In [103…

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
sns.set_theme(color_codes = True)
```

In [104…

```
df = pd.read_csv('NaiveBayesData.csv')
df.head()
```

Out[104…

	Feature_1	Feature_2	Target
0	0.733246	-1.431007	0
1	0.656043	0.842841	1
2	0.537983	-2.056550	0
3	1.289308	-0.079023	1
4	0.720124	0.977237	1

In [105…

```
df.isnull().sum()
```

Out[105…

```
Feature_1    0
Feature_2    0
Target       0
dtype: int64
```

In [106…

```
df.describe()
```

Out[106…

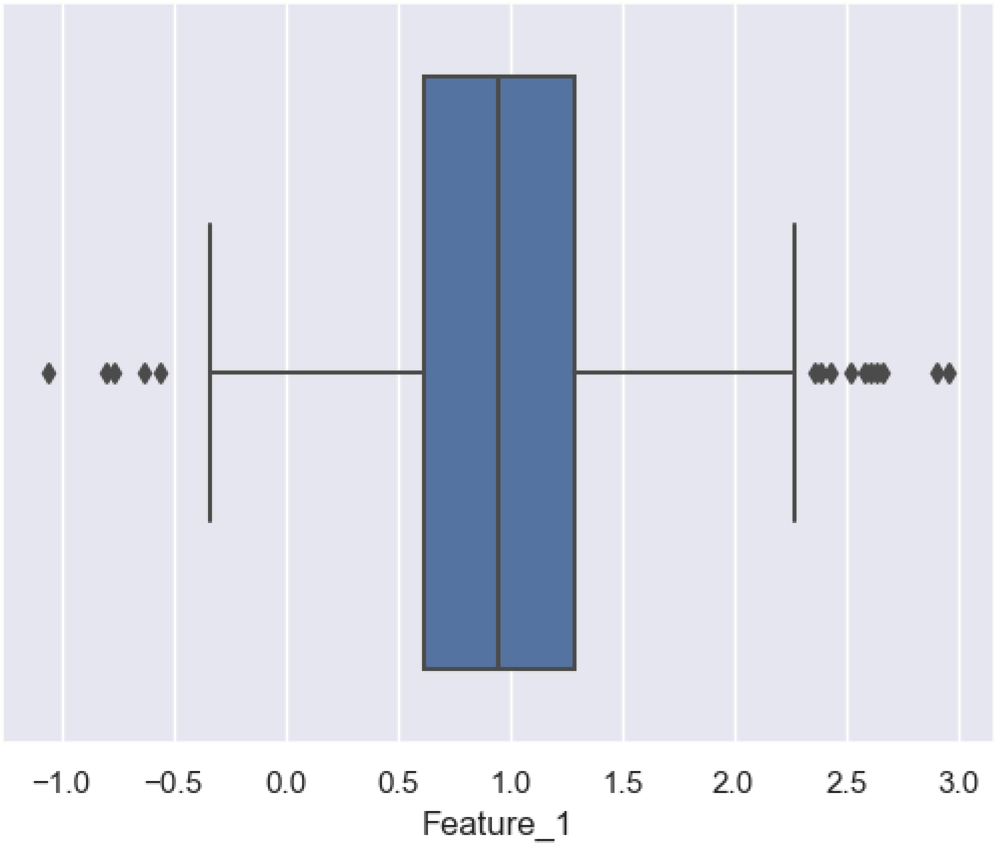
	Feature_1	Feature_2	Target
count	300.000000	300.000000	300.000000
mean	0.967604	0.007175	0.500000
std	0.607250	1.229816	0.500835
min	-1.062240	-2.823751	0.000000
25%	0.621191	-1.089332	0.000000
50%	0.952689	-0.108417	0.500000
75%	1.289528	1.045620	1.000000
max	2.961783	2.885964	1.000000

In [107…

```
sns.boxplot(x=df["Feature_1"])
```

```
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

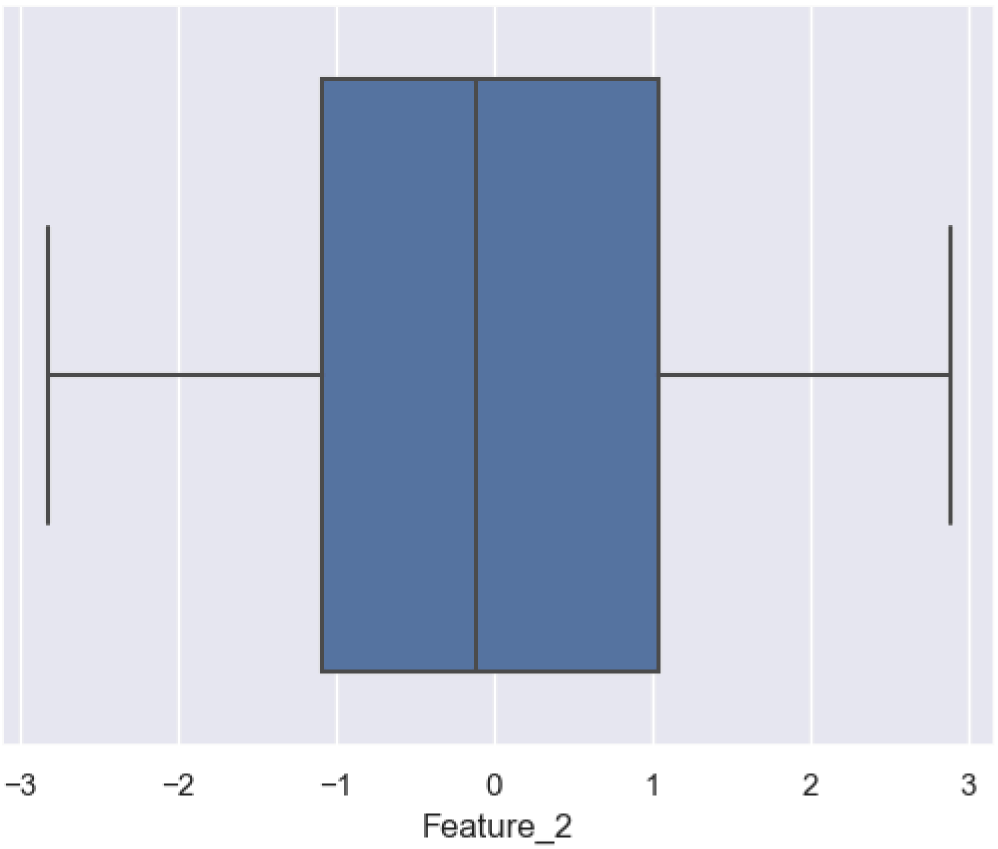
```
Out[107…   <Axes: xlabel='Feature_1'>
```



```
In [108…   sns.boxplot(x=df["Feature_2"])
```

```
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

```
Out[108…   <Axes: xlabel='Feature_2'>
```

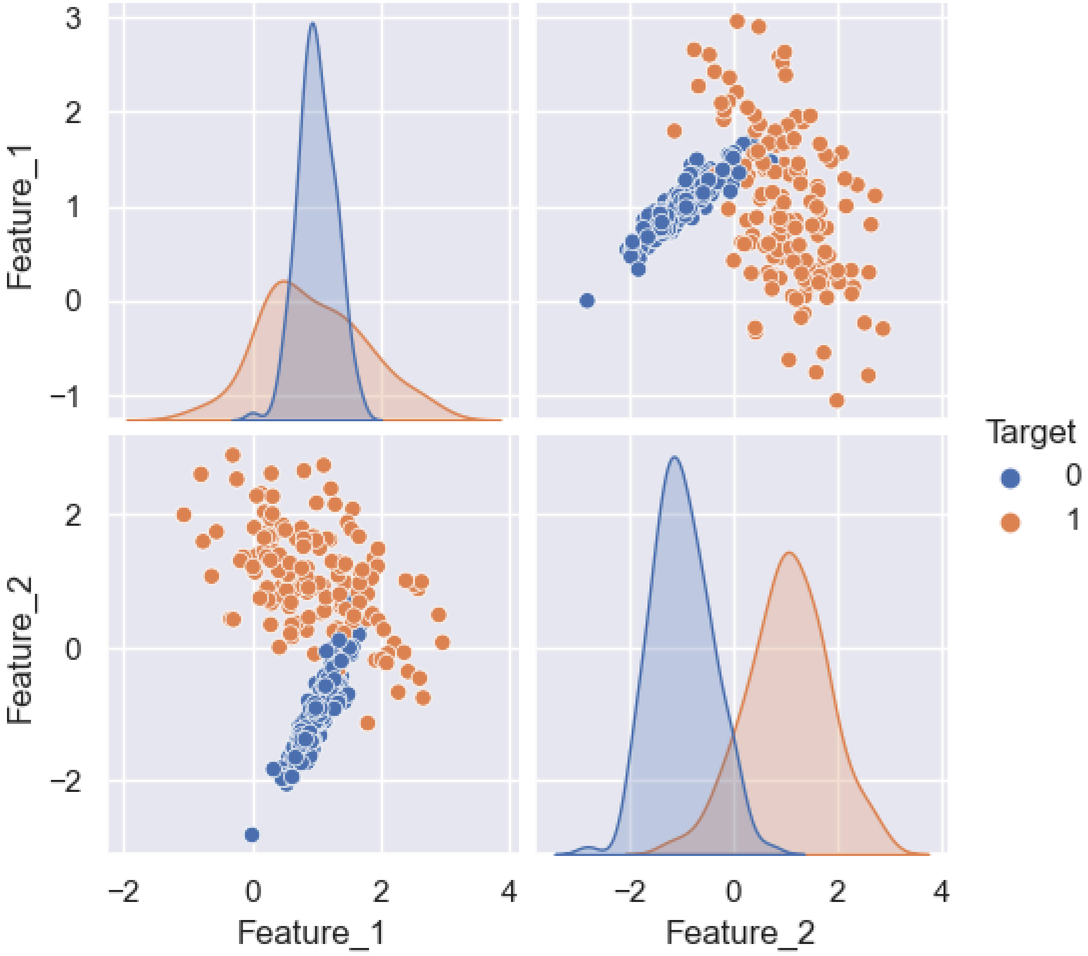


In [109…

```
sns.pairplot(df, kind="scatter", hue="Target")
plt.show()
```

```
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

```
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

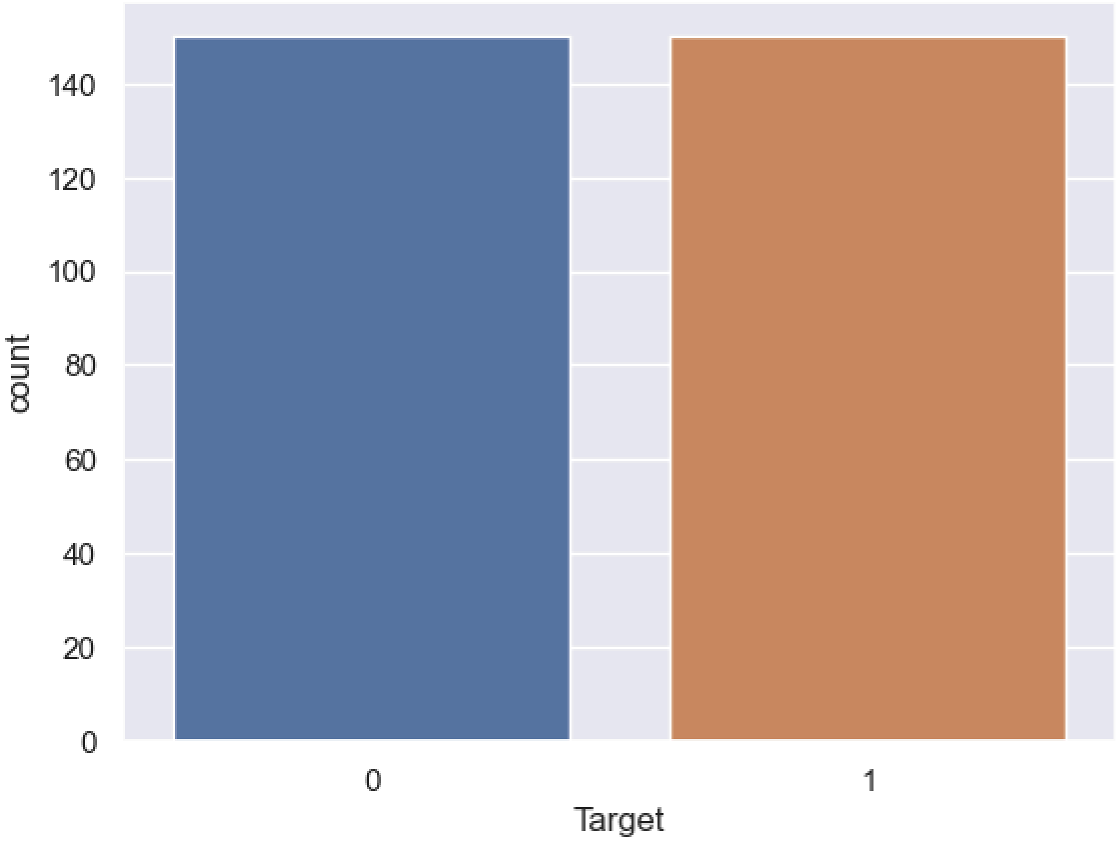


In [110…

```
sns.countplot(data=df, x='Target')
```

```
print(df.Target.value_counts())
```

```
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sjkar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
Target
0    150
1    150
Name: count, dtype: int64
```



In [111…

```
class_counts = df['Target'].value_counts()

class_proportion = class_counts / len(df['Target'])

print(class_proportion)
```

```
Target
0    0.5
1    0.5
Name: count, dtype: float64
```

*Count is same therfore no need of resampling

In [112…

```
X = df.drop('Target', axis=1)
y = df['Target']
```

In [113…

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

Training Gaussian Naive Bayes model

In [114…

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
```

Out[114…

```
▾ GaussianNB

GaussianNB()
```

Prediction and Evaluation

In [115…

```
y_pred = gnb.predict(X_test)
```

In [116…

```
from sklearn.metrics import accuracy_score
y_pred = gnb.predict(X_test)
print("Accuracy Score :", round(accuracy_score(y_test, y_pred)*100 ,2), "%")
```

```
Accuracy Score : 98.33 %
```

In [117…

```
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
print('F-1 Score : ',(f1_score(y_test, y_pred)))
print('Precision Score : ',(precision_score(y_test, y_pred)))
print('Recall Score : ',(recall_score(y_test, y_pred)))
```

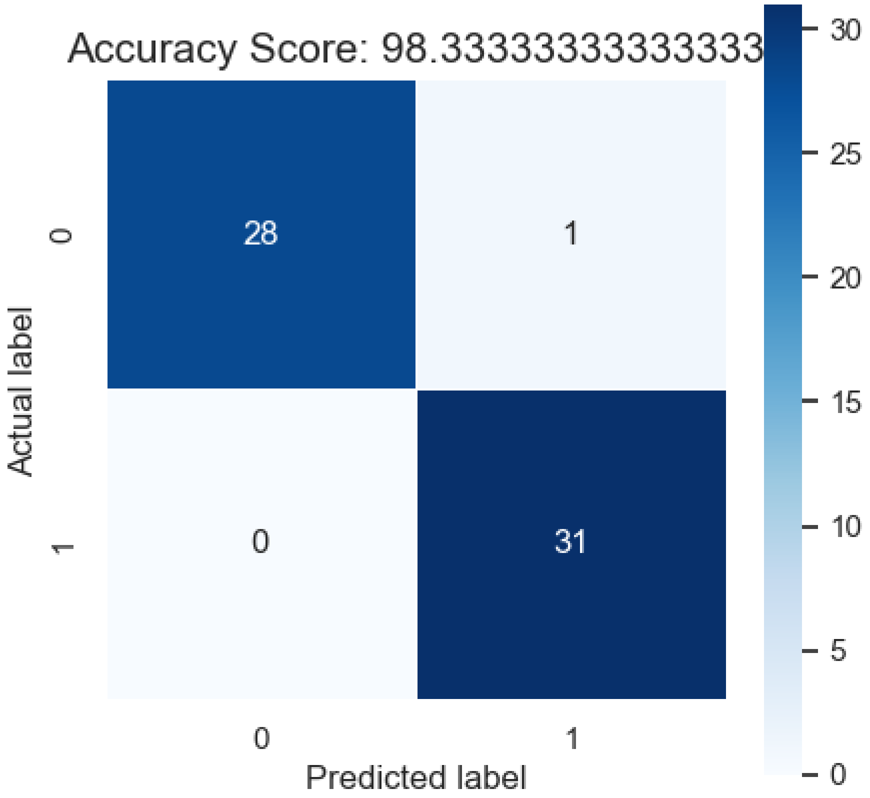
```
F-1 Score :  0.9841269841269841
Precision Score :  0.96875
Recall Score :  1.0
```

In [118…

```
from sklearn.metrics import classification_report, confusion_matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True,square = True,  cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(gnb.score(X_test, y_test)*100)
plt.title(all_sample_title, size = 15)
```

Out[118…

```
Text(0.5, 1.0, 'Accuracy Score: 98.33333333333333')
```



In [119…

```python
from sklearn.model_selection import cross_val_score

cv_scores = cross_val_score(gnb, X, y, cv=5)

print("Cross-validated accuracy scores:", cv_scores)
print("Average cross-validated accuracy:", cv_scores.mean())
```

```
Cross-validated accuracy scores: [0.91666667 0.95       0.96666667 0.96666667 0.95
]
Average cross-validated accuracy: 0.95
```

Sample data evaluation

In [120…

```python
new_sample = [[0.5, -1.2]]
```

In [121…

```python
predicted_class = gnb.predict(new_sample)
print("\nPredicted class for new sample data point:", predicted_class[0])
```

```
Predicted class for new sample data point: 0
```

```
C:\Users\sjkar\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does
not have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```