```
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
```

```
sms_data = pd.read_csv('/content/sms_spam_dataset.csv')
```

```
sms_data['label'] = sms_data['label'].map({'spam': 1, 'ham': 0})
```

```
X_train, X_test, y_train, y_test = train_test_split(sms_data['message'], sms_data['label'], test_size=0.2, random_state=42)
```

X_train, X_test, y_train, y_test = train_test_split(sms_data['message'], sms_data['label'], test_size=0.2, random_state=42)

```
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

```
naive_bayes = MultinomialNB()
naive_bayes.fit(X_train_tfidf, y_train)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
y_pred = naive_bayes.predict(X_test_tfidf)
```

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 1.0

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         1

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2
```

```
# Import necessary libraries
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB

# Load the SMS spam dataset (example dataset)
sms_data = pd.read_csv('sms_spam_dataset.csv')

# Preprocess the data
sms_data['label'] = sms_data['label'].map({'spam': 1, 'ham': 0})

# Vectorize the text data using TF-IDF
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(sms_data['message'])
y = sms_data['label']

# Train a Naive Bayes classifier
naive_bayes = MultinomialNB()
naive_bayes.fit(X, y)

# Take input from the user
user_input = input("Enter an SMS message: ")

# Vectorize the user input
user_input_tfidf = vectorizer.transform([user_input])

# Make prediction
prediction = naive_bayes.predict(user_input_tfidf)
```

```
# Output the prediction
if prediction == 1:
    print("The message is spam.")
else:
    print("The message is not spam (ham).")
```

```
Enter an SMS message: "WINNER!! As a valued network customer you have been selected to receivea £900 prize reward! To claim call 09
The message is spam.
```