PROJECT

Title of Project

Women Cloth Reviews Prediction with Multinomial Naïve Bayes

Objective

The objective of this project is to predict the sentiment of women's clothing reviews using the Multinomial Naïve Bayes algorithm. This involves classifying the reviews as positive or negative based on the text data.

Data Source

The data source for this project is a dataset containing women's clothing reviews. This dataset includes various features such as review text, rating, and other relevant information.

Program

```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
from wordcloud import WordCloud

# Title of Project
```

```
print("Project: Women Cloth Reviews Prediction with Multinomial Naïve Bayes")


# Objective

print("\nObjective: To predict the sentiment of women's clothing reviews using the Multinomial Naïve Bayes algorithm.")


# Import Data

data = pd.read_csv('women_cloth_reviews.csv')


# Describe Data

print("\nData Description:")

print(data.head())

print(data.info())

print(data.describe())


# Data Visualization

print("\nData Visualization:")


# Distribution of ratings

sns.countplot(x='Rating', data=data)

plt.title('Distribution of Ratings')

plt.show()


# Word cloud of review text

wordcloud = WordCloud(width=800, height=400, background_color='white').generate(' '.join(data['Review Text'].dropna()))

plt.figure(figsize=(10, 5))
```

```
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Review Text')
plt.show()

# Data Preprocessing
print("\nData Preprocessing:")
data = data.dropna(subset=['Review Text'])
data['Sentiment'] = data['Rating'].apply(lambda x: 1 if x > 3 else 0)

# Define Target Variable (y) and Feature Variables (X)
print("\nDefining Target Variable (y) and Feature Variables (X):")
X = data['Review Text']
y = data['Sentiment']
print(f"X (Feature Variables): {X.head()}")
print(f"y (Target Variable): {y.head()}")

# Train Test Split
print("\nSplitting the Data into Training and Test Sets:")
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Modeling
print("\nModeling with Multinomial Naïve Bayes:")
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
model = MultinomialNB()
```

```
model.fit(X_train_vec, y_train)

# Model Evaluation
print("\nModel Evaluation:")
y_pred = model.predict(X_test_vec)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy}')
print('Confusion Matrix:')
print(conf_matrix)
print('Classification Report:')
print(class_report)

# Prediction
print("\nExample Prediction:")
example_review = ["This dress is amazing and fits perfectly!"]
example_review_vec = vectorizer.transform(example_review)
prediction = model.predict(example_review_vec)
print(f'Prediction for example review: {"Positive" if prediction[0] == 1 else
"Negative"}')

print("\nProject completed successfully!")
```

Explanation

This project aims to predict the sentiment of women's clothing reviews using the Multinomial Naïve Bayes algorithm. The dataset is preprocessed to handle missing values and convert ratings into binary sentiment labels. The text data is vectorized using CountVectorizer, and the model is trained and evaluated on this vectorized data. The model's performance is assessed using accuracy, confusion matrix, and classification report. Finally, the model is used to predict the sentiment of an example review.