BA820 - Project Milestone 1

Cover Page

- **Project Title:** San Francisco Building Complaints Segmentation
- **Section and Team Number:** B1 Team7
- **Members:** Yu-Hsiang (Rick) Wang, Ching-Hsuan (Shawn) Lin, Kuang-Ching (Amanda) Ting, Ming-Hua (Jasmine) Tsai
- **Date:** Feb 19, 2025

Proposal

Problem statement

Managing complaints and inspections is resource-intensive, yet manual classification is inefficient. Complaints lack prioritization and grouping, causing delays. We will use unsupervised machine learning (e.g., K-Means, Hierarchical Clustering) to identify patterns, classify complaints, and improve prioritization. This approach enhances response times, optimizes resource allocation, and streamlines issue resolution by efficiently assigning complaints to the right departments. By uncovering hidden patterns in unstructured data, it reduces manual effort and improves decision-making.

Key Challenges

In this project, the raw data is relatively messy, requiring extensive preprocessing. This includes tasks such as filling in missing values, checking data types, and encoding categorical data.

Exploratory Data Analysis (EDA)

- **Data Source:** San Francisco Government's Open Data (https://reurl.cc/eGKE5K)

- **Summary Statistics:** Based on the mean, median, and standard deviation, the Street Number data is highly dispersed, while the ZIP Code values are more concentrated. The Supervisor District distribution appears relatively balanced across different areas.

	Street Number	ZIP Code	Supervisor District
count	310076.000000	309933.000000	309546.000000
mean	1198.582122	94115.473493	5.786946
std	1162.789639	9.120995	3.076982
min	0.000000	94102.000000	1.000000
25%	320.000000	94109.000000	3.000000
50%	832.000000	94114.000000	5.000000
75%	1719.000000	94122.000000	9.000000
max	9490.000000	94158.000000	11.000000

- **Outliers and Patterns:** Since most of our data consists of categorical variables, we have not yet isolated outliers. In the next step, we will apply dimensionality reduction to separate them and analyze the differences.

- **Preprocessing:** Time-related fields like data_loaded_at and data_as_of were unnecessary and could introduce noise. Closed Date is due to unresolved cases, so we imputed them, while excessive location details were dropped in favor of ZIP Code.

- **Observation:**
  1. Mission and Tenderloin have the highest complaint volumes, Bayview Hunters Point and Sunset/Parkside show more dispersed complaints.
  2. Complaints steadily increased from the 1990s to the late 2010s, possibly and a sharp decline in 2023–2024 suggests new regulations.
  3. High-density complaint clusters suggest areas needing priority inspections and targeted interventions.

Analysis Plan

Tasks

- **Data Preprocessing**: Remove complaint descriptions to build an initial model with structured features (e.g., complaint type, location, time). Clean the text data later by removing stop words, punctuation, and special characters, converting to lowercase, and applying lemmatization.
- **Initial Clustering:** Implement a K-Means model on structured features for a quick segmentation of complaints and to assess severity-level distinctions.
- **Clustering Comparison:** Compare K-Means results with Hierarchical Clustering to determine which method yields more interpretable and actionable groupings.
- **Text Mining & Topic Modeling:** Reintroduce complaint descriptions and apply Latent Dirichlet Allocation (LDA) to uncover underlying topics in the text data.
- **Dimensionality Reduction & Visualization:** Use PCA or t-SNE to visualize and compare clustering results across different algorithms and incorporate text-derived features.

Analysis Methods

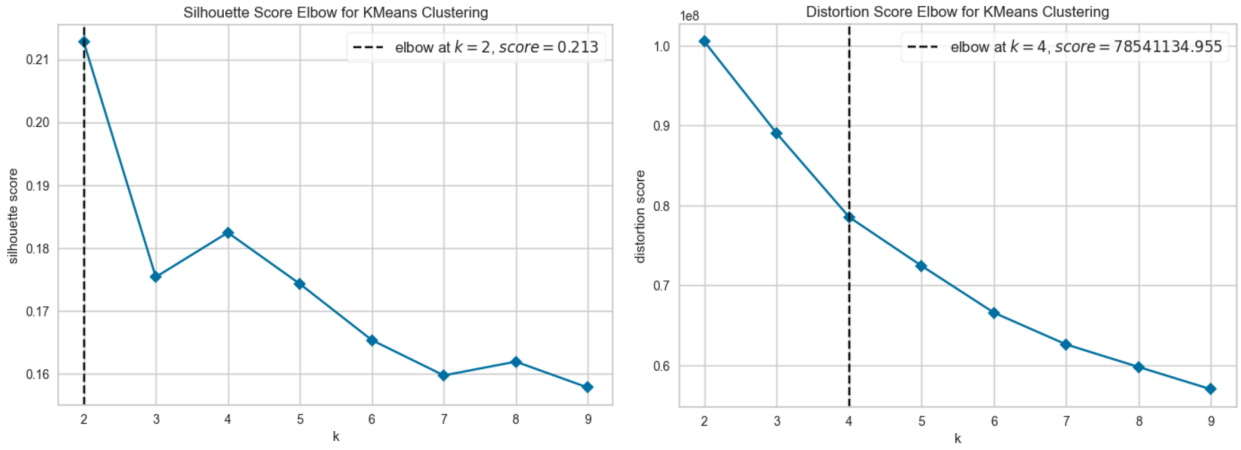
- **K-Means & Hierarchical Clustering**: Two clustering approaches for grouping complaints based on severity and other factors.
- **Text Preprocessing**: Standard techniques (stop word removal, lowercasing, lemmatization) to clean and normalize textual data.
- **Topic Modeling (LDA)**: Unsupervised method to reveal hidden themes in complaint descriptions.
- **Dimensionality Reduction (PCA, t-SNE)**: Techniques to project high-dimensional data into a lower-dimensional space for visualization and to identify separable clusters.

Justifications

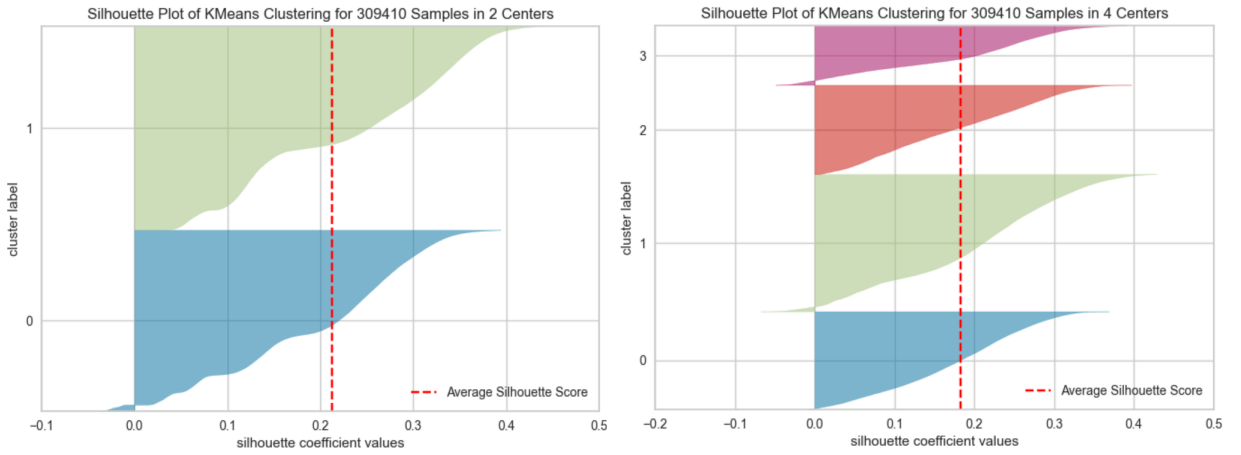
- **Clustering Methods**: K-Means offers quick, initial segmentation, while Hierarchical Clustering allows for more nuanced groupings and dendrogram-based insights, helping compare interpretability and actionability.
- **Text Mining & LDA**: Including textual descriptions enriches feature sets and uncovers latent topics that purely structured data might overlook, leading to more accurate or meaningful segmentation.
- **Dimensionality Reduction**: PCA and t-SNE facilitate the visualization of complex data, making it easier to evaluate cluster separability and detect potential outliers.

Preliminary Results and Analysis

We started the preliminary analysis by constructing a basic K-means model with the initial setting of 2 clusters and a random state. Then, we visualize the silhouette score and distortion elbow for cluster numbers ranging from 2 to 10.



In the above plots, the silhouette score analysis indicates that two clusters (k = 2) provide the best clustering performance. As the number of clusters increases, the silhouette score decreases, suggesting that additional clusters lead to weaker separation and less well-defined groupings. As for the distortion elbow method suggests that k = 4 is the optimal choice, as the curve exhibits a bend at this point.



The above silhouette plot for k = 2 shows two fairly distinct clusters with a slightly higher silhouette score. The silhouette plot for k = 4 indicates that some clusters contain misclassified points with negative silhouette values, meaning that increasing the number of clusters may not improve separation. The results present a trade-off between having fewer, well-defined clusters (k = 2) and a more granular but potentially weaker segmentation (k = 4).

Our initial K-means clustering results suggest that the data naturally falls into two main groups. These two clusters are more clearly defined, meaning that data points within each cluster are relatively similar. However, using four clusters instead of two allows for more differentiation but at the cost of weaker cluster cohesion, as some data points do not fit well within any single cluster.

The elbow method suggests k = 4, which would provide a finer breakdown of the data, while the silhouette score suggests k = 2, favoring a more general segmentation. These findings indicate that choosing k = 2 is better for clearly separating the data, while k = 4 could be useful if a more detailed segmentation is needed, despite some overlap in cluster assignments.

Next Steps

To refine these results, we should explore hierarchical clustering to compare its performance and confirm whether two or four clusters offer the best separation. Additionally, dimensionality reduction techniques like PCA could help remove noise and improve clustering accuracy. Another key area for improvement is the use of text mining on the description column, which was not included in the current clustering approach. By extracting features from complaint descriptions and incorporating them into the model, we could improve cluster interpretation and potentially reveal more meaningful subcategories within the data.

Appendix

- **Contribution**

Team Member	Has Done	Planned to Do
Yu-Hsiang (Rick) Wang	Data Preprocessing Model Construction Deliverable Drafting	Dimensionality Reduction Conduct Text Mining
Ching-Hsuan (Shawn) Lin	EDA Preliminary Analysis Deliverable Drafting	Conduct Text Mining Construct LDA Model
Kuang-Ching (Amanda) Ting	EDA Model Construction Deliverable Drafting	Dimensionality Reduction Construct Advanced Cluster Models
Ming-Hua (Jasmine) Tsai	Data Preprocessing Preliminary Analysis Deliverable Drafting	Construct LDA Model Construct Advanced Cluster Models

- **GitHub Project:** https://github.com/shanelin0107/BA820_Group_Project_Team7.git
- **References:**
  - ChatGPT: https://chatgpt.com/share/67b6a8ac-4e4c-8009-8670-3f927ce830c9
  - City and County of San Francisco. Annual Budget Reports, Department of Building Inspection (DBI) (2018-2022). https://sf.gov
  - San Francisco Open Data. Building Complaints and Violations Dataset (2017-2022). https://data.sfgov.org
  - San Francisco Controller's Office. Audit Report on Building Complaint Classification Efficiency (2021). https://sfcontroller.org
- **Timeline**
  - **Week 1 2/20~ 2/26**
    - Implement Hierarchical Clustering and compare its grouping with the K-Means results to assess interpretability and actionability.
    - **Text Preprocessing**: Reintroduce complaint descriptions. Perform text cleaning: remove stop words, punctuation, and special characters; convert text to lowercase; apply lemmatization.
    - **Text Vectorization**: Convert the cleaned text data into numerical representations (TF-IDF, Bag-of-Words).
  - **Week2 2/27~3/5**
    - **Topic Modeling**: Apply LDA (or an alternative method) to extract latent topics from the text.

- **Dimensionality Reduction & Visualization**: Use PCA or t-SNE to visualize the clustering results from both structured data and text-derived features.
- **Integration & Analysis**: Integrate insights from both clustering and topic modeling, and assign severity levels to the clusters.
- **Review & Reporting**: Refine the models and visualizations. Prepare and finalize the milestone report.

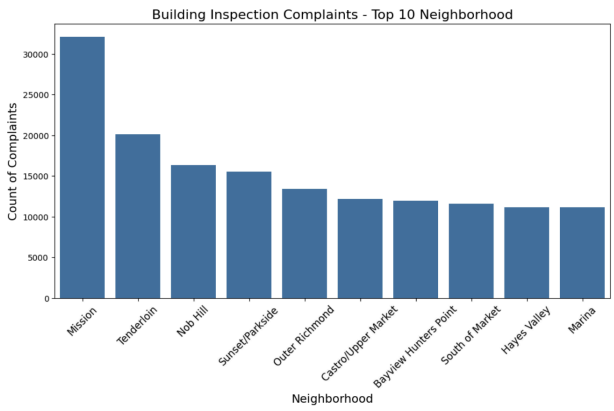
- **Supplemental Data and Results**
  - Label Encoder to convert categorical columns.

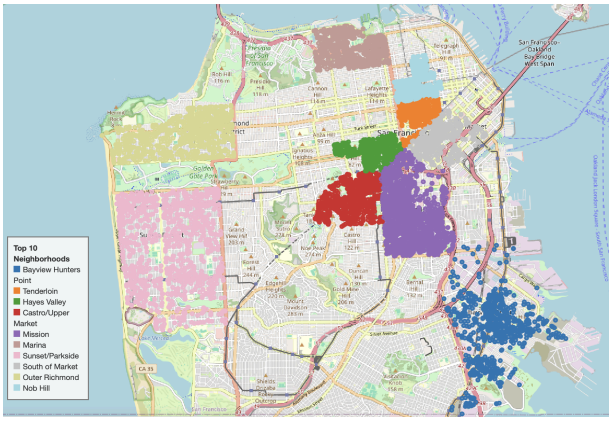
```
from sklearn.preprocessing import LabelEncoder
categorical_columns = ['Status', 'Receiving Division', 'Assigned Division', 'Analysis Neighborhood']

label_encoder = LabelEncoder()
for col in categorical_columns:
    df_clean_raw[col] = label_encoder.fit_transform(df_clean_raw[col])


✓ 0.3s
```

  - Barplot and Map to show the top 10 receiving and assigned divisions.





  - Visualize Distortion Elbow

```
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer
kmeans = KMeans(2, random_state= 42)

kmeans.fit(df_clean_raw)


visualizer = KElbowVisualizer(
    kmeans, k=(2,10), metric='distortion', timings=False #metric='silhouette' metric='distortion'
)

visualizer.fit(df_clean_raw)        # Fit the data to the visualizer
visualizer.show()
```

- Visualize Silhouette Plots

```
from yellowbrick.cluster import SilhouetteVisualizer

kmeans_model = KMeans(2, random_state= 42)
visualizer = SilhouetteVisualizer(kmeans_model, colors='yellowbrick', timings=False)

visualizer.fit(df_clean_raw)        # Fit the data to the visualizer
visualizer.show()        # Finalize and render the figure
```