**SMS Spam Detection using Machine learning**

* **Source code**

**import pandas as pd**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.feature\_extraction.text import TfidfVectorizer**

**from sklearn.naive\_bayes import MultinomialNB**

**from sklearn.metrics import accuracy\_score, classification\_report**

**import re**

**import nltk**

**from nltk.corpus import stopwords**

**# Download NLTK stopwords if not already done**

**nltk.download('stopwords')**

**stop\_words = set(stopwords.words('english'))**

**# Sample dataset (replace with real data like SMS Spam Collection for production)**

**data = {**

**'message': [**

**"Congratulations! You've won a free ticket to Bahamas. Call now!",**

**"Hey, are we still on for lunch tomorrow?",**

**"URGENT: Your account has been suspended. Verify now.",**

**"Thanks for the invite, I'll be there.",**

**"Win a brand new car! Just click this link.",**

**"Meeting at 3 PM in the conference room.",**

**"Free money! Transfer to your account instantly.",**

**"Can you send me the report by EOD?",**

**"You've been selected for a prize! Claim it here.",**

**"Reminder: Doctor's appointment at 10 AM."**

**],**

**'label': ['spam', 'ham', 'spam', 'ham', 'spam', 'ham', 'spam', 'ham', 'spam', 'ham']**

**}**

**df = pd.DataFrame(data)**

**# Preprocessing function**

**def preprocess\_text(text):**

**text = text.lower() # Lowercase**

**text = re.sub(r'[^\w\s]', '', text) # Remove punctuation**

**words = text.split()**

**words = [word for word in words if word not in stop\_words] # Remove stopwords**

**return ' '.join(words)**

**# Apply preprocessing**

**df['processed\_message'] = df['message'].apply(preprocess\_text)**

**# Split data**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['processed\_message'], df['label'], test\_size=0.3, random\_state=42)**

**# Vectorize text using TF-IDF**

**vectorizer = TfidfVectorizer(max\_features=1000) # Limit features for simplicity**

**X\_train\_vec = vectorizer.fit\_transform(X\_train)**

**X\_test\_vec = vectorizer.transform(X\_test)**

**# Train model**

**model = MultinomialNB()**

**model.fit(X\_train\_vec, y\_train)**

**# Predict and evaluate**

**y\_pred = model.predict(X\_test\_vec)**

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

**print("Classification Report:\n", classification\_report(y\_test, y\_pred))**

**# Function to predict new messages**

**def predict\_spam(message):**

**processed = preprocess\_text(message)**

**vec = vectorizer.transform([processed])**

**prediction = model.predict(vec)[0]**

**return prediction**

**# Example usage**

**new\_message = "Claim your free iPhone now!"**

**print(f"Prediction for '{new\_message}': {predict\_spam(new\_message)}")**