Ariel University

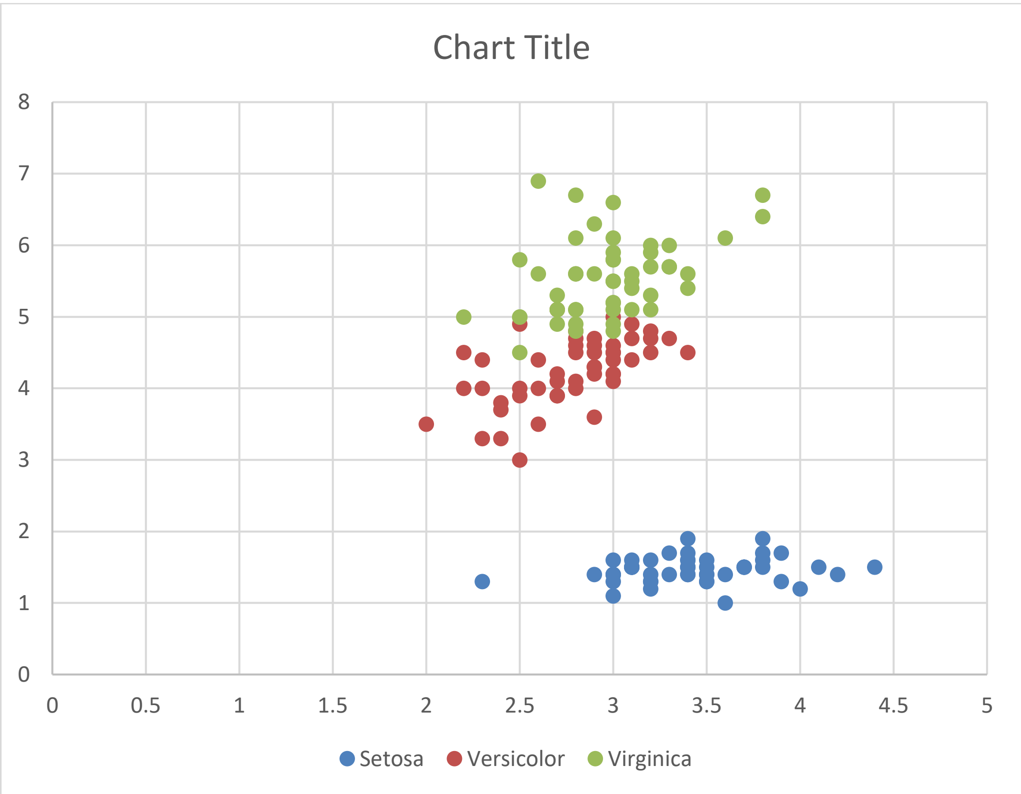
Machine Learning

Homework 2

Problem 1. Class **C** has VC-dimension d. Class **C'** includes all objects that are formed by intersections and unions (in any order) of s objects in **C**. Give an upper bound on the VC-dimension of **C'**.

Note: For the next two problems, hand in code, and also hand in the answers in a separate file, not together with the code.

Problem 2. The famous UCI Iris data set contains information on 150 flowers from three species of iris: Setosa, Versicolor and Virginica. For this assignment we will take only the second and third features for each flower:



Implement the Perceptron algorithm *as learned in class* on this data set (without normalizing the vectors).

1. Run Perceptron on Setosa and Versicolor. What is the final vector? How many mistakes were made by Perceptron? What is the true maximum margin?
2. Run Perceptron on Setosa and Virginica. What is the final vector? How many mistakes were made by Percepron? What is the true maximum margin?
3. Compare the two results above, and explain how and why they differ.
4. What would happen if we ran Perceptron on Versicolor and Virginica?

Problem 3. Now let's take Versicolor and Virginica alone. Each pair of points in this set can define a line that passes through the two points. The set of all such lines is our hypothesis set **H**, that is our set of rules. Implement Adaboost using the above set of rules.

One run of Adaboost is as follows: Split the data randomly into ½ test (T) and ½ train (S). Use the points of S (not T) to define the hypothesis set of lines. Run Adaboost on S to identify the 8 most important lines h_i and their respective weights α_i. For each k=1,…,8, compute the empirical error of the function H_k on the training set, and the true error of H_k on the test set:

$$H_k(x) = sign(\sum_{i=1}^{k} \alpha_i h_i(x))$$

$$\bar{e}(H_k) = \frac{1}{n}\sum_{x_i \in S}[y_i \neq H_k(x)]$$

$$e(H_k) = \frac{1}{n}\sum_{x_i \in T}[y_i \neq H_k(x)]$$

Execute 100 runs of Adaboost, and report $\bar{e}(H_k)$ and $e(H_k)$ averaged over the 100 runs. Hand in code in python (write at the top which version of python you're using) and printouts of the value of each H_k for each dataset (total: 16 values). Answer the following:

Analyze the behavior of Adaboost on train and test. Do you see any exceptional behavior? Do you see overfitting? Explain.