**Assignment 3 – Misha Ward**

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

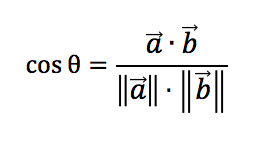
1. **Briefly outline the idea and major steps of decision tree classification.**

Decision tree classification is based on the idea of having your data flow through a tree in which each node represents a certain feature of the data to get more precise outcomes. Each link between nodes represents a choice that the data represents. Learning is accomplished by splitting the data into subsets of the values. For example, if your first feature is “isTrue”, then the node of the data would be “isTrue”. Splitting the values based on “Yes” or “No” would create new subtrees for new nodes of other features of the dataset. Continuing with the example above, values of “Yes” would create a right subtree and “No” left subtree and continue to do this for each feature of the dataset. The steps in this classification problem are as follows:

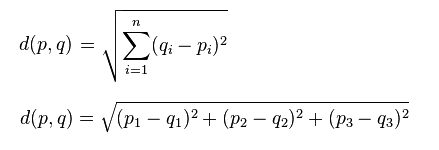
1. Assume features are categorical
2. Construct tree in top down recursive manner.
3. Start training the root of the tree.
4. New examples are selected broken up into segments recursively
5. Finally, data is selected on an impurity function.
6. **What is the basic idea of k-nearest neighbor? How to compute distance between two points?**

The basic idea of k-nearest neighbor is to classify data based on proximity to other data. As discussed in class, if the data looks like a duck, acts like a duck, and quacks like a duck, then it must be a duck. This underlying idea of k-nearest neighbor is implemented through saving all data and their distance from each other. Each data point is read and compared to other data points to determine the classification of the label attribute. There are multiple ways of finding the similarity of two data points including Euclidian and cosine similarity. By using Euclidian distance, you will be finding how close other data points are while cosine similarity would be determining how similar the points are. Of course, there are other measures of distance like Hamming and Kernel but these were not discussed in class. Prior to computing distances, one should consider to use preprocessing to scale the values into a common range. The formulas for both Euclidian and cosine similarity are below.

*Cosine Similarity*



*Euclidian Distance*



**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.

Step 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Row Labels | Count of BUY | Formula | Calculation | Result |
| NO | 21 | NO / ALL | 21 / 50 | 0.42 |
| YES | 29 | YES / ALL | 29 / 50 | 0.58 |
| **Grand Total** | **50** |  |  |  |

Step 2: Create tables below for each feature

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Count of BUY** | Column Labels |  |  |  |  | Results | |
| Row Labels | NO | YES | Formula | No Calculation | Yes Calculation | YES | NO |
| ACER | 2 | 3 | Label Count / Total Count | 2 / 21 | 3 / 29 | 10.3% | 9.5% |
| APPLE | 3 | 2 | Label Count / Total Count | 3 / 21 | 2 / 29 | 6.9% | 14.3% |
| ASUS | 2 | 3 | Label Count / Total Count | 2 / 21 | 3 / 29 | 10.3% | 9.5% |
| DELL | 2 | 3 | Label Count / Total Count | 2 / 21 | 3 / 29 | 10.3% | 9.5% |
| HP | 4 | 7 | Label Count / Total Count | 4 / 21 | 7 / 29 | 24.1% | 19.0% |
| LENEVO | 2 | 2 | Label Count / Total Count | 2 / 21 | 2 / 29 | 6.9% | 9.5% |
| SAMSUNG | 3 | 2 | Label Count / Total Count | 3 / 21 | 2 / 29 | 6.9% | 14.3% |
| SONY | 2 | 3 | Label Count / Total Count | 2 / 21 | 3 / 29 | 10.3% | 9.5% |
| TOSHIBA | 1 | 4 | Label Count / Total Count | 1 / 21 | 4 / 29 | 13.8% | 4.8% |

Step 3: Calculate probabilities for data:

Example Test Data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Manufacturer** | **RAM** | **CAPACITY** | **WARRANTY** | **BATTERY** | **COST** | **BUY** |
|  | DELL | 8GB | 500GB | 0 | 4hrs | LOW | NO |
| Yes | 0.103448276 | 0.206896552 | 0.137931034 | 0.034482759 | 0.103448276 | 0.482758621 | 0.0000029 |
| No | 0.095238095 | 0.476190476 | 0.285714286 | 0.523809524 | 0.428571429 | 0.523809524 | 0.0006399 |

Index all Yes weights and no weights for the data, then multiply all probabilities together and multiply this product by the total probability of sell or buy (respective to yes or No).

Calculation for YES: (.1034 \* .2068 \* .1379 \* .0344 \* .1034 \* .4827 ) \* .58 = . 0000029

Calculation for NO: (.0953 \* .4761 \* .2857 \* .5238 \* .4285 \* .5238 ) \* .42 = . 0006399

Step 4: Compare the probabilities and if the product probability for YES is larger than NO, then the prediction will be YES, otherwise NO. Using the example above, you would select “NO” as the guess.

Step 5: Results – compare results to create confusion matrix and analyze effectiveness of model.

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

1. **(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.

**Confusion Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted: Yes** | **Predicted: No** | **Total** |
| **Actual: Yes** | 5 | 1 | 6 |
| **Actual: No** | 2 | 2 | 4 |
| **Total** | **7** | **3** | **10** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Formula** | **Calculation** | **Final Answer** |
| Accuracy | (TP + TN) / All | (5 + 2) / 10 | **0.700** |
| Precision | TP / (TP + FP) | 5 / (5 + 2) | **0.714** |
| Recall | TP / (TP + FN) | 5 / (5 + 1) | **0.833** |

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