Contents

[2. Assignment 2](#_Toc126410553)

[2.1. Case 2](#_Toc126410554)

[2.2. Assignment Goals 2](#_Toc126410555)

[2.2.1. Visualization Deliverables 3](#_Toc126410556)

[2.2.2. OTHER DELIVERABLES 3](#_Toc126410557)

[3. Documentation 4](#_Toc126410558)

[3.1. Vision Diagram 4](#_Toc126410559)

[3.2. Data Wrangling and Cleansing 5](#_Toc126410560)

[3.3. Database Installation 6](#_Toc126410561)

[3.4. Data Mapping and Integration 7](#_Toc126410562)

[3.5. Data Validation and Data Visualization 8](#_Toc126410563)

[3.6. System Integration and User Acceptance Testing 9](#_Toc126410564)

[3.7. Challenges Encountered 10](#_Toc126410565)

[3.8. End User Instructions 11](#_Toc126410566)

# Assignment

## Case

Fire Department Calls for Service – 2001 to 2023.

The project deals with getting insightful information from the data collected by the Fire Department in San Fransisco. With data processing, data wrangling, and using a database to store the data, analytical dashboards were created for easy understanding through visualizations.



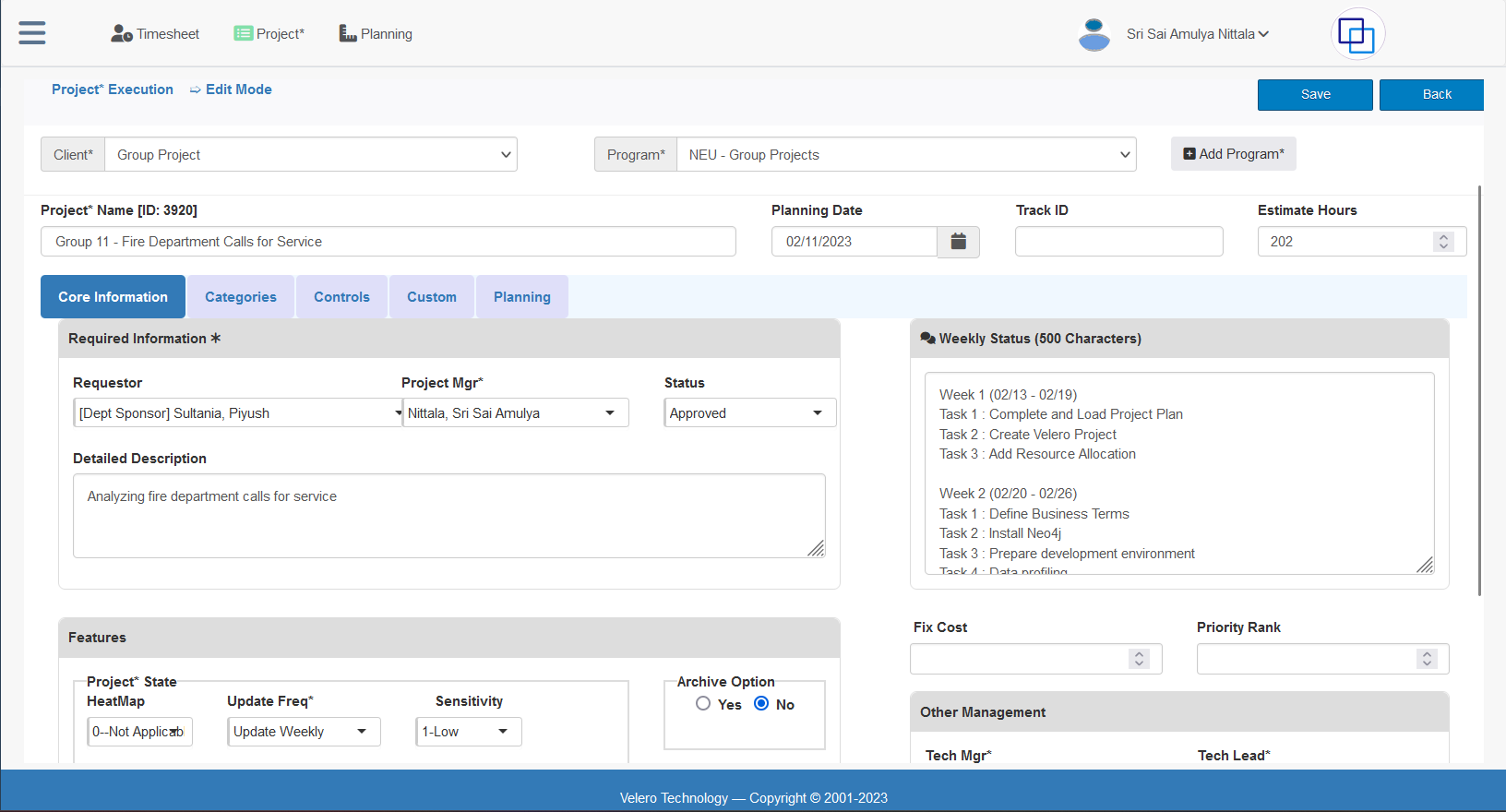
Tools used:

* Jupyter Notebooks (Python) – for data processing and data wrangling
* Neo4j (Cypher) - for data storage
* Tableau – for data visualizations

## Assignment Goals

To work with datasets, Perform/Create:

* Create your group assignment project in Velero:
  + Project



* + Graphical user interface, application

    Description automatically generatedProject Plan
  + Graphical user interface, application

    Description automatically generatedResource Allocation
  + Graphical user interface

    Description automatically generatedTimesheet
  + Graphical user interface, text, application

    Description automatically generatedIssues & Risks

### Visualization Deliverables

Once you wrangle/clean/join/integrate the data, import the data into **NEO4J** and illustrate how to use the appropriate graph to illustrate various aspects of analysis.

* Who is using this dashboard and how do they benefit from your dashboard?

The dashboard and analysis from the visualizations will be beneficial and useful to all governing authorities of the emergency response department. The fire department, medical services, and other emergency response teams. Essential information can be retrieved where improvements can be made based on many factors.

* What value would be generated using this dashboard?

The aim of this dashboard is to understand resource allocation, preparedness, identify areas of improvement in different departments, cities, neighborhoods, and supervising districts. It also aims at finding areas where education can be provided to residents of cities where it is needed.

* What data are used for dimensions and columns are used for measurement?

**Dimensions:** City, NeighborhoodName, StationArea, SupervisorDistrict, Time, Address, UnitId, CallFinalDisposition, CallType, and CallTypeGroup, ALSUnit, Month

**Measures:** Year, Priority, ReactionCategory

* Did you generate any new dimensions?

A new dimension was generated using two of the columns in the original dataset. The columns are DispatchDtTm(the time at which the unit was dispatched from the station) and OnSceneDtTm(the time at which the unit reached the address). The difference of the two columns was taken and based on the time, the values were categorized into the following:

* Prompt (0 – 3 min)
* Delayed (3+ min)

This column is named ReactionCategory

### OTHER DELIVERABLES

All the files related to the project (Jupyter Notebook, Cypher code, Tableau Worksheet) can be found int the submission file.

# Documentation

## Vision Diagram

Diagram

Description automatically generated

## Data Wrangling and Cleansing

**About Dataset**

The dataset contains information about fire calls for service from the years 2001 to 2023. It includes data on all emergency units that responded to these calls. Each record in the dataset includes details such as the call, the incident, address, the unit sent and its relevant time intervals. The information related to the call was the call number, time the call was received, the cause and risk of the incident, the priority assigned to it and the final disposition. Since most calls require the response of multiple units, there are multiple records for each call number. The addresses in the dataset are associated with a city, zipcode, coordinates, neighborhood, station area, supervising district, and box. The specific address was not provided, the address was only provided in the form of blocks.

* **Data profiling**

To get a basic understanding of the data pandas profiling was used. It is an open-source Python module with which we can quickly do an exploratory data analysis with just a few lines of code.

Table

Description automatically generated with medium confidence

* **Correlation**

Chart

Description automatically generatedThe correlation between columns shows that most of the columns have invalid coefficient for correlation. From the **correlation matrix** in the columns selected for analysis, we can find that **Analysis boundaries and Analysis Neighborhoods** are highly corelated and mean the same. So, one of them could be dropped.

* **Missing values**

Chart, waterfall chart

Description automatically generatedHeatmap of missing values in the columns selected for analysis.

* **Data manipulation and wrangling**

Out of all the columns only some were used for analysis and the following is how they were treated for null values, manipulation, and wrangling.

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Data Type** | **Description** | **Treated** |
| Call Type Group | Text | The category into which the incident falls into: Fire, Alarm, Potentially Life-Threatening, and Non-Life Threatening | Based on the Call Type (the cause of the fire) and common logic, the null values in Call Type Group were filled in by matching the cause and the risk it falls into |
| City | Text | The city in which the incident occurred | The missing values of City were filled in by matching the zip code and city |
| Station Area | Text | The station area associated with the address of the incident | The missing values of Station Area were filled in by matching the City and Station Area |
| Reaction Category | Text | The category into which the response time of the unit to the address falls into | The difference between DispatchDtTm and OnSceneDtTm was taken and based on the minutes the unit was assigned a category as follows:   * Prompt: 0-3 mins * Delayed: 3+ mins |
| Time | Text | The time at which the call was made i.e, AM and PM | The time at which the call was made, extracted from the column Call Date |
| Month | Integer | The month in which the call was made | The month was extracted from the column Call Date |
| Year | Integer | The year when the call was made | The year was extracted from the column Call Date |

After treating null values, data processing, manipulation and wrangling, these are the final columns in the dataset.

|  |  |
| --- | --- |
| **COULMN** | **DESCRIPTION** |
| RowID | Unique identifier for each call |
| Year | The year in which the call was made |
| Month | The month in which the call was made |
| Time | The time at which the call was made, either AM or PM |
| CallType | The type of call received by the Fire Department into categories: "Fire", "Medical Incident", "Alarms", "Traffic Collision", etc. |
| CallTypeGroup | The group into which the call received by the Fire Department falls into:   * Fire * Alarm * Potentially Life-Threatening * Non-Life-Threatening |
| CallFinalDisposition | The final decision made by the unit after being addressed |
| Priority | The priority level given to the incident. Includes:   * 1 – Emergency * 2 – Urgent * 3 – Non-urgent * A – Fire * B – Non-Fire * E – EMS * I – Non-EMS |
| UnitId | The identifier of the unit sent to the location of the incident |
| ALSUnit | Whether an Advanced Life Support unit was sent or not (Boolean) |
| Address | The address where the incident occurred |
| City | City in which the address of the incident occurred |
| NeighborhoodName | Identifies the neighborhood in which the call was received. |
| StationArea | Station associated with the address |
| Supervisor District | Supervising District associated with the address |
| ReactionCategory | The category into which the response time of the unit falls into (Prompt, Delay) |

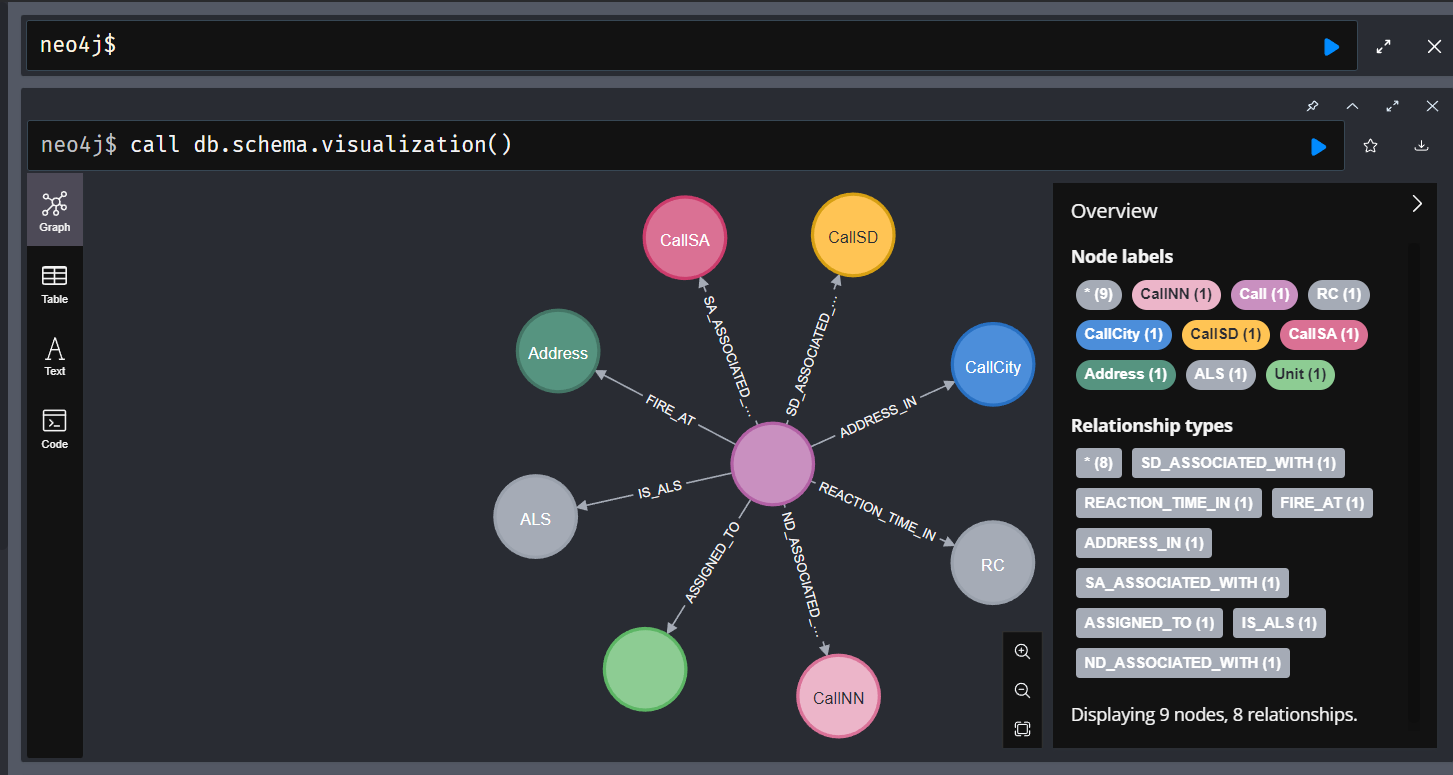
## Database Installation

**Neo4j Database**

The Neo4j database is a type of database that organizes data as graphs, providing reliable data integrity and following the ACID principle (which stands for Atomicity, Consistency, Isolation, and Durability). Similar to SQL for RDBMS, the Graph database has its own language called Cypher, used for accessing data.

* Downloaded Neo4j 1.5.7 and created a database of version 5.6.0
* Graphical user interface, application

  Description automatically generatedInstalled its respective APOC for enabling connection with Tableau.



## Data Mapping and Integration

To load data into Neo4j, a data model is to be designed for our business case. Below are the nodes and relationships and a snippet of the Cypher code.

|  |
| --- |
| Nodes |
| Call |
| Unit |
| ReactCategory |
| ALS |
| Address |
| City |
| CallNN |
| CallSD |
| CallSA |

|  |
| --- |
| Relationships |
| ADDRESS\_IN |
| ASSIGNED\_TO |
| FIRE\_AT |
| IS-ALS |
| REACTION\_TIME\_IN |
| ND\_ASSOCIATED\_WITH |
| SD\_ASSOCIATED\_WITH |
| SA\_ASSOCIATED\_WITH |

Text

Description automatically generatedCreation and loading data into the node “Call”:

Building the relationship between “Call” node and “Unit” node:

Text

Description automatically generated

## Data Validation and Data Visualization

As a leading data visualization tool, Tableau has many desirable and unique features. Its powerful data discovery and exploration application allows you to answer important questions in seconds. Tableau's drag and drop interface to visualize any data, explore different views, and even combine multiple databases easily. It does not require any complex scripting.

Diagram

Description automatically generated

1. **Number of calls in San Fransisco by Neighborhoods:**

A picture containing application

Description automatically generated

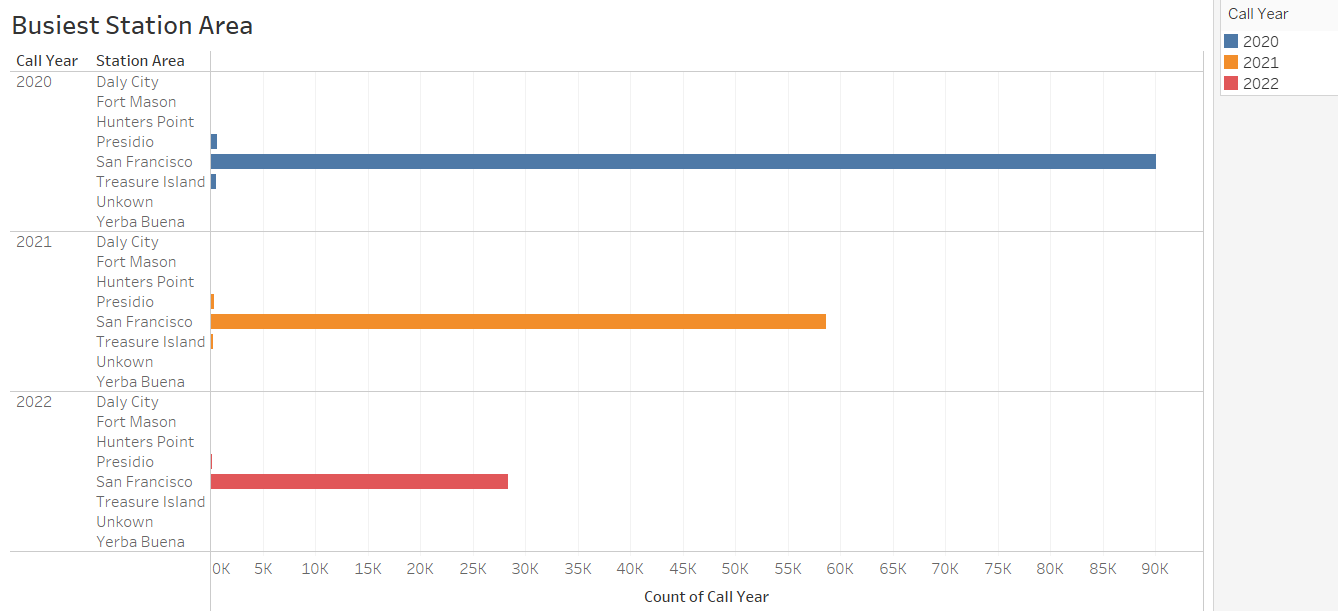
Here, the greatest number of calls are in San Fransisco for different priorities. And by neighborhoods, Tendorloin (not seen) has the highest number of calls at night (PM). Lastly, for priorities A & E Tendorloin still has the greatest number of calls. Hence, all emergency response stations in Tendorloin should always be alert and have resources at night.

1. **Which cause of incident is the most common?**

Chart

Description automatically generated

The most common cause for calls is medical incidents for which ALS Unit was sent meaning it is a life-threatening cause and should be addressed seriously. Hospitals, emergency units and urgent care units should be prepared all year round.

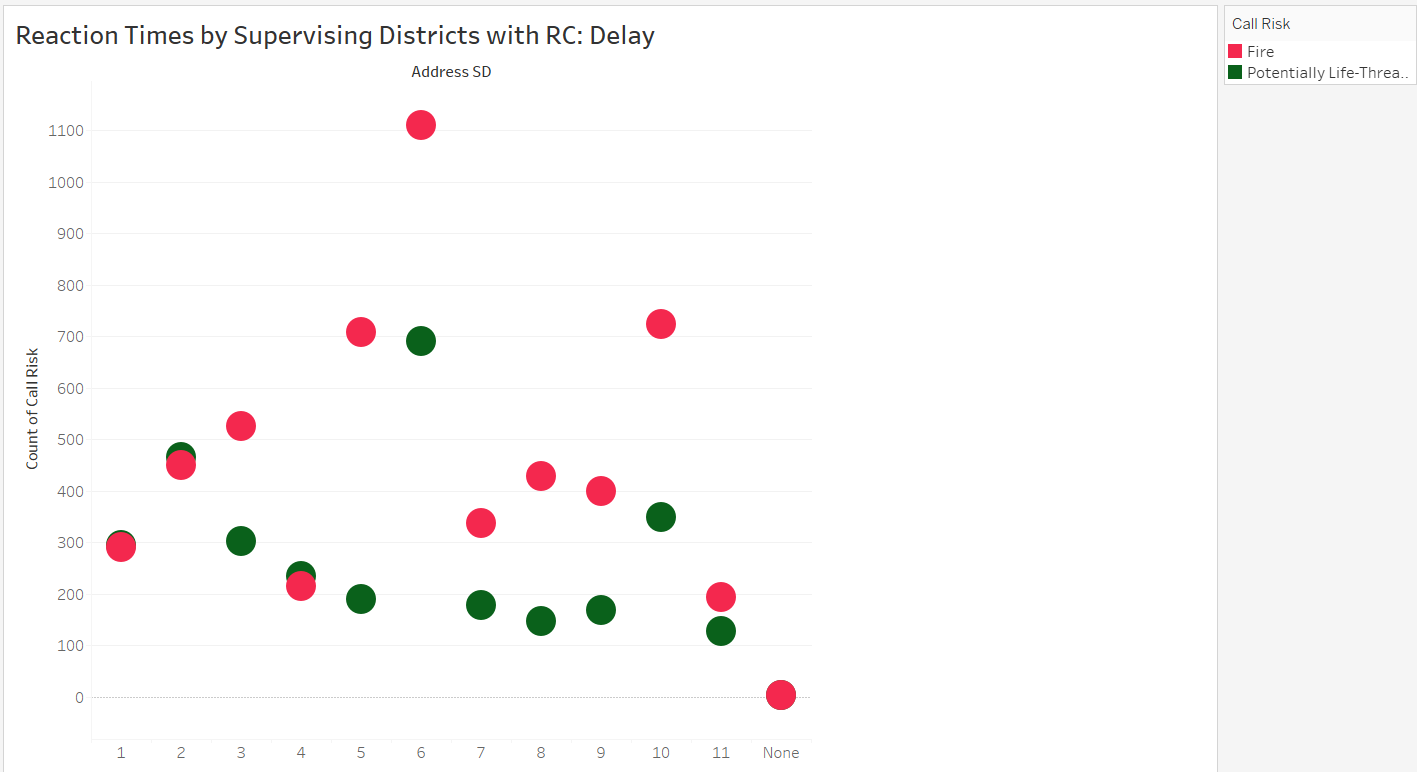
1. **Which station area can expect the greatest number of calls?**

In 2020, 2021, and 2022 San Fransisco has the busiest stations, and the pattern can be assumed to me the same for the year 2023

1. Chart, bar chart, waterfall chart

   Description automatically generated**Reaction Times by Calls of different priorities for disposition “Gone on Arrival.”**

Even for calls of priority 3, there have been many cases where the units have arrived late and the victims have passed away.

1. **Reaction Times by different Supervising Districts**

With Fire and Potentially Life-Threatening being the most dangerous risks districts 6 and 10 should find areas for improvement to reduce their response time to such dangerous situations.

1. Chart, bubble chart

   Description automatically generated**Which city has the highest number of call risk: Alarm?**
2. **Chart, text, scatter chart

   Description automatically generatedDistribution of call disposition by risk and supervising district**

**Target Audience**

The target audience for the dataset are groups interested in understanding the nature and frequency of emergency incidents reported to the San Francisco Fire Department.

* **Emergency responders and Fire Department** personnel who want to gain insights into the types of calls they are responding to and where those calls are most frequent.
* **Data scientists** who want to use this data as part of a larger analysis or modeling project.
* **Journalists or news organizations** who want to report on trends or patterns in emergency incidents in San Francisco.
* **Residents** of San Francisco who are interested in understanding the nature and frequency of emergency incidents in their neighborhoods or citywide.

## System Integration and User Acceptance Testing

Graphical user interface, text, application

Description automatically generated**Neo4j – Count of Records Display**

**Tableau – Count of Records Display across categories**

Background pattern

Description automatically generated with low confidence

The count and of records display across tools ensures system integration and matching records supports user acceptance testing.

## Challenges Encountered

**Lack of expertise in Neo4j and understanding cypher queries**

Inefficiency and loss in operation and production of products due to lack of expertise puts the business at risk. The severity of the failure and its related impacts can depend on the reasons for the expertise that results in inefficiency and loss of operation and production.

*Solution*

Learning to use the database through documentation and online courses

Practice on the default Movies database for better understanding

**Compatibility and interfacing issues**

Universality of data is not guaranteed, and business dashboards often encounter the challenge of connectivity and compatibility. When a dashboard is unable to connect with a crucial business system, the information it offers may be erroneous, outdated, or limited in its practicality.

*Solution*

To overcome this issue, users must enter data manually, which effectively undermines the purpose of the dashboard.

**Data Loading Issues**

The size of the dataset was almost 6 million rows which were difficult to load into Neo4j due to memory issues.

*Solution*

Performed Bias Variance Analysis on the dataset using random shuffling. Using the minimum percentage of data that needs to be consumed across all variables to achieve higher accuracy, it can be concluded that the dataset does not need to be consumed entirely. This means that the analysis can be conducted on a smaller sample of the dataset achieving the same amount of accuracy, which can help in reducing computational costs and processing time.

## End User Instructions

The end user instructions are divided into 3 parts for easy understanding and execution.

1. **Data preparation for final dataset**

* Download the dataset from [https://data.sfgov.org/Public-Safety/Fire-Department-Calls-for-Service/nuek-vuh3](https://nam12.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdata.sfgov.org%2FPublic-Safety%2FFire-Department-Calls-for-Service%2Fnuek-vuh3&data=05%7C01%7Ck.heydari%40northeastern.edu%7C9c7559e4cce04782a81608db039f3c03%7Ca8eec281aaa34daeac9b9a398b9215e7%7C0%7C0%7C638107754660790685%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=7nHkXloaDj%2B3f0sM8AcjlH2XWZmRGsYeomZ0Uq09ofk%3D&reserved=0)
* Install Python 3.x (<https://www.python.org/downloads/>) and Anaconda (<https://docs.anaconda.com/anaconda/install/index.html>) according to system
* Start Anaconda and launch Jupyter Notebooks
* Open the attached Jupyter Notebook and run each cell, then the final dataset will get saved as “Cleaned\_data\_FINAL.csv”

1. **Data loading in Neo4j**

* Download Neo4j 1.5.7 (<https://neo4j.com/download/>) according to system
* Create a database within the project “Example Project” and reset its password
* Download the APOC for the database (needed for Tableau connection)
* Click on the database and open the import folder and place the dataset in that folder
* Start the database
* Once started open the Neo4j browser for data loading
* Copy and paste the attached Cypher code in the cell and execute
* The progress can be seen in the left panel

1. **Data visualization using Tableau**

* Download JDBC driver from the attached file
* Open Tableau and click on other databased and open JDBC database
* Enter **jdbc:neo4j://localhost:7687/neo4j?&UID=neo4j&PWD=password** and the password set for the database
* Open the attached Tableau workbook and update the data and view all the dashboards