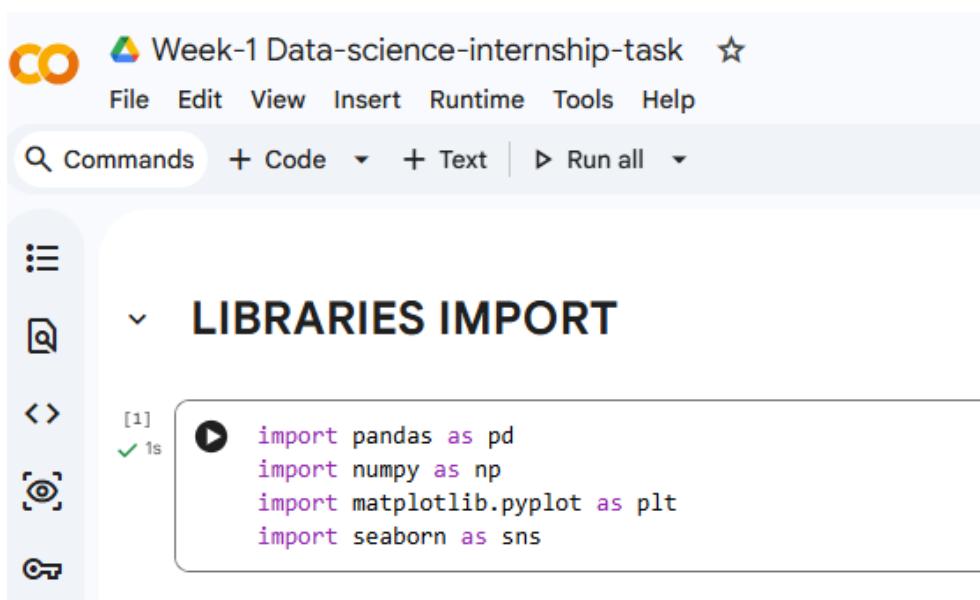
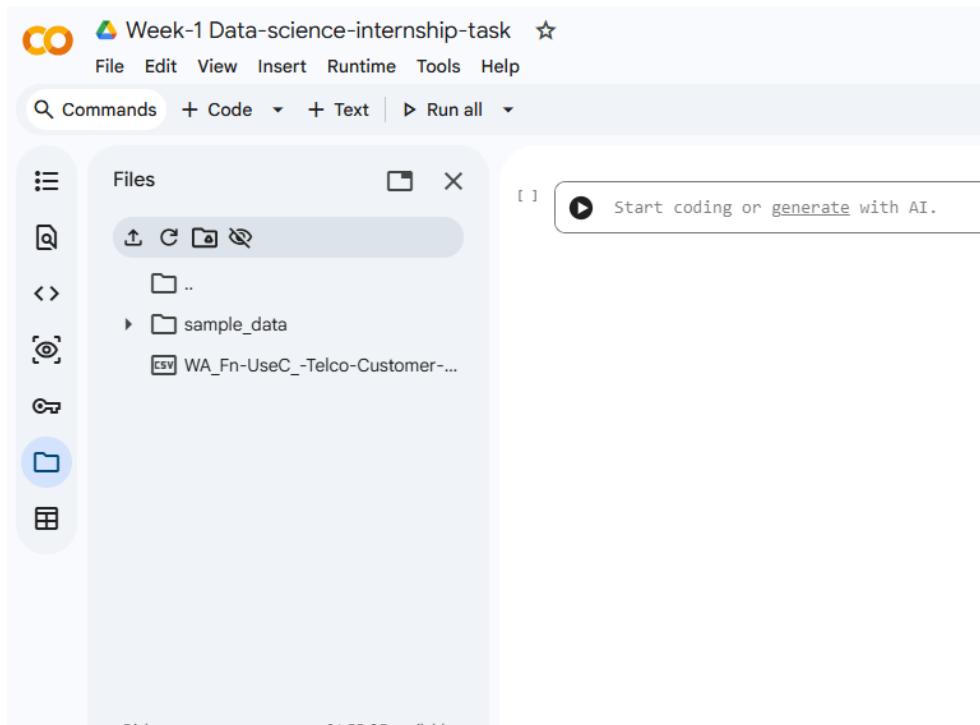


Introduction: This report documents the work completed for assignment in the Data Science Internship. The objective of this task was to perform advanced data cleaning and preprocessing on a real-world business dataset. The focus was on identifying data quality issues and applying professional techniques to make the dataset ready for analysis.



The screenshot shows a Jupyter Notebook interface with a code cell containing `df = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")` and `df.head()`. Below the code is a preview of the first five rows of the DataFrame:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	... DeviceProtection	TotalCharges
0	VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No

Dataset Description: The Telco Customer Churn dataset was selected for this assignment. It contains detailed information about telecom customers including demographic details, billing information, and service usage. The dataset included several missing values, inconsistent formats, and potential outliers, making it suitable for this cleaning task.

The screenshot shows a Jupyter Notebook interface with a code cell containing:

```
[6] df.shape
df.info()
df.describe(include='all')
```

The output shows the DataFrame structure:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   customerID      7043 non-null   object 
 1   gender          7043 non-null   object 
 2   SeniorCitizen   7043 non-null   int64  
 3   Partner         7043 non-null   object 
 4   Dependents     7043 non-null   object 
 5   tenure          7043 non-null   int64  
 6   PhoneService    7043 non-null   object 
 7   MultipleLines   7043 non-null   object 
 8   InternetService 7043 non-null   object 
 9   OnlineSecurity  7043 non-null   object 
 10  OnlineBackup    7043 non-null   object 
 11  DeviceProtection 7043 non-null   object 
 12  TechSupport    7043 non-null   object 
 13  StreamingTV    7043 non-null   object 
 14  StreamingMovies 7043 non-null   object 
 15  Contract        7043 non-null   object 
 16  PaperlessBilling 7043 non-null   object
```

```

Week-1 Data-science-internship-task
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
    19 TotalCharges    7843 non-null object
    20 Churn        7843 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
[1]: df.describe()
   customerID  gender  SeniorCitizen  Partner  Dependents  tenure  PhoneService  MultipleLines  InternetService  OnlineSecurity  ...
count      7043     7043  7043.000000     7043  7043.000000     7043     7043     7043     7043  7043  ...
unique      7043      2           NaN       2           2           NaN       2           3           3           3           ...
top  3186-AJIEK      Male          NaN      No          No          NaN      Yes          No  Fiber optic      No  ...
freq         1      3555          NaN      NaN      3641      4933          NaN      6361      3390      3096      3498  ...
mean        NaN        NaN  0.162147      NaN        NaN      32.371149      NaN        NaN        NaN        NaN        NaN  ...
std         NaN        NaN  0.368612      NaN        NaN      24.559481      NaN        NaN        NaN        NaN        NaN  ...
min         NaN        NaN  0.000000      NaN        NaN  0.000000      NaN        NaN        NaN        NaN        NaN  ...
25%        NaN        NaN  0.000000      NaN        NaN  9.000000      NaN        NaN        NaN        NaN        NaN  ...
50%        NaN        NaN  0.000000      NaN        NaN  29.000000      NaN        NaN        NaN        NaN        NaN  ...
75%        NaN        NaN  0.000000      NaN        NaN  55.000000      NaN        NaN        NaN        NaN        NaN  ...
max         NaN        NaN  1.000000      NaN        NaN  72.000000      NaN        NaN        NaN        NaN        NaN  ...

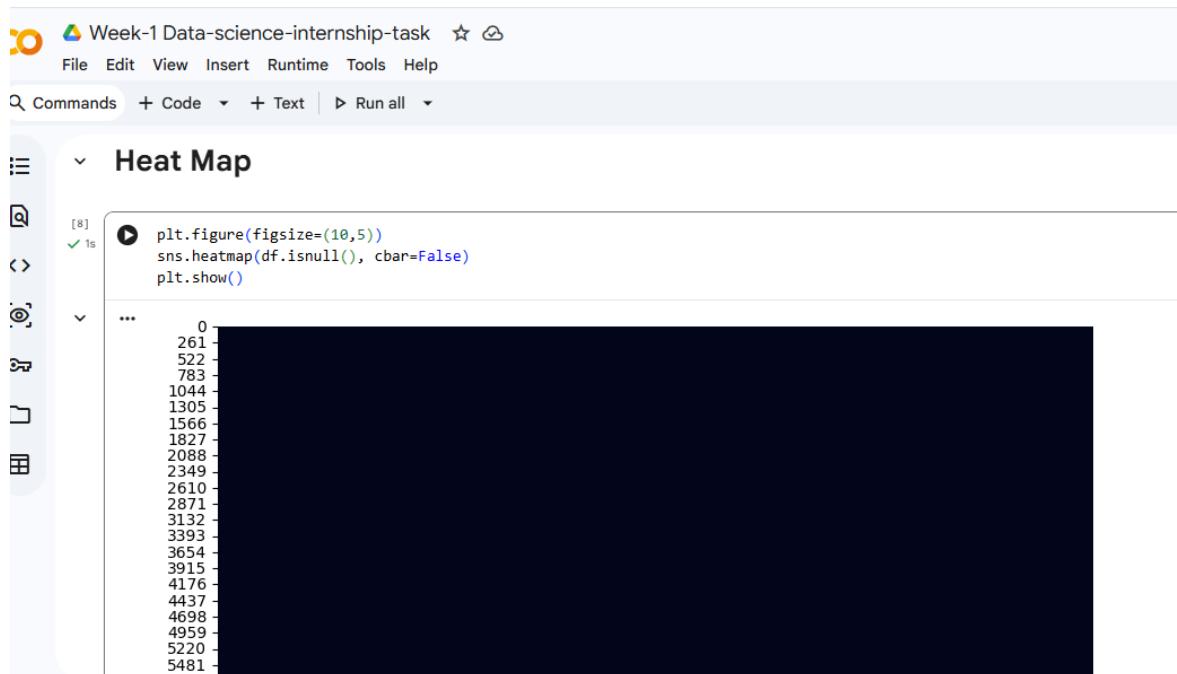
```

Missing Value Analysis: An in-depth missing value analysis was conducted to understand which columns required attention. Both tabular summaries and visualization techniques were used to analyze the distribution of missing data.

```

df.isnull().sum()
[7]: 0s
[7]: ... 0
[7]:     customerID  0
[7]:     gender      0
[7]:     SeniorCitizen  0
[7]:     Partner      0
[7]:     Dependents   0
[7]:     tenure       0
[7]:     PhoneService  0
[7]:     MultipleLines 0
[7]:     InternetService  0
[7]:     OnlineSecurity  0

```



Data Type Correction: During preprocessing, the TotalCharges column was found to contain non-numeric values stored as text. This issue was corrected by converting the column into proper numeric format to ensure consistency and accuracy.

The table shows the first few rows of a DataFrame with various columns and their corresponding values. The 'TotalCharges' column has been converted to numeric type.

```

df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df

```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	... DeviceProtection
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...
...
7499	6840-MAIA	Male	n	Max	Max	Max	Max	Max	Max	Max	Max

Missing Value Imputation: Two different imputation techniques were applied. First, statistical imputation using mean and mode was used for appropriate columns. Then, KNN Imputation was implemented as an advanced technique. After imputation, the dataset was verified to ensure no missing values remained.

```
[12] df['TotalCharges'].fillna(df['TotalCharges'].mean(), inplace=True)
df['PaymentMethod'].fillna(df['PaymentMethod'].mode()[0], inplace=True)

... /tmp/ipython-input-3296583753.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an in-place method will never work because the intermediate object on which we are setting values always behave
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead

df['TotalCharges'].fillna(df['TotalCharges'].mean(), inplace=True)
/tmp/ipython-input-3296583753.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an in-place method will never work because the intermediate object on which we are setting values always behave
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead

df['PaymentMethod'].fillna(df['PaymentMethod'].mode()[0], inplace=True)
```

Outlier Detection & Treatment:

Outliers in the MonthlyCharges column were detected using the IQR method. Identified extreme values were treated through documented filtering to improve the reliability of the dataset.

```
[13] Q1 = df['MonthlyCharges'].quantile(0.25)
Q3 = df['MonthlyCharges'].quantile(0.75)
IQR = Q3 - Q1

lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR

... (7043, 21)
```

Final Dataset Export:

The screenshot shows two Jupyter Notebook sessions.

Session 1: Titled "Check Final Data". It displays the first five rows of a DataFrame named "df".

```
[11]: df.head()
df.tail()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	... DeviceProtection	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	No
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	Yes

5 rows x 21 columns

Session 2: Titled "Cleaned CSV save". It shows the code used to save the cleaned DataFrame to a CSV file.

```
[12]: df.to_csv("cleaned_customer_churn_data.csv", index=False)
```

The sidebar shows a file tree with a "sample_data" folder containing "WA_Fn-UseC_-Telco-Customer-...csv" and "cleaned_customer_churn_data.csv".

Conclusion: This assignment helped me understand how real-world business data often contains many quality issues. By applying Python-based preprocessing techniques, I was able to convert messy data into a clean and structured format. The final dataset is now consistent and ready for further analytics and machine learning tasks.