



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Proyecto 2

Laboratorio Inteligencia Artificial 1
2 Semestre 2024
Oscar Humberto Morales Oviedo
200011652

Manual Técnico

Proyecto: Inteligencia Artificial 2024 2S - Proyecto 2

Autor: Oscar Morales - 20001652

Fecha de publicación: 2024

Versión: 1.0

Índice

1. **Introducción**
2. **Arquitectura del Sistema**
 - 2.1 Frontend
 - 2.2 Backend (Opcional)
 - 2.3 Base de Datos (Opcional)
3. **Componentes del Sistema**
 - 3.1 Estructura de Carpetas y Archivos
 - 3.2 Descripción de Scripts Principales
4. **Configuración del Entorno**
5. **Flujo de Datos y Procesos**
 - 5.1 Carga y Lectura de Archivos CSV
 - 5.2 Selección y Configuración de Modelos
 - 5.3 Procesamiento de Datos
6. **Modelos de Machine Learning Implementados**
 - 6.1 Regresión Lineal
 - 6.2 Regresión Polinomial
 - 6.3 K-Means Clustering
7. **Descripción de Funcionalidades**
 - 7.1 Carga de Datos
 - 7.2 Entrenamiento de Modelos
 - 7.3 Predicciones y Visualización
8. **Consideraciones de Seguridad**
9. **Pruebas y Validación**
10. **Solución de Problemas y Depuración**
11. **Anexos**

1. Introducción

Este manual técnico documenta el diseño, desarrollo y funcionamiento del proyecto "Inteligencia Artificial 2024 2S - Proyecto 2", un sitio web orientado a la visualización de modelos de machine learning utilizando HTML, JavaScript y la biblioteca de gráficos y modelos de machine learning Tytus.js. La estructura de este documento está dirigida a

desarrolladores y administradores que busquen comprender el sistema en su totalidad y realizar posibles mejoras.

2. Arquitectura del Sistema

Este sistema es una aplicación web estática que utiliza una arquitectura **cliente-servidor** simplificada. Se compone de un frontend en HTML, JavaScript y CSS que interactúa directamente con el navegador del usuario y depende de las bibliotecas de machine learning en JavaScript.

2.1 Frontend

El frontend es el componente principal de este proyecto y se compone de HTML, CSS y JavaScript. La estructura y estilo del sitio se gestionan con **Tailwind CSS**, y las visualizaciones de los resultados de los modelos se generan utilizando Google Charts y la biblioteca de gráficos Vis.js para visualizaciones interactivas.

2.2 Backend (Opcional)

Aunque esta aplicación actual no incluye un backend en Node.js o cualquier otro entorno, se podría expandir en el futuro para incluir APIs que almacenen datos, registros de usuarios y modelos entrenados en una base de datos remota o local.

2.3 Base de Datos (Opcional)

No se requiere una base de datos para el funcionamiento actual de esta aplicación, ya que toda la manipulación de datos se realiza en el navegador. Sin embargo, si se expandiera el sistema para almacenar modelos entrenados o resultados de predicciones, se recomendaría implementar una base de datos NoSQL como MongoDB.

3. Componentes del Sistema

3.1 Estructura de Carpetas y Archivos

La estructura básica del proyecto es la siguiente:

```
Proyecto2-IA/
|
├── index.html          # Archivo principal de la interfaz de usuario
├── style.css           # Estilos adicionales
├── dist/
│   ├── tytus.js       # Biblioteca de machine learning Tytus.js
│   └── tytusjs.js     # Biblioteca de machine learning Tytus.js
├── index.js           # Lógica y control de interfaz en JavaScript
├── assets/            # Imágenes y otros recursos
├── js/
│   ├── app.js         # Biblioteca de machine learning Tytus.js
│   └── tytus.js       # Biblioteca de machine learning Tytus.js
└── README.md          # Documentación general del proyecto
```

3.2 Descripción de Scripts Principales

1. **index.html**: Contiene la estructura de la página, incluyendo los campos de carga de archivos, controles de selección de modelos, botones de entrenamiento, predicción, y las secciones de visualización de resultados.
2. **index.js**: Archivo JavaScript que maneja la lógica principal de la aplicación. Este archivo gestiona la carga de datos, los eventos de selección de modelos, la interacción con Tytus.js para entrenamiento y predicción, y la visualización de gráficos.
3. **tytus.js**: Biblioteca de machine learning que implementa los modelos y algoritmos necesarios para el procesamiento y cálculo de predicciones.

4. Configuración del Entorno

Para ejecutar el proyecto, asegúrate de contar con:

1. **Navegador Web**: Preferiblemente Google Chrome o Firefox.

2. **Tailwind CSS CDN:** La hoja de estilo se carga directamente desde el CDN, por lo que no es necesario instalarlo localmente.
3. **Bibliotecas JavaScript:**
 - **Tytus.js:** Debe estar disponible en la carpeta `dist/` para que el sistema pueda acceder a los modelos de machine learning.
 - **Google Charts** y **Vis.js:** Se cargan a través de sus respectivos CDN para generar gráficos.

5. Flujo de Datos y Procesos

5.1 Carga y Lectura de Archivos CSV

1. El usuario carga un archivo CSV mediante el control de entrada.
2. El archivo se lee en el cliente usando JavaScript, permitiendo extraer los datos en formato tabular.
3. Si se selecciona la opción de ignorar el encabezado, el sistema omite la primera fila.

5.2 Selección y Configuración de Modelos

Al seleccionar un modelo de la lista, la interfaz muestra los campos y controles específicos para cada modelo, permitiendo configurar parámetros (por ejemplo, grados en la regresión polinomial, cantidad de clases en K-Means).

5.3 Procesamiento de Datos

1. Los datos cargados se preprocesan para garantizar que el modelo seleccionado pueda manejar los valores de entrada.
2. Se ejecuta el entrenamiento, y los resultados (pesos o clusters) se almacenan en variables temporales para su uso en la fase de predicción.

6. Modelos de Machine Learning Implementados

6.1 Regresión Lineal

- Algoritmo sencillo que establece una relación lineal entre las variables de entrada y de salida.
- Utiliza mínimos cuadrados para ajustar una línea que minimiza la suma de los errores cuadráticos.

6.2 Regresión Polinomial

- Expansión de la regresión lineal que permite modelar relaciones no lineales.
- Los datos son elevados a distintas potencias (grados) para ajustar mejor la curva a los datos.

6.3 K-Means Clustering

- Algoritmo de agrupamiento no supervisado que clasifica datos en **k** clases basándose en la cercanía.
- Ideal para dividir los datos en grupos homogéneos.

7. Descripción de Funcionalidades

7.1 Carga de Datos

Permite seleccionar y cargar archivos CSV que contienen datos de entrada. La interfaz maneja la extracción de datos y opcionalmente omite el encabezado.

7.2 Entrenamiento de Modelos

Para cada modelo, el sistema ofrece controles específicos:

- **Entrenar modelo:** Calcula los parámetros óptimos para el modelo.
- **Parámetros adicionales:** Como el rango de predicción o el número de clases en K-Means.

7.3 Predicciones y Visualización

1. **Predicciones:** Una vez entrenado el modelo, se puede hacer una predicción basada en valores de entrada proporcionados por el usuario.
2. **Visualización:** Los resultados se muestran en gráficos de Google Charts o Vis.js.

8. Consideraciones de Seguridad

1. **Validación de Datos:** Se recomienda implementar verificaciones adicionales para el formato y el contenido de los archivos CSV.
2. **Protección de Entradas:** Asegúrate de que los datos ingresados no contengan scripts maliciosos para prevenir ataques XSS.
3. **Uso de HTTPS:** Si el sistema se publica en un servidor, se recomienda el uso de HTTPS para proteger la información.

9. Pruebas y Validación

Para asegurar la funcionalidad y confiabilidad, se recomienda realizar las siguientes pruebas:

1. **Pruebas de Integración:** Verificar que los datos se carguen y que los modelos se entrenen y predigan correctamente.
2. **Pruebas de Rendimiento:** Evaluar la carga en el navegador con archivos de distintos tamaños.
3. **Pruebas de Compatibilidad:** Asegurarse de que el sistema funcione en diferentes navegadores y versiones.

10. Solución de Problemas y Depuración

- **No se cargan los datos del archivo CSV:**
 - Verifica el formato del archivo y asegúrate de que el sistema esté configurado para ignorar o incluir el encabezado según corresponda.
- **Error en el entrenamiento o predicción:**
 - Confirma que los parámetros sean adecuados para el modelo elegido.
 - Verifica la consola del navegador para corregir mensajes de error.

11. Anexos

A. Ejemplo de CSV para Regresión Lineal

```
variable_1, variable_2  
5, 15  
6, 18  
...
```

B. Configuración recomendada del navegador

- Permitir ventanas emergentes y configuración de permisos para leer archivos locales.