In [550…

```
# Import relevant libraries and modules.

import pandas as pd
from sklearn import naive_bayes
from sklearn import model_selection
from sklearn import metrics
```

In [552…

```
data = pd.read_csv(r'C:\Users\HP\Desktop\Advance Data Analyst\6. The nuts and bolts of ML\2. Module 2\1. PACE in ML\Files\extracted_nba_players_data.csv')
data.head(10)
```

Out[552…

	fg	3p	ft	reb	ast	stl	blk	tov	target_5yrs	total_points	efficiency
0	34.7	25.0	69.9	4.1	1.9	0.4	0.4	1.3	0	266.4	0.270073
1	29.6	23.5	76.5	2.4	3.7	1.1	0.5	1.6	0	252.0	0.267658
2	42.2	24.4	67.0	2.2	1.0	0.5	0.3	1.0	0	384.8	0.339869
3	42.6	22.6	68.9	1.9	0.8	0.6	0.1	1.0	1	330.6	0.491379
4	52.4	0.0	67.4	2.5	0.3	0.3	0.4	0.8	1	216.0	0.391304
5	42.3	32.5	73.2	0.8	1.8	0.4	0.0	0.7	0	277.5	0.324561
6	43.5	50.0	81.1	2.0	0.6	0.2	0.1	0.7	1	409.2	0.605505
7	41.5	30.0	87.5	1.7	0.2	0.2	0.1	0.7	1	273.6	0.553398
8	39.2	23.3	71.4	0.8	2.3	0.3	0.0	1.1	0	156.0	0.242424
9	38.3	21.4	67.8	1.1	0.3	0.2	0.0	0.7	0	155.4	0.435294

In [554…

```
# Define the y (target) variable.

y = extracted_data['target_5yrs']

# Define the X (predictor) variables.

X = extracted_data.drop('target_5yrs', axis = 1)
```

In [556…

```
# Display the first 10 rows of your target data.

y.head(10)
```

Out[556…

```
0    0
1    0
2    0
3    1
4    1
5    0
6    1
7    1
8    0
9    0
Name: target_5yrs, dtype: int64
```

In [558…

```
# Display the first 10 rows of your predictor variables.

X.head(10)
```

Out[558…

	fg	3p	ft	reb	ast	stl	blk	tov	total_points	efficiency
0	34.7	25.0	69.9	4.1	1.9	0.4	0.4	1.3	266.4	0.270073
1	29.6	23.5	76.5	2.4	3.7	1.1	0.5	1.6	252.0	0.267658
2	42.2	24.4	67.0	2.2	1.0	0.5	0.3	1.0	384.8	0.339869
3	42.6	22.6	68.9	1.9	0.8	0.6	0.1	1.0	330.6	0.491379
4	52.4	0.0	67.4	2.5	0.3	0.3	0.4	0.8	216.0	0.391304
5	42.3	32.5	73.2	0.8	1.8	0.4	0.0	0.7	277.5	0.324561
6	43.5	50.0	81.1	2.0	0.6	0.2	0.1	0.7	409.2	0.605505
7	41.5	30.0	87.5	1.7	0.2	0.2	0.1	0.7	273.6	0.553398
8	39.2	23.3	71.4	0.8	2.3	0.3	0.0	1.1	156.0	0.242424
9	38.3	21.4	67.8	1.1	0.3	0.2	0.0	0.7	155.4	0.435294

In [560…

```
# Perform the split operation on your data.
# Assign the outputs as follows: X_train, X_test, y_train, y_test.

X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, test_size=0.25, random_state=0)
```

In [562…

```
# Print the shape (rows, columns) of the output from the train-test split.

# Print the shape of X_train.

print(X_train.shape)

# Print the shape of X_test.

print(X_test.shape)

# Print the shape of y_train.

print(y_train.shape)

# Print the shape of y_test.

print(y_test.shape)
```

```
(1005, 10)
(335, 10)
(1005,)
(335,)
```

In [564…

```
# Assign `nb` to be the appropriate implementation of Naive Bayes.

nb = naive_bayes.GaussianNB()

# Fit the model on your training data.

nb.fit(X_train, y_train)

# Apply your model to predict on your test data. Call this "y_pred".

y_pred = nb.predict(X_test)
```

In [566…

```
# Print your accuracy score.

print('accuracy score:'), print(metrics.accuracy_score(y_test, y_pred))

# Print your precision score.

print('precision score:'), print(metrics.precision_score(y_test, y_pred))

# Print your recall score.

print('recall score:'), print(metrics.recall_score(y_test, y_pred))

# Print your f1 score.

print('f1 score:'), print(metrics.f1_score(y_test, y_pred))
```

```
accuracy score:
0.6895522388059702
precision score:
0.8405797101449275
recall score:
0.5858585858585859
f1 score:
0.6904761904761905
```

Out[566…

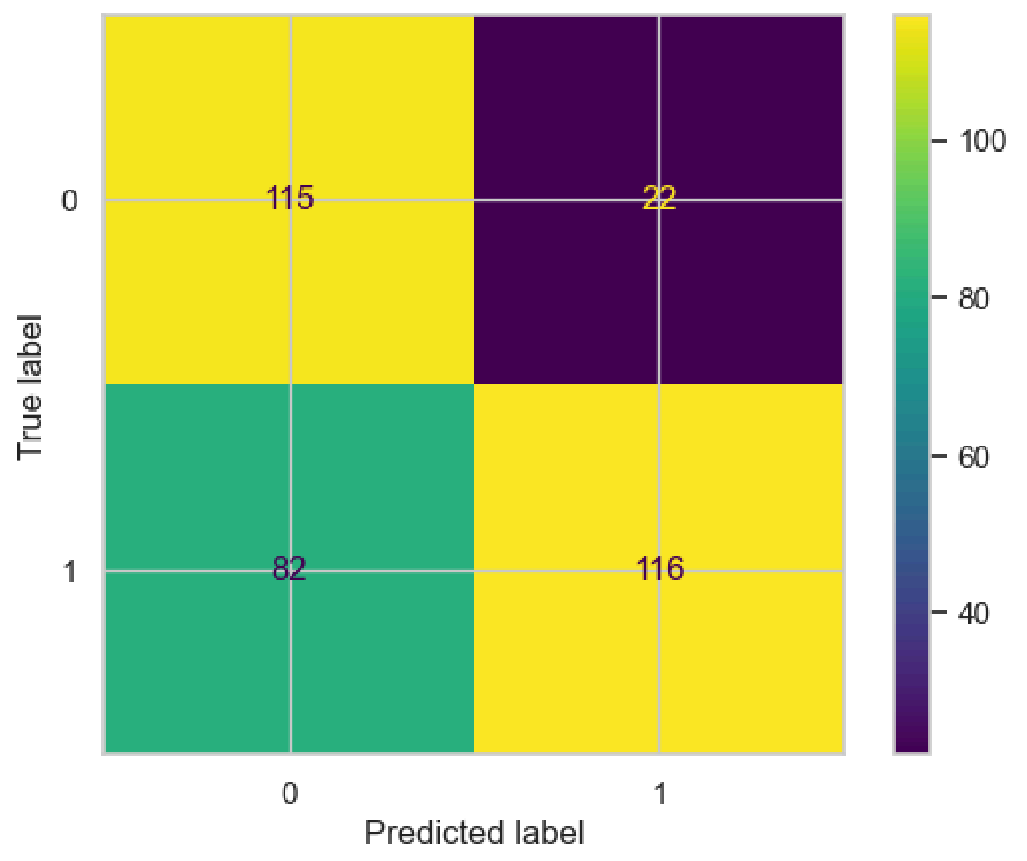
```
(None, None)
```

In [568…

```
# Construct and display your confusion matrix.

# Construct the confusion matrix for your predicted and test values.

cm = metrics.confusion_matrix(y_test, y_pred)

# Create the display for your confusion matrix.

disp = metrics.ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=nb.classes_)

# Plot the visual in-line.

disp.plot()
```

Out[568…

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x27bf18d2a50>
```



In [ ]:

```
# key takeaways

# The evaluation of the model is important to inform if the model has delivered accurate predictions.
# Splitting the data was important for ensuring that there was new data for the model to test its predictive performance.
# Each metric provided an evaluation from a different standpoint, and accuracy alone was not a strong way to evaluate the model.
# Effective assessments balance the true/false positives versus true/false negatives through the confusion matrix and F1 score.

# How would you present your results to your team?

# Showcase the data used to create the prediction and the performance of the model overall.
# Review the sample output of the features and the confusion matrix to indicate the model's performance.
# Highlight the metric values, emphasizing the F1 score.

# How would you summarize your findings to stakeholders?
# The model created provides some value in predicting an NBA player's chances of playing for five years or more.
# Notably, the model performed better at predicting true positives than it did at predicting true negatives.
# In other words, it more accurately identified those players who will likely play for more than five years than it did those who likely will not.
```