3. Experiments / Proof of concept evaluation

3.1 Preprocessing

The original dataset contained more than 7 million records of crimes from 2001 till present. With crime data, it is important to understand the most recent crime patterns so as to take proactive measures to avoid it. Hence, the analysis was performed on current year 2021 data. We also performed analysis on Pre and Post Covid data that has data from the past 3 years. As part of this, this step involved a one-time effort of segregating the relevant data into individual files so as to avoid loading the whole 7M records into memory.

3.2 Implementation

The execution was performed as a two-part process. First part involved Association Rule Mining and the second was Clustering.

3.2.1 Association Rule Mining

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. It identifies the pattern of a consequent being present when an antecedent is present. In this scenario, association rule mining was performed on two different subsets of the Crime data. The methodology for rule mining was the same for both the subsets.

Preprocessing

Following preprocessing strategies were applied:

1. Drop redundant or unuseful columns: Following columns were dropped from the dataset as they were redundant or provided unuseful data: *ID, Case Number, IUCR, Beat, Updated On, Latitude, Longitude, Location, FBI Code, Block, X Coordinate, Y Coordinate, Date, Year*

2. Add new column: A new formatted *Month* column was added instead of *Date* as to limit the groups formed by grouping on Date. If there are 365 distinct dates, it would create 365 different groups and wouldn't be able to associate with other records.

3. Convert the record data to transaction data: Association analysis is usually performed on transaction data. For this, the record data was converted to tuples. Every value of a column should be uniquely identifiable in a transaction. Directly converting a record to a tuple wouldn't work. For eg: Value *12* in a tuple could be *District*, *Ward* or even *Month*. Hence, to uniquely identify individual column values, column name was prefixed to the value. For example, *12* in *District* was converted to *District-12*, *12* in *Month* was converted to *Month-12*.

Methodology

In an effort to generate the rules, we tried 3 different algorithms for pattern mining from crimes of 2021.

1. Apriori algorithm: Apriori is an algorithm for frequent item set mining and association rule learning over transaction data. It proceeds by identifying the frequent individual items in the dataset and

extending them to larger and larger item sets as long as those item sets appear sufficiently often in the dataset. The preprocessed crime data was passed as an input to the Apriori implementation. It also takes hyperparameter *min_support* as an input. *Support* defines the frequency of the occurrence of an itemset. In this dataset, the attributes of the crime data were widely distributed. Hence, the support was kept to be in range of [0.05, 0.5]. *Confidence* gives the likeliness of the consequent being present, when an antecedent is present. The 2021 data was processed for multiple combinations of support, lift and confidence. Following image shows the number of rules generated for each combination.

Support	Confidence	Lift	Number of rules generated	Time taken
0.05	0.9	1	85	16s
0.05	0.8	1	110	14s
0.10	0.8	1	18	2s
0.10	0.9	1	11	2s
0.20	0.9	1	1	50ms
0.5	0.9	1	0	34ms

Table 1.1

2. Efficient Apriori: It provides an widely used efficient implementation of Apriori as a Python package. The runtime in comparison to plain Apriori was remarkable. The only drawback for this was it didn't return the exact support, confidence and lift of each itemset. This was required in this case to draw out the exact conclusions. The rules generated from this implementation were the same as that of the Apriori. The performance of this algorithm can be seen in table 1.2.

3. FP Growth: FP-growth is an improved version of the Apriori Algorithm which is widely used for frequent pattern mining. It optimizes the Apriori by reducing the disk I/O and subsequently computing power. It generates the FP-tree and uses it to directly generate large itemsets. It uses a recursive divide-and-conquer approach to mine frequent patterns from the large itemsets. The rules generated from this implementation were the same as that of the Apriori. The performance of this algorithm can be seen in table 1.2.

4. Comparison of the algorithms: Table 1.2 shows the comparison between the runtime of the different algorithms. Efficient Apriori was the fastest but lacked some data in the returned rules. FP Growth was approximately 10x faster than plain Apriori.

	Apriori	Efficient Apriori	FP Growth
Support	0.05	0.05	0.05
Confidence	0.9	0.9	0.9
Lift	1	1	1
Time taken	16s	0.84s	1.8s

Table 1.2

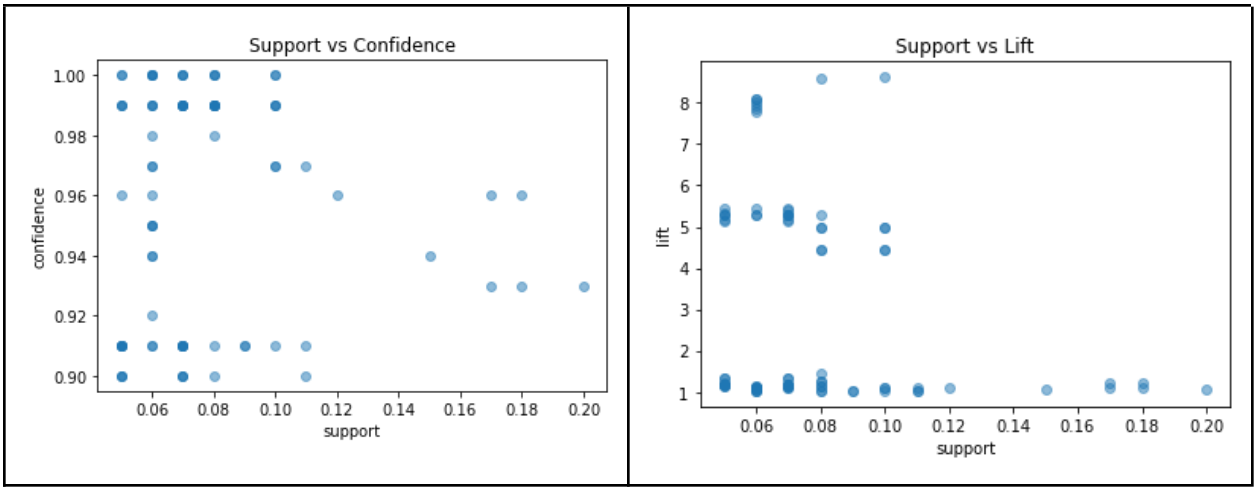
Analysis

Implementation was performed on 2 different types of data. First, the data of 2021 crimes was analysed to understand the recent patterns of crime. Second, the data was divided into 2 parts: Pre-Covid and Post-Covid. The intention was to find the difference in the crime patterns at both periods.

A. 2021 crime data

Dataset : Crime data of **01/01/2021 - 11/10/2021** - Total **158084 records**

Graphs: Fig 1.2 shows the distribution of Support vs Confidence and Support vs Lift.



Frequent patterns: In the Apriori analysis, 85 frequent patterns were generated for support 0.05, confidence 0.9 and lift 1. Some of the samples of the rules generated are shown in fig 1.2. Out of 85 rules generated, around 10 rules were meaningful(highlighted in blue), Rest others provided relations which were obvious.

	A	B	C	D	E
1	antecedents	consequents	support	confidence	lift
20	'Description-DOMESTIC BATTERY SIMPLE', 'Arrest-False'	'Domestic-True'	0.08	0.99	4.43
21	'Description-DOMESTIC BATTERY SIMPLE', 'Primary Type-BATTERY', 'Arrest-Fal	'Domestic-True'	0.08	0.99	4.43
22	'Domestic-False', 'Description-OVER $500'	'Primary Type-THEFT', 'Arrest-False'	0.05	0.99	5.44
23	'Domestic-False', 'Description-OVER $500'	'Arrest-False'	0.05	0.99	1.13
24	'Domestic-False', 'Primary Type-THEFT', 'Description-OVER $500'	'Arrest-False'	0.05	0.99	1.13
25	'Description-OVER $500'	'Primary Type-THEFT', 'Arrest-False'	0.06	0.99	5.43
26	'Description-OVER $500'	'Arrest-False'	0.06	0.99	1.13
27	'Description-OVER $500', 'Primary Type-THEFT'	'Arrest-False'	0.06	0.99	1.13
28	'Primary Type-DECEPTIVE PRACTICE'	'Arrest-False'	0.08	0.99	1.13
29	'Domestic-False', 'Primary Type-DECEPTIVE PRACTICE'	'Arrest-False'	0.08	0.99	1.13
30	'Domestic-False', 'Description-$500 AND UNDER'	'Primary Type-THEFT', 'Arrest-False'	0.07	0.99	5.43
31	'Domestic-False', 'Description-$500 AND UNDER'	'Arrest-False'	0.07	0.99	1.13
32	'Domestic-False', 'Description-$500 AND UNDER', 'Primary Type-THEFT'	'Arrest-False'	0.07	0.99	1.13
33	'Description-$500 AND UNDER'	'Primary Type-THEFT', 'Arrest-False'	0.07	0.99	5.42
34	'Description-$500 AND UNDER'	'Arrest-False'	0.07	0.99	1.12
35	'Description-$500 AND UNDER', 'Primary Type-THEFT'	'Arrest-False'	0.07	0.99	1.12

Fig 1.2

Result: After analysing all the rules, some of the interesting conclusions that can be drawn are as follows:

- In districts 4 and 8, no arrests have been made for more than 90% of the crimes.

- The months of August and September contribute to 20% of the annual crime. And yet no arrests were made for more than 90% of the cases.
- In following crimes, arrests rates are very low:
    - Theft over $500 - 1%
    - Theft under $500 - 1%
    - Deceptive practice - 1%
    - Criminal Damage to vehicle - 2%
    - Simple Assault - 6%
    - Crimes committed at residences, apartments - 6%, contributes to 25% of the crimes.

B. Pre and Post Covid data

3.2.2 Clustering

4 Discussion & Conclusions

4.1 Decisions made

Association rule mining led to believe that arrest rates are very low in districts 8 and 9. It is also low for the most common occurring crimes like Theft, Damages to vehicle, Crimes committed at residences and apartments.

4.2 Difficulties faced

The initial file size was huge(1.8 GB) and was killing the kernel. Splitting the file according to the required helped to smoothen the further analysis process.

4.3 Things that worked

Association rule mining led to identification of meaningful patterns related to arrests in some specific locations, months of the year and the crimes.

4.4 Things that didn’t work well

There were expectations that Pre Covid and Post Covid analysis would provide some interesting patterns around the different types of crimes.

4.5 Conclusion

5 Project Plan / Task Distribution

Sandesh Gupta

- Generate all data required for association rule mining:
  - last 2 years
  - last 4 years
  - 2021 data
  - Pre covid and post covid
- Pre-process data for ARM
  - Decide relevant columns
  - Drop irrelevant columns
  - Format columns
  - Convert records to transaction data
- Research on how to run Apriori, Efficient Apriori, FP Growth
- Run Association analysis on 2021 data
  - Run the algorithms, format itemsets and write to CSV
- Analyse and formulate the rules post Association Rule Mining. Interpret the rules and derive conclusions.

6 References

https://en.wikipedia.org/wiki/Apriori_algorithm
http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/
https://en.wikipedia.org/wiki/Association_rule_learning
https://medium.com/analytics-vidhya/association-analysis-in-python-2b955d0180c
https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2/data
https://pypi.org/project/efficient-apriori/
https://www.geeksforgeeks.org/implementing-apriori-algorithm-in-python/
http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/