Universidad de
SanAndrés

I302 - Aprendizaje Automático y Aprendizaje Profundo

2do Semestre 2024

Trabajo Práctico 3

Fecha de entrega: Domingo 22 de septiembre, 23:59 hs.

Formato de entrega: Los archivos desarrollados deben ser entregados en un archivo comprimido .zip a través del Campus Virtual, utilizando el siguiente formato de nombre de archivo: *Apellido_Nombre_TP3.zip*. Se aceptará únicamente 1 archivo por estudiante. En caso de que el nombre del archivo no cumpla con la nomenclatura especificada, el trabajo no será corregido.

La carpeta comprimida deberá constar de N sub-carpetas, una por cada problema del TP (es decir, cada problema tiene su sub-carpeta denominada "Problema N"). Dentro de cada sub-carpeta deberá incluir un Jupyter Notebook llamado *Entrega_Problema_N.ipynb* en el cual se den las respuestas a los incisos del problema y se muestren los gráficos resultantes. Puede agregar resultados o análisis adicionales si lo considera necesario. Se recomienda fuertemente no realizar todo el desarrollo dentro del Jupyter Notebook; en su lugar, se sugiere usar archivos .py para desarrollar el código, siguiendo las buenas prácticas de programación y modularización vistas en clase.

Se recomienda seguir la estructura de archivos sugerida al final del trabajo práctico.

---

1. **Diagnóstico de Cancer Mamario.** El conjunto de datos de este problema consta de características computadas a partir del procesamiento de imágenes digitales de biopsias de células mamarias. Esto incluye características relacionadas con el tamaño, la forma y la textura de las células. Además, se incluye el diagnóstico del tumor como benigno o maligno. Para una descripción más detallada del dataset consulte el archivo *breast_cancer_description.md*.

   a) Utilizando el conjunto de datos de desarrollo *breast_cancer_dev.csv* desarrollar distintos modelos de regresión logística binaria con regularización L2, para cada uno aplicando una de las siguientes técnicas de re-balanceo. Para ajustar el hiperparámetro de regularización puede utilizar F-Score como métrica de performance.

      1) Sin aplicar ninguna técnica de re-balanceo.
      2) Undersampling: eliminar muestras de la clase mayoritaria de manera aleatoria hasta que ambas clases tengan igual proporción.
      3) Oversampling mediante duplicación: duplicar muestras de la clase minoritaria de manera aleatoria, hasta que que ambas clases tengan igual proporción.
      4) Oversampling mediante SMOTE (Synthetic Minority Oversampling Technique): hasta que ambas clases tengan igual proporción.
      5) Cost re-weighting: en la función de costo, multiplicar los terminos que dependen de las muestras de la clase minoritaria por un factor $C = \frac{\pi_2}{\pi_1}$, donde π_1 es la probabilidad a-priori de la clase minoritoria y π_2 el de la clase mayoritaria. Esto efectivamente re-balancea la importancia de tener errores de clasificacion de ambas clases.

   b) Utilizando el conjunto de datos de test *breast_cancer_test.csv* evalue la performance de cada uno de los modelos desarrollados anteriormente computando las siguientes métricas:

      - Matriz de confusión
      - Accuracy
      - Precision
      - Recall
      - F-Score
      - Curva PR
      - Curva ROC
      - AUC-ROC
      - AUC-PR

      Para las curvas PR y ROC, grafique las curvas de cada modelo sobre el mismo grafico, de manera de poder comparar las características de cada uno. Para las metricas de performance escalares, ponga los resultados de cada modelo en una sola tabla, como se muestra a continuación:

Modelo	Accuracy	Precision	Recall	F-Score	AUC-ROC	AUC-PR
Sin rebalanceo						
Undersampling						
Oversampling duplicate						
Oversampling SMOTE						
Cost re-weighting						

c) En base a los resultados obtenidos definir qué modelo enviará a producción (es decir, cual considera que es "el mejor") y justifcar por qué.

2. **Riesgo de Diabetes.** La diabetes es una afección crónica que afecta el equilibrio de insulina y los niveles de azúcar en sangre. Es uno de los principales factores de riesgo para la salud en adultos a nivel mundial. Una vez que la diabetes se desarrolla, es imposible revertirla, por lo que es fundamental predecir la probabilidad de su aparición con el fin de intervenir a tiempo y prevenir su avance. En este problema, desarrollarán distintos modelos predictivos para estimar la probabilidad de que una persona se encuentre en uno de tres estadíos de la diabetes ("no diabetes", "pre-diabetes", "diabetes") utilizando los datos recolectados por el Center for Disease Control (CDC) de EE.UU. Para una descripción más detallada del dataset consulte el archivo *diabetes_ description.md*.

   a) Utilizando el conjunto de datos *diabetes_ dev.csv* desarrollar un modelo predicivo con cada una de las arquitecturas:

      1) Análisis discriminante lineal (Linear Discriminant Analysis, LDA).
      2) Regresión logística multi-clase. Recuerde que puede agregar regularización.
      3) Bosque aleatorio (Random Forest) utilizando la entropía como criterio de división. Se recomienda experimentar con diferentes configuraciones de hiperparámetros (número de árboles, profundidad máxima, etc.) y seleccionar la mejor combinación en función de los resultados obtenidos.

   b) Evaluar las distintas métricas de performance (matriz de confusión, accuracy, precision, recall, f-score, curva PR, curva ROC, AUC-ROC y AUC-PR) de cada modelo sobre el conjunto de datos *diabetes_ test.csv*, y presentar los resultados de manera compacta para facilitar la comparación entre los modelos. Definir cuál de ellos enviará a producción y justificar por qué.

   c) OPCIONAL: Utilizando el conjunto de desarrollo, elaborar un modelo ensamblado mediante "stacking" de los tres modelos anteriores, y evaluar las distintas métricas de performance del modelo ensamblado sobre el conjunto de test. ¿Mejoró la performance?

Estructura Sugerida para la Entrega del Trabajo Práctico

Para organizar el desarrollo de este trabajo práctico de manera efectiva, recomendamos modularizar las diferentes funcionalidades en archivos .py y carpetas separadas de forma de facilitar la reutilización de código y la depuración. Una posible estructura de entrega podría ser:

```
Apellido_Nombre_TP3.zip
|- Problema 1/
    |- data/
        |- raw/           # Datos originales sin modificar
            |- breast_cancer_dev.csv
            |- breast_cancer_test.csv
        |- processed/     # Datos procesados y curados
    |- src/
        |- Entrega_Problema_1.ipynb    # Respuestas de los incisos
        |- ...                         # Agregar archivos .py necesarios

|- Problema 2/
    |- data/
        |- raw/           # Datos originales sin modificar
            |- diabetes_dev.csv
            |- diabetes_test.csv
        |- processed/     # Datos procesados y curados
    |- src/
        |- Entrega_Problema_2.ipynb    # Respuestas de los incisos
        |- ...                         # Agregar archivos .py necesarios

|- requirements.txt       # Especificar dependencias del proyecto
|- README.md              # Descripción del TP e instrucciones de uso
|- ...                    # Agregar archivos .py según sea necesario
```

Esta estructura es flexible. Se pueden agregar o eliminar archivos según sea necesario, pero es obligatorio incluir un Jupyter Notebook para cada problema con todas las respuestas a los incisos.