```
# Importing necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

# Load the dataset using read_csv
iris_data = pd.read_csv('Iris.csv')

# One-hot encode categorical variables
iris_data = pd.get_dummies(iris_data, columns=['Designation',
'Education', 'Marital Status', 'Field', 'Race', 'Gender', 'Country'])

# Assuming the dataset has features (X) and labels (y)
X = iris_data.drop('Salary', axis=1)  # Features variables for column
y = iris_data['Salary']  # Labels

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Initialize Naive Bayes classifier
nb_classifier = GaussianNB()

# Train the classifier
nb_classifier.fit(X_train, y_train)

# Predict on the test set
y_pred = nb_classifier.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.3821587594042684
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

# Load the dataset using read_csv
playtennis_data = pd.read_csv('heart.csv')

# One-hot encode categorical variables (but keep the target column for
the label)
playtennis_data = pd.get_dummies(playtennis_data, columns=['age',
'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg',
                                                    'thalach',
'exang', 'oldpeak', 'slope', 'ca', 'thal'],
                                 drop_first=True)  # drop_first=True
```

```
avoids multicollinearity issues

# The target column is likely 'target', so assign that to y
X = playtennis_data.drop('target', axis=1)  # Features variables
y = playtennis_data['target']  # Labels (heart disease)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Initialize Naive Bayes classifier
nb_classifier = GaussianNB()

# Train the classifier
nb_classifier.fit(X_train, y_train)

# Predict on the test set
y_pred = nb_classifier.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.8682926829268293
```