TELEDUC
Educación a Distancia

Herramientas avanzadas de programación en *Python* para procesamiento de datos

Resumen

Módulo 1
Ciencia de datos

Se define como “la disciplina que permite encontrar patrones predecibles en sets de datos estructurados, y no estructurados” (Dhar, 2013). También como “un concepto que busca unificar estadísticas, análisis de datos, inteligencia artificial y cualquier otro método similar para poder entender y analizar fenómenos con los datos.” (Hayashi, 1998). Un ejemplo de su uso es la astronomía, donde diariamente se genera un exabyte de datos (la cantidad de información que se genera en internet en un solo día).

Python es un lenguaje de programación que permite ocupar la ciencia de datos. Específicamente, existen paquetes (como el paquete *random*) que permite hacer esto. Los paquetes también pueden llamarse librerías. Una librería muy usada para procesar datos de forma masiva y eficiente es *Pandas*.

Pandas es una librería de código abierto, que provee estructuras de datos de alto rendimiento y fáciles de usar. Además, suministra herramientas para el análisis de datos en *Python*.

Librería *Pandas*

Las estructuras de datos básicas en *Pandas* son: series y Data Frames.

- Serie: Es una lista de datos con un largo fijo, aquí no se agregan o quitan elementos, sólo se modifican. Además, los datos de una serie son del mismo tipo.
- Data Frame: Los Data Frames son la estructura más usada en la librería *Pandas*. Se percibe como una matriz de datos. Cada columna, representa datos para una variable específica. Y cada fila corresponde a medidas o valores de cada instancia. Se observa que cada columna es de un tipo específico, y cada fila tiene valores de distintos tipos. En esta estructura de datos, se agregan filas y columnas según la preferencia del programador. Las columnas de un Data Frame son series. Es

importante destacar que cada fila tendrá un índice, que permitirá acceder a esa fila en particular y su información específica.

Se crea un Data Frame a partir de una lista de listas con el siguiente código:

```
import Pandas as pd

data = [lista de listas]

df = pd.DataFrame(data)
```

En el caso anterior, la primera línea sirve para cargar la librería *Pandas* para usarla posteriormente. Esta la cargamos con el nombre pd, que es el que ocuparemos después para poder acceder a sus funciones. Luego, creamos un Data Frame mediante la función DataFrame(), ingresando como parámetro la lista de listas creada anteriormente.

No obstante, en este caso, el Data Frame generado no tiene nombres para sus columnas. Se añade nombres a las columnas agregando un segundo parámetro columns a la función DataFrame() con una lista de *strings* que representen los nombres de las columnas. Esto se puede hacer de la siguiente manera:

```
df = pd.DataFrame(data, columns=[lista nombres columnas])
```

Se carga un archivo CSV en un *Data Frame* con el siguiente código (asumiendo que ya cargamos la librería *Pandas*):

```
df = pd.read_csv("nombre archivo CSV",encoding="latin-
                    1",sep=";")
```

Mediante la función read_csv() sobre la librería *Pandas*, se lee un archivo CSV por el nombre del archivo. Luego, se ingresa como segundo parámetro encoding="latin-1" para leer caracteres únicos del español (como tildes, ¡, ñ, etc). Y el tercer parámetro sep=";" para asegurarnos de que *Pandas* lea correctamente el archivo CSV.

A continuación se presentarán algunas operaciones comunes con la librería *Pandas*:

- Ver columnas: Se utiliza para listar las columnas de un Data Frame usando la función dtypes(). Esto se aplica directamente sobre la variable que contiene al Data Frame. De esta forma se observan los nombres de las columnas y sus tipos. Si se encuentra un Data Frame en una variable de nombre df, se ejecuta esta operación de la siguiente manera:

```
df.dtypes()
```

- Extraer columna: Se observan todos los datos de una columna específica. Si hay un Data Frame en una variable de nombre df, se realiza esta operación de la siguiente manera:

```
df["nombre columna"]
```

- Extraer fila: Se observan todos los datos de una fila específica. Si se encuentra un Data Frame en una variable de nombre df, se ejecuta esta operación de la siguiente manera:

```
df.loc[identificador fila]
```

- Extraer filas: Se encuentran todos los datos de un conjunto de filas. Si se encuentra un Data Frame en una variable de nombre df, se realiza esta operación de la siguiente manera:

```
df.loc[identificador fila inicial: identificador fila final]
```

Importante: A diferencia de un intervalo de *strings*, en este efectivamente se toma desde el valor inicial al final. Por ejemplo, si uno tiene un intervalo 0:4, entonces mostrará las filas con identificadores 0,1,2,3 y 4.

- Extraer valor: Se observa un dato específico (columna) de una fila. Si se encuentra un Data Frame en una variable de nombre df, se efectúa esta operación de la siguiente manera:

```
df.loc[identificador fila fila][ "nombre columna"]
```

- Filtrar datos: Se emplea para filtrar datos según los valores de ciertas columnas. Si se tiene un Data Frame en una variable de nombre df, se ejecuta esta operación de la siguiente manera:

```
df.loc[df['Nombre columna'] (operación lógica)]
```

- Agregar columna: Si se tiene un Data Frame en una variable de nombre df, se agrega una columna con un valor específico de la siguiente manera:

```
df["nombre nueva columna"] = valor específico
```

También se agrega una nueva columna como el resultado de una operación con otra columna.

```
df["nueva columna"] = (operación con otra columna del Data
                       Frame)
```

- Editar columna: Se edita el contenido de una columna asignándole un nuevo valor (que puede ser específico o el resultado con otra columna). Si hay *un* Data Frame en una variable de nombre df, se realiza esta operación de la siguiente manera:

```
df["columna del Data Frame"] = valor específico
```

También se edita una columna como el resultado de una operación con otra columna.

```
df["columna del Data Frame"] = (operación con otra columna del
                                Data Frame)
```

- Borrar columna: Se borra una columna en un Data Frame. Si se encuentra un Data Frame en una variable de nombre df, se ejecuta esta operación de la siguiente manera:

```
del df["nombre columna"]
```

- Calcular estadísticos descriptivos: se calculan los estadísticos descriptivos de una Data Frame. Si hay un Data Frame en una variable de nombre df, se efectúa esta operación de la siguiente manera:

```
df.describe()
```

- Guardar en archivo CSV: Permite guardar un Data Frame en un archivo CSV. Si se encuentra un Data Frame en una variable de nombre df, se realiza esta operación de la siguiente manera:

```
df.to_csv("nombre archivo CSV", index=false)
```

Referencias bibliográficas

- Dhar, V. (2013). *Data Science and Prediction*. *Communications of the ACM*.
- Hayashi, C. (1998). *What is data science? Fundamental concepts and a heuristic example. Data Science, Classification, and Related Methods*, 40–51.