Data Collection and Preprocessing Phase

Date	4 June,2024
Team ID	SWTID1719938571
Project Title	Walmart Sales Analysis for Retail Industry with Machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing

In this phase, we will statistically analyze dataset variables to identify patterns and outliers. Python will be used for preprocessing tasks, including normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring the quality of data for subsequent analysis and modeling. This process will form a strong foundation for generating reliable insights and accurate predictions.

Section	**Description**
Data Overview	<u>Dimensions:</u> 421570 rows ×17 columns <u>Descriptive Statistics:</u> `merged_data.describe()` count 421570.000000 421570.000000 421570.000000 421570.000000 421570.000000 421570.000000 421570.000000 421570.000000 421570.000000 421570.000000 mean 22.200546 44.260317 15981.258123 60.090059 3.361027 2590.074819 879.974298 468.087665 1083.132268 1662.772385 std 12.785297 30.492054 22711.183519 18.447931 0.458515 6052.385934 5084.538801 5528.873453 3894.529945 4207.829321 min 1.000000 1.000000 -4988.940000 -2.060000 2.472000 0.000000 -265.760000 -29.100000 0.000000 0.000000 25% 11.000000 18.000000 2079.650000 46.680000 2.933000 0.000000 0.000000 0.000000 0.000000 0.000000 50% 22.000000 37.000000 7612.030000 62.090000 3.452000 0.000000 0.000000 0.000000 0.000000 0.000000 75% 33.000000 74.000000 20205.852500 74.280000 3.738000 2809.050000 2.200000 4.540000 425.290000 2168.040000 max 45.000000 99.000000 693099.360000 100.140000 4.468000 88646.760000 104519.540000 141630.610000 67474.850000 108519.280000
Univariate Analysis	Univariate Analysis `merged_data["Type"].value_counts()` Type A 214993 B 162752 C 42473 Name: count, dtype: int64

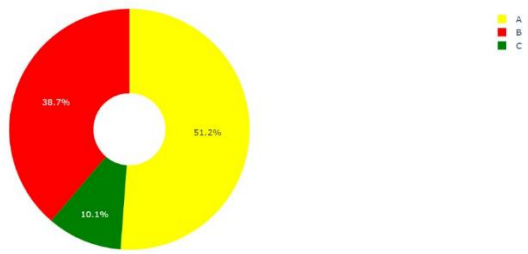
```
[29] print("Maximum Weekly Sales:", merged_data["Weekly_Sales"].max())
     print("Minimum Weekly Sales:", merged_data["Weekly_Sales"].min())
     print("Average Weekly Sales:", merged_data["Weekly_Sales"].mean())
```

```
Maximum Weekly Sales: 693099.36
Minimum Weekly Sales: 0.0
Average Weekly Sales: 16031.557675777807
```

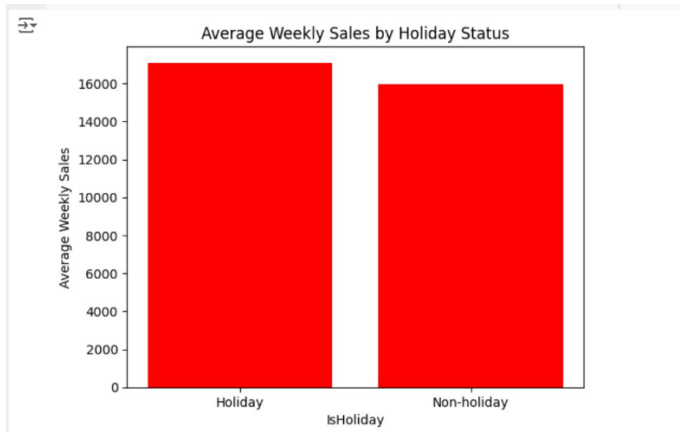
```
print(f"Maximum Temperature: {merged_data['Temperature'].max():.2f}°C")
print(f"Minimum Temperature: {merged_data['Temperature'].min():.2f}°C")
print(f"Average Temperature: {merged_data['Temperature'].mean():.2f}°C")
```

```
Maximum Temperature: 100.14°C
Minimum Temperature: 5.54°C
Average Temperature: 60.10°C
```

Pie Chart for Type of Store

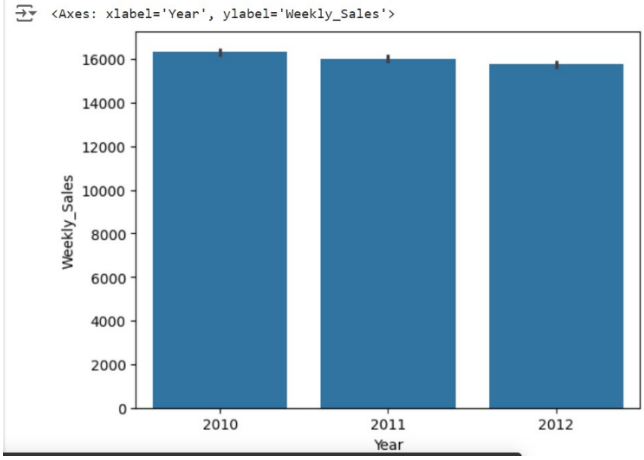


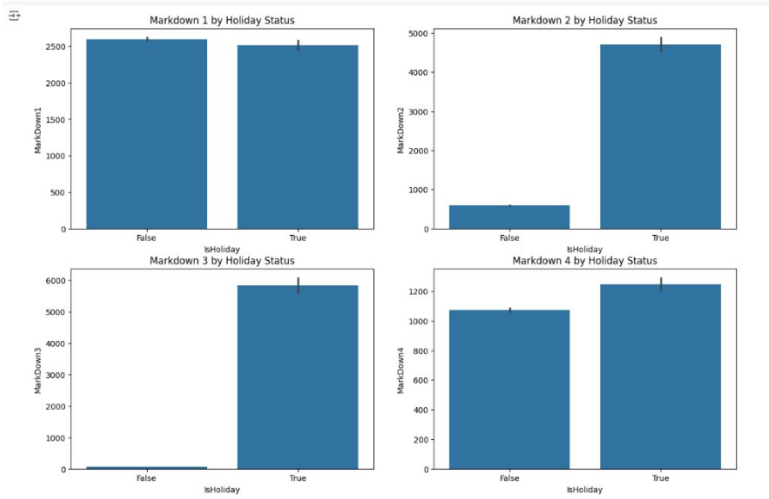
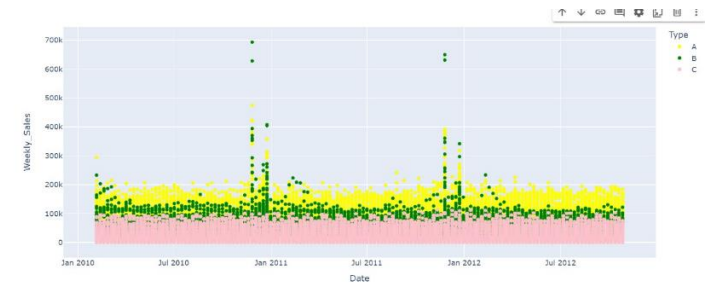
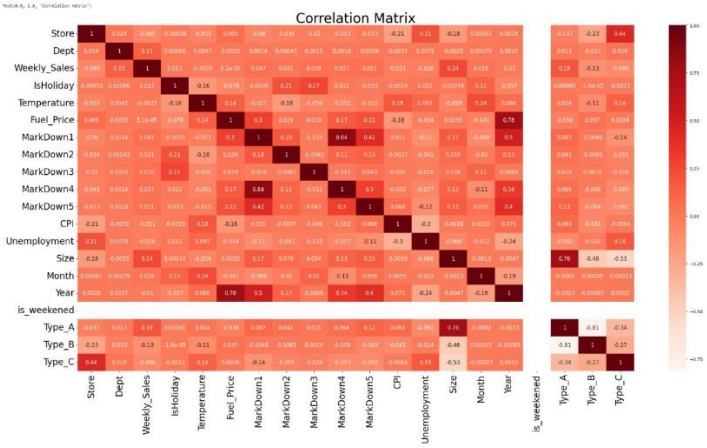
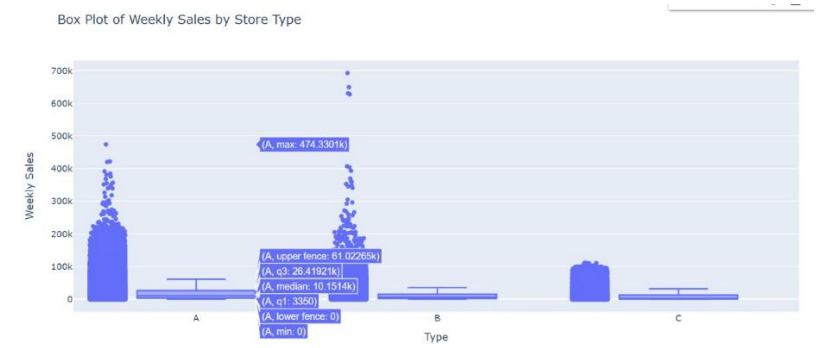
Bivariate Analysis




```
# bar graph
sns.barplot(x=merged_data["Year"],y=merged_data["Weekly_Sales"])
```

```
<Axes: xlabel='Year', ylabel='Weekly_Sales'>
```



	
Multivariate Analysis	 
Outliers and Anomalies	

Data Preprocessing Code Screenshots

Loading Data

```
[2] # reading all the csv files
    stores = pd.read_csv("stores.csv")
    features = pd.read_csv("features.csv.zip")
    train = pd.read_csv("train.csv.zip")
    test = pd.read_csv("test.csv.zip")
```

```
[12] # merging all the csv files
     # all the csv files have store column in common.
     merged_data = train.merge(features,on=["Store","Date"] ,how= 'inner').merge(stores ,on=["Store"] ,how ='inner')
```

```
merged_data
```

	Store	Dept	Date	Weekly_Sales	IsHoliday_x	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unempl
0	1	1	2010-02-05	24924.50	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	
1	1	2	2010-02-05	50605.27	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	
2	1	3	2010-02-05	13740.12	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	
3	1	4	2010-02-05	39954.04	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	
4	1	5	2010-02-05	32229.38	False	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	

Handling Missing Data

Code for identifying and handling missing values.

```
# Handling the null values
merged_data["MarkDown1"] = merged_data["MarkDown1"].replace(np.nan,0)
merged_data["MarkDown2"] = merged_data["MarkDown2"].replace(np.nan,0)
merged_data["MarkDown3"] = merged_data["MarkDown3"].replace(np.nan,0)
merged_data["MarkDown4"] = merged_data["MarkDown4"].replace(np.nan,0)
merged_data["MarkDown5"] = merged_data["MarkDown5"].replace(np.nan,0)
```

Data Transformation

Code for transforming variables (scaling, normalization).

```
# changing the categorical value type into numbers
merged_data = pd.get_dummies(merged_data,columns=["Type"])
```

```
merged_data["is_weekened"].replace({False:0,True:1},inplace=True)
```

```
merged_data["IsHoliday"].replace({False:0,True:1},inplace=True)
```

```
# Scaling the data
sc = StandardScaler()
X = sc.fit_transform(X)
print(X)
```

Feature Engineering

Code for creating new features or modifying existing ones.

```
# Date ,type and isholiday needs to be converted to numbers
merged_data["Date"] = pd.to_datetime(merged_data["Date"])
merged_data.loc[:,"DayofWeek"] =merged_data.loc[:,"Date"].dt.day_name()
merged_data.loc[:,"Month"]  = merged_data.loc[:,"Date"].dt.month
merged_data.loc[:,"Year"] = merged_data.loc[:,"Date"].dt.year
```

Save Processed Data

Code to save the cleaned and processed data for future use.