spam sms detection

Build an AI model that can classify SMS messages as spam or legitimate. Use techniques like TF-IDF or word embeddings with classifiers like Naive Bayes,Logistic Regression,or Support Vector Machines to identify spam messages.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score, f1_score, confusion_matrix
from sklearn.naive_bayes import MultinomialNB
```

In [2]:
```
file_path='C:\\Users\\Dayakar\\Desktop\\DS  Assigments\\internship 27\\archive (7)\\spam.csv'
spam_data=pd.read_csv(file_path,encoding='Latin1')
spam_data
```

Out[2]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows × 5 columns

In [3]:
```
spam_data.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)
```

In [4]:
```
spam_data
```

Out[4]:

	v1	v2
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ì_ b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

In [5]:
```
spam_data.size
```

Out[5]:
```
11144
```

In [6]:
```
spam_data.shape
```

Out[6]:
```
(5572, 2)
```

In [7]:
```
spam_data.columns
```

Out[7]:
```
Index(['v1', 'v2'], dtype='object')
```

In [8]:
```
spam_data.dtypes
```

Out[8]:
```
v1    object
v2    object
dtype: object
```

In [9]:
```
spam_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   v1      5572 non-null   object
 1   v2      5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

In [10]:
```
spam_data.isnull().sum()
```

Out[10]:
```
v1    0
v2    0
dtype: int64
```

In [11]:
```
spam_data.describe()
```

Out[11]:

	v1	v2
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

In [12]:
```
spam_data.columns=['Category', 'Message']
spam_data
```

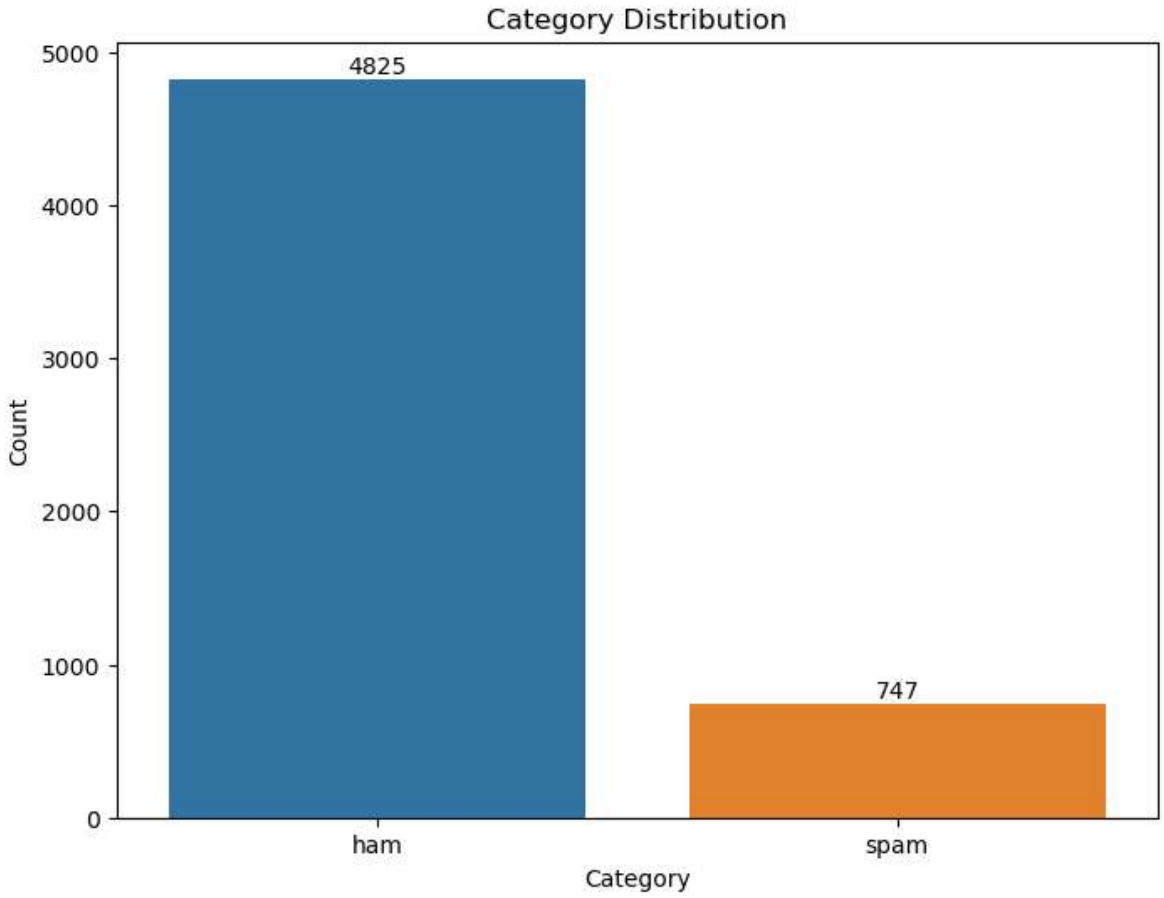
Out[12]:

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ì_ b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

In [15]:
```
category_counts = spam_data['Category'].value_counts().reset_index()
category_counts.columns = ['Category', 'Count']
plt.figure(figsize=(8, 6))
sns.barplot(x='Category', y='Count', data=category_counts)
plt.xlabel('Category')
plt.ylabel('Count')
plt.title('Category Distribution')

for i, count in enumerate(category_counts['Count']):
    plt.text(i, count, str(count), ha='center', va='bottom')
plt.show()
```



In [16]:
```
spam_data['spam']= spam_data['Category'].apply(lambda x: 1 if x=='spam' else 0)
spam_data
```

Out[16]:

	Category	Message	spam
0	ham	Go until jurong point, crazy.. Available only ...	0
1	ham	Ok lar... Joking wif u oni...	0
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	1
3	ham	U dun say so early hor... U c already then say...	0
4	ham	Nah I don't think he goes to usf, he lives aro...	0
...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	1
5568	ham	Will ì_ b going to esplanade fr home?	0
5569	ham	Pity, * was in mood for that. So...any other s...	0
5570	ham	The guy did some bitching but I acted like i'd...	0
5571	ham	Rofl. Its true to its name	0

5572 rows × 3 columns

Traing and Testing of Data

In [17]:
```
X_train, X_test, y_train, y_test = train_test_split(spam_data.Message,spam_data.spam, test_size=0.2
```

In [18]:
```
from sklearn.feature_extraction.text import CountVectorizer
featurer = CountVectorizer()
X_train_count = featurer.fit_transform(X_train.values)
```

In [19]:
```
X_train_count
```

Out[19]:
```
<4457x7729 sparse matrix of type '<class 'numpy.int64'>'
        with 59397 stored elements in Compressed Sparse Row format>
```

Applying the Naive Bayes Method

In [20]:
```
model = MultinomialNB()
model.fit(X_train_count,y_train)
```

Out[20]:
```
MultinomialNB
MultinomialNB()
```

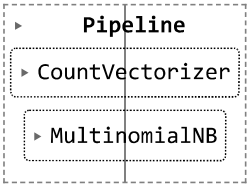
In [21]:
```
X_test_count = featurer.transform(X_test)
model.score(X_test_count, y_test)
```

Out[21]:
```
0.9847533632286996
```

In [22]:
```
from sklearn.pipeline import Pipeline
clf = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('nb', MultinomialNB())
])
```

In [23]:
```
clf.fit(X_train, y_train)
```

Out[23]:



In [24]:
```
clf.score(X_test,y_test)
```

Out[24]:
```
0.9847533632286996
```

Now design a pre_build model to detect spam and not spam message

In [25]:
```
# Pre-trained model
pretrained_model = model
new_sentences = [
    "Your account have 100 debeted, is waiting to be collected. Simply text the password \MIX\" to 8
]

new_sentences_count = featurer.transform(new_sentences)
# Predict whether each sentence is spam (1) or not (0)
predictions = pretrained_model.predict(new_sentences_count)

for sentence, prediction in zip(new_sentences, predictions):
    if prediction == 1:
        print(f"'{sentence}' is a spam message.")
    else:
        print(f"'{sentence}' is not a spam message.")
```

```
'Your account have 100 debeted, is waiting to be collected. Simply text the password \MIX" to 8506
9 to verify. Get Usher and Britney. FML' is a spam message.
```

In [ ]: