Data Wrangling
Capstone Project # 1 - Predicting Popularity of YouTube Videos

Data Set

The data set used for this project comes from Kaggle. There are a number of csv and json files available on Kaggle under the name “Trending YouTube Videos Statistics”. For this project, csv files are being utilized. The dataset includes data on trending videos. Data is included for several countries including the USA and Mexico. Data includes 16 columns (fields) in its raw form for both the USA and Mexico videos. There are a total of 40,949 records found in the USA csv file and 40,451 records found in Mexico csv files. The data for the most part is clean.

Data Wrangling Code

https://github.com/fariha23/YouTube_Data_Analysis_Video_Categories

Data Wrangling

Steps:

1. **Removing unnecessary columns**
   After loading files in Pandas as dataframes (namely, usa_df and mx_df) and looking at the output of “.info()” method on both dataframes (df) it is observed that no null records in either of the dfs. The attribute “.columns” shows that following are the columns included in the data.

   'video_id', 'trending_date', 'title', 'channel_title', 'category_id', 'publish_time', 'tags', 'views', 'likes', 'dislikes', 'comment_count', 'thumbnail_link', 'comments_disabled', 'ratings_disabled', video_error_or_removed', 'description'

   The aim of the project is to predict which genre (category) of videos is most popular in different countries. In order to make only relevant fields available for analysis, following fields(columns) were dropped from original dfs. To drop these columns “.drop()” method was used.

   ‘'video_id','thumbnail_link','description','tags','channel_title'

2. **Adding “category_name” column**

One important field missing from both csv files is category name (‘category_name’). The reason is that every country’s data doesn’t have the same 1:1 translation between category_id and category_name.

3. **Dropping rows with NaN values**
   After the new column was added to both dataframes, the USA data frame shows that there are no null columns. The Mexico data frame shows 252 rows of NaN. Upon investigating, it was found that the Mexico data frame has a category id of 29 whereas the translation dictionary does not contain that specific number. Therefore a NaN was written in those rows under column “category_name” of the data frame. After reviewing several of these 252 videos it is determined that these videos do not belong to one specific category. Therefore these rows will be dropped from the Mexico data frame. Since there are 40K+ rows of data in the data frame, the impact of dropping 252 rows will not be significant when the prediction model is built for the project. To drop these rows, “.drop()” method is used with a boolean condition for the specific null values in column “category_name”. To find nulls, “.isnull ()” method is used.

4. **Outliers**
   Outliers are extreme values in a data set that makes the data pattern skew. These values are sometimes warranted . For example highly popular YouTube videos can acquire a couple of millions of views and for the purposes of this project those outliers can help determine the most popular category of videos. To find the outliers, boxplot was graphed using Seaborn Library,

   ```
   sns.boxplot(x=usa_df_clean['views'])
   ```

   Summary statistics were also collected on both cleaned data frames using .describe() methods. Both results can be found in the jupyter notebook saved under the github url provided above.

5. **Appending Mexico Data Frame with USA Data Frame**
   In order to keep things simple, appended Mexico’s data frame under the USA's data frame. A column ‘country’ was added to both data frames(‘USA’ and ‘Mexico’). Reset the index afterwards.
   ```
   combined_usa_mx_df=usa_df_clean.append(mx_df_clean)
   combined_usa_mx_df.reset_index(inplace = True, drop = True)
   ```
6. **Dropping duplicates**
   Drop duplicates in the data frame combined_usa_mx_df and reset index
   ```
   combined_usa_mx_df.drop_duplicates(inplace=True)
   combined_usa_mx_df.reset_index(inplace = True, drop = True)
   ```

7. **DateTime columns: publish_time and trending_date**

Columns publish_time and trending_date are changed into DateTime data type using pd.to_datetime. Changing Trending_Date was not straightforward. A function was written to change format YYDDMM into ISO80010.

8. **Dropping rows where publish_date is greater than trending_date**
   It can be noticed that there is some data in the dataframe where the publish_time of the video is after the trending time. For example the official release date of "Childish Gambino - This is America" was May 5, 2008 (https://en.wikipedia.org/wiki/This_Is_America_(song)) so the trending_date of this video cannot be before May 5, 2008. Therefore all the rows where trending_date was less than publish_time were dropped.