**Amazon Product Review Analysis**

**Objective:**

This project covers your skills in using various aspects of data analysis tools effectively.

Your solution should include a good analysis of the data and make use of the best approach for forecasting.

A good solution has a sound application of programming and algorithmic knowledge that matches the given problem statement.

Put together your modular coding and analytical skills in use for this project.

**Amazon Product Review Analysis :**

The year was 1994 when Bezos launched Amazon out of his garage. In 1995, the first product was launched by Amazon. It was a book that was sold to 50 states in 45 different countries within 30 days (Oberlo 2021).

Within 26 years, Amazon holds the title of the world’s largest online retailer and has become a household name. Amazon has become synonymous with online shopping and continues to grow by developing new products, acquisitions, and different service offerings to enlarge the customer base.

Nowadays, people (almost 150.6 million) turn on the Amazon app for everything. Several types of research have proved that customers trust Amazon (Statista 2019). On average, the small and medium-sized businesses located in the USA sell more than 4,000 items per minute (Amazon 2019), which leads to millions of product reviews on Amazon.

Reviews tell what products and features are trending, what is in demand, what is no longer relevant, how products and those of competitors are doing, and much more.

It is observed that the maximum number of customers look at product reviews before they make a purchase. Survey results show that positive product reviews are a key factor for purchasing by 57 percent of Amazon buyers (Statista, 2019).

As product reviews are often the deciding factor for many customers, it’s very important to have a well-automated system for monitoring them.

The traditional manual process of Amazon product reviews is time-consuming and inefficient when millions of reviews are being posted all the time. It doesn’t show any trend or patterns over time. Moreover, it is tough to understand customers’ sentiment towards any product or its delivery.

Review analysis must dynamically adjust to the changing trend.

**Case Study: Amazon Product Review Analysis :**

Thomas, a global market analyst, wishes to develop an automated system to analyze and monitor an enormous number of reviews. By monitoring the entire review history of products, he wishes to analyze tone, language, keywords, and trends over time to provide valuable insights that increase the success rate of existing and new products and marketing campaigns.

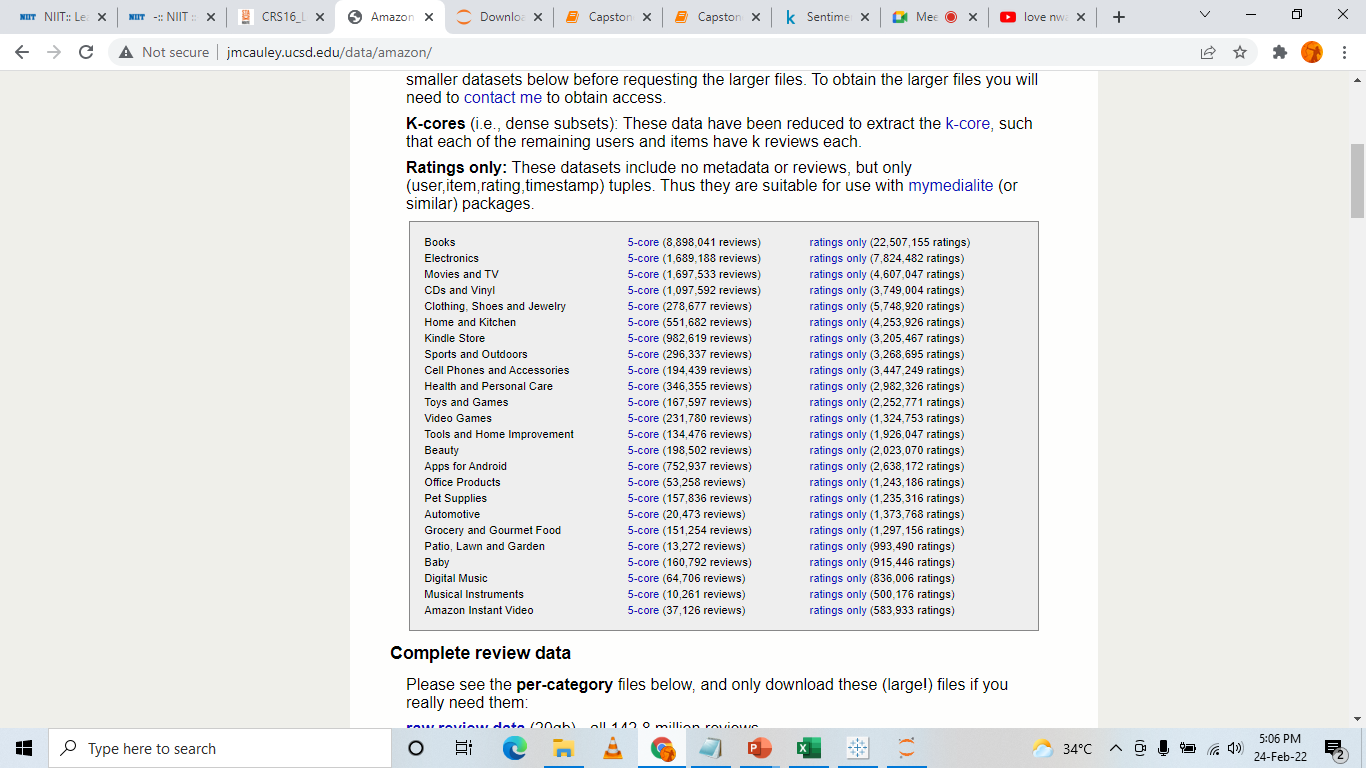
**Scenario 1: Inventory Optimization and Demand Forecasting**

Optimize inventory management by identifying the product categories (Clustering as an outcome of text processing) on the customer review data. Predict what kind of products could be in demand (Time Series Analysis).

**Performing Text and Data Preprocessing**

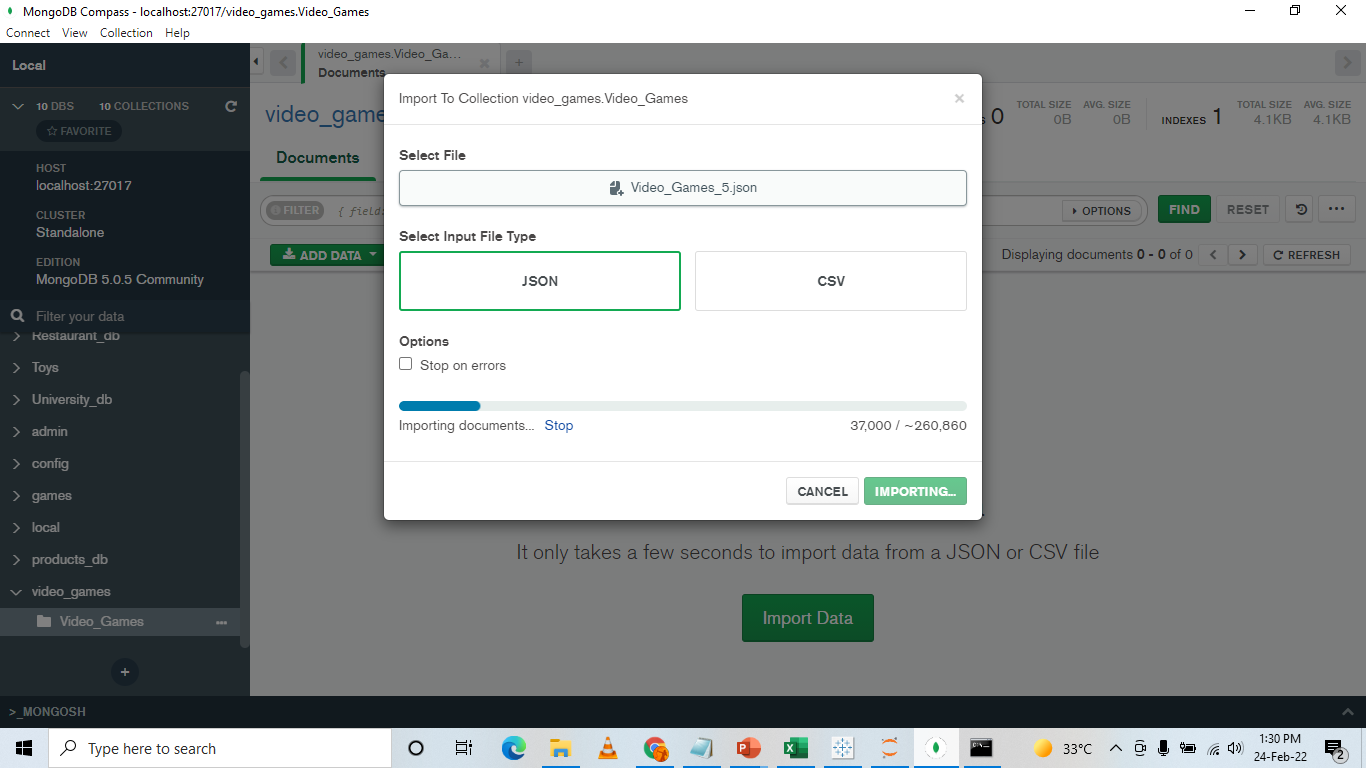
**Data Source:**

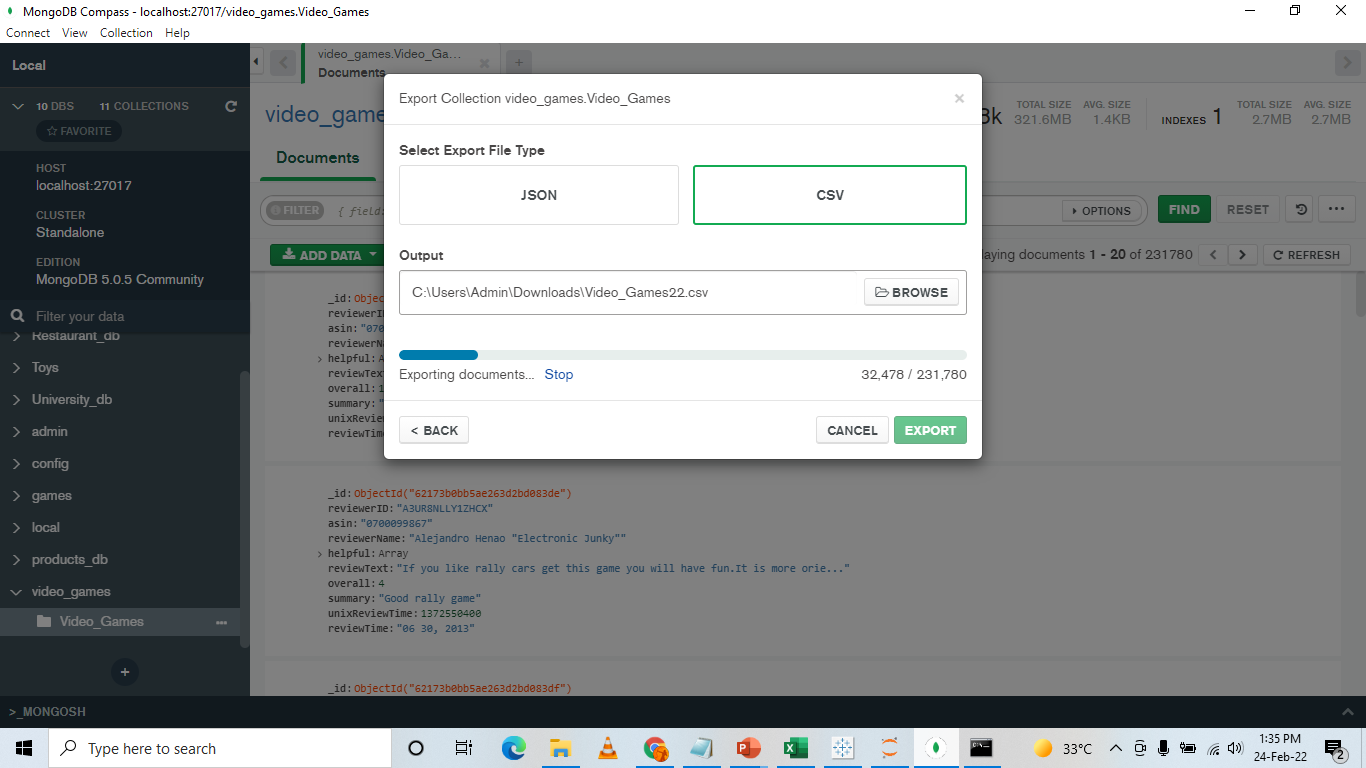
We are downloading the datasets from http://jmcauley.ucsd.edu/data/amazon/ and I am using video\_games and toys\_and\_games pair for this project.



**Converting the JSON format file to CSV file:**

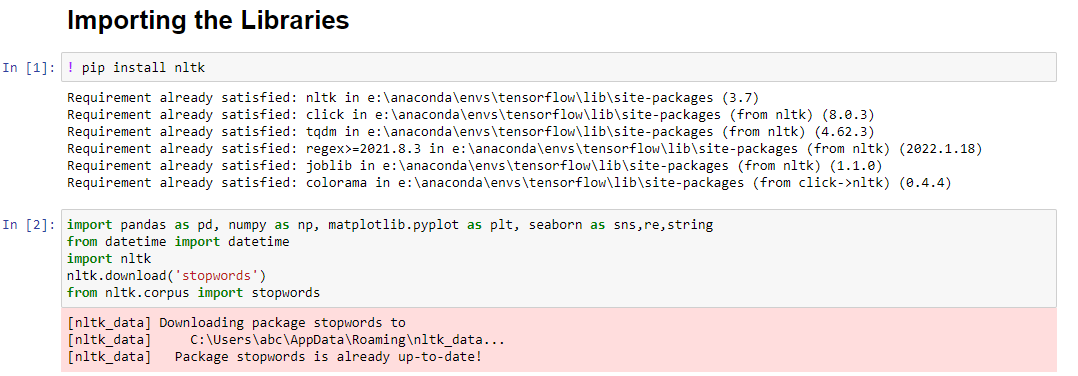
When we are extracting the files, we are getting it in json format. We are using MongoDB to convert Json file to a CSV file, and storing it in a folder in C-Drive.





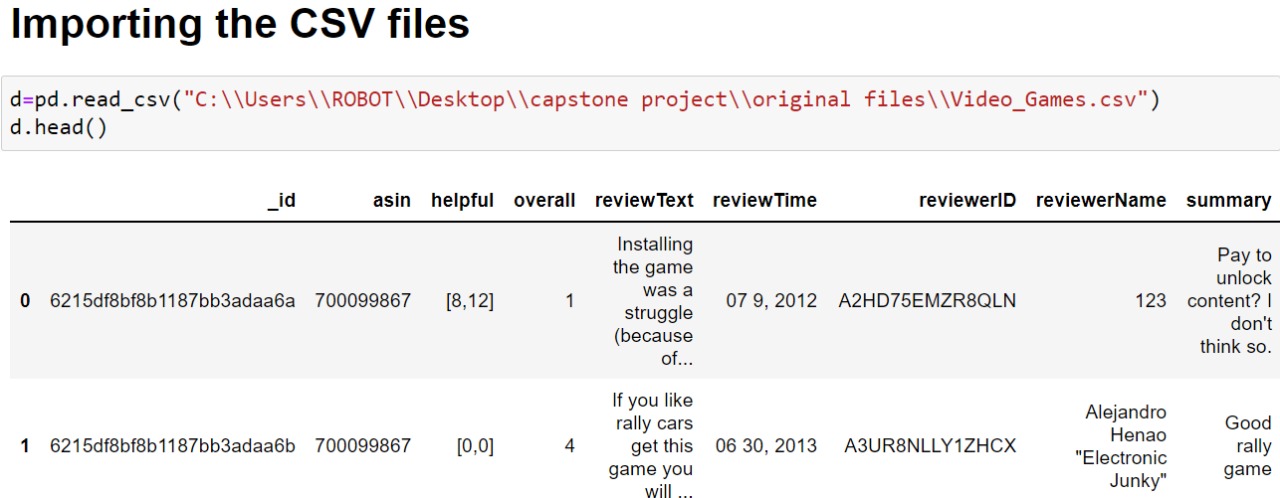
**Importing the necessary libraries:**

I am importing pandas, regex, NumPy, matplotlib, nltk, stopwords, seaborn and datetime functions. These were the only functions needed for the data cleaning part.



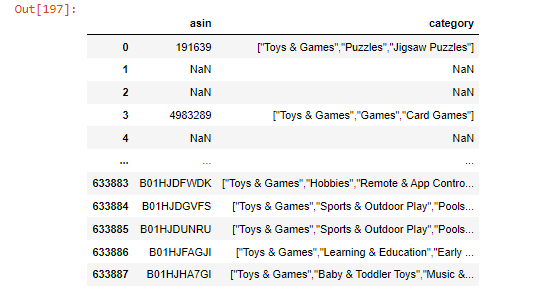
**Importing the CSV files:**

We are importing both the files from the folder into the python codebook.









**Merging the two datasets:**

We merged core data and metadata using inner join with asin as the common field because we need both main \_category and reviews in one dataset for our analysis.

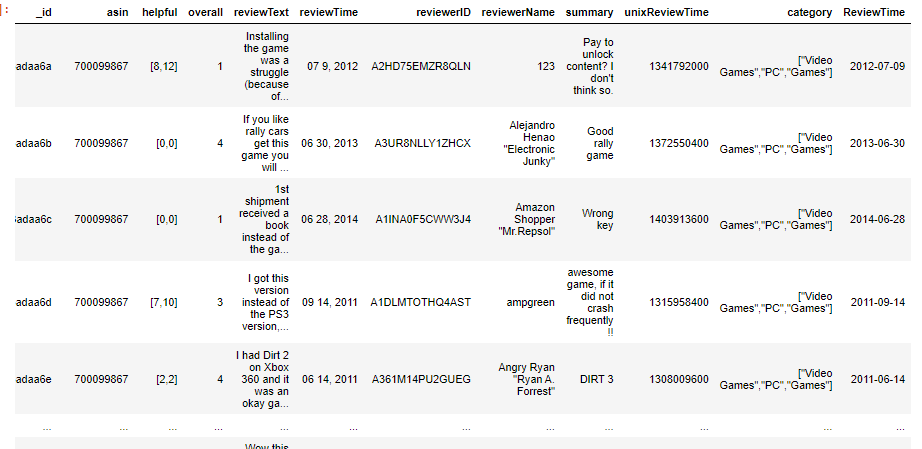
****

****

**Converting Unix Review time to date time format and storing it in a new column:**

I am making a copy of the dataframe, where we imported the data. In the dataset there are 2 columns, one being **unixReviewTime** and the other being **reviewTime**. We are using the **unixReviewTime** and changing it into datetime format and storing it in a new column named ReviewTime. To do this we are defining a function which we are using to do the same process in both the dataframes.



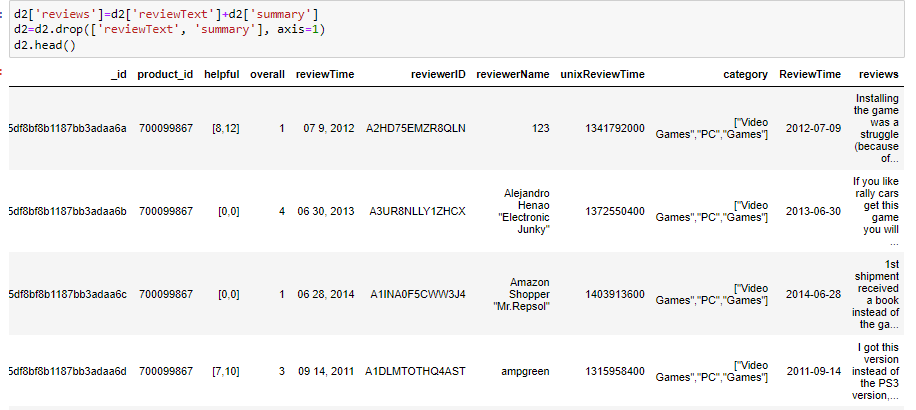


**Extra Step:**

We are changing the name of the of the column asin to product\_id, so that its easier for us to identify the column later.

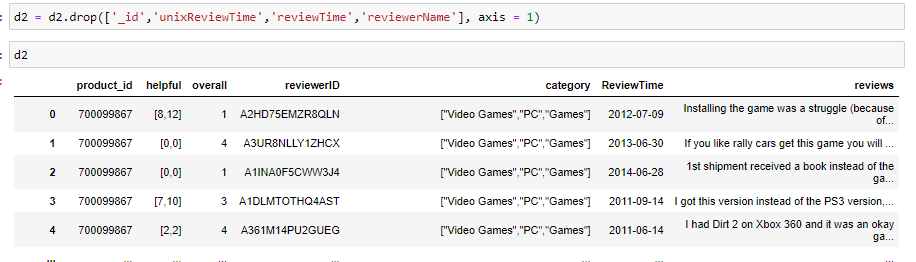
**Joining Summary and Review text together to form reviews column and then dropping those columns:**

We are making a new column called Reviews, by combining **reviewText and** **summary** as both the columns contain the review of the customer, and its better to use a single column as it will help us in the analysis. Then we are dropping those columns as they are no longer required.



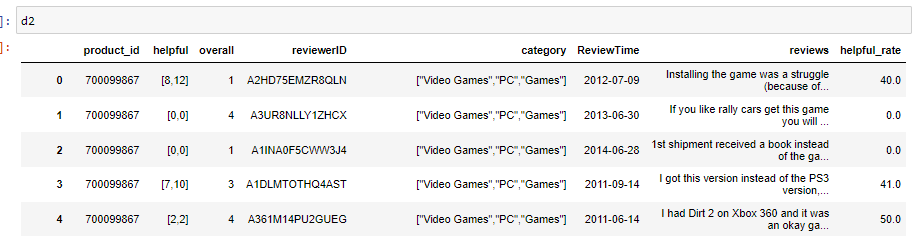
**Dropping columns not required for further analysis:**

We are dropping the columns (\_id, unixReviewTime, reviewTime and reviewerName) as it is not required for further analysis. The unixReviewTime and reviewTime are not required as we have made a separate ReviewTime column in date time format, and reviewerName is not important, as we already have the reviewer ID.



**Making a new column called helpful\_rate to see what percent of the reviews that were written by the user were helpful to other people:**

We have a column called helpful where the data is given in the form of a object, so we are splitting the data into 2 columns, and removing the brackets from it, and then concatenating the columns and then defining a function to remove any whitespaces from it. Then we are changing it into intiger format and then we are dividing the values and multiplying it with hundred to get the percentage of helpful reviews by the user. For rows with NaN value we are replacing it with 0 and for the other cells we are rounding the values to 2 decimal places. Then we are making a new column called helpful\_rate and storing the percentages in that column.

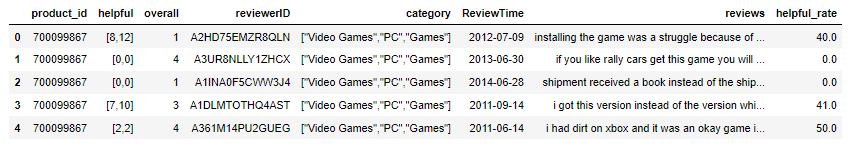


**Missing Value Treatment:**

We are searching for null values in the dataset and then removing all the rows with null values, but before this we are storing the data in a new dataframe.

**Cleaning the data, by removing any special characters, URLs, white spaces, punctuations and making uppercase letters to lower case:**

We are defining a function and in there we are removing special characters, URLs, white spaces, punctuations and making uppercase letters to lowercase letters, as this format will be needed for further analysis when we are doing sentiment analysis



**Removing the stopwords so that it can be used for sentiment analysis:**

We are using a lambda function to remove all the stopwords from the column.



**Removing tags from the datasets:**

In the datasets in the category column there are characters like ‘<span>’ which is not required, so we are defining a function to remove them.



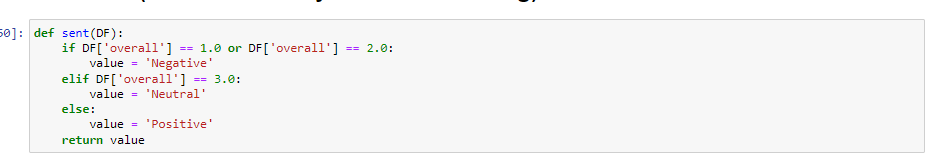
After this we are exporting the data in the form of CSV, which will be used for further analysis. We are the same process for both the datasets.



**Sentiment Analysis**

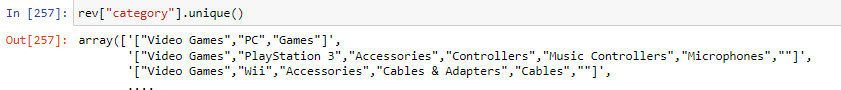
**Defining a function to separate overall ratings into positive, negative and neutral:**

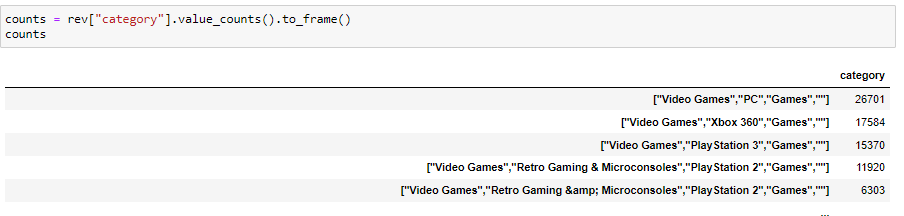
We defined a function where we are splitting the overall ratings into positive and negative. Then we are making a column called Sentiment, where we are getting all the ratings which are either positive or negative.



**Grouping all the unique categories:**

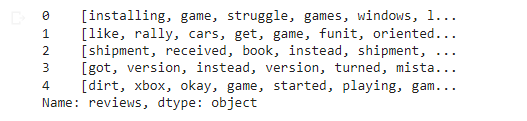
Making a new dataframe called rev and storing a copy of the old dataframe in it. Then checking for all the unique categories and finding the count of the number of reviews in each category and then arranging it in descending order and storing it in a dataframe called counts.

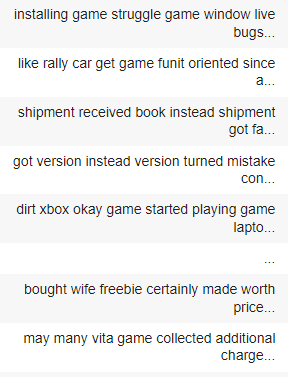


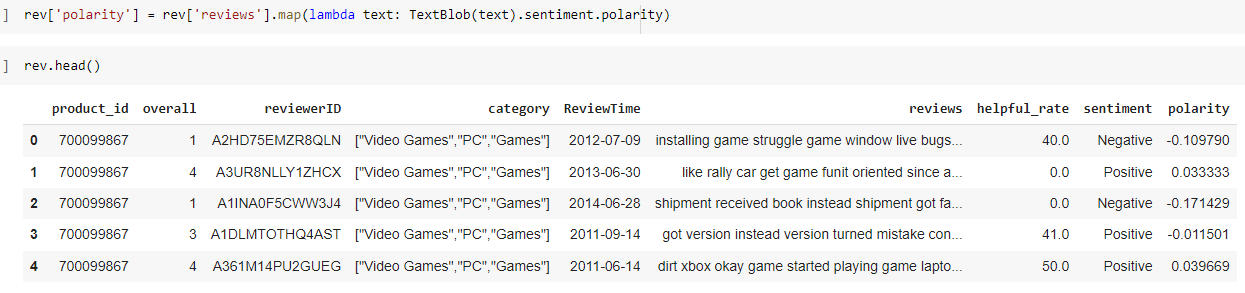


**Tokenising and Lemmatising the review column to find polarity:**

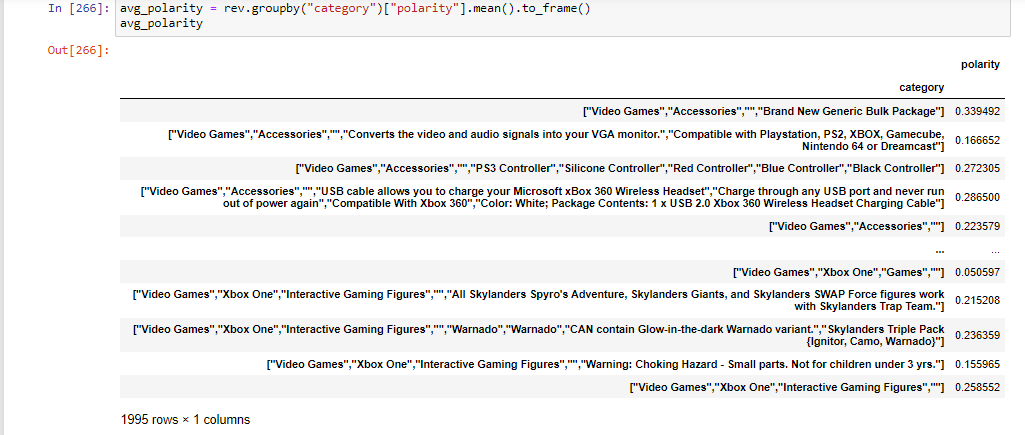
Tokenising the review text, and then lemmatising it. From there we are finding the polarity of the texts from the reviews, with positive score means positive sentiment and negative score being negative sentiment.

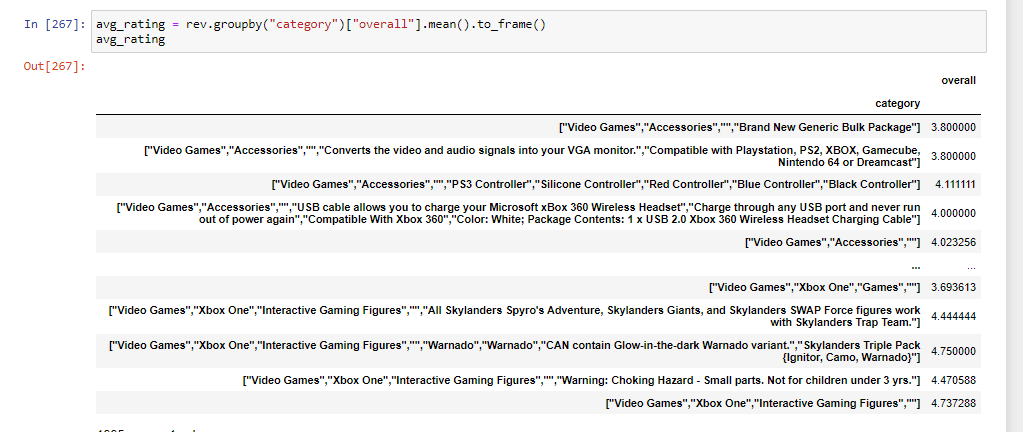




 **Finding the mean polarity and mean ratings for each category group:**

Finding the mean polarity and mean ratings for each category and checking if the overall mean polarity for a category is positive or negative. Then making a new dataframe called fin\_rev and joining the mean polarity and mean ratings with counts dataframe.







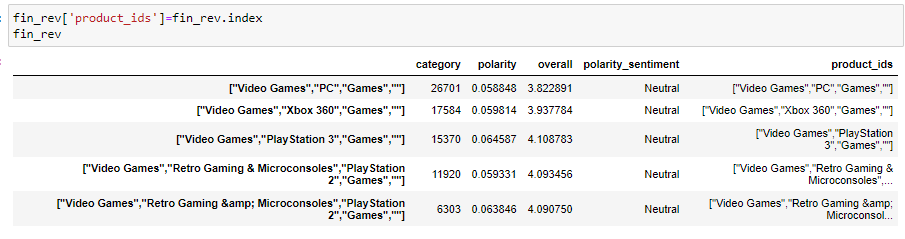
**Defining a function to split the polarity into positive, negative and neutral sentiments:**

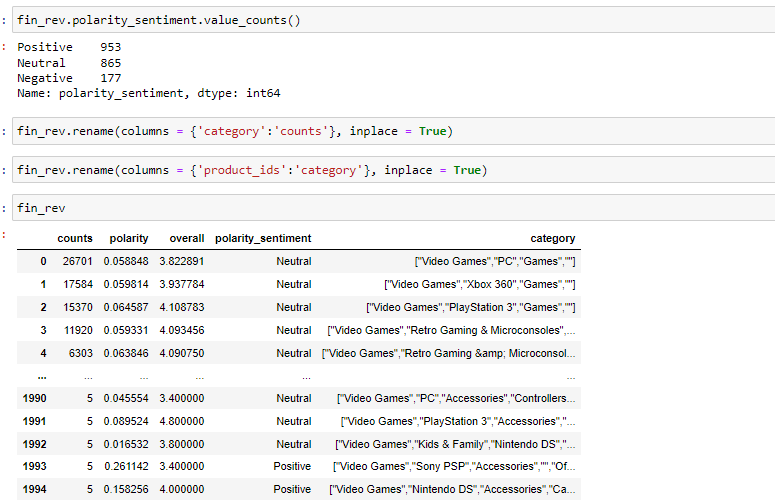
Defining a function called sent1 which is splitting polarities into positive and negative and then adding a new column called polarity\_sentiment where we are applying the function.



**Making a new column called product\_ids where we are storing the values of the index in the column :**

Making a new column called product\_ids where we are storing the values of the index in the column, as we need it for label encoding and then clustering of the data. Then renaming the category column to counts and product\_ids column to categories and then getting our final fin\_rev dataframe.

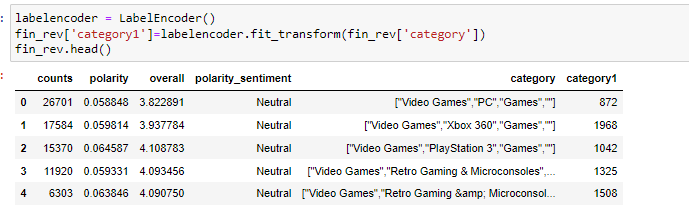




**Clustering**

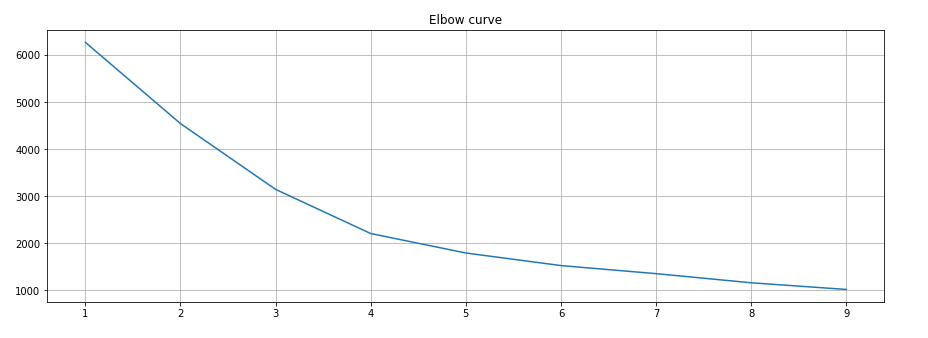
**Label encoding the categories column so that it can be used in clustering of the data :**

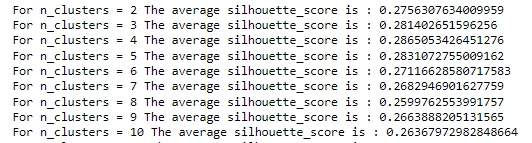
Label encoding the categories column and making a column called category1, where the label encoded values of category are stored.



**Making the elbow curve and finding the silhouette scores :**

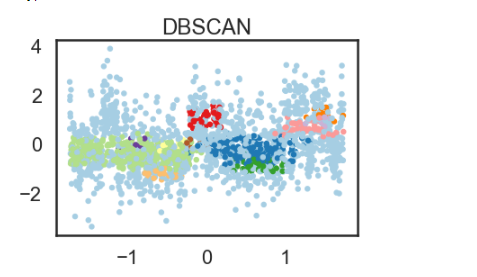
Making the elbow curve and finding the silhouette scores, which will be used to check which clustering method would be better.

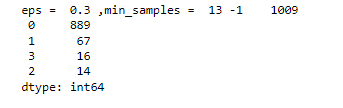


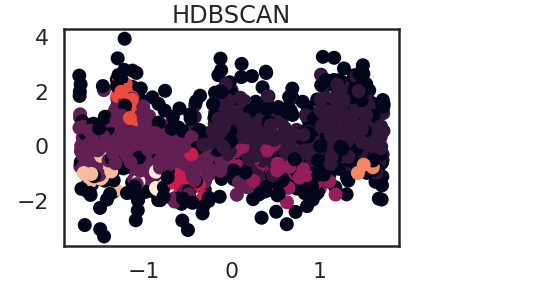


**Doing DBSCAN and HDBSCAN clustering :**

Performing DBSCAN and HDBSCAN clustering on the data to find out which clustering technique is the best for this dataset.









**Comparing the silhouette scores:**

**DBSCAN**



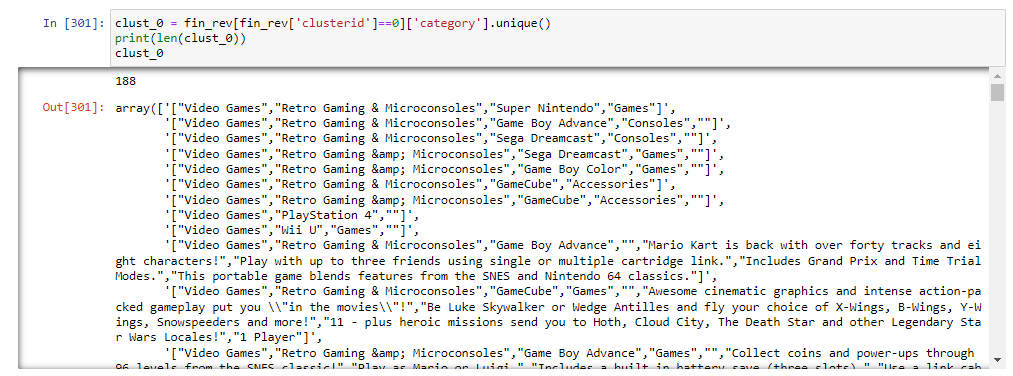
**HDBSCAN**

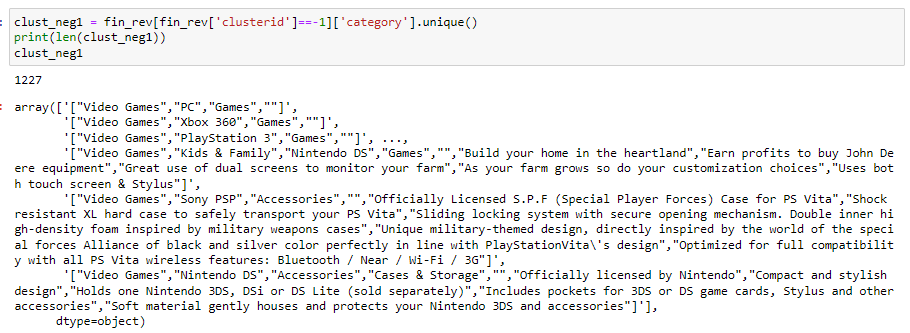


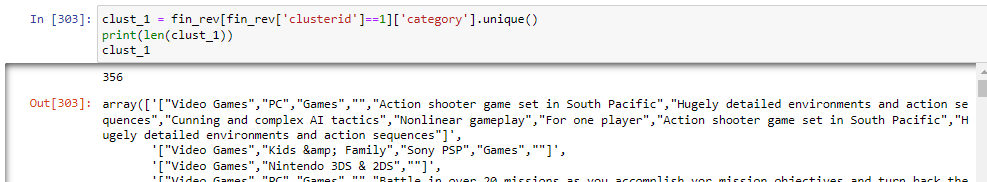
As the silhouette score for DBSCAN is more than HDBSCAN, so we are going to use DBSCAN for further analysis.

**Using a filter to create 3 dataframes based on their cluster\_ids, 0,-1 and 1:**

Used a filter to store all the categories belonging to a particular cluster\_id together. Did this process individually for all the 3 cluster\_ids. Then sorted the different categories on the basis of mean of overall reviews for the particular cluster\_id and stored it in 3 individual dataframes. Found all the top 10 categories in each individual cluster\_ids, and also the last 10 categories.

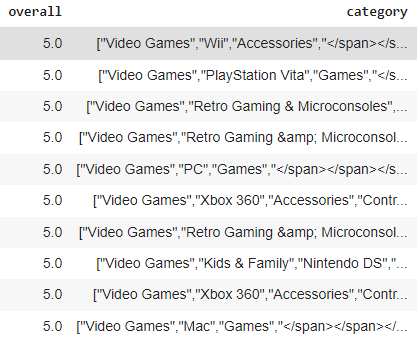






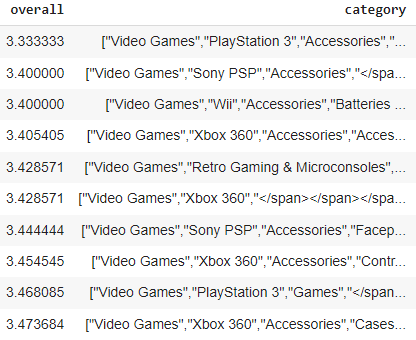
**Top 10 categories based on overall ratings:**

****

****

****

**Last 10 categories based on overall ratings:**

****

****

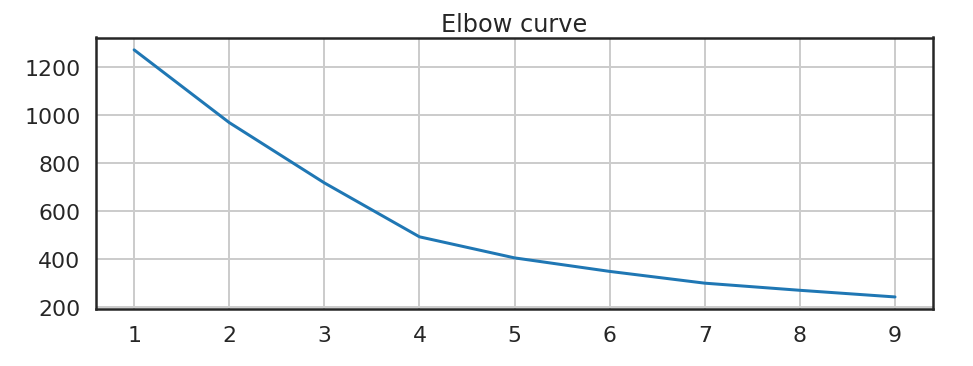
****

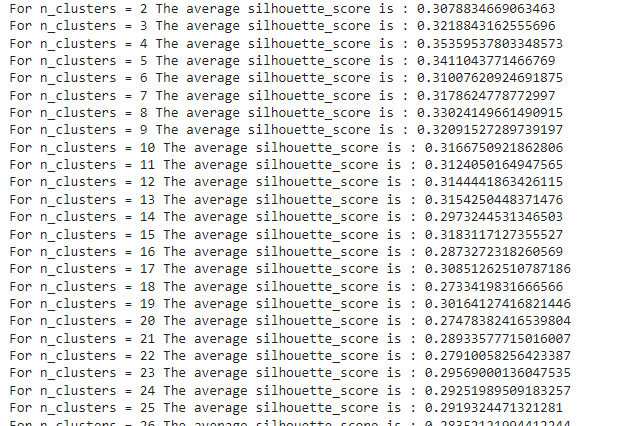
**DBSCAN will be used in this dataset because it has a better silhouette score than HDBSCAN.**

**We are repeating the same process for dataframe 2.**

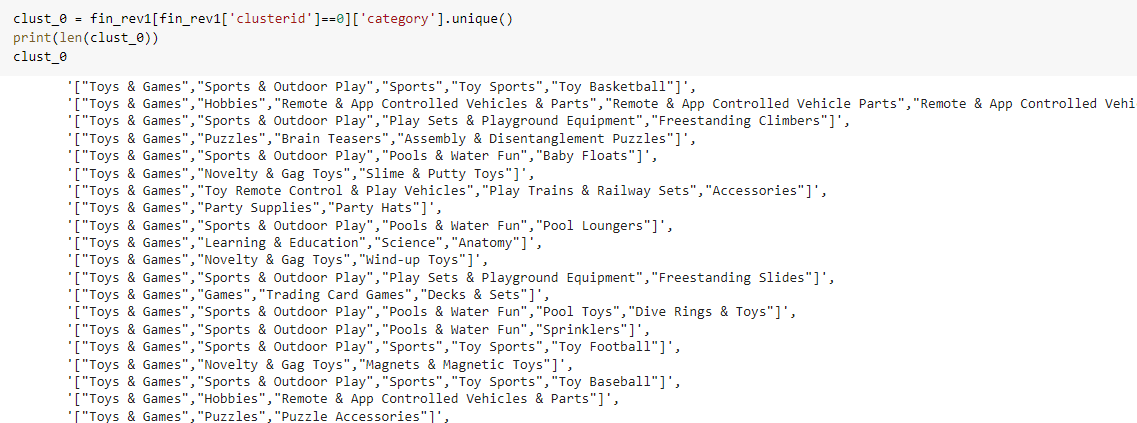
**Clustering of Dataframe 2:**

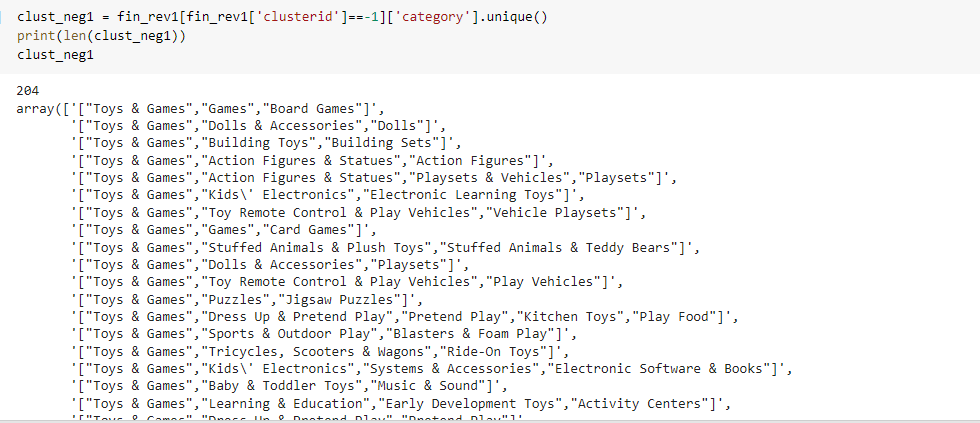
**Making the elbow curve and finding the silhouette scores :**

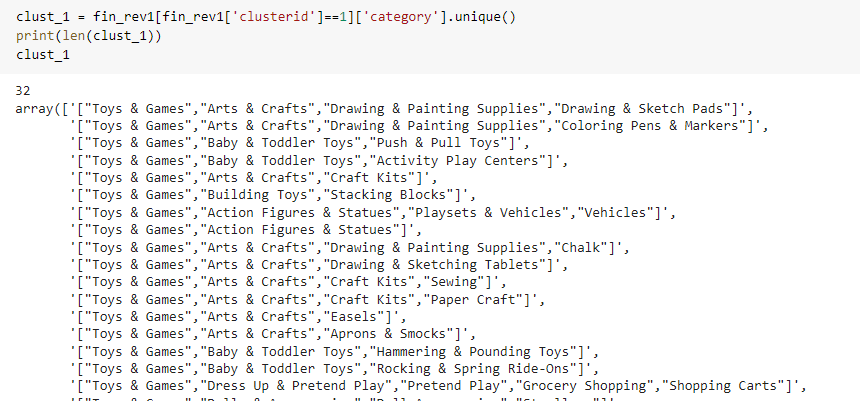




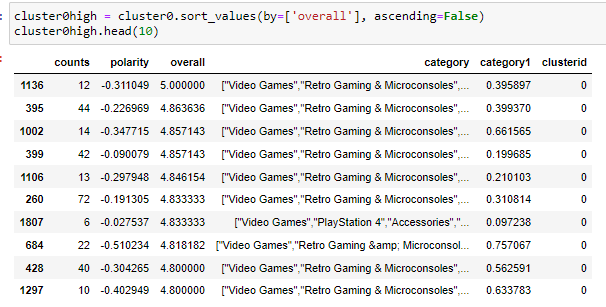
**Output:**





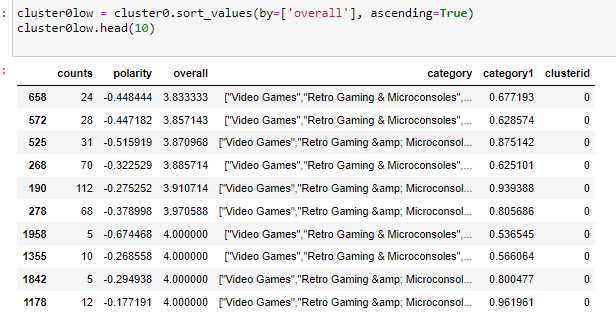


**Sorted the values in descending order with respect to rating to find all the categories with a high rating:**

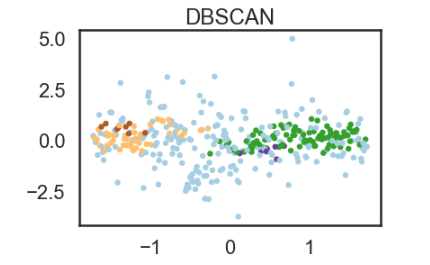
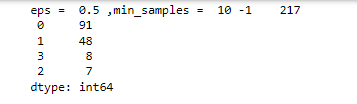


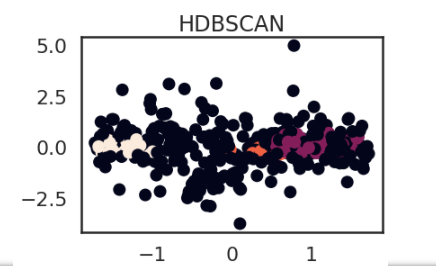
**Repeated this method for the other 2 clusters ids.**

**Sorted the values in ascending order with respect to rating to find all the categories with a low rating:**

**Repeated the same method for the other two cluster ids.**

**Doing DBSCAN and HDBSCAN clustering for the Toys and Games Dataset :**







**DBSCAN:**



**HDBSCAN:**



**DBSCAN will be used in this dataset because it has a better silhouette score than HDBSCAN.**

**Rest of the process is exactly the same as we had done in the first dataset.**

**Time Series Analysis**

**Choosing the right model for time series forecasting:**

Choosing ARMA Model as it has the lowest RMSE and MSE score.

**ARMA Model:**



**ARIMA Model:**



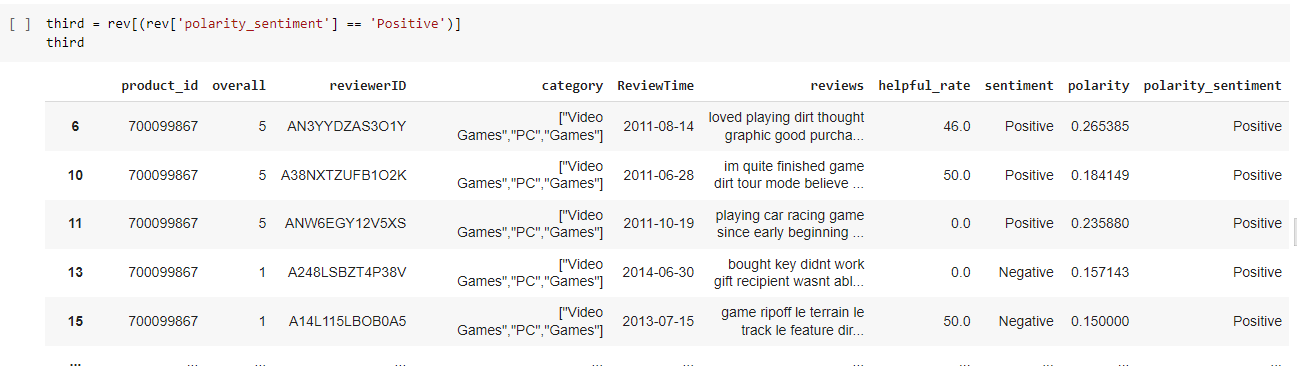
**SARIMA Model:**

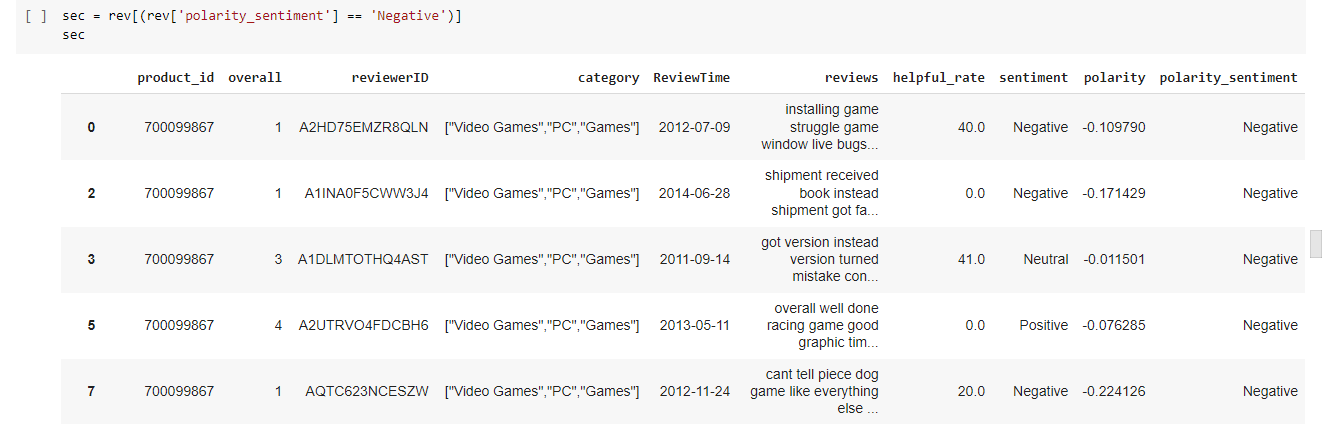


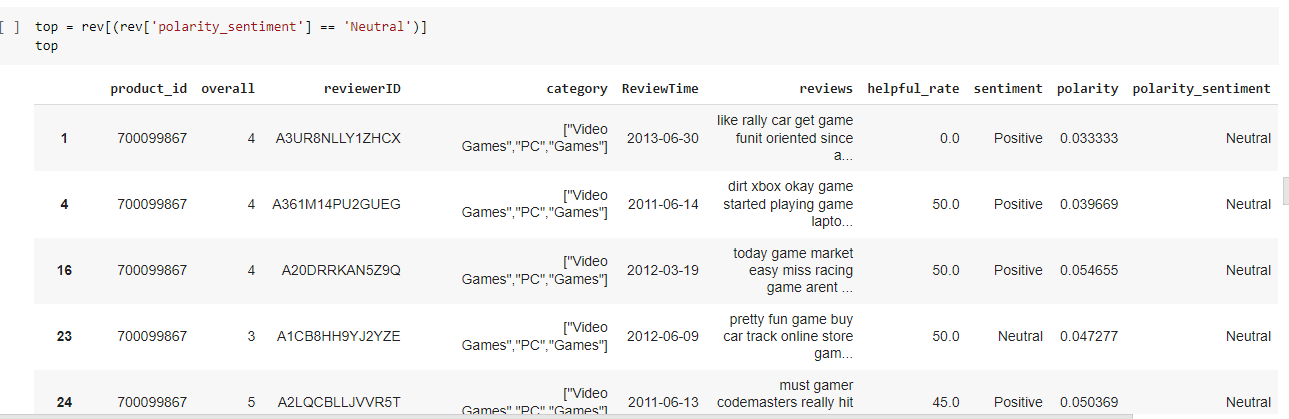
**ARMA model has the lowest MSE and RMSE score so that was used time series analysis.**

**Using a filter to get the positive, negative and neutral sentiments in 3 different dataframes:**

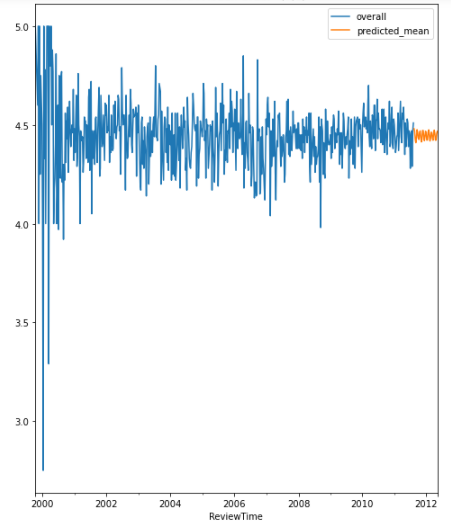
Using a filter to get positive, negative and neutral polarity\_sentiments in 3 different dataframes, to forecast which sentiment has the best increase in ratings over the next few months.

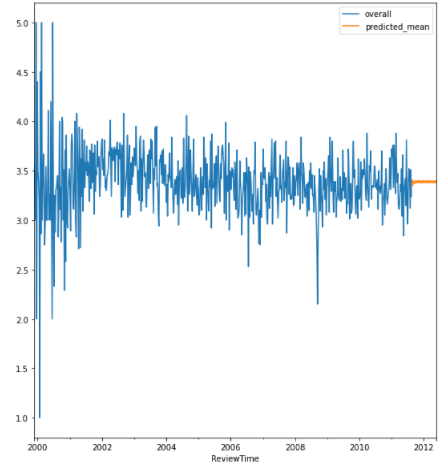


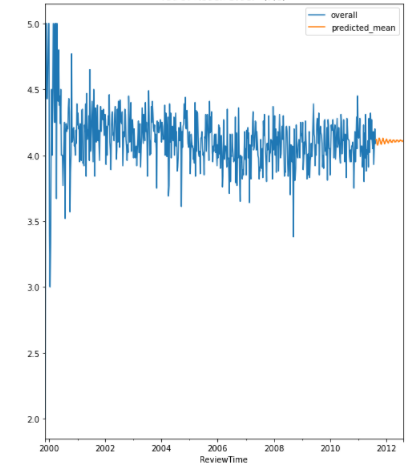




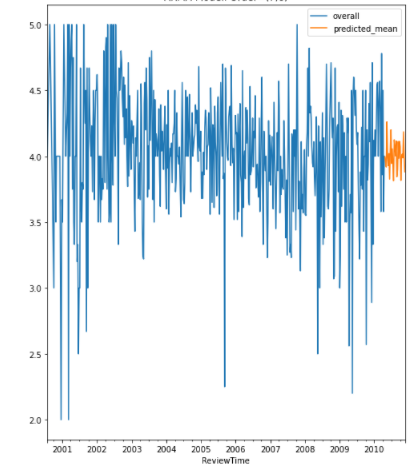
**Forecasting the results:**

****

****

****

**Finding the forecast of a particular category from Positive reviews as it had an upward trend:**

****

Repeated the same process for Toys and Games Dataset, and there instead of finding the trend for a category in categories with neutral sentiments, found it for categories with Negative sentiment as it had the upward trend there.