Name: Sri Vamshi Polela

CSDA 5430 Predictive Analytics

Week 1 Assignment

1. Compare and contrast supervised and unsupervised learning in predictive data analytics. Use the case study of *hurricane and store readiness* presented in class to support your discussion. Make sure to identify the business goal and how predictive analytics can help business decision making.

**Supervised Learning VS Unsupervised Learning:**

* Supervised learning uses labeled data (input & output) to train the machine. It learns from past data or previous experience. Then it makes predictions or classifications.
* Algorithms come under supervised learning: Classification and prediction.
* Unsupervised learning uses unlabeled data to discover patterns. The algorithms here learn on itself and able to give inferences about the data.
* Algorithms: Clustering and Association.

**Business Goals:**

* The goal in the hurricane and store readiness using predictive analytics is to increase the sales or which product is in demand with the help of past data. So that they can heavily stock up on that product.
* Unsupervised learning looks for hidden patterns in unlabeled data, which makes it possible to find out about unexpected local product demand. This method helps with proactive stock management in advance of a hurricane's arrival, as well as providing insights that might not be immediately evident.

**Decision Making:**

* Predictive analytics identifies high-demand products during hurricanes, aiding in effective inventory management.
* It also forecasts the percentage increase in stock required for specific products, ensuring proactive preparation for future hurricanes. This approach optimizes resource allocation and enhances business resilience.

1. A data analytics algorithm has been applied to a transaction dataset and the following confusion matrix is received. According to the confusion matrix, how many total cases did the classifier make classification on? Out of those total cases, how many cases did the classifier predict as 1(fraudulent) and how many as 0 (non-fraudulent)? In reality, how many cases in the sample were fraudulent and how many were non-fraudulent? Assuming the important class is fraudulent, calculate accuracy, error rate, sensitivity, and specificity.

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | | |
|  | Actual Class | |
| Predicted Class | 1 (fraudulent) | 0 (non-fraudulent) |
| 1 (fraudulent) | 2 | 5 |
| 0 (non-fraudulent) | 8 | 35 |

**Solution**: From the above confusion matrix, To calculate the total number of cases, sum up all of the values in the confusion matrix.



* The total no. of cases = TP + FP + TN + FN

= 2 + 5 + 35 + 8

= 50

**Predictions:**

* The classifier predicts as 1(fraudulent) = TP + FP = 2+5 =7
* The classifier predicts as 0 (non-fraudulent) = FN + TN = 35+ 8 = 43

**Actuals:**

* The actual fraudulent cases: TP + FN = 2 + 8 = 10
* Actual non-fraudulent cases: FP + TN = 5 + 35 = 40

**Accuracy:**

* The ratio of correctly predicted cases both (True Positives and True Negatives) and total no. of cases.

= = = 0.74

**Error Rate:**

* It is calculated by ratio of total numbers of incorrect predictions (FN + FP) and total no. of cases. = = = 0.26

**Sensitivity (True Positive Rate or Recall):**

* It is calculated as the number of correct positive predictions (TP) divided by total number of positives.

= = = 0.2

**Specificity (True Negative Rate):**

* It is calculated as the number of correct negative predictions (TN) divided by total number of negatives.

= = = 0.875

1. Classification trees provide easily understandable classification rules, and each terminal node of a classification tree is equivalent to a classification rule in IF-THEN logical statement. Follow the classification rule examples on PPT Ch08 and provide the equivalent rule set for the following classification tree for the case study in chapter 8. The outcome variable of the model is *cd* (chronic disease). Remove redundancy if exists.

A diagram of a flowchart

Description automatically generated

**Rule set:**

Rule 1: IF CHARLSONSCORE < =0 THEN class = minor\_disease.

Rule 2: IF CHARLSONSCORE > 0 AND CHARLSONSCORE > 5 THEN class = minor\_disease.

Rule 3: IF CHARLSONSCORE > 0 AND CHARLSONSCORE <= 5 AND AGE > 47 THEN

class = severe\_disease.

Rule 4: IF CHARLSONSCORE > 0 AND CHARLSONSCORE <= 5 AND AGE <= 47 AND MSA\_Q\_08 > 1 THEN class = severe\_disease.

Rules 5: IF CHARLSONSCORE > 0 AND CHARLSONSCORE <= 5 AND AGE <= 47 AND MSA\_Q\_08 <=1 THEN class = minor\_disease.

From the ruleset, Rule 2 is a subset of Rule1 and Rule 4 is a subset of Rule 3. Therefore we can remove Rule 2 and Rule 4 without losing any information.

Rule 1: IF CHARLSONSCORE < =0 THEN class = minor\_disease.

Rule 3: IF CHARLSONSCORE > 0 AND CHARLSONSCORE <= 5 AND AGE > 47 THEN

class = severe\_disease.

Rules 5: IF CHARLSONSCORE > 0 AND CHARLSONSCORE <= 5 AND AGE <= 47 AND MSA\_Q\_08 <=1 THEN class = minor\_disease.

1. On the decision tree shown below from the case study in chapter 8, the root node 1 is split into two nodes 2 & 3 based on splitting the attribute CHARLSONSCORE >= 0.5. Calculate entropies for before split, after split, and the information gain (IG) of the split.

*A diagram of a number of patients

Description automatically generated*

**Solution: Entropies before split is nothing but Entropy for node 1 or root node:**

**A mathematical equation with numbers and symbols

Description automatically generated**

**Entropy Formula:**

**Entropy** = -p(k1)\*log2(p(k1))-p(k2)\*log2(p(k2))

Where p(k1) is the probability of class A and p(k2) is the probability of class B

The maximum probability of a node is 1, therefore p(k1)=1-p(k2)

**A close-up of a text

Description automatically generated**Here we two classes : minor\_disease and severe\_disease **Rcode for node 1:**

**Entropy for node 1:** (-0.47) log2 (0.47) – (0.53) log2 (0.53)

= -0.47\*(-1.0893) -0.53\*(-0.916)

= 0.511971+0.48548

= 0.997451 or 0.997

**In both the methods the result is like 0.997**

**A close-up of a number

Description automatically generatedEntropies after split is the entropy of node 2 and node 3:** **Rcode for node 2:**

**Entropy for node 2:** (-0.59) log2 (0.59) – (0.41) log2 (0.41)

= -0.59\*(-0.7612) -0.41\*(-1.2863)

= 0.449108+0.527383

= 0.976491 or 0.976

**Rcode for node 3:**

A close-up of a text

Description automatically generated**Entropy for node 3:** (-0.26) log2 (0.26) – (0.74) log2 (0.74)

= -0.26\*( -1.9434) -0.74\*(-0.4344)

= 0.505284+0.321456

= 0.82674 or 0.826

**Information Gain:** It is used to evaluate the effectiveness of split.

**Formula**: Entropy (before split) – Weighted Average entropy of split nodes (children’s nodes)

= Entropy (root node) – [(0.64) Entropy(node2) + (0.36) Entropy(node3)]

= 0.997 – [(0.64\*0.976) + (0.36\*0.826)]

= 0.997 – [0.625 + 0.298]

= 0.077

1. Universal Bank is a relatively young bank growing rapidly in terms of overall customer acquisition. The majority of these customers are liability customers (depositors) with varying sizes of relationship with the bank. The customer base of asset customers (borrowers) is quite small, and the bank is interested in expanding this base rapidly to bring in more loan business. In particular, it wants to explore ways of converting its liability customers to personal loan customers (while retaining them as depositors).

A campaign that the bank ran last year for liability customers showed a healthy conversion rate of over 9% success. This has encouraged the retail marketing department to devise smarter campaigns with better target marketing. The goal is to use *k*-NN to predict whether a new customer will accept a loan offer. This will serve as the basis for the design of a new campaign.

The file *UniversalBank.csv* contains data on 5000 customers. The data include customer demographic information (age, income, etc.), the customer’s relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign.

In R, perform the following steps as discussed in Ch08 on the data.

* Step 1: Collecting data (Discuss the business problem and how the data can support the analytics, identify the important class.)

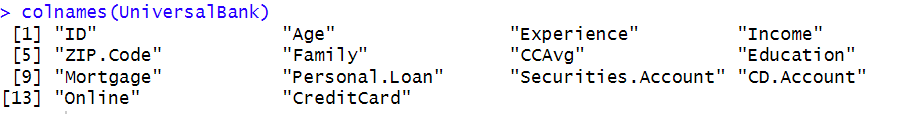
**Business problem:**

* Universal bank is trying to increase its asset customer base by converting liability customers into personal loan customers while maintaining them as depositors.

**Data Understanding:**

* The data improves analytics by allowing for demographic segmentation of customers and investigating the impact of existing connections on loan acceptance. Mortgage and securities accounts offer information about client behavior.
* The **important class** is personal loan which is target variable. By focusing on this class they can design effective campaigns.
* Step 2: Exploring data and preparing data (Remove zipcode, making sure categorical data are factored correctly. Create training partition (60%) and testing partition (40%) with randomized partitioning. Set a seed so the results can be reproduced. What is the percentage of customers who accepted the personal loan? How about this percentage in training and testing partitions, respectively?)

**Data Exploration:**

****

**Removing the ZIPcode:**

A close-up of a computer screen

Description automatically generated

* The percentage of customers who accepted the personal loan is 9.6%

After seeing the class distribution, we can say that dataset is imbalanced dataset.

A close-up of a computer screen

Description automatically generated

* **The percentage after Partitioning the Data for personal loan is same for train data and test data (9.6%)**

A screenshot of a computer code

Description automatically generated

* Step 3: Training a model on the data (Use the function C5.0() from the C50 package for the classification tree algorithm. Remember to ignore customer id as part of predictors. What is the tree size? Get a plot of the tree.)

**Sol:** **Tree size:** 11

**Tree plot:**

A diagram of a company

Description automatically generated with medium confidence

* Step 4: Evaluating model performance (Evaluate the model performance against the test data with confusion matrix. Identify accuracy, sensitivity, and specificity.)

**Predicting the model performance against the test data with confusion matrix with imbalanced dataset.**

* Accuracy = 98.65%
* Sensitivity = 99.78%
* Specificity = 88.02 %

A screenshot of a computer

Description automatically generated

* Step 5: Improving model performance with trail option (Try trials=6 and trials=10, compare with the original performance. Optionally, perform parameter tuning by identifying the best complexity parameter (cp) value and generating a pruned tree.)

**Try with Trials = 6: Try with trails =10**

Here the performance is slightly Here the model performance is increased by 0.1%

decreased by 0.6% .

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

**Parameter Tuning:**

A screenshot of a computer code

Description automatically generated

Best Cp is 0.10000 when the xerror is the least ie 0.20486

**Best Pruned tree:**

A diagram of a network

Description automatically generated