Project – Wrangle and Analyze Data.
As the name suggests this project is about wrangling and analyzing data. The dataset that will be worked on; wrangled is that of tweet archive of user @dog_rates.

Typical steps in Data Wrangling process are Gather – Assess – Clean. In this project we go through each step starting with gathering data.

The data that was wrangled and analyzed was that of tweet archive of twitter user @dog_rates (WeRateDogs.) . There were multiple Pieces of data involved in this project:

1) First piece is sent exclusively via en email to Udacity by WeRateDogs for the purpose of this project.
2) Second piece builds on first piece with added information not available in first piece with columns such as retweet_count, favorite_count. Etc.
3) Third piece is an image prediction file which is an output of a neural network whose output contains image predictions regarding breed of the dog.

Varied methods were employed to get each piece of data. Out of the above stated, the first and third piece was supplied by Udacity whereas second piece was downloaded by Twitter API.

When it came to getting the data which was sitting outside programming environment, pandas package was instrumental. Each of the 3 pieces of data was read in a dataframe.

Twitter-archive-enhanced.csv was in .csv format so it was imported using pd.read_csv(file_name) function.

Tweet-json.txt was not in csv format, it was instead in json (Java Script Object Notation) format. Therefore I first read single line from the text file and then used loads function from json module on that line. This converted the data in the line to dictionaries, out of which I appended relevant values to a dataframe ultimately constituting as a single row in the dataframe. This process was repeated in a for loop reading each line in the tweet-json.txt.

Image_predictions.tsv file was located on a remote server, I downloaded it programmatically using requests library and imported it in Jupyter notebook with pd.read_csv by specifying separator as '\t' thereby reading tab separated variable.

This way I had 3 dataframes which contained data originally in different formats and yet now was in a uniform structure of dataframe.

Now let's move on to Assessing data which is the second step in wrangling process. When looking at quality of the data, a guiding light is Data quality dimensions. There are 4 data quality dimensions; Completeness, Validity, Accuracy, and Consistency. They act as beacons in the quest to identify data quality issues. Let's look at how I uncovered each dirty data issue.

Data quality issues were identified by combination of 2 approaches i.e. visual assessment & programmatic assessment. Out of these approaches, visual assessment is a simple one. Though simple, it consists of several tools/functions. In my case I used following functions to perform quick assessment:

1) Df_name.sample(n) = this function gives n random rows at a time.

2) Df_name.info() = this gave me name, datatype, non null count of each column.
3) Df_name.describe() = this gave me bunch of statistical descriptors. This also gave me sense of outliers.
4) Df_name.corr() = this picked all numeric columns and gave me correlation between each pair. This function is useful as it gives sense of what columns could turn out to be useful for deeper analysis.
5) Sns.distplot(df_name) = this function provides a histogram which gives a sense of distribution, it also can be used to identify potential outliers.

Using visual assessment I discovered some issues and also shortlisted some columns to perform deeper analysis using programmatic assessment. Following are functions I used on shortlisted columns to find more data issues which were in accordance with project motivation.

1) Df_name.column_name.value_counts() = I used this so as to understand outliers, data corruption, data entry error etc.
2) Df_name.info() = this gave me name, datatype, non null count of each column.
3) Df_name.describe() = this gave me bunch of statistical descriptors. This also gave me sense of outliers.
4) Df_name.sample(7)[[column_1, column_2]] = This is a very useful function as it allows me to see sample of specified number of rows. Every run gave me new sample so as to not drive my analysis in only one direction and thereby allowing me to see more possibilities with the data.

Messy data issues were discovered using project motivation and objectives as main input. In accordance with that I recognized, noted that individual pieces should be merged in order to create a master dataframe. In that pursuit few columns were identified which represented data duplication, which was also noted in the list of data issues.

This leads to last step in the data wrangling process; cleaning the data.

As is the custom, I made copy of each dataframe before performing any data cleaning tasks so as to prepare for a contingency in case of mistakes as to avoid loss of data.

Protocol/steps involved in data cleaning are Define, Code, Test.

Following are some important functions used in the cleaning process

1) Df_name.column_name.replace()
2) pd.to_datetime(df_name)
3) df_name.drop()

Once data issues were cleaned, only thing remaining was merging individual cleaned pieces in a master dataframe. For this Pd.merge() was used. Once created, master dataframe was written to the disk storage using df_name.to_csv('file_name.csv', index=False) signifying end of Data Wrangling stage.

--Saurabh Daphtardar