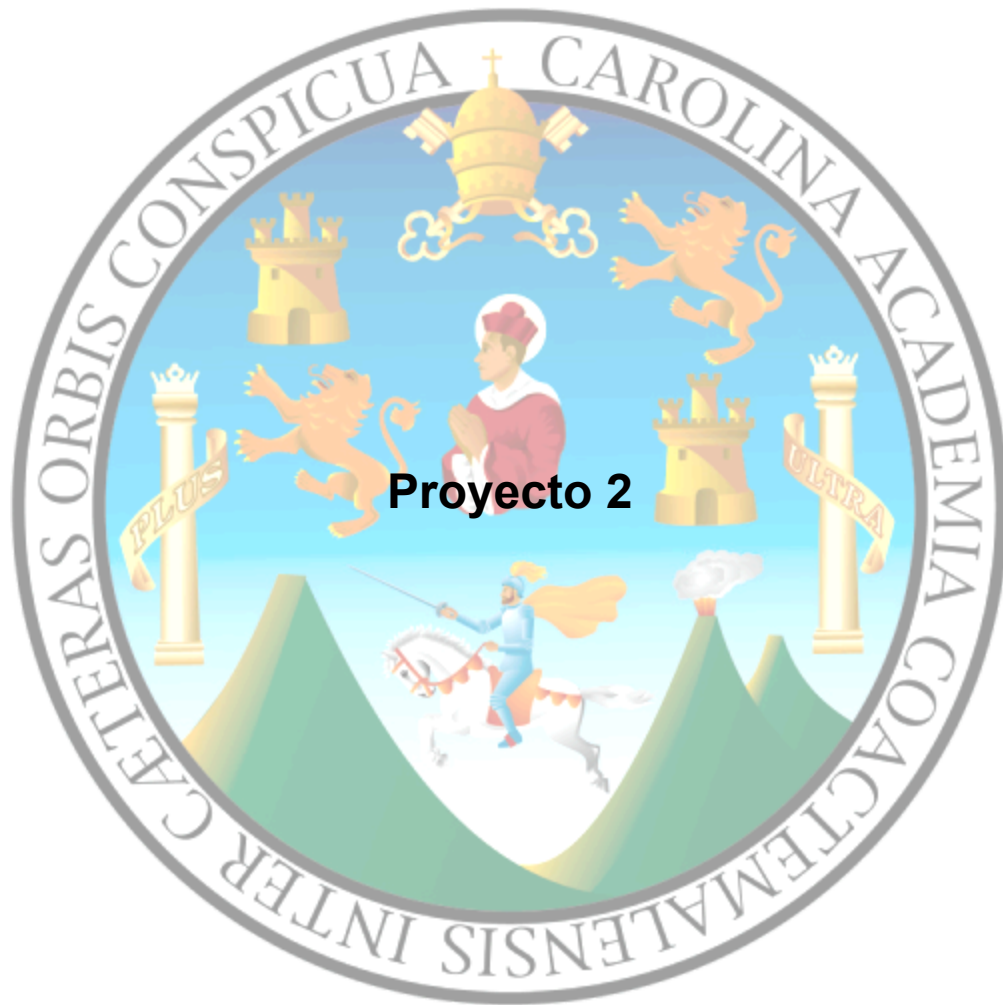


Universidad De San Carlos De Guatemala
Laboratorio Inteligencia Artificial 1
Ing. Luis Espino
Aux. Erick Sandoval
Segundo Semestre 2024



Proyecto 2

NOMBRES

Eddy Fernando Díaz Galindo

CARNET:

201906558

OBJETIVOS

General: Desarrollar una interfaz web interactiva que permita la carga y procesamiento de conjuntos de datos en formato CSV utilizando la biblioteca Tytus.js, implementando algoritmos de regresión lineal y polinómica para realizar predicciones y visualizar los resultados en gráficos generados con Google Charts.

Objetivos Específicos

1. Crear funcionalidades que permitan a los usuarios seleccionar un archivo CSV, validar su formato y extraer los datos necesarios para el entrenamiento de los modelos de regresión, asegurando la correcta manipulación de los datos de entrada a través de la biblioteca Tytus.js para facilitar la gestión y almacenamiento eficiente de los datos.
2. Desarrollar algoritmos de regresión lineal y polinómica que procesen los datos cargados, realicen predicciones y presenten los resultados en tablas y gráficos interactivos, permitiendo a los usuarios comprender el rendimiento del modelo y los resultados obtenidos, integrando visualizaciones generadas con Google Charts.

MANUAL DE USO DEL CÓDIGO HTML Y JAVASCRIPT

Este proyecto presenta una interfaz para ejecutar algoritmos de regresión lineal y polinómica a partir de datos cargados desde un archivo CSV. La estructura HTML incluye un encabezado que establece el título y la descripción del proyecto, seguido de un contenedor principal que utiliza Bootstrap para el diseño. Los usuarios pueden seleccionar el algoritmo deseado mediante un menú desplegable, y cargar un archivo CSV a través de un campo de entrada, que activará la carga de datos al seleccionarlo.

La interfaz incluye un panel para ingresar parámetros específicos de la regresión polinómica, que se muestra solo si se selecciona ese algoritmo. También cuenta con botones para predecir y graficar resultados, que están deshabilitados hasta que se carguen los datos. Dos tablas permiten visualizar los datos cargados y los resultados de las predicciones, mientras que un contenedor dedicado muestra gráficos generados por Google Charts. La interacción entre la interfaz y el script de JavaScript `script.js` permite procesar los datos, realizar predicciones y visualizar resultados de manera dinámica.

El uso efectivo de la aplicación requiere acceso a Internet para cargar las bibliotecas externas, así como un archivo CSV bien formado que incluya las columnas necesarias (`XTrain` y `YTrain`). La documentación proporciona una visión general clara de la estructura y funcionalidad de la interfaz, facilitando el entendimiento del flujo de datos y la visualización en el contexto del aprendizaje automático.

1. Variables Globales

- **xTrain**: Arreglo que almacena los datos de entrada para el entrenamiento del modelo.
- **yTrain**: Arreglo que almacena los resultados correspondientes a **xTrain**.
- **yPredict**: Arreglo que almacenará las predicciones realizadas por el modelo.
- **dataRows**: Almacena las filas de datos extraídas de un archivo CSV.
- **dataGrades**: Arreglo para almacenar grados u otros parámetros que afectan a la regresión.
- **selectedAlgorithm**: Cadena que indica el algoritmo de regresión seleccionado, inicializado en "linear".

2. Funciones

updateAlgorithm()

Esta función se encarga de actualizar el algoritmo de regresión seleccionado basado en la entrada del usuario.

Uso:

- Se llama automáticamente cuando el usuario selecciona un nuevo algoritmo en un elemento `<select>` con id "algorithmSelect".
- Resetea el campo de entrada para grados y habilita/deshabilita botones de predicción y graficación.
- Muestra u oculta parámetros específicos de la regresión polinómica.

loadCSV()

Carga un archivo CSV y procesa su contenido para extraer datos.

Uso:

- Llama a esta función al hacer clic en un botón que invoca la carga de un archivo.
- Resetea los arreglos globales y utiliza un objeto `FileReader` para leer el archivo.
- Si se carga correctamente, llama a `processCSVData(csvData)` para procesar el contenido y `populateTable(dataRows)` para mostrar los datos en una tabla.

grades()

Agrega grados o valores ingresados por el usuario a la variable `dataGrades`.

Uso:

- Se invoca al hacer clic en un botón relacionado con la carga de grados.
- Valida el formato de entrada (debe ser una lista de números enteros separados por comas) antes de agregarlos a `dataGrades`.

processCSVData(csvData)

Procesa el contenido de un archivo CSV y extrae datos relevantes.

Uso:

- Llama a esta función dentro de `loadCSV()` para procesar el texto CSV.

- Extrae los encabezados y valida que contenga las columnas necesarias (XTrain y YTrain).
- Almacena los valores de **XTrain** y **YTrain** en los arreglos globales **xTrain** y **yTrain**.

populateTable(dataRows)

Llena una tabla HTML con los datos almacenados en **dataRows**.

Uso:

- Se llama dentro de **loadCSV()** después de procesar los datos del CSV.
- Limpia la tabla existente y agrega las nuevas filas de datos.

model_type()

Determina el tipo de modelo a usar y ejecuta el ajuste correspondiente.

Uso:

- Se invoca cuando el usuario decide realizar una predicción.
- Llama a **runLinearRegression()** o **runPolyRegression()** basado en el algoritmo seleccionado.

runPolyRegression()

Ejecuta la regresión polinómica y muestra los resultados.

Uso:

- Se llama si el algoritmo seleccionado es "polynomial".
- Utiliza una clase **PolynomialRegression** para ajustar el modelo y hacer predicciones.
- Muestra el error y las predicciones en una tabla.

runLinearRegression()

Ejecuta la regresión lineal y muestra los resultados.

Uso:

- Se invoca si el algoritmo seleccionado es "linear".
- Utiliza una clase **LinearRegression** para ajustar el modelo y calcular errores y parámetros.
- Muestra la pendiente, el intercepto, el error cuadrático medio y las predicciones en una tabla.

callDraw()

Prepara y dibuja el gráfico utilizando Google Charts.

Uso:

- Se llama cuando el usuario desea ver el gráfico de resultados.
- Une los arreglos de datos en un formato adecuado para Google Charts y dibuja el gráfico.

drawChart(dataArray)

Dibuja un gráfico usando Google Charts basado en los datos proporcionados.

Uso:

- Se invoca dentro de `callDraw()`.
- Configura el tipo de gráfico y muestra los datos en la sección de gráfico definida en HTML.

CONCLUSIONES

- La implementación de la funcionalidad para la carga y procesamiento de datos demostró ser efectiva, permitiendo a los usuarios seleccionar archivos CSV, validar su formato y extraer correctamente los datos requeridos para el entrenamiento de los modelos de regresión. El uso de la biblioteca Tytus.js facilitó la gestión de datos, asegurando un almacenamiento eficiente y optimizando el flujo de datos entre la interfaz y el procesamiento subyacente.
- La ejecución de los algoritmos de regresión lineal y polinómica permitió a los usuarios obtener predicciones precisas basadas en los datos ingresados. Las visualizaciones generadas con Google Charts proporcionaron una representación clara y comprensible de los resultados, lo que mejoró la interpretación del rendimiento del modelo. Esto confirmó que la interfaz no solo cumplió con los objetivos planteados, sino que también ofreció una herramienta valiosa para el análisis de datos en el contexto del aprendizaje automático.