Análisis de series temporales

Práctico de Introducción al Aprendizaje Automático

En este práctico vamos a introducirnos en las primeras herramientas de aprendizaje automático. Se propone diseñar e implementar algunos modelos simples y definir métricas para ver como performan.

Instalación

Recomiendo instalar un ambiente virtual con las librerías estándar para procesar datos y visualizarlos, para evitar contratiempos.
Las siguientes librerías son más que suficiente para cumplir con los objetivos de este práctico: **jupyterlab, numpy**, **pandas**, **matplotlib**, **scikit-learn**.

Informe

Se deberá realizar un informe en formato jupyter notebook donde se presente la información solicitada, y se explique las decisiones tomadas a lo largo del análisis y los procesos de curación. Se espera que el mismo notebook donde se realizan las acciones, tenga un formato de informe con títulos y texto en markdown en cada sección, como se muestra en el informe de ejemplo.
Una vez terminado el informe se deberá **exportar a HTML** para evitar problemas de lectura y facilitar la corrección.
El informe se puede organizar a su mejor criterio, sin omitir la información solicitada.
Se espera que se realicen gráficos cada vez que se pueda, aunque no se lo pida de forma explícita.
Recuerden no comitear el dataset ni los modelos entrenados.

Implementación

Para simplificar la tarea en este práctico solo vamos a diferenciar entre envíos **rápidos** y **lentos**, donde un envío es rápido si llega antes de **3 días hábiles**, y lento si llega después de 3 días hábiles.
Esto nos reduce los problemas de clasificación a solo **2 clases**.

Definiciones básicas:

- Definir una **métrica binaria** para evaluar los modelos
- Diseñar un modelo **baseline** para los envíos de SP a SP, como el modelo más simple posible. Para esto no necesitamos machine learning, solo proponer una heurística a partir de los datos observados en los prácticos de análisis
- Calcular la métrica y la matriz de confusión para el baseline

Preparación de los features:

- Con la intención de salvar las rutas poco representadas, implementar una codificación para los features: **sender_zipcode** y **receiver_zipcode**
- Seleccionar un conjunto de features para entrenar modelos de machine learning

Clustering:

- Clusterizar los envíos basados únicamente en las **rutas**. Para esto recomiendo utilizar **KMeans**
- (*) Cual es el número óptimo de clusters? (Ver método de **Elbow** con KMeans)
- Describir brevemente las características interesante de los clusters

Modelos lineales:

- Leer sobre **accuracy**, **precision** y **recall** para agregarlas al pool de métricas que vamos a utilizar.
- Implementar un modelo basado en **regresión lineal**, calcular las métricas y la matriz de confusión
- Implementar un modelo basado en **regresión logística,** calcular las métricas y la matriz de confusión
- Cual es la principal diferencia entre estos modelos? Tuviste que hacer algún tipo de post-procesamiento?
- Estandarizar los features seleccionados y re entrenar los modelos. Las métricas mejoran? Explicar por qué.

Recuerden que todas las respuestas y decisiones tomadas tienen que estar justificadas con datos o gráficos. Se evaluará la legibilidad del notebook, el detalle a la hora de responder las preguntas y mostrar la información solicitada, y además que los gráficos utilizados sean apropiados y correctos.