| PROGRAMA DE FORMACIÓN | Gestión analítica de datos |
| --- | --- |

| COMPETENCIA | 220501096 desarrollar la solución de *software* de acuerdo con el diseño y metodologías de desarrollo. | RESULTADOS DE APRENDIZAJE | 220501096-01 Codificar la lógica de programación, según metodología y herramienta de desarrollo. |
| --- | --- | --- | --- |

| NÚMERO DEL COMPONENTE FORMATIVO | CF14 |
| --- | --- |
| NOMBRE DEL COMPONENTE FORMATIVO | Profundización en *Data Science.* |
| BREVE DESCRIPCIÓN | En este componente, se estudia el *Data Science* o ciencia de datos, con sus componentes para el procesamiento y análisis de los mismos; para este proceso, se aplican herramientas de tratamiento y manipulación de datos como Numpy, Pandas, y Matplotlib*.* Finalmente, con modelos de *Machine learning* se generan predicciones para orientar a las organizaciones en la formulación de estrategias y la toma de decisiones. |
| PALABRAS CLAVE | *Machine learning, Big data,* redes neuronales, *deep learning* y probabilidad. |

| ÁREA OCUPACIONAL | Ciencias naturales, aplicadas y relacionadas |
| --- | --- |
| IDIOMA | Español |

**Tabla de contenido**

1 *Machine learning* y *Data science*

2 Modelos de *Machine learning*

# Introducción

| Cuadro de texto |
| --- |
| Apreciado aprendiz, bienvenido a este componente formativo, donde se desarrollarán aspectos generales y claves sobre la ciencia de los datos; conviértase en todo un experto en *Data science* y en modelos de *Machine learning*.  Afiance sus saberes y habilidades para el empleo de algoritmos de *Machine learning,* de acuerdo con las necesidades del negocio y las metodologías de análisis de datos.  En el siguiente video conocerá, de forma general, la temática que se estudiará a lo largo del componente formativo. |

**Video introductorio**

| **Tipo de recurso** | Video spot animado | | | |
| --- | --- | --- | --- | --- |
| **NOTA** | **La totalidad del texto locutado para el video no debe superar las 500 palabras aproximadamente** | | | |
| **Título** | Profundización en *Data science* | | | |
| **Escena** | **Imagen** | **Sonido** | **Narración (voz en off)** | **Texto** |
| **1** | Ilustración de concepto informe de datos  Imagen: 228131\_i1401 | NA | Las empresas y organizaciones están generando datos de forma exponencial que necesitan ser procesados y analizados | Las empresas generan grandes cantidades de datos. |
| **2** | Personas sentadas en una mesa  Descripción generada automáticamente con confianza media  Imagen: 228131\_i1402 | NA | para hacer predicciones que orienten las estrategias de negocio en el futuro y poder generar de forma oportuna decisiones de valor, | Predecir el comportamiento del cliente. |
| **3** | Logro de objetivos y concepto de trabajo en equipo, crecimiento profesional y cooperación para el desarrollo del proyecto.  Imagen: 228131\_i1403 | NA | que les garanticen la permanencia y resiliencia en un mercado cada vez más competitivo. | Garantizar competitividad. |
| **4** | Ilustración del concepto de análisis  Imagen: 228131\_i1404 | NA | Con la ciencia de datos, las organizaciones pueden obtener perfiles de cada uno de sus clientes | Con los datos se puede conocer los clientes. |
| **5** | Ilustración de vector de concepto abstracto de estilo de vida minimalista. consumo minimalista, reducción de compras, vida sencilla, menor carga económica, estilo de vida de bajos gastos, pocas posesiones metáfora abstracta  Imagen: 228131\_i1405 | NA | y con ello, predecir cuáles son sus necesidades, | Necesidades. |
| **6** | Icono  Descripción generada automáticamente  Imagen: 228131\_i1406 | NA | por ejemplo; una aseguradora podría deducir a partir de modelos entrenados de *Machine learning*, qué tipo de pólizas debe ofrecer a un determinado cliente, | *Machine learning* puede predecir lo que desea el cliente. |
| **7** | Personas portadoras de iconos relacionados con el seguro.  Imagen: 228131\_i1407 | NA | ya sea un seguro de vida, una póliza de salud, un seguro vehicular, un esquema de pensiones, entre otros. | Oferta de servicios. |
| **8** | Ilustración del concepto de presentación digital  Imagen: 228131\_i1408 | NA | Lo anterior basado en el comportamiento de los datos para perfiles similares a los del nuevo usuario | A partir del análisis de los datos. |
| **9** | Seguro, protección contra pérdidas económicas. gestión de riesgos. seguro de salud, vida, propiedad, ingresos.  Imagen: 228131\_i1409 | NA | de manera que, si la persona está en un rango de edad entre los 20 y 30 años, le podría interesar contratar más un plan de pensiones que un seguro de vida; | Segmentación de clientes de acuerdo con el comportamiento de los datos.  (texto para dos escenas 9 y 10) |
| **10** | Ilustración de personas mayores con seguro de vida  Imagen: 228131\_i1410 | NA | los cuales, por lo general son adquiridos por personas con una edad avanzada. |  |
| **11** | Imagen que contiene Icono  Descripción generada automáticamente  Imagen: 228131\_i1411 | NA | Cada día se generan y almacenan millones de datos que están esperando ser explorados, transformados y cargados por las nuevas tecnologías para obtener su máximo valor. | Transformación de datos. |
| **12** | Ilustración del concepto de codificación manual  Imagen: 228131\_i1412 | NA | El dominio de los datos que otorga el *Data science* es una habilidad profesional muy demandada por las empresas | Oportunidades laborales para científicos de datos. |
| **13** | Ilustración de concepto informe de datos  Imagen: 228131\_i1413 | NA | que buscan personas que les brinden opciones para tomar decisiones basadas en datos y con esto minimizar el grado de incertidumbre que se genera en los nuevos mercados. | Toma de decisiones basadas den datos. |
| **14** | Pequeños científicos que desarrollan ia utilizando el aprendizaje automático. ilustración de vector plano de datos de computación cerebral. inteligencia artificial, tecnología, concepto científico para banner, diseño de sitios web o página web de inicio  Imagen: 228131\_i1414 | NA | Los sistemas que utilizan inteligencia artificial (IA) y en particular *Machine learning* son los que pueden aprender y mejorar de forma automática a partir del uso de la experiencia | Inteligencia artificial y aprendizaje de máquina para mejorar la experiencia del usuario. |
| **15** | Programadores y chatbot que procesan el lenguaje natural. procesamiento de lenguaje natural, lenguaje natural de chatbot, concepto de ciencia de lenguaje natural  Imagen: 228131\_i1415 | NA | sin que se hayan programado para ello y por lo tanto, deben resolver todos los problemas con datos previos que se hayan recolectado de experiencias pasadas. | Resolver problemas.  *Machine learning* para mejorar la oferta de servicios. |
| **16** | Imagen que contiene interior, agua, tabla, luz  Descripción generada automáticamente  Imagen: 228131\_i1416 | NA | Por ejemplo, un sistema de *Machine learning* que permite combatir el fraude en una organización |
| **17** | Robar concepto de datos con ladrón  Imagen: 228131\_i1417 | NA | debe estar soportado por datos relacionados con fraudes perpetrados en el pasado. | Aprendizaje automático basado en datos históricos. |
| **18** | Composición abstracta de medicina innovadora con imagen de android que muestra elementos de la ilustración de vector de interfaz médica hud  Imagen: 228131\_i1418 | NA | Por otra parte, están los sistemas de IA que tratan los problemas que no se pueden resolver mediante experiencias pasadas | Resolver problemas con IA. |
| **19** | Los científicos de personas diminutas identifican las emociones de las mujeres a partir de la voz y el rostro. detección de emociones, reconocimiento del estado emocional, concepto de tecnología de sensor emo. ilustración aislada violeta vibrante brillante  Imagen: 228131\_i1419 | NA | como el análisis de sentimientos y reconocimiento de voz que dependen del contexto o el estado de ánimo. | Por ejemplo, el análisis de los sentimientos. |
| **20** | Icono isométrico con chatbot de servicio de atención médica en línea hablando con personas 3d  Imagen: 228131\_i1420 | NA | En este componente formativo aprenderá las técnicas más relevantes de *Machine learning* | Técnicas de *machine learning.* |
| **21** | Icono de comunicación plana  228131\_i1221 | NA | y cómo aplicarlas a casos de usos prácticos que existen en el contexto organizacional. | Aplicaciones en las organizaciones. |
| **22** | Ilustración colorida del programador trabajando  Imagen: 228131\_i1422 | NA | Para ello, se utilizará el lenguaje de programación *Python* como elemento clave en el análisis de los datos.  El desarrollo del componente formativo empezará con el estudio de las librerías de *Python* más importantes en el *Data science* como *Numpy*, Pandas y Matplotlib. | *Python* herramienta para analizar datos. |
|  | Seguidamente abordará lo relacionado con el aprendizaje supervisado y no supervisado. Por último, se explorarán los modelos de *Machine learning*. |
| **23** | Código de programación resumen de antecedentes de la tecnología de desarrollador de software y script de computadora  Imagen: 228131\_i1423 | NA | Finalmente, con el estudio de este componente formativo se estará en la capacidad de emplear algoritmos de *Machine learning* | Crear algoritmos *machine learning.*  Necesidades del negocio. |
| **24** | Ilustración del concepto de empresa  Imagen: 228131\_i1424 | NA | de acuerdo con las necesidades del negocio y las metodologías de análisis de los datos. |
| **Nombre del archivo** | **228131\_v01** | | | |

**Desarrollo de contenidos**

# *Machine learning* y *Data science*

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| *Machine learning* es el proceso por el cual las máquinas “aprenden” con base en los datos o información previamente suministrada. A diferencia de la programación tradicional, en donde las instrucciones son específicas y dependen del tipo de problemática por resolver; en el *Machine learning,* se desarrollan algoritmos generales que tienen la capacidad de obtener patrones aplicables a diferentes tipos de datos. Artificial intelligence and data concept  Imagen: 228131\_i1425 | |
| Por lo tanto, una aplicación como *Machine learning,* que está diseñada para realizar, por ejemplo, una clasificación de fotos con números aleatorios, podría también utilizarse para clasificar imágenes de señales de tránsito, debido a que ambas problemáticas se resuelven con un algoritmo de clasificación de datos etiquetados. (Bobadilla, 2020)  CientÃ­ficos que estudian las conexiones neuronales. programadores escribiendo cÃ³digos para cerebro de mÃ¡quina  Imagen: 228131\_i1426 | |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| La Ciencia de datos o “*Data science*”, se utiliza para el estudio de grandes volúmenes de datos, los cuales, en su mayoría, son generados en Internet y, por consiguiente, cuando se diligencia un formulario o se realizan búsquedas e interacciones con el contenido de una página web, se estarán dejando registros en una base de datos, que podrán ser procesados con *Data science.*  Smart Male IT Programer Working on Desktop Computer. Software Development / Code Writing / Website Design / Database Architecture. Technical Department Office.  Imagen: 228131\_i1427 | |
| Para implementar técnicas de carga, manipulación y visualización de datos en *Machine learning* y *Data science,* con el lenguaje de programación Python, se estudiarán algunas de sus principales y más famosas librerías, como son NumPy, Pandas y Matplotlib. | |

| Cuadro de texto |
| --- |
| Para reforzar los conceptos sobre este tema, se le invita a ver el artículo “Uso de *Machine learning* en la creación de páginas web a medida de los usuarios”de Ana Milagros Quispe Rodríguez (2020), que se encuentra en el material complementario. |

## NumPy

| **Tipo de recurso** | Acordeón tipo 2 |
| --- | --- |
| **Introducción** | NumPy es una poderosa librería de Python, que se especializa en analizar y realizar cálculos numéricos, a los datos, con eficiencia, particularmente cuando el insumo se fundamenta en grandes cantidades de información. Estalibreríainserta en Python los arrays*,* para representar conjuntos de datos de un mismo tipo y varias dimensiones. Así mismo, incorpora funciones para manipular eficientemente los datos. |
| Education and complex concept  Imagen: 228131\_i1428 | |
| **Utilizar NumPy**  Para utilizar la librería de NumPy, esta se debe cargar de la siguiente manera:   | import numpy | | --- |   Una vez importada, se pueden utilizar todas las funcionalidades que incorpora dicha librería, cabe resaltar que es posible emplear la librería con el nombre por defecto “numpy”, o también de forma práctica, a través de una abreviación, como se muestra a continuación:   | import numpy as np | | --- |   De esta forma, en adelante, cuando se desee utilizar la librería numPy, solo se debe escribir “np”.  A continuación, un ejemplo básico de un array con numPy:   | import numpy as np  ejemplo\_lista = [10, 20, 30,40,50]  np.array(ejemplo\_lista) | | --- |   La salida del código anterior mostrará como resultado lo siguiente:   | array([10,20,30,40,50]) | | --- |   La expresión array() muestra que la salida del código ya no será una lista normal de Python, sino un arreglo de numPy. | |
| **Definición de unarray**  Para definir arreglos, se deben identificar las dimensiones de este y emplear la sintaxis que se muestra a continuación:   * Arreglos de 1 dimensión:  | arreglo\_1 = np.array([1,2,3,4])  print(arreglo\_1)  #Salida:  [10 20 30 40] | | --- |  * Arreglos de 2 dimensiones      | arreglo\_2 = np.array([[1,2,3,4], [5,6,7,8]])  print(arreglo\_2)  #Salida:  [[1 2 3 4]  [5 6 7 8]] | | --- |  * Arreglos de 3 dimensiones  | arreglo\_3 = np.array ([[[1,2,3], [4,5,6]], [[7,8,9], [11,12,13]]])  print(arreglo\_3)  #Salida:  [[[ 1 2 3]  [ 4 5 6]]  [[ 7 8 9]  [11 12 13]]] | | --- | | |
| **Acceso a unarray**  Para acceder a los datos que se encuentran en un array, se utilizan los índices, teniendo en cuenta que la posición inicial siempre es ‘0’; de manera que si se quiere tener acceso al primer valor del “arreglo\_2”, que es el número “1”, el código a utilizar es:   | print(arreglo\_2[0,0])  #Salida:  1 | | --- |   Por lo tanto, si se requiere acceder al segundo valor del “arreglo\_2”, que es el número 2, se debe:   | print(arreglo\_2[0,1])  #Salida:  2 | | --- | | |
| **Atributos de unarray**  Para conocer las características que describen un array, se pueden utilizar las siguientes funciones de numPy:   * ndim: devuelve la cantidad de dimensiones de un arreglo.  | arreglo\_2.ndim  #Salida:  2 | | --- |  * shape: devuelve las dimensiones del arreglo en forma de tupla.  | arreglo\_2.shape  #Salida:  (2, 4) | | --- |  * size: devuelve la cantidad de elementos del arreglo.  | arreglo\_2.size  #Salida:  8 | | --- |  * dtype: devuelve el tipo de datos que contiene el arreglo.      | arreglo\_2.dtype  #Salida:  dtype('int64') | | --- | | |
| **Creación de unarray**  NumPy se puede integrar con funciones que facilitan la creación de arreglos, entre las cuales se encuentran las siguientes:   * **Random:** permite crear un array con valores aleatorios.      | np.random.randn(2, 4)  #Salida:  array ([[ 1.08140609, 1.27142365, -0.94476481, 0.87678435],  [ 0.32765834, -0.82676935, 0.01746584, 0.89508321]]) | | --- |   Como se puede evidenciar, se creó un arreglo de 2 dimensiones, con 2 filas y 4 columnas de números aleatorios.   * **Zeros:** esta función devuelve un array con elementos en “cero”.   Ejemplo 1: se creará un arreglo unidimensional de 5 elementos.     | np.zeros(5)  #Salida:  array([0., 0., 0., 0., 0.]) | | --- |   Ejemplo 2: se creará un arreglo de 2 dimensiones de 2 filas y 3 columnas.   | np.zeros((2, 3))  #Salida:  array([[0., 0., 0.],  [0., 0., 0.]]) | | --- |  * **Arrange:** esta función de numPy es utilizada para retornar un arreglo con números enteros que estén en un intervalo, donde el primer parámetro indica el inicio con valores mayores o iguales (>=) y el segundo parámetro el final, con valores menores (<).  | np.arange(1,5)  #Salida:  array([1, 2, 3, 4]) | | --- | | |

| Cuadro de texto |
| --- |
| Para reforzar sus conceptos sobre este tema, consulte el video de Alberola Oltra, J. M. (2021), operaciones básicasconnumPy, que se encuentra en el material complementario. |

## Pandas

| **Tipo de recurso** | Acordeón tipo 1 |
| --- | --- |
| **Introducción** | Pandas es una librería de Python experta en la manipulación y análisis de datos y, además, ofrece visualizaciones que permiten simplificar la estadística descriptiva, mediante la aplicación de funciones simples; esta capacidad minimiza el esfuerzo en el análisis exploratorio de los datos, al igual que la exhibición de los resultados. (Toro, 2022)  Beneficios de usar Pandas:   * Capacidad de sacar provecho de métodos nativos para manipular datos faltantes. * Uso simple para describir y clasificar datos. * Creación rápida de gráficos. * Manejo eficiente de imágenes a través de indexación booleana. * Posibilidad para trabajar con muchos tipos de datos: enteros, cadenas de textos, números decimales, objetos, entre otros. * Fácil integración con la librería NumPy. |
| Modern flat design for analysis website banner. Vector illustration concept for business analysis, market research, product testing, data analysis.  Imagen: 228131\_i1429 | |
| **Clases de datos en Pandas**  Pandas soporta varias arquitecturas de datos, entre las que se destacan las series y los Dataframes. A continuación, se detalla cada una de ellas:     * **Series:** una serie es un objeto similar a un array de una dimensión que permite varias formas para la indexación de datos del mismo tipo (Toro, 2022).   Las series tienen un tamaño fijo que no se puede modificar; sin embargo, su contenido puede ser actualizado a través del índice que identifica a cada elemento dentro de la serie.    En el siguiente esquema, se muestra una serie que contiene algunas competencias para el análisis de datos:  **~~Figura 1~~** *~~Serie en pandas~~* | |
| **Creación de series**    Las series se pueden crear a partir de listas y diccionarios; estas estructuras de datos se basan en la librería numPy, pero con nuevas funcionalidades.   * + **Creación de series con listas:**en este tipo de series, se retorna un objeto con los elementos de la lista. La sintaxis es: pandas.Series(data, índice, tipo).Los parámetros, índices y tipo son opcionales; en caso de no pasarlos a la función, esta crea el índice por defecto, e infiere el tipo de datos.   En el siguiente ejemplo, se crea una serie a partir de una lista:   | import pandas as pd  serie\_lista = pd.Series(['Analisis', 'Programacion', 'Visualizacion', 'Despliegue'], dtype='string')  print(serie\_lista)  #salida:  0 Analisis  1 Programacion  2 Visualizacion  3 Despliegue  dtype: string | | --- | |  |   Similar que en la librería NumPy, para utilizar Pandas se deberá importar así: “import pandas as pd”.   * + **Creación de series con diccionarios:** este tipo de series retornan un objeto con los elementos del diccionario y las filas que se le ingresan como índices. La sintaxis es: pandas.Series(diccionario, índice). El parámetro índice es opcional, sino, se pasa a la función y se emplean como índice, los valores del diccionario:  | import pandas as pd  serie\_diccionario = pd.Series({'Analisis':4, 'Programacion':8, 'Visualizacion':3, 'Despliegue':5})  print(serie\_diccionario)  #salida:  Analisis 4  Programacion 8  Visualizacion 3  Despliegue 5  dtype: int64 | | --- | |  |   La salida indica que los índices son las descripciones y los datos son los valores numéricos. | |
| * **Dataframe:** los dataframe son objetos de estructuras de datos similares a una tabla, donde las columnas corresponden a elementos de un mismo tipo y las filas son registros que pueden contener distintos tipos de datos.   Los dataframes poseen dos índices, uno para las columnas y otro para las filas. El acceso a los datos se puede realizar a través de estos.  **~~Figura 2~~**  Dataframe en Pandas | |
| **Creación de dataframe:** para crear un dataframe, se encuentran varios métodos, entre los cuales, se destacan: diccionario de listas, lista de listas, lista de diccionario, por array, por ficheros CSV o Excel, entre otros.     * **Creación de dataframe por diccionario de listas:** Los dataframe por diccionarios, retornan un objeto, donde las columnas son los valores del diccionario y la lista de filas debe tener el mismo tamaño de las columnas. A continuación, se presenta un ejemplo para mayor claridad:  | import pandas as pd  data = {'nombre':['Juan', 'Maria', 'Jose', 'Pedro'],  'edad':[25, 31, 22, 45],  'habilidad':['Analisis', 'Programacion', 'Visualizacion', 'Despliegue'], }  dataframe = pd.DataFrame(data)  print(dataframe)  #Salida:  nombre edad habilidad  0 Juan 25 Analisis  1 Maria 31 Programacion  2 Jose 22 Visualizacion  3 Pedro 45 Despliegue | | --- | |  |  * **Creación de dataframe por lista de listas:** los dataframesde lista de listas, retornan un objeto con los elementos de la lista de listas, donde las columnas deben ser declaradas con el parámetro “columns” y las filas creadas con el mismo tamaño de las columnas; en caso contrario, se llenarán los datos faltantes con valores NaN.        | import pandas as pd  dataframe = pd.DataFrame([['Juan', 'Analisis'], ['Maria', 'Programacion'], ['Jose', 'Visualizacion']], columns=['Nombre', 'Habilidad'])  print(dataframe)  #Salida:  Nombre Habilidad  0 Juan Analisis  1 Maria Programacion  2 Jose Visualizacion | | --- | |  |  * **Creación de dataframe por lista de diccionario:** los dataframes de lista de diccionario, retornan un objeto con los elementos de la lista del diccionario, en donde se crean, por bloque, los valores, tanto de columnas como de filas.      | import pandas as pd  dataframe = pd.DataFrame([{'Nombre':'Juan', 'Habilidad': 'Analisis'}, {'Nombre':'Maria', 'Habilidad': 'Programacion'}, {'Nombre':'Jose'}])  print(dataframe)  #Salida:  Nombre Habilidad  0 Juan Analisis  1 Maria Programacion  2 Jose NaN | | --- | |  |  * **Creación de dataframe por array:** los dataframe por array, retornan un objeto donde los elementos de las columnas y filas son los declarados en el array.        | import pandas as pd  import numpy as np  dataframe = pd.DataFrame(np.random.randn(3, 4), columns=['1', '2', '3','4'])  print(dataframe)  #Salida:  1 2 3 4  0 1.520768 0.781591 -0.889866 0.731697  1 -1.582151 1.708422 -0.659952 -0.307672  2 0.178816 -2.236286 -0.196871 1.445878 | | --- | | |
| **Atributos de un dataframe**: la manipulación de los dataframes, se realiza a través de métodos o propiedades:   * + **info():** este método retorna información referente a la estructura del dataframe, como: cantidad de filas, columnas, índices, entre otros.   + **shape:** retorna la cantidad de filas y columnas.   + **size:** retorna la cantidad de elementos del dataframe.   + **columns:** retorna el nombre de las columnas del dataframe.   + **Index:** retorna el nombre de las filas del dataframe.   + **dtypes:** retorna el tipo de datos de la columna seleccionada del dataframe.   + **head(x):** retorna las primeras “x” filas del dataframe.   + **Tail(x):** retorna las últimas “x” filas del dataframe.   A continuación, se muestra, a través de un ejemplo, el resultado de cada uno de los métodos mencionados:     | import pandas as pd  datos = {'nombre':['Juan', 'Maria', 'Jose', 'Pedro'],  'edad':[25, 31, 22, 45],  'habilidad':['Analisis', 'Programacion', 'Visualizacion', 'Despliegue'], }  dataframe = pd.DataFrame(datos)  print(dataframe)  #Salida:  nombre edad habilidad  0 Juan 25 Analisis  1 Maria 31 Programacion  2 Jose 22 Visualizacion  3 Pedro 45 Despliegue  dataframe.info()  #Salida:  RangeIndex: 4 entries, 0 to 3  Data columns (total 3 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 nombre 4 non-null object  1 edad 4 non-null int64  2 habilidad 4 non-null object  dtypes: int64(1), object(2)  memory usage: 224.0+ bytes  dataframe.shape  #Salida:  (4, 3)  dataframe.size  #Salida:  12  dataframe.columns  #Salida:  Index(['nombre', 'edad', 'habilidad'], dtype='object')  dataframe.index  #Salida:  RangeIndex(start=0, stop=4, step=1)  dataframe.dtypes  #Salida:  nombre object  edad int64  habilidad object  dtype: object  dataframe.head(2)  #Salida:  nombre edad habilidad  0 Juan 25 Analisis  1 Maria 31 Programacion  dataframe.tail(2)  #Salida:  nombre edad habilidad  2 Jose 22 Visualizacion  3 Pedro 45 Despliegue | | --- | | |

| Cuadro de texto |
| --- |
| Para ampliar el tema de manejo de datos con Pandas, ver la siguiente videoclase:  VER VIDEO 228131\_v02 - Manejo de datos con pandas (en la carpeta CF14\_228131 – Videoclase) |

| Cuadro de texto |
| --- |
| Refuerce este tema, visualizando atentamente el video **Introducción a la librería Pandas**, que se encuentra en el material complementario. |

## Matplotlib

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| Matplotlib es una librería de Python que se especializa en la generación de gráficos de dos dimensiones, siendo una de las aplicaciones históricas dentro de su repositorio y empleada por la gran mayoría de los usuarios de la comunidad. Es de gran utilidad en el campo científico, debido a su potencial para elaborar gráficos de alta calidad. (Cuevas, 2018) | |
| Es importante mencionar que la librería Matplotlib depende de la librería NumPy para su correcto funcionamiento. Además, en principio se diseñó para apoyar los componentes del lenguaje de programación MATLAB. | |

**Crear gráficos con Matplotlib**

| **Tipo de recurso** | Rutas /Pasos Horizontal | |
| --- | --- | --- |
| **Introducción** | Matplotlib posee un módulo llamado Pyplotque proporciona una interfaz sencilla y de fácil uso para generar gráficos. También es significativamente valiosa para crear figuras en pocas líneas de código, sin declarar los respectivos ejes.  En los siguientes pasos, se mostrará de forma general, la creación de un gráfico usando Matplotlib y su módulo Pyplot: | |
| **Paso 1** | Paso 1. Importar la librería Matplotlib y el módulo Pyplot. | Imagen: 228131\_i1430 |
| **Paso 2** | Paso 2. Establecer el tipo de figura que tendrá el gráfico. | Imagen: 228131\_i1431 |
| **Paso 3** | Paso 3. Trazar los datos sobre los ejes del plano. | Imagen: 228131\_i1432 |
| **Paso 4** | Paso 4. Ajustar y/o personalizar el gráfico. | Imagen: 228131\_i1433 |
| **Paso 5** | Paso 5.Mostrar el gráfico. | Imagen: 228131\_i1434 |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| En el siguiente ejemplo, se encuentra la creación de un gráfico con la librería Matplotlib y el módulo Pyplot. Para esto, se crearán dos variables (x, y), a las que posteriormente se les asignan valores y, finalmente, se mostrará la gráfica resultante: | |
| | import matplotlib.pyplot as plt  X = [1, 2, 3, 4]  Y = [25, 12, 33, 30]  plt.plot(X,Y)  plt.show()  Salida:  **Figura 1** *Ejemplo de gráfico con Matplotlib* | | --- | | |

**Tipos de gráficos**

| **Tipo de recurso** | Acordeón tipo 1 |
| --- | --- |
| **Introducción** | Con Matplotlib se podrán crear los gráficos más comunes y útiles para el análisis de los datos; además se pueden realizar combinaciones entre ellos, lo que permite flexibilidad y personalización en las visualizaciones. A continuación, se presentan algunos ejemplos: |
| Imagen: 228131\_i1435 | |
| **Gráfico de dispersión**   * La función scatter(a, b)crea un diagrama de puntos, donde las coordenadas **“**a” y “b” forman los puntos del plano para el eje “x” y el eje “y”, respectivamente:      | import matplotlib.pyplot as plt  fig, grafico = plt.subplots()  grafico.scatter([10, 20, 30, 40], [20, 30, 10, 25])  plt.show()  Salida:  **~~Figura 4~~**  *~~Gráfico de Dispersión~~* | | --- | | |
| **Gráfico de líneas**  La función plot(a, b) crea una figura geométrica plana, donde las coordenadas “a” y “b” forman los puntos del plano para el eje “x” y el eje “y” respectivamente y luego se conectan los segmentos consecutivamente:   | import matplotlib.pyplot as plt  fig, grafico = plt.subplots()  grafico.plot([10, 20, 30, 40], [20, 30, 10, 25])  plt.show()  Salida:  **~~Figura 5~~**  *~~Gráfico de Línea~~* | | --- | | |
| **Gráfico de barras verticales**  La función bar(a, b) crea un diagrama de barras, donde la coordenada “a” es la posición de las barras en el eje “x” y “b” es la altura de las barras en el eje “y”.     | import matplotlib.pyplot as plt  fig, grafico = plt.subplots()  grafico.bar([10, 20, 30, 40], [20, 30, 10, 25])  plt.show()  Salida:  **~~Figura 6~~**  *~~Gráfico de barra~~* | | --- | | |
| **Histogramas**  La función hist(a, bins)crea un histograma a partir de la agrupación de las frecuencias de una lista “a”. El eje “y” muestra la cantidad de veces que se repite el dato que se encuentra en el eje “x”.     | import numpy as np  import matplotlib.pyplot as plt  fig, grafico = plt.subplots()  a = np.random.normal(5, 2, size=1000)  grafico.hist(a, np.arange(0, 11))  plt.show()  Salida:  **~~Figura 7~~**  *~~Histograma~~* | | --- | | |
| **Gráfico Boxplot**  La función boxplot(a) crea un diagrama de tipo caja y bigotes, a partir de los datos suministrados en la lista “a”. El objetivo principal con este tipo de gráficos, es el de mostrar las características comunes de los datos.     | import matplotlib.pyplot as plt  fig, grafico = plt.subplots()  grafico.boxplot([10, 20, 10, 20, 30, 40, 30, 30, 50, 70])  plt.show()  Salida:  **~~Figura 8~~**  *~~Boxplot~~* | | --- | | |

| Cuadro de texto |
| --- |
| Amplíe y refuerce conceptos y saberes sobre Matplotlib, explorando el video de Alberola Oltra, J. M. (2021) **Matplotlib**, el cual se encuentra en el material complementario. |

## Aprendizaje supervisado

| **Tipo de recurso** | Slider Presentación | |
| --- | --- | --- |
| **Introducción** | El aprendizaje supervisado es un tipo de *Machine learning* en el cual se toma una muestra de datos que contiene etiquetas para, luego, a través de diferentes algoritmos, entrenar un modelo que sea capaz de predecir o clasificar una nueva entrada de datos que no esté en los datos originales o de muestra. Por ejemplo, se tiene una serie de imágenes de diferentes animales con sus respectivos metadatos, asociados a la clase de animal que pertenece; así: (imagen1.png, “pájaro”), (imagen2.png,” perro”), (imagen3.png, “gato”). Con base en estos datos, se podría implementar un modelo de clasificación que sea capaz de predecir si una nueva imagen que se ingresa al modelo, es un pájaro, un perro o un gato. (Bobadilla, 2020) | |
| **Técnicas de aprendizaje supervisado**  El aprendizaje supervisado cuenta con dos técnicas principales; la regresión y la clasificación. Estas se diferencian fundamentalmente en el valor que intentan predecir, por un lado, la regresión predice números continuos que puede tomar un rango muy amplio de valores, por otra parte, la clasificación solo tiene dos opciones 1 y 0, que pueden interpretarse como si o no. Para mayor claridad sobre estas técnicas a continuación se ilustran por medio de un caso de uso:   * Regresión: se requiere predecir el costo que tendría una casa con base en las características de entrada como; estrato, número de habitaciones, antigüedad, ubicación, tamaño, entre otros. * Clasificación: se requiere predecir si un correo electrónico es o no, un correo malicioso o *spam*. | | Imagen: 228131\_i1436 |
| **Algoritmos de aprendizaje supervisado**  Entre los algoritmos más populares de aprendizaje supervisado están:   * Regresión lineal. * Regresión polinomial. * Vecinos más cercanos (KNN). * Máquina de soporte vectorial (SVM). * Árbol de decisión. * Regresión logística.   ~~En la siguiente~~ **~~unidad~~**~~, llamada “modelos de~~ *~~Machine learning”,~~*~~se ahondará en cada uno de los algoritmos antes descritos.~~ | | Imagen: 228131\_i1437 |

## Aprendizaje no supervisado

| Tipo de recurso | | Cajón de texto de color |
| --- | --- | --- |
| El aprendizaje no supervisado es una categoría del *Machine learning* que encuentra o infiere, por sí misma, estructuras y patrones en los datos, sin que estos se encuentren etiquetados, clasificados y categorizados. Su función principal, a diferencia del aprendizaje supervisado, no es predecir, sino que se encarga de describir las estructuras del conjunto de datos que se le suministre. | | |
| Por ejemplo, se requiere obtener un clasificador de tipos de autos, por lo que no sería viable etiquetar a los millones de autos que hay en las calles. En este caso, aplica un algoritmo de aprendizaje no supervisado, para agruparlos por las características similares. | | |
| Tipo de recurso | Cajón de texto de color | |
| **Algoritmos de aprendizaje no supervisado**  Entre los algoritmos de aprendizaje no supervisado se encuentran los de agrupamiento o *clustering,* y los de generalización; los algoritmos de agrupamiento, se encargan de dividir los datos con relación a características que no conoce y con base en su criterio; por otro lado, los algoritmos de generalización, construyen las variables particulares de alto nivel; por ejemplo, en los sistemas de recomendación o gestión de riesgos. | | |
|  | | |

# Modelos de *Machine learning*

| **Tipo de recurso** | Rutas / Pasos. Verticales 1 |
| --- | --- |
| **Introducción** | En este tema, ahondaremos en los distintos modelos de *Machine learning*, tanto de aprendizaje supervisado, como de aprendizaje no supervisado. En general, estos modelos cumplen con los siguientes pasos o agenda: |
| Imagen: 228131\_i1438 | |
| **Botón 1** | **Entendimiento de la problemática**  Conocer el objetivo y variable a predecir e identificar el tipo de modelo a aplicar: regresión, clasificación o agrupación. |
| **Botón 2** | **Recopilación de los datos**  Preparar y dividir los datos en entrenamiento y validación o testeo. |
| **Botón 3** | **Entrenamiento del modelo**  Aplicar el modelo a los datos de entrenamiento. |
| **Botón 4** | **Medición de desempeño**  Evaluarel modelo a partir de sus métricas y rendimiento. |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| En la siguiente figura, se ilustra el ciclo de un modelo de *Machine learning*, donde se inicia con el suministro de los datos; luego se aplica el modelo de *Machine learning* para generar una predicción y, finalmente, se actualiza para entrar en una nueva iteración.  **Figura 2** *Ciclo de un modelo de Machine learning* | |
| Antes de entrar en detalle con los modelos de *Machine learning,* es pertinente mencionar a una de las librerías más populares de Data Science en Python, llamada Scikit-learn*,* la cual contiene algoritmos de clasificación, regresión y agrupamiento. Además, es *open* *source* y de fácil aprendizaje, razón por la cual se ha convertido en la favorita de los científicos de datos. | |

| Cuadro de texto |
| --- |
| Para mayor profundización en la librería de Scikit-learn, acceder a su sitio oficial: <https://scikit-learn.org/stable/> |

| **Tipo de recurso** | Infografía estática |
| --- | --- |
| **Texto introductorio** | Explore los puntos interactivos del siguiente esquema y conozca, más y mejor, las fases del *Machine learning*: |
| **~~Figura 10~~** *~~Fases del machine learning~~*    Nota. Ruta de acceso <https://1drv.ms/u/s!AmN5mLAAnCdygcJ9Jxc2eOPeJ59GUQ?e=Br3teN> | |
| **Código de la imagen** | 228131\_i1439 |

## Regresión y clasificación

| **Tipo de recurso** | | Pestañas o tabs Verticales |
| --- | --- | --- |
| **Introducción** | | Los algoritmos de regresión y clasificación, forman parte del aprendizaje supervisado; por lo tanto, estos modelos aprenden de forma interactiva, con los datos, para realizar tareas específicas y predecir resultados conocidos. A continuación, se conocerá en detalle, el funcionamiento de estos algoritmos: |
| Imagen general que ilustre el tema    Imagen: 228131\_i1439 | | |
| **Algoritmos de regresión** | **Algoritmos de regresión**  Los algoritmos de regresiónson un tipo de aprendizaje inductivo que predice valores numéricos a partir de un dominio suministrado de atributos con valores continuos; estos se podrán describir a través de una función hipótesis para la predicción de eventos futuros. Es importante mencionar que los algoritmos de regresión no devuelven una predicción exacta sobre los eventos futuros, sino que entregan una aproximación. (Bosch, *et al*., 2019)  **~~Figura 11~~**  *Ejemplo de un algoritmo de regresión*  Gráfico, Gráfico de dispersión  Descripción generada automáticamente  **Casos de uso.** Los algoritmos de regresión pueden utilizarse con éxito en los siguientes escenarios: evaluar tendencias y estimaciones de ventas, analizar elasticidad en los precios de productos, analizar datos deportivos, evaluar riesgos en una compañía de seguros, entre otros. Para profundizar en la implementación de un modelo de regresión, se plantea un ejercicio desarrollado en Google Colab, donde se requiere predecir el costo de un incidente de ciberseguridad, con base en los equipos que fueron afectados:  VER ANEXO 01: REGRESIÓN.PDF | |
| **Algoritmos de clasificación** | **Algoritmos de clasificación**  Los algoritmos de clasificación dividen los datos en propiedades que se conocen de antemano, por ejemplo: documentos basados en el idioma, canciones clasificadas por género, camisas separadas por colores, entre otros. En este tipo de modelos los datos siempre deben estar etiquetados con propiedades que permitan que el algoritmo pueda establecer una clase en relación con sus características.  **~~Figura 12~~** *~~Ejemplo de un Algoritmo de Clasificación~~*    **Casos de uso.** Los algoritmos de clasificación, pueden utilizarse con éxito en los siguientes escenarios: clasificar mensajes maliciosos en correos electrónicos, detectar fraudes, analizar sentimientos basados en una publicación en redes sociales, realizar búsqueda de documentos similares, detectar el lenguaje de un documento o página web, entre otros.  Para profundizar en la implementación de un modelo de clasificación, se plantea un ejercicio desarrollado en Google Colab, donde se requiere predecir si un correo es o no, un *spam*.  VER ANEXO 02: Clasificacion.pdf | |

## K vecinos más cercanos (KNN)

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| El algoritmo del vecino más cercano o KNN adopta como base la capacidad de relacionar nuevas predicciones con experiencias pasadas, algo similar a lo que realizan los seres humanos cuando se basan en la experiencia para resolver nuevos problemas. | |
| **Casos de uso.** Algunos de los campos en los cuales se utiliza esta técnica, son:   * En soluciones avanzadas de visión computarizada, para la detección, por ejemplo, de delincuentes en un aeropuerto. * En sistemas de recomendación como el usado por Netflix, donde pueden predecir si a un determinado usuario le gustará una nueva película o serie. * En el campo de la medicina, para la detección de patrones en los datos genéticos que ayudan en el diagnóstico de enfermedades. | |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| Este algoritmo se utiliza ampliamente en contextos empresariales. En la siguiente tabla, se presentan algunas ventajas y desventajas de este algoritmo.  **Tabla 1**  *Ventajas y desventajas del algoritmo*   | Ventajas | Desventajas | | --- | --- | | Sencillo y efectivo. | No genera un modelo, lo que reduce la capacidad de entender la relación de las propiedades de la clase. | | No hace suposiciones sobre la distribución de datos subyacente. | Requiere la selección de un K. | | El entrenamiento del modelo es muy rápido. | El resultado de la clasificación es lento. | | |
|  | |

| Tipo de recurso | | Cajón de texto de color |
| --- | --- | --- |
| **Funcionamiento del modelo:**  El algoritmo elige el valor de K y los datos de entrenamiento que han sido previamente clasificados en distintas categorías, con sus respectivas etiquetas; posteriormente, toma cada registro que no ha sido etiquetado de los datos de prueba o testeo, e identifica cuáles son los datos de entrenamiento (los etiquetados) que se encuentran “más cercanos” y que tengan las mismas similitudes para, finalmente, asignar una etiqueta. | | |
|  | | |
| Tipo de recurso | Cajón de texto de color | |
| Para ilustrar este algoritmo con un caso cotidiano se puede implementar en la predicción de los nuevos amigos de un estudiante que llega a un salón clase. El algoritmo podría elegir a 6 compañeros que se sientan cerca de él y validar, por ejemplo, qué deporte practican cada uno de ellos para finalmente predecir que será amigo de los compañeros que practican el deporte con más resultados. | | |
|  | | |

## Máquina de soporte vectorial (SVM):

| **Tipo de recurso** | Acordeón tipo 1 |
| --- | --- |
| **Introducción** | El algoritmo de máquina de soporte vectorial o SVM, se puede utilizar tanto para problemas de regresión, como de clasificación, aunque es más empleado por su facilidad de implementación en modelos de clasificación. El principio detrás de este algoritmo es el de trazar dos líneas entre los puntos con mayor distancia entre ellos, para hallar la forma óptima de ordenar las clases. Los vectores que separan los puntos, se llaman vectores de soporte. |
| Imagen general que ilustre el tema    Imagen: 228131\_i1440 | |
| **Algoritmo de máquina de soporte vectorial**  El principio detrás de este algoritmo es trazar dos líneas entre los puntos con mayor distancia entre ellos para hallar la forma óptima de ordenar las clases. Los vectores que separan los puntos se llaman vectores de soporte.  **~~Figura 13~~** *~~Máquina de Soporte Vectorial~~* | |
| Casos de usos  Algunos de los campos en los cuales se utiliza esta técnica, son:   * Reconocimiento visual de caracteres. * Identificación automática de rostros para facilitar el enfoque de cámaras digitales. * Filtrar correos maliciosos o *spam*. * Identificación de imágenes por parte de los satélites (clasificar las partes que corresponden a agua, tierra, nubes, montañas, etc.)   Para mayor profundización en la implementación de este algoritmo consultar la documentación oficial de Scikit-learnen: <https://scikit-learn.org/stable/modules/svm.html> | |

## Árboles de decisión

| **Tipo de recurso** | Acordeón tipo 1 |
| --- | --- |
| **Introducción** | Este algoritmo utiliza la estructura de un árbol para diseñar la relación entre las propiedades de los datos y las posibles predicciones. |
| Imagen general que ilustre el tema    Imagen: 228131\_i1441 | |
| **Algoritmo árbol de decisión**  El principio básico de este algoritmo es la aplicación del enfoque “divide y vencerás”; dado que separa los datos en varios subconjuntos que posteriormente son divididos nuevamente en subconjuntos más pequeños y así sucesivamente hasta que se cumple con el objetivo de obtener datos homogéneos o el criterio de identificación.  A continuación, se detallan algunas ventajas y desventajas de esta técnica:  **Tabla 2**  *Ventajas y desventajas del algoritmo*   | Ventajas | Desventajas | | --- | --- | | Requieren poca preparación de los datos para la implementación del modelo. | Se pueden crear árboles demasiado complejos que no caractericen bien los datos. | | Permiten tratar problemas que requieren varias salidas o predicciones. | Predicciones no uniformes, sino aproximaciones. | | Fáciles de aprender e interpretar sus salidas. | Inestabilidad en sus estructuras, ya que una pequeña variación puede generar un árbol totalmente distinto. | | |
| Casos de usos: algunos de los campos en los cuales se utilizan esta técnica son:   * Predicciones sobre la oferta de un determinado producto a un cliente. * Estimar el valor a cobrar de la prima de seguros a los asegurados. | |

| **Tipo de recurso** | Infografía estática |
| --- | --- |
| **Texto introductorio** | Los árboles de decisión, son ampliamente usados en la solución de problemas cotidianos, porque permiten comparar acciones y resultados, basados en la relación costo/beneficio. |
| El siguiente esquema favorece la comprensión del proceso de creación de un árbol de decisión. Explore cada uno de sus puntos interactivos:  **Figura 3** *Creación de árbol de decisión*    Ruta de Acceso: <https://1drv.ms/u/s!AmN5mLAAnCdygcJ2z87K6oIwrxHTiA?e=YzSFQC> | |
| **Código de la imagen** | Imagen: 228131\_i1442 |

## *Clustering*

| **Tipo de recurso** | Acordeón tipo 1 |
| --- | --- |
| **Introducción** | El algoritmo *clustering* o de agrupamiento, es un método de aprendizaje no supervisado que agrupa los datos en subconjuntos llamados *clusters*. Los objetos o datos que contienen los *clusters,* comparten atributos similares entre sí, pero distintos a los datos de otros *clusters*. |
| Imagen general que ilustre el tema    Imagen: 228131\_i1443 | |
| **Algoritmo *clustering***  Esta técnica también se denomina “segmentación” y es utilizada para hallar atributos que son implícitos en los datos, es decir, encuentran etiquetas parecidas entre los miembros del clúster y, a su vez, delimitan la distancia con otros *clusters*.  Un ejemplo clásico para la aplicación de la técnica de agrupamiento, son los comercios en línea; estos podrían estudiar a los clientes que realizan compras en cierta época del año y establecer sus preferencias. Con la aplicación de un algoritmo de *clustering,* se puede segmentar a los clientes por grupos, de acuerdo con las predicciones y tomar decisiones sobre los productos que se deben ofrecer y aplicarles las respectivas promociones o descuentos.  Analice la siguiente figura:  **Figura 4** *Clustering* | |
| Casos de usos: algunos de los campos en los cuales se utiliza esta técnica, son:   * Investigación y segmentación de mercado para fidelización y clasificación de clientes. * Reconocimiento de patrones. * Detección de fraudes a través de comportamientos anormales en los datos. * Procesamiento de imágenes. | |

## Redes neuronales *Deep learning*

| **Tipo de recurso** | Acordeón tipo 2 |
| --- | --- |
| **Introducción** | Las redes neuronales son algoritmos de inteligencia artificial, que procesan los datos de forma similar al cerebro humano. Hacen parte de una rama del aprendizaje de máquina llamado aprendizaje profundo o *Deep learning.* Las redes neuronales están formadas por nodos o neuronas que se conectan entre sí, para crear capas una debajo de la otra. |
| Imagen: 228131\_i1444 | |
| Casos de usos: algunos de los campos en los cuales se utiliza esta técnica son:   * Diagnóstico de enfermedades a través de la clasificación de imágenes médicas. * Publicidad y mercadeo basado en filtros de las redes sociales y análisis de la información. * Pronóstico de hechos financieros soportado en información histórica, por ejemplo, valores de acciones en la bolsa de valores. * Predicción de artículos nuevos que pueden generar interés a un usuario específico. * Corrección y sugerencia de palabras de forma automática. * Procesamiento de datos complejos como el análisis de ADN. | |
| **Aplicaciones**  Dentro de las aplicaciones más destacadas de las redes neuronales se encuentran las siguientes:   * **Visión artificial:** estas aplicaciones extraen información desde imágenes o videos y son procesadas con algoritmos de *Deep learning,* que facilitan la distinción y reconocimiento de forma similar a la visión humana. * **Reconocimiento de voz:** con las redes neuronales se puede analizar el habla humana independiente de las variaciones en su tono, idioma o acentos. Como ejemplo se resaltan los asistentes Alexa y Cortana de los gigantes tecnológicos Amazon y Microsoft respectivamente. * **Procesamiento de lenguaje natural:** la implementación de redes neuronales permite el procesamiento de texto natural escrito por los humanos y obtiene un significado a partir de estos. Los chatbots o agentes virtuales son el ejemplo más representativo de esta aplicación. * **Motores de recomendaciones**: las redes neuronales son utilizadas para realizar seguimiento a la actividad de las personas en la web y con ello generar recomendaciones muy personalizadas. | |
| ***Deep learning* versus *Machine learning***  Para que los modelos de *Machine* *learning* funcionen correctamente, es necesario la intervención humana; los científicos de datos se encargan, de forma manual, de realizar los procesos de extracción, transformación y carga de los datos ETL. Por otro lado, con el *Deep learning,* el científico de datos solamente le suministra los datos al modelo sin procesarlos, porque la red de este se encargará de obtener los atributos de los datos por sí sola, e irá aprendiendo de forma independiente. | |

| Cuadro de texto |
| --- |
| Profundice y complete la información, explorando el PDF “**Redes neuronales artificiales**” que se encuentra en el material complementario. |

| Cuadro de texto |
| --- |
| ~~Para afianzar los conocimientos en~~ *~~Machine learning~~*~~, descargar los siguientes archivos de códigos.~~  **VER CARPETA CF14\_228131 - Códigos**  228131\_i432 |

Recuerde explorar los demás recursos que se encuentran disponibles en este componente formativo; para ello, diríjase al menú principal, donde encontrará la síntesis, una actividad didáctica para reforzar los conceptos estudiados, material complementario, entre otros.

**SÍntesis**

| **Tipo de recurso** | Síntesis |
| --- | --- |
| ~~Nombre del mapa:~~ *~~Data science~~* ~~Síntesis~~: | |
| **Introducción** | El siguiente mapa integra los criterios y especificidades de los conocimientos expuestos en el presente componente formativo: |
| **~~Figura 16~~** *~~Mapa mental - Data science~~*    Imagen: 228131\_i1445  *Nota.* Adaptado de Mindmeister (s.f), <https://mm.tt/map/2416234474?t=AabeDnq3dR> | |

**Actividad interactiva**

| **Tipo de recurso** | Actividad didáctica. Arrastrar y soltar | |
| --- | --- | --- |
| Esta actividad le permitirá determinar el grado de apropiación de los contenidos del componente formativo.  De acuerdo con la definición planteada en la columna izquierda, arrastre cada término al lugar que considere correcto de la columna derecha. | | Imagen: 228131\_i1446 |
| Se especializa en analizar y realizar cálculos numéricos a los datos con eficiencia, particularmente, cuando el insumo son grandes cantidades de información. | | Librería *Numpy* |
| Función de *numpy* que devuelve las dimensiones del arreglo en forma de tupla. | | *Shape* |
| Se especializa en la manipulación y análisis de datos. Además, ofrece visualizaciones que permiten simplificar la estadística descriptiva mediante la aplicación de funciones simples | | Librería Pandas |
| Función que devuelve las primeras “x” filas de un *Dataframe*. | | . head(x): |
| Función que crea un histograma a partir de la agrupación de las frecuencias de una lista “a”. | | hist(a, bins) |
| Esta técnica también se denomina “segmentación” y es utilizada para hallar atributos que son implícitos en los datos. | | *Clustering* |
| Se aplica para visión artificial, reconocimiento de voz, procesamiento de lenguaje natural y motores de recomendaciones, | | Redes Neuronales |
| **RETROALIMENTACIÓN** | | |
| **Respuesta correcta:**  ¡Felicitaciones! Usted ha superado la prueba con éxito. Ello evidencia su apropiación de los contenidos y temáticas. Procure hacer un refuerzo de los conceptos y acciones estudiados. ¡**Adelante**! | | **Respuesta incorrecta:**  ¡**Atención**! Sus respuestas sugieren hacer un repaso del componente formativo. Ello le permitirá apropiarse más y mejor de los temas, conceptos y acciones estudiados. ¡**Adelante**! |

**Material complementario**

| Tipo de recurso | Material complementario | | |
| --- | --- | --- | --- |
| Tema | Referencia APA del material | Tipo | Enlace |
| ***Machine learning* y *Data science*** | Alberola, J. (2021). *Matplotlib* [video]. Universidad Politécnica de Valencia. <https://riunet.upv.es/handle/10251/168966> | Video | <https://riunet.upv.es/handle/10251/168966> |
| ***Machine learning* y *Data science*** | Alberola J. (2021). *Operaciones básicas con Numpy* [video]. Universidad Politécnica de Valencia.<https://riunet.upv.es/handle/10251/168551> | Video | <https://riunet.upv.es/handle/10251/168551> |
| ***Machine learning* y *Data science*** | Lluch, J. (2022). *Introducción a la librería Pandas.* [video]. Universidad Politécnica de Valencia. <https://riunet.upv.es/handle/10251/183074> | Video | <https://riunet.upv.es/handle/10251/183074> |
| ***Machine learning* y *Data science*** | Quispe, A. (2020). Uso de Machine Learning en la creación de páginas web a medida de los usuarios. *Campus,* *25*(30), p. 337-344. <https://www.usmp.edu.pe/campus/pdf/revista30/articulo9.pdf> | Artículo | <https://www.usmp.edu.pe/campus/pdf/revista30/articulo9.pdf> |
| **Modelos de *Machine learning*** | Repetur, A. (2019). *Redes neuronales artificiales.* [Trabajo final presentado para obtener el título de  Licenciado en Ciencias Matemáticas]. Universidad Nacional del Centro de la Provincia de Buenos Aires. <https://underpost.net/ir/pdf/redes/ciencias_matematicas_redes_neuronales.pdf> | Trabajo final | <https://underpost.net/ir/pdf/redes/ciencias_matematicas_redes_neuronales.pdf> |

**GLOSARIO**

| **Tipo de recurso** | Glosario |
| --- | --- |
| Aprendizaje no supervisado | El aprendizaje no supervisado es una categoría del *Machine learning* que encuentra o infiere por sí mismo estructuras y patrones en los datos sin que ellos estén etiquetados, clasificados y categorizados. |
| Aprendizaje Supervisado | es un tipo de *Machine learning* en el cual se toma una muestra de datos que contiene etiquetas para, luego, a través de diferentes algoritmos, entrenar un modelo que sea capaz de predecir o clasificar una nueva entrada de datos que no esté en los datos originales o de muestra. |
| *Clustering* | es un método de aprendizaje no supervisado que agrupa los datos en subconjuntos llamados *clusters*. |
| *Machine learning* | Es el proceso por el cual las máquinas “aprenden” con base en los datos o información previamente suministrada. |
| Matplotlib | Es una librería de Python que se especializa en la generación de gráficos de dos dimensiones, siendo una de las aplicaciones históricas dentro de su repositorio y empleada por la gran mayoría de los usuarios de la comunidad. |
| Modelos de clasificación | Dividen los datos en propiedades que se conocen de antemano. |
| Modelos de regresión | Se trata de un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica, en función de las variables independientes o predictoras. |
| NumPy | Es una poderosa librería de Python que se especializa en analizar y realizar cálculos numéricos a los datos con eficiencia, particularmente cuando el insumo son grandes cantidades de información. |
| Pandas | Es una librería de Python experta en la manipulación y análisis de datos. Además, ofrece visualizaciones que permiten simplificar la estadística descriptiva mediante la aplicación de funciones simples. |
| Redes neuronales | Son algoritmos de inteligencia artificial que procesan los datos de forma similar al cerebro humano. |
| Visión artificial | Aplicación de las redes neuronales que extraen información desde imágenes o videos y son procesadas con algoritmos de *Deep learning* que facilitan la distinción y reconocimiento de forma similar a la visión humana. |

**Referentes bibliográficos**

| **Tipo de recurso** | Bibliografía |
| --- | --- |
| Bobadilla, J. (2020). *Machine learning y deep learning: usando python, scikit y keras.* Ediciones de la U. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=10277> | |
| Bosch, A. Casas, J. & Lozano,T. (2019). *Deep learning: principios y fundamentos.* Editorial UOC. <https://elibro-net.bdigital.sena.edu.co/es/ereader/senavirtual/126167> | |
| Cuevas, A. (2018). *Aplicaciones gráficas con Python 3.* Ediciones de la U. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=9084> | |
| Toro, F. (2022). *Ciencia de los datos con Python.* Ecoe Ediciones. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=20994> | |
|  | |



