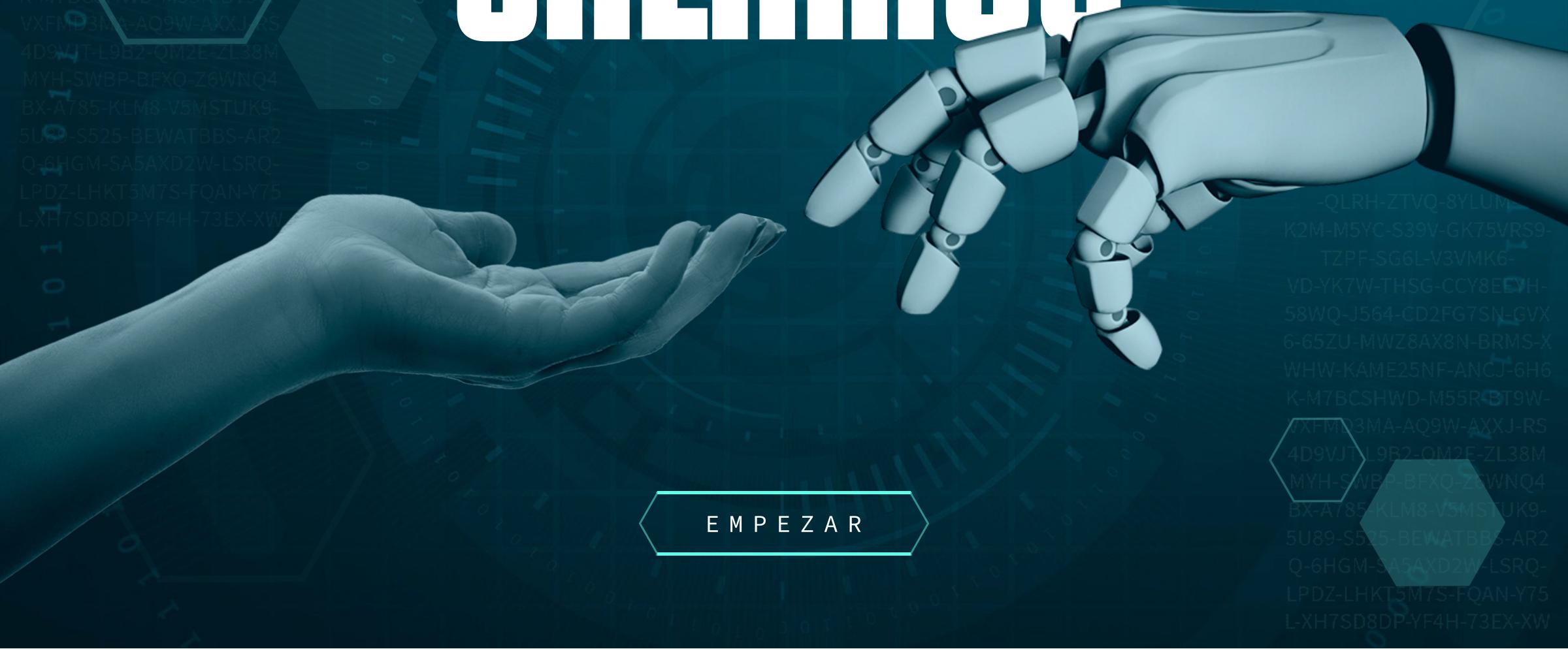


PREDICCIÓN

# SALARIOS



EMPEZAR

# ÍNDICE

O B J E T I V O S

M A C H I N E   L E A R N I N G

D A T O S

R E S U L T A D O S

P R O B L E M A S

C O N C L U S I O N E S

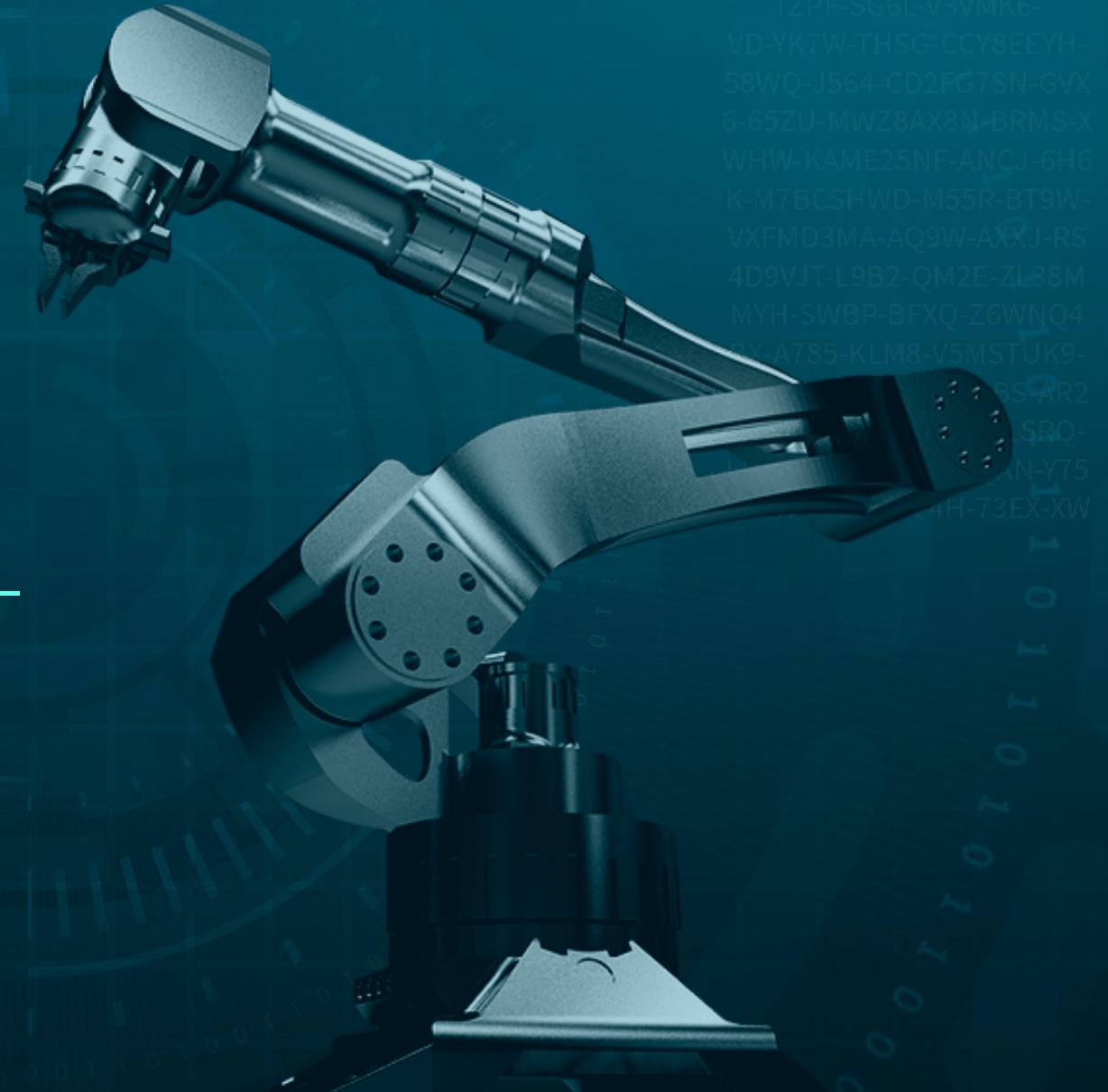
M Y S Q L

G R A C I A S

# OBJETIVO

---

Con una competición de kaggle **predecir el salario**, de diferentes puestos de trabajo relacionados con los datos, en dólares americanos.



# DATOS

En la competición de kaggle se nos proporciona tres csv. Un dataset de 500 filas con los datos completos (**train**); los datos de los que tenemos que predecir el salario, 107 filas, en dólares americanos (**test**); y un ejemplo del csv que hay que subir a la competición (**sample**).



# PROBLEMAS

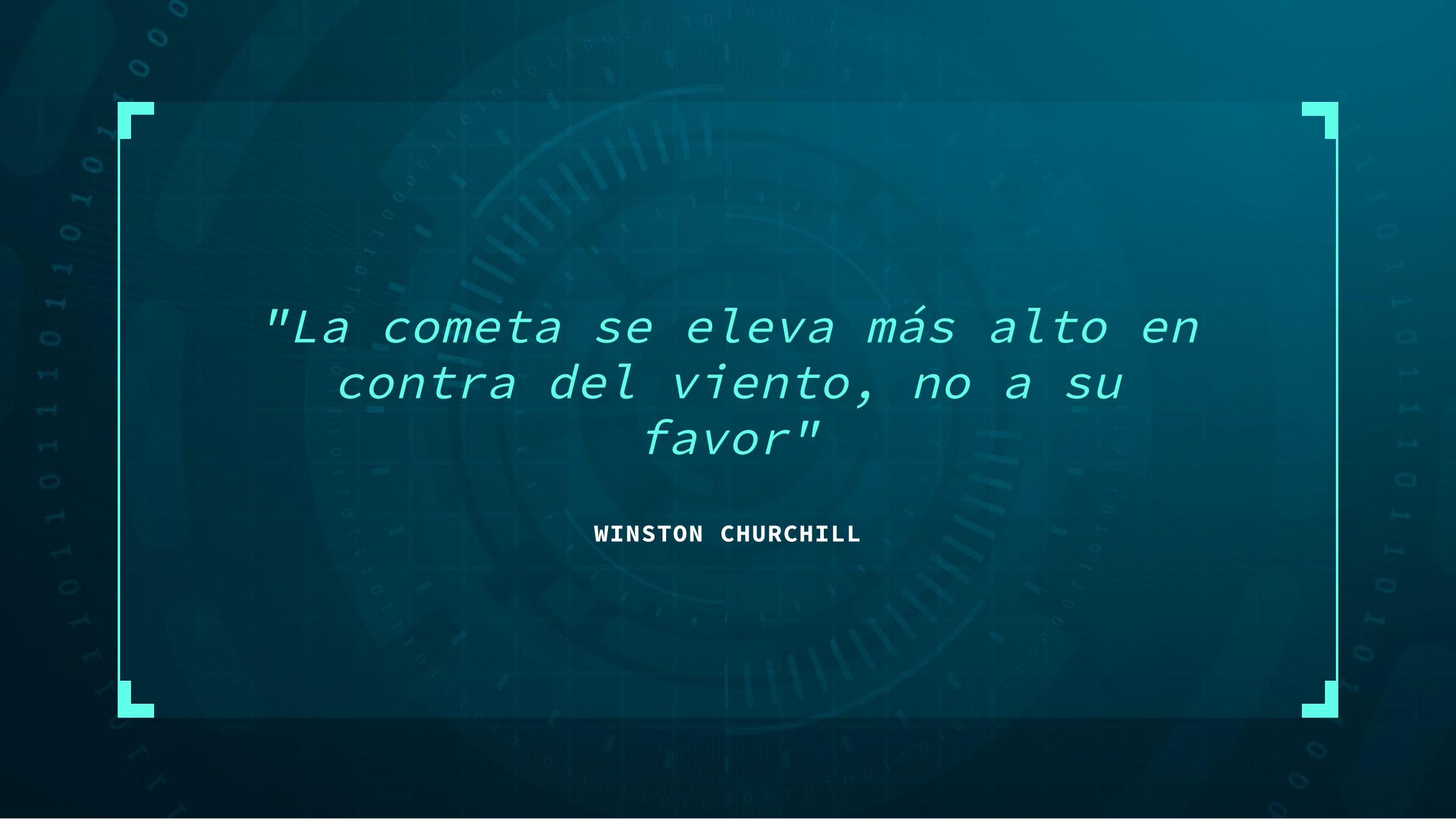
---

- De los 107 salarios que queremos predecir, **69 filas**, tienen el salario en dólares americanos.
- Los dataframes tienen 11 columnas, por lo que hay que establecer la **correlación** que tienen con el salario.
- Hay **valores únicos** en el dataframe de test que no están en el de train.
- Tenemos **pocos datos** con los que hacer el train.
- Qué consideramos una **buenas predicción** de los salarios, cuántos puntos son pocos para la competición.
- ¿Podemos únicamente usar **Machine Learning** con los datos de los que disponemos?

# ¿MACHINE LEARNING O MYSQL?



- Es como se ha **planteado** la resolución del proyecto.
- Supone días de trabajo, **es más lento**.
- Dónde está la línea entre una **buenas predicción** y una mala.
- Va en contra de lo esperado, **saltarse las normas** mola.
- Resultados inmediatos y **certeros**, consigue el fin que se propone.
- ¿Podemos conseguir un **error de cero**?



*"La cometa se eleva más alto en contra del viento, no a su favor"*

WINSTON CHURCHILL



# MySQL

- Creamos un **nuevo schema** e importamos los datos que tenemos.
- **Dos tablas:** datos de los salarios y el test.
- **Quitamos** todos los datos que estén en **dólares americanos**.
- Unimos las dos tablas con un **left join** para quedarnos con todas las filas de test. Creamos una **tabla temporal**.
- Esta consulta, nos arroja 16 resultados más de los esperados (38).

# MySQL

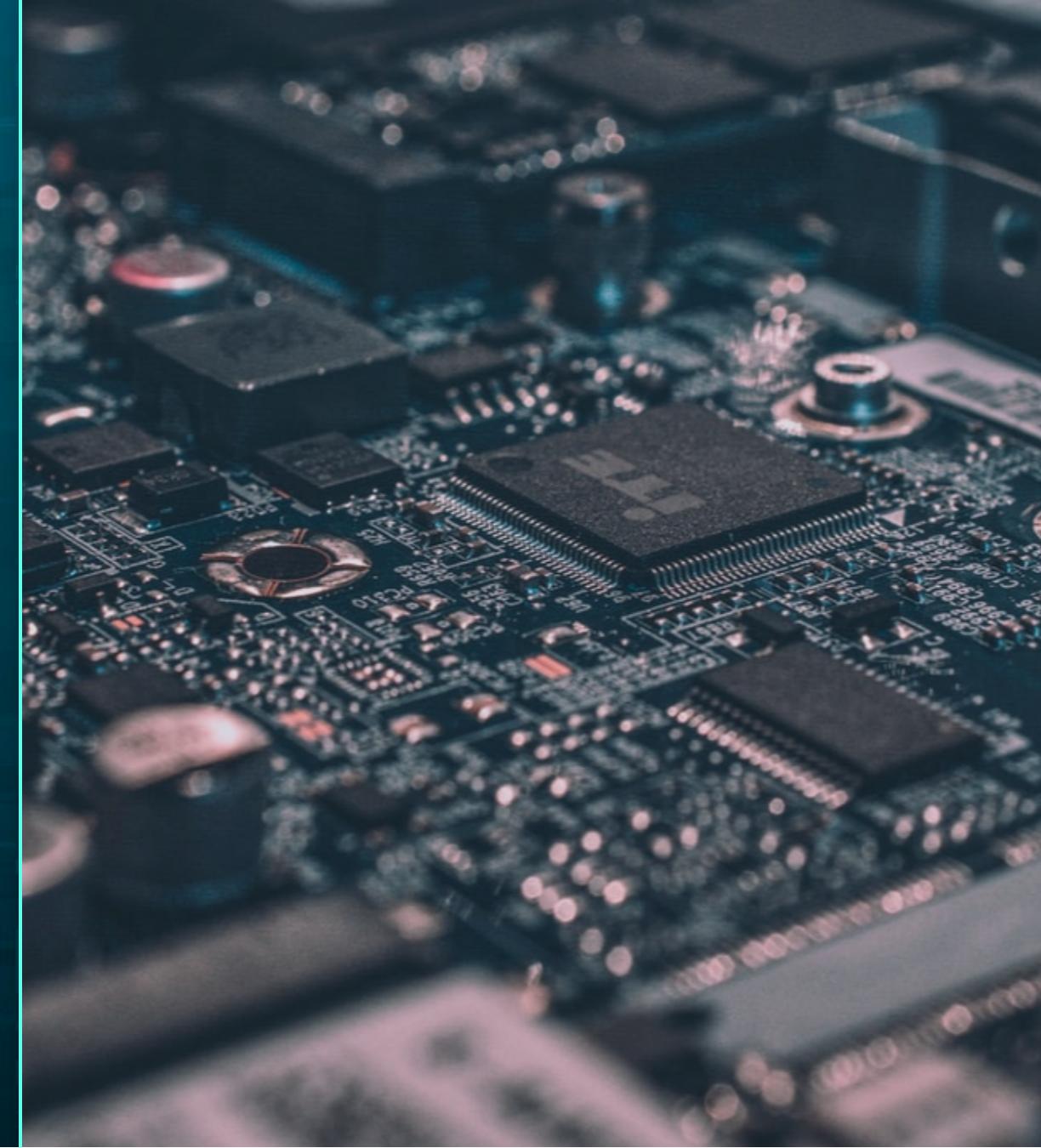
Seleccionamos el id de las filas a predecir que **nos dan más de un resultado** y ponemos más atención en estas.

Generamos **otra tabla temporal** de los datos que teníamos desde el inicio en dólares.

**Unimos las dos tablas** temporales y nos quedamos con las dos columnas que nos interesan (id, salary\_in\_usd).

Exportamos el resultado y lo subimos a **kaggle**.

La primera vez obtenemos **18k puntos**; y, finalmente, conseguimos cero puntos.



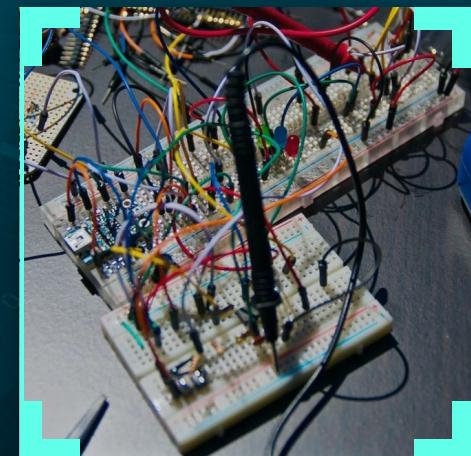
# MACHINE LEARNING

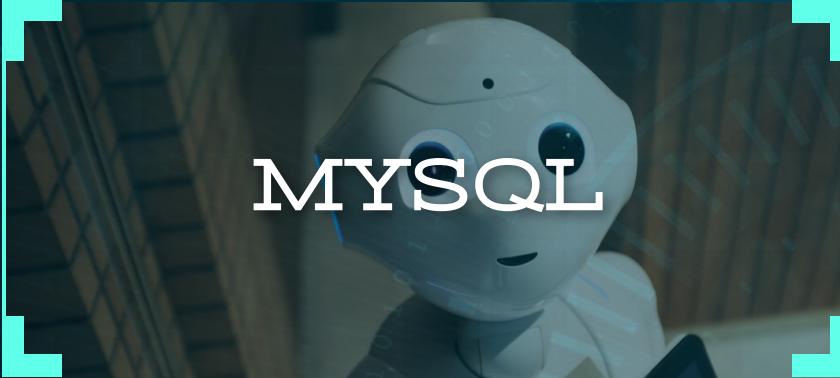
Evaluamos **qué método** es el más apropiado, con el que vamos a trabajar.

Calculamos la **correlación** de las columnas con el salario y eliminamos las que no nos interesan, el resto de columnas, las cambiamos a valores numéricicos.

Sacamos distintos datos cambiando **atributos de H2O** (max\_models, max\_runtime).

Calculando únicamente los datos que no estaban en USD, conseguimos una **puntuación de 18k**, intentando predecir todos los salarios (pese a ser innecesario en este caso), se consiguen **30k puntos**.

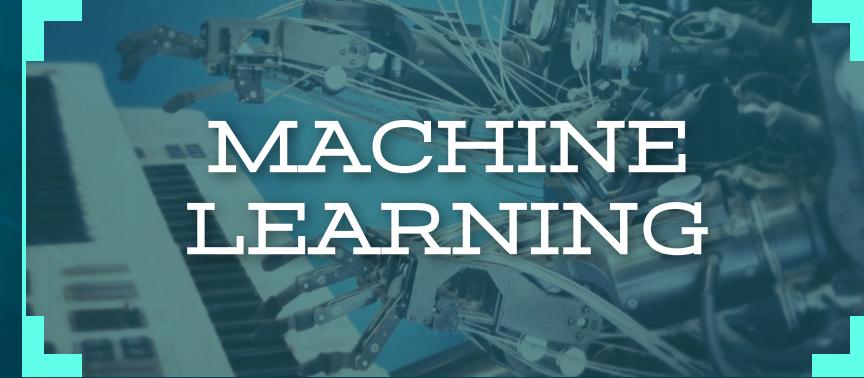




## MySQL

Con los datos que teníamos, hemos podido hacerlo con una herramienta que no fuera para predecir los salarios. Los resultados son muy buenos, pero el trabajo realizado es en gran medida irreutilizable, ya que no estamos haciendo una predicción.

VS



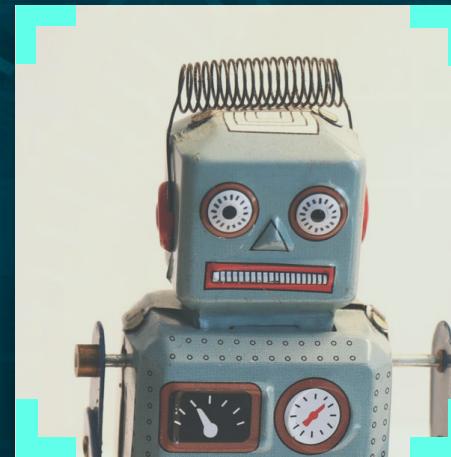
## MACHINE LEARNING

Los datos son peores que con MySQL, pero son datos sacados con una predicción. Tendríamos que preguntarnos si 30k de error es mucho o poco teniendo en cuenta el rango de salarios que hay.

# CONCLUSIÓN

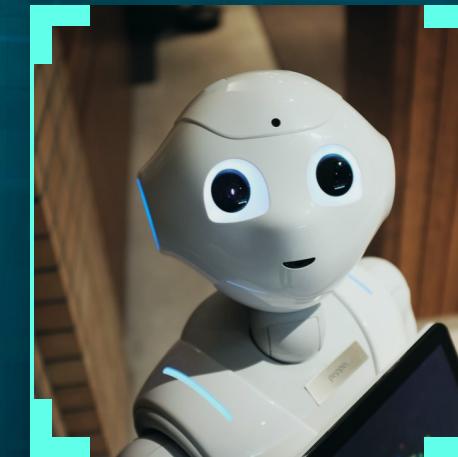
*La cabra siempre tira al monte*

# EQUIPO



**Jenn Atance**

No sabe vivir sin  
SQLite



**MackitoBook**

Colaboración  
necesaria





# iGRACIAS!