

AMAZON ALEXA REVIEWS SENTIMENT ANALYSIS

SUBMITTED BY: SAHIL ABBAS NAQVI

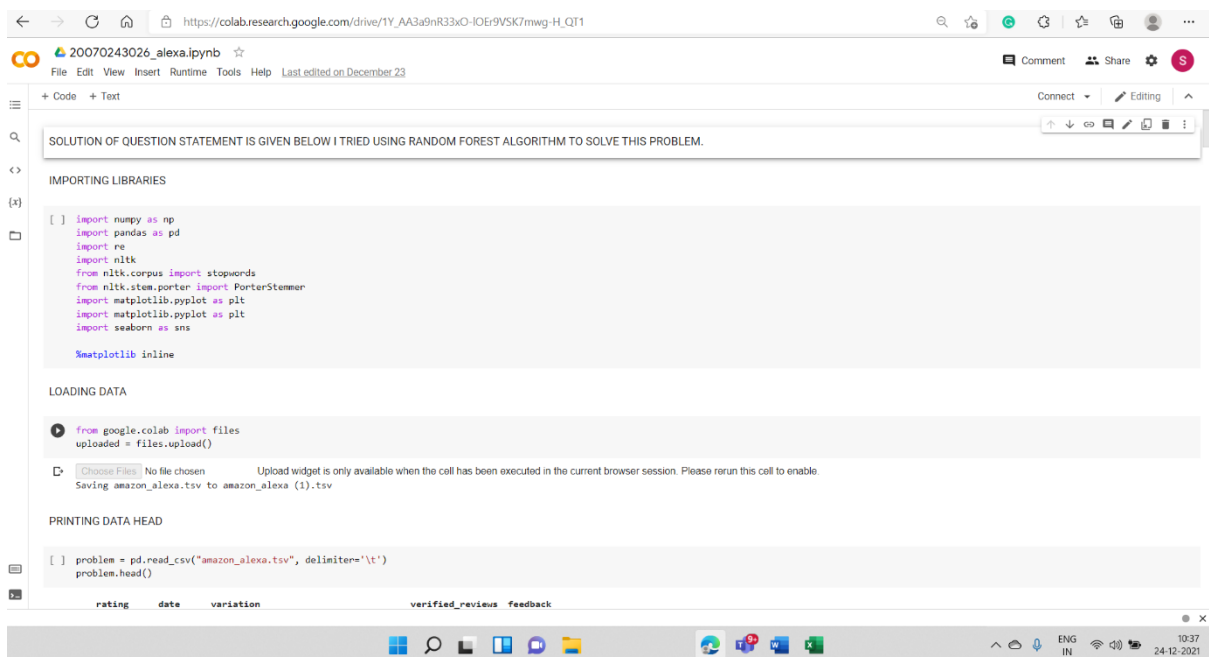
PRN: 20070243026

INTRODUCTION

Amazon is an American multinational corporation that focuses on e-commerce, cloud computing, digital streaming, and artificial intelligence products. But it is mainly known for its e-commerce platform which is one of the biggest online shopping platforms today. There are so many customers buying products from Amazon that today Amazon earns an average of \$ 638.1 million per day. So having such a large customer base, it will turn out to be an amazing data science project if we can analyse the sentiments of Amazon Alexa reviews. So, in this project, I will walk you through the task of Amazon Alexa Reviews Sentiment Analysis with Python.

Amazon Alexa Reviews Sentiment Analysis with Python

The dataset I'm using for the task of Amazon Alexa reviews sentiment analysis was downloaded from Kaggle. This dataset contains the product reviews of over 3150 customers who have purchased Alexa from Amazon. So, let's start this task by importing the necessary Python libraries and the dataset:



```
20070243026_alexa.ipynb
File Edit View Insert Runtime Tools Help Last edited on December 23

+ Code + Text

SOLUTION OF QUESTION STATEMENT IS GIVEN BELOW I TRIED USING RANDOM FOREST ALGORITHM TO SOLVE THIS PROBLEM.

IMPORTING LIBRARIES

[ ] import numpy as np
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

LOADING DATA

from google.colab import files
uploaded = files.upload()

[ ] Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving amazon_alex.tsv to amazon_alex (1).tsv

PRINTING DATA HEAD

[ ] problem = pd.read_csv("amazon_alex.tsv", delimiter='\t')
problem.head()

rating    date    variation    verified_reviews    feedback
```

The screenshot shows a Jupyter Notebook titled '20070243026_alexas.ipynb' in Google Colab. The code cell contains the following Python code:

```
problem = pd.read_csv("amazon_alexas.tsv", delimiter='\t')
problem.head()
```

The output displays the first five rows of the dataset:

	rating	date	variation	verified_review	feedback
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1

Below this, the notebook shows the tail of the dataset:

```
problem.tail()
```

	rating	date	variation	verified_review	feedback
3145	5	30-Jul-18	Black Dot	Perfect for kids, adults and everyone in betwe...	1
3146	5	30-Jul-18	Black Dot	Listening to music, searching locations, check...	1
3147	5	30-Jul-18	Black Dot	I do love these things, I have them running my...	1
3148	5	30-Jul-18	White Dot	Only complaint I have is that the sound qualit...	1
3149	4	29-Jul-18	Black Dot	Good	1

The notebook also includes a section titled 'EDA ON OUR GIVEN DATA SET FOR BETTER UNDERSTANDING'.

DATA PRE-PROCESSING

For better understanding of our dataset pre-processing is must. So, I have decided to apply some pre-processing steps on my dataset so that I gain more information out of it.

First I describe my dataset using **describe()** method and then I check for null values if there are any null values exist.

The screenshot shows a Jupyter Notebook titled '20070243026_alexas.ipynb' in Google Colab. The code cell contains the following Python code:

```
problem.describe()

[ ] problem.isnull().sum()

[ ] problem.groupby('rating').count()
```

The output displays the statistical summary of the dataset:

	rating	feedback
count	3150.000000	3150.000000
mean	4.463175	0.918413
std	1.068508	0.273778
min	1.000000	0.000000
25%	4.000000	1.000000
50%	5.000000	1.000000
75%	5.000000	1.000000
max	5.000000	1.000000

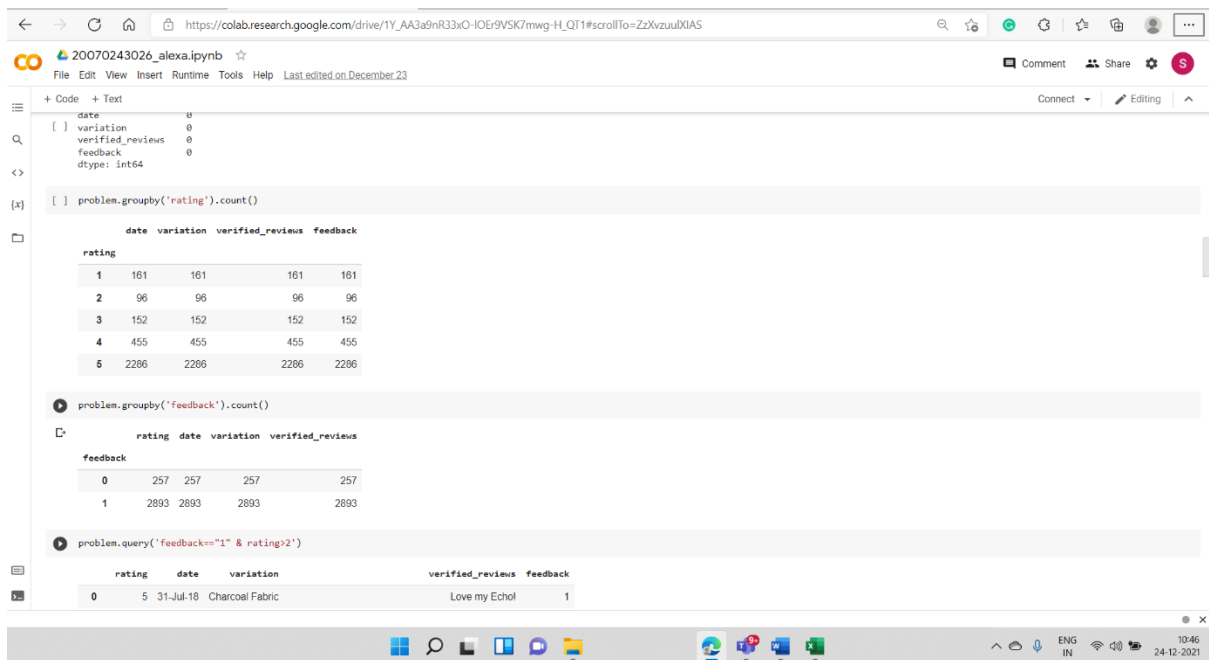
The output also shows the result of the null value check:

```
[ ] problem.isnull().sum()
rating      0
date        0
variation   0
verified_review 0
feedback    0
dtype: int64
```

Finally, the output shows the count of data points grouped by rating:

```
[ ] problem.groupby('rating').count()
date variation verified_review feedback
```

After that I grouped my data by using **groupby()** method based on ratings and feedback column.



The screenshot shows a Google Colab notebook with the following code and output:

```
problem.groupby('rating').count()
```

rating	date	variation	verified_reviews	feedback
1	161	161	161	161
2	96	96	96	96
3	152	152	152	152
4	455	455	455	455
5	2286	2286	2286	2286

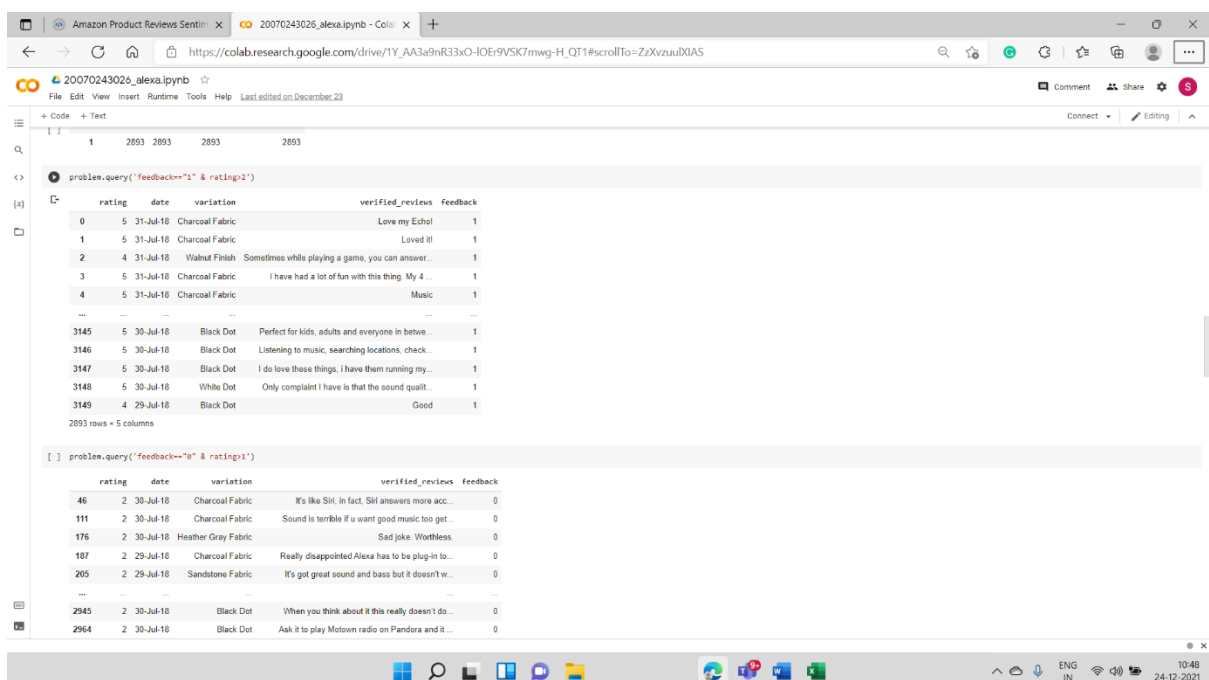
```
problem.groupby('feedback').count()
```

feedback	rating	date	variation	verified_reviews
0	257	257	257	257
1	2893	2893	2893	2893

```
problem.query('feedback=="1" & ratings>2')
```

rating	date	variation	verified_reviews	feedback	
0	5	31-Jul-18	Charcoal Fabric	Love my Echol	1

Then I use **query()** method to know what number of records exist for different feedback and ratings.



The screenshot shows a Google Colab notebook with the following code and output:

```
problem.query('feedback=="1" & ratings>2')
```

rating	date	variation	verified_reviews	feedback	
0	5	31-Jul-18	Charcoal Fabric	Love my Echol	1
1	5	31-Jul-18	Charcoal Fabric	Loved It!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1
...
3145	5	30-Jul-18	Black Dot	Perfect for kids, adults and everyone in betwe...	1
3146	5	30-Jul-18	Black Dot	Listening to music, searching locations, check...	1
3147	5	30-Jul-18	Black Dot	I do love these things, I have them running my...	1
3148	5	30-Jul-18	White Dot	Only complaint I have is that the sound qual...	1
3149	4	29-Jul-18	Black Dot	Good	1

2893 rows x 5 columns

```
[ ] problem.query('feedback=="0" & ratings>1')
```

rating	date	variation	verified_reviews	feedback	
46	2	30-Jul-18	Charcoal Fabric	It's like Siri, in fact, Siri answers more acc...	0
111	2	30-Jul-18	Charcoal Fabric	Sound is terrible if u want good music too get...	0
176	2	30-Jul-18	Heather Gray Fabric	Sad joke. Worthless.	0
187	2	29-Jul-18	Charcoal Fabric	Really disappointed Alexa has to be plug-in to...	0
205	2	29-Jul-18	Sandstone Fabric	It's got great sound and bass but it doesn't w...	0
...
2945	2	30-Jul-18	Black Dot	When you think about it this really doesn't do...	0
2964	2	30-Jul-18	Black Dot	Ask it to play Motown radio on Pandora and it...	0

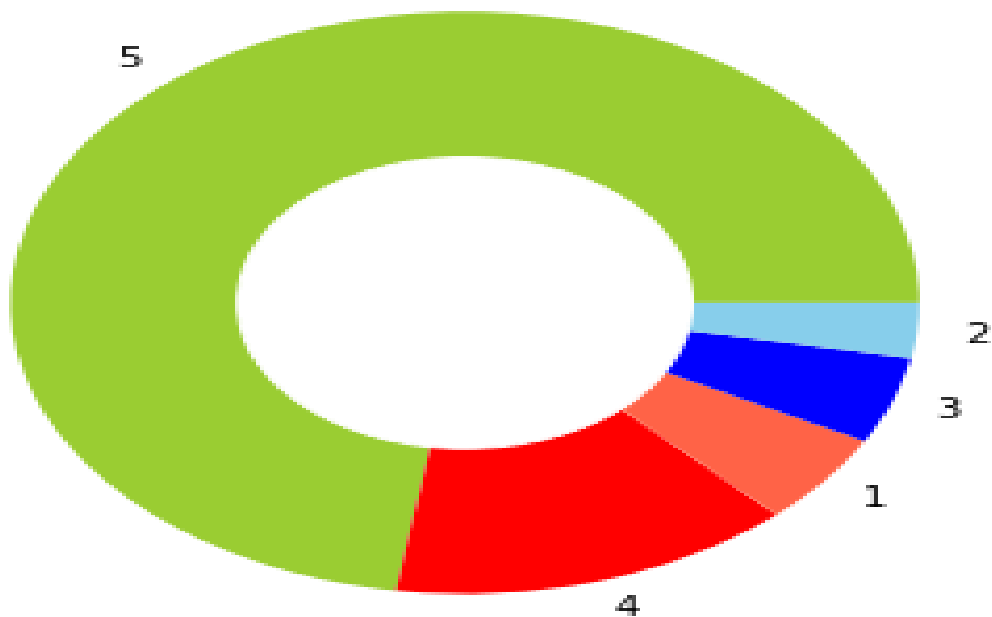
SENTIMENT ANALYSIS OF AMAZON ALEXA REVIEWS

The Rating column of this dataset contains the ratings that customers have given to the Amazon Alexa based on their experience with the product. So, let's look at the rating breakdown to see how most customers rate Alexa they buy from Amazon:

```
ratings = problem["rating"].value_counts()
numbers = ratings.index
quantity = ratings.values

custom_colors = ["yellowgreen", "red", "tomato", "blue", "skyblue"]
plt.figure(figsize=(5, 5))
plt.pie(quantity, labels=numbers, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Alexa_Ratings", fontsize=20)
plt.show()
```

Alexa_Reviews



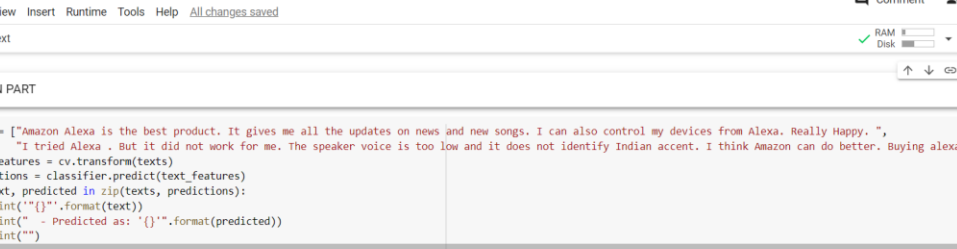
According to the figure above, more than half of people rated Alexa they bought from Amazon with 5 stars, which is good.

BUILDING RANDOM FOREST MODEL

[illegible]

Mean Absolute Error of the Random Forest Model that we build was found out to be **0.05** which is extremely good as per my opinion. And the accuracy score of this model was found out to be **94%**. Similarly, the F1 score is somewhere around **96**.

PREDICTION PART



```
0s completed at 11:13 AM

[ ] texts = ["Amazon Alexa is the best product. It gives me all the updates on news and new songs. I can also control my devices from Alexa. Really Happy. ",
            "I tried Alexa . But it did not work for me. The speaker voice is too low and it does not identify Indian accent. I think Amazon can do better. Buying alexa is waste of money"]

text_features = cv.transform(texts)
predictions = classifier.predict(text_features)
for text, predicted in zip(texts, predictions):
    print("{} {}".format(text))
    print(" - Predicted as: {}".format(predicted))
    print("")

"Amazon Alexa is the best product. It gives me all the updates on news and new songs. I can also control my devices from Alexa. Really Happy. "
- Predicted as: '1'

"I tried Alexa . But it did not work for me. The speaker voice is too low and it does not identify Indian accent. I think Amazon can do better. Buying alexa is waste of money"
- Predicted as: '1'
```