Project 04: Decision Tree, Naive Bayes, K-means.

Rohan Bharadwaj (109758985)
Shashank Jain (109956091)
Ashish Goel (109753528)

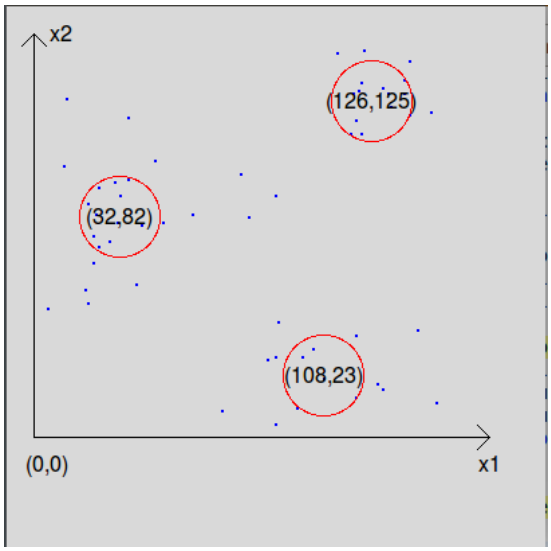
Analysis

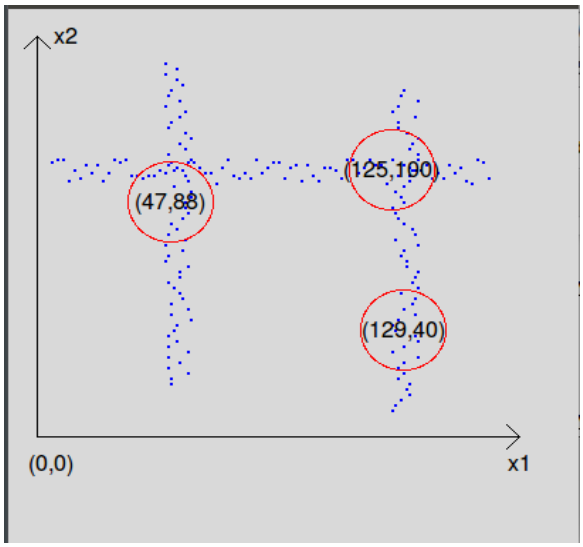
python learning.py -q1 -q2 -q3.1 -q3.2
Question 1 accuracy: 0.924012
Question 2 accuracy: 0.927052

Question 3.1 centroid: (32,82)
Question 3.1 centroid: (108,23)
Question 3.1 centroid: (126,125)
Press any key to continue.

Question 3.2 centroid: (47,88)
Question 3.2 centroid: (129,40)
Question 3.2 centroid: (125,100)
Press any key to continue.

Note: we have taken initial centroid as [(30, 30), (150, 30), (90, 130)]:





Decision Tree

Description

We have used ID3 algorithm to achieve our goal. We're also maintaining a default value at every node, where default points to the maximum occurrence of a classifier,say, 'democrat' or 'republican; from the dataset at the node. This has been done, because during problem solving stage while traversing down the decision tree,if there is no new path for a particular feature then we return the default value instead of traversing further down.

Other than that, it is just basic decision tree implementation where we are splitting on max-gain attributes at each level. Once we reach a node where all classifier results are same, or there are no attributes left then we return from that node, choosing default value as a result for that path.

Results

python learning.py -q1
Question 1 accuracy: 0.924012

Naive Bayes

Description

In the learning phase, we basically store a dictionary with key containing both of the classifiers,i.e, republican and democrat's. This dictionary consists of the conditional probability of each of the possible feature value given a classifier. At the problem solving step, we just multiply the probability found in learning phase for each of the given values of the features, to eventually find out the classifier with maximum probability of occurrence.
We've used additive smoothing to smoothen out our results from Naive Bayes. The value of k chosen is given below. The formula we have chosen for smoothing is as follows:-

Results:

When democrat is chosen for tiebreak:-

python learning.py -q2
For k=0.000634514758844 (With Smoothing)
Question 2 accuracy: 0.927052

(Without Smoothing)
Question 2 accuracy: 0.920973

When republican is chosen for tiebreak:-

python learning.py -q2
For k=0.000634514758844 (With Smoothing)
Question 2 accuracy: 0.927052

(Without Smoothing)
Question 2 accuracy: 0.884498

K-means

Description

We are clustering two set of points in different regions based on the **Euclidean distance** between the points and the initial centroids. We are recomputing centroids and repeating the process of clustering with the new centroids, till we get no more change in the centroid values.

Results

With initial centroids as [(30, 30), (150, 30), (90, 130)]:
python learning.py -q3.1 -q3.2
Question 3.1 centroid: (32,82)
Question 3.1 centroid: (108,23)
Question 3.1 centroid: (126,125)

Question 3.2 centroid: (47,88)
Question 3.2 centroid: (129,40)
Question 3.2 centroid: (125,100)

With initial centroids as [(30, 60), (150, 60), (90, 130)]:
Question 3.1 centroid: (32,82)
Question 3.1 centroid: (108,23)
Question 3.1 centroid: (126,125)

Question 3.2 centroid: (48,47)
Question 3.2 centroid: (128,80)
Question 3.2 centroid: (48,103)