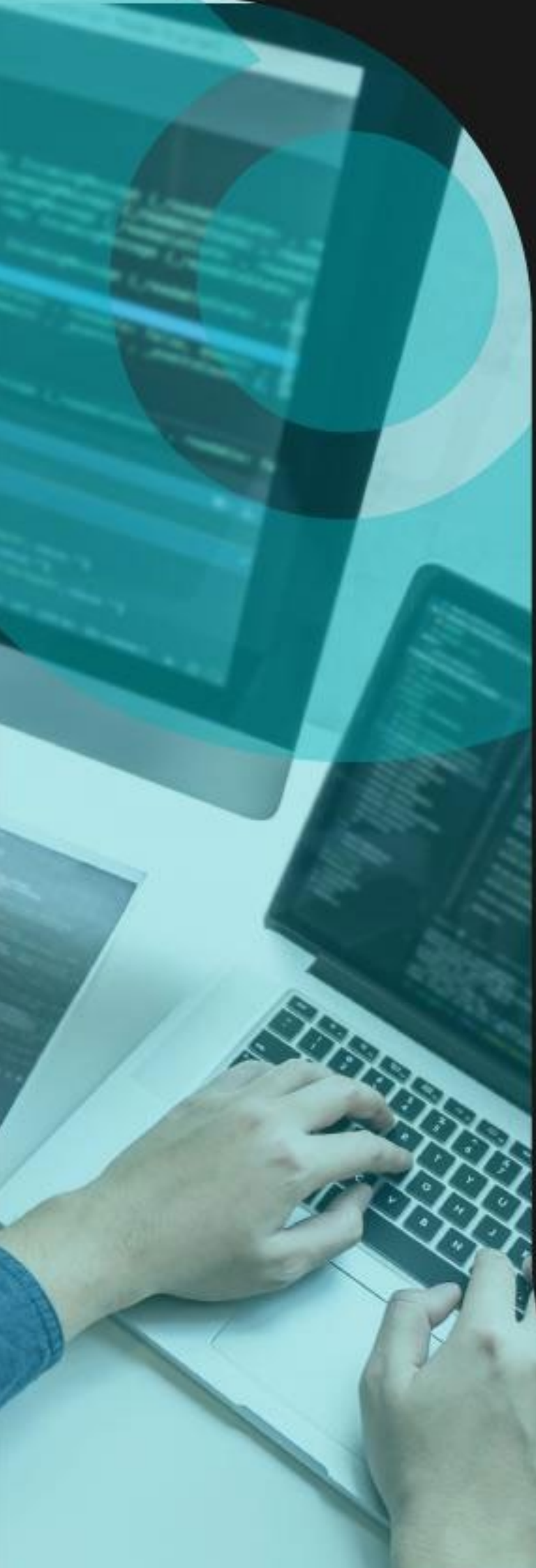


AWAKELAB

BASECAMP

Ciencia de Datos

Módulo: Aprendizaje de Máquina Supervisado

Aprendizaje Esperado

---

2. Elaborar un modelo predictivo de regresión lineal simple utilizando técnicas de validación cruzada y el lenguaje Python para resolver un problema.

Algoritmos regresivos

¿Qué son?

Son conocidos también como **forward-backward algorithm** son utilizados en algoritmos de inferencia para modelos de Markov (Clic aquí para saber más acerca de esto). La programación de este algoritmo es de forma dinámica, pues optimiza los valores de las distribuciones marginales, la metodología es en dos partes: primero avanza en el tiempo mientras que en la segunda parte retrocede en el tiempo; de ahí el nombre de algoritmo de avance-retroceso .

El término algoritmo de avance-retroceso también se utiliza para referirse a cualquier algoritmo que pertenezca a la clase general de algoritmos que operan en modelos de secuencia de manera de avance-retroceso.

Para esta sección nos interesará la relación entre dos o más variables, pero ¿dónde se puede usar esta relación?.

Se suele utilizar en las Ciencias Sociales con el fin de determinar si existe, o no, **relación causal** entre una variable dependiente (Y) y un conjunto de otras variables explicativas (X). Asimismo, el modelo busca determinar cuál será el impacto sobre la variable Y ante un cambio en las variables explicativas (X).

Cuándo se requieren un algoritmo regresivo

Por ejemplo, un economista podría estar interesado en determinar la relación entre el ingreso de los trabajadores y su nivel de educación. Para esto, podría llevar a cabo un modelo de regresión en el que la *variable independiente (Y), será el ingreso del trabajador. En cuanto a las variables explicativas (X), se deben incluir todas aquellas que podrían explicar el ingreso entre las que se encuentran por supuesto la educación, la experiencia, la educación de los padres, etc.*

Algoritmo autorregresivo: Regresión lineal

Supongamos que interesa pronosticar el resultado de una variable Y, basándose en valores de otras variables, $X_1, \ldots, X_n$.

Supongamos que tenemos una medición continua. Nos interesa pronosticar, mediante una combinación lineal de variables, **preferentemente independientes entre ellas,** una variable de interés Y.

La recta o plano que forma esta combinación lineal representa el valor esperado de Y, condicionado a la información $X_1, \ldots, X_k$. La estimación de los coeficientes que ponderan cada variable se obtienen mediante **mínimos cuadrados**.

¿Cuál es el método de los mínimos cuadrados?

El método de mínimos cuadrados es una forma de análisis de regresión matemática que se utiliza para determinar la línea de mejor ajuste para un conjunto de datos, proporcionando una demostración visual de la relación entre los puntos de datos. Cada punto de datos representa la relación entre una variable independiente conocida y una variable dependiente desconocida.

El caso más simple es denominado Modelo de Regresión Lineal Simple, que es cuando se utiliza una variable predictora, denominada X, para pronosticar una variable respuesta Y. El modelo está dado por:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \text{ con } \epsilon_i \overset{\text{iid}}{\sim} Normal(0, \sigma^2)$$

En donde:

- β_0 es el valor de la variable Y cuando la variable X vale 0.
- β_1 es el cambio en la variable Y cuando la variable X aumenta en una unidad.

Con lo anterior se concluye que $y_i|X = x_i \sim Normal(\beta_0 + \beta_1 x_i, \sigma^2)$ para todo i.

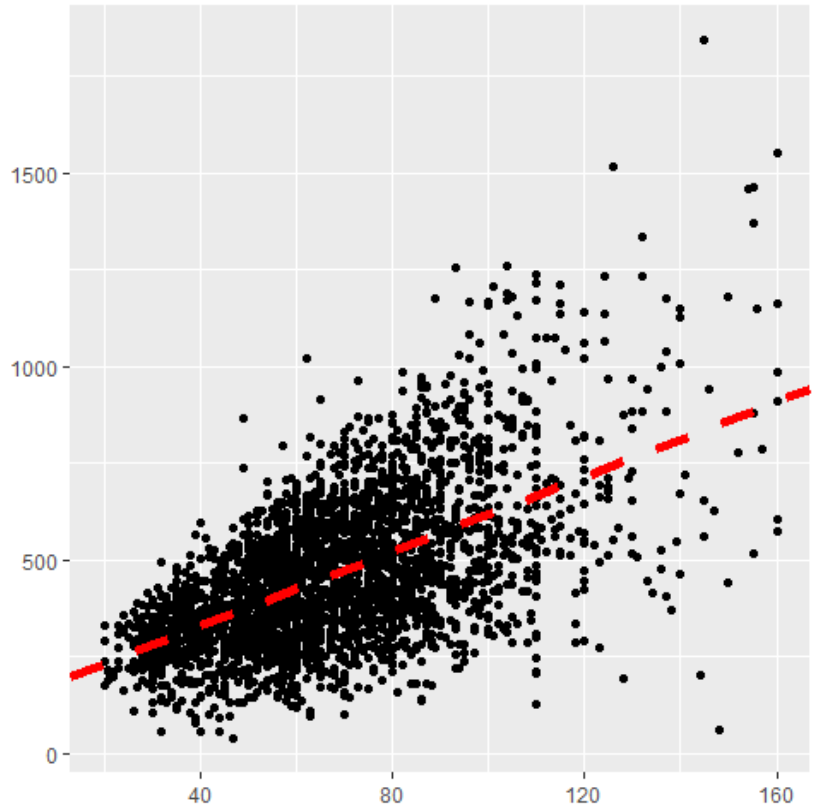
El ajuste de un modelo de regresión busca encontrar estimaciones para β_0 y β_1 a partir de una base de datos. Estas estimaciones se denotan con el superíndice gorro (^), esto es, $\hat{\beta}_0$ y $\hat{\beta}_1$. Con estos parámetros estimados se obtiene un modelo que nos permite obtener estimaciones para y, las cuales se denotan por $\hat{y}$.

Regresión Lineal Simple

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

Los residuos del modelo para la $i-$ ésima observación se denotan por e_i, y corresponde a la diferencia entre el valor real y la predicción del modelo:

$$e_i = y_i - \hat{y}_i$$



¿Por qué lineal?

Dos preguntas importantes:

I) Probar si los parámetros obtenidos son significativos implica comprobar si la variable explica la respuesta.

$$H_0: \beta_i = 0 \; vs \; H_1: \beta_i \neq 0$$

Se realizará un test t individual para cada parámetro. Además, se puede obtener los intervalos de confianza de cada uno de ellos.

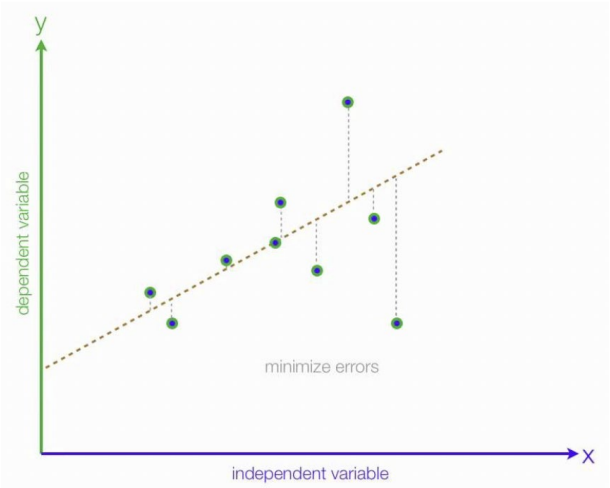
II) Probar si el modelo obtenido es significativo implica comprobar si el modelo es correcto.

$$H_0: \beta_0 = \beta_1 = \cdots = \beta_k = 0 H_1: Existen\ i, j: \beta_i \neq b_j$$

Se realizará un test F a partir de la tabla ANOVA del modelo. En este caso, se reemplaza la idea de Tratamiento por Regresión.

Métricas

El objetivo de la regresión lineal es encontrar una línea que minimice el error de predicción de todos los puntos de datos.



El paso esencial en cualquier modelo de aprendizaje automático es evaluar la precisión del modelo. Las métricas de error cuadrático medio, error absoluto medio, error cuadrático medio y R-cuadrado o coeficiente de determinación se utilizan para evaluar el rendimiento del modelo en el análisis de regresión.

1. El error cuadrático medio (Mean absolute error - MAE) representa el promedio de la diferencia absoluta entre los valores reales y predichos en el conjunto de datos. Mide el promedio de los residuos en el conjunto de datos.

$$MAE = \frac{1}{N}\sum_{i=1}^{n} |y_i - \hat{y}|$$

2. El error cuadrático medio (Mean Squared Error - MSE) representa el promedio de la diferencia cuadrática entre los valores original y pronosticado en el conjunto de datos. Mide la varianza de los residuos.

$$MSE = \frac{1}{N}\sum_{i=1}^{n} \quad (y_i - \hat{y})^2$$

3. La raíz del error cuadrático medio (Root Mean Squared Error - RMSE) es la raíz cuadrada del error cuadrático medio. Mide la desviación estándar de los residuos.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n} \quad (y_i - \hat{y})^2}$$

4. El coeficiente de determinación o R-cuadrado (R^2) representa la proporción de la varianza en la variable dependiente que es explicada por el modelo de regresión lineal. Es una puntuación sin escala, es decir, independientemente de que los valores sean pequeños o grandes, el valor de R cuadrado será menor que uno.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \quad (y_i - \hat{y})^2}{\sum_{i=1}^{n} \quad \left(y_i - \underline{y}\right)^2}$$

5. La R cuadrada ajustada es una versión modificada de la R cuadrada, y se ajusta por la cantidad de variables independientes en el modelo, y siempre será menor o igual que R^2. En la fórmula a continuación, **n** es la cantidad de observaciones en los datos y **k** es el número de variables independientes en los datos.

$$R_{adj}{}^2 = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1}\right]$$

Diferencias entre estas métricas de evaluación

- El error cuadrático medio (MSE) y el error cuadrático medio penalizan los grandes errores de predicción frente al error absoluto medio (MAE). Sin embargo, RMSE se usa mucho más que MSE para evaluar el rendimiento del modelo de regresión con otros modelos

aleatorios, ya que tiene las mismas unidades que la variable dependiente (eje Y).

- MSE es una función diferenciable que facilita la realización de operaciones matemáticas en comparación con una función no diferenciable como MAE. Por lo tanto, en muchos modelos, RMSE se usa como una métrica predeterminada para calcular la función de pérdida a pesar de ser más difícil de interpretar que MAE.
- MAE es más robusto a los datos con valores atípicos.
- El valor más bajo de MAE, MSE y RMSE implica una mayor precisión de un modelo de regresión. Sin embargo, se considera deseable un valor más alto de R cuadrado.
- R Squared y R Squared ajustado se utilizan para explicar qué tan bien las variables independientes en el modelo de regresión lineal explican la variabilidad en la variable dependiente. El valor de R Squared siempre aumenta con la adición de las variables independientes, lo que podría conducir a la adición de las variables redundantes en nuestro modelo. Sin embargo, el R-cuadrado ajustado resuelve este problema.
- La R cuadrada ajustada tiene en cuenta el número de variables predictoras y se utiliza para determinar el número de variables independientes en nuestro modelo. El valor de la R cuadrada ajustada disminuye si el aumento de la R cuadrada por la variable adicional no es lo suficientemente significativo.
- Para comparar la precisión entre diferentes modelos de regresión lineal, RMSE es una mejor opción que R Squared.

¿Cuál escoger?

Por lo tanto, si se compara la precisión de la predicción entre diferentes modelos de regresión lineal (LR), entonces RMSE es una mejor opción, ya que es simple de calcular y diferenciable. Sin embargo, si su conjunto de datos tiene valores atípicos, elija MAE sobre RMSE.

Además, el número de variables predictoras en un modelo de regresión lineal está determinado por R cuadrado ajustado, y elija RMSE sobre R cuadrado ajustado si le interesa evaluar la precisión de la predicción entre diferentes modelos LR.

Validación cruzada

La validación cruzada (CV) es una técnica muy útil para evaluar la efectividad del modelo, particularmente en los casos en que se necesita mitigar el sobreajuste. Consiste en una técnica para obtener una estimación más confiable del rendimiento del modelo utilizando solo sus datos de entrenamiento. También es útil para determinar los hiper parámetros del modelo, en el sentido de obtener un error de prueba más bajo.

El objetivo es predecir el valor de una variable para individuos (datos) de fuera de la muestra, con el menor error posible, a veces es inaccesible, ya que no se tiene datos fuera de la muestra.

Para estimar este error fuera de la muestra, se debe tomar la muestra actual:

- Elegir una parte como **conjunto de entrenamiento**
- Elegir otra parte como **conjunto de validación**
- **Ajustar el modelo** al conjunto de entrenamiento
- **Valorar el error** en el conjunto de validación

El mejor modelo será aquel que dé menor error en la parte de validación.

Problemas de regresión lineal simple

La regresión como herramienta ayuda a agrupar datos para ayudar a las personas y las empresas a tomar decisiones informadas. Hay diferentes variables en juego en la regresión, incluida una variable dependiente, la variable principal que está tratando de comprender, y una variable independiente, factores que pueden tener un impacto en la variable dependiente.

Para que el análisis de regresión funcione, debe recopilar todos los datos relevantes. Se puede presentar en un gráfico, con un eje x y un eje y.

Hay varias razones principales por las que las personas usan el análisis de regresión:

- Para predecir condiciones económicas futuras, tendencias o valores.
- Para determinar la relación entre dos o más variables.
- Entender cómo cambia una variable cuando cambia otra

Scikit-learn

Uso de la librería

Scikit-learn es una librería de código abierto que unifica bajo un único marco los principales algoritmos y funciones, facilitando en gran medida todas las etapas de preprocesado, entrenamiento, optimización y validación de modelos predictivos.

Para trabajar con algoritmos de *machine learning*, es imprescindible el uso de una fuente de datos a partir de la cual los métodos que vamos a emplear sean capaces de obtener los patrones e información oculta. Estos conjuntos de datos también se conocen como *dataset*.

Puesto que vamos a trabajar con Scikit-learn, usaremos alguno de los *dataset* que esta librería ya tiene integrado. Podemos echar un vistazo a los principales *dataset* que ya tiene integrada la librería. En este artículo vamos a realizar un ejemplo de sistema de clasificación usando el Iris dataset que ya está integrado dentro de la librería. Este dataset contiene ejemplos con cuatro atributos numéricos:

1. Longitud del sépalo en centímetros.
2. Anchura del sépalo en centímetros.
3. Longitud del pétalo en centímetros.
4. Anchura del pétalo en centímetros.

Cada uno de los ejemplos se clasifica en tres posibles tipos de Iris: Iris-Setosa, Iris-Versicolor e Iris-Virgínica. Utilizamos este *dataset* ya que es sencillo en el número de atributos y que todos los atributos son numéricos y por lo tanto no necesitan transformación alguna.

Aplicación en Python

En un primer paso necesitaremos instalar la librería scikit-learn en nuestro entorno virtual o notebook de Colab. Para ellos ejecutaremos el siguiente código:

```
pip install scikit-learn
```

El sistema nos informará de que todo es correcto y podremos acceder a la versión instalada mediante el código Python:

```
import sklearn as skl
from sklearn import linear_model

# Modelo de entrenamiento
lm = linear_model.LinearRegression()
```

Para importar el dataset que hemos elegido anteriormente debemos usar el siguiente código:

```
# Importamos el dataset

import sklearn.datasets
import pandas as pd

iris =
pd.read_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data', header=None)

iris.columns =
['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm','Species']
```

```
iris.sample(10)

dataset = sklearn.datasets.load_iris()
```

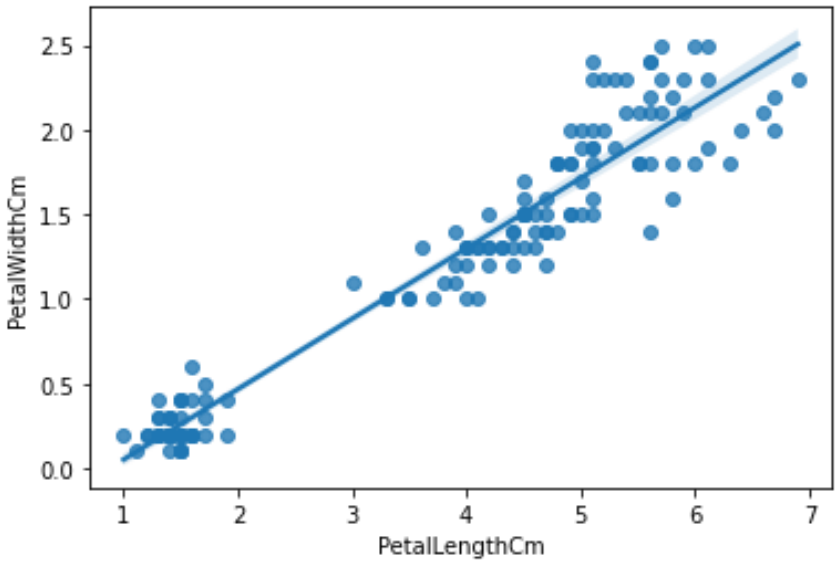
Dentro de la variable *dataset* tendremos la descripción del *dataset* (daaset.DESCR) una lista con el nombre de los atributos de cada una de la instancias (dataset.feature_names), la ruta del fichero que contiene el *dataset* (dataset.filename), una matriz con los valores de los atributos para cada una de las instancias (dataset.data), un array con el valor numérico de la clasificación objetivo (dataset.target) y el nombre de la clase para cada valor numérico (dataset.target_names). Con esto ya estamos listos para empezar a trabajar.

Una pre-visualización:

```
import seaborn as sns
%matplotlib inline

# A través del dibujo, puede hacer una observación
intuitiva sobre la relación lineal de los datos

sns.regplot(x='PetalLengthCm',y='PetalWidthCm',data=i
ris)
```



Set de datos de entrenamiento y testeo

Una de las principales cosas a tener en cuenta cuándo estamos preparando el dataset para realizar un entrenamiento es que deberemos separar las instancias o ejemplos del mismo en varios grupos. En los casos más simples será necesario formar al menos dos grupos de instancias: entrenamiento y test. Estos dos grupos se utilizan para fines diferentes. Así, el grupo de instancias de entrenamiento se usa para entrenar o ajustar el modelo matemático que estamos usando para resolver nuestro problema de aprendizaje automático; mientras que el grupo de test se usa para evaluar el desempeño del modelo con un conjunto de datos que no se hayan usado para entrenar.

Existen otras formas de separar un *dataset* para empezar un entrenamiento, pero lo dejaremos para artículos más avanzados. Generalmente hablando se suele separar el *dataset* en un 20% para test (o menos si el número de instancias del mismo es alto). Para realizar esta labor tenemos algunas funciones que nos ayudan en scikit-learn. Usaremos la variable X para referirnos a los atributos de las instancias y la variable y para referirnos a las respuestas que esperamos de nuestro sistema o etiquetas. Así para realizar la separación de nuestro *dataset* haremos lo siguiente:

```
from sklearn.model_selection import train_test_split

# Renombramos los valores para que X sean los
atributos e Y sean las respuestas del sistema

X = dataset.data
y = dataset.target

# Realizamos la partición de nuestro dataset en un
conjunto de entrenamiento y otro de test (20%)

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size = 0.2, random_state
= 42)
```

Modelo de entrenamiento

```
from sklearn import linear_model

lm = linear_model.LinearRegression()
features=['PetalLengthCm']

X=iris[features]
y=iris['PetalWidthCm']

print(X.shape,y.shape)

# Agregue dos características, X tendrá una dimensión
más
features=['PetalLengthCm','SepalLengthCm']

#Utiliza X, y para entrenar a la modelo
model = lm.fit(X,y)
```

```
print(model.intercept_,model.coef_)

#De los resultados de la impresión puede obtener la
intersección y el coeficiente del modelo de regresión
```

Evaluación del desempeño predictivo

Para evaluar el rendimiento del modelo obtenido, el conjunto de datos debe dividirse en un conjunto de entrenamiento y un conjunto de prueba, el modelo se aprende en el conjunto de entrenamiento y el error se evalúa en el conjunto de prueba

Divida las muestras en el conjunto de datos en varias partes iguales, una de las cuales se toma como conjunto de prueba cada vez y el resto de los datos como conjunto de entrenamiento. Utilice los datos del conjunto de prueba para evaluar y probar el modelo aprendido del conjunto de entrenamiento, es decir, realizar una verificación cruzada. En comparación con la división aleatoria, una parte del conjunto de entrenamiento siempre se divide en el conjunto de prueba. La verificación cruzada es dividir los datos en varias partes. Cada vez que se usa una parte diferente como conjunto de prueba, cada parte se usa como conjunto de prueba y conjunto de entrenamiento.

```
from sklearn.model_selection import cross_val_score

# Obtenga el error de 5 verificaciones cruzadas.
Tenga en cuenta que cross_val_score() está precedido
por un signo negativo, y se obtiene el error de valor
absoluto promedio de cada modelo de regresión

scores = -cross_val_score(lm, X, y, cv = 5, scoring =
'neg_mean_absolute_error')
print(scores)
```

```
# Promedio, como resultado del error

import numpy as np
print(np.mean(scores))

# Intente modificar los parámetros de puntuación para
obtener MSE

# Intente agregar y eliminar variables, ¿qué modelo
es mejor para la predicción de PetalWidthCm? ¿Qué
variables son útiles para predecir PetalWidthCm?
```

Referencias

[1] Modelos de regresión

https://economipedia.com/definiciones/modelo-de-regresion.html

[2] Librería Scikit-Learn de Python

https://aprendeia.com/libreria-scikit-learn-de-python/

[3] Métricas

https://www.google.com/search?q=M%C3%A9tricas+de+evaluaci%C3%B3n+del+modelo+regresivo+(MAE%2C+MSE%2C+RMSE)&rlz=1C5CHFA_enHU941HU941&oq=M%C3%A9tricas++de++evaluaci%C3%B3n++del++modelo++regresivo++(MAE%2C+MSE%2C+RMSE)&aqs=chrome..69i57.454j0j7&sourceid=chrome&ie=UTF-8

[4] Modelos de Markov

https://www.elsevier.es/es-revista-farmacia-hospitalaria-121-articulo-introduccion-utilizacion-modelos-markov-el-10017809

Material Complementario

[1] Algoritmos progresivos y regresivos

https://www.youtube.com/watch?v=-M6HC7lkIA8

[2] Regresión lineal

https://www.youtube.com/watch?v=SsFBnvkoZa4

[3] Validación cruzada

https://www.youtube.com/watch?v=82GsWZkZ7ss