Practical Questions:

Q7:

Implemented in *models.py*

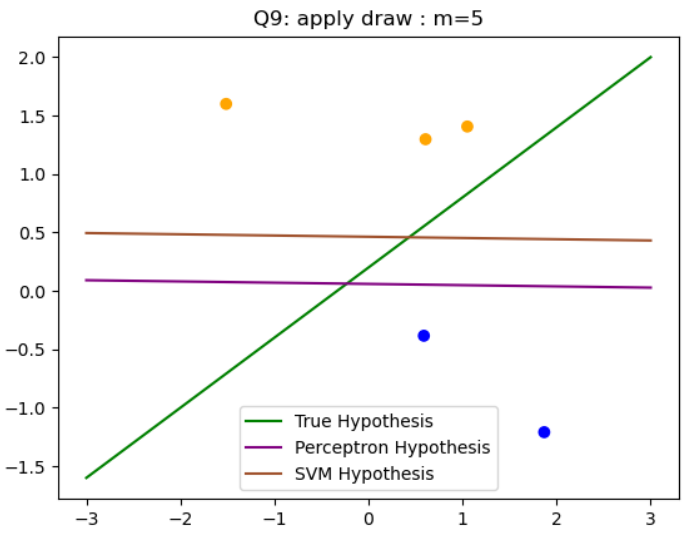
Q8:

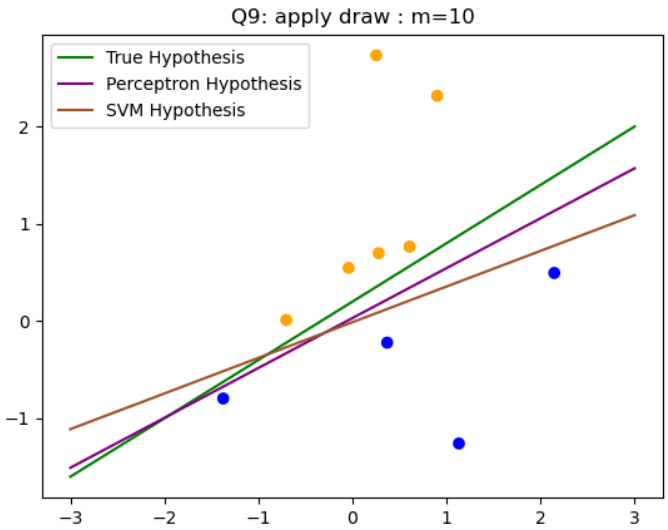
Implemented in *comparison.py*

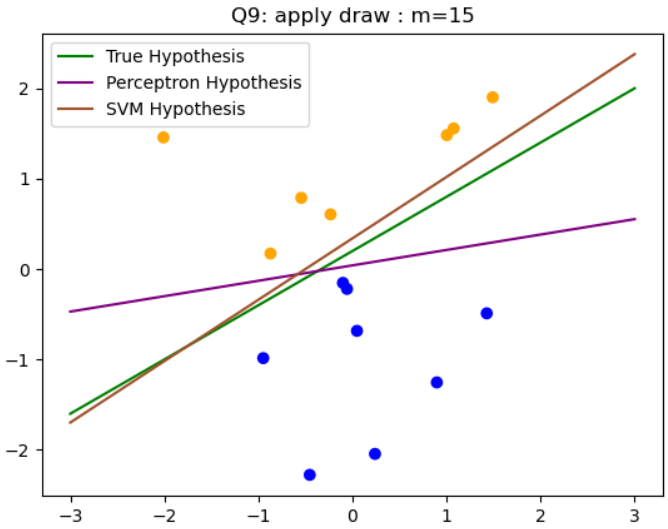
Q9:

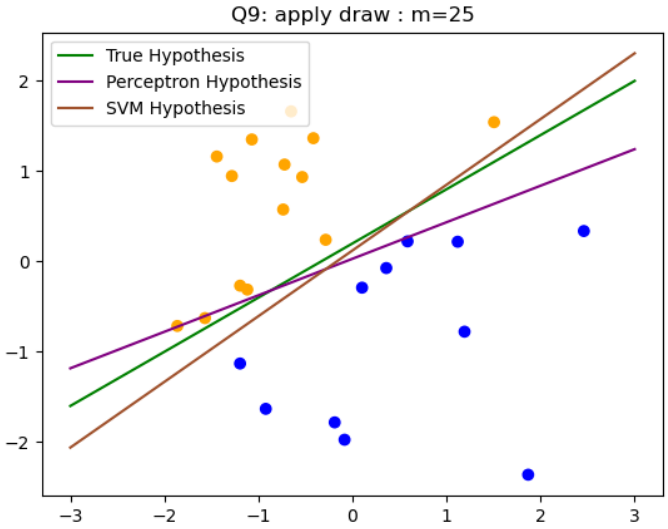
The code that generates the graph is in the function *"apply_draw"*

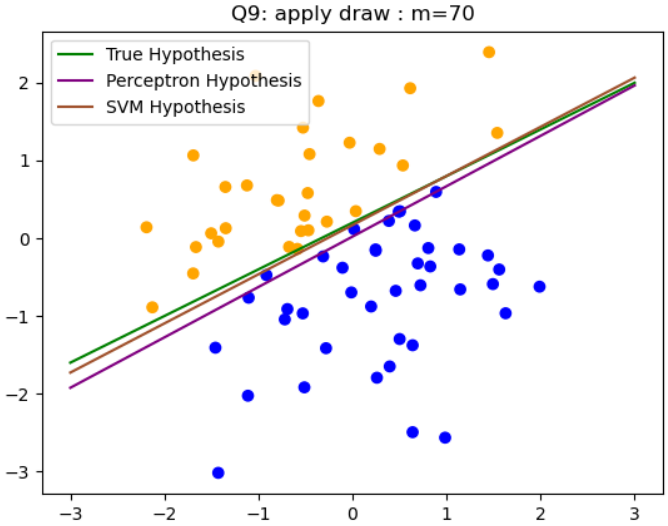
The requested Graphs:





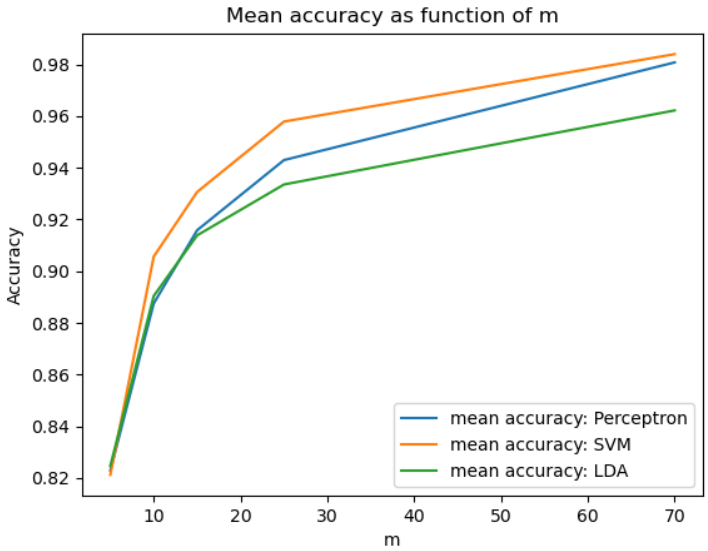






Q10:

The requested Graph:



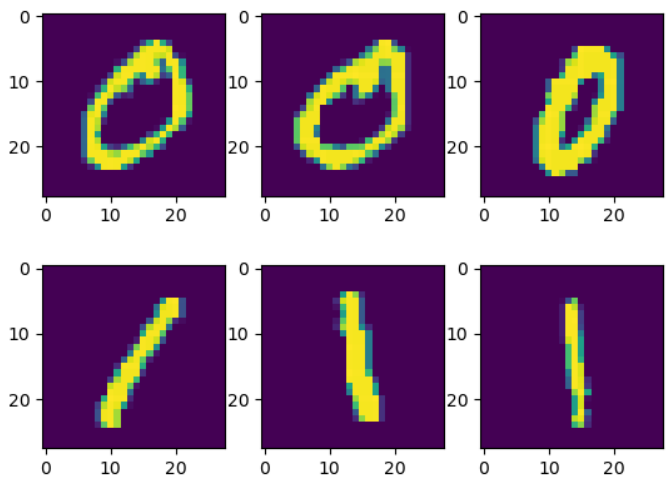
Q11:

The SVM model did better in this instance.
This can be explained by the fact that this model is based on maximizing the margins between the labels of the actual samples and not on guessing the actual model.
Perceptron is a close second, and the LDA did worse, probably because it wrongly assumes the true distribution is made of 2 gaussians.

Q12:

implemented in fumction *play_with_ds*

Requested images:

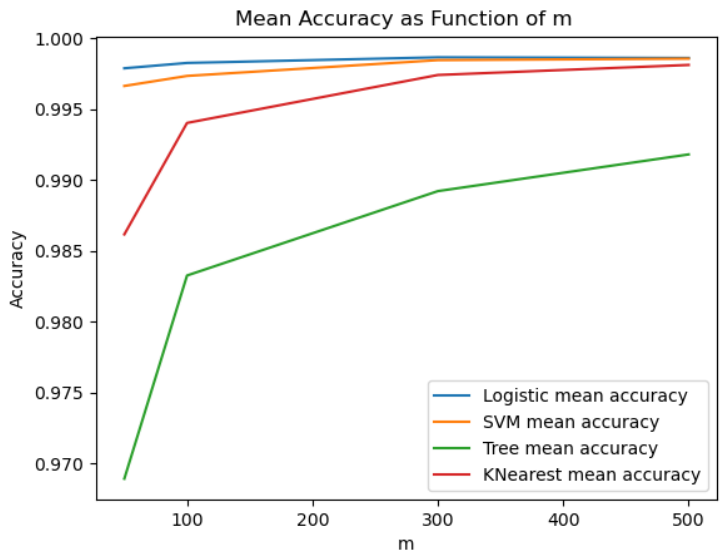


Q13:

implemented in function *rearrange_data*

Q14:

Requested graph




```
Logistic time:[0.0096577  0.01057609 0.01440191 0.02400233]
SVM time:[0.03295195 0.04133796 0.05709656 0.07117558]
Tree time:[0.01100782 0.01535079 0.0314739  0.05162932]
KNearest time:[0.23499504 0.2456953  0.27822554 0.30706699]
```

We see that the decision tree is the fastest in terms of running time (on average), while K-nearest neighbors is the slowest.
This is caused by the nature of calculations of K-nearest neighbors which is heavier than others due to its recursive action.