PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
Facultad de Ingeniería
IIC2613 Inteligencia Artificial
2024-1

Tarea 4
Aprendizaje de Máquina

Fecha de entrega: Jueves 13 de junio a las 23:59 hrs

Aspectos generales

Formato y plazo de entrega

El formato de entrega es **únicamente** un archivo con extensión `.ipynb` para tanto las respuestas de código como teóricas. El lugar de entrega es en el repositorio de la tarea, en la branch por defecto, hasta el jueves 13 de junio a las 23:59 hrs. Para crear tu repositorio, debes entrar en el enlace del anuncio de la tarea en Canvas. Por último, recuerda que los cupones de atraso son días **reales** (hábiles o no) extra.

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 puedan utilizar distintos algoritmos para resolver problemas de clasificación y regresión, aplicándolos en problemas donde pueden ser de gran utilidad. Es fundamental que pongan énfasis en las justificaciones de sus respuestas, cuidando la redacción, ortografía; manteniendo el código ordenado y comentado. Aquellas respuestas que solo presenten resultados o código (sin contexto ni comentarios) no serán consideradas, mientras que tareas desordenadas pueden ser objeto de descuentos.

1. Reflexión (2 pts.)

Esta sección de la tarea consiste en una serie de preguntas de reflexión que tendrás que responder basándote en el podcast de la profesora Jocelyn Dunstan "Ciencia de Datos con Jocelyn Dunstan" que está disponible en Spotify. Escucha uno a mas capítulos del podcast y escoge un episodio para contestar las siguientes preguntas:

1.1. (0.4 ptos.) ¿Por qué escogiste este episodio?

1.2. (0.8 ptos.) ¿Cuál crees que es la relevancia del tema tratado en el capítulo dentro del campo de la inteligencia artificial, machine learning o ciencia de datos?

1.3. (0.8 ptos.) Busca al menos una referencia externa que enriquezca alguna idea mencionada en el capítulo y discute.

2. DCClasificador de Sismos (4 pts.)

El objetivo de esta actividad es entrenar algoritmos de clasificación y regresión para estimar la posibilidad de tsunami y magnitud en grados richter de un sismo dada informacion sobre este.

Para llevar esto a cabo se dispone de una base de datos que contiene todos los sismos entre los años 1990 y 2023 (mas de 3.5M de muestras), de la cual se ha tomado un segmento de 10.000 muestras para poder implementar algoritmos de aprendizaje supervisado. El set de datos original se encuentra disponible en Kaggle, junto con el detalle sobre sus contenidos.

Las preguntas del enunciado también se encuentran disponibles en el Notebook de la entrega, siendo este el lugar donde las respuestas deberán ir incluidas (celdas de Markdown para respuestas teóricas y de Python para implementaciones en código)

2.1. Lectura y estudio de los datos

1. Utilizando pandas, lee la información contenida en el archivo `earthquakes.csv`. Una vez que la información esté cargada, comenta sobre los elementos que contiene. Para esto debes investigar la fuente de los datos en este link y visualizar los datos con pandas para su entendimiento.

2. Indica cuál es el atributo que se utilizará como etiqueta para entrenar los modelos. Utiliza las funciones `describe()` y `count()` para realizar un análisis inicial del DataFrame. Indica la cantidad de datos con los que se está trabajando, el tipo de datos presentes y cualquier otra información que consideres relevante. Una de las actividades en esta tarea consiste en predecir la posibilidad de un tsunami dada el resto de la información recompilada de un sismo, indica que columna usaremos de etiqueta y por qué es importante mantenerla oculta al momento de testear la clasificación.

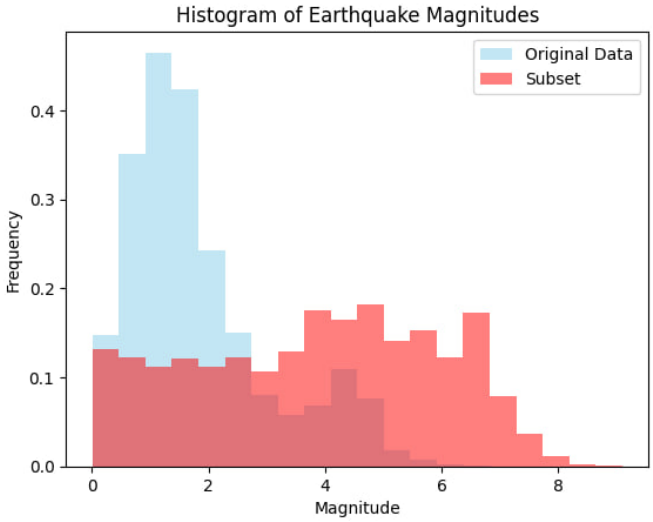
3. Investiga sobre qué es el balance/desbalance de clases, cómo puede afectar este a un algoritmo clasificador e indica alguna técnica para contrarrestar el desbalance en problemas de clasificación. Finalmente, indica la distribución de datos por clase, si están balanceados o no y cómo esto podría beneficiar/afectar al rendimiento de los modelos que entrenaremos en esta tarea.

2.2. Preprocesamiento de los datos

En esta sección, deberás llevar a cabo un tratamiento de los datos para garantizar que los clasificadores puedan ser entrenados de manera adecuada. Es importante resaltar que todas las decisiones que tomes deberán estar debidamente justificadas. Para comenzar responde a las preguntas: ¿Por qué es importante preprocesar los datos y cómo puede esto afectar a nuestro problema de clasificación?

1. Los datos cargados corresponden a un segmento de un set de datos mas grande sobre terremotos a lo largo del mundo, el set de datos original tiene 3.5 millones de filas pero solamente unas 10.000 fueron extraídas, guardando todas las muestras con tsunami (debido a lo poco frecuente de la

clase) y tratando de balancear la magnitud de un terremoto para no tener sobrerepresentacion de algunos valores. A continuación se muestra un gráfico de la distribución (normalizada) de la columna `magnitude` antes y después de muestrear el set original:



En caso de que hubiesemos dispuesto de pocos datos originalmente ¿qué técnicas podrías usar para enriquecer el dataset? Investiga en internet y reporta al menos dos metodologias.

2. Revisa si el nuevo DataFrame contiene valores nulos (NaN). En caso afirmativo, deberás tomar la decisión de eliminar las filas o columnas que contengan estos datos nulos, o si prefieres, reemplazar estos con algún otro valor. Si no existen valores nulos, indícalo. ¿Por qué es importante manejar este tipo de valores?.

3. ¿Qué son las variables categóricas? ¿Hay este tipo de variables en este conjunto de datos? Explica cómo se pueden manejar estas y en caso de haberlas realiza un desarrollo para poder entrenar los algoritmos correctamente.

   Se recomienda **fuertemente** revisar el siguiente link sobre métodos de encoding.

4. Elimina las columnas que no consideres necesarias o que creas que no aportan información relevante para realizar la clasificación. Debes justificar por qué eliminas cada columna. Responde a la pregunta ¿Cómo pueden afectar columnas con información no relevante al modelo de clasificación?

   Se recomienda **fuertemente** usar la función `pairplot()` de Seaborn para visualizar la correlación entre características, aunque recuerda que **correlación no siempre implica causalidad**.

5. Nos importa que tanto nuestro set de entrenamiento como nuestro set de testeo sea igual de representativos de la distribución de los datos, por lo que tomaremos en consideración esto a la hora de separar los datos en train/test. Afortunadamente, la funcion `train_test_split()` de SciKit Learn nos permite llevar esto a cabo mediante su atributo `stratify`. Usaremos un 20 % de los datos para evaluar nuestros modelos.

   En esta sección solo debes reemplazar la variable `df` con cual sea en la que estés almacenando los datos.

6. Como fue comentado anteriormente, dentro de esta tarea llevaremos a cabo dos tareas sobre el set de datos, clasificación de tsunamis y regresión de la magnitud de un sismo utilizando información externa sobre este. Por tanto, deberás separar las columnas `tsunami` y `magnitude` del set de datos para utilizarlas como vectores de etiquetas $\vec{y}$ para tus clasificadores.

Dentro de este apartado se espera que tengas 2 matrices y 4 vectores:

- `X_train` y `X_test`: Matrices que contienen la información sobre cada sismo (excepto las columnas `tsunami` y `magnitude`).
- `y_train_tsunami` y `y_test_tsunami`: Vectores de entrenamiento y evaluacion que contienen un 0 para sismos sin tsunamis y un 1 para aquellos que si.
- `y_train_magnitude` y `y_test_magnitude`: Vectores de entrenamiento y evaluacion que contienen la magnitud para cada sismo dentro de `X_train` y `X_test` respectivamente.

7. ¿Qué es la normalización de datos y por qué puede ser beneficioso realizarla para un problema de clasificación? ¿Existe alguna columna que pueda ser conveniente normalizar? En caso de haberla utiliza algún algoritmo de normalización, si no justifica por qué no es conveniente realizar dicha operación.

   Es importante que la obtencion de parametros para normalizacion de los datos se lleva a cabo en el conjunto de entrenamiento y luego se aplica sobre el de evaluacion, si utilizas `StandardScaler` de sklearn, deberas hacer ajuste del scaler (usando `.fit()`) sobre la columna en el conjunto de entrenamiento y luego hacer `.transform()` sobre ambos conjuntos.

2.3. Explorando los algoritmos de un problema de clasificación

Responde las siguientes preguntas de análisis para un problema de clasificación y los algoritmos que se utilizan para estos:

1. ¿Cuál es el principal objetivo de un problema de clasificación?
2. ¿Qué casos de uso comunes de clasificación existen en el mundo real? ¿Qué impacto pueden tener estos en las personas? Menciona al menos 3.
3. ¿Qué relevancia e impacto puede tener el éxito o fracaso de este problema de clasificación en la vida real? ¿De qué modo un algoritmo de clasificación como este podría beneficiar a la sociedad chilena para prepararse ante desastres naturales?
4. ¿Cuáles son los factores que deberían influir en la elección de un algoritmo de clasificación para un conjunto de datos determinado?
5. ¿Qué son los hiperparámetros y cómo estos podrían afectar al rendimiento de un modelo de clasificación?

2.4. Implementación de Clasificadores

Utilizando el DataFrame con los datos ya preprocesados, deberás entrenar cuatro modelos distintos para clasificar si los sismos pueden generar tsunamis o no. Para esto, se deben entrenar los clasificadores de KNN, árboles de decisión, Random Forest y SVM.

2.5. Cálculo de métricas

1. Genera una tabla comparativa en la que se indiquen las métricas de accuracy, precisión y recall para cada uno de los modelos.
2. Investiga qué es la métrica F1-score y calcúlala a partir de los resultados, no puedes usar implementaciones ya existentes para calcularlo. Luego argumenta, ¿qué mide el score F1 en problemas de clasificacion binaria?
3. Grafica utilizando la librería `matplotlib` el F1-score para cada modelo.

4. Señala cuál de los modelos obtuvo las mejores métricas y rendimiento.

5. Para el algoritmo con los mejores resultados elige 2 hiperparámetros y entrena 3 modelos variando estos, con este el mismo algoritmo, realiza predicciones con este y luego evalúalos. ¿Qué combinación de hiperparámetros es la con mejores resultados?

6. Para el modelo de mejor desempeño del apartado anterior calcula su matriz de confusion utilizando el metodo de sklearn `confusion_matrix()` y visualizala utilizando `ConfusionMatrixDisplay`. Luego comenta, ¿en qué tiende a equivocarse el modelo entrenado?

2.6. Análisis de desempeño y resultados

1. Explica por qué crees que algunos algoritmos tuvieron mejores resultados que otros para este set de datos. Basa tu respuesta en material bibliográfico, ¿significa eso que algunos algoritmos tienen siempre mejores resultados que otros?

2. ¿Qué medidas se podrían tomar para mejorar el rendimiento de los modelos?

3. ¿Por qué son tan importantes las métricas estadísticas para los modelos y/o clasificadores realizados anteriormente?

4. ¿Qué utilidad le vas al uso de técnicas como esta en la vida real para tener un impacto positivo en la sociedad? ¿Qué tipo de decisiones se podrían tomar o qué medidas se podrían implementar en base a las predicciones de un algoritmo como este?

2.7. Implementar modelos de regresión

Utilizando el DataFrame con los datos ya preprocesados, entrena cuatro regresores distintos para predecir la magnitud de los sismos a partir de otras características. Con los datos de entrenamiento, realiza una predicción de regresión con los modelos de KNN, árboles de decisión, Random Forest y SVM.

2.8. Análisis de regresión

1. Para esta parte debes generar un gráfico en el cual se indiquen los RMSE para cada regresión.

2. ¿Qué significa tener un mayor o menor RMSE y que implicancias tienen en la clasificación y sus resultados?

3. Al realizar la comparación entre cada modelo de regresión y su RMSE, detalle la razón de la obtención de estos resultados específicos, el porque un modelo de regresión es mejor que otro dada su construcción y explicar en tus propias palabras el funcionamiento de estos modelos de regresión.

Archivos entregados

- `earthquakes.csv`: Archivo .csv que contiene una muestra del set de datos de terremotos de Kaggle y sobre el cual se trabajara a lo largo de esta tarea.

- `T4.ipynb`: Notebook sobre el cual debera trabajarse para la tarea, en el deberan ir las respuestas tanto teoricas como a nivel de codigo. Debe ser entregado con todas sus celdas corridas.