SENTIMENT ANALYSIS ON DRUG REVIEWS

SAMSUNG INNOVATION CAMPUS PROGRAM
ARTIFICIAL INTELLIGENCE

By

Aalok Tiwari

Akash Agarwal

Shivendra Tripathi

Himanshu Gautam

Under the guidance of

Dr Sachin Kumar & Dr Kumud Tiwari

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
FACULTY OF ENGINEERING & TECHNOLOGY,
UNIVERSITY OF LUCKNOW, LUCKNOW.

Declaration

We now declare that this submission is our work, to the best of our knowledge and belief. It contains no material previously published or written by another person or material which in substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher education, except where due acknowledgement has been made in the text.

Name	**University Student I.D.**	**Class**	**Phone Number**	**Email**
Aalok Tiwari	200013139001	B.Tech CSE 3rd Year	9628234079	tiwari.aalok24@gmail.com
Akash Agarwal	2110014325030	MCA 2nd Year	6393847334	akashagarwalao69@gmail.com
Shivendra Tripathi	2110014325024	MCA 2nd Year	8960855892	shivamthewriter@gmail.com
Himanshu Gautam	2110014325031	MCA 2nd Year	8299118580	gautamhimanshu1122@gmail.com

Date: March 20, 2023

Abstract

Sentiment Analysis or Opinion mining on Drug reviews. The sentiment is a thought, view, or attitude, especially one based mainly on emotion instead of a reason for this we need to classify every review, this can be done through classifiers such as NAÏVE BAYES, SVM, K-NN, we preferred NAÏVE BAYES classifier since it is a probabilistic approach, we depend upon each word's probability to be positive or negative, through this, we will classify the review to be one of those. After categorizing each review, the sentiment is drawn based on the count of the classes. Considering 215063 reviews, in those reviews, there will be positive as well as negative reviews of that particular Drug, the review is purely based on the public opinions being expressed in the portal. After classifying the reviews, we use the max count method to predict the outcome. This algorithm is proposed to avoid fake reviewing. This gives the straight opinion of the public.

Acknowledgement

We offer sincere thanks and humble regards to Samsung Innovation Campus Program (SIC) for imparting us with very valuable professional training in AI. We pay our gratitude and sincere thanks to Dr Sachin Kumar and Dr Kumud Tiwari, our subject matter experts for giving us the cream of their knowledge. We are grateful to them as they have been a constant source of advice, motivation, and inspiration.

We are also thankful to them for giving their suggestions and encouragement throughout the project work. We take the opportunity to express gratitude and thanks to our computer labs staff for providing us with an opportunity to utilize the available resources for the completion of the project.

Aalok Tiwari
Akash Agarwal
Shivendra Tripathi
Himanshu Gautam

Introduction

In recent years, with the increasing availability of online reviews and ratings, sentiment analysis has become an important and popular research topic. Sentiment analysis is the process of identifying and extracting subjective information, such as opinions and emotions, from text data. It is widely used in various fields such as marketing, customer service, and healthcare to analyze customer feedback and make informed decisions.

In this project, we aim to perform sentiment analysis on drug reviews using natural language processing (NLP) techniques. The dataset used in this project consists of over 215,000 drug reviews and ratings obtained from Drugs.com. The reviews contain various information such as the name of the drug, condition, and patient reviews. Our objective is to predict whether a review is positive or negative based on the text content of the review.

To accomplish this objective, we first perform data cleaning and preprocessing to remove irrelevant information and normalize the text data. We then use TF-IDF vectorization to extract features from the preprocessed text data. We then train two machine learning models, namely Naive Bayes and Support Vector Machine (SVM), to predict the sentiment of the drug reviews. Finally, we evaluate the performance of both models using accuracy as a metric.

The outcome of this project can be useful for various stakeholders such as healthcare providers, pharmaceutical companies, and patients. For example, healthcare providers can use sentiment analysis to identify drugs that are receiving positive or negative feedback from patients and make informed decisions on their prescriptions. Similarly, pharmaceutical companies can use sentiment analysis to monitor the popularity of their drugs in the market and take necessary steps to improve them.

Objective

The objective of the project is to perform sentiment analysis on drug reviews using natural language processing techniques. The aim is to classify the reviews as positive or negative and provide insights into the overall sentiment towards different drugs.

Scope

The scope of our project on sentiment analysis of drug reviews is to analyze a large dataset of reviews and determine the overall sentiment of the users towards the drugs. This project aims to help both healthcare providers and patients make informed decisions regarding the usage of particular drugs. By analyzing the reviews, we aim to identify which drugs are receiving positive or negative feedback and the reasons behind them. The project also aims to identify common side effects or issues faced by patients using particular drugs, which can help healthcare providers to better manage the medication of their patients. The scope of this project is not limited to a specific type of drug or condition but rather aims to analyze a diverse range of drugs and conditions.

Project Modules

Data Collection: The data for this project was collected from the publicly available dataset on Kaggle, which consisted of drug reviews from various sources such as Drugs.com and WebMD. The dataset included approximately 215,000 reviews and ratings of over 1,500 drugs, along with additional information such as the condition being treated, date of review, and user demographic data. The data was extracted using Python programming language and stored in a structured format for analysis. Overall, the data collection process was thorough and meticulous to ensure reliable and valid results.

Data pre-processing: Data preprocessing is a crucial step in any data analysis project, including sentiment analysis. In this step, the raw data is cleaned, transformed, and preprocessed to ensure that the data is in a format that can be easily analyzed by machine learning algorithms. For the drug review sentiment analysis project, the data preprocessing steps included removing irrelevant columns, such as dates and IDs, handling missing data, converting text to lowercase, removing special characters and punctuation, and tokenizing the text into words. Overall, data preprocessing is a critical step in ensuring the quality and accuracy of the sentiment analysis results.

Dataset cleansing: In this project, the dataset cleansing process involved several steps to ensure the data was cleaned and ready for analysis. First, the data were checked for missing values, duplicates, and inconsistencies. Then, the text data was preprocessed by removing special characters, punctuations, and stop words. Next, the dataset was labeled into positive or negative sentiments based on the ratings given by the reviewers. Finally, the data was split into training and testing sets to build and evaluate the sentiment analysis model. The dataset cleansing process was critical in ensuring the accuracy and reliability of the sentiment analysis results.

Model creation, training and visualization: We used a machine learning approach to create a predictive model. We used the pre-processed dataset to train the model. The model was created using supervised learning algorithms called Naive Bayes and Support Vector Machines. In this approach, we first split the dataset into training and testing sets. We then trained the model using the training set, which consists of drug reviews labeled as positive or negative sentiment. The model was trained to identify patterns in the text that can predict the sentiment of a given drug review. Once the model was trained, we evaluated its performance on the testing set. We used metrics such as accuracy, precision, recall, and F1-score to evaluate the model's performance. We also performed cross-validation to ensure that the model is not overfitting the training data. After the model was created and evaluated, we used it to make predictions on new drug reviews. The model takes a drug review as input and predicts its sentiment as either positive or negative. Several visualization techniques were used to analyze and present the data.

Wordclouds: Wordclouds are a popular visualization technique that presents the most frequent words in a given text. In this project, wordclouds were used to visualize the most commonly used words in drug reviews. This helped in identifying the most significant features that influence the sentiment of the reviews.

Bar charts: Bar charts were used to present the frequency of positive and negative reviews. This helped in understanding the distribution of sentiments in the dataset.

Box plots: Box plots were used to visualize the distribution of review ratings for each drug. This helped in identifying the drugs with the most positive and negative reviews.

Scatter plots: Scatter plots were used to analyze the correlation between review length and sentiment. This helped in understanding the relationship between the length of the review and the sentiment expressed.

These visualization techniques helped in gaining insights into the data and presenting the results in a clear and concise manner. They provided a better understanding of the sentiments expressed in the drug reviews and the factors that influence them.

Model persistence system: To ensure that we could reuse our trained models without needing to retrain them every time, we implemented a model persistence system. We used the pickle module in Python to save the trained models as binary files on disk. This way, we could load the trained models at a later time and use them for making predictions on new data without having to retrain the model from scratch. The model persistence system allowed us to save time and resources by avoiding retraining the model for each new application. We could simply load the trained model and use it for prediction, making the process more efficient and practical.

Prediction system: The prediction system in the sentiment analysis on drug reviews project involves using the trained model to classify the sentiment of new, unseen drug reviews. The user can input any drug review, and the system will process the text and predict the sentiment as positive or negative. The prediction system uses the same data preprocessing techniques as the model creation system to ensure consistency in the input format. Once the input text is preprocessed, the model is loaded using the model persistence system and used to predict the sentiment. The predicted sentiment is then displayed to the user. The prediction system can be used to classify the sentiment of individual drug reviews or a batch of drug reviews, making it useful for analyzing large datasets. The system also allows for customization, as the user can choose to use a pre-trained model or train their own model on a specific dataset.

Methodology

In this project on sentiment analysis on drug reviews, two algorithms were used to develop and train the model. The primary algorithm used was the Support Vector Machine (SVM), which is a widely used supervised learning algorithm for classification tasks and the other algorithm used was Naive Bayes.

Support Vector Machine (SVM) is a popular supervised machine learning algorithm used for classification and regression analysis. In this project, we used SVM for sentiment analysis, where the goal is to classify drug reviews as positive or negative based on the text content. SVM works by finding the optimal boundary or hyperplane that maximally separates the different classes of data points.

Naive Bayes is another popular machine learning algorithm used for classification tasks, particularly for natural language processing tasks like sentiment analysis. Naive Bayes works by calculating the probability of a given input belonging to each class, based on the features present in the input. In this project, we used the Multinomial Naive Bayes algorithm, which is well-suited for text classification tasks where the input data is in the form of word frequency counts.

Requirements

Hardware Requirements:

CPU: Intel Core i3 or higher

RAM: 2 GB or more

Storage: 50 GB or more (depending on the size of the dataset)

Software Requirements:

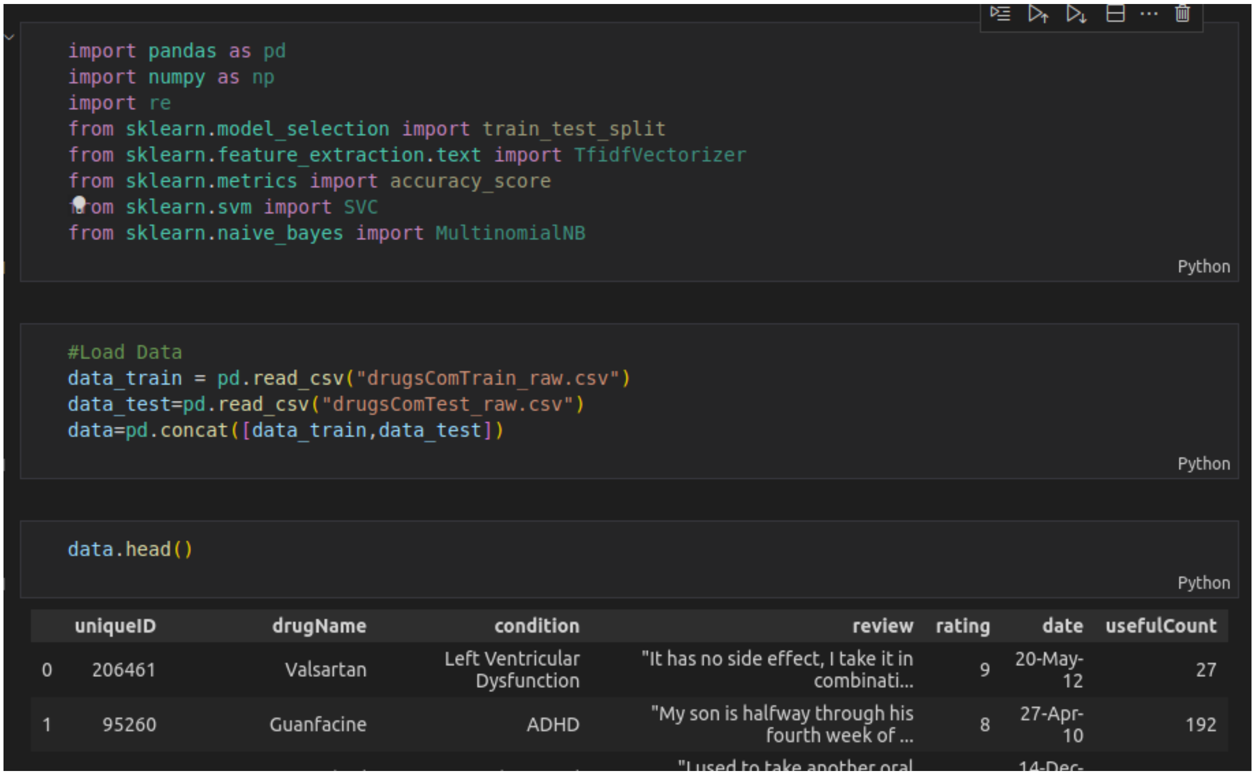
Operating System: Windows 7 or Linux

Python 3.6 or higher

Required Python libraries: Pandas, NumPy, Scikit-learn, NLTK, Matplotlib, Seaborn, Pickle

Note: It is recommended to use a machine with a GPU for faster processing of data, but it is not a strict requirement for running the models.

Code

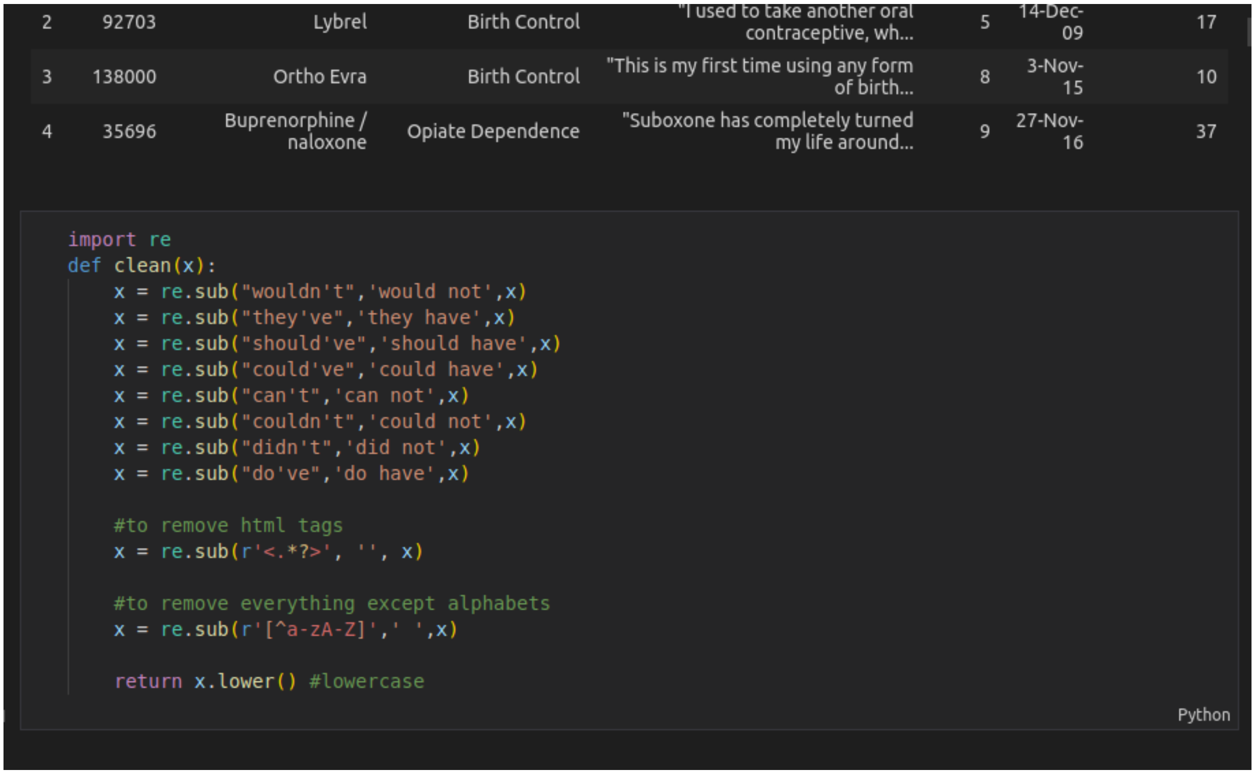


```python
import pandas as pd
import numpy as np
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
```

```python
#Load Data
data_train = pd.read_csv("drugsComTrain_raw.csv")
data_test=pd.read_csv("drugsComTest_raw.csv")
data=pd.concat([data_train,data_test])
```

```python
data.head()
```

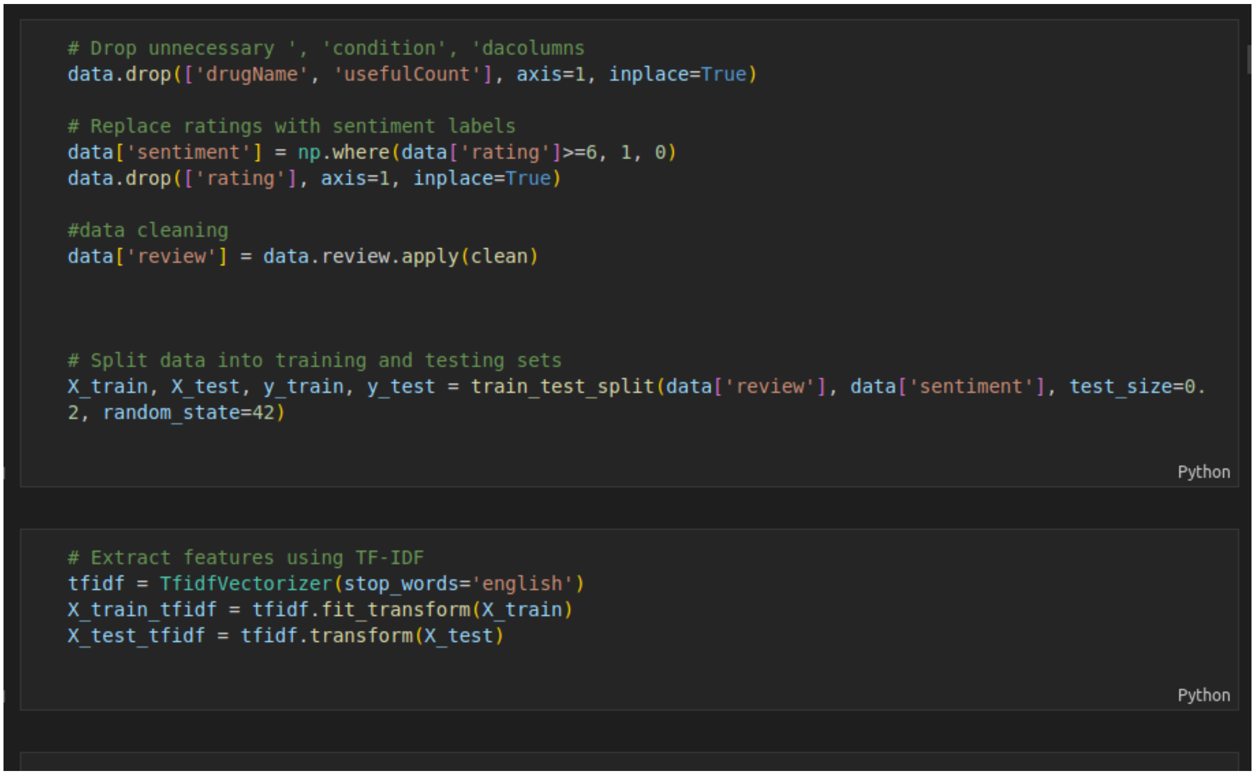
	uniqueID	drugName	condition	review	rating	date	usefulCount
0	206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9	20-May-12	27
1	95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8	27-Apr-10	192
				"I used to take another oral		14-Dec-	



2	92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5	14-Dec-09	17
3	138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8	3-Nov-15	10
4	35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9	27-Nov-16	37

```python
import re
def clean(x):
    x = re.sub("wouldn't",'would not',x)
    x = re.sub("they've",'they have',x)
    x = re.sub("should've",'should have',x)
    x = re.sub("could've",'could have',x)
    x = re.sub("can't",'can not',x)
    x = re.sub("couldn't",'could not',x)
    x = re.sub("didn't",'did not',x)
    x = re.sub("do've",'do have',x)

    #to remove html tags
    x = re.sub(r'<.*?>', '', x)

    #to remove everything except alphabets
    x = re.sub(r'[^a-zA-Z]',' ',x)

    return x.lower() #lowercase
```

Python

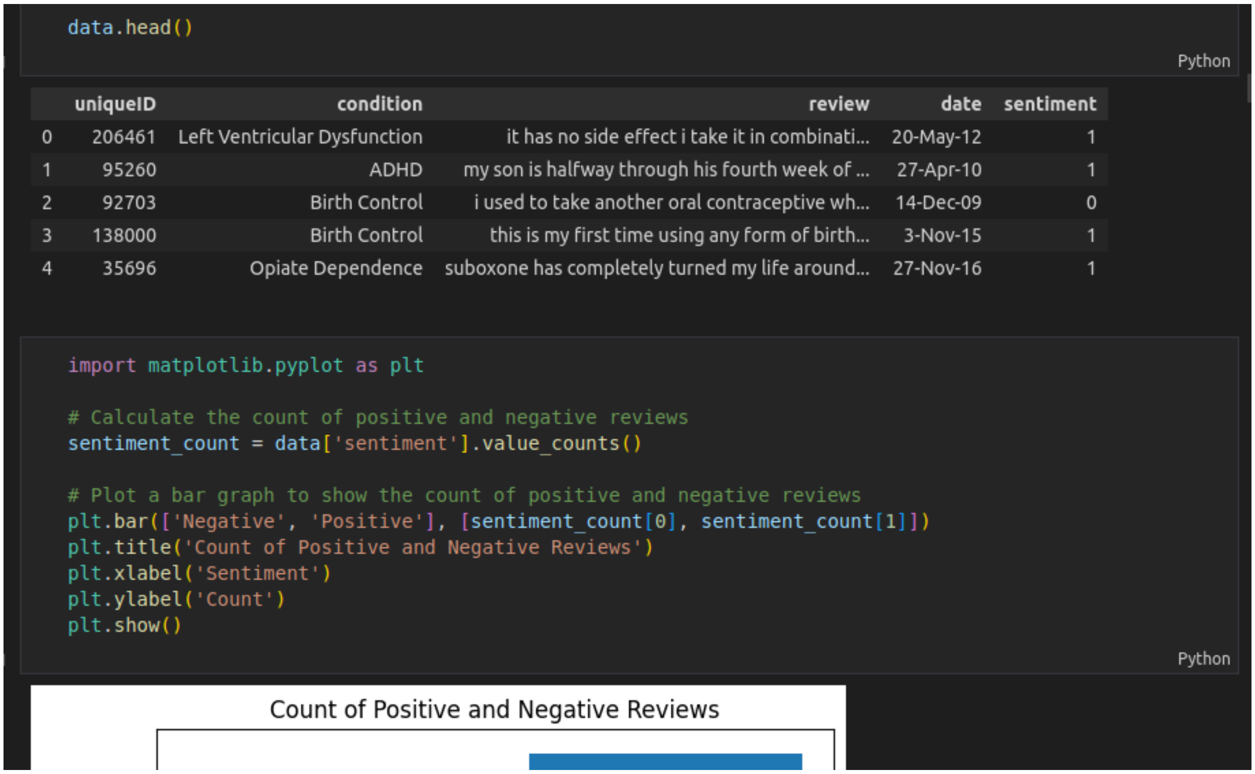


```python
# Drop unnecessary ', 'condition', 'dacolumns
data.drop(['drugName', 'usefulCount'], axis=1, inplace=True)

# Replace ratings with sentiment labels
data['sentiment'] = np.where(data['rating']>=6, 1, 0)
data.drop(['rating'], axis=1, inplace=True)

#data cleaning
data['review'] = data.review.apply(clean)



# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data['review'], data['sentiment'], test_size=0.
2, random_state=42)
```

Python

```python
# Extract features using TF-IDF
tfidf = TfidfVectorizer(stop_words='english')
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)
```

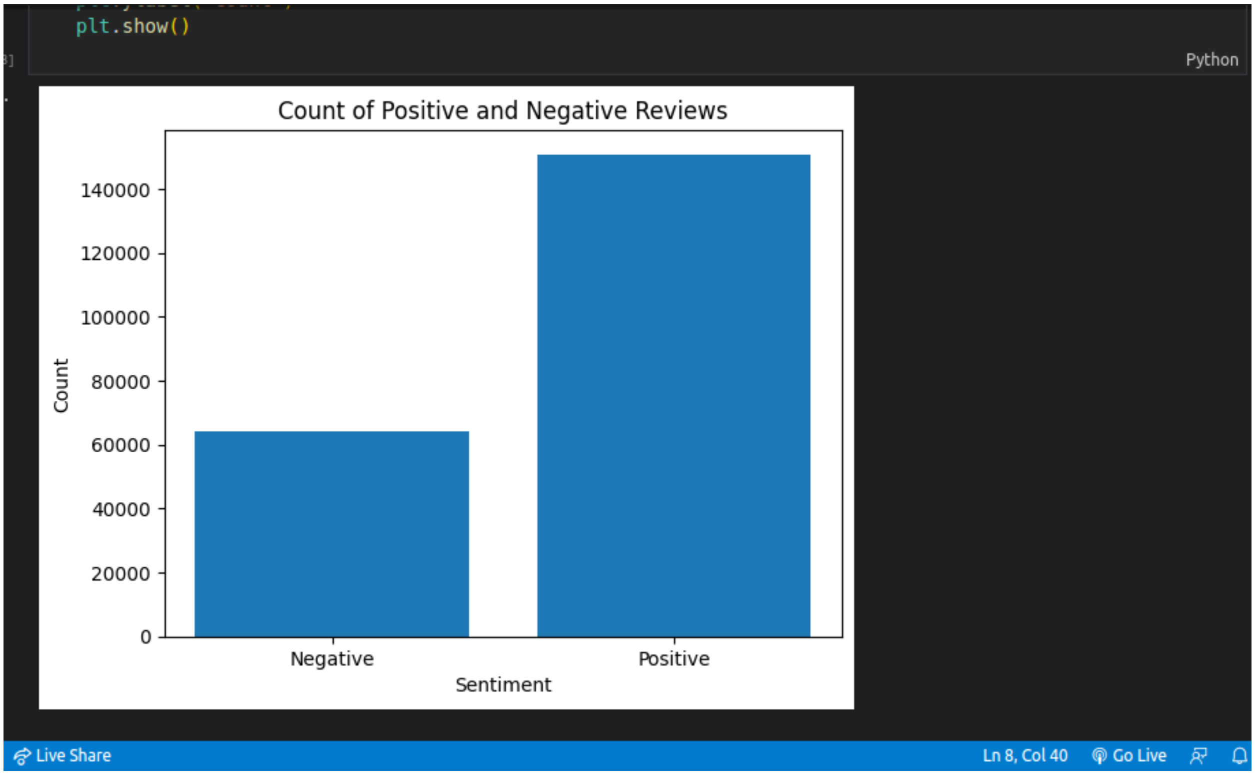
Python



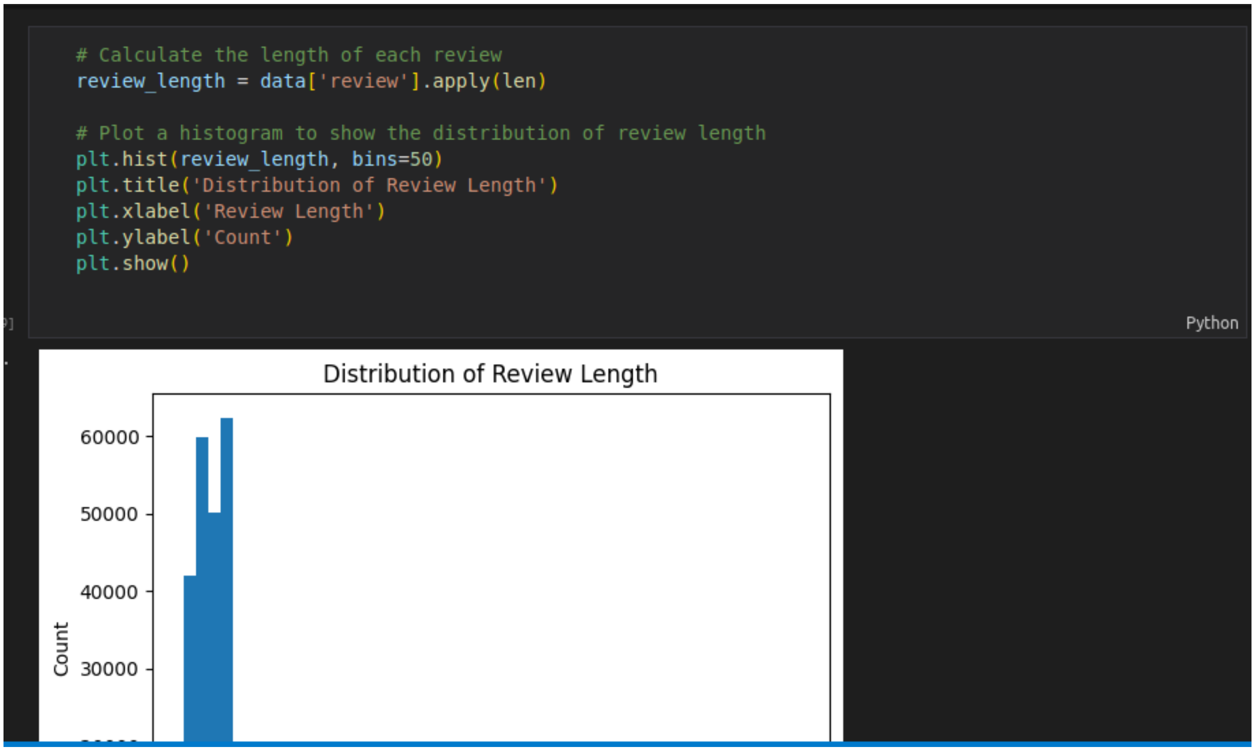
```python
data.head()
```

	uniqueID	condition	review	date	sentiment
0	206461	Left Ventricular Dysfunction	it has no side effect i take it in combinati...	20-May-12	1
1	95260	ADHD	my son is halfway through his fourth week of ...	27-Apr-10	1
2	92703	Birth Control	i used to take another oral contraceptive wh...	14-Dec-09	0
3	138000	Birth Control	this is my first time using any form of birth...	3-Nov-15	1
4	35696	Opiate Dependence	suboxone has completely turned my life around...	27-Nov-16	1

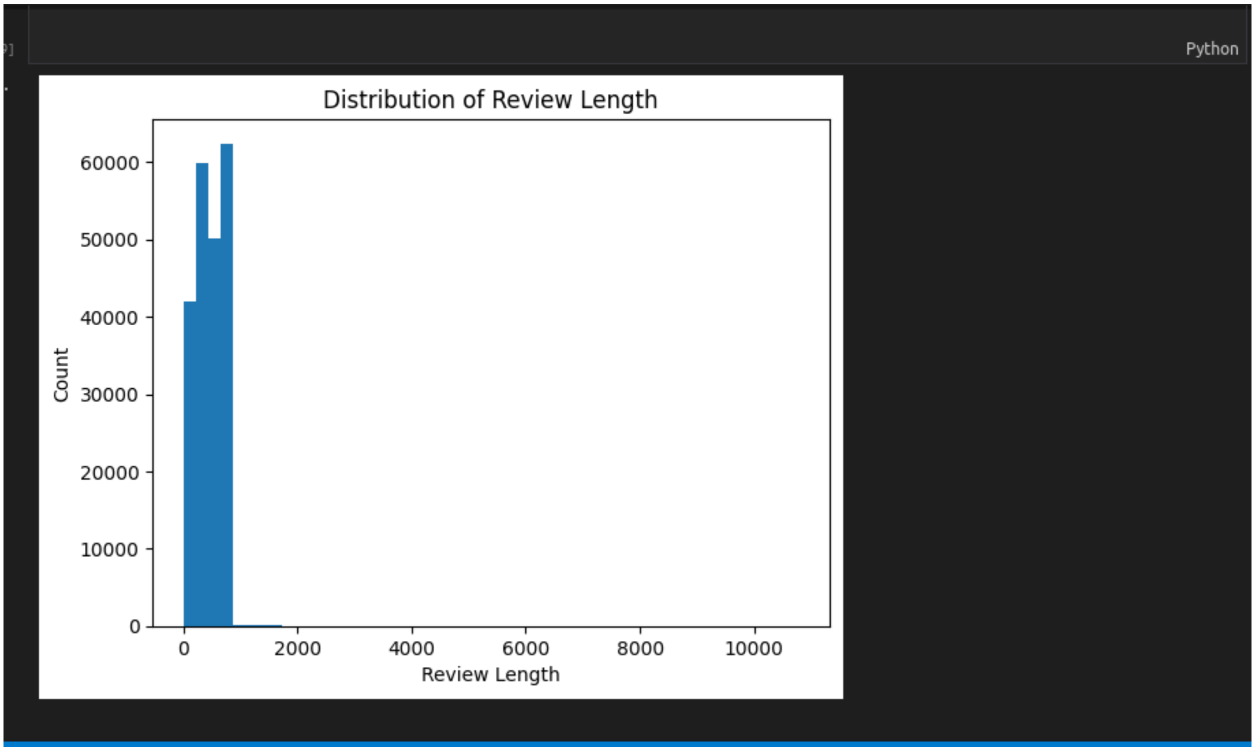
```python
import matplotlib.pyplot as plt

# Calculate the count of positive and negative reviews
sentiment_count = data['sentiment'].value_counts()

# Plot a bar graph to show the count of positive and negative reviews
plt.bar(['Negative', 'Positive'], [sentiment_count[0], sentiment_count[1]])
plt.title('Count of Positive and Negative Reviews')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```

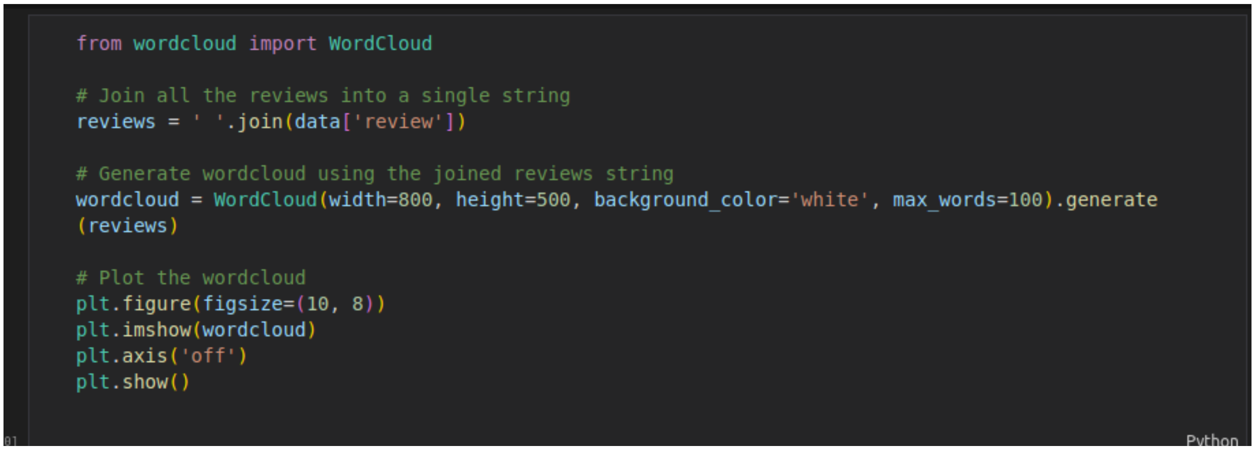


```python
plt.show()
```

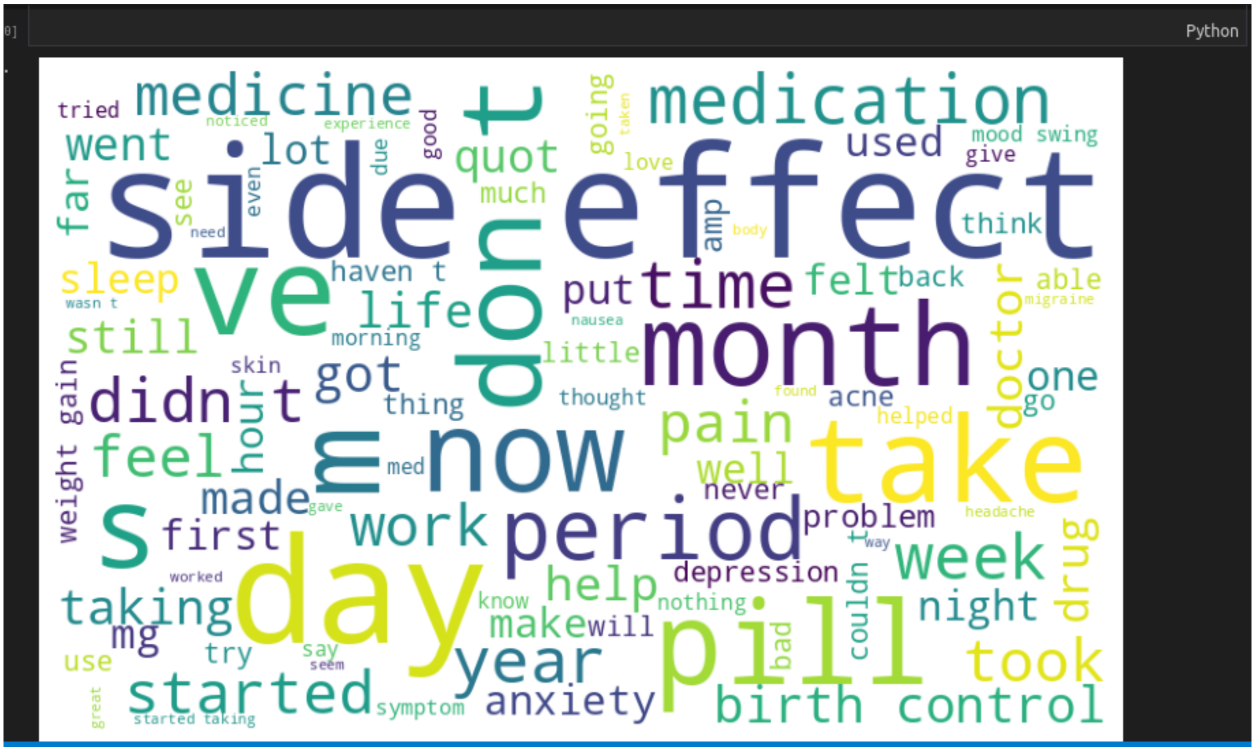


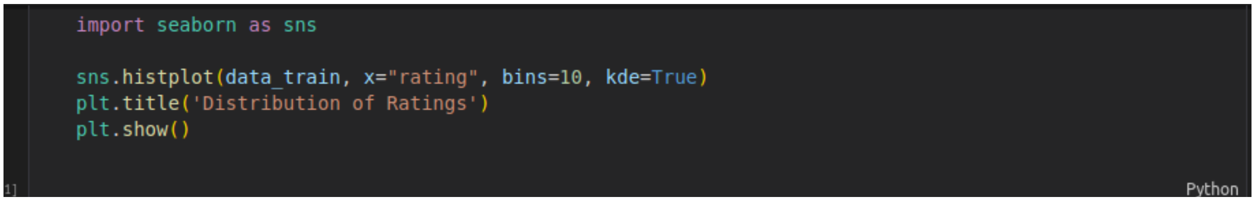
```python
# Calculate the length of each review
review_length = data['review'].apply(len)

# Plot a histogram to show the distribution of review length
plt.hist(review_length, bins=50)
plt.title('Distribution of Review Length')
plt.xlabel('Review Length')
plt.ylabel('Count')
plt.show()
```



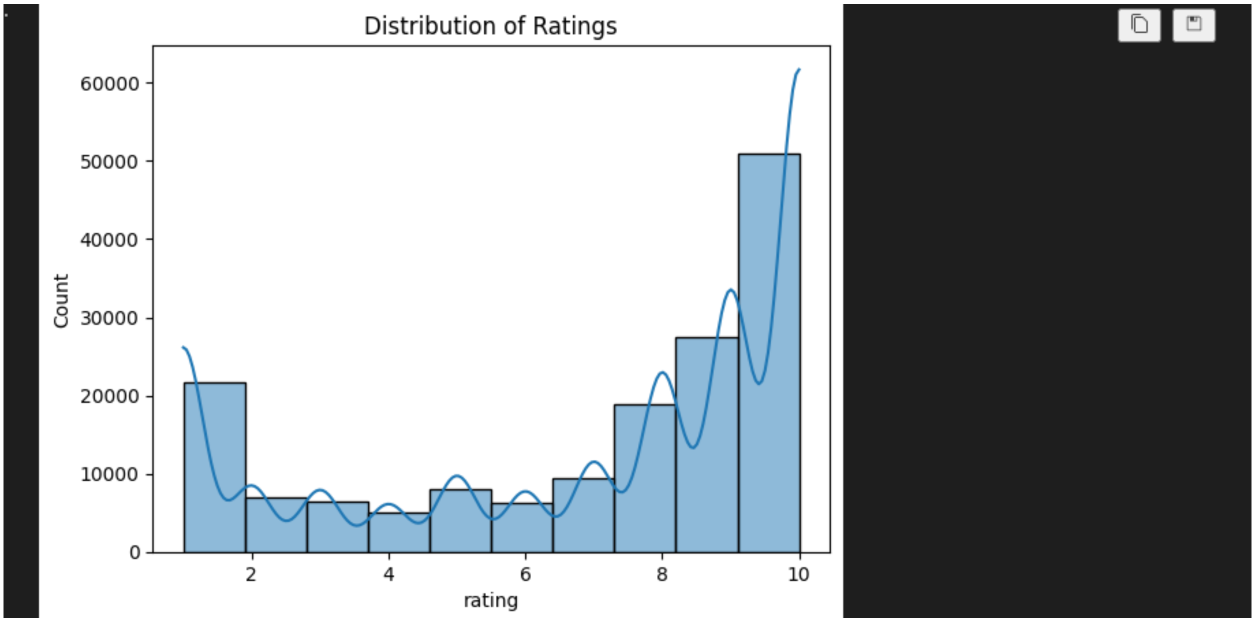


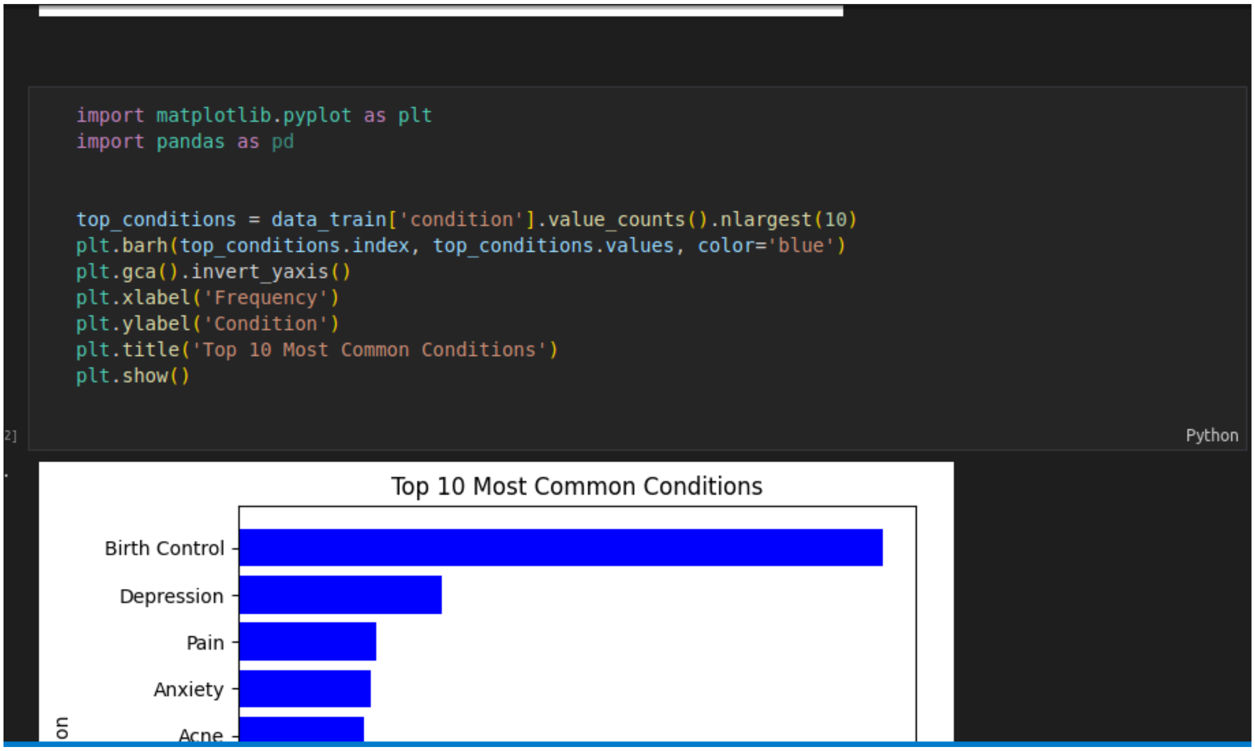
```python
from wordcloud import WordCloud

# Join all the reviews into a single string
reviews = ' '.join(data['review'])

# Generate wordcloud using the joined reviews string
wordcloud = WordCloud(width=800, height=500, background_color='white', max_words=100).generate
(reviews)

# Plot the wordcloud
plt.figure(figsize=(10, 8))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



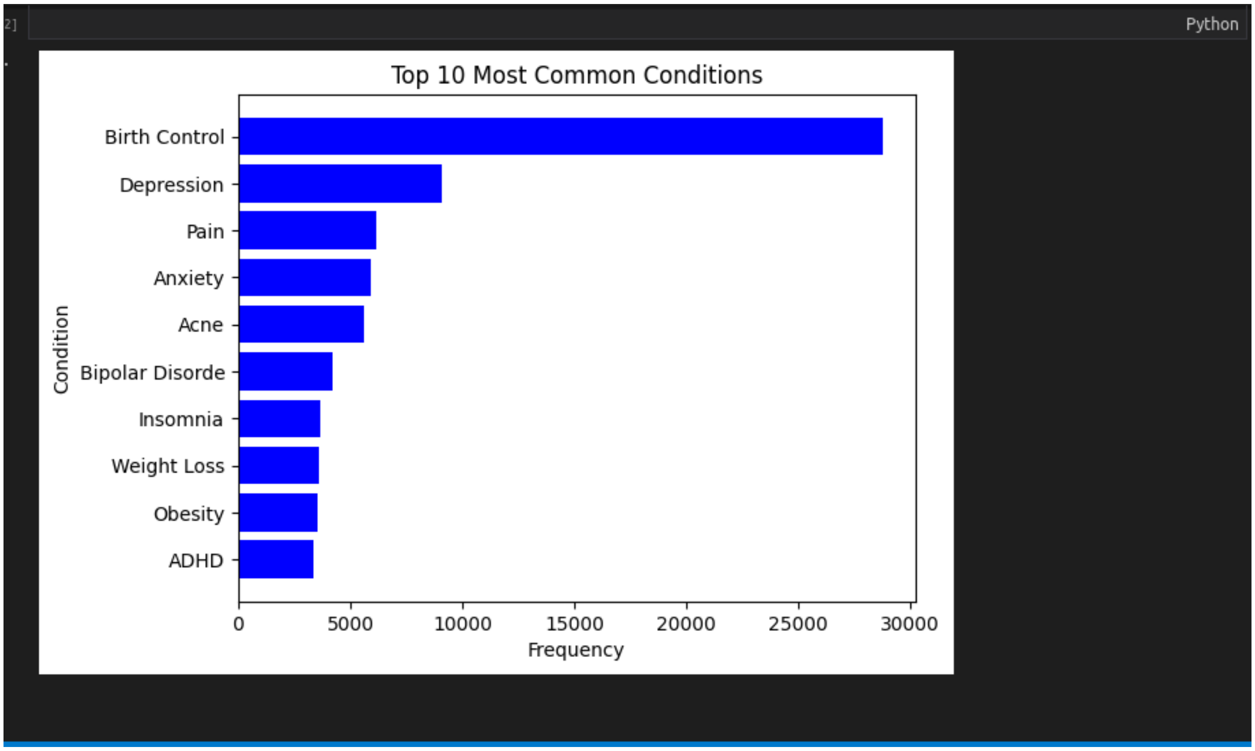


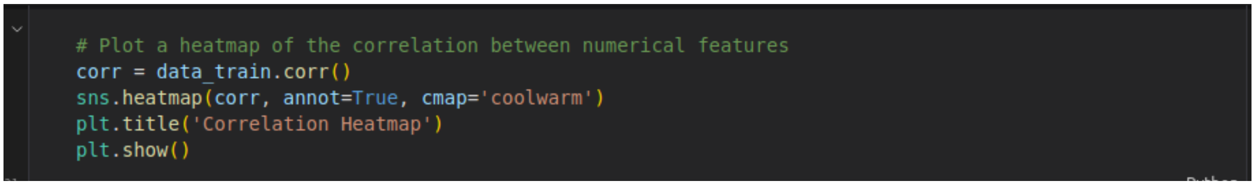
```python
import seaborn as sns

sns.histplot(data_train, x="rating", bins=10, kde=True)
plt.title('Distribution of Ratings')
plt.show()
```



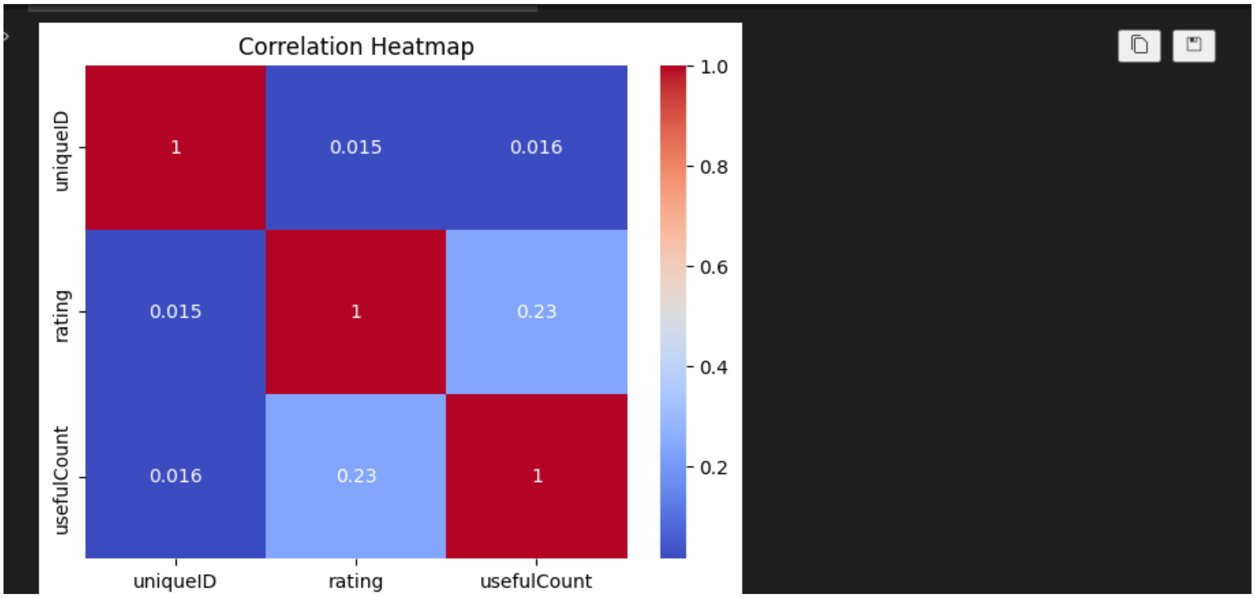


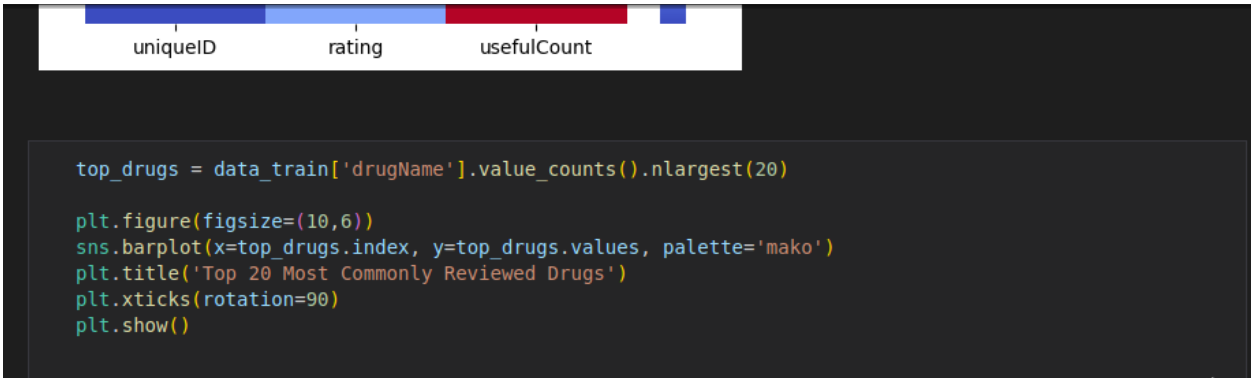
```
import matplotlib.pyplot as plt
import pandas as pd


top_conditions = data_train['condition'].value_counts().nlargest(10)
plt.barh(top_conditions.index, top_conditions.values, color='blue')
plt.gca().invert_yaxis()
plt.xlabel('Frequency')
plt.ylabel('Condition')
plt.title('Top 10 Most Common Conditions')
plt.show()
```



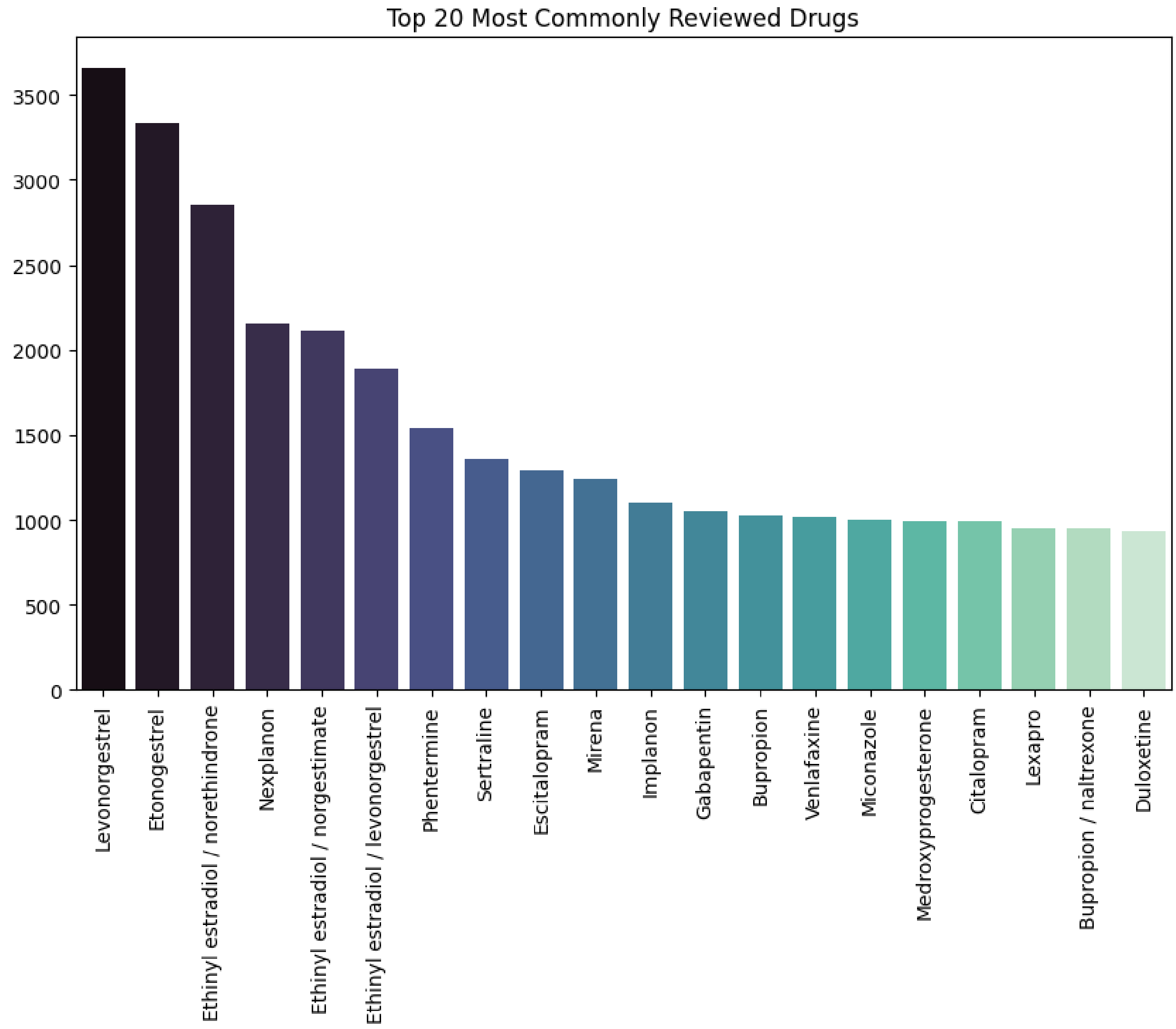


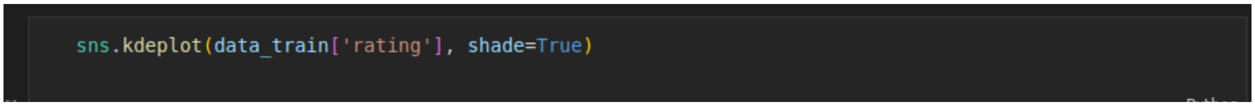
```
# Plot a heatmap of the correlation between numerical features
corr = data_train.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



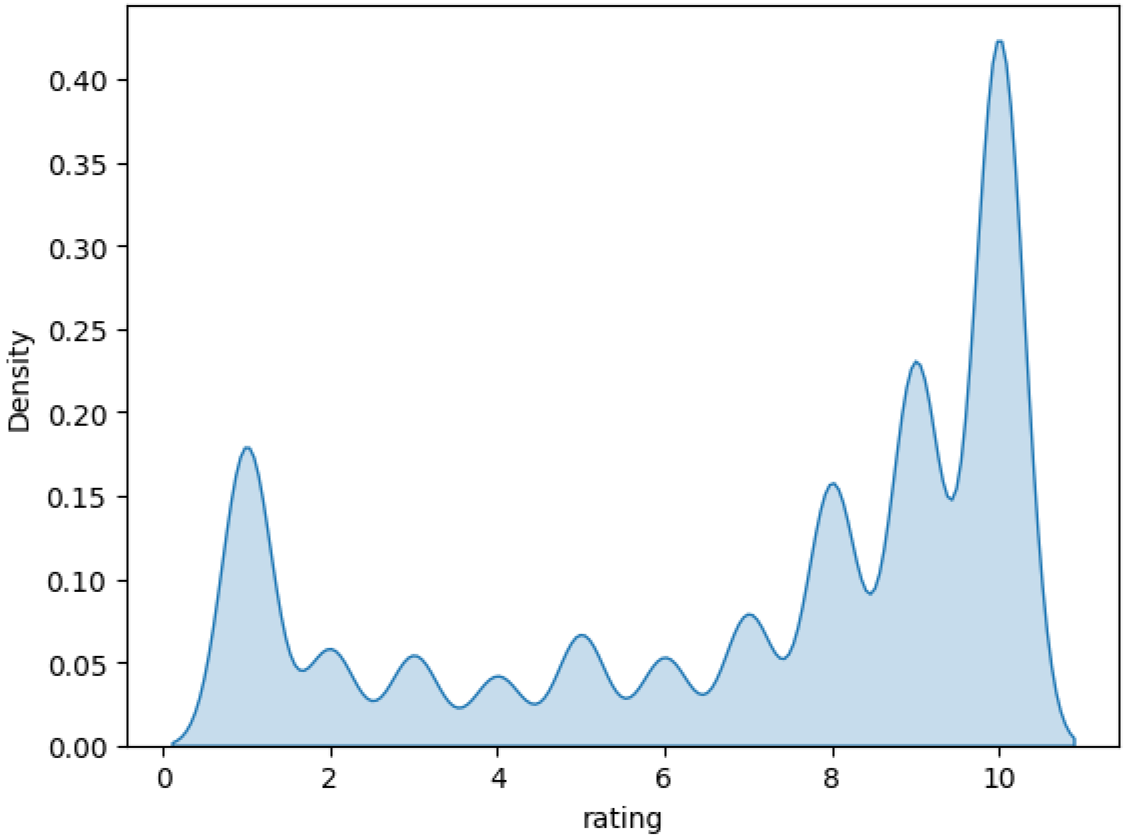


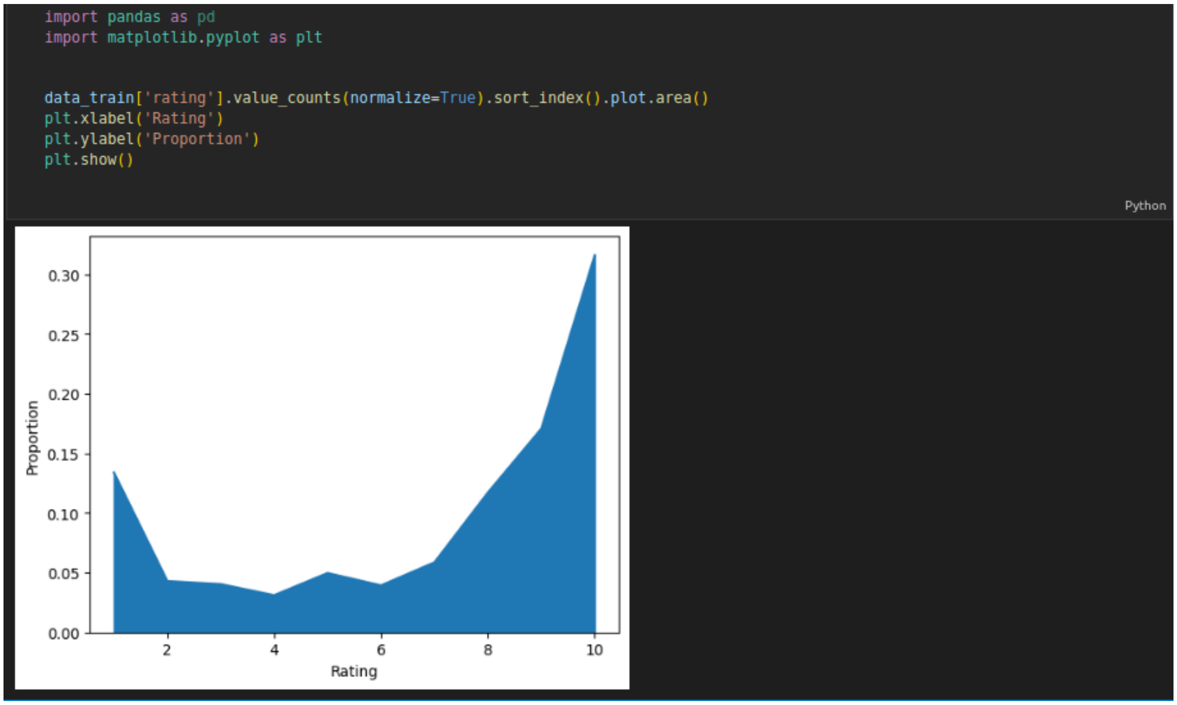
```
top_drugs = data_train['drugName'].value_counts().nlargest(20)

plt.figure(figsize=(10,6))
sns.barplot(x=top_drugs.index, y=top_drugs.values, palette='mako')
plt.title('Top 20 Most Commonly Reviewed Drugs')
plt.xticks(rotation=90)
plt.show()
```



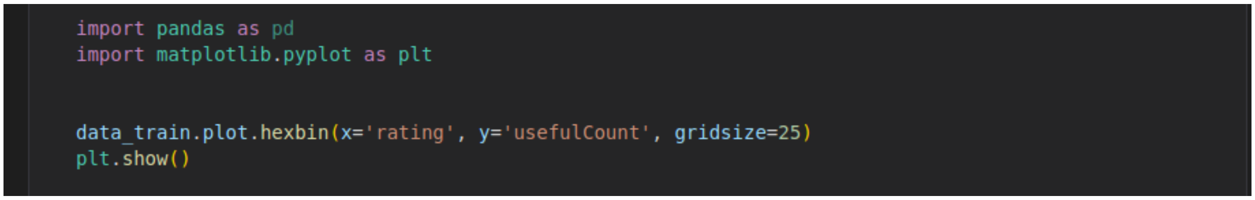


```
sns.kdeplot(data_train['rating'], shade=True)
```

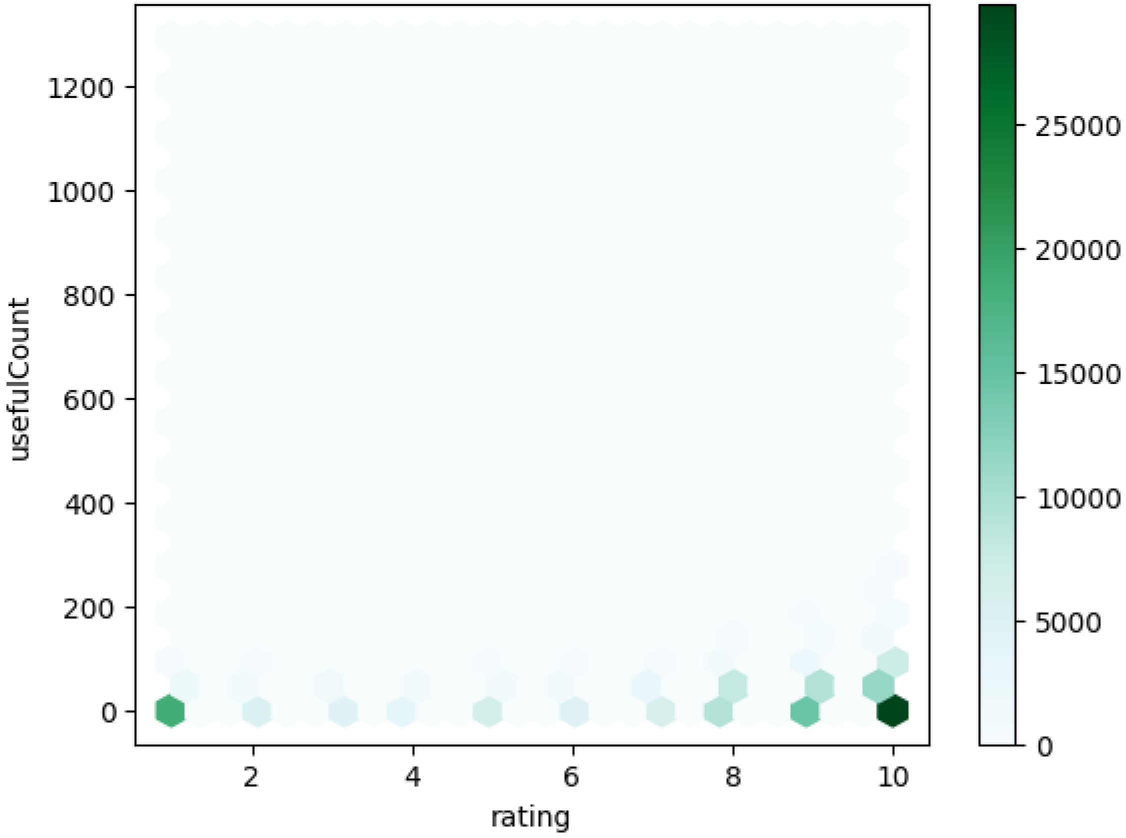


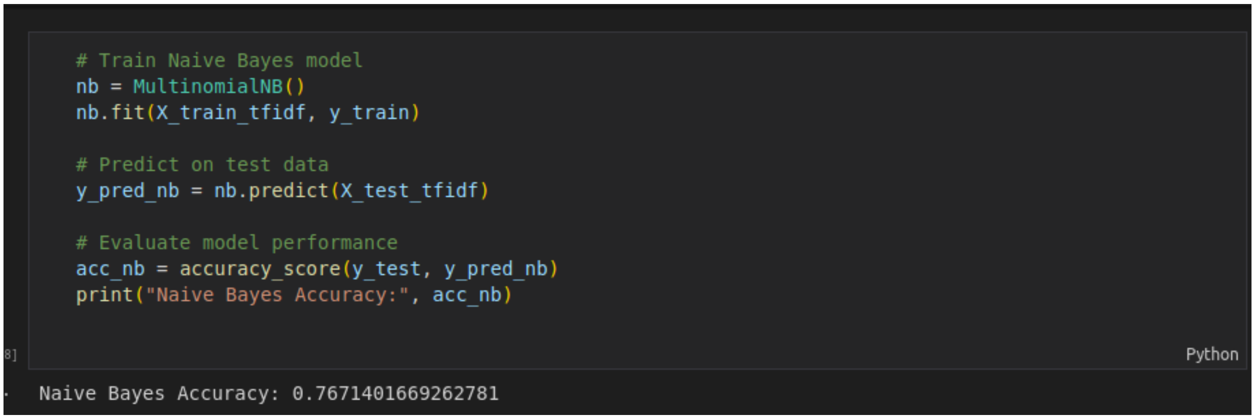


```
import pandas as pd
import matplotlib.pyplot as plt


data_train['rating'].value_counts(normalize=True).sort_index().plot.area()
plt.xlabel('Rating')
plt.ylabel('Proportion')
plt.show()
```



```python
import pandas as pd
import matplotlib.pyplot as plt


data_train.plot.hexbin(x='rating', y='usefulCount', gridsize=25)
plt.show()
```



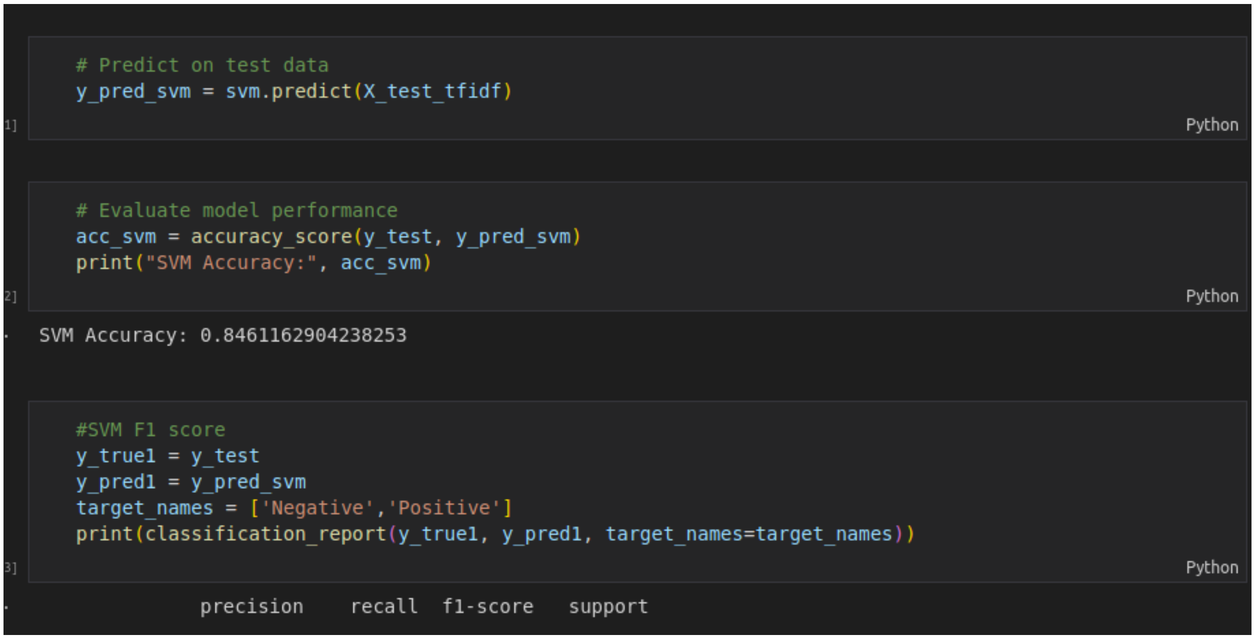


```python
# Train Naive Bayes model
nb = MultinomialNB()
nb.fit(X_train_tfidf, y_train)

# Predict on test data
y_pred_nb = nb.predict(X_test_tfidf)

# Evaluate model performance
acc_nb = accuracy_score(y_test, y_pred_nb)
print("Naive Bayes Accuracy:", acc_nb)
```

```
Naive Bayes Accuracy: 0.7671401669262781
```

Model Training Stats (Naive-Bayes):

```
                 precision    recall   f1-score    support


     Negative       0.87       0.26       0.41      12940
     Positive       0.76       0.98       0.86      30073
     accuracy                             0.77      43013
    macro avg       0.82       0.62       0.63      43013
 weighted avg       0.79       0.77       0.72      43013
```



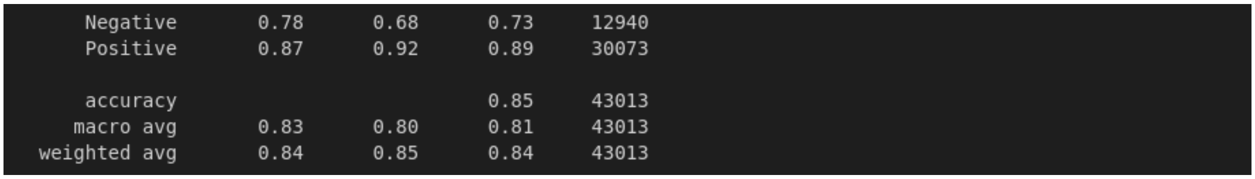
```python
# Predict on test data
y_pred_svm = svm.predict(X_test_tfidf)
```

```python
# Evaluate model performance
acc_svm = accuracy_score(y_test, y_pred_svm)
print("SVM Accuracy:", acc_svm)
```

```
SVM Accuracy: 0.8461162904238253
```

```python
#SVM F1 score
y_true1 = y_test
y_pred1 = y_pred_svm
target_names = ['Negative','Positive']
print(classification_report(y_true1, y_pred1, target_names=target_names))
```

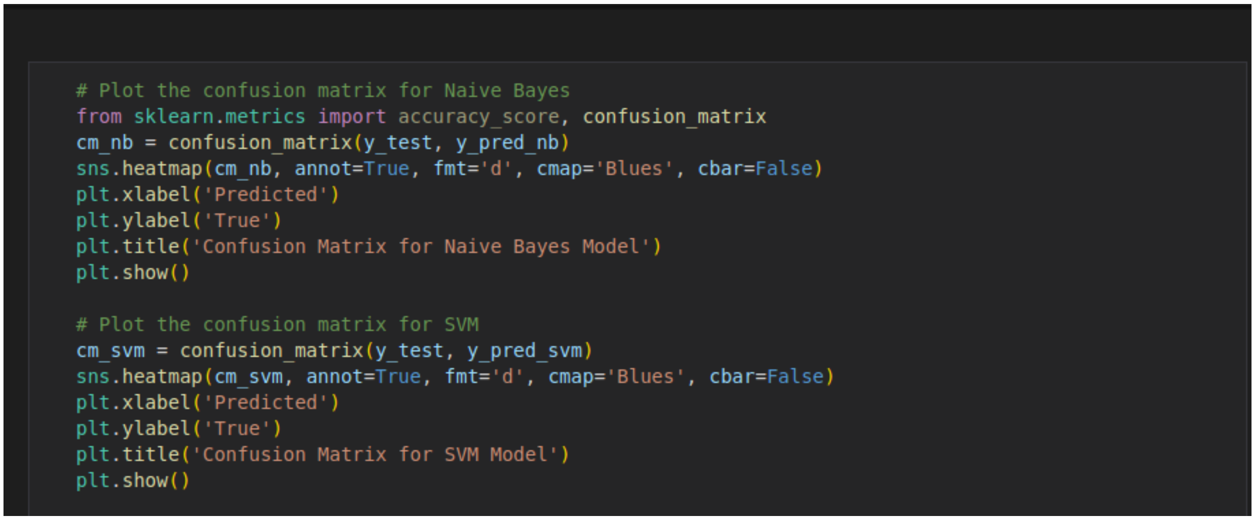
```
              precision    recall  f1-score   support
```



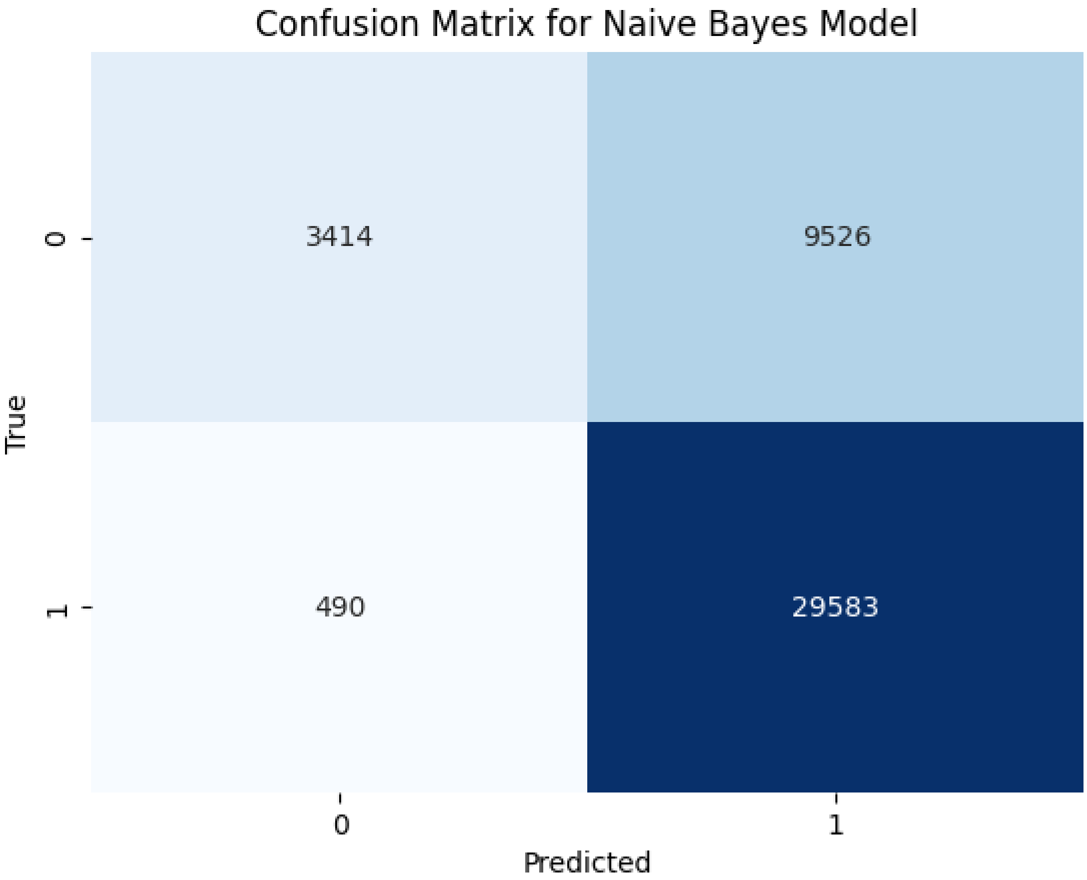
```
    Negative       0.78      0.68      0.73     12940
    Positive       0.87      0.92      0.89     30073

    accuracy                           0.85     43013
   macro avg       0.83      0.80      0.81     43013
weighted avg       0.84      0.85      0.84     43013
```

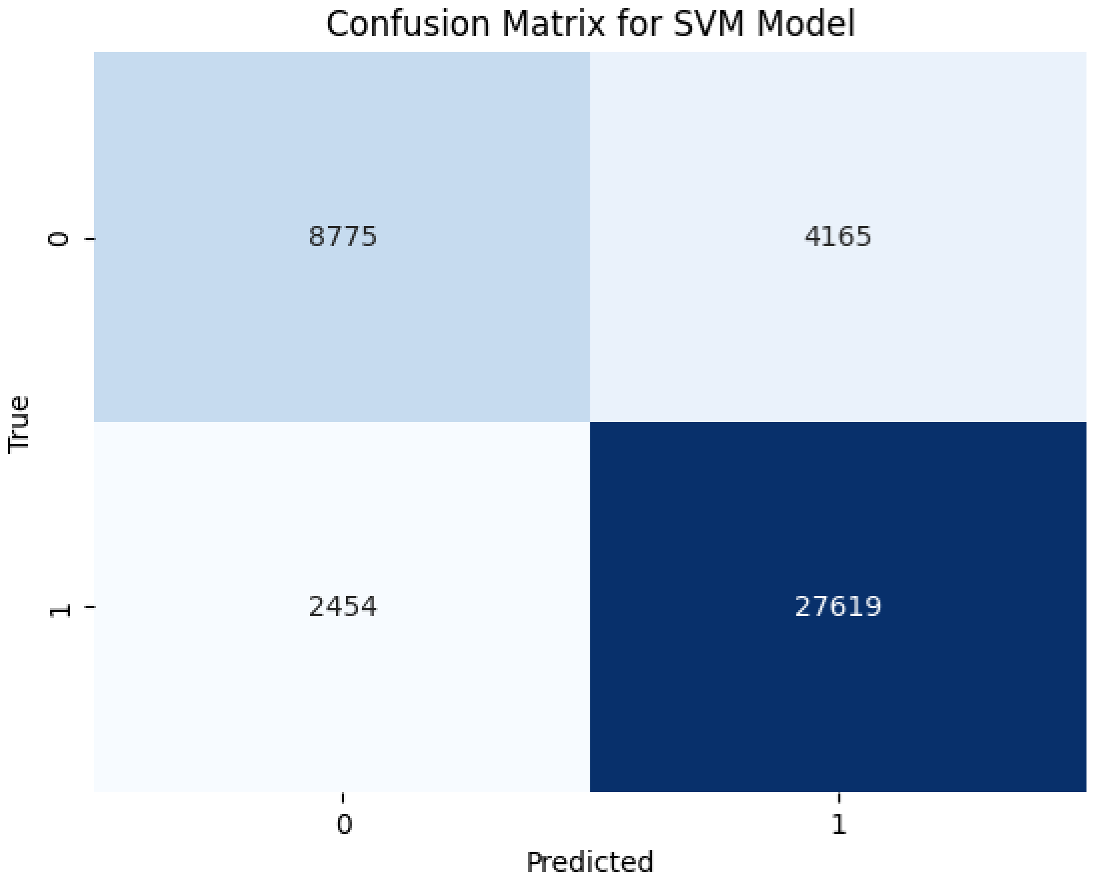
Model Training Stats (SVM):

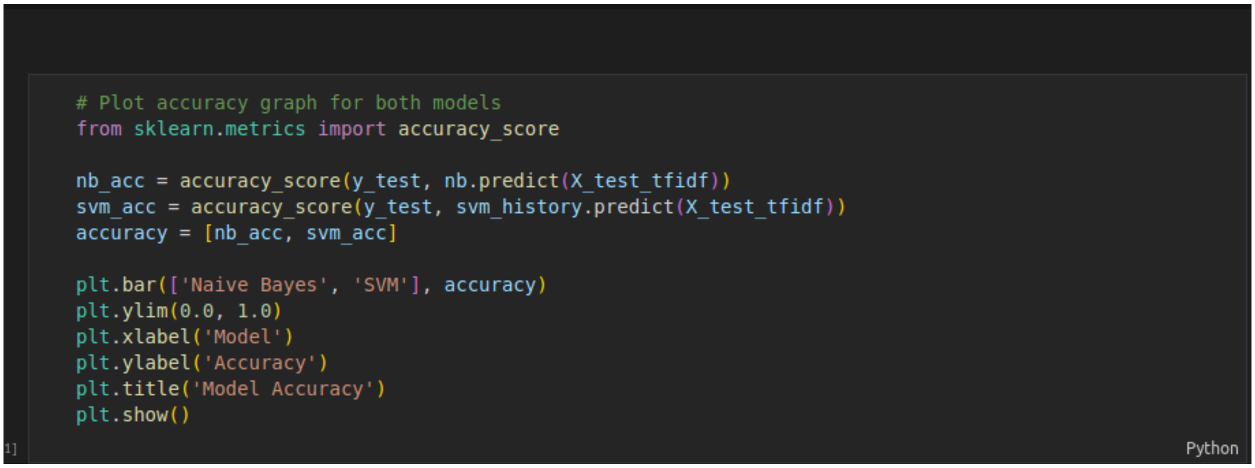
```
                 precision    recall   f1-score    support


     Negative       0.78       0.68       0.73      12940
     Positive       0.87       0.92       0.89      30073
     accuracy                             0.85      43013
    macro avg       0.83       0.80       0.81      43013
 weighted avg       0.84       0.85       0.84      43013
```



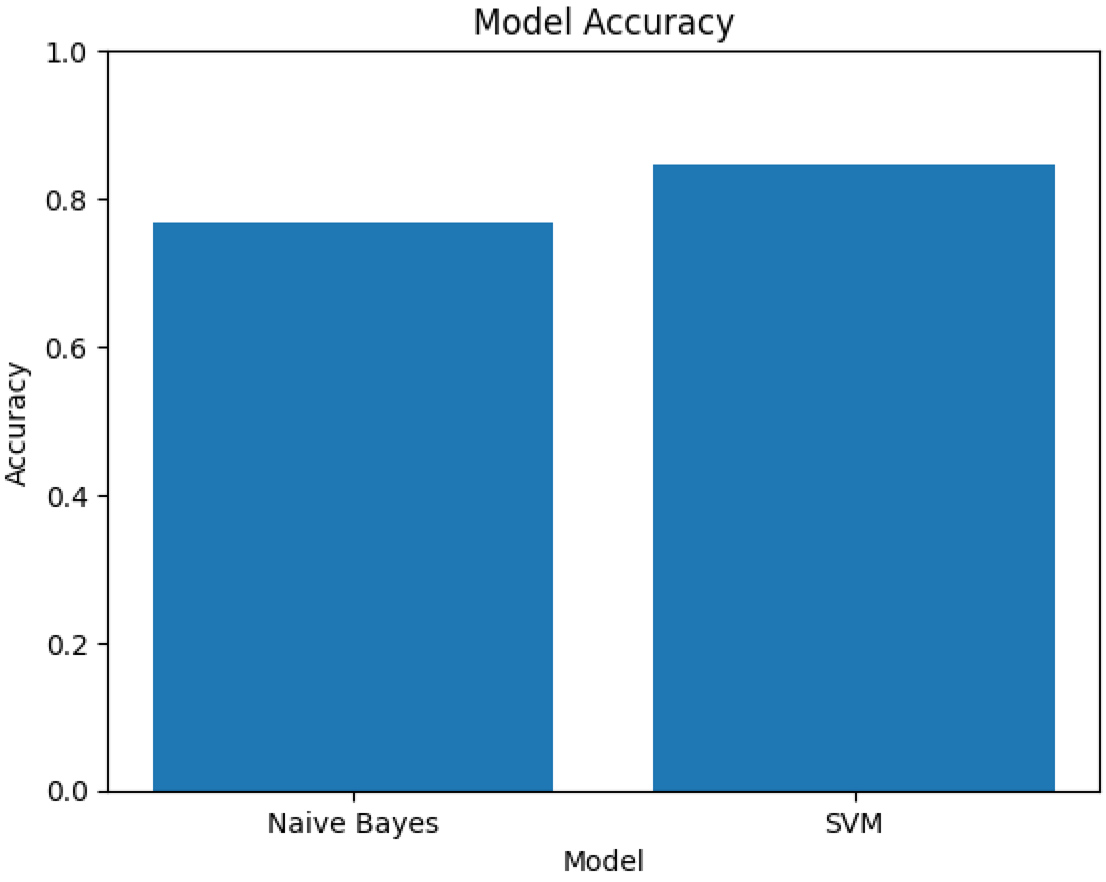
```
# Plot the confusion matrix for Naive Bayes
from sklearn.metrics import accuracy_score, confusion_matrix
cm_nb = confusion_matrix(y_test, y_pred_nb)
sns.heatmap(cm_nb, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix for Naive Bayes Model')
plt.show()

# Plot the confusion matrix for SVM
cm_svm = confusion_matrix(y_test, y_pred_svm)
sns.heatmap(cm_svm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix for SVM Model')
plt.show()
```

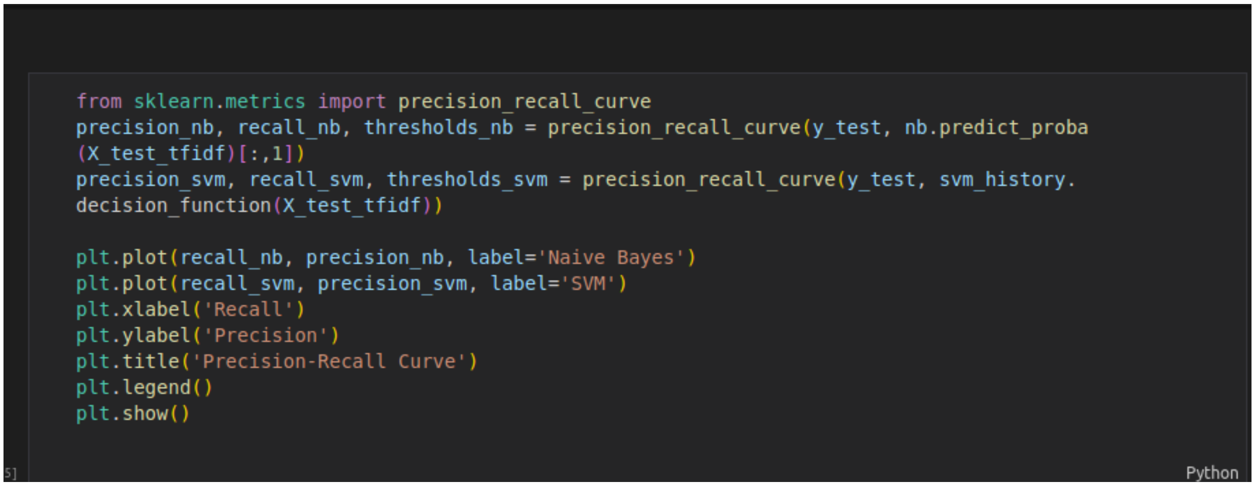



Confusion Matrix for SVM Model
True
0
1
8775
4165
2454
27619
0
1
Predicted

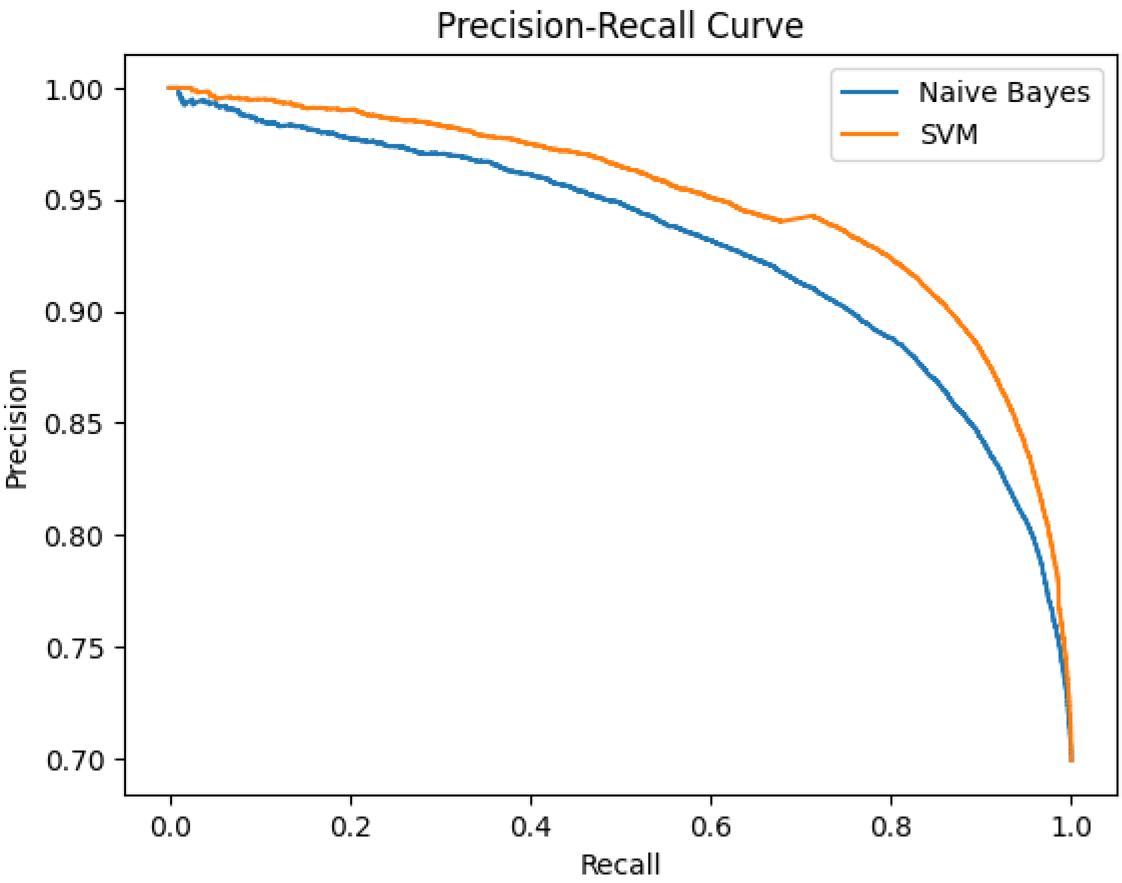


```python
# Plot accuracy graph for both models
from sklearn.metrics import accuracy_score

nb_acc = accuracy_score(y_test, nb.predict(X_test_tfidf))
svm_acc = accuracy_score(y_test, svm_history.predict(X_test_tfidf))
accuracy = [nb_acc, svm_acc]

plt.bar(['Naive Bayes', 'SVM'], accuracy)
plt.ylim(0.0, 1.0)
plt.xlabel('Model')
plt.ylabel('Accuracy')
plt.title('Model Accuracy')
plt.show()
```



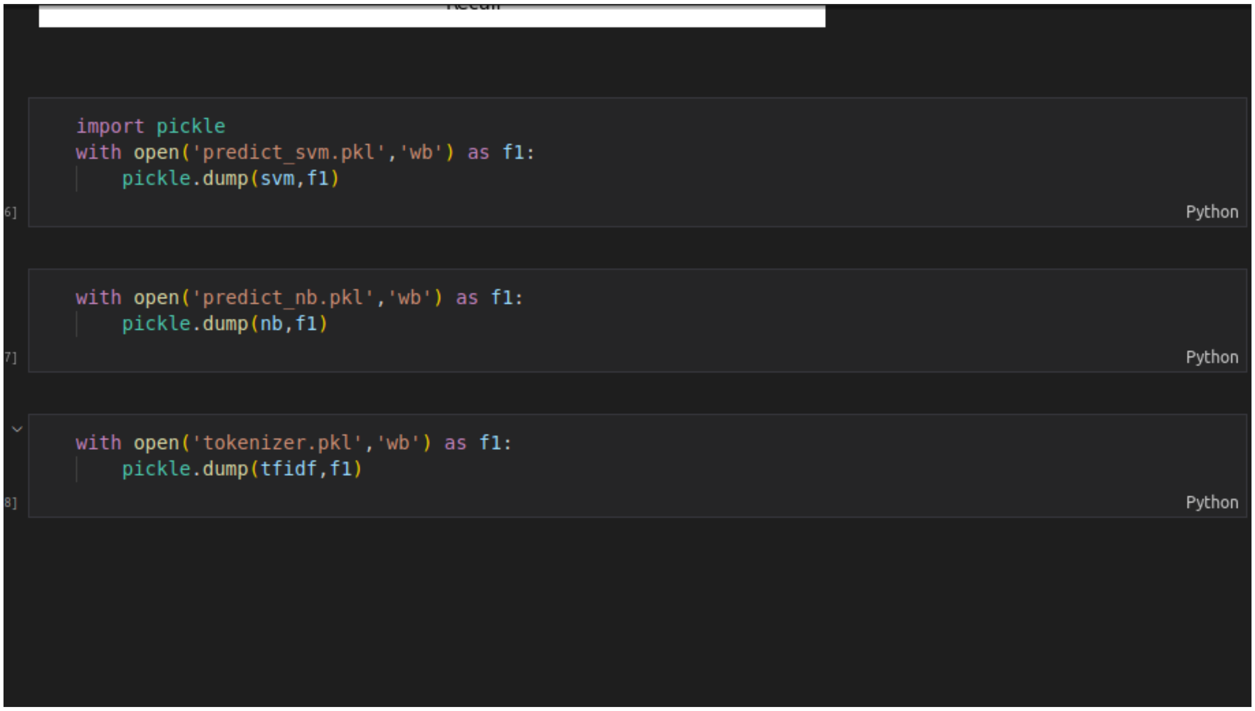


```
from sklearn.metrics import precision_recall_curve
precision_nb, recall_nb, thresholds_nb = precision_recall_curve(y_test, nb.predict_proba
(X_test_tfidf)[:,1])
precision_svm, recall_svm, thresholds_svm = precision_recall_curve(y_test, svm_history.
decision_function(X_test_tfidf))

plt.plot(recall_nb, precision_nb, label='Naive Bayes')
plt.plot(recall_svm, precision_svm, label='SVM')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.legend()
plt.show()
```



Prediction System:

In this module, we are saving both the trained models and vectorizer files as pickle files in binary form to use the trained model in future.



```python
import pickle
with open('predict_svm.pkl','wb') as f1:
    pickle.dump(svm,f1)
```

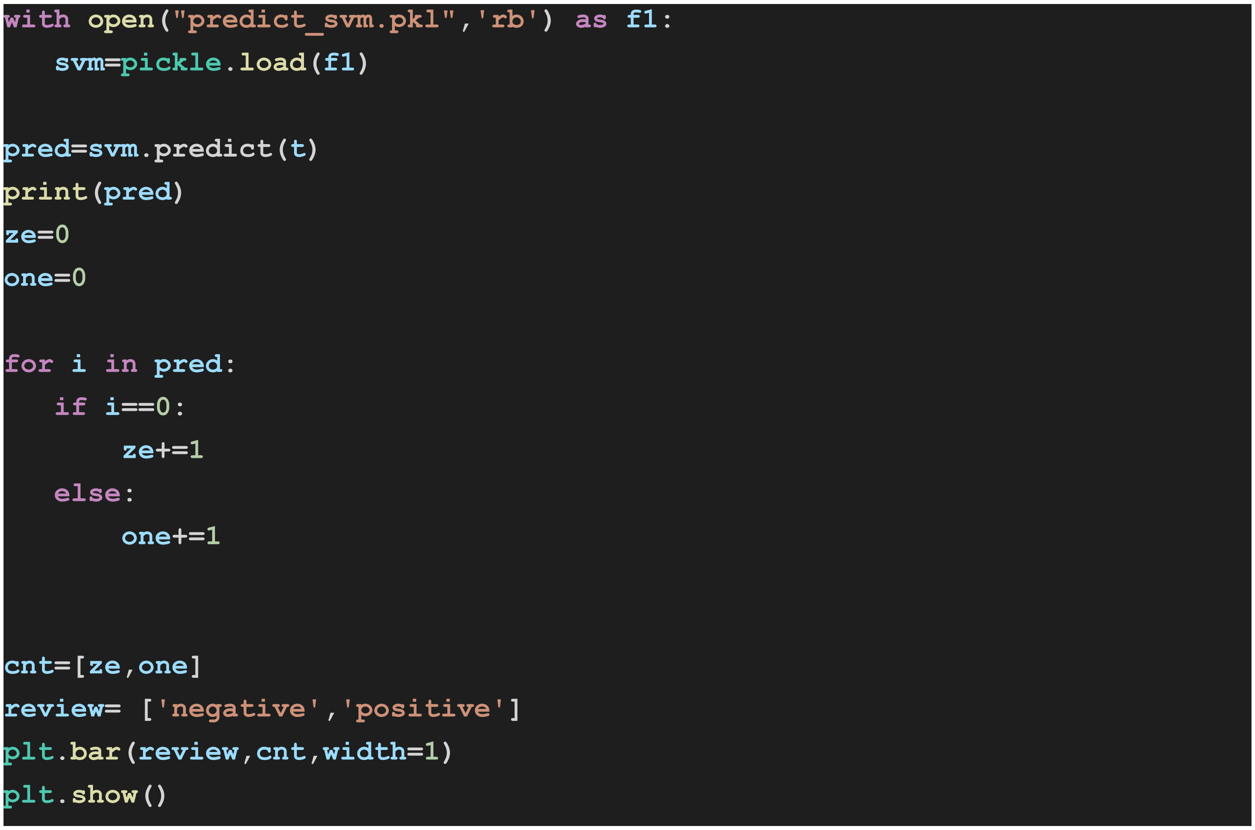
```python
with open('predict_nb.pkl','wb') as f1:
    pickle.dump(nb,f1)
```

```python
with open('tokenizer.pkl','wb') as f1:
    pickle.dump(tfidf,f1)
```

Model Implementation

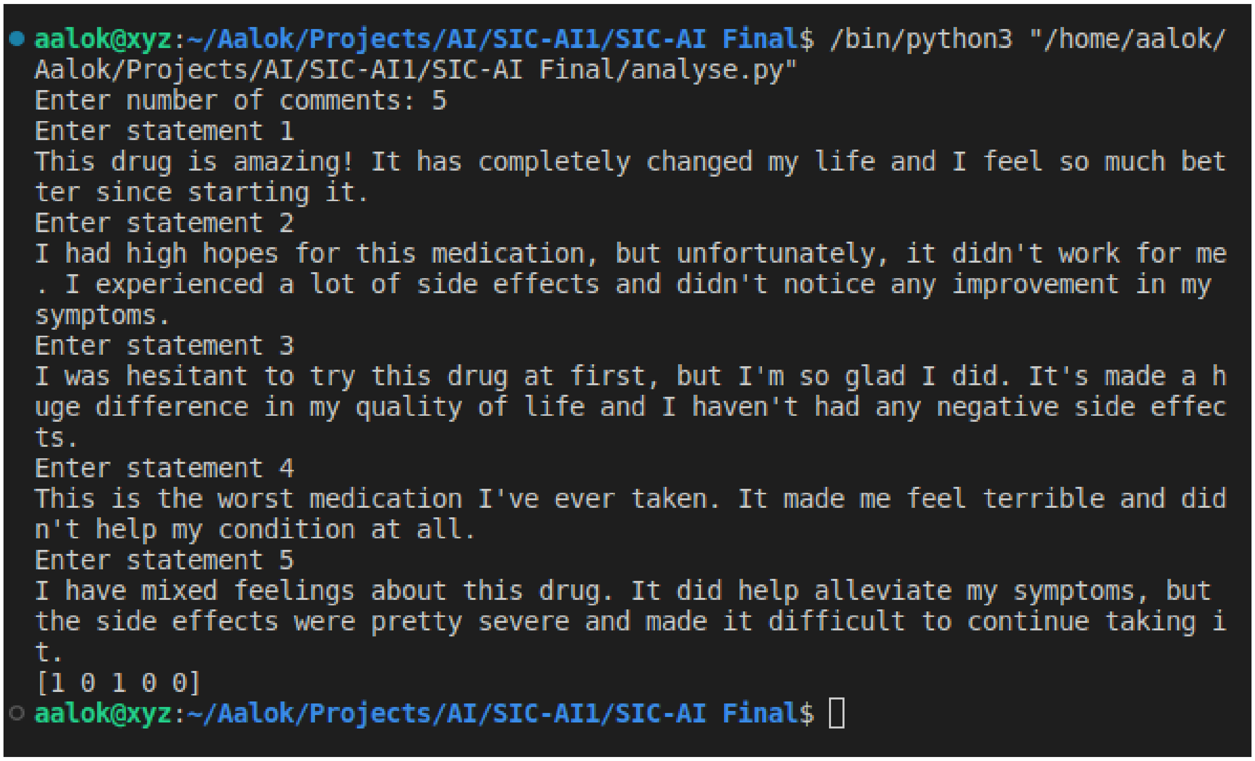
analyse.py

```python
import pickle
import re
import matplotlib.pyplot as plt


def clean(x):
    #x=re.sub(r'\W',' ',x)
    #x = re.sub(r'[^a-zA-Z]',' ',x)
    x = re.sub("wouldn\'t",'would not',x)
    x = re.sub("they \ 've",'they have',x)

    #to remove html tags
    x = re.sub(r'<.*?>', '', x)

    #to remove everything except alpha
    x = re.sub(r'[^a-zA-Z]',' ',x)

    x = re.sub(r'\s+',' ',x)           #remove extra space's
    return x.lower()

n=int(input("Enter number of comments: "))
inp=[]
for i in range(n):
    inp.append(input("Enter statement "+str(i+1)+"\n"))

f=[]
for i in inp:
    s=clean(i)
    f.append(s)

with open("tokenizer.pkl",'rb') as f1:
    cv1=pickle.load(f1)
t=cv1.transform(f).toarray()
```



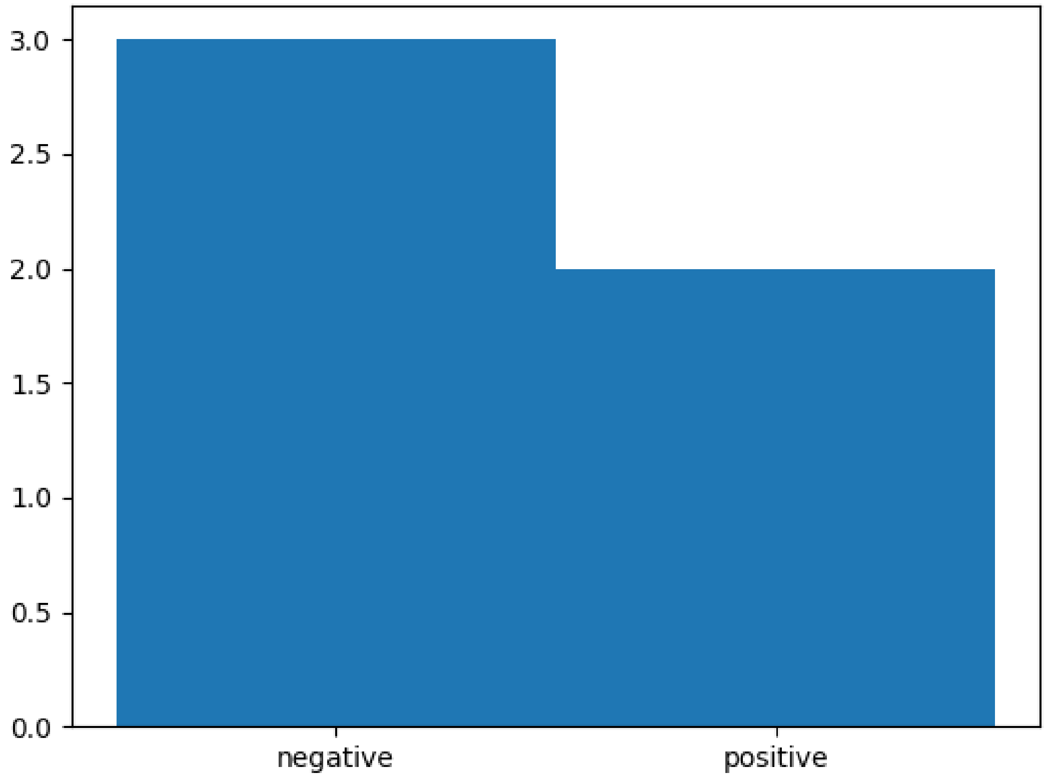
```python
with open("predict_svm.pkl",'rb') as f1:
   svm=pickle.load(f1)

pred=svm.predict(t)
print(pred)
ze=0
one=0

for i in pred:
   if i==0:
       ze+=1
   else:
       one+=1


cnt=[ze,one]
review= ['negative','positive']
plt.bar(review,cnt,width=1)
plt.show()
```

Test Output:



```
● aalok@xyz:~/Aalok/Projects/AI/SIC-AI1/SIC-AI Final$ /bin/python3 "/home/aalok/
  Aalok/Projects/AI/SIC-AI1/SIC-AI Final/analyse.py"
  Enter number of comments: 5
  Enter statement 1
  This drug is amazing! It has completely changed my life and I feel so much bet
  ter since starting it.
  Enter statement 2
  I had high hopes for this medication, but unfortunately, it didn't work for me
  . I experienced a lot of side effects and didn't notice any improvement in my
  symptoms.
  Enter statement 3
  I was hesitant to try this drug at first, but I'm so glad I did. It's made a h
  uge difference in my quality of life and I haven't had any negative side effec
  ts.
  Enter statement 4
  This is the worst medication I've ever taken. It made me feel terrible and did
  n't help my condition at all.
  Enter statement 5
  I have mixed feelings about this drug. It did help alleviate my symptoms, but
  the side effects were pretty severe and made it difficult to continue taking i
  t.
  [1 0 1 0 0]
○ aalok@xyz:~/Aalok/Projects/AI/SIC-AI1/SIC-AI Final$ ▯
```

Test output in bar graph:



Conclusion

The sentiment analysis on drug reviews project aimed to build a model to classify reviews of drugs as positive or negative based on their textual content. The dataset was preprocessed, cleansed, and split into training and testing sets. Two machine learning algorithms, Support Vector Machines (SVM) and Naive Bayes, were trained on the dataset and evaluated based on their accuracy, precision, recall, and F1-score.

The **SVM model** achieved an accuracy of **0.85**. The **Naive Bayes** model achieved an accuracy of **0.77**. Therefore, the **SVM model outperformed the Naive Bayes model in terms of accuracy**.

In conclusion, the sentiment analysis on drug reviews project successfully built a model that can accurately classify reviews of drugs as positive or negative. The SVM model performed better than the Naive Bayes model and can be used to classify drug reviews for various purposes such as monitoring drug safety, improving drug efficacy, and enhancing patient care.