```
#importing libraries
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

#loading the data
data = pd.read_csv('/content/spam.csv', encoding='ISO-8859-1')

#displaying first five elements
data.head()

     v1                                                 v2 Unnamed: 2
\
0   ham  Go until jurong point, crazy.. Available only ...        NaN

1   ham                      Ok lar... Joking wif u oni...        NaN

2  spam  Free entry in 2 a wkly comp to win FA Cup fina...        NaN

3   ham  U dun say so early hor... U c already then say...        NaN

4   ham  Nah I don't think he goes to usf, he lives aro...        NaN


  Unnamed: 3 Unnamed: 4
0        NaN        NaN
1        NaN        NaN
2        NaN        NaN
3        NaN        NaN
4        NaN        NaN

#displaying the last five elements
data.tail()

        v1                                                  v2 Unnamed:
2  \
5567  spam  This is the 2nd time we have tried 2 contact u...
NaN
5568   ham              Will Ì_ b going to esplanade fr home?
NaN
5569   ham  Pity, * was in mood for that. So...any other s...
NaN
5570   ham  The guy did some bitching but I acted like i'd...
NaN
5571   ham                         Rofl. Its true to its name
NaN
```

```
     Unnamed: 3 Unnamed: 4
5567        NaN        NaN
5568        NaN        NaN
5569        NaN        NaN
5570        NaN        NaN
5571        NaN        NaN
```

```python
data = data[['v1', 'v2']]

#renaming the columns
data.columns = ['label', 'text']

data.head()
```

```
  label                                               text
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...
```

```python
# converting labels to 1 for spam, 0 for not spam
data['label'] = (data['label'] == 'spam').astype(int)

data.head()
```

```
   label                                               text
0      0  Go until jurong point, crazy.. Available only ...
1      0                      Ok lar... Joking wif u oni...
2      1  Free entry in 2 a wkly comp to win FA Cup fina...
3      0  U dun say so early hor... U c already then say...
4      0  Nah I don't think he goes to usf, he lives aro...
```

```python
# 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)

# creating bow and TF-IDF representations
vectorizer_bow = CountVectorizer()
X_train_bow = vectorizer_bow.fit_transform(X_train)
X_test_bow = vectorizer_bow.transform(X_test)

vectorizer_tfidf = TfidfVectorizer()
X_train_tfidf = vectorizer_tfidf.fit_transform(X_train)
X_test_tfidf = vectorizer_tfidf.transform(X_test)

# training naive bayes classifiers
nb_bow = MultinomialNB()
nb_tfidf = MultinomialNB()
```

```
nb_bow.fit(X_train_bow, y_train)
nb_tfidf.fit(X_train_tfidf, y_train)
```

```
MultinomialNB()
```

```
#making predictions
y_pred_bow = nb_bow.predict(X_test_bow)
y_pred_tfidf = nb_tfidf.predict(X_test_tfidf)

#calculating accuracy
accuracy_bow = accuracy_score(y_test, y_pred_bow)
accuracy_tfidf = accuracy_score(y_test, y_pred_tfidf)

print(f'Accuracy (BoW): {accuracy_bow}')
print(f'Accuracy (TF-IDF): {accuracy_tfidf}')
```

```
Accuracy (BoW): 0.9838565022421525
Accuracy (TF-IDF): 0.9623318385650225
```

By implementing Bag of Words and TF-IDF representations with a Naive Bayes classifier, we achieved an accuracy rate of around 98 percent that allowed us to distinguish between spam and non-spam SMS messages. On caluculating accuracy we noticed about the model performance.It has high accuracy rate