Data Science Project Report (Week 8)

Number of team members: 1
Member Name: Yan-Ping Yu
Email: yy3407@columbia.edu
Nationality: Taiwan
College: Columbia University
Specialization: Data Science

1. Problem description:

Our client operates within the beverage industry in Australia. The primary challenge is to develop multivariate forecasting models, utilizing machine learning or deep learning techniques, to accurately predict weekly demand for their products. This initiative is driven by the need to analyze historical time series data, incorporating various factors that influence demand, to forecast the quantity of items required by customers each week.

2. What type of data you have got for analysis:

DataSet Links to an external site.

3. Data Understanding:

1. **Introduction to the Data**

   Source: The data are provided by Data Glacier canvas.

   Scope and Scale: There are 1219 rows and 12 features in the dataset. The time period covers from Feb 2017 to Dec 2020.

2. **Data Description**

   Product: Identifier for the product.

   Date: Sales recorded weekly from Feb 2017 to Dec 2020.

   Sales: Weekly sales quantity.

   Price Discount (%): The percentage discount applied to the product's price.

   In-Store Promo: Indicator (0 or 1) whether there was an in-store promotion.

   Catalogue Promo: Indicator (0 or 1) whether there was a catalogue promotion.

   Store End Promo: Indicator (0 or 1) whether there was a store end promotion.

   Google Mobility: An indicator (0 or 1) representing restrictions on mobility,

potentially due to COVID-19.
Covid_Flag: An indicator (0 or 1) denoting whether there were COVID-19 restrictions.
V_DAY: Indicator (0 or 1) for Valentine's Day.
EASTER: Indicator (0 or 1) for Easter.
CHRISTMAS: Indicator (0 or 1) for Christmas, though it doesn't apply within this dataset's timeframe.

4. **What are the problems in the data (number of NA values, outliers , skewed etc)? What approaches you are trying to apply on your data set to overcome problems like NA value, outlier etc and why?**

1. Missing Values

```
# Check NA values
data.isna().sum()
```

```
Product              0
date                 0
Sales                0
Price Discount (%)   0
In-Store Promo       0
Catalogue Promo      0
Store End Promo      0
Google_Mobility      0
Covid_Flag           0
V_DAY                0
EASTER               0
CHRISTMAS            0
dtype: int64
```

Fortunately, my dataset doesn't have any missing values. This is a relief because missing data can significantly impact the analysis, leading to biased results if not handled properly. However, had there been missing values, I would have considered techniques like imputation or removal, depending on the context and the proportion of missing data.

2. Outliers

```
def numeric_outlier(column):
    mean = column.mean()
    std = column.std()
    threshold = 3
    z_score = (column - mean)/std
    outlier = data[np.abs(z_score) > threshold]
    return outlier  #True for outliers
```

I've identified outliers using the formula above in my dataset, particularly in features like **Google_Mobility** and various promotional indicators. Outliers can skew the analysis and potentially mislead any conclusions drawn from the data. To tackle this,

I'm considering several strategies:

- **Winsorizing**: This will limit extreme values in the dataset, reducing the influence of outliers without completely removing them. It's a balanced approach that retains most of the data integrity.
- **Transformation**: I plan to apply transformations such as logarithmic or square root transformations to variables that are significantly skewed. This should help in normalizing the distributions and reducing the impact of outliers.

3.Skewness

```
def skew(column):
    num = column.skew()
    return num
```

The skewness identified by the formula above in my dataset, especially in **Sales**, **Catalogue Promo**, and **Google_Mobility**, among others, indicates that the data distribution deviates from normality. This skewness can be problematic for algorithms that assume normally distributed data. Here's how I plan to address it:

- **Data Transformation**: I'm leaning towards applying transformations like the Box-Cox or Yeo-Johnson methods, especially for heavily skewed data. These transformations can help in stabilizing variance and making the data more normal-like, which is beneficial for subsequent analysis.
- **Normalization**: Standardizing features by removing the mean and scaling to unit variance could also be a useful preprocessing step, particularly for models sensitive to the scale of the data.