Inteligencia Artificial 2026

Primer Proyecto

02.marzo.2026

En este primer proyecto se abordarán agentes reactivos, con base en modelos de *machine learning*, ya sea para regresión o clasificación supervisada. Deberá elegir una de las siguientes opciones de proyecto.

Opción A: Predicción de Tarifas de Transporte en NYC

El objetivo de este proyecto es desarrollar un modelo predictivo capaz de estimar el costo de un viaje en taxi (`fare_amount`) en la ciudad de Nueva York, dadas las coordenadas de inicio y fin, la fecha, la hora y el número de pasajeros. Tome en cuenta que en el conjunto de datos proporcionado, se tienen datos ruidosos, valores faltantes y debe cubrir una fase de preparación de datos y construcción de ingeniería de características (Feature Engineering).

Obtención de los Datos

Para acceder a los datos, deben seguir estos pasos:

1. Ingresar a la URL:
   `https://www.kaggle.com/competitions/new-york-city-taxi-fare-prediction/data`
2. Descargar el archivo `train.csv`.
   Nota: El archivo original es muy pesado (5 GB+). Para fines del proyecto, se recomienda cargar una muestra de 1,000,000 de filas utilizando el parámetro `nrows` en `pandas.read_csv()`.

Fases del Proyecto

- Análisis Exploratorio de datos y Limpieza
  Antes de entrenar cualquier algoritmo, es imperativo asegurar la calidad de la información. Tome en cuenta:
  - Limpieza Geográfica: Eliminar registros con latitudes o longitudes que no pertenezcan a la zona metropolitana de NYC.
  - Limpieza de Etiquetas: Eliminar viajes con tarifas negativas o de $0.00.
  - Análisis de Outliers: Identificar y tratar viajes con distancias extremadamente largas o número de pasajeros inverosímiles (ej. 0 o más de 7).
- Ingeniería de Características (Feature Engineering)
  El modelo será tan bueno como los datos que reciba. Deben crear nuevas variables:
  - Distancia: Implementar la Fórmula de Haversine para calcular la distancia circular entre puntos geográficos.
  - Variables Temporales: Extraer del timestamp la hora del día, el día de la semana, el mes y el año.
  - Zonas de Interés: (Opcional) Crear variables binarias si el origen o destino es un aeropuerto (JFK, LaGuardia, Newark).

Experimentación con Modelos

La elección del algoritmo es libre. Deben comparar al menos tres de las siguientes arquitecturas y justificar cuál ofrece el mejor balance entre precisión y costo computacional:

- K-Nearest Neighbors (KNN)
- Random Forest o Gradient Boosting (XGBoost/LightGBM)
- Redes Neuronales Densas (MLP)
- Regresión Lineal Multivariada.

Está totalmente prohibido el uso de modelos pre-entrenados. Ustede debe entrenar todos los modelos a usar. Cualquier uso o sospecha de uso de modelos pre-entrenados se sancionará con una nota de 0 en el proyecto.

Para la optimización y evaluación de sus algoritmos, deberá hacer una selección adecuada de los hiperparámetros, y deberá mostrar tablas de métricas que permitan evaluar el desempeño de sus algoritmos. Conviene mostrar comparaciones entre los datos reales y predichos, ejemplos de puntos u observaciones con alta influencia en el modelos, y métricas promedio del error de predicción.

Opción B: Detección de Ciberacoso en Redes Sociales

En este proyecto el objetivo es desarrollar un sistema de clasificación automática que identifique si un mensaje de texto (*Tweet*) contiene contenido de odio o acoso. El dataset contiene más de 47,000 *tweets* etiquetados en categorías como: Age, Ethnicity, Gender, Religion, Other Cyberbullying y Not Cyberbullying.

Obtención de los Datos

Para acceder a los datos, deben seguir estos pasos:

1. Ingresar al url:
   `https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification`.
2. Descargar el archivo `cyberbullying_tweets.csv`.
   Nota: El dataset está bien balanceado, lo que permite enfocarse más en el procesamiento del texto que en corregir desequilibrios de clase.

Fases del Proyecto

- Pre-procesamiento de Texto (NLP Pipeline)
  A diferencia de los datos numéricos, el texto requiere una limpieza profunda para que los algoritmos puedan procesarlo:
  - Limpieza: Eliminar URLs, menciones (usuario), hashtags, caracteres especiales y emojis.
  - Normalización: Convertir todo el texto a minúsculas y eliminar *stop-words*.
  - Tokenización y Lematización: Reducir las palabras a su raíz (ej. "corriendo" -¿ "correr") para reducir la dimensionalidad.
- Vectorización (De Texto a Números). Los modelos de IA no leen palabras, leen números. Deberá implementar y comparar varios esquemas de vectorización:
  - Bag of Words (BoW) o TF-IDF: Para modelos estadísticos tradicionales.
  - Word Embeddings: (Opcional) Uso de representaciones vectoriales densas si deciden usar Redes Neuronales.

Experimentación con Modelos

La elección del algoritmo es libre. Deben comparar al menos tres de las siguientes arquitecturas y justificar cuál ofrece el mejor balance entre precisión y costo computacional:

- Modelos Probabilísticos: Naive Bayes (clásico para texto).
- Modelos Lineales: Regresión Logística o Support Vector Machines (SVM).
- Ensambles: Random Forest o Gradient Boosting.
- Deep Learning: Redes Neuronales Densas o capas de Recurrente (SimpleRNN/LSTM).

Está totalmente prohibido el uso de modelos pre-entrenados. Ustede debe entrenar todos los modelos a usar. Cualquier uso o sospecha de uso de modelos pre-entrenados se sancionará con una nota de 0 en el proyecto.

Para la optimización y evaluación de sus algoritmos, deberá hacer una selección adecuada de los hiperparámetros, y deberá mostrar tablas de métricas que permitan evaluar el desempeño de sus algoritmos. Conviene mostrar matrices de confusión, ejemplos de clases que son comunes de confunción, y ejemplos de tweets bien y mal clasificados.

Entregables (para ambos proyectos)

1. Reporte técnico del proyecto
2. Código y recursos utilizados.
   El código debe subirse a Canvas. Por favor, todo el código debe ser empaquetado el un archivo `.zip` o `.rar`, y entregarlo junto con el reporte en formato `.pdf`.
   Tomar en cuenta que en el reporte no debe ir código. El reporte debe entregarse fuera del archivo comprimido, para facilitar la lectura. **El no cumplimiento de lo anterior tendrá una penalización sobre su nota del proyecto**.

El reporte debe incluir:

- Presentación de los datos, análisis exploratorio. Gráficas de distribuciones o correlación. Análisis de datos faltantes o atípicos. Técnicas de tratamiento de datos faltantes. Tratamiento de datos atípicos. Transformaciones aplicadas a los datos (One-hot, reescala, BOW, TF-IDF, Vectorización, ...).
- Partición del conjunto train y test. Explicación de los métodos utilizados.
- Descripción técnica de los modelos probados: arquitectura, parámetros e hiperparámetros. Deberá indicar una justificación de éstos.
- Justificación del modelo: Por qué el algoritmo elegido es superior a los demás probados.
- Tablas comparativas: Comparar métricas de entrenamiento vs. test para evitar el overfitting. Visualización de Residuos: Un gráfico de dispersión de Predicciones vs. Valores Reales / o matrices de confusión.
- Pruebas de Usuario: Una función que reciba inputs (coordenadas y fecha/hora, o un tweet) y devuelva la predicción.

Fecha de Entrega: lunes 6 de abril.
Se asignará una hora para que cada grupo presente sus resultados.