```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
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
```

```
data = pd.read_csv('/content/spam.csv',encoding = 'ISO-8859-1')
data.head()
```

	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
		U dun say so early hor... U c already			

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

```
data.describe
```

```
pandas.core.generic.NDFrame.describe
def describe(percentiles=None, include=None, exclude=None) -> Self
```

```
std             NaN      1.0
min             NaN      1.0
25%             NaN      1.5
50%             NaN      2.0
75%             NaN      2.5
max             NaN      3.0
```

Dropping the unwanted columns

```
data = data.drop(columns=data.columns[2:5])
data.head()
```

	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...

Next steps: Generate code with `data` View recommended plots New interactive sheet

```
data.columns = ['Category', 'Message']
data
```

	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

Next steps: Generate code with `data` View recommended plots New interactive sheet

To Check the null value

```
data.isnull().sum()
```

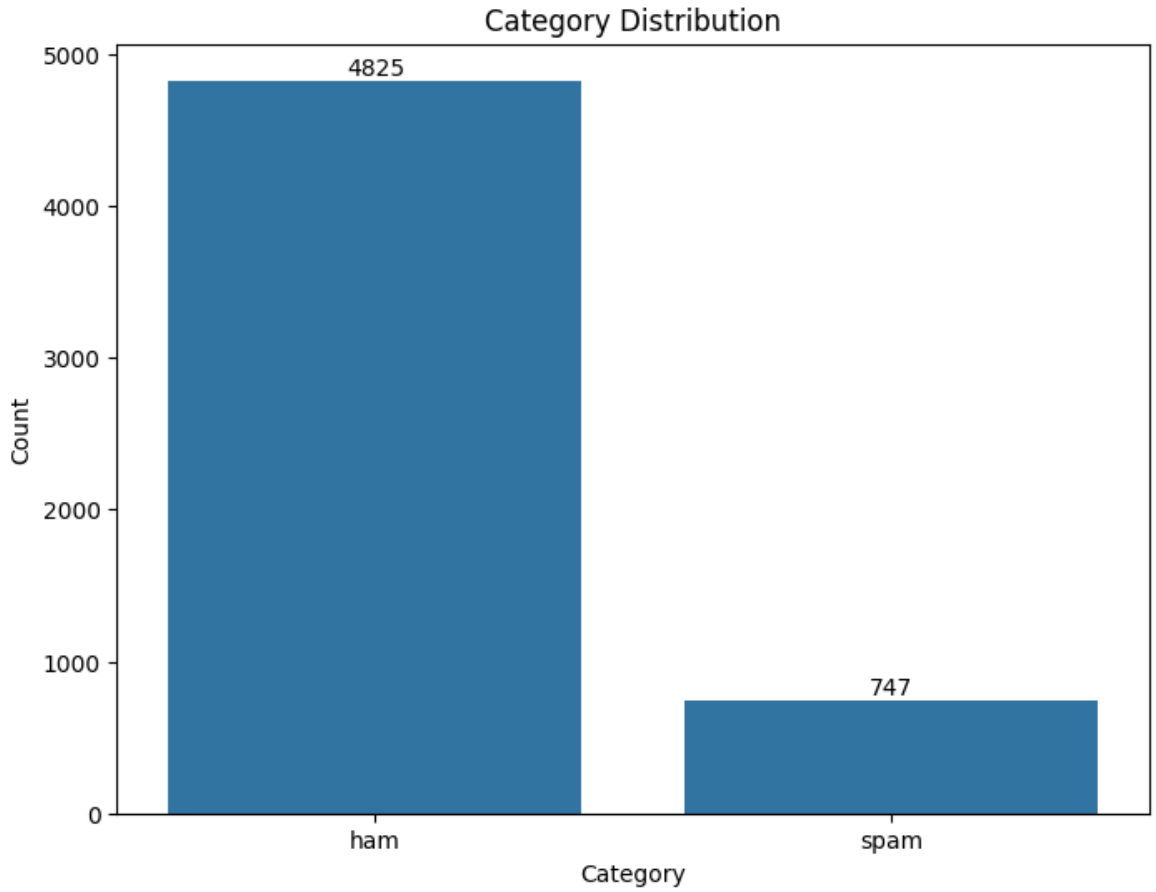
	0
Category	0
Message	0

dtype: int64

Visualisation of data

```
category_counts = 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()
```




```
data['spam']=data['Category'].apply(lambda x:1 if x=='spam' else 0)
data.head(5)
```

	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

Next steps: Generate code with `data` View recommended plots New interactive sheet

Training and testing of data

```
x_train,x_test,y_train,y_test=train_test_split(data.Message,data.spam,test_size=0.2)
x_train.head()
```

	Message
2735	I meant middle left or right?
5044	We have sent JD for Customer Service cum Accou...
1600	K.then any other special?
2723	Tunde, how are you doing. This is just wishing...
4374	Ur TONEXS subscription has been renewed and yo...

dtype: object

```
from sklearn.feature_extraction.text import CountVectorizer
featurer = CountVectorizer()
x_train_count = featurer.fit_transform(x_train.values)
```

```
x_train_count
```

```
<4457x7807 sparse matrix of type '<class 'numpy.int64'>'
        with 59145 stored elements in Compressed Sparse Row format>
```

Applying Logistic Regression

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train_count,y_train)
x_test_count = featurer.transform(x_test)
model.score(x_test_count, y_test)
```

```
0.9847533632286996
```

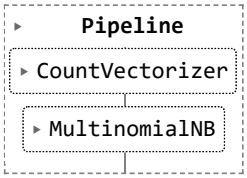
Aplying Naive Bayes Method

```
model_1 = MultinomialNB()
model_1.fit(x_train_count,y_train)
X_test_count = featurer.transform(x_test)
model_1.score(x_test_count,y_test)
```

```
0.9847533632286996
```

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

```
clf.fit(x_train,y_train)
```




```
clf.score(x_test,y_test)
```

```
0.9847533632286996
```

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

```
# Pre-trained model
pretrained_model = model
new_sentences = [
    "Your account have 100 debeted, is waiting to be collected. Simply text the password \MIX\" to 85069 to v
]

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

```
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 85069 to
```