Women Cloth Reviews Prediction with Multi Nomial Naïve Bayes full python code

To build a prediction model for women’s clothing reviews using the Multinomial Naive Bayes algorithm, we’ll use Python libraries like pandas, sklearn, and nltk. Below is a step-by-step guide, including the full code.

**Step 1: Install Necessary Libraries**

You may need to install the following libraries if you haven't already:

bash

Copy code

pip install pandas scikit-learn nltk

**Step 2: Import Libraries**

python

Copy code

import pandas as pd

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 classification\_report, accuracy\_score

import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords

import re

**Step 3: Load the Data**

Assuming you have a CSV file named women\_clothing\_reviews.csv with columns review and label:

python

Copy code

# Load the dataset

data = pd.read\_csv('women\_clothing\_reviews.csv')

# Display the first few rows

print(data.head())

**Step 4: Data Preprocessing**

We will clean the text data by removing punctuation and stop words.

python

Copy code

# Function to clean the reviews

def clean\_text(text):

text = re.sub(r'\W', ' ', text) # Remove punctuation

text = text.lower() # Convert to lowercase

text = ' '.join([word for word in text.split() if word not in stopwords.words('english')]) # Remove stop words

return text

# Apply the cleaning function

data['cleaned\_review'] = data['review'].apply(clean\_text)

**Step 5: Split the Data**

python

Copy code

# Split the dataset into training and testing sets

X = data['cleaned\_review']

y = data['label']

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

**Step 6: Feature Extraction**

Convert the text data into a format suitable for machine learning using Count Vectorization.

python

Copy code

# Initialize CountVectorizer

vectorizer = CountVectorizer()

# Fit and transform the training data

X\_train\_vectorized = vectorizer.fit\_transform(X\_train)

# Transform the test data

X\_test\_vectorized = vectorizer.transform(X\_test)

**Step 7: Train the Model**

Now, we can train the Multinomial Naive Bayes model.

python

Copy code

# Initialize the model

model = MultinomialNB()

# Train the model

model.fit(X\_train\_vectorized, y\_train)

**Step 8: Make Predictions**

python

Copy code

# Make predictions

y\_pred = model.predict(X\_test\_vectorized)

**Step 9: Evaluate the Model**

python

Copy code

# Evaluate the model

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

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

**Step 10: Use the Model for Prediction**

You can now use the model to predict new reviews.

python

Copy code

# Function to predict new reviews

def predict\_review(review):

cleaned\_review = clean\_text(review)

vectorized\_review = vectorizer.transform([cleaned\_review])

prediction = model.predict(vectorized\_review)

return prediction[0]

# Example usage

new\_review = "I love this dress, it fits perfectly!"

print("Predicted label:", predict\_review(new\_review))