MSA 2024 Phase 2 - Part 1 Analysis and Preprocessing

In [ ]:
```
import pandas as pd
import numpy  as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn as skl
```

1. Find all variables and understand them

In [ ]:
```
# Choose W Store Sales as the dataset and merged the three tables
url_features="https://raw.githubusercontent.com/NZMSA/2024-Phase-2/main/data-science/0.%20Resources/datasets/W%20st
df_features=pd.read_csv(url_features)
df_features.info()

url_sales="https://raw.githubusercontent.com/NZMSA/2024-Phase-2/main/data-science/0.%20Resources/datasets/W%20store
df_sales=pd.read_csv(url_sales)
df_sales.info()

url_stores="https://raw.githubusercontent.com/NZMSA/2024-Phase-2/main/data-science/0.%20Resources/datasets/W%20stor
df_stores=pd.read_csv(url_stores)
df_stores.info()

# Merging the three tables by the unique keys
merged_data=pd.merge(df_sales, df_features, on=['Store','Date','IsHoliday'],how='left')
merged_data=pd.merge(merged_data, df_stores, on='Store', how='left')

merged_data=merged_data.sort_values(by=['Store','Dept','Date'])
```

2. Setting the labels and the distribution of values in columns

In [ ]:
```
#setting the following 12 weekly sales as labels
for i in range(1,13):
```

```
 merged_data[f'Weekly_Sales_{i+1}w']=merged_data.groupby(['Store','Dept'])['Weekly_Sales'].shift(-i)
print("The number of rows before data processing:", merged_data.shape[0])
merged_data=merged_data.dropna(subset=['Weekly_Sales_13w'])
print("The number of rows after data processing:", merged_data.shape[0])

# the distribution of values in columns
merged_data.info()
merged_selected_types=merged_data.select_dtypes(include=['float64', 'int64'])
mean=merged_selected_types.mean()
variance=merged_selected_types.var()
std=merged_selected_types.std()
quantiles=merged_selected_types.quantile([0,0.05,0.25,0.5,0.75,0.90,0.95,0.96,0.97,0.98,0.99,1])
print(f"\n Mean:\n{ mean} \n Variance :\n{variance} \n Standard deviation:\n{std} \n Quantiles:\n{quantiles}\n")

plt.figure(figsize=(10,6))
sns.heatmap(merged_selected_types.isnull(),cbar=False, cmap='viridis', yticklabels=False)
plt.title('Missing Values Heatmap in Merged Data')
plt.show()
```

3. Clean data

In [ ]:

```
#Data cleaning
# Consider the solution to process the missing values in columns
for i in range(1, 6):
    print(f"the number of 0 in MarkDown{i} is: {(merged_data[f'MarkDown{i}'] == 0).sum()}")
# Considering that the proportion of missing values in columns Markdown1-5 exceeds 70%, and there are valid values
# in order to avoid unexpected impacts on the model results, these columns will not be considered as input variable
# in the subsequent modeling process.

#Transfering the bool variable into numeric
merged_data['IsHoliday']=merged_data['IsHoliday'].astype(int)
# Convert "Type" to numeric type, create a mapping dictionary, and use the map method to convert the type to intege
type_mapping = {'A': 0, 'B': 1, 'C': 2}
merged_data['Type'] = merged_data['Type'].map(type_mapping)
# to avoid the influence of outliers in y lables, we drop the values which are lager than 90% quantile and smaller
data_frames = {}  # Used to store processed dataframes
for i in range(2,14):
    quantile_10 = merged_data[f'Weekly_Sales_{i}w'].quantile(0.10)
    quantile_90 = merged_data[f'Weekly_Sales_{i}w'].quantile(0.90)
```

```
    filtered_data = merged_data[(merged_data[f'Weekly_Sales_{i}w'] >= quantile_10) & (merged_data[f'Weekly_Sales_{i
    # Delete sales data for other weeks, ensuring that there is only one y label at a time
    cols_to_keep = [col for col in filtered_data.columns if col == f'Weekly_Sales_{i}w' or 'Weekly_Sales_' not in c
    filtered_data = filtered_data[cols_to_keep]
    data_frames[f'Weekly_Sales_{i}w'] = filtered_data

for week, df in data_frames.items():
    print(f"Number of rows retained for {week}:\n {df.info()} ")
```

4. Visualise data

In [ ]:

```
# Draw the histograms and box graphics to show the distribution of values and outliers intuitively and directly
# Set the style of graphics
sns.set(style="whitegrid")

# Iterate the DataFrame
for week, df in data_frames.items():
    # Select columns of type float64 and int64
    numerical_df = df.select_dtypes(include=['float64', 'int64'])

    for column in numerical_df.columns:
        # Draw histograms
        plt.figure(figsize=(10, 6))
        sns.histplot(numerical_df[column], bins=10, kde=True)
        plt.title(f'Histogram of {column} with {week}')
        plt.xlabel(f'{column}')
        plt.ylabel('Frequency')
        plt.show()
        plt.close()

        # Draw box diagrams
        plt.figure(figsize=(10, 6))
        sns.boxplot(x=numerical_df[column])
        plt.title(f'Box Plot of {column} in Data with {week}')
        plt.xlabel(column)
        plt.ylabel('Value')
        plt.show()
        plt.close()
```

5. Identify correlated variables

In [ ]:
```
# Correlation coefficient check
for week, df in data_frames.items():
    numerical_df = df.select_dtypes(include=['float64', 'int64'])
    # Calculate the correlation matrix
    correlation = numerical_df.corr(method="spearman")
    print(f"Spearman Rank Correlation:\n {correlation}")

    # Create a heatmap with seaborn
    plt.figure(figsize=(10, 10))
    sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
    plt.title(f'Feature Correlation Matrix Heatmap with {week}')
    plt.savefig('Feature Correlation Matrix Heatmap.png', dpi=300)
    plt.show()
```

6. Summary

Data Selection and Preparation: I selected the "w" store dataset, merging features, stores, and sales tables, with the goal of predicting sales for the next 12 weeks.

Data Analysis: The dataset contains 382,955 rows and 27 columns. Key findings include missing values in MarkDown1-5 exceeding 70%, and "Store" and "Dept" being categorical despite being numeric."Type" and "IsHoliday" need conversion to numeric formats. Significant variability was noted in several columns.

Data Cleaning: Data cleaning involved converting data types, handling missing values, and processing outliers. Separate datasets were stored for different target variables.

Visualization: Heatmaps, histograms, and box plots helped visualize missing values and data distribution, enhancing dataset understanding.

Correlation Analysis: Strong correlations were found between weekly sales and the target, as well as between other variables like CPI, unemployment, and Markdown values. This informed potential reductions in model inputs.

Conclusion: The initial data exploration and analysis provide a strong foundation for subsequent modeling.