Proyecto:

MDS7202: Laboratorio de Programación Científica para Ciencia de Datos

Cuerpo Docente:

- Profesor: Ignacio Meza De La Jara, Sebastián Tinoco
- Auxiliar: Catherine Benavides, Consuelo Rojas
- Ayudante: Eduardo Moya, Nicolás Ojeda

Motivación

De la Sombra a la Luz: El Legado de la Familia Mancini

En la oscura ciudad de Noirville, un banco ha sido conocido durante años como un centro de operaciones de la mafia. Bajo la dirección del infame Don Vittorio "El Zorro" Mancini, el banco prosperó, pero su reputación quedó manchada por actividades ilícitas y operaciones fraudulentas. Sin embargo, el curso de los acontecimientos cambió radicalmente cuando Don Vittorio falleció, dejando el banco en manos de su hijo, Marco Mancini.

Marco, un joven ambicioso con una visión diferente, se ha propuesto limpiar la reputación del banco y convertirlo en una institución legítima y respetada. Para lograrlo, necesita identificar y eliminar a todos los clientes con un pasado fraudulento que todavía utilizan el banco para sus actividades criminales.

Es así como ustedes entran a participar en un concurso de ML creado por Marco, creando un modelo de aprendizaje automático (ML) con el cual puedan identificar los casos fraudulentos. Esta es su oportunidad de demostrar sus habilidades y ayudar a Marco Mancini a transformar el legado de su familia, llevando al banco a una nueva era.

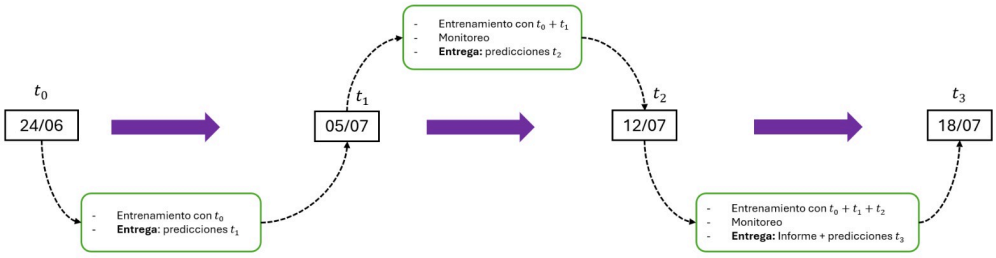
El concurso consta de 3 etapas y se les subirán datos semanalmente como en un problema real. En cada etapa del concurso deberán subir los resultados de su mejor modelo a CodaLab con la función `generateFiles` disponible en el anexo. Tienen hasta 3 intentos para subir sus modelos en cada etapa del concurso.

- Las fechas de las competencias son:
  - Entrega Parcial 1: 24/Junio al 04/Julio
  - Entrega Parcial 2: 05/Julio al 11/Julio
  - Entrega Final: 12/Julio al 18/Julio

Por favor, lean detalladamente las instrucciones de la tarea antes de empezar a escribir.

Reglas

- El proyecto consta de 2 entregas parciales y una entrega final, con las siguiente fechas de entrega:
  - Entrega parcial 1: 04/Julio
  - Entrega parcial 2: 11/Julio
  - Entrega final: 18/Julio



- Cualquier duda fuera del horario de clases al foro. Mensajes al equipo docente serán respondidos por este medio.
- Estrictamente prohibida la copia.
- Pueden usar cualquier material del curso que estimen conveniente.
- Hay BONUS para los equipos que tengan los mejores resultados en la competencia.
- **Grupos de 2 personas.**
- Por cada entrega que no suban a CodaLab, se descontará 1 punto de su nota final.

- Entregables:
  - Para la Entrega Parcial 1, tienen a su disposición los datos bancarios en el archivo **X_t0.csv** e **y_t0.csv** para el modelamiento, y **X_t1.csv** para generar las predicciones para la primera competencia en CodaLab.
  - Al final de cada entrega (parcial 1, parcial 2 y final), tienen que seleccionar el mejor modelo y predecir el conjunto de prueba para reportar sus resultados en la competencia de CodaLab. Para esto, utilizar la función `generateFiles` que se encuentra en Anexos **para generar el archivo a entregar en la competencia**.
  - Para la entrega final deberán incluir, adicionalmente, un informe que abarque todo su trabajo realizado. Se recomienda ir escribiendo el informe en paralelo a la creación de los modelos.

Notas adicionales

- No necesitan tener un rendimiento cercano al 100% para tener una resolución exitosa en el proyecto.
- Utilizar paralelización para acelerar búsquedas. Esto podría ser una buena solución para el caso de que la búsqueda de hiperparámetros sea muy lenta. En caso de tener problemas de RAM, reducir la cantidad de `jobs` a algo que su computador/interprete web pueda procesar.
- Generar grillas de búsquedas razonables. Entre más grande es la grilla, más lento el proceso de búsqueda. Utilice grillas de tamaños adecuados para que la búsqueda converga en tiempos razonables y no se demore 3.5 eternidades en terminar.

Objetivos: Entrega Parcial 1

A continuación, se detallan las tareas que deberían realizar en la entrega parcial 1 (04/Julio). **Recuerden que el entregable es predecir con su mejor modelo sobre los clientes de X_t1.csv y subirlo a CodaLab**.

- Análisis exploratorio de datos.
- Pre-Procesamiento, estos pueden incluir:
  - Estandarizar de filas y/o columnas.
  - Disminución de dimensionalidad.
  - Discretización de variables númericas a categoricas.
  - Manejo de datos nulos.
  - Etc.
- Hold Out, realizar un Hold Out de un 70/30 para el entrenamiento y testeo.
- Modelo baseline (sencillo).
- 3 modelos de ML distintos al baseline. Es importante destacar que en esta iteración del proyecto sólo pueden utilizar **pipelines de ScikitLearn**. La utilización de cualquier otro elementos es penalizado con puntaje 0 en las secciones implicadas.
- Optimización de modelos de ML.
- Interpretabilidad del modelo con mejores resultados.

Link Competencia CodaLab

Acuerdese de utilizar el archivo correspondiente para hacer la entrega de resultados a la competencia. Utilizar los datos equivocados se va a ver reflejado en un bajo desempeño en la tabla de resultados.

Instrucciones del Informe

La siguiente lista detalla las secciones que debe contener su notebook para resolver el proyecto.
Es importante que al momento de desarrollar cada una de las secciones, estas sean escritas en un formato tipo **informe**, donde describan detalladamente cada uno de los puntos realizados.

1. Introducción [0.25 puntos]

Esta sección es una muy breve introducción con todo lo necesario para entender que hicieron en su proyecto.

- Describir brevemente el problema planteado (¿Qué se intenta predecir?)
- Describir brevemente los datos de entrada que les provee el problema.
- Describir las métricas que utilizarán para evaluar los modelos generados. Elijan **una métrica** adecuada para el desarrollo del proyecto **según la tarea que deben resolver y la institución a la cuál será su contraparte** y luego justifiquen su elección. Considerando que los datos presentan desbalanceo y que el uso de la métrica 'accuracy' sería incorrecto, enfoquen su elección en una de las métricas precision, recall o f1-score y en la clase que será evaluada.
- [Escribir al final] Describir brevemente los modelos que usaron para resolver el problema (incluyendo las transformaciones intermedias de datos).
- [Escribir al final] Indicar si lograron resolver el problema a través de su modelo final. Indiquen además si creen que los resultados de su mejor modelo son aceptables y como les fue con respecto al resto de los equipos.

2. Modelos con Scikit-Learn (Entrega Parcial 1)

2.1 Análisis Exploratorio de Datos [0.5 puntos]

Esta sección consiste en realizar un análisis exploratorio de datos para investigar patrones, tendencias y relaciones en un conjunto de datos.

2.2 Pre-Procesamiento de datos [0.25 puntos]

Sección consiste en la realización de una limpieza o preprocesamiento de los datos para la creación posterior de los modelos.

Recuerde ejecutar `train_test_split` para generar un conjunto de entrenamiento y validación.

Se recomienda utilizar distintos tipos de procesamientos, como:

- ColumnTransformer
- Imputación de nulos
- Discretización de variables
- Etc.

2.3 Baseline [0.25 puntos]

En esta sección se debe detallar la creación del modelo más básico posible que resuelva el problema. La idea es utilizar este modelo de manera comparativa a los modelos a crear en la sección 5 (modelos de ML).

Implemente, entrene y evalué un modelo enfocado en resolver el problema de clasificación. Para esto, utilice **Pipeline**. Cada pipeline debe contener, el preprocesamiento anterior y un clasificador.

Imprimir `classification_report` .

2.4 Modelos de ML [0.5 puntos]

En esta sección, se explicitan los otros 3 modelos realizados. Explicar diferencias e hiperparámetros. Estos modelos también deben contar con un `Pipeline` y obtener su `classification_report` .

Dentro de los clasificadores a utilizar, tienen:

- `LogisticRegression` .
- `KNeighborsClassifier` .
- `DecisionTreeClassifier`
- `SVC`
- `RandomForestClassifier`
- `LightGBMClassifier` (del paquete `lightgbm`)
- `XGBClassifier` (del paquete `xgboost`)

Responder a las siguientes preguntas:

- ¿Hay algún clasificador entrenado mejor que el azar (`Baseline`)?
- ¿Cuál es el mejor clasificador entrenado?
- ¿Por qué el mejor clasificador es mejor que los otros?
- Respecto al tiempo de entrenamiento, con cual cree que sería mejor experimentar (piense en el tiempo que le tomaría pasar el modelo por una grilla de optimización de hiperparámetros).

Finalmente, de los 3 modelos **elija solo 1** para desarrollar las siguientes secciones. Justifique su elección en términos metodológicos.

2.5 Optimización de modelos [0.5 puntos]

Acá se explica la forma en la que se realizó la optimización de los modelos. Hiperparámetros ocupados.
Deberán usar `Optuna` para tunear hiperparámetros. Además de crear pipelines para cada uno de los modelos.

Algunas ideas para mejorar el rendimiento de sus modelos:

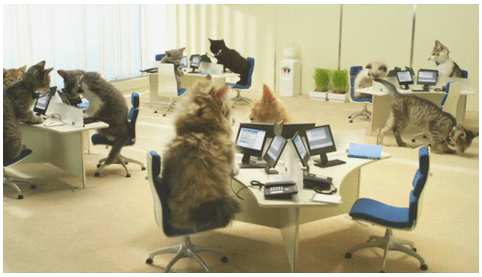
- Técnicas de selección de atributos.
- Variar el imputador de datos, en caso de usarlo.

2.6 Interpretabilidad [0.5 puntos]

Utilización de `SHAP` , `Anchors` u otras herramientas de interpretabilidad, para ver la importancia de cada atributo en el modelo final. Explicar o justificar la importancia de cada uno.

3. MLOPS (Entrega Parcial 2 y Entrega Final) [2.5 puntos]

Las especificaciones de esta sección se darán en el enunciado de la Parte 2, el día 05/Julio.



4. Resultados [0.5 puntos]

Resultados de los modelos a lo largo de las iteraciones. Recordar que van a haber 3 épocas de entrenamiento, por lo que se necesitan tener los resultados de los modelos al paso del tiempo, para ver los rendimientos y entender que tuvo que ser ajustado o cambiado para resolver el problema.

Se espera que en esta sección ustedes puedan comentar en **términos metodológicos** los resultados encontrados respondiendo preguntas como: ¿Cual fue el performance de sus modelos para resolver el problema? ¿Qué significa esto en términos prácticos? ¿Cómo fueron cambiando sus resultados con el paso de las iteraciones? ¿Qué fenómeno o hiperparámetro podría estar explicando gran parte de sus resultados? ¿Qué es lo que hace que exista un overfitting o underfitting en los resultados de los distintos modelos y como influye esto en su elección del mejor modelo? En otras palabras, intenten explicar sus resultados lo mejor posible desde el punto de vista del modelo.

5. Conclusiones [0.25 puntos]

Conclusiones generales del proyecto. En esta sección se espera que puedan concluir de forma general sobre el proyecto realizado. Es indispensable que conecten todas las secciones del informe, desde la Introducción hasta sus Resultados. Entre los temas a describir, deben mencionar los modelos con mejores resultados, conclusiones generales sobre los datos y las herramientas utilizadas.

Anexos

Función para exportar los resultados de su mejor modelo a CodaLab.

```
from zipfile import ZipFile
import os

def generateFiles(predict_data, clf_pipe):
    """Genera los archivos a subir en CodaLab

    Input
    ---------------
    predict_data: Dataframe con los datos de entrada a predecir
    clf_pipe: pipeline del clf

    Ouput
    ---------------
    archivo de txt
    """
    y_pred_clf = clf_pipe.predict_proba(predict_data)[:, 1]

    with open('./predictions.txt', 'w') as f:
        for item in y_pred_clf:
            f.write("%s\n" % item)

    with ZipFile('predictions.zip', 'w') as zipObj:
       zipObj.write('predictions.txt')
    os.remove('predictions.txt')

generateFiles(X_test, dummy_clf)
```