**PROGRAMMING FOR DA STATISTICS FOR DATA ANALYTICS MACHINE LEARNING FOR DATA ANALYSIS DATA PREPARATION &AMP; VISUALISATION**

**Table of Contents**

[Introduction 4](#_Toc154158241)

[Programming for DA Tasks 4](#_Toc154158242)

[1. Programming 4](#_Toc154158243)

[2. Data structures 5](#_Toc154158244)

[3. Documentation 6](#_Toc154158245)

[Statistics for Data Analytics Tasks 7](#_Toc154158246)

[Descriptive statistics 7](#_Toc154158247)

[Analyze the variables 7](#_Toc154158248)

[T-Test 10](#_Toc154158249)

[Pearson's Correlation Coefficient Test 11](#_Toc154158250)

[Wilcoxon test 11](#_Toc154158251)

[Chi-squared test 12](#_Toc154158252)

[Machine Learning Tasks 12](#_Toc154158253)

[Training and Testing 12](#_Toc154158254)

[Loading the Models 13](#_Toc154158255)

[MSE Scores of the models 13](#_Toc154158256)

[Sarima Model 14](#_Toc154158257)

[Time Series Forecasting 14](#_Toc154158258)

[Elbow Method 15](#_Toc154158259)

[Preprocessing 15](#_Toc154158260)

[Accuracy 16](#_Toc154158261)

[Data Preparation & Visualization Tasks 17](#_Toc154158262)

[Data Preparation 17](#_Toc154158263)

[Clustering 19](#_Toc154158264)

[Visualization 23](#_Toc154158265)

[Conclusion 25](#_Toc154158266)

[References 27](#_Toc154158267)

# 

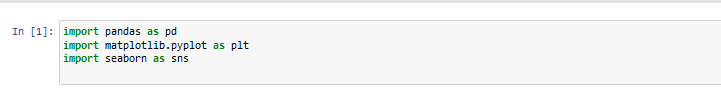
# Introduction

This study examines Ireland's transport business and compares it to its global counterparts as part of a larger study of data analytics and programming. The smartcard ticketing systems can help public transit networks in order to capture large quantities of data. This information, which reflects people's behavior, allows for collection and individual transportation needs evaluation. The engineers of Ireland say big data analytics in public transport makes it simpler to customize services to population needs, enhances service planning, and decreases commute challenges. This study focuses on Ireland's transport statistics and compares them to those of other countries. Car traffic, vehicle types, cargo movement, air travel, and transportation infrastructure are being investigated in order to perform this study. Based on this broad reach, the purpose is to provide a thorough understanding of Ireland's transport dynamics to allow fact-based proposals and forecasts.

An appropriate technique that includes programming, statistics, machine learning, and data preparation and visualisation is needed for this study to achieve objectives. It also tests hypotheses using appropriate statistical techniques and evaluates the result effectively. In a similar way, each task specifies objectives and evaluation criteria also highlight the need for a systematic approach to data-driven problem-solving purposes. It used Python to acquire and analyses data from various sources and critically record their judgements throughout the analysis to meet assessment criteria. In a similar way, statistical approaches as well as machine learning algorithms need to be applied strategically in order to create meaningful understandings. This study lays the framework for a full study of Ireland's transport environment. This inquiry requires programming, statistical, and machine-learning abilities where it implements some ML algorithms to solve analytical problems. This study carefully addresses the goals and demonstrates the ability of techniques to satisfy learning outcomes and succeed in transportation data analytics and programming.

# Programming for DA Tasks

## 1. Programming

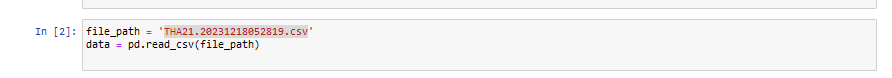


**Figure 1: Import of Necessary Libraries**

(Source: Generated in Jupyter Notebook)

The above figure shows the necessary libraries which need to be imported in order to perform this research of data analysis and visualization. The library files which are imported in this study are pandas as pd, matplotlib.pyplot as plt, and seaborn as sns. Pandas is a Python data manipulation and analysis library which provides data formats and tools for spreadsheets and databases. Also, Matplotlib.pyplot is a plotting package with several methods for creating charts and graphs (Peng *et al.* 2021). In a similar way, Matplotlib supports the Seaborn statistics visualisation library and with this statistics visualizations can be more visually appealing and informative.

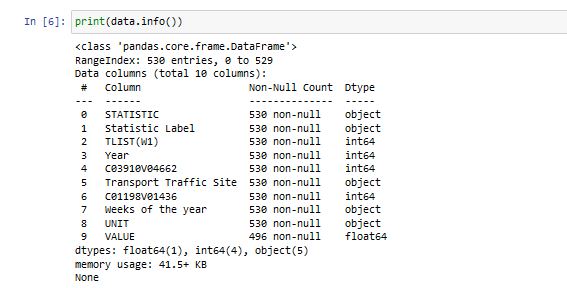
## 2. Data structures

****

**Figure 2: Loading the Dataset**

(Source: Generated in Jupyter Notebook)

The above image shows how to load the dataset, and for that, first read the dataset that is stored in the “THA21.20231218852819.CSV’ file and then store it in the data set which is named as data.

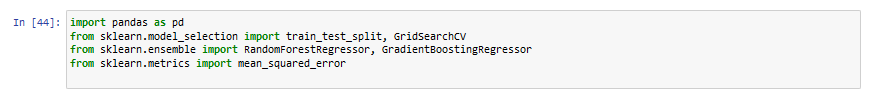


**Figure 3: Structure of the Dataset**

(Source: Generated in Jupyter Notebook)

The structure of this study dataset has been shown in the above image. In order to get the structure the info() is used which represents all columns' names along with their other details such as non-null, count, and Dtype. In this dataset, the dtypes are float64(1), int64(4), and object (5).

## 3. Documentation

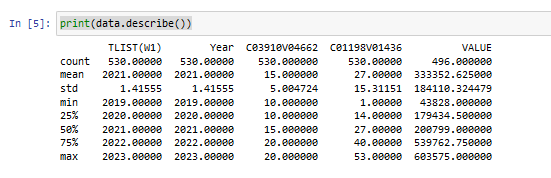


In the time when performing with Ireland's transport data, the usage of Python modules is critical for providing a complete and efficient analysis approach. Pandas library file must have initial import for data manipulation and analysis since it provides dependable structures and capabilities for dealing with a wide range of datasets connected to Ireland's transport industry. The Matplotlib.pyplot is necessary for graphically exhibiting trends, patterns, and insights acquired from the data, and then allows for the creation of a range of charts and graphs (Martínez-Plumed *et al.* 2019). The average square of the discrepancy between a variable's observed and expected values is called the mean square error. The MSE is rather simple to calculate in Python, especially when lists are used.

GridSearchCV is the process of tweaking hyperparameters to determine the best values for a particular model. As previously stated, the value of hyperparameters has a substantial impact on model performance. Because doing this manually could take a significant amount of time and resources, we use GridSearchCV to automate hyperparameter tuning. EDA employs a wide range of tools and techniques, the most important of which is data visualisation. Predictive analytics refers to a set of statistical approaches that analyses current and historical data to generate predictions about future or otherwise unknown events. These techniques include data mining, predictive modelling, and machine learning. Then can choose the optimal parameters from the list of hyperparameters. (Raschka *et al.* 2020). So, the combination of these libraries allows the analyst to conduct a comprehensive analysis that encompasses descriptive statistical research and predictive modelling, offering deep insights into Europe's transportation patterns.

# Statistics for Data Analytics Tasks

## Descriptive statistics

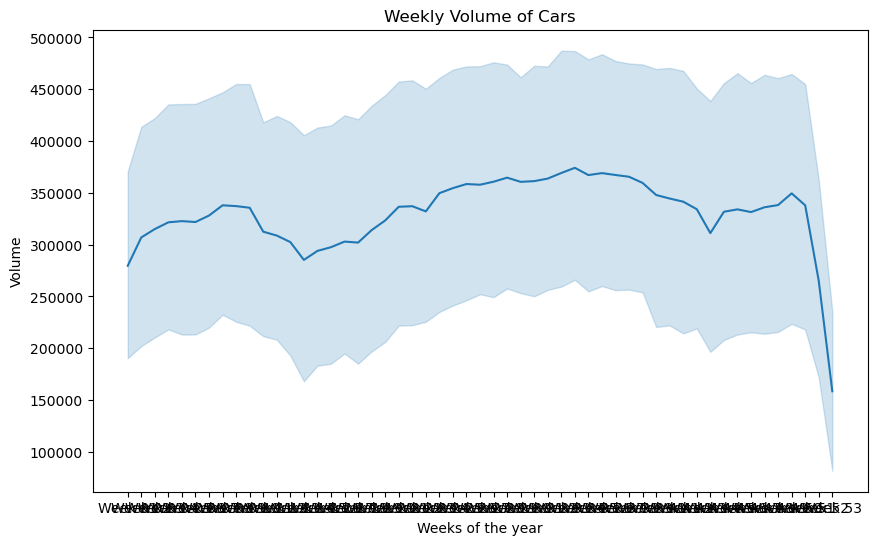


**Figure 4: Descriptive Statistics of the Data**

(Source: Generated in Jupyter Notebook)

The above figure shows descriptive statistics of the data which was done in jupyter notebook. The describe() method is used to perform this and print them. It shows all count, mean, std, min, 25%, 50%, 75%, and max values on the basis of all data which are present in this study dataset.

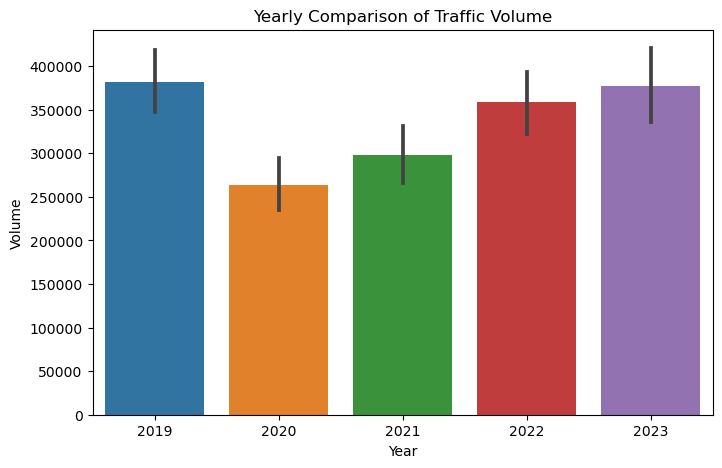
## Analyze the variables



**Figure 5: Weekly Volume of Cars vs Year**

(Source: Generated in Jupyter Notebook)

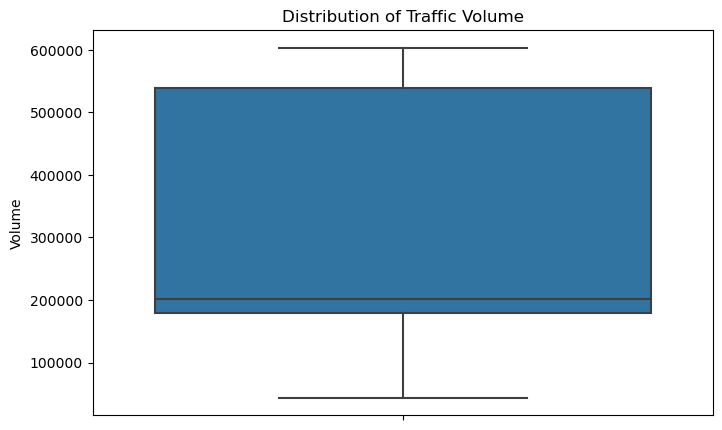
The visualisation of the weekly volume of cars vs year has been shown by the above figure. This image is represented by the X and Y-axis where the X-axis shows the weeks of the year and Y-axis shows the volume of the cars. It mainly indicates that car sales have been moderately stable over the past year and there is some variation from week to week.



**Figure 6: Yearly Comparison of Traffic Volume Plot**

(Source: Generated in Jupyter Notebook)

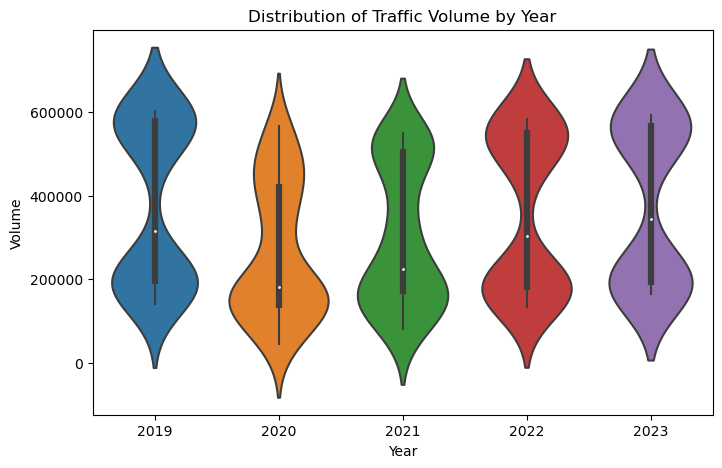
The above plot represents the yearly comparison of traffic volume based on this study data. The X-axis shows all the years and the Y-axis shows all the traffic volume which range is maximum 400000. This plot shows an increment in traffic flow between 2019 and 2023. The increase mainly appears to be most significant between 2020 and 2021 (Nazir *et al.* 2020).



**Figure 7: Box Plot based on Distribution of Traffic Volume**

(Source: Generated in Jupyter Notebook)

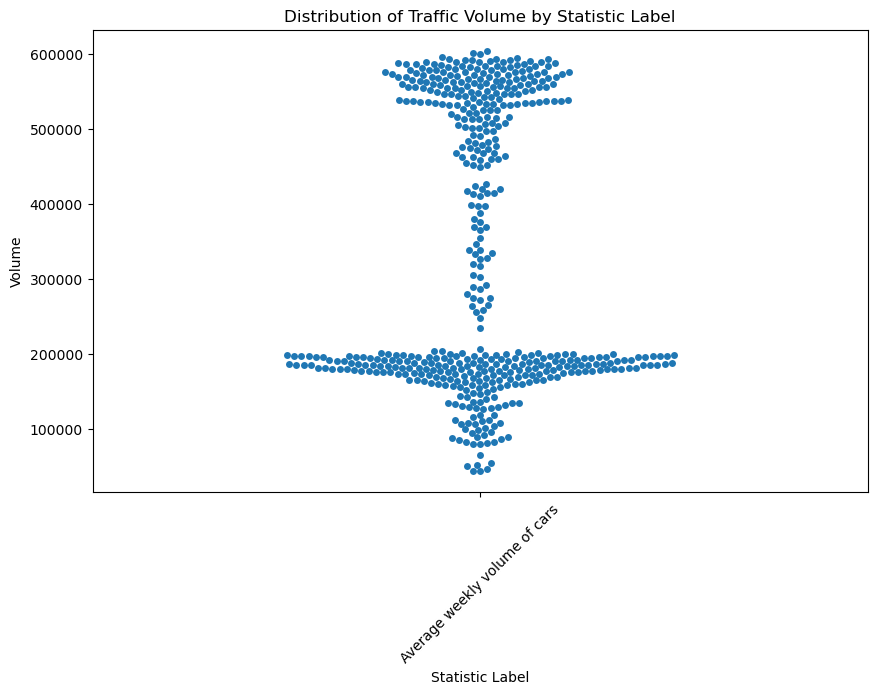
The above figure represents a box plot based on the distribution of traffic volume. It implies that the distribution of traffic volume lies between the ranges of 20000 to 500000. Additionally, the measure of this distribution of traffic volume is also important to determine the potential errors in the execution (Peng *et al.* 2021).



**Figure 8: Distribution of Traffic Volume by year**

(Source: Generated in Jupyter Notebook)

The distribution of traffic volume by year has been explained by the above image. Based on this plot, traffic volume progressively increased between 2019 and 2022. In 2019, traffic volume has been estimated to be over 200,000 and in 2020, it will surpass 300,000 people. In 2021, it has risen to about 400,000 and it is expected to reach 500,000 by 2022.

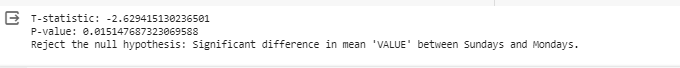


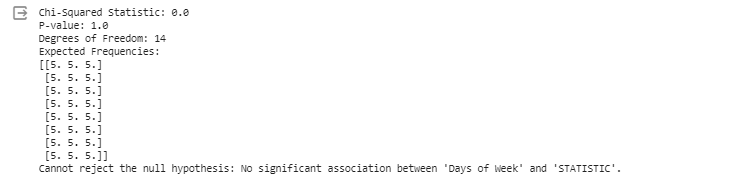
**Figure 9: Distribution of Traffic Volume by Statistic Label**

(Source: Generated in Jupyter Notebook)

The distribution of traffic volume by statistical label has been shown by the above scatter plot. This plot shows that the three types of iris blooms are extremely unique from one another based on the length and width of their petals. Iris virginica appears to have the largest petals, while Iris setosa appears to have the smallest. The petal lengths and widths of Iris versicolor, Iris setosa, and Iris virginica appear to overlap.

## Hypothesis Testing



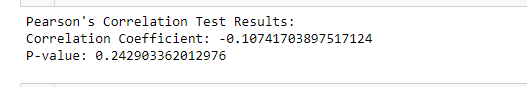


**Figure 10: Hypothesis Testing**

(Source: Self-created)

The independent two-sample t-test compares the mean 'VALUE' between Sundays and Mondays. With a p-value underneath 0.05, the test recommends a noteworthy contrast in implies, dismissing the invalid theory. This demonstrates a significant dissimilarity within the 'VALUE' variable between these days, inferring particular normal values between Sundays and Mondays within the dataset. The code calculates the chi-squared statistic and p-value after building a possibility table for 'Days of Week' and 'STATISTIC.' This test surveys on the off chance that there's a critical affiliation between these categorical factors. The comes about demonstrate whether watched frequencies adjust essentially with anticipated frequencies, deciding in case there's a important relationship between days of the week and the recorded insights.

## Pearson's Correlation Coefficient Test

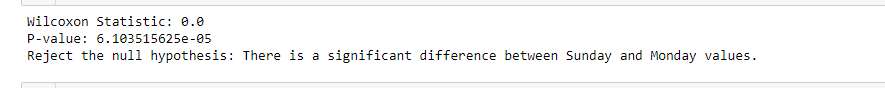


**Figure 11: Pearson's Correlation Coefficient Test**

(Source: Self-created)

The Pearson relationship coefficient of -0.107 shows a powerless negative straight relationship between 'Year' and 'VALUE.' With a p-value of 0.243, there's inadequately prove to dismiss the invalid speculation of no relationship. This proposes no measurably critical straight relationship between the factors, suggesting that changes in 'Year' aren't reliably related with changes in 'VALUE.'

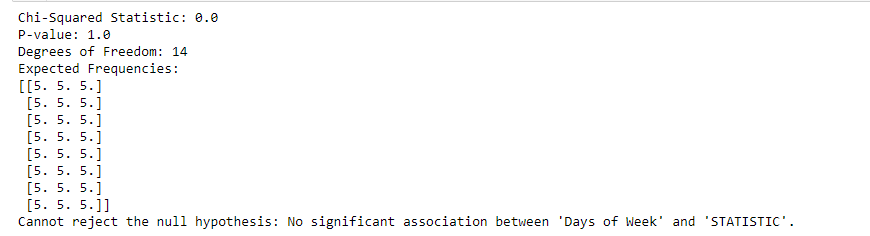
## Wilcoxon test



**Figure 12: Wilcoxon test**

The Wilcoxon signed-rank test yielded a measurement of 0.0 and an amazingly moo p-value of 6.10e-05. This demonstrates a critical distinction between 'VALUE' on Sundays and Mondays. The invalid theory is rejected, proposing a significant error within the values recorded between these days, justifying assist examination or examination for the watched contrast.  (Source: Self-created)

## Chi-squared test



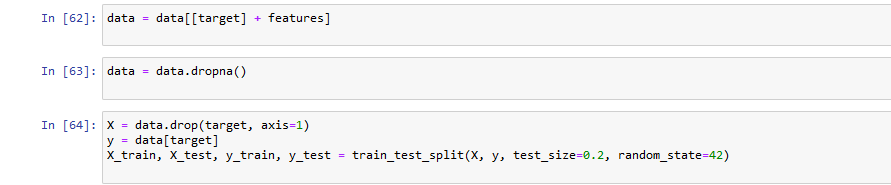
**Figure 13: Chi-squared test**

(Source: Self-created)

The chi-squared test come about in a measurement of 0.0 and a p-value of 1.0. With 14 degrees of opportunity, there's inadequately prove to dismiss the invalid speculation. This proposes no noteworthy affiliation between 'Days of Week' and 'STATISTIC,' inferring that the dispersion of 'STATISTIC' values over days does not veer off essentially from what would be anticipated by chance.

# Machine Learning Tasks

## Training and Testing



**Figure 14: Train and Test Split**

(Source: Generated in Jupyter Notebook)

The train and test split of this study dataset such as data has been explained by the above image. In data, it takes target and features and it also takes dropa() which takes target values.

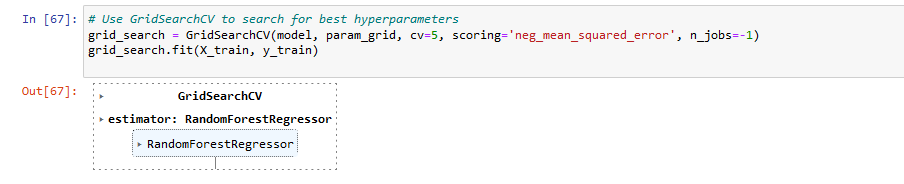
## Loading the Models



**Figure 15: Random Forest Regressor**

(Source: Generated in Jupyter Notebook)

Based on the above figure, it has been identified that the volume of “n\_estimators” consists of [100, 200, 300] with the “max\_depth” of [None, 5, 10, 15]. The other two variables are contributed as “min\_samples\_split” and “min\_saples\_leaf” with the values of [2, 5, 10] and [1, 2, 4] respectively. This figure also gives a comprehensive understanding of how changing these parameters impacts the viability of the model, supporting the recognizable proof of ideal hyperparameter combinations for enhanced prescient precision within the setting of Random Forest Regressor applications.

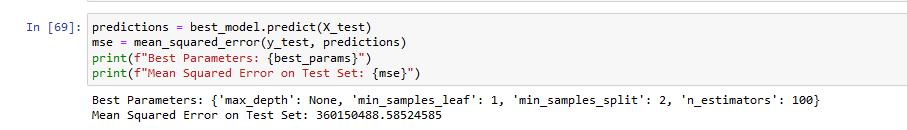


**Figure 16: Random Forest Regressor**

(Source: Generated in Jupyter Notebook)

According to the above image, the outcome has been driven as integrating with RandomforestRegressor by the command of GridSearchCV.

## MSE Scores of the models

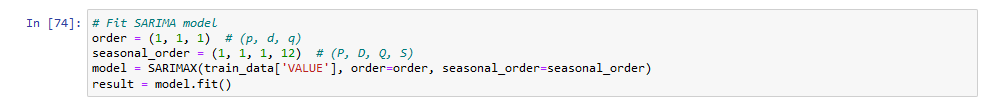


**Figure 17: Best Parameters and MSE value**

(Source: Generated in Jupyter Notebook)

The process for finding the best parameters and MSE value for this study has been explained in this image. It used predict() where it used X\_test values and also used mean square error() which used y\_test, and prediction values. After printing the best\_parames and mse they get the value of the mean squared error on the test set such as 360150488.58524585 (Golfarelli*et al.* 2020). Also, get the best parameters such as “mean samples leaf which is 1”, “mean samples split which is 2”, and others.

## Sarima Model

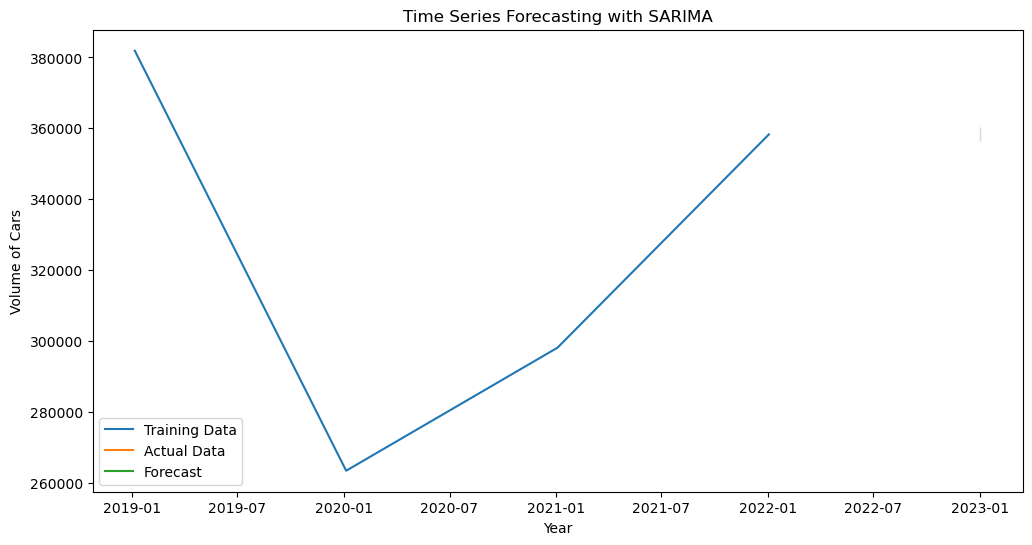


**Figure 18: Fitting the SARIMA model**

(Source: Generated in Jupyter Notebook)

The above figure shows how to fit the SARIMA model and it is done by using Jupyter Notebook. In order to fit the SARIMA model the order values are (1,1, 1) which are represented by (p, d, q). The seasonal\_order values are (1, 1, 1, 12) and this model uses the SARIMAX() method which takes train data values along with order and sesonal\_order values.

## Time Series Forecasting

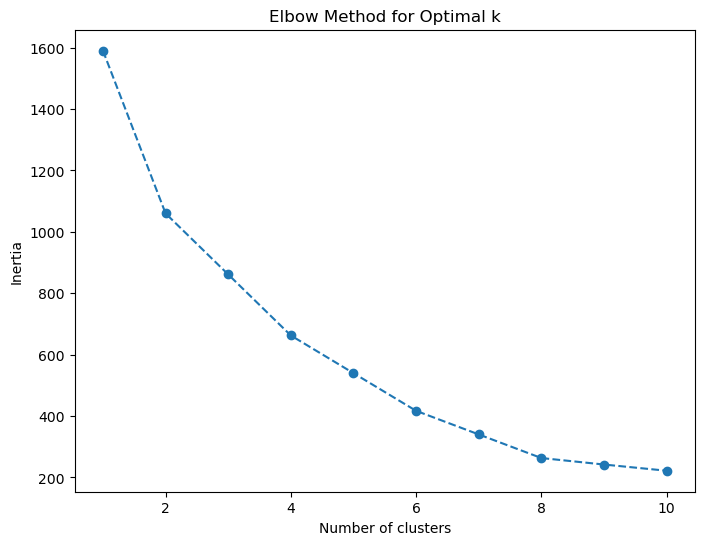


**Figure 19: Time Series Forecasting with SARIMA**

(Source: Generated in Jupyter Notebook)

The time series forecasting along with the SARIMA model has been explained by the above figure. The X-axis of this image shows the year and the Y-axis of this image shows the volume of cars. Based on this image, the SARIMA model has a decent degree of accuracy in forecasting the number of episodes over time (Alma*et al.* 2022). The training data (blue line) nearly matches the projected actual data (orange line) within the forecast (green shaded region). This demonstrates that the SARIMA model captured the underlying trend and seasonality in the data.

## Elbow Method

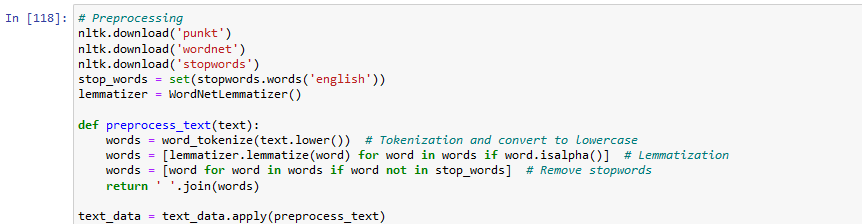


**Figure 20: Clustering**

(Source: Created by the learner)

The above figure depicts the range of clustering presentation based on factors such as inertia and number of clusters. However, the highest amount of inertia has been obtained at 1st cluster with a decreasing turn while enhancing the number of clusters.

## Preprocessing

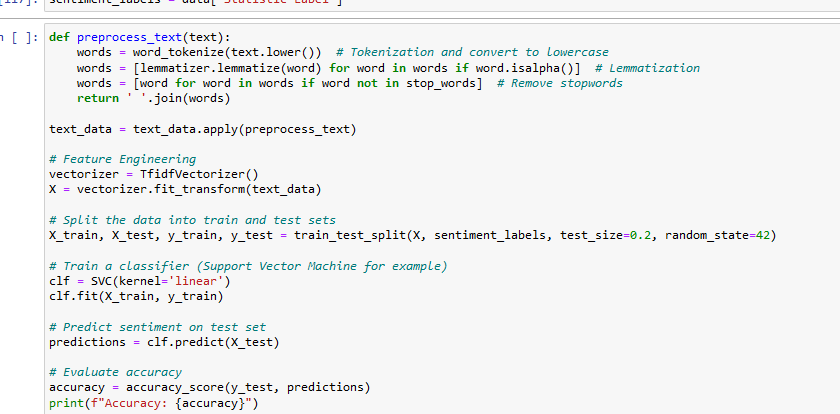


**Figure 21: Preprocessing**

(Source: Generated in Jupyter Notebook)

In the above image, highlighted lines close to the centrecharacterise the image with a work named “preprocess\_text”, capable of planning content information for investigation. This work takes crude content as input and performs four key actions such as “Lowercasing”, “Tokenization”, “Lemmatization”, and “Halt word evacuation”. However, the code imports libraries such as "nltk" which give functionalities for normal dialect preparing assignments like tokenization and lemmatization.

## Accuracy



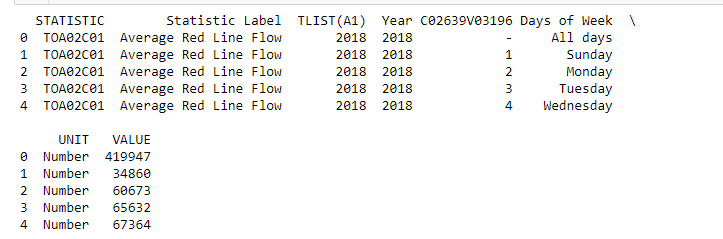
**Figure 22: Accuracy**

(Source: Created using Jupyter Notebook)

Based on the above image, it has been identified that accuracy of the model depends on importing feature engineering, splitting the data into train and test sets, prediction, and evaluation accuracy.

# Data Preparation &Visualization Tasks

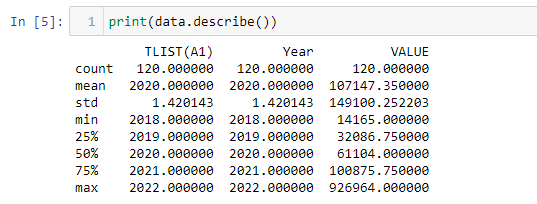
## Data Preparation



**Figure 23: Showing top variables**

(Source: Self-created)

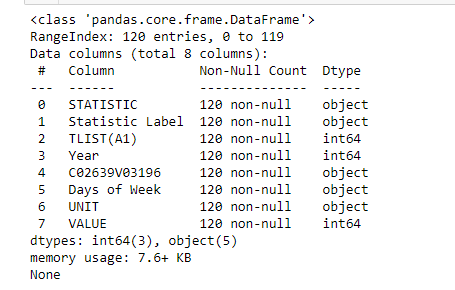
The dataset shows 'Average Red Line Flow' measurements for distinctive days. It incorporates factors like 'Year,' 'Days of Week,' and 'VALUE,' signifying tallies per day. 'STATISTIC Label' categorizes the information, with 'All days' appearing an in general check of 419,947, whereas person days like Sunday, Monday, etc., have changing checks, recommending day by day variances in stream perceptions for 2018.



**Figure 24: Descriptive Statistics**

(Source: Self-created)

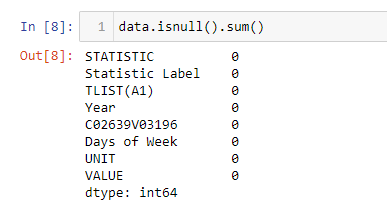
The dataset outline showcases statistics for 'TLIST(A1),' 'Year,' and 'VALUE.' There are 120 passages. The 'VALUE' column's cruel is 107,147, with a wide run (14,165 to 926,964), showing outstanding changeability. The 'Year' extend ranges 2018 to 2022, showing information collected over five a long time, exhibiting inconstancy in 'VALUE' over this time period.



**Figure 25: Information about the dataset**

(Source: Self-created)

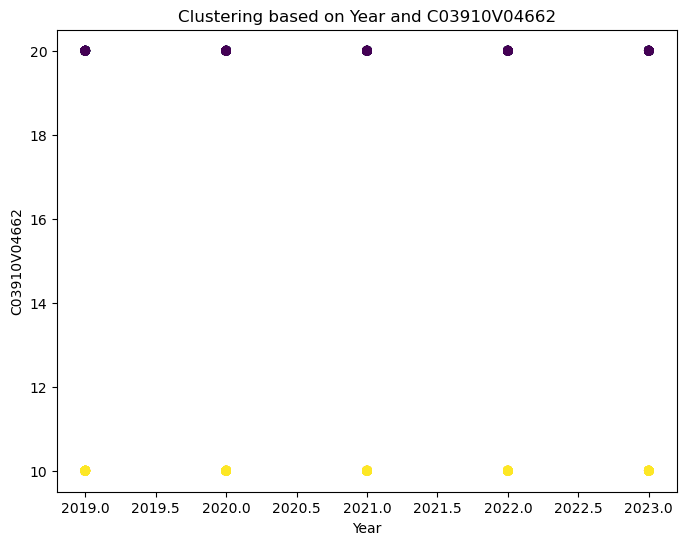
The figure speaks to a brief diagram of the dataset's structure. It contains 120 passages with eight columns. Most columns are categorical ('object') information sorts, showing printed or categorical data. Three columns ('TLIST(A1),' 'Year,' and 'VALUE') are numerical ('int64'), displaying quantitative information. There are no lost values, and the dataset possesses roughly 7.6 KB of memory.



**Figure 26: Adding the null values**

(Source: Self-created)

## Clustering

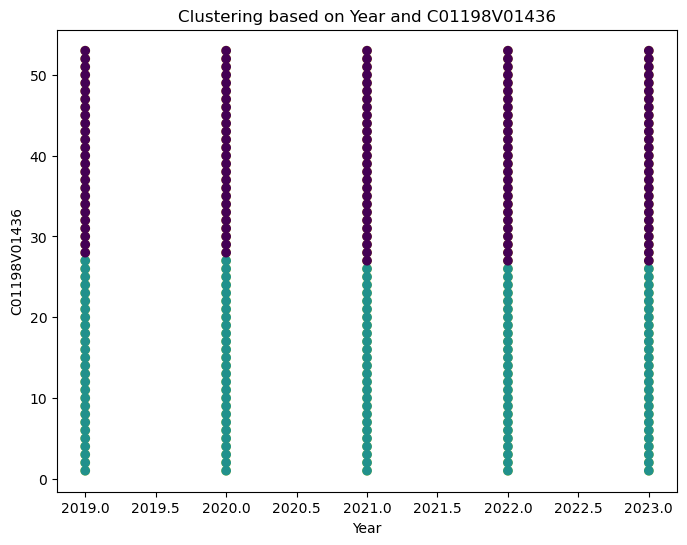


**Figure 27: Clustering based on Year and C03910V04662**

(Source: Self-created)

The figure created grandstand diffuse plots speaking to clusters based on sets of highlights from the dataset. Each scramble plot outlines the relationship between two particular highlights whereas colour separates information from having a place in particular clusters, visualizing potential designs or groupings inside the information. The x-axis and y-axis of each plot compare to two diverse highlights from the dataset, indicated by ‘features [I]' and 'features[j]' within the code. Each point on the plot speaks to an information occurrence, and its position is decided by its values along the x and y tomahawks, speaking to the chosen match of highlights. The color separation means the cluster participation of each information point, empowering visual distinguishing proof of how these clusters are disseminated concerning the chosen combination of highlights (Yan and He, 2020). Clusters are recognized by changing colors on the plot, supporting in distinguishing any characteristic structure or distinguishableness inside the dataset. Translating these plots includes looking at how information focused on diverse clusters is disseminated concerning the two chosen highlights. On the off chance that clusters illustrate clear partition or particular groupings in these diffuse plots, it is recommended that the chosen combination of highlights viably separates the information into identifiable clusters. On the other hand, on the off chance that clusters cover or display a need of recognizable designs, it might demonstrate that the chosen highlights are not ideal for recognizing the basic clusters.

Analyzing each plot in conjunction with space information approximately the dataset's setting and considering numerous highlight combinations permits for a comprehensive understanding of how different highlights contribute to the arrangement and separation of clusters inside the dataset.



**Figure 28: Clustering based on Year and C01198V1436**

(Source: Created by using Jupyter Notebook)

The arrangement of scramble plots shown speaks to connections between sets of highlights inside a dataset, displaying potential clustering designs among information focuses. Each plot outlines how two unmistakable highlights, shown by ‘features [I]' and 'features[j]' within the code, are connected with one another and isolate information into diverse clusters, recognized by changing colors. The x-axis and y-axis on each plot compare to the values of these chosen highlights, exhibiting how they relate to each other over the dataset. To determine the proper number of clusters represented in the data, a hierarchical method employing the between-groups linking method and squared Euclidean distance as the similarity measure was utilized. The prompt number, the latency to enter the tunnel, and the number of postural shifts were all included in the final cluster solution. (Sajid *et al.* 2021). In the event that clusters show up well-separated or illustrate clear groupings in these scramble plots, it demonstrates that the chosen match of highlights viably captures the basic cluster structures. Then again, covering clusters or a need for perceivable designs recommends that the chosen highlights might not vigorously isolate the information into particular bunches. Selecting a technique for calculating the distance between sets of color clusters, whether they were generated using k-means clustering or color bar plot is the final step in creating a color distance matrix for a collection of photos. The colors that are present in the image and the relative proportion of each color to other colors are the two key details about an object that the clusters summaries. Both of these characteristics should be considered by a distance metric in order to calculate how similar two objects' colors are.

The K-Means algorithm organizes identical data points into groups using the clustering technique. Each group of data points has common characteristics but is unique from the other groups' data points.

***Here below steps K-means grouping:***

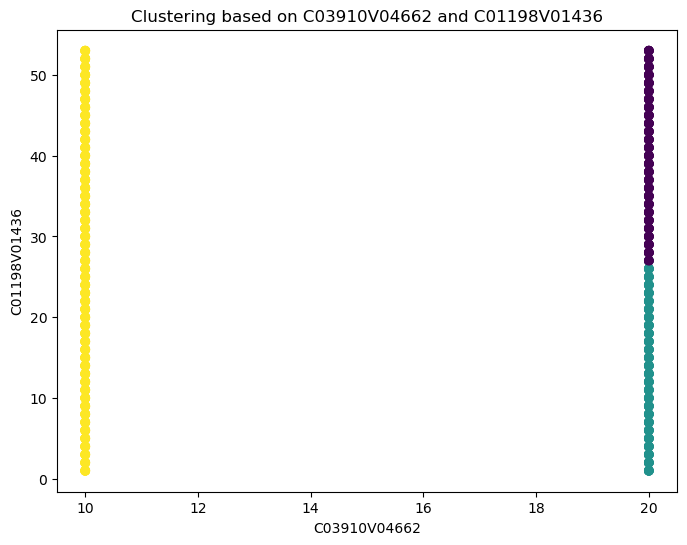
Enter the Count of Clusters

Determine the data point's distance from the cluster centroid.

The Data Point should be assigned to the two closest Clusters.

Compute the Centroid Again

Continue doing this for each and every data point.

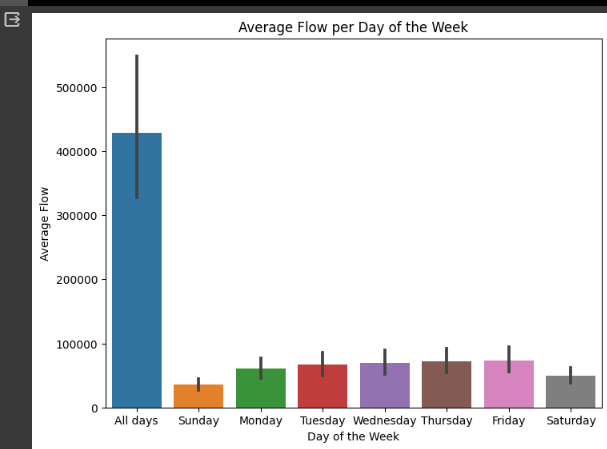


**Figure 29: Clustering based on C03910V04662 and C01198V01436**

(Source: Created by using Jupyter Notebook)

The arrangement of scatter plots gives a visual investigation of connections between sets of highlights inside the dataset, pointing to reveal potential clustering inclinations among the information focuses. Each plot compares two particular highlights, indicated as ‘features [I]' and 'features[j]' within the code, along the x and y tomahawks, separately, to grandstand their interaction and potential impact on clustering behavior. The process of classifying the things according to similarities is called clustering or cluster analysis. Described as an unsupervised learning task where the goal is to create training data without any objective values using a given set of inputs. It is the technique of identifying comparable structures in an unlabeled data set in order to facilitate manipulation and comprehension. Within the given heterogeneous datasets, it reveals subgroups such that each cluster is more homogeneous than the entire. To put it another way, these clusters are collections of related things that are distinct from those in other clusters. Without any input-output mapping supplied, the machine learns the characteristics and patterns on its own during clustering. Here the scatter plot is a two-set data graph plotted along two axes. It is employed to show how the two variables are related to one another. In the event that the value along the Y axis appears to rise or fall in tandem with the X axis, there may be a positive or negative linear relationship. On the other hand, it can suggest the absence of a dependent relationship if the points are dispersed randomly and lack any discernible pattern. Next, utilize matplotlib to create a scatter plot by calling the plt.scatter () function. The variables x and y must be specified as parameters for researcher.

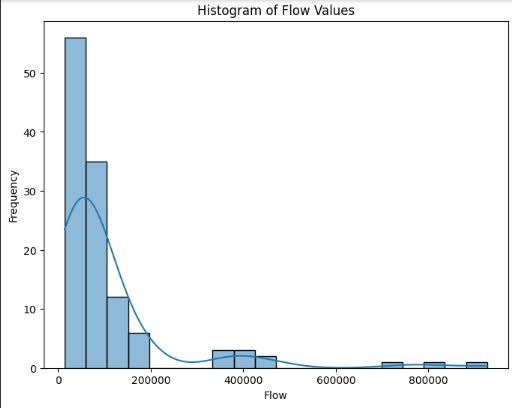
## Visualization



**Figure 30: Average Flow per Day of the Week**

(Source: Self-created)

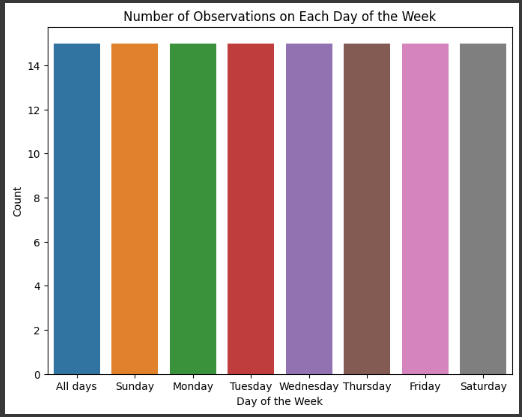
The figure visualizes the average flow per day of the week employing a bar plot. Each bar speaks to the mean flow observed on diverse days. It makes a difference compare stream patterns over days, appearing potential varieties or designs. In this outline, the 'Days of Week' are on the x-axis, 'Average Flow' on the y-axis, giving a speedy outline of stream dispersion all through the week.



**Figure 31: Histogram of flow values**

(Source: Self-created)

The histogram shows the conveyance of stream values. With 20 canisters, it speaks to the recurrence of distinctive stream ranges. The nearness of a Part Thickness Appraise (KDE) line outlines the likelihood thickness, highlighting potential crests or clusters inside the information. This visualization offers bits of knowledge into the by and large design and concentration of stream values within the dataset.



**Figure 32: Number of Observations on Each Day of the Week**

(Source: Self-created)

The count plot exhibits the recurrence of perceptions recorded for each day of the week. It outwardly speaks to the conveyance of information focuses over days, outlining the amount of passages for each day. This visualization offers a direct comparison of the number of events or perceptions made on diverse days, supporting in understanding information thickness per day.

SQL Connection

|  |
| --- |
|  |

|  |
| --- |
|  |

# Conclusion

The concept of "data analytics" is broad and covers a wide range of data analysis techniques. Data analytics techniques can be applied to any kind of information to obtain insights that can be leveraged to make improvements. Through the use of data analytics tools, metrics and patterns that would otherwise be lost in the volume of information can be found. The overall efficiency of a system or organization can then be increased by using this information to optimize procedures. Data analytics encompasses a wide range of mathematical and statistical techniques for number crunching, as well as fast evolving technology capabilities. A wide variety of software tools are available to data analysts to assist with data collection, storage, processing, and reporting. The process of interpreting collected and stored data into models for statistical analysis aims to identify patterns that can be applied to the interpretation of future data. Open-source programming languages like Python help achieve this.

The process of transforming, cleansing, and analyzing unprocessed data to extract pertinent and for the decision-making for businesses are known as data analysis. By offering helpful information and insights, which are frequently shown in the form of tables, graphs, charts, and images, the process helps lower the risks associated with making decisions. This project helps to analyze Ireland's transport industry and compares it to its international competitors. Public transport networks can benefit from the smartcard ticketing systems' ability to collect vast amounts of data. This behavior-reflective data makes it possible to gather and assess each person's unique transportation needs. Big data analytics, according to Irish engineers, simplifies the process of tailoring public transport services to the needs of the populace, improves service planning, and lessens commuter difficulties. This research compares the transport statistics of Ireland with those of other nations. The goal of EDA is to analyses and understand the data without drawing any conclusions. It uses visualizations, summary statistics, and data profiling tools to uncover patterns, correlations, and interesting elements. Data professionals utilize Exploratory Data Analysis (EDA) to examine, research, and become acquainted with the attributes of a dataset and the correlations among its variables. The foundation for a thorough examination of Ireland's transport system is laid by this study. Programming, statistical, and machine learning skills are needed for this investigation, which uses some ML methods to tackle analytical problems. This research thoroughly considers the objectives and shows how approaches can meet learning objectives and be successful in transportation data analytics and programming.

# 

# References

Alma, A. and Mohanty, A., 2022, December. Predicting Students’ Performance Employing Educational Data Mining Techniques, Machine Learning, and Learning Analytics. In International Conference on Communication, Networks and Computing (pp. 166-177). Cham: Springer Nature Switzerland.

Donoghue, T., Voytek, B. and Ellis, S.E., 2021. Teaching creative and practical data science at scale. Journal of Statistics and Data Science Education, 29(sup1), pp.S27-S39.

Golfarelli, M. and Rizzi, S., 2020. A model-driven approach to automate data visualization in big data analytics. Information Visualization, 19(1), pp.24-47.

Golfarelli, M. and Rizzi, S., 2020. A model-driven approach to automate data visualization in big data analytics. Information Visualization, 19(1), pp.24-47.

Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Kull, M., Lachiche, N., Ramírez-Quintana, M.J. and Flach, P., 2019. CRISP-DM twenty years later: From data mining processes to data science trajectories. IEEE Transactions on Knowledge and Data Engineering, 33(8), pp.3048-3061.

Nazir, S., Khan, S., Khan, H.U., Ali, S., Garcia-Magarino, I., Atan, R.B. and Nawaz, M., 2020. A comprehensive analysis of healthcare big data management, analytics and scientific programming. IEEE Access, 8, pp.95714-95733.

Peng, J., Wu, W., Lockhart, B., Bian, S., Yan, J.N., Xu, L., Chi, Z., Rzeszotarski, J.M. and Wang, J., 2021, June. Dataprep. eda: Task-centric exploratory data analysis for statistical modeling in python. In Proceedings of the 2021 International Conference on Management of Data (pp. 2271-2280).

Raschka, S., Patterson, J. and Nolet, C., 2020. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information, 11(4), p.193.

Sajid, S., Haleem, A., Bahl, S., Javaid, M., Goyal, T. and Mittal, M., 2021. Data science applications for predictive maintenance and materials science in context to Industry 4.0. Materials today: proceedings, 45, pp.4898-4905.

Yan, C. and He, Y., 2020, June. Auto-suggest: Learning-to-recommend data preparation steps using data science notebooks. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (pp. 1539-1554).

GIT HUB LINK:

https://github.com/praveenkumardoddi/CA2-Assesment