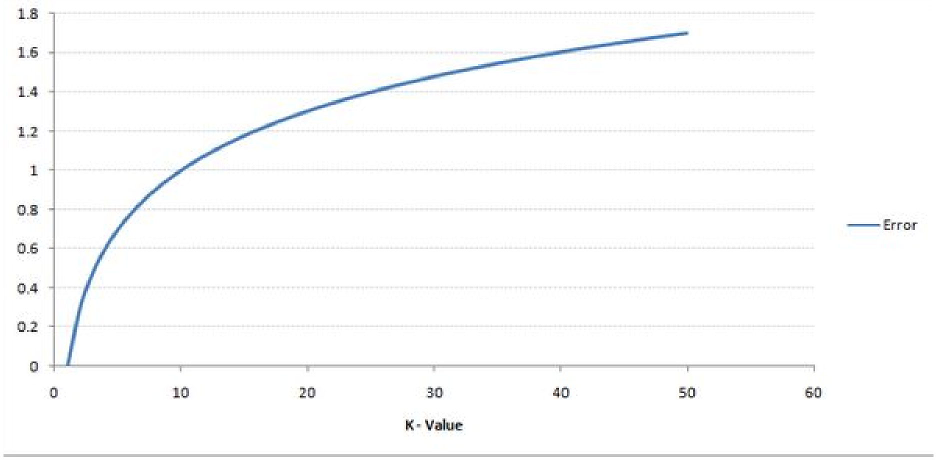
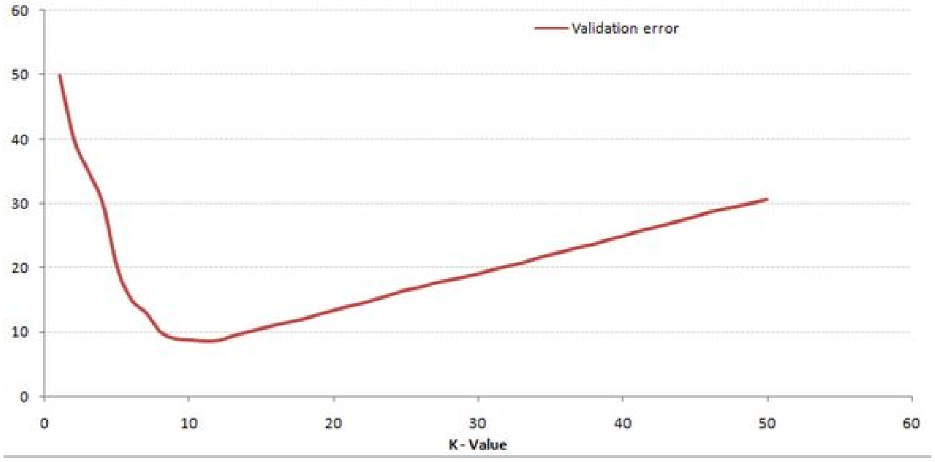
REPORT
ASSIGNMENT 1 - AML(CS6510)

Note: **Please read readme to get a description of the files in folder 3 and 4.**

1. KNN Question
   a) As k varies from 1 to n the training error increases. It is zero for k=1 as the node is nearest neighbour of itself.



   b) Generalisation error first decreases then it reaches a minima and then increases as k varies from 1 to n. If k is too small, result is sensitive to noise points. If k is too large, neighborhood may include points from other classes.



   c) i) In higher dimensions, all data points lie on the surface of the unit hypersphere! This makes euclidean distance a meaningless metric. Euclidean distance is unhelpful in high dimensions because all vectors are almost equidistant to the search query vector (imagine multiple points lying more or less on a circle with the query point at the center; the distance from the query to all data points in the search space is almost the same).

ii) A high dimensional space requires a lot of memory and is many times computationally infeasible to compute.

d) No, it’s not possible to achieve this. The classification of the mentioned Decision Tree will not be exactly similar as that of KNN (K=1). This is because Decision Tree divides the region in the form of grid. On the other hand KNN in KNN surface of detection is a circle(using euclidean distance metric).

2. Solved in the coming images

2) (a) class 1 ⇒ $\{0.5, 0.1, 0.2, 0.4, 0.3, 0.2, 0.1, 0.2$
$0.35, 0.25\}$

class 2 ⇒ $\{0.9, 0.8, 0.75, 1.0\}$

number of samples in class 1 is 10. Class 2 has 4 samples.

$$\boxed{\text{We know that :- } \; p(c/x) = \frac{p(x|c)\,p(C)}{p(x)}}$$

μ_1: mean of class 1 ; μ_2: mean of class 2

σ_1^2: varionce of class 1 ; σ_2^2: varianu of class 2

$\mu_1 = 0.26$; $\mu_2 = 0.8625$

$\sigma_1^2 = 0.0149$; $\sigma_2^2 = 0.0092$

mean of sample is the maximum likelihood estimator for a Gaussian mean

$$P(x) = p(x/c_1)\,p(c_1) + p(x/c_2)\,p(c_2)$$

$$P(x|c_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \times \exp\left(\frac{(x-\mu_k)^2}{-2\sigma_k^2}\right)$$

Putting values we get ~~$P(C_1|x=0.6)$~~

$$\boxed{P(C_1 \mid x == 0.6) = 0.63}$$

$$\boxed{\begin{aligned} P(C_1) &= 10|14 \\ P(C_2) &= 4|14 \end{aligned}}$$

⇒ Class Probabilities

(b) Vector of attributes for document x =

$$[1, 0, 0, 1, 1, 1, 1, 0]$$

attributes: [goal, football, golf, defence, offense, wicket, office, strategy]

$$P(\text{Politics} \mid x) = \frac{P(x \mid \text{politics}) \times P(\text{politics})}{P(x)} \quad - \text{①}$$

As there is one column ~~in sport~~ with all zeroes, we can use laplace smoothing. 1 is added to every attribute value - class combination.

$$P(x) = P(x \mid \text{politics})\, P(\text{politics}) + P(x \mid \text{sports})\, P(\text{sports})$$

$$P(\text{politics}) = 1/2$$

$$P(\text{sports}) = 1/2$$

$$P(x \mid \text{politics}) = \frac{3 \times 6 \times 6 \times 6 \times 6 \times 2 \times 5 \times 2}{8^8}$$

$$P(x \mid \text{sports}) = \frac{5 \times 3 \times 6 \times 5 \times 2 \times 2 \times 1 \times 6}{8^8}$$

Putting values in ① we get,

$$P(\text{politics} \mid x) = 0.878$$

3. Decision Tree was implemented for this question. The trees were evaluated with following metrics(Accuracy is measured out of 1):
   i) Information Gain using Entropy
   ii) Pruning and Information Gain using entropy
   iii) Gini Index
   iv) Gini index and pruning
   a) Accuracy of Information Gain using entropy metric was **0.8134**. A decline was observed when Gini index was used as metric. It reduced to **0.8116**. Pruning with information gain(Entropy) was the best performing algorithm. It had an accuracy of **0.8226** for threshold value of 0.43. Gini with pruning performed with an accuracy of **0.8175** for threshold value of 0.17. Overall the order of accuracy was**(Pruning and Information Gain(Entropy)>Pruning and Gini>Information Gain(Entropy)>Gini Index)**.
   b) When pruning is introduced alongside any of the impurity metrics the accuracy should increase as they improve the generalisation performance of the algorithm. Decision Tree without pruning may overfit on the training data. Therefore we are observing an improvement in performance as compared to using any of the impurity metric alone.

      Generalisation performance of Gini Index and Entropy metric may vary from data-set to data-set. Gini impurity and Information Gain Entropy are pretty much the same. And people do use the values interchangeably. Generally, your performance will not change much whether you use Gini impurity or Entropy. However, Entropy evaluation is a little bit slower than Gini Index evaluation because of high computation cost of calculating logarithm Function.

      Since, the attributes have continuous values we have create a split using some metric. Generalisation performance of Gini index or Entropy also depends on the way we are making the split. In my code slightly changing the split metric improves the performance for Gini index as compared to Entropy.

4.
   a) Top two scores: 0.61735(Decision Tree), 0.75553(Multinomial naive bayes).
   b) The number of ingredients(features) in training data is 6714. It is very dimension for some algorithms which gets easily struct by curse of dimensionality. KNN algorithm gets hit by curse of dimensionality in this problem. Therefore It had the worst performance as compared to the other two.

      Decision Trees tends to overfit on the data. Hence their, generalisation performance is observed lower than naive bayes classifier.

      Naive bayes tend to outperform Decision Trees when the number of classification labels(possible target classes) is high. Therefore, we observed a better score for naive bayes classifier against Decision Trees.

      The calling function for KNN is commented. If you want to create a file corresponding to it then please uncomment line 100 of the file 4.py.