EXTRACTION

1. SALES ZIP FILE (5 YEARS)

Zip file pulled from Kaggle containing 3 datasets sales, prices, and quant\_fe. The sales data contain how many pieces sold in a day for a product. It also contains valuable column such as category. The data range from 1 to 1966 days, approximately 5 years worth of sales. For the prices file it contains similar information to sales pieces but instead of pieces sold in the day columns it shows the price of the product. This allow us to view prices changes over time. The last file was quant\_fe but we were unable to determine was the values under the column as the link did not specify the meaning of the values. This file was disregarded and not used. The two files sales and price were read through jupyter notebook using the pandas module.

2. ONLINE PRODUCT DATABASE

The e-commerce product listing was extracted from dataworld as a csv file. It contains over 30,000 rows. Important columns in the dataset are unique id, brand name, regular price, sale price and category. This dataset was also read through jupyter notebook using the pandas module.

3. WALMART SALES 2010-2012

Data file was extracted from Kaggle. 3 datasets were used; features, stores, and training. The 3 csv files were read in jupyter notebook using the pandas module. The dataset contain weekly sales across 45 stores. It also contained socioeconomic variables such as CPI and unemployment rate. Other important columns were temperature and fuel prices.

4. API HOLIDAY

Data was collected from calendarific API. The holidays that were looked at were Christmas Day, Christmas Eve, Independence Day, New Year's Day, and Thanksgiving Day due to them being major holidays in the United States. A for loop was created to request get the dates from 2010-2012 in the United States for the 5 holidays that were selected. The column of the table are holiday name, year and holiday date.

5. WALMART DAILY STOCK DATA

Data was extracted from Kaggle containing daily stock prices from 1972-2020. Columns it contained were date, open, closing, high, low of stock price and the volume of stock sold. The dataset was read through jupyter notebook using the pandas module.

TRANSFORMATION

1. SALES ZIP FILE (5 YEARS)

For both sales and price, the day columns were transposed so the sales and prices changes can be viewed by row. The dataset did not specify when the data was collected so we were unable to determine the exact start or end date of the data. After doing some research it was discovered that Walmart is closed only on Christmas Day. Using this information, it was determined that there were days with 0 sales for all items and they were 365 days apart, one range being 366 days apart likely due to a leap year. This helped determine what day and month the sales were during. The only thing that could not be determined was the year but knowing the day and month can help in identifying trends and how sales is impacted from price changes. After determining the date, day and month were included into the dataset. Columns such as state\_id and cat\_id (category) were also included in replacement of their strings to help reduce storage space. Since the product ID was a string, in order to reduce storage a product ID table was created with integer id values. Various columns were also deleted as we did not have much information regarding it and their values. After cleaning it resulted in the below table:

SALES TABLE

UNIQUE PRODUCT ID TABLE STATE ID TABLE CAT ID TABLE

For the price file, a similar process was done like the sales dataset. Item, category and state were replaced with their integer ID. The day and month were also included in the dataset. The dataset for the price changes is very large and in order to gain insightful information it was decided that instead of showing price changes in every day a “Markup” and “Markdown” column was created. These columns would count the numbers of markdown and markdown every 6 months over the 5 years. It was decided to work initially with 6 months in order to simplify the data but next time every month may be considered instead to have a more detailed look at price changes. This was calculated by sort the dataset by the product name and day. If the next row price increase it would be counted as a Markup, if the next row was lower then it would be counted as a Markdown, and if there was no change it would continue onto the next row without counting anything. The function would sum these counts and then reset the sum once it reached 6 months and this process would iterate. From this function it resulted in the following table:

PRICE MAKRUP MARKDOWN TABLE (6 MONTH/5 YEARS)

2. ONLINE PRODUCT DATABASE

Various columns were deleted such as URL link, UPC and timestamp of data collection. The columns were deleted due to not being able to gain insightful information from them. The main columns cleaned was the categories column as it was formatted to include all it’s sub-categories. It was decided to focus on categories and not the sub-categories as the other datasets did not have the column so the online product database online alone would not be enough to get important findings from sub-categories. After splitting the first category from the column, the categories column was overwritten with only the first category. Once that was completed the string was then replaced with a category integer ID. The cleaned table is the following:

ONLINE PRODUCT TABLE

3. WALMART SALES 2010-2012

The 3 datasets were merge on stores for features, stores, and training and on inner to eliminate an NA’s. After columns were deleted such as the price markdown columns due to majority of the values being NA’s and it was not determined was the values meant for stores. The data was also broken down from store and category level but due to the category being a category integer ID and being unable to determine the actual category it is it was decided to group the weekly sales by store to remove category.

4. API HOLIDAY

The holiday table required no transformation or cleaning as the data frame was created during the extraction process of getting the holiday dates from the holiday API. This table was used to merge on the Walmart sales 2010-2012 dataset in order to identify when holidays occurs and how it impacted the weekly sales.

5. WALMART DAILY STOCK DATA

The data was filtered on the beginning and end data of the 2010-2012 Walmart dataset. Since the stock data is daily a week column was added to identify the day that follow under that week. This was done using the datetime and timedelta module and function.

LIMITATION/PROBLEMS/WHAT TO DO FOR NEXT TIME

One of the issues with the sales and price column was since it doesn’t give us the exact start date it’s hardier to compare the performance of sales to the other tables. It’ll mainly be used to identify trends and relationship between sales and price changes.

Another problem that occurred during the processed of extraction and transforming the dataset was the creation of too many tables. The team initially had created additional tables where aggregate functions, calculations or merge of similar datasets occurred causing either repeated data or adding tables that could be implemented easily using SQL statements. The team decided it would be simpler to separate the table and if there was a need to have a merge dataset it would be easy to write a merge statement on the SQL query. This helped to reduce the number of tables to only the essential dataset tables and simplified the database.

Another problem that occurs was our sales and prices file were too big for github and SQL. In order to resolve this we have to implement a git ignore of those file.