



OJASHWEE RAMAN

Email: ojashwee.raman2021@vitstudent.ac.in | Phone: 9693592086 | LinkedIn: [linkedin.com/in/ojashweeaman](https://www.linkedin.com/in/ojashweeaman)

Classification of articles based on title and content.

Code:

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB

# Load and preprocess the datasets
classification_data = pd.read_csv('Dataset_Classification.csv')
description_data = pd.read_csv('Categories_Description.csv')

# Assuming classification_data has 'content' and 'category' columns
X_train = classification_data['content']
y_train = classification_data['title']

# Assuming description_data has 'description' and 'name' columns
X_test = description_data['description']
names = description_data['name']

# Create a CountVectorizer to convert text into numerical features
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Train a Multinomial Naive Bayes classifier
classifier = MultinomialNB()
classifier.fit(X_train_vec, y_train)

# Classify descriptions and create a new dataframe
classified_categories = classifier.predict(X_test_vec)
classified_df = pd.DataFrame({'content': X_test, 'description':
classified_categories, 'name': names})

# Save the final classified dataset to a CSV file
classified_df.to_csv('Final_Classified_Dataset.csv', index=False)
```

Open-Sou17-08-2023rce AI Model Used:

We used a simple text classification model based on the **Multinomial Naive Bayes algorithm**. Naive Bayes is a probabilistic algorithm often used for text classification tasks. It's considered a relatively lightweight and easy-to-implement algorithm suitable for simple classification tasks.

Method Used:

The method used in this code is text classification. Text classification is the process of assigning predefined categories or labels to textual data based on its content. In this case, the method involves training a model on a labeled dataset (with content and corresponding categories), then using the trained model to predict categories for a new dataset (with descriptions). The method leverages the bag-of-words representation and the Multinomial Naive Bayes algorithm for classification.

Libraries Used:

- **pandas**: For data manipulation and analysis.
- **CountVectorizer from sklearn.feature_extraction.text**: For converting text data into numerical features.
- **MultinomialNB from sklearn.naive_bayes**: For implementing the Multinomial Naive Bayes algorithm.

Working of the Code:

- Load the 'Dataset_Classification.csv' and 'Categories_Description.csv' datasets using the `pd.read_csv` function from the `pandas` library.
- Separate the training data ('content' and 'category') from the 'Dataset_Classification.csv' and the testing data ('description' and 'name') from the 'Categories_Description.csv'.
- Create a `CountVectorizer` object to convert text data into numerical features. Fit it on the training data ('content') using the `fit_transform` method and transform the testing data ('description') using the `transform` method.
- Train a `MultinomialNB` classifier using the training data ('content' and 'category').
- Use the trained classifier to predict categories for the testing data ('description').
- Create a new dataframe containing the classified descriptions, their corresponding names, and the original descriptions.
- Save the new dataframe to a CSV file named 'Final_Classified_Dataset.csv'.