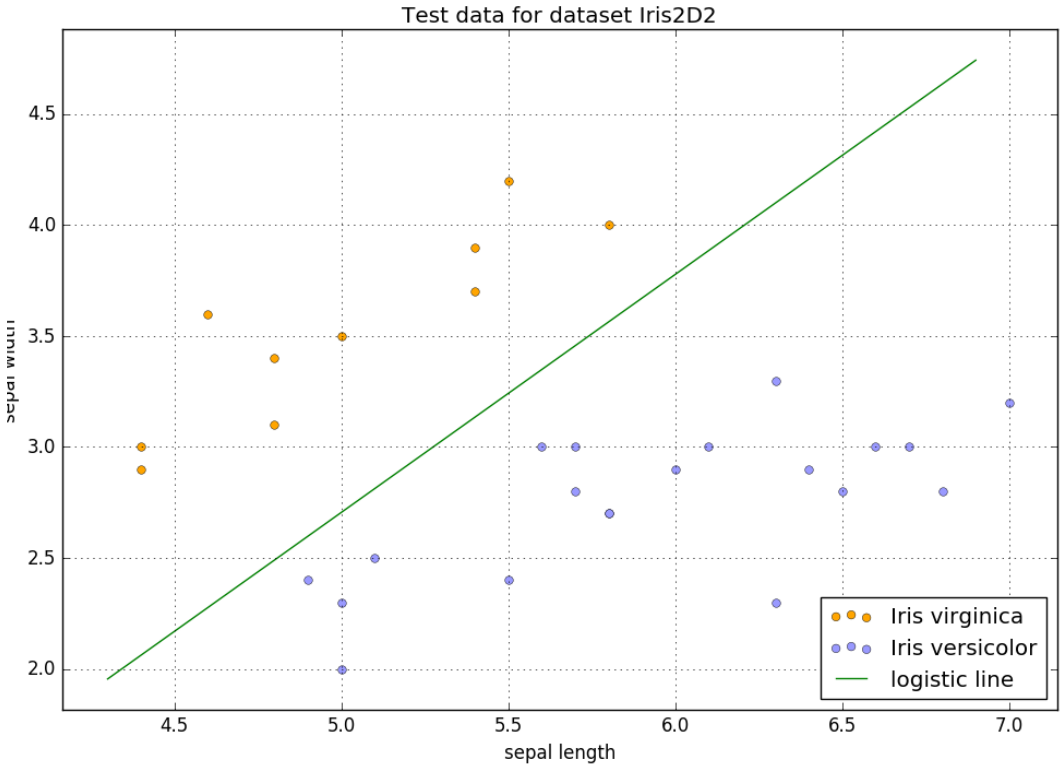
From the plots it can be said that the data points in the 2D2 dataset could be linearly separated 'better' then the points in data set 2D1. This is because the two classes clusters in separated areas of the plots. On the other hand, in the 2D1 plot for both train and test data there can be spotted some points of the two different classes really close together, the clouds are not well separated.

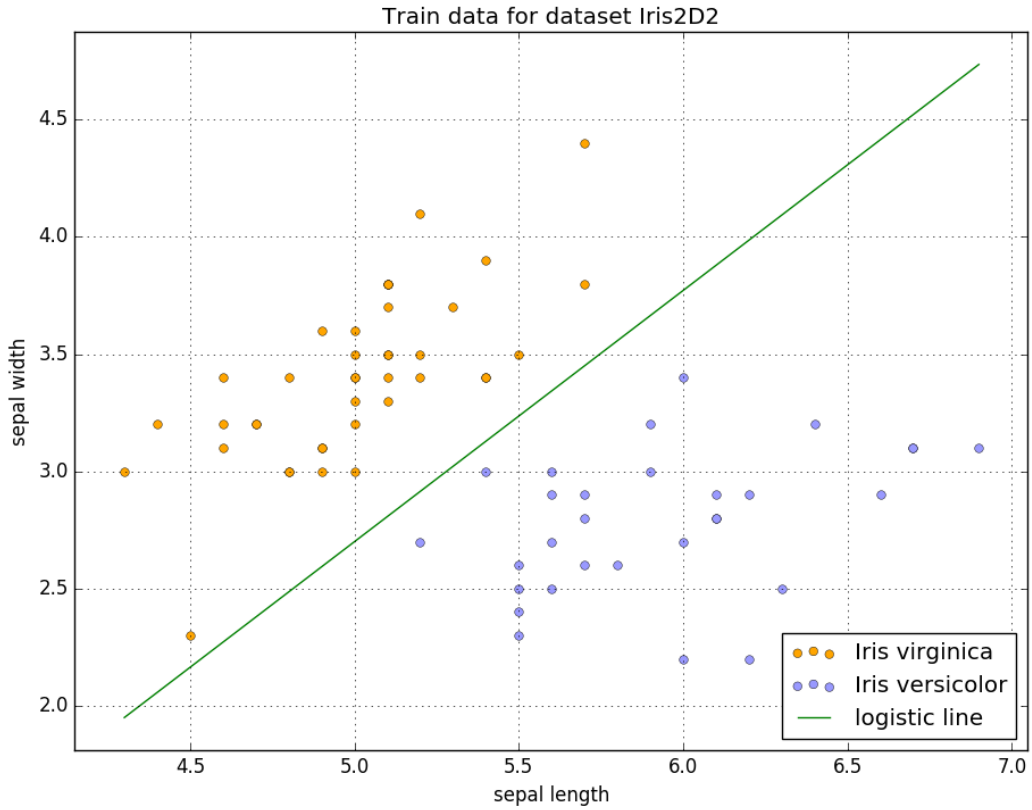
4) Logistic regression is applied to the datasets of Iris2D1 and Iris 2D2.
The three parameters for data 2D1 are: [**-26.6835763 -4.15431041 10.71338671**]
The three parameters for data 2D2 are: [**-42.64114662 17.2208129 -16.09699443**]

The 0-1 error loss for train set 2D1 is: **0.042**, the 0-1 error loss for test set 2D1 is: **0.10**. The 0-1 error loss for train set 2D2 is: **0.00**, the 0-1 error loss for test set 2D2 is: **0.00**.
The plots below report the logistic line found using the three parameters for each data set.









The plots help us confirming the observation made in point 1. In fact it is possible to notice that all the points in the 2D2 set are well classiefied and this is also confirmed by the 0-1 loss error being equal to zero.
On the other hand, some of the points in data set 2D1, both train and test, are misclassified by the model, this is also the cause of the 0-1 error loss being greater than zero.

Exercise 7

1) For logistic regression, show that:

$$\nabla E_{in}(w) = -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n x_n}{1+e^{y_n w^T x_n}} = \frac{1}{N}\sum_{n=1}^{N} -y_n x_n \theta(-y_n w^T x_n)$$

$$\frac{d}{dw}\left(\frac{1}{N}\sum_{i=1}^{N}\ln(1+e^{-y_n x_i^T w})\right) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{1+e^{-y_n x_i^T w}}$$

$$=$$

$$\frac{d}{dw}(1+e^{-y_n x_n^T w}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{1+e^{-y_n x_n^T w}}\left[-y_n [x_n]_i\right] = e^{-y_n x_n^T w}(-y_n [x_n]_i)$$

$$=$$

$$\frac{1}{N}\sum_{n=1}^{N} -y_n [x_n]_i \theta(-y_n x_n^T w)$$

2) From this it is possible to say that the influence of correct classification has smaller impact on the gradient than a misclassified example. The error measure E_{in} is small when $y_n w^T x_n$ is large and *positive*, this implies that an example is considered correctly classified when the signs of $y_n \; and \; x_n$ are the same.