Data Wrangling Report

Capstone Project 1

Raviteja Bodla

Data Science Career Track

Springboard

INTRODUCTION:

This report is initial report for my capstone project as a part of 'Data Science Career track' Bootcamp with Springboard. This report describes the data wrangling techniques used as a part of this project. This objective of the capstone project is to predict what the customer will buy in the next order based on previous buying behavior. This objective of this particular report is to describe the techniques used in obtaining the data and cleaning the data.

DATA AND IT'S DESCRIPTION:

The data used for this project is obtained from Kaggle competition 'Instacart Market Basket Analysis' . The data contains 6 comma separated files (csv) files :

S.no	File	Shape	Attribues
1	aisles.csv	134 x 2	1) aisle_id 2) aisle
2	departments.csv	21 x 2	1) department_id 2) department
3	orders.csv	3.42 million x 7	1) order_id 2) user_id 3) eval_set 4) order_number 5) order_dow 6) order_hour_of_day 7) days_since_prior_order
4	order_products_prior.csv	3.24 million x 4	1) order_id 2) product_id 3) add_to_cart_order 4) reordered
5	order_products_train.csv	1.38 million x 4	1) order_id 2) product_id 3) add_to_cart_order 4) reordered
6	products.csv	4970 x 4	1) product_id 2) product_name 3) aisle_id 4) department_id

A detailed description of the data will be included in the final project report.

DATA IMPORT:

After downloading the data from Kaggle competition page(https://www.kaggle.com/c/instacart-market-basket-analysis), all the files in the file repository were obtained using glob function from 'glob' library. The files were uploaded using read_csv fuction from 'pandas' library.

DATA CLEANING:

As the data was downloaded from the Kaggle compition, the data was already clean. However, I would search for wrong data types, null values and outliers. After importing the data, names were assigned to each dataset using a for loop and *dataset.name* method, as there were many files and naming them will help in further processes.

1) **Data types**: After uploading the data it was observed that all the *_id variables in all the datasets got uploaded as data type 'int64', however these variables have discrete data and should be of type 'object'. Using dataset.series.astype('object') all the *_id variables in all the datasets were changed to 'object' data type.
2) **Null values**: Finding null values in all the data sets one after the other would be tedious and time consuming. Creating a function would make the process easier. For the purpose of finding the null values I have created a user-defined function (null_columns) and using a list of all the datasets and for loop, columns with null values were found.

   Only 'days_since_prior_order' column of 'orders' dataset had 206209 null values. All the other datasets had no null values. By observing all the null values, it was observed that 'days_since_prior_order' column of first order of all the customers are null values, so the null values can be filled with '0'. Using 'fillna()' function of pandas, all the null values were filled with 0.

3) **Outliers**: Datasets 'aisles', 'products' and 'departments' have only categorical data so these datasets were not checked for outliers. For datasets 'orders', 'orders_products_prior', 'orders_products_train' were checked for outliers. Values below 2.5 percentile and above 97.5 percentile were considered outliers for the purpose of this analysis. 'Percentile' function from numpy library was used inside a for loop to create array of 2.5 percentile value and 97.5 percentile value in a particular variable in the dataset and using this array, outlier values were sliced out of the dataset. After looping on all the required datasets, it was found that there were no outliers in any of the data sets.