**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| Programa de formación | Creación de modelos de predicción y aplicación de algoritmos de predicción con Python |

|  |  |  |  |
| --- | --- | --- | --- |
| Competencia | 220501114- Sistematizar datos masivos de acuerdo con métodos de analítica y herramientas tecnológicas | Resultado de aprendizaje | 220501114-02 - Aplicar el algoritmo de predicción seleccionado de acuerdo a los requerimientos establecidos.  220501114-03 - Evaluar el modelo construido de acuerdo con los resultados obtenidos. |

|  |  |
| --- | --- |
| Número del componente formativo | 02 |
| Nombre del componente formativo | Modelos de predicción |
| Breve descripción | Este componente formativo está enfocado en reconocer cómo funcionan los algoritmos de regresión y algoritmos de clasificación para realizar entrenamientos y predicción de datos, que permitan tomar decisiones asertivas en un contexto determinado. Se profundizará en el funcionamiento del modelo *Machine Learning* y Python, como lenguaje para realizar algoritmos de inteligencia artificial debido a su exploración estadística y librerías gráficas. |
| Palabras clave | Aprendizaje supervisado, algoritmos, inteligencia artificial, máquina de aprendizaje*,* Python. |

|  |  |
| --- | --- |
| Área ocupacional | 6 - Ventas y Servicios |
| Idioma | Español |

1. **TABLA DE CONTENIDOS**

**Introducción**

1. **Introducción a los modelos de predicción**

**2. Librerías Python**

**3. Algoritmos de aprendizaje supervisado en acción con Python**

3.1 Algoritmo de regresión con Python

3.2 Algoritmo de clasificación con Python

3.3 Árboles de clasificación con Python

3.4 K vecinos más cercanos con Python

3.5 Redes neuronales con Python

**4. Algoritmos de aprendizaje no supervisado en acción con Python**

4.1 *Clusterin*g con Python

4.2 Reducción de componentes con Python

**5. Midiendo el error de un modelo predictivo con Python**

5.1 Error en modelos de regresión

5.2 Evaluación de modelos de clasificación

**6. Informe final con Python**

**Síntesis**

**B.** I**NTRODUCCIÓN**

¿Sabías que a través de los años se ha ido aumentando el interés y la utilización del *Machine Learning*? Así es, su implementación cada vez es más constante; puede decirse que esta se ha posicionado como una de las disciplinas que se están aplicando casi en todas las áreas de investigación académica e industrial.

*Machine Learning* hace referencia a un grupo de algoritmos que permiten realizar la identificación de patrones presentados en colección de datos, a partir de los cuales se pueden generar diversas estructuras y modelos que permiten la predicción de información que no había podido ser analizada.

Es por ello que se han generado una serie de herramientas mediante las cuales se pueden acceder a métodos predictivos muy poderosos. Un ejemplo claro de ello es el lenguaje de programación Python. El siguiente video presenta, de modo introductorio, un poco más sobre este concepto y el lenguaje a trabajar para realizar algoritmos de inteligencia artificial.

CF02\_video\_Introducción

**C. DESARROLLO DE CONTENIDOS**

# **1. Introducción a los modelos de predicción**

Para comenzar es importante hacer un recorrido por el origen de Python*,* uno de los lenguajes de programación más usados en la actualidad, el cual fue creado en 1991 en el Instituto Nacional de Matemáticas de Holanda por Guido van Rossum,quien lo llamó Python, por un grupo de comedia denominado así.

Es uno de los lenguajes más populares que existen en el mundo, ya que sobrepasa en popularidad a R, y es idóneo para la ciencia de datos. El siguiente recurso da a conocer las características más representativas de la herramienta.

|  |
| --- |
| **CF02\_1\_Infografía interactiva\_características de Python** |

Ahora bien, después de conocer las cualidades de Python,y para empezar a entender el tema de predicciones, se va a trabajar el presente componente en función del siguiente ejercicio práctico denominado “**Entendiendo el crecimiento y la clasificación de los pingüinos del Archipiélago Palmer en la Antártida”,** preste mucha atención.

El biólogo y estadista de la Gran Bretaña Ronsal Fisher ha estudiado los pingüinos encontrados en el Archipiélago Palmer en la Antártida. A partir de ahí, ha generado características de tres diferentes especies de pingüinos, Adelié, Gentoo y Chinstrap en un *dataset* de 344 filas y 7 columnas.

|  |  |
| --- | --- |
| El *dataset* consta de los siguientes campos:   * **Species:** especie de pingüino que puede tomar valores Adelié, Gentoo o Chinstrap. * **Island**: isla de donde proviene el pingüino. * **bill\_length\_mm**: longitud del pico en mm. * **bill\_depth\_mm**: profundidad de del pico en mm. * **flipper\_length\_mm**: longitud de la aleta del pingüino. * **body\_mass\_g**: masa del cuerpo en gramos. * **sex**: sexo del pingüino. | Ilustración del concepto de familia de pingüinos vector gratuito |

Para estudiar un algoritmo de regresión se tomará una variable independiente para la longitud de la aleta del pingüino (flipper\_length\_mm); luego, con esta variable se predecirá la masa y el peso del pingüino (body\_mass\_g) y, para estudiar los algoritmos de clasificación y *clustering,* se usarán las variables necesarias para clasificar un pingüino en cualquiera de las tres especies.

Los pasos mínimos necesarios para implementar un modelo de predicción con Pythonse describen a continuación:

|  |
| --- |
| **CF02\_1\_Infografía estática\_pasos de implementación** |

Los datos se obtienen importando las librerías Pythonde acuerdo con el códig*o:*

|  |
| --- |
| import seaborn as sns  import numpy as np  df\_pinguinos = sns.load\_dataset("penguins")  df\_pinguinos.isnull().sum()  df\_pinguinos.dropna(inplace=True) |

Una muestra de estos datos es:

**Tabla 1.**

*Muestra de los datos*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| species | island |  | bill\_length\_mm | bill\_depth\_mm | flipper\_length\_mm | body\_mass\_g | sex |
| 0 | Adelie | Torgersen | 39.1 | 18.7 | 181.0 | 3750.0 | Male |
| 1 | Adelie | Torgersen | 39.5 | 17.4 | 186.0 | 3800.0 | Female |
| 2 | Adelie | Torgersen | 40.3 | 18.0 | 195.0 | 3250.0 | Female |
| 4 | Adelie | Torgersen | 36.7 | 19.3 | 193.0 | 3450.0 | Female |
| 5 | Adelie | Torgersen | 39.3 | 20.6 | 190.0 | 3650.0 | Male |

Inicialmente se construye una gráfica *pair lot* o gráfica de pares, con la cual se pueden observar todas las relaciones de las columnas o valores entre sí. Es importante tener en cuenta que las variables pueden ser continuas o categóricas, las cuales se usan para entender el mejor conjunto de características y explicar la relación entre dos variables o para formar los *clústeres* más separados, también permiten formar modelos de clasificación simples observando y separando los datos.

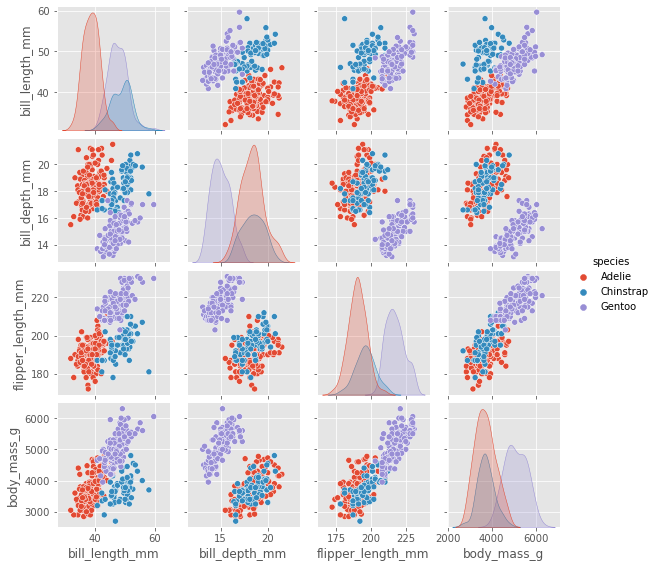
El comando a utilizar es:

|  |
| --- |
| sns.pairplot(df\_pinguinos, hue='species', height=2) |

El gráfico resultante es el que se observa a continuación:

**Figura 1**

*Gráfico de datos iniciales*



Analizando la gráfica se pueden determinar los siguientes resultados:

* La especie Adelie (naranja) se diferencia de las especies Gentoo (color morado) y Chinstrap (color azul), dependiendo de la longitud del pico (bill\_length\_mm).
* La especie Gentoo (morado) se diferencia de las especies Adelie (naranja) y Chinstrap (azul), dependiendo de la profundidad del pico.

## **2. Librerías Python**

Es importante destacar que una librería en el contexto de la programación corresponde a un conjunto de funcionalidades que permiten a determinado usuario llevar a cabo diferentes tareas que antes no podía realizar.

Las librerías de Pythondan respuesta a un conjunto de implementaciones que facilitan codificar dicho lenguaje, con el objetivo de crear interfaces independientes.

Cada librería de Python está conformada por diferentes módulos con funciones específicas y que varía dependiendo del sistema operativo en el que se utilizan.

El siguiente recurso expone las principales librerías de Python:

|  |
| --- |
| **CF02\_2\_Infografía interactiva\_librerías de Python** |

# **3. Algoritmos de aprendizaje supervisado en acción con Python**

Incluyen algoritmos que toman una muestra de datos y sus respectivas salidas, el objetivo principal es que el algoritmo aprenda cómo se relacionan los datos de entrada X y sus correspondientes salidas Y, a partirde un conjunto de entrenamiento escogido para tal fin.

Con el conocimiento adquirido se puede predecir nuevas salidas a través de nuevos datos **X**.

Estos algoritmos se llaman supervisados porque el modelo aprendió de respuestas y etiquetas conocidas de antemano en la fase de entrenamiento. Los métodos de aprendizaje supervisado son de dos tipos principales, esto depende del problema que se desea resolver y pueden ser: algoritmos de regresión y algoritmos de clasificación.

## **3.1 Algoritmo de regresión con Python**

La regresión lineal se utiliza para estimar valores reales como ventas, costos, cantidad de llamadas en una central, gastos, etc., en función de unas variables predictoras de tipo continuo. La mejor ecuación que se ajusta se llama **línea de regresión** y la ecuación utilizada es:

Donde:

: es la variable dependiente.

: es la pendiente de la línea de regresión.

: es la intersección con el eje Y.

La regresión lineal puede ser simple o múltiple, la primera se caracteriza por tener una variable independiente, contrario a la regresión múltiple que se caracteriza por tener más de una variable independiente.

Volviendo al ejemplo propuesto se utilizará la librería *scikit-learn,* que es una librería para el aprendizaje automático. En el siguiente recurso se pueden identificar los módulos a implementar y los pasos a seguir:

|  |
| --- |
| **CF02\_3\_1\_pasos verticales\_módulos** |

## **3.2 Algoritmo de clasificación con Python**

La construcción de los modelos de clasificación es una de las tareas más usadas en el aprendizaje automático, cuando los datos de los cuales se va a aprender cuentan con atributos de entrada y estos se encuentran relacionados con valores discretos, clases o etiquetas.

Las tareas para resolver un problema de clasificación involucra un conjunto de datos de entrenamiento, en los cuales se tienen puntos de datos etiquetados con sus categorías o clases correctas. Los modelos de clasificación más usados son la regresión logística*, Naive Bayes*, los soportes de máquinas vectoriales, los modelos no paramétricos como k vecinos más cercanos, los métodos de clasificación por medio de árboles de decisión y las redes neuronales.

|  |  |
| --- | --- |
| Las aplicaciones de los modelos de clasificación son infinitas. Algunos ejemplos son el reconocimiento de imágenes, el reconocimiento de voz, los juegos de computador, los carros autónomos, la predicción de enfermedades, las predicciones financieras como la bolsa de valores, la identificación de defectos de fabricación, la identificación de posibles fraudes bancarios, etc. | Los programadores de desarrollo de software investigan el código. vector gratuito |

**3.3 Árboles de clasificación con Python**

Son muy usados en la planeación, la estadística y el aprendizaje automático, usa una estructura de árbol de condiciones y las consecuencias para evaluar posibles eventos de un problema en particular.

Cada cuadrado es un nodo, los nodos de más abajo se denominan las hojas del árbol, imaginando un árbol al revés. Para iniciar las predicciones se comienza desde la raíz del árbol, es decir, desde el nodo superior; cada nodo es evaluado y saltando al siguiente nodo, según la decisión correcta.

Un árbol de decisiones se usa, entonces, para predecir la clase o el valor de la variable dependiente, aprendiendo de las reglas del árbol de decisiones encontradas con los datos de entrenamiento.

A continuación se explica cómo aplicar el árbol de decisiones al ejercicio planteado.

|  |
| --- |
| **CF02\_3\_3\_pasos horizontales\_árbol\_de\_decisiones** |

## **3.4 K vecinos más cercanos con Python**

Trabaja comparando la distancia de una observación a consultar con otras muestras del conjunto de entrenamiento y seleccionando los k vecinos más cercanos. La mayoría de las clases a las que pertenecen esos vecinos más cercanos será la clase a la que pertenezca la observación a consultar.

En el siguiente recurso se puede observar cómo hacer uso de dicho algoritmo.

|  |
| --- |
| **CF02\_3\_4\_slider\_aplicación de K vecinos cercanos** |

## **3.5. Redes neuronales con Python**

Son una técnica de inteligencia artificial basada en la forma como funcionan las neuronas en el cerebro. Consiste en unidades llamadas neuronas, las cuales tienen algunas características particulares, las cuales se muestran a continuación.

|  |
| --- |
| **CF02\_3\_5\_imagen\_interactiva\_características\_redes\_neuronales** |

|  |  |
| --- | --- |
| Revise el siguiente documento en el que se muestra un ejemplo sencillo con Python, haciendo uso de las redes neuronales. | **Ejemplo redes neuronales** |

# **4. Algoritmos de aprendizaje no supervisado en acción con Python**

En el aprendizaje supervisado el objetivo es mapear una entrada con una salida, con estos datos se puede entrenar un modelo y servir como predictor para datos nuevos. En el aprendizaje no supervisado solo existen datos de entrada, el objetivo es describir similitudes en los datos de entrada y observar patrones que permitan agruparlos en *clúster*. Todos los patrones, las asociaciones, las relaciones y los *clústeres* son extraídos de los mismos datos.

## **4.1 *Clustering* conPython**

La agrupación de datos es un método de clasificación no supervisado, cuyo principal objetivo es buscar patrones ocultos en los datos y con esto crear grupos específicos llamados *clústeres.* Uno de los algoritmos más usados es el *k-means* o k-medias por su fácil implementación, se inicia escogiendo un valor de k que se refiere a la cantidad de *clústeres* y la inicialización aleatoria de centroides.

En el siguiente recurso se realizala aplicación del *clustering* en el ejemplo que se ha venido trabajando:

|  |
| --- |
| **CF02\_4\_1\_pestañas\_horizontales\_clustering\_con\_Python** |

## **4.2. Reducción de componentes con Python**

La reducción de dimensiones y básicamente el análisis de componentes principales es un procedimiento estadístico para convertir un conjunto de observaciones de variables, posiblemente correlacionadas, en un conjunto reducido de variables no correlacionadas.

Cada uno de los nuevos componentes principales son elegidos de tal manera que describa la mayor parte de la varianza del modelo y todos estos componentes principales sean ortogonales entre sí. Las nuevas variables o componentes principales obtenidos tendrán un máximo aporte para explicar el resultado final.

En resumen, el principal objetivo del análisis de componentes principales es reducir un gran número de variables a un mínimo número de variables que expliquen de forma significativa un modelo.

A pesar de que el análisis de los componentes principales es bastante complejo, las librerías construidas en Pythonlo hacen muy sencillo. A través del siguiente recurso se puede percibir cómo implementar la reducción al ejercicio que se ha venido trabajando:

|  |
| --- |
| **CF02\_4\_2\_pestañas\_verticales\_reduciendo el componente** |

# **5. Midiendo el error de un modelo predictivo con Python**

## Tanto los modelos de predicción como los modelos de clasificación o *clustering* requieren ser medidos de una u otra forma para conocer si el modelo aprendió o no aprendió; tal como se mide una calificación de unos estudiantes sobre una materia en particular, de esta forma lo que se hace es cuantificar la calidad, la precisión y el rendimiento de las predicciones que se están ofreciendo.

## Estas mediciones se denominan métricas de evaluación. En algoritmos de clasificación existen medidas como la matriz de confusión y en los modelos de regresión existen el error medio absoluto, el error cuadrático medio y el R cuadrado, que se detallan a continuación.

## **5.1 Error en modelos de regresión**

En un modelo de regresión se estima el valor numérico de una cantidad no conocida de acuerdo con unas variables características, la diferencia entre el valor predicho y el valor real es el error.

Existen varios tipos de métricas para evaluar el error, los más conocidos son el error cuadrático medio o *root mean squared* *error* (RMSE), el error absoluto medio o *mean absolute* *error* (MAE) y el R cuadrado.

En el siguiente recurso educativo se puede conocer en qué consiste cada uno:

|  |
| --- |
| **CF02\_5\_1\_Acordeón\_métricas para evaluar el error** |

## **5.2 Evaluación de modelos de clasificación**

Una matriz de confusión es una herramienta de evaluación supervisada de aprendizaje automático que proporciona más información sobre la precisión general de un algoritmo de clasificación de aprendizaje automático. A diferencia de una métrica de precisión simple, que se calcula dividiendo el número de registros predichos correctamente por el número total de registros, las matrices de confusión devuelven cuatro métricas únicas con las que se puede trabajar.

* TP = verdaderos positivos – son aquellos casos correctamente predichos, es decir, el valor real es positivo y la predicción también lo es.
* FP = falso positivo – el valor real es negativo y el valor predicho es positivo.
* FN = falso negativo – en este caso el valor real es positivo y el valor predicho es negativo.
* TN = verdadero negativo – en este caso el valor real es negativo, igual que el valor predicho.

|  |
| --- |
| from sklearn.metrics import accuracy\_score  from sklearn.metrics import confusion\_matrix |

En Pythonse resume acá cómo calcular las métricas de arriba, usando el ejercicio de regresión logística con el *dataset* seleccionado.

A continuación se puede identificar el proceso para calcular cada una de las métricas establecidas:

|  |
| --- |
| **CF02\_5\_2\_Pasos horizontales\_Cálculo de métricas** |

Con estas métricas se puede evaluar cualquier modelo de clasificación en aprendizaje automático y revisar si son apropiadas para tomar decisiones y retroalimentar el algoritmo de aprendizaje automático.

# **6. Informe final con Python**

Anaconda es una *suite* compleja para la ciencia de datos, las aplicaciones de aprendizaje automático, el procesamiento de datos y las predicciones, etc., es de código abierto y abarca gran cantidad de librerías para el desarrollo con Python. Una de las principales utilidades de Pythonaparte de ser un lenguaje de programación interesante es que proporciona todas las herramientas necesarias para realizar las tareas de programación, estadística, comunicación y transmisión de conocimiento.

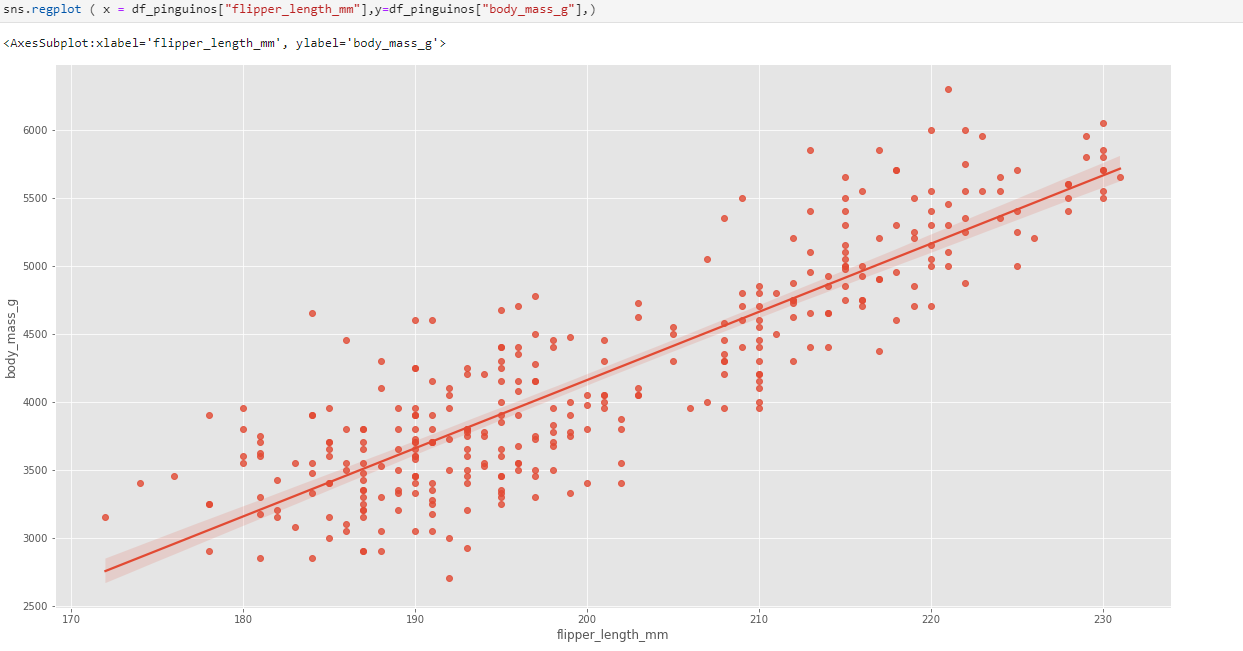
Entre los paquetes de Anaconda existe JupyterLab que es un editor basado en la Web con el que se crean *notebooks*, permite documentar los proyectos de la ciencia de datos, puesto que integra código fuente, ecuaciones, gráficos estadísticos, y de esta forma complementa proyectos de investigación, proyectos estadísticos y análisis exploratorio de datos.

Los informes finales de la ciencia de datos con Python se pueden generar usando esta herramienta y de una vez ir mostrando los resultados a los usuarios de un algoritmo de *Machine Learning.*

Es realmente sencillo mostrar un gráfico producto del análisis exploratorio de datos como, por ejemplo, la regresión lineal que se muestra en la figura.

**Figura 2**

*Regresión lineal*



Observe el siguiente video en el que se explica cómo configurar el entorno de trabajo de JupyterLab:

CF02\_6\_Video experto\_Pasos entorno de trabajo Jupiter Lab

**Síntesis**

En los últimos años han cobrado gran importancia los datos y más aun con la llegada de las nuevas Tecnologías de la Información y la Comunicación, pues estas permiten de cierta manera el registro y el control de información de forma ágil, sencilla y efectiva.

Actualmente, existen muchas herramientas y aplicaciones para analizar y predecir los datos, este componente en particular reconoce algunos algoritmos que permiten dichos procesos. Su correcta aplicación facilita la toma de decisiones acertada en cualquier tipo de contexto.

Una breve revisión de los temas vistos se encuentra en el siguiente esquema.

|  |
| --- |
| **CF02\_Síntesis** |

**D. ACTIVIDADES DIDÁCTICAS**

|  |  |
| --- | --- |
| **DESCRIPCIÓN DE LA ACTIVIDAD DIDÁCTICA** | |
| **Nombre de la actividad** | Algoritmos de aprendizaje |
| **Objetivo de la actividad** | Identificar los diferentes algoritmos de aprendizaje supervisado y no supervisado y los componentes principales implementados con *Python,* para el análisis de los datos y el desarrollo de predicciones a partir de los requerimientos establecidos. |
| **Tipo de actividad sugerida** | Selección múltiple |
| **Archivo de la actividad**  **(Anexo donde se describe la actividad propuesta)** | Carpeta Formatos DI: CF02\_Actividad didáctica |

**E. MATERIAL COMPLEMENTARIO**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tema** | **Referencia APA del material** | **Tipo de material**  **(Video, capítulo de libro, artículo, otro)** | **Enlace del recurso o**  **archivo del documento o material** |
| **2. Librerías de Python** | Escuela de Bayes. (2021). *Las 7 librerías más importantes para el análisis de datos en el lenguaje de programación de Pytho*n [video]. YouTube. <https://www.youtube.com/watch?v=guADTdAhLtA> | Video | <https://www.youtube.com/watch?v=guADTdAhLtA> |
| **3.3. Árboles de clasificación con Python** | AMP Tech. (2020). *Machine Learning episodio 3. Árboles de decisiones* [video]. YouTube. <https://www.youtube.com/watch?v=269QJ5joMCc> | Video | <https://www.youtube.com/watch?v=269QJ5joMCc> |
| **3.5. Redes neuronales con Python** | Ringa Tech. (2020). *Tu primera red neuronal en Python y Tensorflow* [video]. YouTube. <https://www.youtube.com/watch?v=iX_on3VxZzk> | Video | <https://www.youtube.com/watch?v=iX_on3VxZzk> |
| **4. Aprendizaje supervisado y no supervisado** | DotCSV. (2020) *¿Qué es el aprendizaje supervisado y no supervisado?* [video]. YouTube. <https://www.youtube.com/watch?v=oT3arRRB2Cw&t=27s> | Video | <https://www.youtube.com/watch?v=oT3arRRB2Cw&t=27s> |

**F. GLOSARIO**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| Aprendizaje automático: | rama de la inteligencia artificial, cuyo objetivo es implementar técnicas que permitan a los computadores aprender mediante un proceso de inducción del conocimiento. |
| Aprendizaje automático no supervisado: | hace referencia al proceso en el cual el algoritmo identifica patrones y saca conclusiones de los datos que se le proporcionan. |
| Aprendizaje automático supervisado: | hace referencia al proceso en el cual el algoritmo recibe datos de entrenamiento consistente en datos etiquetados. |
| Clasificación: | proceso de predecir con qué etiqueta se puede relacionar una observación. |
| Entrenamiento: | proceso que se realiza para que los modelos aprendan de los datos. |
| Evaluación: | análisis de eficiencia con el que el modelo predice los datos, generalmente se contrasta con una colección de prueba separada previamente. |
| Inteligencia artificial: | sistemas informáticos que pueden aprender como aprende un ser humano. |
| *K-means*: | lenguaje de alto nivel, usado para construir todo tipo de aplicaciones y muy usado en la ciencia de datos. |
| K-vecinos más cercanos: | algoritmo de clasificación básico, cuyo objetivo es detectar patrones usando una cantidad k de observaciones cercanas a la observación que se quiere clasificar. |
| *Machine Learning:* | aprendizaje automático. |
| Método del codo: | método consistente en ejecutar k-*means* para un *clúster* hasta n *clúster* y graficar la inercia por cada uno, que es la sumatoria de la distancia al cuadrado desde cada observación hasta el centroide, el valor k se toma de la gráfica. |
| Predicciones: | capacidad del modelo para clasificar entradas nuevas, de acuerdo con un entrenamiento previo. |
| Preprocesamiento: | manipulación que se realiza a los datos con el objetivo de entregarlos al modelo como este lo requiera. |
| *Python:* | proceso criptográfico que proporciona comunicaciones seguras a través de las redes, haciendo que la información entre extremos se transporte de forma segura mediante el uso de criptografía. |
| Redes neuronales: | está conformado por neuronas interconectadas en capas imitando el cerebro humano, cada conexión tiene un peso y cada neurona presenta un sesgo. |
| Regresión lineal: | modelo del aprendizaje de tipo supervisado, que consiste en obtener una relación entre variables de entrada o predictores de tipo numérico con una salida de tipo numérico. |
| Regresión logística: | modelo que realiza un prejuicio del resultado de una variable categórica a partir de variables predictoras. |

**G. REFERENCIAS BIBLIOGRÁFICAS**

Anaconda Inc. (2018). *Anaconda distribution*. Anaconda Inc. <https://docs.anaconda.com/free/anaconda/>

Codificandobits. (2021). *¿Se requiere SQL para trabajar en Machine Learning?* Codificandobits. <https://www.codificandobits.com/blog/sql-machine-learning/>

González, L. (2023). *Regresión logística – Teoría.* Aprende IA. <https://aprendeia.com/algoritmo-regresion-logistica-machine-learning-teoria/#Diferencias_entre_Regresion_Lineal_y_Regresion_Logistica>

Horst, A., Hill, A. y Gorman, K. (2020). *Palmerpenguins*. Palmer Penguins.

<https://allisonhorst.github.io/palmerpenguins/>

Roman, V. (2019). *Machine Learning: cómo desarrollar un modelo desde cero*. Medium. <https://medium.com/datos-y-ciencia/machine-learning-c%C3%B3mo-desarrollar-un-modelo-desde-cero-cc17654f0d48>

**H. CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor (es)** | Héctor Henry Jurado Soto | Experto temático - Contratista | Regional Cauca – Centro de Teleinformática y Producción Industrial | Junio 2022 |
| Jeimy Lorena Romero Perilla | Diseñadora instruccional | Regional Norte de Santander - Centro de la Industria, la Empresa y los Servicios | Junio 2022 |
| Carolina Coca Salazar | Asesora metodológica | Regional Distrito Capital- Centro de Diseño y Metrología | Julio 2022 |
|  | Julia Isabel Roberto | Correctora de estilo | Regional Distrito Capital - Centro de Diseño y Metrología | Septiembre 2022 |
|  | Miroslava González Hernández | Diseñadora instruccional | Regional Santander - Centro Industrial del Diseño y la Manufactura | Agosto 2023 |
|  | Rafael Neftalí Lizcano Reyes | Responsable Equipo desarrollo curricular | Regional Santander - Centro Industrial del Diseño y la Manufactura | Agosto 2023 |

**I. CONTROL DE CAMBIOS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** | **Razón del Cambio** |
| **Autor (es)** |  |  |  |  |  |