Experiment 2: Study the use of metrics and scoring to quantify the quality of a model's prediction and plot the graphs

Theory:

1. **matplotlib**

Matplotlib is a Python 2D plotting library, and with this library, we need to import a sub-library pyplot. This library is used to plot any type of charts in Python for the code. There are five key plots that are used for data visualization. They are Bar graph (.bar()), Scatter plot (.scatter()), Histogram (.hist()), Area plot (.fill_between()), and Pie plot (.pie()). The plot() function is used to draw points (markers) in a diagram. By default, the plot() function draws a line from point to point.

Implementation:

1. **sklearn.metrics.accuracy_score**

Syntax:

```
sklearn.metrics.accuracy_score(y_true, y_pred) ¶
```

Gives accuracy classification score.

2. **sklearn.metrics.precision_score**

Syntax:

```
sklearn.metrics.precision_score(y_true, y_pred, labels=None, pos_label=1) ¶
```

Compute the precision.

3. **sklearn.metrics.recall_score**

Syntax:

```
sklearn.metrics.recall_score(y_true, y_pred, labels=None, pos_label=1) ¶
```

Compute the recall.

4. **sklearn.metrics.f1_score**

Syntax:

```
sklearn.metrics.f1_score(y_true, y_pred, *, labels=None, pos_label=1) ¶
```

Compute the F1 score, also known as balanced F-score or F-measure.

5. **sklearn.metrics.confusion_matrix**

Syntax:

```
sklearn.metrics.confusion_matrix(y_true, y_pred, labels=None) ¶
```

Compute confusion matrix to evaluate the accuracy of a classification.

Conclusion: In this way, we understood the metrics to evaluate the performance of the trained model. We also studied the matplotlib library to visualize the data.