Juan Manuel Guevara Jimenez
Luis Diego Flores Arguedas

Reporte de proyecto 1 Inteligencia Artificial

Manual de instalación

Para iniciar el programa se debe ejecutar el módulo menú.py. Para el correcto uso del programa se requiere la instalación de las siguientes librerías:

- Sys
- Math
- Pandas

Para añadir las librerías, debe escribirse el comando -pip install- seguido del nombre de la librería en línea de comandos.

Uso

Al ejecutar el programa se mostrará un menú con las siguientes opciones, las cuales deben ser seleccionadas ingresando el respectivo número asociado a ellas:

- Leer archivo .csv
- Realizar normalización del archivo
- Ejecutar árboles de decisión
- Realizar cross-validation de resultados
- Salir

Reporte de métodos

Reader.py

Métodos:

- read()
- get_matrix()

Read(): Método utilizado para la lectura de archivos csv. Se decide que el método lea predeterminadamente un archivo llamado data.csv con la misma ubicación del módulo. Los datos leídos son almacenados en una matriz lógica que será utilizada en módulos posteriores para realizar los cálculos de normalización y los árboles de decisión.
Get matrix: Éste método se utiliza desde otros módulos para la obtención y manipulación de la matriz leída.

Standardization.py

Métodos:

- get_col()

- mean()
- standardDeviation()
- zscore()
- normalization()
- get_z_matrix()
- to_csv()

Get col: La finalidad de este método es conseguir leer una columna de la matriz ingresada ignorando el encabezado que asocia la pertenencia de los datos.
Mean: Método que calcula la media resultante de los datos que son ingresados mediante un arreglo.
Standard Deviation: Recibe un arreglo de datos y una media con los cuales calcula la desviación estándar.
Z score: Función que devuelve la normalización según los datos ingresados: elemento de la matriz, valor promedio de la columna y desviación estándar.
Normalization: Genera una matriz normalizada a partir del set de datos leído en reader, ésta nueva matriz se define dentro de los módulos como z_matrix.
To csv: Salida de la matriz formalizada en formato CSV, en caso de querer consultar dicha columna durante el tiempo de ejecución causa agotamiento de la memoria del programa, forzando a reiniciar.

Decision_tree.py

Clases:

- Node

Métodos:

- select_threshold()
- info_entropy()
- remainder()
- info_gain()
- num_class()
- choose_attr()
- build_tree()
- predict()
- print_tree()
- clean()
- x_aux()
- random_data_sets()
- trainingForest()
- predictForest()
- test_predictions()
- main()

Select threshold: Selecciona el umbral del atributo para dividir el conjunto de datos de prueba. El umbral elegido divide los datos de prueba de modo que se maximice la ganancia

de información. Recibe un DataFrame (tipo de dato de la librería pandas), un atributo y la predicción del atributo.
Info entropy: Calcula el contenido de información (entropía) de los datos de prueba.
Remainder: Calcula el promedio ponderado de la entropía después de una prueba de atributo.
Info gain: Calcula la ganancia de información de la prueba de atributo basada en un umbral dado. Los umbrales no siempre se mantienen para un mismo atributo.
Num class: Conteo de resultados afirmativos y negativos para el set de datos.
Choose att: Selecciona del set de atributos el atributo y umbral que tengan mayor ganancia de información.
Build tree: Construye el árbol de decisión basado en la información de entrenamiento.
Predict: Dado un training data, realiza una predicción entre M o B basado en el árbol de decisión.
Print tree: Función implementada que permite visualizar de manera sencilla los árboles que son construidos, mostrando de manera sencillo si el nodo es una predicción o un atributo y a que nivel del árbol pertenecen. Solo necesita de entrada el nodo raiz del arbol y un valor del nivel donde empieza.
Clean: Función utilizada que recibe el dataset, y cambia los valores que no son números y los transforma en datos de 1 y 0 que nos permite manejar el data set con más facilidad.
X aux: Utilizada por random datasets para ignorar la información irrelevante en la estructura de datos dada.
Random datasets: Selección de sets de datos que serán utilizados para la generación de árboles en el random forest. Realiza un aproximado del set de datos para generar un número impar de sets de datos.
Training forest: es el método encargado de entrenar los diferentes tipos de árboles y crear un bosque de árboles entrenados con el cual podremos realizar la predicciones más adelante.
Predict forest: función encargada de recorrer un csv de testeo en el cual pasa fila por fila y nos da la predicción de cada ejemplo que recibe comparando el ejemplo con el bosque de árboles de decisiones que habíamos entrenado.
Test predictions: función encargada de darnos el orden lógico de nuestro método de random forest, el cual recibe el csv de testeo, y dentro se encarga de procesar los datos de entrenamiento y entrenar el bosque y de ultimo probar los ejemplos y sus predicciones.
Main: Programa principal que ejecutará el módulo.