

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Inteligencia Artificial 1 – Sección A
Ing. Luis Fernando Espino Barrios
Aux. Erick Eden Sandoval Ramirez



Proyecto 2 – Manual Técnico

Segundo semestre regular, 2024

Índice

Objetivos	3
Introducción.....	4
Desarrollo.....	5
Conclusiones.....	9

Objetivos

- **Desarrollar un Sistema de Predicción:** Implementar un sistema de predicción basado en diversos algoritmos de Machine Learning, permitiendo a los usuarios cargar archivos CSV como entradas de datos.
- **Interfaz de Usuario Amigable:** Proporcionar una interfaz gráfica que facilite la selección de algoritmos y la visualización de resultados de predicción.
- **Versatilidad en Algoritmos:** Incluir múltiples algoritmos de Machine Learning (como Árbol de Decisión, Regresión Lineal, Naive Bayes, Redes Neuronales, etc.) para ofrecer opciones de análisis a los usuarios.
- **Visualización de Resultados:** Mostrar gráficamente los resultados de las predicciones y el árbol de decisión para una mejor interpretación.

Introducción

En la actualidad, el Machine Learning es una herramienta poderosa que permite realizar predicciones basadas en datos. Este proyecto tiene como objetivo crear una aplicación web interactiva que utilice TytusJS para implementar algoritmos de Machine Learning. Los usuarios podrán cargar archivos CSV que contengan datos de entrenamiento y realizar predicciones utilizando distintos algoritmos.

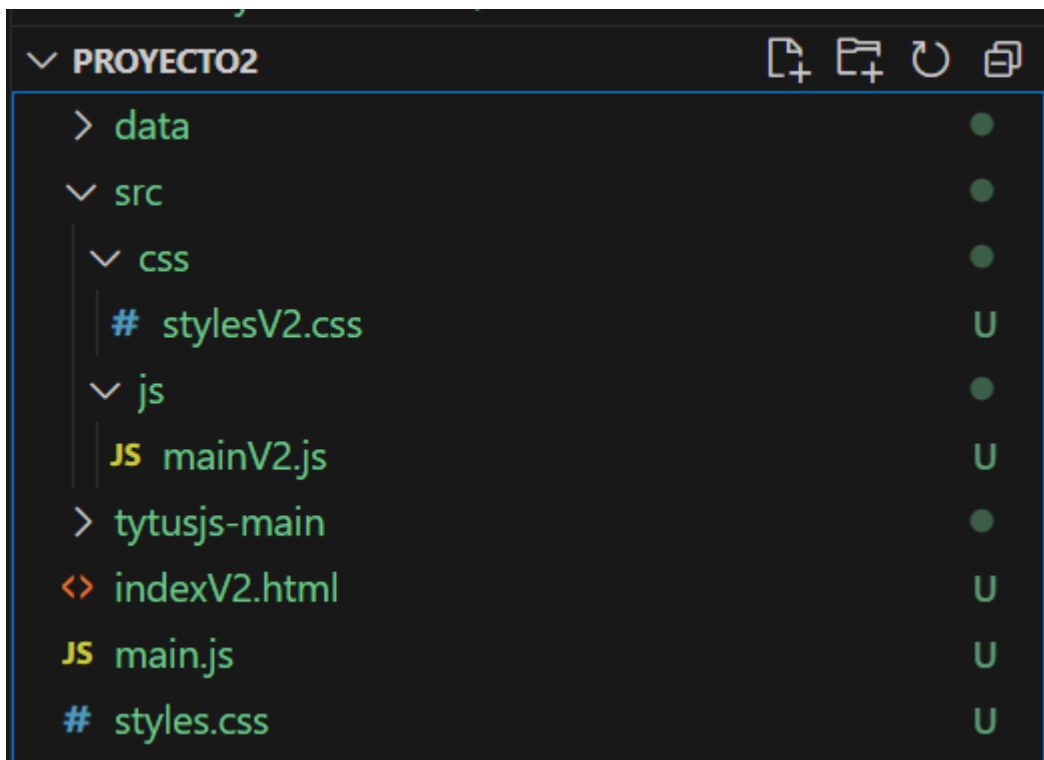
La aplicación se compone de una interfaz que permite seleccionar el algoritmo deseado, cargar los datos y visualizar los resultados. Esto facilita el uso de técnicas de Machine Learning incluso para aquellos que no tienen un profundo conocimiento técnico en la materia.

Desarrollo

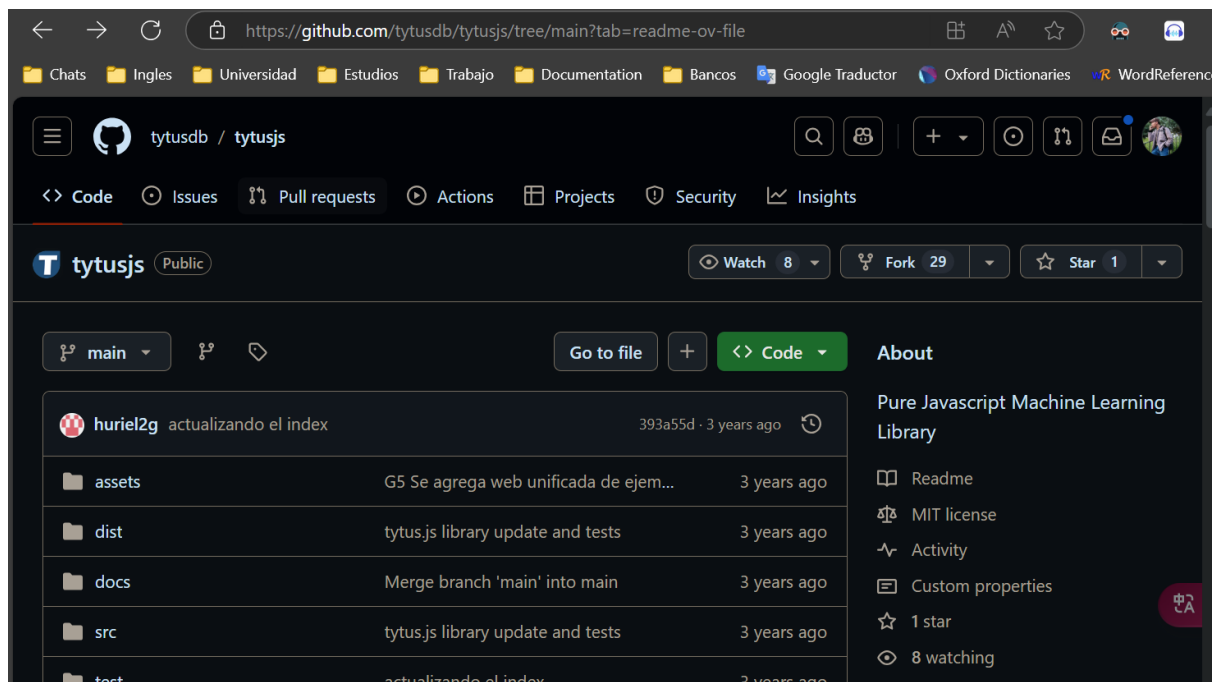
Estructura del Proyecto

El proyecto está organizado en tres archivos principales:

1. **indexV2.html**: Contiene la estructura HTML de la aplicación, donde se definen los elementos de la interfaz de usuario, como botones y campos de entrada.
2. **stylesV2.css**: Contiene los estilos de la aplicación, asegurando una presentación visual coherente y atractiva.
3. **mainV2.js**: Contiene la lógica principal del proyecto, que incluye la carga de datos, la implementación de los algoritmos de Machine Learning y la gestión de eventos en la interfaz de usuario.

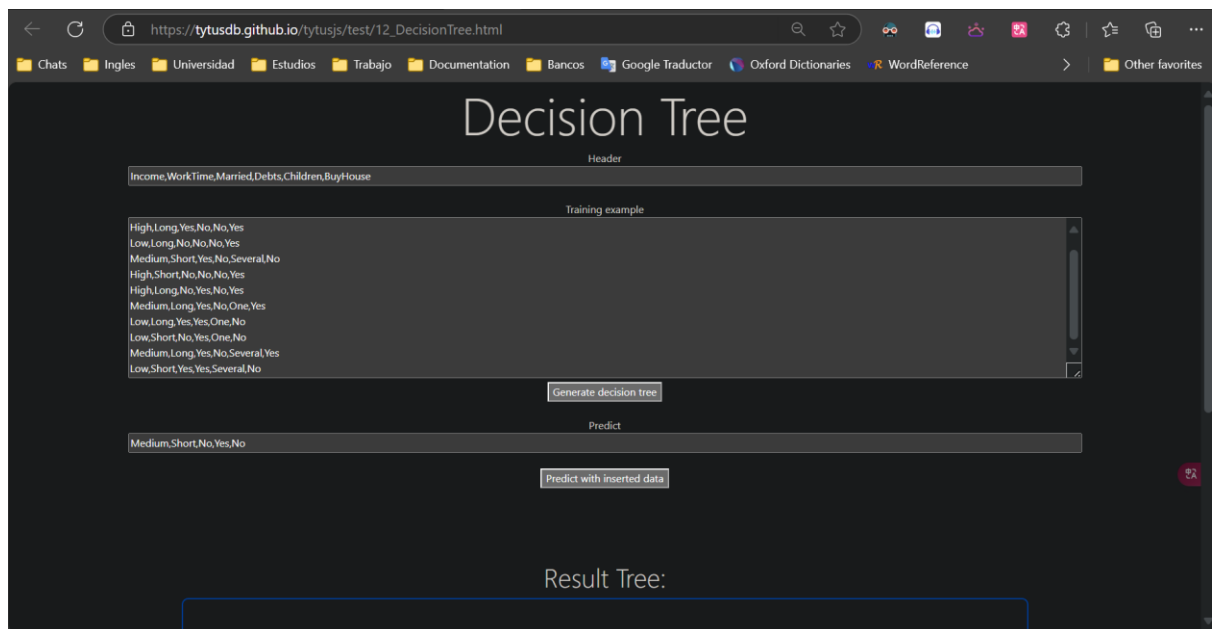


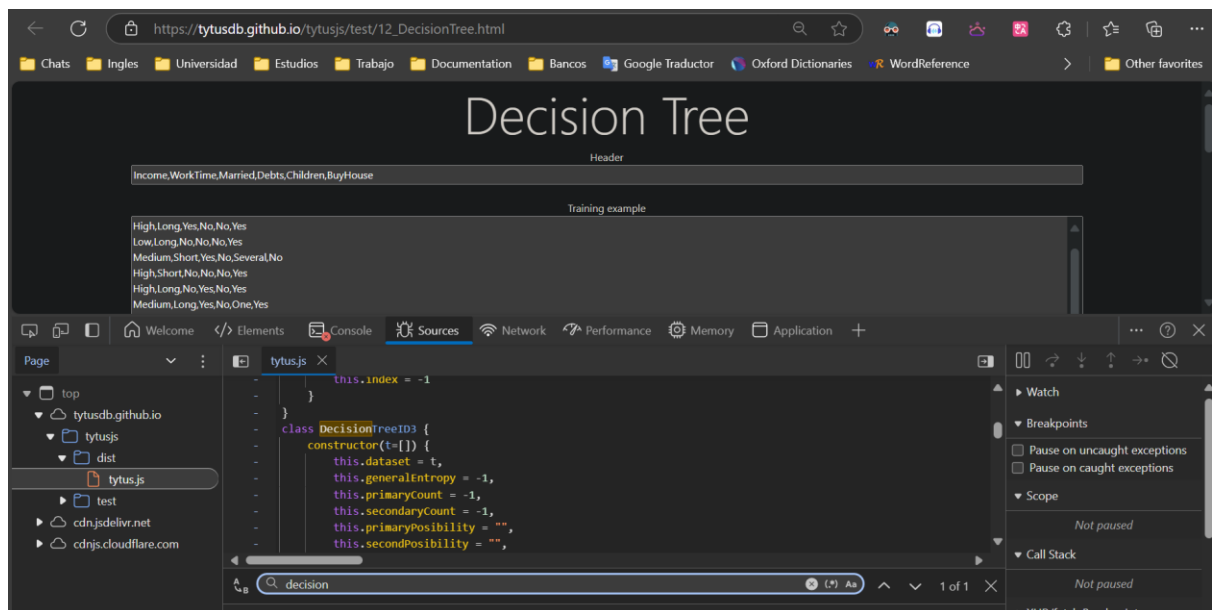
Para uso de la librería de tytusjs se descarga el proyecto del repositorio oficial el cual contiene todas las clases y funciones necesarias para hacer uso de los algoritmos.



<https://github.com/tytusdb/tytusjs.git>

En dicho enlace se encontrará adicional del proyecto, los ejemplos que el mismo autor publica para uso de las librerías y que el usuario se puede ir guiando, leyendo su estructura y propia implementación en las páginas publicadas.

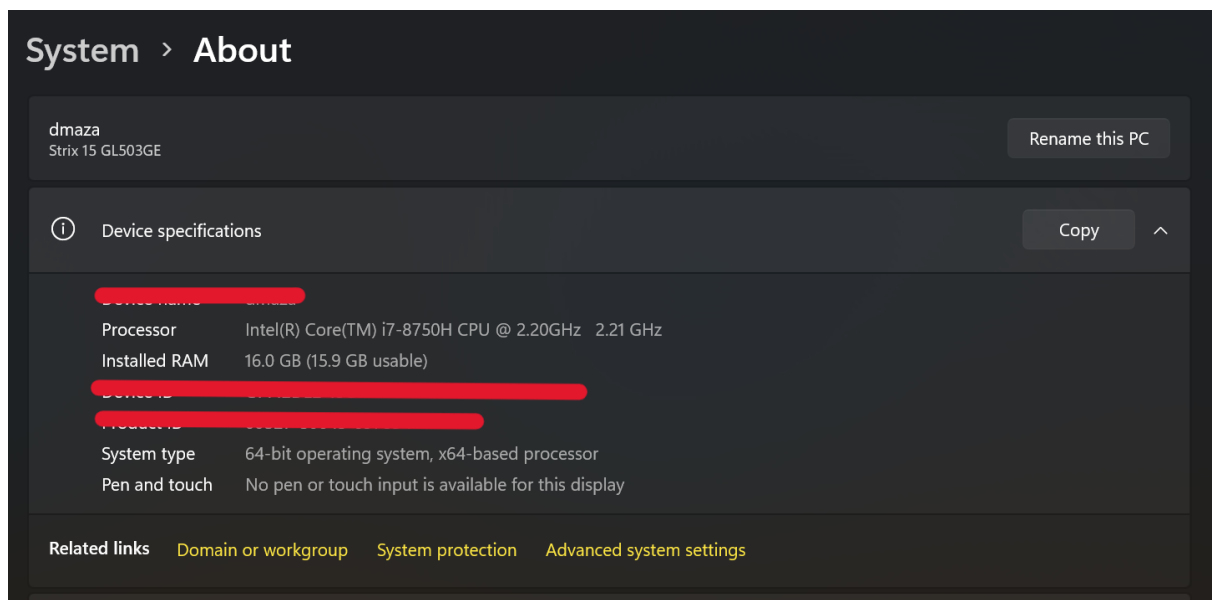




Así es como básicamente se puede ir extrayendo la información de la implementación de métodos de la librería como tal, por si igual se desea implementar algún otro en el futuro.

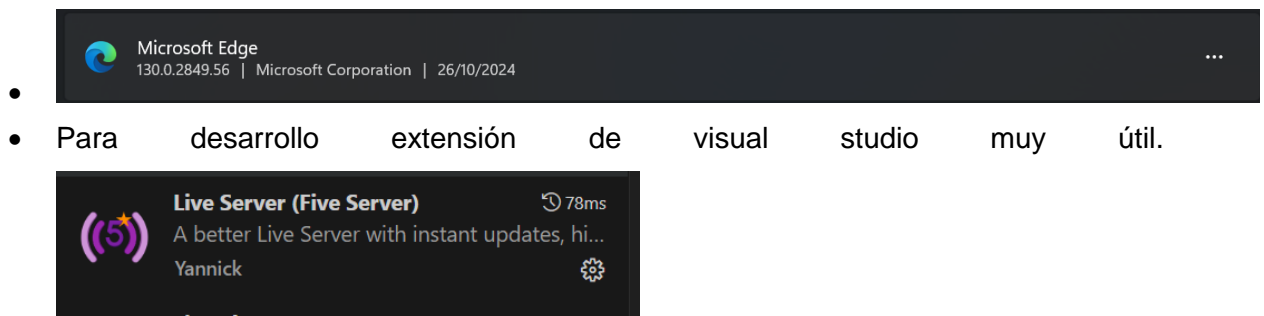
Herramientas técnicas utilizadas

Hardware



Software

- Windows 11
- Visual Studio Code



- Para desarrollo extensión de visual studio muy útil.

Flujo general del aplicativo

1. **Carga de Datos:** El usuario selecciona dos archivos CSV que se cargarán como conjuntos de datos. Estos archivos deben contener las características y etiquetas necesarias para el entrenamiento del modelo.
2. **Selección de Algoritmo:** Se proporciona un menú desplegable donde el usuario puede elegir el algoritmo que desea utilizar para el análisis.
3. **Entrenamiento del Modelo:** Al hacer clic en el botón "Entrenar", el modelo seleccionado se entrena utilizando los datos proporcionados.
4. **Predicción:** Una vez entrenado el modelo, el usuario puede realizar predicciones basadas en nuevas entradas.
5. **Visualización:** Se muestran los resultados de la predicción y, si corresponde, la representación gráfica del árbol de decisión.

Ejemplo de consumo de librería y creación de algoritmos

```
function handleTrainDecisionTree() {  
  console.log({ 'Data3': data3, 'Data2': data2 });  
  
  decisionTreeModel = new DecisionTreeID3(data3);  
  root = decisionTreeModel.train(decisionTreeModel.dataset);  
  
  resCnainer.style.display = 'block';  
  console.log('results', { 'Decision Tree': root });  
  
  graphBn.style.display = 'block';  
}
```

Este fragmento muestra cómo se inicializa y entrena el modelo de árbol de decisión, seguido de la visualización de los resultados. Cabe resaltar que todo el aplicativo se centra en un único documento html, manejando la lógica y separación de funciones desde javascript.

Conclusiones

El proyecto "Machine Learning con TytusJS" proporciona una herramienta eficaz para realizar predicciones utilizando diferentes algoritmos de Machine Learning. Su interfaz amigable y su capacidad de visualizar resultados hacen que la aplicación sea accesible tanto para principiantes como para usuarios avanzados.

Futuras Mejoras

- **Integración de más algoritmos:** Incluir otros métodos de Machine Learning y sus variantes para enriquecer la experiencia del usuario.
- **Mejora en la Visualización:** Implementar gráficos interactivos más avanzados que permitan una mejor comprensión de los resultados.
- **Optimización del Rendimiento:** Asegurar que la aplicación sea eficiente y pueda manejar grandes volúmenes de datos sin problemas de rendimiento.