Assignment 3

Task I. Basic concepts (10 points each, total 20 points)

1. Briefly outline the idea and major steps of decision tree classification.

A decision tree is an example of a classification that uses a tree data structures to determine the final decision / classification, where the leaf nodes acts as all of the final decisions

In order to make a decision tree, ① first we need to determine the classes we are trying to predict.

② Second, we build the actual decision tree out of the features. In order to achieve the most optimized decision tree, we must consider the purity of the dataset and it's feature.

③ To inspect the dataset's purity, we use it's entropy value, which we can get by calculating:

$$H[D] = -\sum_{j=1}^{|C|} P(c_j) \log_2 P(c_j)$$

④ Next, we calculate the entropy for each feature, which we can get from:

$$H_{A_i}[D] = \sum_{j=1}^{v} \frac{|D_j|}{|D|} H[D_j]$$

the smaller the entropy, the purer the split

(5) After that, we determin which attribute is the most optimized decision boundary by subtracting the attribut'es entropy from the overall dataset entropy, where the attribute with the biggest difference in entropy (highest information gain) will get picke as a node in the decision tree

(6) Finally, we build the tree recursively until there's no more impure attributes / splits

2. What is the basic idea of k-nearest neighbor? How to compute distance between two points?

KNN is the idea of looking at clusters of labeled data points, and trying to predict (label) a new datapoint based on how many similar labeled datapoints around it.

Then, we take the k amount of the nearest datapoints and determine the class of the new point based on the majority of the k neighbors. Depending on the implementation, we can consider the actual distance of the neighbor and the new datapoint as weights to give nearer datapoints (more similar) an advantage.

Distance in KNN can be measured with various options such as:

- Euclidian distance

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

- Cosine distance

$$d(x, y) = x \cdot y$$

- etc

Task II. Naïve Bayes classification (30 points)

Manufacturer	RAM	CAPACITY	WARRANTY	BATTERY	COST	BUY
DELL	4GB	500GB	0	4hrs	LOW	**NO**
TOSHIBA	8GB	1TB	1	12hrs	HIGH	**YES**
SAMSUNG	16GB	1TB	1	4hrs	LOW	**YES**
LENEVO	8GB	2TB	0	12hrs	LOW	**NO**
TOSHIBA	16GB	500GB	2	12hrs	LOW	**YES**
SONY	16GB	1TB	2	4hrs	HIGH	**YES**
SAMSUNG	4GB	1TB	0	12hrs	LOW	**NO**
LENEVO	8GB	500GB	1	8hrs	LOW	**NO**
APPLE	4GB	2TB	2	12hrs	LOW	**YES**
DELL	16GB	2TB	0	8hrs	LOW	**YES**
SONY	8GB	500GB	1	18hrs	HIGH	**NO**
TOSHIBA	4GB	4TB	2	18hrs	VERY HIGH	**NO**
HP	16GB	2TB	2	4hrs	LOW	**NO**
APPLE	4GB	1TB	0	18hrs	HIGH	**NO**
ASUS	16GB	500GB	3	8hrs	HIGH	**YES**
SAMSUNG	8GB	2TB	0	18hrs	HIGH	**NO**
DELL	4GB	4TB	1	8hrs	LOW	**YES**
HP	16GB	500GB	2	8hrs	LOW	**YES**
SAMSUNG	4GB	2TB	0	18hrs	HIGH	**NO**
DELL	8GB	4TB	1	18hrs	HIGH	**YES**
HP	16GB	1TB	3	4hrs	HIGH	**NO**
ASUS	4GB	500GB	0	12hrs	HIGH	**NO**
SONY	16GB	4TB	1	8hrs	LOW	**YES**
HP	8GB	2TB	3	12hrs	LOW	**YES**
SAMSUNG	4GB	4TB	2	8hrs	LOW	**YES**
TOSHIBA	16GB	500GB	3	18hrs	HIGH	**YES**
APPLE	4GB	1TB	0	4hrs	HIGH	**NO**
ACER	4GB	4TB	1	12hrs	LOW	**YES**
ASUS	4GB	8TB	3	12hrs	LOW	**YES**
DELL	8GB	500GB	3	4hrs	LOW	**NO**
HP	16GB	4TB	3	18hrs	VERY HIGH	**YES**
ACER	8GB	4TB	0	8hrs	HIGH	**NO**
HP	16GB	8TB	3	8hrs	VERY HIGH	**YES**
HP	8GB	1TB	1	18hrs	LOW	**YES**
APPLE	8GB	500GB	1	8hrs	LOW	**NO**
TOSHIBA	8GB	8TB	2	18hrs	VERY HIGH	**YES**
HP	16GB	8TB	3	12hrs	VERY HIGH	**YES**
ACER	8GB	1TB	0	4hrs	LOW	**NO**
APPLE	4GB	8TB	3	12hrs	HIGH	**YES**
SONY	16GB	8TB	2	4hrs	HIGH	**YES**
LENEVO	16GB	1TB	3	18hrs	LOW	**YES**
ACER	8GB	8TB	3	12hrs	HIGH	**YES**
HP	4GB	2TB	1	8hrs	LOW	**YES**

ASUS	8GB	8TB	3	18hrs	VERY HIGH	**NO**
HP	8GB	4TB	2	4hrs	LOW	**NO**
ACER	16GB	8TB	2	12hrs	HIGH	**YES**
SONY	4GB	2TB	0	4hrs	LOW	**NO**
LENEVO	16GB	4TB	3	8hrs	VERY HIGH	**YES**
ASUS	16GB	2TB	1	12hrs	HIGH	**YES**
HP	4GB	8TB	2	4hrs	LOW	**NO**

Figure 1: Training Data Set

Manufacturer	RAM	CAPACITY	WARRANTY	BATTERY	COST	BUY (Actual Label)
DELL	8GB	500GB	0	4hrs	LOW	NO
LENEVO	16GB	1TB	3	8hrs	HIGH	YES
HP	4GB	2TB	1	12hrs	LOW	YES
APPLE	8GB	4TB	2	4hrs	HIGH	NO
ASUS	16GB	500GB	0	18hrs	VERY HIGH	YES
DELL	8GB	8TB	2	8hrs	LOW	YES
TOSHIBA	4GB	1TB	1	12hrs	VERY HIGH	NO
ACER	16GB	4TB	2	8hrs	HIGH	YES
SONY	8GB	2TB	3	18hrs	VERY HIGH	YES
SAMSUNG	4GB	8TB	2	8hrs	HIGH	NO

Figure 2: Testing Data Set

Consider the above training and testing data sets. The training data set contains 50 data points and the testing data set contains 10 data points.

All the attributes in the data set are nominal. The attributes along with their possible nominal values are shown below.

'Manufacturer' {'DELL', 'LENEVO', 'HP', 'APPLE', 'ASUS', 'TOSHIBA', 'ACER', 'SONY', 'SAMSUNG'}

'RAM' {'4GB', '8GB', '16GB'}

'CAPACITY' {'500GB', '1TB', '2TB', '4TB', '8TB'}

'WARRANTY' {'0', '1', '2', '3'}

'BATTERY' {'4hrs', '8hrs', '12hrs', '18hrs'}

'COST' {'LOW', 'HIGH', 'VERY HIGH'}

'BUY' {'YES', 'NO'}

The 'BUY' attribute is the label that tells us if a customer would buy a laptop or not considering the values of the other attributes.

1. **(20 total)** Use the Naïve Bayes Algorithm to build a model using the training data set and use this model to predict the label, which is the 'BUY' attribute for the testing data set.

$$P(C_i|X) = \frac{P(X|C_i)\,P(C_i)}{P(X)}$$

→ we're just comparing them so no need for denominator

n = 50

Manufacturer

```
P(DELL | YES) = 3/29  = 0.1034
P(DELL | NO)  = 2/21  = 0.0952

P(TOSHIBA | YES) = 4/29      = 0.1379
P(TOSHIBA | NO)  = 1/21      = 0.0476

P(SAMSUNG | YES) = 2/29      = 0.0690
P(SAMSUNG | NO)  = 3/21      = 0.1429

P(LENEVO | YES) = 2/29       = 0.0690
P(LENEVO | NO)  = 2/21       = 0.0952

P(SONY | YES) = 3/29  = 0.1034
P(SONY | NO)  = 2/21  = 0.0952

P(APPLE | YES) = 2/29 = 0.0690
P(APPLE | NO)  = 3/21 = 0.1429

P(HP | YES) = 7/29    = 0.2414
P(HP | NO)  = 4/21    = 0.1905

P(ASUS | YES) = 3/29  = 0.1034
P(ASUS | NO)  = 2/21  = 0.0952
```

P(ACER | YES) = 3/29 = 0.1034
P(ACER | NO) = 2/21 = 0.0952

RAM
P(4GB | YES) = 7/29   = 0.2414
P(4GB | NO) = 9/21   = 0.4286

P(8GB | YES) = 6/29   = 0.2069
P(8GB | NO) = 10/21 = 0.4762

P(16GB | YES) = 16/29 = 0.5517
P(16GB | NO) = 2/21 = 0.0952

CAPACITY
P(500GB | YES) = 4/29 = 0.1379
P(500GB | NO) = 6/21 = 0.2857

P(1TB | YES) = 5/29   = 0.1724
P(1TB | NO) = 5/21   = 0.2381

P(2TB | YES) = 5/29   = 0.1724
P(2TB | NO) = 5/21   = 0.2381

P(4TB | YES) = 7/29   = 0.2414
P(4TB | NO) = 3/21   = 0.1429

P(8TB | YES) = 8/29   = 0.2759
P(8TB | NO) = 2/21   = 0.0952

WARRANTY
P(0 | YES) = 1/29   = 0.0345
P(0 | NO) = 11/21   = 0.5238

P(1 | YES) = 9/29   = 0.3103
P(1 | NO) = 3/21   = 0.1429

P(2 | YES) = 8/29   = 0.2759
P(2 | NO) = 4/21   = 0.1905

P(3 | YES) = 11/29   = 0.3793
P(3 | NO) = 3/21   = 0.1429

BATTERY
P(4hrs | YES) = 3/29 = 0.1034
P(4hrs | NO) = 9/21 = 0.4286

P(12hrs | YES) = 11/29   = 0.3793
P(12hrs | NO) = 3/21 = 0.1429

P(8hrs | YES) = 9/29 = 0.3103
P(8hrs | NO) = 3/21 = 0.1429

P(18hrs | YES) = 6/29 = 0.2069
P(18hrs | NO) = 6/21 = 0.2857

COST
P(LOW | YES) = 14/29 = 0.4828
P(LOW | NO) = 11/21 = 0.5238

P(HIGH | YES) = 10/29 = 0.3448
P(HIGH | NO) = 8/21 = 0.3810

P(VERY HIGH | YES) = 5/29   = 0.1724
P(VERY HIGH | NO) = 2/21   = 0.0952

----------test data result----------

P(YES) = 0.58
P(NO) = 0.42

P(X0 | YES) = 0.1034 * 0.2069 * 0.1379 * 0.0345 * 0.1034 * 0.4828 = 5.083862540756031e-06
P(X0 | YES) * P(YES) = **2.9486402736384976e-06**
P(X0 | NO) = 0.0952 * 0.4762 * 0.2857 * 0.5238 * 0.4286 * 0.5238 = 0.001523678563007414
P(X0 | NO) * P(NO) = **0.0006399449964631139**

Predicition: NO

P(X1 | YES) = 0.0690 * 0.5517 * 0.1724 * 0.3793 * 0.3103 * 0.3448 = 0.0002662975616586492
P(X1 | YES) * P(YES) = **0.00015445258576201652**
P(X1 | NO) = 0.0952 * 0.0952 * 0.2381 * 0.1429 * 0.1429 * 0.3810 = 1.6789846424324115e-05
P(X1 | NO) * P(NO) = **7.051735498216128e-06**

Predicition: YES

P(X2 | YES) = 0.2414 * 0.2414 * 0.1724 * 0.3103 * 0.3793 * 0.4828 = 0.0005708753978057294
P(X2 | YES) * P(YES) = **0.000331107730727323**
P(X2 | NO) = 0.1905 * 0.4286 * 0.2381 * 0.1429 * 0.1429 * 0.5238 = 0.00020777434950101098
P(X2 | NO) * P(NO) = **8.726522679042461e-05**

Predicition: YES

P(X3 | YES) = 0.0690 * 0.2069 * 0.2414 * 0.2759 * 0.1034 * 0.3448 = 3.3892416938373536e-05

P(X3 | YES) * P(YES) = **1.965760182425665e-05**
P(X3 | NO) = 0.1429 * 0.4762 * 0.1429 * 0.1905 * 0.4286 * 0.3810 = 0.0003022172356378341
P(X3 | NO) * P(NO) = **0.0001269312389678903**

Predicition: NO

P(X4 | YES) = 0.1034 * 0.5517 * 0.1379 * 0.0345 * 0.2069 * 0.1724 = 9.683547696678155e-06
P(X4 | YES) * P(YES) = **5.6164576640733294e-06**
P(X4 | NO) = 0.0952 * 0.0952 * 0.2857 * 0.5238 * 0.2857 * 0.0952 = 3.693766213351306e-05
P(X4 | NO) * P(NO) = **1.5513818096075484e-05**

Predicition: NO

P(X5 | YES) = 0.1034 * 0.2069 * 0.2759 * 0.2759 * 0.3103 * 0.4828 = 0.0002440254019562895
P(X5 | YES) * P(YES) = **0.00014153473313464792**
P(X5 | NO) = 0.0952 * 0.4762 * 0.0952 * 0.1905 * 0.1429 * 0.5238 = 6.156277022252177e-05
P(X5 | NO) * P(NO) = **2.5856363493459143e-05**

Predicition: YES

P(X6 | YES) = 0.1379 * 0.2414 * 0.1724 * 0.3103 * 0.3793 * 0.1724 = 0.00011650518322565905
P(X6 | YES) * P(YES) = **6.757300627088225e-05**
P(X6 | NO) = 0.0476 * 0.4286 * 0.2381 * 0.1429 * 0.1429 * 0.0952 = 9.444288613682317e-06
P(X6 | NO) * P(NO) = **3.966601217746573e-06**

Predicition: YES

P(X7 | YES) = 0.1034 * 0.5517 * 0.2414 * 0.2759 * 0.3103 * 0.3448 = 0.0004067090032604825
P(X7 | YES) * P(YES) = **0.00023589122189107982**
P(X7 | NO) = 0.0952 * 0.0952 * 0.1429 * 0.1905 * 0.1429 * 0.3810 = 1.3431877139459292e-05
P(X7 | NO) * P(NO) = **5.641388398572902e-06**

Predicition: YES

P(X8 | YES) = 0.1034 * 0.2069 * 0.1724 * 0.3793 * 0.2069 * 0.1724 = 4.993079281099673e-05
P(X8 | YES) * P(YES) = **2.89598598303781e-05**
P(X8 | NO) = 0.0952 * 0.4762 * 0.2381 * 0.1429 * 0.2857 * 0.0952 = 4.1974616060810284e-05
P(X8 | NO) * P(NO) = **1.762933874554032e-05**

Predicition: YES

P(X9 | YES) = 0.0690 * 0.2414 * 0.2759 * 0.2759 * 0.3103 * 0.3448 =

0.00013556966775349417
P(X9 | YES) * P(YES) = **7.863040729702662e-05**
P(X9 | NO) = 0.1429 * 0.4286 * 0.0952 * 0.1905 * 0.1429 * 0.3810 = 6.044344712756683e-05
P(X9 | NO) * P(NO) = **2.5386247793578066e-05**

Predicition: YES

	Prediction	Answer
0	NO	NO
1	YES	YES
2	YES	YES
3	NO	NO
4	NO	YES

	Prediction	Answer
5	YES	YES
6	YES	NO
7	YES	YES
8	YES	YES
9	YES	NO

2. **(10 total)** After predicting the label for the testing data set, you will have both the predicted and the actual label for the data set. Use this information to calculate the Accuracy, Precision and Recall to evaluate the performance of the model that you just created.

```
Accuracy   = (TP + TN) / ALL
           = 7/10 = 0.7

Precision  = TP / (TP + FP)
           = 5 / (5 + 2) = 5 / 7 = 0.71428

Recall     = TP / (TP + FN)
           = 5 / (5 + 1) = 5 / 6 = 0.83333

F-1 Score  = (2 * precision * recall) / (precision + recall)
           = 0.76923
```

From calculating the F-1 score, we can conclude that the model performed well; however, it still needs improvement before being deployed into the real world.

Please indicate each step on how you predicted the label of each data point in the test data set as discussed in the slides.