**Q1. Association rules with Apriori.**

**The file ./specs/gpa question1.csv contains data scholar data related to a**

**selected sample of students. There might be interesting rules that can be extracted from this file.**

**1. Filter out the count attribute as this will not be included in the rule**

**generation.**

**2. Use the Apriori algorithm to generate frequent itemsets from the input**

**data. When doing so, only select frequent itemsets with a support of at**

**least 15% (so, the minimum support should be 0.15). How many frequent**

**itemsets are produced? How big are they? Include this information in**

**your report.**

**3. Save the generated itemsets in ./output/question1 out apriori.csv,**

**making sure to include the support column.**

**4. Using these frequent itemsets, generate a first batch of association rules**

**with a minimum confidence of 0.9. How many rules are produced? For**

**each rule, include a short description in your report.**

**5. Save the generated rules in ./output/question1 out rules9.csv, making sure to include the support and confidence columns.**

**6. Generate a second batch of association rules, but this time use a minimum**

**confidence of 0.7. How many rules are produced this time? Again, shortly**

**describe the outcome in your report.**

**7. Save the generated rules in ./output/question1 out rules7.csv in the**

**same format as the previous rule batch.**

Read the data from file using pandas and stored it into a DataFrame.

1. To remove the column ***count*** we can drop it using ***pandas.drop(‘count’,axis=1).***
2. **Support**: Refers to the default popularity of an item and can be calculated by finding number of transactions containing a particular item divided by total number of transactions.

-To generate the frequent itemsets using apriori, we first need to convert the categorical data to numerical using OnehotEncoding. For that I have used ***get\_dummies*** functionality inside pandas to convert categorical data to numerical data.

- Once that done we can apply apriori algorithm to the newly created data set. - Also we can specify the ***min\_support=0.15*** as we want the support of 15%.

- Total 19 frequent itemsets are created, maximum length of itemsets is 2.

1. Saved the data to file ***question1\_out\_apriori.csv.***
2. To generate the association rule using the above created data we have to first import the package called ***association\_rules*** in ***mlxtend.frequent\_patterns*** .

-**Confidence:** refers to the likelihood that an item B is also bought if item A is bought. It can be calculated by finding the number of transactions where A and B are bought together, divided by total number of transactions where A is bought.

- Association Rule: Function to generate association rules from frequent itemsets

- Following is the syntax to apply the rule with 90% confidence ***association\_rules(freq\_itemsets, metric="confidence", min\_threshold=0.9).***

*-* Only one rule is created after passing the data from association\_rules.

- Confidence refers to the likelihood that an item B is also bought if item A is bought. So It can be seen that **(category\_21...25)** &**(category\_junior),** have a confidence of **1.0** indicating they would be occuring together.

1. Saved the data to file ***question1\_out\_rules9.csv.***
2. To generate a second batch with confidence of 70% we can use the following syntax ***association\_rules(freq\_itemsets, metric="confidence", min\_threshold=0.7).***

***-*** 3 rules are produces.

1. (category\_philosophy)->(category\_26...30). This signifies that when a person is studying philosophy, the age of the person is most likely to be in range of (26-30). With support of 20% and confidence 71%.
2. (category\_Ph.D)->(category\_26...30). This signifies that when a person is studying Ph.D, the age of the person is most likely to be in range of (26-30). With support of 16% and confidence 80%.
3. (category\_21...25)->(category\_junior). This signifies that when a person in range of (26-30), The person is most likely to be a junior student. With support of 16% and confidence 100%.
4. Saved the data to file ***question1\_out\_rules7.csv.***

**Q2. Association rules with FP-Growth.**

**Question 2: Association rules with FP-Growth**

**The file ./specs/bank data question2.csv contains customer records from**

**the marketing department of a financial firm. The data contains the following**

**fields:**

**id : a unique identification number**

**age : age of customer in years (numeric)**

**sex : MALE / FEMALE**

**region : inner city / rural / suburban / town**

**income : income of customer (numeric)**

**married : if the customer is married - YES / NO**

**children : number of children (numeric)**

**car : if the customer owns a car - YES / NO**

**save acct : if the customer has a saving account - YES / NO**

**current acct : if the customer has a current account - YESY / NO**

**mortgage : if the customer has a mortgage - YES / NO**

**pep : if the customer signed for a Personal Equity Plan after the last mailing -**

**YES / NO**

**1. Filter out the id attribute as this will not be include in the rule generation.**

**2. Discretize the numeric attributes into 3 bins of equal width, the filter out**

**the original attributes. When doing so, only select frequent itemsets with**

**a support of at least 20% (so, the minimum support should be 0.2).**

**3. Use the FP-Growth algorithm to generate frequent itemsets from the**

**data. How many frequent itemsets are produced? How big are they?**

**Include this information in your report.**

**4. Save the generated itemsets in ./output/question1 out fpgrowth.csv**

**5. Using the obtained frequent itemsets, generate association rules. Experiment with different confidence values, selecting a value that produces at**

**least 10 rules. What is this value? Include it in your report.**

**6. Save the generated rules in ./output/question2 out rules.csv**

**7. Select the top 2 most interesting rules and for each specify the following**

**in your report:**

**• an explanation of the pattern and why you believe it is interesting**

**based on the business objectives of the company;**

**• any recommendations based on the discovered rule that might help**

**the company to better understand behavior of its customers or in its**

**marketing campaign.**

**Note: The top 2 most interesting rules may not be the top 2 rules in**

**the result set. They are rules that provide some non-trivial, actionable**

**knowledge based on the underlying business objectives.**

Read the data from file using pandas and stored it into a DataFrame.

1. To filter the id attribute we can drop the column from the dataset using ***pd.drop(col\_name,axis=1)***.
2. We have to Discretize the numerical data into 3 bins. To do that we can use ***pandas.cut(col\_name, bins=3).*** Once the columns are binned we have to categorize the data using ***get\_dummies*** functionality inside pandas.

- We have to import the ***fpgrowth*** from ***mlxtend.frequent\_patterns*** and pass the categorized data from it. To have a support of 20% we have to specify ***min\_support=20***.

1. After passing the data from fpgrowth function we have a large dataset of around **231** rows and 2 columns(support, itemsets).
2. Saved the output from fpgrowth to file ***question2\_out\_fpgrowth.csv***.
3. From the obtained frequent itemsets we have created the rules by passing it from the ***association\_rules()*** function. As the requirement was to get at least 10 rules by changing the confidence values.

- It is observed from the output that when we specify the confidence below **0.79** i,e below 79% then we get more than 10 records.

1. Saved the rules to file ***question2\_out\_rules.csv***.
2. Interesting rules:

* {'save\_act\_YES', 'sex\_FEMALE'}-> {'current\_act\_YES'} 25: Looking at this rule it can be seen that females who have savings account have 77% confidence to have a current account. This infers that the females are more likely to have daily transactions giving opportunity to banks to target them for loans.
* {'mortgage\_NO', 'pep\_NO'})->{'current\_act\_YES'}: Here we can see that lift is greater than 1. If lift is > 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.

So if a person doesn’t have a mortgage and pep the he would have a current account which could help companies to target them to have a loan for starting business.

These rules could help companies to identify the age group to target for offer in equity plans and loans provision to potential customers.