

PROYECTO TALENTO TECH 2.0

Diseño curricular

Nombre del Bootcamp	Inteligencia Artificial
Nivel del Bootcamp	Explorador
Experto temático:	UDEA - UCALDAS
Perfil mínimo de ingreso	<ul style="list-style-type: none"> • Contar con conocimientos básicos en programación, ideal manejo de Python. • Habilidades básicas de comprensión de lectura, matemáticas, razonamiento abstracto. • Manejo básico de un sistema operativo (Windows, MacOS, linux).
Perfil mínimo de egreso	<ul style="list-style-type: none"> • Tratar conceptos fundamentales de la inteligencia artificial. • Utilizar las aplicaciones prácticas de la inteligencia artificial. • Aplicar el proceso general para la construcción de modelos de machine learning. • Conocer las técnicas más comunes de aprendizaje inductivo y supervisado. • Identificar un diseño experimental adecuado para la implementación de proyectos de machine learning.
<p align="center">Criterios de aprobación del curso</p> <ul style="list-style-type: none"> • Curso nivelatorio de inglés. • Asistir como mínimo al 75% de las clases del curso. • Completar y presentar satisfactoriamente todos los proyectos y ejercicios prácticos 	

Justificación

Teniendo en cuenta el auge de la IA en esta nueva oleada de avances significativos, la ciencia de datos y el aprendizaje de máquina es esencial en la actualidad debido a la creciente cantidad de datos disponibles en todas las industrias y sectores. En un mundo impulsado por la información, la capacidad de recopilar, analizar y extraer conocimientos significativos de estos datos se ha vuelto crítica para el éxito empresarial y la toma de decisiones informadas.

Este curso proporciona a los estudiantes las habilidades necesarias para navegar por este vasto océano de datos, desde la recopilación y limpieza inicial hasta el análisis y la visualización. Al comprender cómo utilizar herramientas y técnicas de análisis de datos, los estudiantes pueden identificar tendencias, patrones y relaciones ocultas que de otro modo podrían pasar desapercibidas, lo que les permite tomar decisiones estratégicas respaldadas por evidencia y mejorar la eficiencia operativa en sus organizaciones.

Además, la ciencia de datos y el aprendizaje de máquina equipa a los estudiantes con habilidades altamente demandadas en el mercado laboral actual. Con la explosión de datos en prácticamente todos los aspectos de la vida y los negocios, existe una creciente demanda de profesionales capacitados en el manejo y análisis de datos. Al completar un curso de ciencia de datos y aprendizaje de máquina, los estudiantes se posicionan como candidatos altamente competitivos en una variedad de campos, incluyendo tecnología, finanzas, salud, marketing y más.

Este curso también promueve el pensamiento crítico y la resolución de problemas. Al enfrentarse a conjuntos de datos complejos y desordenados, los estudiantes desarrollan habilidades analíticas y de resolución de problemas que son valiosas en cualquier contexto. A través de proyectos prácticos y estudios de casos del mundo real, los estudiantes aprenden a formular preguntas, diseñar experimentos y llegar a conclusiones significativas utilizando métodos científicos rigurosos.

Metodología, Objetivos**Metodología**

- Explicación magistral por parte del profesor utilizando ayudas audiovisuales y servidor de contenidos. (Lecturas).
- Talleres grupales en clase y extraclase para intercambiar conocimientos y conceptos frente al tema.
- Discusiones grupales
- Sesiones de resolución de problemas

- Talleres en clase y extraclase.

Objetivos

Objetivo General:

Desarrollar competencias teóricas y prácticas en el campo de la inteligencia artificial, enfocándose en la comprensión de conceptos fundamentales, la aplicación de técnicas de ciencia de datos y machine learning para el desarrollo de proyectos prácticos basados en escenarios reales, incluyendo la construcción de chatbots y asistentes virtuales primarios.

Objetivos Específicos:

- Comprender los conceptos fundamentales de la inteligencia artificial, ciencia de datos y machine learning y su relevancia en diversas aplicaciones prácticas, adquiriendo una base sólida en la teoría y los principios subyacentes de esta disciplina.
- Aplicar el proceso general de construcción de modelos de machine learning, desde la identificación del problema hasta la evaluación del modelo, utilizando técnicas de aprendizaje inductivo y supervisado.
- Desarrollar y poner en práctica proyectos de machine learning, incluyendo el diseño experimental adecuado, con énfasis en la construcción y optimización de chatbots y asistentes virtuales primarios que demuestren la aplicabilidad de las técnicas aprendidas.

Estructura

Kit de presaberes

El enlace con las preguntas de presaberes se puede encontrar en el siguiente enlace:

<https://docs.google.com/document/d/1909cRoq4kp0xwr2l0JtjUwBd5FcrYbt6/edit?usp=sharing&ouid=101194662912274647727&rtpof=true&sd=true>

Kit del programador

1. Computadora y Sistema Operativo

- **Requisitos del Equipo:**
 - Procesador: Intel i3 o equivalente
 - Memoria RAM: Mínimo 8 GB
 - Almacenamiento: Al menos 100 GB de espacio libre

- Sistema Operativo: Windows 10, macOS, o Linux (Ubuntu 18.04+)

2. Entorno de Desarrollo Integrado (IDE)

- **Jupyter Notebook:** Herramienta interactiva para desarrollar y compartir código en Python.
 - Descargar Jupyter Notebook - <https://jupyter.org/install>
- **PyCharm:** IDE completo para Python que ofrece herramientas avanzadas de desarrollo.
 - Descargar PyCharm - <https://www.jetbrains.com/pycharm/download/>

3. Lenguaje de Programación

- **Python:** Lenguaje principal utilizado en el curso.
 - Descargar Python - <https://www.python.org/downloads/>

4. Bibliotecas y Frameworks de Python

- **NumPy:** Para operaciones matemáticas y manejo de matrices.
 - Instalación: `pip install numpy` - <https://numpy.org/>
- **pandas:** Para la manipulación y análisis de datos.
 - Instalación: `pip install pandas` - <https://pandas.pydata.org/>
- **Matplotlib y Seaborn:** Para la visualización de datos.
 - Instalación: `pip install matplotlib seaborn` - <https://matplotlib.org/> y <https://seaborn.pydata.org/>
- **scikit-learn:** Para algoritmos de machine learning.
 - Instalación: `pip install scikit-learn` - <https://scikit-learn.org/>

5. Gestión de Versiones

- **Git:** Sistema de control de versiones para gestionar y compartir código.
 - Descargar Git - <https://git-scm.com/downloads>
- **GitHub:** Plataforma para alojar repositorios de Git y colaborar con otros programadores.
 - Crear cuenta en GitHub - <https://github.com/>

6. Entornos Virtuales

- **virtualenv:** Herramienta para crear entornos virtuales en Python.
 - Instalación: `pip install virtualenv` - <https://pypi.org/project/virtualenv/>

- **Anaconda:** Distribución de Python que incluye muchas de las bibliotecas necesarias y herramientas para la ciencia de datos.
 - Descargar Anaconda - <https://www.anaconda.com/products/distribution>
- **Microsoft Azure:** Plataforma en la nube con servicios de IA como Azure Machine Learning.
 - Crear cuenta en Azure - <https://azure.microsoft.com/>

7. Recursos Educativos y Documentación

- **Kaggle:** Plataforma de competencias de data science con datasets y kernels.
 - Registrarse en Kaggle - <https://www.kaggle.com/>
- **Documentación Oficial:** Acceso a la documentación oficial de las herramientas y bibliotecas utilizadas.
 - Python: Documentación de Python - <https://docs.python.org/3/>
 - NumPy: Documentación de NumPy - <https://numpy.org/doc/>
 - Pandas: Documentación de pandas - <https://pandas.pydata.org/pandas-docs/stable/>
 - Matplotlib: Documentación de Matplotlib - <https://matplotlib.org/stable/contents.html>
 - Scikit-learn: Documentación de scikit-learn - <https://scikit-learn.org/stable/documentation.html>

Cronograma: 159 horas (105 Técnicas, 15 habilidades de poder, 24 inglés técnico, 15 proyectos)

Proyectos a desarrollar:

El proyecto a desarrollar en el bootcamp se enmarca en una de las las siguientes líneas:

- **Transición energética justa**, democratización de la generación y consumo energético, desarrollo de comunidades energéticas, impulso a energías limpias (hidrógeno verde, eólica, solar, entre otras) y minerales estratégicos.
- **Ciencia, tecnología e innovación** para la transformación productiva y la resolución de desafíos sociales, económicos y ambientales, contribuyendo a la construcción de una sociedad del conocimiento.

En esta serie de proyectos prácticos, exploraremos las diferentes etapas del ciclo de vida de proyectos de machine learning (ML), centrándonos en la identificación de problemas, el análisis de datos, la implementación de modelos y su despliegue. Los tres proyectos están interrelacionados y proporcionan una experiencia completa, desde la detección del problema hasta la implementación, evaluación y despliegue de modelos de ML. A continuación, se

detalla cada proyecto y cómo se relacionan entre sí.

Proyecto 1: Fases Iniciales del Ciclo de Vida de un Proyecto de ML

Objetivo:

Ilustrar las primeras fases del ciclo de vida de un proyecto de ML, incluyendo la detección del problema, la identificación de datos y stakeholders, y el análisis exploratorio de datos.

Descripción:

- Contexto y Problemas: Los estudiantes trabajarán con datos relacionados con la transición energética justa, democratización del consumo energético, y la transformación productiva mediante ciencia y tecnología.
- Herramientas: Jupyter Notebooks, Python, pandas, numpy.
- Etapas:
 - Introducción teórica sobre el ciclo de vida de proyectos de ML.
 - Selección del contexto y problema específico.
 - Análisis exploratorio de datos, incluyendo carga de datos, evaluación de calidad, tratamiento de datos ausentes, normalización y análisis univariado, bivariado y multivariado.

Proyecto 2: Aplicación de Modelos de Machine Learning

Objetivo:

Aplicar un proceso completo de aprendizaje de máquina, desde la preparación de datos hasta la comparación y ajuste de modelos.

Descripción:

- Proceso:
 - División del conjunto de datos en entrenamiento y prueba.
 - Preprocesamiento de datos para estandarización o normalización.
 - Entrenamiento y comparación de varios modelos (regresión logística, árboles de decisión, máquinas de soporte vectorial).
 - Evaluación del rendimiento de los modelos utilizando métricas adecuadas.
 - Ajuste de hiperparámetros para mejorar el rendimiento.
 - Interpretación de resultados y reflexión sobre mejoras futuras.

Proyecto 3: Implementación y Despliegue de un Clasificador Naive Bayes

Objetivo:

Implementar y desplegar un clasificador Naive Bayes básico para clasificar frases en categorías específicas.

Descripción:

- Proceso:
 - Definición y etiquetado de un conjunto de datos con frases.
 - Creación de un diccionario de palabras y cálculo de frecuencias.
 - Implementación del clasificador Naive Bayes con suavización de Laplace.
 - Pruebas del clasificador con nuevas frases para evaluar precisión y efectividad.
 - Despliegue del modelo: Implementación del clasificador en un entorno de producción utilizando herramientas como Flask o FastAPI para crear una API que permita el uso del modelo por aplicaciones externas.

Misión 1

**Entrenamiento
Misión 1**

Mapa de ruta:

- **Objetivo**

El objetivo general de esta misión es enseñar a los estudiantes los conceptos y habilidades esenciales para desarrollar y gestionar proyectos de aprendizaje de máquina, enfocándose en la identificación del problema, la recolección y preparación de datos, y el análisis exploratorio. Los estudiantes aprenderán a definir problemas de aprendizaje de máquina, recolectar datos de diversas fuentes, aplicar técnicas de visualización y análisis de datos, y preparar conjuntos de datos para la creación y evaluación de modelos.

- **Habilidades digitales**

- Conocimientos Básicos de Programación
- Conceptos Básicos de Estadística y Matemáticas
- Habilidades en el Manejo de Datos

- **Competencias**

- Capacidad para analizar problemas y encontrar soluciones efectivas.
- Habilidad para explicar ideas y resultados de manera clara a diferentes audiencias.
- Capacidad para colaborar con otros y compartir conocimientos para alcanzar objetivos comunes.

- **Contenidos (Temáticas)**

Nombre de la Misión 1: Fundamentos de Aprendizaje de máquina y Preparación de Datos

Introducción

- Tipos de aprendizaje en la IA
 - Supervisado
 - No supervisado
 - Reforzado

	<p>Historia Ciclo de vida de una aplicación de Aprendizaje de máquina</p> <ul style="list-style-type: none"> • Identificación del Problema • Recolección de Datos • Preparación de Datos • Ingeniería de modelos • Evaluación del Modelo • Despliegue • Mantenimiento y Actualización <ul style="list-style-type: none"> · Identificación del Problema <ul style="list-style-type: none"> • Definición del objetivo • Análisis de viabilidad · Recolección de Datos <ul style="list-style-type: none"> • Fuentes de datos <ul style="list-style-type: none"> ○ Ejemplos de conjuntos de datos disponibles ○ Tipos de bases de datos ○ Carga de datos ○ Generación de datos sintéticos • Calidad y cantidad de datos · Visualización de datos <ul style="list-style-type: none"> • Tipos de gráficos <ul style="list-style-type: none"> ○ Histogramas ○ Cajas ○ Densidad · Análisis exploratorio de datos <ul style="list-style-type: none"> • Dimensiones de la Calidad de Datos:
--	---

	<ul style="list-style-type: none"> • Completitud, consistencia, exactitud • Detección y tratamiento de datos ausentes • Descarte • Imputación <ul style="list-style-type: none"> ○ Media o mediana ○ Regresión ○ Hot deck • Normalización de datos <ul style="list-style-type: none"> ○ Minmax ○ Robust ○ Análisis univariable ○ Validación de distribución normal ○ Estadísticas descriptivas • Análisis bivariado <ul style="list-style-type: none"> ○ Correlación • Análisis multivariado <ul style="list-style-type: none"> ○ PCA • Ejecución del proyecto <p>El proyecto práctico tiene como objetivo ilustrar las primeras fases del ciclo de vida de un proyecto de machine learning (ML), centrándose en la detección del problema, la identificación de datos potenciales y la identificación de los interesados, así como en el posible impacto de la solución. Los estudiantes trabajarán con un conjunto de datos públicos en el marco de las siguientes líneas:</p> <ul style="list-style-type: none"> ○ Transición energética justa, democratización de la generación y consumo energético, desarrollo de comunidades energéticas, impulso a energías limpias (hidrógeno verde, eólica, solar, entre otras) y minerales estratégicos. ○ Ciencia, tecnología e innovación para la transformación productiva y la resolución de desafíos sociales, económicos y ambientales, contribuyendo a la construcción de una
--	---

	<p>sociedad del conocimiento.</p> <p>Para esto deben utilizar herramientas de análisis de datos como Jupyter Notebooks, Python, pandas y numpy.</p> <p>El desarrollo del proyecto se dividirá en varias etapas. Primero, en la introducción teórica, los estudiantes revisarán las diferentes fases del ciclo de vida de un proyecto de ML, enfocándose en la identificación del problema y la recolección de datos. Discutirán las técnicas y metodologías para identificar problemas, seleccionar variables relevantes y la importancia de identificar a los stakeholders.</p> <p>En la selección del contexto y problema, los estudiantes revisarán ejemplos de contextos regionales, como la transición energética justa y la democratización del consumo energético, o la ciencia, tecnología e innovación para la transformación productiva. Luego, elegirán un contexto basado en el conjunto de datos. Discutirán la relevancia del problema y los interesados involucrados, como agricultores, investigadores y agencias gubernamentales.</p> <p>Finalmente, en el análisis de datos, los estudiantes presentarán un conjunto de datos relacionado con el problema seleccionado. Demostrarán cómo explorar y comprender el conjunto de datos utilizando herramientas de análisis de datos. Primero, mostrarán cómo cargar el conjunto de datos en un DataFrame usando pandas y revisarán la cantidad de columnas y filas para obtener una visión general del tamaño del conjunto de datos. Identificarán los tipos de datos de cada columna, como numéricos, categóricos o booleanos, y discutirán su importancia en el análisis y modelado. Evaluarán la calidad de los datos en términos de completitud, consistencia y exactitud, y abordarán la detección y tratamiento de datos ausentes mediante técnicas como el descarte o la imputación (media, mediana, regresión, Hot deck). Aplicarán técnicas de normalización de datos, incluyendo Min-Max Scaling y escalado robusto, para mejorar el rendimiento del modelo. Realizarán un análisis univariable validando la distribución normal de las variables y calculando estadísticas descriptivas como media, mediana, moda y desviación estándar. En el análisis bivariado, calcularán y visualizarán la correlación entre variables para explorar sus relaciones. Finalmente, llevarán a cabo un análisis multivariado mediante un Análisis de Componentes Principales (PCA) para reducir la dimensionalidad y crear visualizaciones complejas que permitan</p>
--	---

	<p>entender mejor la interacción entre múltiples variables. Concluirán el proyecto con una reflexión sobre el proceso seguido, los desafíos encontrados y las lecciones aprendidas.</p>
--	---

	<p>Preparación</p> <p>Tipos de aprendizaje en la IA</p> <ul style="list-style-type: none"> ● Supervisado: Los modelos son entrenados con un conjunto de datos etiquetados. Ejemplos incluyen clasificación y regresión. ● No supervisado: Los modelos encuentran patrones en datos no etiquetados. Ejemplos incluyen clustering y reducción de dimensionalidad. ● Reforzado: Los modelos aprenden a tomar decisiones secuenciales para maximizar una recompensa. Ejemplos incluyen juegos y control robótico. <p>Historia y Ciclo de Vida de una Aplicación de Aprendizaje de Máquina</p> <p>Identificación del Problema</p> <ul style="list-style-type: none"> ● Definición del objetivo: Determinar claramente qué se quiere lograr con el modelo. ● Análisis de viabilidad: Evaluar si el problema es adecuado para ser abordado con ML y si se dispone de datos suficientes y de calidad. <p>Recolección de Datos</p> <ul style="list-style-type: none"> ● Fuentes de datos: <ul style="list-style-type: none"> ○ Ejemplos de conjuntos de datos disponibles: UCI Machine Learning Repository, Kaggle Datasets. ○ Tipos de bases de datos: Relacionales (SQL), NoSQL, APIs. ○ Carga de datos: Procesos para importar datos desde diferentes fuentes. ○ Generación de datos sintéticos: Crear datos artificiales para aumentar el tamaño del conjunto de datos. ● Calidad y cantidad de datos: Evaluar si los datos son suficientes y de buena calidad para el modelado. <p>Preparación de Datos</p>
--	---

	<ul style="list-style-type: none"> ● Visualización de datos: <ul style="list-style-type: none"> ○ Tipos de gráficos: Histogramas, diagramas de caja, gráficos de densidad. ● Análisis exploratorio de datos (EDA): <ul style="list-style-type: none"> ○ Dimensiones de la calidad de datos: Completitud, consistencia, exactitud. ○ Detección y tratamiento de datos ausentes: <ul style="list-style-type: none"> ■ Descarte: Eliminar registros con datos faltantes. ■ Imputación: Rellenar datos faltantes con métodos como media, mediana, regresión, hot deck. ○ Normalización de datos: <ul style="list-style-type: none"> ■ MinMax: Escalar datos entre un rango específico. ■ Robust: Escalar datos usando la mediana y el rango intercuartílico. ○ Análisis univariado: Exploración de una sola variable a la vez. <ul style="list-style-type: none"> ■ Validación de distribución normal. ■ Estadísticas descriptivas. ○ Análisis bivariado: Exploración de la relación entre dos variables. <ul style="list-style-type: none"> ■ Correlación: Medir la relación entre variables. ○ Análisis multivariado: Exploración de la relación entre múltiples variables. <ul style="list-style-type: none"> ■ PCA (Análisis de Componentes Principales): Técnica para reducir la dimensionalidad de los datos. <p>Ingeniería de Modelos</p> <ul style="list-style-type: none"> ● Selección y entrenamiento de modelos adecuados basados en los datos y el problema. <p>Evaluación del Modelo</p> <ul style="list-style-type: none"> ● Validación del rendimiento del modelo usando métricas
--	--

	<p>apropiadas.</p> <p>Despliegue</p> <ul style="list-style-type: none"> Implementación del modelo en un entorno de producción para que se pueda usar en la práctica. <p>Mantenimiento y Actualización</p> <ul style="list-style-type: none"> Monitoreo del rendimiento del modelo y actualización periódica con nuevos datos para mantener su efectividad.
Experiencia a Misión 1	<p>Simulación</p> <p>Objetivo</p> <p>El objetivo principal de esta práctica es que los estudiantes adquieran las habilidades fundamentales para iniciar un proyecto de aprendizaje automático. A través de un caso práctico, los participantes aprenderán a: Identificar y definir problemas relevantes en el ámbito del aprendizaje automático, considerando contextos reales como la transición energética y la innovación tecnológica.</p> <p>Recolectar y explorar datos utilizando herramientas como pandas, identificando características clave y realizando una limpieza inicial de los datos.</p> <p>Visualizar datos de manera efectiva para comprender su distribución y relaciones.</p> <p>Realizar un análisis exploratorio de datos (EDA) completo, incluyendo la evaluación de la calidad de los datos, el manejo de datos faltantes y la exploración de variables univariadas y multivariadas.</p> <p>Materiales Necesarios</p> <p>Conjunto de datos públicos sobre agricultura (por ejemplo, datos sobre producción de cultivos).</p> <p>Herramientas de análisis de datos (Jupyter Notebooks, Python, pandas, numpy).</p> <p>Desarrollo</p>

	<p>1. Introducción Teórica El profesor repasará a los estudiantes las diferentes fases del ciclo de vida de un proyecto de aprendizaje de máquina, enfocándose en las primeras fases correspondientes a la identificación del problema y recolección de datos. Se discutirán las técnicas y metodologías utilizadas para identificar problemas en machine learning, la identificación y selección de variables relevantes, y la importancia de identificar a los stakeholders (interesados) y su rol en el proyecto.</p> <p>2. Selección del Contexto y Problema El profesor presentará varios ejemplos de contextos regionales en el marco de las siguientes líneas: Transición energética justa, democratización de la generación y el consumo energético, desarrollo de comunidades energéticas, impulso a las energías limpias (hidrógeno verde, eólica, solar, entre otras) y minerales estratégicos. Ciencia, tecnología e innovación para la transformación productiva y la resolución de desafíos sociales, económicos y ambientales del país; y la construcción de una sociedad del conocimiento. Elegirá un contexto basado en el conjunto de datos disponible, por ejemplo, "Predecir el rendimiento de los cultivos". Guiará una discusión sobre por qué este problema es relevante y quiénes son los interesados, como agricultores, investigadores y agencias gubernamentales.</p> <p>3. Análisis de Datos</p> <p>3.1. Carga y Exploración del Conjunto de Datos El profesor mostrará cómo cargar el conjunto de datos en un DataFrame utilizando pandas y revisará la cantidad de columnas y filas para proporcionar una idea general del tamaño del conjunto de datos.</p> <p>3.2. Tipos de Datos Identificará los tipos de datos de cada columna (numéricos, categóricos, booleanos, etc.) y discutirá la importancia de los tipos de datos en el análisis y modelado.</p> <p>3.3. Cantidad de Datos Faltantes Detectará la cantidad de datos faltantes en cada columna y discutirá sobre el impacto que estos datos faltantes tienen en el desarrollo del modelo de machine learning.</p>
--	---

	<p>3.4. Verificación de Reglas de Rango y Dominio</p> <p>Identificará y marcará los datos basura, como valores erróneos o irrelevantes, revisará valores fuera de un rango esperado o texto en columnas numéricas, y discutirá estrategias para limpiar estos datos basura.</p> <p>4. Visualización de Datos</p> <p>4.1. Tipos de Gráficos</p> <p>El profesor enseñará a los estudiantes cómo crear diferentes tipos de gráficos para visualizar la distribución y características de los datos:</p> <p>Histogramas para visualizar la distribución de variables numéricas.</p> <p>Gráficos de caja (Box plots) para visualizar la dispersión y posibles outliers.</p> <p>Gráficos de densidad para visualizar la distribución de variables continuas.</p> <p>5. Análisis Exploratorio de Datos</p> <p>5.1. Dimensiones de la Calidad de Datos</p> <p>El profesor evaluará junto a los estudiantes la completitud, consistencia y exactitud de los datos.</p> <p>5.2. Detección y Tratamiento de Datos Ausentes</p> <p>El profesor enseñará a los estudiantes cómo manejar los datos faltantes mediante el descarte de estos datos o utilizando técnicas de imputación como media, mediana, regresión o Hot deck.</p> <p>5.3. Normalización de Datos</p> <p>El profesor mostrará cómo aplicar técnicas de Min-Max Scaling y escalado robusto para normalizar los datos y mejorar el rendimiento del modelo.</p> <p>5.4. Análisis Univariante</p> <p>El profesor guiará a los estudiantes en la validación de la distribución normal de las variables y el cálculo de estadísticas descriptivas como media, mediana, moda, desviación estándar, entre otras.</p> <p>5.5. Análisis Bivariado</p> <p>El profesor enseñará cómo calcular y visualizar la correlación entre variables para explorar la relación entre dos variables.</p> <p>5.6. Análisis Multivariado</p>
--	---

	<p>El profesor guiará a los estudiantes en la realización de un Análisis de Componentes Principales (PCA) para reducir la dimensionalidad y visualizar la interacción entre múltiples variables, y en la creación de visualizaciones complejas para entender mejor la interacción entre múltiples variables.</p>
<p>Conexión Misión 1</p>	<p>Co-creación</p> <p>En esta práctica, los estudiantes deben crear un video en el que presenten un proyecto de machine learning desde su inicio hasta el análisis de datos. Primero, deben ofrecer una breve introducción teórica sobre las fases del ciclo de vida de un proyecto de machine learning, discutiendo técnicas para identificar problemas y la importancia de los stakeholders. Luego, deben seleccionar un contexto relevante que esté en el marco de las siguientes 2 líneas:</p> <ul style="list-style-type: none"> ● Transición energética justa, democratización de la generación y consumo energético, desarrollo de comunidades energéticas, impulso a energías limpias (hidrógeno verde, eólica, solar, entre otras) y minerales estratégicos. ● Ciencia, tecnología e innovación para la transformación productiva y la resolución de desafíos sociales, económicos y ambientales, contribuyendo a la construcción de una sociedad del conocimiento. <p>A continuación, mostrarán cómo realizar un análisis de datos, incluyendo la carga y exploración del conjunto de datos, identificación de tipos de datos, manejo de datos faltantes, y normalización. Finalmente, deben reflexionar sobre el proceso seguido, los desafíos encontrados y las lecciones aprendidas.</p> <p>Nota: Se recomienda que el dataset sea acorde para un problema de clasificación o regresión con datos multivariados. (Esto es importante definirlo con miras a la práctica final)</p> <p>Composición del video</p> <p>1. Introducción Teórica (Breve)</p> <ul style="list-style-type: none"> ● Repasar las diferentes fases del ciclo de vida de un proyecto de machine learning.

- Discutir las técnicas y metodologías utilizadas para identificar problemas en machine learning.
- Destacar la importancia de identificar a los stakeholders (interesados) y su rol en el proyecto.

2. Selección del Contexto y Problema

- Revisar varios ejemplos de contextos regionales tales como
 - Transición energética justa, democratización de la generación y el consumo energético, desarrollo de comunidades energéticas, impulso a las energías limpias, (hidrógeno verde, eólica, solar, entre otras) y minerales estratégicos.
 - Ciencia, tecnología e innovación para la transformación productiva y la resolución de desafíos sociales, económicos y ambientales del país; y la construcción de una sociedad del conocimiento.
- Discutir por qué este problema es relevante.
- Identificar a los interesados

3. Análisis de Datos

Carga y Exploración del Conjunto de Datos:

- Mostrar cómo cargar el conjunto de datos en un DataFrame utilizando pandas.
- Revisar la cantidad de columnas y filas para tener una idea general del tamaño del conjunto de datos.

Tipos de Datos:

- Identificar los tipos de datos de cada columna (numéricos, categóricos, booleanos, etc.).
- Discutir la importancia de los tipos de datos en el análisis y modelado.

Cantidad de Datos Faltantes:

- Detectar la cantidad de datos faltantes en cada columna.
- Discutir sobre el impacto que tiene los datos faltantes en el problema en cuestión en fases más avanzadas del desarrollo del modelo de machine learning

Verificación de Reglas de Rango y Dominio:

- Identificar y marcar los datos basura, como valores erróneos o irrelevantes.

	<ul style="list-style-type: none"> ○ Revisar valores fuera de un rango esperado o texto en columnas numéricas. ○ Discutir estrategias para limpiar los datos basura. <p>Dimensiones de la Calidad de Datos:</p> <ul style="list-style-type: none"> ● Evaluar la completitud, consistencia y exactitud de los datos. <p>Detección y Tratamiento de Datos Ausentes:</p> <ul style="list-style-type: none"> ● Mostrar cómo manejar los datos faltantes mediante el descarte o técnicas de imputación (media, mediana, regresión, Hot deck). <p>Normalización de Datos:</p> <ul style="list-style-type: none"> ● Aplicar técnicas de Min-Max Scaling y escalado robusto para normalizar los datos y mejorar el rendimiento del modelo. <p>Análisis Univariable:</p> <ul style="list-style-type: none"> ● Validar la distribución normal de las variables. ● Calcular estadísticas descriptivas como media, mediana, moda y desviación estándar. <p>Análisis Bivariado:</p> <ul style="list-style-type: none"> ● Calcular y visualizar la correlación entre variables para explorar la relación entre dos variables. <p>Análisis Multivariado:</p> <ul style="list-style-type: none"> ● Realizar un Análisis de Componentes Principales (PCA) para reducir la dimensionalidad. ● Crear visualizaciones complejas para entender mejor la interacción entre múltiples variables. <p>4. Reflexión y Conclusión</p> <ul style="list-style-type: none"> ● Reflexionar sobre el proceso seguido y los desafíos encontrados.
--	--

	<p>Momento para conectar con el sector laboral, interactuar con profesionales, participar en conferencias, seminarios web, ferias de empleo y eventos de networking. Adicional, presentar sus proyectos digitales o ideas emprendedoras para atraer inversores, colaboradores o clientes interesados en sus proyectos digitales.</p>
MISIÓN 2	
<p>Entrenamiento Misión 2</p>	<p>Mapa de ruta:</p> <ul style="list-style-type: none"> ● Objetivo <p>El objetivo general de esta misión es entender la manera de desarrollar, ajustar y evaluar modelos de aprendizaje de máquina. El curso se centrará en técnicas de aprendizaje supervisado como regresión lineal, regresión logística, KNN, Naive Bayes, árboles de decisión y máquinas de soporte vectorial, y en técnicas de aprendizaje no supervisado como K-Means. Los estudiantes aprenderán a manejar problemas de sobreajuste y subajuste, aplicar técnicas de validación y ajuste de hiperparámetros, y utilizar métricas clave para evaluar la precisión y efectividad de los modelos.</p> ● Habilidades digitales <ul style="list-style-type: none"> ○ Conocimientos básicos en la carga y manipulación de datos usando python ○ Conocimientos básicos en la carga y manipulación de datos ● Competencias <ul style="list-style-type: none"> ● Capacidad para evaluar la calidad de los datos, identificar problemas y aplicar soluciones adecuadas en el proceso de análisis. ● Habilidad para abordar y resolver desafíos relacionados con la visualización de datos, la normalización y el manejo de datos ausentes. ● Competencia para presentar datos y resultados de manera clara y efectiva utilizando visualizaciones adecuadas y resúmenes estadísticos. ● Capacidad para ajustar técnicas y herramientas según las

	<p>necesidades del proyecto y las características específicas del conjunto de datos.</p> <ul style="list-style-type: none"> Habilidad para colaborar con otros en la recopilación, análisis y presentación de datos, integrando diversos puntos de vista y aportaciones. <ul style="list-style-type: none"> Contenidos <p>Nombre de la misión 2: Modelos de Aprendizaje Supervisado y no Supervisado</p> <p>Aprendizaje supervisado</p> <ul style="list-style-type: none"> Regresión lineal Regresión logística KNN Naive Bayes Árboles de decisión Máquinas de soporte vectorial <p>Aprendizaje no supervisado</p> <ul style="list-style-type: none"> K-Means <p>Afinamiento de modelos</p> <ul style="list-style-type: none"> Sobre-ajuste y sub-ajuste Técnicas para tratamiento <ul style="list-style-type: none"> Hold Out Validación cruzada Ajuste de Hiperparámetros <p>Validación y Métricas</p> <ul style="list-style-type: none"> Métricas de modelos de regresión <ul style="list-style-type: none"> Error Cuadrático Medio (Mean Squared Error, MSE) Raíz del Error Cuadrático Medio (Root Mean Squared Error, RMSE) Error Absoluto Medio (Mean Absolute Error, MAE) R-cuadrado (Coefficient of Determination, R^2) R-cuadrado Ajustado (Adjusted R^2) Métricas de modelos de clasificación
--	---

	<ul style="list-style-type: none"> ○ Matriz de confusión ○ F1 Score ○ Precisión ○ Recall <ul style="list-style-type: none"> ● Ejecución del proyecto <p>El proyecto consiste en aplicar un proceso completo de aprendizaje de máquina que incluye la división del conjunto de datos en entrenamiento y prueba, seguido del preprocesamiento para estandarizar o normalizar las características. Luego, se entrenarán y compararán varios modelos de machine learning, como regresión logística, árboles de decisión y máquinas de soporte vectorial, utilizando métricas de rendimiento para evaluar su eficacia. Posteriormente, se ajustarán los hiperparámetros de los modelos seleccionados para mejorar su rendimiento. Finalmente, se interpretarán los resultados obtenidos, reflexionando sobre el impacto de cada modelo y posibles mejoras futuras.</p>
	<p>Preparación</p> <p>Regresión Lineal</p> <p>La regresión lineal es una técnica utilizada para predecir un valor continuo basado en una o más variables independientes. Se busca encontrar una relación lineal que minimice la diferencia entre los valores predichos y los valores reales. Es útil para tareas como predecir el precio de una casa en función de sus características.</p> <p>Regresión Logística</p> <p>La regresión logística se utiliza para predecir la probabilidad de un evento binario (por ejemplo, sí/no, verdadero/falso). Es útil para clasificaciones donde el resultado es una de dos categorías, como diagnosticar si un paciente tiene una enfermedad o no.</p> <p>Naive Bayes</p>

Naive Bayes es un clasificador que aplica el teorema de Bayes bajo la suposición de que las características son independientes entre sí. Es rápido y eficiente para grandes conjuntos de datos y se usa comúnmente en la clasificación de texto, como filtrar correos electrónicos spam.

Árboles de Decisión

Los árboles de decisión son modelos que dividen los datos en subconjuntos basados en decisiones que maximizan la separación de las clases objetivo. Son intuitivos y fáciles de interpretar, y se utilizan en aplicaciones como decisiones de crédito y diagnóstico médico.

Máquinas de Soporte Vectorial (SVM)

Las SVM son algoritmos de clasificación que buscan el límite óptimo que separa las diferentes clases con el mayor margen posible. Son especialmente útiles en problemas de alta dimensionalidad como el reconocimiento de imágenes y el análisis de texto.

Aprendizaje No Supervisado

K-Means

K-Means es un algoritmo de agrupación que divide los datos en K grupos basándose en la similitud de las características. Se utiliza para descubrir patrones ocultos en datos sin etiquetas, como la segmentación de clientes en marketing.

Afinamiento de Modelos

Sobreajuste y Subajuste

- **Sobreajuste:** Ocurre cuando un modelo se ajusta demasiado bien a los datos de entrenamiento, capturando ruido y patrones irrelevantes, lo que resulta en un rendimiento deficiente en nuevos datos.

- **Subajuste:** Ocurre cuando un modelo es demasiado simple para capturar los patrones en los datos de entrenamiento, resultando en un rendimiento pobre tanto en datos de entrenamiento como de prueba.

Técnicas para Tratamiento

- **Hold Out:** Dividir los datos en dos conjuntos: uno para entrenamiento y otro para prueba, para evaluar el rendimiento del modelo en datos no vistos.
- **Validación Cruzada:** Dividir los datos en varios subconjuntos y entrenar el modelo múltiples veces, cada vez con un subconjunto diferente como prueba, para obtener una evaluación más robusta del modelo.
- **Ajuste de Hiperparámetros:** Proceso de encontrar los mejores parámetros para el modelo que no se aprenden durante el entrenamiento, optimizando así el rendimiento del modelo.

Validación y Métricas

Métricas de Modelos de Regresión

- **Error Cuadrático Medio (MSE):** Mide la media de los errores cuadrados entre los valores predichos y los reales, penalizando errores grandes.
- **Raíz del Error Cuadrático Medio (RMSE):** Similar al MSE, pero en la misma escala que los valores predichos.
- **Error Absoluto Medio (MAE):** Promedio de los errores absolutos entre los valores predichos y los reales, proporcionando una medida fácil de interpretar.
- **R-cuadrado (R^2):** Indica la proporción de la variación en la variable dependiente que es explicada por las variables independientes del modelo.
- **R-cuadrado Ajustado:** Similar a R^2 , pero ajustado por el número de predictores en el modelo, penalizando por complejidad.

	<p>innecesaria.</p> <p>Métricas de Modelos de Clasificación</p> <ul style="list-style-type: none"> ● Matriz de Confusión: Tabla que muestra las verdaderas etiquetas frente a las predicciones del modelo, ayudando a evaluar la precisión del modelo. <ul style="list-style-type: none"> ○ True Positives (TP): Predicciones correctas de la clase positiva. ○ True Negatives (TN): Predicciones correctas de la clase negativa. ○ False Positives (FP): Predicciones incorrectas de la clase positiva. ○ False Negatives (FN): Predicciones incorrectas de la clase negativa. ● F1 Score: Combina precisión y recall en una sola métrica, proporcionando un balance entre ambas. ● Precisión: Proporción de predicciones positivas correctas sobre todas las predicciones positivas. ● Recall: Proporción de verdaderos positivos sobre todos los casos positivos reales.
<p>Experiencia a Misión 2</p>	<p>Simulación: Desarrollo Modelos de Machine Learning</p> <p>Para esta simulación se tomará como punto de partida el dataset elegido desde la primera misión, se asume que ya está concluido el análisis exploratorio de los datos realizado en la misión 2.</p> <p>El profesor debe asegurarse de que todos los estudiantes tengan acceso a los datos y las bibliotecas necesarias instaladas.</p> <p>Proporcionar un entorno de desarrollo adecuado (Jupyter Notebook, Google Colab, etc.).</p> <p>Paso 1: División del Conjunto de Datos</p> <p>Objetivo: Dividir el conjunto de datos en conjuntos de entrenamiento y prueba.</p> <p>Instrucciones:</p> <p>Utilizar <code>train_test_split</code> de sklearn para dividir los datos.</p>

	<p>Establecer una proporción de, por ejemplo, 80% para entrenamiento y 20% para prueba.</p> <p>Asegurarse de incluir la variable objetivo.</p> <p>Opcional: Utilizar validación cruzada (<code>cross_val_score</code> de <code>sklearn</code>) para una evaluación más robusta del modelo. Explicar a los estudiantes cómo la validación cruzada puede ayudar a evaluar la estabilidad y la generalización del modelo al dividir los datos en múltiples subconjuntos de entrenamiento y prueba.</p> <p>Paso 2: Implementación de Múltiples Modelos</p> <p>Objetivo: Entrenar varios modelos de Machine Learning y compararlos.</p> <p>Instrucciones:</p> <p>Seleccionar al menos tres modelos diferentes, por ejemplo: Regresión Logística, Árbol de Decisión y SVM.</p> <p>Entrenar cada modelo usando los datos de entrenamiento.</p> <p>Realizar predicciones con los datos de prueba.</p> <p>Paso 3: Evaluación de los Modelos</p> <p>Objetivo: Evaluar y comparar el rendimiento de los modelos.</p> <p>Instrucciones:</p> <p>Utilizar métricas como precisión, recall, F1-score y/o AUC-ROC.</p> <p>Crear una tabla comparativa para visualizar el rendimiento de cada modelo.</p> <p>Paso 4: Ajuste de Hiperparámetros</p> <p>Objetivo: Mejorar el rendimiento de los modelos ajustando los hiperparámetros.</p> <p>Instrucciones:</p> <p>Seleccionar uno o dos modelos para el ajuste de hiperparámetros.</p> <p>Usar <code>GridSearchCV</code> o <code>RandomizedSearchCV</code> de <code>sklearn</code> para encontrar los mejores parámetros.</p> <p>Volver a entrenar los modelos con los mejores parámetros encontrados y evaluar nuevamente su rendimiento.</p> <p>Paso 5: Interpretación de Resultados</p> <p>Objetivo: Analizar y discutir los resultados obtenidos.</p> <p>Instrucciones:</p> <p>Interpretar las métricas de rendimiento y discutir posibles razones para las</p>
--	--

	<p>diferencias entre los modelos. Considerar la importancia de las características en los modelos entrenados y cómo podrían influir en las predicciones. Reflexionar sobre posibles mejoras futuras y pasos adicionales que podrían tomarse.</p> <p>Cápsulas</p> <p>Recursos externos:</p> <p>Videos Regresión Lineal y Mínimos Cuadrados Ordinarios DotCSV: https://youtu.be/k964_uNn3l0?si=Z2kXHths20YwPEwy KNN: https://youtu.be/5l9AJD601n0 Naive Bayes y matriz de confusión: https://youtu.be/N4NSRYgKtxl Matriz de correlación y selección de características: https://youtu.be/ubOL_sLXzNU</p> <p>Ejemplos prácticos IA NOTEBOOK #1 Regresión Lineal y Mínimos Cuadrados Ordinarios Programando IA: https://youtu.be/w2RJ1D6kz-o?si=7MAgywOPdNH9GvAK Model Naive Bayes para clasificación con Python https://youtu.be/l3C6bZMZCDI?si=hkRfA1tkW_anb21W REGRESIÓN LOGÍSTICA - PRÁCTICA #38 Curso Machine Learning con Python https://youtu.be/rUHZb_TzWVs?si=bNeISbCO548y46V1 VECTORES DE SOPORTE REGRESIÓN - SCIKIT LEARN #22 Curso Machine Learning con Python: https://youtu.be/xzkvLAGBSkA?si=P3fp4kKbqL923elu</p> <p>Infografías 100-Days-Of-ML-Code https://github.com/Avik-Jain/100-Days-Of-ML-Code</p>
Conexión Misión 2	<p>Momento para conectar con el sector laboral, interactuar con profesionales, participar en conferencias, seminarios web, ferias de empleo y eventos de networking. Adicional, presentar sus proyectos digitales o ideas emprendedoras para atraer inversores, colaboradores o clientes interesados en sus proyectos digitales.</p>

	MISIÓN 3
Entrenamiento Misión 3	<p>Mapa de ruta: Objetivo</p> <p>Habilidades digitales</p> <ul style="list-style-type: none"> • Comprender los principios básicos de ciencia de datos, incluyendo la carga y manipulación de datos en Python usando bibliotecas como Pandas y NumPy. • Habilidad para cargar, limpiar, y transformar datos utilizando herramientas y bibliotecas de Python. • Habilidad para realizar un análisis exploratorio de datos, identificando patrones, tendencias y anomalías mediante técnicas de visualización y estadísticas descriptivas. • Implementación y evaluación de modelos de machine learning • Competencias • Capacidad para analizar y visualizar datos de manera efectiva, interpretando resultados y extrayendo conclusiones relevantes para la toma de decisiones. • Habilidad para identificar y resolver problemas en conjuntos de datos, como datos faltantes y sobreajuste, aplicando las técnicas adecuadas para mejorar la calidad del modelo. • Competencia en seleccionar y aplicar algoritmos adecuados para problemas específicos, y en ajustar y optimizar modelos para maximizar su rendimiento. • <p>Contenidos (Temáticas)</p> <p>Nombre de la misión 3:</p> <ul style="list-style-type: none"> • Procesamiento del lenguaje natural • Limpieza y Preprocesamiento de Datos: • Tokenización • Normalización • Eliminación de Stopwords • Lematización • Stemming

	<ul style="list-style-type: none"> • Representación de Texto • Bolsa de Palabras • TF-IDF • Análisis de sentimientos utilizando modelos de aprendizaje de máquina • Regresión Logística • Naive Bayes • Despliegue de modelos • Desarrollo de APIs RESTful para el acceso a modelos. <p>Ejecución del proyecto</p> <p>El proyecto de esta misión consiste en la implementación de un clasificador Naive Bayes básico para procesar frases y clasificarlas en categorías como saludo, queja, reclamo, petición o despedida. Los estudiantes definirán un conjunto de datos con frases etiquetadas, crearán un diccionario de palabras y calcularán frecuencias, y luego implementarán el clasificador Naive Bayes aplicando el conteo de palabras y la suavización de Laplace. Finalmente, probarán el clasificador con nuevas frases para evaluar su precisión y efectividad en la categorización.</p>
	<p>Preparación</p> <p>Procesamiento del Lenguaje Natural (PLN)</p> <p>Limpieza y Preprocesamiento de Datos</p> <p>Tokenización</p> <p>La tokenización es el proceso de dividir un texto en unidades más pequeñas, como palabras o frases. Esto se hace para facilitar el análisis y la manipulación de texto en tareas posteriores.</p> <ul style="list-style-type: none"> • Ejemplo: "El perro corre rápido" se convierte en ["El", "perro", "corre", "rápido"]. <p>Normalización</p>

La normalización implica transformar el texto a una forma consistente y estandarizada. Esto puede incluir la conversión a minúsculas, eliminación de puntuación y corrección ortográfica.

- **Ejemplo:** "El PERRO corre RÁPIDO!!!" se convierte en "el perro corre rápido".

Eliminación de Stopwords

Las stopwords son palabras comunes y de poca relevancia que se eliminan del texto para reducir el ruido y centrarse en las palabras más significativas.

- **Ejemplo:** Palabras como "el", "de", "y" se eliminan de "El perro corre rápido" para obtener ["perro", "corre", "rápido"].

Lematización

La lematización convierte las palabras a su forma base o lema, considerando el contexto y la parte del discurso. Esto ayuda a agrupar palabras con el mismo significado.

- **Ejemplo:** "corriendo" y "corre" se lematizan a "correr".

Stemming

El stemming es un proceso similar a la lematización, pero más rudimentario, que recorta las palabras a su raíz básica sin considerar el contexto gramatical.

- **Ejemplo:** "corriendo" y "corre" se reducen a "corr".

Representación de Texto

Bolsa de Palabras (Bag of Words)

La Bolsa de Palabras es un método de representación de texto que cuenta la frecuencia de las palabras en un documento sin considerar el orden. Cada documento se representa como un vector de frecuencias de palabras.

- **Ejemplo:** Para los textos "el perro corre rápido" y "el gato corre rápido", el vocabulario sería ["el", "perro", "corre", "rápido", "gato"], y los

vectores serían [1, 1, 1, 1, 0] y [1, 0, 1, 1, 1] respectivamente.

TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF es una técnica que refleja la importancia de una palabra en un documento y en una colección de documentos. Combina la frecuencia de términos (TF) con la frecuencia inversa de documentos (IDF).

- **Aplicación:** TF-IDF se usa para identificar palabras importantes en un conjunto de documentos, mejorando la relevancia en tareas como la búsqueda de información y la clasificación de texto.

Análisis de Sentimientos utilizando Modelos de Aprendizaje de Máquina

Regresión Logística

La regresión logística se utiliza para clasificar el sentimiento de un texto como positivo o negativo. Entrena el modelo en datos etiquetados y luego predice la probabilidad de que un nuevo texto pertenezca a una categoría.

- **Aplicación:** Clasificación de opiniones de productos, análisis de reseñas de películas.

Naive Bayes

El clasificador Naive Bayes es eficaz para el análisis de sentimientos debido a su simplicidad y eficiencia. Calcula la probabilidad de que un texto pertenezca a una categoría basándose en la ocurrencia de palabras.

- **Aplicación:** Filtrado de spam, análisis de sentimientos en redes sociales.

Desarrollo de APIs RESTful para el Acceso a Modelos

El despliegue de modelos de aprendizaje de máquina a menudo implica la creación de APIs RESTful que permiten a las aplicaciones acceder a los modelos y hacer predicciones en tiempo real. Una API RESTful es una interfaz que sigue principios de diseño REST (Representational State Transfer) y permite interactuar con servicios web.

- **Pasos para desarrollar una API RESTful:**

	<ol style="list-style-type: none"> 1. Diseño de la API: Definir los endpoints, métodos HTTP (GET, POST, PUT, DELETE) y los formatos de datos (JSON, XML). 2. Implementación: Utilizar frameworks como Flask o Django en Python para construir la API. Estos frameworks facilitan la creación de endpoints y la integración con modelos de ML. 3. Integración del Modelo: Cargar el modelo entrenado y utilizarlo en los endpoints para hacer predicciones basadas en las solicitudes de los usuarios. 4. Pruebas: Asegurarse de que la API funcione correctamente con diferentes entradas y manejar errores adecuadamente. <ul style="list-style-type: none"> • Ejemplo de endpoint de predicción: Un endpoint <code>/predict</code> que recibe un texto en una solicitud POST, lo preprocesa y pasa al modelo para obtener una predicción de sentimiento.
<p>Experiencia a Misión 3</p>	<p>Simulación</p> <p>Implementación de un modelo básico de machine learning basado en Naive Bayes</p> <p>En esta simulación, el profesor va a implementar un clasificador Naive Bayes básico para identificar si una frase es un saludo, una queja, un reclamo, una petición o una despedida, utilizando únicamente el conteo de palabras. A continuación, se detallan los pasos que seguirá el profesor para explicar esta actividad a los estudiantes.</p> <p>Se puede tomar como referencia el siguiente repositorio: https://github.com/felipebuitragocarmona/CodigoBase_ChatBot_NaiveBayes</p> <p>Paso 1: Definir el Conjunto de Datos El profesor definirá un conjunto de datos con frases de ejemplo clasificadas en cada categoría: Saludo: "Hola", "Buenos días", "Buenas tardes", "¿Cómo estás?" Queja: "Quisiera hacer una queja", "No estoy satisfecho con el servicio", "El producto llegó dañado" Reclamo: "Necesito reclamar por un error", "Quiero hacer un reclamo", "Esto es inaceptable, quiero una solución"</p>

	<p>Petición: "Me gustaría pedir información", "Necesito saber más detalles sobre el producto", "Por favor, envíenme más información"</p> <p>Despedida: "Adiós", "Hasta luego", "Nos vemos", "Chao"</p> <p>Cada frase se asociará con una etiqueta que indique su categoría.</p> <p>Paso 2: Crear un Diccionario de Palabras y Calcular Frecuencias El profesor creará un diccionario que cuente cuántas veces aparece cada palabra en cada categoría:</p> <ul style="list-style-type: none"> • Contar palabras por categoría: Por cada frase, se dividirán las palabras y se contarán las apariciones de cada palabra en su respectiva categoría. • Calcular la frecuencia total de palabras: Se sumarán todas las apariciones de palabras en cada categoría para obtener la frecuencia total de palabras por categoría. • Calcular el número total de frases: Se contará el total de frases en el conjunto de datos. <p>Paso 3: Implementar el Clasificador Naive Bayes Para clasificar una nueva frase, el profesor seguirá estos pasos: Dividir la frase en palabras: Se convertirán todas las palabras a minúsculas y se dividirán. Calcular las probabilidades por categoría:</p> <ul style="list-style-type: none"> • Para cada categoría, se empezará con la probabilidad de la categoría (cuántas frases pertenecen a esa categoría dividido por el total de frases). • Para cada palabra en la nueva frase, se calculará la probabilidad de que esa palabra aparezca en la categoría utilizando el conteo de palabras y aplicando la suavización de Laplace (sumar 1 al contador para evitar divisiones por cero). • Se sumarán los logaritmos de estas probabilidades para obtener el puntaje total para cada categoría. <p>1. Elegir la categoría con la mayor probabilidad: La categoría con el puntaje más alto será la predicción del clasificador para la nueva frase.</p> <p>Paso 4: Probar el Clasificador</p>
--	--

	<p>Para asegurarse de que el clasificador funcione correctamente, el profesor probará algunas frases nuevas, como:</p> <p>"Hola, ¿cómo estás?"</p> <p>"Quisiera hacer una queja sobre el servicio"</p> <p>"Esto es inaceptable"</p> <p>"Necesito información sobre el producto"</p> <p>"Hasta luego, nos vemos"</p> <ul style="list-style-type: none"> • Se observará qué categoría predice el clasificador para cada frase y se verificará que las predicciones sean razonables. • Paso 5: Exponer el Modelo a través de una API en Python • Para hacer que el clasificador Naive Bayes sea accesible a través de una API, el profesor debe seguir estos pasos: • Crear un Servidor API: Utilizar un framework de Python como Flask o FastAPI para crear un servidor API que pueda recibir solicitudes HTTP. • Integrar el Modelo: Cargar el modelo Naive Bayes y el conjunto de datos en el servidor API. Esto incluye: <ul style="list-style-type: none"> • Cargar los conteos de palabras y frecuencias. • Implementar la lógica del clasificador según los pasos detallados anteriormente. • Definir los Endpoints de la API: Implementar endpoints que permitan al usuario enviar frases al modelo para su clasificación. Por ejemplo: <ul style="list-style-type: none"> • POST /classify: Recibe una frase y devuelve la categoría predicha.
	<p>Cápsulas</p> <p>Recursos externos:</p> <p>Videos</p> <p>Natural Language Processing (NLP) Zero to Hero TensorFlow https://youtu.be/fNxaJsNG3-s?si=e8FDLvi-S5TCCnc0</p> <p>Ejemplos prácticos</p> <p>Curso de Procesamiento del Lenguaje Natural (NLP): https://youtu.be/Ulp0ZfxeqvA?si=NKzQqZ9SVSpk1JSw</p> <p>Introduction NLP Tutorial For Beginners In Python</p>

	<p>https://youtu.be/R-AG4-qZs1A?si=wxNewEPluHr-QsuY</p> <p>Infografías</p> <p>https://view.genially.com/613e61e093a2cf0d146b1c0e/interactive-content-infografia-ia-y-procesamiento-del-lenguaje-natural</p>
<p>Conexión Misión 3</p>	<p>Co-creación</p> <p>ChatBot Virtual Primario Preparativos Iniciales</p> <ul style="list-style-type: none"> • Se tomará como referencia el código explicado por el profesor en la presente misión • Ampliar el Conjunto de Datos • Los estudiantes deben ampliar el conjunto de datos original añadiendo más categorías y frases. Asegurarse de incluir al menos 5 categorías adicionales y varias frases por categoría. Ejemplos de nuevas categorías pueden ser: • Información de contacto: "¿Cómo puedo contactarlos?", "¿Cuál es el número de teléfono?" • Horario de atención: "¿Cuál es el horario de atención?", "¿En qué días están abiertos?" • Precios: "¿Cuánto cuesta el producto?", "¿Cuál es el precio de servicio?" • Política de devolución: "¿Cuál es la política de devoluciones?", "¿Puedo devolver el producto?" • Soporte técnico: "Tengo un problema técnico", "¿Cómo puedo solucionar un error?" • Cada frase nueva debe estar claramente etiquetada con la categoría correspondiente. • Actualizar el Diccionario de Palabras y Calcular Frecuencias • Los estudiantes deben actualizar el diccionario de palabras para incluir las nuevas frases y recalcular las frecuencias de palabras por categoría, así como las frecuencias totales. • Asegurarse de ajustar el número total de frases para reflejar la ampliación del dataset.

	<ul style="list-style-type: none"> • Implementar el Clasificador Naive Bayes • Modificar el código del clasificador Naive Bayes para manejar el nuevo conjunto de datos ampliado. • Verificar que el clasificador pueda predecir correctamente la categoría de nuevas frases. • Programar Respuestas Predefinidas • Definir respuestas preestablecidas para cada categoría. Por ejemplo: • Saludo: "¡Hola! ¿Cómo puedo ayudarte hoy?" • Queja: "Lamentamos los inconvenientes. ¿Podrías darnos más detalles para resolver tu queja?" • Reclamo: "Vamos a revisar tu reclamo. Por favor, proporciona más información." • Petición: "Entendido. Estamos procesando tu solicitud de información." • Despedida: "Gracias por contactarnos. ¡Hasta luego!" • Información de contacto: "Puedes contactarnos al 123-456-7890." • Horario de atención: "Nuestro horario de atención es de lunes a viernes, de 9:00 a 18:00." • Precios: "El precio del producto es de \$99.99." • Política de devolución: "Puedes devolver el producto dentro de los 30 días posteriores a la compra." • Soporte técnico: "Para soporte técnico, por favor visita nuestro centro de ayuda en [enlace]." • Desarrollar el Chatbot • Implementar una función de chatbot que utilice el clasificador Naive Bayes para determinar la categoría de una entrada del usuario y responda con la respuesta predefinida correspondiente. • El chatbot debe ser capaz de manejar entradas de texto, clasificar la categoría y proporcionar la respuesta adecuada. • Probar y Evaluar el Chatbot • Los estudiantes deben probar el chatbot con una variedad de preguntas para asegurarse de que clasifica
--	---

	<p>correctamente y responde apropiadamente.</p> <ul style="list-style-type: none"> • Evaluar el rendimiento del chatbot y ajustar el conjunto de datos y las respuestas si es necesario. • Exponer el Modelo a través de una API en Python • Crear un Servidor API: Utilizar un framework de Python como Flask o FastAPI para crear un servidor API que pueda recibir solicitudes HTTP. • Integrar el Modelo: Cargar el modelo Naive Bayes y el conjunto de datos en el servidor API. Esto incluye: <ul style="list-style-type: none"> • Cargar los conteos de palabras y frecuencias. • Definir los Endpoints de la API: Implementar endpoints que permitan al usuario enviar frases al modelo para su clasificación. • Entregables • Código: Código fuente que incluye el clasificador Naive Bayes actualizado y el chatbot. • Video: Un video explicativo donde se demuestre cómo funciona el chatbot, incluyendo ejemplos de preguntas y respuestas.
--	--