constructer\FINAL PROJECT

```
### **Title of Project**
**Predicting Women's Clothing Reviews using Multinomial Naive Bayes**

### **Objective**
The objective of this project is to predict the sentiment (positive or negative) of women's clothing reviews based on text data using a Multinomial Naive Bayes model. The goal is to build a model that can accurately classify reviews, helping retailers understand customer sentiment.

### **Data Source**
The dataset for this project can be sourced from online retail platforms, such as Kaggle or other repositories that offer customer reviews data. The dataset should include features such as `Review Text`, `Rating`, `Sentiment` (positive/negative), etc.

For example:
- [Women's E-Commerce Clothing Reviews Dataset](https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews)

### **Import Library**

```python
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
```

Import Data

```python
data = pd.read_csv('Womens Clothing E-Commerce Reviews.csv')
```

Describe Data

Provide an overview of the dataset:

```python
print(data.info())
print(data.describe())
print(data.head())
```

Check for missing values, unique values in each column, and basic statistics.

```

````
Data Visualization

Visualize the distribution of key features like `Rating`, `Sentiment`, and the length of `Review Text`.

```python
sns.countplot(x='Rating', data=data)
plt.show()

data['Review Length'] = data['Review Text'].apply(len)
sns.histplot(data['Review Length'], bins=20)
plt.show()
```

Data Preprocessing

1. **Handling Missing Values:** Drop or fill any missing values in the `Review Text` or `Sentiment` columns.

   ```python
   data.dropna(subset=['Review Text'], inplace=True)
   ```

2. **Text Processing:** Convert the review text to lowercase, remove punctuation, and apply tokenization or stemming if necessary.

   ```python
   data['Review Text'] = data['Review Text'].str.lower()
   ```

3. **Vectorization:** Convert the text data into numerical data using `CountVectorizer`.

   ```python
   vectorizer = CountVectorizer(stop_words='english')
   X = vectorizer.fit_transform(data['Review Text'])
   ```

Define Target Variable (y) and Feature Variables (X)

```python
y = data['Sentiment']  # Assuming 'Sentiment' is already defined as positive/negative
```

Train Test Split

Split the data into training and testing sets:

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```
````

```
### **Modeling**

Train the Multinomial Naive Bayes model:

```python
model = MultinomialNB()
model.fit(X_train, y_train)
```

Model Evaluation

Evaluate the model's performance using the test data:

```python
y_pred = model.predict(X_test)
print(f'Accuracy: {accuracy_score(y_test, y_pred)}')
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Prediction

Predict sentiments on new or unseen reviews:

```python
new_reviews = ["This dress is amazing!", "I didn't like the fabric at all."]
new_reviews_transformed = vectorizer.transform(new_reviews)
predictions = model.predict(new_reviews_transformed)
print(predictions)
```

Explanation

Summarize the findings, discussing the model's accuracy, common errors (based on the confusion matrix), and how the model can be used in real-world applications. Consider providing recommendations for improving the model or integrating it into a retail system to automatically analyze customer sentiment.
```