

## **Data Wrangling:**

Working with a Kaggle Dataset, most of the data was already cleanly organized. This includes data being organized into 3 categorical datasets: items.csv, item\_categories.csv, and shops.csv. The training and testing data were also cleanly provided in sales\_train\_v2.csv and test.csv. After all the datasets were properly imported into Jupyter Notebook, the first step was to use the .head() and .info() parameters of dataframes to see how the data was organized.

## **Cleaning the Dataset and Dealing with Missing Values:**

Immediately I realized that the date column of the training data had a date column, but the dtype was made up of strings. In addition, the string format of the date (DD.MM.YYYY) did not allow me to use parse parameter. As a result, I first defined a function that restructured the string format (YYYY.MM.DD) to an easier to understand and organize format. With a properly restructured string format, Pandas was able to easily parse the date column and convert all entries into datetime objects. I also did a quick search for null values and found none so no steps were necessary to address null values.

## **Locating and Dealing with Potential Outliers:**

Following that, I used the describe parameter to do a quick search for outliers in the dataset. Immediately, I noticed potential outliers in the item\_price and item\_cnt\_day columns. In the item\_price column, there was a single negative value (-1.0) and a single extreme value (307980.0) that was multitudes higher than the second largest. Since there was only 1 entry of each, I felt it was safe to call these outliers and remove them from the analysis. In addition, it didn't make logical sense to sell items at a negative price or at that high of a price. For the item\_cnt\_day column, there were more than 7300 entries that were negative. It was highly likely that these are due to returns of an item, but since no information was provided or any additional information could be gathered, I chose to make two datasets, 1 removing the negative entries, and 1 keeping them. I will apply my model to both datasets to check for performance.