ML-ASSIGNMENT-2(Q1)

Aayush Atul Verma 2017A7PS0061P
Jithin Kalluakalam Sojan 2017A7PS0163P
Agastya Sampath 2017A3PS0359P

The Dataset

The Iris dataset has been used for the following questions. There are 50 data points for each of the 3 labels – setosa, versicolor and virginica forming a total of 150 entries in our dataset. Each data point has four different features namely Sepal length, Sepal width, Petal length, Petal width.

Since the four features make it tough to visualize and classify, we reduce the number of dimensions to 2 using PCA to make it easier to visualize the spread of the data. Since, active learning requires a larger amount of unlabeled data as compared to labeled data, we remove labels from 90% of the entries. (135 data points) using numpy.

Tools Used

We have used Python for the assignment. The modules used to get the inbuilt methods for implementing Active Learning are as follows: numpy, modal, sklearn, matplotlib, random, math

UNCERTAINTY SAMPLING SELECTION STRATEGY

Strategy is to query the instances which the classifier is least certain how to label.

Number of labeled points in the dataset initially: 15
Number of unlabeled points in the dataset: 135
Classification algorithm used: K-nearest neighbors algorithm

Scenario 1

Stream-Based Active Learning

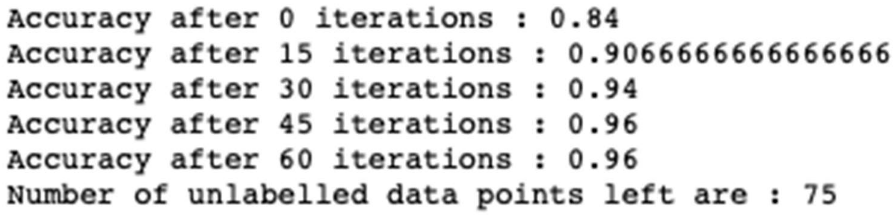
In this algorithm we sample one instance at a time and then decide whether to query it or not based on a certain threshold value. This value compared to a threshold is the informativeness measure of each instance.

This measure can be calculated by three **uncertainty sampling** techniques:

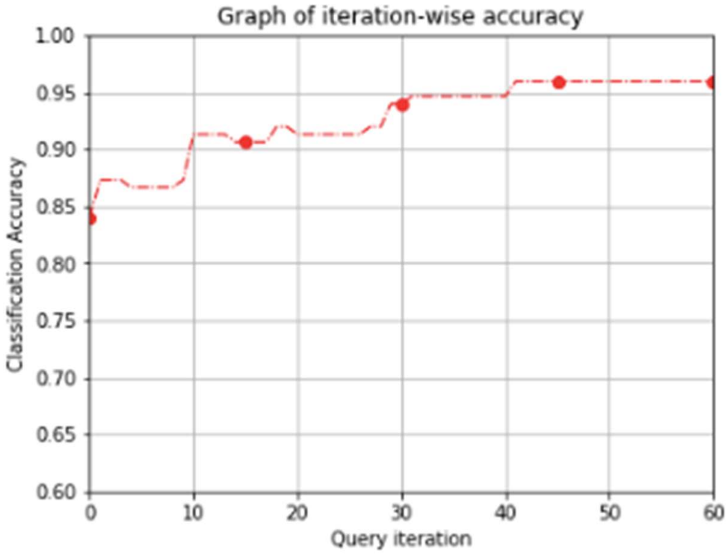
1) Least Confident strategy
2) Margin Sampling
3) Entropy

After every 15 queries, 10% additional data points are labeled. A total of 60 queries done, i.e. 40% additional data points are labeled and the accuracy after every additional labelling of 10% is printed on the console.
The threshold value used is 0.5.

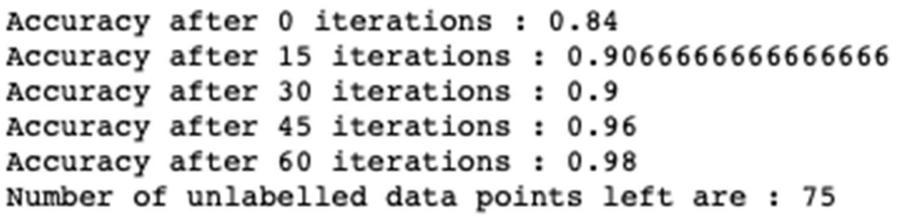
- ***Least Confident Strategy***
  Only the most probable label is considered in this strategy and the rest is ignored. This value is then compared with the strategy.



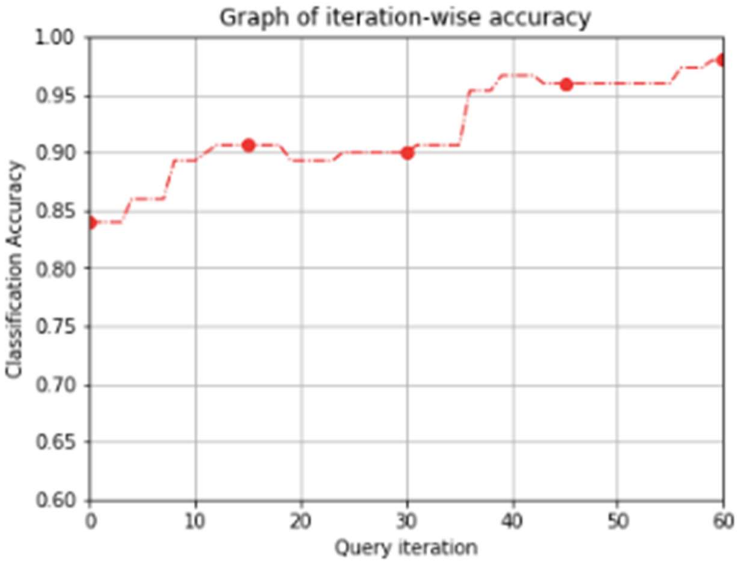
```
Accuracy after 0 iterations : 0.84
Accuracy after 15 iterations : 0.9066666666666666
Accuracy after 30 iterations : 0.94
Accuracy after 45 iterations : 0.96
Accuracy after 60 iterations : 0.96
Number of unlabelled data points left are : 75
```



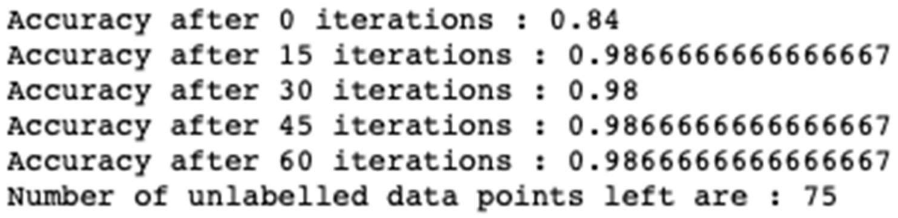
- ***Margin Sampling***
  Considers the difference between two-most probable labels only and this value is then compared with a threshold value.



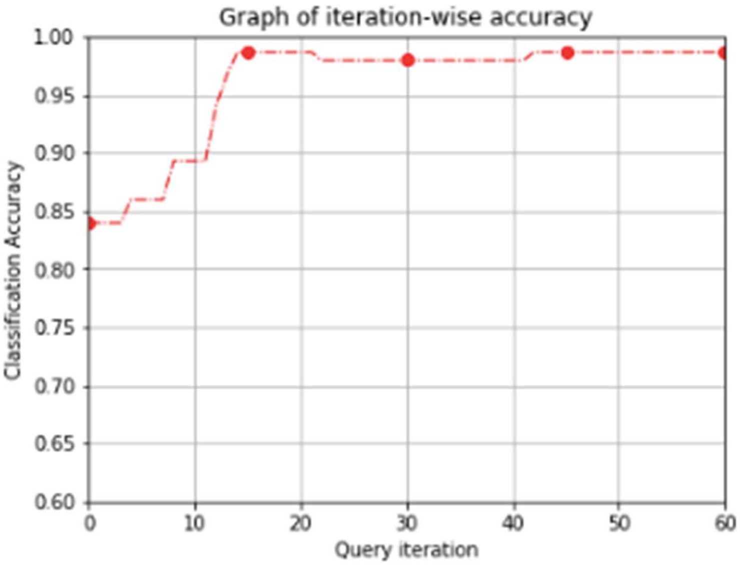
```
Accuracy after 0 iterations : 0.84
Accuracy after 15 iterations : 0.9066666666666666
Accuracy after 30 iterations : 0.9
Accuracy after 45 iterations : 0.96
Accuracy after 60 iterations : 0.98
Number of unlabelled data points left are : 75
```



- ***Entropy***
  Utilizes the entropy function on the probability of all labels of that instance to get a final value. This value is then compared with the threshold value.



```
Accuracy after 0 iterations : 0.84
Accuracy after 15 iterations : 0.9866666666666667
Accuracy after 30 iterations : 0.98
Accuracy after 45 iterations : 0.9866666666666667
Accuracy after 60 iterations : 0.9866666666666667
Number of unlabelled data points left are : 75
```



Conclusion: The three strategies give an almost similar final accuracy with entropy as a measure of informativeness marginally outperforming the other two strategies.

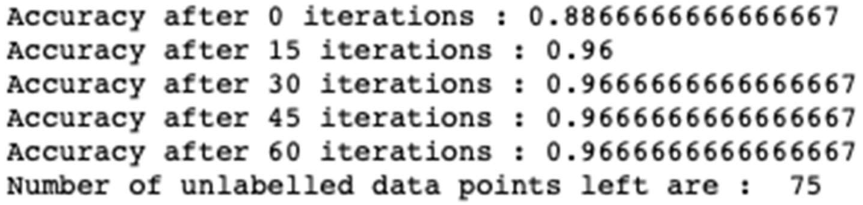
Scenario 2

Pool-Based Active Learning

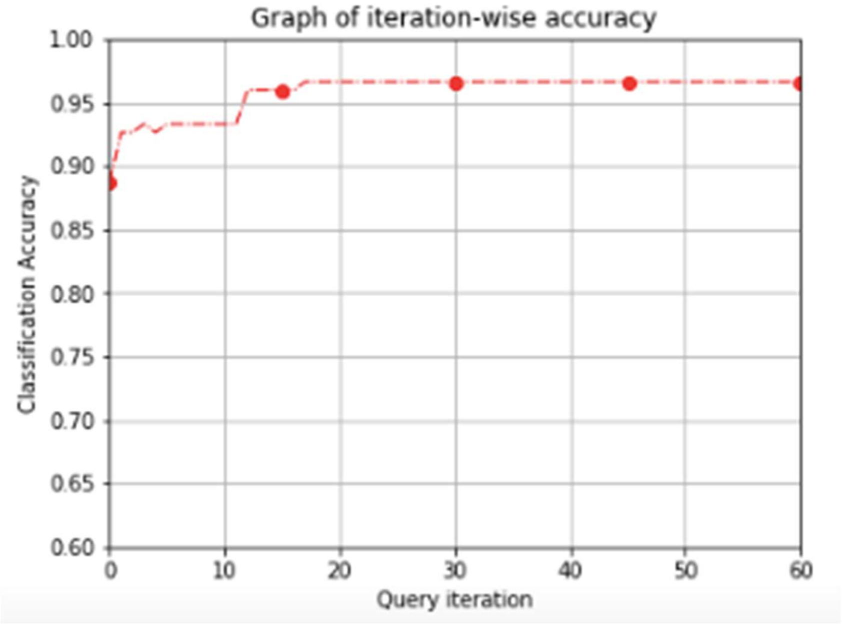
In this algorithm, there is large pool of unlabeled instances from which the best instance is sampled out to be queried. Hence the instances are queried out in a greedy manner based on its informativeness measure.

This measure can also be calculated based on the three **uncertainty sampling** techniques:

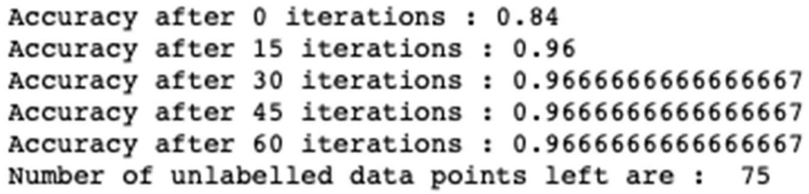
- ***Least Confident Strategy***



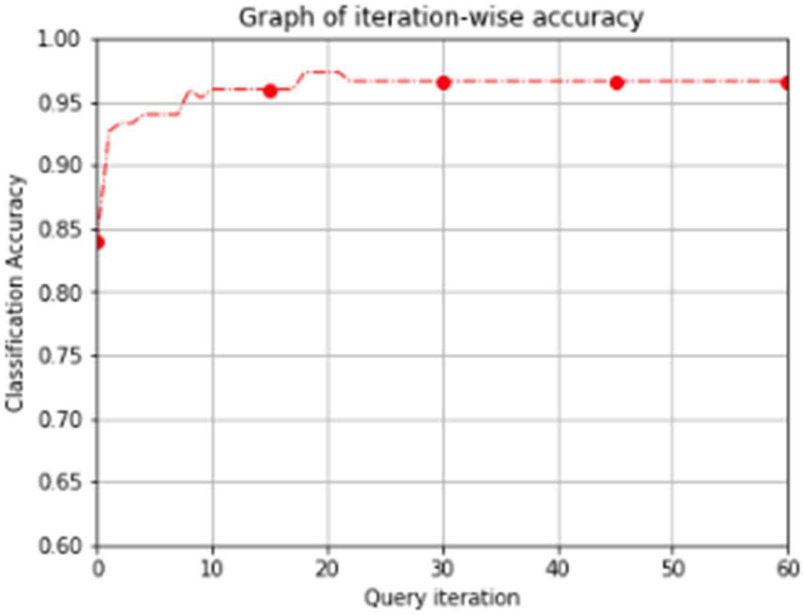
```
Accuracy after 0 iterations : 0.8866666666666667
Accuracy after 15 iterations : 0.96
Accuracy after 30 iterations : 0.9666666666666667
Accuracy after 45 iterations : 0.9666666666666667
Accuracy after 60 iterations : 0.9666666666666667
Number of unlabelled data points left are :  75
```



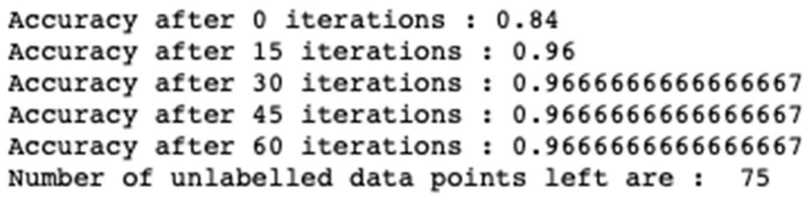
- ***Margin Sampling***



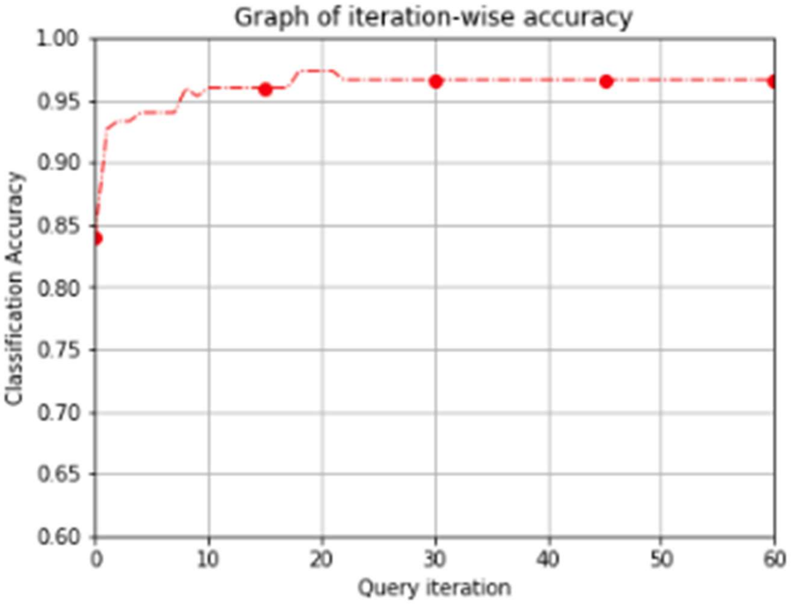
```
Accuracy after 0 iterations : 0.84
Accuracy after 15 iterations : 0.96
Accuracy after 30 iterations : 0.9666666666666667
Accuracy after 45 iterations : 0.9666666666666667
Accuracy after 60 iterations : 0.9666666666666667
Number of unlabelled data points left are :  75
```



- ***Entropy***



```
Accuracy after 0 iterations : 0.84
Accuracy after 15 iterations : 0.96
Accuracy after 30 iterations : 0.9666666666666667
Accuracy after 45 iterations : 0.9666666666666667
Accuracy after 60 iterations : 0.9666666666666667
Number of unlabelled data points left are :  75
```



Conclusion: The three strategies perform almost equivalently.
Hence entropy as a measure slightly outperforms the other two uncertainty sampling techniques.

QUERY BY COMMITTEE (QBC) SELECTION STRATEGY

In this strategy there are multiple models (committee of classifiers) involved which are all trained on the labeled data with each representing competing hypotheses. This leads to quantitative disagreements amongst the classifiers and this is used as a measure to choose an instance to query. The instance that causes maximum disagreement is likely to be most informative and is hence queried.

Number of labeled points in the dataset initially: 15
Number of unlabeled points in the dataset: 135
Classification algorithm: 5 classifiers used.

- 3 classifiers use the Random Forest algorithm
- 2 classifiers use the KNN algorithm

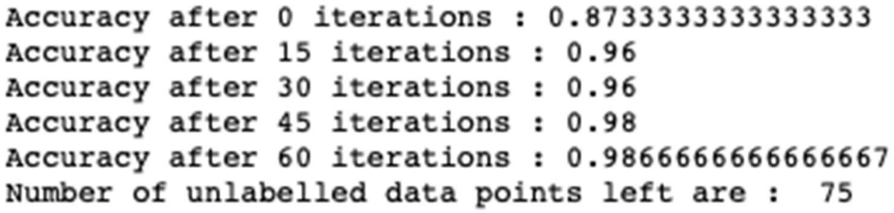
Scenario 1

Stream-Based Active Learning

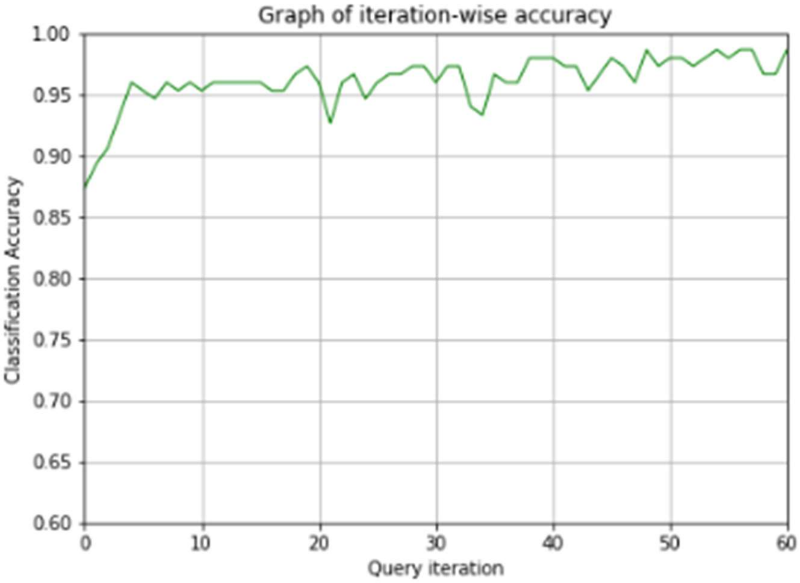
The disagreements can be measured by two techniques:

1) Vote Entropy
2) KL – Divergence

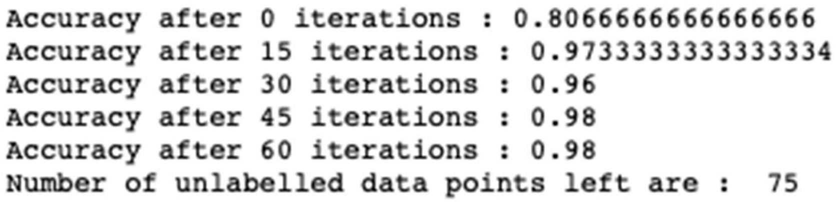
- ***Vote Entropy***



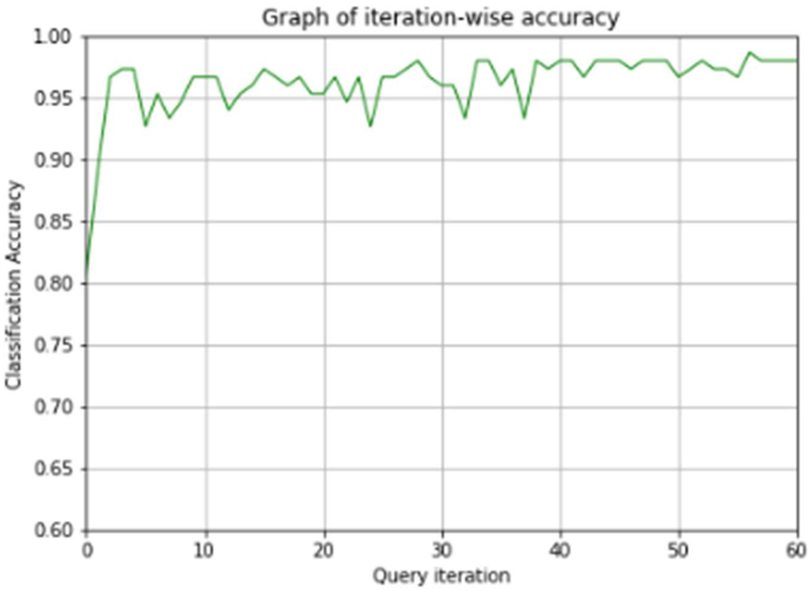
```
Accuracy after 0 iterations : 0.8733333333333333
Accuracy after 15 iterations : 0.96
Accuracy after 30 iterations : 0.96
Accuracy after 45 iterations : 0.98
Accuracy after 60 iterations : 0.9866666666666667
Number of unlabelled data points left are :  75
```



- ***KL - Divergence***



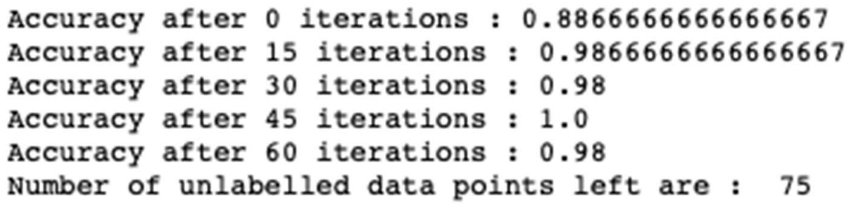
```
Accuracy after 0 iterations : 0.8066666666666666
Accuracy after 15 iterations : 0.9733333333333334
Accuracy after 30 iterations : 0.96
Accuracy after 45 iterations : 0.98
Accuracy after 60 iterations : 0.98
Number of unlabelled data points left are :  75
```



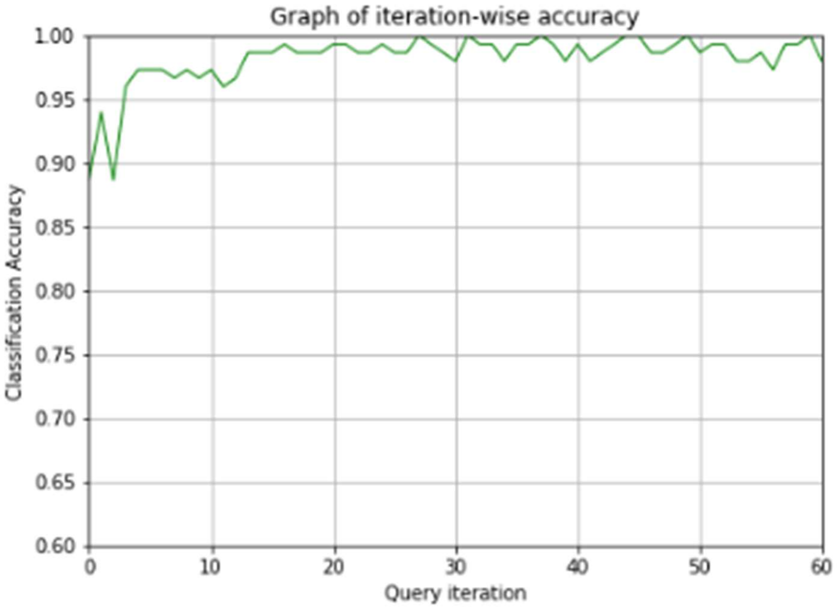
Scenario 2

Pool-Based Active Learning

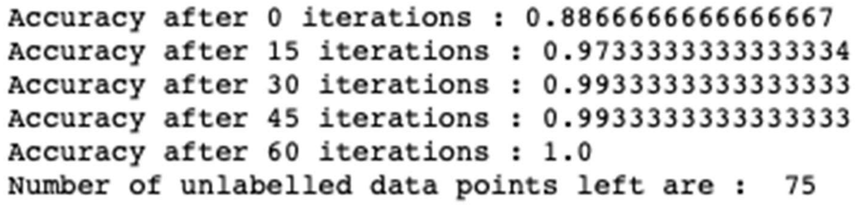
- ***Vote Entropy***



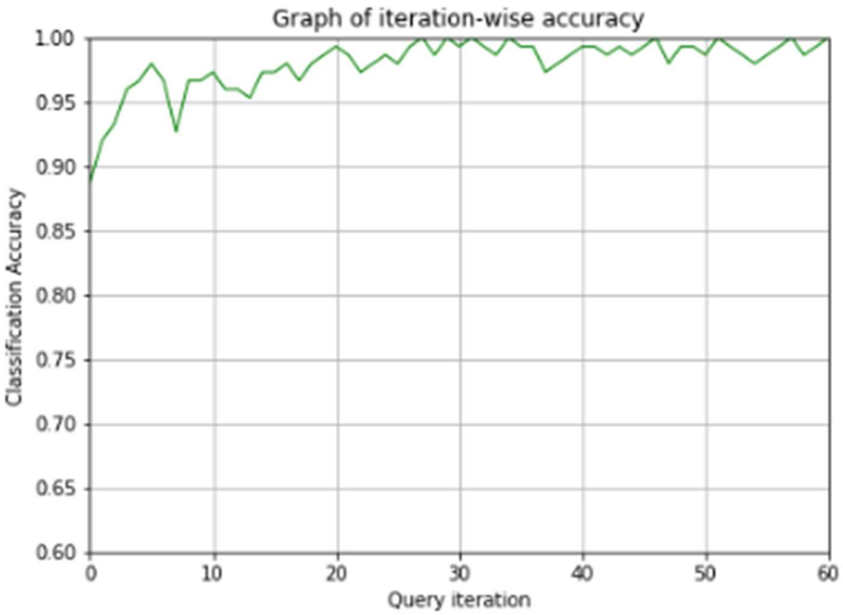
```
Accuracy after 0 iterations : 0.8866666666666667
Accuracy after 15 iterations : 0.9866666666666667
Accuracy after 30 iterations : 0.98
Accuracy after 45 iterations : 1.0
Accuracy after 60 iterations : 0.98
Number of unlabelled data points left are :  75
```



- ***KL – Divergence***

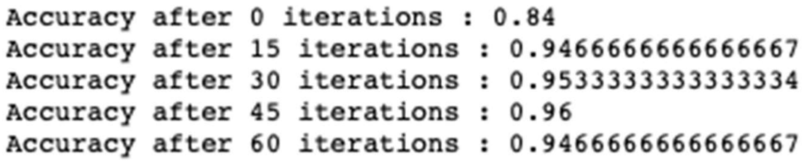


```
Accuracy after 0 iterations : 0.8866666666666667
Accuracy after 15 iterations : 0.9733333333333334
Accuracy after 30 iterations : 0.9933333333333333
Accuracy after 45 iterations : 0.9933333333333333
Accuracy after 60 iterations : 1.0
Number of unlabelled data points left are :  75
```



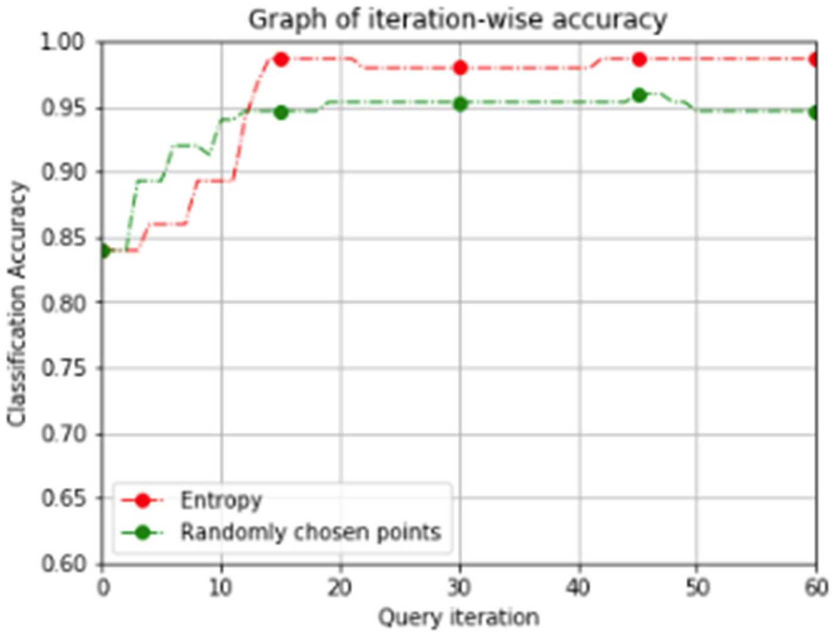
Conclusion: Both the strategies perform almost equally well. The QBC strategy outperforms the uncertainty sampling strategy.

The model that performed the best by uncertainty sampling (part (i)) is by choosing the stream-based active learning technique along with entropy as a measure of informativeness.
To compare this with randomly chosen points we use the same classifier (K-nearest neighbors algorithm used with n_neighbors=5) with the same initial training set.
The following are the results and comparison.



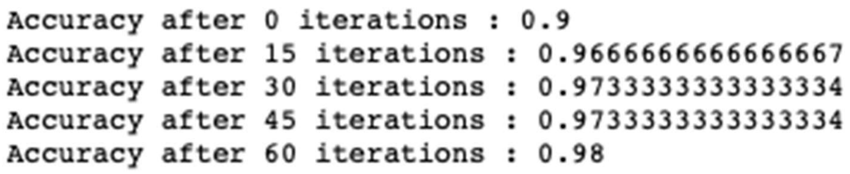
```
Accuracy after 0 iterations : 0.84
Accuracy after 15 iterations : 0.9466666666666667
Accuracy after 30 iterations : 0.9533333333333334
Accuracy after 45 iterations : 0.96
Accuracy after 60 iterations : 0.9466666666666667
```

Fig. Accuracy after training with the addition of a new random data point from the unlabeled set



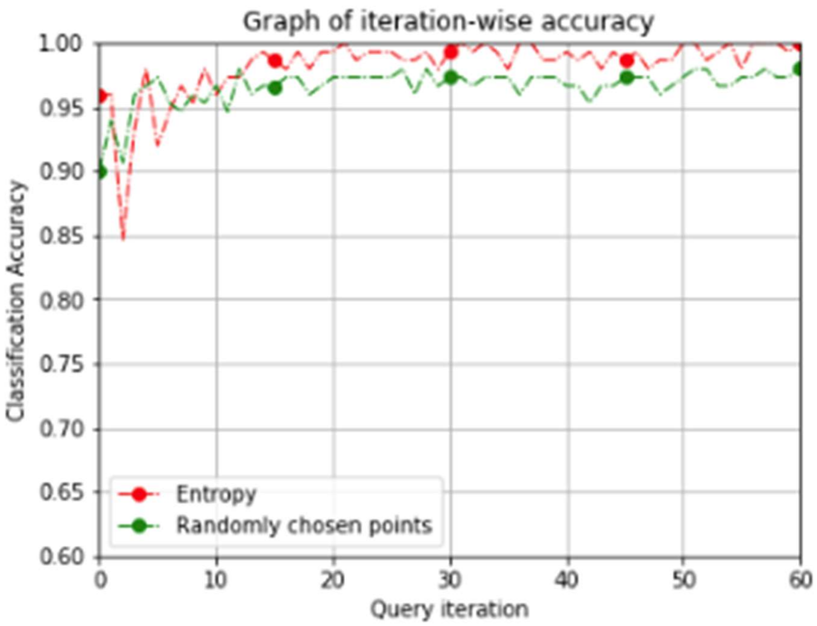
Graph. Comparison between using entropy as a measure versus choosing new points at random

The model that performed the best by the QBC strategy (part (ii)) is by choosing the pool-based active learning technique along with KL-divergence as a measure of disagreement.
To compare this with randomly chosen points we use the same committee (5 different classifiers) with the same initial training set.
The following are the results and comparison.



```
Accuracy after 0 iterations : 0.9
Accuracy after 15 iterations : 0.9666666666666667
Accuracy after 30 iterations : 0.9733333333333334
Accuracy after 45 iterations : 0.9733333333333334
Accuracy after 60 iterations : 0.98
```

Fig. Accuracy after training with the addition of a new random data point from the unlabeled set



Graph. Comparison between using KL-divergence as a measure of disagreement vs. choosing new points at random

Version Space

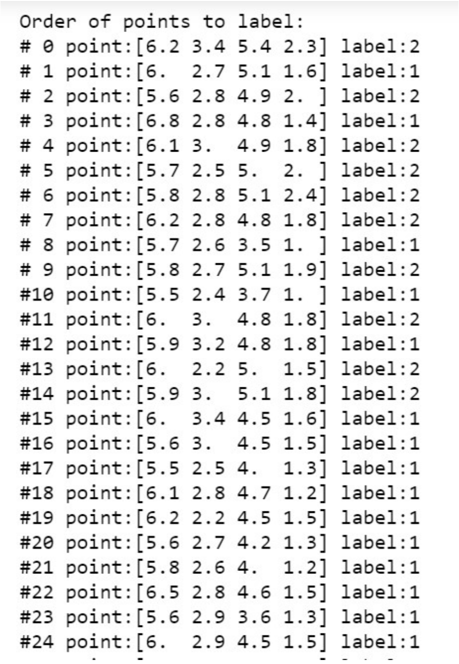
This experiment is done on the QBC in pool based active learning. The learners are the same and before the model is trained through the active learning iterations, the unlabeled points are run through the learners individually.

Now, the points are taken into a dataset if there is a disagreement among any of the models in the committee. The size of the dataset turns out to be the size of the version space. An example of the size of the version space based on the models used in the committee for QBC is given below.

```
Size of version space: 55 points.
```

The points are then compared for measure of disagreement. An assumption that almost always holds true is that the point with the maximum measure of disagreement is the one that is the most informative, and hence would reduce the version space the most.

Therefore, the points are sorted based on the measure of disagreement calculated using entropy. The version space points in the example based on the committee members of the QBC simulation are ordered based on the decreasing effect on reducing the size of the version space are given below.

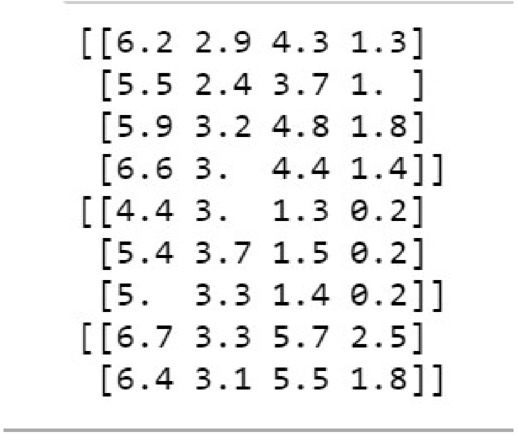


```
Order of points to label:
# 0 point:[6.2 3.4 5.4 2.3] label:2
# 1 point:[6.  2.7 5.1 1.6] label:1
# 2 point:[5.6 2.8 4.9 2. ] label:2
# 3 point:[6.8 2.8 4.8 1.4] label:1
# 4 point:[6.1 3.  4.9 1.8] label:2
# 5 point:[5.7 2.5 5.  2. ] label:2
# 6 point:[5.8 2.8 5.1 2.4] label:2
# 7 point:[6.2 2.8 4.8 1.8] label:2
# 8 point:[5.7 2.6 3.5 1. ] label:1
# 9 point:[5.8 2.7 5.1 1.9] label:2
#10 point:[5.5 2.4 3.7 1. ] label:1
#11 point:[6.  3.  4.8 1.8] label:2
#12 point:[5.9 3.2 4.8 1.8] label:1
#13 point:[6.  2.2 5.  1.5] label:2
#14 point:[5.9 3.  5.1 1.8] label:2
#15 point:[6.  3.4 4.5 1.6] label:1
#16 point:[5.6 3.  4.5 1.5] label:1
#17 point:[5.5 2.5 4.  1.3] label:1
#18 point:[6.1 2.8 4.7 1.2] label:1
#19 point:[6.2 2.2 4.5 1.5] label:1
#20 point:[5.6 2.7 4.2 1.3] label:1
#21 point:[5.8 2.6 4.  1.2] label:1
#22 point:[6.5 2.8 4.6 1.5] label:1
#23 point:[5.6 2.9 3.6 1.3] label:1
#24 point:[6.  2.9 4.5 1.5] label:1
```

Labeling Using K-Means Clustering

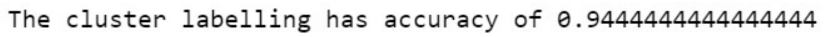
The labelling of points using K-Means is done by first taking 40% of the unlabeled data points. After that, using the K-Means model from sklearn, the points are clustered into 3.

Then 20% of the points are taken from each cluster. These representative points in an example run can be seen here. Each outer array represents a cluster.



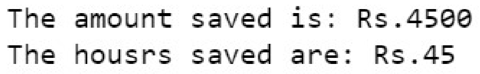
```
[[6.2 2.9 4.3 1.3]
 [5.5 2.4 3.7 1. ]
 [5.9 3.2 4.8 1.8]
 [6.6 3.  4.4 1.4]]
[[4.4 3.  1.3 0.2]
 [5.4 3.7 1.5 0.2]
 [5.  3.3 1.4 0.2]]
[[6.7 3.3 5.7 2.5]
 [6.4 3.1 5.5 1.8]]
```

The labels of these representative points are considered for each cluster and every point is given the label that has the maximum vote. Then the accuracy of the labelling is calculated by comparing with the actual labels of the points. The accuracy printed for the sample run is as follows.



```
The cluster labelling has accuracy of 0.9444444444444444
```

The cost saved is considered as the amount and time it would take to label all 40% of the points and the amount and time it would take to label only 20% of each cluster. An example of this is given from the sample run.



```
The amount saved is: Rs.4500
The housrs saved are: Rs.45
```