Predicción de éxito estudiantil con árboles de decisión.

Sara Rodríguez Velásquez
Stiven Yepes Vanegas
Medellín, 19/05/2020

Estructuras de Datos Diseñada

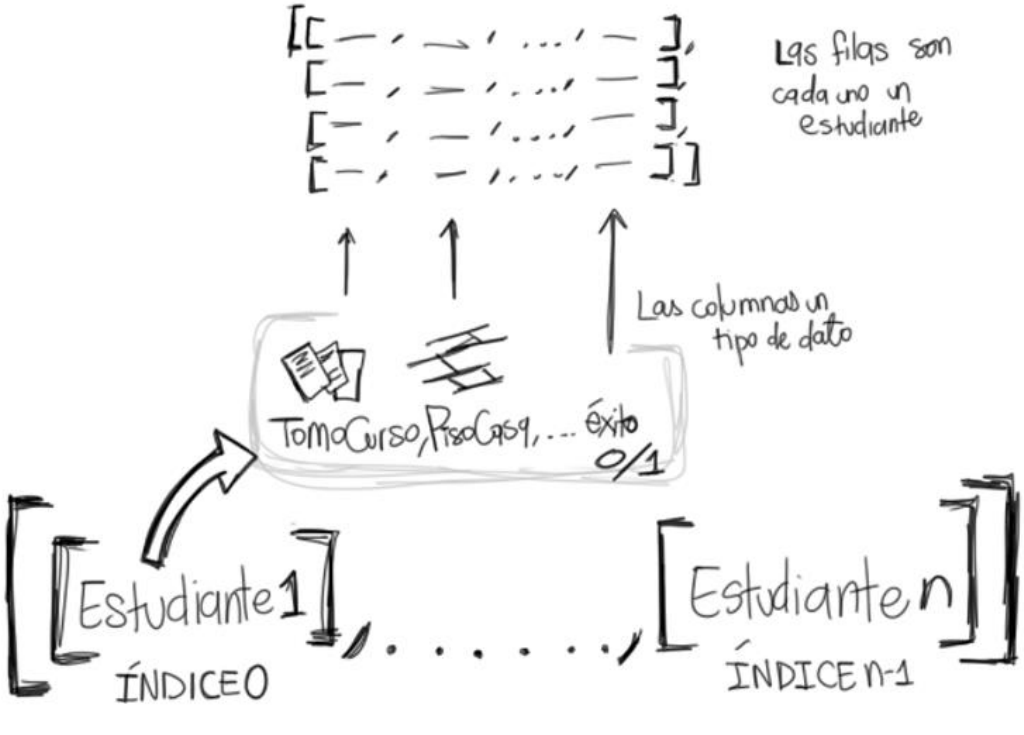


Gráfico 1: Arreglo numpy con datos de personas agrupados en cada element (element) de este.

Operaciones de la Estructura de Datos

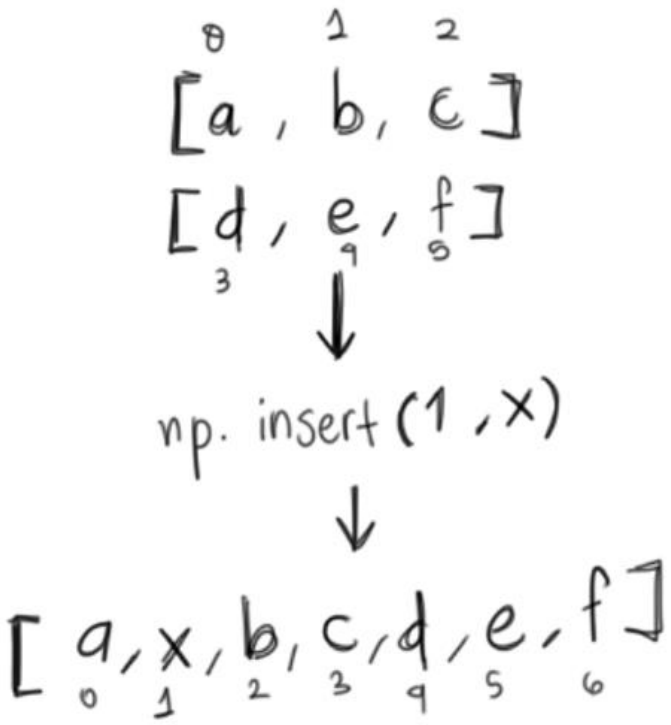
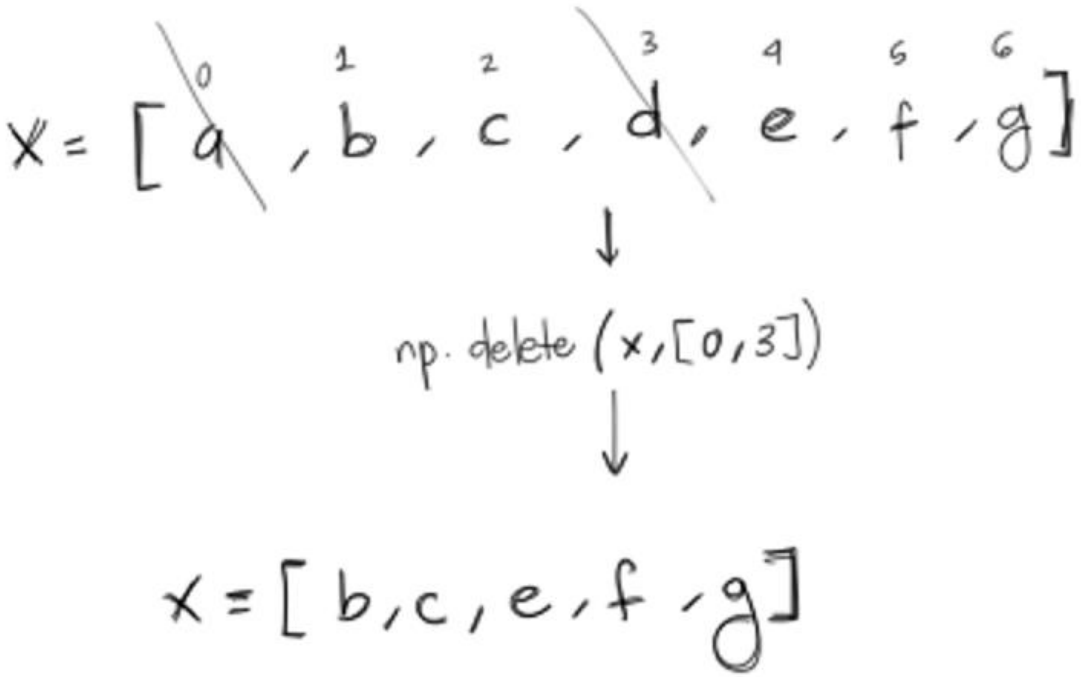


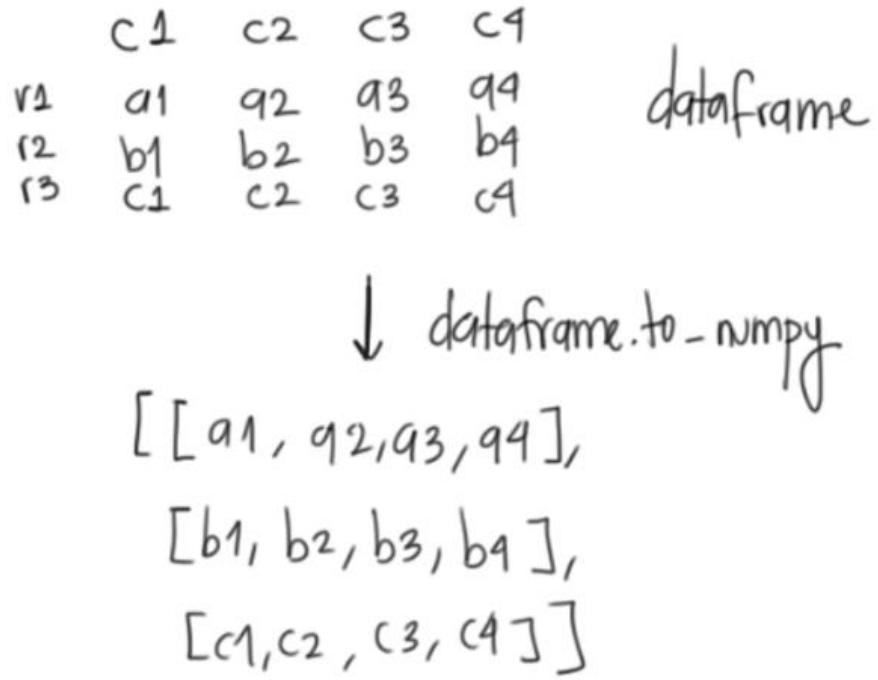
Gráfico 2: Inserción de un dato x en un arreglo de letras indicando solo la posición.

Operaciones de la Estructura de Datos



Gráfica 3: Borrado de 2 datos de un arreglo de letras, indicando la posición de borrado.

Operaciones de la Estructura de Datos



Gráfica 4: Conversión de un formato de pandas dataset a un arreglo numpy.

Complejidad de las operaciones de la estructutra de datos

Método	Complejidad
Función view (Ver elemento)	O(1)
Crear array vacío (numpy.empty)	O(1)
Crear array con ceros (numpy.zeros)	O(n)
Crear array multidimensional con ceros (numpy.zeros)	O(nm), n= filas, m=columnas
Función Copy (Copiar array)	O(n)
Función Copy (Copiar array multidimensional)	O(nm) n= filas, m=columnas

Tabla 1: Complejidad de las operaciones de la estructura de datos

Criterios de Diseño de la Estructura de Datos

- Se realiza la lectura del archivo csv con Pandas, para garantizar un cortado certero de los datos, y por las facilidades que tiene para rellenar los datos sin contestar con el valor string: “desconocido”, y de esta forma prevenir anomalías en la posterior construcción del árbol de decisión.

- El archivo se convierte a la estructura de array(arreglo) numpy, porque es mucho mas liviano para trabajar los datos, consume menor memoria y tenemos tamaños de datasets fijos.

- La razón de usar esta estructura es la habilidad para llamar los datos por categorías como columnas y filas y que es más liviano que el panda’s dataframe, una opción similar pero mucho mas lenta.

Consumo de Tiempo y Memoria

	LECTURA DE CSV	CONSTRUCCIÓN ARRAY	LECTURA Y CONSTRUCCIÓN
Conjunto de datos 1	0,045s	0,002s	0,06s
Conjunto de datos 2	0,1s	0,009s	0,16s
Conjunto de datos 3	0,13s	0,011s	0,2s
Conjunto de datos 4	0,16s	0,014s	0,25s
Conjunto de datos 5	0,22s	0,021s	0,34s
Conjunto de datos 6	0,28s	0,028s	0,44s
Conjunto de datos 7	0,35s	0,037s	0,55s
Conjunto de datos 8	0,45s	0,048s	0,72s
Conjunto de datos 9	0,63s	0,068s	1s
Conjunto de datos 10	0,8s	0,091s	1,3s

Tabla 2: Tiempos de ejecución de las operaciones de la estructura de datos con diferentes conjuntos de datos

Mediciones de tiempo con la lectura del archivo .csv en Panda, Construcción de un arreglo numpy y ambas operaciones juntas.

Consumo de Tiempo y Memoria

	CONSTRUCCIÓN ARRAY	CONSTRUCCIÓN LIST
Conjunto de datos 1	0,002s	0.003s
Conjunto de datos 2	0,009s	0.014s
Conjunto de datos 3	0,011s	0,2s
Conjunto de datos 4	0,014s	0.018s
Conjunto de datos 5	0,021s	0.047s
Conjunto de datos 6	0,028s	0.069s
Conjunto de datos 7	0,037s	0.105s
Conjunto de datos 8	0,048s	0.125s
Conjunto de datos 9	0,068s	0.164s
Conjunto de datos 10	0,091s	0.197s

Tabla 3: Comparación de tiempos de ejecución de las operaciones de la estructura de datos entre el array Numpy y las Listas de Python

Comparación de tiempo de la construcción del archivo con arreglos de numpy vs. Con listas de python.

Consumo de Tiempo y Memoria

	LECTURA DE CSV	CONSTRUCCIÓN ARRAY	LECTURA Y CONSTRUCCIÓN
Conjunto de datos 1	4.0664 MiB	2.3007 MiB	8.2656 MiB
Conjunto de datos 2	4.0039 MiB	7.1445 MiB	17.2890 MiB
Conjunto de datos 3	7.0429 MiB	9.0585 MiB	15.1601 MiB
Conjunto de datos 4	4.1289 MiB	11.6757 MiB	16.7656 MiB
Conjunto de datos 5	7.0390 MiB	16.7851 MiB	24.6445 MiB
Conjunto de datos 6	13.4609 MiB	21.4257 MiB	34.9648 MiB
Conjunto de datos 7	16.7460 MiB	28.3164 MiB	42.2968 MiB
Conjunto de datos 8	20.3203 MiB	36.4960 MiB	55.0156 MiB
Conjunto de datos 9	28.1406 MiB	51.7148 MiB	77.8632 MiB
Conjunto de datos 10	35.6484 MiB	67.0234 MiB	101.8164 MiB

Tabla 4: Consumo de memoria de la estructura de datos con diferentes conjuntos de datos

Medición de valores en uso de memoria por los procedimientos de lectura, creación del array y ambos pasos.

Software Desarrollado

Próximamente