

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

INTELIGENCIA ARTIFICIAL 1

Segundo Semestre 2024



Manual de técnico

Nombre	Carnet
Jorge Mario Cano Blanco	201902128

Métodos Utilizados

Métodos utilizados para el desarrollo del proyecto utilizando javascript y la librería de tytus para utilizar Machine Learning.

Método de Carga de CSV que se utiliza para verificar si en el inputfile está el archivo correcto si no es así da un error y le pide al usuario que suba un archivo correcto.

```
function CargarCSV() {  
  const fileInput = document.getElementById('fileInput');  
  const file = fileInput.files[0];  
  
  if (!file) {  
    alert('Por favor, selecciona un archivo CSV.');    return;  
  }  
  
  const reader = new FileReader();  
  reader.onload = function (event) {  
    const csvData = event.target.result;  
    processCSV(csvData);  
  };  
  reader.readAsText(file);  
}
```

Método para procesar el csv donde hace un Split donde hayan saltos de línea y después un Split para las comas después separa el archivo en eje x y eje y para que se pueda empezar a utilizar el csv.

```
function processCSV(data) {  
  const lines = data.split('\n');  
  const headers = lines[0].split(',').map(header => header.trim());  
  xTrain = [];  
  yTrain = [];  
  
  for (let i = 1; i < lines.length; i++) {  
    const line = lines[i].split(',');  
    if (line.length === 2) {  
      xTrain.push(parseFloat(line[0]));  
      yTrain.push(parseFloat(line[1]));  
    }  
  }  
  document.getElementById("log").innerHTML += '<br>CSV cargado exitosamente.';  
  
  displayCSVPreview(headers, lines.slice(1, 6));  
}
```

Método para mostrar las primeras filas del csv que se subió en una tabla para que el usuario vea que si tiene los datos que el necesita.

```
function displayCSVPreview(headers, rows) {
  const table = document.getElementById("csvPreview");
  table.innerHTML = ''; // Limpiar contenido previo

  // Crear la fila de encabezados
  const headerRow = document.createElement('tr');
  headers.forEach(header => {
    const th = document.createElement('th');
    th.textContent = header;
    headerRow.appendChild(th);
  });
  table.appendChild(headerRow);

  // Agregar las primeras 5 filas
  rows.forEach(row => {
    const dataRow = document.createElement('tr');
    const cells = row.split(',');
    cells.forEach(cell => {
      const td = document.createElement('td');
      td.textContent = cell.trim();
      dataRow.appendChild(td);
    });
    table.appendChild(dataRow);
  });
}
```

Método para parsear el csv donde hace dos splits para buscar los saltos de línea y las comas después verifica que tenga una fila de encabezados y una fila de datos para el correcto funcionamiento.

```
function parseCSV(contents) {
  const rows = contents.trim().split("\n");
  const headers = rows[0].split(",");
  const parsedData = [];

  if (rows.length < 2) {
    alert("El archivo CSV debe contener al menos una fila de datos.");
    return null;
  }

  for (let i = 1; i < rows.length; i++) {
    const values = rows[i].split(",").map(Number);

    if (values.length === 3 && !values.some(isNaN)) {
      parsedData.push(values); // Agregar solo los datos
    } else {
      alert("Error: Asegúrate de que el archivo CSV tenga tres columnas numéricas.");
      return null;
    }
  }

  return parsedData;
}
```

Método para dibujar la gráfica de regresión lineal donde jala los datos del csv que ya están en las variables de los encabezados.

```

function drawChart(xTrain, yTrain, yPredict) {
  const dataArray = [['X', 'Y Real', 'Y Predicción']];
  for (let i = 0; i < xTrain.length; i++) {
    dataArray.push([xTrain[i], yTrain[i], yPredict[i]]);
  }

  const data = google.visualization.arrayToDataTable(dataArray);

  const options = {
    title: 'Regresión Lineal',
    seriesType: 'scatter',
    series: {1: {type: 'line'}}},
    hAxis: { title: 'X' },
    vAxis: { title: 'Y' }
  };

  const chart = new google.visualization.ComboChart(document.getElementById('chart_div'));
  chart.draw(data, options);
}

```

Método para entrenar modelo de regresión lineal que se llaman a los encabezados de x y y y después solo se utilizan los métodos de la librería tytus.

```

function EntrenarModelo() {
  document.getElementById("log").innerHTML = "";
  const model = document.getElementById('modelSelect').value;

  if (model === 'regresionLineal') {
    CargarCSV();
    linear = new LinearRegression();
    linear.fit(xTrain, yTrain);
    document.getElementById("log").innerHTML += '<br>Modelo de Regresión Lineal entrenado.<br>';
  }
}

```

```

document.getElementById("log").innerHTML += `<br>Modelo de Regresión Lineal Entrenado.<br>`
} else if (model === 'regresionPolinomial') {
  CargarCSV();
  polynomial = new PolynomialRegression();

  // Entrenamiento y predicción para grados 2, 3 y 4
  polynomial.fit(xTrain, yTrain, 2);
  yPredictDegree2 = polynomial.predict(xTrain);
  r2 = polynomial.getError();

  polynomial.fit(xTrain, yTrain, 3);
  yPredictDegree3 = polynomial.predict(xTrain);
  r3 = polynomial.getError();

  polynomial.fit(xTrain, yTrain, 4);
  yPredictDegree4 = polynomial.predict(xTrain);
  r4 = polynomial.getError();

  // Redondear valores
  [yPredictDegree2, yPredictDegree3, yPredictDegree4].forEach(arr => {
    for (let i = 0; i < arr.length; i++) {
      arr[i] = Number(arr[i].toFixed(2));
    }
  });

  document.getElementById("log").innerHTML += `
<div class="log-entry">
  <h3>Modelo de Regresión Polinomial Entrenado</h3>
  <p><strong>X Train:</strong> ${xTrain.join(', ')}</p>

```

Predicciones para la polinomial donde se llaman a los ejes de x y después se llama al método de predict de la librería para que este haga sus predicciones.

```

function predictPolynomial() {
  if (polynomial) {
    const predictArray = xTrain;
    yPredictDegree2 = polynomial.predict(predictArray);
    yPredictDegree3 = polynomial.predict(predictArray);
    yPredictDegree4 = polynomial.predict(predictArray);
    document.getElementById("log").innerHTML += '<br>Predicción Polinomial realizada.';
  } else {
    alert('Primero entrena el modelo de regresión polinomial antes de predecir.');
```

Función donde se llama a la predicción polinomial y también se hace la predicción para la regresión lineal para que ya se las muestre al usuario y el las pueda ver.

```
function predictModel() {  
  const model = document.getElementById('modelSelect').value;  
  if (model === 'regresionPolinomial') {  
    predictPolynomial();  
  } else {  
    if (linear) {  
      yPredict = linear.predict(xTrain);  
      document.getElementById("log").innerHTML += '<br>Predicción realizada.<br>Y Predict: ' + yPredict;  
    } else {  
      alert('Primero entrena el modelo antes de predecir.');    }  
  }  
}
```