Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
Computación: Ciencia y Tecnología del Mundo Digital

Tarea Grande 2

Machine Learning

Ayudantes: Felipe García, Astrid San Martín
Profesor Denis Parra

Anunciada: 20 de mayo de 2021

Indicaciones

- Fecha de Entrega: 10 de junio de 2021, 20:00 hrs.
- La entrega de la tarea debe realizarse en el repositorio privado GitHub asignado para esta evaluación.
- El descuento por cada hora o fracción de atraso es 1.5 puntos de la nota final.
- Esta tarea debe realizarse de máximo 2 personas. En caso de copia la tarea será evaluada con nota 1.0 junto con las sanciones disciplinarias correspondientes.
- Las funciones de librería utilizadas deberán ser todas oportunamente explicadas, junto con sus argumentos. De lo contrario, **no se otorgará el puntaje correspondiente.**
- Al graficar, comente brevemente que realiza cada función ocupada.
  Ejemplo: "print('Hello World') # Imprime el texto dado"

Objetivo

Los objetivos de esta tarea son:

- Familiarizarse, analizar en profundidad y procesar un set de datos.
- Hacer uso de la librería Pandas de Python para el procesamiento de datos.
- Hacer uso de la librería `scikit-learn` de Python.

- Hacer uso de librerías gráficas de Python como `Matplotlib`, `Altair` o `Seaborn`
- Usar modelos de clasificación y regresión (aprendizaje supervisado) y medir su rendimiento.

Descripción de la tarea

Es parte ya de nuestra vida diaria el uso de teléfonos celulares, siendo testigos de cómo han ido cambiando de tamaño y capacidad (ver figura 1), cada vez tienen más y mejores aplicaciones que nos permiten desarrollar tanto nuestra vida personal (a veces no tanto) y profesional de mejor manera.

Figura 1: Celulares entre 1997 y 2003 (`Fuente`)

Se estima que existen actualmente en el mundo más de $7,950,000,000$[1] teléfonos celulares, y creciendo. Esta cantidad ha sido posible debido a la variada oferta de teléfonos celulares, pasando por celulares de gama baja, media y los más deseados de gama alta, con un sin número de características que los hacen ser productos bastante caros.

[1] Bureau, US Census. Çensus Bureau Projects U.S. and World Populations on New Year's Day". The United States Census Bureau. Retrieved March 23, 2020

En esta tarea harán análisis exploratorio de un un set de datos de características técnicas de celulares y su relación con el rango de precio `celulares.csv` que podrán encontrarlo en el repositorio del curso. En este archivo encontrarán información de $2{,}200$ celulares tales como memoria ram, número de núcleos, si tiene o no bluetooth, etc. A su vez, tendrán la oportunidad de explorar dos métodos de aprendizaje supervisado de *machine learning*.

Para desarrollar esta tarea deberán trabajar en grupos de a dos, para lo cual recibirán un enlace de Github Classroom donde el primer integrante del grupo deberá registrar el equipo, y el segundo deberá incorporarse al equipo, es importante que sólo el primer integrante cree el equipo y el repositorio. Deben empezar la tarea buscando a un compañero y formando un grupo. **No se permitirán entregas individuales.** Por cierto, deben considerar que los nombres de los equipos tendrán que ser elegidos acorde a la seriedad de la instancia de evaluación de este curso, como lo es una tarea. Consecuentemente, habrá descuentos por nombres poco adecuados. Se sugiere que el nombre del grupo sean los apellidos de los integrantes.

El formato de entrega de cada una de las partes que consta esta tarea deberá ser en un único Jupyter notebook (`.ipynb`). Asimismo, la entrega deben realizarla en el repositorio asignado a cada pareja. Para trabajar con Jupyter notebook es recomendable que usen Google Colab, así evitaran tener que instalar Jupyter localmente y la instalación de todas las demás librerías a utilizar en esta tarea [2].

Esta tarea está dividida en tres partes, donde las dos primeras constan de trabajo de programación usando las librerías ya mencionadas (Pandas, `scikit-learn` y alguna de visualización como `Matplotlib`, `Altair` o `Seaborn`). La última sección corresponde a un informe donde deberán responder preguntas sobre lo realizado en las primeras partes. Al respecto, deben tomar en cuenta que esta última parte de la tarea es la que posee mayor puntaje por lo cual, es importante que pongan mucha atención a las preguntas planteadas antes de empezar a trabajar, es así como podrán focalizar de manera efectiva el desarrollo de la tarea.

Set de datos

Los datos para realizar la tarea se encuentran en el archivo `celulares.csv`, donde cada fila corresponde a un modelo, con 21 atributos, donde `price_range` e `int_memory` serán las etiquetas para las tareas

[2] Si luego de probar con Colab tuviesen problemas que no le permitieran hacer la tarea, pueden hacerla localmente con Jupyter Notebook, pero recuerde usar Python 3 y también crear un archivo **requirements.txt** con las versiones de las bibliotecas que usó (pandas, scikit-learn, altair y otras)

de clasificación (la variable que hay que predecir en el clasificador) y regresión (la variable que hay que predecir en el regresor), respectivamente. Aquí se describen los atributos por columna:

- `battery_power`: Energía total que la batería puede almacenar en mAh.
- `blue`: Si tiene bluetooth o no.
- `clock_speed`: Velocidad a la cual el microprocesador ejecuta las tareas.
- `dual_sim`: Si tiene dual sim o no.
- `fc`: Cámara frontal mega pixeles.
- `four_g`: Si tiene 4G o no.
- `int_memory`: Memoria interna en Gigabytes.
- `m_dep`: Ancho del celular en cm.
- `mobile_wt`: Peso del celular.
- `n_cores`: Número de núcleos del procesador.
- `pc`: Cámara primaria mega pixeles.
- `px_height`: Altura de resolución en pixeles.
- `px_width`: Ancho de resolución en pixeles.
- `ram`: Random Access Memory en Mega Bytes.
- `sc_h`: Altura de la pantalla en cm.
- `sc_w`: Ancho de la pantalla en cm.
- `talk_time`: Tiempo de duración de la batería en uso.
- `three_g`: Si tiene 3G o no.
- `touch_screen`: Si tiene touch screen o no.
- `wifi`: Si tiene wifi o no.
- `price_range`: Etiqueta con rango de precios 0 bajo costo, 1 de costo medio, 2 de alto costo, 3 de muy alto costo.

Parte 1: Procesamiento de los datos (1,5 ptos)

En esta primera parte deberán enfocarse en leer, analizar y limpiar los datos, es decir, deben familiarizarse con la base de datos. Para ello, es obligatorio hacer uso de la librería Pandas y queda **prohibido utilizar ciclos en esta parte (i.e., *while, for,* etc.)**, pues se espera que el código sea eficiente. El código debe estar comentado en cada paso, explicando lo realizado y las toma de decisiones.

Análisis de las características (0.8): Como se detalló, en la base de datos las características o atributos que tiene dan información sobre las características y precios de distintos modelos de celulares. Por lo tanto, deben analizar la asignación de la clase ó etiqueta y poner atención si algunas de esas características tienen o no influencia en la clase. Es así como, después de cargar los datos, deberán eliminar las columnas que consideren que no aportan información para el objetivo de la tarea. Tengan en consideración que en el informe deberán justificar las decisiones que tomen. Deberán presentar al menos 3 gráficos, considerando tres tipos diferentes (pairplot, swarmplot, histogram, etc.), que caractericen los datos con su explicación correspondiente sobre qué se observa, y qué nos dice esto de los datos.

Limpiar valores nulos (0.5): El siguiente paso que deben realizar es buscar si el dataset contiene valores nulos. Así, usando la librería Pandas, deberán encontrar los valores nulos o `NaN` (*Not a Number*) dentro del DataFrame. Si encuentran estos valores nulos deberán tomar la decisión sobre qué hacer con ellos (imputar un valor, eliminar el dato, etc.) y justificar debidamente en el informe.

Separar la clase y normalizar (0.2): Por último, deberán separar el dataset en una matriz de características (*features*) y un vector de clases para, luego, normalizar las características. Para esta tarea, como ya se mencionó, se esperan UNA separación donde el vector de clase corresponda al rango de precios en el caso del clasificador y memoria interna en el caso del regresor.

De aquí en adelante, será con estos datos ya procesados con los cuales se continuará trabajando, es decir, sin columnas innecesarias, sin datos nulos y con valores normalizados.

Parte 2: Clasificación automática y regresión (2 ptos)

En esta segunda parte de la tarea, deberán hacer uso de un modelo de clasificación y otro de regresión. Será necesario como primer paso entrenar el algoritmo usando un set de entrenamiento, y posteriormente evaluar el desempeño del mismo usando un set de testeo. En el caso de la predicción para el

modelo de clasificación, se entregan los datos a clasificar con la etiqueta de `price_range`, con el objeto de medir qué tan bien resulta la clasificación. Ahora, en el caso de la predicción para el modelo de regresión, se entregan los datos a predecir con la etiqueta de `int_memory`, con el objeto de medir qué tan bien resulta la regresión.
Deben recordar `price_range` es la variable que hay que predecir en el clasificador, `int_memory` es la variable que hay que predecir en el regresor.
Finalmente, se deberá utilizar algunas métricas ó estadísticas de evaluación para chequear cómo lo hizo el modelo entrenado.

Dentro de las herramientas disponibles en el área de *Machine Learning*, existen varios modelos de clasificación. Para esta tarea utilizarán un algoritmo que tuvieron la oportunidad de conocer en la lectura asignada: Naive Bayes (NB)
Como modelo de regresión harán uso de el algoritmo Decision Tree (DT) en su versión de regresión, en este caso sólo deberán reportar el error del modelo.
En esta ocasión, utilizarán la librería `scikit-learn` para trabajar en la definición y entrenamiento del modelo. Una vez más deberán tener presente que el código debe estar comentado en cada paso, explicando lo realizado y las toma de decisiones.

Separar datos de entrenamiento y pruebas (0.3): En orden de preparar los datos para el entrenamiento, deberán separar el set de datos en set de entrenamiento y set de pruebas. Para definir el porcentaje de datos a usar para entrenamiento y testeo, podrán buscar en la literatura en internet las proporciones recomendadas, una vez más deberán reportar la proporción escogida justificando.

Instanciar y entrenar el clasificador (0.5): Ahora con el set de datos de entrenamiento definido, deben instanciar el modelo NB y entrenarlo. Para este algoritmo deben utilizar como etiqueta el atributo de rango de precios `price_range`.

Instanciar y entrenar el regresor (0.5): Asimismo, deberán instanciar el modelo Decisión Tree Regressor y entrenarlo usando el set de entrenamiento. En este caso deberán usar como etiqueta la memoria interna `int_memory`.

Visualizar la importancia de las features del regresor DT Regressor (0.2): Debes realizar un gráfico sobre la importancia de la features en el regresor y comentar lo observado acorde al contexto de los datos.

Calcular el rendimiento del clasificador y regresor (0.5): Una vez entrenado el clasificador (etiqueta `price_range`), corresponde testear el rendimiento de cada clase. Para poder conocer el rendimiento tendrán que utilizar las métricas: *accuracy*, *precision*, *recall* y *f-1 score*. De este modo, deberán hacer la predicción usando el set de prueba o testeo, y comparar las predicciones con las clases reales. Se debe reportar las métricas para cada clase en el set de pruebas.
A su vez, luego de entrenar el regresor (etiqueta `int_memory`), deben reportar una métrica de error i.e. mean square error.

Para un correcto análisis del entrenamiento y rendimiento de los modelos, los pasos señalados anteriormente deberán ser repetidos probando distintos hiper-parámetros (deben mostrar explícitamente en una celda Markdown las combinaciones probadas). El detalle de los hiper-parámetros a modificar pueden encontrarlos en la documentación de `scikit-learn`. Finalmente, deben escoger la combinación de hiper-parámetros que muestre el mejor rendimiento, siendo este set de hiper-parámetros el que deberá quedar en el código. En el informe deberán reportar y explicar las pruebas realizadas, incluyendo la toma de decisiones que se hicieron.

Parte 3: Informe (2,5 ptos)

En base a todo lo realizado hasta ahora, deberán responder algunas preguntas con el objetivo de evaluar los conceptos aplicados en esta tarea. Recuerden poner particular atención a esta parte de la tarea, la parte más importante de tu tarea.

La siguientes preguntas deberán ser incluídas en el archivo `.ipynb`, usando una celda en formato *markdown*. Las respuestas tienen una extensión máxima de 10 líneas. Deberán hacer uso de la letra por defecto en el formato *markdown*, de no cumplir con estas restricciones existirá una penalización en el puntaje.

1. ¿Cómo resolvieron el tema de los valores nulos? Justifique. (0.3)
2. ¿Qué normalizaste, filas o columnas? ¿Por qué? ¿Para qué sirve normalizar los datos? ¿Qué tipo de normalización usaste y por qué? Justifique. (0.3)
3. ¿Qué gráficos utilizaste para caracterizar los datos?¿Por qué? ¿Qué observaste de los datos, encontraste alguna característica particular? Justifique. (0.3)

4. ¿Por qué se separan los datos en set de entrenamiento y set de pruebas? ¿Qué proporción de los datos utilizaste para cada uno y por qué? Justifique. (0.3)

5. ¿Qué hiper-parámetros modificaste para probar tu clasificador? ¿Y el regresor? ¿Cuáles combinaciones de parámetros te dieron mejores resultados y por qué crees que es así? Justifique. (0.3)

6. Para el clasificador, explica la diferencia entre las métricas del set de pruebas para cada clase, ¿Qué nos dice de la calidad del clasificador por cada clase? ¿Hay alguna clase que tenga un mejor resultado en la clasificación?. Justifique. (0.3)

7. Para el regresor, ¿Qué nos dice el error sobre la calidad de la regresión? Justifique. (0.3)

8. ¿Qué observaste en la importancia de las features en el Decision Tree Regressor? Acorde al contexto del set de datos ¿parece razonable la importancia de las features encontradas para realizar la regresión? Justifique. (0.4)

Bonus A: Reducción de dimensionalidad y clustering (0,3 ptos)

Este bonus solo será contabilizado si la nota obtenida en la tarea es igual o superior a 4.

Para obtener el puntaje del bonus, deberán realizar dos tareas nuevas muy comúnes y útiles cuando se trabaja con un set de datos. Al igual que en las secciones anteriores, el código debe estar comentado en cada paso, explicando lo realizado y las toma de decisiones.

La primera tarea corresponde a reducción de dimensionalidad, en palabras simples, la idea es tomar todas las características del set de datos y hacer una transformación lineal que nos entregue la misma información, pero con un número menor de características. Esto nos permite, por ejemplo, llevar un set de datos en tres dimensiones a dos dimensiones para poder visualizarlo de forma más sencilla.

La segunda tarea corresponde a clustering, es decir, encontrar etiquetas para datos sin clase asignada previamente, agrupándolos. La idea de este tipo algoritmos es poder separar los datos cuando NO conocemos las clases, siendo este un ejemplo de aprendizaje no supervizado.

A diferencia de la parte anterior, no será necesario usar un set de entrenamiento y un set de pruebas, sino que utilizarán el conjunto de datos obtenidos en la primera parte.

Reducción de dimensionalidad: Para realizar reducción de dimensionalidad deberán usar una técnica llamada *T-distributed Stochastic Neighbor Embedding* (t-SNE), de modo que los datos queden representados con solo dos características (columnas) derivadas de las características originales.

Visualizar datos reducidos: Una vez realizada la reducción de la matriz de características a dos columnas, deberán mostrar la reducción resultante graficando los datos en dos dimensiones usando la librería gráfica que ustedes elijan. Cada eje del gráfico obtenido debe corresponder a cada una de las componentes de t-SNE, y se espera que cada punto pueda ser identificado con su clase respectiva (ya sea según color o forma del marcador). Deberán explicar cada paso realizado, explicando qué parámetros se escogieron y por qué.

Predecir las clases usando clustering: En esta etapa, desestimarán las clases reales, y sólo usando las dos características resultantes de t-SNE, deberán aplicar dos algoritmos de clustering para dividir los datos. En concreto, deberán obtener una predicción de clase para los datos reducidos. Algo muy importante a considerar es que los algoritmos de clustering a ser utilizados deberán ser K-Means (basado en centroides) y DBSCAN (basado en densidad). Deberán explicar claramente qué hiper-parámetros fueron escogidos y por qué.

Visualizar los clusters: Nuevamente, deberán graficar los datos en dos dimensiones usando la librería gráfica que ustedes elijan. Se espera que cada eje del gráfico sea una de las componentes de t-SNE y, cada dato pueda ser identificado con la clase que predijo (según el color o la forma del marcador). Deberán explicar lo observado acorde a contexto del set de datos, cómo se relaciona lo observado con los atributos trabajados. Además, deberán probar con otras dos clases extraídas de los atributos y determinar si hay alguna feature que explique los clusters.

Bonus B: Concurso de Memes (0,2 ptos)

1. Para esta tarea, tendrán una bonificación de dos décimas aquellos alumnos que elaboren y envíen los 3 mejores memes dentro del curso. Para hacer envío del meme, junto a su tarea, deben entregar el archivo en formato JPG o PNG o GIF para luego ser evaluado por nuestro comité especializado en memes. Este archivo debe tener el siguiente nombre: meme_TG1_apellidos.gif
2. Nuestro comité de memes tendrá en mente la siguiente rúbrica para la evaluación:
    - Nivel de creatividad.Relación con los contenidos del curso.

- Sentido del humor.
- Respeto con los compañeros/ayudantes/profesores al usar imágenes de otras personas, siempre verificar la autorización del otro para evitar ofensas.

3. Las decisiones del comité no son apelables.

Formato de entrega

La entrega de esta tarea se hará por medio del repositorio GitHub privado del grupo, creado a través de este link, donde deberán entregar un único archivo `.ipynb` con todo el desarrollo de la tarea y preguntas del informe respondidas al final de este. Si la tarea fue realizada con Google Collab, entonces desde ahí mismo pueden descargar el archivo `.ipynb` y subirlo al repositorio.

Es sumamente importante que elimines el *output* de tu Jupyter notebook. Tampoco debes subir el dataset (celulares`.csv`) a tu repositorio. No seguir estas instrucciones significará un descuento. Además, deberán ingresar el nombre de ambos integrantes al principio del archivo y citar **todo** código externo usado.

Entregas Atrasadas

Si así lo desea, existe la posibilidad de entregar la tarea fuera de plazo, con hasta 3 horas de atraso. Cualquier entrega realizada pasada la hora estipulada de entrega será considerada como atrasada, **sin excepciones**. Cada hora o fracción implicará un descuento de **1.5 puntos** a su nota final.
Debido a lo anterior, recomendamos fuertemente enviar su tarea con anticipación, realizando commits intermedios de ser necesario.