

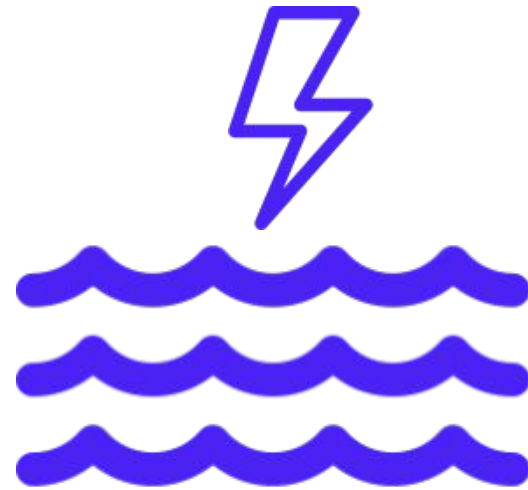
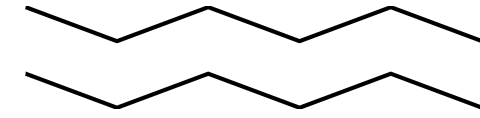


# Predicción del tamaño de olas usando Machine Learning

Andrés Cervantes

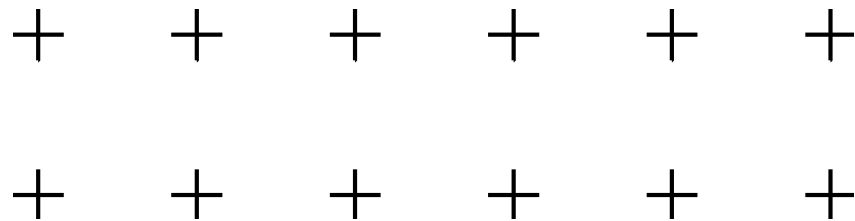
[github.com/andrescerv/BEDU-Data-Analysis](https://github.com/andrescerv/BEDU-Data-Analysis)

# Problema a solucionar

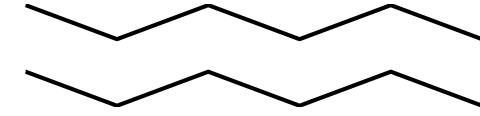


Surfear depende de condiciones meteorológicas.

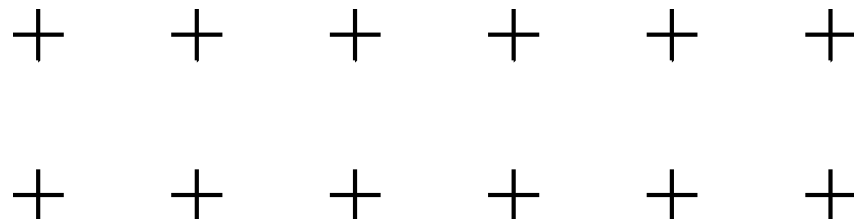
¿cómo se puede planear una sesión de surf si **las olas son altamente dependientes de factores naturales?**



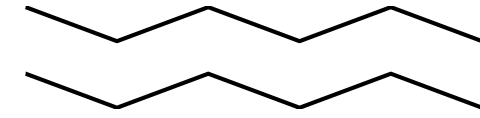
# • • • • Propuesta



**Predecir el tamaño y calidad  
de las olas a través de modelos  
de Machine Learning.**



# • • • • Usuarios potenciales

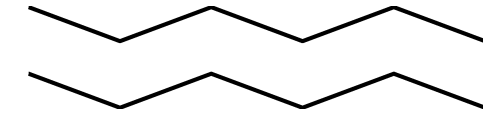


- Cliente objetivo:
  - cualquier *surfer*
  - organizadores de torneos
    - [World Surf League](#)
    - [Red Bull](#)
  - agencias de viaje
  - hoteles

+ + + + + +

+ + + + + +

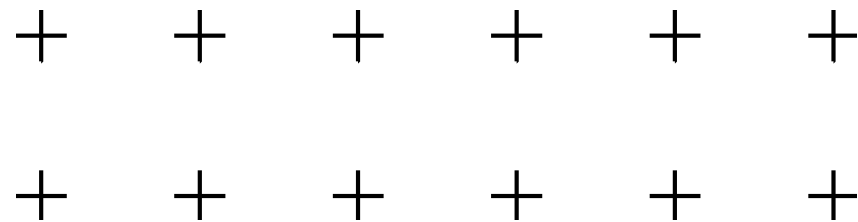
# ■ ■ ■ ■ Próximos pasos



Robustecer el modelo con más datos.

*Scrappear* fuentes de meteorología.

Predecir la *calidad* de la ola.



# 1. EDA



- ▪ ▪ ▪ **Existen 7 columnas, de las cuales nos interesan las últimas 5 para predecir la altura de las olas del set\*:**

```
waves_data.columns
```

```
Index(['time', 'wave_height', 'max_wave_height', 'zero_upcrossing_wave_period',  
      'peak_energy_wave_period', 'peak_direction', 'temperature'],  
      dtype='object')
```

- time: Time where the data was recorded (each 30 minutes), starting on 2020-01-01, ending on 2019-06-30.
- wave\_height: Significant wave height, an average of the highest third of the waves in a record
- max\_wave\_height: The maximum wave height in the record
- zero\_upcrossing\_wave\_period: The zero upcrossing wave period
- peak\_energy\_wave\_period: The peak energy wave period
- peak\_direction: Direction (related to true north) from which the peak period waves are coming from
- temperature: Approximation of sea surface temperature

+ + + + + +  
+ + + + + +

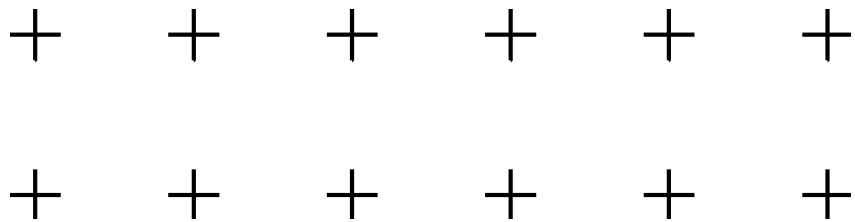
\* en este contexto, set se refiere al promedio de la altura del tercio de olas más grandes registradas en 30 minutos.



# Muestra del dataset:

```
waves_data_clean.tail()
```

	time	wave_height	max_wave_height	zero_upcrossing_wave_period	peak_energy_wave_period	peak_direction	temperature
43723	30/06/2019 21:30	2.299	3.60	9.281	12.765	94.0	21.95
43724	30/06/2019 22:00	2.075	3.04	9.303	12.722	95.0	21.95
43725	30/06/2019 22:30	2.157	3.43	9.168	12.890	97.0	21.95
43726	30/06/2019 23:00	2.087	2.84	8.706	10.963	92.0	21.95
43727	30/06/2019 23:30	1.926	2.98	8.509	12.228	84.0	21.95





■ ■ ■ ■

# Se tienen 43k registros limpios. La ola “típica” mide entre 0.7 y 1.7 metros.

```
waves_data_clean.describe()
```

	wave_height	max_wave_height	zero_upcrossing_wave_period	peak_energy_wave_period	peak_direction	temperature
count	43454.000000	43454.000000	43454.000000	43454.000000	43454.000000	43454.000000
mean	1.237799	2.090125	5.619685	9.011972	98.626594	23.949641
std	0.528608	0.897640	0.928533	2.390107	24.275165	2.231022
min	0.294000	0.510000	3.076000	2.720000	5.000000	19.800000
25%	0.839000	1.410000	4.981000	7.292000	85.000000	21.900000
50%	1.130000	1.900000	5.530000	8.886000	101.000000	23.950000
75%	1.544000	2.600000	6.166000	10.677000	116.000000	26.050000
max	4.257000	7.906000	10.921000	21.121000	358.000000	28.650000

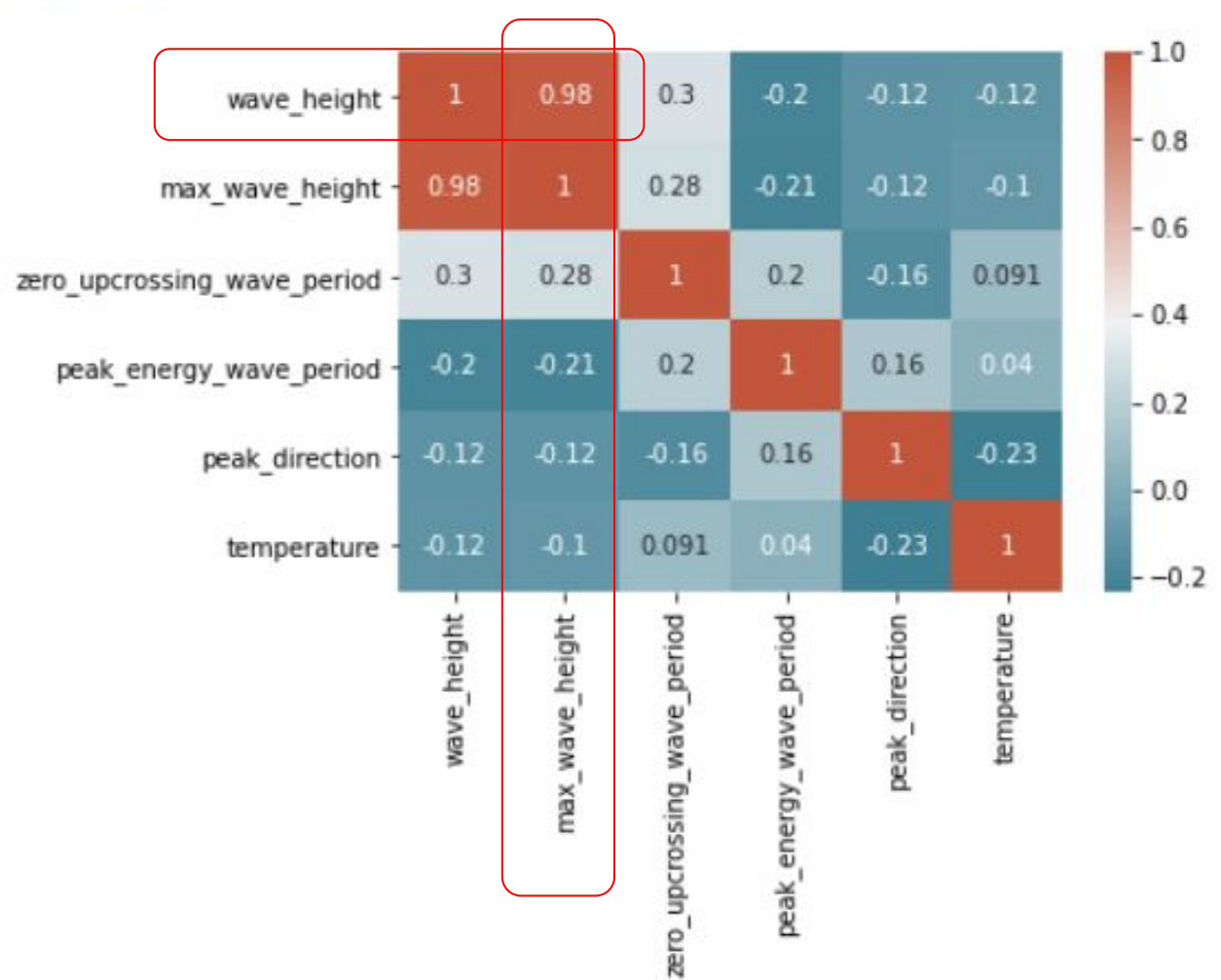
+ + + + + +

+ + + + + +

■ ■ ■ ■

La única correlación relevante es entre ‘wave\_height’ vs. ‘max\_wave\_height’ (0.98).

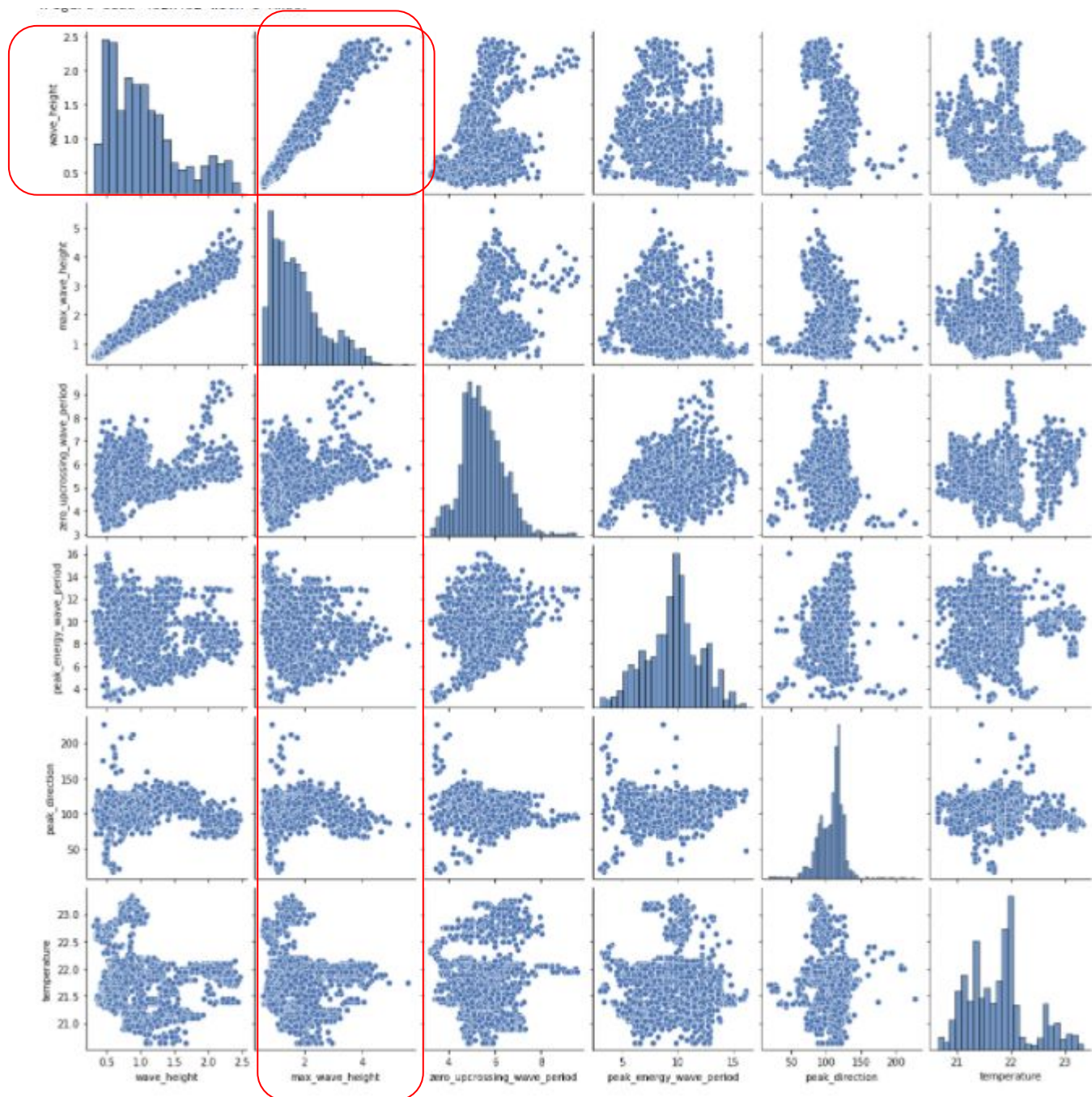
+ + + + + +  
+ + + + + +



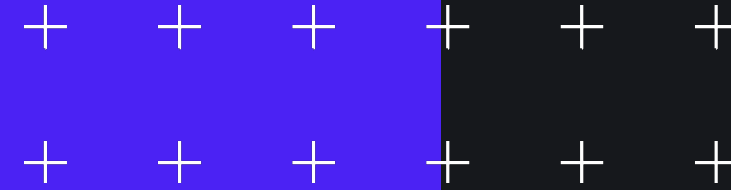
■ ■ ■ ■

La única correlación relevante es entre ‘wave\_height’ vs. ‘max\_wave\_height’ (0.98).

+ + + + +  
+ + + + +



## 2. Machine Learning





# Se usó Random Forest para predecir la altura de la ola:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error

y = waves_data_sample.wave_height
y_tr = y.iloc[:1295]
y_te = y.iloc[1295:]

features = ['zero_upcrossing_wave_period', 'peak_energy_wave_period', 'peak_direction', 'temperature']
X = waves_data_sample[features]
X_tr = X.iloc[:1295]
X_te = X.iloc[1295:]

# 1/4 select
rf_model = RandomForestRegressor(random_state=1)
# 2/4 fit
rf_model.fit(X_tr, y_tr)
# 3/4 predict
wave_height_preds = rf_model.predict(X_te)
# 4/4 validate
MAE = mean_absolute_error(y_te, wave_height_preds)
```

+ + + + + +  
+ + + + + +

■ ■ ■ ■

## Se usaron dos modelos:

**El primer modelo *no* considera la ola más grande del set:**

```
features = ['zero_upcrossing_wave_period', 'peak_energy_wave_period', 'peak_direction', 'temperature']
X = waves_data_sample[features]
X_tr = X.iloc[:1295]
X_te = X.iloc[1295:]
```

**El segundo modelo *sí* considera la ola más grande del set:**

```
features = ['max_wave_height', 'zero_upcrossing_wave_period', 'peak_energy_wave_period', 'peak_direction', 'temperature']
Z = waves_data_sample[features]
Z_tr = Z.iloc[:1295]
Z_te = Z.iloc[1295:]
```




+ + + + + +  
+ + + + + +

■ ■ ■ ■ ■  
**El segundo modelo reduce el error en un 74%.**

**El primer modelo tiene un error de 0.97 metros por predicción:**

```
features = ['zero_upcrossing_wave_period', 'peak_energy_wave_period', 'peak_direction', 'temperature']  
X = waves_data_sample[features]  
X_tr = X.iloc[:1295]  
X_te = X.iloc[1295:]
```

The Mean Absolute Error of the first model is 0.97 meters.



**El primer modelo tiene un error de 0.25 metros por predicción:**

```
features = ['max_wave_height', 'zero_upcrossing_wave_period', 'peak_energy_wave_period', 'peak_direction', 'temperature']  
Z = waves_data_sample[features]  
Z_tr = Z.iloc[:1295]  
Z_te = Z.iloc[1295:]
```

The Mean Absolute Error of the second model is 0.25 meters.

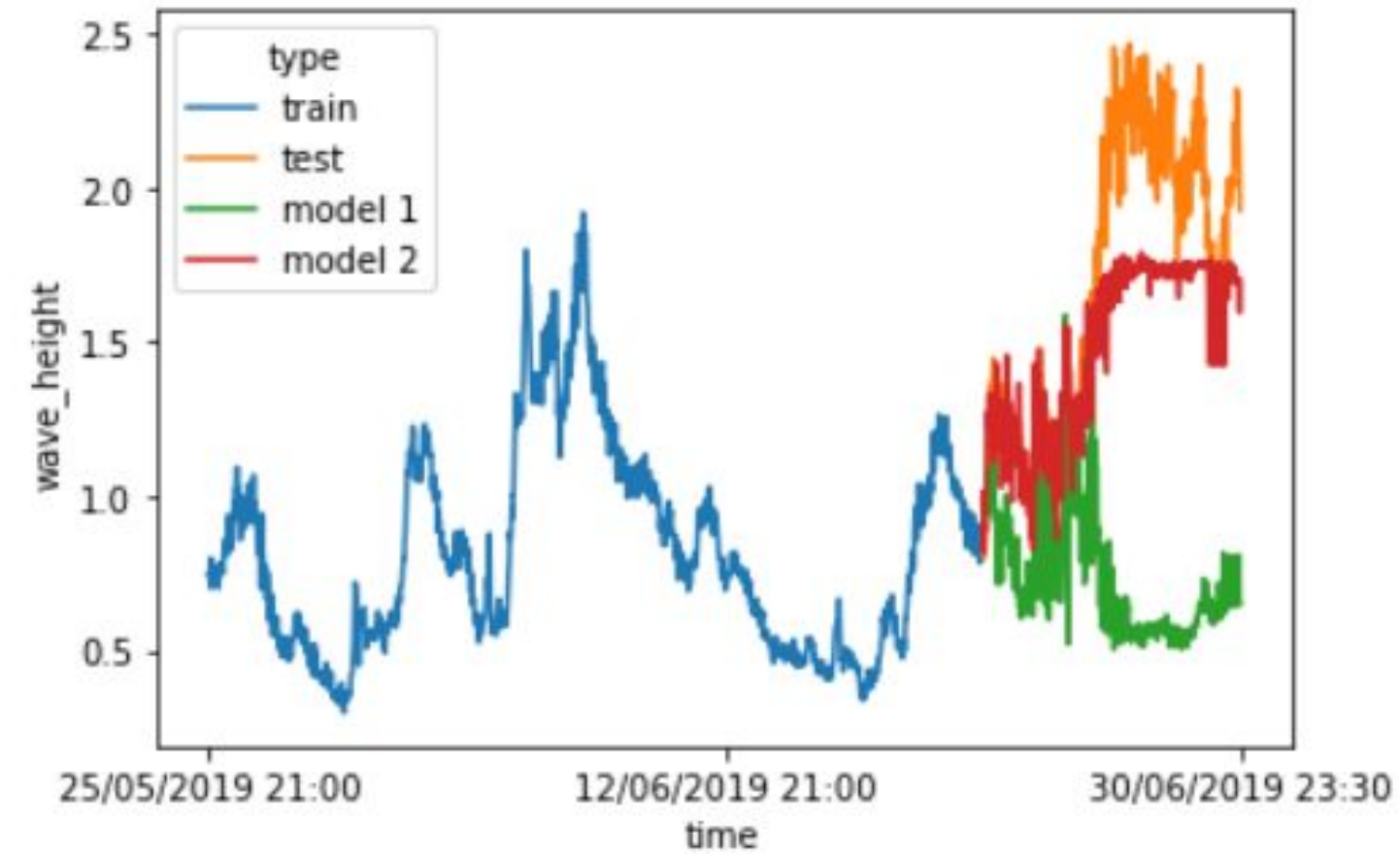


+ + + + + +

+ + + + + +

■ ■ ■ ■

# El segundo modelo hace mejores predicciones:





# conclusión



El **segundo modelo** es mucho mejor que el primer modelo, reduciendo el Error Promedio Absoluto en un 75% porque **sí considera el factor más importante**: ola más grande en el set.

Sin embargo, el **segundo modelo es poco práctico** ya que el set y la ola más grande del set ocurren al mismo tiempo. No puede pasar una sin la otra.

Siguientes pasos:

- robustecer el modelo
- *scrappear* datos meteorológicos
- estudiar el *swell*
- predecir la *calidad* de la ola