KAGGLE COMPETITION

FEATURE IMPUTATION WITH A HEAT FLUX DATASET

Presentado por Ismael Merino

INDICE

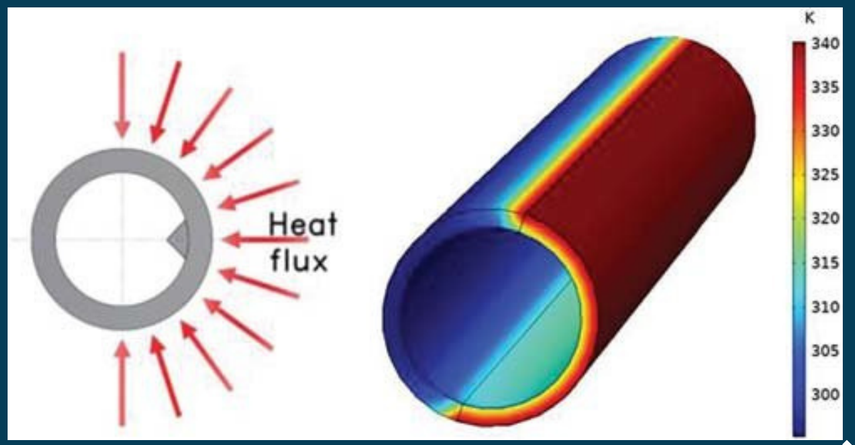
¿Qué queremos predecir?

QUEREMOS PREDECIR LA VARIABLE X_E_OUT (EQUILIBRIUM QUALITY), SE REFIERE A LA CALIDAD DE EQUILIBRIO EN UN SISTEMA DE EBULLICIÓN.

ESTA CARACTERÍSTICA REPRESENTA LA PROPORCIÓN O FRACCIÓN DE LÍQUIDO EN UNA MEZCLA LÍQUIDO-VAPOR EN EL PUNTO DE EBULLICIÓN CRÍTICO.

LA METRICA A UTILIZAR SERÁ EL RMSE (ROOT MEAN SQUARED ERROR)

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



Variables

AUTHOR - EL AUTOR DE LA PUBLICACIÓN CUYOS EXPERIMENTOS

GEOMETRY - LA GEOMETRÍA DEL CALENTADOR

PRESSURE [MPA] - PRESIÓN DEL LÍQUIDO

MASS_FLUX [KG/M2S] - LA MASA QUE SE DESPLAZA A TRAVÉS DE UNA UNIDAD DE ÁREA

D_E [MM] - EL DIÁMETRO EQUIVALENTE DEL CANAL

D_H [MM] - EL DIÁMETRO CALENTADO DEL CANAL

LENGTH [MM] - LA LONGITUD CALENTADA DEL CANAL

CHF_EXP [MW/M2] - EL FLUJO DE CALOR CRÍTICO DE CADA EXPERIMENTO

X_E_OUT [-] - LA CALIDAD DE EQUILIBRIO LOCAL/SALIDA

1. Imputación de nulos

DataFrame original

id	author	geometry	pressure [MPa]	mass_flux [kg/m2-s]	x_e_out [-]	D_e [mm]	D_h [mm]	length [mm]	chf_exp [MW/m2]
0	Thompson	tube	7.00	3770.0	0.1754	NaN	10.8	432.0	3.6
1	Thompson	tube	NaN	6049.0	-0.0416	10.3	10.3	762.0	6.2
2	Thompson	NaN	13.79	2034.0	0.0335	7.7	7.7	457.0	2.5
3	Beus	annulus	13.79	3679.0	-0.0279	5.6	15.2	2134.0	3.0
4	NaN	tube	13.79	686.0	NaN	11.1	11.1	457.0	2.8
5	NaN	NaN	17.24	3648.0	-0.0711	NaN	1.9	696.0	3.6
6	Thompson	NaN	6.89	549.0	0.1203	12.8	12.8	1930.0	2.6
7	Peskov	tube	18.00	750.0	NaN	10.0	10.0	1650.0	2.2
8	NaN	tube	12.07	4042.0	-0.0536	NaN	NaN	152.0	5.6
9	Peskov	tube	12.00	1617.0	0.1228	10.0	10.0	520.0	2.2

1. Imputación de nulos

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31644 entries, 0 to 31643
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    31644 non-null  int64  
 1   author                26620 non-null  object 
 2   geometry              26144 non-null  object 
 3   pressure [MPa]        27192 non-null  float64
 4   mass_flux [kg/m2-s]   26853 non-null  float64
 5   x_e_out [-]           21229 non-null  float64
 6   D_e [mm]              26156 non-null  float64
 7   D_h [mm]              27055 non-null  float64
 8   length [mm]           26885 non-null  float64
 9   chf_exp [MW/m2]       31644 non-null  float64
dtypes: float64(7), int64(1), object(2)
memory usage: 2.4+ MB
```

Observamos que tenemos 2 variables categóricas (author y geometry) y muchos nulos en el resto de columnas

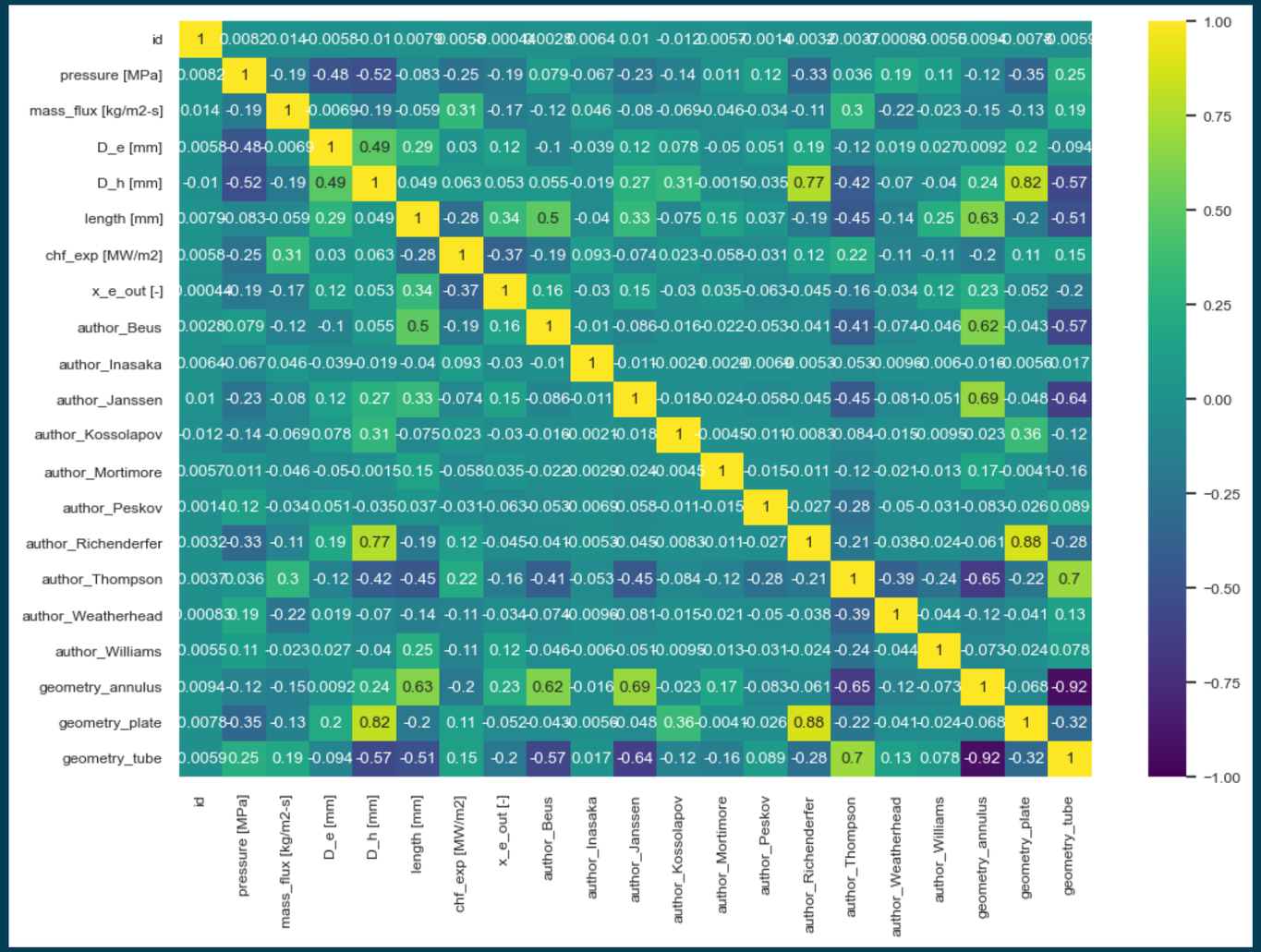
1. Imputación de nulos

	author	geometry	0
0	Beus	annulus	1575
1	Beus	tube	29
2	Inasaka	plate	1
3	Inasaka	tube	45
4	Janssen	annulus	2684
5	Janssen	plate	1
6	Janssen	tube	31
7	Kossolapov	annulus	1
8	Kossolapov	plate	97
9	Kossolapov	tube	3
10	Mortimore	annulus	189
11	Mortimore	plate	2
12	Mortimore	tube	6

13	Peskov	annulus	1
14	Peskov	plate	3
15	Peskov	tube	1080
16	Richenderfer	annulus	6
17	Richenderfer	plate	504
18	Richenderfer	tube	35
19	Thompson	annulus	9
20	Thompson	plate	11
21	Thompson	tube	17376
22	Weatherhead	annulus	1
23	Weatherhead	tube	2039
24	Williams	annulus	1
25	Williams	plate	1
26	Williams	tube	889

Imputamos las variables categóricas author y geometry por su moda cuando la relacionamos entre ellas

2. EDA

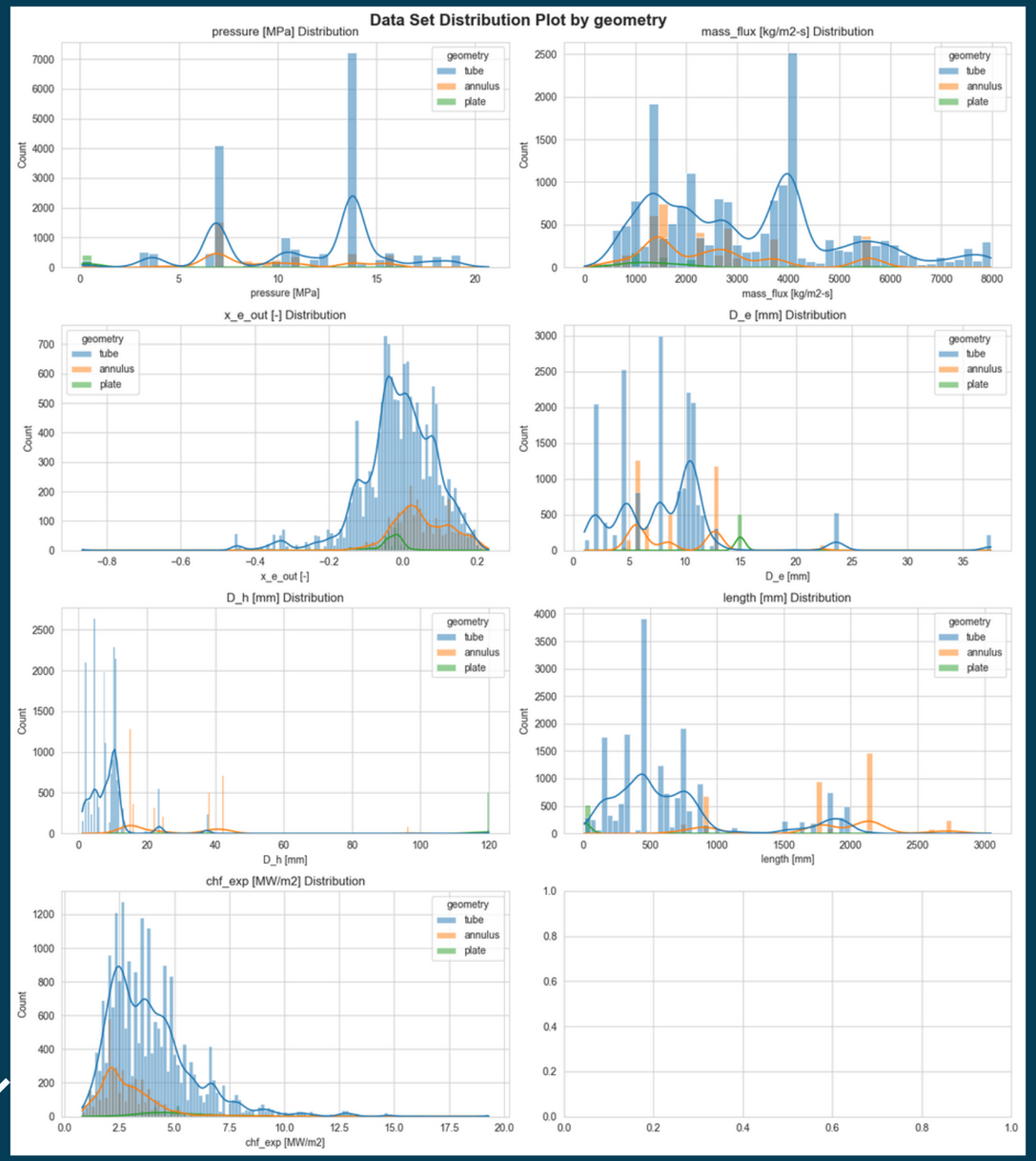


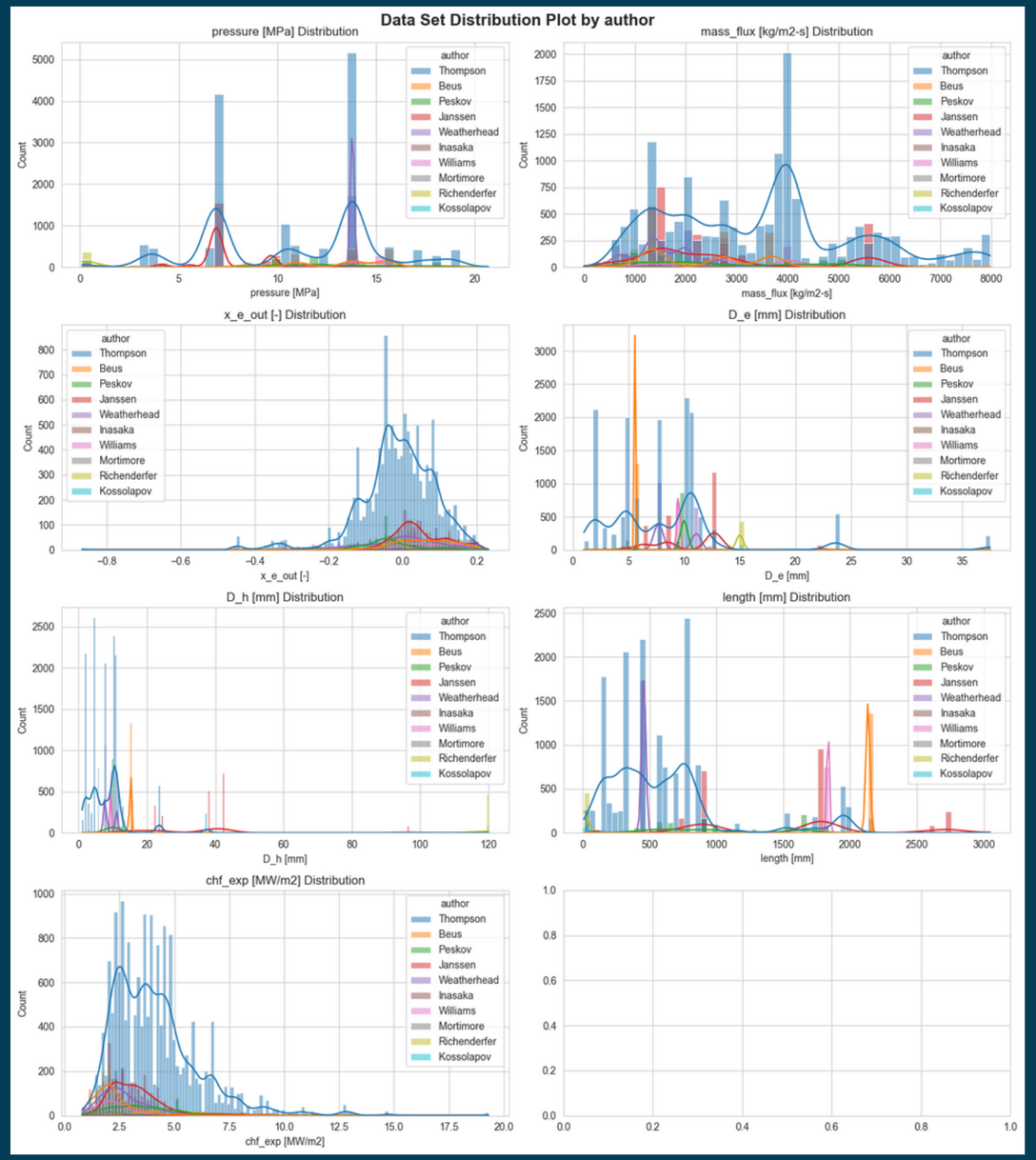
Matriz correlación

Imputamos el resto de variables con algunas relaciones correlaciones como el D_e y D_h.

El resto de variables las imputamos con KNNImputer.

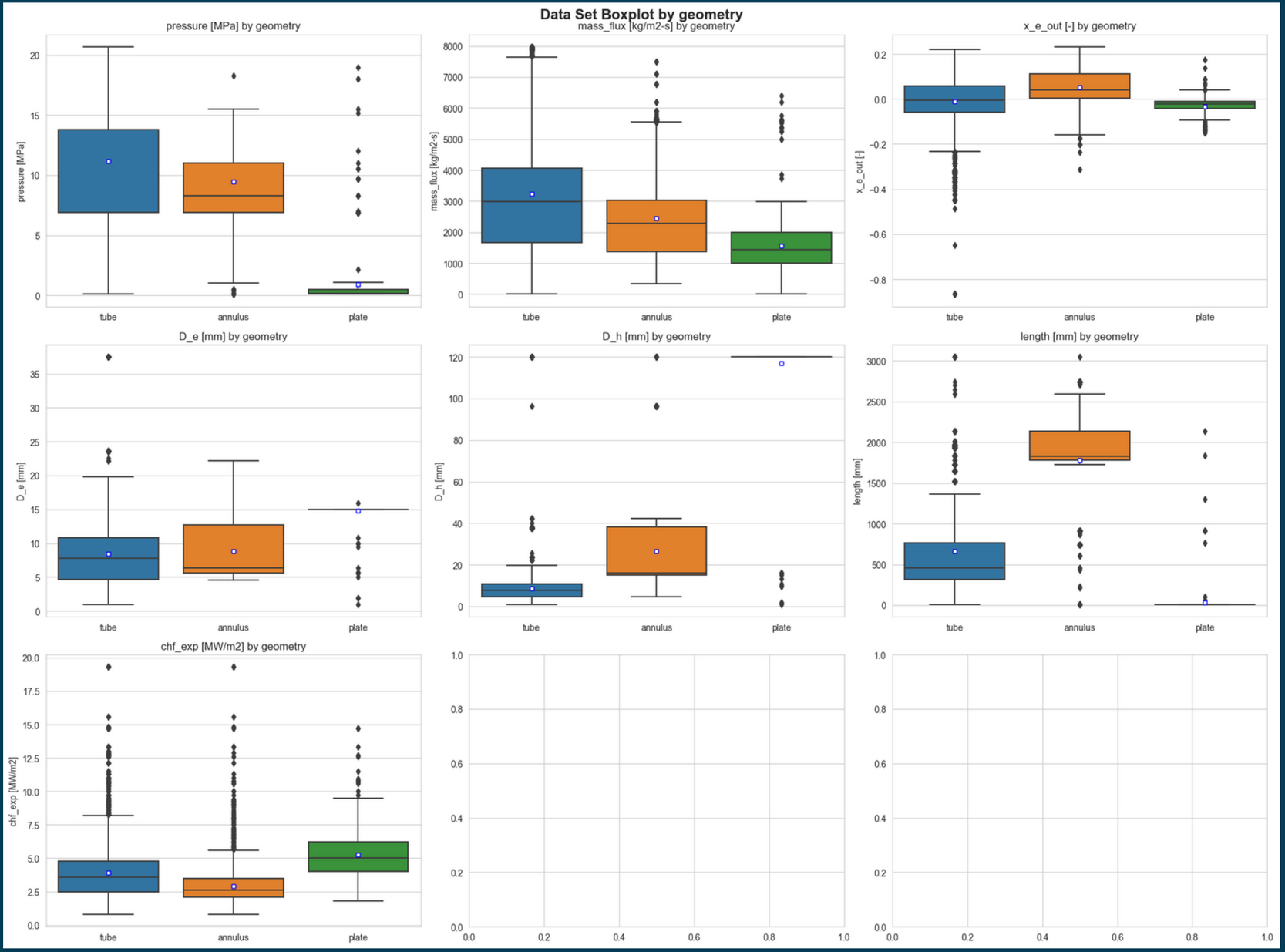
2. EDA Diagramas





2. EDA

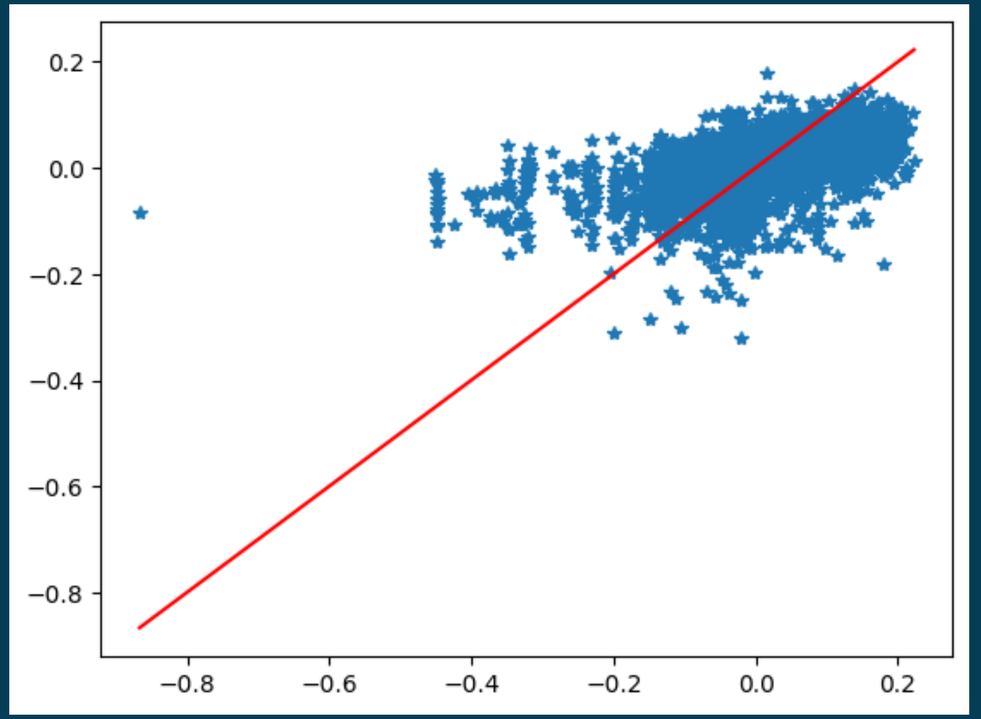
Boxplot



3. Modelos Machine Learning

3.1 Regresión Lineal

Creamos un primer modelo de Regresión Lineal, separando las variables de X en X_train y X_test. Hacemos lo mismo en la Y y calculamos el RMSE y predecimos



RMSE KAGGLE= 0.0853

R2 = 0.274

3. Modelos Machine Learning

3.2 Random Forest

```
rf_model = RandomForestRegressor(max_depth= 12, min_samples_leaf= 5,
                                 min_samples_split= 10, n_estimators= 100)
```

Creamos un segundo modelo de Random Forest, separando las variables de X en X_train y X_test. Hacemos lo mismo en la Y y calculamos el RMSE y predecimos

RMSE TRAIN = 0.0758

RMSE TEST = 0.0754

R2 = 0.433

RMSE KAGGLE = 0.0793

3. Modelos Machine Learning

3.3 XG Boost

```
model = XGBRegressor(colsample_bytree=0.5, learning_rate=0.01,
                     max_depth=5, n_estimators=1000, subsample=0.6)
```

Creamos un tercer modelo de XG Boost, separando las variables de X en X_train y X_test. Hacemos lo mismo en la Y y calculamos el RMSE y predecimos

RMSE TRAIN = 0.0745

RMSE TEST = 0.0776

R2 = 0.452

RMSE KAGGLE = 0.0793

3. Modelos Machine Learning

3.4 Gradient Boosting

```
gb_model = GradientBoostingRegressor(max_depth=5, n_estimators=100,
                                     learning_rate=0.1)
```

Creamos un cuarto modelo de Gradient Boosting, separando las variables de X en X_train y X_test. Hacemos lo mismo en la Y y calculamos el RMSE y predecimos

RMSE TRAIN = 0.0766

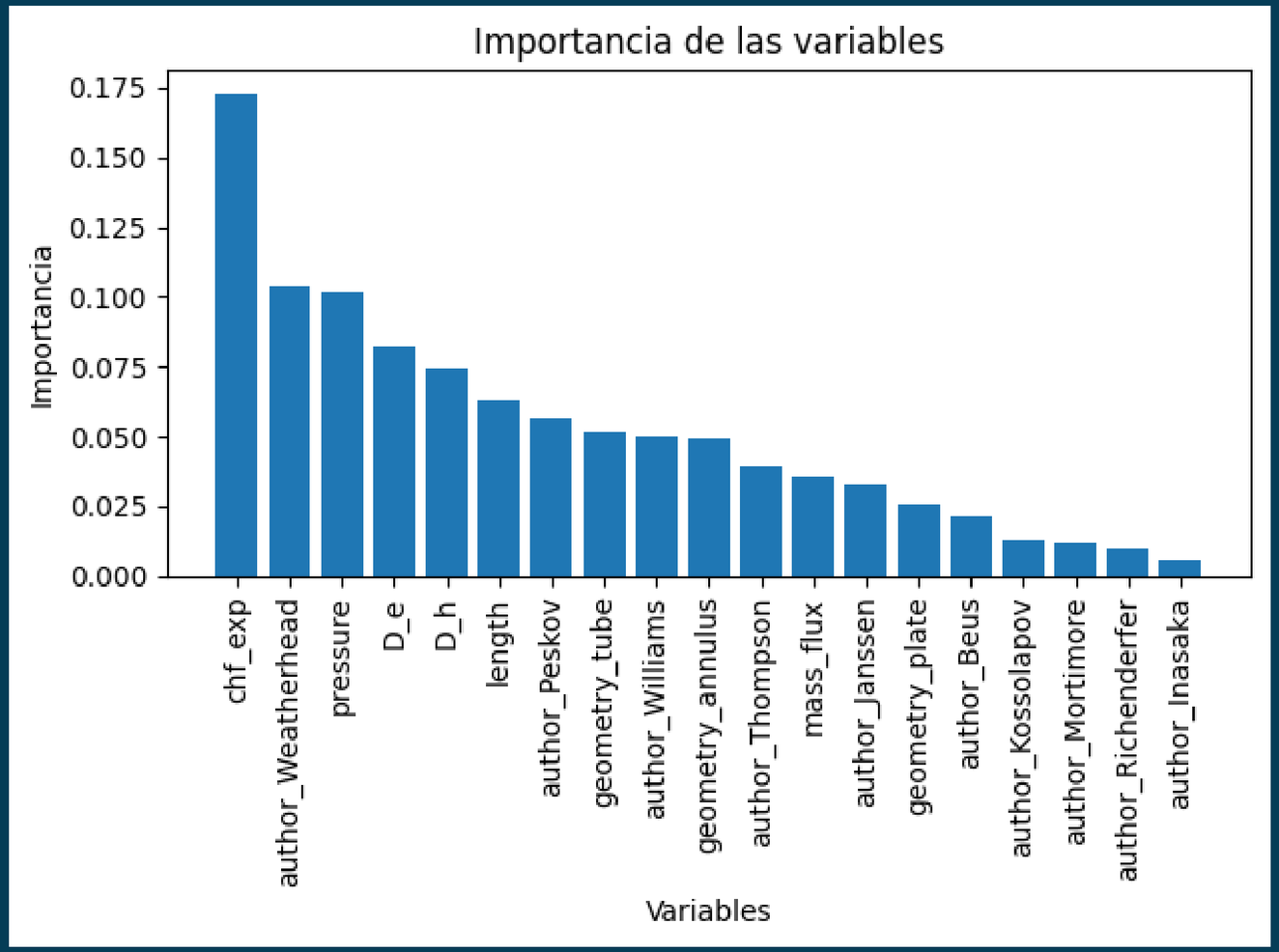
RMSE TEST = 0.0780

R2 = 0.423

RMSE KAGGLE = 0.0797

3. Modelos Machine Learning

Feature importances XG Boost



3. Modelos Machine Learning

3.5 Ensamble

```
voting_reg1 = VotingRegressor([('rf', rf_pipeline), ('XG', xgb_pipeline),
                               ('GraBoosst', gb_pipeline)],
                              weights=[2/10, 7/10, 1/10])
```

Con los modelos anteriores, hago un ensamble con XG Boost, Random Forest y Gradient Forest, dando mas peso del primero al ultimo, generandome el mejor modelo.
Concateno al X_train el Dataset original y borro author y geometry

RMSE TRAIN = 0.0739

RMSE TEST = 0.0744

R2 = 0.475

RMSE KAGGLE = 0.0763

4. Modelo Deep Learning

Keras

Creamos un modelo con Keras, dandole una capa densa de 64 neuronas con funcion relu, y una neurona de salida que de la predicción.

RMSE TRAIN = 0.327

RMSE TEST = 0.345

R2 = 0.100

RMSE KAGGLE = ...

5. Conclusión

Nuestro mejor modelo es el ensamble con XG Boost, Random Forest y Gradient Forest

Modelos	RMSE Train	RMSE Test	RMSE Kaggle	R2
Regresión Lineal	0.0845	0.0847	0.0853	0.2740
Random Forest	0.0758	0.0754	0.0793	0.4330
XG Boost	0.0745	0.0776	0.0793	0.4520
Grandient Boosting	0.0766	0.0780	0.0797	0.4230
Ensamble	0.0739	0.0744	0.0763	0.4750
Deep Learning	0.3270	0.345		0.1000

Gracias por su atención

ISMAEL MERINO
The Bridge 2023