MAHATMA EDUCATION SOCIETY’S

PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE (Autonomous)

NEW PANVEL

PROJECT REPORT ON

**‘Sentiment Analysis of Amazon Alexa Customer Reviews’**

IN PARTIAL FULFILLMENT OF

MASTER OF SCIENCE DATA ANALYTICS (PART II) SEMESTER III– 2024-25

PROJECT GUIDE

Prof. Dipti Khiste

SUBMITTED BY: SHREYA BHATTACHARJEE

ROLL NO: 6861

**PILLAI COLLEGE OF ARTS,COMMERCE & SCIENCE (Autonomous)**

**Re-accredited “A” Grade by NAAC (3rd Cycle) **

**Project Completion Certificate**

This is to certify that**Shreya Bhattacharjee** of **M.Sc. Data Analytics Part – 2** has completed the project titled ‘**Sentiment Analysis of Amazon Alexa Customer Reviews**’ of subject **Sentiment Analysis** under our guidance and supervision **Prof. Dipti Khiste** during the academic year 2024-25 in the department of Computer Science.

**INDEX**

|  |  |  |
| --- | --- | --- |
| SR NO. | CONTENT |  |
| 1. | INTRODUCTION |  |
| 2. | PROJECT SUMMARY   * Project Overview * Objective * Key Components * Findings * Challenges * Future |  |
| 3. | PYTHON CODE AND OUTPUT   * Model Training * Model Evaluation |  |
| 4. | CONCLUSION |  |

**INTRODUCTION**

The "Sentiment Analysis of Amazon Alexa Customer Reviews" project aims to analyze customer feedback to understand the overall sentiment towards Amazon Alexa products. By leveraging natural language processing (NLP) techniques, the project identifies and classifies the sentiment expressed in customer reviews as positive, negative, or neutral. This analysis provides valuable insights into customer satisfaction and areas for improvement. The project utilizes various machine learning algorithms and data visualization tools to present findings in an easily interpretable format, ultimately helping Amazon to enhance product features and customer experience. By understanding the sentiment trends, Amazon can prioritize product updates and address common customer concerns more effectively. Furthermore, this analysis can assist in marketing strategies by highlighting the strengths appreciated by users, thereby fostering a stronger connection with the customer base.

**PROJECT SUMMARY**

**PROJECT OVERVIEW**:

The "Sentiment Analysis of Amazon Alexa Customer Reviews" project focuses on developing a system to analyze customer feedback and understand the overall sentiment towards Amazon Alexa products. The primary goal is to classify customer reviews into positive, negative, or neutral categories and provide actionable insights for product improvement and customer satisfaction enhancement.

**OBJECTIVE:**

The goal of this project is to develop a system for analyzing the sentiment of Amazon Alexa customer reviews. The system aims to determine whether customer feedback has a positive, negative, or neutral sentiment and to identify key areas for improvement based on this sentiment analysis.

**KEY COMPONENTS:**

**1. Data Collection:**

Gather a dataset of Amazon Alexa customer reviews. This data can be sourced from Amazon's website, Kaggle, or other relevant datasets.

**2. Sentiment Analysis:**

**Text Processing:** Use natural language processing (NLP) techniques to pre-process customer reviews (e.g., tokenization, stemming, and removal of stop words).

**Sentiment Classification**: Apply sentiment analysis models to classify the sentiment of each review as positive, negative, or neutral. This can be achieved using machine learning models such as Logistic Regression, Multinomial Naive Bayes, Support Vector Machine.

**3. Correlation Analysis:**

Analyze the relationship between the sentiment scores of customer reviews and product ratings. This involves examining how positive or negative feedback correlates with overall product ratings and identifying any patterns or trends.

**4. Model Evaluation:**

Evaluate the performance of sentiment analysis models using metrics such as accuracy, precision, recall, and F1 score.

**5. Application:**

**Customer Insights**: Provide actionable insights for Amazon based on the sentiment analysis, helping them make informed decisions about product improvements and customer service enhancements.

**Product Development:** Use sentiment insights to prioritize product updates and feature enhancements based on customer feedback.

**FINDINGS:**

**Sentiment Impact:** Positive reviews generally correlate with higher product ratings, while negative reviews are associated with lower ratings. Neutral reviews have a less predictable impact.

**Predictive Insights:** The sentiment analysis provided valuable predictive insights into customer satisfaction and areas for potential product improvement.

**CHALLENGES:**

**Data Quality**: Ensuring the accuracy and relevance of customer reviews and product rating data.

**Sentiment Complexity**: Handling nuances in language and context to improve sentiment classification accuracy.

**Volume of Data:** Managing and analyzing a large volume of customer feedback efficiently.

**FUTURE DIRECTIONS:**

**Model Improvement**: Enhance sentiment analysis models by integrating more sophisticated NLP techniques and expanding training datasets.

**Real-Time Analysis**: Develop systems for real-time sentiment analysis to provide immediate insights and impact assessments.

**Holistic Approach:** Incorporate additional data sources, such as social media sentiment and competitor product reviews, for a more comprehensive analysis of customer feedback.

**OUTCOME:**

The successful implementation of this project will result in a robust tool for analysing and interpreting the sentiment of Amazon Alexa customer reviews. This can offer valuable insights for Amazon to enhance product features, improve customer satisfaction, and make informed decisions about product development and marketing strategies.

**PYTHON CODE AND OUTPUT:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

from matplotlib import style

style.use("ggplot")

import seaborn as sns

import re

import nltk

nltk.download('stopwords')

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Unzipping corpora/stopwords.zip.

True

from nltk.tokenize import word\_tokenize

from nltk.stem import PorterStemmer

from nltk.corpus import stopwords

stop\_words = set(stopwords.words('english'))

from wordcloud import WordCloud

from sklearn.feature\_extraction.text import CountVectorizer

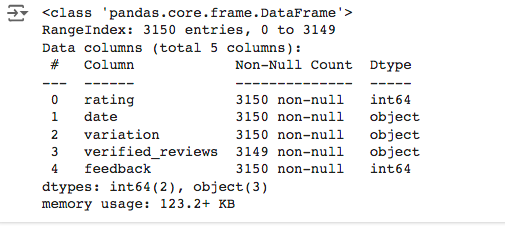
from sklearn.model\_selection import train\_test\_split

df = pd.read\_csv("/content/amazon\_alexa.tsv" , sep = '\t')

df.head()



df.info()



sns.countplot(x='rating', data=df, palette=colors)

plt.show()

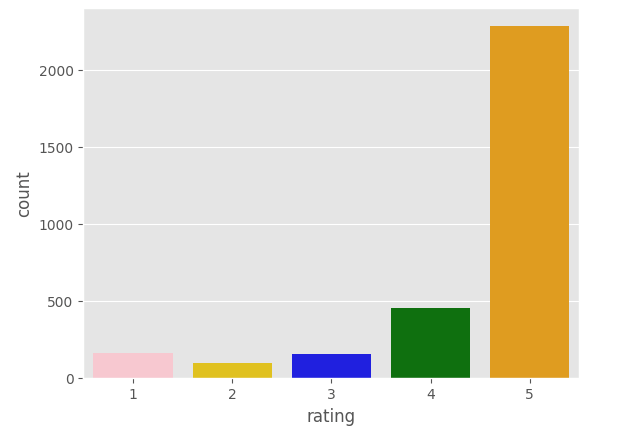


fig = plt.figure(figsize = (7,7))

colors = ('red', 'gold','yellowgreen','cyan','orange')

wp = {'linewidth': 2, 'edgecolor':'black'}

tags = df['rating'].value\_counts()

explode = (0.2,0.2,0.2,0.3,0.2)

tags.plot(kind ='pie', autopct = '%1.1f',colors = colors, shadow = True,

startangle =0, wedgeprops =wp, explode = explode, label ='')

plt.title('Distrubution of the different Ratings')

plt.show()

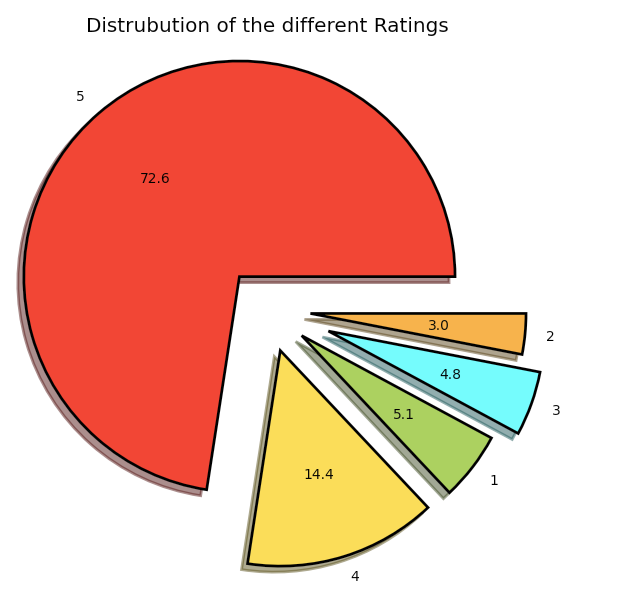
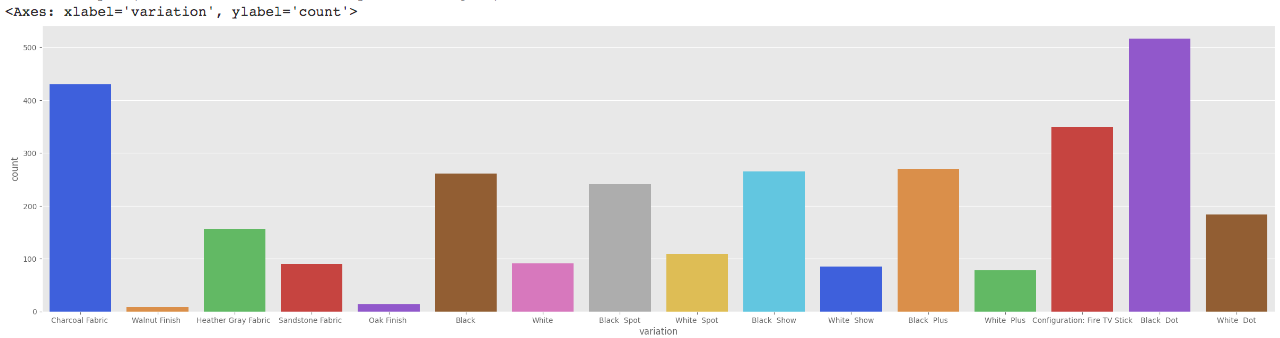
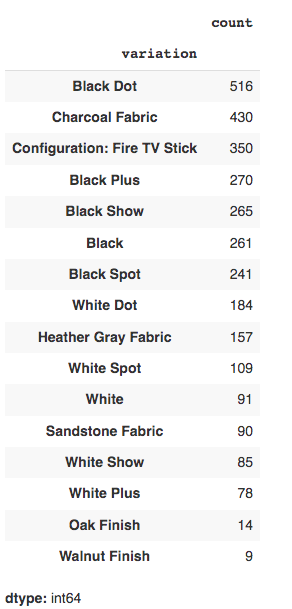


fig = plt.figure(figsize = (30,7))

sns.countplot(x='variation', data = df, palette='bright')

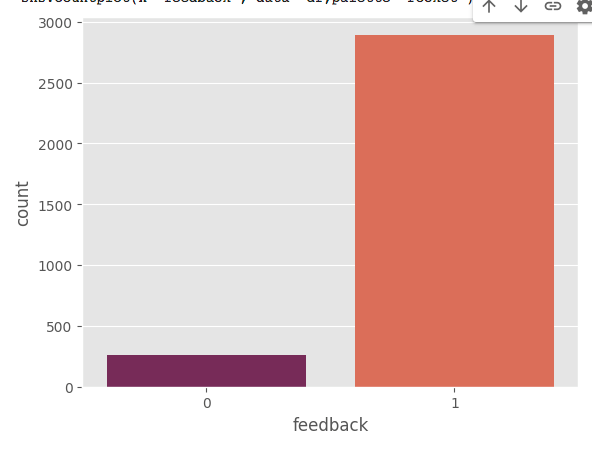


df['variation'].value\_counts()



sns.countplot(x='feedback', data =df,palette='rocket') # Replace rocket with a string that represents a valid palette name.

plt.show()



ratings = df["rating"].value\_counts()

numbers = ratings.index

quantity = ratings.values

custom\_colors = ["skyblue", "yellowgreen", 'tomato', "blue", "red"]

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("Amazon Alexa Reviews", fontsize=20)

plt.show()



import nltk

nltk.download('vader\_lexicon') # Download lexicon for sentiment analysis

from nltk.sentiment.vader import SentimentIntensityAnalyzer # Import the class

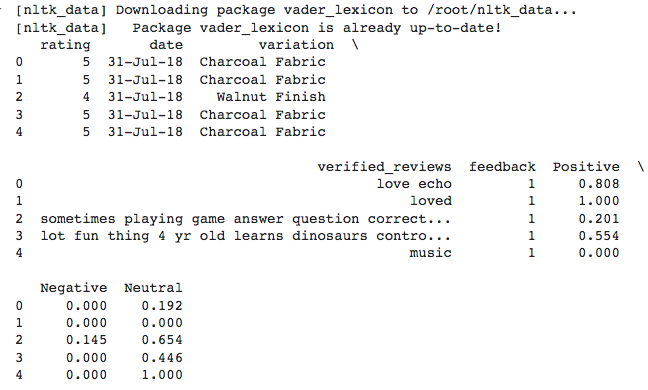
sentiments = SentimentIntensityAnalyzer()

df["Positive"] = [sentiments.polarity\_scores(i)["pos"] for i in df["verified\_reviews"]]

df["Negative"] = [sentiments.polarity\_scores(i)["neg"] for i in df["verified\_reviews"]]

df["Neutral"] = [sentiments.polarity\_scores(i)["neu"] for i in df["verified\_reviews"]]

print(df.head())



x = sum(df["Positive"])

y = sum(df["Negative"])

z = sum(df["Neutral"])

def sentiment\_score(a, b, c):

if (a>b) and (a>c):

print("Positive 😊 ")

elif (b>a) and (b>c):

print("Negative 😠 ")

else:

print("Neutral 🙂 ")

sentiment\_score(x, y, z)



print("Positive: ", x)

print("Negative: ", y)

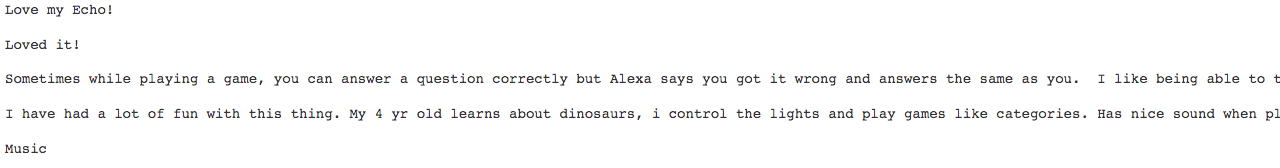
print("Neutral: ", z)



So we can see that Positive and Neutral are above 1000 where Negative is below 100. So this means that most of the customers of Amazon Alexa are satisfied with its services.

for i in range(5):

print(df['verified\_reviews'].iloc[i],"\n")



import nltk

nltk.download('punkt')

def data\_processing(text):

if isinstance(text, str): # Check if text is a string

text = text.lower()

text = re.sub(r"http\S+www\S+|https\S+", '', text, flags=re.MULTILINE)

text = re.sub(r'[^\w\s]', '', text)

text\_tokens = word\_tokenize(text)

filtered\_text = [w for w in text\_tokens if not w in stop\_words]

return " ".join(filtered\_text)

else:

return "" # Return empty string for non-string values

df.verified\_reviews = df['verified\_reviews'].apply(data\_processing)

pos\_reviews = df[df.feedback == 1]

pos\_reviews.head()



text = ' '.join([word for word in pos\_reviews['verified\_reviews']])

plt.figure(figsize=(20,15), facecolor='None')

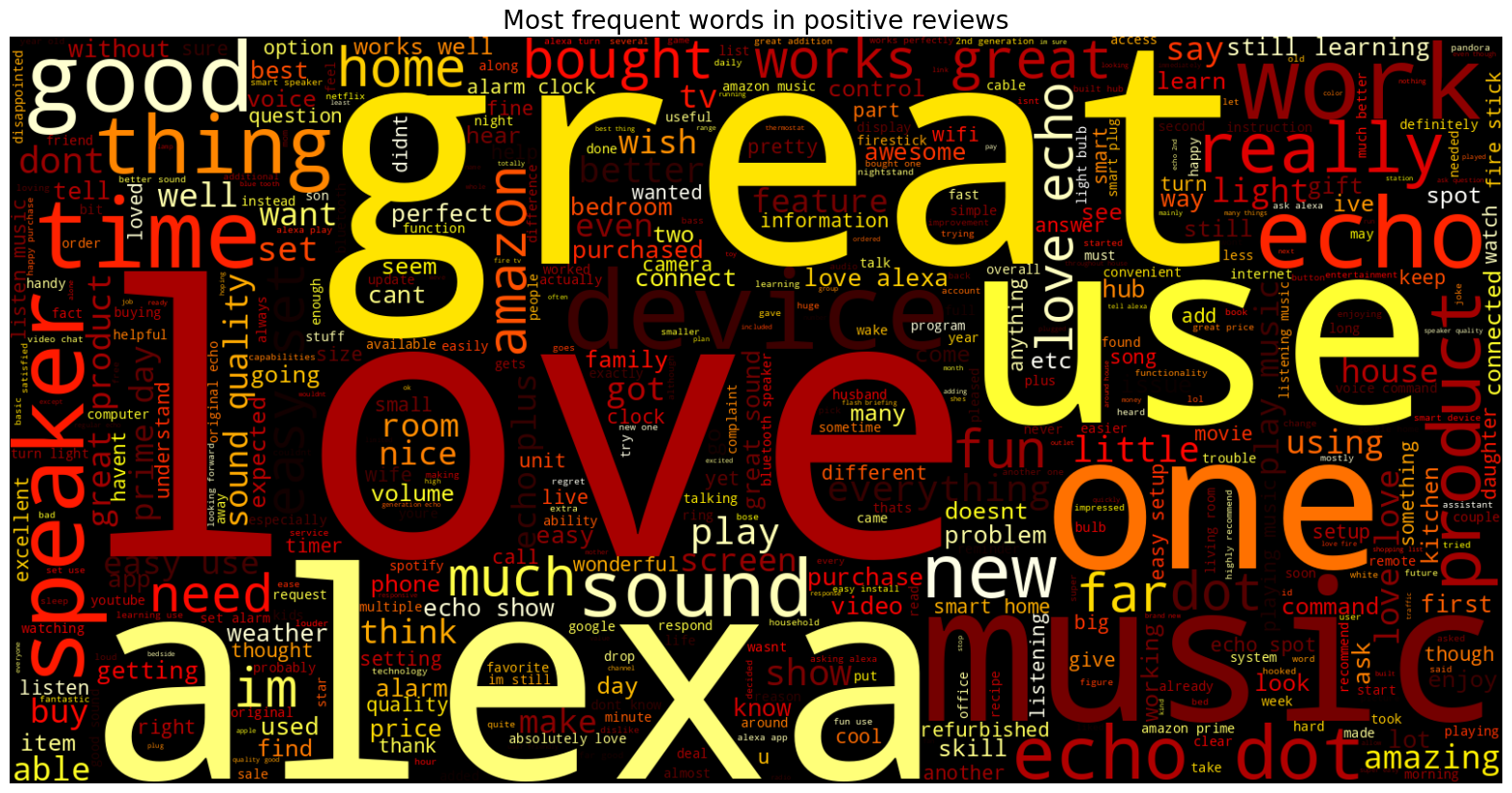
wordcloud = WordCloud(max\_words=500, width=1600, height=800, colormap='hot').generate(text)

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.title('Most frequent words in positive reviews', fontsize=19)

plt.show()



neg\_reviews = df[df.feedback==0]

neg\_reviews.head()



text = ' '.join([word for word in neg\_reviews['verified\_reviews']])

plt.figure(figsize=(20,15), facecolor='None')

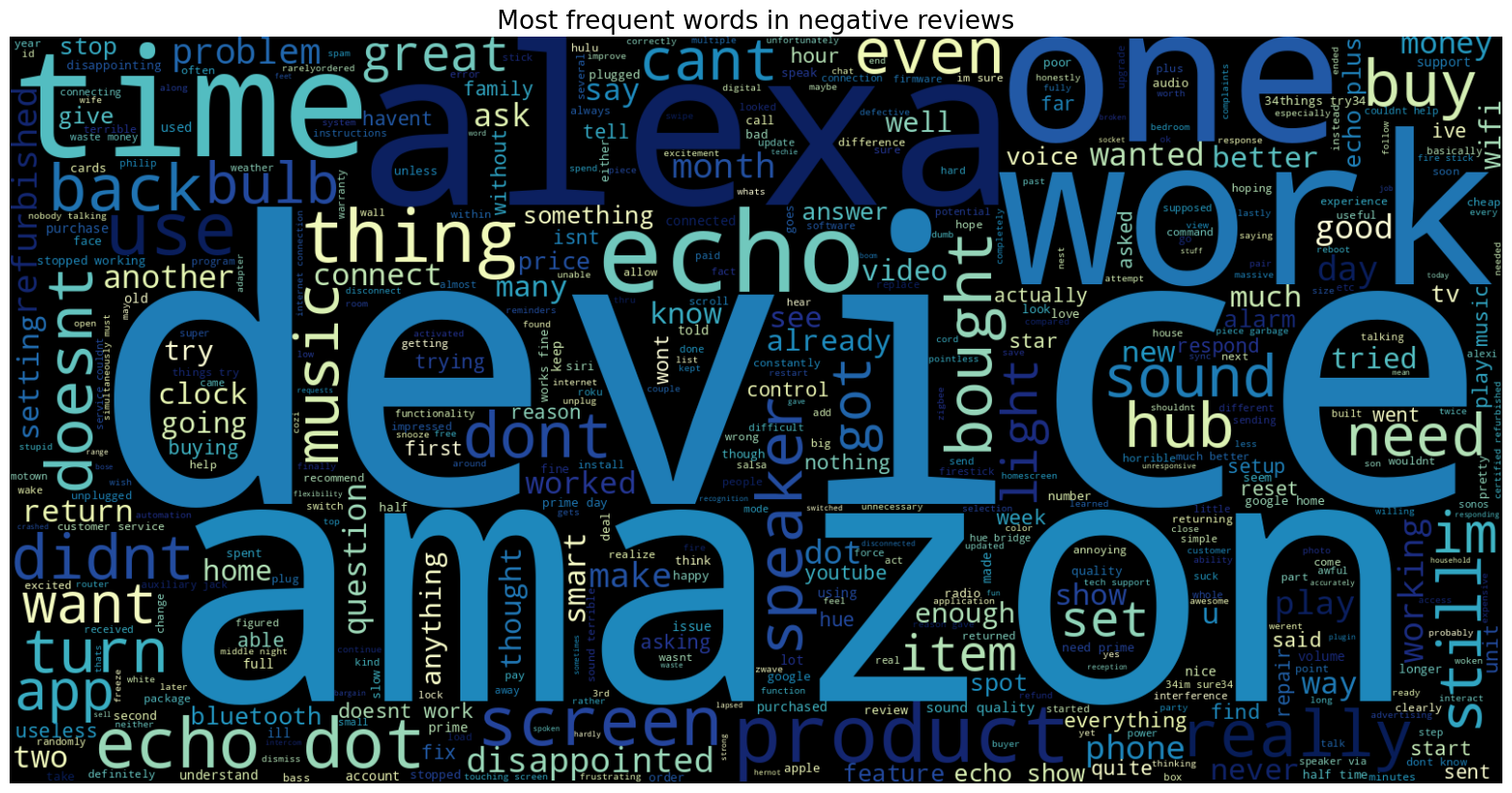
wordcloud = WordCloud(max\_words=500, width=1600, height=800, colormap='YlGnBu').generate(text)

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.title('Most frequent words in negative reviews', fontsize=19)

plt.show()



X = df['verified\_reviews']

Y = df['feedback']

cv = CountVectorizer()

X = cv.fit\_transform(df['verified\_reviews'])

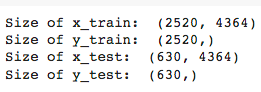
x\_train, x\_test, y\_train, y\_test = train\_test\_split(X,Y, test\_size=0.2, random\_state=42)

print("Size of x\_train: ",(x\_train.shape))

print("Size of y\_train: ",(y\_train.shape))

print("Size of x\_test: ",(x\_test.shape))

print("Size of y\_test: ",(y\_test.shape))



from sklearn.linear\_model import LogisticRegression

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

logreg = LogisticRegression()

logreg.fit(x\_train, y\_train)

logreg\_pred = logreg.predict(x\_test)

logreg\_acc = accuracy\_score(logreg\_pred, y\_test)

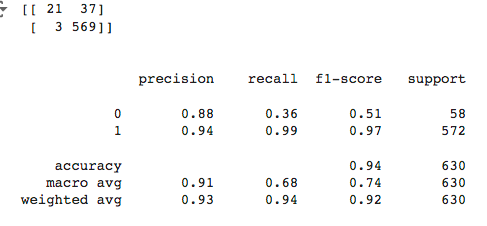
print("Test accuracy: {:.2f}%".format(logreg\_acc\*100))



print(confusion\_matrix(y\_test, logreg\_pred))

print("\n")

print(classification\_report(y\_test, logreg\_pred))



mnb = MultinomialNB()

mnb.fit(x\_train, y\_train)

mnb\_pred = mnb.predict(x\_test)

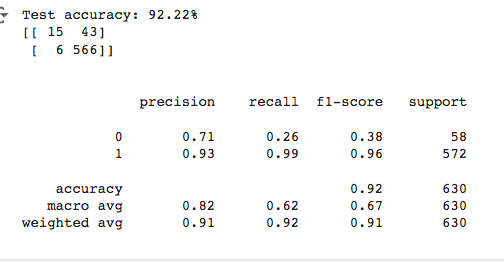
mnb\_acc = accuracy\_score(mnb\_pred, y\_test)

print("Test accuracy: {:.2f}%".format(mnb\_acc\*100))

print(confusion\_matrix(y\_test, mnb\_pred))

print("\n")

print(classification\_report(y\_test, mnb\_pred))



from sklearn.svm import SVC

svm = SVC()

svm.fit(x\_train, y\_train)

svm\_pred = svm.predict(x\_test)

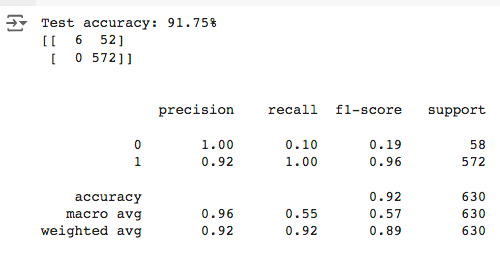
svm\_acc = accuracy\_score(svm\_pred, y\_test)

print("Test accuracy: {:.2f}%".format(svm\_acc\*100))

print(confusion\_matrix(y\_test, svm\_pred))

print("\n")

print(classification\_report(y\_test, svm\_pred))



from sklearn.model\_selection import GridSearchCV

# Hyperparameter tuning for Logistic Regression

param\_grid = {'C': [0.1, 1, 10]}

grid\_search = GridSearchCV(logreg, param\_grid, cv=5)

grid\_search.fit(x\_train, y\_train)

print("Best parameters for Logistic Regression:", grid\_search.best\_params\_)

print("Best cross-validation score for Logistic Regression:", grid\_search.best\_score\_)

# Hyperparameter tuning for Multinomial Naive Bayes

param\_grid = {'alpha': [0.1, 1, 10]}

grid\_search = GridSearchCV(mnb, param\_grid, cv=5)

grid\_search.fit(x\_train, y\_train)

print("Best parameters for Multinomial Naive Bayes:", grid\_search.best\_params\_)

print("Best cross-validation score for Multinomial Naive Bayes:", grid\_search.best\_score\_)

# Hyperparameter tuning for Support Vector Machine

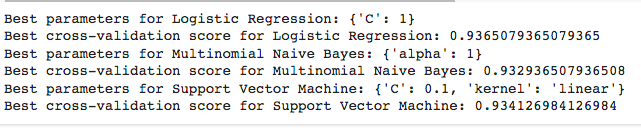
param\_grid = {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf']}

grid\_search = GridSearchCV(svm, param\_grid, cv=5)

grid\_search.fit(x\_train, y\_train)

print("Best parameters for Support Vector Machine:", grid\_search.best\_params\_)

print("Best cross-validation score for Support Vector Machine:", grid\_search.best\_score\_)



import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Assuming you have already defined logreg, mnb, svm, and their predictions

models = [

('Logistic Regression', logreg, logreg\_pred),

('Multinomial Naive Bayes', mnb, mnb\_pred),

('Support Vector Machine', svm, svm\_pred)

]

for name, model, pred in models:

print(f"--- {name} ---")

print("Accuracy: {:.2f}%".format(accuracy\_score(y\_test, pred) \* 100))

# Confusion Matrix Visualization

cm = confusion\_matrix(y\_test, pred)

plt.figure(figsize=(5, 5))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.title(f'Confusion Matrix - {name}')

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.show()

# Classification Report Visualization (Bar Chart)

report = classification\_report(y\_test, pred, output\_dict=True)

metrics = ['precision', 'recall', 'f1-score']

classes = list(report.keys())[:-3] # Exclude 'accuracy', 'macro avg', 'weighted avg'

plt.figure(figsize=(8, 6))

for metric in metrics:

scores = [report[class\_][metric] for class\_ in classes]

plt.bar(classes, scores, label=metric)

plt.xlabel('Class')

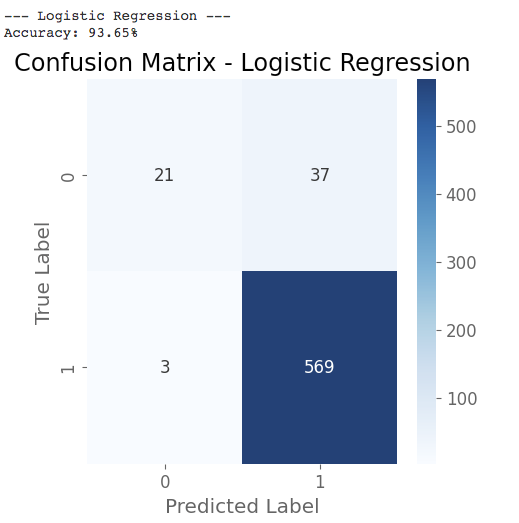
plt.ylabel('Score')

plt.title(f'Classification Report - {name}')

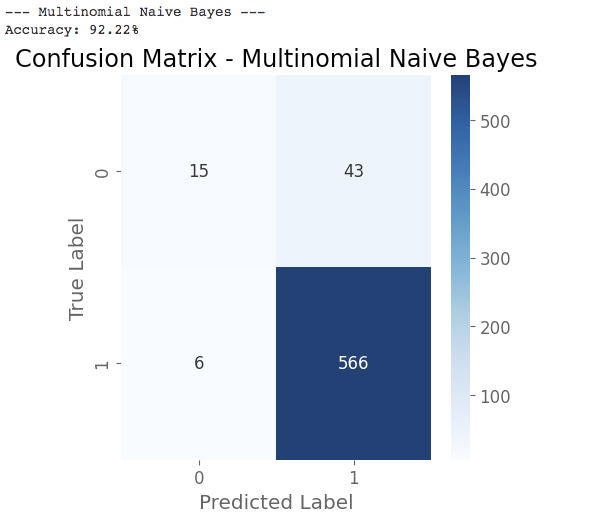
plt.legend()

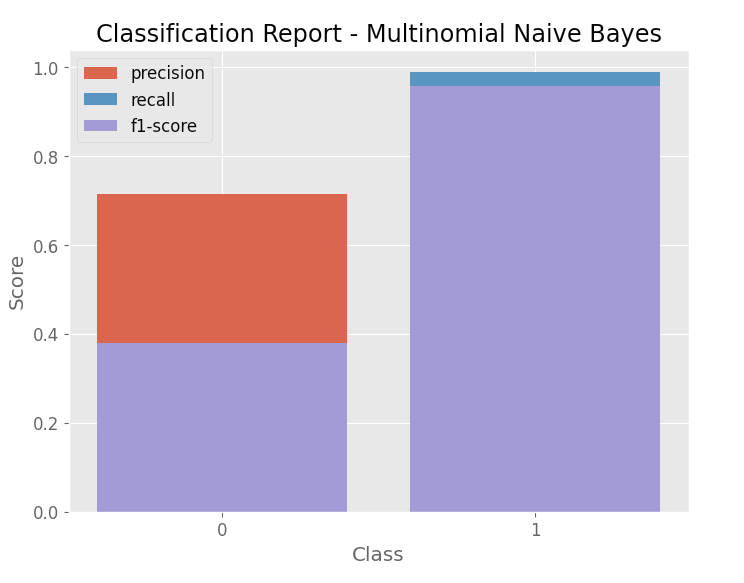
plt.show()

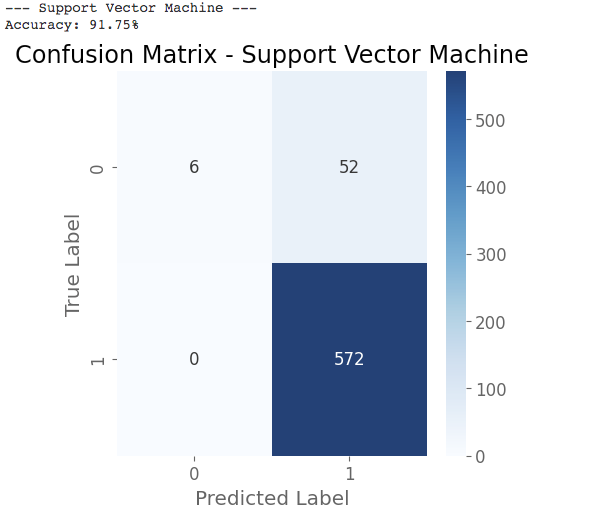
print("\n")

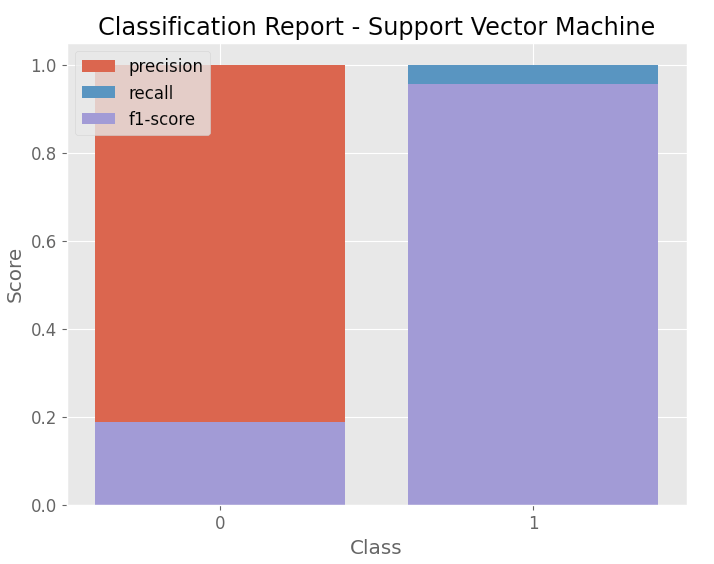












**CONCLUSION**

In conclusion, this project demonstrates the potential of sentiment analysis in transforming customer feedback into actionable insights, ultimately helping Amazon to enhance its products and strengthen its relationship with customers. Further advancements in NLP techniques and the integration of additional data sources will continue to improve the accuracy and impact of sentiment analysis in the future.