Association Rule Mining of Emergency 911 Calls

Genevieve Mortensen, Mahsa Monshizadeh, Aditya Shahapure

December 2023

Abstract

This proposal presents a methodology for data mining to analyze the 911 emergency dataset using association rule mining techniques. The objective is to uncover valuable insights from the dataset, identifying meaningful patterns and associations. By doing so, the project aims to shed light on the influential factors affecting the occurrence and characteristics of emergency calls. The anticipated results of this study aim to support a more data-centric and streamlined emergency response system. This, in turn, is expected to optimize resource allocation and enhance the overall handling of critical situations, ultimately contributing to the community's public safety and well-being.

Keywords

data mining, 911 emergency, association rule mining, emergency calls, public safety

1 Introduction

Emergency services play a pivotal role in the fabric of any community, serving as a crucial lifeline during unforeseen crises. The 911 emergency call dataset, generating a wealth of data, provides a unique and valuable opportunity to extract actionable insights aimed at enhancing emergency response systems. By applying association rule mining to this dataset, a thorough exploration of intricate relationships among various parameters associated with emergency calls—such as time, location, incident type, and other pertinent factors—becomes possible.

Delving deeper into the intricacies of this dataset allows us to unravel hidden patterns and correlations, shedding light on the nuanced dynamics of emergency situations. The objective is to go beyond conventional analyses and gain a comprehensive understanding of the multifaceted interplay between different variables. This deeper insight holds the potential to revolutionize decision-making processes within emergency services. It enables the identification of trends and dependencies that may have previously gone unnoticed, empowering authorities to make more informed and strategic choices.

Moreover, the outcomes of this analysis can significantly impact resource deployment strategies. By pinpointing correlations between specific factors and emergency incidents, we can optimize the allocation of resources, ensuring a more efficient and effective response. The overarching goal is to streamline emergency services, making them not only more responsive but also better tailored to the unique demands of each situation. In essence, the application of association rule mining to the 911 emergency call dataset represents a crucial step towards fortifying community safety and well-being through data-driven improvements in emergency services.

2 Method

The proposed methodology employs a comprehensive approach to analyze the 911 emergency dataset, leveraging association rule mining techniques, with a primary focus on the Apriori algorithm. This method unfolds through distinct phases, encompassing data preprocessing, the application of the Apriori algorithm, and the subsequent interpretation of association rules.

To implement this project, we deviated from the initially suggested R implementation via GeeksforGeeks and instead utilized the Python programming language. This shift allowed us to harness the capabilities of relevant data mining and visualization libraries for a more versatile and efficient workflow. The project made use of Kaggle as the platform for dataset access and management, providing a collaborative and accessible environment.

The analysis began with a meticulous data preprocessing stage, ensuring the dataset's readiness for subsequent mining processes. Subsequently, the Apriori algorithm, a well-established association rule mining technique, was applied to uncover meaningful patterns within the dataset. Visualization of intriguing relationships was achieved through exploratory data analysis, utilizing Python-based libraries to enhance interpretability.

In addition to the programming language and libraries, Google Colab was employed as the primary development environment. This cloud-based platform facilitated seamless collaboration, resource sharing, and ensured accessibility to the necessary computational power for efficient processing of the extensive emergency dataset.

In summary, the methodology integrates Python, Kaggle, and Google Colab, combining data preprocessing, Apriori algorithm application, and rule interpretation to extract meaningful insights from the 911 emergency dataset. The inclusion of exploratory data analysis and external validation through news coverage enhances the robustness of the findings, contributing to a more comprehensive understanding of emergency call patterns.

2.1 Dataset

The 911 emergency call dataset used in this study is publicly accessible through the Emergency Dispatch Operation department for Montgomery County, PA, and is available on Kaggle as a unified CSV file containing 663,282 unique emergency calls. The dataset encompasses various attributes related to individual 911 calls responded to by the emergency dispatch for a single county. Information includes details about the specific time and location of each emergency, descriptions of the incidents, and labels associated with the services provided in response to dispatched calls.

The dataset provides location details through multiple attributes, including longitude, latitude, township, ZIP code, and general address. With distinct and diverse categories within the dataset, there is ample potential to extract numerous meaningful association rules. Notably, latitude, longitude, and ZIP code are represented as float values, while the emergency description (desc), emergency title, date/time (timeStamp), township (twp), and general address (addr) are represented as object values.

2.2 Data Preprocessing and Exploratory Data Analysis

In the preprocessing phase, we made adjustments to enhance the generalizability of the dataset. The continuous ”timeStamp” attribute was discretized, considering the season of the year and the time of day. This transformation contributes to better generalizability across diverse scenarios. Additionally, rows with missing values were removed to ensure data integrity.

Further refinement involved the removal of less pertinent columns, resulting in a streamlined dataset. Notably, the ”service” column, representing fire, Emergency Medical Services (ESM), or traffic incidents, was retained for analysis. Figure 1a illustrates the distribution of services, offering an overview of their prevalence.

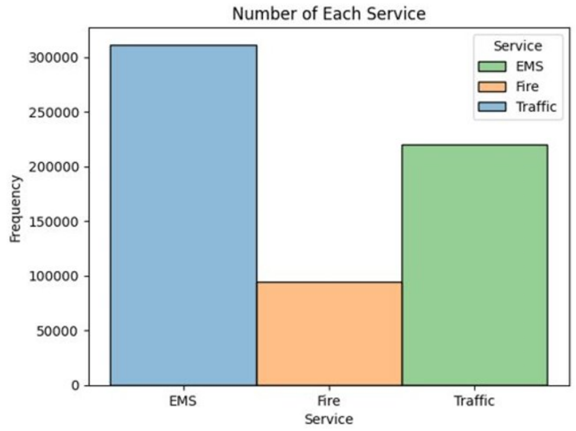

Figure 1a: Histogram showing the frequency of each emergency service getting dispatched to an emergency location.

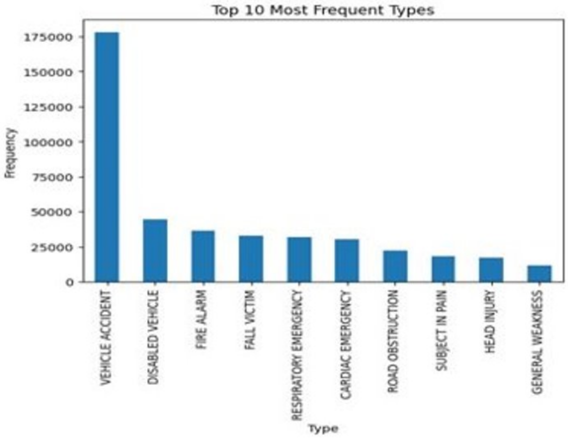

Figure 1b: Histogram showing the frequencies of the top ten most frequent accidents.

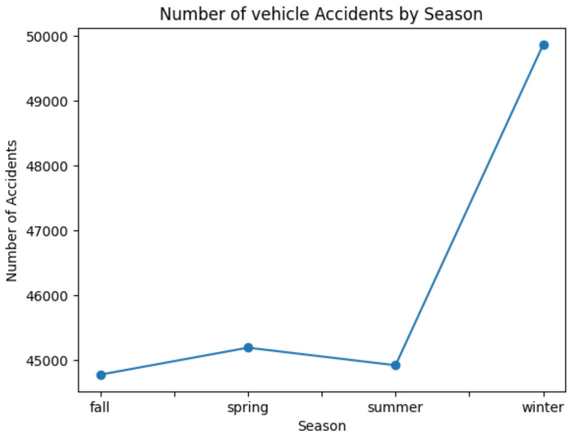

Figure 2a: Line plot showing the number of vehicle accidents per season.

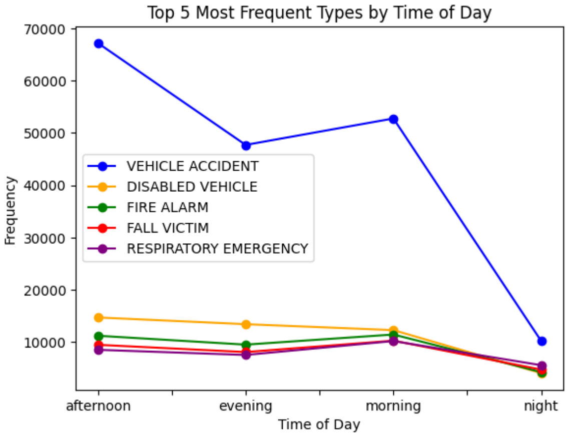

Figure 2b: Line plot showing the top five most frequent types of emergencies per each time of the day. Days were categorized equally into 4 intervals, each lasting 6 hours long.

Another significant column, the ”type of incident,” was explored to understand the top 10 incident categories, as depicted in Figure 1b. Additionally, columns detailing the season, time of day, and day of the week were introduced for a more comprehensive analysis.

During the Exploratory Data Analysis (EDA) phase, insights were uncovered and visualized. The visualization in a) shows that most emergencies are dispatched to EMS whereras b) shows that most emergencies are vehicle accidents. One might think that traffic police would be called more often considering this observation. Figure 2a showcases the number of vehicle accidents distributed across different seasons. Meanwhile, Figure 2b provides a glimpse into the top five most frequent incident types categorized by the time of day. We can expect some temporal association rules to reflect the results of our EDA. These results offer valuable perspectives on the dataset’s characteristics, providing a foundation for subsequent analyses.

2.3 Association Rule Mining

Association Rule Mining stands as a powerful technique for uncovering meaningful relationships within expansive datasets, guided by the parameters of Support and Confidence. Support gauges the frequency of itemset occurrence, while Confidence measures the strength of associations between items in transactions. To navigate our extensive dataset, we opted for a lower minimum support (0.01) to ensure ample itemset discovery, and a high Confidence threshold (0.8) to enhance the reliability of identified rules.

2.3.1 Apriori Algorithm Approaches

The Apriori algorithm, a seminal contribution by Rakesh Agrawal and Ramakrishnan Srikant in 1994, was selected as the cornerstone for our Association Rule Mining endeavors. Apriori efficiently uncovers frequent itemsets, paving the way for the generation of insightful association rules. Its foundational principle, the "apriori property," asserts that if an itemset is frequent, all its subsets must also be frequent—a guiding notion that underpins the algorithm's systematic approach.

Key Steps in Apriori Algorithm:

Frequent Itemset Generation: Apriori initiates the process by identifying itemsets that meet a minimum support threshold, efficiently pruning infrequent itemsets from further consideration.

Candidate Generation: Employing a breadth-first search strategy, Apriori systematically generates candidate itemsets, avoiding redundancy by excluding subsets of infrequent itemsets.

Association Rule Generation: Leveraging the identified frequent itemsets, Apriori crafts association rules based on the concept of confidence, highlighting statistically significant relationships between different items.

Rule Pruning: To enhance interpretability, Apriori applies a confidence threshold, pruning rules that fail to meet this criterion and refining the set of meaningful associations.

Comparative Analysis of Apriori Approaches:

Three distinct approaches were implemented to harness the power of the Apriori algorithm, each shedding light on its nuances and efficiency:

1. **Apriori with Candidate Generator of Fk-1 * Fk-1 (about 45 minutes):**

   This approach involves generating candidate itemsets by combining frequent itemsets with themselves. While time-efficient, it may not be the most memory-efficient and is influenced by dataset characteristics.

2. **Apriori with Candidate Generator of Fk-1 * Fk1 (about 1h and 50 minutes):**

   Candidate itemsets are generated by combining frequent itemsets with frequent singletons. This method is more time-consuming but yields a potentially more comprehensive set of rules, impacting computation time and memory usage.

   We conduct a comparison between first 2 apriori methods you can find the results in figure 3.

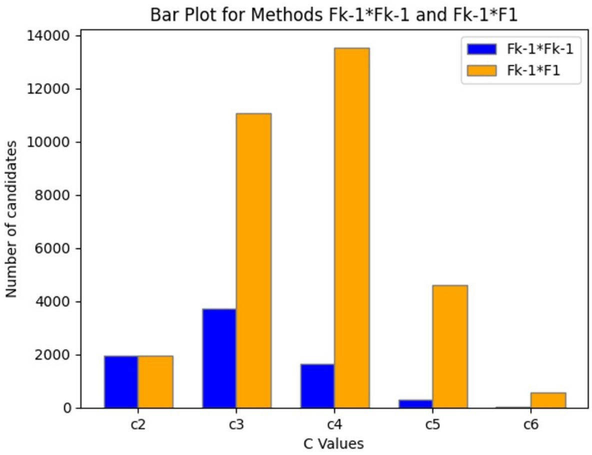

Figure 3: For each non-empty set of candidates generated by each implementation of Apriori, the alternative method gave fewer candidates than the traditional method and is therefore more computationally space efficient.

3. **Using Built-in Function from MLxtend (about 2 minutes):** Leveraging the built-in functions of the MLxtend library proved remarkably efficient, requiring only about 2 minutes. This approach strikes a balance between implementation speed and comprehensive rule generation, making it advantageous for quick prototyping and initial analysis.

Formulas: The support for an itemset I will be calculated using the formula

$$\text{support}(I) = \frac{\text{total number of transactions}}{\text{number of transactions containing } I}$$

Moreover, the confidence for an association rule $X \Rightarrow Y$ will be computed as:

$$\text{confidence}(X \Rightarrow Y) = \frac{\text{support}(X)}{\text{support}(X \cup Y)}$$

Furthermore, the lift for an association rule $X \Rightarrow Y$ will be determined using:

$$\text{lift}(X \Rightarrow Y) = \frac{\text{support}(Y)}{\text{confidence}(X \Rightarrow Y)}$$

2.4 Implementation in Python

The Python programming language served as the foundation for the entire project implementation. We harnessed the power of well-established data manipulation libraries, such as Pandas, to handle dataset manipulation seamlessly. For the implementation of the Apriori algorithm, a specialized library, mlxtend, was employed to leverage its capabilities in association rule mining.

Python's inherent flexibility, coupled with its extensive array of libraries, made it the optimal choice for tackling intricate data mining tasks. This strategic choice ensured the project's efficiency and effectiveness in analyzing the complexities of the 911 emergency dataset.

3 Results

The running times associated with each of the 3 approaches to the Apriori Algorithm were noted as:

- Using the built-in python package called 'mlextend': 1 minute, 12 seconds
- Using the coded Fk-1*Fk-1 method for candidates generation: 45 minutes
- Using the coded Fk-1*F1 method for candidates generation: 1 hour, 50 minutes

We found 191 association rules using our implementation of the Apriori algoritm with Fk-1*Fk-1 candidate generation. We also found 100 association rules with the Fk-1*F1 method. Using the MLxtend Python 3 library, we found 196 association rules. Of the rules found between candidate generation methods for our implementation, we obtained the following similar rules:

→ {'VEHICLE ACCIDENT', 'LOWER MERION'} ⇒ {'weekday'}   with confidence = 0.819

We can interpret the above rule as when a 'VEHICLE ACCIDENT' occurs in the area of 'LOWER MERION', it is likely to happen on a 'weekday'. Further, the corresponding value of the confidence metric indicates the likelihood that the rule is true. In this case, it suggests that in 81.9% of instances where there's a 'VEHICLE ACCIDENT' in 'LOWER MERION', it occurs on a 'weekday'.

→ {'ABINGTON', 'weekday', 'VEHICLE ACCIDENT'} ⇒ {'Traffic'}   with confidence = 0.893

Likewise, from the above rule, we can conclude that when there's a 'VEHICLE ACCIDENT' in 'ABINGTON' and it happens on a 'weekday', it's associated with 'Traffic'.The corresponding high confidence level of 89.4% indicates that when these conditions are met ('ABINGTON', 'weekday', 'VEHICLE ACCIDENT'), it is highly likely that the incident is related to 'Traffic'.

→ {'RESPIRATORY EMERGENCY', 'summer'} ⇒ {'EMS'}   with confidence = 0.99

Similarly, from the above rule, it can be inferred that the instances of 'RESPIRATORY EMERGENCY' occurring in the 'summer' are likely to require 'EMS' (Emergency Medical Services) assistance. Furthermore, the extremely high confidence of 99.97% indicates a strong association between 'RESPIRATORY EMERGENCY' incidents in the 'summer' requiring 'EMS'.

→ {'winter', 'FIRE ALARM'} ⇒ {'Fire'}   with confidence = 0.997

As for the above rule, it can be said that the instances occurring in 'winter' and associated with 'FIRE ALARM' are highly likely to result in a 'Fire'. The confidence value assosicated with this suggests that in almost all instances (99.7%) where 'winter' and 'FIRE ALARM' co-occur, they lead to a 'Fire'. In fact, this aligns perfectly with our notion of fire alarms being set on in case of fire.

→ {'weekday', 'winter', 'VEHICLE ACCIDENT'} ⇒ {'Traffic'}   with confidence = 0.8617

The above rule implies that incidents involving 'winter', 'weekday', and 'VEHICLE ACCIDENT' are often associated with 'Traffic'. As for the confidence value, it indicates that in 86.2% of cases where 'weekday', 'winter', and 'VEHICLE ACCIDENT' coincide, the incident is related to 'Traffic'.

All in all, we broadly came up with 3 kinds of associations, viz., geographical, temporal and weekday.

4 Discussion

The motivation behind coming up with this topic was to figure out a way in which the 911 dispatch team could better anticipate the type of emergency, which in turn would expedite the process of resolving the issues at hand. Further, this type of analysis can prove to be helpful for devising action plans. For example, if a vehicle accident occurs in Lower Merion, given the high probability of it occurring on a weekday, the action plan could look something like allocation of more resources to that particular area on those days. Similarly, knowing that traffic is likely to be heavy in Abington on weekdays during vehicle accidents, dispatchers can advise motorists to take alternate routes.

As far as the scope for improvement is concerned, the core Apriori algorithm can be replaced with a much more efficient and faster FP-Growth algorithm. Several other enhancements could be taken into account, on individual parts of the overall program:

- Advanced data structures like tries can be implemented for more efficient candidate generation.
- For support counting, redundant calculations can be avoided by caching/storing support counts or using dynamic programming techniques.
- Parallel-processing techniques can be utilized to enhance performance, like doing multiple support calculations or candidate generation simultaneously.
- Powerful distributed computing frameworks like Apache Spark could be used to significantly speed up the processing of massive datasets.

For the sake of experiment, we explored Apache Spark framework's python API 'PySpark' and used its RDD-based library 'FPGrowth' to check the performance when using this technique over Apriori algorithm. The results proved to be significantly superior as it took only about 14 seconds to run (while in the Spark environment within Google Colab). This can be considered as a groundwork for further research on optimisation.

5 Author contribution statement

We would like to thank the faculty members for giving us an opportunity to work on this project. The completion of this project shows how we worked together as a team, bringing different skills and knowledge to make it happen.

- Genevieve Mortensen: Proposed idea and found references, worked extensively on the EDA and results interpretation.
- Mahsa Monshizadeh: Did some preprocessing and EDA and Coded the logic behind implementing the Apriori algorithm using 3 approaches and compared them.
- Aditya Shahapure: Explored the techniques which could be used for further optimisation, tried experimenting with Spark framework.

Apart from this, we all worked in a collaborative manner while brainstorming different topics, making the project proposal, presentation and report.

6 References

- https://github.com/tsengupta715/NLP-for-Ambulance-Calls

- https://www.geeksforgeeks.org/project-idea-analysis-emergency-911-calls-using-association-rule-mining/

- https://www.kaggle.com/datasets/mchirico/montcoalert

- https://www.cs.rit.edu/usr/local/pub/GraduateProjects/2165/gkm9983/Report.pdf

- https://data.cincinnati-oh.gov/Safety/Cincinnati-Fire-Incidents-CAD-including-EMS-ALS-BL/vnsz-a3wp/data

- https://www.scaler.com/topics/apriori-algorithm-in-data-mining/

- https://www.analyticsvidhya.com/blog/2020/11/a-must-read-guide-on-how-to-work-with-pyspark-on-google-colab-for-data-scientists/

- https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html