Configuration Manual

MSc Research Project
Data Analytics

Forename Surname
Student ID: x22104275

School of Computing
National College of Ireland

Supervisor: Mayank Jain

National College of Ireland

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Aravind Hallimysore Kalegowda

Student ID: X22104275

Programme: Data Analytics **Year:** 2023

Module: MSc Research Project

Lecturer: Mayank Jain

Submission Due Date: 31/Jan/2024

Project Title: Utilizing Predictive Analytics to Enhance Retail Business Performance

Word Count: …1280… **Page Count:** ..8..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Aravind Hallimysore Kalegowda……………

Date: ……………………28/01/2024………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☒
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☒
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☒

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Aravind Hallimysore Kalegowda
X22104275

1 Introduction

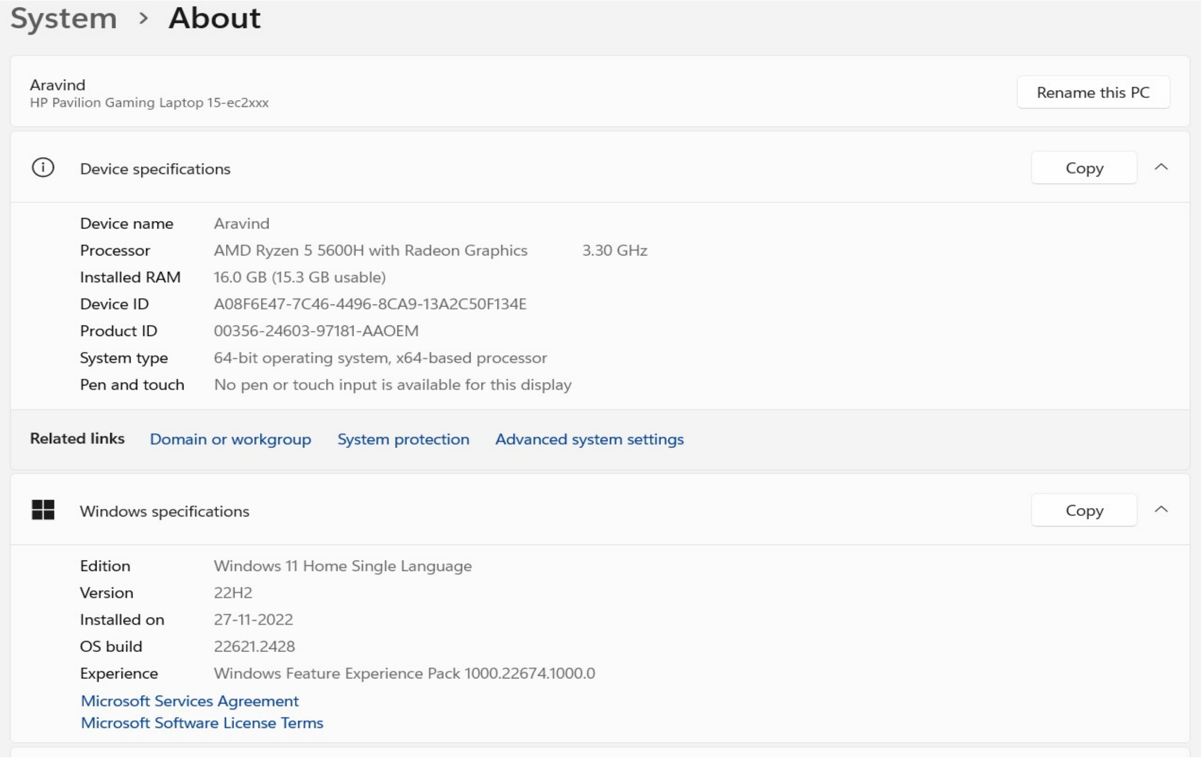
The aim of this manual is to gives the steps and configurations used in the current research. This guide provides complete information about the software and hardware setups along with the libraries utilized in this project. It also outlines the coding process and instructions for running the code.

2 System Specifications

2.1 Hardware Specifications:

The below table and Fig.1. gives the details of hardware details of the local system used in this project.

Operating system	Windows 11
RAM	16GB (Minimum 8gb required)
Hard Disk	512GB



Aravind
HP Pavilion Gaming Laptop 15-ec2xxx

Device specifications

Device name	Aravind
Processor	AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz
Installed RAM	16.0 GB (15.3 GB usable)
Device ID	A08F6E47-7C46-4496-8CA9-13A2C50F134E
Product ID	00356-24603-97181-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Windows specifications

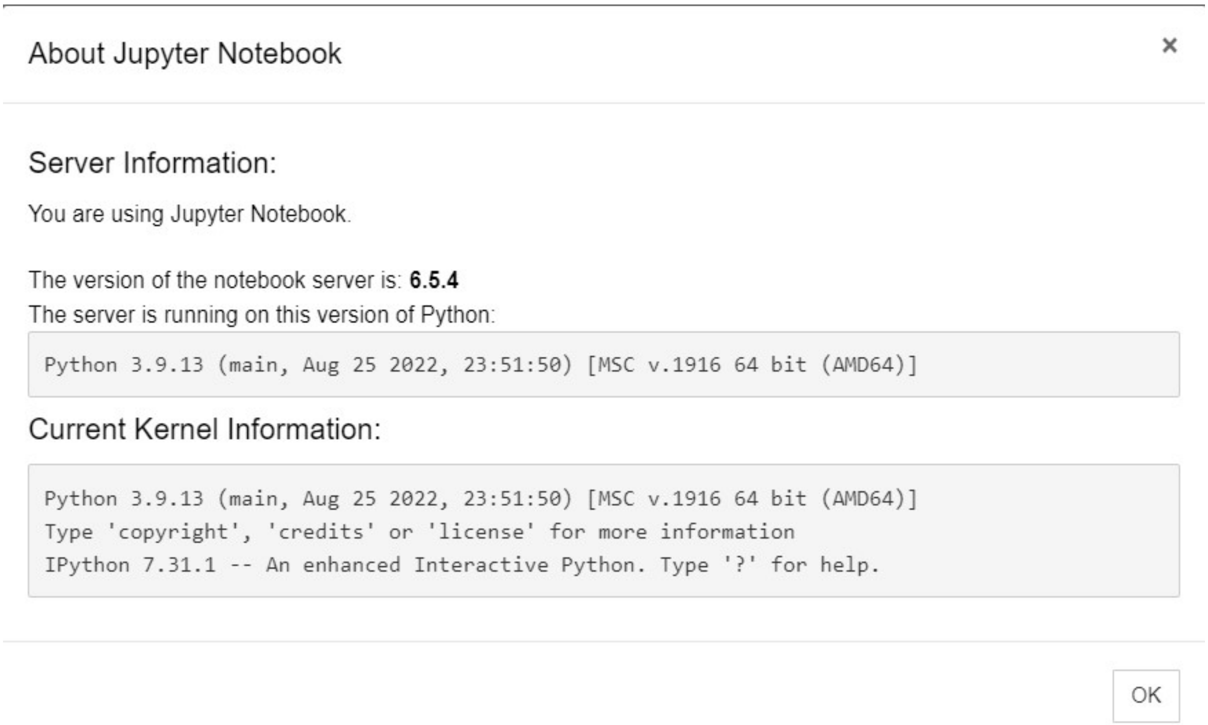
Edition	Windows 11 Home Single Language
Version	22H2
Installed on	27-11-2022
OS build	22621.2428
Experience	Windows Feature Experience Pack 1000.22674.1000.0

Fig.1. Hardware specification

2.2 Software Specifications:

The below table and Fig.1 gives the details of Software used in this project.

IDE	Jupyter Notebook
Software language	Python(3.9.13)
Other Tool	CSV (For Data Exploration)



About Jupyter Notebook

Server Information:

You are using Jupyter Notebook.

The version of the notebook server is: **6.5.4**
The server is running on this version of Python:

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
```

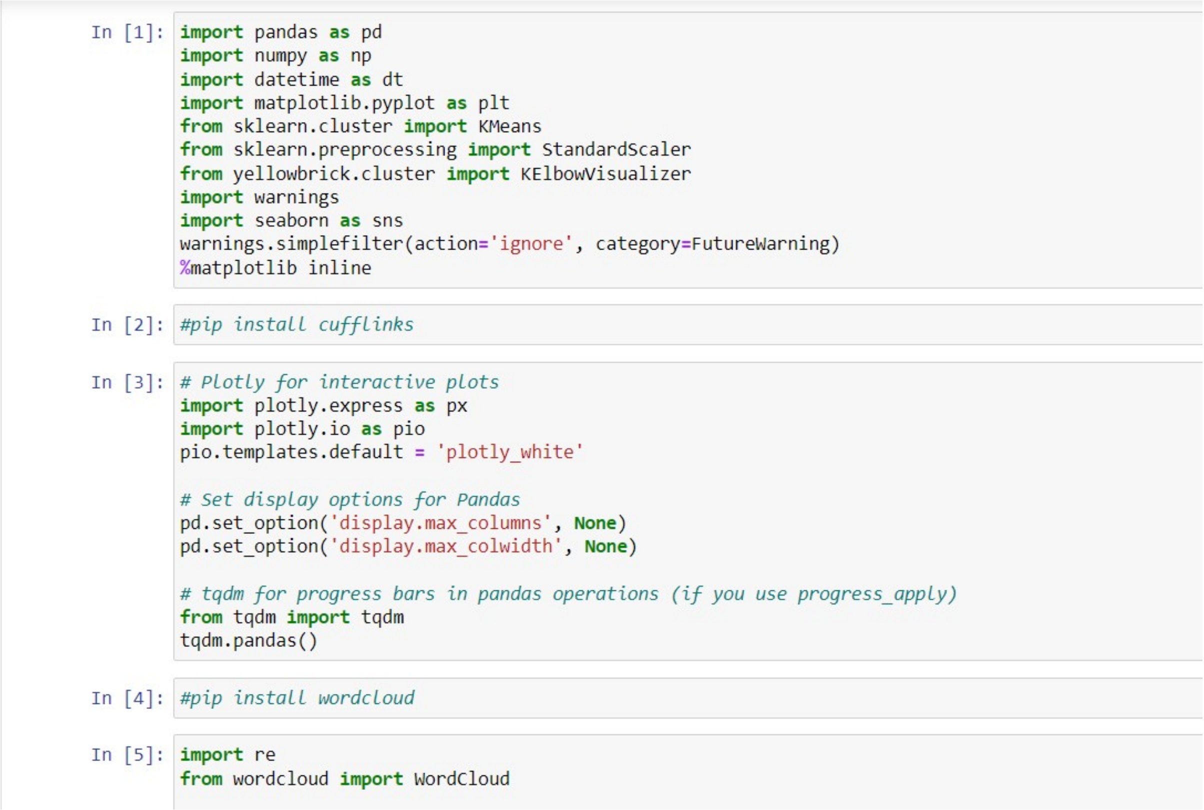
Current Kernel Information:

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.31.1 -- An enhanced Interactive Python. Type '?' for help.
```

Fig.2. Software Specifications

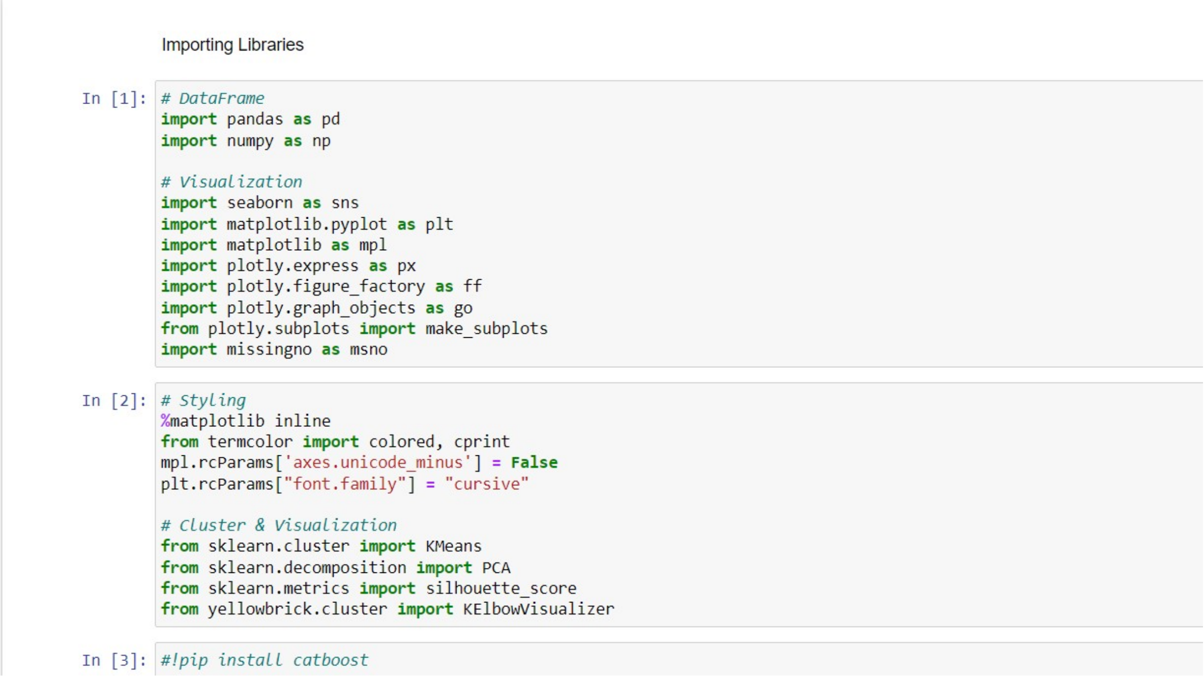
3 Libraries Installation

In this section of the Configuration Manual should be followed by installation commands for these libraries to set up the environment properly. Fig.2. and Fig.3. gives the all the libraries used in Customer_segmention and Store_sales_prediction part of coding respectively.



```
In [1]: import pandas as pd
        import numpy as np
        import datetime as dt
        import matplotlib.pyplot as plt
        from sklearn.cluster import KMeans
        from sklearn.preprocessing import StandardScaler
        from yellowbrick.cluster import KElbowVisualizer
        import warnings
        import seaborn as sns
        warnings.simplefilter(action='ignore', category=FutureWarning)
        %matplotlib inline

In [2]: #pip install cufflinks

In [3]: # Plotly for interactive plots
        import plotly.express as px
        import plotly.io as pio
        pio.templates.default = 'plotly_white'

        # Set display options for Pandas
        pd.set_option('display.max_columns', None)
        pd.set_option('display.max_colwidth', None)

        # tqdm for progress bars in pandas operations (if you use progress_apply)
        from tqdm import tqdm
        tqdm.pandas()

In [4]: #pip install wordcloud

In [5]: import re
        from wordcloud import WordCloud
```

Fig.3. Libraries installed for Customer_segmentaion



Importing Libraries

```
In [1]: # DataFrame
        import pandas as pd
        import numpy as np

        # Visualization
        import seaborn as sns
        import matplotlib.pyplot as plt
        import matplotlib as mpl
        import plotly.express as px
        import plotly.figure_factory as ff
        import plotly.graph_objects as go
        from plotly.subplots import make_subplots
        import missingno as msno

In [2]: # Styling
        %matplotlib inline
        from termcolor import colored, cprint
        mpl.rcParams['axes.unicode_minus'] = False
        plt.rcParams["font.family"] = "cursive"

        # Cluster & Visualization
        from sklearn.cluster import KMeans
        from sklearn.decomposition import PCA
        from sklearn.metrics import silhouette_score
        from yellowbrick.cluster import KElbowVisualizer

In [3]: #!pip install catboost
```

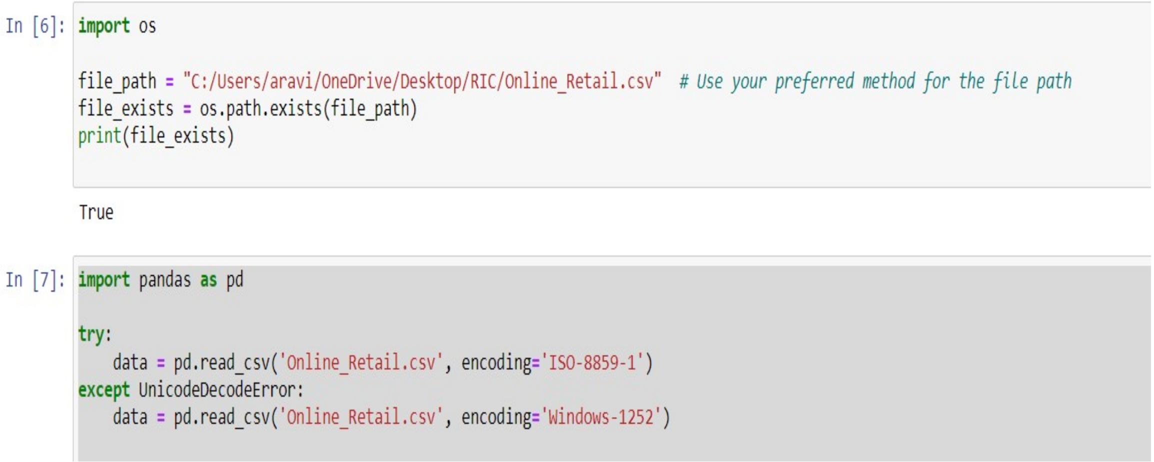
Fig.4. Libraries installed for Store_sales_prediction

For " Customer segmentation":

In the Fif.3. " Customer segmentation " of this research data handling and calculations rely on pandas for its powerful data manipulation capabilities and NumPy for efficient numerical computations. The datetime library is instrumental in managing date and time data, an essential aspect of customer segmentation analysis. Visualizations play a critical role in our research, with matplotlib.pyplot providing a wide range of static and interactive plotting options, while seaborn offers sophisticated statistical visualizations. plotly.express takes our data presentation to the next level with interactive and advanced plotly visualizations. On the machine learning front, sklearn.cluster houses the KMeans algorithm used for clustering, with sklearn.preprocessing offering tools like StandardScaler for feature scaling. The yellowbrick.cluster module's KElbowVisualizer is a pivotal tool in selecting the optimal number of clusters. Utility-wise, the warnings library helps manage potential warnings, and the re module supports regular expression operations, crucial for data cleaning. The wordcloud library allows for the visual representation of text data, and for progress tracking in long-running pandas operations, tqdm adds a practical progress bar.

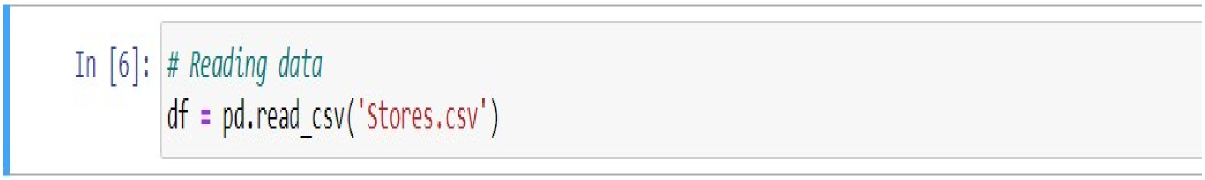
For "Store_Sales_Prediction":
In the Fig.4. "Store_Sales_Prediction" pandas and numpy are foundational for data manipulation and numerical analysis. seaborn and the matplotlib suite, which includes matplotlib.pyplot, facilitate the creation of both static and interactive visualizations. For a more dynamic and interactive user experience, we incorporate plotly's suite, including plotly.express, plotly.figure_factory, plotly.graph_objects, and plotly.subplots. The missingno library is particularly useful for handling and visualizing missing data, which is a common issue in real-world datasets. For output styling, term color enhances our textual output with color aiding in the differentiation of output significance. The machine learning and evaluation process is supported by sklearn.cluster for implementing the KMeans algorithm. The sklearn.metrics module provides evaluation metrics, like the silhouette score, which are essential for model assessment. Lastly, yellowbrick.cluster's KElbowVisualizer is again utilized for determining the best number of clusters for the KMeans algorithm.

4 Importing Files



```
In [6]: import os

        file_path = "C:/Users/aravi/OneDrive/Desktop/RIC/Online_Retail.csv"  # Use your preferred method for the file path
        file_exists = os.path.exists(file_path)
        print(file_exists)

        True

In [7]: import pandas as pd

        try:
            data = pd.read_csv('Online_Retail.csv', encoding='ISO-8859-1')
        except UnicodeDecodeError:
            data = pd.read_csv('Online_Retail.csv', encoding='Windows-1252')
```

Fig.5. Importing dataset file for Customer segmentation

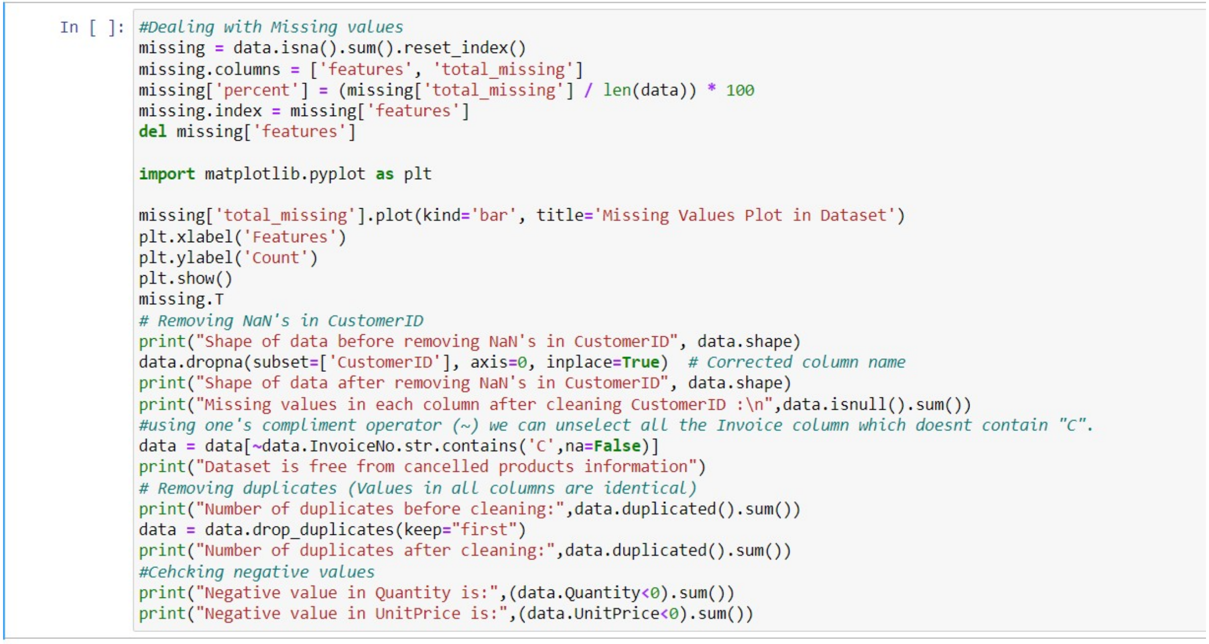


```
In [6]: # Reading data
        df = pd.read_csv('Stores.csv')
```

Fig.6. Importing dataset file for Sales Prediction

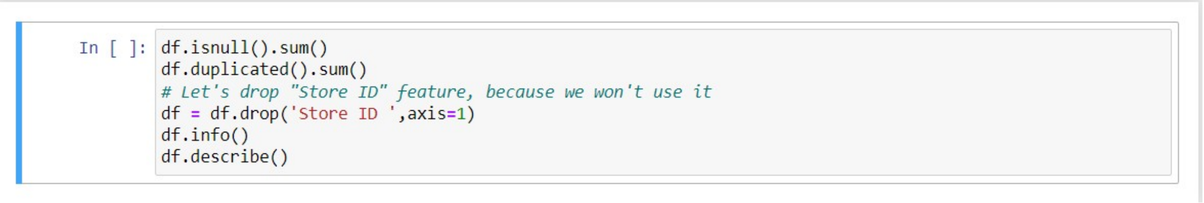
In the "Importing File" section of Customer segmentation Fig.5., guides the loading the "Online_Retail.csv" dataset into the project. Using the pandas library attempted to read the file with standard encodings, falling back to an alternative if the first attempt fails. It's important to check that the file path matches where the stored dataset on local machine to ensure the data loads correctly. Similarly for the Sales Prediction Fig.6. also pandas library is used in this research to load the dataset “store.csv” from local machine.

5 Preprocessing



```
In [ ]: #Dealing with Missing values
        missing = data.isna().sum().reset_index()
        missing.columns = ['features', 'total_missing']
        missing['percent'] = (missing['total_missing'] / len(data)) * 100
        missing.index = missing['features']
        del missing['features']

        import matplotlib.pyplot as plt

        missing['total_missing'].plot(kind='bar', title='Missing Values Plot in Dataset')
        plt.xlabel('Features')
        plt.ylabel('Count')
        plt.show()
        missing.T
        # Removing NaN's in CustomerID
        print("Shape of data before removing NaN's in CustomerID", data.shape)
        data.dropna(subset=['CustomerID'], axis=0, inplace=True)  # Corrected column name
        print("Shape of data after removing NaN's in CustomerID", data.shape)
        print("Missing values in each column after cleaning CustomerID :\n",data.isnull().sum())
        #using one's compliment operator (~) we can unselect all the Invoice column which doesnt contain "C".
        data = data[~data.InvoiceNo.str.contains('C',na=False)]
        print("Dataset is free from cancelled products information")
        # Removing duplicates (Values in all columns are identical)
        print("Number of duplicates before cleaning:",data.duplicated().sum())
        data = data.drop_duplicates(keep="first")
        print("Number of duplicates after cleaning:",data.duplicated().sum())
        #Cehcking negative values
        print("Negative value in Quantity is:",(data.Quantity<0).sum())
        print("Negative value in UnitPrice is:",(data.UnitPrice<0).sum())
```

Fig.7. Preprocessing steps for Customer segmentation data



```
In [ ]: df.isnull().sum()
        df.duplicated().sum()
        # Let's drop "Store ID" feature, because we won't use it
        df = df.drop('Store ID ',axis=1)
        df.info()
        df.describe()
```

Fig.8. Preprocessing steps of Store Sale prediction data

In this research on Customer segmentation and Store_Sales_Prediction dataset has been pre-processed to improve accuracy of the experiment (Zelaya, 2019). In Fig.7 and Fig.8. shows the steps used for Customer segmentation and Store_Sales_Prediction preprocessing respectively.

In both " Customer segmentation " and "Store_Sales_Prediction" experiment of this research have applied several data preprocessing steps which are crucial for accurate analysis. These steps include cleaning the data by removing duplicates and handling missing values. We also perform data transformation where necessary such as converting data types for proper analysis and scaling numerical values to standardize ranges for machine learning models. And also feature extraction is conducted to create new variables that can provide more insight during the modeling phase. These preprocessing tasks ensure that the data is clean, well-structured, and ready for the subsequent stages of our analysis.

6 Model Training

In the " Customer segmentation " Experiment model training is performed by fitting the KMeans algorithm to the customer data after preprocessing. This helps in determine the best number of clusters by analyzing the inertia of different cluster counts and choosing the one that provides a balance between cluster compactness and quantity.

In the "Store_Sales_Prediction" Experiment training is done by various predictive models by dividing dataset into training and test sets, ensuring that each model is given the opportunity to learn from the data and then validated on unseen data. The models are fine tuned to predict store sales with the goal of achieving the lowest possible prediction error.

7 Model Building

In this research on Customer segmentation and Store_Sales_Prediction have experimented building some Machine learning models (Willi Richart, 2013).
In the "Customer segmentation" of this research utilized the KMeans clustering algorithm to segment customers into groups based on purchasing patterns. The number of clusters is optimized using an elbow method visualization which helps in determining the most appropriate cluster count for the segmentation.

For the "Store_Sales_Prediction" experiment used various regression models including Random Forest Regressor, Decision tree Regressor, Linear Regressor, LGBM Regressor, XGB Regressor and CatBoost Regressor are employed to forecast sales. Each model's performance is rigorously evaluated using cross-validation techniques to ensure that our sales predictions are both accurate and reliable.

8 References

Willi Richart, L. P. C., 2013. *Building Machine learning systems with Python.* Birmingham: s.n.

Zelaya, C. V. G., 2019. *Towards Explaining the Effects of Data Preprocessing on Machine Learning.* Macao, China: s.n.