

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
INTELIGENCIA ARTIFICIAL
ING. LUIS FERNANDO ESPINO
AUX. ERICK SANDOVAL



MANUAL TÉCNICO

Daniel Rolando Sotz Alvarado

201430496

Guatemala, 2 de noviembre 2024

Introducción

Este documento ofrece una visión general de la aplicación, su propósito y la forma de utilización de la misma para diferentes usuarios. Describe brevemente los modelos de machine learning disponibles y los tipos de problemas que cada uno puede resolver (por ejemplo, regresión, clasificación, clustering).

Requisitos

- Navegador web: La aplicación funciona en cualquier navegador moderno que soporte JavaScript.
- Conexión a Internet: Se necesitan algunas bibliotecas alojadas en línea.

Librerías

- TytusJS: Librería de machine learning para trabajar con modelos como regresión, árboles de decisión, redes neuronales, entre otros.
- Chart.js: Utilizado para la visualización de datos en gráficos.
- vis-network: Herramienta para visualizar grafos y redes

Estructura Proyecto

index.html: Archivo principal que contiene el código HTML y CSS necesarios. Contiene los siguientes elementos:

- Selector carga de datos
- ComboBox para seleccionar modelo
- Botones
- Tablas resultados
- Gráficas resultados

proyecto2.js: Archivo que contiene el código JavaScript.

Funcionamiento

1. La aplicación realiza una manipulación dinámica para construir una interfaz interactiva que permite cargar datos, seleccionar modelos de machine learning, entrenar, predecir, y visualizar resultados.
2. La aplicación escucha el evento **change** en el input de tipo archivo para cargar el archivo CSV seleccionado por el usuario. Una vez cargado, el archivo se procesa, y

sus datos se convierten en una tabla HTML. Esta tabla se inserta en el DOM para que el usuario pueda ver los datos cargados.

3. El menú desplegable que permite seleccionar el modelo de machine learning escucha el evento input para ejecutar una función que carga los elementos de configuración específicos para cada modelo.
4. Cada modelo tiene configuraciones específicas que se muestran en el DOM de forma dinámica según los requisitos del modelo. Una función central genera los selects y labels adecuados para las variables de entrada, salida, y otros parámetros específicos del modelo.
 - a. **Regresión:** Configura las variables de entrenamiento (**xTrain** y **yTrain**) y prueba (**xTest**), y, en el caso de la regresión polinomial, el grado del polinomio.
 - b. **Árbol de Decisión y Naïve Bayes:** Permiten seleccionar la variable objetivo (columna a predecir) y las características o causas que afectarán a la predicción.
5. Los botones de acción (**Entrenar**, **Ejecutar** y **Generar Gráficas**) permiten realizar las operaciones del modelo seleccionado. Estos botones escuchan eventos de clic y ejecutan las funciones correspondientes a la operación solicitada.
6. Los resultados de entrenamiento y predicción se generan en tablas HTML o en gráficos específicos, y se insertan en divisiones adecuadas.

Definición Modelos

Regresión Lineal

Un modelo de regresión lineal es básicamente un modelo que nos permite describir cómo interviene una variable 'X' sobre otra variable 'Y'.

- X: Variable independiente o exógena.
- Y: Variable dependiente o endógena.

El objetivo de la regresión lineal es poder obtener estimaciones razonables de Y para distintos valores de X a partir de una muestra de 'n' pares de valores (x1, y1), ..., (xn, yn).

Ejemplos aplicativos de la regresión lineal:

- Cómo influye la estatura del padre sobre la de sus hijos.
- Determinar el precio de una casa en función de su superficie.
- Conocer la mortalidad y el consumo medio diario de cigarrillos.
- Abundancia de algún parásito en función de la temperatura.
- Aproximar la calificación obtenida en una materia según el número de horas de estudio semanal.
- Prever el tiempo de computación de un programa en función de la velocidad del procesador.

Objetivos de conocer y entender la regresión lineal:

- Poder construir un modelo de regresión lineal simple que describa cómo influye una variable 'X' sobre otra variable 'Y'.
- Poder obtener estimaciones puntuales de los parámetros de un modelo de regresión lineal.
- Poder determinar el valor promedio de la variable 'Y' para cualquier valor de la variable 'X'.
- Poder predecir futuros de la variable dependiente 'Y'.

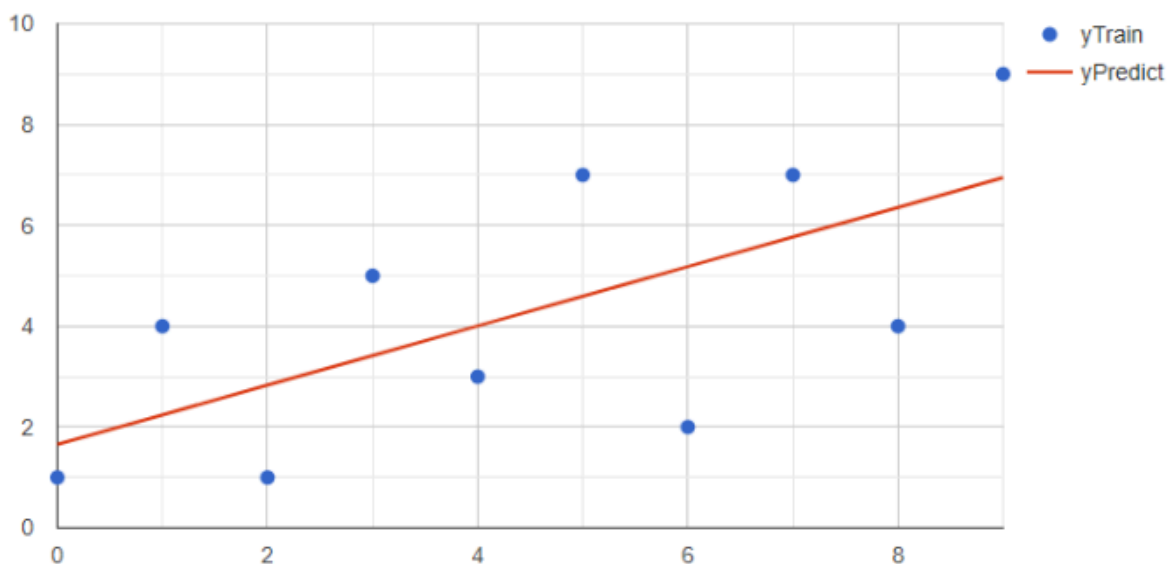
Ejemplo:

La muestra que se utilizó para la realización de la prueba de la regresión lineal fue la siguiente:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

[1, 4, 1, 5, 3, 7, 2, 7, 4, 9]

Graficando dicha muestra se obtuvo la siguiente gráfica.



Regresión Polinomial

Es una predicción de una variable de respuesta cuantitativa a partir de una variable predictora cuantitativa, donde la relación se modela como una función polinomial de orden n.

La regresión lineal se puede calcular aplicando mínimos cuadrados, así como la solución de incógnitas resultantes como son el error absoluto y cuadrático, la definición adecuada de la función objetivo utilizada por el método y obtención de un algoritmo matemático implementado en un lenguaje de programación de libre elección capaz de mostrar los polinomios de mínimos cuadrados y sus valores de aproximación.

Objetivo:

Consiste en encontrar una función que mejor se ajuste a los datos que se proporcionan o mejor representen la gráfica que se ajuste mejor al modelo. Se debe saber que el error absoluto es igual a la sumatoria de la diferencia entre la función modelada y la polinómica y los valores de la variable independiente elevados todos al cuadrado.

Ecuación de regresión polinomial:

$$y=a+bx+cx^2+dx^3\dots$$

Ejemplo de regresión polinomial:

X Train: [0,1,2,3,4,5,6,7,8,9]

Y Train: [0,15,5,25,40,35,10,5,45,35]

X To Predict: [0,1,2,3,4,5,6,7,8,9]

Y Prediction Degree 2: [4.68,10.14,14.95,19.11,22.64,25.52,27.75,29.34,30.29,30.59]

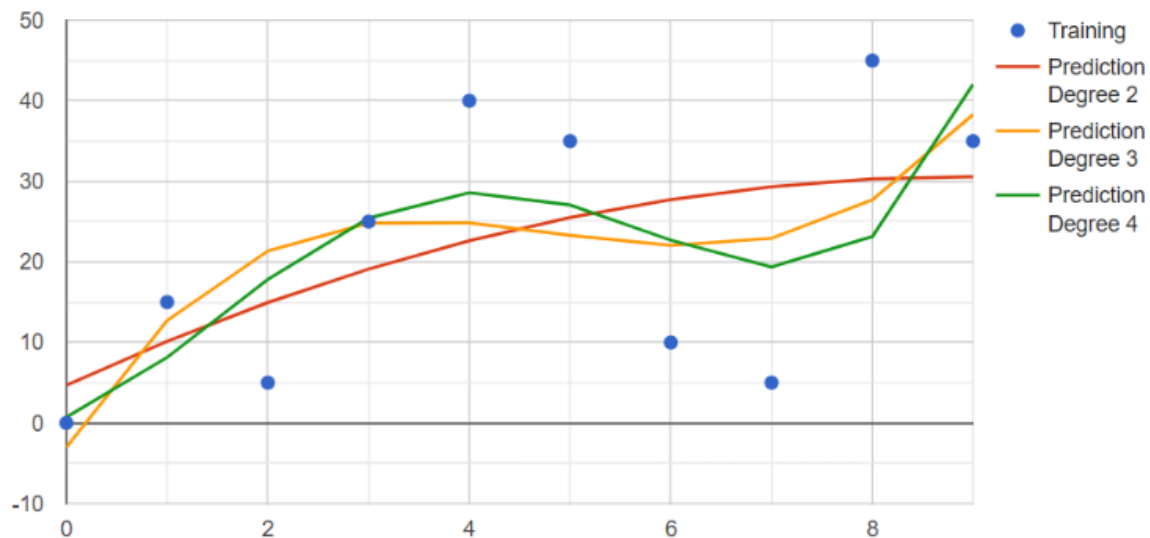
Y Prediction Degree 3: [-3.03,12.71,21.37,24.8,24.84,23.31,22.06,22.92,27.72,38.3]

Y Prediction Degree 4: [0.72,8.13,17.84,25.43,28.58,27.06,22.68,19.38,23.14,42.05]

R² for Degree 2: 0.3

R² for Degree 3: 0.42

R² for Degree 4: 0.47



Árbol de Decisión

Un árbol de decisión es un mapa de los posibles resultados de una serie de decisiones relacionadas. Permite que un individuo o una organización comparen posibles acciones entre sí según sus costos, probabilidades y beneficios. Se pueden usar para dirigir un intercambio de ideas informal o trazar un algoritmo que anticipe matemáticamente la mejor opción.

Un árbol de decisión, por lo general, comienza con un único nodo y luego se ramifica en resultados posibles. Cada uno de esos resultados crea nodos adicionales, que se ramifican en otras posibilidades. Esto le da una forma similar a la de un árbol.

Hay tres tipos diferentes de nodos: nodos de probabilidad, nodos de decisión y nodos terminales. Un nodo de probabilidad, representado con un círculo, muestra las probabilidades de ciertos resultados. Un nodo de decisión, representado con un cuadrado, muestra una decisión que se tomará, y un nodo terminal muestra el resultado definitivo de una ruta de decisión.

Configuración

- **Variable objetivo:** La columna a predecir.
- **Características:** Las columnas de entrada que influyen en la predicción.

Implementación

- **train():** Genera el árbol de decisión a partir de los datos de entrenamiento.
- **test():** Clasifica una muestra de prueba.
- **graph():** Visualiza el árbol con `vis-network` para mostrar las decisiones en nodos y ramas.