**CSE 537 - Artificial Intelligence**

**Report: Project 4**

**Khan Mostafa**

109365509

**Abhijit Betigeri**

109229784

{khan.mostafa, abhijit.betigeri}@stonybrook.edu

Department of Computer Science

Stony Brook University

## **Decision Tree, Naïve Bayes, K-means**

## **Q1. Implementing the Decision Tree Algorithm**

**Methodology Used:** We use the ID3 to build a decision tree from the fixed set of data instances given. The resulting tree is used to classify the future samples – since classes created by ID3 are inductive, here we use the training set given, and classes created by ID3 are used for all future instances . The leaf nodes of the decision tree contains class name & non-leaf node contains the decision made. The decision node here is the attribute test with each branch being the possible value of the attribute. ID3 used information gain logic to see which attribute goes into the decision node.

We use the property of information gain for attribute selection describing how well the given attributes separates the training examples into targeted classes. The one with the highest information is selected. To define gain we use the concept of entropy – signifying the amount of information in an attribute.

**Execution Details**

We have reporting the results till the threshold of 50.

**Command:**

python learning.py –q1

w/th 0: Question 1 accuracy: 0.920973

w/th 1: Question 1 accuracy: 0.920973

w/th 2: Question 1 accuracy: 0.920973

w/th 3: Question 1 accuracy: 0.942249

w/th 4: Question 1 accuracy: 0.942249

w/th 5: Question 1 accuracy: 0.942249

w/th 6: Question 1 accuracy: 0.936170

w/th 7: Question 1 accuracy: 0.936170

w/th 8: Question 1 accuracy: 0.936170

w/th 9: Question 1 accuracy: 0.957447

w/th 10: Question 1 accuracy: 0.957447

w/th 11: Question 1 accuracy: 0.957447

w/th 12: Question 1 accuracy: 0.957447

w/th 13: Question 1 accuracy: 0.957447

w/th 14: Question 1 accuracy: 0.957447

w/th 15: Question 1 accuracy: 0.957447

w/th 16: Question 1 accuracy: 0.957447

w/th 17: Question 1 accuracy: 0.957447

w/th 18: Question 1 accuracy: 0.957447

w/th 19: Question 1 accuracy: 0.957447

w/th 20: Question 1 accuracy: 0.957447

w/th 21: Question 1 accuracy: 0.957447

w/th 22: Question 1 accuracy: 0.957447

w/th 23: Question 1 accuracy: 0.957447

w/th 24: Question 1 accuracy: 0.957447

w/th 25: Question 1 accuracy: 0.957447

w/th 26: Question 1 accuracy: 0.957447

w/th 27: Question 1 accuracy: 0.957447

w/th 28: Question 1 accuracy: 0.957447

w/th 29: Question 1 accuracy: 0.957447

w/th 30: Question 1 accuracy: 0.957447

w/th 31: Question 1 accuracy: 0.957447

w/th 32: Question 1 accuracy: 0.957447

w/th 33: Question 1 accuracy: 0.957447

w/th 34: Question 1 accuracy: 0.957447

w/th 35: Question 1 accuracy: 0.957447

w/th 36: Question 1 accuracy: 0.957447

w/th 37: Question 1 accuracy: 0.957447

w/th 38: Question 1 accuracy: 0.957447

w/th 39: Question 1 accuracy: 0.957447

w/th 40: Question 1 accuracy: 0.957447

w/th 41: Question 1 accuracy: 0.957447

w/th 42: Question 1 accuracy: 0.957447

w/th 43: Question 1 accuracy: 0.957447

w/th 44: Question 1 accuracy: 0.957447

w/th 45: Question 1 accuracy: 0.957447

w/th 46: Question 1 accuracy: 0.957447

w/th 47: Question 1 accuracy: 0.957447

w/th 48: Question 1 accuracy: 0.957447

w/th 49: Question 1 accuracy: 0.957447

**Q2. Implementing the Naïve Bayes Algorithm**

**Methodology Used:** In Naïve Bayes classifier an assumption that the presence of a particular feature of the class is unrelated to the presence of any other feature. Using the Bayesian interpretation there is a linking of the degree of belief in a proposition before and after accounting for evidence.

We use a technique called laplace smoothing for the parameter estimation which accounts for the unobserved event. The technique is more robust and will not fail completely when the data that has never been observed in the training shows up.

**Execution Details**

* **The accuracy with smoothing enabled – 0.902736:**

python learning.py –q2

**Accuracy: 0.902736**

* **The accuracy without smoothing – 0.884498:**

python learning.py –q2

**Accuracy: 0.884498**

**Add for democrat/republic default ??**

**Q3. Implementing the K-means Algorithm**

**Methodology Used:** This unsupervised learning algorithm is used in classifying the data set through a certain number of clusters fixed a priori.

The steps used are:

1. We place K points into the space represented by the objects that are being clustered. Use the initial set of the centroids already given.
2. Assign each object to the group that has the closest centroid.
3. As soon all the objects have been assigned, recalculate the positions of the K centroids.
4. Repeat steps 2 and 3 untill centroids no longer move or change. This helps in separation of the objects into groups from which the metric to be minimized is calculated.

**Execution Details**

**Command 1:**

python learning.py –q3.1 –q3.2

(With two different centroid combinations:

centroids = [(30, 60), (150, 60), (90, 130)]

centroids = [(30, 30), (150, 30), (90, 130)])

**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)**

**Appendix:**

Put all results here.