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

Tarea Grande 2

Machine Learning

Ayudantes: Astrid San Martín, Ricardo Schilling

Profesor: Denis Parra

Anunciada: 11 de Octubre de 2021

Indicaciones

- Fecha de Entrega: 12 de Noviembre, hasta las 19:59.
- La entrega de la tarea debe realizarse en el repositorio privado GitHub asignado para esta evaluación.
- Todas las tareas se entregarán máximo a las 19:59 del plazo indicado.
- **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.**

Objetivos

Los objetivos de esta tarea son:

- Estudiar, analizar y procesar el set de datos propuesto.
- Analizar y seleccionar las mejores características del set de datos propuesto.
- Conocer y hacer uso de un método de clasificación automática de inteligencia de máquina.
- Conocer y hacer uso de métricas para medir el rendimiento de la clasificación realizada por el método de aprendizaje automático escogido.

- Hacer uso de la librería Pandas de Python para el procesamiento de datos.
- Hacer uso de la librería `scikit-learn` de Python.
- Realizar la visualización de los resultados con una librería de python a elección (altair, matplotlib, seaborn, etc.).

Descripción de la tarea

En esta tarea tendrán la oportunidad de explorar un método de aprendizaje automático, de *machine learning*. En particular, en esta tarea usaremos un método de aprendizaje supervisado para la clasificación automática.

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. **Está permitido hacer esta tarea de forma individual, sin embargo la dificultad está calibrada para un trabajo en parejas.**

El formato de entrega de cada una de las partes que consta esta tarea deberá ser un único Jupyter notebook (`.ipynb`). Asimismo, la entrega deben realizarla en el repositorio asignado a cada pareja, **para crearlo deben usar el siguiente link**. 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 [1].

Esta tarea está dividida en tres partes, en cada parte deberán aplicar conocimientos de programación con las librerías ya mencionadas (Pandas, `scikit-learn` y, opcionalmente, `Seaborn`), pero además, deberán responder preguntas que justifiquen sus conocimientos respecto al tema, esto es igual de importante como la parte de programación. A su vez, el código deberá estar bien comentado en cada paso que realicen, debe quedar claro que comprenden las funciones utilizadas y los pasos realizados.

[1] Si luego de probar muchas con Colab tuviese problemas que no le permitieran hacer la tarea, puede 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)

Motivación

Dentro del área del aprendizaje de máquina tenemos algoritmos de aprendizaje supervisado, donde conocemos previamente la categoría o clase de cada instancia que tenemos, y algoritmos de aprendizaje no-supervisado, donde no tenemos información previa sobre la categoría o clase de las instancias. Por su puesto, a su vez, tenemos datos de la más diversa naturaleza por ejemplo series de tiempo, imágenes, registros de transacciones, etc. y que pueden venir como datos continuos, discretos, categóricos, etc. También, existe una gran variedad de métodos de clasificación, los que dependiendo de su estructura interna, serán más adecuados para datos de una naturaleza específica, lo que hace muy importante conocer y entender muy bien la estructura de cada algoritmo a utilizar.

El objetivo principal de esta tarea es que investiguen por su propia cuenta y realicen pruebas con los datos entregados sobre una amplia variedad de clasificadores, siendo capaces de comparar los resultados obtenidos y comprender el funcionamiento interno de sus mejores modelos.

Set de datos

Para esta tarea tendrán a su disposición el set de datos: `flowers.csv`, el cual deberán descargar directamente usando el siguiente link.
Este dataset corresponde a features obtenidas luego de pasar imágenes de flores a través de dos redes neuronales diferentes. Estas features corresponden a características que la red fue capaz de capturar entregando "números"para cada instancia que logran describir el contenido de la imagen.
Junto a estos números abstractos, tendrán también a que flor corresponde cada imágen, es decir su categoría o clase a la cual pertence, de manera que puedan entrenar clasificadores que sean capaces de predecir el **tipo de flor** dadas las features entregadas.

(a) Ejemplo de *daisy*.

(b) Ejemplo de *sunflower*.

(c) Ejemplo de *rose*.

Figura 1: Ejemplo de imágenes de las cuales extrajimos sus *features*.

Parte 1: Procesamiento de los datos (1.5 ptos)

Como primer paso para esta tarea deberán estudiar y realizar el preprocesamiento de los datos, con el objetivo de familiarizarse y dejarlos preparados para la siguiente parte de la tarea. Para esta parte es **obligatorio hacer uso de la librería Pandas** y queda **prohibido utilizar ciclos en esta parte**, pues se espera que el código sea eficiente.

Cargar dataset (0.1p): Para poder iniciar esta tarea, es necesario que carguen la base de datos: `flowers.csv`

Análisis de las características (0.6p): Para realizar este paso, deberán estudiar los datasets usando Jupyter Notebook, usando funciones de pandas que entreguen estadísticas de los datos (i.e. `describe()` `de pandas`), y realizando algunos gráficos de distribución de los datos, verificando outliers, features de baja varianza, datos faltantes, etc., siempre **justificando** las decisiones que tomen. Deberán mostrar y comentar al menos un gráfico que muestre, por ejemplo: la distribución de las diferentes flores en el dataset.

En esta paso también deberán limpiar valores nulos, en caso de existir, además de deshacerse de features que no sirvan para la clasificación.

Normalizar y crear label (0.4p): Para este paso, procederemos a normalizar las características, llevando los datos a un rango de valores común, y crearemos un label en función del tipo de flor. Deberán generar una matriz de atributos y un vector de label.

Separar datos de entrenamiento y pruebas (0.4p): 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.

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.

Es sumamente importante que todas las decisiones tomadas en el análisis de características y la normalización estén explicadas como comentario en el mismo notebook. Es necesario, por ejemplo, explicar qué método de normalización usan, cómo quedan los datos luego de

normalizar y bajo qué criterios eliminaron ciertas features.

No explicar una decisión implica que no habrá puntaje asociado, sin excepciones.

Parte 2: Clasificación (3 ptos)

En esta parte conoceremos y estudiaremos distintos tipos de clasificadores, los cuales representan ejemplos de aprendizaje supervisado. **A diferencia de la parte 1** donde está prohibido usar ciclos, en esta parte **si podrán hacer uso de estos.**
Luego de realizar esta separación, deberán instanciar y entrenar **10 clasificadores** distintos, los cuales deben ser capaces de predecir el tipo de flor, el que está disponible en la columna llamada **label** en su set de datos. Para estos clasificadores deberán probar distintos hiperparámetros, a modo de buscar cuáles son los que entregan los mejores rendimientos. El detalle de los hiperparámetros a modificar pueden encontrarlos en la documentación de la librería `scikit-learn`. Finalmente, deben escoger la combinación de hiperparámetros que muestre el mejor rendimiento, siendo este set de hiperparámetros el que deberá quedar en el código. Deben probar al menos 3 combinaciones distintas.

Para cada uno de estos clasificadores, deben explicar en qué consiste el clasificador, qué hiperparámetros decidieron cambiar, qué significan los hiperparámetros que cambiaron y reportar las métricas de rendimiento de este. Como mínimo deberán reportar el accuracy del modelo junto con precision, recall y f1-score por cada clase.
Al igual que la parte 1, es sumamente necesario que justifiquen la elección de hiperparámetros, reporten las métricas pedidas y expliquen los clasificadores utilizados. Cada clasificador tiene el mismo puntaje y para ser asignado puntaje el clasificador debe estar bien explicado.

Parte 3: Reducción de dimensionalidad (1.5 ptos)

En esta parte de la tarea, deberán realizar reducción de dimensionalidad. Esto permite que un investigador, por ejemplo, pueda visualizar un resumen"de los datos, pero preservando la información. De esta forma, podemos tener una mejor idea de cómo están distribuidos los datos en este espacio y cómo se podrían trabajar.
Para esta tarea, deberán trabajar la información mediante alguno de los métodos conocidos como PCA (*Principal Component Analysis*), t-SNE (*T-distributed Stochastic Neighbor Embedding*) o UMAP (*Uniform Manifold Approximation and Projection for Dimension Reduction*). El objetivo es que visualicen,

ya sea de forma bidimensional o tridimensional, la tendencia de los datos y la formación de posibles clusters de interés en función de sus features. En palabras simples, la idea es tomar todas las características del set de datos y hacer una transformación lineal[2] que nos entregue una información similar, pero de forma comprimida 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 rescatando la mayor información posible.

Para esta parte deberán utilizar la librería `Scikit-learn`, ya sea usando el algoritmo de *PCA*, *t-SNE* o *UMAP*[3] sobre los datos. Posteriormente, deberán realizar la visualización usando librerías gráficas de Matplotlib, Seaborn o Altair[4]

Para la visualización, se espera que cada eje del gráfico obtenido corresponda a cada una de las componentes del reductor de dimensiones y que cada punto pueda ser identificado con su respectivo label o categoría (mediante colores, figuras u otros, e incluyendo la leyenda). Como se ha realizado en los pasos anteriores, deberán comentar **comenten** el gráfico obtenido.

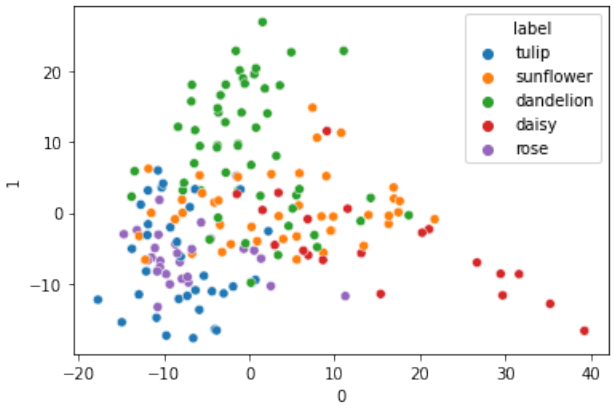

Figura 2: Ejemplo de reducción de dimensionalidad, usando PCA en un subconjunto de los datos.

Bonus Concurso de Memes (0,3 puntos)

1. Como ya saben, para cada tarea tenemos implementado nuestro concurso de memes. Tendrán una bonificación de tres décimas aquellos alumnos que elaboren y envíen los 3 mejores memes dentro

[2] Es real, hay muchísima Álgebra Lineal en Computación (O_O;).

[3] Si quieren usar UMAP, deben usar una librería distinta, llamada **umap**

[4] Por defecto Altair no permite hacer gráficos con más de 5000 puntos. Para deshabilitar esta restricción debes hacer lo que dice la documentación.

del curso. Para hacer envío del meme, junto a su tarea, deben entregar el archivo en formato de imagen JPEG, PNG o GIF; para luego ser evaluado por nuestro comité especializado en memes. El archivo debe tener el siguiente formato: **meme_TC2_apellidos.extensión** y debe estar en su repositorio de la tarea.

2. Nuestro comité de memes tendrá en mente la siguiente rúbrica para la evaluación:
    - Nivel de creatividad
    - Relación con los contenidos de la tarea
    - Sentido de humor
    - Respeto con los compañeros/ayudantes/profesores al usar imágenes de otras personas, siempre verificar la autorización 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. Si la tarea fue realizada con Google Colab, entonces desde ahí mismo pueden descargar el archivo `.ipynb` y subirlo al repositorio.

Es sumamente importante que no suban el dataset (archivo `csv`) a su repositorio. No seguir estas instrucciones significará un descuento importante. Además, deberán ingresar el nombre de ambos integrantes al principio del archivo y citar **todo** código y fuentes de información externas.

Entregas Atrasadas

No se aceptan entregas fuera del horario publicado, se revisará la hora del último commit en GitHub. No entregar dentro del horario estipulado es equivalente a no entregar y se califica con la nota mínima 1.0.
No se aceptará ninguna tarea fuera de plazo, **sin excepciones**. Recuerden hacer entregas parciales para evitar mala nota en caso de no entregar a tiempo.