Laboratorio 9.
Spark MLlib

INSTRUCCIONES:

En esta guía se detallan las instrucciones para poder completar los ejercicios que complementan el contenido visto en la semana. El objetivo de la misma es que el estudiante se familiarice y explore de primera mano el módulo de Spark Mllib.

Realizar de forma **grupal**, si surge alguna duda siéntase en la comodidad y libertad de hacer preguntas al catedrático. Para este único laboratorio, deberá utilizar su ambiente dentro de **Databricks**. **NOTA**: es importante que su ambiente lo conecte con un repositorio de GIT para asegurar el trabajo colaborativo de la mejor manera posible.

PREPARACIÓN DEL AMBIENTE

El enfoque para este laboratorio será que trabajen desde los datos limpios o unificados del laboratorio anterior. Se recomienda que tengan un único dataframe, idealmente en formato Parquet. Con ello, necesitarán explorar patrones con algoritmos no supervisados (KMeans, PCA), y luego construyan pipelines de clasificación o regresión (por ejemplo, predecir el tipo de accidente, el número de fallecidos o la severidad, etc).

En este laboratorio nos enfocaremos en aplicar técnicas de Machine Learning con PySpark MLlib, tanto para explorar patrones como para predecir la severidad y frecuencia de los accidentes.

EJERCICIOS

La fuente de datos corresponde a los accidentes de tránsito en Guatemala (2013–2023), divididos en tres grandes bases:

- Hechos de tránsito: número de accidentes clasificados por año, mes, departamento, día de la semana, hora, tipo de accidente, y en algunos casos por zona (en el municipio de Guatemala).

- Vehículos involucrados: cantidad y características de vehículos (tipo, color, modelo), además de información sobre los conductores (sexo, edad, condición).

- Fallecidos y lesionados: víctimas por accidentes, con desagregación por edad, sexo, tipo de accidente, hora, día, mes y departamento.

Un detalle importante es que los datos están en distintas hojas de un mismo Excel, y no existe un ID único que relacione directamente las tres bases. Para unirlas habrá que apoyarse en columnas comunes como: año, mes, día, hora, departamento, zona, tipo de accidente, etc.

ANÁLISIS EXPLORATORIO AVANZADO Y SEGMENTACIÓN (25 pts)

A continuación, responda las siguientes preguntas:

1. (Preparación) Cargar los DataFrames finales del laboratorio anterior (hechos, vehículos, fallecidos/lesionados) desde los archivos Parquet. Mostrar el esquema y las primeras filas de cada uno. Crear un DataFrame combinado con variables relevantes para el análisis exploratorio: año, mes, día de la semana, hora, zona, departamento, tipo de accidente, número de vehículos involucrados, lesionados y fallecidos.

2. (5 pts) Realizar un análisis de correlaciones entre las variables numéricas. Utilizar **VectorAssembler** y **Correlation.corr()** de MLlib. Comentar qué variables parecen tener relación con la severidad del accidente.

3. (10 pts) Aplicar una reducción de dimensionalidad con PCA (Principal Component Analysis) sobre las variables numéricas (hora, número de vehículos, lesionados, fallecidos, etc.) y mostrar los dos primeros componentes. Graficar con **display()** o **toPandas().plot()** la distribución de los accidentes en ese plano PCA.

4. (10 pts) Implementar una segmentación usando KMeans sobre las variables estandarizadas (por ejemplo: hora, número de vehículos, lesionados, fallecidos).
    a. Determinar un número adecuado de clústers (k=3 o 4).
    b. Mostrar el conteo de accidentes por clúster y caracterizar cada uno (por ejemplo: un clúster con más accidentes nocturnos o con más fallecidos).

MODELADO SUPERVISADO PARA PREDICCIÓN (75 pts)

5. (5 pts) Crear una nueva columna severidad que clasifique los accidentes en categorías:
    a. Leve: sin fallecidos, menos de 2 lesionados
    b. Moderado: 1 fallecido o entre 2–5 lesionados
    c. Grave: más de 1 fallecido o más de 5 lesionados

Mostrar la distribución de esta variable.

6. (5 pts) Dividir los datos en entrenamiento (70%) y prueba (30%). Utilizar una semilla fija para reproducibilidad.

7. (10 pts) Construir un pipeline de clasificación que incluya:
    a. StringIndexer para variables categóricas (tipo_accidente, departamento, zona, día_semana)
    b. VectorAssembler para variables numéricas (hora, número de vehículos, lesionados, fallecidos, etc.)
    c. StandardScaler
    d. Modelo de RandomForestClassifier
    e. Entrenar el modelo y mostrar las métricas de precisión, recall y F1-score.

8. (15 pts) Comparar el rendimiento del modelo anterior con un **LogisticRegression** y un **DecisionTreeClassifier** utilizando el mismo pipeline.
    a. Mostrar una tabla con las métricas principales para cada modelo.
    b. Discutir brevemente cuál se comporta mejor y por qué podría ser.

9. (15 pts) Crear un modelo de regresión lineal que intente predecir el número de fallecidos en función de las variables tipo de accidente, zona, hora, número de vehículos y lesionados.
    a. Evaluar con RMSE, MAE y R^2.
    b. Guardar el modelo en formato .parquet o .model.

10. (5 pts) Recargar el modelo guardado y probarlo sobre un subconjunto nuevo (por ejemplo, datos del año 2023). Mostrar los resultados reales vs. predichos.

11. (5 pts) Graficar las predicciones del modelo de regresión (predicho vs real) y discutir si el modelo tiende a subestimar o sobreestimar los valores altos.

EVALUACIÓN

- **(25 puntos)** Haber completado con exactitud la primera sección.
- **(75 puntos)** Haber respondido de manera correcta evidenciando el paso a paso de su solución para toda la segunda sección.

NOTA: para poder obtener derecho a nota, cada estudiante del grupo deberá haber entregado el ejercicio del día Lunes 06 de Octubre.

MATERIAL A ENTREGAR

- Script de Python (.ipynb) que utilizó con la discusión generada utilizando markdown.
- Link del repositorio usado para versionar el código.

NOTA: se deberá de evidenciar el trabajo colaborativo de todos los integrantes para obtar por calificación.

FECHAS DE ENTREGA

- **AVANCE:** Jueves 9 de octubre de 2025 17:20hrs: Actividades de la **1 a la 6** de la sección de ejercicios.
- **DOCUMENTO FINAL COMPLETO:** Domingo 12 de octubre de 2025 a las 23:59.