

## PRACTICAL-4

Write a program to calculate values of different Web IR performance matrices. Create a dataset for the same and model values should be predicted automatically (can be given by probability measure). Draw confusion matrix and calculate TP,TN, FP,FN for the dataset. Then value of parameters should be calculated.

### **Solution:**

The website collects various student details related to placement, including CGPA, age, gender, and work experience.

After entering these details, the user provides the actual placement status (whether the student was placed or not).

The model then generates a random prediction for placement.

Based on the actual and predicted values, the confusion matrix and performance metrics (such as Precision, Recall, F1-Score, and Accuracy) are updated dynamically.

Here's screenshot of the live working website.

The screenshot shows a web browser window with the title 'Web IR Performance Metrics Calculator'. The page has a light gray background. On the left, there is a form titled 'Enter Student Details' with the following fields: 'CGPA:' with a text input containing '4', 'Age:' with a text input containing '19', 'Gender:' with a dropdown menu showing 'Male', 'Work Experience (Years):' with a text input containing '9', and 'Placed (1: Yes, 0: No):' with a dropdown menu showing 'No'. A green 'Submit & Calculate' button is at the bottom of the form. To the right of the form, there is a 'Confusion Matrix' table and a 'Performance Metrics' section. The 'Confusion Matrix' table has two columns: 'Predicted Positive' and 'Predicted Negative', and two rows: 'Actual Positive' and 'Actual Negative'. The values are: Actual Positive (26, 25), Actual Negative (11, 22). The 'Performance Metrics' section lists: Precision: 0.70, Recall: 0.51, F1-Score: 0.59, and Accuracy: 0.57.

	Predicted Positive	Predicted Negative
Actual Positive	26	25
Actual Negative	11	22

**Performance Metrics**

Precision: 0.70  
Recall: 0.51  
F1-Score: 0.59  
Accuracy: 0.57

# CODE

## index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Web IR Performance Metrics</title>
  <link rel="stylesheet" href="styles.css">
</head>

<body>
  <h2>Web IR Performance Metrics Calculator</h2>

  <form id="userForm">
    <h3>Enter Student Details</h3>
    <label for="cgpa">CGPA:</label>
    <input type="number" id="cgpa" step="0.1" min="0" max="10"
required><br>

    <label for="age">Age:</label>
    <input type="number" id="age" min="18" max="60" required><br>

    <label for="gender">Gender:</label>
    <select id="gender">
      <option value="Male">Male</option>
      <option value="Female">Female</option>
      <option value="Other">Other</option>
    </select><br>

    <label for="experience">Work Experience (Years):</label>
    <input type="number" id="experience" min="0" max="20" required><br>

    <label for="actualPlacement">Placed (1: Yes, 0: No):</label>
    <select id="actualPlacement">
      <option value="1">Yes</option>
      <option value="0">No</option>
    </select><br>

    <button type="submit" class="generateButton">Submit &
Calculate</button>
  </form>

  <div class="container">
    <h3>Confusion Matrix</h3>
```

```

<table>
  <tr>
    <th></th>
    <th>Predicted Positive</th>
    <th>Predicted Negative</th>
  </tr>
  <tr>
    <th>Actual Positive</th>
    <td id="tp">0</td>
    <td id="fn">0</td>
  </tr>
  <tr>
    <th>Actual Negative</th>
    <td id="fp">0</td>
    <td id="tn">0</td>
  </tr>
</table>

<h3>Performance Metrics</h3>
<p>Precision: <span id="precision">0</span></p>
<p>Recall: <span id="recall">0</span></p>
<p>F1-Score: <span id="f1score">0</span></p>
<p>Accuracy: <span id="accuracy">0</span></p>
</div>

<script src="index.js"></script>
</body>
</html>

```

index.js

```

document.getElementById("userForm").addEventListener("submit", function
(event) {
  event.preventDefault();

  let actualPlacement =
parseInt(document.getElementById("actualPlacement").value);

  let predictedPlacement = Math.random() > 0.5 ? 1 : 0;

  let tp = parseInt(document.getElementById("tp").textContent);
  let tn = parseInt(document.getElementById("tn").textContent);
  let fp = parseInt(document.getElementById("fp").textContent);
  let fn = parseInt(document.getElementById("fn").textContent);

```

```

    if (actualPlacement === 1 && predictedPlacement === 1) {
        tp++;
    } else if (actualPlacement === 1 && predictedPlacement === 0) {
        fn++;
    } else if (actualPlacement === 0 && predictedPlacement === 1) {
        fp++;
    } else if (actualPlacement === 0 && predictedPlacement === 0) {
        tn++;
    }

    document.getElementById("tp").textContent = tp;
    document.getElementById("tn").textContent = tn;
    document.getElementById("fp").textContent = fp;
    document.getElementById("fn").textContent = fn;

    let precision = tp + fp === 0 ? 0 : (tp / (tp + fp)).toFixed(2);
    let recall = tp + fn === 0 ? 0 : (tp / (tp + fn)).toFixed(2);
    let f1score = (precision == 0 || recall == 0) ? 0 : ((2 * precision *
recall) / (parseFloat(precision) + parseFloat(recall))).toFixed(2);
    let accuracy = ((tp + tn) / (tp + tn + fp + fn)).toFixed(2);

    document.getElementById("precision").textContent = precision;
    document.getElementById("recall").textContent = recall;
    document.getElementById("f1score").textContent = f1score;
    document.getElementById("accuracy").textContent = accuracy;
});

```

styles.css

```

body {
    font-family: Arial, sans-serif;
    text-align: center;
    margin: 20px;
    background-color: #f4f4f4;
}

h2,
h3 {
    color: #333;
}

form {
    background: white;
    padding: 15px;
    border-radius: 8px;
    display: inline-block;

```

```
        margin-bottom: 20px;
    }

    label {
        font-weight: bold;
    }

    input,
    select {
        width: 80%;
        padding: 5px;
        margin: 5px 0;
        display: block;
        text-align: center;
    }

    button {
        background: #28a745;
        color: white;
        padding: 8px 12px;
        border: none;
        cursor: pointer;
        margin-top: 10px;
    }

    button:hover {
        background: #218838;
    }

    .container {
        background: white;
        padding: 15px;
        border-radius: 8px;
        display: inline-block;
    }

    table {
        width: 100%;
        border-collapse: collapse;
        margin: 10px 0;
    }

    th,
    td {
        border: 1px solid black;
        padding: 8px;
        text-align: center;
    }
}
```