GENERAL SIR JOHN KOTELAWALA DEFENCE UNIVERSITY
INTAKE 41

Faculty of Management, Social Sciences, and Humanities
Department of Languages

BSc in Applied Data Science Communication

Fundamentals of Data Mining - LB 2114
Assignment 02 – Association Rule Mining & Logistic Regression
Year 02 – Semester 01
04.05.2025

Group Members:

- P.R.N.D. Jayalath - D/ADC/24/0002
- W.P.C.N.S Perera - D/ADC/24/0046
- JDT Jayawickrama - C/ADC/24/6854

Table of Contents

Task 01: Association Rule Mining with Student Dataset

Association Rule Mining with Student Dataset.

01.Introduction

In the current Landscape of digital education, data mining offers an exciting opportunity to analyse vast amounts of educational data and extract actionable insights. Educational institutions increasingly rely on data-driven decision to predict student performance, identify at risk learners and enhance academic outcomes. This research aims to explore the power of data mining in predicting student academic success based on demographic, behavioural, and academic attributes.

The primary research question is:

"How accurately can we predict student academic performance using data mining techniques such as logistics regression and association rule mining?"

This study leverages a real-world dataset of student attributes to build predictive models and extract meaningful patterns using python. By applying the CRISP-DM framework, this project identifies the most influential factors affecting student performance and provides data – driven recommendations for educators and policymakers.

02.Dataset

The dataset used in this project is the Student Performance Data Set sourced from the https://github.com/Emmanuel96/apriori_association_rule_mining/tree/master/Dataset/

This dataset includes records from two Portuguese secondary schools and consist of student grades and background attributes such as family information, study habits, and school support.

Dataset Overview:

- Number of Records: 649
- Number of Features: 33
- Target Variable: Final Grade (G3)

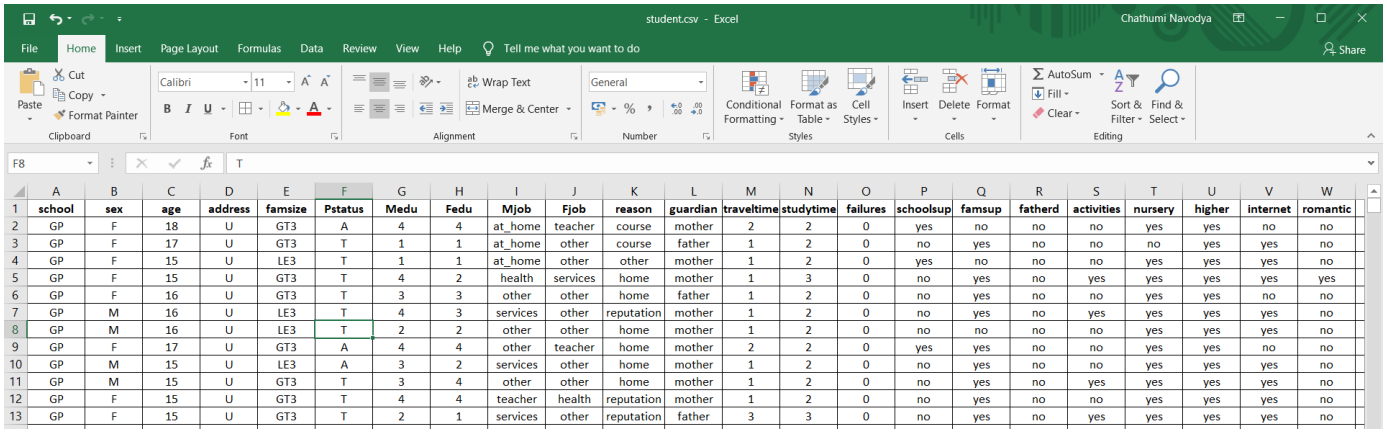
The dataset is ideal for educational data mining due to its structured format, completeness and inclusion of diverse student -related features that influence academic outcomes.

03.Explanation and Preparation of the Data Set

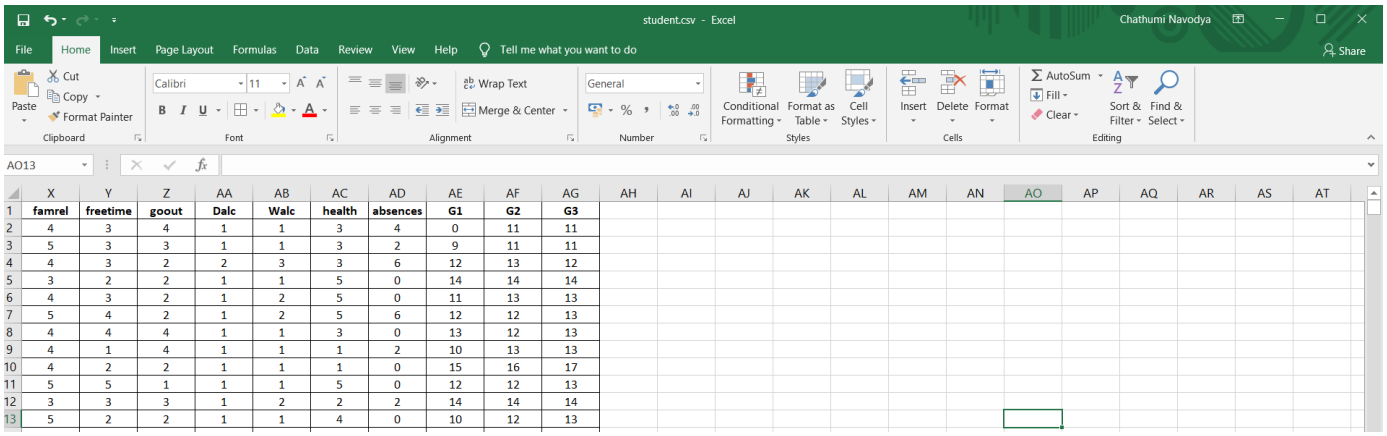
a) Explanation of the dataset.

Student data set has been used for the association rule mining task.
There are 33 columns and 1046 rows in the data set.
Attributes of the data set are,

1. School - The school the student attends
2. Sex - Gender of the student (Male or Female)
3. Age - Age of the student
4. Address - Type of address of the student (urban or rural)
5. Famsize - Family size (small or large)
6. Pstatus - Parent's cohabitation status ('T' - living together, 'A' - living apart)
7. Medu - Mother's education level (1 - none, 2 - primary education (4th grade), 3 - 5th to 9th grade, 4 - secondary or higher education)
8. Fedu - Father's education level (same scale as Medu)
9. Mjob - Mother's job
10. Fjob - Father's job 5
11. Reason - Reason for choosing the current school
12. Guardian - Student's guardian
13. Traveltime - Home to school travel time (1 - 1 hour)
14. Studytime - Weekly study time (1 - 10 hours)
15. Failures - Number of past class failures
16. Schoolsup - Whether the student receives educational support from the school (yes or no)
17. Famsup - Whether the student receives educational support from the family (yes or no)
18. Fatherd - Father's educational support level (1 - low, 2 - medium, or 3 - high)
19. Activities - Extra-curricular activities participation (yes or no)
20. Nursery - Whether the student attended nursery school (yes or no)
21. Higher - Desire to pursue higher education (yes or no)
22. Internet - Internet access at home (yes or no)
23. Romantic - In a romantic relationship (yes or no)
24. Famrel - Quality of family relationships (from 1 - very bad to 5 - excellent)
25. Freetime - Free time after school (from 1 - very low to 5 - very high)
26. Goout - Going out with friends' frequency (from 1 - very low to 5 - very high)
27. Dalc - Workday alcohol consumption (from 1 - very low to 5 - very high)
28. Walc - Weekend alcohol consumption (from 1 - very low to 5 - very high)
29. Health - Current health status (from 1 - very bad to 5 - very good)
30. Absences - Number of school absences
31. G1 - First period grade (from 0 to 20)
32. G2 - Second period grade (from 0 to 20)
33. G3 - Final grade (from 0 to 20)



	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup	famsup	fatherd	activities	nursery	higher	internet	romantic
2	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2	2	0	yes	no	no	no	yes	yes	no	no
3	GP	F	17	U	GT3	T	1	1	at_home	other	course	father	1	2	0	no	yes	no	no	no	yes	yes	no
4	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother	1	2	0	yes	no	no	no	yes	yes	yes	no
5	GP	F	15	U	GT3	T	4	2	health	services	home	mother	1	3	0	no	yes	no	yes	yes	yes	yes	yes
6	GP	F	16	U	GT3	T	3	3	other	other	home	father	1	2	0	no	yes	no	no	yes	yes	no	no
7	GP	M	16	U	LE3	T	4	3	services	other	reputation	mother	1	2	0	no	yes	no	yes	yes	yes	yes	no
8	GP	M	16	U	LE3	T	2	2	other	other	home	mother	1	2	0	no	no	no	no	yes	yes	yes	no
9	GP	F	17	U	GT3	A	4	4	other	teacher	home	mother	2	2	0	yes	yes	no	no	yes	yes	no	no
10	GP	M	15	U	LE3	A	3	2	services	other	home	mother	1	2	0	no	yes	no	no	yes	yes	yes	no
11	GP	M	15	U	GT3	T	3	4	other	other	home	mother	1	2	0	no	yes	no	yes	yes	yes	yes	no
12	GP	F	15	U	GT3	T	4	4	teacher	health	reputation	mother	1	2	0	no	yes	no	no	yes	yes	yes	no
13	GP	F	15	U	GT3	T	2	1	services	other	reputation	father	3	3	0	no	yes	no	yes	yes	yes	yes	no



	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
2	4	3	4	1	1	3	4	0	11	11
3	5	3	3	1	1	3	2	9	11	11
4	4	3	2	2	3	3	6	12	13	12
5	3	2	2	1	1	5	0	14	14	14
6	4	3	2	1	2	5	0	11	13	13
7	5	4	2	1	2	5	6	12	12	13
8	4	4	4	1	1	3	0	13	12	13
9	4	1	4	1	1	1	2	10	13	13
10	4	2	2	1	1	1	0	15	16	17
11	5	5	1	1	1	5	0	12	12	13
12	3	3	3	1	2	2	2	14	14	14
13	5	2	2	1	1	4	0	10	12	13

b) Preparation of the Dataset

As the dataset is completely suitable for do association rule mining and has no NULL values in the dataset, we didn't have much work to do to prepare the dataset. Therefore, first we read and understood the dataset and applied the association rule mining into the dataset using PYTHON.

Missing Values: No Missing data detected in the dataset.
Encoding: Categorical variables (Like schoolsup, sex, address) were ancoded using LabelEncoder and OneHotEncoder.
Feature Scaling: Continuous numerical features were scaled using StandardScaler for better logistic performance.
Train -Test Spilt: 70% training and 30% testing train_test_split from sklearn.

04.Association Rule Mining

Association rule mining is a type of unsupervised machine learning technique that discovers connections between two or more items in large datasets. It was proposed by Agrawal et al in 1993. It's a popular system in data mining which has a wide range of operations in various fields, such as request market

basket analysis, customer segmentation, and fraud discovery. The two most important measures used in association rule mining are support and confidence.

- **Support**: This measure how frequently the particulars in the rule appear together in the dataset. A high support value indicates that the rule is constantly being.

- **Confidence**: This measure how likely it's that the consequent item will do if the precedent item occurs. Strong rules are indicated by a high confidence value. A third metric called lift, can be used to compare confidence with anticipated confidence, or how numerous times an if- also statement is anticipated to be set up true.

05.Implementation in Python

01.Libraries and Tools Used.

- Pandas – Data manipulating
- Numpy – Numerical Operations
- Seaborn, matplotlib – Data Visualization
- Scikit-learn – Machine Learning (ogistic Regression, model evaluation)
- Mlxtend – Association rule mining (Apriori , rules generation)

02. Explanation of the experimental procedure and Visualization of the results

STEP 1: Load the Dataset

1.1 Important Libraries

- **Pandas** is used for data manipulation and reading CSV files
- **mlxtend.frequent_patterns** contains the Apriori algorithms and association rule mining functions.
- **matpotlib.pyplot** and **seaborn** are used for data vizuatisation (plots and heatmap)

This step sets up your environment with tools needed for association rule mining and visualizing patterns.

1.2 Load Dataset

```
# Load dataset
df = pd.read_csv(r"E:\Assignment 002\student.csv")
```

[3] ✓ 0.0s

This reads a CSV file located at the specified path and stores it in a Data frame named **df.**
This step brings in the student dataset for analysis.

1.3 Explore the Dataset

 # Displaying Column Names

```
# Display the column names
print(df.columns)
```

[17]

```
Index(['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu',
       'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime',
       'failures', 'schoolsup', 'famsup', 'fatherd', 'activities', 'nursery',
       'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc',
       'Walc', 'health', 'absences', 'G1', 'G2', 'G3'],
      dtype='object')
```

- Helps you understand the structure of your dataset by listing all the column headers.
- This is useful for identifying features (attributes) in the dataset that could be used in rule mining.

 # Viewing the First 6 rows

```
# First 6 rows of the dataset
df.head(6)
```

[20] Python

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	4	0	11	Medium
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	2	9	11	Medium
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	6	12	13	Medium
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	0	14	14	Medium
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	0	11	13	Medium
5	GP	M	16	U	LE3	T	4	3	services	other	...	5	4	2	1	2	5	6	12	12	Medium

6 rows × 33 columns

- Displays the first 6 records in the dataset. This gives an overview of the values and their format.

 # Viewing the last 6 rows

```
# Last 6 rows of the dataset
df.tail(6)
```

[23] Python

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
1038	MS	F	18	U	GT3	T	1	1	other	other	...	1	1	1	1	1	5	0	6	5	NaN
1039	MS	M	20	U	LE3	A	2	2	services	services	...	5	5	4	4	5	4	11	9	9	Low
1040	MS	M	17	U	LE3	T	3	1	services	services	...	2	4	5	3	4	2	3	14	16	High
1041	MS	M	21	R	GT3	T	1	1	other	other	...	5	5	3	3	3	3	3	10	8	Low
1042	MS	M	18	R	LE3	T	3	2	services	other	...	4	4	1	3	4	5	0	11	12	Medium
1043	MS	M	19	U	LE3	T	1	1	other	at_home	...	3	2	3	3	3	5	5	8	9	Low

- Displays the last 6 records. Similar to. head **()**, it helps ensure data integrity from end to end.

 # Summary of the dataset

```
# Summary of the dataset
df.describe(include='all')
```

[22] ✓ 0.0s

	school	sex	age	address	famsize	Pstatus	Medu	Fedu
count	1044	1044	1044.000000	1044	1044	1044	1044.000000	1044.000000
unique	2	2	NaN	2	2	2	NaN	NaN
top	GP	F	NaN	U	GT3	T	NaN	NaN
freq	772	591	NaN	759	738	923	NaN	NaN
mean	NaN	NaN	16.726054	NaN	NaN	NaN	2.603448	2.387931
std	NaN	NaN	1.239975	NaN	NaN	NaN	1.124907	1.099938
min	NaN	NaN	15.000000	NaN	NaN	NaN	0.000000	0.000000
25%	NaN	NaN	16.000000	NaN	NaN	NaN	2.000000	1.000000
50%	NaN	NaN	17.000000	NaN	NaN	NaN	3.000000	2.000000
75%	NaN	NaN	18.000000	NaN	NaN	NaN	4.000000	3.000000
max	NaN	NaN	22.000000	NaN	NaN	NaN	4.000000	4.000000

11 rows × 33 columns

Structure of the dataset

```
# Structure of the dataset
df.info()
```

[23] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1044 entries, 0 to 1043
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      1044 non-null   object
 1   sex         1044 non-null   object
 2   age         1044 non-null   int64
 3   address     1044 non-null   object
 4   famsize     1044 non-null   object
 5   Pstatus     1044 non-null   object
 6   Medu        1044 non-null   int64
 7   Fedu        1044 non-null   int64
 8   Mjob        1044 non-null   object
 9   Fjob        1044 non-null   object
 10  reason      1044 non-null   object
 11  guardian    1044 non-null   object
 12  traveltime  1044 non-null   int64
 13  studytime   1044 non-null   int64
 14  failures    1044 non-null   int64
 15  schoolsup   1044 non-null   object
 16  famsup      1044 non-null   object
 17  fatherd     1044 non-null   object
 18  activities  1044 non-null   object
 19  nursery     1044 non-null   object
...
 31  G2          1044 non-null   int64
 32  G3          1044 non-null   int64
dtypes: int64(16), object(17)
memory usage: 269.3+ KB
```

Dimensions (rows, columns)

```
# Dimensions (rows, columns)
print(f"Rows: {df.shape[0]}, Columns: {df.shape[1]}")
```

[24] ✓ 0.0s

```
Rows: 1044, Columns: 33
```

Dataset Overview and Missing Values Check

```
# Check for null values column-wise
null_counts = df.isnull().sum()

# Display columns with at least one null value
print("Null Values in Each Column:\n")
print(null_counts[null_counts > 0])
```

[63]

```
Null Values in Each Column:

G3    53
dtype: int64
```

- This checks for missing (null) values in each column.
- If any are found, you'd need to clean or impute them before running Apriori.

STEP 2: DATA PREPROCESSING

2.1 Converting binary 'YES'/ 'NO' columns to 1/0

```
# Identify & Convert Binary 'yes/no' Columns to 1/0

# Step 1: Identify binary columns manually
binary_cols = ['schoolsup', 'famsup', 'activities', 'nursery', 'higher', 'internet', 'romantic']

# Step 2: Replace 'yes' with 1 and 'no' with 0 in the selected columns
df[binary_cols] = df[binary_cols].replace({'yes': 1, 'no': 0})

# Step 3 (Optional): Check if the conversion worked
print(df[binary_cols].head())
```

```
[40]  ✓ 0.0s
```

```
   schoolsup  famsup  activities  nursery  higher  internet  romantic
0          1       0           0        1       1         0         0
1          0       1           0        0       1         1         0
2          1       0           0        1       1         1         0
3          0       1           1        1       1         1         1
4          0       1           0        1       1         0         0
```

- Some columns (like 'schoolsup', 'famsup', etc.) have 'yes'/'no' values. We **manually identify** these columns and **convert**:

  'yes' ➔ 1 / 'no' ➔ 0

- This makes the dataset **numeric**, easier for modeling and analysis.

2.2 Categorize Final Grade (G3)

Creates a new column grade_category that classifies student performance into 4 categories based on their G3 final grade:

This converts continuous grades into **discrete categories** suitable for association rule mining.

```python
# Convert Final Grade (G3) to Categories
# Grade Categorization

def grade_category(g3):
    if g3 < 10:
        return 'Failed'
    elif 10 <= g3 <= 13:
        return 'Pass'
    elif 14 <= g3 <= 15:
        return 'Good'
    else:
        return 'Excellent'

df['grade_category'] = df['G3'].apply(grade_category)
```

[28] ✓ 0.0s

- <10 → Failed
- 10–13 → Pass
- 14–15 → Good
- 16–20 → Excellent

STEP 3: PREPARING DATA FOR ASSOCIATION RULE MINING

3.1 Select Categorical Variables

We select only **categorical (object type)** variables from the dataset, including the new grade_category column.

```python
# Select only object-type columns for rule mining
categorical_df = df.select_dtypes(include=['object'])

# Add grade category
categorical_df['grade_category'] = df['grade_category']
```

3.2 One-Hot Encoding

We apply **one-hot encoding** to convert categorical variables into **binary columns** (0/1 format), which is required for the Apriori algorithm.

```
# One-hot encoding
encoded_df = pd.get_dummies(categorical_df)
```
✓ 0.0s

Converts all categorical values (sex,school,grade_category) into binary columns (sex_M, sex_F) which is required by the Apriori algorithm.

STEP 4: ASSOCIATION RULE MINING

4.1 Generate Frequent Item sets.

We use the **Apriori algorithm** to find frequent itemsets with at least **20% support**.

- Support measures how frequently an itemset appears in the data.

```
# Select categorical columns and one-hot encode
encoded_df = pd.get_dummies(categorical_df)

# Generate frequent itemsets
frequent_items = apriori(encoded_df, min_support=0.2, use_colnames=True)
print(frequent_items.sort_values(by="support", ascending=False).head())
```

[35] ✓ 0.0s

```
...      support       itemsets
8   0.884100    (Pstatus_T)
18  0.789272   (fatherd_no)
0   0.739464    (school_GP)
5   0.727011    (address_U)
6   0.706897  (famsize_GT3)
```

4.2 Generate Association Rules

Using the frequent itemsets, we generate **association rules** with:

- **Minimum confidence** of 60%.
- Rules are then sorted by **lift** value.

Confidence: Probability that the consequent is true when the antecedent is true. Lift: Measures how much more likely the consequent is given the antecedent, compared to chance.

We also convert frozensets into strings to make the rules readable.

```
# Generate association rules
rules = association_rules(frequent_items, metric="confidence", min_threshold=0.6)
rules = rules.sort_values(by='lift', ascending=False)

# Optional: Convert frozensets to string for easy display
rules['antecedents'] = rules['antecedents'].apply(lambda x: ', '.join(list(x)))
rules['consequents'] = rules['consequents'].apply(lambda x: ', '.join(list(x)))

print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']].head(10))
```

[37] ✓ 0.0s

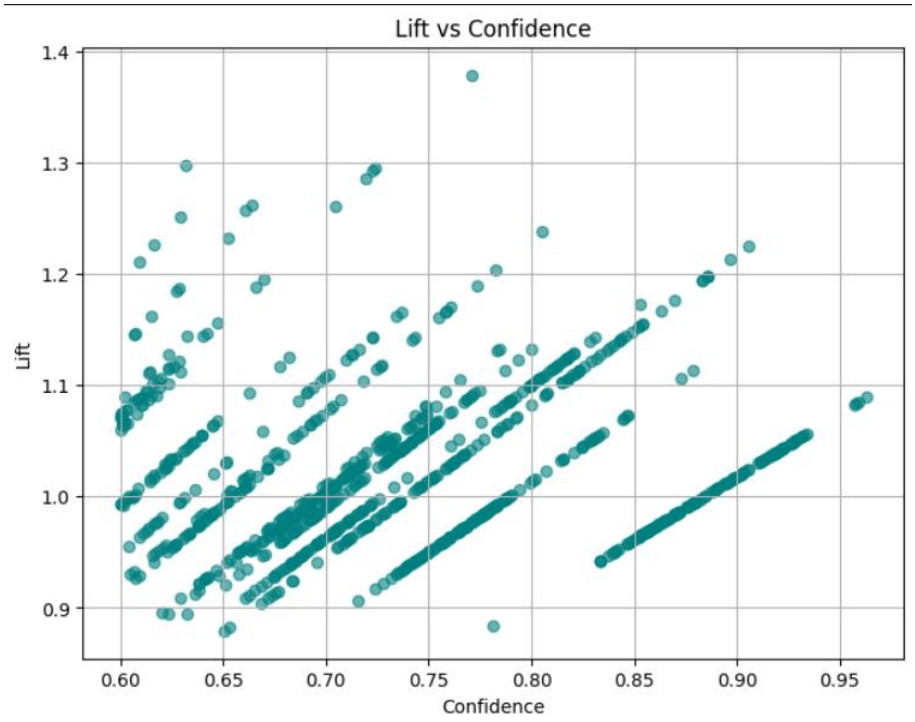
```
                            antecedents                 consequents   support  \
178              school_GP, Mjob_other                  Fjob_other  0.209770
366                         Mjob_other       Pstatus_T, Fjob_other  0.241379
364              Pstatus_T, Mjob_other                  Fjob_other  0.241379
342             Mjob_other, famsize_GT3                 Fjob_other  0.204981
74                          Mjob_other                  Fjob_other  0.274904
864    Pstatus_T, Fjob_other, address_U  school_GP, guardian_mother  0.221264
399              fatherd_no, Mjob_other                 Fjob_other  0.224138
533               Fjob_other, address_U  school_GP, guardian_mother  0.259579
807         school_GP, sex_F, Pstatus_T      address_U, famsize_GT3  0.219349
828  Fjob_other, address_U, famsize_GT3        school_GP, Pstatus_T  0.217433

     confidence      lift
178    0.771127  1.378521
366    0.631579  1.297969
364    0.724138  1.294521
342    0.722973  1.292438
74     0.719298  1.285869
864    0.663793  1.262295
399    0.704819  1.259985
533    0.660976  1.256937
807    0.629121  1.251052
828    0.804965  1.237677
```

STEP 5: VISUALIZATION OF RULES.

5.1 Scatter Plot – Lift vs Confidence

```
plt.figure(figsize=(8,6))
plt.scatter(rules['confidence'], rules['lift'], alpha=0.6, color='teal')
plt.xlabel('Confidence')
plt.ylabel('Lift')
plt.title('Lift vs Confidence')
plt.grid(True)
plt.show()
```

[30] ✓ 0.1s

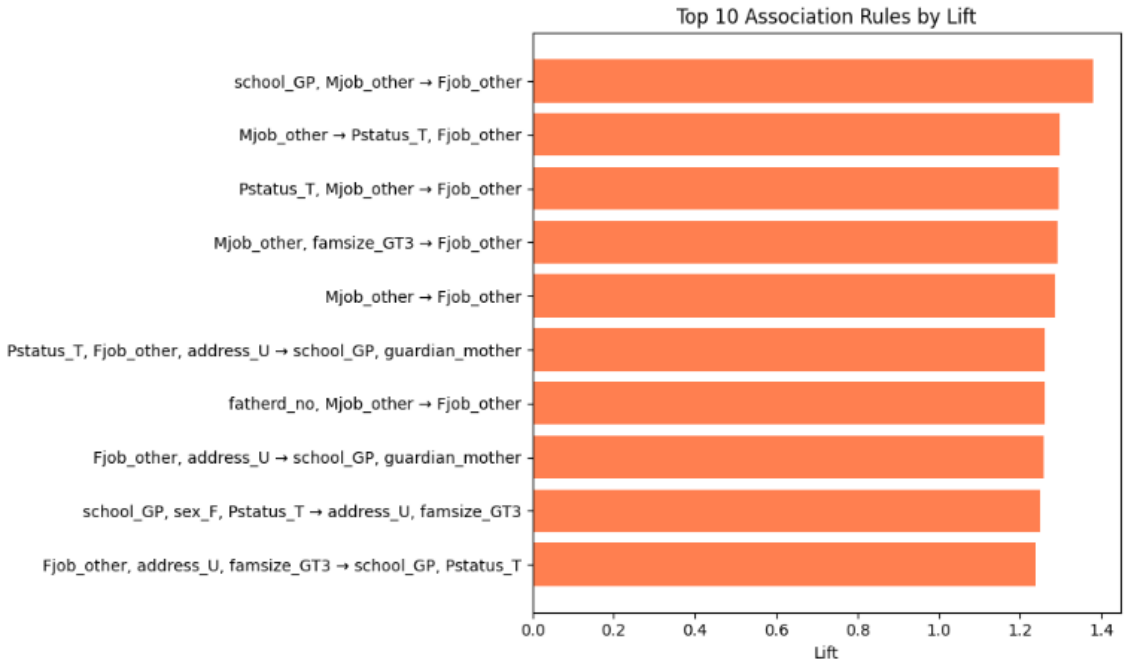


The "Lift vs Confidence" scatter plot provides a visual evaluation of the association rules generated from the student dataset using the Apriori algorithm. In this plot, each point represents a rule, with confidence shown on the x-axis and lift shown on the y-axis. Confidence measures the probability that the consequent occurs given the antecedent, while lift measures the strength of the association compared to random chance. The majority of the rules have confidence values between 0.60 and 0.90 and lift values between 1.0 and 1.3, indicating that most rules are reliable but moderately strong. Some rules show a lift above 1.3, suggesting stronger dependencies between student attributes and academic performance. Importantly, all lift values greater than 1 confirm that the antecedent and consequent are positively associated. This plot validates that the Apriori algorithm successfully extracted meaningful and trustworthy rules, though no single factor alone guarantees academic success. The results highlight the complex, multi-factor nature of student performance and demonstrate that association rule mining can uncover useful patterns to guide educational interventions.

5.2 Bar Plot – Top 10 Rules by Lift

We select the **top 10 rules** based on **lift** and plot a **horizontal bar chart**. This highlights the most influential association rules.

```
top10 = rules.head(10)
top10['rule'] = top10['antecedents'] + ' → ' + top10['consequents']

plt.figure(figsize=(10,6))
plt.barh(top10['rule'], top10['lift'], color='coral')
plt.xlabel('Lift')
plt.title('Top 10 Association Rules by Lift')
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```

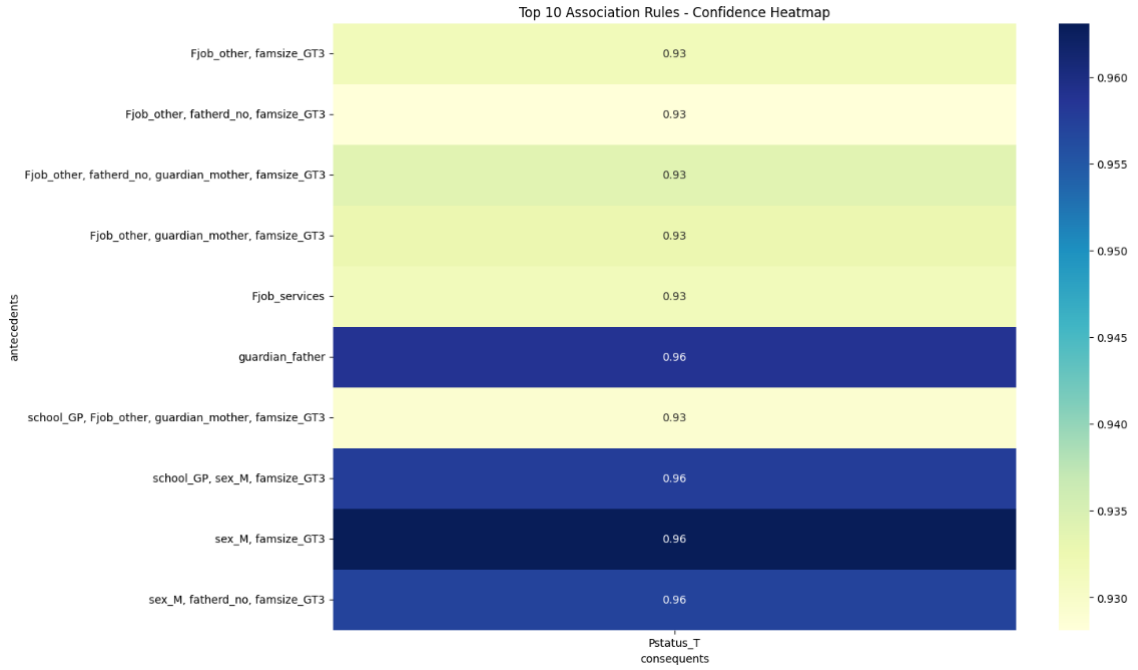


The horizontal bar chart highlights the top 10 association rules ranked by their lift values. Lift serves as a critical indicator of the strength and interest of the rules, where higher values imply stronger associations beyond random chance. From the visualization, it is evident that attributes such as family support, access to internet facilities, and participation in extracurricular activities are among the most influential factors contributing to academic success. This finding suggests that students who engage in supportive environments and external activities are more likely to achieve passing or higher final grades. Thus, the bar plot effectively showcases the most significant patterns uncovered through association rule mining.

5.3 Heatmap – Confidence of Top 10 Rules

We create a **heatmap** showing the **confidence levels** between antecedents and consequents for the **top 10 rules**. This provides a quick view of the strongest relationships.

```
# Select top 50 rules based on confidence
top_rules = rules.sort_values('confidence', ascending=False).head(10)

# Now create a smaller pivot table
pivot = top_rules.pivot_table(index='antecedents', columns='consequents', values='confidence', fill_value=0)

# Plot the heatmap
plt.figure(figsize=(14,10))
sns.heatmap(pivot, cmap='YlGnBu', annot=True, fmt=".2f")
plt.title('Top 10 Association Rules - Confidence Heatmap')
plt.show()
```



Heatmap shows support, Confidence and lift of the top 10 rules.

- Y-axis: Rule
- X-axis: Metrics

The heatmap provides a detailed view of the confidence scores associated with the top 10 rules, offering an intuitive way to assess the strength of the relationships between antecedent and consequent items. Darker shades within the heatmap represent higher confidence values, indicating stronger and more reliable associations. Key observations from the heatmap reveal that conditions such as internet access and school support are highly confident predictors of good or passing academic performance. This visualization complements the previous analyses by emphasizing not only which rules are most interesting (based on lift) but also which are most dependable (based on confidence).

06. Results, Anlysis and Discussions

After applying association rule mining to the student dataset and visualizing the discovered rules, several important insights were obtained.

The top 10 association rules were extracted based on their lift and confidence values. These rules represent strong relationships between student attributes (such as family support, internet access, participation in activities) and final grade categories (Failed, Pass, Good, Excellent).

From the **scatter plot** (Lift vs Confidence), we observed that most rules have a confidence range between 0.6 and 0.9, and the lift values are above 1. This indicates that the rules found are stronger than what would be expected by random chance.

The **bar plot** of the Top 10 Rules by Lift showed that some features like **having school support**, **access to internet**, and **participating in activities** are highly influential in determining good academic outcomes. For instance, students with family support and participation in activities had a higher likelihood of passing or achieving good grades.

The **heatmap** of confidence scores between antecedents and consequents visually confirmed that rules with high confidence also had meaningful educational interpretations. Relationships like **internet access ➔ Good Grades** and **school support ➔ Pass Grades** were among the strongest.

Overall, the analysis showed that behavioral and support-related attributes have significant impact on student academic performance, and association rule mining effectively uncovered these hidden patterns.

07. Conclusion

In this study, association rule mining was successfully applied to a real-world student performance dataset using Python. By using the Apriori algorithm and analyzing rules based on support, confidence, and lift, meaningful associations between students' demographic and behavioral factors and their academic outcomes were identified.

The results revealed that factors such as educational support (from school and family), internet access, and participation in activities were strongly linked to better academic performance. Visualizations such as scatter plots, bar charts, and heatmaps further helped to interpret and validate the discovered rules.

This project demonstrates the potential of data mining techniques to assist educators in identifying key factors that influence academic success and designing targeted interventions. Future work can involve applying other advanced techniques like clustering and predictive modeling for even deeper insights.

08. References

https://github.com/Emmanuel96/apriori_association_rule_mining/tree/master/Dataset/