IS 665 Data Analytics

Fall 2023

Part 1. Statistics Basics

1.

a.

The table below shows the information for ICU and non-ICU patients, with and without fever, and the total counts.

	ICU	**Non-ICU**	**Total**
Fever	36	100	136
No Fever	0	2	2
Total	36	102	138

b.

Given that a certain patient has a fever, the likelihood that they will be in the ICU can be calculated by dividing the number of fever patients in the ICU by the total number of fever patients.

P(ICU|Fever)
= Total number of fever patients / Number of fever patients in ICU
= 136 / 36
~ 0.265

So, the probability is approximately 26.5%.

c.

Given that a patient is in the ICU, the likelihood that he/she has a fever is calculated by dividing the number of ICU patients with fever by the total number of ICU patients.

P(Fever|ICU)
= Total number of ICU patients / Number of ICU patients with fever
= 36 / 36
= 1

This means that the probability is 100%, or in other words, every patient in the ICU has a fever according to this data.

………

2.

From the table and the newspaper story, we can create equations for the total number of fish caught by contestants in various groups:

The winner caught 15 fish.
Contestants who caught 3 or more fish averaged 6 fish each.
Contestants who caught 12 or fewer fish averaged 5 fish each.

Here's how to set up the calculations:

Let Xi be the number of contestants who caught i fish for i=4 to 12, since these are the unknowns.

The total number of fish caught by those who caught 3 or more fish can be represented as:

$3\times23+4\times X4 +5\times X5 +\cdots+12\times X12 +13\times5+14\times2+15\times1$

The total number of fish caught by those who caught 12 or fewer fish can be represented as:

$0\times9+1\times5+2\times7+3\times23+4\times X4 +5\times X5 +\cdots+12\times X12$

Given the averages, we also have:

$(3\times23+4\times X4 +5\times X5 +\cdots+12\times X12 +13\times5+14\times2+15\times1) / (23+X4 +X5 +\cdots+X12 +5+2+1) = 6$

$(0\times9+1\times5+2\times7+3\times23+4\times X4 +5\times X5 +\cdots+12\times X12) / (9+5+7+23+X4 +X5 +\cdots+X12) = 5$

Since we have two equations with many unknowns, we cannot find a unique solution without additional information. If we had the total number of contestants or the total number for one of the missing values of n, we could solve the system of equations.

In the absence of this information, we cannot calculate the exact total number of fish caught during the festival. If you can provide the missing information, we can proceed with the calculations.

……………………………………………………………………………………………………….

3.

(a)

A confusion matrix is a table used to describe the performance of a classification model. It has four parts: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN).

- TP is when the actual is 1 and the flag is also 1, i.e. model correctly predicts a churn.
- TN is when the actual is 0 and the flag is 0, so the model correctly predicts no churn.
- FP is when the actual is 0 but the flag is 1, a false alarm.

- FN is when the actual is 1 but the flag is 0, a miss.

P(actual=1) is basically the sum of TP and FN divided by the total number of observations because it's the probability that the customer actually churns regardless of what the model predicts.

P(flag=1) is the sum of TP and FP divided by the total number of observations because it's the probability that the classifier flags a churn whether it's right or wrong.

(b)

Now, 'a' is defined as P(actual=1|flag=1), which means the probability that the actual churn happens given that the model has flagged a churn. So, that's TP divided by (TP + FP), because out of all the predicted churns, we want to know how many were actually correct.

'b' is P(actual=0|flag=0), so the probability that there's no churn given the model didn't flag it. That's TN divided by (TN + FN), which is the proportion of true non-churns in all the non-flagged cases.

(c)

We know that, churn base rate is 5%, which means that only 5% of the customers are expected to churn.

If 'a' is 0.8, it means the model catches 80% of all actual churns.
If 'b' is 0.9, it means that the model correctly identifies 90% of the non-churns.

If the churn rate is only 5%, and we're catching 80% of those with a model, the false positive rate is the percentage of non-churners that we incorrectly identify as churners, which would be 10% given that 'b' is 0.9.

In a real-world scenario, because the churn rate is so low, even a small false positive rate can mean that many non-churning customers are incorrectly flagged as churning, which would make the precision (the proportion of true positives in all positive predictions) much lower than the claimed 80%.

Proof with Example:

For instance, let's assume there are 1000 customers. With a 5% churn rate, that's 50 actual churns and 950 non-churns. If 'a' is 0.8, we're correctly identifying 40 of those churns. If 'b' is 0.9, we're correctly identifying 855 of the non-churns.

We have 40 true positives (TP), meaning 40 customers were correctly identified as churners. There are 95 false positives (FP), where non-churners were incorrectly flagged as churners.

We have 855 true negatives (TN), so 855 non-churners were correctly not flagged.
And there are 10 false negatives (FN), where churners were missed by the model.

Now, the precision of this model, which is the number of correct positive predictions (TP) out of all positive predictions (TP + FP), is roughly 29.63%. This is way lower than the 80% Dr. Lin claimed.

The overall accuracy, the number of correct predictions (TP + TN) out of all customers, is 89.5%. This might seem high but given the low churn rate and the high false positive rate, the precision is not as good as Dr. Lin suggested.

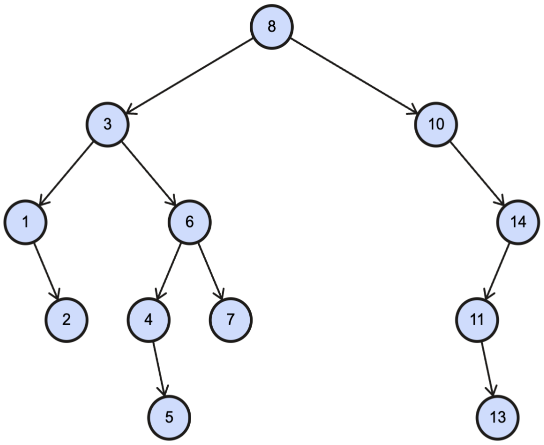
So, Dr. Lin's claim about the classifier's performance is not realistic. A model with an 80% hit rate on a 5% churn rate would not have a precision close to 80% because of the imbalance in the number of actual churns versus non-churns. In simple terms, the model would be wrong a lot of the time it predicts churn because there are so few churns to begin with

……….

Part 2. Data Structure Basics

1.
The order of visit using Depth First Search (DFS) starting at node A is: A, B, C, E, D, F, H, G.
The order of visit using Breadth First Search (BFS) starting at node A is: A, B, D, E, C, F, G, H.

2.



………

Part 3. Data Warehousing

a.

Measures:

- Access Frequency: This is a count measure representing how often agents access web pages. It is a derived measure that can be calculated by counting the number of records in the fact table for each combination of dimensions.

Constraints:

The constraints for the business question are attributes that will be used to filter or group the data in the reports. They are:

- Agent Position: This is the role or job title of the agent which is stored in the attribute Agent.Position. It will be used to report the access frequency by different positions.
- Web Page Type: This refers to the category of the webpage (e.g., support page, product description) and is stored in the attribute WebPage.PageType. It is used to filter the access frequency by the type of web page.
- Customer State: The state from which the customer is calling, which is stored in Customer.State. It will be used to filter the data to include only those customers from a specific state.
- Time (Month): The month when the webpage was accessed, which can be derived from a Time dimension table with attributes for Day, Month, and Year. The attribute Time.Month will be used to group the data by each month.

These constraints will be used in SQL queries or in business intelligence tools to slice and dice the data, allowing for detailed analysis as per the business question's requirements. In a star schema, these would typically correspond to fields in the dimension tables that are joined to the fact table where the measure is recorded.

b.

Based on the metadata and the business question provided, we can design a star schema for the database. The star schema is optimized for query performance and ease of understanding and consists of one or more fact tables referencing any number of dimension tables. For the given question, we need to report on the frequency of agents in different positions accessing web pages of each page type to perform customer support tasks for customers from each state in each month.

Here is a proposed star schema:

Fact Table:

FactAgentWebPageAccess

- Attributes: Agent_ID (link to Agent dimension), WebPage_ID (link to WebPage dimension), Time_ID (link to Time dimension), Access_Frequency (measure)
- Measures: Access_Frequency (this is a count of the number of times an agent accesses a webpage)
- Constraints: Agent.Position, WebPage.PageType, Customer.State, Time.Month

Dimension Tables:

Agent Dimension (DimAgent):
Attributes: Agent_ID (primary key), AName, Position, Experience_Level, Branch, Manager_ID
This will store the details of the agents including their position which is required for the constraint.

WebPage Dimension (DimWebPage):
Attributes: WebPage_ID (primary key), PageURL, PageType, Description
This will store the details of web pages accessed by agents, including the type of page which is required for the constraint.

Customer Dimension (DimCustomer):
Attributes: Customer_ID (primary key), Name, Gender, State, Country, Income_Level, Education_Level
This will hold customer information including the state which is a constraint in the business question.

Time Dimension (DimTime):
Attributes: Time_ID (primary key), Day, Month, Year
This is a standard time dimension which will allow us to filter by month as required in the business question.

Task Dimension (DimTask):
Attributes: Task_ID (primary key), Begin_Time, End_Time, TC_ID (link to Task Category dimension)
This could be used to track the tasks during which web pages are accessed if required.

Task Category Dimension (DimTaskCategory):
Attributes: TC_ID (primary key), TC_Name

This gives us additional information on the category of tasks being performed.

The fact table will record each instance of an agent accessing a webpage, which can be counted and grouped by the various dimensions to answer the business question. The schema is designed to be simple for quick querying and easy to understand for reporting purposes. Each dimension table is connected to the fact table through a foreign key that is a primary key in the respective dimension table. The attributes included in each table are chosen based on their relevance to the business question and the need for meaningful queries.
……

Part 4. Data Mining

NOTE: Please refer python code submitted in the jupyter notebook

…………………………………………………………………………………………………

APPLICATIONS

Case 1:

NOTE: Please refer python code submitted in the notebook

Association Rule Mining Analysis

Methodology and Parameters:
Data Source: Emergency events in scenic areas.
Approach: Association Rule Mining using the Apriori algorithm.
Support Threshold: 0.013333 (1.33%).
Confidence Threshold: 1.0 (100%).

Top 5 Rules Discovered:
Rule 1:
Antecedents: (Unnamed: 13, Unnamed: 10)
Consequents: (Unnamed: 12)
Support: 0.013333, Confidence: 1.0, Lift: 75.0
Rule 2:
Antecedents: (Unnamed: 12)
Consequents: (Unnamed: 13, Unnamed: 10)
Support: 0.013333, Confidence: 1.0, Lift: 75.0
Rule 3:
Antecedents: (Unnamed: 18, Unnamed: 13)
Consequents: (Unnamed: 12)
Support: 0.013333, Confidence: 1.0, Lift: 75.0
Rule 4:
Antecedents: (Unnamed: 12)
Consequents: (Unnamed: 18, Unnamed: 13)
Support: 0.013333, Confidence: 1.0, Lift: 75.0
Rule 5:
Antecedents: (Unnamed: 18, Unnamed: 23)
Consequents: (Unnamed: 12)
Support: 0.013333, Confidence: 1.0, Lift: 75.0

Interpretation and Recommendations:

- The identified rules demonstrate a perfect correlation (100% confidence) between specific combinations of factors (represented by 'Unnamed' columns) in the context of emergency events in scenic areas.

- The high lift value of 75.0 across these rules suggests a significantly stronger association between the antecedents and consequents compared to random chance. This indicates that certain combinations of factors are highly predictive of specific outcomes or responses.
- Strategic Implications:
- It is recommended to closely investigate and understand the specific factors (currently unnamed) represented by these columns.

Upon identification, these factors can be used to develop targeted strategies for emergency response in scenic areas. For example, if a combination of factors consistently predicts a particular type of emergency event, resources and response mechanisms can be aligned accordingly for more efficient management.

Cautions:

- While the rules indicate strong statistical associations, they do not imply causality.
- Further investigation is necessary to understand the underlying reasons for these associations and to validate the practical applicability of these findings.

Conclusion:
The Association Rule Mining analysis reveals critical insights into the relationships between various factors involved in emergency events in scenic areas. The high confidence and lift values of the top rules suggest specific combinations of factors that are strongly predictive of certain outcomes. These findings can guide the development of more effective strategies for emergency preparedness and response, provided that the specific nature of the factors represented by the 'Unnamed' columns is thoroughly investigated and understood.

……………………………………………………………………………………………………

Case 2:

NOTE: Please refer python code submitted in the notebook

a.
The primary research question is: "Can we predict whether a billionaire is self-made based on factors such as net worth, country, source of wealth, rank, age, residence, citizenship, marital status, number of children, and education?" This problem is significant as it explores the characteristics that might influence a person's journey to becoming a self-made billionaire, providing insights into patterns of wealth accumulation and socioeconomic factors.

b.
To address this question, the following attributes were utilized:

Attributes Used:

- NetWorth: Total net worth in billions.
- Country: Country of citizenship.

- Source: Primary source of wealth.
- Rank: Forbes billionaire rank.
- Age: Age of the individual.
- Residence: Primary place of residence.
- Citizenship: Country of citizenship.
- Status: Marital status.
- Children: Number of children.
- Education: Highest level of education attained.

Data Mining Algorithm:

- Algorithm: Random Forest Classifier.
- Reason for Selection: This algorithm is effective for classification tasks and can handle a mix of numerical and categorical data. It's also robust to overfitting and can capture complex nonlinear relationships.

Prediction and Tuning:
Label: Self_made (Binary: 0 for not self-made, 1 for self-made).
Parameters Tuned:

1. n_estimators: Number of trees in the forest.
2. max_depth: Maximum depth of the tree.
3. min_samples_split: Minimum number of samples required to split an internal node.
4. min_samples_leaf: Minimum number of samples required to be at a leaf node.

c.
Discuss Findings
The hyperparameter tuning improved the accuracy of the Random Forest model to approximately 72.36%, a slight increase from the initial accuracy of 71.36%. The best parameters identified were: max_depth: None, min_samples_leaf: 4, min_samples_split: 10, and n_estimators: 150.

Key observations from the classification report are:

- The model's precision for predicting self-made billionaires (class 1) is 74%, and the recall is 91%. This suggests the model is quite effective at correctly identifying self-made billionaires but is less effective at correctly identifying individuals who are not self-made (class 0).
- The precision for class 0 is 61%, with a recall of 31%. This lower recall rate indicates the model often misses classifying non-self-made billionaires accurately.
- The overall accuracy of 72.36% indicates the model's reasonable effectiveness in distinguishing between self-made and non-self-made billionaires, given the complexity of socioeconomic factors influencing this status.

These findings suggest that while the model has decent predictive power, there is room for improvement, particularly in correctly identifying non-self-made billionaires. Future work might explore additional features, different classification algorithms, or more sophisticated ensemble methods to enhance predictive performance.