Data Wrangling Strategy

Data wrangling is the terminology used to describe the process of transforming raw data to a clean and organized format ready for use. The most common data structure used to "wrangle" data is the data frame. For this particular project, we imported the data from CSV file by using Pandas. We will take the following steps to perform data wrangling.

1. Describing the data

The first step is to view some characteristics of a DataFrame. One of the easiest things we can do after loading the data is to view the first few rows using head(). We can also take a look at the number of rows and column using .shape attribute of the DataFrame. By using .info attribute or .dtypes attribute, we are able to view data type of each variables.

For any numerical variables, we can use .describe() method to get descriptive statistics. Pandas offers variance (var), standard deviation (std), kurtosis (kurt), skewness (skew), standard error of the mean (sem), mode (mode), median (median), and a number of others. Pandas also provides a large set of summary functions that operate on different kinds of pandas objects (DataFrame columns, Series, GroupBy) and produce single values for each of the groups. When applied to a DataFrame, the result is returned as a pandas Series for each column. For any categorical variables, both .unique and .value_counts are useful for manipulating and exploring the data.

2. Dropping duplicate rows

The method .drop_duplicate can help perform this task.

3. We have determined the date type of each variables in Step 1.

For numerical features, the scaling matters for some machine learning algorithms. By checking the descriptive statistics, we can decide whether the data standardization is needed.

To detect missing data, the functions isnull and notnull return booleans indicating whether a value is missing. If missing data exist, we need to figure out its size and randomness. If the size is minimal and not highly dependent on the target variable, we can probably remove them. Otherwise, we should apply different imputation

strategy such as mean or median. To detect outliers, a box-and-whisker or histogram plot helps in this regard. Numpy .percentile() command is also an effective tool to identify the extreme observations and define the upper and lower boundaries. The interquartile range (IQR) method can be used to fix outliers.

Once we have detected the missing values for categorical features, we can apply frequent category imputation strategy, replacing all occurrences of missing values with the most frequent value. Alternatively, we can treat missing data as an additional label or category of the variable so that the importance of 'missingness' can be captured. By using .value_counts() and unique() methods, we can find out the cardinality of a categorical variable and rare labels if possible. Both high cardinality and rare labels may cause overfitting. Removing them can help improve machine learning performance.