# Guion

## (FRAN) Introducción

**[DIAPOSITIVA 4]**

Hablar sobre las herramientas que hemos usado: scikit-learn para RandomForest, Numpy, Jupyter, Pandas, seaborn, etc. y para que las hemos usado (muy brevemente todo que son 10 min)

## Práctica 3 – Naive Bayes

### (LUIS) Estructura del algoritmo

**[DIAPOSITIVA 5]**

Describir brevemente la estructura que se ha planteado para el seguimiento del algoritmo

### (FRAN) Descripción del dataset

Nº de casos utilizados (filas), descripción de atributos (columnas), tipos de datos de cada columna, valores posibles. TAMBIEN DECIR EL OBJETIVO DE LA PREDICCIÖN de cada DATASET (QUÉ SE PRETENDE PREDECIR Y OBTENER COMO RESULTADO).

Mostrar imágenes o gráficas con el % de valores de cada columna y su frecuencia

#### Set 1

**[DIAPOSITIVA 6]**

#### Set 2

**[DIAPOSITIVA 7]**

### (FRAN) Exploración y preprocesamiento de datos

**[DIAPOSITIVA 8]**

Definir atributos objetivo (target) y características (featuress). Así como la frecuencia de los atributos objetivo en cada feature.

**[DIAPOSITIVA 9]**

Hablar de la limpieza que hemos hecho, atributos que hemos quitado porque restaban importancia al modelo, modificación de valores, normalización. También decir que para el set 2 hemos normalizado los dos atributos target (piso y edificio) a predecir, en un solo atributo textual para no enfrentarnos a un problema de multi-target regression (en la que es necesario predecir más de un tipo de atributo u objetivo a partir de los datos de entrada). Y que, para el entrenamiento del modelo, se ha hecho un *encoding* de esos valores textuales del nuevo atributo “FLOOR\_BUILDING” a valores numéricos enteros (“2\_1” esta codificado a 0, “0\_0” codificado a 1, “3\_2” es 2 y así) y dices que esto se conoce como clasificación multiclase en ambos casos (el resultado de la predicción no es binario 0-1, sí-no, sino que el resultado pueden ser más de 2 tipos de valores).

### (LUIS) Procedimiento de entrenamiento del modelo

**[DIAPOSITIVA 10]**

Hablar de vectorización de los atributos objetivo para facilitar la tarea al modelo y del encoding (aquí eran todo números así que no hacía falta)

Hablar de overfitting y underfitting y las distintas pruebas que se han hecho

Hablar de tasa del error y precisión del modelo

El set 1 tiene menos atributos (1 target y 7 features) por lo que la media del error es menor y la tendencia de la precisión es mayor, con un cjto de entrenamiento pequeño, provocando un tiempo de generación del modelo menor que el set 2.

El cual tiene mucho más tamaño (casi 6 veces más grande que el set 1), y diferente configuración (2 targets: predecir piso y edificio a partir de 520 puntos y 520 features). Por lo que la media del error es mucho más grande, la precisión es mucho menor debido a la gran cantidad de datos que tiene que manejar (520 ptos) y el tiempo de entrenamiento es significativamente mayor, aunque no demasiado, debido a que los cálculos computaciones que realiza el modelo es a muy alto nivel (matemáticas básicas) ya que el algoritmo que se ha utilizado no es una implementación de ninguna biblioteca sino una implementación desde cero usando el lenguaje de programación.

Hablar por qué es mejor la variación de datos de 70-30 frente al resto (aquí meter lo de underfitting y overfitting)

### (LUIS) Procedimiento de predicción del modelo

**[DIAPOSITIVA 11]**

Se puede ver que el modelo de 70-30 con un 98 % de precisión para el set 1 acierta la gran mayoría de valores en la predicción.

**[DIAPOSITIVA 12]**

Mientras que el modelo del set 2 , debido a las razones descritas anteriormente, no genera una predicción muy buena.

### (LUIS) Estadísticas generales y conclusiones extraídas

**[DIAPOSITIVA 13]**

Decir que una es una matriz Scatter o de dispersión

Y la de la derecha una matriz de confusión que permite ver qué valores se asocian correctamente entre los valores reales y los predichos. Por ejemplo, en el caso del set 1, el modelo suele confundir las posiciones del piso 2 y las predice como piso 3 (por eso la celda tiene valor 5). Aunque el resto son fallos menores ya que el modelo tiene una alto valor de predicción

**[DIAPOSITIVA 14]**

Lo mismo para el set 2. No enrrollarse mucho

## Práctica 4 – Arboles de decisión

### (LUIS) Estructura del algoritmo

**[DIAPOSITIVA 15]**

Describir brevemente la estructura que se ha planteado para el seguimiento del algoritmo (la misma que el otro solo que ahora con un gridsearch para búsqueda de los mejors hiperparámetros porque no es una implementación a pelo sino una implementación de una biblioteca que requiere una configuración determinada de entrada)

### (FRAN) Descripción del dataset (Set 1)

**[DIAPOSITIVA 16]**

Nº de casos utilizados (filas), descripción de los atributos (columnas), tipos de datos de cada columna, valores posibles. TAMBIEN DECIR EL OBJETIVO DE LA PREDICCIÖN (QUÉ SE PRETENDE PREDECIR Y OBTENER COMO RESULTADO).

Poner fuente de origen del dataset (Kaggle) y echarle flores a Kaggle y por qué hemos usado ese dataset.

Y decir los valores de la tabla de targets más frecuentes.

### (FRAN) Exploración y preprocesamiento de datos

**[DIAPOSITIVA 17]**

Definir atributos objetivo (target) y características (features), limpieza que hemos hecho, atributos que hemos quitado porque restaban importancia al modelo, modificación de valores, normalización.

### (LUIS) Procedimiento de entrenamiento del modelo y uso de scikit-learn y funcionamiento del algoritmo DecisionTree

**[DIAPOSITIVA 18]**

Hablar de vectorización de los atributos objetivo para facilitar la tarea al modelo y del encoding

Hablar del gridsearch (Cross validation y random) -> <https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74>

Hablar en el gridsearch de qué es cada hiperparámetro (max\_depth, n\_estimators, …)

Hablar de overfitting y underfitting

Hablar de tasa del error y precisión del modelo

Hablar de hiperparámetros

Hablar de scikit-learn y echarle flores y a su facilidad para implementar algoritmos y crear modelos de aprendizaje supervisado desde 0 (para aquellos que no son data scientists)

### (LUIS) Procedimiento de predicción del modelo y árbol generado

**[DIAPOSITIVA 19]**

Hablar valores comparativos entre aquellos que se han predicho y los reales (faltan géneros que no ha tomado en la predicción)

**[DIAPOSITIVA 20]**

Hablar de que a mayor número de árboles en los hiperparámetros de la generación del randomforest mayor es el tiempo de entrenamiento ya que se generan más ramas y mayor es la precisión cuadrática (hablar de F1 score) en datos de entrenamiento y de prueba de forma lineal.

Lo mismo ocurre en la gráfica de la derecha al variar el número de parámetros (features) en el set de entrenamiento, cuanto mayor es el número de de características mayor es la precisión del modelo (hasta estabilizarse sobre las 5-6 features) y mayor es el tiempo de entrenamiento (ya que el cjto de datos de entrada es mayor)

**[DIAPOSITIVA 21]**

En la matriz scatter o de dispersión de la izquierda (ponemos un ejemplo de que hemos usado esta gráfica para observar la relación entre parámetros en los valores de predicción) se puede ver una relación entre dos parámetros que casan (popularidad de una canción con respecto al nivel de baile que genera en la gente). Se ve que la tendencia es que a mayor popularidad, tiende a habar menos géneros que sean bailables, estando la mayoría en un grado medio.

Hablar que la matriz de confusión (derecha) nos permite ver cual es la relación entre los diferentes parámetros y cual combina mejor con otros a la hora de entrenar el modelo (los que están en negro son los que peor combinan)

#### Profundidad 2

#### Profundidad 3

#### Profundidad 4

#### Profundidad n

## (LUIS Y FRAN) Conclusiones finales

**[DIAPOSITIVA 25]**

Hablar del Naive Bayes y del DecisionTree (modo *entropy* para simular la ganancia del algoritmo ID3) y arboles de decisión en problemas de este tipo:

Las aplicaciones habituales para el uso e implementación de este algoritmo suelen ser en problemas que cumplen tres requisitos diferenciados: en aquellos cuyo objetivo consiste en la clasificación de texto (análisis de sentimiento, *clusterización*, etc.), cuando el conjunto de datos que se utiliza en la problemática es enorme y cuando el conjunto de entrenamiento es , proporcionalmente, pequeño en comparación con el conjunto total. -> <https://towardsdatascience.com/all-about-naive-bayes-8e13cef044cf>

<https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html>