Data Collection and Preprocessing Phase

Date	07 April 2024
Team ID	722312
Project Title	Walmart Sales Analysis For Retail Industry With Machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	**Description**
Data Overview	Basic statistics, dimensions, and structure of the data.
Univariate Analysis	Exploration of individual variables (mean, median, mode, etc.).
Bivariate Analysis	Relationships between two variables (correlation, scatter plots).
Multivariate Analysis	Patterns and relationships involving multiple variables.
Outliers and Anomalies	Identification and treatment of outliers.
Data Preprocessing Code Screenshots	

Loading Data	Code to load the dataset into the preferred environment (e.g., Python, R). [3]: ``` train = pd.read_csv("F:/smartinternz/train.csv") features = pd.read_csv("F:/smartinternz/features.csv") stores = pd.read_csv("F:/smartinternz/stores.csv") test = pd.read_csv("F:/smartinternz/test.csv") train.head() ```

Handling Missing Data	Code for identifying and handling missing values.
	
	`data.isnull().sum()`
	
	Store 0
	Dept 0
	Date 0
	Weekly_Sales 0
	IsHoliday_x 0
	Temperature 0
	Fuel_Price 0
	MarkDown1 270889
	MarkDown2 310322
	MarkDown3 284479
	MarkDown4 286603
	MarkDown5 270138
	CPI 0
	Unemployment 0
	IsHoliday_y 0
	Type 0
	Size 0
	dtype: int64
	
	`data['MarkDown1']=data['MarkDown1'].replace(np.nan,0)`
	`data['MarkDown2']=data['MarkDown2'].replace(np.nan,0)`
	`data['MarkDown3']=data['MarkDown3'].replace(np.nan,0)`
	`data['MarkDown4']=data['MarkDown4'].replace(np.nan,0)`
	`data['MarkDown5']=data['MarkDown5'].replace(np.nan,0)`
	
	`data.isnull().sum()`
	
	Store 0
	Dept 0
	Date 0
	Weekly_Sales 0
	IsHoliday_x 0
	Temperature 0
	Fuel_Price 0
	MarkDown1 0
	MarkDown2 0
	MarkDown3 0
	MarkDown4 0
	MarkDown5 0
	CPI 0
	Unemployment 0
	IsHoliday_y 0

Data Transformation	Code for transforming variables (scaling, normalization).
	`data=data[data['Weekly_Sales']>=0]`
	`data.describe()`

	Store	Dept	Weekly_Sales	Temperature	Fuel_Price	MarkDo
count	420285.000000	420285.000000	420285.000000	420285.000000	420285.000000	420285.00(
mean	22.195477	44.242771	16030.329773	60.090474	3.360888	2590.18
std	12.787213	30.507197	22728.500149	18.448260	0.458523	6053.22
min	1.000000	1.000000	0.000000	-2.060000	2.472000	0.00(
25%	11.000000	18.000000	2117.560000	46.680000	2.933000	0.00(
50%	22.000000	37.000000	7659.090000	62.090000	3.452000	0.00(
75%	33.000000	74.000000	20268.380000	74.280000	3.738000	2801.50(
max	45.000000	99.000000	693099.360000	100.140000	4.468000	88646.76(

Feature Engineering

Code for creating new features or modifying existing ones.

```
type_dummies = pd.get_dummies(data['Type'], prefix='Type')
type_dummies = type_dummies.astype(int)
data = pd.concat([data, type_dummies], axis=1)
data.drop(columns=['Type'], inplace=True)
```

```
data
```

Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday_y	Size	Type_A	Type_B	Type_C
42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	False	151315	1	0	0
42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	False	151315	1	0	0
42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	False	151315	1	0	0
42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	False	151315	1	0	0
42.31	2.572	0.00	0.00	0.0	0.00	0.00	211.096358	8.106	False	151315	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	False	118221	0	1	0
58.85	3.882	4018.91	58.08	100.0	211.94	858.33	192.308899	8.667	False	118221	0	1	0

```
data['Date']=pd.to_datetime(data['Date'])
```

```
data['month']=data['Date'].dt.month
data['year']=data['Date'].dt.year
```

```
data[['Date','month','year']].head()
```

	Date	month	year
0	2010-02-05	2	2010
1	2010-02-05	2	2010
2	2010-02-05	2	2010
3	2010-02-05	2	2010
4	2010-02-05	2	2010

```
data['dateofweek_name']=data['Date'].dt.day_name()
data[['Date','dateofweek_name']].head()
```

	Date	dateofweek_name
0	2010-02-05	Friday
1	2010-02-05	Friday
2	2010-02-05	Friday
3	2010-02-05	Friday
4	2010-02-05	Friday

<table>
<tr>
<td>Feature Engineering</td>
<td>

```
data['is_weekend']=np.where(data['dateofweek_name'].isin(['Sunday','Saturday']),1,0)
data[['Date','is_weekend']]
```

	Date	is_weekend
0	2010-02-05	0
1	2010-02-05	0
2	2010-02-05	0
3	2010-02-05	0
4	2010-02-05	0
...	...	...
421565	2012-10-26	0
421566	2012-10-26	0
421567	2012-10-26	0
421568	2012-10-26	0
421569	2012-10-26	0

420285 rows × 2 columns

```
data['IsHoliday_x']=data['IsHoliday_x'].astype(int)
data['IsHoliday_y']=data['IsHoliday_y'].astype(int)
del data['dateofweek_name']
```

```
data.head()
```

kly_Sales	IsHoliday_x	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	...	MarkDown5	CPI	Unemployment	IsHoliday_y	Size	Type_A	Type_
24924.50	0	42.31	2.572	0.0	0.0	0.0	...	0.0	211.096358	8.106	0	151315	1	
50605.27	0	42.31	2.572	0.0	0.0	0.0	...	0.0	211.096358	8.106	0	151315	1	
13740.12	0	42.31	2.572	0.0	0.0	0.0	...	0.0	211.096358	8.106	0	151315	1	
39954.04	0	42.31	2.572	0.0	0.0	0.0	...	0.0	211.096358	8.106	0	151315	1	
32229.38	0	42.31	2.572	0.0	0.0	0.0	...	0.0	211.096358	8.106	0	151315	1	

```
data['Date'] = pd.to_datetime(data['Date'])
data['Month'] = data['Date'].dt.month
data['Year'] = data['Date'].dt.year
data['Is_Weekend'] = (data['Date'].dt.dayofweek >= 5).astype(int)
#data.drop(columns=['Date'], inplace=True)
```

```
del data['is_weekend']
data
```

</td>
</tr>
<tr>
<td>Save Processed Data</td>
<td>Code to save the cleaned and processed data for future use.

```
data.to_csv('D:/merged_data.csv',index=False)
```

</td>
</tr>
</table>