1. Supercised Learning

El Supercised Learning se puede clasificar en dos grandes grupos:

- Supercised Learning
  - Classification
    - Regresión logística
    - K-Nearest Neighbors (KNN)
    - Decision Trees
    - Random Forest
  - Regression
    - Linear Regression

Existen métodos y algoritmos que se pueden usar en ambos problemas:

Supervised Learning	**Classification**	**Regression**
Logistic Regression	✓	
k-Nearest Neighbors (KNN)	✓	✓
Decision Trees	✓	✓
Random Forest	✓	✓
Support vector machine (SVM)	✓	
Linear Regression		✓

Cuadro 1: Cuadros

Métricas Clasificación:

- Precisión.
- Recall (exhaustividad).
- Valor-F (F_1)

Ver video 03b min 40. Matriz de confucion.

1.1. Classification

En el contexto del aprendizaje automático supervisado, la variable predictora (también conocida como variable independiente o característica) es una variable que se utiliza para predecir el valor de la variable objetivo (también conocida como variable dependiente o target). Las variables predictoras son los datos de entrada al modelo y la variable objetivo es el resultado que se desea predecir.

1.1.1. Regresión logística.

La regresión logística es un tipo de algoritmo de aprendizaje supervisado que se utiliza principalmente para resolver tareas de "clasificación" binarias, aunque contiene la palabra "regresión". Se lo utiliza para predecir el resultado de una

variable categórica (una variable que puede adoptar un número limitado de categorías) en función de las **variables predictoras** o independientes. Un ejemplo de clasificación podría ser la detección de spam: un programa de aprendizaje automático puede aprender a marcar el spam después de recibir ejemplos de correos electrónicos spam marcados (**variable objetivo** "target" sería una variable binaria que indica si un correo electrónico es spam o no) por los usuarios y ejemplos de correos electrónicos regulares no spam (también llamados "ham"). Ejemplo Python [3].

Tambien existe la regresion logistica multinomial, que es una generalizacion de la regresion logistica. En este caso la variable objetivo puede tomar mas de dos valores. Ver regresion_logistica_02.ipynb

1.1.2. K-Nearest Neighbors (KNN).

Se utiliza para clasificación y regresión. k vecinos más cercanos es un método de clasificación no paramétrico. Video de youtube [4].
Se utiliza principalmente para la clasificación de datos no lineales y para resolver problemas de clasificación en los que los datos son muy complejos y desestructurados.

Resumen:

- Es sensible a conjuntos de datos no balanceados.
- Es muy sensible aoutliers.
- La normalización de los datos de entrenamiento puede mejorar drásticamente su precisión
- Si se aplica en un comjunto de datos desbalanceado, puede ser que el algoritmo siempre prediga la clase mayoritaria.

1.1.3. Decision Trees

Los árboles de decisión son modelos ampliamente utilizados para tareas de clasificación y regresión. Esencialmente, aprenden una jerarquía de preguntas if/else, que conducen a una decisión. Estas preguntas son similares a las preguntas que podría hacer en un juego de 20 preguntas. Imagina que quieres distinguir entre los siguientes cuatro animales: osos, halcones, pingüinos y delfines. Su objetivo es llegar a la respuesta correcta haciendo la menor cantidad posible de preguntas si/si no. Puede comenzar preguntando si el animal tiene plumas, una pregunta que reduce sus posibles animales a solo dos. Si la respuesta es "sí", puedes hacer otra pregunta que podría ayudarte a distinguir entre halcones y pingüinos. Por ejemplo, podría preguntar si el animal puede volar. Si el animal no tiene plumas, sus posibles opciones de animales son delfines y osos, y deberá hacer una pregunta para distinguir entre estos dos animales, por ejemplo, preguntar si el animal tiene aletas.

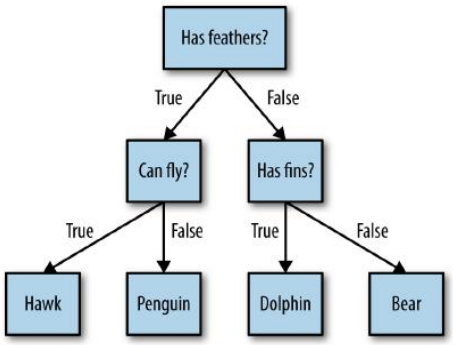

Figura 1: Ejemplo de un árbol de decisión.

Existen varios algoritmos ID3, C4.5 y CART. Pagina buena [2]:

• **ID3 - Iterative Dichotomiser 3:** Genera un árbol de decision a partir de un conjunto de ejemplos.
Este algoritmo usa las metricas de *entropía* y *ganancia de la información*. Cuando se usa la librería sklearm *sklearn.tree.DecisionTreeClassifier* en el parámetro *criterion* colocar *entropy*.

- Un nodo principal llamado raíz en la parte superior.
- Nodos terminales. como su nombre lo indica, son nodos donde termina el flujo y que ya no son raiz de ningún otro nodo. Estos nodos terminales deben contener una respuesta, o sea, la clasificación a que pertenece el objeto que ha conducido hasta él.
- Los demás nodos representan preguntas con respecto al valor de uno de los atributos.
- Las líneas nodos representan preguntas con respecto al valor de uno de sus atributos.
- Las líneas representa las posibles respuestas que los atributos pueden tomar.

Algoritmo básico

1. Calcular la entropía para todas las clases.
2. Calcular la entropía para cada valor posible de cada atributo.
3. Seleccionar el mejor atributo basado en la reducción de la entropía. usando el calculo de la ganancia de la información.
4. Iterar, para cada sub-nodo, Excluyendo el nodo raíz, que ya fue usado.

- **C4.5:** Utiliza la metrica *gini*.

  1. Mitiga el sobreajuste por que emplea inherentemente el proceso de poda de un solo paso.
  2. Funciona para datos discretos y continuos.
  3. Es útil para datos incompletos.

  Metricas:

  - **Entropía:**
    La medida del desorden o la medida de la pureza. Básicamente, es la medida de la impureza o aleatoriedad de los datos.

    Para calcular la entropia de n clases se utiliza la fórmula:

    $$H(S) = \sum_{i=1}^{n} -p_i \cdot \log_2(p_i) \tag{1}$$

    Dónde:

    - S: es una lista de valores posibles.
    - p_i: es la probabilidad de los valores.
    - i: cada uno de los valores.

    Importante:

    - Para una muestra homogénea la entropia es igual a cero 0. Si no existe aleatoriedad, es decir, una moneda cargada.
    - La máxima entropía viene dada por $\log_2(n)$, n son los posibles valores de salida. Si $n = 2$ (true o false) entonces, la máxima entropía es 1. O sea es la máxima incertidumbre, ejemplo moneda equilibrada.

  - **Ganancia de la Información:** La ganancia de la información se aplica a cuantificar qué característica, de un conjunto de datos dados, proporciona la máxima información sobre la clasificación.

1.1.4. Random Forest - Bosques aleatorios.

Usa una técnica, o meta-algoritmo llamado Bootstrap aggregating. Se crean m tablas reducidas en atributos y para cada una de ellas entrenamos un árbol.

1.1.5. Support Vector Machines (SVM)

Es un algoritmo de aprendizaje supervisado que se puede utilizar para problemas de clasificación o regresión. El algoritmo SVM utiliza un hiperplano para separar los datos en clases. El hiperplano se selecciona de tal manera que maximiza la distancia entre los puntos de datos de las clases. El hiperplano se puede utilizar para clasificar nuevos puntos de datos.

Ventajas:

- Efectivo en espacios de alta dimensión.
- Efectivo en casos en que el número de dimensiones es mayor que el número de muestras.
- Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamados vectores de soporte), por lo que también es eficiente en memoria.
- Versátil: se pueden especificar diferentes funciones del núcleo para la función de decisión. Se proporcionan núcleos comunes, pero también es posible especificar núcleos personalizados.

Desventajas:

- Si el número de características es mucho mayor que el número de muestras, evite el exceso de ajuste al elegir las funciones del núcleo y el término de regularización es crucial.
- Los SVM no proporcionan directamente estimaciones de probabilidad, estas se calculan utilizando una validación cruzada de cinco veces.

Parámetros:

- **C:** Parámetro de regularización. El parámetro C controla el comercio entre el ajuste de los datos de entrenamiento y la suavidad de la superficie de decisión. Un C alto significa que el clasificador intentará ajustar los datos de entrenamiento lo mejor posible, mientras que un C bajo significa que el clasificador buscará una superficie de decisión que esté lo más suave posible.
- **kernel:** Especifica el tipo de kernel que se utilizará en el algoritmo. Debe ser uno de 'lineal', 'poli', 'rbf', 'sigmoid', 'precomputed' o una llamada a un kernel personalizado.
- **degree:** Grado de la función del núcleo polinomial ('poly'). Ignorado por todos los demás núcleos.
- **gamma:** Coeficiente para 'rbf', 'poly' y 'sigmoide'. Si gamma es 'auto', entonces 1 / n_features se utilizará en su lugar.

- **coef0:** Término independiente en función del núcleo. Solo es significativo en 'poly' y 'sigmoide'.
- **probability:** Habilita la estimación de la probabilidad. Debe estar habilitado antes de llamar a fit, y se basa en una validación cruzada de cinco veces. Deshabilitarlo puede acelerar el cálculo cuando se usa svm en grandes conjuntos de datos.
- **shrinking:** Habilita o deshabilita el encogimiento heurístico de la función de decisión. Debe estar habilitado para el uso de la validación cruzada de probabilidad. Deshabilitarlo puede dar una pequeña ganancia de rendimiento.

Linealmente separable: es un conjunto de datos que se puede separar en dos grupos distintos de manera que no haya puntos de datos que se superpongan entre los dos grupos.

1.2. Regresión

Los modelos que existe para la regresión son:

- Regresión lineal y multiple: *sklearn.linear_ model.Regresión Lineal*
- Regresión polinómica: *sklearn.preprocessing.PolynomialFeatures*

1.2.1. Regresión Lineal

La regresión lineal es un método estadístico que se utiliza para estudiar la relación entre una variable dependiente (también conocida como variable objetivo) y una o más variables independientes (también conocidas como variables predictoras). El objetivo de la regresión lineal es encontrar la línea (recta) que mejor se ajuste a los datos y pueda utilizarse para hacer predicciones.

$$y = a \cdot x + b \tag{2}$$

1.2.2. Regresión Lineal Múltiple

Consiste en predecir una respuesta numérica y en base a múltiples variables predictoras $x_1, x_2, ..., x_n$, suponiendo una relación lineal.

$$y = a_1 \cdot x_1 + a_2 \cdot x_2 + ... + a_n \cdot x_n + b \tag{3}$$

1.3. Métricas Regresión:

Las métricas de evaluación son medidas utilizadas para evaluar el rendimiento de un modelo de aprendizaje automático. En el caso de los problemas de regresión en el aprendizaje supervisado, estas métricas nos ayudan a determinar qué tan bien nuestro modelo está haciendo predicciones cuantitativas, como

valores continuos.

Metricas para la regresión:

- Raíz del error cuadrático medio **(RMSE)**.
- Error absoluto medio **(MAE)**.
- Error cuadratico medio **(MSE)**.
- Suma Residual de los cuadrados **RSS**.

Para cada una de las métricas:

- m : número de instancias
- h : Funcion hipotesis, es el modelo entrenado. En este caso regresion lineal.
- x : Todos los valores de entrada, todas las columnas.

Definición 1.1. (Error cuadrático medio **(MSE)**)

$$MSE(X,h) = \frac{1}{m} \cdot RSS = \frac{1}{m} \cdot \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)})^2 \tag{4}$$

Definición 1.2. (Error absoluto medio **(MAE)**

$$MAE(X,h) = \frac{1}{m} \cdot \sum_{i=1}^{m} |h(x^{(i)} - y^{(i)}| \tag{5}$$

Definición 1.3. (Raíz del error cuadrático medio **(RMSE)**

$$RMSE(X,h) = \sqrt{\frac{1}{m} \cdot \sum_{i=1}^{m} (h(x^{(i)} - y^{(i)}))^2} \tag{6}$$