Huriye Güngör

78014

CSSM 502 Final Project Report

In my final project, I preferred to use Google Play Store Apps data because I am a loyal Android user and interested in how different metrics affect the reviews of a specific app. Let us start by importing necessary libraries for data preprocessing, such as Numpy, Pandas for visualization, Matplotlib, and Seaborn, and finally, I will be using Sweetviz library to receive automatic visualizations. You can find the HTML report of Sweetviz library output attached.

I will be dealing with the first 10 thousand rows of the data. After importing, let us observe the first 5 rows of it. The data consist of 13 features: App, Category, Rating, Reviews, Size of the app, Total install number, Type of app, which is free or paid, Price, Content Rating, Genres, Last updated date, the current version of the app, and minimum android version requirement.

Let us make a quick overview of the data by using the Sweetviz library. It can be seen that the App column has nearly 9 thousand distinct values, which are classified into 33 different categories. The majority of those apps are in the family category. Game, tools, and medical categories follow the family category (See Figure 1).

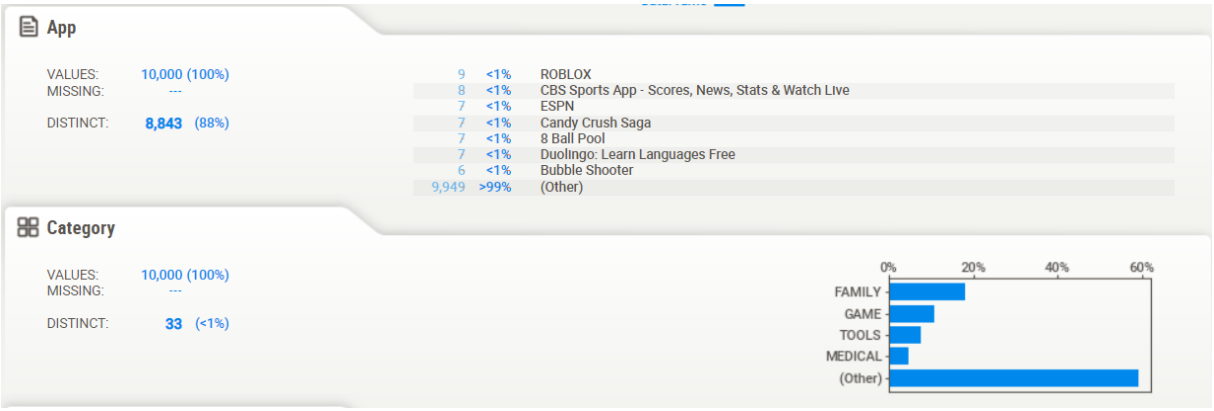

Figure 1: Sweetviz output of App and Category columns

Rating column has a left-skewed normal behavior, as can be seen from Figure 2. Moreover, average and median values are very close to each other, which also supports that this column has a normal-like distribution.

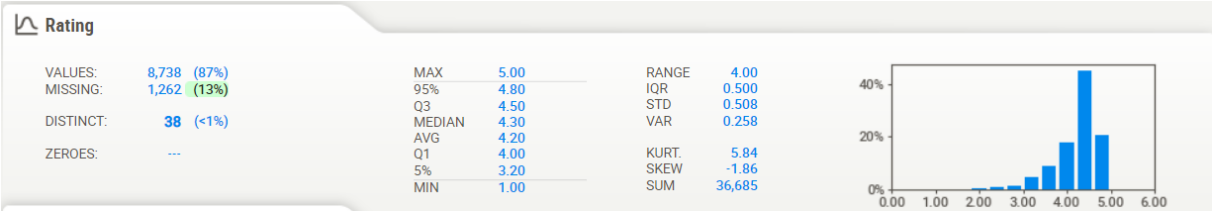

Figure 2: Sweetviz output of Rating column

95% of the Reviews column lies between zero and 1.6 million reviews which means most of the apps have no reviews at all, and this column has an outlier on its max value, as it can be seen from Figure 3.

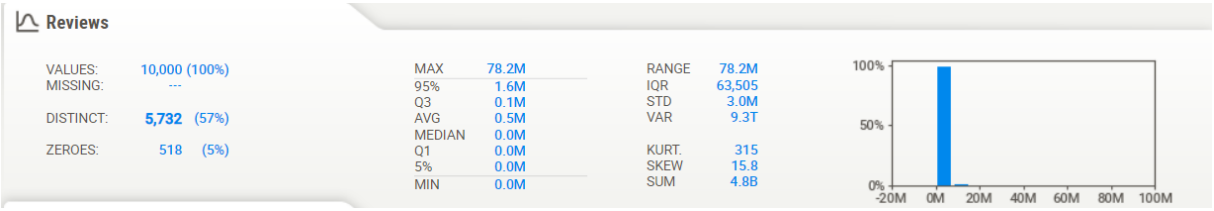

Figure 3: Sweetviz output of Reviews column

Size column gives little information about the data since most of the apps' sizes vary with device or are labeled as Other (See Figure 4).

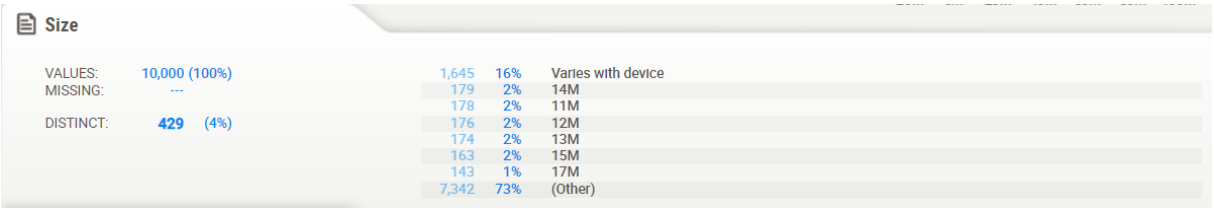

Figure 4: Sweetviz output of Size column

Currently installs column has a non-numeric data type; therefore, we cannot clearly observe its behavior of it by looking at this horizontal bar chart (Figure 5). In the next step, I will be converting it to a numeric data type.

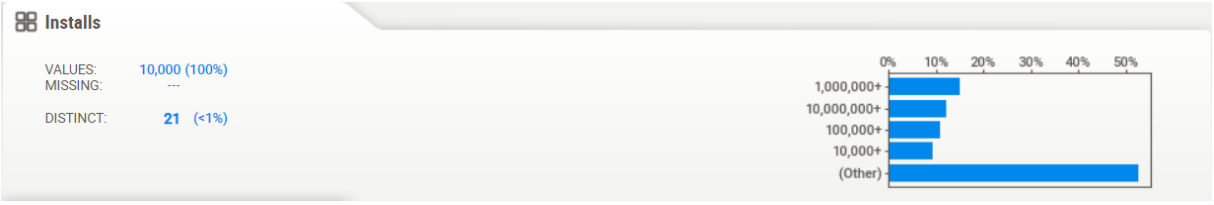

Figure 5: Sweetviz output of Installs column

From the Type graph in Figure 6, we can understand that more than %90 of the apps are free. Moreover, the price column is also supportive in this manner since most of the apps seem to have zero price.

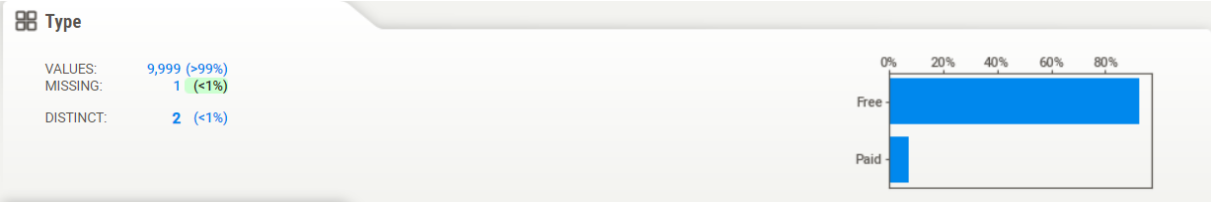

Figure 6: Sweetviz output of Type column

Content rating shows that %80 of the apps are suitable for everyone (Figure 7).

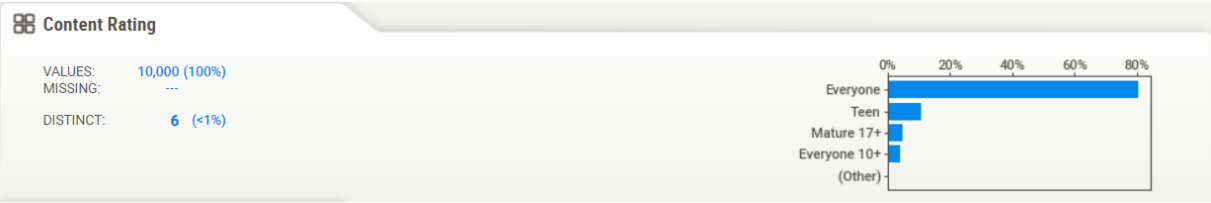

Figure 7: Sweetviz output of Content Rating column

I will not be dealing with the rest of the columns, so let us skip them.

From above, we saw that Installs and Price columns have unwanted characters such as a plus sign, dollar sign, or comma. Therefore, I deleted these characters and converted columns to a numeric type.

	Rating	Reviews	Installs	Price
count	8738.000000	1.000000e+04	9.999000e+03	10000.000000
mean	4.198363	4.775455e+05	1.661268e+07	1.093744
std	0.507935	3.044132e+06	8.839671e+07	16.600865
min	1.000000	0.000000e+00	0.000000e+00	0.000000
25%	4.000000	4.600000e+01	5.000000e+03	0.000000
50%	4.300000	2.801500e+03	1.000000e+05	0.000000
75%	4.500000	6.355125e+04	5.000000e+06	0.000000
max	5.000000	7.815831e+07	1.000000e+09	400.000000

Figure 8: Descriptive Statistics

After preprocessing the data, I continued with exploratory data analysis. In Figure 8, we can see the descriptive statistics. I prepared some visuals in order to better understand the

behavior of data. Firstly, I dropped duplicated rows with respect to the App column. After that, I preferred to group the data by category column, aggregated Ratings column by mean and Reviews column by their sum. In order to see which category has the highest rating, I sorted the data by descending order. From the chart, it can be seen from Figure 9 that the Events category has the highest average rating while the Dating category has the lowest.

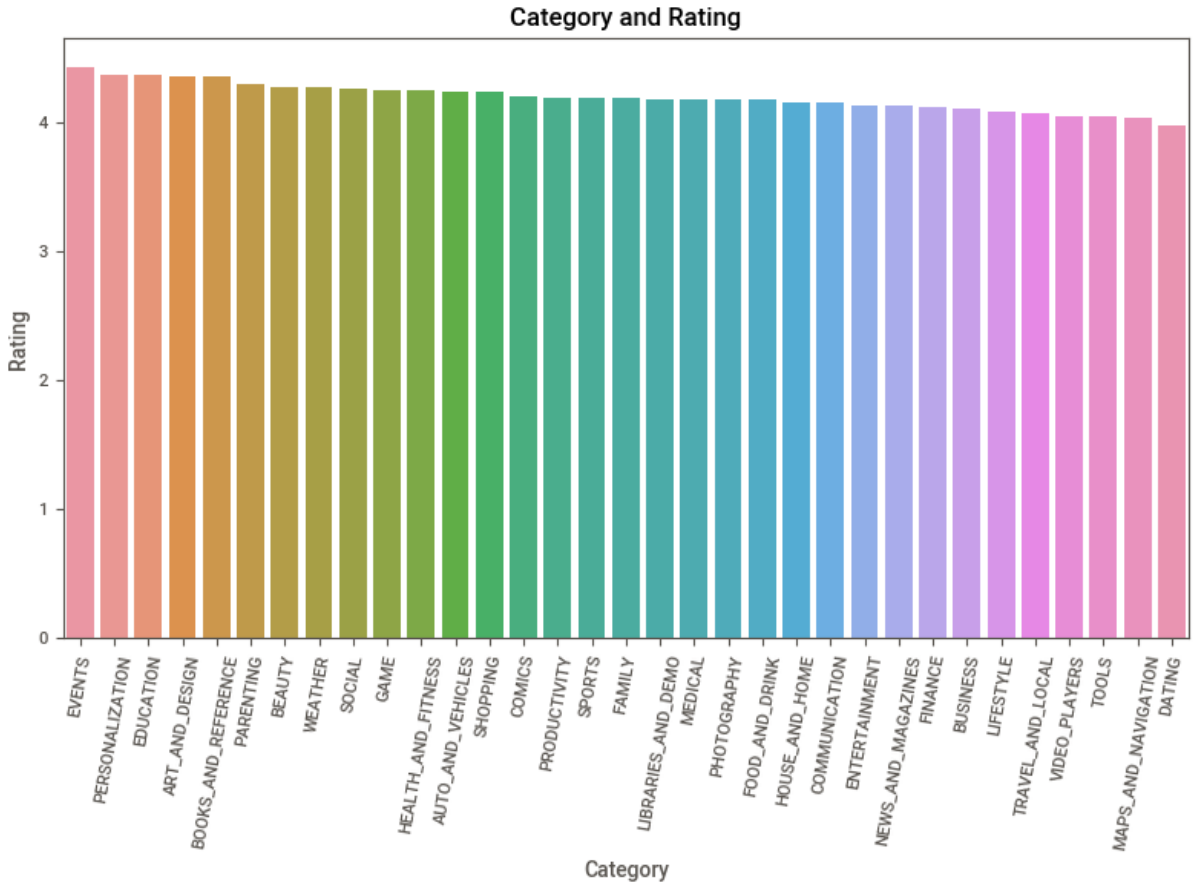

Figure 9: Category vs. Mean Rating

By applying the same procedure for Reviews, we can understand that the Game category has the highest total reviews (See Figure 10). If we were to look at the total number of installs, similarly, the game category also has the highest number of total installs, and the communication category follows it (See Figure 11).

Now, let us have a look at the total number of reviews with respect to the apps (Figure 12), not categories. As can be seen, Facebook, WhatsApp, and Instagram have the top 3 highest reviews; however, they are not in the games category. How can this be possible? We saw from the Sweetviz part that the games category has the second highest number of apps in this data. Therefore, their sum is greater than communication and social categories.

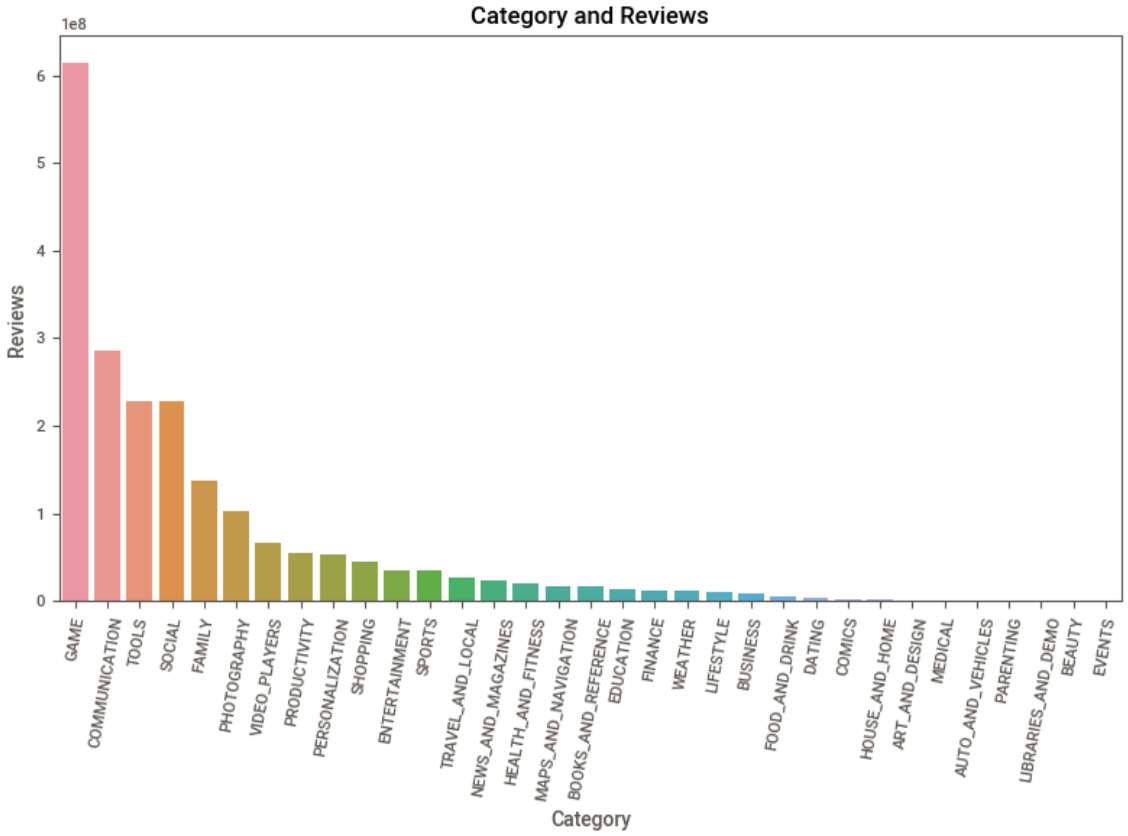

Figure 10: Category vs. Total Reviews

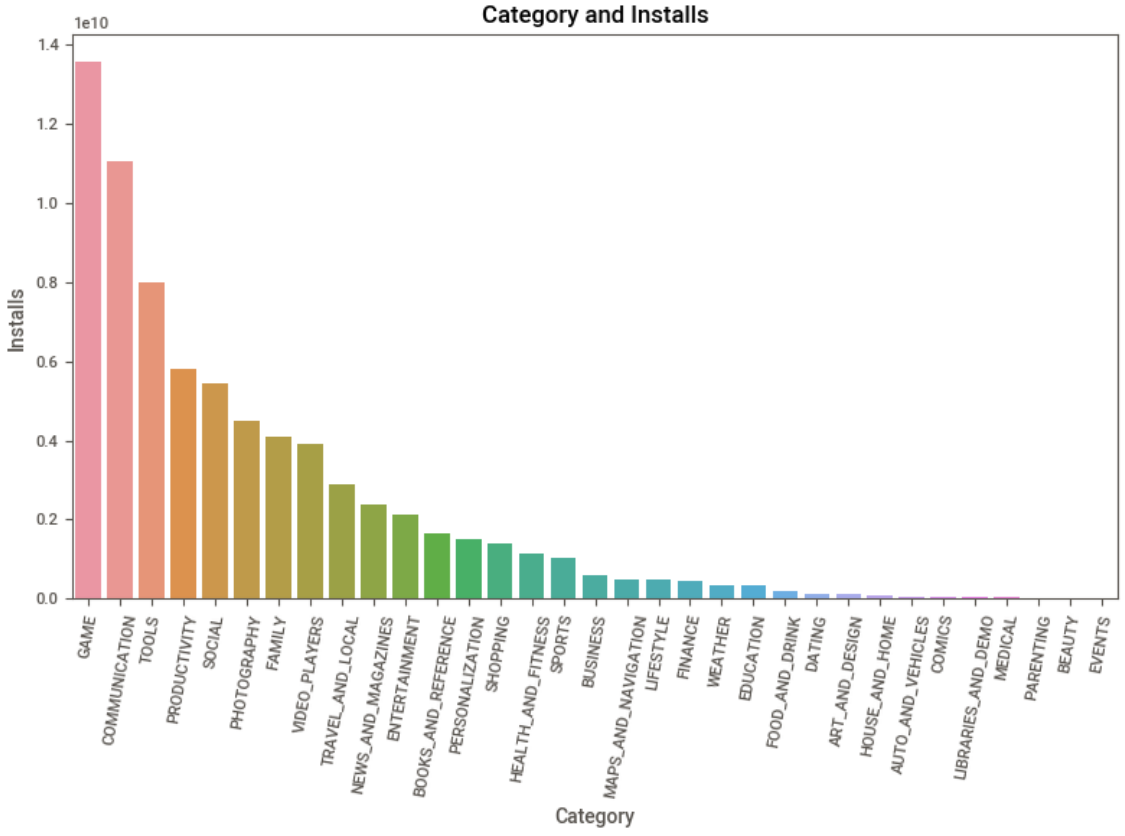

Figure 11: Category vs. Total Installs

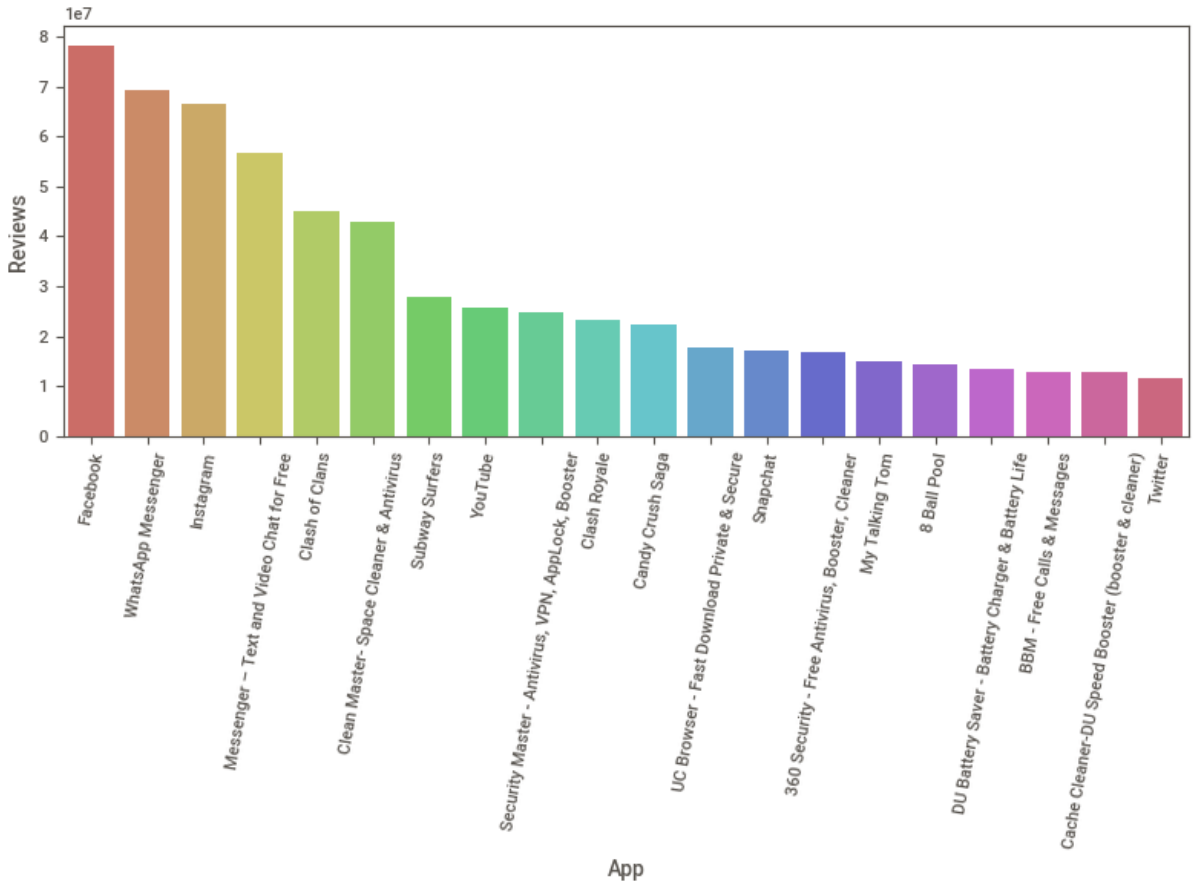

Figure 12: Apps vs. Total Reviews

Now, let's group this data by type column, which has two different values, such as Free or Paid (Figure 13). From this table, it is clear that free apps have much higher reviews compared to paid ones (521 fold), as expected. However, paid apps have a higher mean rating.

Type	Reviews	Rating
Free	2064834655	4.172233
Paid	6405610	4.271174

Figure 13: Total Reviews and Mean Rating with respect to Type column

From Figure 14, we can see the correlation between numeric columns; in order to better understand the high correlation, I constructed a heat map (Figure 15). As you can see from here, the highest correlation takes place between the Installs column and the Reviews column. This is parallel to common sense because we expect that high installation may result in high reviews.

	Rating	Reviews	Installs	Price
Rating	1.000000	0.056175	0.040293	-0.022608
Reviews	0.056175	1.000000	0.624765	-0.008079
Installs	0.040293	0.624765	1.000000	-0.009952
Price	-0.022608	-0.008079	-0.009952	1.000000

Figure 14: Correlation Matrix

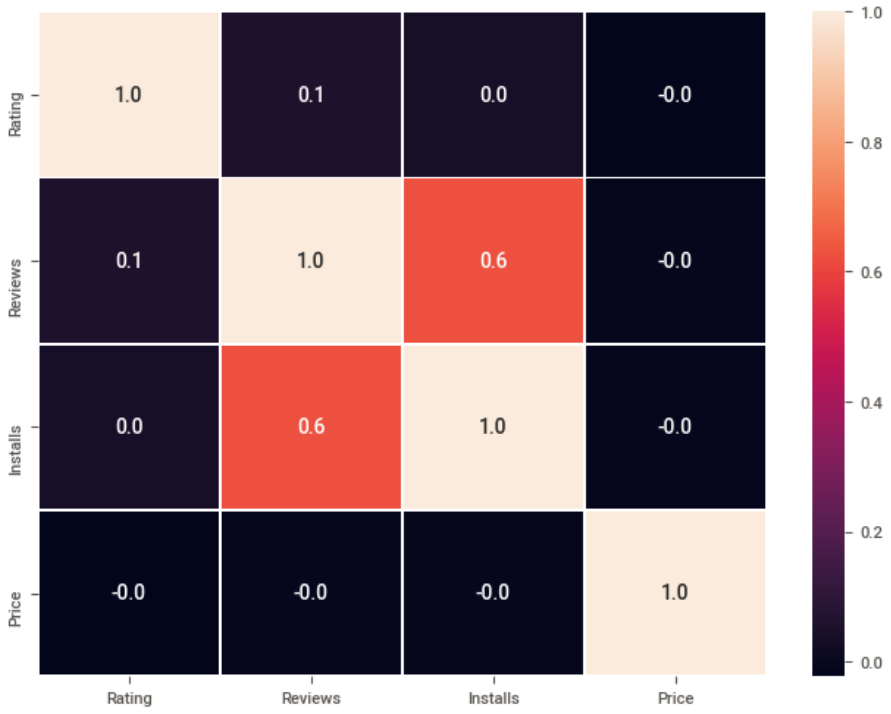

Figure 15: Correlation Matrix in Heat Map format

Having completed the exploratory data analysis part, I continue with the machine learning part by partitioning the data with train and test sections. In this part, I will use unsupervised machine learning methods. I chose three numeric features, namely rating, installs, and price, to construct a prediction model for my target column, which is Reviews. Firstly, I started with a linear regression model to serve this aim. Model intercept and coefficients of Rating, Installs, and Price columns are calculated as follows:

Model Intercept: -518066.22475449

Rating Coefficient: 1.47599681e+05

Installs Coefficient: 1.58356705e-02

Price Coefficient: -2.50155825e+02

Finally, the accuracy of this model is calculated as %40. By using the cross-validation method, we can see that there are even worse accuracy scores, and the median of them is 0.368. This value is not desired, so I need to search for different methods. Therefore, I tried Gaussian Naïve Bayes Algorithm; however, the accuracy score was even worse (%0.8).

Finally, I decided to use Polynomial Regression with a range of 0 to 7. You can see the resulting figure of training and validation scores below. It can be seen from the figure when the degree is equal to 2; we get higher accuracy results. Now, let's prove it by using Grid Search Algorithm. Grid Search best parameter results are as follows:{'linearregression__fit_intercept': False, 'linearregression__normalize': True, 'polynomialfeatures__degree': 2}. The result proves that choosing a degree of 2 is consistent with the result we get from Figure 16. For the degree of 2, the median of cross-validation scores is calculated as 0.427, which is slightly better than the result of degree 1 (0.368). I know that even with this improvement, the result is not satisfying. However, it was the best that I could achieve among the algorithms that I have studied so far.

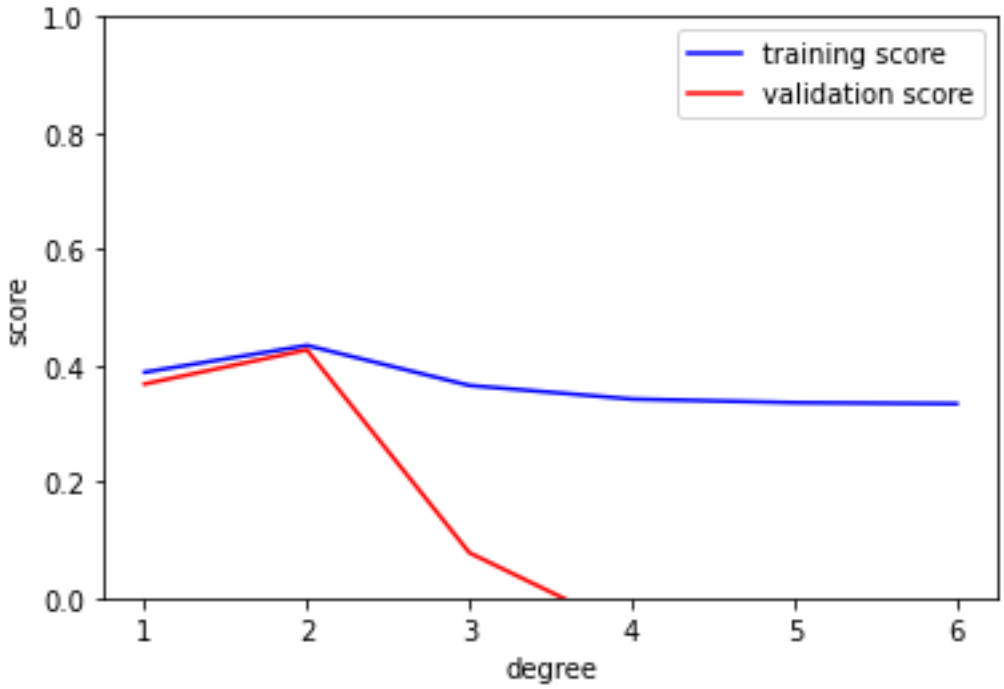

Figure 16: Training and Validation Scores

To conclude, working with Google Play Store Apps data was informative because the content is familiar and easy to understand. I hope that I will get better accuracy scores with different machine learning algorithms in the future.