

# Práctica 2: Descripción y criterios de corrección

## Aprendizaje Automático

La práctica 2 está orientada a la resolución de un problema del mundo real. Para ello, durante la primera mitad del cuatrimestre cada equipo de prácticas habrá realizado lo siguiente:

- Descarga o elaboración de la base de datos con la que se va a trabajar.
- Desarrollo de las funciones de AA como consecuencia de la práctica 1.

El desarrollo de la práctica 2 se plasmará en una memoria final que será evaluada. La redacción de la memoria deberá ser realizado por cada equipo de prácticas, y esta se realizará de forma incremental, para lo cual se realizará un seguimiento del trabajo realizado por cada equipo de prácticas, así como correcciones de la memoria desarrollada, durante las clases de prácticas. El objetivo de estas correcciones será mejorar la memoria, como documento en el que se plasma el trabajo realizado por cada equipo, para aumentar la calidad del trabajo realizado y aprender a realizar la redacción de un documento técnico. Por lo tanto, durante cada clase de prácticas cada equipo recibirá una nota de cada parte de la memoria, así como una descripción de lo que les falta para alcanzar la nota máxima en esas partes, en función de los criterios que se especifican más abajo en este documento. De esta forma, en cada clase de prácticas no solamente se incluirán nuevas secciones, sino que se podrán modificar las anteriores para mejorar la memoria e intentar alcanzar la nota máxima.

Por lo tanto, en la segunda mitad del cuatrimestre, las prácticas de esta asignatura se centran en realizar distintas aproximaciones de cara a resolver el problema escogido, junto con la correspondiente redacción de la memoria.

A finales de cuatrimestre se solicitará una entrega final con todo el material desarrollado, lo cual incluye:

- Código fuente utilizado.
- Base de datos usada.
- Memoria redactada.
- Presentación realizada en clase (si se hace).

A la hora de desarrollar el código, se debería seguir una estructura similar a esta:

- Crear una carpeta con un nombre similar a *fonts*, donde se sitúe el código de la práctica 1.
- Crear una carpeta con un nombre similar a *datasets*, donde estarán los datos usados en las distintas aproximaciones. Dentro de esta carpeta se pueden crear carpetas nuevas, siendo la forma habitual el crear una carpeta para cada clase. Por ejemplo, si el trabajo es de detección de ojos, esta carpeta tendrá una que se llamará *ojo* y otra que se llamará *no-ojo*. Esta estructura puede cambiar como el equipo de trabajo considere necesario. Por ejemplo, es posible que a lo largo de las distintas aproximaciones se carguen datos distintos, por lo que los nombres de las carpetas podrían modificarse para ajustarse a esto, o incluso crear una jerarquía de carpetas para las distintas aproximaciones.
- En la carpeta raíz, crear un archivo para cada aproximación, con nombres significativos como *aproxAA.jl* o *aproxDeepLearning.jl*. Estos archivos deberían de ser muy sencillos, puesto que sólo deberían realizar las siguientes operaciones:
  - Fijar la semilla aleatoria para garantizar la repetibilidad de los resultados.
  - Cargar los datos.
  - Extraer las características de esa aproximación.
  - Para el caso de la aproximación basada en modelos de AA clásicos, llamar a la función *modelCrossValidation* para realizar validación cruzada con distintos modelos y configuraciones de parámetros. Para la aproximación con *Deep Learning*, se utilizará una función similar, a desarrollar por cada equipo, como se verá posteriormente.

Una cuestión importante es que al ejecutar estos archivos el sistema debería entrenar los modelos y mostrar por pantalla los mismos resultados, gráficas y tablas que aparezcan en la memoria, en las secciones correspondientes a cada aproximación.

Con respecto a la memoria, esta constituirá el centro de la evaluación del trabajo, puesto que cada parte se debería ver plasmada en ella. En este trabajo se pedirán dos aproximaciones. La primera de ellas será utilizando técnicas clásicas de AA: Redes de Neuronas, kNN, Árboles de Decisión, SVM y DoME. La segunda será utilizando *Deep Learning*.

La memoria del trabajo realizado en esta asignatura debería tener los siguientes apartados:

1. Introducción
2. Descripción del problema
3. Análisis bibliográfico
4. Aplicación de modelos de AA
5. Aplicación de *Deep Learning*
6. Conclusiones
7. Trabajo futuro
8. Bibliografía

En la mayoría de las publicaciones que encontraréis en general no se seguirá este esquema. Sin embargo, los trabajos publicados en general sí tienen la misma información, aunque esté organizada de otra manera.

A continuación, se describe la información que debería especificar cada una de las secciones, junto con los criterios usados para su corrección y la puntuación de cada parte que, en total, suman 2.5 puntos de la nota final:

1. Introducción. Esta sección contiene una introducción al trabajo, en la que, comenzando de forma genérica al problema a resolver, se describa la problemática asociada, abordando temas como la importancia de este, las posibles ventajas de resolverlo, o las desventajas de no hacerlo.

Esta sección debería contener también los objetivos del trabajo, así como una descripción general del problema a resolver, dejando los detalles concretos para secciones posteriores.

**Puntuación:** 0.15 puntos. Se valorará la correcta contextualización del problema y descripción del mismo, así como la claridad de la explicación.

2. Descripción del problema. Esta sección debería contener una descripción detallada del problema concreto a resolver, dando detalles como pueden ser los siguientes:

- Restricciones que se aplican al problema a resolver. Por ejemplo: imágenes siempre horizontales, fotografías de personas con el fondo siempre blanco, etc.

- Descripción de la BD, aportando información como cuántos patrones tiene o qué naturaleza tienen los patrones, así como de qué información se dispone para cada instancia.
- Origen de la base de datos, tanto en el caso de haber sido desarrollada por el equipo de trabajo, como haber sido descargada de internet u otra fuente. En este último caso, es importante especificar la fuente.
- Propiedades de los datos que el equipo de trabajo considere importantes. Entre ellas, las propiedades estadísticas pueden ser de utilidad para comprender decisiones que se tomen con posterioridad. Por ejemplo, si los valores en un determinado atributo están acotados y estos valores se distribuyen de una forma más o menos uniforme en el interior, esto parece indicar que la mejor forma de normalizar sería entre máximo y mínimo.

**Puntuación:** 0.15 puntos. Se valorará la claridad de la explicación y de los detalles aportados sobre los datos con los que se trabajará. Una explicación que es necesaria en esta parte es la justificación de qué métrica o métricas se utilizarán para valorar y comparar los clasificadores que se generen, y por qué son más adecuadas que el resto de métricas.

3. **Análisis bibliográfico.** Esta sección debería contener un resumen de los trabajos más recientes e importantes en el ámbito tratado, con las referencias bibliográficas de trabajos en donde se resuelva el mismo tipo de problema. En muchas ocasiones no se encontrarán trabajos que resuelvan exactamente el mismo problema; de todas formas, es interesante describir y referenciar trabajos que resuelvan el mismo problema o similares aplicando las mismas técnicas que se emplearán en este trabajo, o aplicando diferentes tipos de técnicas.

Cada referencia debe describirse brevemente, de tal forma que al lector se le dé una idea de cómo está el estado actual de la cuestión al resolver este problema. Para citar estas referencias bibliográficas se debe utilizar un estilo o norma concreto, el mismo en todo el documento. En las memorias a presentar en esta asignatura, no se os va a exigir un estilo concreto; por ese motivo, se recomienda que se use el mismo estilo que se usará en la elaboración del Trabajo Fin de Grado, puesto que es algo que tendréis que consultar más adelante en la carrera.

Una forma muy habitual pero incorrecta de escribir esta sección es poner los trabajos encontrados como una lista. En lugar de eso, lo que habría que hacer es describirlos de forma

continúa en el texto, con un hilo conductor. A continuación, se muestra un ejemplo, en el que se puede ver que cuando se referencia cada trabajo se hace una breve descripción del mismo:

“Algunos trabajos siguen aproximaciones que utilizan modelos bayesianos, como el que presentan G. Greenspan y D. Geiger [Greenspan 2004], en el que se modela la distribución de los haplotipos mediante una red bayesiana, integrando la identificación de bloques de haplotipos y la resolución de haplotipos, o el que presentan D. V. Conti y W. J. Gaurderman [Conti 2004], basado en un framework de modelización lineal generalizada para representar la arquitectura genética subyacente a través de combinaciones lineales de SNPs, realizando la selección del modelo adecuado utilizando, a su vez, un modelo de Bayes. En el último caso, sin embargo, se observa un mal funcionamiento al añadirla estimación de la fase de los haplotipos al modelo, posiblemente debido al ruido introducido por la incertidumbre resultado de realizar la estimación de la fase.”

**Puntuación:** 0.15 puntos. Se valorará el número de trabajos descritos, además de las propias descripciones. En este aspecto, para obtener la nota máxima será necesario incluir al menos 8 trabajos y describirlos brevemente. En ocasiones no será posible encontrar referencias directamente relacionadas con la temática a desarrollar, en estos casos se podrán analizar trabajos con temáticas similares. Como se ha expuesto anteriormente, estos trabajos deberán estar correctamente referenciados siguiendo un estilo concreto, siendo estas referencias de alguna publicación en libros, revistas, actas de congreso o publicaciones similares.

4. Aplicación de modelos de AA. Esta sección contendrá todos los datos del desarrollo del sistema en sí en la aproximación realizada por el equipo para resolver el problema. Esta sección se dividirá en varias partes que tengan una naturaleza claramente diferenciada:

- Descripción. Esta sección debería describir cómo es la solución propuesta por el equipo, incluyendo tareas como preprocesado de los datos y normalización, extracción de características, etc. Todas las decisiones de implementación que se tomen deberán estar justificadas, por ejemplo:
  - Si hay algún atributo irrelevante que se va a eliminar, justificad esto.
  - Si los datos requieren de normalización, justificar por qué el tipo escogido es el más adecuado. Si no requieren normalización, justificar por qué no se realiza.

- Si se realiza algún otro tipo de preprocesado, describirlo adecuadamente.

**Puntuación:** 0.4 puntos. Se valorará la claridad de la descripción, de esta parte, incluyendo conceptos como:

- Descripción razonada de las características, apoyándose en gráficas y/o imágenes explicativas.
- Descripción de la base de datos utilizada en esta aproximación. A pesar de que en la sección “Descripción del problema” ya se hayan descrito los datos, es posible que esta varíe ligeramente según las características utilizadas, por lo que es conveniente tener una descripción incluyendo cuántos patrones se han usado, cuántas entradas, clases, etc.
- Preprocesado de los datos, que generalmente suele ser relativo a la normalización de los mismos. Es necesario justificar el porqué del tipo de normalización utilizado, así como los parámetros de normalización (mínimo, máximo, media, etc.), o por qué no se realiza normalización.
- Otros datos relativos a los experimentos que se van a llevar a cabo, como metodología, número de folds, etc.
- Cualquier otro material como gráficas en las que se muestren los datos puede ser de interés para esta parte.

En este apartado se valorará especialmente el nivel de elaboración de las características utilizadas.

- Resultados. Esta sección contiene las descripciones de la parte experimental, con información como los modelos utilizados junto con sus hiperparámetros y sus valores, la metodología utilizada, etc.

La idea de esta sección es que con los datos aportados, junto con la anterior, una persona cualquiera sea capaz de descargarse (u obtener) la base de datos, preprocesarla de la misma manera, y obtener los mismos resultados. Para cada algoritmo es necesario experimentar con los distintos valores de los parámetros más importantes.

Una vez ejecutadas las pruebas, en la memoria se detallarán los resultados en forma de tablas y/o gráficas para cada modelo y configuración utilizado. También son de interés las matrices de confusión. Cuando se muestren tablas en la memoria, mostrar no solamente el valor promedio, sino también las desviaciones típicas.

En general, todas estas formas de mostrar resultados deberían de reflejar los resultados de test. Si se desea, se pueden incluir resultados medidos en el conjunto de entrenamiento; sin embargo, no pueden omitirse los datos en el conjunto de test.

**Puntuación:** 0.3 puntos. Esta sección contiene la parte experimental. En ella, se valora la claridad de las explicaciones, así como el número de experimentos. Es necesario realizar experimentación con las 5 técnicas (Redes de Neuronas, SVM, árboles de decisión, kNN y DoME), y, para cada una, probar con distintos hiperparámetros.

- Para el caso de RR.NN.AA., probar al menos 8 arquitecturas distintas, entre una y 2 capas ocultas.
- Para SVM, probar con distintos kernels y valores de C. Como mínimo, 8 configuraciones de hiperparámetros de SVM.
- Para Árboles de Decisión, probar al menos 6 valores de profundidad distintos.
- Para kNN, probar al menos 6 valores de k distintos.
- Para DoME, probar al menos 8 valores distintos de número de nodos, en el intervalo donde se hayan obtenido los mejores resultados.

En todos los experimentos realizados, es necesario usar las métricas descritas en la sección 2 de la memoria para evaluar y comparar los modelos obtenidos.

- **Discusión.** En esta parte se comenta razonadamente, el porqué de los resultados anteriores. El objetivo de esta sección es dar al lector una mayor comprensión de cómo funciona el sistema, comentando cuestiones como:
  - Si los resultados se considera que han sido buenos o malos, y por qué un determinado valor de una métrica se considera un buen o mal resultado.
  - Si las características extraídas, en el caso de procesado de imágenes o señales, han sido correctas o no. En este aspecto, comentar también por qué algunos

conjuntos de características podrían dar mejores resultados que otras, o qué otras características podrían funcionar mejor.

- Qué algoritmo ha funcionado mejor, y cuál peor, y a qué se cree que es debido.
- Si hay algún resultado que se considere interesante o imprevisto, es susceptible de ser comentado también en esta sección.

**Puntuación:** 0.3 puntos. En esta sección se valorará la claridad de la explicación y los razonamientos empleados, indicando el impacto que tienen en el sistema desarrollado, que deberán apoyarse en los resultados obtenidos, así como en otros que se quieran mostrar aquí, como pueden ser matrices de confusión o distintas gráficas explicativas.

## 5. Aplicación de *Deep Learning*.

**Puntuación:** 0.75 puntos. Para optar a la máxima nota, es necesario seguir una estructura similar a la de una aproximación anterior, que será evaluada siguiendo criterios similares, con las siguientes diferencias:

- Descripción: **Puntuación: 0.25.** La nota es inferior a la de la sección anterior porque en esta no se realiza extracción de características, sino que la propia imagen o señal se utiliza como entrada a la red convolucional
- Resultados: **Puntuación: 0.25.** La nota es inferior a la de la sección anterior porque solamente se experimentará con un único tipo de modelo, redes convolucionales. Sin embargo, los resultados deberán ser expuestos mediante tablas, gráficos y/o matrices de confusión como en la sección anterior. Se deberá experimentar con un mínimo de 10 arquitecturas distintas para optar a la máxima calificación.
- Discusión: **Puntuación: 0.25.** Similar a la de la sección anterior, pero comparando los resultados obtenidos con redes convolucionales.

Para realizar la aproximación basada en *Deep Learning* se proveerá de un código fuente que podréis modificar para adaptarlo a vuestro problema. Sobre esto, tened en cuenta las siguientes observaciones con respecto a realizar esta aproximación utilizando *Deep Learning*:



- En Deep Learning se suele trabajar con bases de datos enormes, con lo que hacer validación cruzada es muy lento. Por este motivo, aunque desde el punto de vista teórico habría que hacerla, no es algo habitual. Sin embargo, en este trabajo se utilizan bases de datos pequeñas, por lo que sí se deberá hacer validación cruzada. Además, al igual que en el entrenamiento de redes convencionales, en cada *fold* habrá que entrenar la RNA varias veces debido al carácter no determinístico del proceso de entrenamiento de las redes. Para ello, se puede sencillamente modificar el código utilizado para que la función que realiza la validación cruzada admita un nuevo tipo de modelo (redes convolucionales), y, de forma similar a las redes convencionales, itere en un bucle en cuyo interior llame a una función que cree y entrene este tipo de redes.
- Igualmente, como las bases de datos en *Deep Learning* suelen ser muy grandes, los patrones de entrenamiento se suelen dividir en subconjuntos denominados *batches*. En vuestro caso, como el número de patrones no va a ser tan grande, esto no es necesario, así que utilizar todos los patrones de entrenamiento en un único *batch*, como se ha venido haciendo en las prácticas hasta el momento.
- La RNA que os dejamos en Moodle tiene varias capas convolucionales y *maxpool*. Cada capa convolucional implementa filtros de un tamaño para pasar a un número distinto de canales. Al igual que en aproximaciones anteriores, es interesante hacer pruebas con distintas arquitecturas (un entrenamiento con cada una), para hacer una tabla comparativa que poner en la memoria. En este aspecto, para tener la máxima nota en esta parte se exige realizar pruebas con al menos 8 arquitecturas distintas.
- En redes convolucionales las imágenes de entrada son de tamaño fijo, porque se tiene una neurona de entrada por pixel de la imagen y canal (RGB o escala de grises). Como hemos estado trabajando con ventanas de tamaño variable, lo que se suele hacer es cambiar el tamaño de las ventanas para que todas tengan el mismo tamaño. Esto se puede hacer fácilmente en el código después de cargar la base de datos con la función `imresize`, tenéis más documentación en <https://juliaimages.org/latest/pkgsg/transformations/>
- Si en lugar de usar redes convolucionales para procesado de imagen se usan para procesado de señales, es necesario realizar modificaciones adicionales al código. Por ejemplo, el tamaño de los filtros de convolución empleado en el código es (3, 3), en

lugar de ser bidimensional (para imágenes) debería ser de una dimensión (para señales).

- La base de datos del ejemplo (MNIST) ya tiene un conjunto de patrones que debe ser usado para test. En vuestro caso no será así, sino que el conjunto de test será creado mediante validación cruzada.
- Los filtros que utiliza el código son pequeños (3x3), y funcionan bien cuando las imágenes también son pequeñas, como en el caso de MNIST. Si son grandes, compensa reducir las imágenes de tamaño con *imresize*. También se podría aumentar los filtros, pero en ese caso todo el entrenamiento se vuelve más lento.

6. Conclusiones. El objetivo de esta sección es desarrollar las conclusiones que se derivan de este trabajo, siempre teniendo como base los resultados hallados en la sección anterior. Algunas cuestiones que se pueden desarrollar en esta sección son las siguientes:

- Comentar si se considera que los resultados han sido buenos o malos, y, en consecuencia, si el trabajo ha cumplido los objetivos iniciales.
- Comentar la viabilidad de utilizar este tipo de sistemas para resolver el problema, o si acaso se podrían obtener mejores resultados con otro tipo de sistemas.
- Comentar cuál ha sido el mejor de los modelos, junto con el conjunto de valores de hiperparámetros que ofrezcan mejores resultados. Una cuestión a tener en cuenta es que es muy posible que los resultados de varias configuraciones sean parecidos, con lo que es necesario observar las desviaciones típicas para poder valorar si el que ofrece mejores resultados es realmente el mejor. Para poder realizar esta afirmación, sería necesario realizar un test estadístico. A falta del mismo, es mejor no utilizar sentencias con afirmaciones absolutas sobre que una técnica es mejor que el resto, sino redactar algo similar a “los resultados parecen indicar que los SVM con un kernel lineal y valor de  $C=100$  ofrecen mejores resultados que el resto a falta de un test estadístico que lo confirme, dado que el valor promedio en la métrica seleccionada es superior a la del resto de configuraciones, y la desviación típica muestra igualmente una gran diferencia”, o “los resultados muestran unos valores promedio muy similares, y, a pesar de haber una técnica cuyos promedios son superiores, las desviaciones típicas tan amplias hacen suponer que un test estadístico concluiría que no hay diferencia estadísticamente significativa”.

- En esta sección también se pueden incluir comentarios más personales, como por ejemplo cuáles han sido las principales dificultades que el equipo se ha encontrado en el desarrollo del trabajo, o qué se ha aprendido con todo este trabajo.

**Puntuación:** 0.15 puntos. Se valorará la claridad de la explicación, así como que las conclusiones que se extraigan estén apoyadas por los resultados obtenidos.

7. Trabajo futuro. En base a los resultados alcanzados y las conclusiones extraídas, establecer una serie de líneas de trabajo que se podrían iniciar a partir de este. A continuación se muestran varios ejemplos de líneas de trabajo que se podrían desarrollar en esta sección:

- Cómo aplicar este sistema a otros entornos o problemas.
- Cómo aplicar este sistema en el mundo real.
- Aplicar nuevas técnicas de AA o *Deep Learning* en este problema.
- Cómo aplicar este sistema para resolver problemas más complejos, con una mención de qué problemas se podrían resolver, o tal vez el mismo problema sin alguna/s de la/s restricción/s de la sección 2.

**Puntuación:** 0.15 puntos. Se valorará la claridad de la explicación, así como que los trabajos propuestos sean factibles y se deriven del trabajo realizado.

8. Bibliografía. Una bibliografía se compone de una serie de citas bibliográficas. Esta sección no suele contener nada de texto más allá de las citas bibliográficas en sí. La citación bibliográfica está formada por un conjunto de datos que permiten describir e identificar de forma abreviada cualquier tipo de documento para localizarlo posteriormente sin dificultad. Se citarán documentos (artículos de revista, libros, webs...) a lo largo del texto y se incluirán al final de la memoria las referencias bibliográficas correspondientes.

Es necesario utilizar un estilo para poner las referencias (a lo largo del texto) y la bibliografía (en esta sección). En esta asignatura no se obliga a utilizar un estilo concreto, pero, como se ha escrito anteriormente, se recomienda utilizar el estilo descrito en la normativa de Trabajos Fin de Grado.

En base a estos criterios, la memoria será puntuada en cada TGR, dando la posibilidad de que se aumenten los puntos obtenidos en el TGR anterior. La memoria constituye el documento a ser

puntuado puesto que debe ser un reflejo del trabajo realizado. La redacción de la memoria debería ser realizada por el equipo de prácticas. Para asegurarse de esto, se utilizarán herramientas de detección de plagios o de texto sintético.

Además, el código fuente también será analizado para verificar su corrección. Como consecuencia de ello, se podrá tener alguna penalización en la nota final del trabajo si se encuentran errores de concepto graves, como pueden ser:

- Código que no se ejecuta (2 puntos).
- Los resultados no coinciden con los mostrados en la memoria (1 punto).
- Utilizar muestras de test para entrenar los modelos (1 punto).
- etc.