Homework 1

Name – Syed Faizan Ali (UFID –26592828)

Name – Charles Richardson (UFID -)

Q1)

1. Discrete, Qualitative, Ordinal, as the house numbers are not consecutive, hence we cannot assign meaning to differences between those numbers.
2. Continuous, Quantitative, Ratio
3. Discrete, Qualitative, Nominal
4. Discrete, Qualitative, Nominal (but Quantitative and Interval if numbers are ordered)
5. Continuous, Quantitative, Ratio
6. Continuous, Quantitative, Ratio
7. Discrete, Qualitative, Nominal
8. Discrete, Quantitative, Interval
9. Discrete, Qualitative, Nominal, as new numbers could be assigned randomly.
10. Discrete, Quantitative, Interval

Q2)

1. The binary attribute is asymmetric because there is preference on which outcome should be coded as zero or one. Both values 0 and 1 are not equally important.

1. Customer 1 = {1,0,1,0,0,0,0,0,0,0} Customer 2 = {1,1,1,0,0,0,0,0,0,0} Customer 3 = {0,0,0,0,0,1,0,0,0,0} Confusion Matrix

[C2 C1 0 1](#_Toc10313)

[0 7 1](#_Toc10314)

[1 0 2](#_Toc10315)

C3

C1 0 1

# 7 1

# 2 0

SMC(1,2) = (f\_11+ f\_00)/(f\_01+f\_10+f\_11+f\_00 ) = (2+7)/(1+0+2+7) = 9/10 = 0.9

SMC(1,3) = (f\_11+ f\_00)/(f\_01+f\_10+f\_11+f\_00 ) = (0+7)/(1+2+0+7) = 7/10 = 0.7

JC(1,2) = f\_11/(f\_01+f\_10+f\_11 ) = 2/(1+0+2) = 2/3 = 0.67

JC(1,3) = f\_11/(f\_01+f\_10+f\_11 ) = 0/(1+2+0) = 0/3 = 0

Jaccard Coefficient reflects the customer similarity better as customer similarity is the size of interaction divided by the union. IN this case the results are high in Jaccard Coefficient. When measuring similarity between customers based on their transactions, a common approach would be to use the Jaccard similarity coefficient, which measures the proportion of items that are purchased by both customers relative to the total number of items that are purchased by either customer. Since the data in the table consists of counts of items purchased by customers, rather than just binary indicators of whether an item was purchased or not, it may be more appropriate to use a different similarity measure, such as the cosine similarity or the Euclidean distance.

(3) In this case, I would prefer to use the cosine similarity measure because cosine similarity is a normalized measure that is unaffected by the scale of the data, making it a good choice for data where the magnitudes of the attribute values may vary widely. This chose this measure but not others because this measure is least affected by the scale, or the extra zeroes that convey little information.

Q3)

1. The similarity between p and q based on "Cosine" can be calculated as follows:

Cosine(p,q) = (p \* q) / (||p|| \* ||q||) = (13 + 0-3 + 13 + 0-3 + 13 + 0-3) /

(sqrt(1^2+0^2+1^2+0^2+1^2+0^2) \* sqrt(3^2+(-3)^2+3^2+(-3)^2+3^2+(-3)^2)) = 1/sqrt(2) = 0.707

The similarity between p and q based on "Pearson's Correlation" can be calculated as follows:

Correlation(p,q) = cov(p,q) / (std(p) \* std(q)) = (1/3 \* (3-0) + 1/3 \* (-3-0) + 1/3 \* (30)) / (sqrt(1/3 \* (1-0)^2 + 1/3 \* (0-0)^2 + 1/3 \* (1-0)^2) \* sqrt(1/3 \* (3-0)^2 + 1/3 \* (-3-0)^2 + 1/3 \* (3-0)^2)) = 1

1. The similarity between p and q' based on "Cosine" can be calculated as follows:

Cosine(p,q') = (p \* q') / (||p|| \* ||q'||) = (16 + 00 + 16 + 00 + 16 + 00) /

(sqrt(1^2+0^2+1^2+0^2+1^2+0^2) \* sqrt(6^2+0^2+6^2+0^2+6^2+0^2)) = 1

The similarity between p and q' based on "Pearson's Correlation" can be calculated as follows:

Correlation(p,q') = cov(p,q') / (std(p) \* std(q')) = (1/3 \* (6-0) + 1/3 \* (0-0) + 1/3 \*

(6-0)) / (sqrt(1/3 \* (1-0)^2 + 1/3 \* (0-0)^2 + 1/3 \* (1-0)^2) \* sqrt(1/3 \* (6-0)^2 + 1/3 \* (0-0)^2 + 1/3 \* (6-0)^2)) = 1

By comparing results in (1) and (2), we found that the cosine similarity has changed, but the Pearson’s coefficient has remained constant, as the relative codependence of the values between each of the datapoints in p and q has not changed any magnitude.

(3) The similarity between p and q'' based on "Cosine" can be calculated as follows:

Cosine(p,q'') = (p \* q'') / (||p|| \* ||q''||) = (19 + 0-9 + 19 + 0-9 + 19 + 0-9) /

(sqrt(1^2+0^2+1^2+0^2+1^2+0^2) \* sqrt(9^2+(-9)^2+9^2+(-9)^2 = = 1/sqrt(2) =

0.707

Correlation = 1

Comparing the results in (1) and (3), we can see that the cosine similarity and Pearson's correlation coefficient between p and q'' are the same as the cosine similarity and Pearson's correlation coefficient between p and q. This indicates that multiplying all attribute values in q by a constant has not changed the direction of the vectors and has only changed their magnitudes. The magnitude of a vector affects its distance from the origin, but it does not affect the direction of the vector or its similarity to other vectors in the space. Therefore, the similarity between p and q'' is the same as the similarity between p and q.

Q4)

E(S) = (summation from i = 1 to i = n) (-1 \* probability(i) \* log2( probability (i) ) )

1. Total No. of Attributes = 16

Class C0 = 8

P(C0) = 8/16 = ½

Class C1 = 8

P(C1) = 8/16 = ½

Entropy = - [1/2 log 1/2 + 1/2 log ½]

= - [1/2 x -1 + 1/2 x -1]

= - [-1]

= 1

1. I) Customer ID –

For Each of the 16 Customer ID’s, either C1 will be 0 and C0 will be 1 or C1 will be 1 and C0 will be 0

So,

Entropy = - [0/1 log 0/1 + 1/1 log 1/1] \* 16

= 0

II) Housing Type –

Apartment:

Total = 6

C0 = 6

C1 = 0

Entropy = - [6/6 log 6/6 + 0/6 log 0/6]

= 0

House:

Total = 6

C0 = 2

C1 = 4

Entropy = - [2/6 log 2/6 + 4/6 log 4/6]

= 0.9183

Hostel:

Total = 4

C0 = 0

C1 = 4

Entropy = - [0/4 log 0/4 + 4/4 log 4/4]

= 0

III) Gender Male:

Total = 8

C0 = 4

C1 = 4

Entropy = - [4/8 log 4/8 + 4/8 log 4/8]

= 1

Female:

Total = 8

C0 = 4

C1 = 4

Entropy = - [4/8 log 4/8 + 4/8 log 4/8]

= 1

IV) Marital Status Married: Total = 8

C0 = 4

C1 = 4

Entropy = - [4/8 log 4/8 + 4/8 log 4/8]

= 1

Single:

Total = 8

C0 = 4

C1 = 4

Entropy = - [4/8 log 4/8 + 4/8 log 4/8]

= 1

c) Information Gain –

Information Gain (ID) = E(s) – [P(ID) x E(ID)] x 16

= 1 – [1/16 x 0] x 16

= 1 – (0 x 16)

= 1

Information Gain (Housing Type) = E(s) – [P(Apartment) x E(Apartment) + P(Hostel) x E(Hostel) + P(House) x E(House)]

= 1 – [6/16 x 0 + 4/16 x 0 + 6/16 x 0.9183]

= 0.6556

Information Gain (Gender) = E(s) – [P(Male) x E(Male) + P(Female) x E(Female)]

= 1 – [8/16 x 1 + 8/16 x 1]

= 0

Information Gain (Martial Status) = E(s) – [P(Married) x E(Married) + P(Single) x E(Single)]

= 1 – [8/16 x 1 + 8/16 x 1]

= 0

Highest IG = Customer ID

Lowest IG = Marital Status and Gender

d). We choose the attribute with the highest information gain to split at the root node, hence we pick customer ID.

e). for tree 1, the weighted entropy of leaves is the same we have calculated above. Hence decrease = 1 – 0 = 1

for tree 2,

|  |  |  |
| --- | --- | --- |
| Path | C0 | C1 |
| Married Male | 4 | 0 |
| Married Female | 0 | 4 |
| Single Male | 0 | 4 |
| Single Female | 4 | 0 |

Plugging these values into the entropy formula, we get E = 0.

So, decrease = 1 – 0 = 1

For tree 3,

|  |  |  |
| --- | --- | --- |
| Path | C0 | C1 |
| Apartment Male | 3 | 0 |
| Apartment Female | 3 | 0 |
| House Male | 1 | 2 |
| House Female | 1 | 2 |
| Hostel Male | 0 | 2 |
| Hostel Female | 0 | 2 |

Plugging these values into the entropy formula, we get E = 1.8365

Decrease = 1 – 1.8365 = -0.8365

E 1). Tree 3 would be chosen for performing classification. No, initially we had chosen Customer ID to split the tree.

E 2). Entropy was used as the impurity measure. It was used to calculate the

Information Gain of an attribute split. Entropy is a measure of disorder and Information Gain is a measure of clarity that is brought by using a particular attribute as the split. The attribute with the highest Information Gain must always be chosen amongst the options to be used for the split.

Q5)

A. The expected confusion matrix summarizing the expected classifier performance on the two data sets is as follows:

|  |  |  |
| --- | --- | --- |
| DataSet1 | Predicted Negative (-) | Predicted Positive (+) |
| Actual Negative (-) | TN = 1000 x n | FP = 1000 - 1000n |
| Actual Positive (+) | FN = 1000 - 1000m | TP = 1000 x m |
|  |  |  |
| DataSet2 | Predicted Negative (-) | Predicted Positive (+) |
| Actual Negative (-) | TN = 1000 x n | FP = 1000 - 1000n |
| Actual Positive (+) | FN = 100 - 100m | TP = 100 x m |

B.

Accuracy for Dataset 1:

(TP + TN) / (TP + TN + FP + FN) = 1000n + 1000m / 2000

Precision = TP/ (TP + FP) = m / m+1-n

TPR = TP/ (TP + FN) = m

FPR = FP/ (FP + TN) = 1-n

Accuracy for Dataset 2:

(TP + TN) / (TP + TN + FP + FN) = 100m + 1000n / 1100

Precision = TP / (TP + FP) = m / m+10-10n

TPR = TP / (TP + FN) = m

FPR = FP / (FP + TN) = 1-n

C.

i). If the skew in the test data is 1:s, then the accuracy of the algorithm on this data set is:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

= no of pos is 1, no of neg is s

= (m \* 1) + (s\*n) / (1 + s)

s is large

accuracy = m + s.n / 1+s

= s\*n / s

= n

s is Small

accuracy = m + s.n / 1+s

= m + s.n

= m

ii). If s is very large (>>1), then the test data is heavily skewed towards the negative class. In this case, the accuracy of the algorithm will be close to n, since the classifier is likely to predict most instances as negative. If s is very small (<<1), then the test data is heavily skewed towards the positive class. In this case, the accuracy of the algorithm will be close to m, since the classifier is likely to predict most instances as positive.

D. In scenarios where class imbalance is high, precision and recall are better metrics than overall accuracy. Precision captures the fraction of the instances that are truly positive among all instances that are predicted as positive. It is a measure of how confident we are about the predicted positive instances. On the other hand, recall captures the fraction of the truly positive instances that are correctly predicted as positive. It is a measure of how well the classifier can identify positive instances. Overall accuracy can be misleading in such scenarios.

S = large

Accuracy = n

Precision = TP / TP + FP

= m / m + (1 - n)\*s

= m / (m + s - ns)

= m / m + s

= m / s

= 0

Recall = TP / TP + FN

= m / m + (1-m) \* 1

= m / m + 1 - m

= m

Precision and Recall together gives us all the parameter information while Accuracy does not give this information

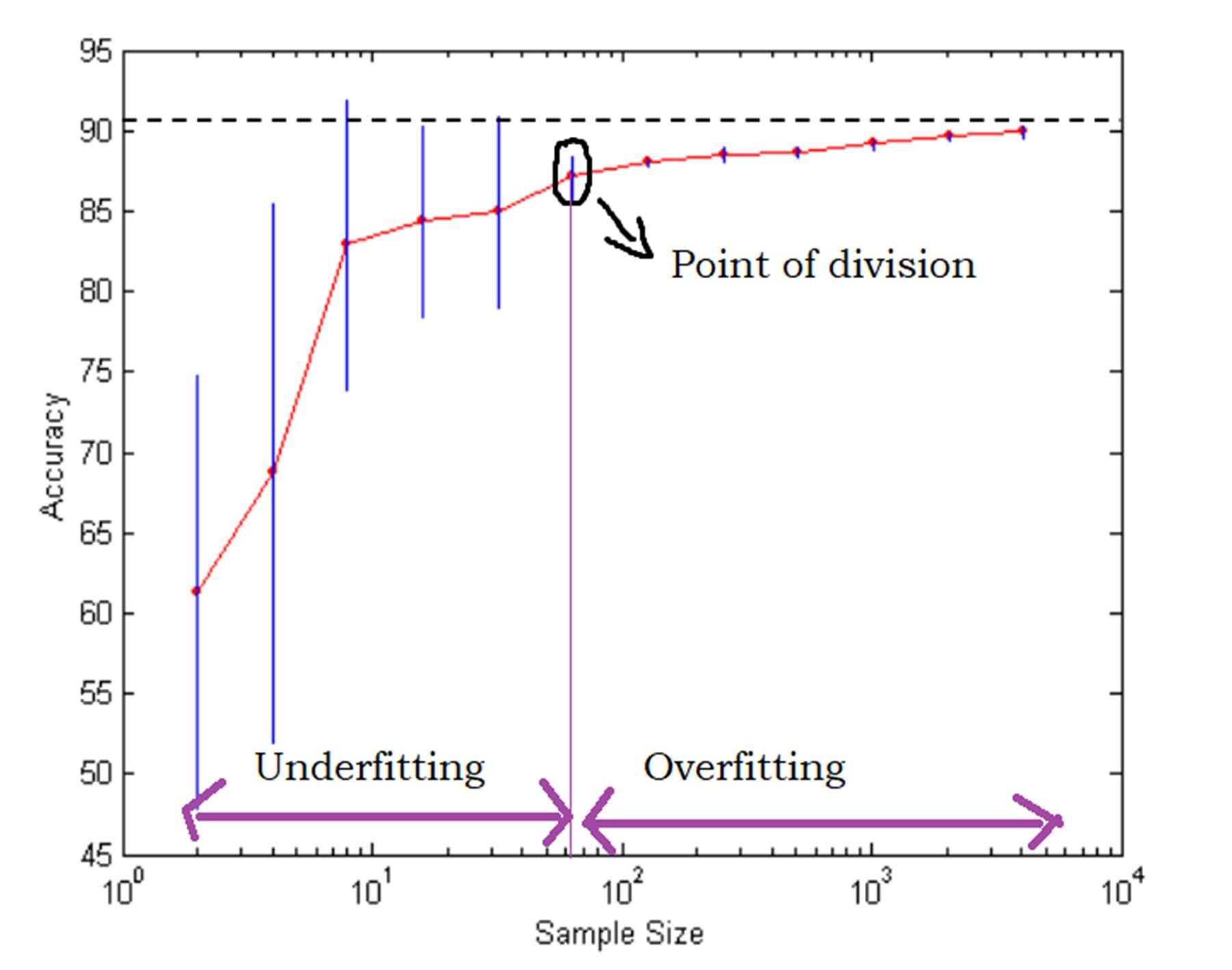
If precision is 0, it tells us that the dataset is imbalanced while recall does not give us that information.

Q6)

1. Decision Trees would be the best classifier for this scenario, as it is able to handle irrelevant attributes and automatically filter out the irrelevant ones while building the tree. In contrast, KNN and ANN may be affected by irrelevant attributes as they rely on distance metrics or neural network connections, respectively, that could be influenced by irrelevant attributes.

1. In this scenario, Decision Trees would still be a good choice, as it can select and combine attributes to create complex decision rules. However, Naïve Bayes would also be a good option, as it assumes independence among attributes and thus can capture interactions among features. KNN, on the other hand, may not perform as well since it relies on distance metrics and may not be able to capture the interactions among features as well as the other two classifiers.

Q7)



Two metrics were used to decide the point.

* + Accuracy: When a model is underfitting, its accuracy levels stay low, as there is much for it to learn. When the model starts to overfit, the relative increases in accuracy (as sample sizes increase) tend to decrease, as there is not much more information for the model to know about.

* + Standard Deviation: When a model is overfitting, the deviations between the accuracy levels in various trails tends to be low, as the model has leant about almost all the sample points.