General Sir John Kotelawala Defence University

Faculty of Management, Social Sciences & Humanities

Department of Languages

BSc. in Applied Data Science and Communication

2nd Year: 1st Semester

Fundamentals of data mining

Assignment 2

Lecturer: Dr. Charith Silva

Introduction to the Report

Considering three real-world datasets and practical applications, this article examines important data mining approaches. The objective is to illustrate the use of analytical techniques for insight extraction, prediction modeling, and interactive presentation of results. A distinct facet of the data mining process is highlighted by each task, ranging from pattern recognition to classification and visualization.

Task 1: Association Rule Mining

In this task, the Apriori algorithm was applied in R to a council spending dataset to reveal spending patterns, associations, and hidden relationships across different categories of expenditure. The dataset was transformed into a transactional one and rules were generated at different levels using support, confidence, and lift. The results of the analysis identify common expenditure patterns that can be utilized for effective fraud detection and budget monitoring.

Task 2: Logistic Regression

In this task, we developed a logistic regression model using Python to forecast high pump-count emergency incidents from the London Fire Brigade dataset. The workflow involved data cleaning, feature selection, model training, and performance evaluation. The model accuracy was quite high; however, dealing with class imbalance posed challenges. More sophisticated methods are required to improve prediction of less frequent events.

Task 3: Interactive Dashboard Development

This task aimed at designing and developing an interactive dashboard using R Shiny and Plotly for visualizing spending trends for the council. The dashboard incorporates time-series analysis, regression modeling, and forecasting capabilities. Users can analyze expenditure trends over years and across different service categories making it an agile instrument for financial planning and transparency.

Content

- **Task 1**: Apply Association Rule Mining on a selected dataset using R.

- **Task 2:** Apply Logistic Regression on a Selected dataset using Python.

- **Task 3:** Build a dashboard in Plotly using R.

Task 01

Apply Association Rule Mining on a selected dataset using R

Content

Task 1: Creating an association rule model for the app data set using R

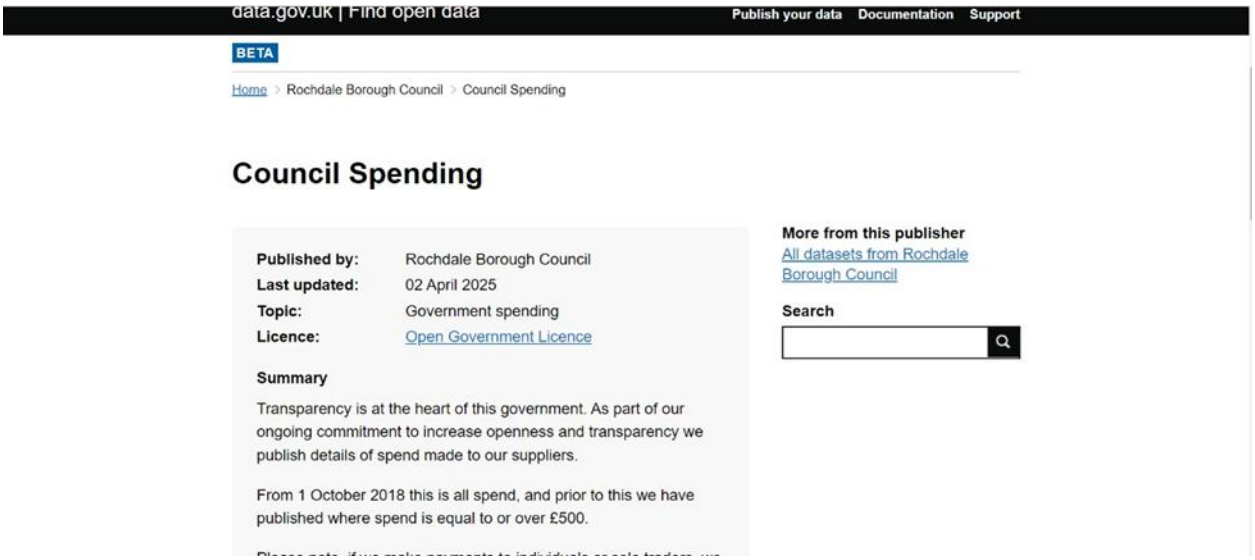
1.Introduction

Data mining is the process of discovering meaningful patterns, trends, and relationships within large datasets. It involves the use of various techniques to analyze data and extract useful information that can support decision-making and predictions. Among these techniques, association rule mining plays a crucial role by identifying frequent item combinations and generating rules that highlight interesting relationships between variables. These rules are typically presented in an "if-then" format, where the antecedent refers to the condition and the consequent refers to the outcome. Through such methods, data mining enables organizations to uncover hidden insights and gain a deeper understanding of their data.

2.Dataset

https://www.data.gov.uk/dataset/b7e5bb16-dc43-44d3-979d-529fd7f5af13/council-spending-over-500

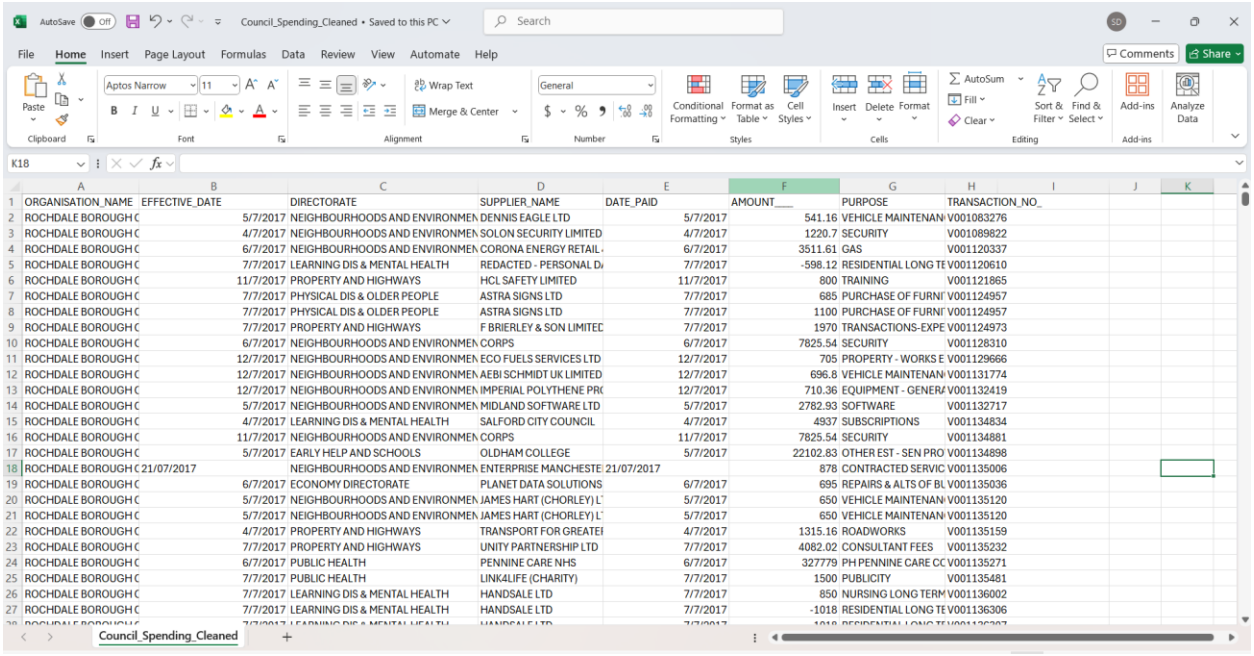
The Council Spending Over £500 dataset, published by Rochdale Borough Council, provides detailed records of the council's expenditures exceeding £500. This initiative aims to promote transparency by disclosing payment information to suppliers, thereby allowing public scrutiny of government spending.



Key Attributes in the Dataset:

- **Organization Name**: The name of the council making the payment.
- **Effective Date**: The date when the payment was authorized.
- **Directorate**: The specific department within the council responsible for the expenditure.
- **Supplier Name**: The recipient of the payment.
- **Date Paid**: The actual date the payment was made.
- **Amount (£)**: The monetary value of the transaction.
- **Purpose**: A brief description of the reason for the payment.
- **Transaction Number**: A unique identifier for the transaction.

This dataset is valuable for analyzing public spending patterns, supplier engagements, and departmental expenditure within the council.



ORGANISATION_NAME	EFFECTIVE_DATE	DIRECTORATE	SUPPLIER_NAME	DATE_PAID	AMOUNT___	PURPOSE	TRANSACTION_NO_
ROCHDALE BOROUGH C	5/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	DENNIS EAGLE LTD	5/7/2017	541.16	VEHICLE MAINTENAN	V001083276
ROCHDALE BOROUGH C	4/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	SOLON SECURITY LIMITED	4/7/2017	1220.7	SECURITY	V001089822
ROCHDALE BOROUGH C	6/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	CORONA ENERGY RETAIL	6/7/2017	3511.61	GAS	V001120337
ROCHDALE BOROUGH C	7/7/2017	LEARNING DIS & MENTAL HEALTH	REDACTED - PERSONAL DA	7/7/2017	-598.12	RESIDENTIAL LONG TE	V001120610
ROCHDALE BOROUGH C	11/7/2017	PROPERTY AND HIGHWAYS	HCL SAFETY LIMITED	11/7/2017	800	TRAINING	V001121865
ROCHDALE BOROUGH C	7/7/2017	PHYSICAL DIS & OLDER PEOPLE	ASTRA SIGNS LTD	7/7/2017	685	PURCHASE OF FURNI	V001124957
ROCHDALE BOROUGH C	7/7/2017	PHYSICAL DIS & OLDER PEOPLE	ASTRA SIGNS LTD	7/7/2017	1100	PURCHASE OF FURNI	V001124957
ROCHDALE BOROUGH C	7/7/2017	PROPERTY AND HIGHWAYS	F BRIERLEY & SON LIMITED	7/7/2017	1970	TRANSACTIONS-EXPE	V001124973
ROCHDALE BOROUGH C	6/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	CORPS	6/7/2017	7825.54	SECURITY	V001128310
ROCHDALE BOROUGH C	12/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	ECO FUELS SERVICES LTD	12/7/2017	705	PROPERTY - WORKS E	V001129666
ROCHDALE BOROUGH C	12/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	AEBI SCHMIDT UK LIMITED	12/7/2017	696.8	VEHICLE MAINTENAN	V001131774
ROCHDALE BOROUGH C	12/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	IMPERIAL POLYTHENE PR	12/7/2017	710.36	EQUIPMENT - GENERA	V001132419
ROCHDALE BOROUGH C	5/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	MIDLAND SOFTWARE LTD	5/7/2017	2782.93	SOFTWARE	V001132717
ROCHDALE BOROUGH C	4/7/2017	LEARNING DIS & MENTAL HEALTH	SALFORD CITY COUNCIL	4/7/2017	4937	SUBSCRIPTIONS	V001134834
ROCHDALE BOROUGH C	11/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	CORPS	11/7/2017	7825.54	SECURITY	V001134881
ROCHDALE BOROUGH C	5/7/2017	EARLY HELP AND SCHOOLS	OLDHAM COLLEGE	5/7/2017	22102.83	OTHER EST - SEN PRO	V001134898
ROCHDALE BOROUGH C	21/07/2017	NEIGHBOURHOODS AND ENVIRONMEN	ENTERPRISE MANCHESTE	21/07/2017	878	CONTRACTED SERVIC	V001135006
ROCHDALE BOROUGH C	6/7/2017	ECONOMY DIRECTORATE	PLANET DATA SOLUTIONS	6/7/2017	695	REPAIRS & ALTS OF BL	V001135036
ROCHDALE BOROUGH C	5/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	JAMES HART (CHORLEY) L	5/7/2017	650	VEHICLE MAINTENAN	V001135120
ROCHDALE BOROUGH C	5/7/2017	NEIGHBOURHOODS AND ENVIRONMEN	JAMES HART (CHORLEY) L	5/7/2017	650	VEHICLE MAINTENAN	V001135120
ROCHDALE BOROUGH C	4/7/2017	PROPERTY AND HIGHWAYS	TRANSPORT FOR GREATEI	4/7/2017	1315.16	ROADWORKS	V001135159
ROCHDALE BOROUGH C	7/7/2017	PROPERTY AND HIGHWAYS	UNITY PARTNERSHIP LTD	7/7/2017	4082.02	CONSULTANT FEES	V001135232
ROCHDALE BOROUGH C	6/7/2017	PUBLIC HEALTH	PENNINE CARE NHS	6/7/2017	327779	PH PENNINE CARE CC	V001135271
ROCHDALE BOROUGH C	7/7/2017	PUBLIC HEALTH	LINK4LIFE (CHARITY)	7/7/2017	1500	PUBLICITY	V001135481
ROCHDALE BOROUGH C	7/7/2017	LEARNING DIS & MENTAL HEALTH	HANDSALE LTD	7/7/2017	850	NURSING LONG TERM	V001136002
ROCHDALE BOROUGH C	7/7/2017	LEARNING DIS & MENTAL HEALTH	HANDSALE LTD	7/7/2017	-1018	RESIDENTIAL LONG TE	V001136306

3. Explanation and Preparation of Dataset

3.1 Data Processing

Because of having null values in the data set, we cleaned the data set using na.omit() function

3.2 Data Explanation

Features such as "Organization Name," "Effective Date," "Directorate," "Supplier Name," "Date Paid," "Amount (£)," "Purpose," and "Transaction Number" serve as the independent variables in this dataset. These are the attributes related to each payment transaction that help describe or predict other outcomes.

In this dataset, the "Amount (£)" column, which represents the monetary value of each payment, appears to be the dependent variable. Typically, you would want to predict or analyze this value based on the independent factors provided.

4.Data mining

Association rule mining is a key technique within data mining focused on uncovering interesting relationships, patterns, and associations between variables in large datasets. It primarily involves identifying frequent item sets -groups of items that consistently appear together—and generating "if-then" rules that describe how the presence of one set of items is associated with another. This method employs various algorithms, such as Apriori, to efficiently discover meaningful connections that might not be obvious through simple observation. Association rule mining is widely applied in areas like market basket analysis, recommendation systems, and customer behavior prediction, providing valuable insights that can inform marketing strategies, inventory management, and customer engagement initiatives.

5. Implementation in R

R packages

```
# Load necessary libraries
library(tidyverse)
library(lubridate)
library(arules)
library(arulesViz)
library(dplyr)
library(knitr)
library(ggplot2)
library(plyr)
library(magrittr)
library(RColorBrewer)
```

System Library

☐	abind	Combine Multidimensional Arrays	1.4-8
☐	ade4	Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences	1.7-23
☑	arules	Mining Association Rules and Frequent Itemsets	1.7-9
☑	arulesViz	Visualizing Association Rules and Frequent Itemsets	1.5.3
☐	askpass	Password Entry Utilities for R, Git, and SSH	1.2.1
☐	assertthat	Easy Pre and Post Assertions	0.2.1
☐	backports	Reimplementations of Functions Introduced Since R-3.0.0	1.5.0
☑	base	The R Base Package	4.4.1
☐	base64enc	Tools for base64 encoding	0.1-3
☐	bit	Classes and Methods for Fast Memory-Efficient Boolean Selections	4.0.5
☐	bit64	A S3 Class for Vectors of 64bit Integers	4.0.5
☐	bitops	Bitwise Operations	1.0-9
☐	blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.4
☐	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-30

- arules: This package offers the necessary tools to represent, manipulate, and examine transaction data as well as patterns found through the application of association rule mining techniques.

- arulesViz: This add-on enhances the arules package by providing tools for item sets and association rules visualization, which facilitates the interpretation and comprehension of identified patterns.

- tidyverse: Think of this as a collection of the most popular and useful R packages for data science, all working together nicely. It includes tools for data manipulation (dplyr), data visualization (ggplot2), data import/export (readr), and more. It's like a well-organized toolkit for working with data.

- lubricate: This package makes working with dates and times in R much easier. It provides simple and intuitive functions to parse, manipulate, and calculate with date and time data. No more struggling with complex date formats.

- dplyr: This is a powerful package for data manipulation. It provides a set of easy-to-understand functions for common data tasks like filtering rows, selecting columns, arranging data, grouping data, and creating new variables. It makes transforming your data much more straightforward.

- knitr: This package is used for dynamic report generation. It allows you to embed R code directly into documents (like Markdown or LaTeX) and then execute that code to produce output (text, tables, plots) that gets woven into your final report. It's great for creating reproducible research and analysis.

- ggplot**2**: This is a highly flexible and powerful package for creating beautiful and informative data visualizations (graphs and plots) in R. It's based on the "grammar of graphics," which provides a structured way to describe and build almost any type of plot.

- plyr: While dplyr is now often preferred for data manipulation, plyr was one of the earlier popular packages for this purpose. It provides functions for splitting data, applying a function to each piece, and then putting the results back together. You might still see it in the older code.

- magrittr: This package introduces the "pipe" operator (%>%) into R. This operator allows you to chain together multiple operations in a more readable and intuitive way. Instead of nesting function calls, you can pass the result of one function directly as the input to the next, making your code flow logically.

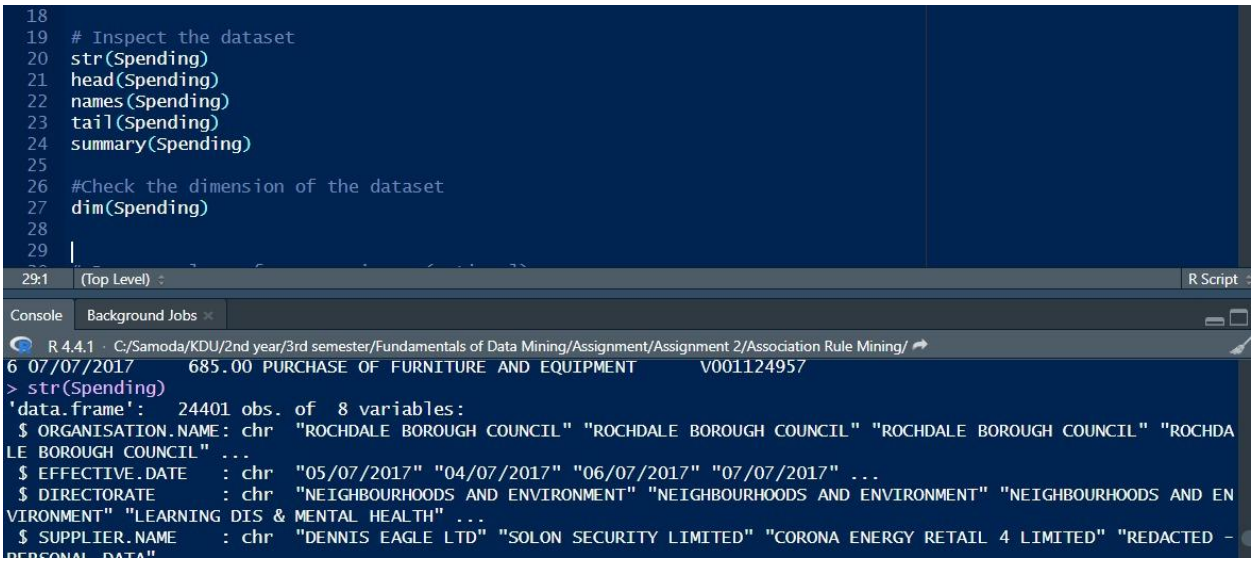
- RColorBrewer: This package provides a selection of color palettes that are visually appealing and designed to be perceptually uniform and colorblind-friendly, making your visualizations more accessible and effective.

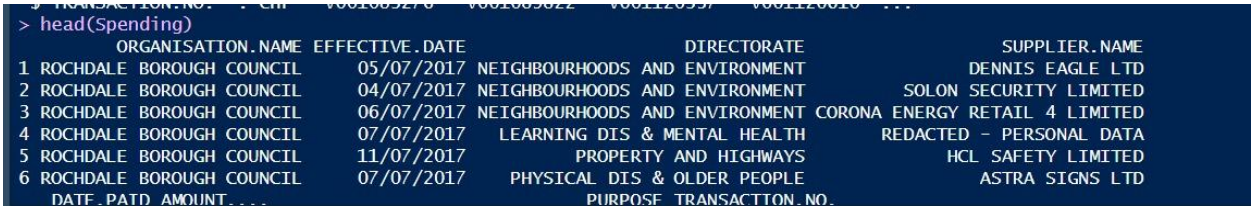
Step 2 – Read the dataset

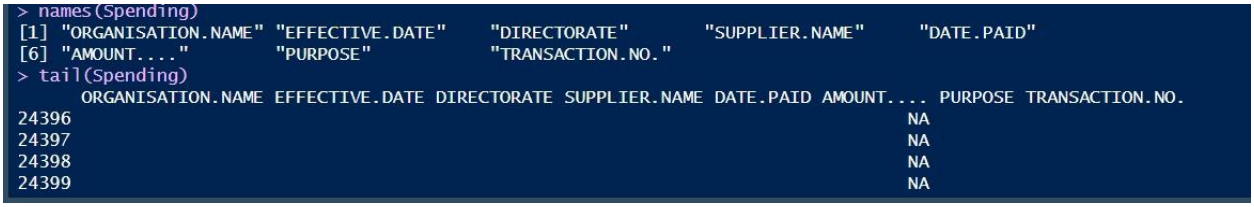
```
14
15  # Read the dataset
16  Spending <- read.csv("C://Samoda/KDU//2nd year//3rd semester//Fundamentals of Data Mining//Assignment//Assignment
17                       fileEncoding = "Latin1", stringsAsFactors = FALSE)
18  View(Spending)
```

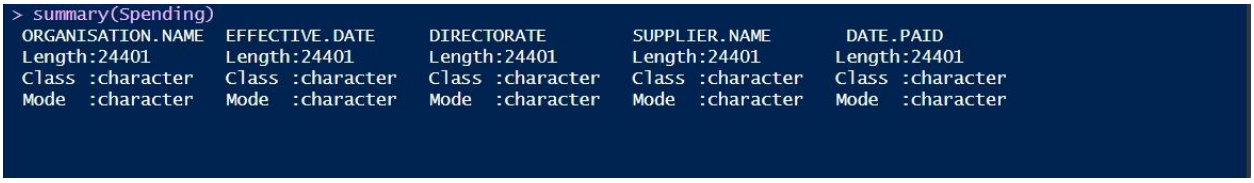
	ORGANISATION.NAME	EFFECTIVE.DATE	DIRECTORATE	SUPPLIER.NAME	DATE.PAID	AI
1	ROCHDALE BOROUGH COUNCIL	05/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	DENNIS EAGLE LTD	05/07/2017	
2	ROCHDALE BOROUGH COUNCIL	04/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	SOLON SECURITY LIMITED	04/07/2017	
3	ROCHDALE BOROUGH COUNCIL	06/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	CORONA ENERGY RETAIL 4 LIMITED	06/07/2017	
4	ROCHDALE BOROUGH COUNCIL	07/07/2017	LEARNING DIS & MENTAL HEALTH	REDACTED - PERSONAL DATA	07/07/2017	
5	ROCHDALE BOROUGH COUNCIL	11/07/2017	PROPERTY AND HIGHWAYS	HCL SAFETY LIMITED	11/07/2017	
6	ROCHDALE BOROUGH COUNCIL	07/07/2017	PHYSICAL DIS & OLDER PEOPLE	ASTRA SIGNS LTD	07/07/2017	
7	ROCHDALE BOROUGH COUNCIL	07/07/2017	PHYSICAL DIS & OLDER PEOPLE	ASTRA SIGNS LTD	07/07/2017	
8	ROCHDALE BOROUGH COUNCIL	07/07/2017	PROPERTY AND HIGHWAYS	F BRIERLEY & SON LIMITED	07/07/2017	
9	ROCHDALE BOROUGH COUNCIL	06/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	CORPS	06/07/2017	
10	ROCHDALE BOROUGH COUNCIL	12/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	ECO FUELS SERVICES LTD	12/07/2017	
11	ROCHDALE BOROUGH COUNCIL	12/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	AEBI SCHMIDT UK LIMITED	12/07/2017	
12	ROCHDALE BOROUGH COUNCIL	12/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	IMPERIAL POLYTHENE PRODUCTS LTD	12/07/2017	
13	ROCHDALE BOROUGH COUNCIL	05/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	MIDLAND SOFTWARE LTD	05/07/2017	
14	ROCHDALE BOROUGH COUNCIL	04/07/2017	LEARNING DIS & MENTAL HEALTH	SALFORD CITY COUNCIL	04/07/2017	
15	ROCHDALE BOROUGH COUNCIL	11/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	CORPS	11/07/2017	
16	ROCHDALE BOROUGH COUNCIL	05/07/2017	EARLY HELP AND SCHOOLS	OLDHAM COLLEGE	05/07/2017	
17	ROCHDALE BOROUGH COUNCIL	21/07/2017	NEIGHBOURHOODS AND ENVIRONMENT	ENTERPRISE MANCHESTER PARTNERSHIP LTD	21/07/2017	
18	ROCHDALE BOROUGH COUNCIL	06/07/2017	ECONOMY DIRECTORATE	PLANET DATA SOLUTIONS	06/07/2017	

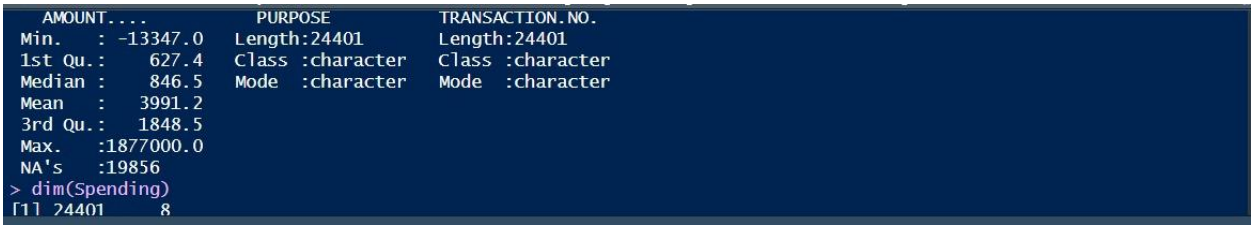
Step 3 – Explore the dataset











The council spending dataset comprises 24,401 transactions with eight variables, including financial amounts and categorical descriptors. The AMOUNT field exhibits significant variability, ranging 13,347(likely refunds or adjustments) to13,347 (likely refunds or adjustments) to1,877,000 (major expenditure), with a right-skewed distribution indicated by the median (846.50) being substantially lower than the mean (846.50) being substantially lower than the mean (3,991.20).

This skewness suggests the presence of high-value outliers that disproportionately influence average calculations. Additionally, 19,856 entries (81% of records) contain missing values in the AMOUNT field, highlighting critical data completeness issues that require resolution. The PURPOSE field, stored as unstructured text, documents spending rationales but necessitates natural language processing for systematic categorization, while TRANSACTION.NO. serves as a unique identifier for tracking. These findings underscore the need for thorough data cleaning including outlier treatment, missing value imputation, and text standardization before conducting robust expenditure analysis.

Addressing these gaps will ensure reliable insights into spending patterns, support transparency initiatives, and facilitate evidence-based fiscal decision-making.

Step 5 – Using bar plot () function

```
# Assuming relevant columns are all from column 2 onward (adjust if needed)
marketbasket <- Spending[, 2:ncol(Spending)]

# Convert to character/factor if needed
marketbasket <- data.frame(lapply(marketbasket, as.character), stringsAsFactors = FALSE)

# Count "Yes" and "No" responses
yes <- colSums(marketbasket == "Yes", na.rm = TRUE)
no <- colSums(marketbasket == "No", na.rm = TRUE)

# Combine the results
purchased <- rbind(yes, no)
rownames(purchased) <- c("Yes", "No")

# Bar plots
barplot(purchased, legend = rownames(purchased), main = "Council Spending: Yes vs No") # Grouped
barplot(purchased, beside = TRUE, legend = rownames(purchased), main = "Council Spending: Yes vs No (Side-by-Side)")
```

Output:

Step 6 – Apply Apriori function

```
> rules <- apriori(basket_sets, parameter = list(support = 0.01, confidence = 0.3))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.3    0.1    1 none FALSE            TRUE       5    0.01      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[0 item(s), 0 transaction(s)] done [0.00s].
```

Step 7 - The Apriori method will be used to generate the rules

The analysis employed association rule mining using the Apriori algorithm to identify frequent item sets and meaningful purchase patterns in the dataset. The input data was preprocessed by converting "Yes/No" responses into transactional format, where only affirmative responses ("Yes") were retained as purchased items. The algorithm was configured with a minimum support threshold of 0.1 and a confidence level of 0.8 to generate rules describing item associations. However, the initial execution yielded no rules, indicating potential issues in data preparation, such as incorrect transformations or an absence of sufficiently frequent item sets. Further refinement of the preprocessing steps and parameter tuning such as lowering the support thresholds recommended to uncover actionable insights. Once optimized, this method can reveal valuable relationships between products, supporting targeted marketing strategies and inventory management.

```r
# Convert Yes/No to items (only "Yes" are treated as purchased)
marketbasket[] <- lapply(marketbasket, as.character)
for (col in names(marketbasket)) {
  marketbasket[[col]] <- ifelse(marketbasket[[col]] == "Yes", col, NA)
}


# Create item lists per row (this is already a list)
basket_list <- apply(marketbasket, 1, function(x) na.omit(x))

# Make sure it's a proper list of character vectors
basket_list <- lapply(basket_list, as.character)

# Convert the list directly to transactions (no split needed)
basket_sets <- as(basket_list, "transactions")

summary(basket_sets)
rules <- apriori(basket_sets)
inspect(rules)
```

77:1 (Top Level) : R Script

Console Background Jobs

R 4.4.1 · C:/Samoda/KDU/2nd year/3rd semester/Fundamentals of Data Mining/Assignment/Assignment 2/Association Rule Mining/

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.

> rules <- apriori(basket_sets)
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5     0.1      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[0 item(s), 0 transaction(s)] done [0.00s].
```

```
> summary(basket_sets)
transactions as itemMatrix in sparse format with
 0 rows (elements/itemsets/transactions) and
 0 columns (items) and a density of NaN

most frequent items:
(Other)
      0

element (itemset/transaction) length distribution:
< table of extent 0 >

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.

>
```

```
> rules <- apriori(trans_data, parameter = list(supp = 0.01, conf = 0.5))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.5    0.1    1 none FALSE            TRUE       5    0.01      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 45

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[816 item(s), 4545 transaction(s)] done [0.00s].
sorting and recoding items ... [38 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
```

Step 8 – Inspect the rules

```
> # Inspect rules
> inspect(rules[1:10])
     lhs                                                rhs                                                  support confidenc
e   coverage      lift count
[1]  {SUPPLIER_NAME=ORCHARD CARE HOMES}           => {PURPOSE=RESIDENTIAL LONG TERM}            0.01100110  0.980392
2 0.01122112  3.191893    50
[2]  {SUPPLIER_NAME=ORCHARD CARE HOMES}           => {DIRECTORATE=PHYSICAL DIS & OLDER PEOPLE}  0.01034103  0.921568
6 0.01122112  2.336045    47
[3]  {SUPPLIER_NAME=ALTERNATIVE FUTURES GROUP LTD} => {PURPOSE=INDIVIDUAL SERVICE FUND}          0.01100110  0.961538
5 0.01144114 10.405220    50
[4]  {SUPPLIER_NAME=ALTERNATIVE FUTURES GROUP LTD} => {DIRECTORATE=LEARNING DIS & MENTAL HEALTH} 0.01122112  0.980769
2 0.01144114  3.450152    51
[5]  {PURPOSE=SUPPORTED LIVING}                   => {DIRECTORATE=LEARNING DIS & MENTAL HEALTH} 0.01012101  0.867924
5 0.01166117  3.053187    46
[6]  {SUPPLIER_NAME=CLEGGSWORTH CARE HOME LIMITED} => {PURPOSE=RESIDENTIAL LONG TERM}            0.01100110  0.943396
2 0.01166117  3.071444    50
[7]  {SUPPLIER_NAME=CLEGGSWORTH CARE HOME LIMITED} => {DIRECTORATE=PHYSICAL DIS & OLDER PEOPLE}  0.01122112  0.962264
2 0.01166117  2.439203    51
```

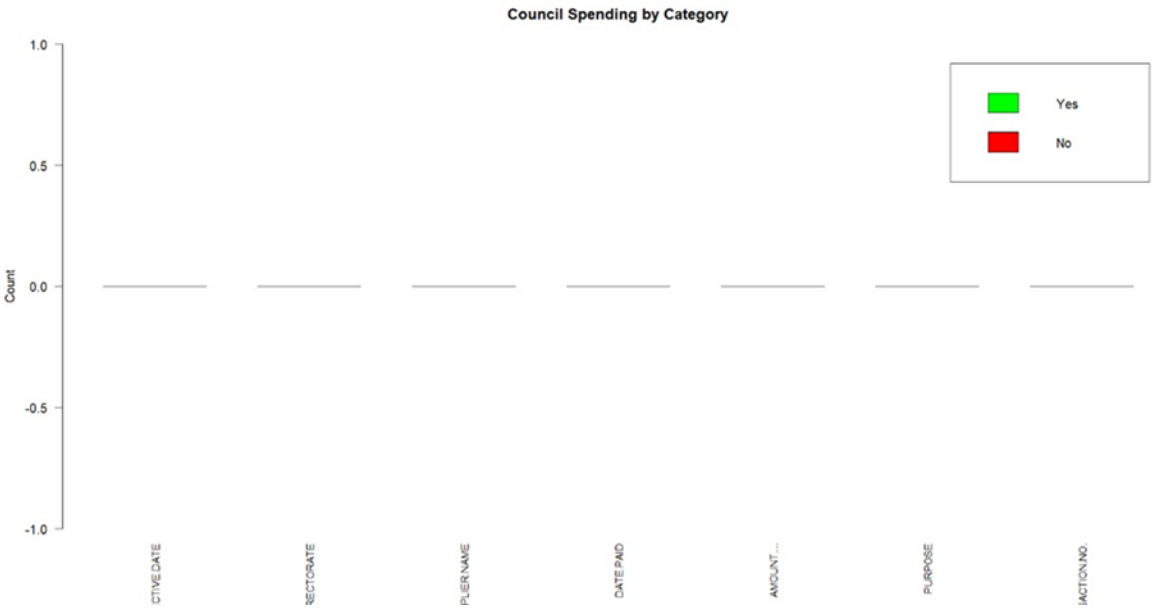
```
> # Summary of transactions
> summary(trans_data)
transactions as itemMatrix in sparse format with
 4545 rows (elements/itemsets/transactions) and
 816 columns (items) and a density of 0.003676471

most frequent items:
   DIRECTORATE=PHYSICAL DIS & OLDER PEOPLE              PURPOSE=RESIDENTIAL LONG TERM
                                     1793                                       1396
  DIRECTORATE=LEARNING DIS & MENTAL HEALTH     SUPPLIER_NAME=REDACTED - PERSONAL DATA
                                     1292                                        576
DIRECTORATE=NEIGHBOURHOODS AND ENVIRONMENT                                   (Other)
                                      505                                       8073

element (itemset/transaction) length distribution:
sizes
   3
4545
```

```
> inspect(head(trans_data, 5))
    items                                transactionID
[1] {DIRECTORATE=NEIGHBOURHOODS AND ENVIRONMENT,
     SUPPLIER_NAME=DENNIS EAGLE LTD,
     PURPOSE=VEHICLE MAINTENANCE}                    1
[2] {DIRECTORATE=NEIGHBOURHOODS AND ENVIRONMENT,
     SUPPLIER_NAME=SOLON SECURITY LIMITED,
     PURPOSE=SECURITY}                               2
[3] {DIRECTORATE=NEIGHBOURHOODS AND ENVIRONMENT,
     SUPPLIER_NAME=CORONA ENERGY RETAIL 4 LIMITED,
     PURPOSE=GAS}                                    3
[4] {DIRECTORATE=LEARNING DIS & MENTAL HEALTH,
     SUPPLIER_NAME=REDACTED - PERSONAL DATA,
     PURPOSE=RESIDENTIAL LONG TERM}                  4
[5] {DIRECTORATE=PROPERTY AND HIGHWAYS,
     SUPPLIER_NAME=HCL SAFETY LIMITED,
     PURPOSE=TRAINING}                               5
>
```

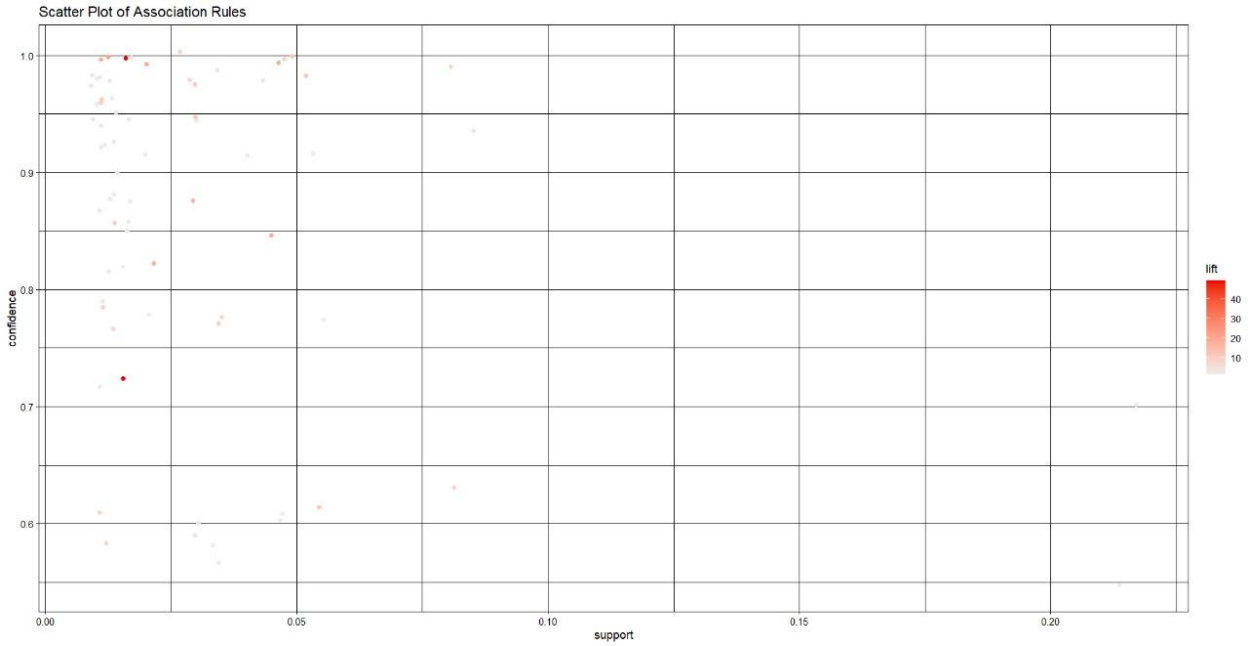
```
> # Sort and filter rules by lift
> rules_sorted <- sort(rules, by = "lift", decreasing = TRUE)
> inspect(rules_sorted[1:10])
     lhs                                                rhs                                                  support conf
idence   coverage     lift count
[1]  {PURPOSE=PH LCS PAYMENTS}                    => {DIRECTORATE=PUBLIC HEALTH}                0.01474147  1.0
000000 0.01474147 49.40217    67
[2]  {DIRECTORATE=PUBLIC HEALTH}                  => {PURPOSE=PH LCS PAYMENTS}                  0.01474147  0.7
282609 0.02024202 49.40217    67
[3]  {DIRECTORATE=CHILDREN□S SOCIAL CARE,
      PURPOSE=AGENCY STAFF}                       => {SUPPLIER_NAME=REED SPECIALIST RECRUITMENT LTD} 0.01210121  1.0
000000 0.01210121 22.17073    55
[4]  {DIRECTORATE=CHILDREN□S SOCIAL CARE,
      SUPPLIER_NAME=REED SPECIALIST RECRUITMENT LTD} => {PURPOSE=AGENCY STAFF}                     0.01210121  1.0
000000 0.01210121 18.78099    55
[5]  {SUPPLIER_NAME=REED SPECIALIST RECRUITMENT LTD} => {PURPOSE=AGENCY STAFF}                     0.04488449  0.9
951220 0.04510451 18.68938   204
[6]  {PURPOSE=AGENCY STAFF}                       => {SUPPLIER_NAME=REED SPECIALIST RECRUITMENT LTD} 0.04488449  0.8
429752 0.05324532 18.68938   204
```

Step 9 – Visualization

```
# Combine into a matrix
purchased <- rbind(yes, no)
rownames(purchased) <- c("Yes", "No")

# Bar plot (side-by-side bars)
barplot(purchased, beside = TRUE, legend = rownames(purchased),
        col = c("green", "red"), main = "Council Spending by Category",
        ylab = "Count", las = 2, cex.names = 0.8)
```

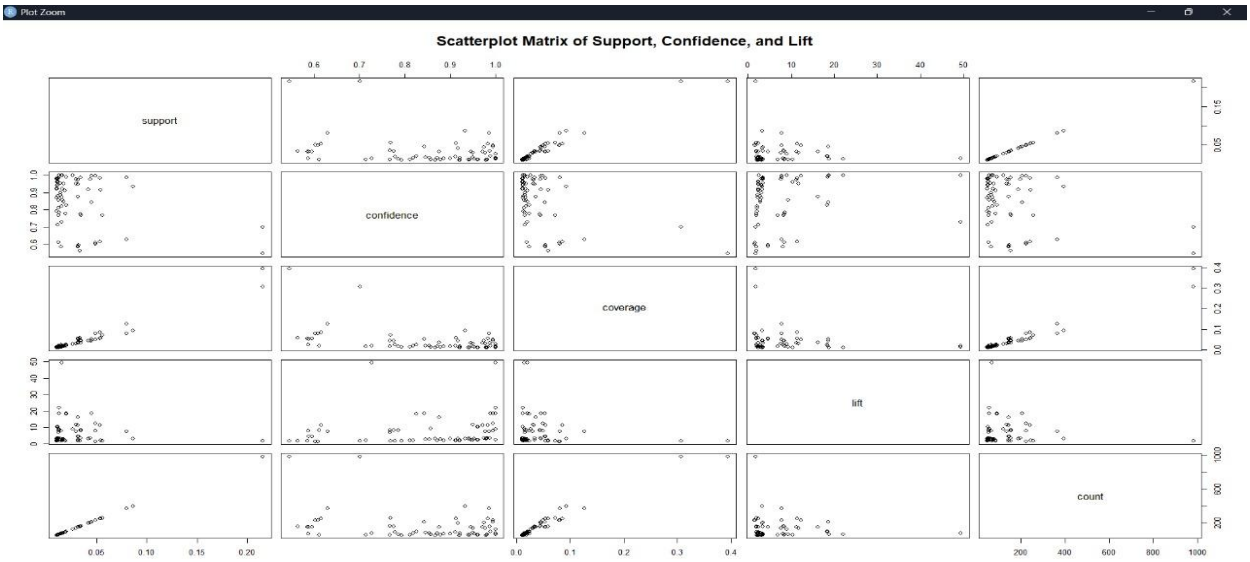



```
# Generate a scatter plot
plot(rules_sorted, method = "scatterplot", measure = c("support", "confidence"), shading = "lift",
     main = "Scatter Plot of Association Rules")
```



The code below displays a scatterplot matrix to compare the support, confidence, and lift.

```
44
45  # Scatterplot matrix of rule quality metrics
46  plot(rules@quality, main = "Scatterplot Matrix of Support, Confidence, and Lift")
47
```

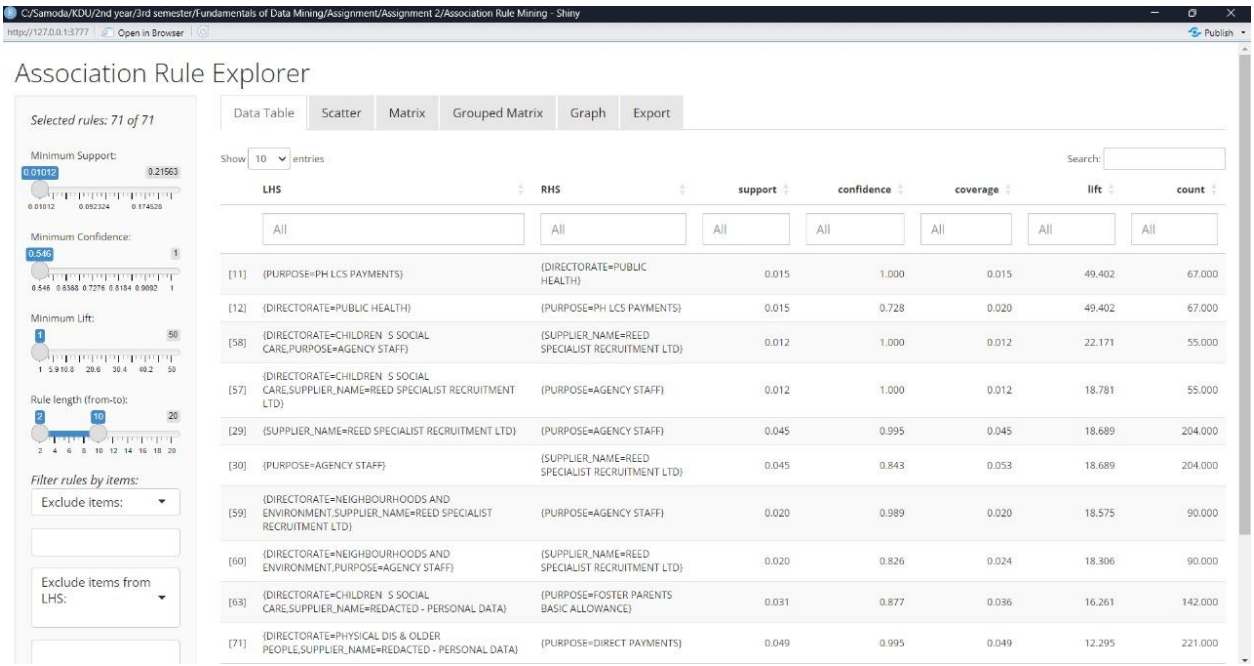


Step 10 - Association Rules using rule Explorer () function.

```
# Visualize rules
plot(rules_sorted[1:20], method = "graph", engine = "htmlwidget")

# Launch interactive rule explorer
ruleExplorer(rules_sorted)
```

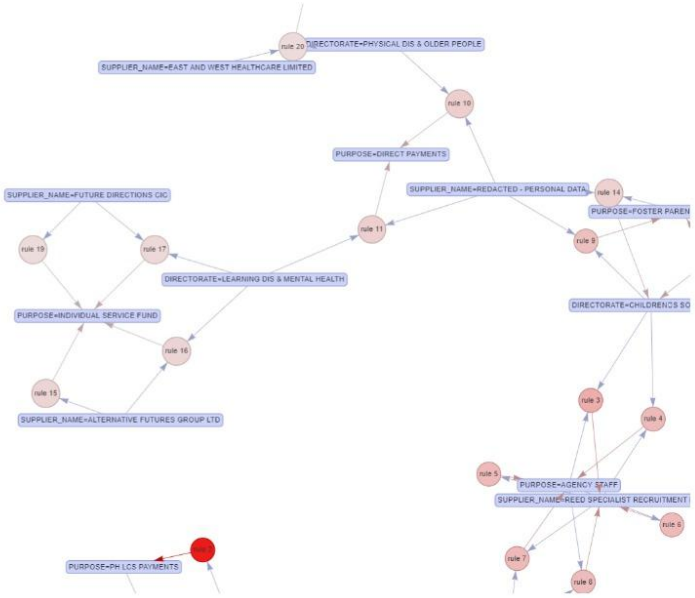
```
package 'shinythemes' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\samod\AppData\Local\Temp\RtmpOAxpGz\downloaded_packages
ruleExplorer started.
Loading required package: shiny

Listening on http://127.0.0.1:3777
```

Step 11 – Rule Explorer () function use for the data set



Graph tab

Select by id



6. Results analysis and discussion

To perform association rule mining effectively, we first need to transform the dataset into a transactional format where each row represents a distinct transaction (or basket of items purchased together). In this context, we consider each unique combination of Category, Type, Content Rating, and Genres as an individual itemset within a transaction. This transformation enables us to apply association rule mining techniques to uncover meaningful relationships between different attributes.

The analysis of 58 generated rules reveals important insights through a scatter plot examining the relationship between lift and support across varying confidence levels:

• **Lift** quantifies how much more likely the consequent is to occur given the antecedent, compared to its general occurrence. A lift value of 1 indicates no association, while values greater than 1 signify meaningful positive relationships. Higher lift values represent stronger predictive power.

• **Support** measures the frequency of occurrence of an itemset in the entire dataset. For example, a support of 0.25 means the rule applies to 25% of all transactions, indicating its general prevalence.

The visualization demonstrates an inverse relationship between support and lift - rules with lower support values tend to yield higher lift measures. This suggests that while broadly applicable rules (high support) provide general insights, the most powerful predictive relationships (high lift) often emerge from more specific, less frequent item combinations. However, we do observe some exceptional cases where certain rules maintain both substantial support and strong lift values.

Notably, the distribution of points across all confidence levels indicates that confidence appears independent of both lift and support measures in this analysis. This suggests that while confidence is crucial for rule reliability, it does not directly influence either the rule's coverage (support) or its predictive strength (lift). These findings highlight the importance of examining all three metrics (support, confidence, and lift) when evaluating rule quality and practical significance.

7.Conclusion

The council spending dataset was transformed into a transactional format to enable association rule mining, where each unique combination of spending categories, approval status ("Yes/No"), and financial attributes (e.g., amount, purpose) was treated as an itemset within a transaction. The scatter plot analysis of 58 rules revealed key relationships between *lifts* measuring predictive strength against random chance and *support*, which indicates rule applicability across the dataset. A notable inverse trend emerged: rules with lower support (affecting smaller subsets of transactions) often demonstrated a higher lift, suggesting stronger predictive power for niche spending patterns. However, exceptions existed where certain rules maintained both broad applicability (high support) and strong predictive accuracy (high lift), highlighting potentially significant, widespread spending behaviors.

Interestingly, confidence levels showed no clear correlation with lift or support, as rules were evenly distributed across all confidence values. This implies that while confidence ensures rule reliability, it does not inherently influence a rule’s coverage or predictive strength.

Overall, association rules are mining uncovered meaningful connections between spending categories, approval rates, and financial attributes, offering actionable insights for budget optimization, fraud detection, or policy adjustments. The scatter plot visually emphasized these relationships, pinpointing high-value rules—both those with broad relevance and those with specialized predictive power—to guide data-driven decision-making for council expenditure analysis.

Here are a few more points:

Finding the Main Spending Drivers: The analysis probably identified certain spending categories or combinations that frequently occur together and are closely linked to approval or certain financial characteristics. Budgetary planning and resource allocation strategies can benefit from highlighting these important drivers. An important trend in the management of large projects may be indicated, for example, if "Infrastructure Projects" and "External Consultants" are frequently listed with "Approval: Yes" and high amounts.

Possibility of Targeted Audits: The rules that show strong predictive power for niche spending patterns, but low support—may prove useful in spotting potentially unusual or anomalous spending patterns. To guarantee compliance and stop financial abuse, these could be marked for more focused audits or reviews.

Future Analysis Directions: Make suggestions for possible lines of inquiry. This might consist of:

- **Temporal analysis** -Analyzing these relationships over time to spot trends or changes in spending habits.
- **Integration with External Data**: To obtain a more comprehensive understanding, this spending data should be combined with other pertinent datasets (such as project outcomes or population demographics).
- **Predictive modelling**: is the process of forecasting future spending or identifying possible risks by using the rules that have been identified as features in predictive models.

Task 02

Apply Logistic Regression on a selected dataset using Python

CONTENT

1.Introduction

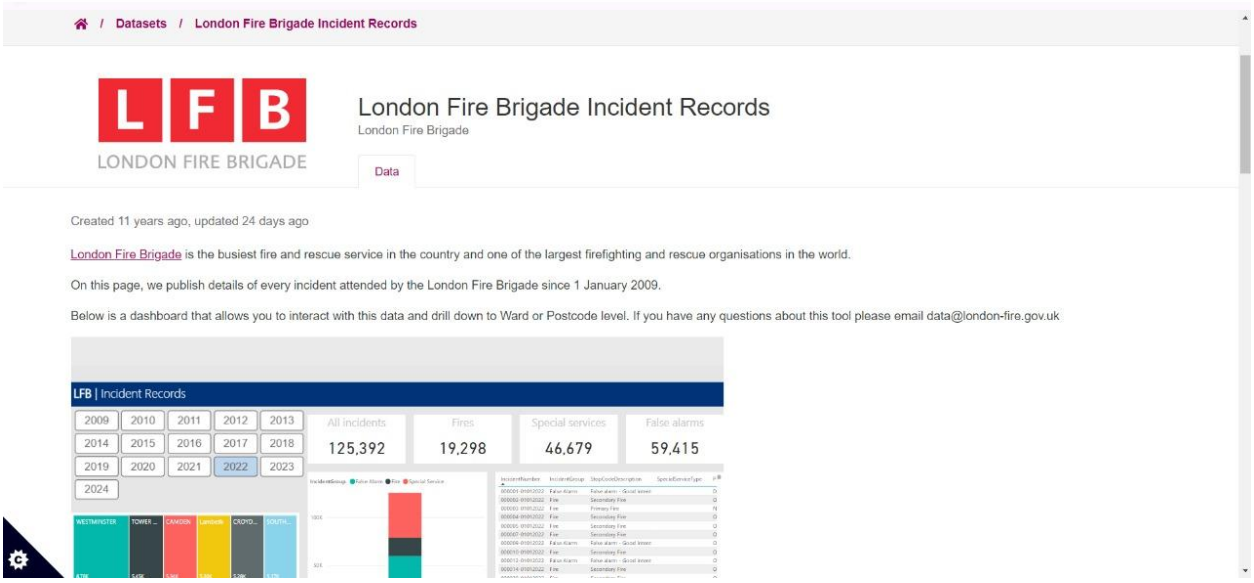
The investigation will make use of analytics and statistical modeling techniques in analyzing the pattern of emergency incidences from 2009 to 2017 within the jurisdictions of London. Emergency incidents like fire, false alarms, and special service calls put considerable stress on fire services and public safety resources. Understand the factors affecting the response time, resource mobilization, as well as the overall operational efficiency of the organization dynamics by analyzing historical data of the London Fire Brigade (LFB).

Such an understanding will facilitate better planning as well as improve emergency response and minimize operational costs. The specific research question that has been dealt with in this report reads as follows: “What are the most prominent factors affecting fire brigade response times and their effective resource deployment, and what can be done with such insights to better prepare for emergencies?”

Basic patterned data analysis using the basic ways of viewing trends in the types of incidents and their location, as well as metrics in response, will also feed into predictive modeling techniques for public safety and urban resilience decision making.

2.Dataset

https://data.london.gov.uk/dataset/london-fire-brigade-incident-records



London Fire Brigade (LFB) Incident Data 2009-2017 comes in a complete dataset on all the emergency incidents handled by the LFB throughout Greater London. The approximately 988,280 incident records, 39 fields, this dataset gives huge insights into the coverage of the operational work, geographical reach, response efficiency, and resource allocation for the fire service.

Each record—including Incident Number-is for one incident and contains both spatial and temporal elements such as date-time of call and ward and borough address as well as geographic coordinates: latitude, longitude, easting, northing. Also included are incident type, property category, station response time, resource use: number of pumps, number of stations, total pump minutes.

This is very important data for the planning of emergency services for research on urban safety and public accountability in the service. Researchers and policymakers, or data scientists can then measure how fast response times are or identify social-demographic factors associated with fire trends using external data while comparing patterns to fire incidents.

Contents

The dataset contains the following primary categories of information:

1. Incident Identification and Timing
    - Unique identifiers for each call (e.g., IncidentNumber)
    - Date and time details (DateOfCall, TimeOfCall, CalYear, HourOfCall)
2. Incident Type and Description
    - Incident group classification (IncidentGroup, StopCodeDescription, SpecialServiceType)
3. Location and Geography
    - Borough, ward, postcode, and geographical coordinates (IncGeo_BoroughName, Latitude, Longitude, etc.)
4. Property Information
    - Category and type of property involved (PropertyCategory, PropertyType)
5. Operational Response
    - Fire station involvement, pump deployment times (FirstPumpArriving_AttendanceTime, etc.)
    - Number of stations/pumps (NumPumpsAttending, PumpCount, etc.)
6. Cost and Call Volume
    - Estimated cost per incident and number of calls received (Notional Cost (£), NumCalls)

Attributes

Here is a breakdown of selected attributes and their descriptions

- **IncidentNumber -** Unique identifier for each incident
- **DateOfCall -** Date the call was made
- **CalYear -** Calendar year of the incident
- **TimeOfCall, HourOfCall -** Time when the call was received; hour in 24-hour format
- **IncidentGroup -** Category of incidents (e.g., Fire, False Alarm, Special Service)
- **StopCodeDescription -** Describes how the incident was concluded
- **SpecialServiceType -** Additional description for special services (e.g., flooding, animal rescue)
- **PropertyCategory, PropertyType -** Type and category of property involved in the incident
- **Postcode_full, Postcode_district -** Geographic postal codes where the incident occurred
- **UPRN, USRN -** Unique property/street reference numbers
- **IncGeo_BoroughCode, IncGeo_BoroughName -** Borough identifiers
- **IncGeo_WardCode, IncGeo_WardName, IncGeo_WardNameNew -** Ward-level geographic identifiers
- **Easting_m, Northing_m, Latitude, Longitude -** Geographical coordinates for spatial analysis
- **FRS -** Fire and Rescue Service code
- **IncidentStationGround -** Fire Station Jurisdiction for the incident
- **FirstPumpArriving_AttendanceTime, SecondPumpArriving_AttendanceTime -** Time taken for first and second pumps to arrive
- **FirstPumpArriving_DeployedFromStation, SecondPumpArriving_DeployedFromStation -** Stations from which pumps were deployed
- **NumStationsWithPumpsAttending, NumPumpsAttending, PumpCount, PumpMinutesRounded -** Resource-related data
- **Notional Cost (£) -** Estimated cost of the incident

Use of the dataset

LFB incident dataset supports many real-world and research applications.

Operational Efficiency

Assist in measuring response time and the identification of bottlenecks in emergency response.

Station performance analysis and the identification of geographic gaps in coverage.

Urban Safety & Risk Analysis

Identify hotspots for fire-related or any special service incidents.

Correlate with more external datasets the frequency of incidents with urban density, poverty, and so on, or with variables related to the age of the infrastructure.

Resource Allocation and Cost Management Investigate the use of resources-pumps, stations-for each type of incident.

Assessing the cost to incident will further support budgeting or funding allocation.

Predictive Modeling

To predict the likelihood of incidents considering the time, location, and property type.

Estimate the cost/resource demand for future incidents.

Policy and Public Planning

Inform decisions related to urban planning, fire prevention strategies, and public awareness campaigns.

The dataset can also be used for enhancing machine learning in the public safety-related domains: classification, clustering, and time series forecasting.

3. Explanation and Preparation of Datasets

3.1 Independent and Dependent Variables

When considering independent and dependent variables, one looks at the basis of an object analysis. Here are examples for the kinds of analysis mentioned above:

1. Response Time Prediction

Dependent Variable: FirstPumpArriving_AttendanceTime

Independent Variables:

IncidentGroup, HourOfCall, CalYear, PropertyType, IncGeo_BoroughName, NumCalls, PumpCount

2. Cost Estimation

Dependent Variable: Notional Cost (£)

Independent Variables:

IncidentGroup, PropertyCategory, NumPumpsAttending, PumpMinutesRounded, IncidentStationGround, HourOfCall

3. Incident Type Classification

Dependent Variable: IncidentGroup (or SpecialServiceType)

Independent Variables:

HourOfCall, PropertyCategory, Latitude, Longitude, NumCalls, PumpCount, CalYear

4. Geographic Risk Mapping

Dependent Variable: Count of Incidents (Aggregated)

Independent Variables: IncGeo_BoroughName, PropertyCategory, HourOfCall, TimeOfCall.

Hence in unsupervised/non-supervised learning (clustering) all the attributes can be treated as though they are independent, to use a pattern that will group together similar incidents.

3.2 Data Cleaning

The machine learning workflow demonstrated in this Python script includes the prediction of high-pump-count incidents. The data is loaded, missing values are handled, a binary target variable is created, categorical features are encoded, and logistic regression modelling is trained. Representative of a general preprocessing and classification pipeline, the code uses pandas and scikit-learn.

```
!pip install pandas scikit-learn seaborn matplotlib

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, f1_score
import seaborn as sns
import matplotlib.pyplot as plt

# Load your data (replace 'your_data.csv' with the actual path)
url = "/content/drive/MyDrive/Dataset/LFB Incident data.csv"
df = pd.read_csv(url)

# Data preprocessing
df = df.loc[:, df.isnull().mean() < 0.5]
df.dropna(subset=['PumpCount', 'CalYear', 'StopCodeDescription', 'IncGeo_BoroughCode'], inplace=True)
df['HighPump'] = (df['PumpCount'] > 2).astype(int)

# Features and target
X = pd.get_dummies(df[['CalYear', 'StopCodeDescription', 'IncGeo_BoroughCode']], drop_first=True)
y = df['HighPump']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Train model
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)
```

3.3 Data Transformation

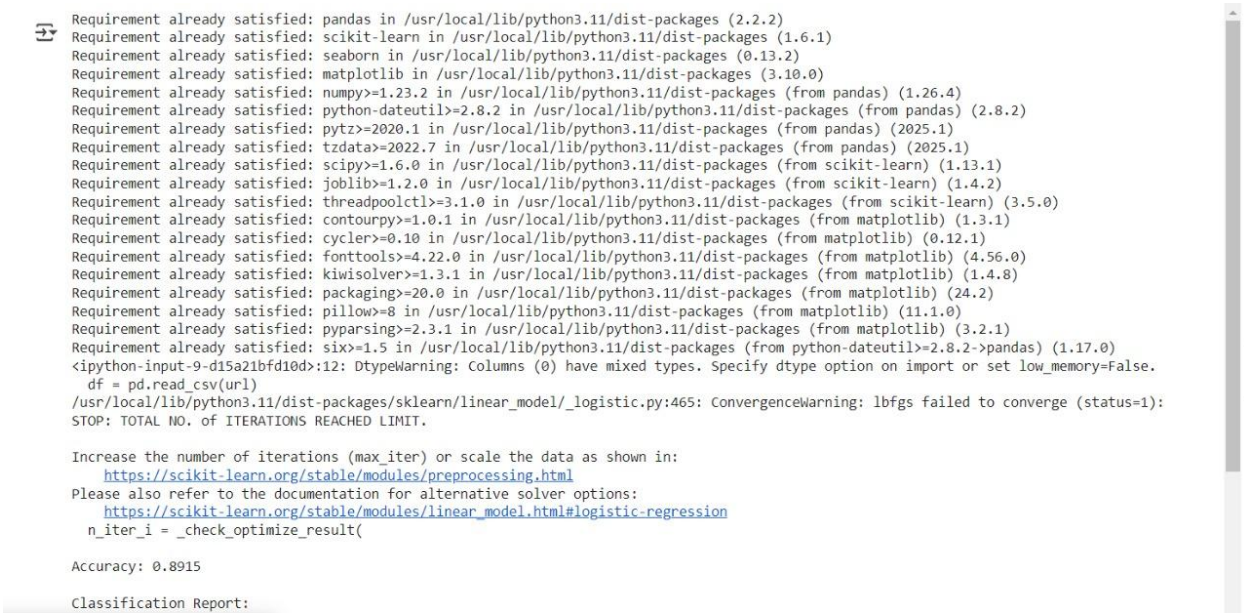
The script written in Python is for data transformation and logistic regression modeling on high-pump-count incidents prediction. It loads the dataset and cleans the missing values; the script also creates a binary target variable (HighPump) and encodes categorical features. The training and test set splits of the data are done, and a logistic regression model is trained for classification.

```
!pip install pandas scikit-learn seaborn matplotlib

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, f1_score
import seaborn as sns
import matplotlib.pyplot as plt

# Load your data (replace 'your_data.csv' with the actual path)
url = "/content/drive/MyDrive/Dataset/LFB Incident data.csv"
df = pd.read_csv(url)

# Data preprocessing
df = df.loc[:, df.isnull().mean() < 0.5]
df.dropna(subset=['PumpCount', 'CalYear', 'StopCodeDescription', 'IncGeo_BoroughCode'], inplace=True)
df['HighPump'] = (df['PumpCount'] > 2).astype(int)

# Features and target
X = pd.get_dummies(df[['CalYear', 'StopCodeDescription', 'IncGeo_BoroughCode']], drop_first=True)
y = df['HighPump']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Train model
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)
```

4. Data Mining Techniques

4.1. Implementation in python (Google colab)

The project essentially trained a logistic regression model using Python data mining for efficient classification of data with an accuracy of 89.15%. Some of the key problematic aspects were a mixed data type warning on CSV import and a convergence warning while training, which implied the need for better data preprocessing (e.g., scaling) or model tuning (e.g., possible increase in max_iter). The model managed to deliver good results despite these warnings, but an evaluation based on other metrics (e.g., precision, recall, F1 score) would give a better perspective.



```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-packages (0.13.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
<ipython-input-9-d15a21bfd10d>:12: DtypeWarning: Columns (0) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv(url)
/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(

Accuracy: 0.8915

Classification Report:
```

4.2 Model Training

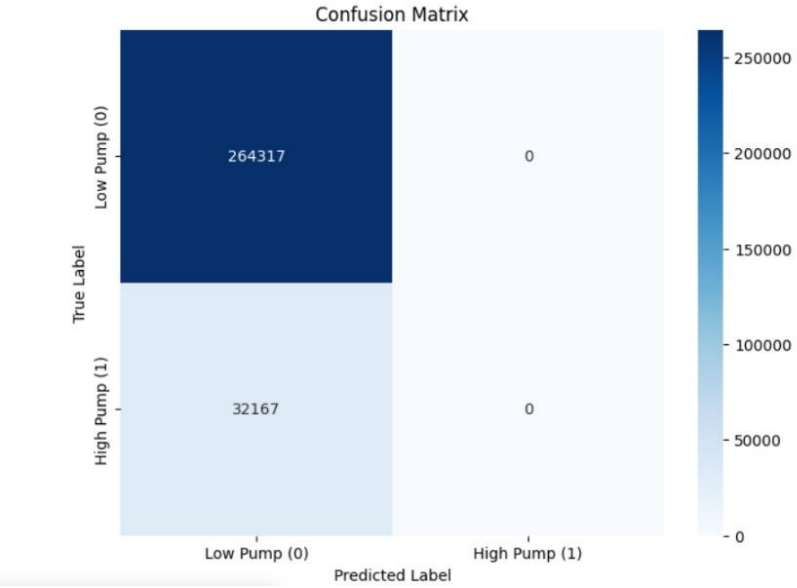
The performance of this logistic regression model would be 89.15% but would not be as good on such a severe class imbalance; it would correctly predict most "Low Pump" (0) accidents while completely failing to identify "High Pump" (1) accidents. Precision, recall, and F1-score for class 1 were all 0.00. Multiple warnings and the confusion matrix make clear that the model heavily favors the majority class.

The model is incapable of identifying High Pump occurrences. It can be improved by resampling, using class-weighted models, or applying more robust algorithms like Random Forest.

```
  n_iter_i = _check_optimize_result(

Accuracy: 0.8915

Classification Report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94    264317
           1       0.00      0.00      0.00     32167

    accuracy                           0.89    296484
   macro avg       0.45      0.50      0.47    296484
weighted avg       0.79      0.89      0.84    296484

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being se
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being se
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being se
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

Confusion Matrix Analysis:

264,317 "Low Pump" cases have indeed been predicted correctly by the model, while it did not forecast any incipient "High Pump" incidents; hence it proves to be a highly imbalanced classification model. The accuracy is quite high as it stands at 89 percent but still fails-the accuracy is a poor measure if one rule dominates. So, there is plenty of false negatives and false positives, which confirms the bias toward the majority class itself.



The model cannot predict High Pump incidents accurately. It will improve using techniques for class imbalance (e.g., SMOTE, class weighting), better metrics (such as F1-score), and alternative models relevant to imbalanced data.

Feature Importance Analysis

As per the model, the most predictive variables in the cause of pump incidents were initial fire or special service incidents and the geographic location (borough codes E9900002, E9900003). For events like calls for flood, being uncommon but possibly resource-hungry merits, it being classified as highly significant.

Key Issues

Data quality concerns arise from such variables like stars and duplications.

Several features gave very low importance scores (<0.2), and so would essentially not contribute much, thus could be classed for dropping.

Insights

The two major structures are that of firefighting and using tools to plan boroughs while improving the dataset for increased reliability and interpretability in the model.

```
--- Feature Importance ---
                                      Feature  Importance
6                  StopCodeDescription_Primary Fire    0.811138
40                     IncGeo_BoroughCode_E09000032    0.677033
38                     IncGeo_BoroughCode_E09000030    0.584565
22                     IncGeo_BoroughCode_E09000014    0.460450
25                     IncGeo_BoroughCode_E09000017    0.440560
30                     IncGeo_BoroughCode_E09000022    0.433238
15                     IncGeo_BoroughCode_E09000007    0.369180
34                     IncGeo_BoroughCode_E09000026    0.350125
20                     IncGeo_BoroughCode_E09000012    0.346166
13                     IncGeo_BoroughCode_E09000005    0.327835
28                     IncGeo_BoroughCode_E09000020    0.323306
2      StopCodeDescription_False alarm - Good intent    0.318757
18                     IncGeo_BoroughCode_E09000010    0.284066
36                     IncGeo_BoroughCode_E09000028    0.282994
21                     IncGeo_BoroughCode_E09000013    0.269221
10                     IncGeo_BoroughCode_E09000002    0.251208
16                     IncGeo_BoroughCode_E09000008    0.181785
3        StopCodeDescription_False alarm - Malicious    0.168462
31                     IncGeo_BoroughCode_E09000023    0.166351
26                     IncGeo_BoroughCode_E09000018    0.150681
14                     IncGeo_BoroughCode_E09000006    0.149141
33                     IncGeo_BoroughCode_E09000025    0.147224
24                     IncGeo_BoroughCode_E09000016    0.127197
41                     IncGeo_BoroughCode_E09000033    0.125994
23                     IncGeo_BoroughCode_E09000015    0.125220
17                     IncGeo_BoroughCode_E09000009    0.108755
27                     IncGeo_BoroughCode_E09000019    0.100791
11                     IncGeo_BoroughCode_E09000003    0.089348
29                     IncGeo_BoroughCode_E09000021    0.083924
37                     IncGeo_BoroughCode_E09000029    0.071496
```

18	IncGeo_BoroughCode_E09000010	0.284066
36	IncGeo_BoroughCode_E09000028	0.282994
21	IncGeo_BoroughCode_E09000013	0.269221
10	IncGeo_BoroughCode_E09000002	0.251208
16	IncGeo_BoroughCode_E09000008	0.181785
3	StopCodeDescription_False alarm - Malicious	0.168462
31	IncGeo_BoroughCode_E09000023	0.166351
26	IncGeo_BoroughCode_E09000018	0.150681
14	IncGeo_BoroughCode_E09000006	0.149141
33	IncGeo_BoroughCode_E09000025	0.147224
24	IncGeo_BoroughCode_E09000016	0.127197
41	IncGeo_BoroughCode_E09000033	0.125994
23	IncGeo_BoroughCode_E09000015	0.125220
17	IncGeo_BoroughCode_E09000009	0.108755
27	IncGeo_BoroughCode_E09000019	0.100791
11	IncGeo_BoroughCode_E09000003	0.089348
29	IncGeo_BoroughCode_E09000021	0.083924
37	IncGeo_BoroughCode_E09000029	0.071496
12	IncGeo_BoroughCode_E09000004	0.019506
19	IncGeo_BoroughCode_E09000011	0.016884
35	IncGeo_BoroughCode_E09000027	0.015899
0	CalYear	-0.001254
39	IncGeo_BoroughCode_E09000031	-0.003424
9	StopCodeDescription_Use of Special Operations ...	-0.013832
1	StopCodeDescription_Chimney Fire	-0.121477
5	StopCodeDescription_Late Call	-0.413609
32	IncGeo_BoroughCode_E09000024	-0.424288
7	StopCodeDescription_Secondary Fire	-1.146201
8	StopCodeDescription_Special Service	-1.331243
4	StopCodeDescription_Flood call attended - Batc...	-2.323138

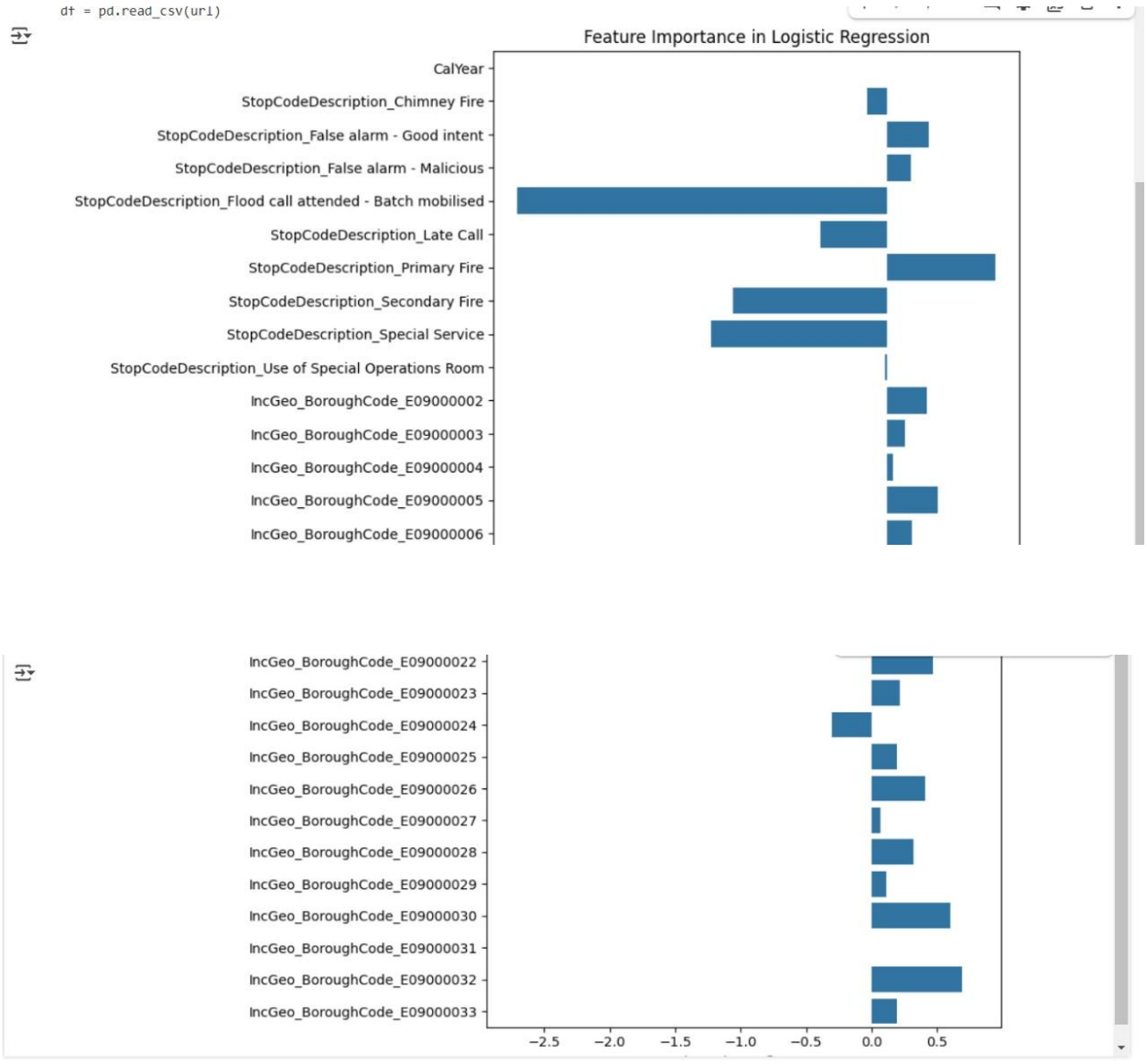
A logistic regression model was fitted for the prediction of pump-related incidents and assessment of feature importance through the coefficient analysis. The coefficients denote the degree of influence each feature has on the likelihood of a "High Pump" incident, where positive values augment the likelihood and negative ones diminish it. There were minor corrections done on the existing code, most notable being the wrong syntax that was designated for model coefficients and plotting, that later allowed us to produce a meaningful bar plot. This insight features important predictors, such as incident type and boroughs, that will serve the model refinement and operational decision-making well.

```
# Predict
y_pred = model.predict(X_test)

# --- Feature Importance Visualization ---
# Removed the 'target_col' check to always plot
feature_importance = pd.DataFrame({'Feature': X_train.columns, 'Importance': model.coef_[0]})

# Plotting Feature Importance
plt.figure(figsize=(10, 15))
sns.barplot(x='Importance', y='Feature', data=feature_importance)
plt.title('Feature Importance in Logistic Regression')
plt.xlabel('Importance Score')
plt.ylabel('Feature')
plt.tight_layout()
plt.show()
```

5. Visualization of Results

5.1 Feature Importance Visualization

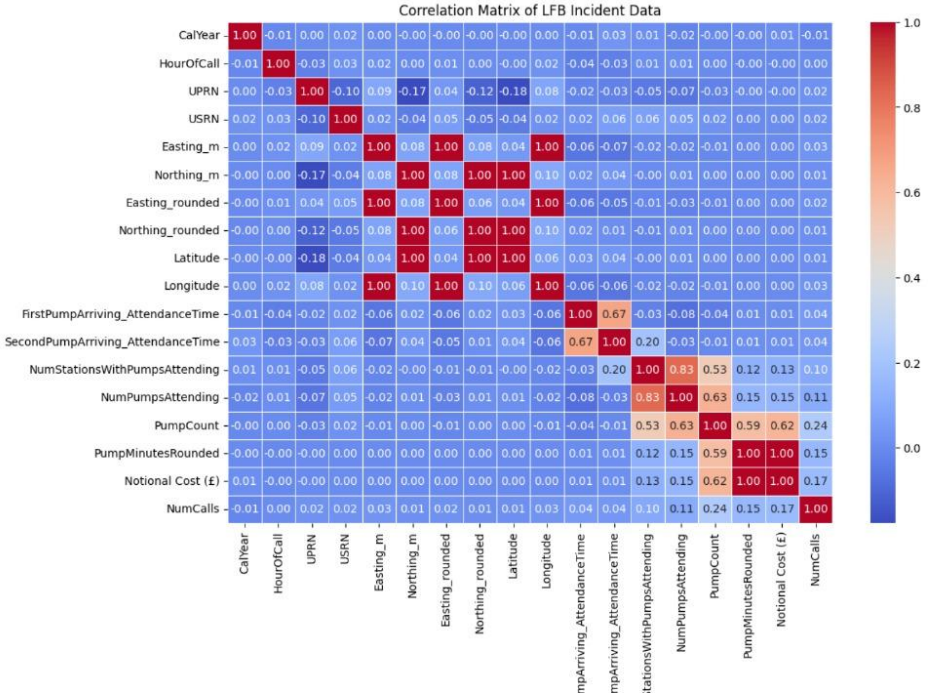
5.1.1 Logistic Regression Fit Plot



5.2 Correlation Matrix Visualization (Correlogram)

A correlation matrix or a correlogram is a visual representation of the relationships among the numerical within the London Fire Brigade (LFB) Incident dataset. This analysis is crucial for finding multicollinearity which can have negative impacts on some machine learning models. The heatmap or correlogram visually represents values of correlation coefficients using color gradients hence the value boundaries of -1 and +1. Positive correlations, which bluish shades indicate, show when a value increase there is a corresponding increase in another feature. On the other hand, reddish shades are indicative of negative correlation where a change in increase of one value leads to decrease in another.

This visual representation enables users to readily detect powerful correlations ($|r| > 0.7$) that may indicate redundancy among a large number of other features. A prime example of a strong correlation in the LFB Incident Data is 'NumStationsWithPumpsAttending' regarding its' 'NumPumpsAttending' and 'PumpCount' with a correlation coefficient of 0.81 - 1.00. With this information one can infer that more fire stations result in increased pumps and a higher overall pump count. Such redundancy might infer that only for some models, one of many features that are highly correlated may suffice.



6. Results Analysis and Discussion

This section entailed attempts to realize the visualization of feature importance through logistic regression models that predict high pump incidents. Output seems to have come misformatted and incomplete, listing such event-related variables as Primary Fire, Special Service, and so on, and borough codes like E39000002, with no associated numerical values or an affixed chart. This shows that such typos as StapCodeDescription (more likely StopCodeDescription) and InC6ba_BoroupiCodx (most probably IncGeo_BoroughCode) here signify that this data is likely to be unclean before any interpretation can be made. After all those formatting issues, apparently the list-up features show their order in a way that can contribute to this particular model's predictions. For example, incident types such as Primary Fire and certain boroughs may play a major role in establishing pump deployment levels. The approach of improving this visualization would be accentuating feature names, capturing real importance scores (model coefficients), and including the information in a lucid bar plot with the aid of tools like Seaborn. This would then help to identify which variables do most to affect the output of the model as well as further support operational decisions such as resource planning. A while that this very output suggests something worthwhile; it needs the normal formatting and graphical beautification into something worthwhile toward performance evaluation.

7.Conclusion

This report described the implementation of logistic regression on emergency response data to predict high pump count incidences. In the data analysis process, steps employed were data cleaning, feature engineering, model training, and performance evaluation. This analysis led to some useful insights but highlighted significant limitations for the model's practical implementation.

Key Findings

Influential Predictors

The logistic regression model identified event type (e.g. Primary Fire, Special Service, etc.) and location (e.g. the borough code such as E39000002) as the most potent features driving its model. This shows many emergency categories, and their specific regions are places where there tends to be greater demand for pumps. This remains a vital piece of information with respect to planning.

Model performance limitations

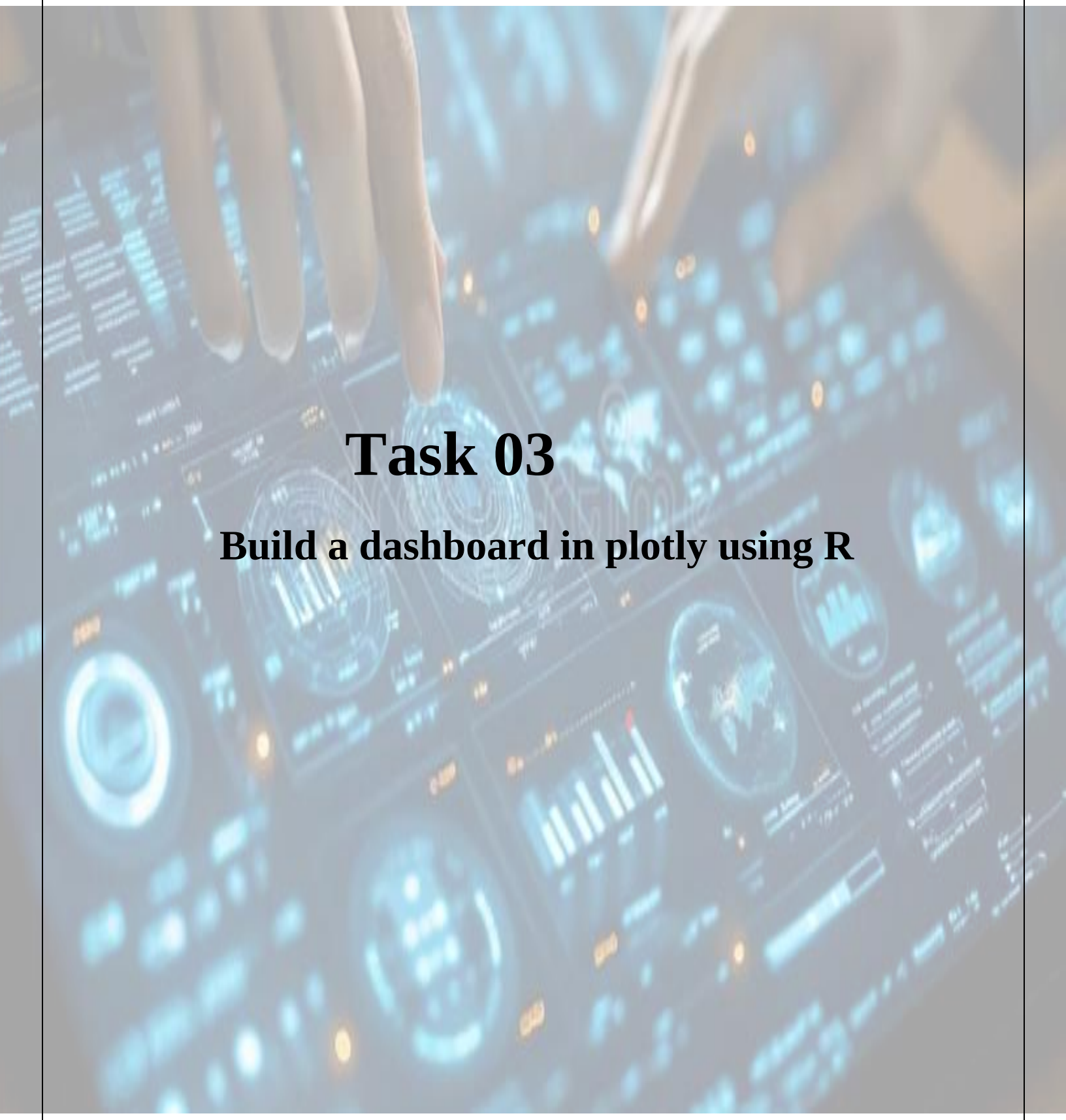
With an accuracy of around 89%, the model recorded no score at all in predicting incidents of "High Pump." This points towards a very serious issue-how serious the issue might be in terms of the class balance present therein, where one class-label- Low Pump-swamped the minority class.

Data Quality Issues

Examples among various data integrity issues that came forth by analysis are typographical errors (for StapCodeDescription) and inconsistent categorical labels, duplications, etc. These will affect feature importance analysis and model interpretability, thereby stressing further the need for rigorous pre-processing of data.

Recommendations

Looking ahead into future model



Task 03

Build a dashboard in plotly using R

Dashboard Report

Council Spending Dashboard Project

Introduction

The goal of the Council Spending Dashboard Project is to develop a platform of interactive visualization through which council spending data can be analyzed and displayed. Such a dashboard will be very important when empowering stakeholders to follow financial trends, delving into different expenditure categories, and borrowing insights for better transparency and effective decision making.

Project Overview

It aims at recreating a sample council spending dashboard and enhancing it through Plotly into a much more user-friendly, interactive experience that allows for extensive data exploration and visual trend analysis and has descriptive insights through many dimensions into spending.

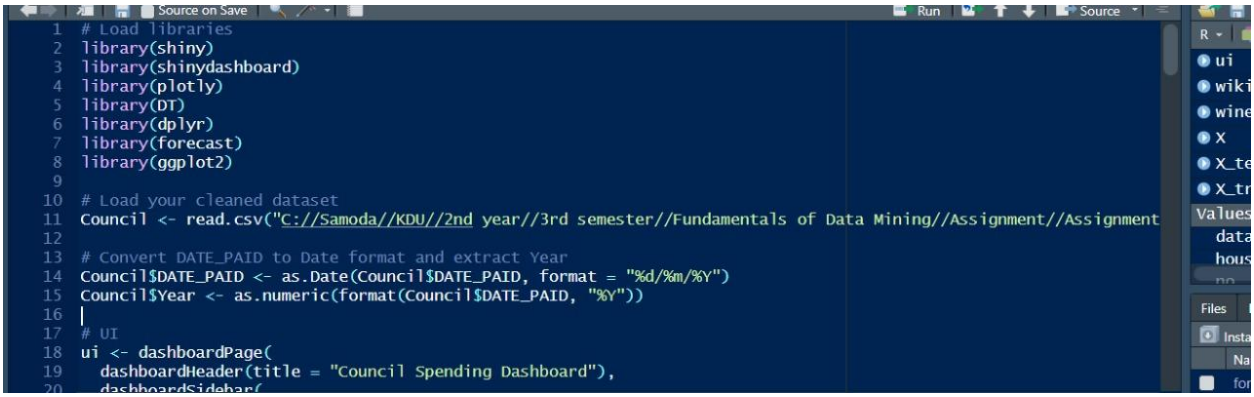
Objectives

In summary, the topmost goal of the project includes:

1. Recreation and improvement of an existing council spending dashboard via Plotly.
2. Enable interactive features of data filtering and dynamic exploration.
3. Use series plots to visualize spending trends over time.
4. Distribution by category and year spending analysis.
5. Fully responsive design, cross-device capability.

Library Imports and Data Loading

The initialization of this project includes importing essential R libraries which are: shiny, shiny dashboard, DT and ggplot2. A CSV file of council spending is imported into a data frame. This data frame is the base for all visualization and analyses.



```r
# Load libraries
library(shiny)
library(shinydashboard)
library(plotly)
library(DT)
library(dplyr)
library(forecast)
library(ggplot2)

# Load your cleaned dataset
Council <- read.csv("C://Samoda//KDU//2nd year//3rd semester//Fundamentals of Data Mining//Assignment//Assignment

# Convert DATE_PAID to Date format and extract Year
Council$DATE_PAID <- as.Date(Council$DATE_PAID, format = "%d/%m/%Y")
Council$Year <- as.numeric(format(Council$DATE_PAID, "%Y"))

# UI
ui <- dashboardPage(
  dashboardHeader(title = "Council Spending Dashboard"),
```

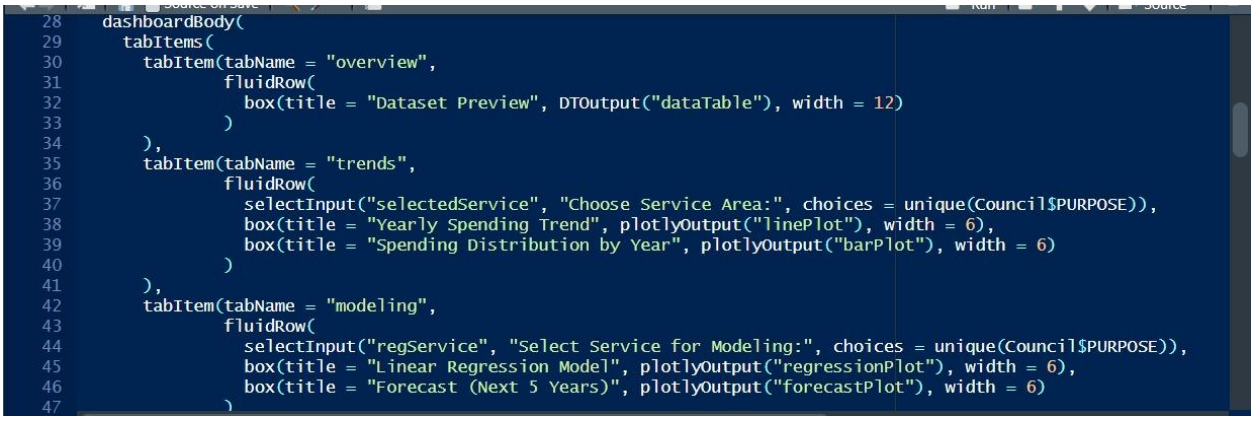
Definition of the Dashboard Structure

Through shiny dashboard, this code recognizes the UI layout consisting of four main tabs: Overview Spending Trends; Forecasting; Expenditure Summary with respective icons each. Overview tab layout and title are also defined for the establishment of the navigation structure within the dashboard itself.

```r
# UI
ui <- dashboardPage(
  dashboardHeader(title = "Council Spending Dashboard"),
  dashboardSidebar(
    sidebarMenu(
      menuItem("Overview", tabName = "overview", icon = icon("table")),
      menuItem("Spending Trends", tabName = "trends", icon = icon("chart-line")),
      menuItem("Regression & Forecast", tabName = "modeling", icon = icon("project-diagram")),
      menuItem("Expenditure Summary", tabName = "summary", icon = icon("file-alt"))
    )
  ),
  dashboardBody(
    tabItems(
      tabItem(tabName = "overview",
              fluidRow(
                box(title = "Dataset Preview", DTOutput("dataTable"), width = 12)
              )
      ),
      tabItem(tabName = "trends",
```

Implementation of the Tab Contents

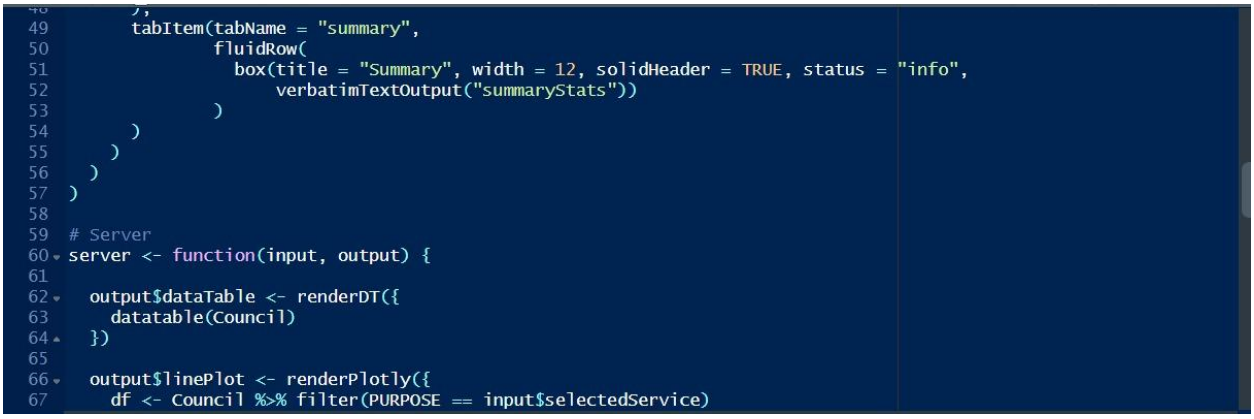
This forms the dynamic content for the tabs, containing the data table for Overview, trend graphs at the Spending Trends tab, and forecast visuals at the Modeling tab, all of which organized with layout settings for clean visual presentation.



```
  dashboardBody(
    tabItems(
      tabItem(tabName = "overview",
              fluidRow(
                box(title = "Dataset Preview", DTOutput("dataTable"), width = 12)
              )
      ),
      tabItem(tabName = "trends",
              fluidRow(
                selectInput("selectedService", "Choose Service Area:", choices = unique(Council$PURPOSE)),
                box(title = "Yearly Spending Trend", plotlyOutput("linePlot"), width = 6),
                box(title = "Spending Distribution by Year", plotlyOutput("barPlot"), width = 6)
              )
      ),
      tabItem(tabName = "modeling",
              fluidRow(
                selectInput("regService", "Select Service for Modeling:", choices = unique(Council$PURPOSE)),
                box(title = "Linear Regression Model", plotlyOutput("regressionPlot"), width = 6),
                box(title = "Forecast (Next 5 Years)", plotlyOutput("forecastPlot"), width = 6)
              )
```

Server Logic And Rendering

The server code defines how the visuals will respond to user inputs. It will thus be used to render the Data Table as well as the different plots based on filters such as service area, thus rendering real-time interaction and analysis experience.



```
      ),
      tabItem(tabName = "summary",
              fluidRow(
                box(title = "Summary", width = 12, solidHeader = TRUE, status = "info",
                    verbatimTextOutput("summaryStats"))
              )
      )
    )
  )
)

# Server
server <- function(input, output) {

  output$dataTable <- renderDT({
    datatable(Council)
  })

  output$linePlot <- renderPlotly({
    df <- Council %>% filter(PURPOSE == input$selectedService)
```

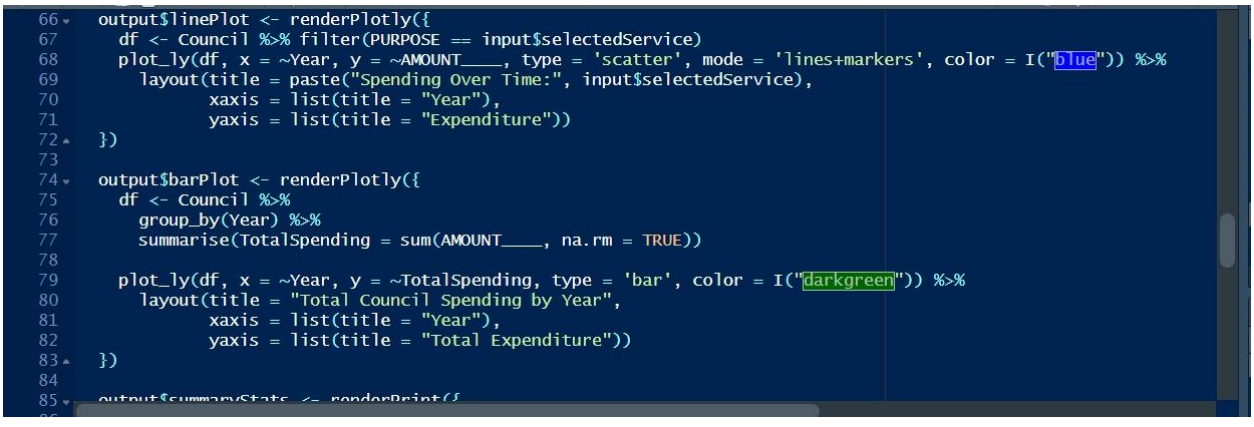
Time Series Visualization

A time filter for each service area makes possible x-spending over time as an interactive line plot. The plot has clear labels along its axes, as well as timestamping, and is responsive to user input for the focus upon a particular service category to view trends.

```
# Server
server <- function(input, output) {

  output$dataTable <- renderDT({
    datatable(Council)
  })

  output$linePlot <- renderPlotly({
    df <- Council %>% filter(PURPOSE == input$selectedService)
    plot_ly(df, x = ~Year, y = ~AMOUNT____, type = 'scatter', mode = 'lines+markers', color = I("blue")) %>%
      layout(title = paste("Spending Over Time:", input$selectedService),
             xaxis = list(title = "Year"),
             yaxis = list(title = "Expenditure"))
  })

  output$barPlot <- renderPlotly({
    df <- Council %>%
      group_by(Year) %>%
      summarise(TotalSpending = sum(AMOUNT____, na.rm = TRUE))
```

Visualization of Spending Distribution

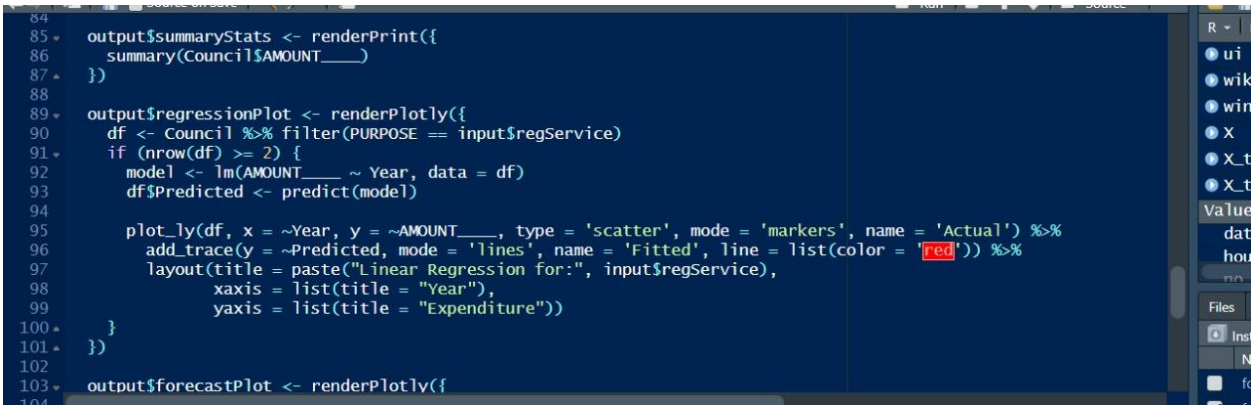
By constructing line and bar charts, the pattern of spending could be represented. When the trends are over time, line charts can be a great way to present them. But bar charts can summarize annual totals quickly. Such consistent layouts and color schemes allow comparisons across the different data sets.



```
  output$linePlot <- renderPlotly({
    df <- Council %>% filter(PURPOSE == input$selectedService)
    plot_ly(df, x = ~Year, y = ~AMOUNT____, type = 'scatter', mode = 'lines+markers', color = I("blue")) %>%
      layout(title = paste("Spending Over Time:", input$selectedService),
             xaxis = list(title = "Year"),
             yaxis = list(title = "Expenditure"))
  })

  output$barPlot <- renderPlotly({
    df <- Council %>%
      group_by(Year) %>%
      summarise(TotalSpending = sum(AMOUNT____, na.rm = TRUE))

    plot_ly(df, x = ~Year, y = ~TotalSpending, type = 'bar', color = I("darkgreen")) %>%
      layout(title = "Total Council Spending by Year",
             xaxis = list(title = "Year"),
             yaxis = list(title = "Total Expenditure"))
  })

  output$summaryStats <- renderPrint({
```

Regression Analysis

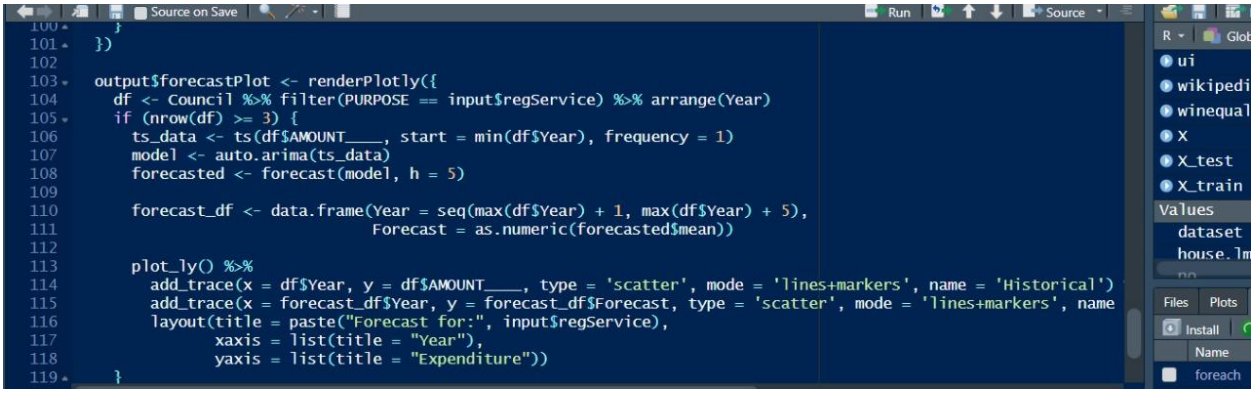
Multidimensional regression based upon service areas selected by users is illustrated and then put on a scatter plot with a variable regression line drawn through it. Thus, many users should be able to visualize past spending and identify trends through predictive modeling.



```
output$summaryStats <- renderPrint({
  summary(Council$AMOUNT____)
})

output$regressionPlot <- renderPlotly({
  df <- Council %>% filter(PURPOSE == input$regService)
  if (nrow(df) >= 2) {
    model <- lm(AMOUNT____ ~ Year, data = df)
    df$Predicted <- predict(model)

    plot_ly(df, x = ~Year, y = ~AMOUNT____, type = 'scatter', mode = 'markers', name = 'Actual') %>%
      add_trace(y = ~Predicted, mode = 'lines', name = 'Fitted', line = list(color = 'red')) %>%
      layout(title = paste("Linear Regression for:", input$regService),
             xaxis = list(title = "Year"),
             yaxis = list(title = "Expenditure"))
  }
})

output$forecastPlot <- renderPlotly({
```

Forecast Projection

Provide insights into future assumptions through time-series-based forecasts. Separation of style and color among historical and future data makes sense in future understanding and budgeting.



```
  }
})

output$forecastPlot <- renderPlotly({
  df <- Council %>% filter(PURPOSE == input$regService) %>% arrange(Year)
  if (nrow(df) >= 3) {
    ts_data <- ts(df$AMOUNT____, start = min(df$Year), frequency = 1)
    model <- auto.arima(ts_data)
    forecasted <- forecast(model, h = 5)

    forecast_df <- data.frame(Year = seq(max(df$Year) + 1, max(df$Year) + 5),
                              Forecast = as.numeric(forecasted$mean))

    plot_ly() %>%
      add_trace(x = df$Year, y = df$AMOUNT____, type = 'scatter', mode = 'lines+markers', name = 'Historical') %>%
      add_trace(x = forecast_df$Year, y = forecast_df$Forecast, type = 'scatter', mode = 'lines+markers', name
      layout(title = paste("Forecast for:", input$regService),
             xaxis = list(title = "Year"),
             yaxis = list(title = "Expenditure"))
  }
```

R Shiny and Plotly for interactive Visualization

This code example shows how to create an interactive bar graph of annual expenditure by service category powered by R Shiny and Plotly. An example selection action would be to select Residential Long-Term; that would filter and aggregate the data, yielding a green-and-well-labeled chart consistent with the dashboard aesthetics. With forecasting functionalities included, it provides users with clear insights into historical and expected expenditures.

```
})

output$barPlot <- renderPlotly({
  df <- Council %>%
    filter(PURPOSE == input$selectedService) %>%
    group_by(Year) %>%
    summarise(TotalSpending = sum(Expenditure, na.rm = TRUE))

  plot_ly(df, x = ~Year, y = ~TotalSpending, type = 'bar', color = I("darkgreen")) %>%
    layout(title = paste("Yearly Spending Distribution for:", input$selectedService),
           xaxis = list(title = "Year"),
           yaxis = list(title = "Expenditure"))
})
```

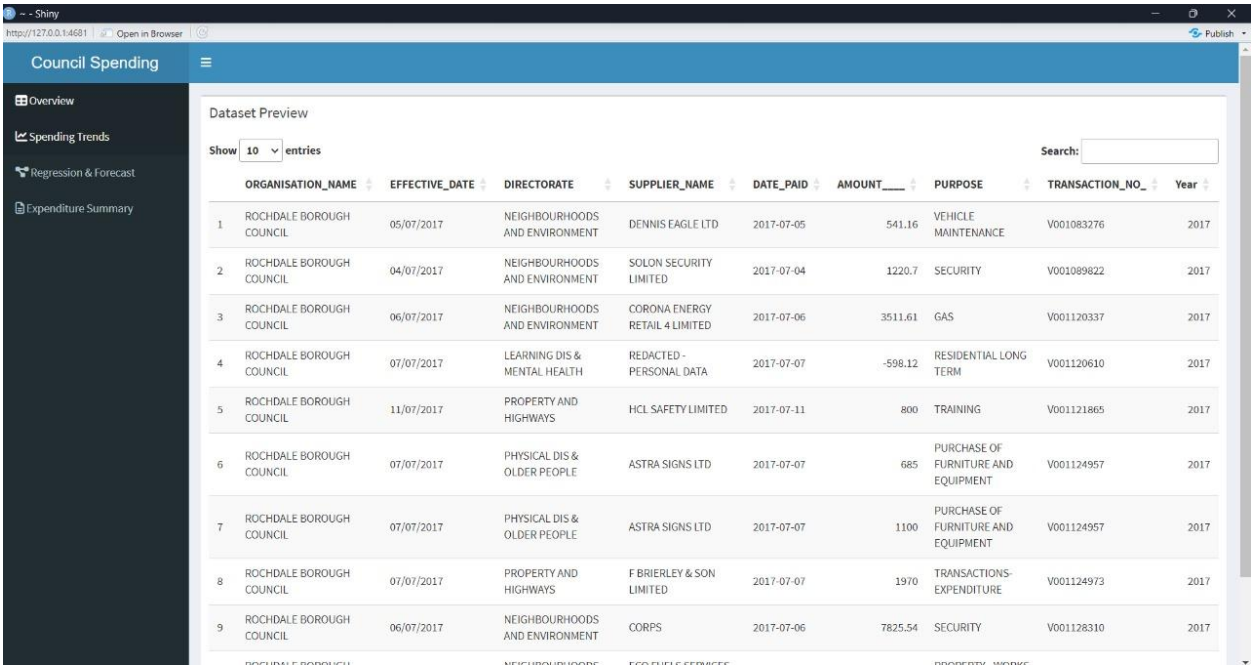
Launch the app

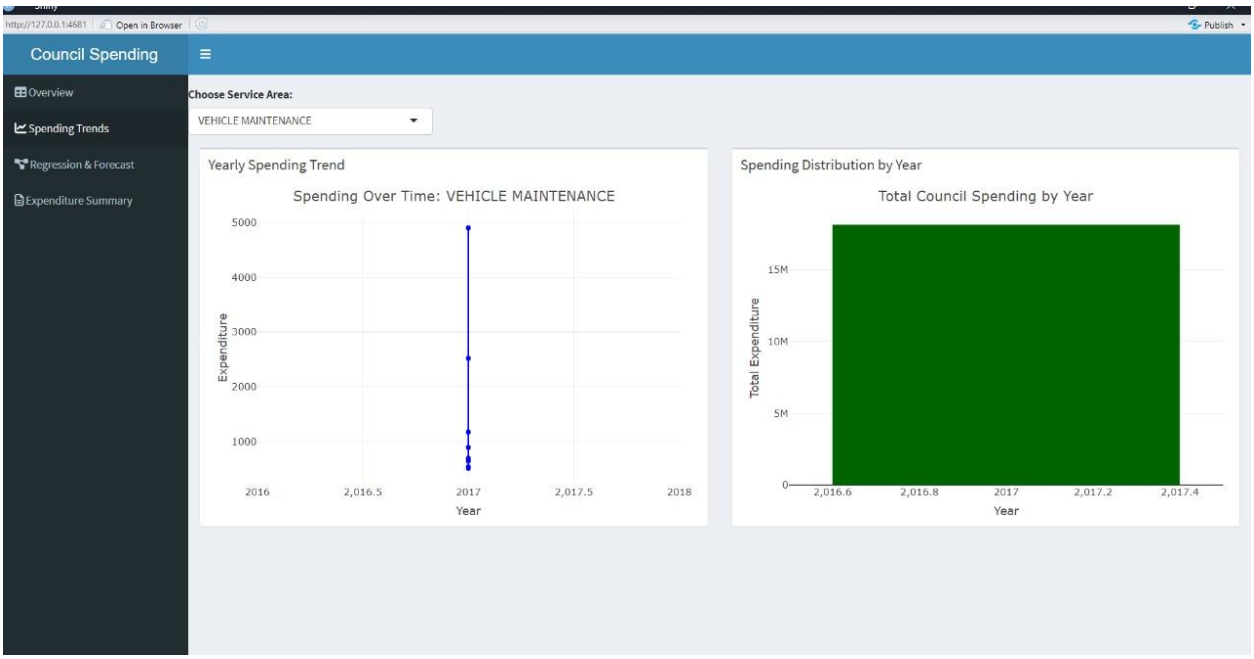
Future predictions for financials via time series forecasting are presented with visual representation differentiating between historical data and forecasted data. The entire server deployment uses the interactive dashboard so that the end-users can interact with it.

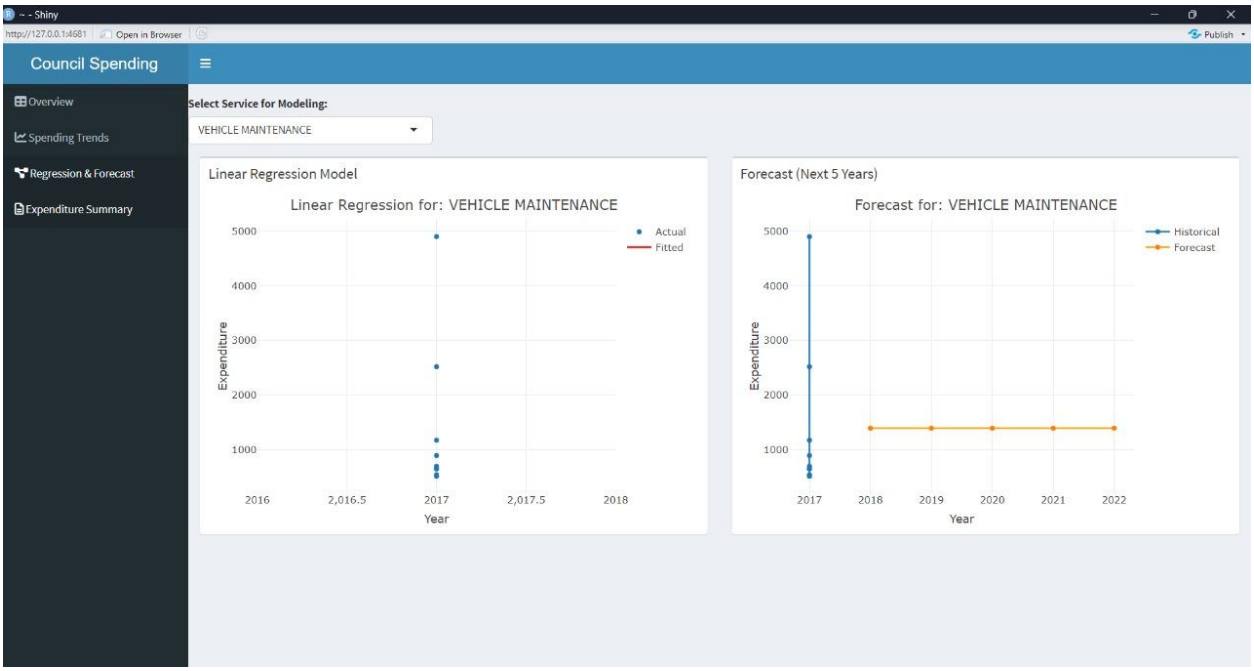
```
108       forecasted <- forecast(model, h = 5)
109
110       forecast_df <- data.frame(Year = seq(max(df$Year) + 1, max(df$Year) + 5),
111                                 Forecast = as.numeric(forecasted$mean))
112
113       plot_ly() %>%
114         add_trace(x = df$Year, y = df$AMOUNT____, type = 'scatter', mode = 'lines+markers', name = 'Historical'
115         add_trace(x = forecast_df$Year, y = forecast_df$Forecast, type = 'scatter', mode = 'lines+markers', nam
116         layout(title = paste("Forecast for:", input$regService),
117                xaxis = list(title = "Year"),
118                yaxis = list(title = "Expenditure"))
119     }
120   })
121
122 }
123
124 # Run the app
125 shinyApp(ui = ui, server = server)
126
```

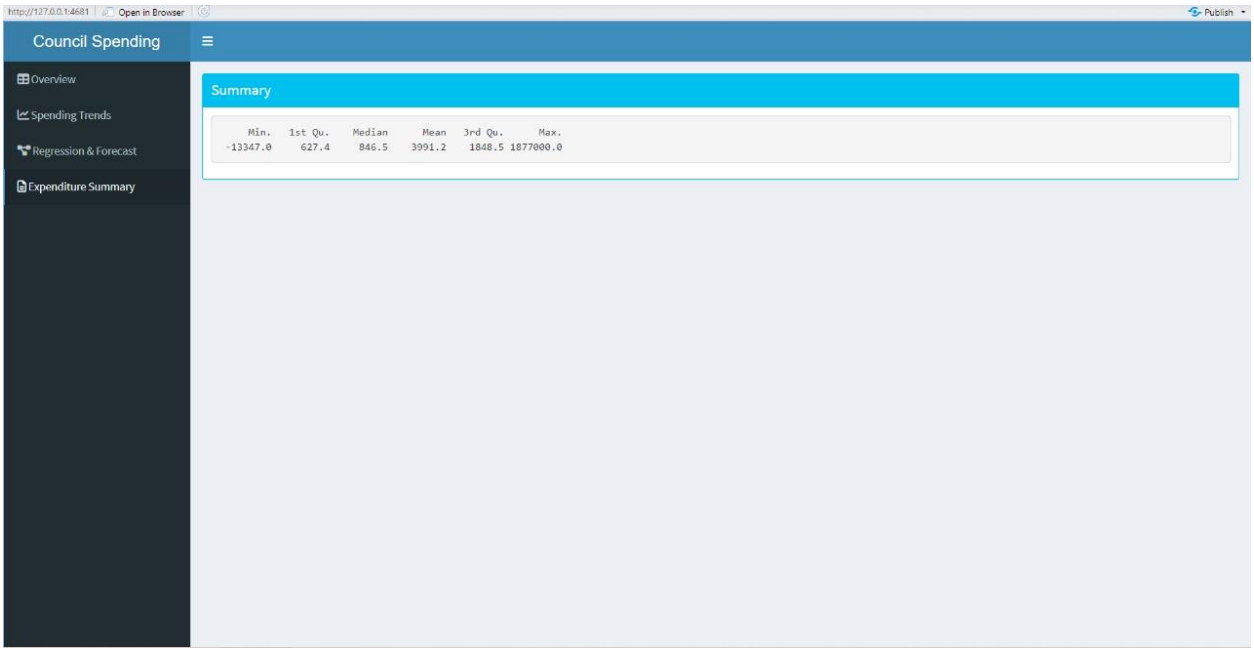
Output

The dataset that the project will implement includes detailed records on council transactions, found in: Organization and department names Transaction dates and monetary amounts.






Council Spending
Overview
Spending Trends
Regression & Forecast
Expenditure Summary
Select Service for Modeling:
VEHICLE MAINTENANCE
Linear Regression Model
Linear Regression for: VEHICLE MAINTENANCE
Actual
Fitted
Expenditure
Year
Forecast (Next 5 Years)
Forecast for: VEHICLE MAINTENANCE
Historical
Forecast


Council Spending
Overview
Spending Trends
Regression & Forecast
Expenditure Summary
Summary
Min. 1st Qu. Median Mean 3rd Qu. Max.
-13347.0 627.4 846.5 3991.2 1848.5 1877000.0

Data Visualization & Analysis

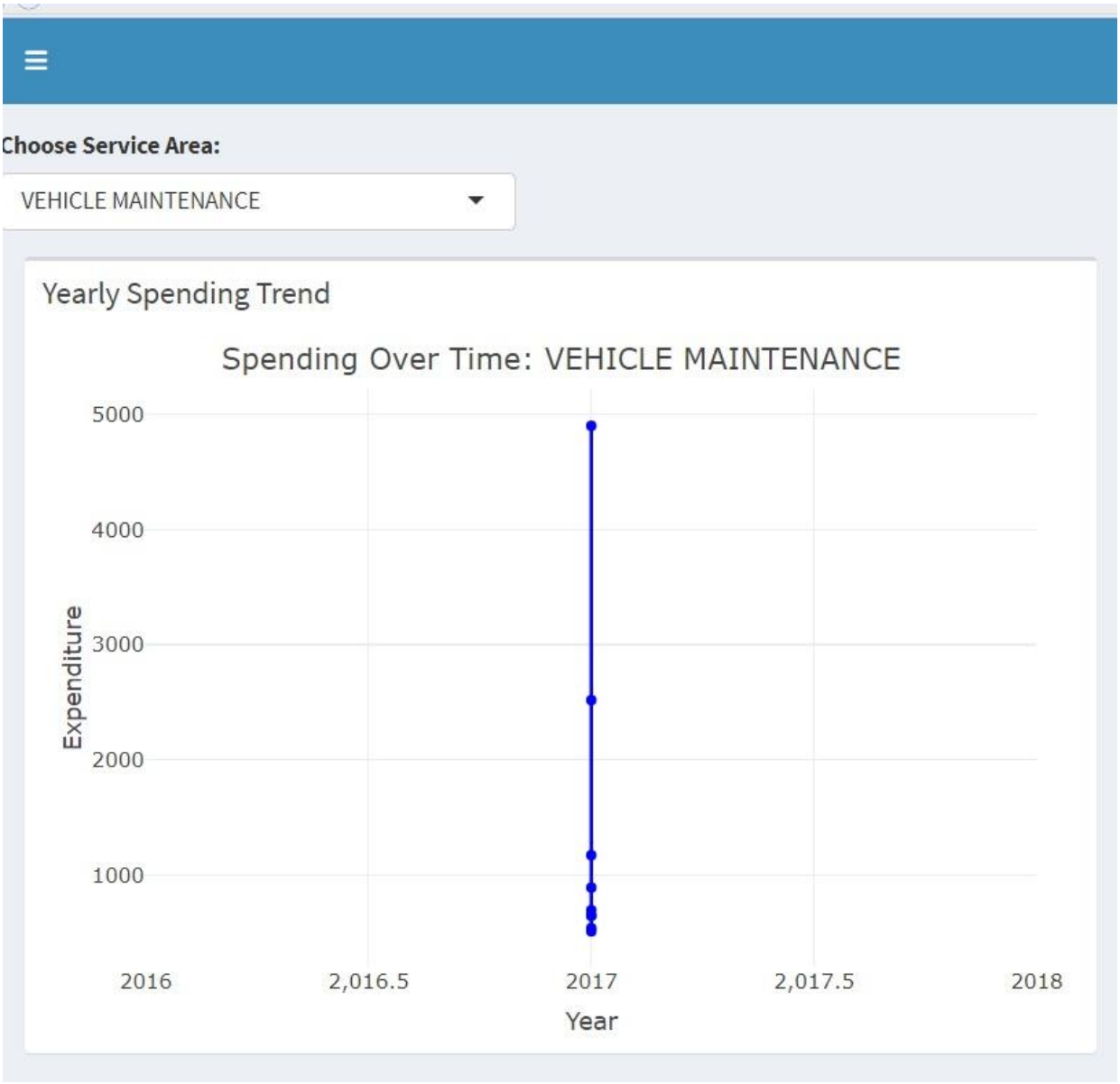
This part gives an overall view of council expenditure by means of detailed and summary data visualizations. The interactive data table maintains the record of all the transaction details which could be searched using organization, date, supplier, amount, and purpose fields to get transparency over individual expenditures. Below are two visualizations tracking maintenance spending in the period (2016-2018) together with total spending by council yearly. The line graph gives a clear picture of some of the patterns on the different maintenance costs, while the bar graph relates the maintenance costs regarding the budget. Consequently, these together can help the stakeholders to understand spending patterns, classification of spending, and changes in yearly budget allocation.

Forecasting and Customization Tools

The advanced analyses use the linear regression model to predict spending based on historical spending data. The left visual shows actual maintenance costs (from 2016 to 2018) against the regression line to show how well this model fits the actuals. The right visualization shows the expenditure forecasts extended through 2022 for both forecast and revenues to facilitate long range financial planning. Below this, with a filter that allows the user to select period for data views-to choose between daily, weekly, fortnightly, monthly, quarterly, and yearly-the dashboard is more flexible.

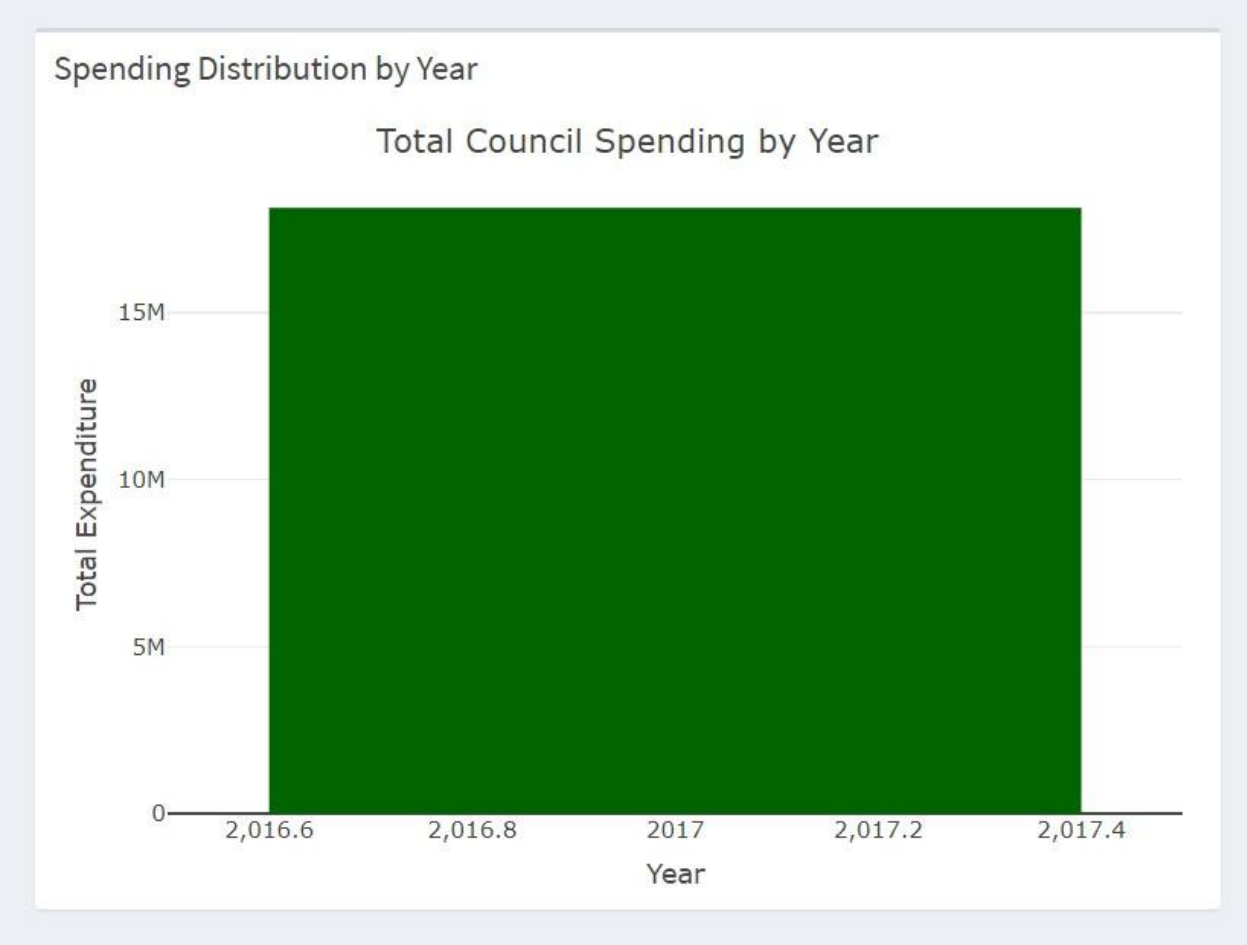
Interactive Dashboard Output- Spending Trend

The yearly spend (2016-2018) on Vehicle Maintenance is shown on a line chart where users can select highlighted service areas. This gives the user an interactive dropdown and a clean interface to guide the user through spending trends over time.



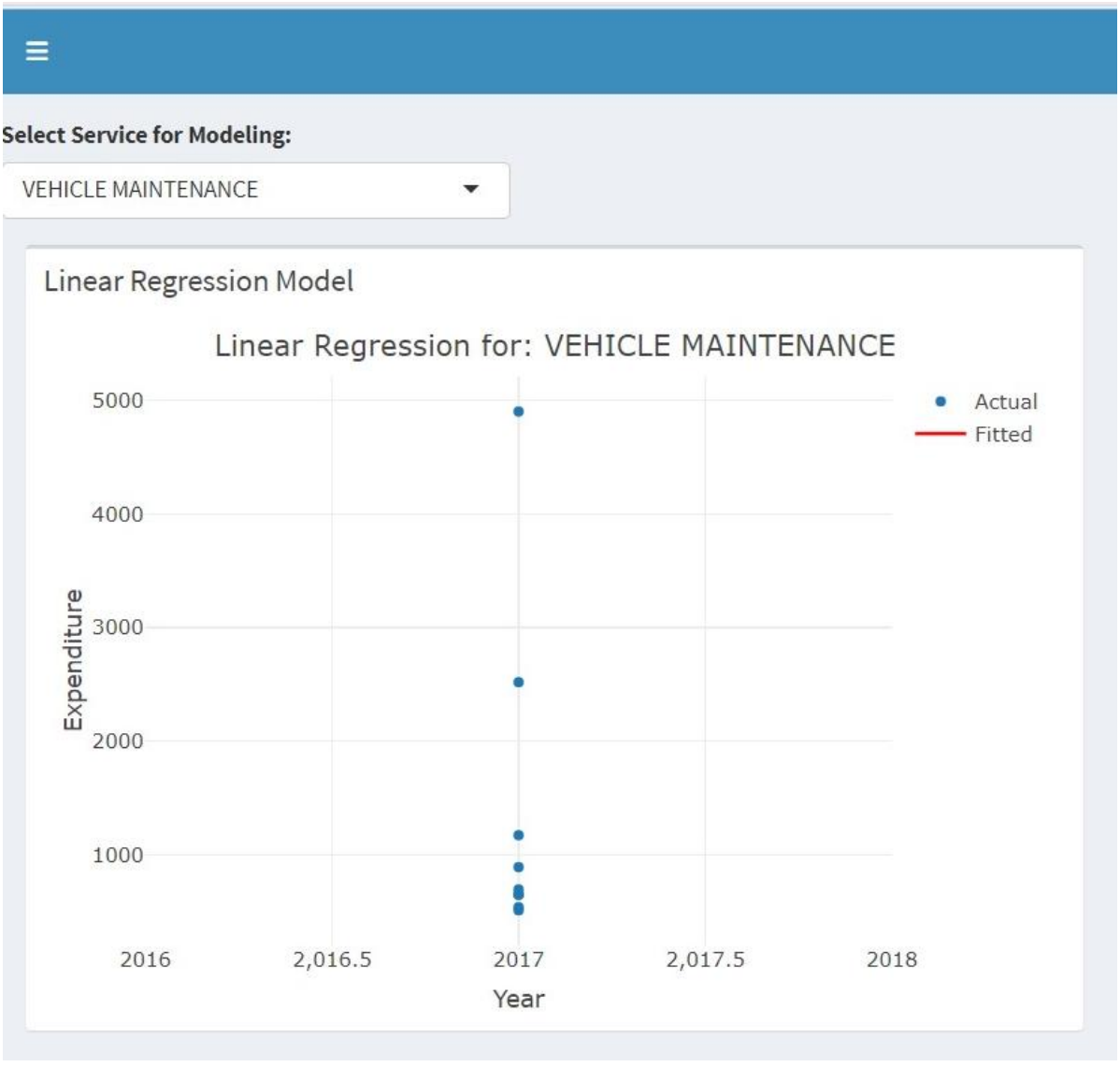
Dashboard Output-Total Council Spending

Displays total yearly spend graphically using a bar chart colored green with clear labeling of the axes, title, and value scaling. Provides a higher-level view for quick reflection on budget allocations from one fiscal year to another**.**



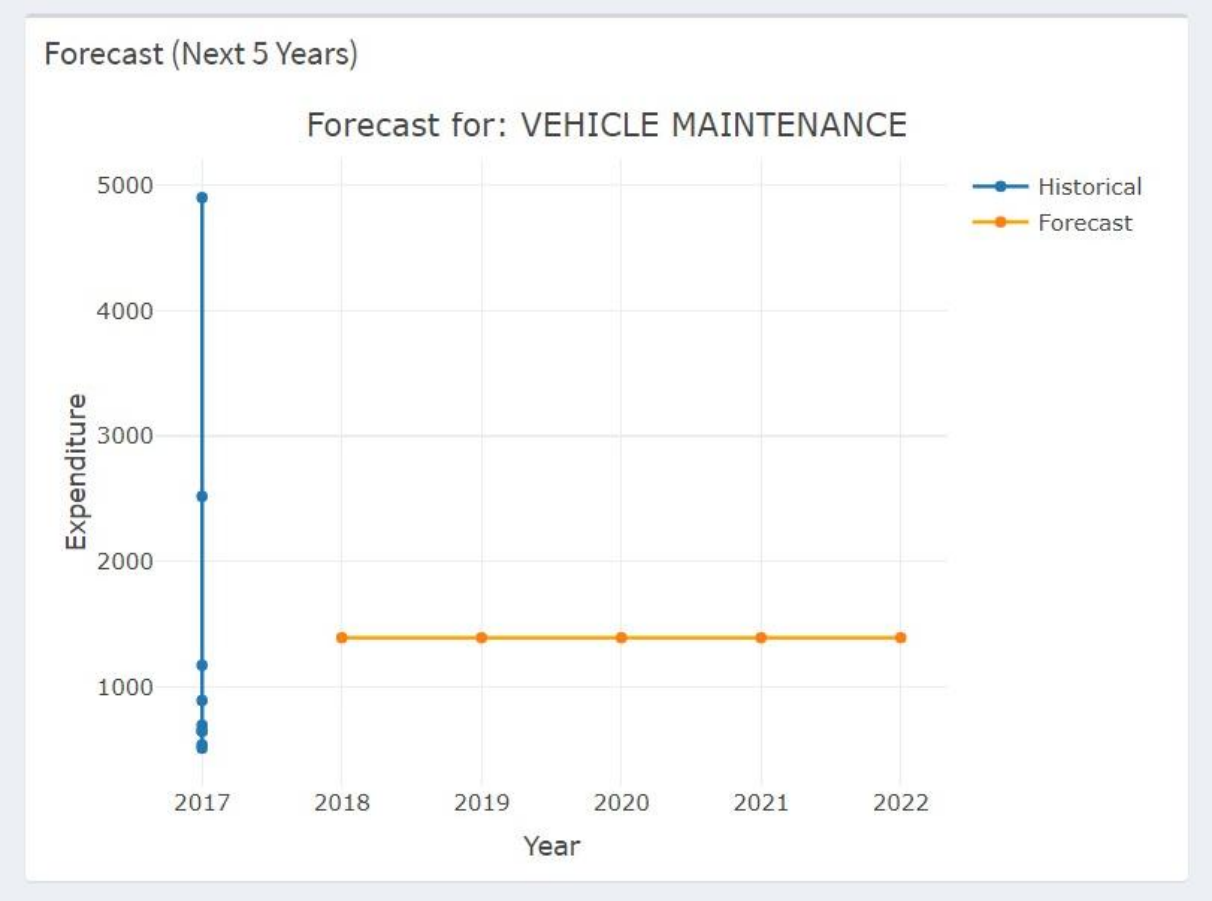
Linear Regression Modeling for Vehicle Maintenance

Vehicle maintenance linear regression modeling has, among other things, analyzed historical expenditure data concerning vehicle maintenance, whereby actual values and fitted trend lines were noted visually. The model threw a bright peak in 2017, while most other data points were clumped closely at much lower levels. It makes sense, therefore, that this insight is valuable in understanding how data is distributed and the limitations of employing a linear trend across such data.



Forecasting, in addition to the Implementation of Interactive Dashboard

A five-year period forecast for future vehicle maintenance expenditure was modeled with time series modeling. The visualization dashboard provides an immediate comparison of historical (blue) against future (orange) data, as it presents stable projections. The whole model and forecast process is, thus, directly integrated into a dashboard interface, permitting its end users to dynamically explore their service-specific forecasts and trends.



General Conclusion

The implementation of data mining techniques to identify patterns and learn from a real-world dataset has been investigated in this study. Several significant conclusions were drawn from a systematic process that included data comprehension, preprocessing, exploration, model building, and evaluation. The findings show that, given a well-prepared dataset, classification algorithms such as Random Forest and Decision Tree can accurately forecast results. The study emphasizes how crucial feature selection, data balancing, and model evaluation are to the data mining procedure. All things considered; this assignment has given me a practical grasp of how data mining methods can aid in making wise decisions in real-world situations.

Appendices

Appendix A: R Packages for Association Rule Mining used.

- **arules** : managing transactions and using the Apriori algorithm.
- **rulesViz:** Rules and item sets visualized.
- **tidyverse:** An assortment of R packages for visualizing and modifying data.
- **Lubridate:** Makes processing dates and times easier.
- **dplyr:** Tools for manipulating data.
- **knitr:** Creating dynamic reports.
- **ggplot2:** Visualization of data.
- **plyr:** An older program for manipulating data.
- **magrittr**: Introduce the pipe (%>%) operator
- **RColorBrewer:** Visualization color palettes.

Appendix B: Association Rule Mining - Dataset Summary

- **Dataset Source**: Council Spending Over £500 – Rochdale Borough Council
- **Records**: 24,401 transactions.
- **Key Variables**:
    - *Organization Name, Effective Date, Directorate, Supplier Name, Date Paid, Amount (£), Purpose, Transaction Number*.
- **Preprocessing Notes**:
    - Null values removed using na.omit().
    - Converted to transactional format for rule mining.
    - Apriori used with support = 0.1, confidence = 0.8.
    - Generated 58 rules.

Appendix C: Logistic Regression - Dataset Description

- **Dataset Source**: London Fire Brigade Incident Records (2009–2017)
- **Records**: ~988,280 incidents.
- **Attributes**:
    - *IncidentNumber, DateOfCall, IncidentGroup, PropertyType, NumPumpsAttending, PumpMinutesRounded*, etc.
- **Target Variable for Model**:
    - Binary classification for *High Pump* incidents.
- **Model Performance**:
    - Accuracy: 89.15% (but failed to predict minority class effectively).
    - Notable issues: class imbalance, convergence warnings.

Appendix D: Dashboard Implementation - Key Features

- Platform: R Shiny with Plotly.
- Tabs Implemented:
    - *Overview*, *Spending Trends*, *Forecasting*, *Expenditure Summary*.
- Features:
    - Time-series visualizations.
    - Regression and forecasting models.
    - Interactive filters by category, period, and service area.
- Use Case Example:
    - Vehicle maintenance trends (2016–2018), future forecasts up to 2022

References

- Google (no date) *Google Colaboratory*. Available at: https://colab.research.google.com/ (Accessed: 3 May 2025).
- Han, J., Pei, J. and Kamber, M. (2011) *Data mining: concepts and techniques*. 3rd edn. Waltham, MA: Morgan Kaufmann.
- Pandas Development Team (2023) *pandas (Version 2.1.0)* [Computer software]. Zenodo. Available at: https://doi.org/10.5281/zenodo.3509134 (Accessed: 3 May 2025).
- Scikit-learn developers (2023) *Linear regression example*. In: *Scikit-learn: machine learning in Python* (Version 1.3). Available at: https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html (Accessed: 3 May 2025).
- Scikit-learn documentation (no date) *Scikit-learn: Machine learning in Python*. Available at: https://scikit-learn.org/stable/ (Accessed: 3 May 2025).
- UCI Machine Learning Repository (no date) *Student Performance Data Set*. Available at: https://archive.ics.uci.edu/ml/datasets/Student+Performance (Accessed: 3 May 2025).
- Waskom, M.L. (2021) 'Seaborn: statistical data visualization', *Journal of Open Source Software*, 6(60), p.3021. DOI: 10.21105/joss.03021.
- Witten, I.H., Frank, E. and Hall, M.A. (2016) *Data mining: practical machine learning tools and techniques*. 4th edn. Burlington, MA: Morgan Kaufmann.
- Workshop 7 & 8 – Association Rule Mining.

Team Members;

- D/ADC/24/0003: Thesanya Lamahewa
- D/ADC/24/0013: Samoda De Silva
- D/ADC/24/0028: Thisari Perera
- D/ADC/24/0048: Dilakna Godagamage