Cambridge Crime

Tinotenda Muchenje

CSIT 558

Descriptive Mining

Description of Dataset

Introduction:

This presentation is on the analysis of crime data in the city of Cambridge from 2009 to 2016. In this study, we will explore a comprehensive dataset obtained from data.world/data-society/cambridge-crime-data-2009-2016. This dataset consists of 56,014 rows and 7 columns, providing valuable information about crimes that occurred in Cambridge during the specified period. The dataset encompasses various aspects, including the type of crime, neighborhood, and time of occurrence, among others.

	File Number	Date of Report	Crime Date Time	Crime	Reporting Area	Neighborhood	Location
0	2009-00002	01/01/2009 12:39:00 AM	1/1/09 0:39	Simple Assault	504.0	Cambridgeport	400 Massachusetts Avenue, Cambridge, MA
1	2009-00003	01/01/2009 01:34:00 AM	1/1/09 1:34	Simple Assault	610.0	Mid-Cambridge	200 HAMPSHIRE STREET, Cambridge, MA
2	2009-00004	01/01/2009 01:43:00 AM	01/01/2009 02:20 - 02:35	Aggravated Assault	708.0	Riverside	DUNSTER STREET & MOUNT AUBURN STREET, Cambridg...

Objective:

The primary objective of this project is to conduct descriptive data mining on the crime dataset to develop a comprehensive understanding of the patterns within the data. Specifically, we aim to explore the relationships and patterns between crime types, neighborhoods, and crime occurrence times. By analyzing these patterns, we seek to gain insights into the occurrences of crime events in the city of Cambridge during the specified timeframe. Understanding these patterns can provide valuable information for law enforcement agencies, city planners, and policymakers in implementing targeted strategies to prevent and address crime. Through our descriptive data mining analysis, we hope to uncover meaningful correlations and trends that contribute to a deeper understanding of crime dynamics in Cambridge.

Data Preprocessing

Data Cleaning:

The following steps were carried out while performing data cleaning:

1. Eliminating empty values
2. Eliminating duplicated values
3. Converting the Crime date column into the correct date format

```
# Counting the number of missing values
df.isna().sum()
df.dropna(inplace=True)
# check for duplicated rows and drop
df.duplicated().sum()
df = df.drop_duplicates()
```

The following snippet displays the resulting data frame after the completion of data cleaning and the conversion of categorical columns into numeric representations.

	File Number	Date of Report	Crime Date Time	Crime	Reporting Area	Neighborhood	Location	Crime Encoded	Neighborhood Encoded	Crime Year	Crime Month	Crime Day of Week	Crime Time
0	2009-00002	01/01/2009 12:39:00 AM	2009-01-01 00:39:00	Simple Assault	504.0	Cambridgeport	400 Massachusetts Avenue, Cambridge, MA	43	2	2009	1	1	0
1	2009-00003	01/01/2009 01:34:00 AM	2009-01-01 01:34:00	Simple Assault	610.0	Mid-Cambridge	200 HAMPSHIRE STREET, Cambridge, MA	43	7	2009	1	1	1
2	2009-00004	01/01/2009 01:43:00 AM	2009-01-01 02:20:00	Aggravated Assault	708.0	Riverside	DUNSTER STREET & MOUNT AUBURN STREET, Cambridg...	2	10	2009	1	1	2

Transforming Categorical Values to Numerical Values:

To facilitate our descriptive analysis, we needed to convert certain categorical values into numerical values. To accomplish this, we utilized Label Encoding, a Python library designed for this specific transformation task. Each categorical value was assigned a numerical representation. Encoding was carried out on the Crime Type and Neighborhood columns.

Example Results:

Crime Type: Simple Assault has a resulting Encoded Value: 43

Neighborhood: Cambridgeport, Encoded Value: 2

```
# label encoding
le = LabelEncoder()
df['Crime Encoded'] = le.fit_transform(df['Crime'])
df['Neighborhood Encoded'] = le.fit_transform(df['Neighborhood'])
```

Dimensionality Reduction

Dimensionality Reduction:
After performing data preprocessing and conversion, an additional step of dimensionality reduction was carried out. Dimensionality reduction plays a crucial role in the descriptive mining phase of the presentation. Dimensionality reduction is a valuable technique in the descriptive mining phase as it enables efficient analysis, facilitates visualization, mitigates the curse of dimensionality, enhances interpretability, and reduces noise in the data, ultimately leading to more meaningful and reliable insights.

Dimensionality Reduction

```
# Dimensionality Reduction of dimensions that are not informative for my specific analysis and can be excluded.
# Remove the columns 'File Number', 'Date of Report', and 'Crime Date Time'
df = df.drop(['File Number', 'Date of Report', 'Crime Date Time', 'Reporting Area'], axis=1)
df
```

	Crime	Neighborhood	Location	Crime Encoded	Neighborhood Encoded	Crime Year	Crime Month	Crime Day of Week	Crime Time
0	Simple Assault	Cambridgeport	400 Massachusetts Avenue, Cambridge, MA	43	2	2009	1	1	0
1	Simple Assault	Mid-Cambridge	200 HAMPSHIRE STREET, Cambridge, MA	43	7	2009	1	1	1
2	Aggravated Assault	Riverside	DUNSTER STREET & MOUNT AUBURN STREET, Cambridg...	2	10	2009	1	1	2

The provided screenshot displays the reduced dimensionality dataframe. Following data preprocessing, label encoding, and dimensionality reduction, the dataset now consists of 55,927 rows and 9 columns.

Correlation Analysis

The purpose of performing a correlation analysis on these variables in the context of descriptive analysis is to explore potential relationships or dependencies between different aspects of the crime dataset. Correlation analysis helps us understand if there are any patterns or associations between variables, which can provide valuable insights for further analysis.

CORRELATION MATRIX

```
# Subset the DataFrame to include only the numerical columns
numerical_data = df[['Crime Encoded', 'Crime Time', 'Neighborhood Encoded']]
# Compute the correlation matrix
correlation_matrix = numerical_data.corr()
# Print the correlation matrix
print(correlation_matrix)
```

RESULTS

	Crime Encoded	Crime Time	Neighborhood Encoded
Crime Encoded	1.000000	0.016599	-0.034873
Crime Time	0.016599	1.000000	-0.005754
Neighborhood Encoded	-0.034873	-0.005754	1.000000

The correlation analysis results for the variables Crime Type, Crime Time, and Neighborhood are as follows:

1. **Crime and Time**: The correlation coefficient between these two variables is 0.016599, which indicates a very weak positive correlation. This suggests that there is a slight tendency for certain types of crimes to occur at specific times, but the relationship is not strong.
2. **Crime and Neighborhood**: The correlation coefficient between these two variables is -0.034873, indicating a very weak negative correlation. This suggests a minimal association between the type of crime and the specific neighborhood where it occurs.
3. **Crime Time and Neighborhood**: The correlation coefficient between these variables is -0.005754, indicating an extremely weak negative correlation. This implies that there is almost no relationship between the time of crime and the specific neighborhood.

In this case, the correlation analysis indicates that there is no significant correlation between the variables Crime Type, Crime Time, and Neighborhood. This suggests that these variables might be relatively independent of each other, and the occurrence of a specific crime type is not strongly influenced by the time of the crime or the neighborhood where it takes place.

Correlation-Pairplot Analysis

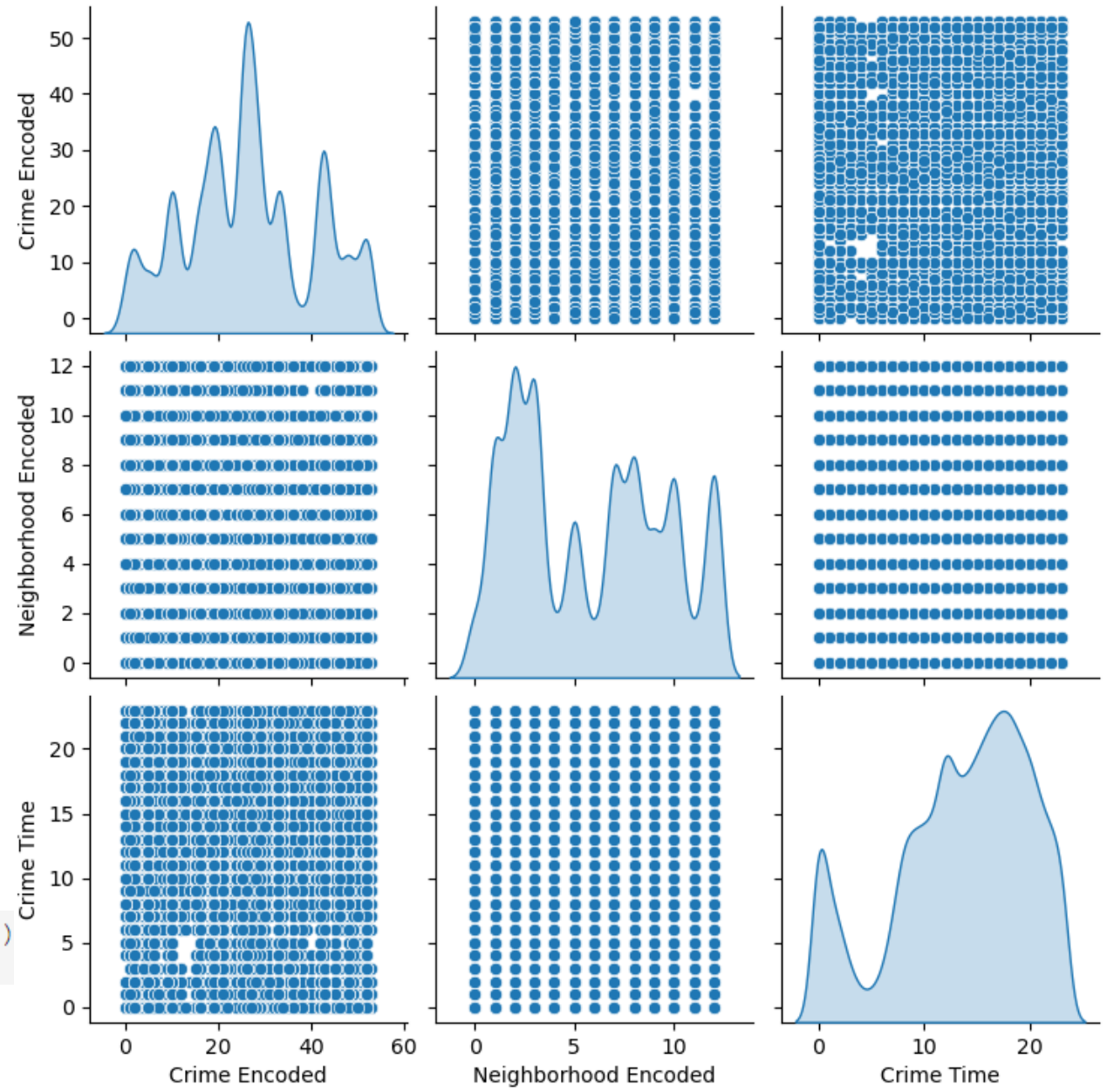
While the correlation coefficient results, may not indicate significant correlations mathematically, I used the pair plot which allows for visual exploration and examination of the relationships between variables. Based on the pair plot, it does not show any significant patterns, outliers, or correlations among the variables as well.

Lack of patterns or correlations: The data points for the crime types, crime time, and neighborhoods appear to be distributed evenly across the plots without forming any distinct patterns or clusters. This suggests that the crime events are randomly distributed and not significantly influenced by time or location factors.

No outliers: There are no obvious outliers or extreme values in the dataset, as the data points appear to be spread evenly throughout the plots. This implies that the crime events have consistent features and are not strongly influenced by a small number of extreme cases.

Equal distribution: The crime events seem to be equally distributed across the variables, meaning the crime types, crime time, and neighborhoods have no significant influence on each other. This may indicate that crime occurrences in the dataset are random and independent events.

```
sns.pairplot(df[['Crime Encoded', 'Neighborhood Encoded', 'Crime Time']], diag_kind='kde')
plt.show()
```

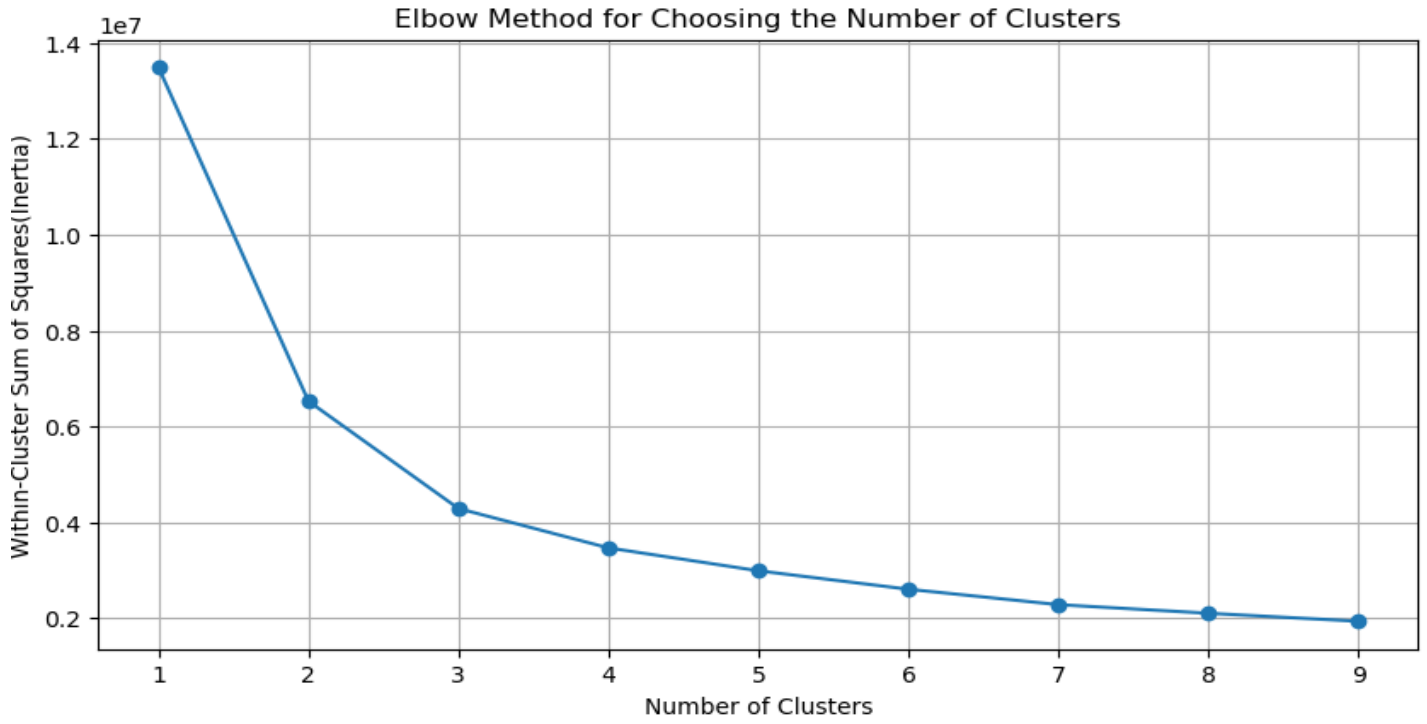


K-Means Clustering

As part of this analysis, the K-means clustering technique was used. By grouping similar crimes, it becomes possible to identify patterns and understand the characteristics of different types of criminal activities. The results can provide meaningful insights, for instance, it may reveal hotspots or high-crime areas where specific types of crimes are concentrated.

Step 1: Determining the number of optimum number of clusters using the Elbow method

```
def optimise_k_means(data, max_k):
    means = []
    inertias = []

    for k in range(1, max_k):
        kmeans = KMeans(n_clusters=k)
        kmeans.fit(data)
        means.append(k)
        inertias.append(kmeans.inertia_)

    # Generate the elbow plot
    fig = plt.subplots(figsize=(10,5))
    plt.plot(means, inertias, 'o-')
    plt.title('Elbow Method for Choosing the Number of Clusters')
    plt.xlabel('Number of Clusters')
    plt.ylabel('Within-Cluster Sum of Squares(Inertia)')
    plt.grid(True)
    plt.show()
```



To determine the optimal number of clusters, I utilized the columns "Crime type," "Neighborhood," and "Crime time“, these variables provide crucial information for the clustering process. Looking at the plot depicted above, it is evident that as the number of clusters increases, the inertia (which represents the sum of squared distances to the nearest cluster center) decreases. Notably, there is a significant change in inertia and slope starting from cluster 2. Based on the diagram, it appears advisable to utilize 3-4 clusters since the data points from cluster 4 are almost within the same inertia.

K-Means Clustering Cntd…

Step 2: Data fitting

Variables: Crime Type, Neighborhood and Crime Time

```
for k in range(1, 6):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(df[['Crime Encoded', 'Crime Time', 'Neighborhood Encoded']])
    df[f'KMeans_{k}'] = kmeans.labels_
```

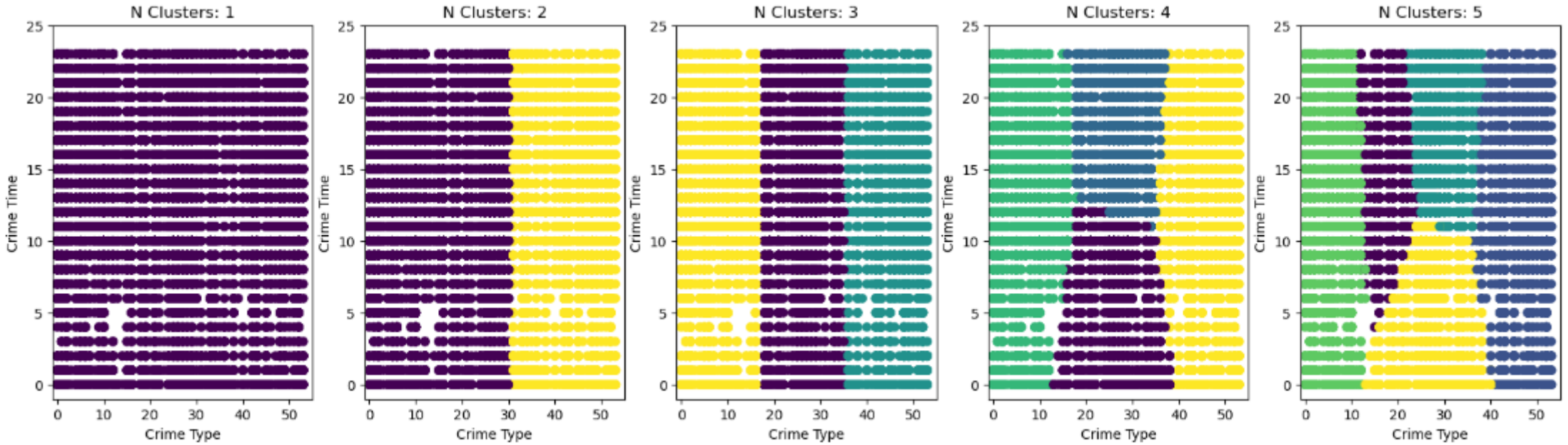
```
fig, axs = plt.subplots(nrows=1, ncols=5, figsize=(20,5))

for i, ax in enumerate(fig.axes, start=1):
    ax.scatter(x=df['Crime Encoded'], y=df['Crime Time'], c=df[f'KMeans_{i}'])
    ax.set_xlim(-1, 55)
    ax.set_ylim(-1, 25)
    ax.set_xlabel('Crime Encoded')
    ax.set_ylabel('Crime Time')
    ax.set_title(f'N Clusters: {i}')
```

Avoiding **Underfitting and Overfitting**: Underfitting occurs when there are too few clusters, resulting in oversimplification and loss of important information. Overfitting, on the other hand, occurs when there are too many clusters, leading to excessive fragmentation and capturing noise or irrelevant patterns. The Elbow method helps find the optimal number of clusters that strikes a balance between these two extremes, maximizing the model's generalizability and effectiveness.

Three experiments were conducted to determine the optimal clustering and fitting of the data.

K-Means Clustering Cntd…

Experiment 1: Clustering Crime Type vs Crime Time

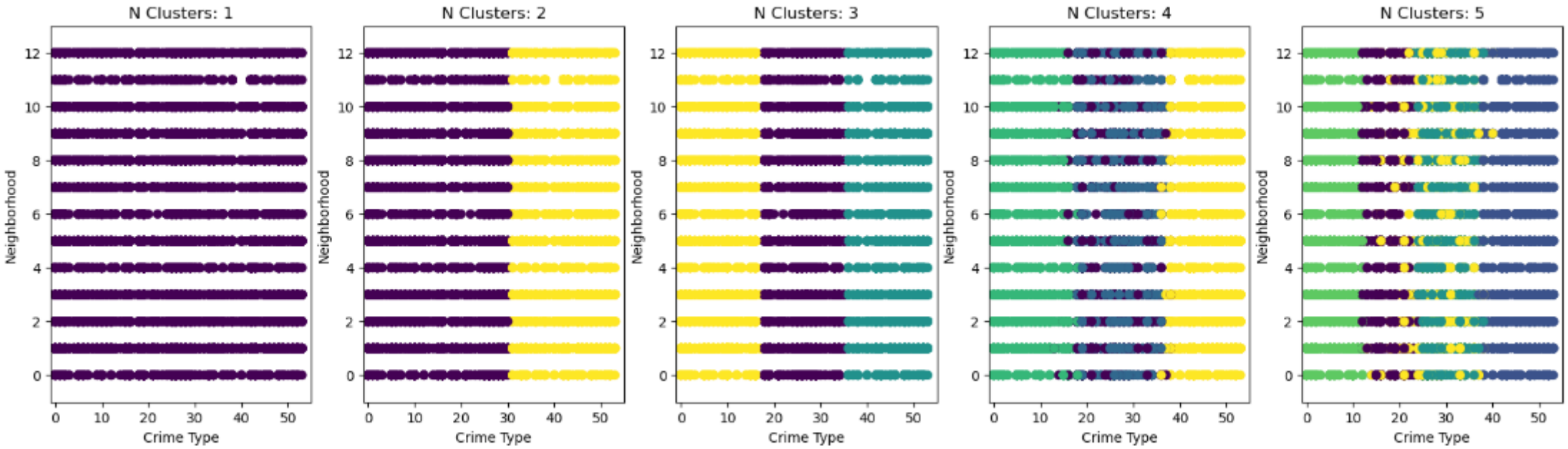


In Experiment 1, K-Means clustering was applied to the Crime Type and Crime Time data. The clustering analysis involved testing different numbers of clusters, ranging from one to five, to determine the optimal number of clusters that yield informative and meaningful results. Based on the obtained graphs, it was found that considering four clusters would be the most suitable choice.

The results of the analysis indicate that crimes encoded between 0 and 10 tend to occur consistently throughout the entire day. On the other hand, crimes encoded between 10 and 40 primarily take place from midnight to midday, while some other crimes within this range occur from midday to midnight. These findings provide insights into the temporal patterns of crime occurrences within the specified encoding ranges.

K-Means Clustering Cntd…

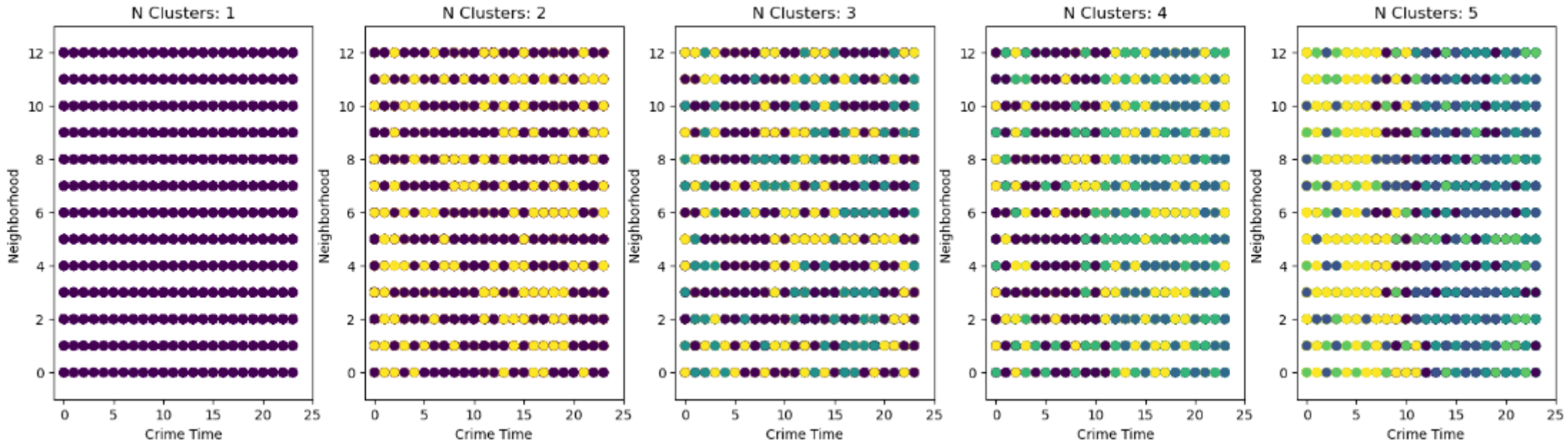
Experiment 2: Clustering Crime Type vs Neighborhood



Based on the results obtained when considering four clusters or all clusters, it was observed that the data points representing crime types and neighborhoods were evenly distributed across the plots. No discernible patterns or distinct clusters were formed, indicating that the occurrence of crime events appears to be randomly distributed and not strongly influenced by location factors. This suggests that the occurrence of crimes does not exhibit any significant spatial correlation or dependency on specific neighborhoods.

K-Means Clustering Cntd…

Experiment 3: Clustering Crime Time vs Neighborhood



Results:

The analysis revealed that the data points representing crime time and neighborhoods exhibited an even distribution across the plots, with no discernible patterns or clusters. This finding indicates that crime events are randomly distributed and not significantly influenced by location factors or the timing of occurrences.

Association Rule Mining - Apriori

Conducting Apriori analysis with Crime Type, Crime Time, and Neighborhood can bring several significant insights, by analyzing association rules.
Expectations: By performing this experiment, we unlock valuable insights regarding co-occurring crime patterns, hotspots, modus operandi, and associations between variables. These findings can inform crime prevention strategies, investigations, policy-making, and resource allocation decisions, ultimately contributing to the improvement of public safety and security.

```
#Crime Type and Neighborhood:
#Investigate associations between crime types and specific neighborhoods.
#This can provide insights into crime patterns within different neighborhoods and potentially uncover
#any localized trends.
itemset = list(df[['Crime Encoded', 'Neighborhood Encoded']].values)

# Apply the apriori algorithm to the itemset
association_rules = apriori(itemset, min_support=0.001, min_confidence=0.2)
association_results = list(association_rules)

# Calculate the confidence and for each rule
for item in association_results:

    #First index of the inner list
    # Contains base item and add item
    pair = item[0]
    items = [x for x in pair]
    print('Rule:' + str(items[0]) + ' -> ' + str(items[1]))

    #Second index of the inner list
    print('Support: ' + str(item[1]))

    #Third index of the inner list located at 0th of the third index of the inner list
    print('Confidence: ' + str(item[2][0][2]))
    print('Lift: ' + str(item[2][0][3]))
    print('=====================================================')
```

Three experiments utilizing the Apriori algorithm were conducted to ascertain and analyze patterns and associations within the data. Through these experiments, the Apriori algorithm was applied with different parameter settings and thresholds to explore various aspects of the data. The goal was to find the optimal configuration that would yield the most significant and informative patterns. By analyzing the resulting frequent itemset and association rules, valuable insights regarding item co-occurrence, dependencies, and correlations were extracted.

Apriori Analysis Cntd...

Association rules were generated based on the **minimum support of 0.001** and **minimum confidence of 0.2.** Each rule consists of an antecedent (left-hand side) and a consequent (right-hand side), along with their associated support and confidence values.

Experiment 1: Association Rule Mining Crime vs Neighborhood

```
Rule:42 -> 3
Support: 0.020508877644071735
Confidence: 0.4114060258249641
Lift: 2.9688651362984215
==================================================
Rule:3 -> 52
Support: 0.010013052729450892
Confidence: 0.2177293934681182
Lift: 1.5712195856117994
==================================================
```

Association 1: Shoplifting(42) -> East Cambridge(3) with Support: 0.02 and Confidence: 0.41. This rule suggests that when a crime of type "Shoplifting" occurs, there is a 0.02 support, indicating that it is present in 2% of the occurrences. The confidence of 0.41 implies that 41% of instances where "Shoplifting" occurs are also associated with the neighborhood "East Cambridge".

Association 2: East Cambridge(3) -> Warrant Arrest(52) with Support: 0.01 and Confidence: 0.21. This rule indicates that when the neighborhood is "East Cambridge", there is a 0.01 support, meaning it is present in 1% of the crime occurrences. The confidence of 0.21 suggests that 21% of instances where the neighborhood is "East Cambridge" is also associated with the crime type "Warrant Arrest".

Experiment 2: Association Rule Mining Crime vs Crime Time

```
Rule:17 -> 47
Support: 0.0012695120424839523
Confidence: 0.27413127413127414
Lift: 4.0612820578383495
==================================================
```

Association 1: 1700hrs(17) -> Taxi Violation(47) with Support: 0.001 and Confidence: 0.27.This association rule indicates that there is a relationship between the occurrence of crimes at 17:00 (5:00 PM) and the specific crime type "Taxi Violation". The support value of 0.001 suggests that this rule appears in 0.1% of the instances in the dataset. The confidence value of 0.27 indicates that when a crime occurs at 17:00, there is a 27% likelihood that it will be classified as "Taxi Violation".

Apriori Analysis Cntd...

Association rules were generated based on the **minimum support of 0.01** and **minimum confidence of 0.2.** Each rule consists of an antecedent (left-hand side) and a consequent (right-hand side), along with their associated support and confidence values.

Experiment 3: Association Rule Mining Crime vs Crime Time vs Neighborhood

```
Rule:42 -> 3
Support: 0.020651921254492462
Confidence: 0.4142754662840746
Lift: 2.78375393522401
==================================================
Rule:27 -> 12
Support: 0.010263379047687163
Confidence: 0.25241864555848725
Lift: 1.5037300372975624
==================================================
```

Association 1: Shoplifting(42) -> East Cambridge(3) with Support: 0.02 and Confidence: 0.41. This rule suggests that when a crime of type "Shoplifting" occurs, there is a 0.02 support, indicating that it is present in 2% of the occurrences. The confidence of 0.41 implies that 41% of instances where "Shoplifting" occurs are also associated with the neighborhood "East Cambridge".

Association 2: Larceny from Person(27) -> West Cambridge(12) with Support: 0.010 and Confidence: 0.25 This association rule suggests a relationship between the crime type "Larceny from Person" and the neighborhood "West Cambridge". The support value of 0.010 indicates that this rule appears in 1% of the instances in the dataset. The confidence value of 0.25 indicates that when a crime of type "Larceny from Person" occurs, there is a 25% likelihood that it will be associated with the neighborhood "West Cambridge".

Support: measures proportion where transactions contain both the antecedent and consequents
Confidence: measures the reliability of the rule.
Lift: measures how much more likely the consequent is given the antecedent, compared to without

Geospatial Analysis

In this section, I conducted geospatial analysis utilizing the crime location data. The significance of this experiment lies in its ability to provide valuable insights and understanding of the spatial distribution of crime incidents. Geospatial analysis allows us to identify areas with a high concentration of crime incidents, commonly referred to as crime hotspots, by pinpointing these hotspots on a map

Step 1: Geocode Addresses to Latitude and Longitude for Geo Analysis

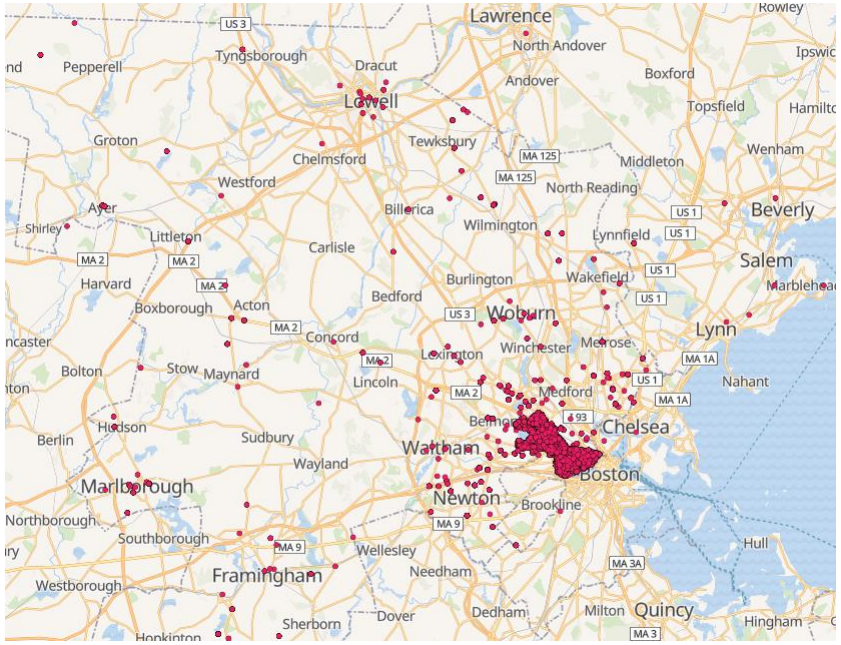
```
locator = Nominatim(user_agent='spatialthoughts', timeout=10)
geocode = RateLimiter(locator.geocode, min_delay_seconds=1)

address_df_pd = address_df.copy()
address_df_pd['location'] = address_df_pd['Full_Address'].progress_apply(geocode)
address_df_pd
```

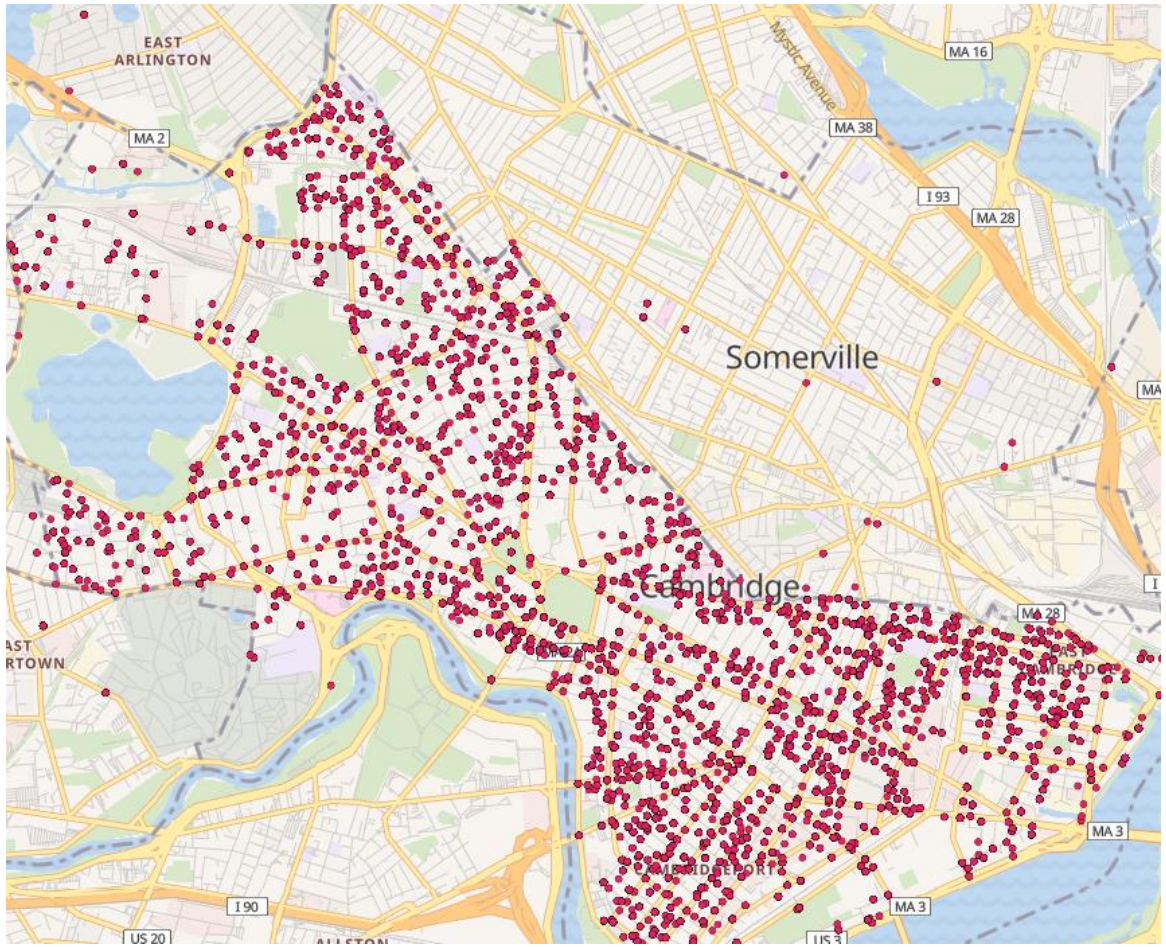
	Address	Latitude	Longitude
0	400 Massachusetts Avenue, Cambridge, MA	42.363310	-71.100180
1	200 HAMPSHIRE STREET, Cambridge, MA	42.373015	-71.100164
2	DUNSTER STREET & MOUNT AUBURN STREET, Cambridg...	42.371930	-71.119530

```
#We create a GeoDataFrame from the latitude and longitude columns.
geometry = gpd.points_from_xy(geocoded_address.Longitude, df.Latitude)
geocoded_address = gpd.GeoDataFrame(df, crs='EPSG:4326', geometry=geometry)
```

Geospatial Analysis Cntd...

RESULTS





The geospatial analysis conducted on the map images reveals important insights about the distribution of crimes in Cambridge. One significant finding is that there is no central place or neighborhood where crimes are concentrated. Instead, the analysis indicates that crimes are fairly distributed throughout the city. The absence of a concentrated crime hotspot suggests that there is no specific area that requires targeted intervention or heightened security measures. Rather the results drawn from the geospatial analysis emphasize the need for a comprehensive and holistic approach to crime prevention in Cambridge.

Conclusion

In conclusion, this project aimed to uncover the most recurring crimes and establish relationships between crime types, crime time, and the corresponding neighborhoods in which these crimes occurred. To achieve this objective, a comprehensive approach was adopted, incorporating descriptive analysis, KMeans Clustering, the Apriori algorithm, and geospatial analysis techniques.

The application of KMeans Clustering enabled the identification of meaningful clusters or groups within the dataset. This clustering analysis helped to uncover potential associations or similarities between different crimes, crime times, and neighborhoods. By grouping similar instances together, we were able to discern underlying patterns and relationships that may have been otherwise overlooked.

The implementation of the Apriori algorithm facilitated the discovery of association rules, highlighting the connections between crime types, crime times, and neighborhoods. These association rules provided valuable insights into the likelihood of certain crimes occurring in specific time periods or locations. The minimum support and confidence thresholds allowed us to focus on significant associations, ensuring the reliability and relevance of the generated rules.

Finally, geospatial analysis played a pivotal role in understanding the spatial distribution of crimes across neighborhoods in the study area. The examination of the map images revealed that crimes were fairly distributed throughout Cambridge, without any concentrated crime hotspots. This finding has significant implications for law enforcement agencies and city planners, as it suggests the need for comprehensive city-wide crime prevention strategies rather than isolated interventions in specific areas.

In summary, this project successfully achieved its objective of identifying recurring crimes and establishing relationships between crime types, crime time, and neighborhoods. The combined application of descriptive analysis, KMeans Clustering, the Apriori algorithm, and geospatial analysis provided a comprehensive and multi-faceted approach to understanding and analyzing the dataset. The insights gained from this project can inform evidence-based decision-making, resource allocation, and the development of effective crime prevention strategies in the study area.

Python Libraries & References

References

Bhat, H. V. (2023, October 09). *What is Geospatial Data and How to Implemement*. Retrieved from analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2023/02/implementing-geospatial-data-analysis-in-data-science-techniques-challenges-trends-and-best-practices/

Data World. (2024). *Cambridge Crime Data 2009-2016*. Retrieved from data.world: https://data.world/data-society/cambridge-crime-data-2009-2016

Intellipaat. (2024, January 4). *Data Science - Apriori Algorithm in Python- Market Basket Analysis*. Retrieved from intellipaat.com: https://intellipaat.com/blog/data-science-apriori-algorithm/

Luke, T. (2022, Apr 10). *Create a K-Means Clustering Algorithm from Scratch in Python*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/create-your-own-k-means-clustering-algorithm-in-python-d7d4c9077670


```python
# importing important libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import mlxtend.preprocessing as preprocessing
from sklearn.preprocessing import StandardScaler, MinMaxScaler, OneHotEncoder, LabelEncoder
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.metrics import silhouette_score
from sklearn.feature_selection import SelectKBest, chi2
from mlxtend.frequent_patterns import apriori, fpgrowth, association_rules
from mlxtend.preprocessing import TransactionEncoder
from statsmodels.tsa.seasonal import seasonal_decompose
from sklearn.ensemble import IsolationForest
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from apyori import apriori
from gsppy.gsp import GSP
from pyspark.ml.fpm import FPGrowth
from pyspark.sql.functions import col
from pyspark import SparkConf, SparkContext
from pyspark.mllib.fpm import FPGrowth
import geopandas as gpd
import leafmap.foliumap as leafmap
from geopy.geocoders import Nominatim
from geopy.extra.rate_limiter import RateLimiter
```

Thank You

24Slides