PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
Facultad de Ingeniería
IIC2613 Inteligencia Artificial
2022-1

Tarea 3

Aspectos generales

Formato y plazo de entrega

El formato de entrega son *jupyter notebooks* (.ipynb)[1] en los directorios de cada pregunta. El lugar de entrega es en el repositorio de la tarea, en la *branch* por defecto, hasta el **lunes 13 de junio a las 23:59 hrs.** Para crear tu repositorio, debes entrar en el enlace del anuncio de publicación de la tarea en Canvas.

Replicabilidad de resultados

Muchas funciones de sklearn tienen componentes aleatorias, lo que hace que en cada ejecución puedas obtener resultados distintos. Para evitar esto, usaremos una *seed* de `numpy` igual a tu número de alumno[2]. Si tu número de alumno fuera `121212` la celda del *notebook* debería quedar de la siguiente manera:

```
# Establece una semilla para resultados replicables
import numpy as np
np.random.seed(121212)
```

Integridad Académica

Este curso se adhiere al Código de Honor establecido por la universidad, el cual tienes el deber de conocer como estudiante. Todo el trabajo hecho en esta tarea debe ser **totalmente individual**. La idea es que te des el tiempo de aprender estos conceptos fundamentales, tanto para el curso, como para tu formación profesional. Las dudas se deben hacer exclusivamente al cuerpo docente a través de las *issues* en GitHub.

Por otra parte, sabemos que estás utilizando material hecho por otras personas, por lo que es importante reconocerlo de la forma apropiada. Todo lo que obtengas de internet debes citarlo de forma correcta (ya sea en APA, ICONTEC o IEEE). Cualquier falta a la ética y/o a la integridad académica será sancionada con la reprobación del curso y los antecedentes serán entregados a la Dirección de Pregrado.

Comentarios adicionales

El objetivo de esta tarea es que comprendan los conceptos básicos de *Machine Learning* y puedan aplicarlos en actividades sencillas. Es fundamental que pongan énfasis en las explicaciones y justificaciones de sus respuestas, cuidando la redacción y ortografía; manteniendo el código ordenado y comentado. Respuestas que solo presenten resultados o código no serán consideradas, mientras que tareas desordenadas pueden ser objeto de descuentos.

NOTA: Debes presentar un README explicando los detalles de tu entrega y todas las librerías que hayas usado. Adicionalmente, **todas las celdas del notebook deben estar ejecutadas**

[1] El nombre del archivo es irrelevante.
[2] Si tu número de alumno termina en J, reemplázala por un 0

1. Árboles de Decisión y Random Forest (Total: 3 pts.)

Para esta parte de la tarea, deberás trabajar con el set de datos `bd_titanic.csv`[3], el cual contiene información respecto a varios de los pasajeros del RMS Titanic, un barco británico que se hundió en su viaje inaugural en 1912 tras colisionar contra un iceberg en el océano atlántico. Entre todos estos datos se encuentran, por ejemplo, el sexo, edad, tipo de ticket comprado, entre otros datos, además de indicar si sobrevivieron o no. Específicamente, el set de datos contiene las siguientes columnas:

- `PassengerId`: Corresponde a un identificador del pasajero.
- `Name`: Corresponde al nombre del pasajero.
- `Ticket`: Corresponde al número de ticket del pasajero.
- `Embarked`: Corresponde al puerto donde embarcó el pasajero. El valor C representa al puerto de Cherbourg, Q al de Queenstown y S al de Southampton.
- `Pclass`: Corresponde a la clase del ticket del pasajero y toma los valores 1, 2 y 3 para primera, segunda y tercera clase, respectivamente.
- `Sex`: Corresponde al sexo del pasajero y puede tomar los valores *male* o *female*.
- `Age`: Corresponde a la edad del pasajero.
- `SibSp`: Corresponde a la cantidad de hermanos(as) y/o parejas del pasajero que iban en el viaje.
- `Parch`: Corresponde a la cantidad de padres y/o hijos(as) del pasajero que iban en el viaje.
- `Fare`: Corresponde a la tarifa del ticket pagada por el pasajero.
- `Survived`: Corresponde a una variable que indica si el pasajero sobrevivió (1) o murió (0).

Tu labor será cargar, analizar, visualizar y procesar este set de datos, de forma que puedas obtener la mayor cantidad de información posible que te ayude a comprender el problema. Luego, deberás utilizar distintos clasificadores simples o ensamblados para poder predecir si un pasajero sobreviviría o no en base a sus características. En particular se usarán Árboles de Decisión y Random Forest.

Actividad 1: Limpieza del set de datos (0.4 pts.)

En un contexto real, es usual tener muchos datos desordenados, información corrupta, redundante o innecesaria. Para esta actividad, deberás cargar el *dataset* y entregar una nueva versión de él, la cual esté limpia y/o trabajada respecto a lo recién mencionado. Es fundamental que expliques y justifiques todas las decisiones que tomes.

Actividad 2: Comprensión de los datos (0.4 pts.)

En cualquier análisis de datos se busca formular una representación matemática o computacional lo más fiel posible al problema. Es por esto que es importante no sesgarse a la hora de desarrollar una solución, evitando caer en una respuesta auto-inducida. Sin embargo, es muy útil interiorizarse con el problema y los datos, buscando posibles relaciones causales y proponiendo distintas hipótesis que se comprueben/-descarten a lo largo del estudio.

[3] `https://www.kaggle.com/c/titanic`

Para esta actividad deberás estudiar los datos y formular al menos 3 hipótesis respecto a los resultados que esperas encontrar o que ya puedes ver, como relaciones entre variables, qué información será más determinista para la sobrevivencia de un pasajero, etc. Se espera que estas hipótesis estén fundamentadas en base al contexto del problema y/o apoyadas con visualizaciones gráficas.

Actividad 3: Visualización de los datos (0.4 pts.)

Ahora que ya tienes tu *dataset* limpio, deberás presentar distintas visualizaciones que permitan observar información valiosa de los datos. Puedes usar cualquier tipo de gráfico y librería, siempre y cuando aporten algo de valor al estudio. Se espera que muestres al menos 4 gráficos y que expliques el motivo por el que consideras relevante esa visualización.

Actividad 4: Pre-procesamiento de los datos (0.4 pts.)

Para esta actividad debes terminar de pre-procesar los datos, de forma que sean aptos para utilizar por los clasificadores. En concreto, se espera que discretices las *features* que sean continuas, además de representar de forma numérica los valores que actualmente no lo sean y todo lo que consideres necesario. Por último, deberás dividir tus datos en sets de entrenamiento y test y responder las siguientes preguntas:

- ¿Qué proporciones usaste para cada set? ¿Por qué?
- ¿Cuántas instancias para cada categoría de `Survived` hay en cada set? ¿Están aceptablemente balanceadas? ¿Qué riesgos existen si es que están demasiado desbalanceadas?

Actividad 5: Entrenando un árbol de decisión (0.4 pts.)

Utilizando la librería sklearn, entrena un árbol de decisión que sea capaz de predecir si un pasajero sobreviviría o no al accidente en base a sus características (las que hayas determinado relevantes) y realiza las clasificaciones sobre el set de test. Para que se te considere la respuesta, deberás encontrar una combinación de hiper-parámetros tal que obtengas un *accuracy* mayor o igual a 65 % en el set de test.

Actividad 6: Ensamblaje personalizado básico (0.6 pts.)

Un ensamblaje es, en palabras sencillas, un modelo que se compone de otros modelos para funcionar. En el caso de Random Forest, este utiliza varios árboles de decisión e implementa un sistema de votación para llegar a la mejor decisión en base a la filosofía de "la sabiduría de la mayoría".

Para esta actividad, deberás construir un Random Forest personalizado, el cual sirva para el *dataset* que limpiaste anteriormente. Este deberá estar conformado por 3 árboles de decisión de la siguiente manera:

- Crea la clase `CustomRandomForest` con sus métodos `fit` y `predict`.
- El método `fit` recibe un *dataset* X con los datos para entrenar el modelo y una serie de datos y con las etiquetas de las instancias en X.

- El método `fit` debe entrenar un árbol A_1 con todos los datos en X y un árbol A_2 que haga lo mismo, pero que se entrene únicamente con las instancias de X en las que falla A_1 al clasificar. Por último, se debe entrenar un árbol A_3 que se entrena con las instancias de X en las que falla A_1 y A_2 al clasificar.
- El método `predict` recibe un set de datos X del mismo tipo que en `train` y debe retornar una serie (del mismo tipo que y en `train`) con los valores asociados a la categoría asignada por el ensamblaje para cada una de las instancias recibidas.
- Investiga sobre el método `predict_proba` de los árboles de decisión[4].
- Para clasificar cada instancia entregada, calcula la probabilidad que asignan los tres árboles a cada una de las dos categorías posibles. Luego, pondera estas probabilidades P_{A_1}, P_{A_2} y P_{A_3} por unos pesos w_1, w_2 y w_3, tales que $\sum_{i=1}^{3} w_i = 1$. Esto significa que la probabilidad para cada categoría debería quedar de la siguiente forma:

  $$P_{A_1} \cdot w_1 + P_{A_2} \cdot w_2 + P_{A_3} \cdot w_3$$

  La categoría que tenga mayor probabilidad ponderada será la que el ensamble deba asignar a la instancia. En caso de haber empate, puedeS escoger cualquiera.

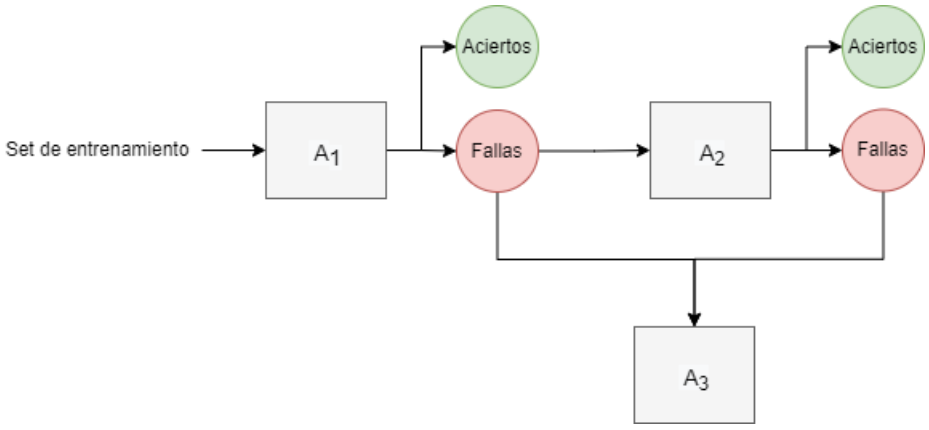

Figura 1: Arquitectura del ensamblaje personalizado

Entrena tu modelo con el set de entrenamiento y evalúalo con el set de test. Para que se te considere la respuesta, deberás encontrar una combinación de hiper-parámetros para los árboles internos tal que obtengas un *accuracy* mayor o igual a 65 % en el set de test.

Actividad 7: Entrenando un Random Forest (0.4 pts.)

Utiliza el modelo Random Forest que provee sklearn, entrénalo con el set de entrenamiento y evalúalo con el set de test. Para que se te considere la respuesta, deberás encontrar una combinación de hiper-parámetros para los árboles internos tal que obtengas un *accuracy* mayor o igual a 65 % en el set de test.

[4]No hace falta describirlo, es solo porque lo necesitarás

Bonus (0.3 pts.)

En clases/ayudantías hemos destacado la fácil interpretabilidad y comprensión de los árboles de decisión como una de sus ventajas, pero lo importante es cómo se lleva esto a la práctica. Para este bonus deberás:

1. Investigar cómo graficar tu árbol de decisión entrenado en la actividad 5.
2. Inventar 2 pasajeros imaginarios, con sus respectivas características. Deberás traspasar estas características a sus *features* representadas numéricamente según el trabajo que hayas hecho al *dataset*.
3. En base a sus características, sigue y explica el camino que van siguiendo a través de las ramas que correspondan según el gráfico de tu árbol y determina cuál es la clasificación asignada. Puedes también añadir una imagen que muestre más claramente la ruta.
4. Entrega los mismos 2 pasajeros imaginarios a tu árbol de decisión de la actividad 5 y verifica que el resultado coincida con los del punto anterior.

2. SVM (Total: 3 pts.)

Buscando nuevos horizontes, vas a Estados Unidos, donde el director de un periódico local te reconoce por tus aportes en el campo de la Inteligencia Artificial. Te comenta que, producto del descuido de uno de sus trabajadores, se han mezclado casi todas sus carpetas con los artículos de diferentes secciones de la compañía y que necesita ayuda lo más urgente posible para poder organizar nuevamente los datos en base a su contenido, pues no contaban con una convención de nombres para los archivos.

El día que se mezclaron todos los documentos, uno de los computadores estaba sin internet, por lo que pudo conservar varios de estos artículos aún estructurados y ordenados. Por lo tanto, decides usar estos datos para entrenar un SVM que luego te permita clasificar las demás noticias. Esta información te la entregan en el archivo `bd_noticias.csv`, el cual contiene las siguientes columnas:

- `news_headline`: Corresponde al titular de la noticia.
- `news_article`: Corresponde al cuerpo de la noticia.
- `news_category`: Corresponde a la categoría/sección de la noticia. Puede tomar los valores *automobile* (automóviles), *science* (ciencia), *sports* (deporte) y *world* (noticias de carácter mundial).

Actividad 1: IA y el procesamiento del lenguaje natural (0.4 pts.)

Comenta al menos 3 tipos de dificultades que tiene el trabajo con lenguaje natural respecto a los datos tradicionales. Menciona un ejemplo otro tipo de datos/información que tengan dificultades similares a las del lenguaje natural y pueda ser interesante para el campo de la IA.

Actividad 2: Quitando el ruido del lenguaje natural (0.5 pts.)

Utilizando la librería spaCy[5] y su modelo pre-entrenado `en_core_web_md`, crea la función `clean_text`, la cual recibe un *string* con un texto en inglés de largo indeterminado y retorna un texto procesado, con las palabras lematizadas, sin *stop-words* y removiendo la puntuación. Puedes usar otras técnicas adicionales

[5]Recuerda ver las cápsulas subidas para familiarizarte con su uso

que consideres útiles para tu trabajo, pero recuerda comentarlas.

Adicionalmente:

- Comenta la importancia de utilizar todas las técnicas usadas para trabajar con lenguaje natural de forma generalizada. Se recomienda usar ejemplos (con palabras).
- Entrega al menos 3 textos de ejemplo creados por ti (recuerda que deben estar en inglés) a la función `clean_text` y comenta brevemente el *output* retornado.

Actividad 3: Representación numérica del lenguaje natural (0.7 pts.)

Para esta actividad deberás hacer un pre-procesamiento del *dataset* entregado en `bd_noticias.csv`, de forma que sea manejable por el modelo clasificador. En particular:

1. Realiza un análisis básico de los datos, decidiendo si vas a usar una, ambas o alguna combinación de las columnas con texto, justificando en todo momento tus decisiones.
2. Usa tu función `clean_text` para procesar y limpiar el texto con el que vayas a trabajar.
3. Codifica a valores numéricos las etiquetas correspondientes a la categoría de las noticias[6]
4. Utilizando las funciones provistas por spaCy, crea un nuevo set de datos que tenga la representación vectorial (*doc2vec*) de los textos. La dimensión del vector queda a elección tuya.
5. Divide el set de datos creado en el paso anterior en sets de entrenamiento y test con sus respectivas etiquetas.

Una vez hecho lo anterior, investiga y comenta en palabras sencillas lo que representan estos vectores. Recuerda citar tus fuentes de forma visible.

Actividad 4: Entrenando un SVM clasificador de noticias (0.5 pts.)

Usando la librería sklearn, crea tu propio clasificador SVM y entrénalo con el set de entrenamiento generado en la actividad anterior. Deberás hacer *cross validation* para validar los mejores hiper-parámetros que encuentres. Para que tu respuesta sea considerada, deberás obtener un *accuracy* medio de al menos 70 % al hacer *cross validation* sobre el set de entrenamiento y un rendimiento mínimo de 70 % de *accuracy* sobre el set de test.

Actividad 5: Análisis de resultados (0.5 pts.)

Calcula y grafica la matriz de confusión del clasificador sobre el set de test. Responde las siguientes preguntas:

- ¿Cuál(es) son las categorías mejor clasificadas? ¿Qué crees que podría explicar ese comportamiento?
- ¿Cuál(es) son las categorías peor clasificadas? ¿Qué crees que podría explicar ese comportamiento?
- ¿Cuáles son las categorías que tienen mayor solapamiento, es decir, se confunden más entre sí? ¿A qué se podría deber ese comportamiento?

[6] La funcionalidad `LabelEncoder` de sklearn te puede ser de utilidad.

Actividad 6: Probando con ejemplos propios (0.4 pts.)

Para esta actividad deberás hacer lo siguiente:

1. Escribe en inglés al menos dos textos que estén intencionalmente asociados a una de las categorías de noticias. Por ejemplo, *"The Barcelona star, Lionel Messi, scored 3 goals in the grand final against Colo-Colo."* estaría asociado a la clase de deportes (`sports`).
2. Hazles el procesamiento correspondiente con la función `clean_text`.
3. Usando spaCy, obtén la representación vectorial para los texto procesados.
4. Ingresa los vectores a tu clasificador ya entrenado para obtener una clasificación de categoría y comenta si coinciden o no con los resultados esperados.

Bonus (0.25 pts.)

Con el paso del tiempo te has aburrido de usar un método tan simple de representación numérica de texto y buscas el estado del arte en esta materia. En tu búsqueda, descubres los métodos con atención y, en particular, te topas con los denominados *transformers* de texto.

- Explica, en menos de media página, en qué consiste un *transformer* o modelo con atención. No es necesario entrar en muchos detalles técnicos.

Referencias

[1] Kaggle. (2022). *Titanic - Machine Learning from Disaster* [Fichero de datos]. Recuperado el 5 de mayo del 2022 en: https://www.kaggle.com/competitions/titanic/overview

[2] Yadav K. (21 de enero del 2021). *News Classification* [Fichero de datos]. Recuperado el 5 de mayo del 2022 en: https://www.kaggle.com/datasets/kishanyadav/inshort-news