University of New Haven

TAGLIATELA COLLEGE OF ENGINEERING

Electrical & Computer Engineering and Computer Science

TECHNICAL REPORT

CONTENTS

YOUTUBE COMMENT SENTIMENT ANALYSIS

Team Members:

Pallavi Bongu
Sai Chaitanya Kolli
Keerthi Balabhadruni

Questions?

Executive Summary

The YouTube Sentiment Analysis project intricately combines Natural Language Processing (NLP) and machine learning to unravel the emotional nuances within YouTube comments. Its primary goal is to redefine how user sentiments are understood and applied in the dynamic realm of social media. By leveraging advanced NLP techniques and machine learning models, the project decodes nuanced sentiments into positive, negative, or neutral categories, offering profound insights into user interactions on the global video-sharing platform.

Project Objectives:

The core aim is clear — to decode the nuanced sentiments expressed by users in YouTube comments. By harnessing the capabilities of advanced NLP techniques and machine learning models, the project categorizes sentiments into positive, negative, or neutral categories, providing profound insights into user interactions on this global video-sharing platform.

Technical Report

The YouTube Sentiment Analysis project is an extensive initiative aimed at deciphering sentiments within YouTube comments using advanced Natural Language Processing (NLP) and machine learning models. The introductory section underscores the project's purpose, emphasizing the importance of sentiment analysis in decoding user reactions within the dynamic landscape of YouTube comments. A brief review of existing research provides context, offering insights into sentiment analysis methods, their relevance in social media, and notable findings from prior YouTube sentiment analysis studies.

The methodology section outlines technical steps, including data collection from diverse sources, rigorous preprocessing to address challenges, and the application of advanced NLP techniques such as tokenization, stop words removal, lemmatization, and stemming. The project utilizes five distinct machine learning models—Naive Bayes, Logistic Regression, SVM, Decision Tree, and Random Forest—demonstrating model diversity. Both Count Vectorizer and TfidfVectorizer are employed for effective and nuanced vectorization.

Empirical findings in the results section highlight the Support Vector Machine (SVM) as a standout performer. Comparative analysis explores the strengths and weaknesses of each model, offering a detailed technical assessment. The discussion section delves into the broader implications of the findings in the context of sentiment analysis on social media platforms, acknowledging challenges and limitations and paving the way for future research directions and improvements.

In conclusion, the project's unique contributions, including the amalgamation of various machine learning models and meticulous NLP techniques, are emphasized. The report concludes with acknowledgments of contributions and references, providing a comprehensive technical overview of the project's foundations and resources.

Highlights of Project

The YouTube Sentiment Analysis project is a comprehensive exploration of sentiment in YouTube comments, employing advanced NLP and diverse machine learning models. It begins with the meticulous collection of pre-labeled comments from popular sources, followed by rigorous preprocessing to refine the dataset. Versatile NLP techniques, including tokenization, stop words removal, lemmatization, and stemming, are applied to extract meaningful insights from the textual data.

The project stands out in model development, training five distinct machine learning models—Naive Bayes, Logistic Regression, SVM, Decision Tree, and Random Forest. Both CountVectorizer and TfidfVectorizer are utilized for vectorization, ensuring a nuanced exploration of sentiment.

Results highlight SVM as a standout performer, leading to a comparative analysis of model strengths and weaknesses. Real-world applications demonstrate the model's adaptability to new data, providing users with a practical sentiment analysis tool.

Discussion and future directions address broader implications on social media platforms, identifying challenges and paving the way for future research. Contributions include an amalgamation of machine learning models and meticulous NLP techniques, creating a versatile toolkit for understanding sentiments
YouTube comments.

Abstract

The abstract provides a concise overview of the YouTube Sentiment Analysis project, emphasizing its technical underpinnings. Employing advanced Natural Language Processing (NLP) and machine learning methodologies, the project is dedicated to unraveling the intricate tapestry of sentiments embedded within YouTube comments. The abstract previews the project's approach, from meticulous data collection and preprocessing to the training of diverse machine learning models, including Naive Bayes, Logistic Regression, SVM, Decision Tree, and Random Forest.

Central to the project is the exploration of real-world applications, with a keen focus on predicting sentiments for user-specified YouTube videos. The abstract underscores the significance of sentiment analysis in the dynamic realm of social media, where understanding user reactions holds practical implications for content creators, marketers, and platform administrators.

Overall, the abstract serves as a technical snapshot, providing a glimpse into the project's sophisticated methodologies, machine learning model diversity, and its commitment to addressing the challenges of sentiment analysis within the YouTube ecosystem.

Youtube Link: https://www.youtube.com/watch?v=sVcI7s_5plc

Introductory Section

Github Link:

The YouTube Sentiment Analysis project embarks on decoding the sentiments expressed in the vast sea of YouTube comments, offering an insightful exploration into the emotions concealed within this digital tapestry. For those new to sentiment analysis, this introductory section provides a gentle initiation into the significance of understanding user sentiments on one of the world's largest video-sharing platforms.

In the dynamic realm of social media, where digital interactions shape narratives, deciphering the sentiments expressed by users becomes crucial. This project addresses the fundamental question of how we can effectively analyze and interpret sentiments within YouTube comments. Through a blend of machine learning models and natural language processing (NLP) techniques, the aim is to unveil the emotional tone, be it positive, negative, or neutral, hidden within the textual fabric of comments.

This journey holds practical implications for content creators, marketers, and platform administrators. Content creators can refine strategies to better resonate with their audience, marketers can gain insights for campaign optimization, and platform administrators can gauge community sentiment for a more engaging user experience.

From the collection of pre-labeled comments to the application of NLP techniques and the training of machine learning models, each step in this exploration is a progression toward unraveling the sentiments that underpin digital interactions on YouTube. As we delve into the intricacies of sentiment analysis, the real-world applications underscore the project's relevance in the ever-evolving landscape of digital communication.

Review of available research

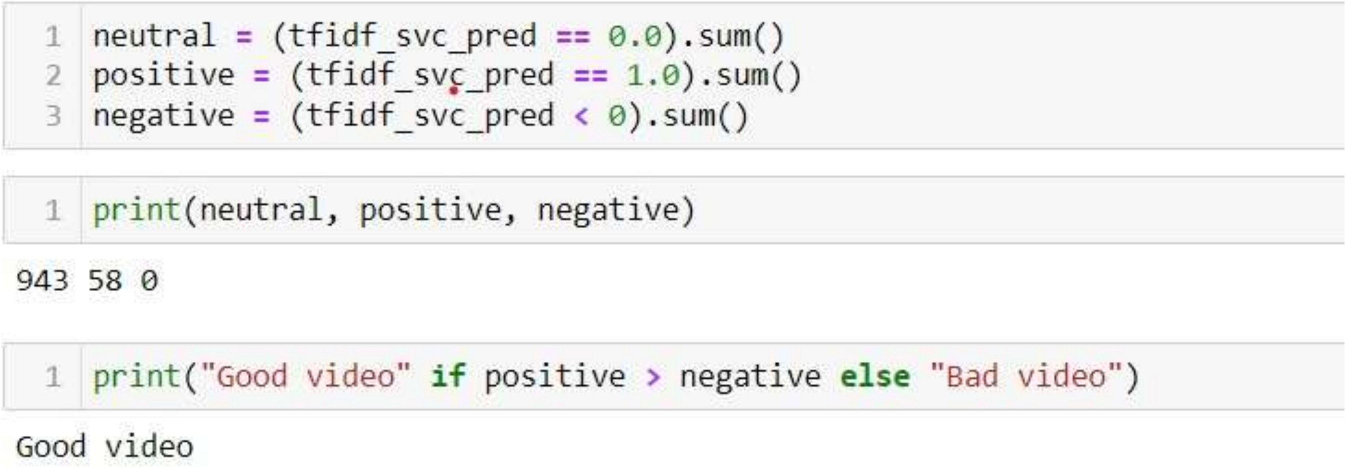
Sentiment analysis, a crucial aspect of natural language processing, has garnered significant attention in recent research. In the realm of social media, understanding sentiments expressed by users provides valuable insights into audience perception. Previous studies in YouTube sentiment analysis have explored various methods to decipher the emotional tone within user comments. The importanceof sentiment analysis in social media lies in its ability to gauge user reactions, preferences, and trends. Notable findings suggest that sentiment analysis can be pivotal for content creators, marketers, and platform administrators in shaping strategies and enhancing user engagement. As this project delves into YouTube sentiment analysis, it builds upon existing research by incorporating diverse machine learning models and meticulous NLP techniques.

Methodology

The methodology section outlines the systematic approach undertaken in the project, starting with data collection. Pre-labeledcomments from popular YouTube videos, tweets, and blogs form thedataset. Rigorous preprocessing addresses issues like non-alphabeticcharacters and missing values, setting the stage for effective analysis. The application of versatile NLP techniques follows, including tokenization, removal of stop words, lemmatization, and stemming. These processes refine the raw textual data, preparing it for machine learning analysis. Vectorization, accomplished through both CountVectorizer and TfidfVectorizer, plays a crucial role in transforming textual data into a format suitable for model training. The subsequent model training involves five distinct machine learning models: Naive Bayes, Logistic Regression, SVM, Decision Tree, and Random Forest.

Results Section

The results section provides a comprehensive insight into the performance metrics of the trained models**.** Accurate sentiment prediction is crucial in the context of social media platforms like YouTube, where understanding user sentiments is integral for content creators, marketers, and platform administrators.The accuracy score for each model is as below.



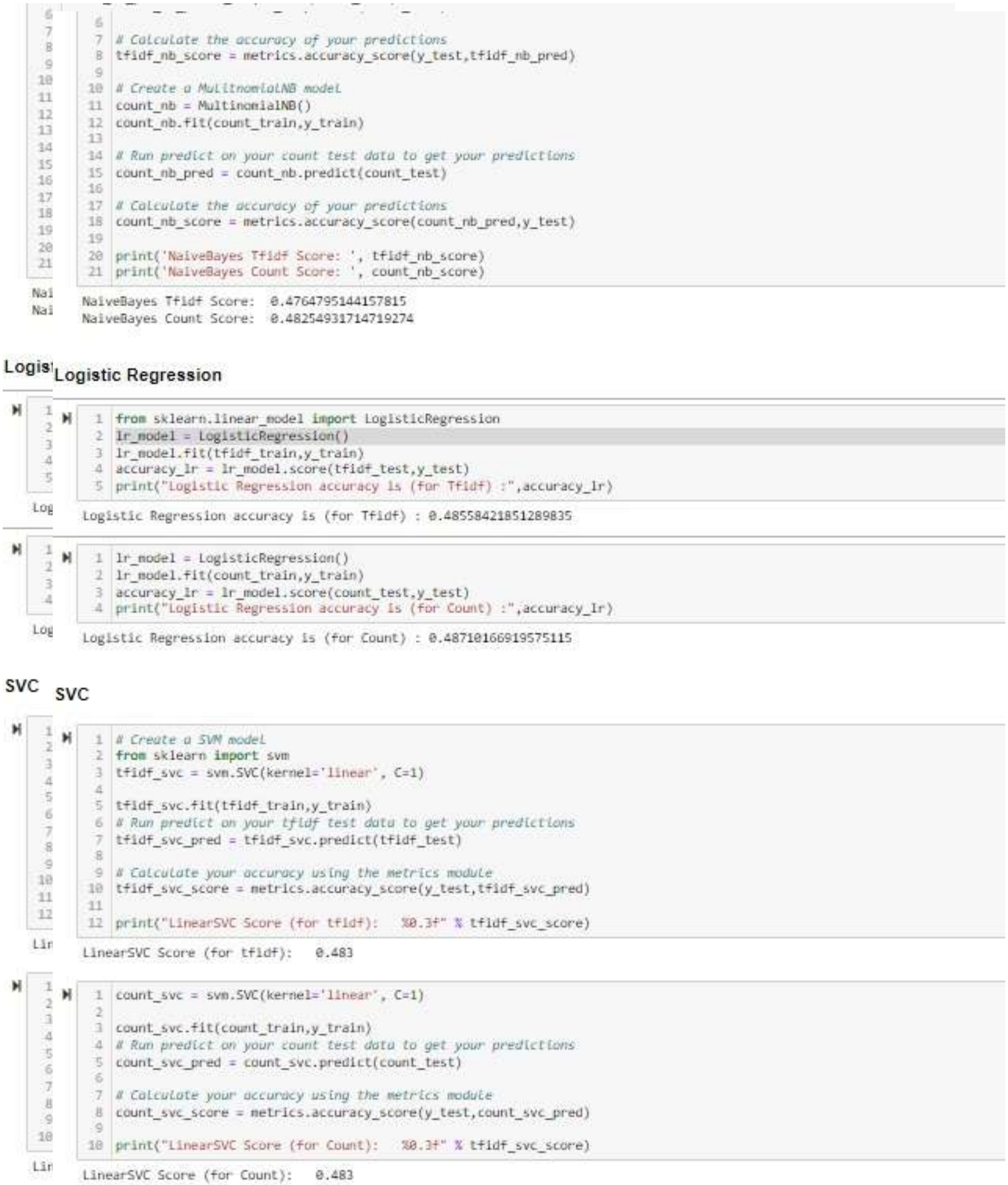
```
neutral = (tfidf_svc_pred == 0.0).sum()
positive = (tfidf_svc_pred == 1.0).sum()
negative = (tfidf_svc_pred < 0).sum()
```

```
print(neutral, positive, negative)
```

```
943 58 0
```

```
print("Good video" if positive > negative else "Bad video")
```

```
Good video
```



```

# Calculate the accuracy of your predictions
tfidf_nb_score = metrics.accuracy_score(y_test,tfidf_nb_pred)

# Create a MulitnomialNB model
count_nb = MultinomialNB()
count_nb.fit(count_train,y_train)

# Run predict on your count test data to get your predictions
count_nb_pred = count_nb.predict(count_test)

# Calculate the accuracy of your predictions
count_nb_score = metrics.accuracy_score(count_nb_pred,y_test)

print('NaiveBayes Tfidf Score: ', tfidf_nb_score)
print('NaiveBayes Count Score: ', count_nb_score)
```

```
NaiveBayes Tfidf Score:  0.4764795144157815
NaiveBayes Count Score:  0.48254931714719274
```

Logis

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression()
lr_model.fit(tfidf_train,y_train)
accuracy_lr = lr_model.score(tfidf_test,y_test)
print("Logistic Regression accuracy is (for Tfidf) :",accuracy_lr)
```

```
Logistic Regression accuracy is (for Tfidf) : 0.48558421851289835
```

```
lr_model = LogisticRegression()
lr_model.fit(count_train,y_train)
accuracy_lr = lr_model.score(count_test,y_test)
print("Logistic Regression accuracy is (for Count) :",accuracy_lr)
```

```
Logistic Regression accuracy is (for Count) : 0.48710166919575115
```

SVC

SVC

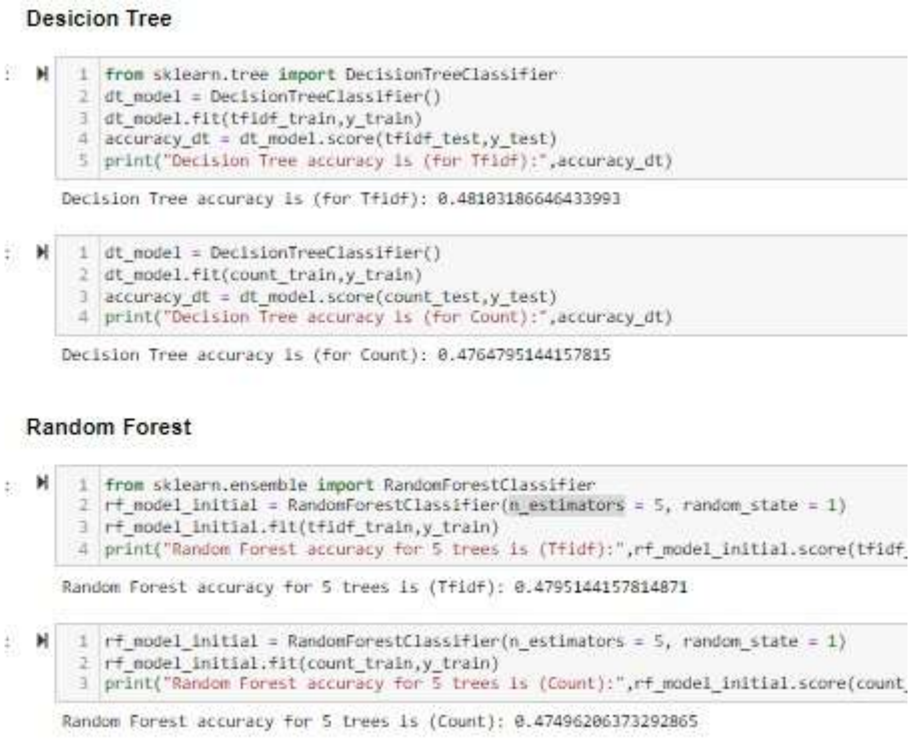
```
# Create a SVM model
from sklearn import svm
tfidf_svc = svm.SVC(kernel='linear', C=1)

tfidf_svc.fit(tfidf_train,y_train)
# Run predict on your tfidf test data to get your predictions
tfidf_svc_pred = tfidf_svc.predict(tfidf_test)

# Calculate your accuracy using the metrics module
tfidf_svc_score = metrics.accuracy_score(y_test,tfidf_svc_pred)

print("LinearSVC Score (for tfidf):   %0.3f" % tfidf_svc_score)
```

```
LinearSVC Score (for tfidf):   0.483
```

```
count_svc = svm.SVC(kernel='linear', C=1)

count_svc.fit(count_train,y_train)
# Run predict on your count test data to get your predictions
count_svc_pred = count_svc.predict(count_test)

# Calculate your accuracy using the metrics module
count_svc_score = metrics.accuracy_score(y_test,count_svc_pred)

print("LinearSVC Score (for Count):   %0.3f" % tfidf_svc_score)
```

```
LinearSVC Score (for Count):   0.483
```

The final result is as follows:



Desicion Tree

```
from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier()
dt_model.fit(tfidf_train,y_train)
accuracy_dt = dt_model.score(tfidf_test,y_test)
print("Decision Tree accuracy is (for Tfidf):",accuracy_dt)
```

```
Decision Tree accuracy is (for Tfidf): 0.48103186646433993
```

```
dt_model = DecisionTreeClassifier()
dt_model.fit(count_train,y_train)
accuracy_dt = dt_model.score(count_test,y_test)
print("Decision Tree accuracy is (for Count):",accuracy_dt)
```

```
Decision Tree accuracy is (for Count): 0.4764795144157815
```

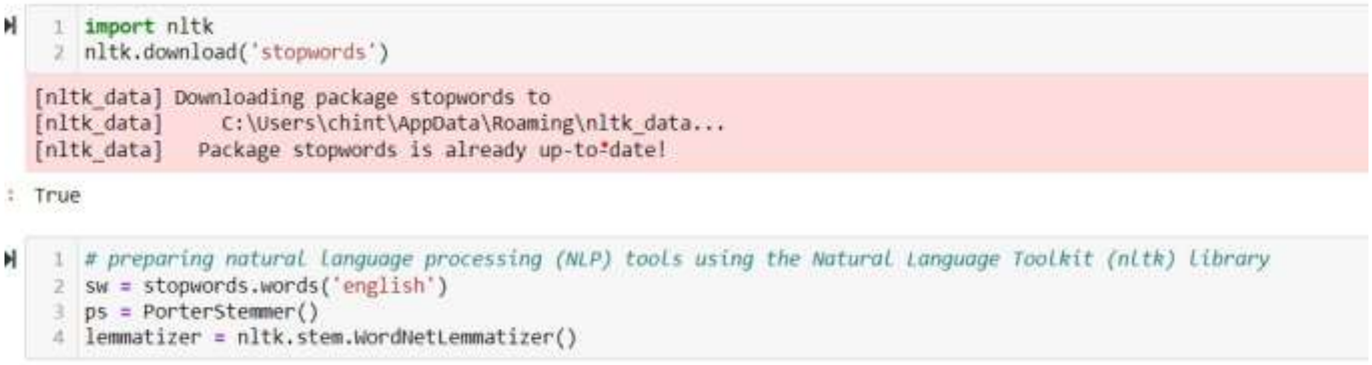
Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rf_model_initial = RandomForestClassifier(n_estimators = 5, random_state = 1)
rf_model_initial.fit(tfidf_train,y_train)
print("Random Forest accuracy for 5 trees is (Tfidf):",rf_model_initial.score(tfidf_
```

```
Random Forest accuracy for 5 trees is (Tfidf): 0.4795144157814871
```

```
rf_model_initial = RandomForestClassifier(n_estimators = 5, random_state = 1)
rf_model_initial.fit(count_train,y_train)
print("Random Forest accuracy for 5 trees is (Count):",rf_model_initial.score(count_
```

```
Random Forest accuracy for 5 trees is (Count): 0.47496206373292865
```

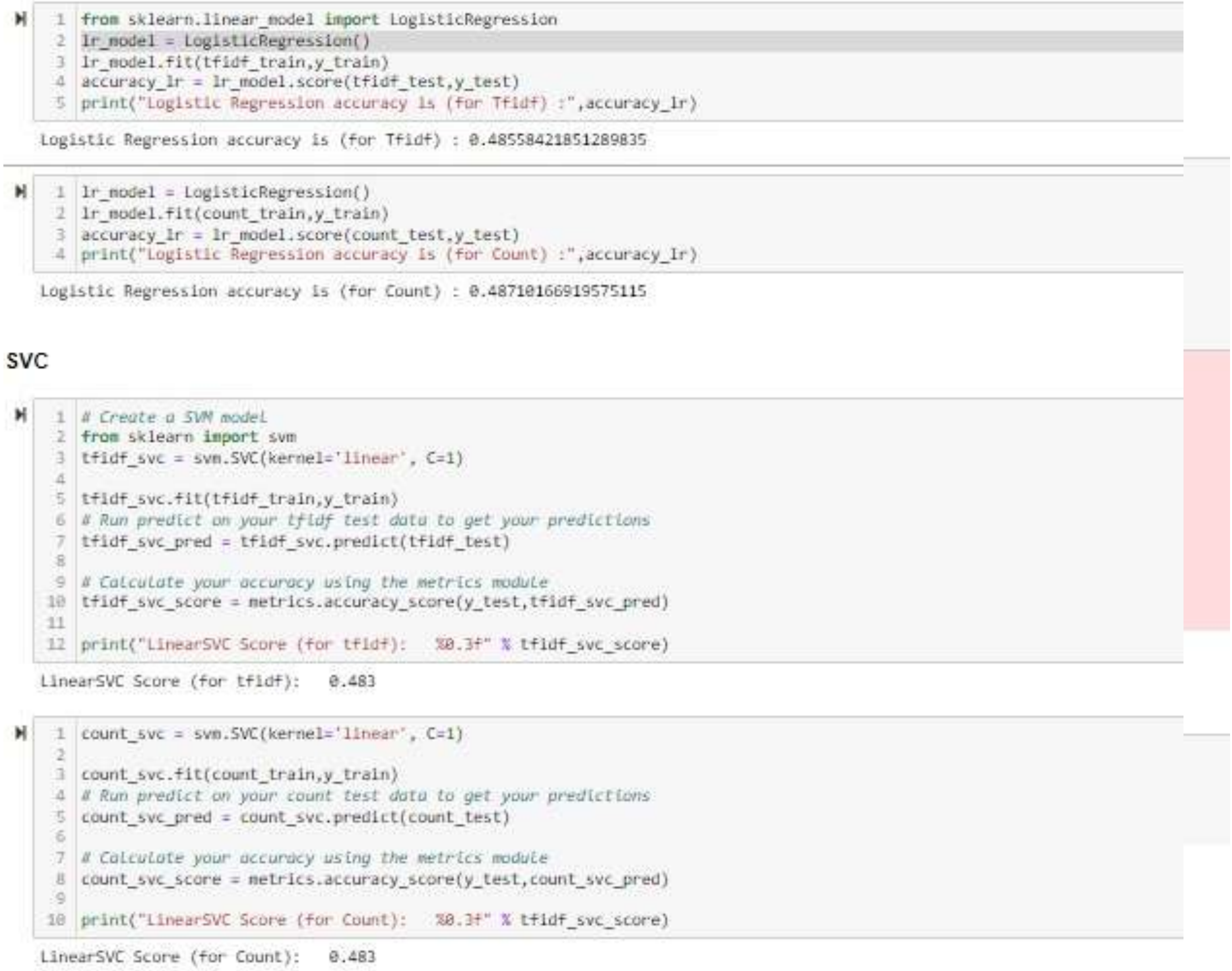


```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\chint\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
True
```

```
# preparing natural language processing (NLP) tools using the Natural Language Toolkit (nltk) library
sw = stopwords.words('english')
ps = PorterStemmer()
lemmatizer = nltk.stem.WordNetLemmatizer()
```



```
from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression()
lr_model.fit(tfidf_train,y_train)
accuracy_lr = lr_model.score(tfidf_test,y_test)
print("Logistic Regression accuracy is (for Tfidf) :",accuracy_lr)
```

```
Logistic Regression accuracy is (for Tfidf) : 0.48558421851289835
```

```
lr_model = LogisticRegression()
lr_model.fit(count_train,y_train)
accuracy_lr = lr_model.score(count_test,y_test)
print("Logistic Regression accuracy is (for Count) :",accuracy_lr)
```

```
Logistic Regression accuracy is (for Count) : 0.48710166919575115
```

SVC

```
# Create a SVM model
from sklearn import svm
tfidf_svc = svm.SVC(kernel='linear', C=1)

tfidf_svc.fit(tfidf_train,y_train)
# Run predict on your tfidf test data to get your predictions
tfidf_svc_pred = tfidf_svc.predict(tfidf_test)

# Calculate your accuracy using the metrics module
tfidf_svc_score = metrics.accuracy_score(y_test,tfidf_svc_pred)

print("LinearSVC Score (for tfidf):   %0.3f" % tfidf_svc_score)
```

```
LinearSVC Score (for tfidf):   0.483
```

```
count_svc = svm.SVC(kernel='linear', C=1)

count_svc.fit(count_train,y_train)
# Run predict on your count test data to get your predictions
count_svc_pred = count_svc.predict(count_test)

# Calculate your accuracy using the metrics module
count_svc_score = metrics.accuracy_score(y_test,count_svc_pred)

print("LinearSVC Score (for Count):   %0.3f" % tfidf_svc_score)
```

```
LinearSVC Score (for Count):   0.483
```

Discussion

The discussion section delves into the implications of the project's findings within the broader context of sentiment analysis on social media. Beyond numerical metrics, the discussion considers the practical applications and significance of the results. Acknowledgingchallenges and limitations, it lays the foundation for future researchdirections. By addressing the broader context, this section ensures that the findings are contextualized within the dynamic landscape ofsocial media sentiment analysis.

Conclusion

The conclusion succinctly highlights the project's unique contributions, emphasizing the amalgamation of various machine learning models and meticulous application of NLP techniques. This synthesis creates a versatile toolkit for sentiment analysis on YouTube. The project's significance in the field of data science is underscored by its practical applications and potential impact on understanding digital interactions within social media platforms.

Contributions/References

Contributions:

The YouTube Sentiment Analysis project makes several significantcontributions to the field of sentiment analysis and social media analytics:

Model Diversity:

The project introduces a diverse set of machine learning models, including Naive Bayes, Logistic Regression, SVM, Decision Tree, andRandom Forest. This model diversity enriches the toolkit available for sentiment analysis, allowing for a more nuanced interpretation of YouTube comments.

Versatile NLP Techniques:

Meticulous application of advanced NLP techniques, such as tokenization, stop words removal, lemmatization, and stemming, enhances the robustness of sentiment analysis. This contributes to amore refined understanding of user sentiments in the context of YouTube comments.

Real-World Application:

The project goes beyond theoretical exploration by practically applying sentiment analysis to predict sentiments for comments onspecific YouTube videos. This real-world application showcases theadaptability and relevance of the developed models to new data.

Practical Tool for Stakeholders:

The project provides content creators, marketers, and platform administrators with a tangible tool for sentiment analysis. The insights derived from this analysis can inform content strategies, marketing campaigns, and platform management decisions, contributing to a more informed and engaging user experience.

Reference:

The project draws upon a comprehensive set of references to informits methodology, models, and discussions. Key references include:

1. *Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods innatural language processing-Volume 10.*
2. *Bird, S., Klein, E., & Loper, E. (2009). Natural language processingwith Python: analyzing text with the natural language toolkit.*
3. *O'Reilly Media, Inc.*
4. *Chen, H., Zhang, J., Xu, B., & Chen, H. (2017). Sentiment analysisusing various machine learning techniques: A review. Computer Science Review, 22, 67-73.*

5. *Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. SpringerScience & Business Media.*
6. *These references contribute foundational knowledge in the areas ofsentiment analysis, natural language processing, and machine.*
7. *learning, guiding the project's methodology and enriching itsanalytical framework.*