phase 4: TRAINING THE MODEL USING ENSEMBLE MODEL ALGORITHM

TOPIC: BUILDING A SMARTER AI-POWERED SPAM CLASSIFIER

STEP 1: WE HAVE TO IMPORT LIBRARY FOR USING ALGORITHM SUCH AS RANDOMFOREST CLASSIFIER AND VOTING CLASSIFIER

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
```

STEP 2:WE HAVE TO SPLIT THE DATASET INTO TRAINING AND TESTING SETS

```
X = data['text']
y = data['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

STEP 3:WE HAVE TO VECTORIZE THE DATASET USING COUNTERVECTORIZER

```
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
```

STEP 4:WE HAVE TO TRAIN THE MODEL

```
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
ensemble_classifier = VotingClassifier(estimators=[('rf', rf_classifier)], voting='hard')

ensemble_classifier.fit(X_train, y_train)
print(X_train)
```

FINAL STEP:AFTER TRAINING THE MODEL THE FINAL OUTPUT HAS BEEN PRINTED

(0, 28704) 1

(0, 1553) 3

(0, 291) 3

(0, 28918) 6

(0, 28349) 1

(0, 6330) 2

(0, 7851) 2

(0, 1991) 2

(0, 9292) 1

(0, 7014) 1

(0, 6430) 1

(0, 29384) 2

(0, 23327) 4

(0, 26823) 1

(0, 24908) 2

(0, 11438) 3

(0, 9296) 3

(0, 24570) 1

(0, 25370) 1

(0, 20129) 4

(0, 11228) 1

(0, 21354) 1

(0, 10204) 1

(0, 7888) 1

(0, 29832) 1

: :

(4135, 5173) 2

(4135, 19523) 4

(4135, 7978) 3

(4135, 26958) 3

(4135, 12665) 2

(4135, 14539) 2

(4135, 9705) 4

(4135, 30090) 1

(4135, 11677) 1

(4135, 671) 4

(4135, 17535) 2

(4135, 4946) 1

(4135, 16641) 3

(4135, 23375) 2

(4135, 10176) 1

(4135, 7810) 3

(4135, 10806) 1

(4135, 972) 1

(4135, 31247) 1

(4135, 11118) 2

(4135, 17582) 1

(4135, 25659) 1

(4135, 9067) 1

(4135, 23522) 1

(4135, 33457) 1

TRAINING CODE :

LANGUAGE :PYTHON

PLATFORM:JUPYTER NOTEBOOK

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
#from sklearn.metrics import accuracy_score, classification_report
#from nltk.corpus import stopwords
#from nltk.stem import PorterStemmer
#from nltk.tokenize import word_tokenize

# Load the dataset
data = pd.read_csv("spam_ham_dataset.csv")

# Preprocess the text data
def preprocess_text(text):
    # Tokenization
    tokens = word_tokenize(text)

    # Removing punctuation and converting to lowercase
    tokens = [word.lower() for word in tokens if word.isalpha()]

    # Removing stopwords
    stop_words = set(stopwords.words("english"))
    tokens = [word for word in tokens if word not in stop_words]
```

```
    # Stemming
    stemmer = PorterStemmer()
    tokens = [stemmer.stem(word) for word in tokens]

    # Join tokens back into a string
    preprocessed_text = " ".join(tokens)

    return preprocessed_text

data['text'] = data['text'].apply(preprocess_text)

# Split the dataset into training and testing sets
X = data['text']
y = data['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Vectorize the text data using CountVectorizer
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)

# Train an ensemble model
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
ensemble_classifier = VotingClassifier(estimators=[('rf', rf_classifier)], voting='hard')
```

```
ensemble_classifier.fit(X_train, y_train)

print(X_train)
```