

# Movie Recommendation System

Name: Olena Panchenko

Email: hel.panchenko@gmail.com

Country: United Kingdom

Specialisation: Data Science

Link: [https://github.com/helenapanchenko/DS\\_Internship\\_Glacier/tree/master/Week9](https://github.com/helenapanchenko/DS_Internship_Glacier/tree/master/Week9)

<b>Problem Description.....</b>	<b>1</b>
<b>Data Preprocessing.....</b>	<b>1</b>
Data Cleansing.....	1
Dealing with Missing Values.....	1
Dealing with Outliers.....	2
Dealing with Duplicates.....	2
Data Transformation.....	2
Change of data type.....	2

## Problem Description

The large company is in the beverages business in Australia. They sell their products through various supermarkets and engage in heavy promotions throughout the year. Various factors like holidays and seasonality also influence their demand. They needed a forecast of each of the products at the item level every week in weekly buckets.

## Data Preprocessing

### Data Cleansing

#### Dealing with Missing Values

None of the columns contain missing values.

#### Dealing with Outliers

We examined the 'Sales' column and the 'Google\_Mobility' column using the Z-score Method which represents the number of standard deviations a data point is from the mean. A common threshold to identify outliers is a Z-score above 3 or below -3 so I'll use it.

**'Sales'**: 26 records with exceptionally high sales were identified as outliers. However, I will not address them, as the significant discounts on these items during those weeks (ranging from 50% to 80%) suggest that the sales increases are not anomalies.

**'Google\_Mobility'**: 36 records with unusually low mobility values were identified as outliers. However, I will not address them, as these values all occurred during the COVID-19 pandemic and are therefore considered normal under the circumstances.

## Dealing with Duplicates

We have no duplicates.

## Data Transformation

### Change of data type

The data type changes applied to the DataFrame involved several key transformations:

- The 'Product' column values were converted from object to string to ensure consistency in text data representation.
- For the 'date' column, values were transformed from object to date format. Although the column's data type shows as an object because it contains Python datetime.date objects, this format is generally suitable for performing date-based operations or comparisons.
- The 'Price Discount (%)' column was converted from object to float for easier numerical analysis.