ACTIVE LEARNING

Prem S P Thakur

Active Learning

Problem :

Imagine You have large amount of ***unlabelled data*** $\boldsymbol{D_u}$ and you have smaller amount of ***labelled data*** $\mathbf{D_L}$.

$$\mathbf{D_u >> D_L}$$

And Now you want to build classifier using those data

Steps

Since, we want to build classifier and all we have right now is the small amount of labeled data

so, first step that we would do is we will take all of this labeled data let's call it as $\mathbf{D_L}$ and will build initial model $\mathbf{M_0}$

1. **$D_L \rightarrow$ Model(M_o)**

now remember we have this larger unlabeled data that we still want to use D_u so that we get improved model. So **we somehow smartly pick some points from D_u** let's call that small subset as s1, and **we obtain human labels** . And then we **retain or update the model** to get better model M_1

2. **$D_u \rightarrow$ Pick/Sample subset of (S_1) $\rightarrow$ Obtain Human Labels $\rightarrow$Retrain /Update**

M_0 **M_1**

This is what Active Learning is about.

Active Learning

So the problem of active learning is this - If you have a large amount of unlabeled data How do you **smartly pick or sample a small subset** let's say S_1 of data from this so as to maximize the lift in the model performance of M_1 as compared to M_0

Assignment Statement

You have to implement active learning algorithm on IRIS dataset.

Consider two features and implement active learning classification using SVM or k-NN classifier and show all the iteration graphically.

About Iris Dataset and Model Selection

This dataset contains data about 3 species(class) of the Iris flower having 150 samples in total and 50 samples of each species. Each sample have 4 features sepal length, sepal width, petal length and petal width , all in centimetres. Each sample is of one of these three species — Iris setosa, Iris versicolor and Iris virginica.

We choose two of the attributes (columns) from the dataset for performing active learning i.e, **Petal Length, Petal Width**

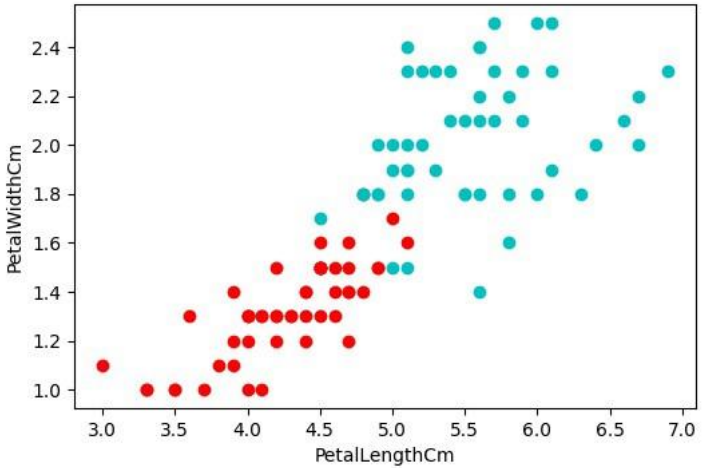
Model for Training : SVM

We here implement SVM for binary classification and therefore **discard the samples of Iris-setosa** and **choose sample that belongs to versicolor and virginica**

Now, We have 100 samples. 50 samples of each class(versicolor, virginica)

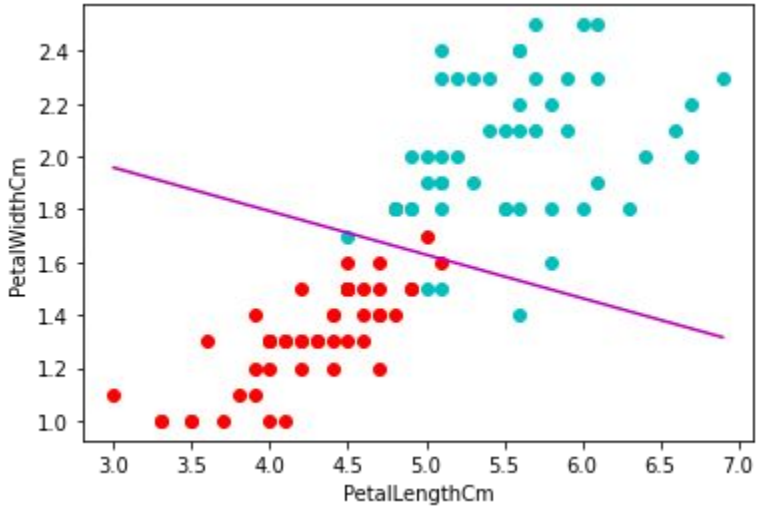
Visualisation

We plot the samples of versicolor and virginica on a 2D graph with versicolor in red and virginica in cyan.



SVM Visualisation

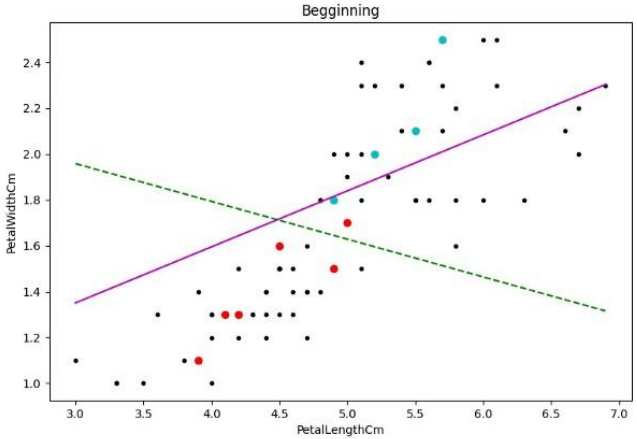
We train a Linear SVM kernel on the entire data to understand the SVM model that we would get when using all the data. Since this is a linear SVM model, the decision boundary (the boundary separating the two classes) will be a straight line.



We say, Magenta Line as ideal decision boundary

Active Learning on IRIS Dataset

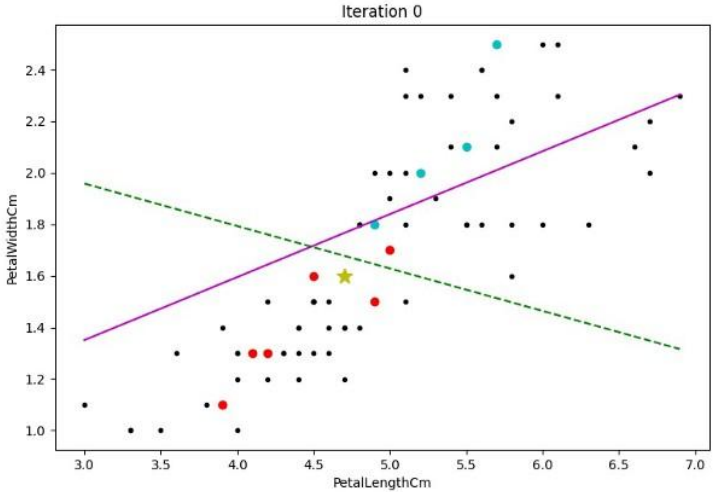
To create labelled and unlabelled data which is not available in IRIS dataset implicitly,
we split the dataset into two parts — pool(80%) and test(20%).
We take the first 10 data points of the pool as the initial train data and the rest 70 points as the unlabelled samples.
We create the beginning plot with all the unlabelled samples, the *ideal decision boundary* and the 10 train data points.



Active Learning on IRIS Dataset

Then, we train an SVM on the train data, and we find the most **ambiguous point** and create a new plot (“Iteration 0”) with this point as a yellow star and also plot the decision boundary of the trained SVM.

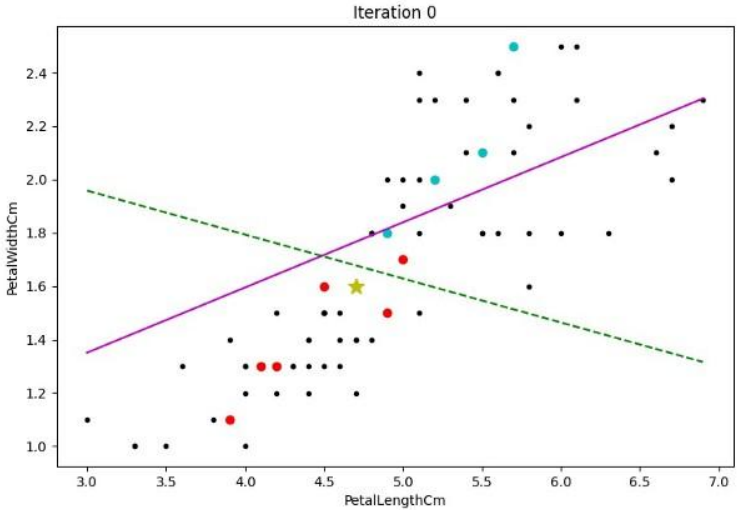
This is how we choose the data points from unlabelled data points and label it and further train.



Active Learning on IRIS Dataset

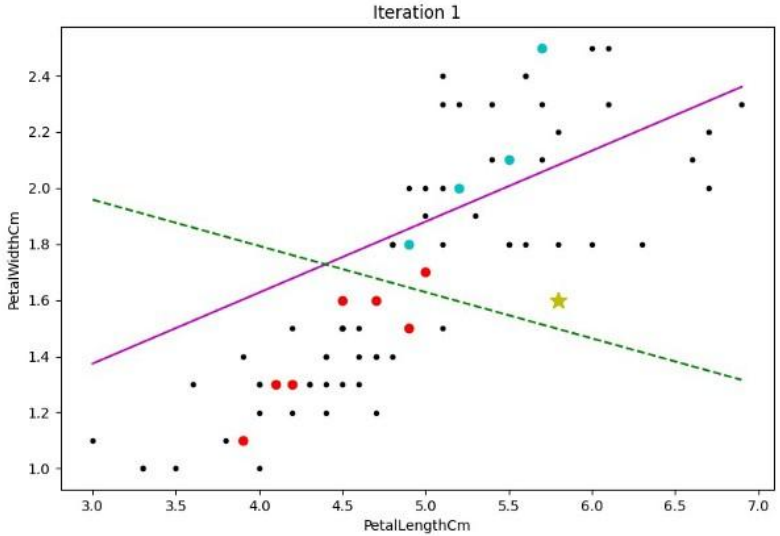
By most Ambiguous Point, what it means is- We created a function find_most_ambiguous which gives a data point that is closer to the decision boundary.
Why so?

For an SVM classifier, if a data point is closer to the decision boundary and less ambiguous if the data point is farther from the decision boundary no matter which side of the decision boundary the point is on.Thus, find_most_ambiguous, gives the unlabelled point that is the closest to the decision boundary.

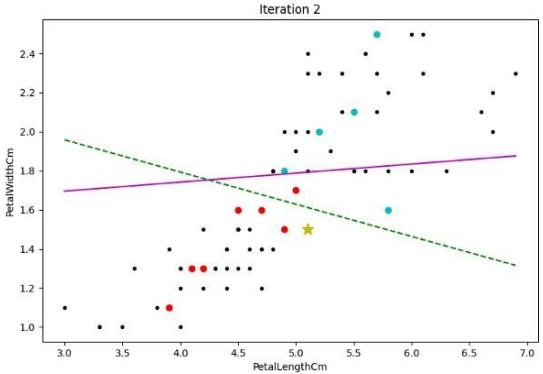


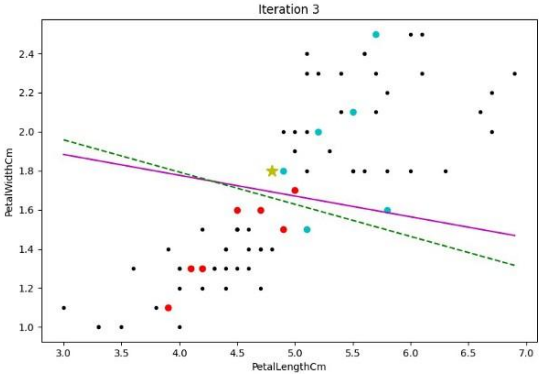
Iteration 1

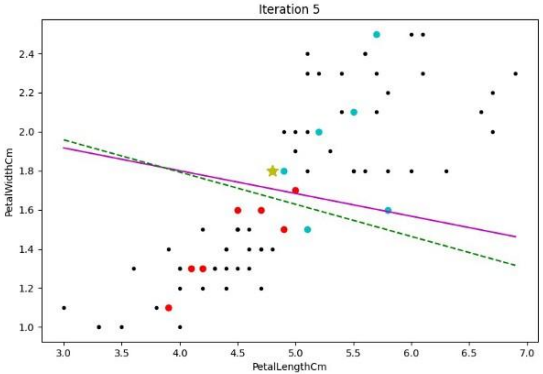
Next, we run the active learning algorithm for 5 iterations. In each of them, we add the most ambiguous point to the training data and train an SVM, find the most unambiguous point at this stage and then create a plot



Further Iterations

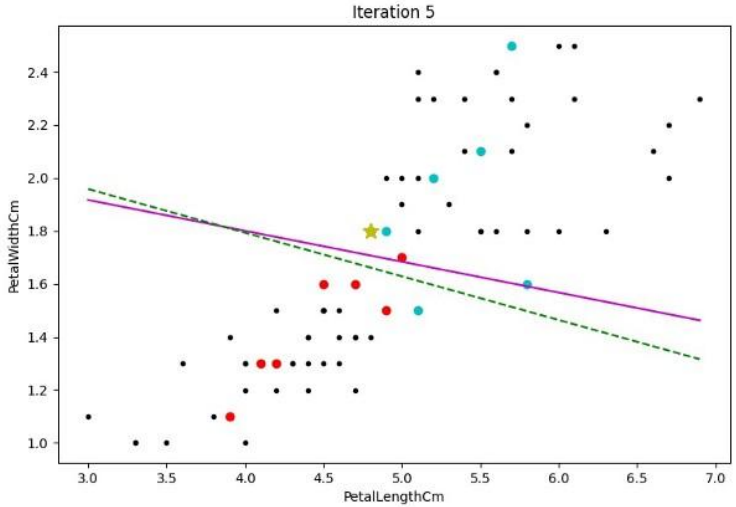






Conclusion

Hence, We see that we have trained a good classifier, that is, a classifier that would perform close to the SVM trained with all the points, although we have used a very small number of points. This is how active learning can be used to create robust models with labelling fewer data points.



References

[1] https://en.wikipedia.org/wiki/Active_learning_(machine_learning)

[2] https://towardsdatascience.com/active-learning-5b9d0955292d

Thanks !!

Attached: Code and Iris dataset in code.zip along with the assignment in google classroom.