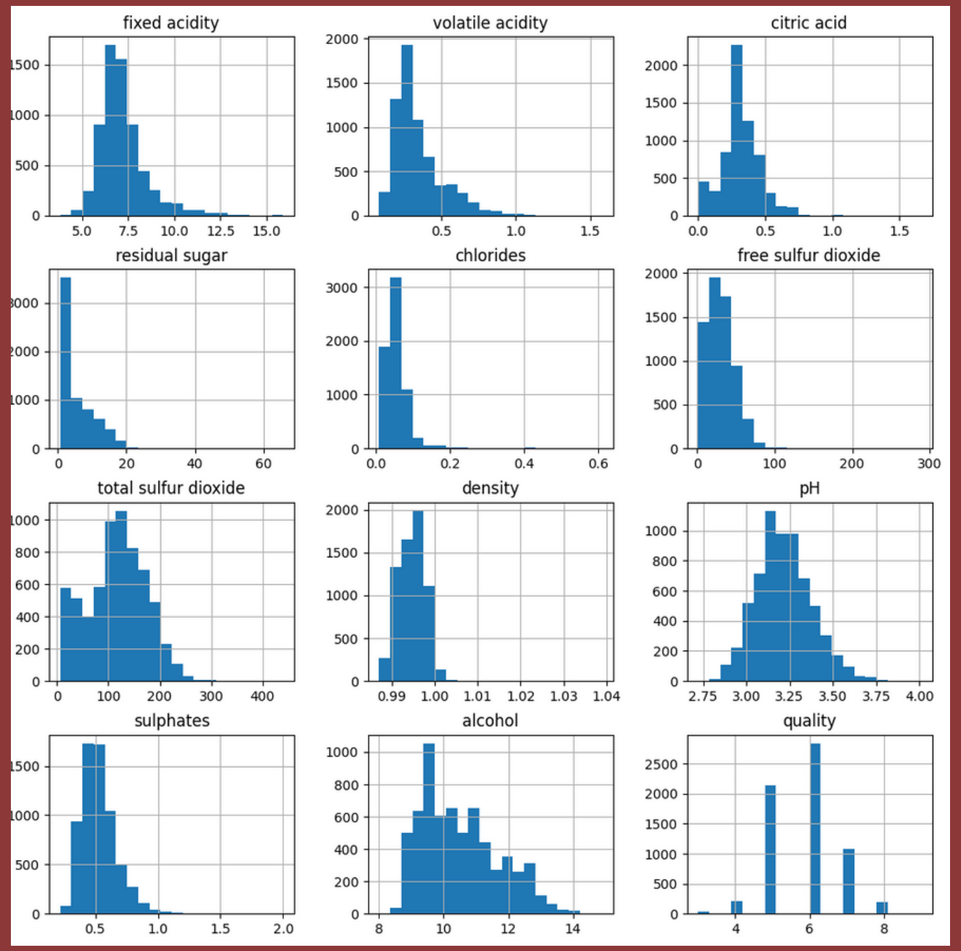
PROYECTO MACHINE LEARNING

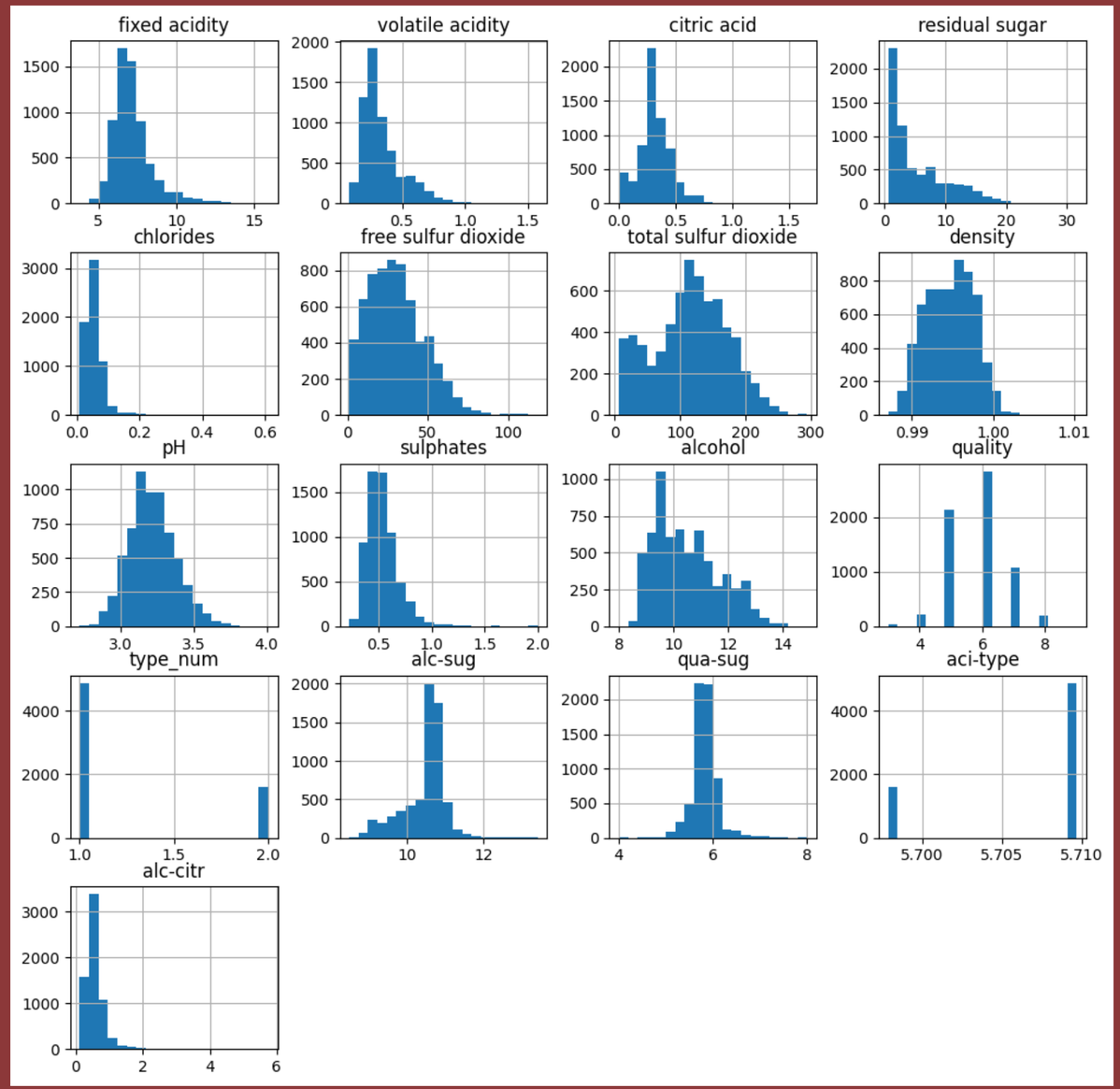
MODELO DE PREDICCION :
CALIDAD DEL VINO

Pasos realizados para llegar al modelo final:

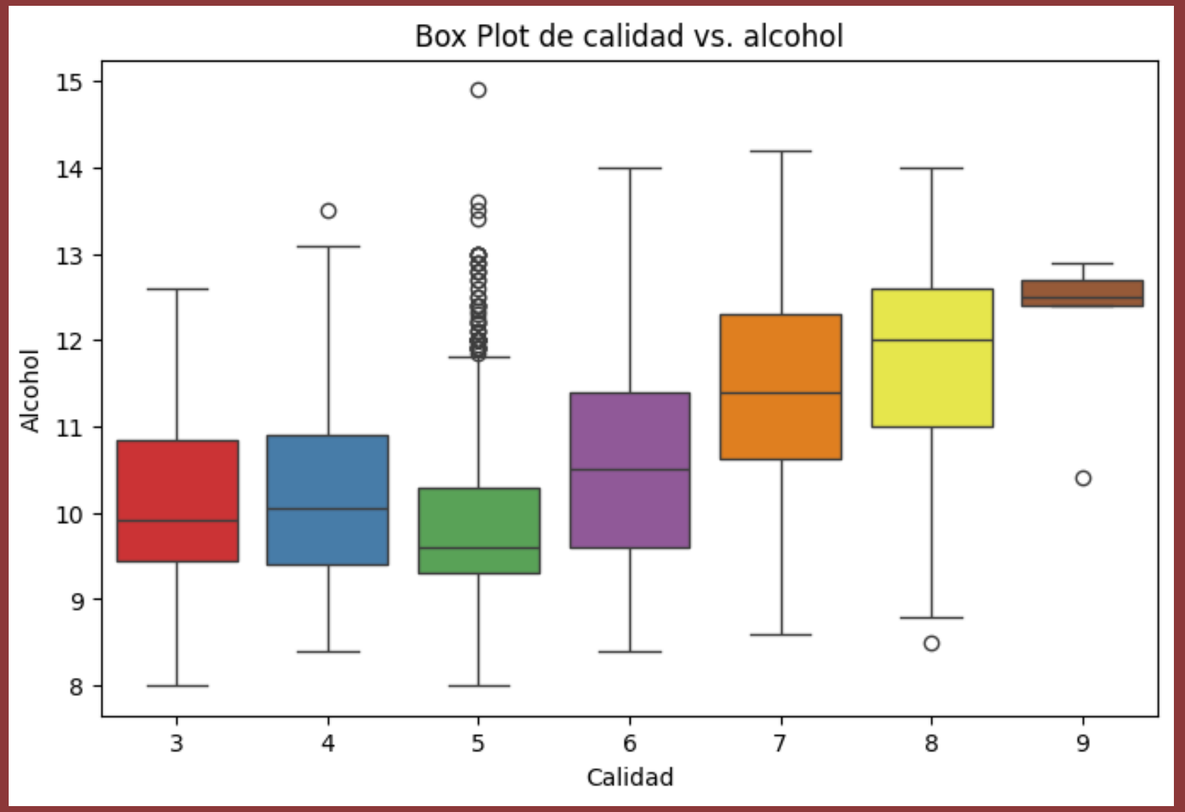
- Obtención de datos
- Procesamiento de datos
- Elección del target
- Estudio de sus variables y correlaciones
- Entrenamiento de nuestros modelos
- Evaluación de los datos finales
- Generación de modelo con posibilidad de actualización

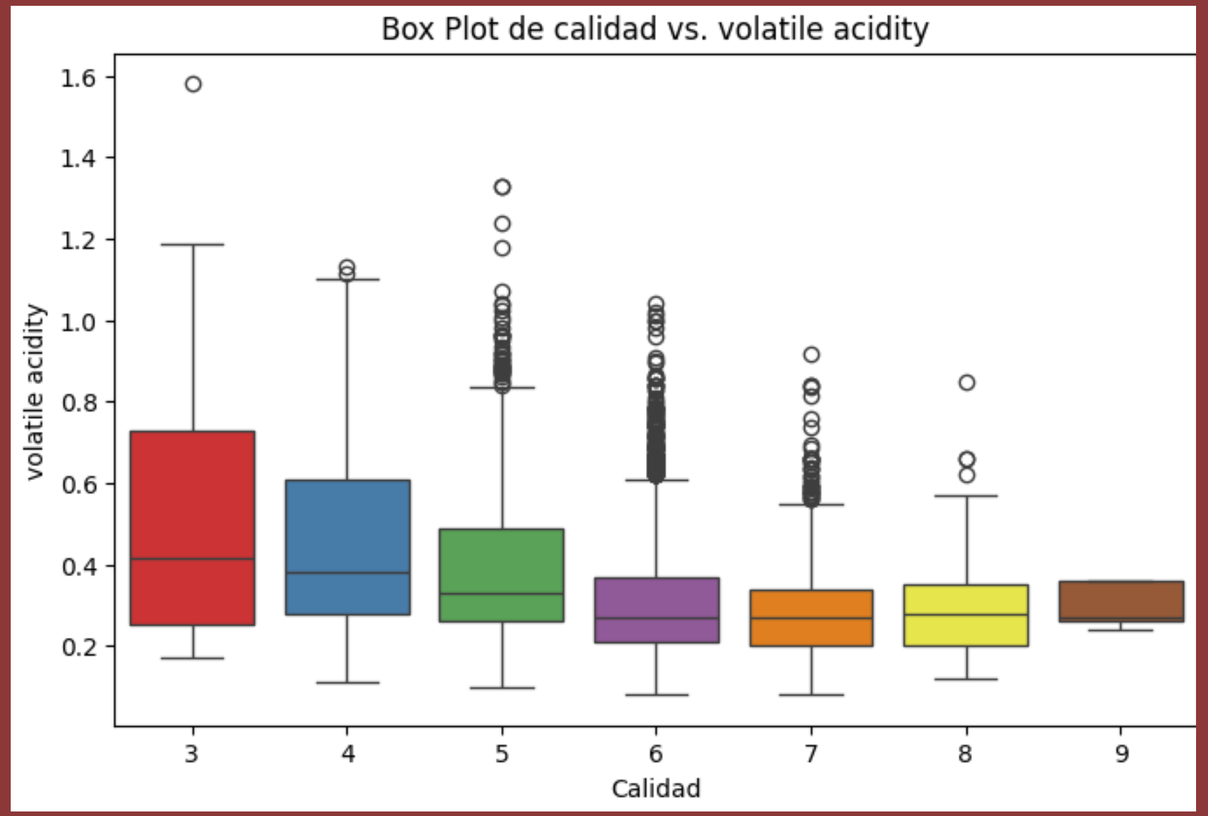
Analisis de distribución de nuestras variables :

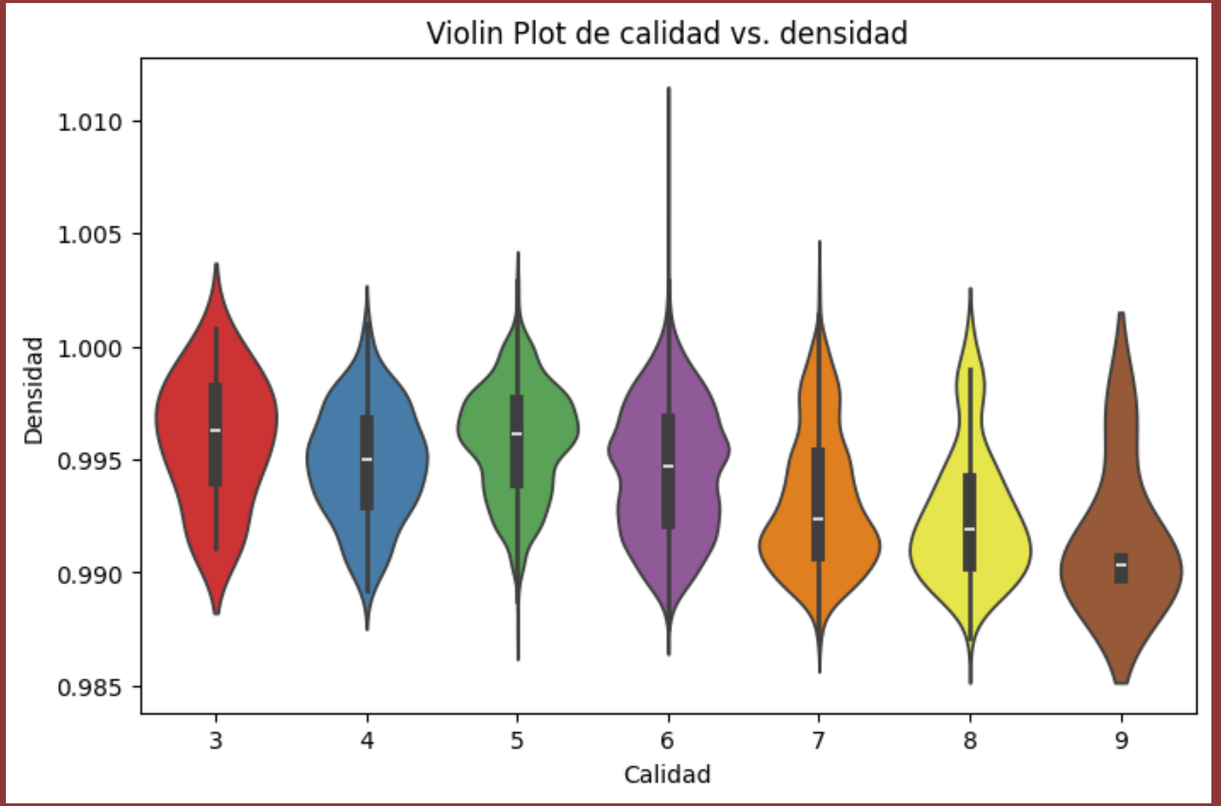


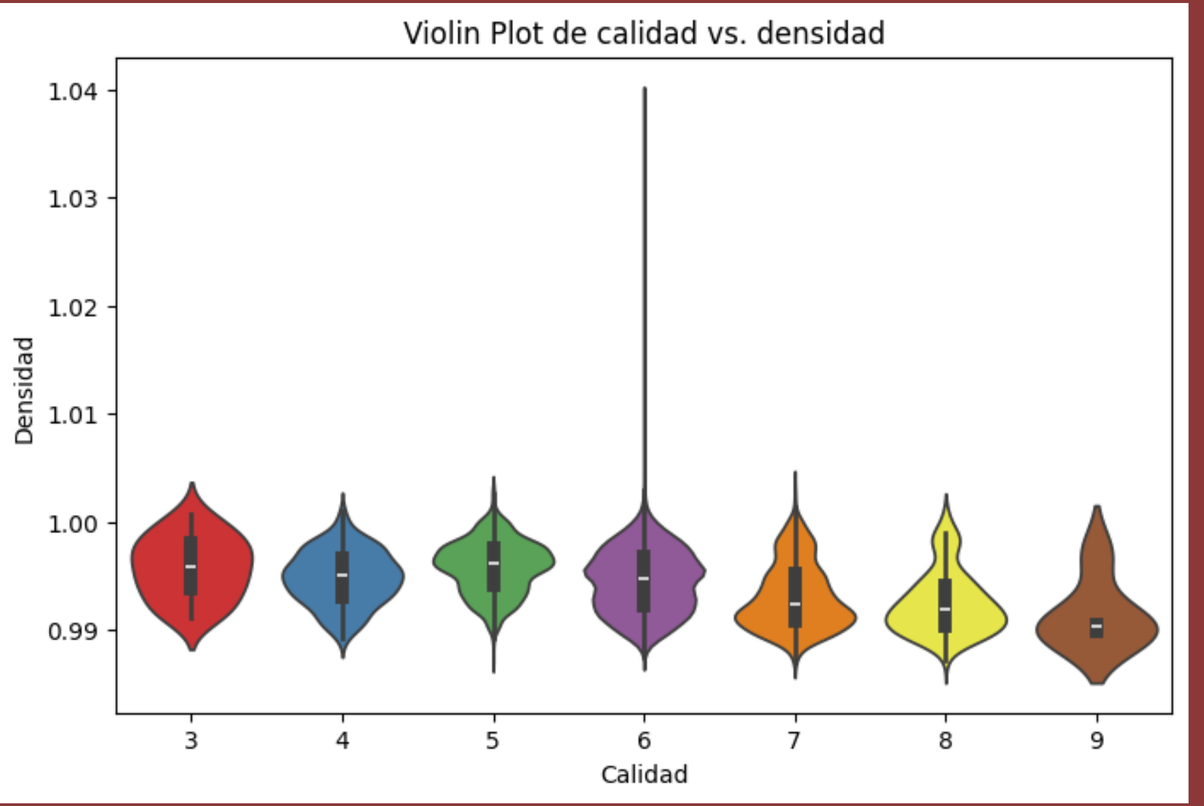


Analizamos la correlación de las variables con nuestro target :

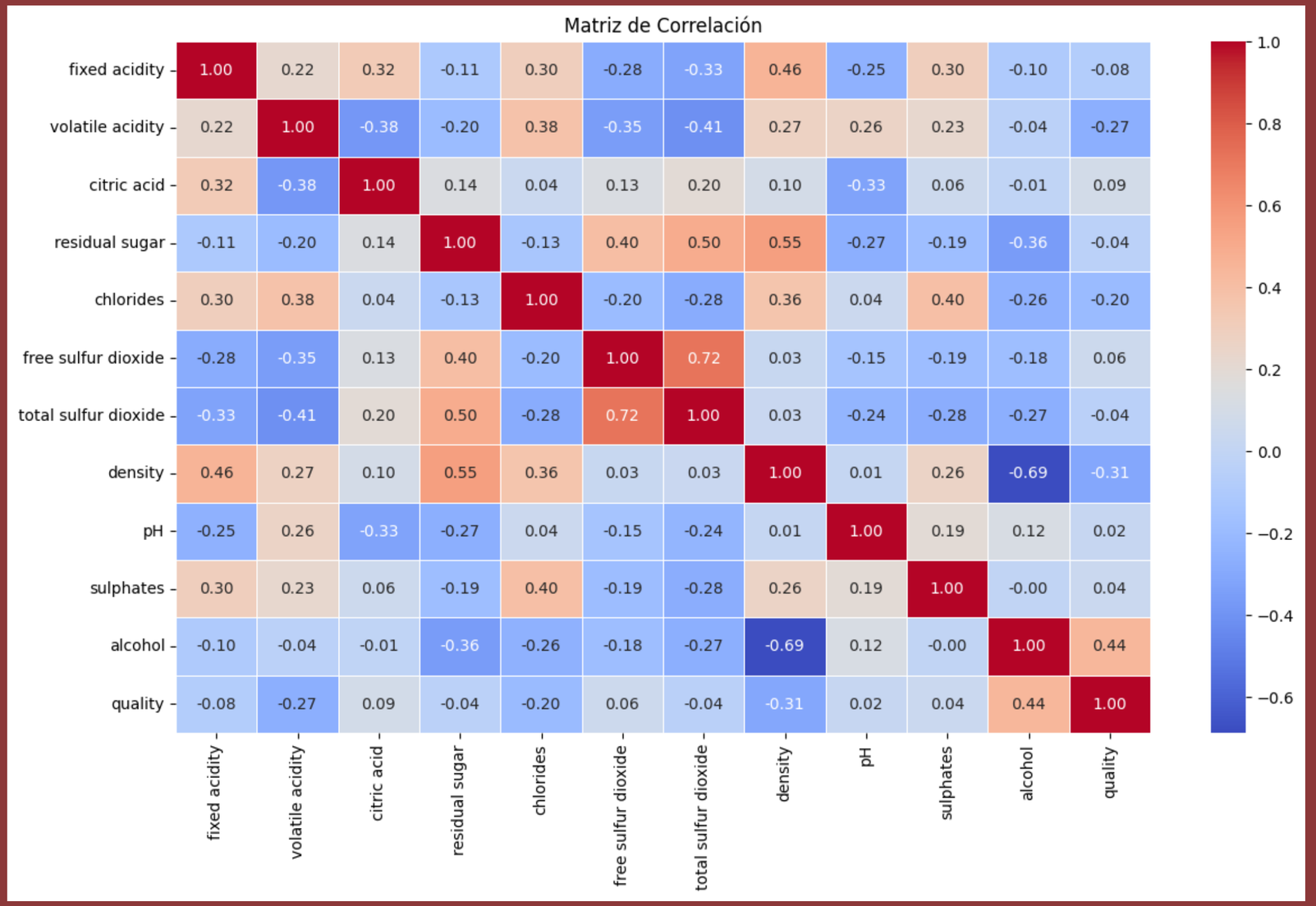




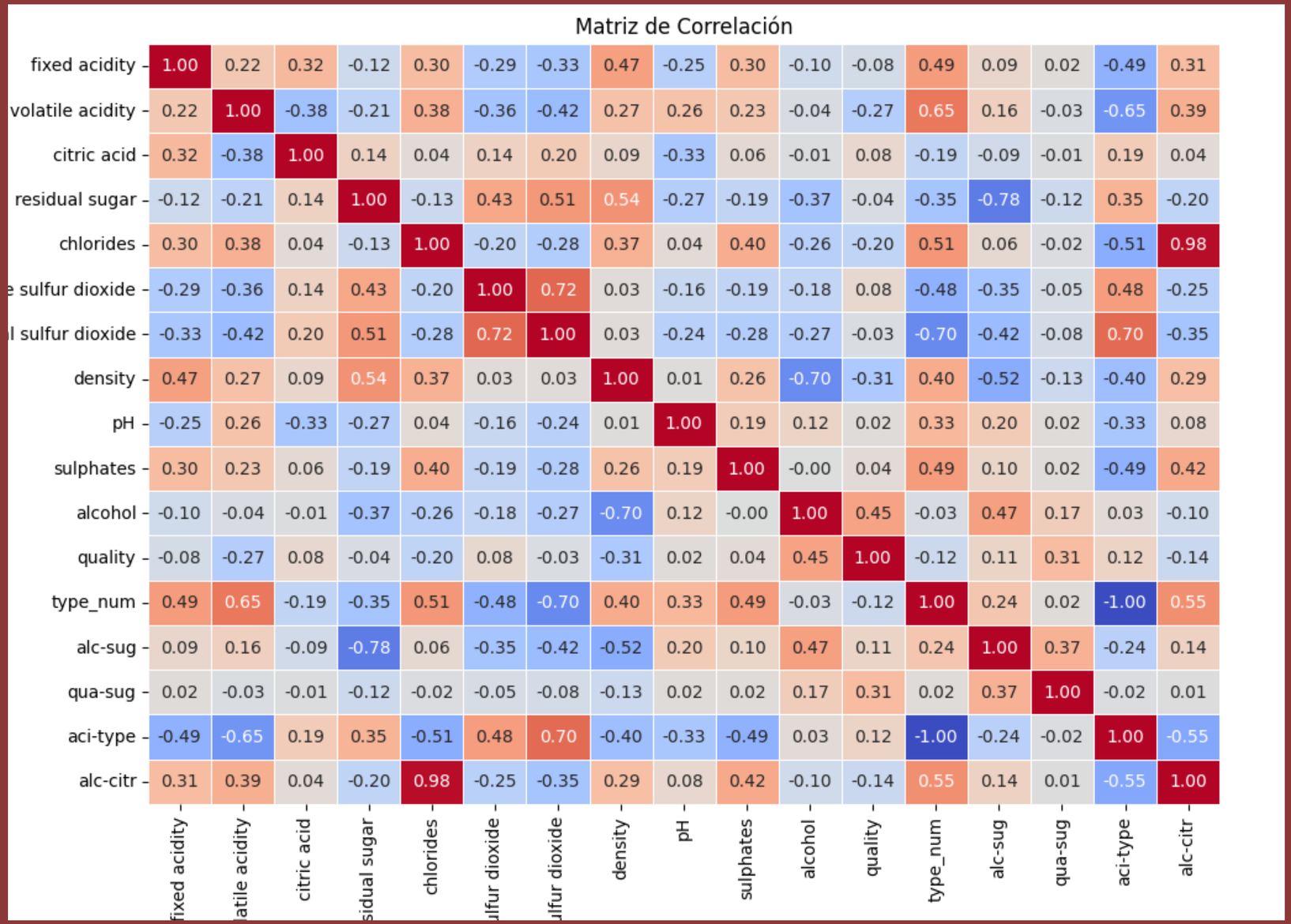

Violin Plot de calidad vs. densidad
Densidad
1.010
1.005
1.000
0.995
0.990
0.985
3
4
5
6
7
8
9
Calidad


Violin Plot de calidad vs. densidad
Densidad
1.04
1.03
1.02
1.01
1.00
0.99
3
4
5
6
7
8
9
Calidad

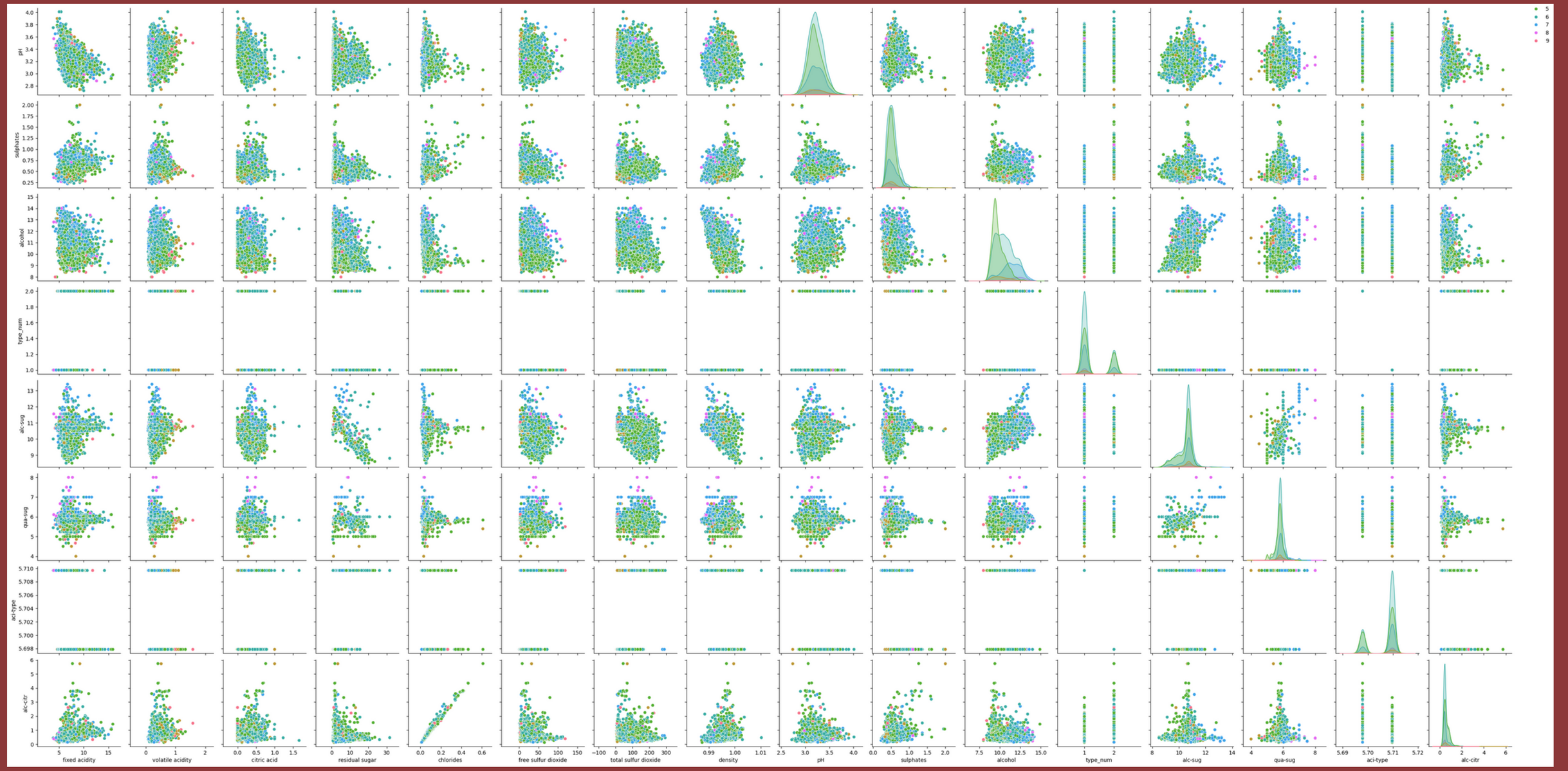
Al tener baja la correlación entre nuestras variables, buscamos relacionarlas entre ellas para conseguir un mayor apoyo a nuestro entrenamiento

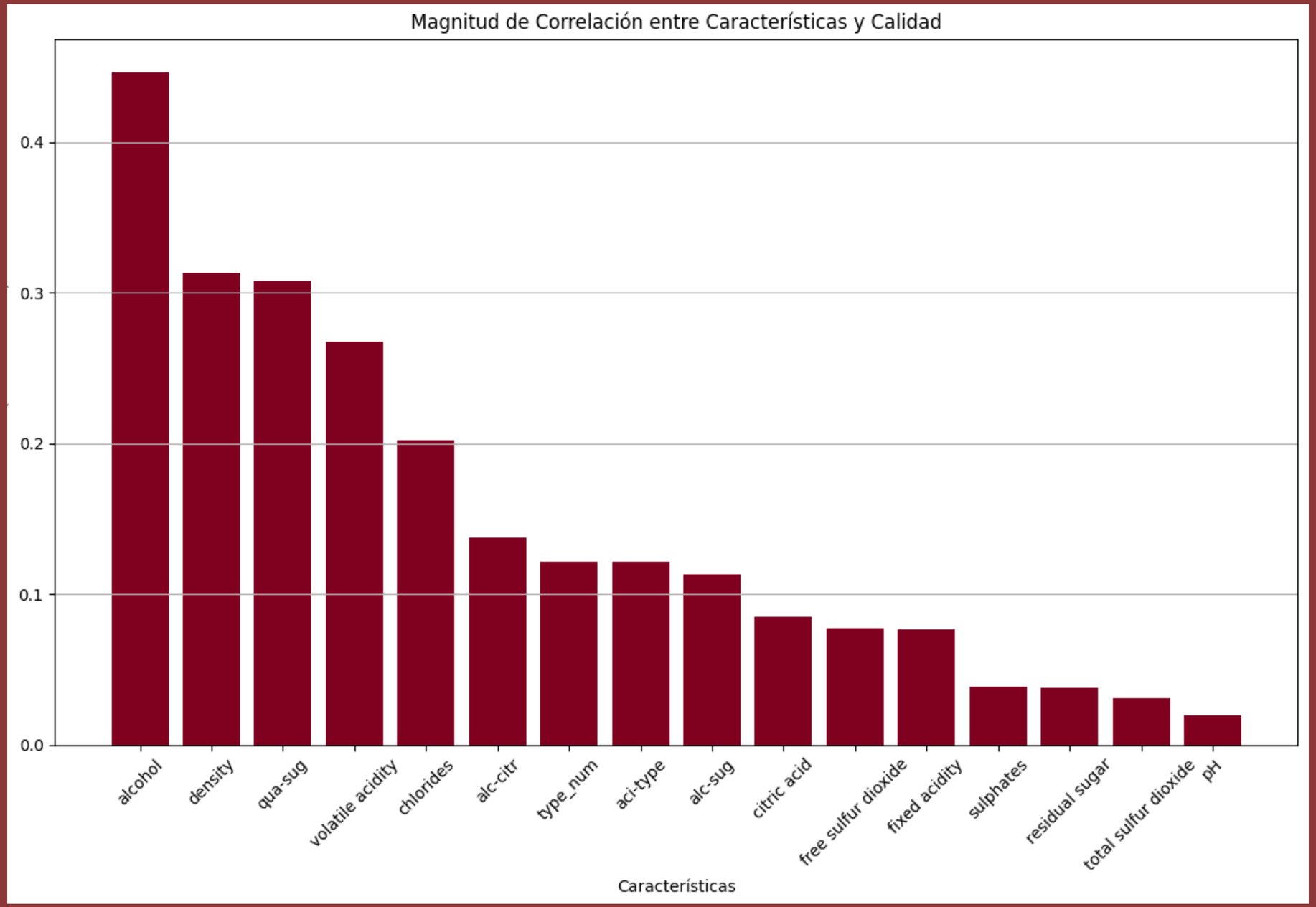


Tras realizar el Feature engineering :

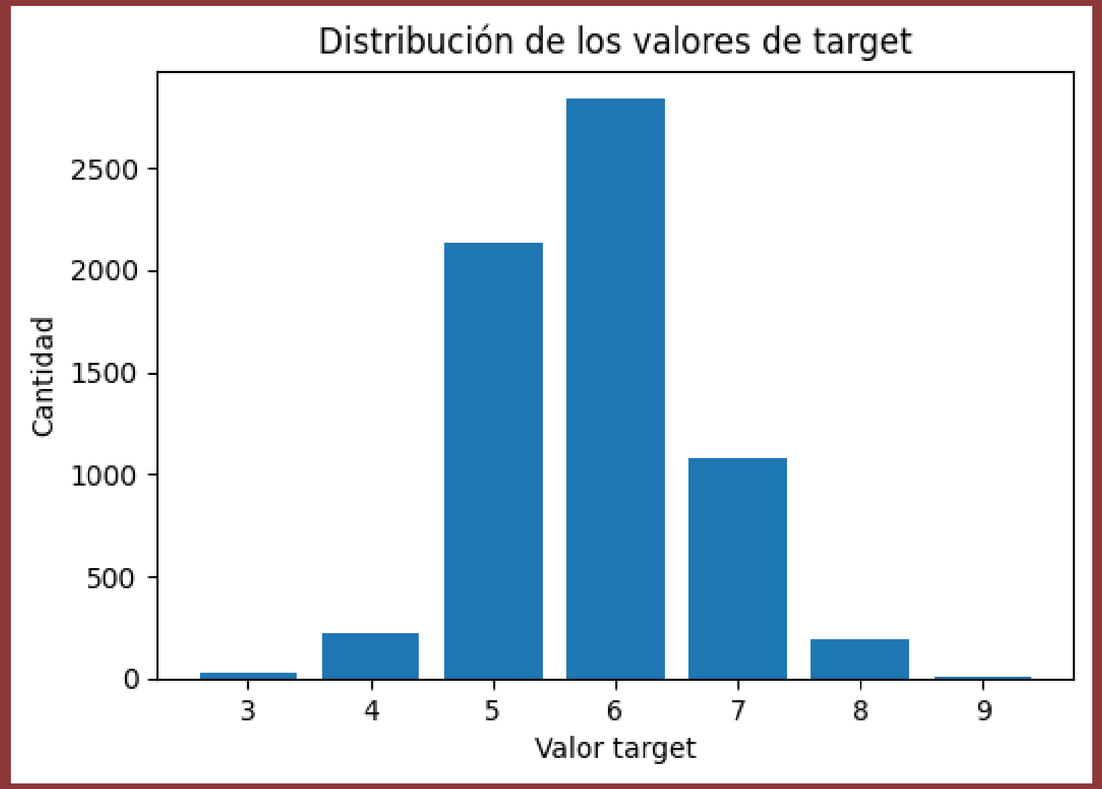


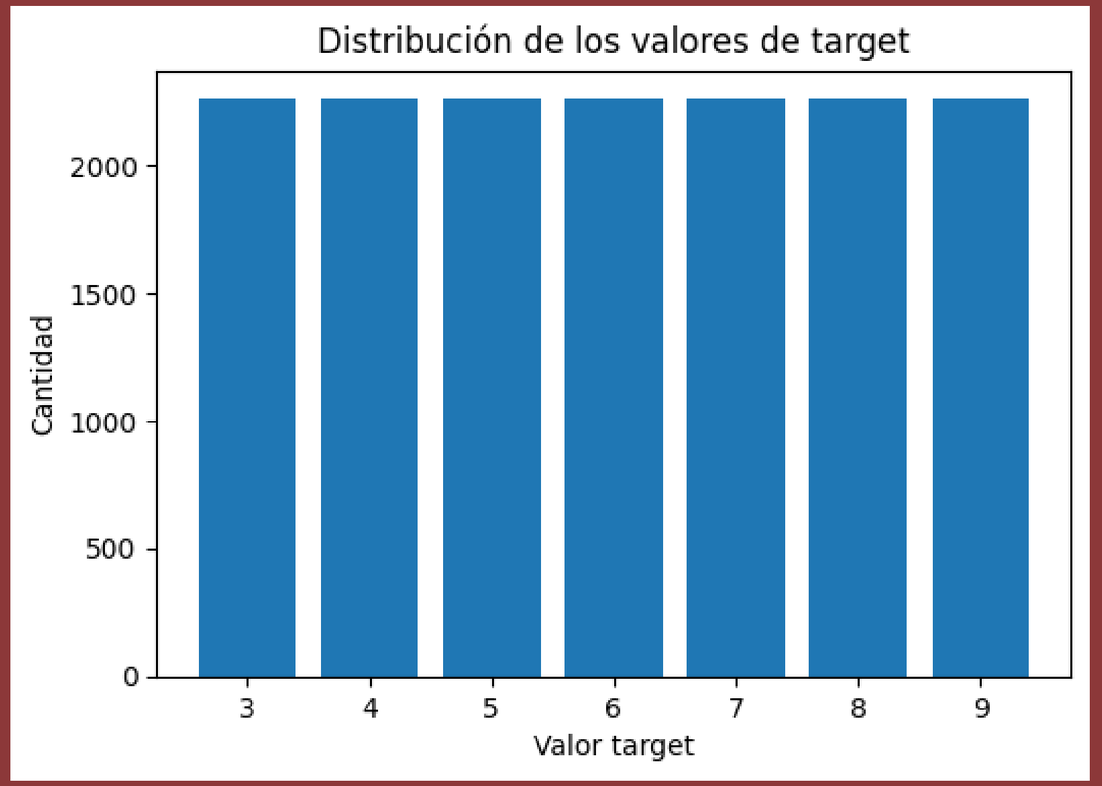

fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
quality
3


fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
type_num
alc-sug
qua-sug
aci-type
alc-citr
5
6
7
8
9


Magnitud de Correlación entre Características y Calidad
0.4
0.3
0.2
0.1
0.0
alcohol
density
qua-sug
volatile acidity
chlorides
alc-citr
type_num
aci-type
alc-sug
citric acid
free sulfur dioxide
fixed acidity
sulphates
residual sugar
total sulfur dioxide
pH
Características

Evaluamos haciendo un RandomOverSampler para que pueda aprender bien de todos nuestros target





Comenzamos con el entrenamiento de nuestros modelos :

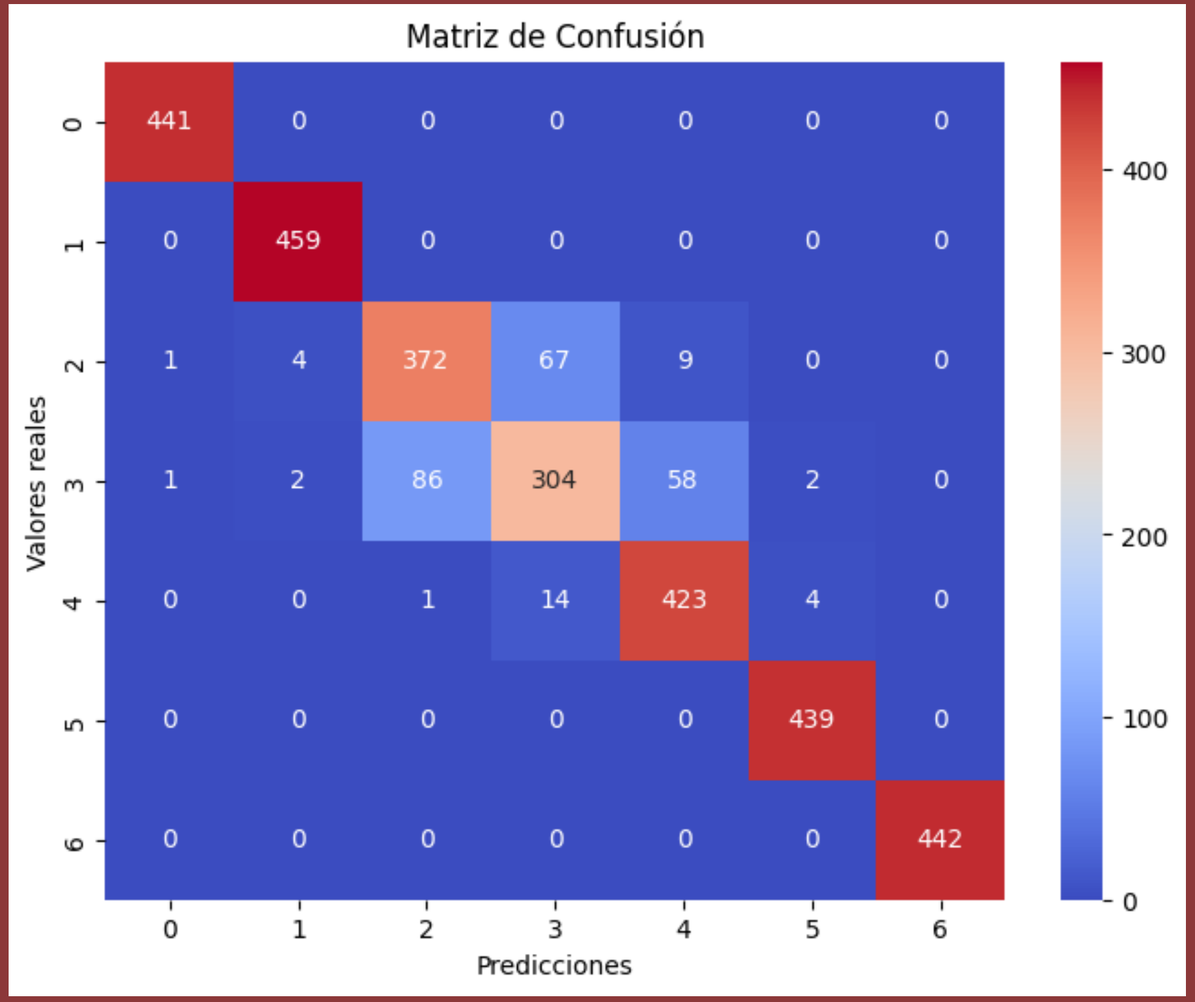
En primer lugar afronte mi problema como clasificación, para que me diera una clasificación del 3 al 9 en función de su calidad.
Una vez hecho esto , entendí que seria mas aconsejable obtener un valor real, mas que una aproximación, por lo que después pase a los modelos de regresión.

A continuación os enseñare mis modelos de clasificación y sus resultados obtenidos

Modelos de clasificación

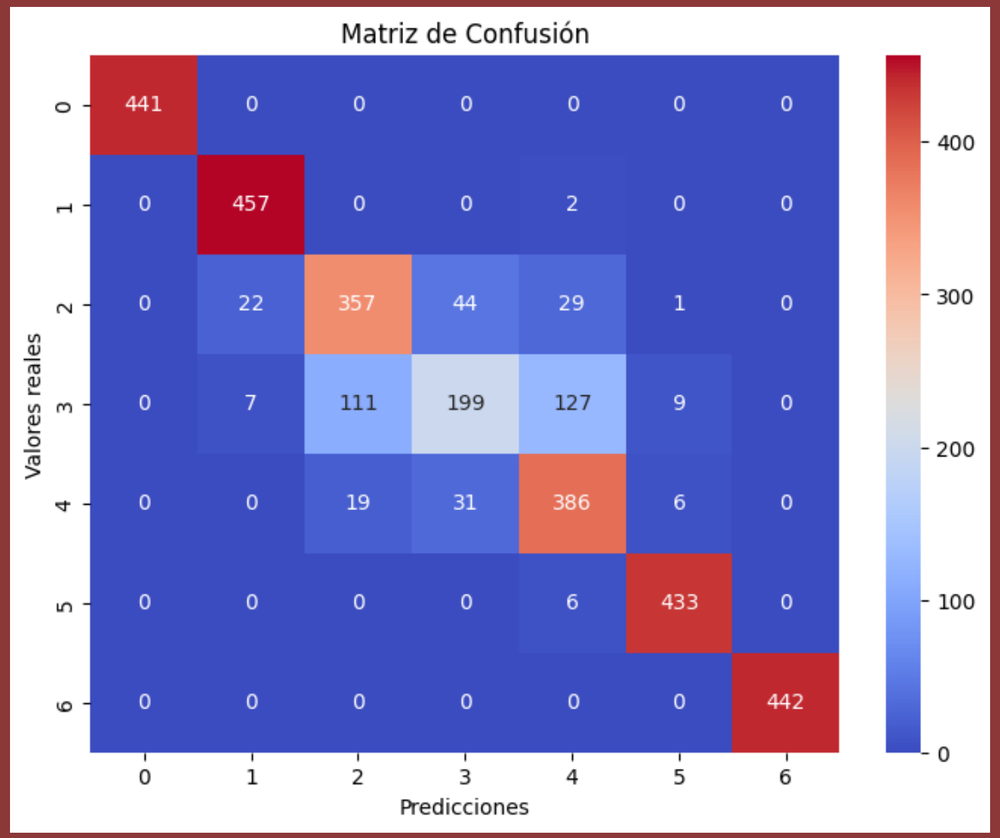
Gradient Boosting

El mejor modelo de clasificación



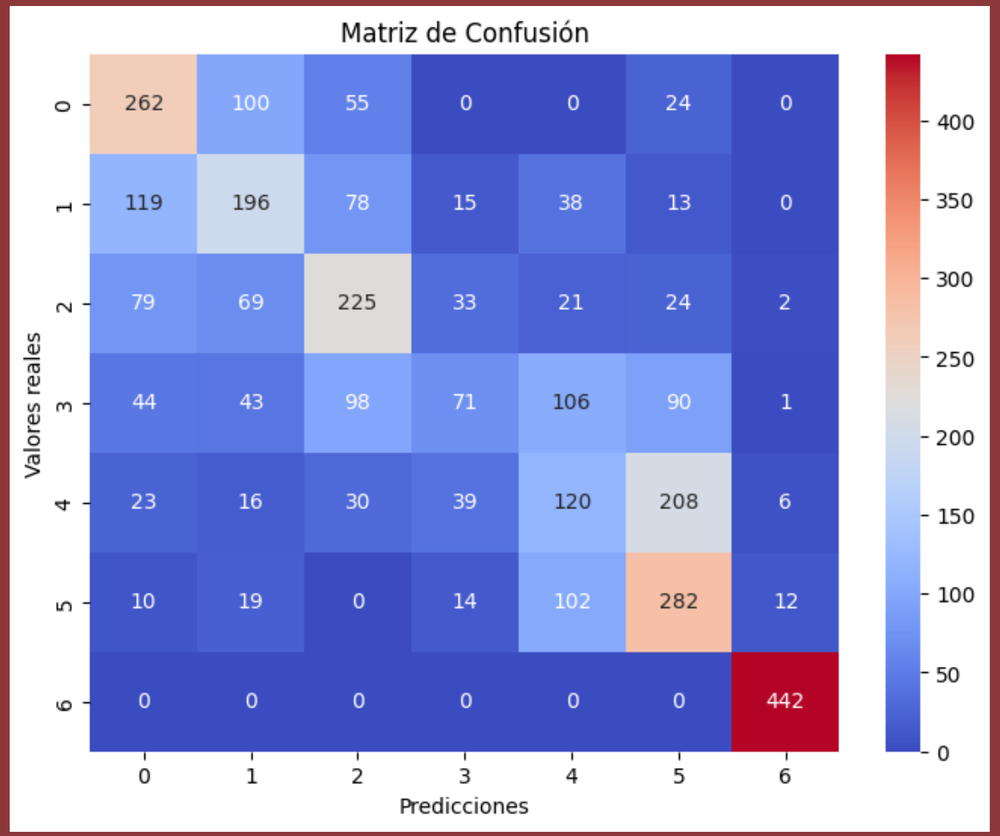
Tasa de aciertos :
92.04%
classifier__learning_rate: 0.2,
'classifier__max_depth': 6,
'classifier__n_estimators': 150

Random Forest



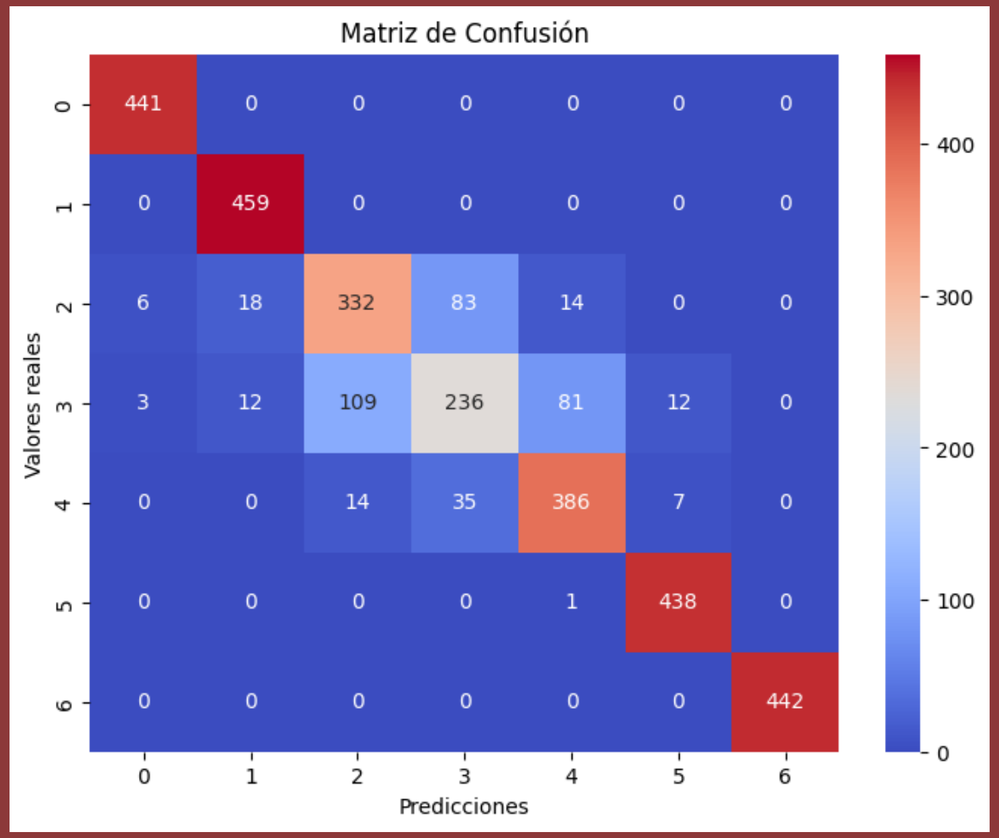
Tasa de aciertos 86.77%
classifier': RandomForestClassifier(),
'classifier__max_depth': 10,
'classifier__max_features': 1,
'classifier__n_estimators': 200

Logistic Regression



Tasa de aciertos : 51.07%
classifier__C': 10,
classifier__max_ite: 100,
classifier__penalty: 'l1'
classifier__solver: 'liblinear

SVC

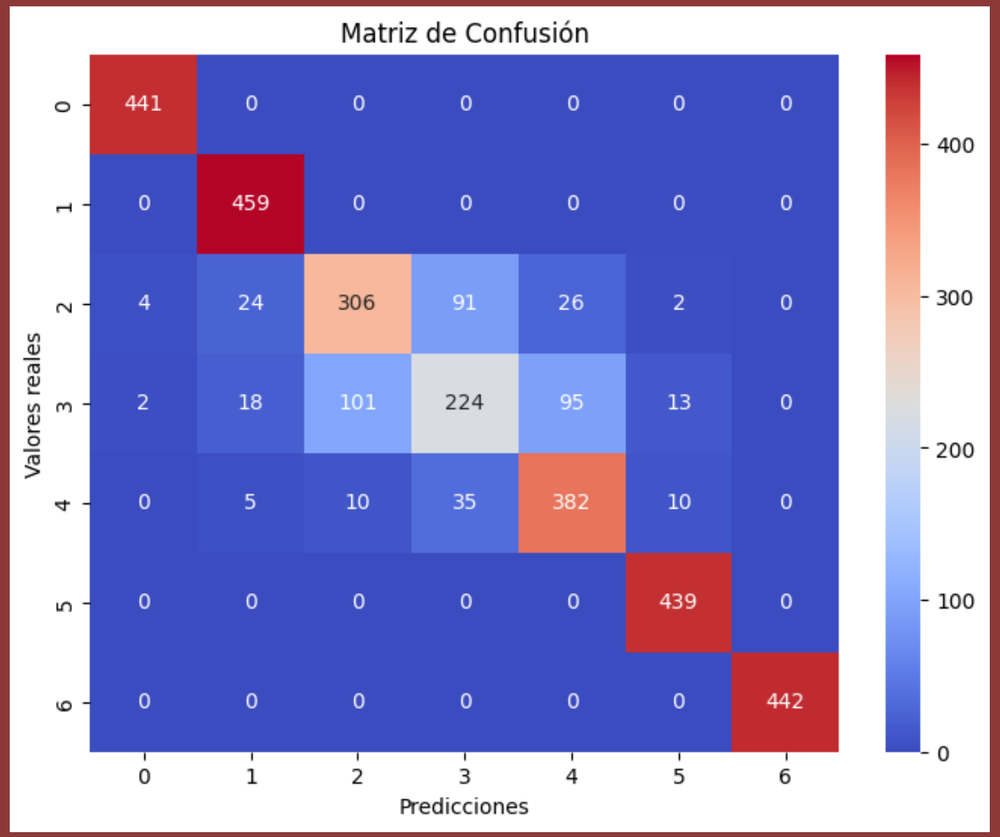


Tasa de Aciertos 87.38%

classifier': SVC(),

'classifier__C': 200

KNN



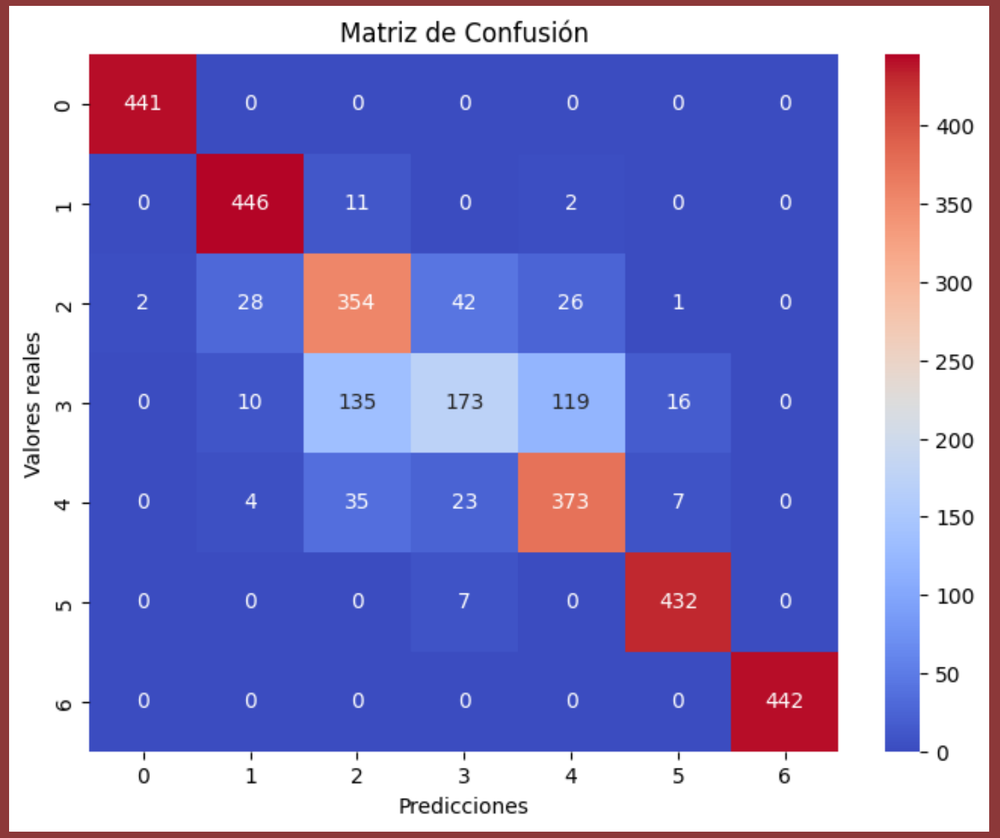
Tasa de aciertos 86.07%

'classifier': KNeighborsClassifier(),

'classifier__n_neighbors': 3

No supervisado PCA con Random Forest

Tasa de acierto 85.04%
classifier__max_depth': 10,
'classifier__min_samples_leaf': 2,
'classifier__n_estimators': 200,
'pca__n_components': 10,
'scaler': StandardScaler()

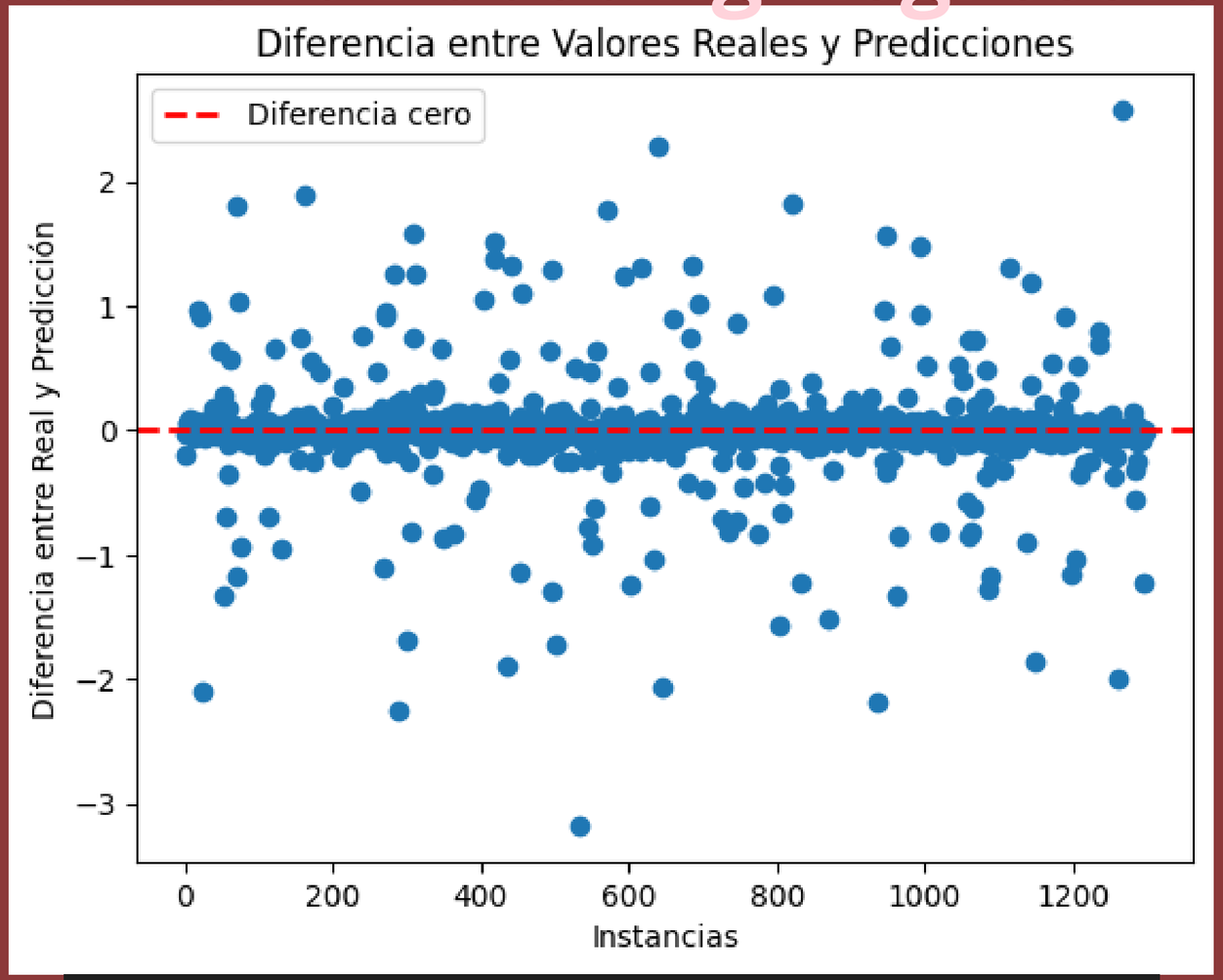


Regression

Una vez probados los modelos de clasificación, decidí probar con los modelos de regresión para ver otra forma para afrontar mi modelo.

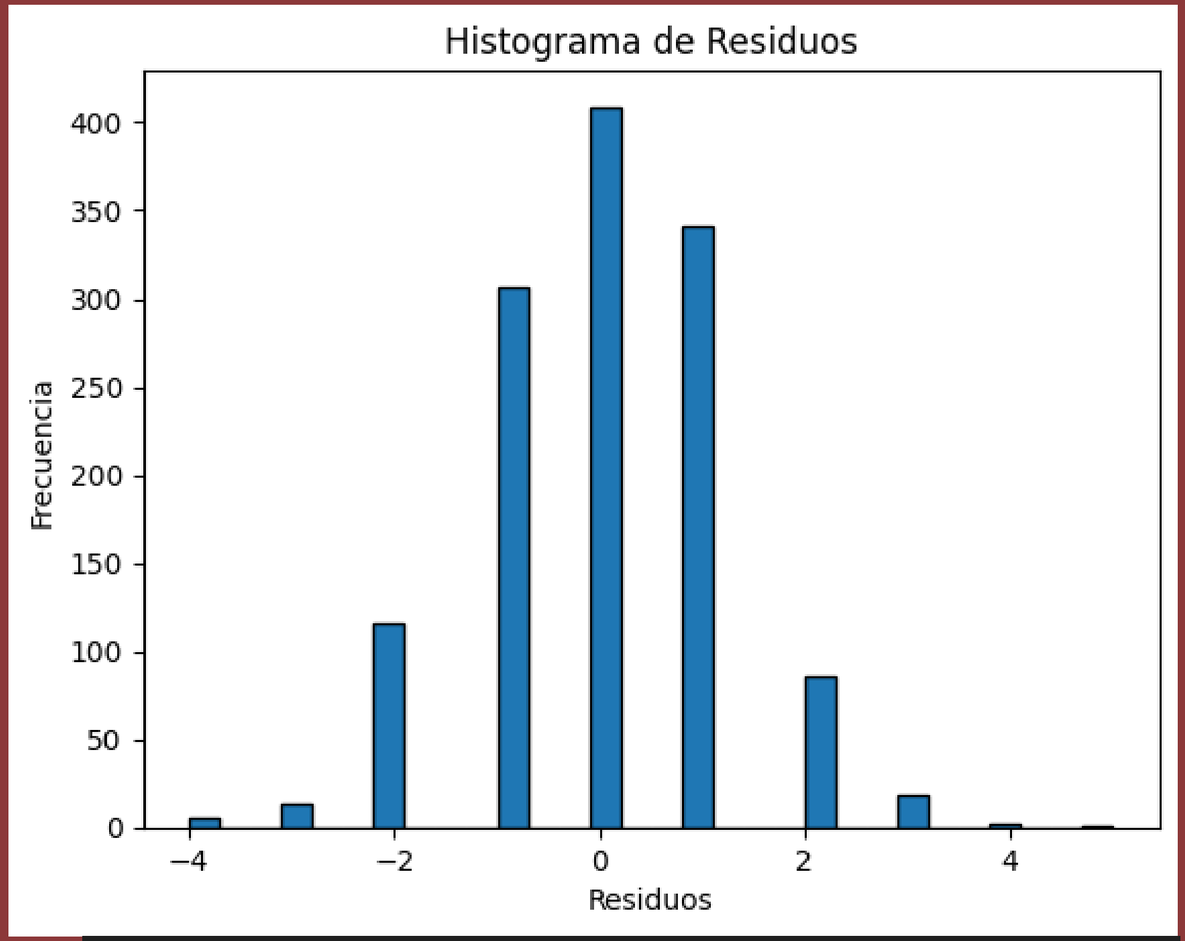
A continuación paso a enseñaros mi mejor modelo de regresión y el modelo final

Gradient Boosting Regression



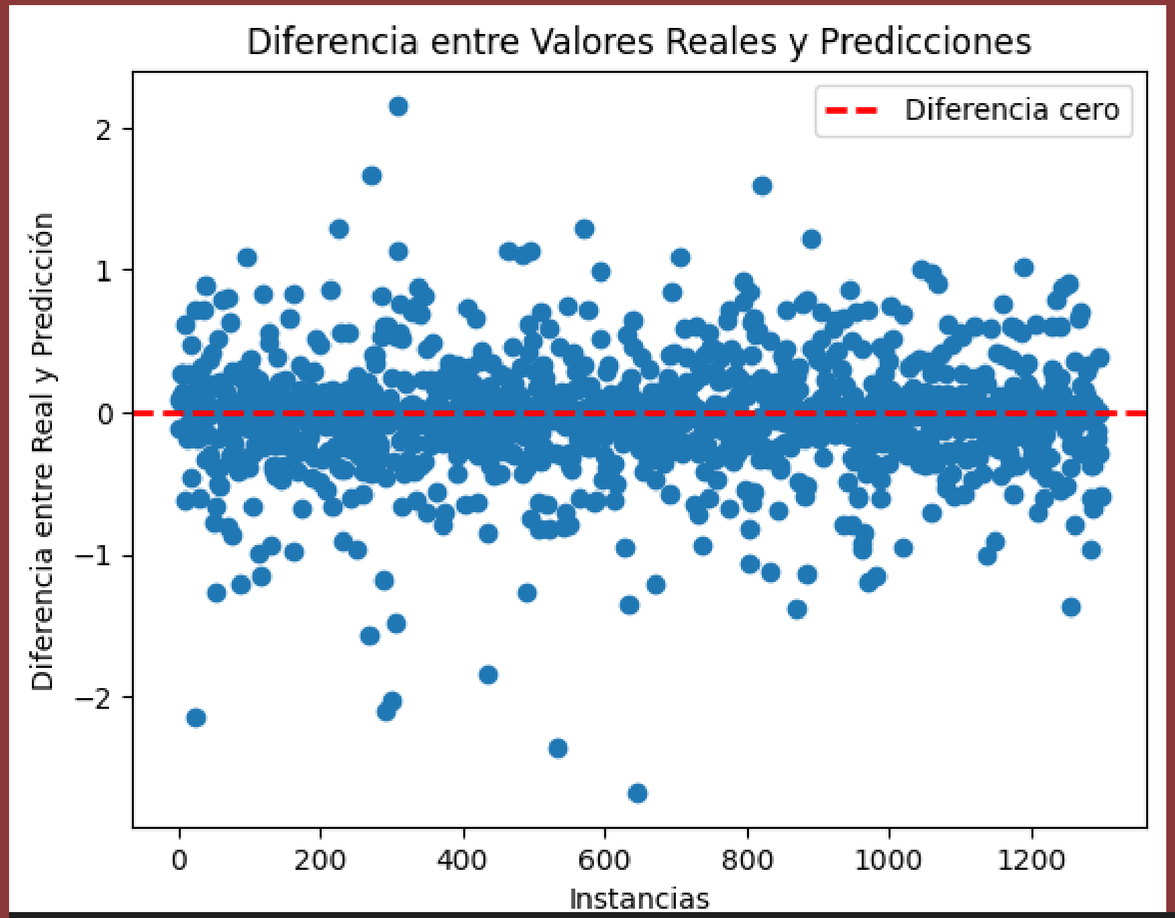

```
Mean Absolute Error (MAE): 0.1437630517740377
Mean Absolute Percentage Error (MAPE): 2.622878504956207%
Mean Squared Error (MSE): 0.13138993696312906
Root Mean Squared Error (RMSE): 0.3624774985611232
```

Decission Tree Regression



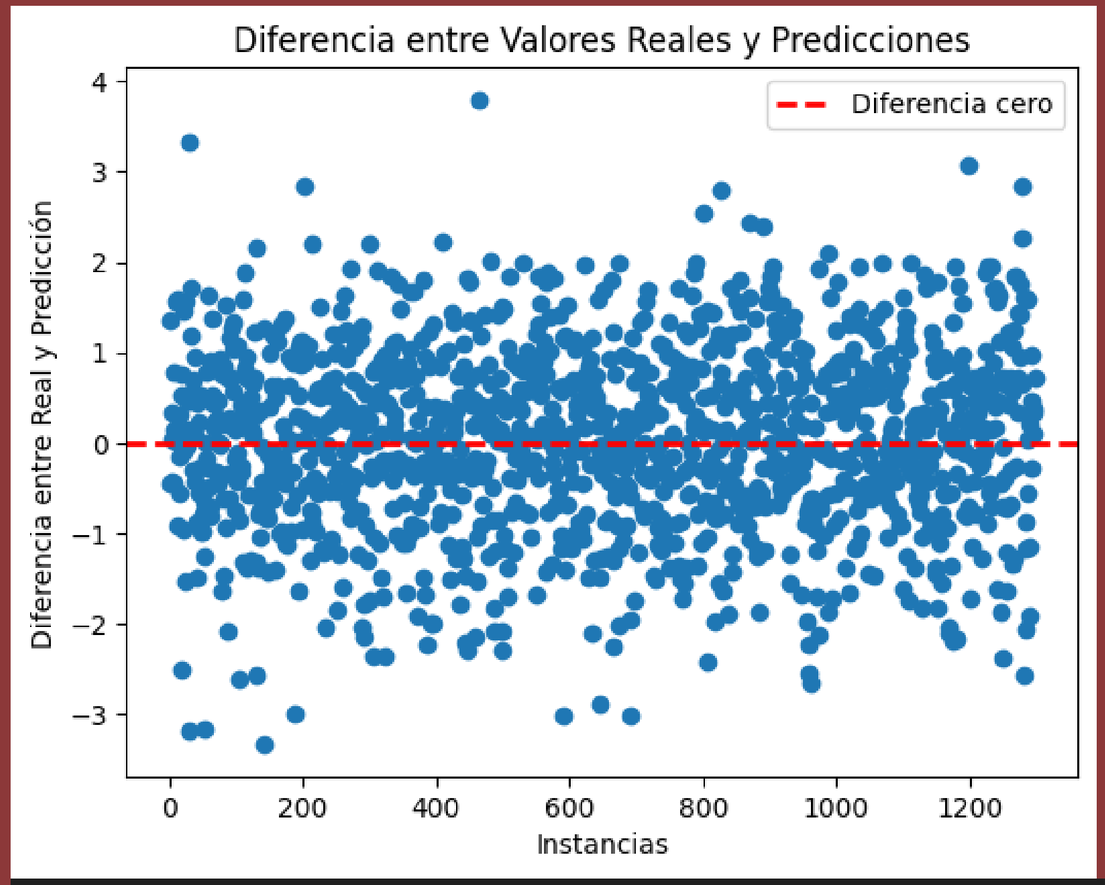

```
Mean Absolute Error (MAE): 0.9090909090909091
Mean Absolute Percentage Error (MAPE): 15.890105657054809%
Mean Squared Error (MSE): 1.448382126348228
Root Mean Squared Error (RMSE): 1.2034874849154966
```

Random Forest Regression



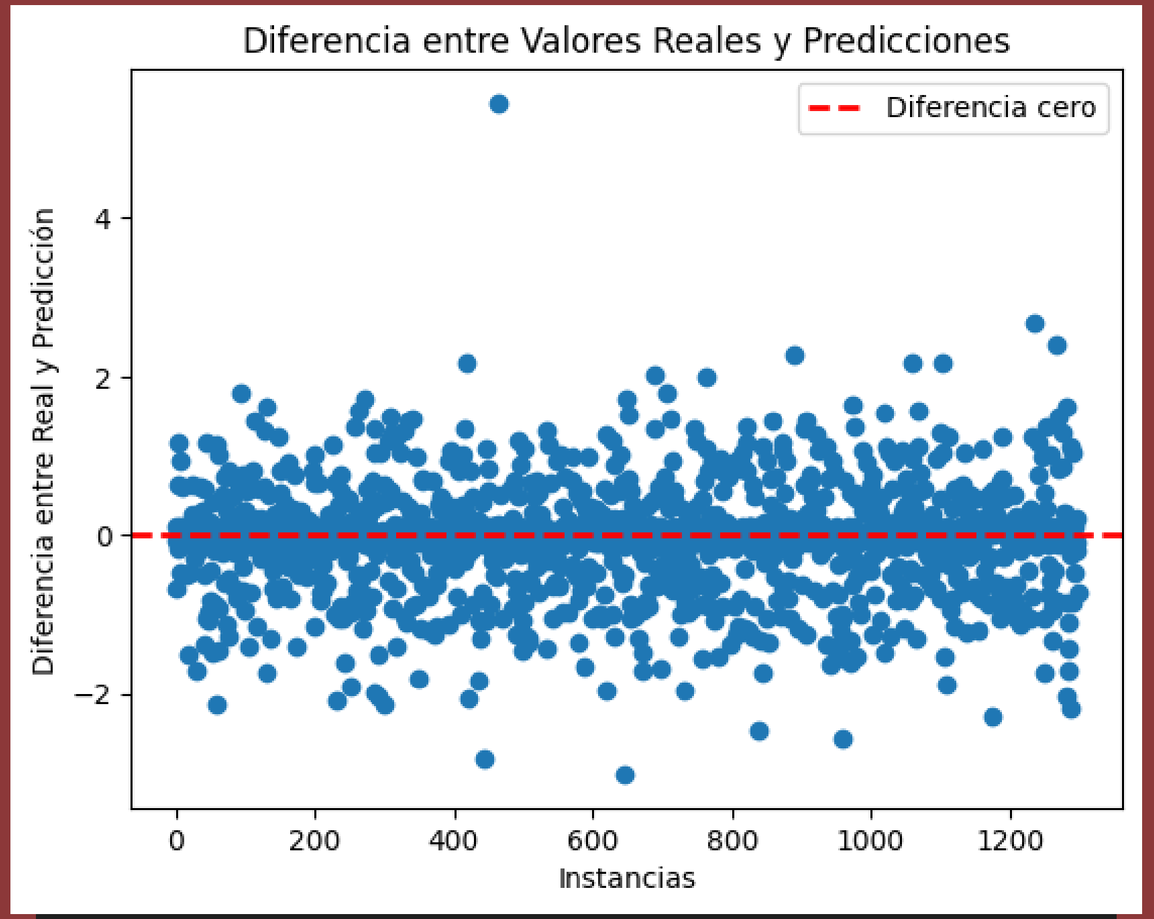

```
Mean Absolute Error (MAE): 0.26286238245014526
Mean Absolute Percentage Error (MAPE): 4.7258278744264395%
Mean Squared Error (MSE): 0.1605248141611384
Root Mean Squared Error (RMSE): 0.40065548063284795
```

LASSO



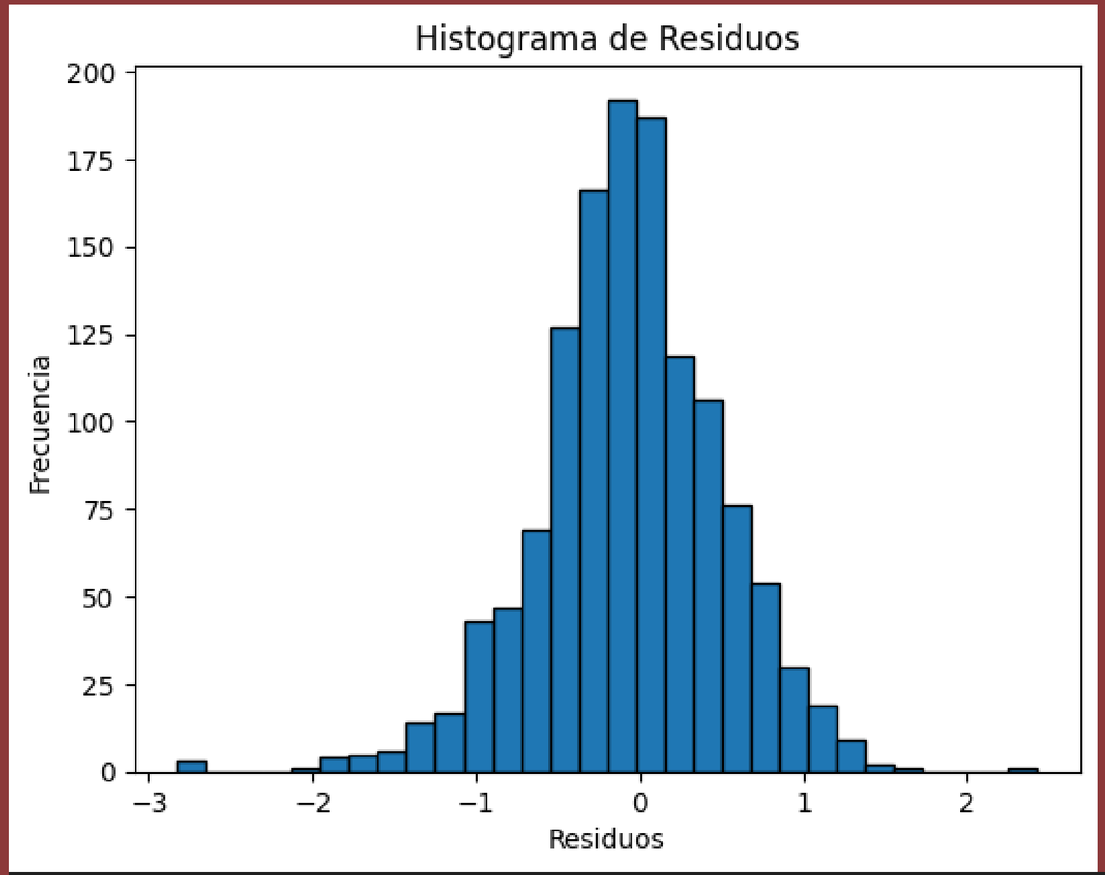

```
Mean Absolute Error (MAE): 0.7998389355733486
Mean Absolute Percentage Error (MAPE): 14.21285908986947%
Mean Squared Error (MSE): 1.0139882637452864
Root Mean Squared Error (RMSE): 1.0069698425202644
```

SVC



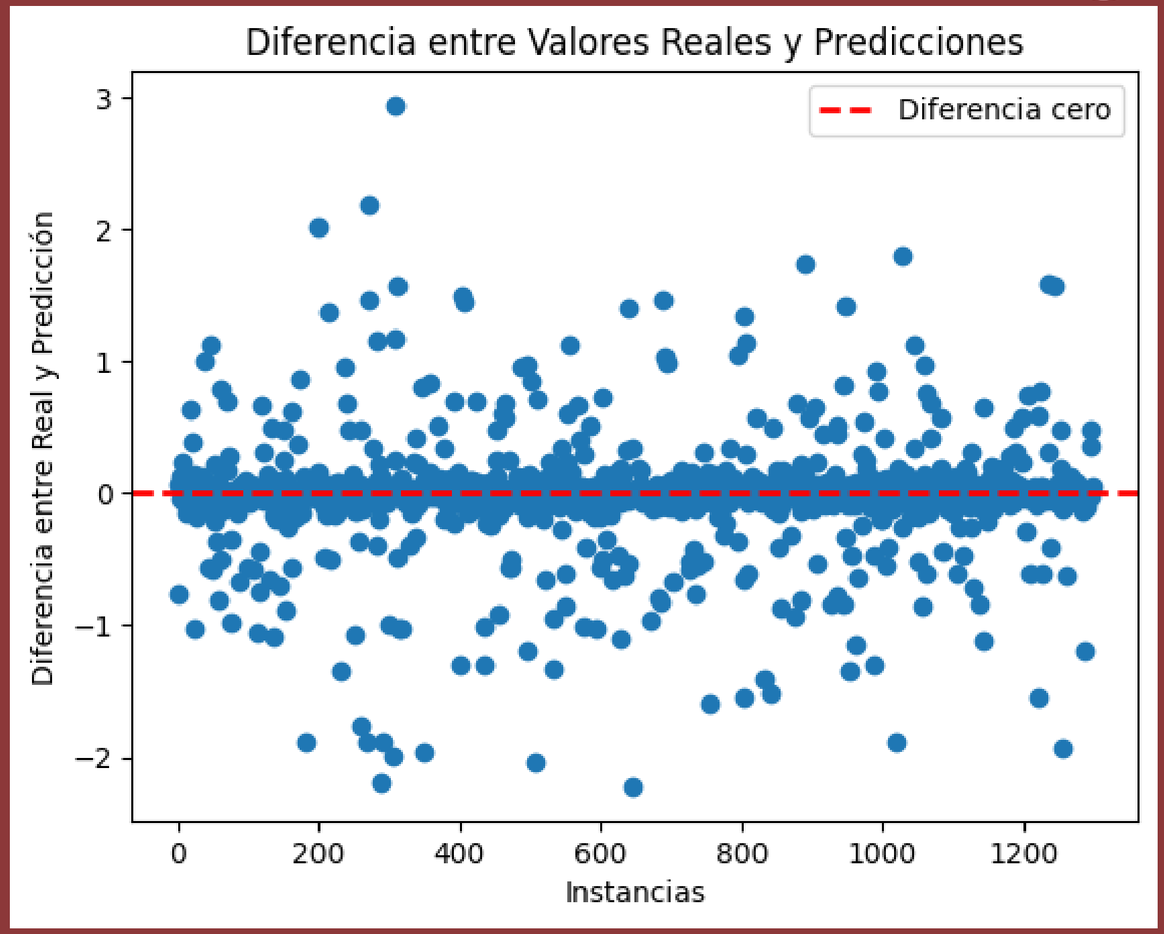

```
Mean Absolute Error (MAE): 0.4950699727764762
Mean Absolute Percentage Error (MAPE): 8.726739301198808%
Mean Squared Error (MSE): 0.5072335220166678
Root Mean Squared Error (RMSE): 0.7122032870021507
```

PCA + Random Fores




```
Mean Absolute Error (MAE): 0.4299380569533648
Mean Absolute Percentage Error (MAPE): 7.6386932085148755%
Mean Squared Error (MSE): 0.3241427107536108
Root Mean Squared Error (RMSE): 0.5693353236482089
```

PCA + Gradient Boosting




```
Mean Absolute Error (MAE): 0.18381300106680273
Mean Absolute Percentage Error (MAPE): 3.317348100547596%
Mean Squared Error (MSE): 0.15903451492197246
Root Mean Squared Error (RMSE): 0.39879131751076585
```

Para finalizar, hemos creado un Streamlit, donde podemos comprobar el funcionamiento de nuestro modelo, insertandole los parametros que queramos comprobar y que nos diga su prediccion.
Paso a enseñaros a utilizarlo