

## Práctica final del módulo de Deep Learning

### Objetivo

El objetivo de la práctica final del módulo de Deep Learning consiste en solucionar un problema del mundo real usando las técnicas vistas en clase. En concreto, lo que se pretende es tratar de predecir el precio de habitaciones de Airbnb utilizando para ello todas las características disponibles en el dataset.

El propósito final no es tener un sistema con una precisión altísima, sino que combinéis distintos tipos de características (numéricas, texto, imágenes...) y que expliquéis cómo lo habéis hecho.

En el siguiente enlace podéis ver un ejemplo de cómo se pueden combinar distintos tipos de características con una red neuronal:

<https://www.pyimagesearch.com/2019/02/04/keras-multiple-inputs-and-mixed-data/>

El conjunto de datos escogido es éste, extraído de Airbnb mediante técnicas de scraping disponible en Kaggle:

<https://www.kaggle.com/datasets/stevezhenghp/airbnb-price-prediction>

Por cuestiones de homogeneidad, se tomarán X muestras (por tanto, X número de imágenes) para la realización de esta práctica. Para ello, se os facilitará un primer Google Colab que tendréis que copiaros y sobre el que tendréis que trabajar. Este Google Colab facilita algunas funciones necesarias para la descarga inicial de datos desde Kaggle. Cabe la posibilidad de que en el momento de realizar la práctica, alguna de estas imágenes no esté disponible. En ese caso, se procurará contar con tantas imágenes como sea posible. Como el dataset está disponible desde Kaggle,, necesitaréis tener una cuenta en dicha plataforma y generar una API Key siguiendo los pasos que se comentan en el **Apéndice A** de este documento.

### Tarea

Diseñar, implementar y evaluar una serie de algoritmos predictivos basados en redes neuronales profundas, así como en una combinación de los mismos, para la tarea de predicción del precio de habitaciones en Airbnb.

Dicha tarea puede ser abordada de manera indistinta como una clasificación de rangos de precios (clasificación) o una predicción directa del valor del precio (regresión). A su vez, la tarea se compone de 4 secciones, cuya finalización completa se exige para superar con éxito el módulo:

- Modelo 1D/tabular: Emplear al menos 1 red neuronal profunda basada en capas Dense para realizar la predicción a partir de datos tabulares.
- Modelo 2D/imágenes: Emplear al menos 1 red neuronal profunda basada en capas convolucionales o capas de atención (Transformers) para realizar la predicción a partir de imágenes.
  - a. Es importante destacar que el alumno debe priorizar el empleo de arquitecturas pre-entrenadas, congelando la parte ya entrenada del modelo y obteniendo embeddings que le permitan afrontar el problema de manera ágil empleando luego un clasificador basado en capas Dense de su elección.
  - b. Tan sólo en el caso de que el alumno disponga de tiempo, se le anima a que de manera opcional entrene por completo un modelo 2D. Sin embargo, se recomienda completar primero toda la práctica antes de abordar este punto.
- Estrategia late-fusion: Empleo de 1 modelo DL/ML que sea capaz de combinar las predicciones realizadas por un modelo 1D y por un modelo 2D para realizar a su vez la predicción.
- Estrategia early-fusion: Empleo de 1 modelo DL/ML que sea capaz de combinar los embeddings obtenidos por un modelo 1D y por un modelo 2D para realizar a su vez la predicción.

#### Criterios de corrección:

Se trata de un problema complejo. Se advierte al alumno que no debe frustrarse si los resultados iniciales no son especialmente halagüeños. Se recomienda completar la práctica antes de intentar optimizar alguna de las partes. Por tanto, se tendrá en cuenta antes:

- **La corrección en las ideas**, expresada como **razonamientos claros y técnicos acerca de las decisiones tomadas** en las diferentes partes que conforman la práctica.
- **Obtención de resultados y evaluación** de los mismos. **Análisis de errores.**
- Experimentos realizados, así como su justificación (no trabajar sin pensar antes). Se valorará muy positivamente la exploración de hiperparámetros, no tanto en aras de obtener un mejor resultado como de informar, entender y valorar su impacto en la solución.
- **Limpieza y claridad del código.**

#### Modo de entrega

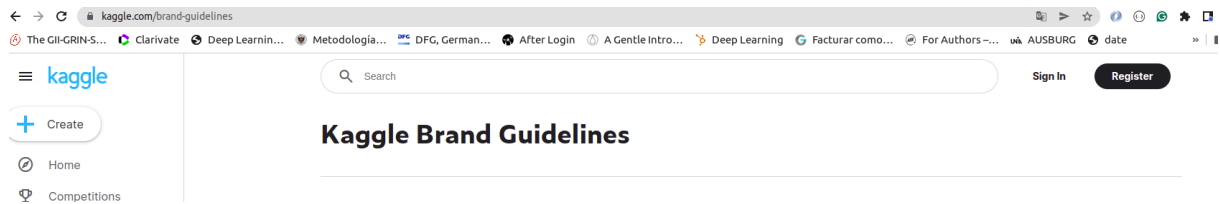
El método de entrega deberá ser mediante Google Colab preferiblemente, o en un repositorio de github de no ser posible utilizar Colab. Recordad que lo importante no es el código, si no que expliquéis lo que habéis hecho y por qué.

*Entrega oficial: Domingo 16/6/2024, 23.59 Hora España (GMT+1)*

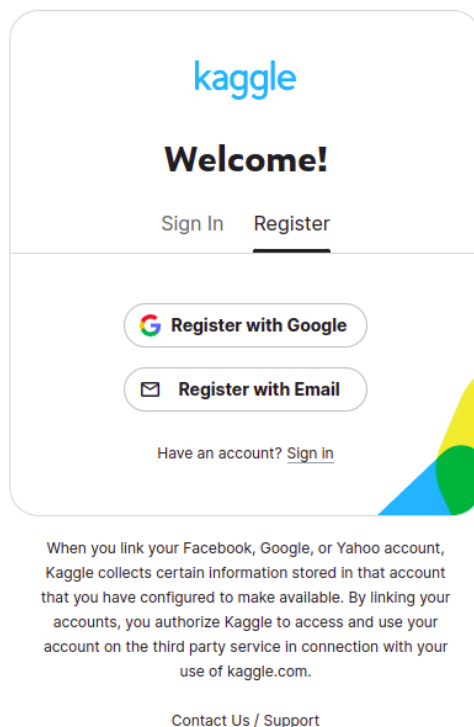
## Apéndice A

### Crear cuenta en Kaggle:

1. Vamos a :
2. Clicamos en el botón negro de Register/Registrar:



3. Nos registramos con la cuenta de Google con la que haremos la práctica clicando en Register with Google/Registrarse con Google:



4. Acabamos el registro rellenando los datos que nos pidan (si nos piden más)



---

## Pasos para crear API key y descargarnos el dataset:

1. En **Settings/Ajustes** > Ir a **API** > **Crear Nuevo Token/Create New Token** > Se descarga el token en un *kaggle.json*
2. Vamos a colab y copiamos (o corremos) lo siguiente:

```
! pip install -q kaggle
```

```
from google.colab import files
```

```
files.upload()
```

Elegir archivos

Ninguno archivo selec.

Cancel upload

3. Clicamos en Elegir archivos y subimos el 'kaggle.json' que se nos ha descargado
4. Una vez subido el fichero, copiamos los siguientes comandos:

```
! mkdir ~/.kaggle
! cp kaggle.json ~/.kaggle/
! chmod 600 ~/.kaggle/kaggle.json
! kaggle datasets list
```

\*\* Podremos seguir estos mismos pasos desde el Google Colab inicial que os facilitaremos.



## Referencias:

- <https://www.kaggle.com/discussions/general/74235>