

**TP Inicial**

**Plataforma de Gestión de Recursos Humanos con ERP y CRM con Inteligencia Artificial**

Laboratorio de Construcción de Software

Comisión 02, miércoles y viernes

**Integrantes:**

Abal, Fernando Gabriel

Bianciotto, Matías Felipe

Gutiérrez, Diego Germán

Roldan Sambrana, Guadalupe Milagros

Sotelo, Evelyn Romina

**Repositorio GitHub:**

<https://github.com/Bianciotto/machine-learning-prototype>

**Investigación Teórica Parte 1**

**Parte 1: ERP, CRM e IA en RRHH.**

**¿Qué es un ERP?**

Un **ERP** (Enterprise Resource Planning) es un sistema de gestión empresarial que permite integrar todas las áreas de la empresa en un solo software. Esto incluye la gestión de finanzas, compras, ventas, producción, inventarios y recursos humanos, entre otros.

Uno de los mejores ejemplos de ERP es **SAP**, una plataforma que ofrece una amplia gama de soluciones empresariales para compañías de todos los tamaños y sectores. Otros ejemplos de ERP son **Oracle**, **Microsoft Dynamics**, **NetSuite** y **Odoo**.

**¿Qué es un CRM?**

Un **CRM** (Customer Relationship Management) es un sistema de gestión de relaciones con los clientes. Su objetivo principal es mejorar la relación con los clientes y aumentar la satisfacción y la fidelización.

Uno de los mejores ejemplos de CRM es **Salesforce**, una plataforma que ofrece una amplia gama de soluciones de gestión de ventas, marketing y atención al cliente. Otros ejemplos de CRM son **HubSpot**, **Zoho CRM** y **Microsoft Dynamics 365**.

**Diferencia entre ERP y CRM**

La principal diferencia entre el **ERP** y el **CRM** es que el primero se centra en la gestión interna de la empresa, mientras que el segundo se centra en la gestión de las relaciones con los clientes, es decir, en tareas y estrategias externas a la empresa.

Un ERP permite a las empresas automatizar y unificar los procesos internos, lo que mejora la eficiencia y la productividad. Por otro lado, un CRM permite a las empresas gestionar de manera efectiva las interacciones con los clientes, lo que aumenta la satisfacción y la fidelización.

**ERP (SAP SuccessFactors):**

**Propósito Primario:** Herramienta integral para la gestión del capital humano (HCM), abarcando desde la administración básica de RRHH hasta el desarrollo del talento y análisis estratégico.

**Funcionalidad en RRHH:**

* Gestión de la experiencia y necesidades del personal (engagement del empleado, alineación de metas y talento humano).
* Monitoreo de horarios, absentismo y productividad.
* Administración de servicios de RRHH y generación de reportes estratégicos.
* Implementación flexible (en la nube o híbrida).

**CRM (Salesforce):**

**Propósito Primario:** Diseñado principalmente para gestionar las relaciones con los clientes, aunque su funcionalidad puede extenderse a procesos de soporte interno como el servicio de RRHH.

**Funcionalidad en RRHH:**

* Permite la creación y gestión de tickets para atender las solicitudes de RRHH.
* Ofrece un marco de trabajo para gestionar tareas de incorporación, personalizable para adaptarse a necesidades específicas.
* Facilita la racionalización de solicitudes de servicio y proporciona una experiencia integrada en la incorporación de empleados.

**SAP** **SuccessFactors ERP** se centra en una **gestión integral y estratégica del talento y los recursos humanos**.

**Salesforce CRM** se centra en la **gestión de tareas de servicio y soporte.**

**Mejoras de eficiencia en procesos de RRHH con la IA.**

Optimización de la información: La IA puede procesar grandes cantidades de información, sobre los empleados y su desempeño, en tiempos relativamente cortos. En conjunto con su capacidad de automatización a tareas administrativas, se vuelve realmente útil en este ámbito.

Proceso de contratación eficiente: Una de las mayores ventajas de la IA en RRHH es que puede mejorar significativamente el proceso de contratación. Las herramientas impulsadas por IA pueden escanear miles de currículums, identificar a los candidatos más cualificados y programar entrevistas sin intervención humana.

Mejora de la experiencia del candidato: Los chatbots basados en IA pueden proporcionar a los candidatos información y asistencia instantáneas, incluso fuera del horario laboral habitual.

Mejora del compromiso de los empleados: La IA puede analizar los datos de los empleados para identificar patrones y tendencias, como altos índices de rotación o baja satisfacción laboral. Esta información se puede utilizar para implementar estrategias dirigidas a mejorar el compromiso de los empleados.

Mayor precisión en las evaluaciones del rendimiento: Las evaluaciones del rendimiento pueden ser un proceso subjetivo, y los prejuicios pueden influir en los resultados. La IA puede proporcionar comentarios imparciales y precisos mediante el análisis de datos como la productividad de los empleados, los registros de asistencia y los comentarios de los clientes. Esto puede ayudar a identificar áreas de mejor. Ayuda a identificar a los empleados con mayor potencial.

Ahorro de costos: puede reducir los costos asociados con la contratación, la formación y la retención de empleados, automatizar las tareas rutinarias de RRHH, liberando al personal para centrarse en iniciativas más estratégicas.

La IA también puede ayudar a personalizar las experiencias de los empleados ofreciéndoles programas de formación y desarrollo personalizados. Por ejemplo, las plataformas de aprendizaje.

La aplicación de la IA en recursos humanos puede presentar ciertos desafíos como: Prejuicios y discriminación, falta de interacción humana, preocupación por la privacidad y la seguridad de los datos, dependencia de la tecnología y falta de flexibilidad, costo de implementación (puede implicar un costo elevado de infraestructura tecnológica).

**Ventajas de ERP en Recursos Humanos**

* *Escalabilidad*: La escalabilidad se refiere a la capacidad que tiene un sistema cualquiera para adaptarse al crecimiento de las demandas que recibe.
* *Automatización de tareas*: Es posible establecer flujos de trabajo que pueden ahorrar mucho tiempo en tareas repetitivas.
* *Disponibilidad de la información*: Es posible generar flujos en donde los encargados de otras áreas puedan solicitar acceso a cierta información y obtenerla inmediatamente.
* *Reducción de los márgenes de error*: A la hora de hacer cálculos para medir al personal, permiten un cálculo exacto y preciso de todos los factores que envuelven el desenvolvimiento del personal.
* Fluidez de la comunicación: Con este sistema se tiene la información centralizada lo cual permite que la comunicación con otros sectores de la empresa sea más fluida. Mejorando la funcionalidad de distintos sectores de la empresa en conjunto.

### **Desafíos de ERP**

* *Mayor complejidad:* ERP es una herramienta de gestión empresarial que abarca todos los aspectos y puede ser bastante compleja.
* *Adición de costos a corto plazo:* no solo el costo del software, sino el costo del tiempo y los recursos necesarios para implementar el sistema y capacitar a los empleados de todos los departamentos.
* *Más tiempo:* Dado que ERP es personalizable, en lugar de ser un software único para todo tipo de organización, puede requerir mucho tiempo.

**Ventajas de CRM en Recursos Humanos**

* *Es un aliado en la toma de decisiones:* permite un seguimiento completo sobre el proceso de venta con objeto de tomar las decisiones idóneas para el negocio.
* *Acceder a la información en tiempo real.*
* *Optimizar los procesos de venta.*
* *Mayor número de ventas.* Al conocer el comportamiento de los clientes y saber con detalle sus preferencias, se puede adelantar sus necesidades y potenciar las ventas.
* *Mejorar el servicio de atención al cliente.* Permite solucionar los problemas o dudas que plantean los clientes de forma rápida y efectiva.

**Desafíos de CRM**

* *Resistencia al Cambio*: Introduciendo una nueva forma de trabajar que puede generar incertidumbre entre los empleados.
* *Integración con Sistemas Existentes:* La falta de integración puede llevar a una duplicación de datos o a la pérdida de información crucial.
* *Personalización del CRM:* Configurarlo para que se ajuste perfectamente a las necesidades de la empresa puede ser un desafío.
* *Capacitación del Personal:* Puede resultar en una subutilización de las funcionalidades del sistema, afectando el retorno de la inversión.

**Fuentes consultadas:**

[**https://www.oracle.com/ar/erp/erp-vs-crm/#link2**](https://www.oracle.com/ar/erp/erp-vs-crm/#link2)

[**https://www.oracle.com/ar/human-capital-management/what-is-hcm/**](https://www.oracle.com/ar/human-capital-management/what-is-hcm/)

[**https://www.oracle.com/ar/human-capital-management/ai-at-work/#recruiting**](https://www.oracle.com/ar/human-capital-management/ai-at-work/#recruiting)

[**https://salesforce-developers.com/**](https://salesforce-developers.com/)

[**https://openwebinars.net/blog/los-mejores-erp-para-el-area-de-rrhh/**](https://openwebinars.net/blog/los-mejores-erp-para-el-area-de-rrhh/)

[**https://www.ibm.com/mx-es/think/insights/enterprise-resource-planning-advantages-disadvantages**](https://www.ibm.com/mx-es/think/insights/enterprise-resource-planning-advantages-disadvantages)

[**Ventajas e inconvenientes de implementar la IA en recursos humanos | Pagina 66, Noticias de Alcoy y de El Comtat**](https://pagina66.com/art/127294/ventajas-e-inconvenientes-de-implementar-la-ia-en-recursos-humanos)

**https://www.softwaredoit.es/software-erp/index.html**

[**https://www.softwaredoit.es/software-crm/index.html**](https://www.softwaredoit.es/software-crm/index.html)

[**https://blog.opresmedia.com/desaf%C3%ADos-comunes-durante-la-implementaci%C3%B3n-del-crm-enfrentando-el-cambio-con-hubspot**](https://blog.opresmedia.com/desaf%C3%ADos-comunes-durante-la-implementaci%C3%B3n-del-crm-enfrentando-el-cambio-con-hubspot)

**https://www.seidor.com/es-ar/blog/ventajas-crm**

**Investigación Teórica Parte 2**

**Parte 2: Herramientas y Técnicas de Machine Learning.**

**¿Qué es Scikit-learn?**

**Scikit-learn** (anteriormente scikits.learn) es una biblioteca para aprendizaje automático de software libre para el lenguaje de programación Python. Es una de las más populares y utilizadas en el mundo, debido a su facilidad de uso y gran cantidad de algoritmos de aprendizaje automático disponibles.

**¿Cómo se usa en Python para desarrollar modelos de machine learning?**

Se creó para ayudar a simplificar el proceso de implementación del aprendizaje automático y los modelos estadísticos en Python. La biblioteca permite a los profesionales implementar rápidamente una amplia gama de algoritmos de aprendizaje automático supervisados y no supervisados mediante una interfaz coherente.

**Machine learning** (ML) es una rama de la inteligencia artificial (IA) y la informática que se centra en el uso de datos y algoritmos para permitir que la IA imite la forma en que los humanos aprenden, mejorando gradualmente su precisión.

Un modelo de Machine Learning es un programa de ordenador que aprende a través de la experiencia. Es decir, se alimenta de un conjunto de datos de entrenamiento para aprender patrones y relaciones entre las variables que le permiten hacer predicciones precisas sobre nuevos datos.

*Scikit-learn* es importante para *Machine Learning* por varias razones:

* Es fácil de usar: Está diseñada para ser fácil de usar y de entender para cualquier persona que tenga conocimientos básicos de programación y estadística. Esto significa que se puede comenzar a usar Scikit-learn rápidamente y empezar a construir modelos de aprendizaje automático sin tener que preocuparte por detalles complejos de implementación.
* Resulta completa: Además ofrece una amplia variedad de algoritmos de aprendizaje automático para resolver una amplia gama de problemas, desde la regresión lineal hasta el aprendizaje profundo. Y, también ofrece herramientas para la selección de características, la optimización de modelos y la evaluación de modelos.
* Es de código abierto: Scikit-learn es de código abierto y es compatible con otros paquetes de Python de código abierto, lo que significa que los usuarios pueden personalizar y ampliar la biblioteca según sus necesidades. Además, también es fácil de integrar con otras herramientas y bibliotecas de Python, lo que la convierte en una herramienta imprescindible para cualquier desarrollador que trabaje en el campo de machine o deep learning.

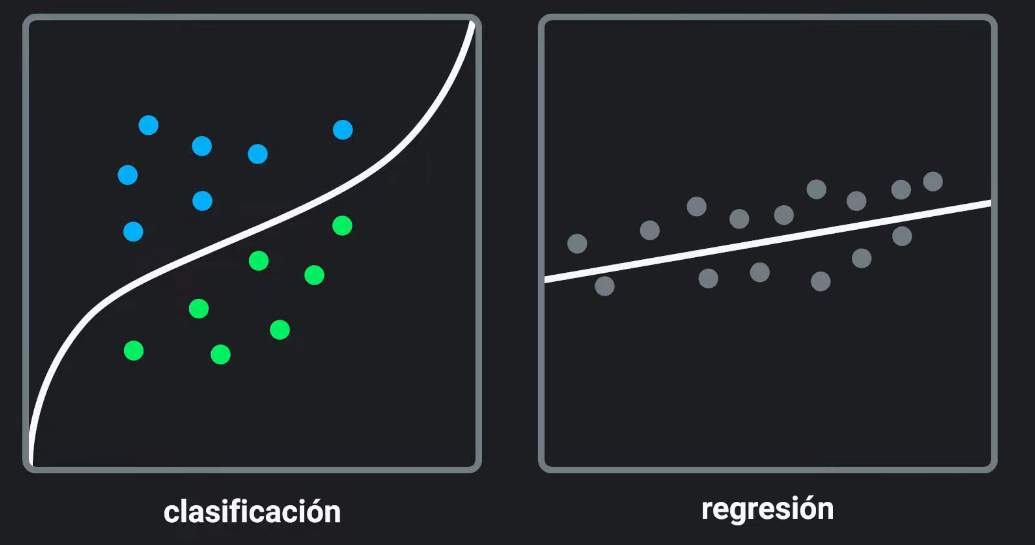
Cuando entrenamos un modelo de *Machine Learning* los enfoques más utilizados para hacerlo son:

* **Aprendizaje supervisado**: el algoritmo aprende de un conjunto de datos que incluye los resultados deseados / etiquetas, con el objetivo de asociar la entrada con la salida.

Ejemplos: regresión lineal, regresión logística, árboles de decisiones.

A su vez este mismo puede ser de dos tipos:

* + Clasificación: el modelo se entrena y aprende a clasificar los datos.
  + Regresión: el modelo aprende a predecir valores continuos.



* **Aprendizaje no supervisado**: el conjunto de datos no contiene etiquetas, el modelo aprende sin una guía, se busca encontrar patrones en los datos mismos.

Ejemplos: K-means, isolation forest.

* **Aprendizaje semi-supervisado**: es un enfoque híbrido que combina elementos del aprendizaje supervisado y el no supervisado. El algoritmo aprende de un conjunto de datos con y sin etiquetas.

Un ejemplo muy conocido de esto es Google Fotos, donde al cargar fotos, el servicio detecta automáticamente la presencia de una persona en varias fotos (no supervisado) y, por otro lado, cuando indico el nombre de una persona en una foto, el servicio automáticamente asocia ese nombre a todas las demás fotos donde aparece esa persona (supervisado).

**Regresión lineal y aprendizaje automático**

Más allá de su utilidad inmediata para determinar el precio de la vivienda, la regresión lineal desempeña un papel importante en el aprendizaje automático.

* Es un modelo básico para comprender técnicas más avanzadas como **la regresión logística**, las **redes neuronales** y **las máquinas de vectores de soporte**.
* Se entrena rápidamente, por lo que es ideal para la creación rápida de prototipos.
* También sirve como base de comparación. Si los modelos más avanzados no lo superan significativamente, puede que su complejidad añadida no esté justificada.
* A diferencia de algunas técnicas (como el aprendizaje profundo), es fácilmente interpretable.
* Puede ayudar en la **selección de características**, identificando los predictores más útiles.

A pesar de su simplicidad, la regresión lineal sigue siendo una herramienta indispensable en el aprendizaje automático debido a su eficacia, interpretabilidad y versatilidad.

**Regresión lineal y la biblioteca scikit-learn**

La biblioteca scikit-learn facilita la aplicación de la regresión lineal. Esta biblioteca tiene muchas ventajas.

* Tiene una interfaz coherente. El código necesario para aplicar los distintos algoritmos de ML es similar.
* El código es sencillo, y se han eliminado los complejos detalles matemáticos y de implementación. Por ejemplo, para ajustar un modelo a los datos de entrenamiento, basta con utilizar la línea model.fit(X\_train, y\_train).
* Facilita el acceso a los coeficientes del modelo.
* Proporciona métricas integradas para evaluar el rendimiento del modelo.
* Es fácil integrar la regresión lineal (o cualquier otro algoritmo de ML) con pasos de preprocesamiento, como el escalado y la selección de características, utilizando Pipeline.

**Regresión Logística**

La regresion logıstica estima la probabilidad de un suceso en funcion de un conjunto de variables explicativas y en la construccion del modelo no hay ningun supuesto en cuanto a la distribucion de probabilidad de las variables por lo que puede incluirse cualquier tipo de variable.

Cuando se desea clasificar un sujeto dentro de uno o mas grupos previamente determinados a partir de un conjunto de caracterısticas observadas del sujeto, es razonable pensar en la utilizacion de una medida probabilıstica.

El modelo de regresion logıstica puede considerarse como una formula para calcular la probabilidad de pertenencia a uno de los grupos, de manera que este estima la probabilidad de que una observacion pertenezca a uno de los grupos. La interpretacion del resultado de la aplicacion de esta metodologıa es sencilla por tratarse en terminos de probabilidad.

## **Tipos de análisis de Regresión Logística**

## Hay tres enfoques para el análisis de regresión logística basados en los resultados de la variable dependiente.

## Regresión logística binaria

## Regresión logística multinomial

* Regresión logística ordinal

### **Regresión logística binaria**

La regresión logística binaria funciona bien para problemas de clasificación binaria que solo tienen dos resultados posibles. La variable dependiente solo puede tener dos valores, como sí y no o 0 y 1.

Aunque la función logística calcula un rango de valores entre 0 y 1, el modelo de regresión binaria redondea la respuesta a los valores más cercanos. Por lo general, las respuestas por debajo de 0,5 se redondean a 0 y las respuestas por encima de 0,5 se redondean a 1, de modo que la función logística devuelve un resultado binario.

**¿Cómo funciona Regresión Logística?**

En la regresión logística se asume:

* Variable de respuesta binaria.
* Observaciones independientes.
* Ausencia de valores atípicos extremos.
* Ausencia de multicolinealidad.



P es la probabilidad (salida de si o no (1 o 0 ) )

Y es la variable dependiente (resultado esperodo condicionado por las caracteristicas)

X es la variable independiente (caracteristacas que influyen en el resultado)

W es el parametro del modelo (modelo de regresion)

La combinacion de X y W conducen a una combinacion lineal, dando

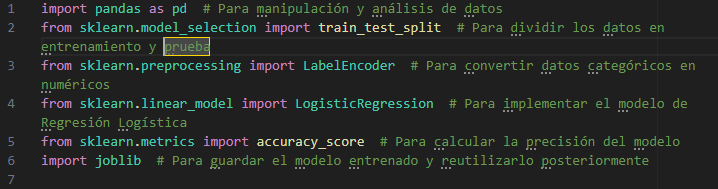


La Función sigmoidea nos permite convertir el valor de entrada en una salida entre 0 y 1.

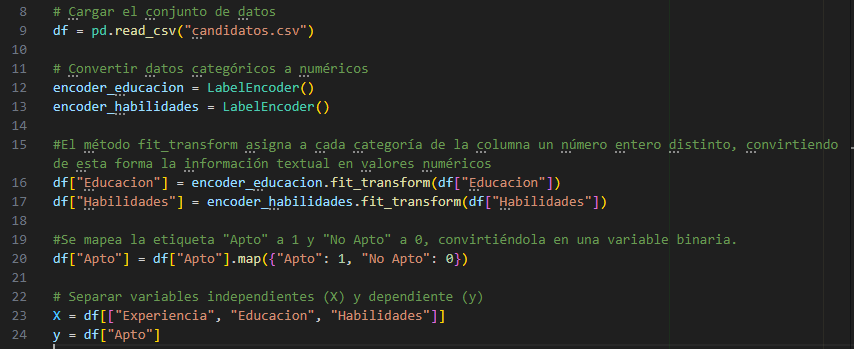
Lo que nos va a permitir reestructurar la ecuación. Como estamos en el supuesto de que solo hay dos respuestas y estas son excluyentes entre sí, se puede avanzar con la utilización de la Función de Bernoulli. Si consideramos todas las observaciones independientes se utilizara la Función de verosimilitud (Likelihood Funtion), se puede profundizar en el funcionamiento de la Función de verosimilitud, pero explicado hasta ente punto nos permite ejemplificar como la Regresión Logística es aplicada al en el desarrollo de nuestro problema.

**Aplicación de Regresión Logística sobre nuestro problema**

**Librerías:**

****

**Preparación de datos:**

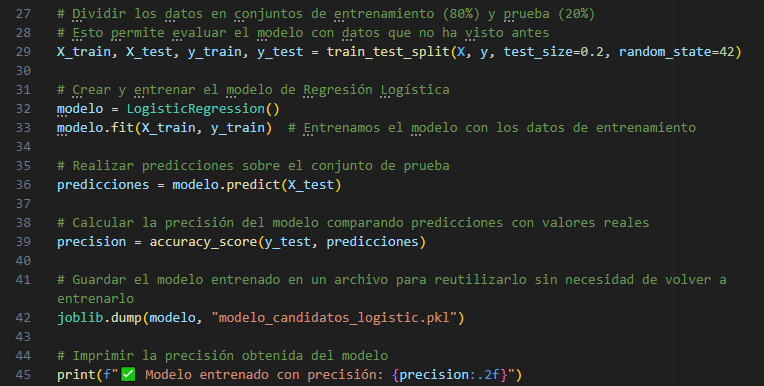
****

**Observemos:**

Los datos que son pertenecientes a las variables independientes X (“Educacion” y “Habilidades”) estan siendo convertidas a valores numéricos.

Los datos perteneciente a la categoría de apto y no apto están respetando lo mencionado anteriormente, quedando los valores de forma binaria.

Los datos pertenecientes al parámetro del modelo W son las variables dependientes almacenados en la variable de nombre “y”.

**Creación del Modelo, Entrenamiento y Predicción:   
**

**Observemos:**

Notemos que nuestras predicciones P están siendo almacenadas en la variable de nombre “predicciones”.

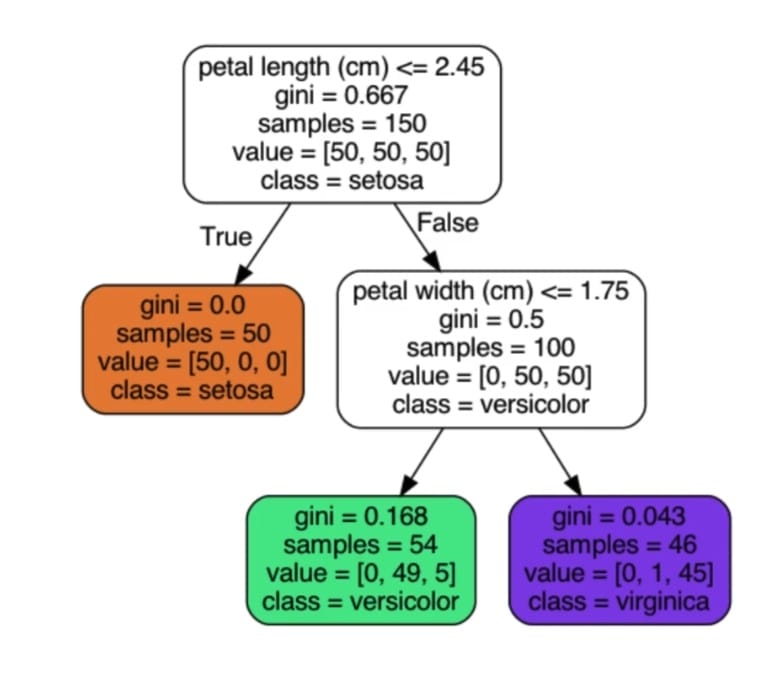
El resultado de la accuracy\_score fue:

Modelo entrenado con precisión: 0.80

**Árbol de Decisiones**

Un árbol de decisión comienza con un nodo raíz, que no tiene ninguna rama entrante. Las ramas salientes del nodo raíz luego alimentan los nodos internos, también conocidos como nodos de decisión. En función de las características disponibles, ambos tipos de nodos realizan evaluaciones para formar subconjuntos homogéneos, que se denotan mediante nodos hoja o nodos terminales. Los nodos hoja representan todos los resultados posibles dentro del conjunto de datos.

**¿Cómo funciona AD?**

El nodo raíz tiene una característica que funciona como condición, teniendo dos posibles salidas. Si se cumple la condición avanza por True, en su defecto, avanza por False.

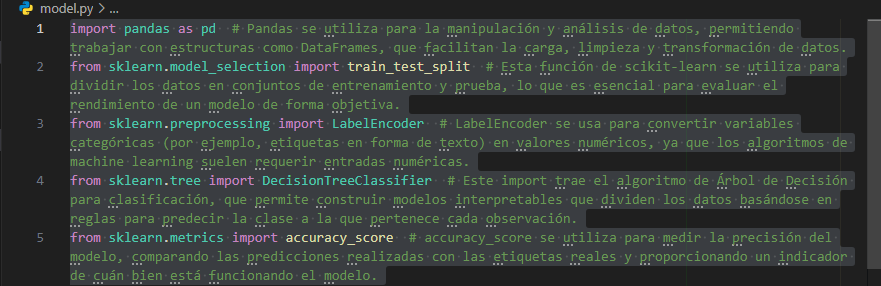
Cada nodo tiene un Gini, La impureza de Gini es la probabilidad de clasificar incorrectamente. Si la impureza es 0 todas las muestras pertenecen a esa clase.

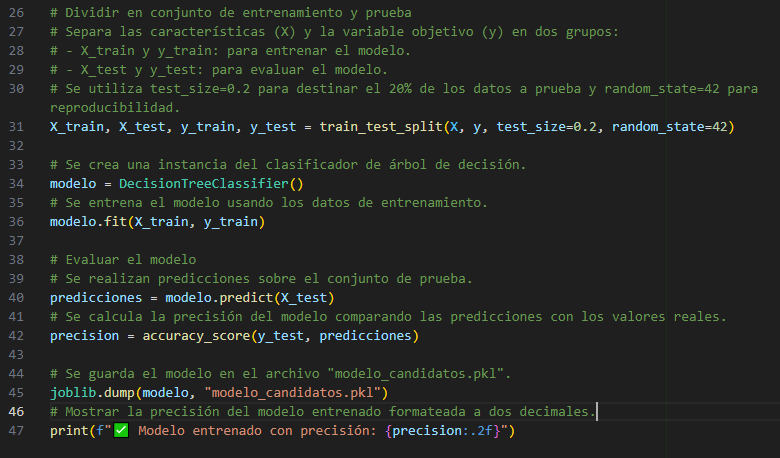
Sample es la cantidad de datos con la que el modelo fue entrenado.

En correlación con el Sample, Value es el conjunto de datos W, siendo cada número perteneciente a una clase en particular.

Class es la clase a la que pertenece a medida que avanza en el árbol.

**Aplicación de Arboles de Deciones sobre Nuestro problema.**

****

****

Observaciones: notemos que la aplicación es similar, a la de Regresion Logistica.  
Los cambios aplicados son:

from sklearn.tree import DecisionTreeClassifier

modelo = DecisionTreeClassifier()

joblib.dump(modelo, "modelo\_candidatos\_logistic.pkl")

**Casos en los que podemos aplicar Regresión Logistica y Árbol de Decisión**

**Sanidad:** Los investigadores médicos planifican la atención y el tratamiento preventivos mediante la predicción de la probabilidad de enfermedad en los pacientes. Utilizan modelos de regresión logística para comparar el impacto de los antecedentes familiares o los genes en las enfermedades.

**Finanzas:** Las empresas financieras tienen que analizar las transacciones financieras en busca de fraudes y evaluar las solicitudes de préstamos y seguros en busca de riesgos. Estos problemas son adecuados para un modelo de regresión logística porque tienen resultados discretos, como alto riesgo o bajo riesgo y fraudulento o no fraudulento.

**Marketing:** Las herramientas de publicidad en línea utilizan el modelo de regresión logística para predecir si los usuarios harán clic en un anuncio. Como resultado, los especialistas en marketing pueden analizar las respuestas de los usuarios a diferentes palabras e imágenes y crear anuncios de alto rendimiento con los que los clientes interactuarán.

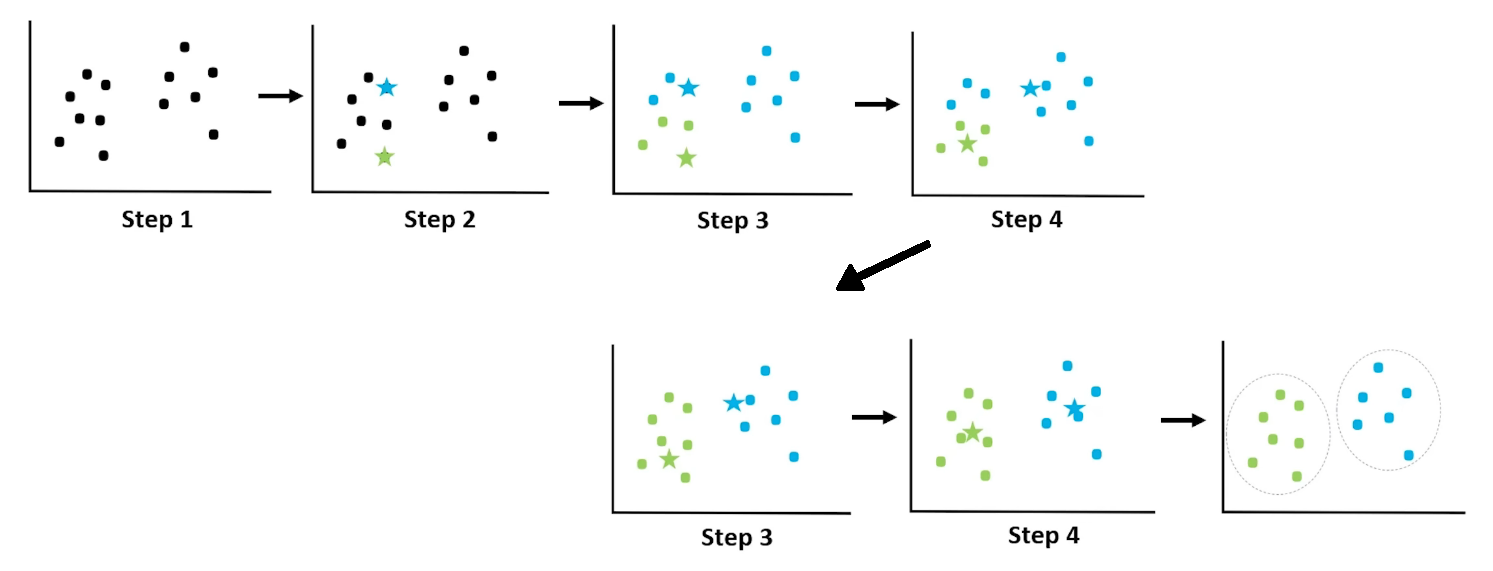
**K-means**

**¿Qué hace?**

Es un algoritmo del tipo de aprendizaje no supervisado, por lo que los datos que utiliza no están etiquetados.

K-means es un algoritmo de agrupamiento iterativo basado en centroides que divide un conjunto de datos en grupos similares en función de la distancia entre sus centroides. El centroide, o centro del clúster, es la media o la mediana de todos los puntos dentro del clúster, según las características de los datos.

**¿Cómo funciona?**



El primer paso es inicializar k centroides donde k es igual al número de clústeres elegidos para un conjunto de datos específico. Este enfoque utiliza métodos de selección aleatoria o muestreo centroide inicial.

El siguiente paso incluye un proceso iterativo de dos etapas basado en el algoritmo de machine learning de maximización de expectativas.El paso de expectativa asigna cada punto de datos a su centroide más cercano en función de la distancia (de nuevo, normalmente euclídea). El paso de maximización calcula la media de todos los puntos de cada conglomerado y reasigna el centro del conglomerado, o centroide. Este proceso se repite hasta que las posiciones de los centroides hayan alcanzado la convergencia o se haya alcanzado el número máximo de iteraciones.

El agrupamiento de medias K es simple pero sensible a las condiciones iniciales y los valores atípicos. Es importante optimizar la inicialización del centroide y el número de clústeres k, para lograr los clústeres más significativos.

Los clústeres de calidad contienen al menos dos propiedades:

1. **Todos los puntos de datos dentro de un clúster deben ser similares.**
2. **Los clústeres deben ser distintos entre sí.**

**¿En qué casos se aplica?**

Se usa comúnmente en la ciencia de datos para la segmentación del mercado, la agrupación de documentos, la segmentación de imágenes y la compresión de imágenes. El algoritmo k-means es un método ampliamente empleado en el análisis de conglomerados porque es eficiente, eficaz y sencillo.

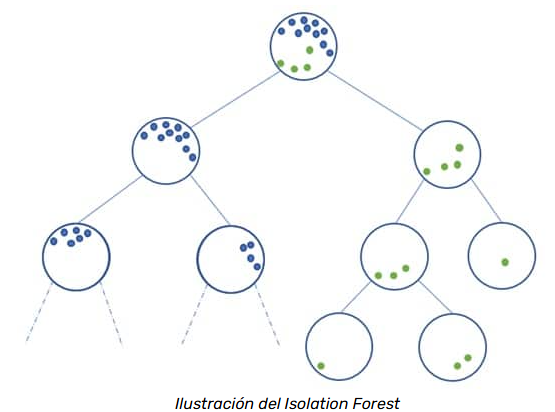
**Isolation Forest**

Isolation Forest es una método no supervisado para identificar anomalías (outliers) cuando los datos no están etiquetados, es decir, no se conoce la clasificación real (anomalía - no anomalía) de las observaciones.

El principio de este algoritmo es:

* Se selecciona una variable (feature) de forma aleatoria.
* A continuación, el conjunto de datos se divide aleatoriamente en función de esta variable para obtener dos subconjuntos de datos.
* Las dos etapas anteriores se repiten hasta aislar un dato.
* Los pasos anteriores se repiten recursivamente.

**¿Cómo se define si un punto es una anomalía o no?** La construcción del algoritmo da lugar a una estructura de árbol, cuyos nodos son los conjuntos particionados y cuyas hojas son los puntos aislados. Intuitivamente, las anomalías tenderán a ser las hojas más cercanas a la raíz del árbol. En función de la distancia a la raíz, se asigna una puntuación de anomalía a cada hoja de un árbol, y esta puntuación de anomalía se promedia sobre todos los árboles que se han construido de forma recursiva, para obtener un resultado final.



Se utiliza para detectar fraudes. Los defraudadores cambian con frecuencia sus estrategias e intentan nuevas formas de cometer fraude. Muchas de estas estrategias serán completamente inesperadas. Por eso métodos no supervisados para detectar anomalías. Funcionan comparando todas las transacciones e identificando las que tienen valores de características inusuales. Lo importante es que esto significa que no tenemos que etiquetar ninguna operación de forma anticipada.

**Fuentes consultadas:**

**https://es.wikipedia.org/wiki/Scikit-learn**

**https://www.datacamp.com/es/tutorial/machine-learning-python**

**https://openwebinars.net/blog/como-entrenar-un-modelo-de-machine-learning-con-scikit-learn/**

**https://www.ibm.com/mx-es/topics/machine-learning**

[**https://repositorio.unal.edu.co/bitstream/handle/unal/2421/42694070\_2009.pdf?seq**](https://repositorio.unal.edu.co/bitstream/handle/unal/2421/42694070_2009.pdf?seq)

[**https://zaguan.unizar.es/record/149521/files/TAZ-TFG-2024-3404.pdf**](https://zaguan.unizar.es/record/149521/files/TAZ-TFG-2024-3404.pdf)

[**https://aws.amazon.com/es/what-is/logistic-regression/#:~:text=La%20regresi%C3%B3n%20log%C3%ADstica%20es%20una,factores%20bas%C3%A1ndose%20en%20el%20otro**](https://aws.amazon.com/es/what-is/logistic-regression/#:~:text=La%20regresi%C3%B3n%20log%C3%ADstica%20es%20una,factores%20bas%C3%A1ndose%20en%20el%20otro)**.**

[**https://www.ibm.com/es-es/topics/logistic-regression**](https://www.ibm.com/es-es/topics/logistic-regression)

[**https://www.ibm.com/es-es/think/topics/decision-trees#:~:text=Un%20%C3%A1rbol%20de%20decisi%C3%B3n%20es,nodos%20internos%20y%20nodos%20hoja**](https://www.ibm.com/es-es/think/topics/decision-trees#:~:text=Un%20%C3%A1rbol%20de%20decisi%C3%B3n%20es,nodos%20internos%20y%20nodos%20hoja)**.**

**https://cienciadedatos.net/documentos/66\_deteccion\_anomalias\_isolationforest**

**https://datascientest.com/es/isolation-forest**

[**https://pyninja.io/**](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbkFsTUtoYmN6YWlHV3VGR25oSzIzV2hmczZ5d3xBQ3Jtc0tuVV9FV1dmVEFYZ2hRT3BFVWFPcHM4V1g5UjVEdlZEV3ZTSElpWlVhMXFqQkNvbjVPM3lwTHNBXzJ5bHJobkdSMEJNaXJwVG1wTnB3TkdSTHJHMENkaFo0bS1lUm1BNEFRMEdrcVNkTGV2MlhJUE1YSQ&q=https%3A%2F%2Fpyninja.io%2F&v=d1tVOcZwM3s)

**https://www.ibm.com/mx-es/topics/k-means-clustering**

**Aplicación Práctica:**

**Desarrollo de un prototipo con *Machine Learning***

**Model.py**

**Preparación de datos:**

Se cargan en la variable “dataSet” el conjunto de datos desde “candidatos.csv”.

Se convierten los datos pertenecientes a las variables independientes “X” (“Experiencia”, “Educación”, “Tecnologías”, “Habilidades”) a valores numéricos gracias a los encoders respectivos.

Los datos pertenecientes a la categoría de apto(1) y no apto(0) están respetando lo mencionado anteriormente, quedando los valores de forma binaria.

En la variable dependiente “Y” se almacenan los datos que queremos predecir, en este caso “apto”.

**Creación del Modelo, Entrenamiento**

Con decisionTreeClasifier() se establece la técnica de machine learning y con accuracy\_score() se evalúa la precisión del modelo.

Con “joblib” se almacena el modelo entrenado en un archivo “.pkl”.

**Predecir.py**

**Preparación de datos:**

- Se importa el modelo entrenado “modelo\_candidatos.pkl”.

- Se guarda en la variable “dataSet” los datos del CSV “candidatos”.

- Se realiza el mapeo de los valores categóricos a números, enumerando a partir del “0” los distintos niveles educativos, las diferentes habilidades y tecnologías requeridas para el puesto. Luego de aplicar los mapeos, se verifica si existen valores nulos en el mapeo y en el caso de existir, da aviso del error en la conversión de los datos.

- En la variable “X” se almacenan las características relevantes a predecir ya convertidas en valores numéricos.

**Predicción:**

Notemos que nuestras predicciones están siendo almacenadas en la variable de nombre “predicciones”. La función predict(X), usa el modelo de árbol de decisión ya entrenado y hace las predicciones sobre nuevos candidatos, devuelve un array de 0s y 1s, donde 1 significa apto y 0 no apto.

Los datos pertenecientes a la categoría de apto(1) y no apto(0) están respetando lo mencionado anteriormente, quedando los valores de forma Binaria. Si la variable pred==1 se lo considera “apto” y se agrega las predicciones al DataFrame. Luego, se actualizan las predicciones en un nuevo CSV.

**Documentación técnica detallada**

**del programa:**

**Clasificación de candidatos**

**basada en habilidades**

*Descripción del modelo.*

Para abordar la clasificación de candidatos, decidimos utilizar **DecisionTreeClassifier() (Árbol de Decisiones)**, un algoritmo de **aprendizaje automático supervisado** dentro de la librería **Scikit-learn**. Este modelo **funciona mediante la segmentación recursiva de los datos**, estructurando decisiones en forma de **árbol**, donde cada nodo representa una condición sobre una característica específica de los candidatos.

El **árbol de decisión** divide progresivamente el conjunto de datos **en función de criterios como experiencia laboral, nivel educativo y habilidades técnicas**, hasta llegar a una clasificación final en una de las categorías definidas (**"Apto" o "No Apto"**).

Instrucciones para ejecutar el modelo de Machine Learning

*Ubicación del archivo de la aplicación*

*Dirígete a la ruta donde está alojada la aplicación “nombre\_final\_de\_la\_app.exe”.*

*Ubica el cursor sobre el archivo nombre\_final\_de\_la\_app.exe para proceder con la ejecución.*

*Distintas formas de ejecución de la aplicación:*

*Opción 1: Mediante el menú contextual*

*Haz clic derecho sobre el archivo nombre\_final\_de\_la\_app.exe.*

*Se desplegará un menú con diferentes opciones.*

*Desliza el cursor sobre "Ejecutar".*

*Presiona clic izquierdo para iniciar la aplicación.*

*Opción 2: Mediante doble clic*

*Posiciona el cursor sobre nombre\_final\_de\_la\_app.exe.*

*Haz doble clic izquierdo sobre el archivo.*

*La aplicación iniciará automáticamente.*

*Opción 3: Mediante selección y tecla "Enter"*

*Ubica el cursor sobre nombre\_final\_de\_la\_app.exe.*

*Haz un clic izquierdo para seleccionar el archivo.*

*Presiona la tecla "Enter" en tu teclado.*

*La aplicación se ejecutará.*

*Opción 4: Desde la terminal (Consola)*

*Abre la terminal (cmd, PowerShell o Terminal de Linux).*

*Ve al directorio donde se encuentra la aplicación con el siguiente comando:*

*cd "ruta/del/archivo"*

*Ejecuta el archivo con el siguiente comando:*

*nombre\_final\_de\_la\_app.exe*

*Este modelo se ejecuta dentro de un servicio Flask, desarrollado para ser utilizado desde un archivo ejecutable (.exe).*

*Una vez iniciada la ejecución del .exe, el sistema prepara el modelo realizando los siguientes pasos:*

*Se obtiene la ruta del archivo .pkl con el metodo get\_path(relative\_path):*

*modelo\_path = get\_path("modelo\_candidatos.pkl")*

*Se utilizael metodo joblib.load() y se le pasa como parametro la ruta del modelo para cargar el modelo y disponiblilizarlo para futuras predicciones:*

*modelo = joblib.load(modelo\_path)*

*Una vez cargado, el modelo estará disponible para realizar predicciones tantas veces como sean solicitadas a través de los servicios ofrecidos por la aplicación Flask.*

*¿Cómo entrenarlo y hacer predicciones?*

Este modelo requiere ser **entrenado con datos estructurados en formato CSV,** elegido por su facilidad de lectura y manipulación. Para ello, proporcionamos un archivo con **120 candidatos ya evaluados**, donde el **80% (96 candidatos) se usa para el entrenamiento** y el **20% restante (24 candidatos) se reserva para la evaluación**, utilizando la métrica accuracy\_score, logrando una **precisión del 96%**.

Una vez entrenado, el modelo se guarda en formato .pkl con la librería joblib, mediante el método dump(), permitiendo su reutilización para clasificar nuevos candidatos.

El **servicio Flask** incorpora la funcionalidad Predecir, que **verifica la correcta carga de un archivo CSV** antes de procesarlo. Luego, convierte las variables categóricas (Educacion, Habilidades, Tecnologías) en valores numéricos (String → int) a través de un **mapeo**, asegurando la compatibilidad con el modelo entrenado.

Posteriormente, el modelo **realiza la predicción de aptitud**, agregando la columna Apto al dataset con valores "Apto" o "No Apto". Los resultados se muestran en la pestaña Resultados como una tabla en Flask, y se almacenan en un archivo CSV llamado resultado\_prediccion.csv.

Dataset ficticio final.

Descripción del data set ficticio:

.

.

.

.

.

Demo funcional (video corto o capturas de pantalla).

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.