Data Mining Techniques on 'CAR' Dataset

Haji Akhundzada,Arzuman Hasanov,Ravan Iskandarov

1 Introduction

Pattern mining plays a pivotal role in data analysis by uncovering hidden associations and trends within large datasets. In this project, we aim to apply data mining techniques to the 'CAR' dataset, which comprises car attributes and their acceptability ratings. The dataset has 1727 rows and 7 columns.The primary objective is to explore the dataset using pattern mining algorithms, specifically Apriori, FP-Growth, and Eclat, to identify significant patterns and association rules.

2 Data Preprocessing

Data preprocessing is a critical step in the data mining process that involves cleaning, transforming, and preparing the raw data for analysis. In the context of the 'CAR' dataset used in this project, several preprocessing steps have been implemented to ensure the data is in a suitable format for applying data mining algorithms effectively.

2.1 Data Loading and Inspection:

The first step in data preprocessing is loading the 'CAR' dataset into a pandas DataFrame, allowing for easy manipulation and analysis of the data.The dataset is inspected to understand its structure, features, and any missing values that may require handling.

2.2 Handling Missing Values:

Missing values in the dataset are identified and addressed to prevent any inconsistencies or errors in the analysis 1.

Techniques such as dropping rows with missing values or imputing missing values with appropriate strategies can be employed to maintain data integrity.

2.3 Label Encoding and Categorical Data:

Categorical variables in the dataset, such as buying price, maintenance cost, and safety rating, are encoded using techniques like LabelEncoder to convert them into numerical format for algorithm compatibility.This transformation ensures that categorical data can be effectively utilized in data mining algorithms that require numerical input.

2.4 Handling Class Imbalance:

Class imbalance, where certain classes in the dataset are underrepresented compared to others(Figure 1), is addressed through techniques like undersampling. Undersampling involves reducing the number of instances in the majority class to balance the class distribution, ensuring fair representation of all classes in the analysis.

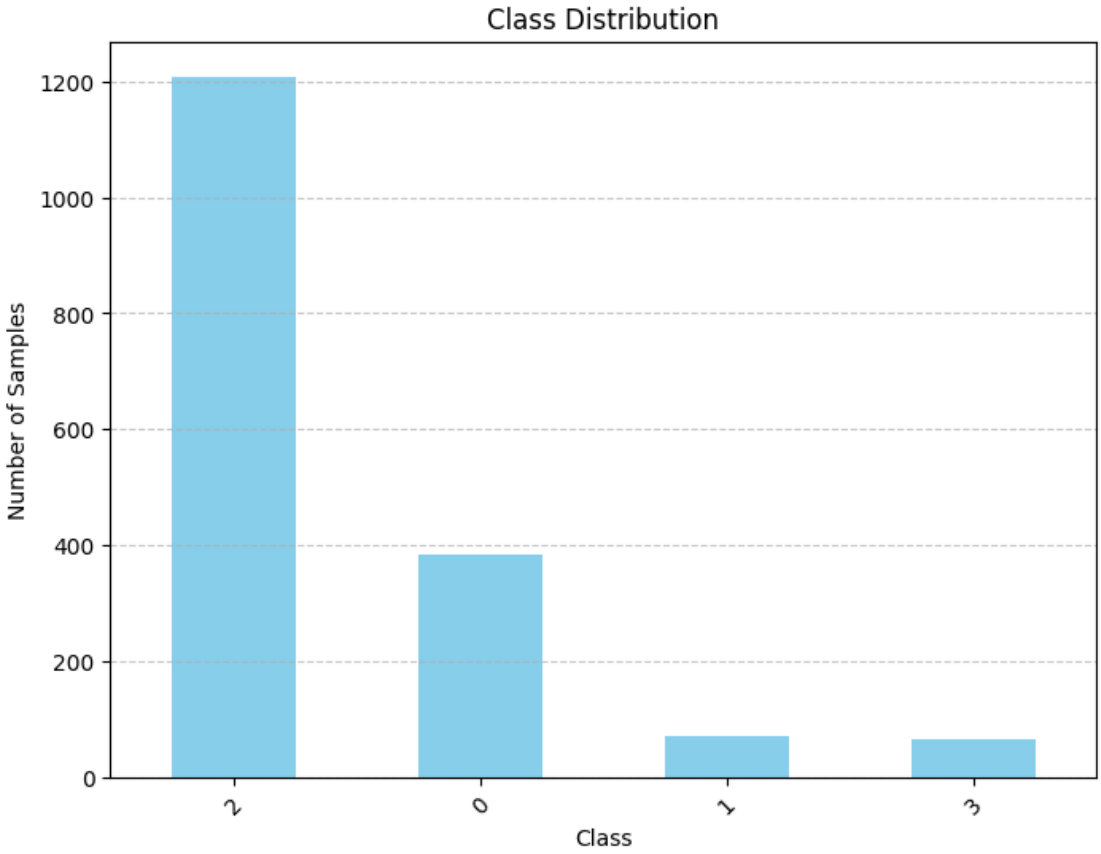

Figure 1

3 Algorithms Selection

Three distinct pattern mining algorithms were chosen for their diverse approaches:

Apriori:

Apriori is a classic association rule mining algorithm used to discover frequent itemsets in transactional databases. It employs a level-wise approach to generate frequent itemsets by iteratively scanning the dataset and pruning infrequent itemsets. Apriori is effective for mining associations between items in categorical datasets and is widely used for market basket analysis and recommendation systems. However, it

can be computationally expensive for large datasets due to multiple passes over the data to find frequent itemsets. It implemented using the spmf library.

FP-Growth:

FP-Growth (Frequent Pattern Growth) is an efficient association rule mining algorithm that uses a tree structure to mine frequent itemsets. It constructs a compact data structure called the FP-tree to encode the dataset and efficiently mine frequent itemsets without generating candidate itemsets.It also implemented using the spmf library.FP-Growth is depth based algorithm.

Eclat:

A vertical data format-based algorithm, implemented using the SPMF library. Unlike Apriori, ECLAT does not generate candidate itemsets but instead focuses on intersecting transaction IDs of items to determine their support.

4 Implementation and Results

4.1 Apriori

The Apriori algorithm, a classic association rule mining technique, was implemented using the SPMF (Sequential Pattern Mining Framework) library. The dataset was preprocessed and converted into a suitable format for sequence mining. The minimum support threshold was set to 0.3, indicating that itemsets occurring in at least 30% of the sequences were considered frequent.

Upon running the Apriori algorithm, the following results were obtained:

```
=============  APRIORI - STATS =============
 Candidates count : 55
 The algorithm stopped at size 2
 Frequent itemsets count : 10
 Maximum memory usage : 8.442001342773438 mb
 Total time ~ 0 ms
===================================================
```

Figure 2

These results indicate that the Apriori algorithm successfully identified frequent itemsets from the dataset based on the specified minimum support threshold. The algorithm terminated at itemset size 2, suggesting that no larger frequent itemsets were found within the given threshold.

Association rule generation(with minsup=0.2, minconf=0.3):

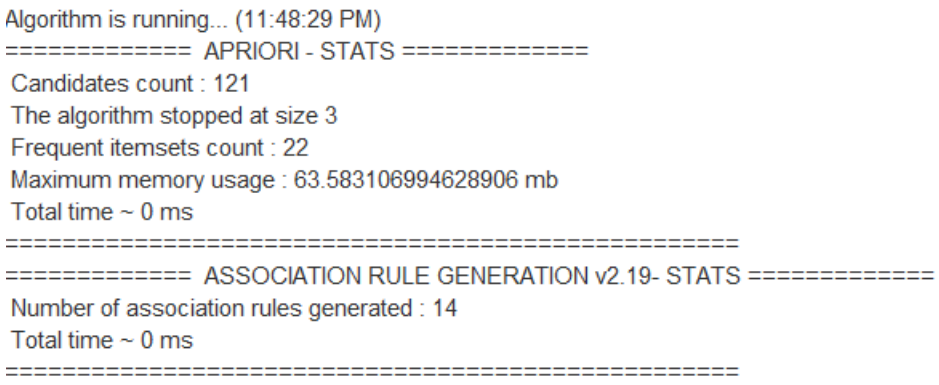

Figure 3

4.2 FP-Growth

The FP-Growth algorithm, another popular method for frequent itemset mining, was implemented using the SPMF library. Similar to the Apriori implementation, the dataset was preprocessed and converted into a suitable format for sequence mining. The minimum support threshold was set to 0.3, indicating that itemsets occurring in at least 30% of the sequences were considered frequent.

Upon running the FP-Growth algorithm, the following results were obtained:

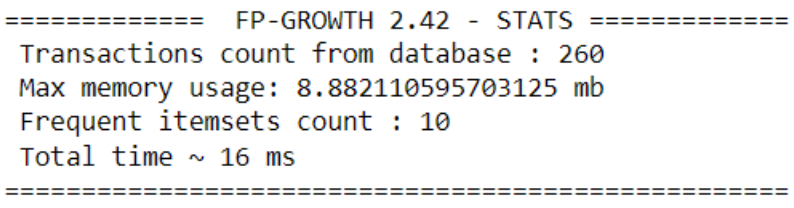

Figure 4

These results highlight the efficiency and effectiveness of the FP-Growth algorithm in identifying frequent itemsets from the dataset. Despite the differences in approach compared to the Apriori algorithm, FP-Growth demonstrated comparable performance in terms of identifying frequent patterns within the dataset.

Association rule generation(with minsup=0.2, minconf=0.3):

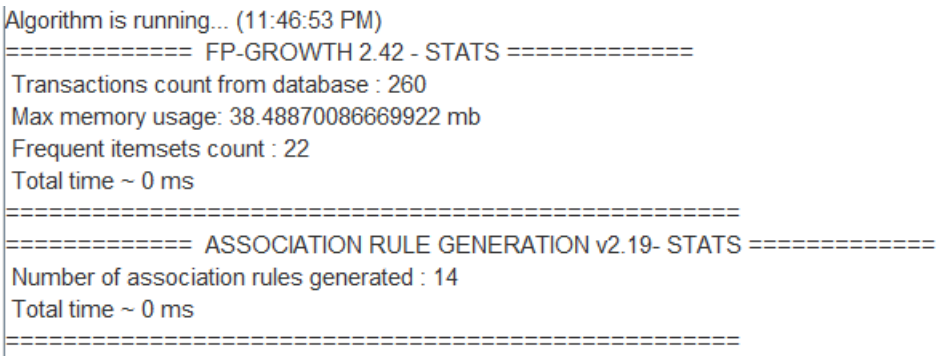

Figure 5

4.3 Eclat

The ECLAT (Equivalence Class Transformation) algorithm, another powerful technique for frequent itemset mining, was implemented using the SPMF library. Similar to the Apriori and FP-Growth implementations, the dataset was preprocessed and converted into a suitable format for sequence mining. The minimum support threshold was set to 0.3, indicating that itemsets occurring in at least 30% of the sequences were considered frequent.

Upon running the ECLAT algorithm, the following results were obtained:

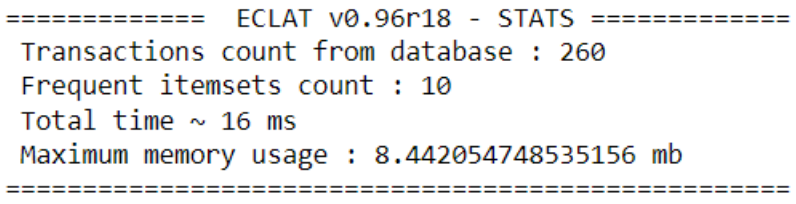

Figure 6

These results demonstrate the efficiency and effectiveness of the ECLAT algorithm in identifying frequent itemsets from the dataset. Despite its simplicity compared to other algorithms like Apriori and FP-Growth, ECLAT achieved competitive performance in terms of memory usage.

5 Comparison

1)Execution Time:

- Apriori: The Apriori algorithm exhibited the fastest execution time among the three algorithms, with a total time of approximately 0 milliseconds.
- FP-Growth: FP-Growth demonstrated a slightly longer execution time compared to Apriori, with a total time of around 16 milliseconds.

- ECLAT: ECLAT performed comparably to Apriori in terms of execution time, with a negligible total time of nearly 0 milliseconds.

2)Memory Usage:

- Apriori: The maximum memory usage observed during the execution of Apriori was approximately 8.44 MB.
- FP-Growth: FP-Growth utilized slightly more memory compared to Apriori, with a maximum memory usage of around 8.88 MB.
- ECLAT: ECLAT consumed a similar amount of memory as Apriori, with a maximum memory usage of approximately 8.44 MB.

3)Scalability:

- Apriori: While Apriori is known for its simplicity and ease of implementation, it may face scalability issues when dealing with large datasets due to the generation of a large number of candidate itemsets.
- FP-Growth: FP-Growth addresses scalability limitations by constructing a compact FP-tree data structure, making it highly efficient for mining frequent patterns from large datasets.
- ECLAT: ECLAT is a vertical approach that avoids candidate generation, making it highly scalable and memory-efficient, particularly suitable for datasets with a high number of transactions.

4)Overall Performance:

- Apriori: Apriori serves as a baseline algorithm for frequent pattern mining and is suitable for small to moderate-sized datasets.
- FP-Growth: FP-Growth offers superior performance in terms of execution time and memory usage compared to Apriori, making it ideal for large-scale datasets.
- ECLAT: ECLAT provides excellent scalability and memory efficiency, making it a preferred choice for datasets with a high number of transactions or limited memory resources.

6 Conclusion

In conclusion, the utilization of Apriori, FP-Growth, and Eclat algorithms on the 'CAR' dataset provided valuable insights into the relationships among car attributes and their acceptability ratings. Each algorithm demonstrated distinct strengths and limitations, emphasizing the significance of selecting the appropriate algorithm

based on dataset characteristics and analytical goals. Future investigations could delve into hybrid methodologies or alternative algorithms to optimize pattern mining efficiency and effectiveness across various domains.