Universidad de
SanAndrés

I302 - Aprendizaje Automático y Aprendizaje Profundo

1er Semestre 2024

Trabajo Práctico 3

Fecha de entrega: Viernes 26 de abril, 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 deberán incluir un Jupyter Notebook (archivo *.ipynb*) en el cual se den las respuestas a los incisos del problema y se muestren los gráficos resultantes. Pueden agregar en el Jupyter Notebook mas resultados o análisis de los expresamente solicitados en los incisos, si lo consideran adecuado. Junto con el Jupyter Notebook podrá haber uno o mas archivos de Python que contengan código que sea parte de la resolución del problema. Dicho código debe seguir las buenas prácticas de programación y modularización que se presentaron en clase.

---

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 masas 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*.

   En este problema, nuestro objetivo es desarrollar varios modelos para clasificar las masas mamarias de cada paciente como benignas o malignas, y luego evaluar la eficacia de cada uno de estos. Para esto, el conjunto de datos se dividió previamente en uno de entrenamiento (*breast_cancer_train.csv*), uno de validación (*breast_cancer_valid.csv*) y uno de testeo (*breast_cancer_test.csv*).

   (a) Derivar la función de costo de "binary cross-entropy" con regularización L_2 sobre los parámetros w, con hiperparámetro λ, aplicando el principio de máxima verosimilitud sobre el conjunto de entrenamiento.

   (b) Implementar los siguientes clasificadores, y para cada uno reportar las siguientes métricas de performance: matriz de confusión, accuracy, precision, recall, el gráfico de la curva ROC y área bajo la curva ROC (AUC-ROC), sobre el conjunto de validación:

      i. Linear Discriminant Analysis (LDA).
      ii. K-nearest neighbours (KNN), donde el hiperparámetro k es ajustado evaluando el AUC-ROC sobre el conjunto de validación.
      iii. Regresión logística con regularización L_2, donde el hiperparámetro λ es ajustado evaluando el AUC-ROC sobre el conjunto de validación.

      NOTA: en este ejercicio se usa el AUC-ROC como métrica de performance para ajustar los hiperparámetros, pero bien se podría haber usado alguna otra métrica de performance. En general, se deberá usar la métrica que mejor cuantifique la "calidad" de un modelo, lo cual puede variar de un problema a otro, y puede ser una cuestión debatible.

   (c) Una vez que haya desarrollado todos los modelos y este satisfecho con la performance de cada uno, evalúe las métricas de performance antes mencionadas para cada uno de los clasificadores desarrollados, mediante validación cruzada con 5 folds (en este caso, se deben juntar los conjuntos de entrenamiento y validación, y hacer validación cruzada sobre estos) y mediante evaluación sobre el test set.

      Comparar los resultados con las métricas reportadas en el inciso anterior, y analizar cualquier diferencia que se observe. En el caso de las métricas escalares, arme una tabla que muestre las métricas evaluadas de las tres maneras distintas (validación, test y validación cruzada). En el caso de las curvas AUC-ROC, hacer un gráfico mostrando las distintas curvas.

2. **Detección de Fraude en Transacciones de Tarjetas de Crédito.** El conjunto de datos de este problema comprende transacciones realizadas mediante tarjetas de crédito en un lapso de dos días, contabilizando 492 casos de fraude entre un total de 284,807 transacciones. Este dataset presenta un desequilibrio considerable, ya que la clase positiva (fraude) representa solo el 0.17 % del total de transacciones. Para una descripción más detallada del dataset, consulte el archivo *credit_card_description.md*.

   En este problema, nuestro objetivo es desarrollar varios modelos para clasificar una transacción como fraudulenta o no, y luego evaluar la eficacia de cada uno. Para esto, el conjunto de datos se dividió previamente en uno de entrenamiento (*credit_card_train.csv*), uno de validación (*credit_card_valid.csv*) y uno de testeo (*credit_card_test.csv*). Estos conjuntos permanecerán fijos durante el desarrollo de los modelos (es decir, en este problema tampoco aplicaremos validación cruzada). En caso de que sea necesario ajustar un hiperparámetro, esto se hará evaluando la métrica de performance sobre el conjunto de validación.

   (a) Implementar los siguientes clasificadores sobre el conjunto de datos de entrenamiento sin aplicar ninguna técnica de re-balanceo, y para cada uno reportar la matriz de confusión, accuracy, precision, recall, curva ROC y área bajo la curva ROC (AUC-ROC), curva PRC (Precision-Recall Curve) y área bajo la curva PRC (AU-PRC) sobre el conjunto de validación:

      i. Red neuronal densamente conectada con una capa oculta de 16 nodos y función de activación sigmoide. Entrenar durante 150 epochs utilizando el optimizador ADAM con un learning rate de 0.001 y un batch size de 2048, y la entropía cruzada binaria como función de costo. Si quiere explorar diferentes configuraciones, y quedarse con la mejor, lo puede hacer.

         Opcional: Implementar la técnica de "Early Stopping" monitoreando AU-PRC (área bajo la curva Precision-Recall) con una paciencia de 10 epochs.

      ii. Bosque aleatorio (Random Forest) con 20 árboles, utilizando la entropía como criterio de división, estableciendo una profundidad máxima de 10 para cada árbol. Si quiere explorar diferentes configuraciones, y quedarse con la mejor, lo puede hacer.

   (b) Volver a entrenar ambos modelos (red neuronal y bosque aleatorio), aplicando cada una de las siguientes técnicas de re-balanceo. Para cada modelo, y cada tecnica de re-balanceo, reportar las métricas de performance mencionadas en el inciso anterior.

      i. Undersampling: eliminar muestras de la clase mayoritaria de manera aleatoria hasta que ambas clases tengan igual proporción.

      ii. Oversampling by duplication: duplicar muestras de la clase minoritaria de manera aleatoria, hasta que que ambas clases tengan igual proporción.

      iii. Cost re-weighting: en la función de costo, multiplicar los terminos que dependen de las muestras de la clase minoritaria por un factor $C = \frac{\pi_2}{\pi_1}$.

iv. SMOTE (Synthetic Minority Oversampling Technique): hasta que ambas clases tengan igual proporción.

Opcional: Explorar distintas configuraciones de red neuronal (diferente número de capas, unidades ocultas, optimizador, learning rate, batch size) y/o del bosque aleatorio y reportar cual cree que es "el mejor".

(c) Sea un problema de clasifiación binaria, donde el proceso estocástico que genera los datos tiene una distribución a posteriori $P(C_j/x)$ tal que para N suficientemente grande las clases no son linealmente separables en el espacio de features (es decir, las muestras de las dos clases tienen algun grado de solapamiento cuando son proyectadas sobre el espacio de features). Sean $\pi_1 = P(C_1)$ y $\pi_2 = P(C_2)$ las probabilidades marginales de que el proceso genere una muestra de la clase 1 y la clase 2, respectivamente, explicar por qué si utilizamos regresión logística binaria, sin aplicar ninguna técnica de re-balanceo, a medida que π_1 tiende a 0 y con N suficientemente grande, el accuracy tiende a 1 mientras que el precision tiende a 0.

NOTA: Este resultado es general, y aplica a cualquier clasificador "no-sezgado" (como los que vimos en clase), no solamente a regresión logística.

3. **Evaluación de Aceptación de Vehículos.** Usted trabaja en el equipo de Machine Learning de una empresa que comercializa vehículos online, y se le ha pedido que desarrolle un modelo para predecir el grado de aceptabilidad de vehículos por parte de los clientes.

   Para ello le ha sido dado un conjunto de datos recolectados durante 2 años de operación de la empresa, en donde se reporta el precio de venta, costo de mantenimiento, número de puertas, capacidad de pasajeros, tamaño del baúl y seguridad estimada de distinos vehículos, así como el grado de aceptación de estos vehículos por parte de los clientes. La aceptación de un vehículo puede interpretarse como la medida en que este cumple con las necesidades, preferencias y requisitos del cliente y se reporta en cuatro categorías: "inaceptable", "aceptable", "buena" y "muy buena". El conjunto de datos se dividió previamente en uno de entrenamiento (*car_train.csv*), uno de validación (*car_valid.csv*) y uno de testeo (*car_test.csv*).

   Desarrollar al menos dos modelos predictivos distintos que estimen el nivel de aceptación de un vehículo, en base a sus atributos. Explicar cual de todos los modelos desarrollados se debe poner en producción, teniendo en cuenta que lo que más le interesa a la empresa es identifcar bien los vehiculos que son "inaceptables", así como los que son "muy buenos", ya que los primeros representan un cliente insastisfecho, y los segundos son los vehículos que se quieren promocionar más.

   Describa claramente cuál fue la estrategia y el proceso que llevó a cabo para llegar al modelo que enviará a producción (o sea, el que usted considera "el mejor" modelo de todos).

   NOTA: cuando se dice "al menos dos modelos predictivos distintos" esto quiere decir dos modelos con arquitecuras distintas. Por ejemplo, LDA y regresion logística son dos arquitecturas distintas, porque por su estructura de ecuaciones parametrizan el espacio

de clasificación de manera diferente, mientras que regresión logistica y red neuronal con activación de salida sigmoide son arquitecturas de alguna manera iguales, ya que la primera es un caso particular de la segunda (regresión logística es esencialmente una red neuronal sin capas ocultas y activación sigmoide). La idea es que generen una diversidad de modelos, y quedarse con el que mejor capacidad predictiva tiene para ese problema en particular.