# Ciencia de Datos

- **Módulo 2**

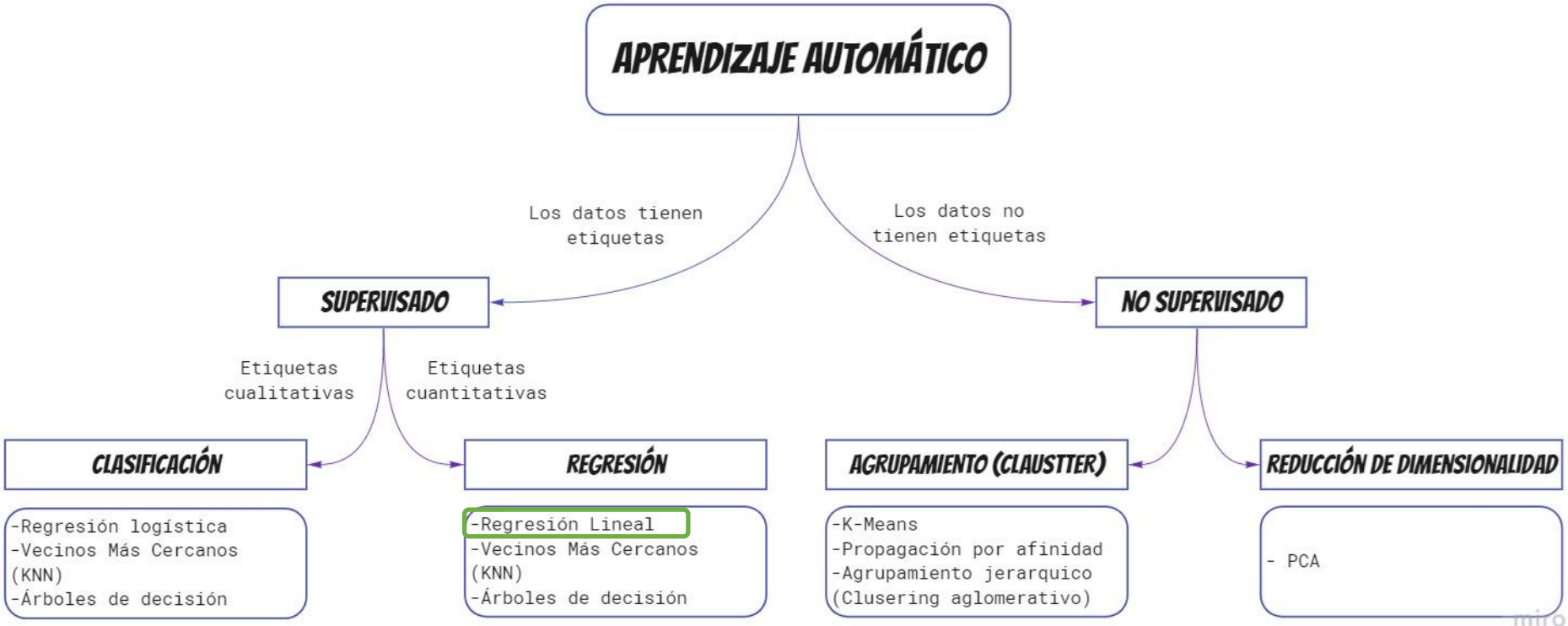
## Regresión lineal

UTNFra

GyEE Gestión y Evaluación Educativa

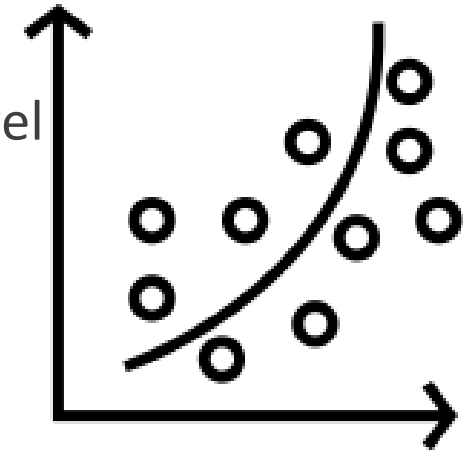
<aprendé programando Virtual>

SECUNDARIA del futuro


APRENDIZAJE AUTOMÁTICO
Los datos tienen etiquetas
Los datos no tienen etiquetas
SUPERVISADO
NO SUPERVISADO
Etiquetas cualitativas
Etiquetas cuantitativas
CLASIFICACIÓN
REGRESIÓN
AGRUPAMIENTO (CLAUSTTER)
REDUCCIÓN DE DIMENSIONALIDAD
-Regresión logística
-Vecinos Más Cercanos (KNN)
-Árboles de decisión
-Regresión Lineal
-Vecinos Más Cercanos (KNN)
-Árboles de decisión
-K-Means
-Propagación por afinidad
-Agrupamiento jerarquico (Clusering aglomerativo)
- PCA

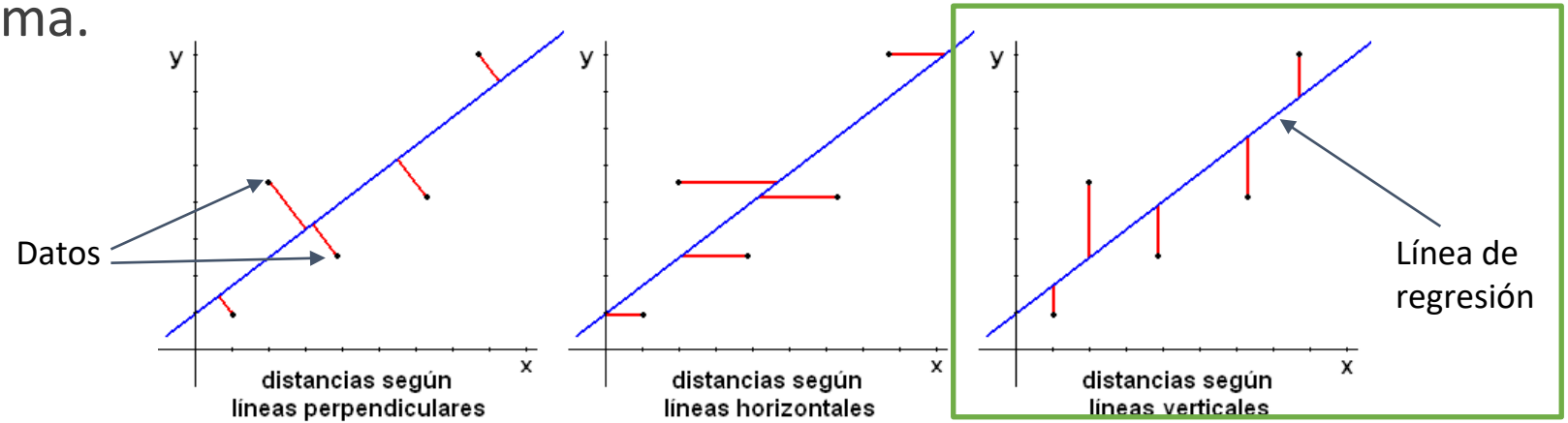
# Regresiones

Cuando nos referimos a un modelo de regresión, son modelos que tratan de predecir variables **numéricas** como precios, cantidades, pesos, etc. Se basan en estimar el valor de la variable a predecir en función de las otras variables observadas.
Por ejemplo, ver la probabilidad de lluvia dado el nivel de humedad y la velocidad del viento. Teniendo esta relación podemos saber cual es la probabilidad de lluvia en cualquier momento en el que tengamos el nivel de humedad y la velocidad del viento.



# Regresiones - Regresión lineal

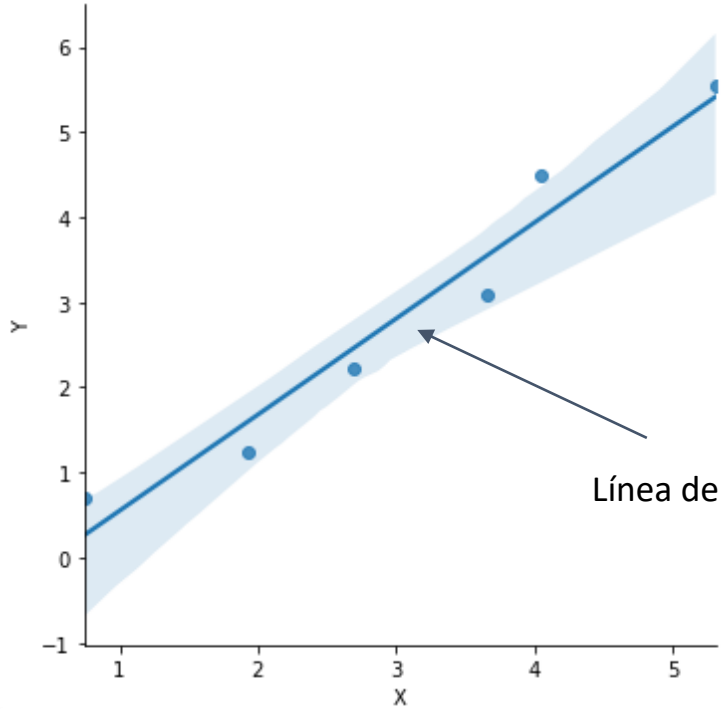
Es el modelo de regresión más básico. El objetivo es buscar la línea recta que mejor se ajuste a los datos conocidos. Esto se hace buscando la línea, tal que, la suma de las distancias de los datos a la línea sea mínima.



# Regresiones - Regresión lineal

Ejemplo:

| X | Y |
|---|---|
| 0.75 | 0.70 |
| 1.93 | 1.23 |
| 2.69 | 2.22 |
| 3.66 | 3.09 |
| 4.05 | 4.5 |
| 5.31 | 5.54 |




Línea de regresión

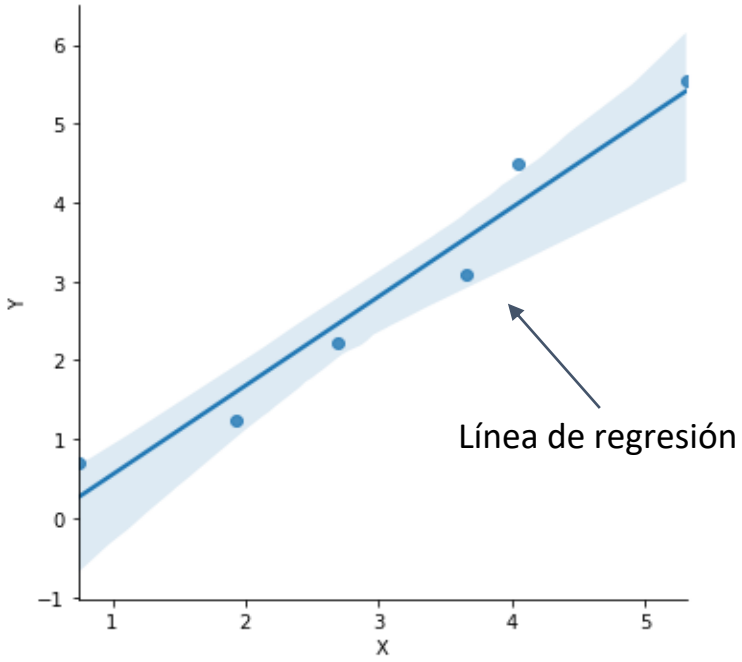
# Regresiones - Regresión lineal

Fórmula:

$$Y = \beta_0 + \beta_1.x$$

Intercepto
(punto origen en eje Y)

Pendiente de línea




| X | Y |
|---|---|
| 0.75 | 0.70 |
| 1.93 | 1.23 |
| 2.69 | 2.22 |
| 3.66 | 3.09 |
| 4.05 | 4.5 |
| 5.31 | 5.54 |

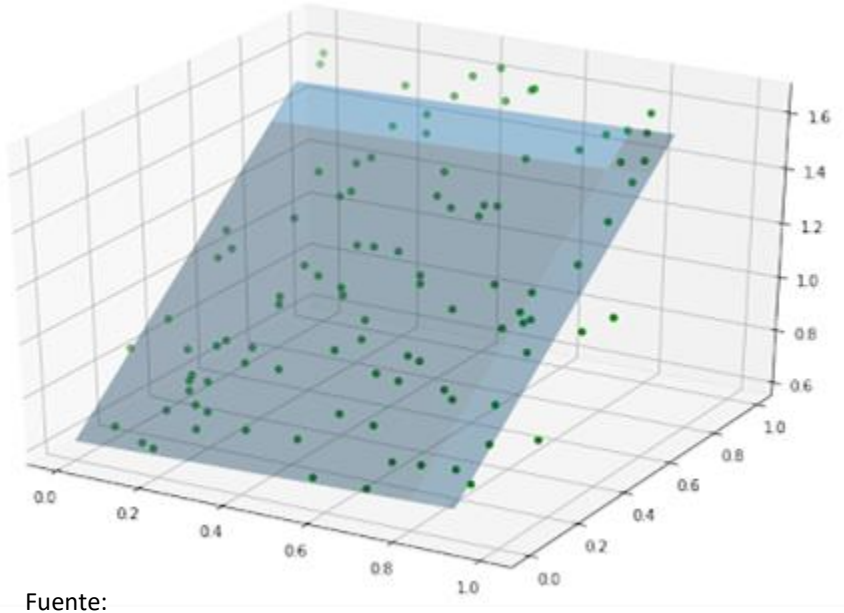
https://phet.colorado.edu/sims/html/least-squares-regression/latest/least-squares-regression_es.html

# Regresiones - Regresión lineal multiple

Fórmula:

$$Y = \beta_0 + \beta_1.x_1 + \beta_2.x_2$$

Parámetros

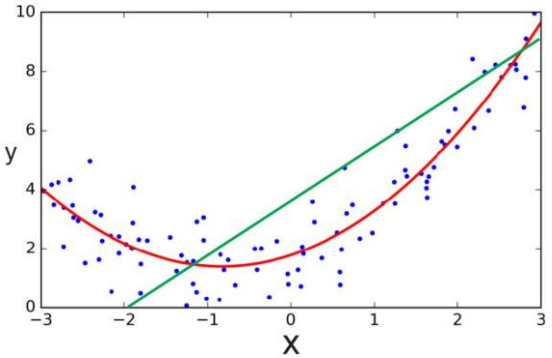


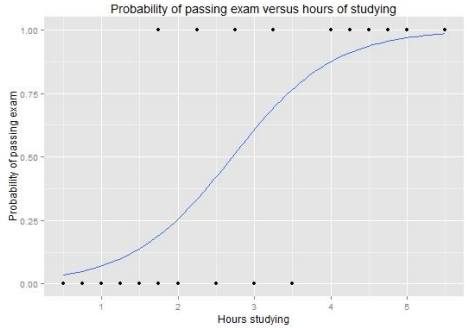

Fuente: http://personal.cimat.mx:8181/~mrivera/cursos/aprendizaje_maquina/regresion/regresion.html

# Regresiones - Otros tipos de regresiones

Así como las regresiones lineales existen varios tipos de regresiones según la forma de la línea a la cual se quieran ajustar los datos. Entre las más comunes está la **polinómica**, en la que, en vez de aproximar a una línea recta, buscamos un polinomio.

Otra regresión muy usada es la **logística**, en este caso la línea generada tiene una forma particular que nos ayuda a aproximar probabilidades de ocurrencia.

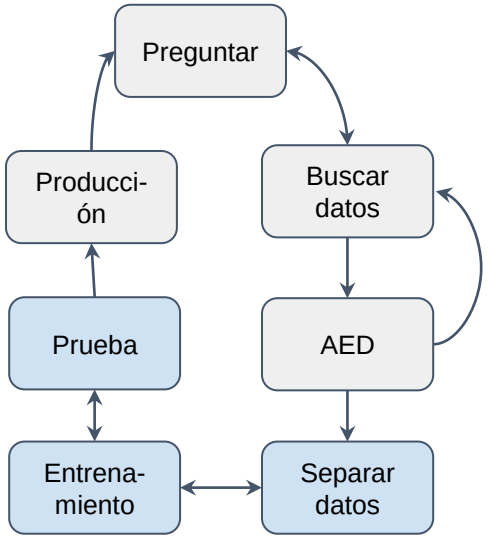


# Aprendizaje Automático - Pasos

En todo modelo de Machine Learning hay algunos **pasos comunes** que nos sirven de guia:

- **Definición del problema**: Siempre que se va a realizar un proyecto de ML es para resolver un problema específico, por ende, primero debemos de realizarnos una pregunta la cual buscaremos resolver.
- **Buscar los datos**: Por lo general debemos definir qué datos necesitamos y de dónde los podemos encontrar. Hoy en día hay muchos datos libres, sean generados por organismos gubernamentales u ONG, o en algunos casos incluso empresas privadas.
- **Análisis exploratorio de datos**: Durante esta etapa se *limpian* los datos y se realiza una exploración de los mismos, buscando características y relaciones que nos ayuden a seleccionar qué variables usar o que modelo usar.

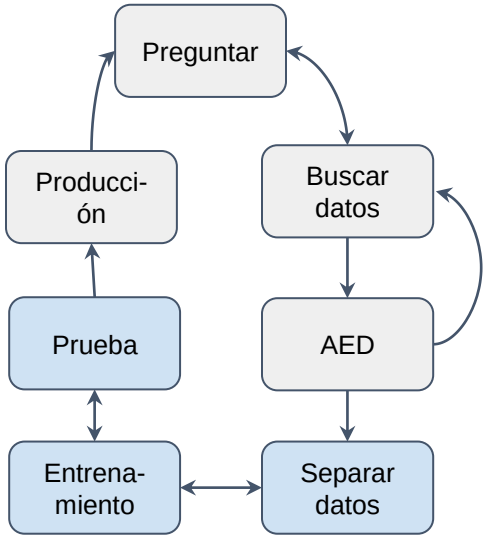


# Aprendizaje Automático - Pasos

- **Dividir los datos de entrenamiento y prueba**. Esta división es de vital importancia para poder definir si el modelo tiene un buen rendimiento o no.

- **Entrenamiento del modelo**: En esta etapa se **entrenan** diferentes de modelos y se **prueban** tomando en cuenta alguna métrica previamente definida, seleccionando el modelo que mejor desempeño tenga.

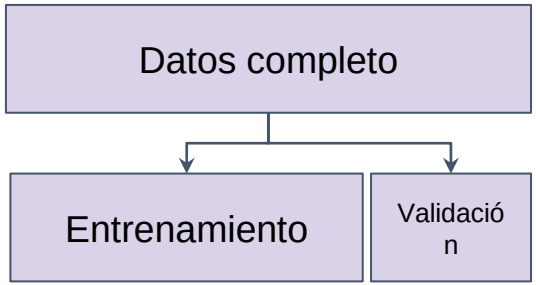
- **Puesta en producción**: Finalmente se realiza la puesta en producción, este paso depende de qué clase de proyecto sea, puede que en algunos casos, colocar en producción solo signifique generar las predicciones para alguna colección de datos, o puede ser implementarlo en un sistema que reciba datos periódicamente.



# Aprendizaje Automático - Pasos

Al entrenar un modelo de aprendizaje automático debemos poder **evaluarlo** para saber que tan bueno se comportara, para esto debemos simular su comportamiento cuando sea puesto en producción, esto lo logramos haciendo que genere estimaciones de datos que nunca antes haya visto el modelo, si, probamos el modelo con datos que fueron usados para entrenar es lo que se llama filtración de datos (data leak) y es considerado uno de los errores más grandes, pues, estaremos diciendo que nuestro modelo tiene un rendimiento que no necesariamente es cierto.

Por eso, lo primero que debemos hacer es **dividir los datos en dos conjuntos**, de entrenamiento y validación o prueba, la cantidad de datos que coloquemos en cada conjunto suele estar dada por un porcentaje del total de datos, y depende principalmente de cuantos datos tengamos, el conjunto de entrenamiento suele tener la mayoría de los datos, ya que, en general, los modelos de aprendizaje profundo necesitan una gran cantidad de datos para tener un buen desempeño. sin embargo, también debemos dejar una cantidad razonable de datos en el conjunto de validación para poder tener una aproximación realista del rendimiento del modelo. Generalmente las proporciones suelen ser de 75% de los datos para entrenar y 25% para validar.



# Definición Datos X e y

Se divide el dataset: creando dos DataFrames: variables que se utilizaran para predecir (X) y la variable a predecir (y)

X

y

| | humedad | velocidad_viento_kmh | rumbo_viento_grados | visibilidad_km | presion_mbar | lluvia | temperatura |
|---|---|---|---|---|---|---|---|
| 0 | 0.92 | 11.27 | 130.0 | 8.05 | 1021.60 | 0 | -0.56 |
| 1 | 0.73 | 20.93 | 330.0 | 16.10 | 1017.00 | 1 | 21.11 |
| 2 | 0.97 | 5.97 | 193.0 | 14.91 | 1013.99 | 1 | 16.60 |
| 3 | 0.82 | 3.22 | 300.0 | 16.10 | 1031.59 | 1 | 1.60 |
| 4 | 0.60 | 10.88 | 116.0 | 9.98 | 1020.88 | 1 | 2.19 |
| 5 | 0.32 | 21.46 | 190.0 | 10.35 | 1015.33 | 1 | 27.54 |
| 6 | 0.84 | 7.97 | 170.0 | 11.13 | 1009.04 | 1 | 19.98 |
| 7 | 0.86 | 14.49 | 30.0 | 15.13 | 1009.60 | 1 | 11.11 |
| 8 | 0.73 | 14.01 | 351.0 | 15.83 | 1018.39 | 1 | 8.41 |
| 9 | 0.81 | 6.44 | 320.0 | 7.86 | 1003.89 | 1 | 1.70 |
| 10 | 0.88 | 14.01 | 141.0 | 6.02 | 1021.28 | 0 | -2.22 |
| 11 | 0.60 | 1.42 | 204.0 | 15.83 | 1019.52 | 1 | 21.90 |
| 12 | 0.87 | 11.03 | 1.0 | 14.91 | 1015.92 | 1 | 17.11 |
| 13 | 0.73 | 4.07 | 297.0 | 9.76 | 1013.06 | 1 | 17.77 |
| 14 | 0.39 | 7.66 | 35.0 | 9.98 | 1025.59 | 1 | 24.95 |

X

y

| humedad | velocidad_viento_kmh | rumbo_viento_grados | visibilidad_km | presion_mbar | lluvia |
|---|---|---|---|---|---|
| 0.92 | 11.27 | 130.0 | 8.05 | 1021.60 | 0 |
| 0.73 | 20.93 | 330.0 | 16.10 | 1017.00 | 1 |
| 0.97 | 5.97 | 193.0 | 14.91 | 1013.99 | 1 |
| 0.82 | 3.22 | 300.0 | 16.10 | 1031.59 | 1 |
| 0.60 | 10.88 | 116.0 | 9.98 | 1020.88 | 1 |
| 0.32 | 21.46 | 190.0 | 10.35 | 1015.33 | 1 |
| 0.84 | 7.97 | 170.0 | 11.13 | 1009.04 | 1 |
| 0.86 | 14.49 | 30.0 | 15.13 | 1009.60 | 1 |
| 0.73 | 14.01 | 351.0 | 15.83 | 1018.39 | 1 |
| 0.81 | 6.44 | 320.0 | 7.86 | 1003.89 | 1 |
| 0.88 | 14.01 | 141.0 | 6.02 | 1021.28 | 0 |
| 0.60 | 1.42 | 204.0 | 15.83 | 1019.52 | 1 |
| 0.87 | 11.03 | 1.0 | 14.91 | 1015.92 | 1 |
| 0.73 | 4.07 | 297.0 | 9.76 | 1013.06 | 1 |
| 0.39 | 7.66 | 35.0 | 9.98 | 1025.59 | 1 |

| temperatura |
|---|
| -0.56 |
| 21.11 |
| 16.60 |
| 1.60 |
| 2.19 |
| 27.54 |
| 19.98 |
| 11.11 |
| 8.41 |
| 1.70 |
| -2.22 |
| 21.90 |
| 17.11 |
| 17.77 |
| 24.95 |

# División Datos de entrenamiento y testeo

Se divide el dataset en los datos que se utilizaran para entrenar el modelo (X_train e y_train) y los datos que se utilizaran para probar (X_test e y_test). Se utiliza la función de Scikit-Learn **train_test_split()**

X | y

Train / Test

| humedad | velocidad_viento_kmh | rumbo_viento_grados | visibilidad_km | presion_mbar | lluvia | temperatura |
|---|---|---|---|---|---|---|
| 0.92 | 11.27 | 130.0 | 8.05 | 1021.60 | 0 | -0.56 |
| 0.73 | 20.93 | 330.0 | 16.10 | 1017.00 | 1 | 21.11 |
| 0.97 | 5.97 | 193.0 | 14.91 | 1013.99 | 1 | 16.60 |
| 0.82 | 3.22 | 300.0 | 16.10 | 1031.59 | 1 | 1.60 |
| 0.60 | 10.88 | 116.0 | 9.98 | 1020.88 | 1 | 2.19 |
| 0.32 | 21.46 | 190.0 | 10.35 | 1015.33 | 1 | 27.54 |
| 0.84 | 7.97 | 170.0 | 11.13 | 1009.04 | 1 | 19.98 |
| 0.86 | 14.49 | 30.0 | 15.13 | 1009.60 | 1 | 11.11 |
| 0.73 | 14.01 | 351.0 | 15.83 | 1018.39 | 1 | 8.41 |
| 0.81 | 6.44 | 320.0 | 7.86 | 1003.89 | 1 | 1.70 |
| 0.88 | 14.01 | 141.0 | 6.02 | 1021.28 | 0 | -2.22 |
| 0.60 | 1.42 | 204.0 | 15.83 | 1019.52 | 1 | 21.90 |
| 0.87 | 11.03 | 1.0 | 14.91 | 1015.92 | 1 | 17.11 |
| 0.73 | 4.07 | 297.0 | 9.76 | 1013.06 | 1 | 17.77 |
| 0.39 | 7.66 | 35.0 | 9.98 | 1025.59 | 1 | 24.95 |

X_train

| humedad | velocidad_viento_kmh | rumbo_viento_grados | visibilidad_km | presion_mbar | lluvia |
|---|---|---|---|---|---|
| 0.92 | 11.27 | 130.0 | 8.05 | 1021.60 | 0 |
| 0.73 | 20.93 | 330.0 | 16.10 | 1017.00 | 1 |
| 0.97 | 5.97 | 193.0 | 14.91 | 1013.99 | 1 |
| 0.82 | 3.22 | 300.0 | 16.10 | 1031.59 | 1 |
| 0.60 | 10.88 | 116.0 | 9.98 | 1020.88 | 1 |
| 0.32 | 21.46 | 190.0 | 10.35 | 1015.33 | 1 |
| 0.84 | 7.97 | 170.0 | 11.13 | 1009.04 | 1 |
| 0.86 | 14.49 | 30.0 | 15.13 | 1009.60 | 1 |
| 0.73 | 14.01 | 351.0 | 15.83 | 1018.39 | 1 |
| 0.81 | 6.44 | 320.0 | 7.86 | 1003.89 | 1 |
| 0.88 | 14.01 | 141.0 | 6.02 | 1021.28 | 0 |

y_train

| temperatura |
|---|
| -0.56 |
| 21.11 |
| 16.60 |
| 1.60 |
| 2.19 |
| 27.54 |
| 19.98 |
| 11.11 |
| 8.41 |
| 1.70 |
| -2.22 |

X_test

| | | | | | |
|---|---|---|---|---|---|
| 0.60 | 1.42 | 204.0 | 15.83 | 1019.52 | 1 |
| 0.87 | 11.03 | 1.0 | 14.91 | 1015.92 | 1 |
| 0.73 | 4.07 | 297.0 | 9.76 | 1013.06 | 1 |
| 0.39 | 7.66 | 35.0 | 9.98 | 1025.59 | 1 |

y_test

| |
|---|
| 21.90 |
| 17.11 |
| 17.77 |
| 24.95 |

UTN F r a · GYE Gestión y Evaluación Educativa · aprendé programando Virtual · SECUNDARIA del futuro

# Regresión lineal

En Python usaremos la librería Scikit-Learn para aplicar los modelos de ML a nuestros datos.
Esta librería de código abierto nos permite entrenar y realizar predicciones en forma secuencial.

# Entrenamiento y testeo

## Entrenamiento del modelo

Se entrena el modelo con los datos de entrenamiento

X_train e y_train

X_train

| humedad | velocidad_viento_kmh | rumbo_viento_grados | visibilidad_km | presion_mbar | lluvia |
|---|---|---|---|---|---|
| 0.92 | 11.27 | 130.0 | 8.05 | 1021.60 | 0 |
| 0.73 | 20.93 | 330.0 | 16.10 | 1017.00 | 1 |
| 0.97 | 5.97 | 193.0 | 14.91 | 1013.99 | 1 |
| 0.82 | 3.22 | 300.0 | 16.10 | 1031.59 | 1 |
| 0.60 | 10.88 | 116.0 | 9.98 | 1020.88 | 1 |
| 0.32 | 21.46 | 190.0 | 10.35 | 1015.33 | 1 |
| 0.84 | 7.97 | 170.0 | 11.13 | 1009.04 | 1 |
| 0.86 | 14.49 | 30.0 | 15.13 | 1009.60 | 1 |
| 0.73 | 14.01 | 351.0 | 15.83 | 1018.39 | 1 |
| 0.81 | 6.44 | 320.0 | 7.86 | 1003.89 | 1 |
| 0.88 | 14.01 | 141.0 | 6.02 | 1021.28 | 0 |

y_train

| temperatura |
|---|
| -0.56 |
| 21.11 |
| 16.60 |
| 1.60 |
| 2.19 |
| 27.54 |
| 19.98 |
| 11.11 |
| 8.41 |
| 1.70 |
| -2.22 |

## Testeo del modelo

Corre el modelo con los datos de Testeo para ver las predicciones que realiza y se guardan los resultados.

Se compara la predicción del modelo con los datos reales (y_test)

X_test

| | | | | | |
|---|---|---|---|---|---|
| 0.60 | 1.42 | 204.0 | 15.83 | 1019.52 | 1 |
| 0.87 | 11.03 | 1.0 | 14.91 | 1015.92 | 1 |
| 0.73 | 4.07 | 297.0 | 9.76 | 1013.06 | 1 |
| 0.39 | 7.66 | 35.0 | 9.98 | 1025.59 | 1 |

Modelo

y_pred

22,94
16,68
19,52
22,6

Comparación
(testeo – métrica)

y_test

21.90
17.11
17.77
24.95