



University
of Windsor

Guided By :
Dr. Prashanth Ranga

Team 4

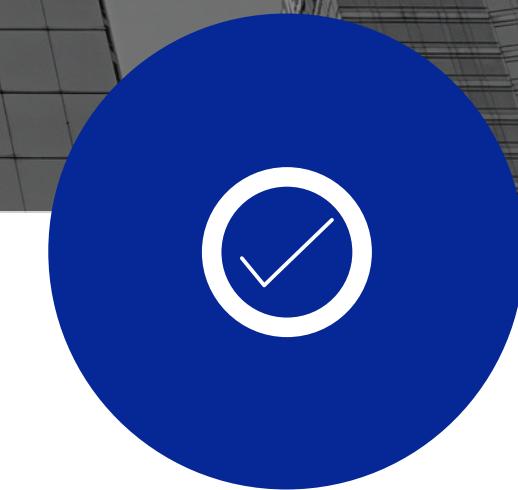
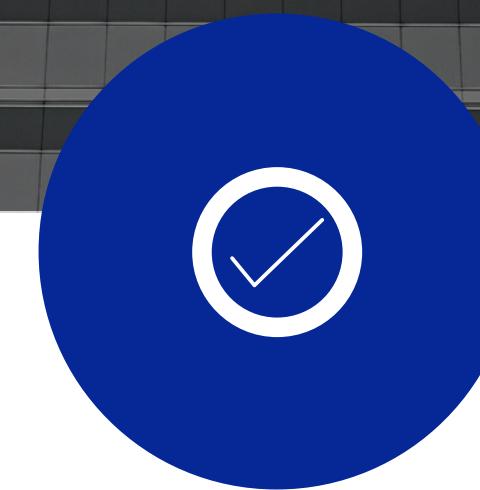
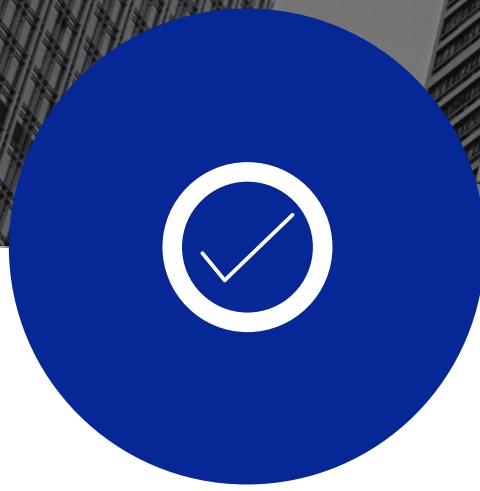
Tixa Patel
Nirali Limbad
Div Patel
Maharshi Prajapati

Sentiment Analysis of Customer Reviews on Amazon Products

Our Agenda

- Problem
- Introduction
- Approach
- Tools and Technologies
- Implementation
- Results
- Demo
- Conclusion
- Future Scope
- References

Problem and Motivation



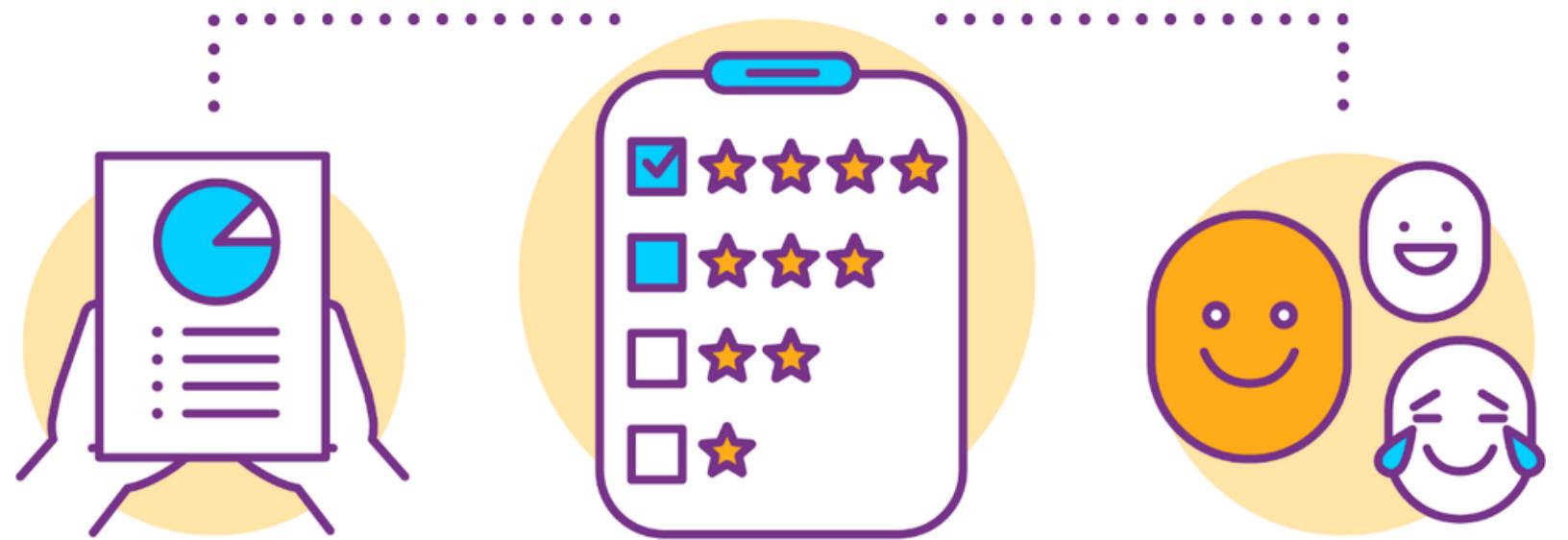
Businesses and researchers in e-commerce platforms might benefit from analysing and utilising user product reviews in emotions.

With the shift from traditional brick-and-mortar retail outlets to online shopping, product reviews are becoming increasingly significant.

In real-time, customers are leaving reviews on product pages. Because of the large number of customer evaluations, it is possible to see how the market reacts to a particular product.

Introduction

We will differentiate Amazon product reviews into two categories: positive and negative which will help to improve the quality of the product and define the market needs for producers.



Sentiment analysis

The producer can filter the negative review and study them to improve the product quality while on the other hand, positive comments will help them in estimating the production in a certain amount of time.

Our Approach

01

Load Data into HDFS

Gather customer reviews and load data into HDFS

02

Data Preprocessing

Cleaning the data and preparing it for a machine learning model. It improves the model's accuracy and efficiency

03

Sentiment Analysis

Using Logistic Regression, to build a sentiment predictive model

04

Data Visualization

Visualize the gathered data in a form of a graph, table, and matrix

Tools and Technologies

Programming Language: Python

Development Tools: Anaconda, Jupyter Notebook

Machine Learning Libraries: NumPy, Pandas, Sklearn

Model: Logistic Regression

Database Management System: Hadoop, Hue

Datasets: Amazon Product Reviews



Implementation

Step-1: Loading Data

HUE

Query ▾

Search data and saved documents...

Jobs cloudera

Tables (3) ▾ +

amazonreviews
reviewid (string)
dateadded (timestamp)
dateupdated (timestamp)
name (string)
asins (string)
brand (string)
categories (string)
primarycategories (string)
imageurls (string)
keys (string)
manufacturer (string)
manufacternumber (string)
reviewsdate (timestamp)
reviewsdateseen (timestamp)
reviewsdidpurchase (string)
reviewsdorecommend (string)
reviewsid (string)
reviewsnumhelpful (string)
reviewsrating (bigint)
reviewssourceurls (string)
reviewstext (string)
reviewstitle (string)
reviewsusername (string)
sourceurls (string)

datafiniti_amazon_consumer_reviews_of_amazon
newreviews

Name	Type		
id	string	AVqVGZNvQMIgsOJE6eUY	AVqVGZNvQMIgsOJE6eUY
dateAdded	timestamp	2017-03-03T16:56:05Z	2017-03-03T16:56:05Z
dateUpdated	timestamp	2018-10-25T16:36:31Z	2018-10-25T16:36:31Z
name	string	Amazon Kindle E-Read...	Amazon Kindle E-Read...
asins	string	B00ZV9PXP2	B00ZV9PXP2
brand	string	Amazon	Amazon
categories	string	Computers,Electronics...	Computers,Electronics...
primaryCategories	string	Electronics	Electronics
imageURLs	string	https://pisces.bbyst...	https://pisces.bbyst...
keys	string	allnewkindleereaderb...	allnewkindleereaderb...
manufacturer	string	Amazon	Amazon
manufacturerNumber	string	B00ZV9PXP2	B00ZV9PXP2

Back Submit

Loading Data

HUE

Query ▾

Search data and saved documents...

Jobs cloudera

☰ HUE

Tables (3) ▾ +

amazonreviews
reviewid (string)
dateadded (timestamp)
dateupdated (timestamp)
name (string)
asins (string)
brand (string)
categories (string)
primarycategories (string)
imageurls (string)
keys (string)
manufacturer (string)
manufacturnumber (string)
reviewsdate (timestamp)
reviewsdateseen (timestamp)
reviewsdidpurchase (string)
reviewsdorecommend (string)
reviewsid (string)
reviewsnump helpful (string)
reviewsrating (bigint)
reviewssourceurls (string)
reviewstext (string)
reviewstitle (string)
reviewsusername (string)
sourceurls (string)
datainiti_amazon_consumer_reviews_of_amazon
newreviews

Name	Type	Value	Value
reviews.date	timestamp	2017-09-03T00:00:00....	2017-06-06T00:00:00....
reviews.dateAdded	string		
reviews.dateSeen	string	2018-05-27T00:00:00Z...	2018-05-27T00:00:00Z...
reviews.doRecommend	string	FALSE	TRUE
reviews.id	bigint		
reviews.numHelpful	bigint	0	0
reviews.rating	bigint	3	5
reviews.sourceURLs	string	http://reviews.bestb...	http://reviews.bestb...
reviews.text	string	I thought it would b...	This kindle is light...
reviews.title	string	Too small	Great light reader...
reviews.username	string	llyyue	Charmi
sourceURLs	string	https://www.newegg.c...	https://www.newegg.c...

Back Submit

Step-2: Text Preprocessing

```
In [9]: #import libraries from Sklearn
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import plot_confusion_matrix, plot_precision_recall_curve
```

```
In [10]: # label Encoder
```

```
vect_data = TfidfVectorizer(ngram_range=(1,2),
                            min_df=3,
                            max_df=0.9,
                            strip_accents='unicode',
                            use_idf=1,
                            smooth_idf=1,
                            sublinear_tf=1)

encoder = LabelEncoder()
```

```
In [11]: # transforming data using fit_transform
```

```
X_train = vect_data.fit_transform(train['reviews'])
X_test = vect_data.transform(test['reviews'])
Y_train = encoder.fit_transform(train['label'])
Y_test = encoder.transform(test['label'])
```

Data Processing

In [12]: # *Logistical Regression*

```
data_model = LogisticRegression(C=4, dual=True, solver='liblinear', random_state=42)
data_model.fit(X_train, Y_train)
```

Out[12]: LogisticRegression(C=4, dual=True, random_state=42, solver='liblinear')

In [13]: # *predicting probabilities*

```
predicts = data_model.predict_proba(X_test)
```

Results

```
In [5]: # Printing data with top 50 reviews
```

```
train.head(50)
```

Out[5]:

	label	reviews
2079998	Negative	Expensive Junk: This product consists of a pie...
1443106	Negative	Toast too dark: Even on the lowest setting, th...
3463669	Positive	Excellent imagery...dumbed down story: I enjoy...
2914699	Negative	Are we pretending everyone is married?: The au...
1603231	Negative	Not worth your time: Might as well just use a ...
2944012	Negative	Book reads like written for grade schoolers: I...
3403602	Negative	Jeanne de Florette & Manon of the Springs: I s...
1039142	Negative	Theater Projector Ceiling Mount: Would not fit...
431645	Positive	This import is sooooooooooooo good: This is a gr...
694263	Negative	Garbage: The handle broke clean off after TWO ...
36844	Positive	Amazing, Fascinating: This was a stunning book...
1915310	Negative	Not great: This book just gives you the basics...
621191	Negative	superficial fluff: This book does not convey m...
939874	Positive	Great Album - some serious stuff: First off, I...

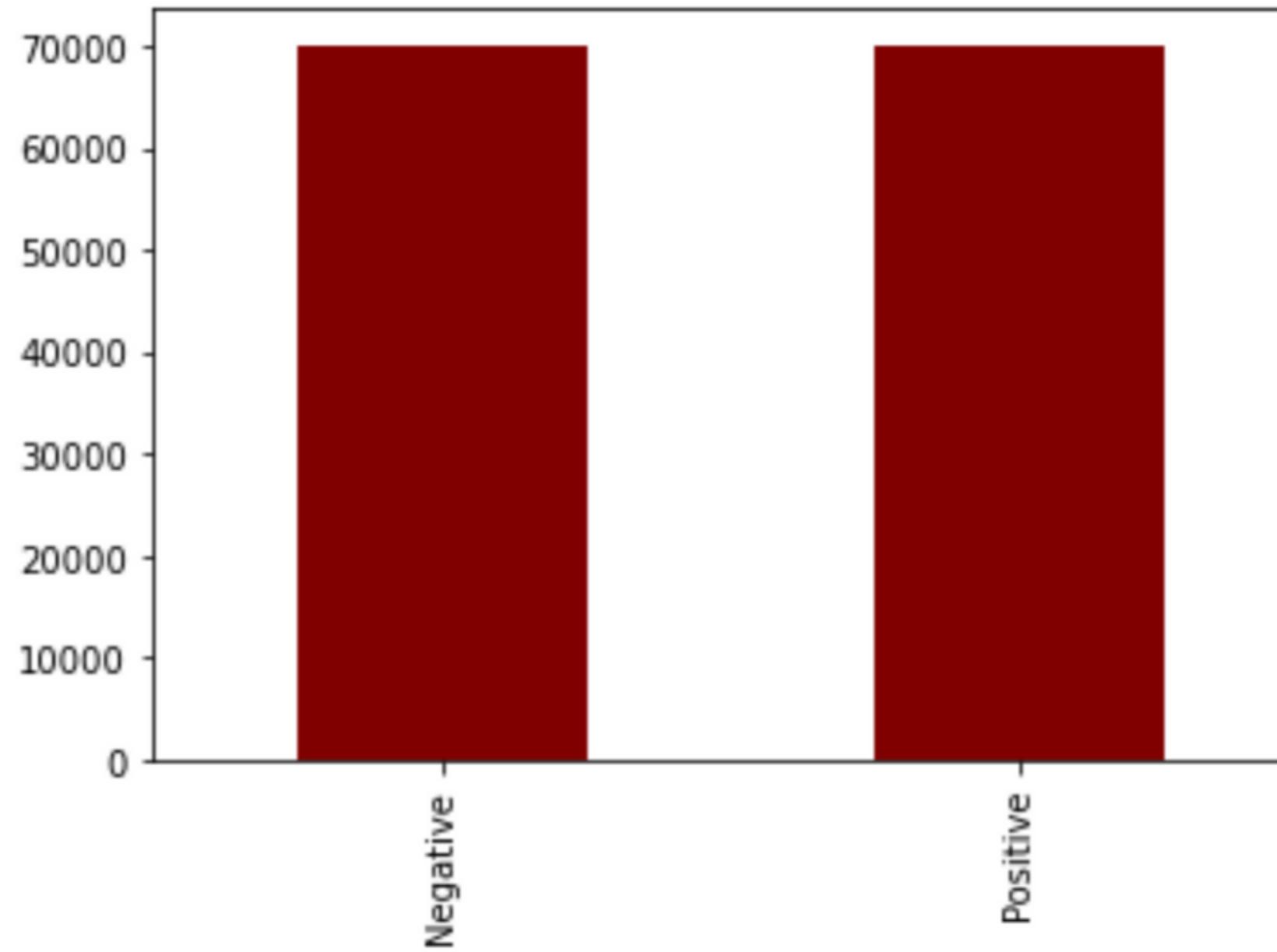
Table Representation for Positive and
Negative Reviews

Results

In [6]: *# plot bar graph for data visualization*

```
train['label'].value_counts().plot(kind='bar', color='maroon')
```

Out[6]: <AxesSubplot:>

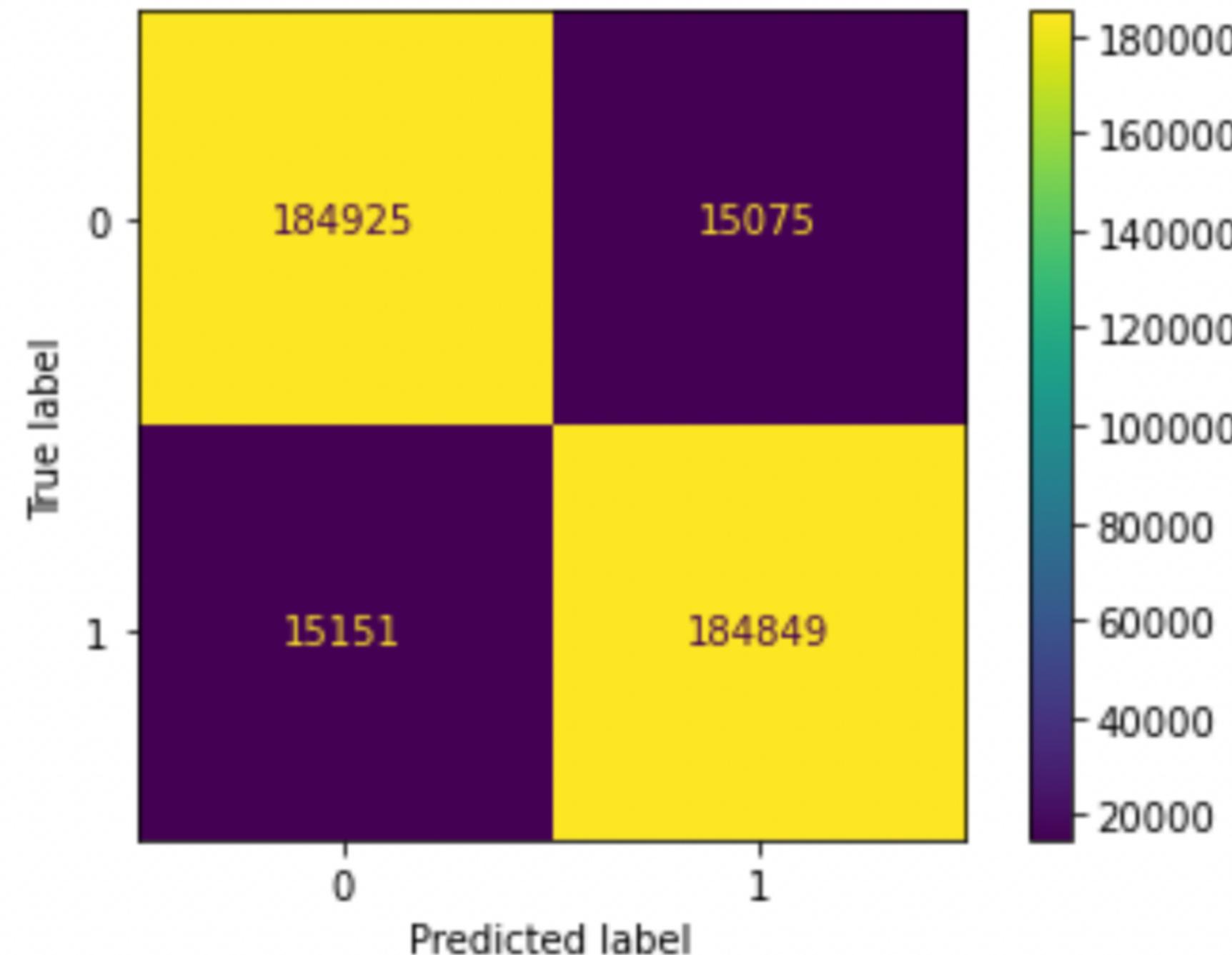


Bar Graph for Data Visualize

Results

```
In [14]: # Matrix
```

```
plot_confusion_matrix(data_model, X_test, Y_test);
```

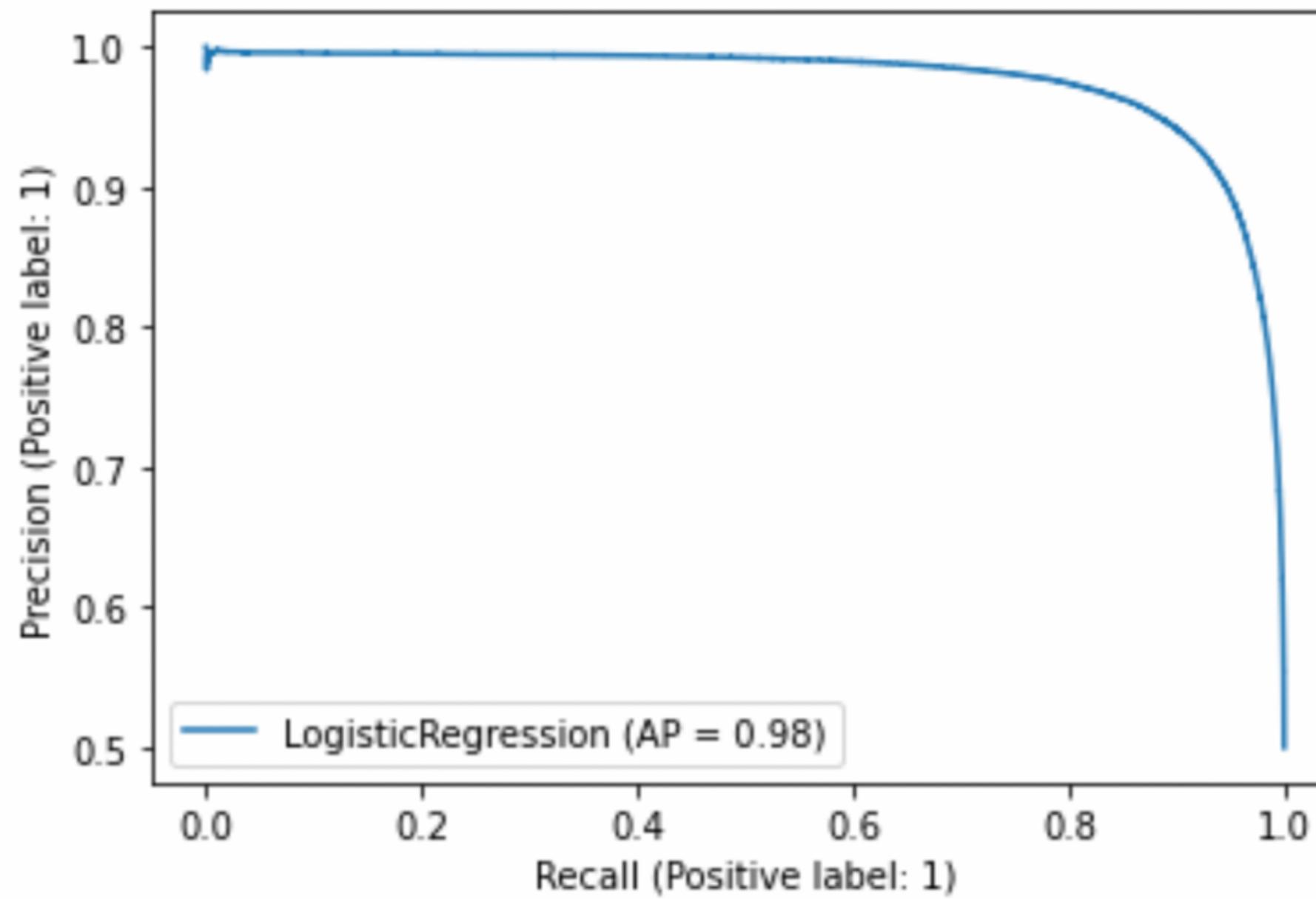


Matrix for Model Accuracy

Results

```
In [15]: # plot curve
```

```
plot_precision_recall_curve(data_model, X_test, Y_test);
```



Precision Recall Curve for Model Precision

Results

```
In [18]: # User Interactive model for providing outcome based on review enter by user

user_input = vect_data.transform([input()])
predict2 = data_model.predict_proba(user_input)
predict3 = data_model.predict(user_input)

if predict3[0]:
    print("\nSentimental Analysis"\nThis review is Positive" " \U0001f600" "\nSentiment Score:", predict2[0][1])

else:
    print("\nSentimental Analysis\nThis review is Negative" " \U0001f621" "\nSentiment Score:", predict2[0][1])
```

Great book for travelling Europe: I currently live in Europe, and this is the book I recommend for my visitors. It covers many countries, colour pictures, and is a nice starter for before you go, and once you are there.

Sentimental Analysis
This review is Positive 😊
Sentiment Score: 0.9980583185819476

```
In [34]: # User Interactive model for providing outcome based on review enter by user

user_input = vect_data.transform([input()])
predict2 = data_model.predict_proba(user_input)
predict3 = data_model.predict(user_input)

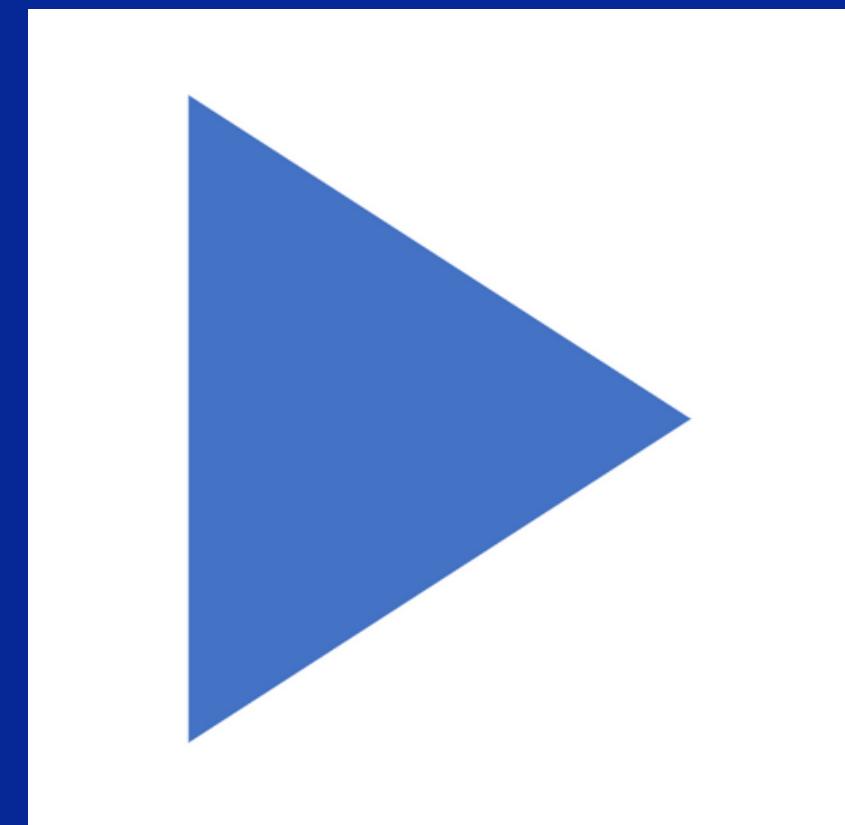
if predict3[0]:
    print("\nSentimental Analysis"\nThis review is Positive" " \U0001f600" "\nSentiment Score:", predict2[0][1])
| 

else:
    print("\nSentimental Analysis\nThis review is Negative" " \U0001f621" "\nSentiment Score:", predict2[0][1])
```

I am very excited to buy this painting for my friend but when I got this product. I am disappointing with this as it is not very frustrating to stick on wall. I waste my money and time on this product.

Sentimental Analysis
This review is Negative 😞
Sentiment Score: 7.450884339937399e-05

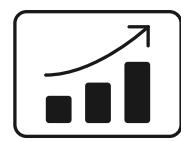
Launching our Sentiment Analysis Demo



Conclusion

- Improve Customer Service
- Develop Quality Products
- Discovering New Marketing Strategies
- Improve Media Perceptions
- Increasing Sales Revenue
- Improve Crises Management

Future Scope



Machine Learning models can not work as human minds. Tone can be difficult to interpret verbally, and even more difficult to figure out in the written word. Our team did a study on this issue and came to know about the tone detector algorithm (Smart sentiment API).

References

- <https://www.kaggle.com/bittlingmayer/amazonreviews>
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Thank you!

Feel free to approach us
if you have any questions.

