```
# Importing essential libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,precision_score,recall_score,
```

```
# File path
file_path = "/content/WA_Fn-UseC_-HR-Employee-Attrition.csv"

# Load the data
data = pd.read_csv(file_path)
```

```
print(data.head)
```

```
<bound method NDFrame.head of       Age  Attrition  BusinessTravel  DailyRate  Department  DistanceFromHome  \
0      41          1               2       1102           2                 1
1      49          0               1        279           1                 8
2      37          1               2       1373           1                 2
3      33          0               1       1392           1                 3
4      27          0               2        591           1                 2
...   ...        ...             ...        ...         ...               ...
1465   36          0               1        884           1                23
1466   39          0               2        613           1                 6
1467   27          0               2        155           1                 4
1468   49          0               1       1023           2                 2
1469   34          0               2        628           1                 8

      Education  EducationField  EmployeeCount  EmployeeNumber  ...  \
0             2               1              1               1  ...
1             1               1              1               2  ...
2             2               4              1               4  ...
3             4               1              1               5  ...
4             1               3              1               7  ...
...         ...             ...            ...             ...  ...
1465          2               3              1            2061  ...
1466          1               3              1            2062  ...
1467          3               1              1            2064  ...
1468          3               3              1            2065  ...
1469          3               3              1            2068  ...

      StandardHours  StockOptionLevel  TotalWorkingYears  \
0                80                 0                  8
1                80                 1                 10
2                80                 0                  7
3                80                 0                  8
4                80                 1                  6
...             ...               ...                ...
1465             80                 1                 17
1466             80                 1                  9
1467             80                 1                  6
1468             80                 0                 17
1469             80                 0                  6

      TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany  \
0                         0                1               6
1                         3                3              10
2                         3                3               0
3                         3                3               8
4                         3                3               2
...                     ...              ...             ...
1465                      3                3               5
1466                      5                3               7
1467                      0                3               6
1468                      3                2               9
1469                      3                4               4

      YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager  \
0                      4                        0                     5
1                      7                        1                     7
2                      0                        0                     0
3                      7                        3                     0
4                      2                        2                     2
```

```
print(data.tail)
```

```
<bound method NDFrame.tail of       Age  Attrition  BusinessTravel  DailyRate  Department  DistanceFromHome  \
0      41          1               2       1102           2                 1
1      49          0               1        279           1                 8
2      37          1               2       1373           1                 2
3      33          0               1       1392           1                 3
```

```
4      27          0               2        591           1                 2
...   ...        ...             ...        ...         ...               ...
1465   36          0               1        884           1                23
1466   39          0               2        613           1                 6
1467   27          0               2        155           1                 4
1468   49          0               1       1023           2                 2
1469   34          0               2        628           1                 8

      Education  EducationField  EmployeeCount  EmployeeNumber  ...  \
0             2               1              1               1  ...
1             1               1              1               2  ...
2             2               4              1               4  ...
3             4               1              1               5  ...
4             1               3              1               7  ...
...         ...             ...            ...             ...  ...
1465          2               3              1            2061  ...
1466          1               3              1            2062  ...
1467          3               1              1            2064  ...
1468          3               3              1            2065  ...
1469          3               3              1            2068  ...

      StandardHours  StockOptionLevel  TotalWorkingYears  \
0                80                 0                  8
1                80                 1                 10
2                80                 0                  7
3                80                 0                  8
4                80                 1                  6
...             ...               ...                ...
1465             80                 1                 17
1466             80                 1                  9
1467             80                 1                  6
1468             80                 0                 17
1469             80                 0                  6

      TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany  \
0                         0                1               6
1                         3                3              10
2                         3                3               0
3                         3                3               8
4                         3                3               2
...                     ...              ...             ...
1465                      3                3               5
1466                      5                3               7
1467                      0                3               6
1468                      3                2               9
1469                      3                4               4

      YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager  \
0                      4                        0                     5
1                      7                        1                     7
2                      0                        0                     0
3                      7                        3                     0
4                      2                        2                     2
```

```
# Handle missing values
if data.isnull().sum().sum() > 0:
    data = data.fillna(method='ffill')  # Forward fill for missing values


# Identify and encode categorical columns
categorical_cols = data.select_dtypes(include=['object']).columns
label_encoders = {}

for col in categorical_cols:
    encoder = LabelEncoder()
    data[col] = encoder.fit_transform(data[col])
    label_encoders[col] = encoder



# Create a new feature
data['TotalWorkingYearsPerJobRole'] = data['TotalWorkingYears'] / (data['NumCompaniesWorked'].replace(0, 1))



print(data)
```

```
3                80                 0                  8
4                80                 1                  6
...             ...               ...                ...
1465             80                 1                 17
1466             80                 1                  9
1467             80                 1                  6
1468             80                 0                 17
1469             80                 0                  6

      TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany  \
0                         0                1               6
1                         3                3              10
2                         3                3               0
3                         3                3               8
4                         3                3               2
...                     ...              ...             ...
1465                      3                3               5
1466                      5                3               7
1467                      0                3               6
1468                      3                2               9
1469                      3                4               4

      YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager  \
0                      4                        0                     5
1                      7                        1                     7
2                      0                        0                     0
3                      7                        3                     0
4                      2                        2                     2
...                  ...                      ...                   ...
1465                   2                        0                     3
1466                   7                        1                     7
1467                   2                        0                     3
1468                   6                        0                     8
1469                   3                        1                     2

      TotalWorkingYearsPerJobRole
0                        1.000000
1                       10.000000
2                        1.166667
3                        8.000000
4                        0.666667
...                           ...
1465                     4.250000
1466                     2.250000
1467                     6.000000
1468                     8.500000
1469                     3.000000

[1470 rows x 36 columns]
```

```
# Extract features and target variable
X = data.drop(columns=['Attrition'])  # Features
y = data['Attrition']  # Target variable



#Split the original data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)


# Further split the training data into training (60%) and validation (40%) in a 3:2 ratio
X_train_split, X_val, y_train_split, y_val = train_test_split(
    X_train, y_train, test_size=0.4, random_state=42, stratify=y_train
)



# Print the shapes of the resulting splits
print("Shapes after splitting:")
print("Training set (60%):", X_train_split.shape, y_train_split.shape)
print("Validation set (40%):", X_val.shape, y_val.shape)
print("Testing set (20% of original):", X_test.shape, y_test.shape)
```

```
Shapes after splitting:
Training set (60%): (705, 35) (705,)
Validation set (40%): (471, 35) (471,)
Testing set (20% of original): (294, 35) (294,)
```

```
# Scale numerical features
numerical_features = X.select_dtypes(include=['number']).columns
scaler = StandardScaler()

X_train_split[numerical_features] = scaler.fit_transform(X_train_split[numerical_features])
```

```
X_val[numerical_features] = scaler.transform(X_val[numerical_features])
X_test[numerical_features] = scaler.transform(X_test[numerical_features])



# Train the Naive Bayes classifier on the reduced training set
nb_classifier = GaussianNB()
nb_classifier.fit(X_train_split, y_train_split)

# Predict on the validation data
y_val_pred = nb_classifier.predict(X_val)


# Evaluate the model on the validation data
print("\n--- Validation Results ---")
print("Validation Accuracy:", accuracy_score(y_val, y_val_pred))

print("Classification Report:\n", classification_report(y_val, y_val_pred))
```

```
--- Validation Results ---
Validation Accuracy: 0.7261146496815286
Confusion Matrix:
 [[291 104]
 [ 25  51]]
Classification Report:
               precision    recall  f1-score   support

           0       0.92      0.74      0.82       395
           1       0.33      0.67      0.44        76

    accuracy                           0.73       471
   macro avg       0.62      0.70      0.63       471
weighted avg       0.83      0.73      0.76       471
```

```
# Test the model on the test set
y_test_pred = nb_classifier.predict(X_test)


# Calculate precision, recall, and F1-score for validation data
val_precision = precision_score(y_val, y_val_pred, average='binary')
val_recall = recall_score(y_val, y_val_pred, average='binary')
val_f1 = f1_score(y_val, y_val_pred, average='binary')
print(f"Validation Precision: {val_precision:.2f}")
print(f"Validation Recall: {val_recall:.2f}")
print(f"Validation F1-Score: {val_f1:.2f}")
print("Confusion Matrix:\n", confusion_matrix(y_val, y_val_pred))
```

```
Validation Precision: 0.33
Validation Recall: 0.67
Validation F1-Score: 0.44
Confusion Matrix:
 [[291 104]
 [ 25  51]]
```

```
# Calculate precision, recall, and F1-score for test data
test_precision = precision_score(y_test, y_test_pred, average='binary')
test_recall = recall_score(y_test, y_test_pred, average='binary')
test_f1 = f1_score(y_test, y_test_pred, average='binary')
print(f"Test Precision: {test_precision:.2f}")
print(f"Test Recall: {test_recall:.2f}")
print(f"Test F1-Score: {test_f1:.2f}")
print("Confusion Matrix:\n", confusion_matrix(y_test, y_test_pred))
```

```
Test Precision: 0.28
Test Recall: 0.66
Test F1-Score: 0.39
Confusion Matrix:
 [[167  80]
 [ 16  31]]
```