



Universidad Nacional de La Matanza

TRABAJO PRÁCTICO

1º CUATRIMESTRE - AÑO 2023

Corrales, Mauro; DNI 40137650

Kuczerawy, Damian; DNI 37807869

Accattoli, Matías Apolo; DNI 33477386

Lopez, German Agustín; DNI 38327209

Ripamonti, Franco; DNI 41543365

Sabes, Franco Damian; DNI 38168884

De Vito, Daniel Ricardo; DNI 38128772

Índice

Índice.....	1
Problema Regresión Lineal.....	2
Análisis de estructura.....	2
Matriz de correlación.....	3
Identificación y Determinación de Features Relevantes.....	4
Separación del Set de Datos.....	7
Normalización.....	7
Problema Regresión Logística.....	8
Observaciones Modelo Inicial.....	8
Selección de Features.....	8
Modelo primera iteración:.....	9
Modelo optimizado:.....	11
Balancear Set de Datos.....	12
Separación de Set de Datos.....	13
Normalización.....	13

Problema Regresión Lineal

Este dataset lo obtuvimos del repositorio de aprendizaje automático de UCI.

<https://archive.ics.uci.edu/ml/datasets/energy+efficiency>

Este estudio analizó la evaluación de los requisitos de carga de calefacción y carga de refrigeración de los edificios (es decir, la eficiencia energética) en función de los parámetros del edificio.

Realizamos análisis energéticos utilizando 12 formas de edificios diferentes. Los edificios se diferencian en cuanto a la superficie acristalada, la distribución de la superficie acristalada y la orientación, entre otros parámetros.

Link al colab:

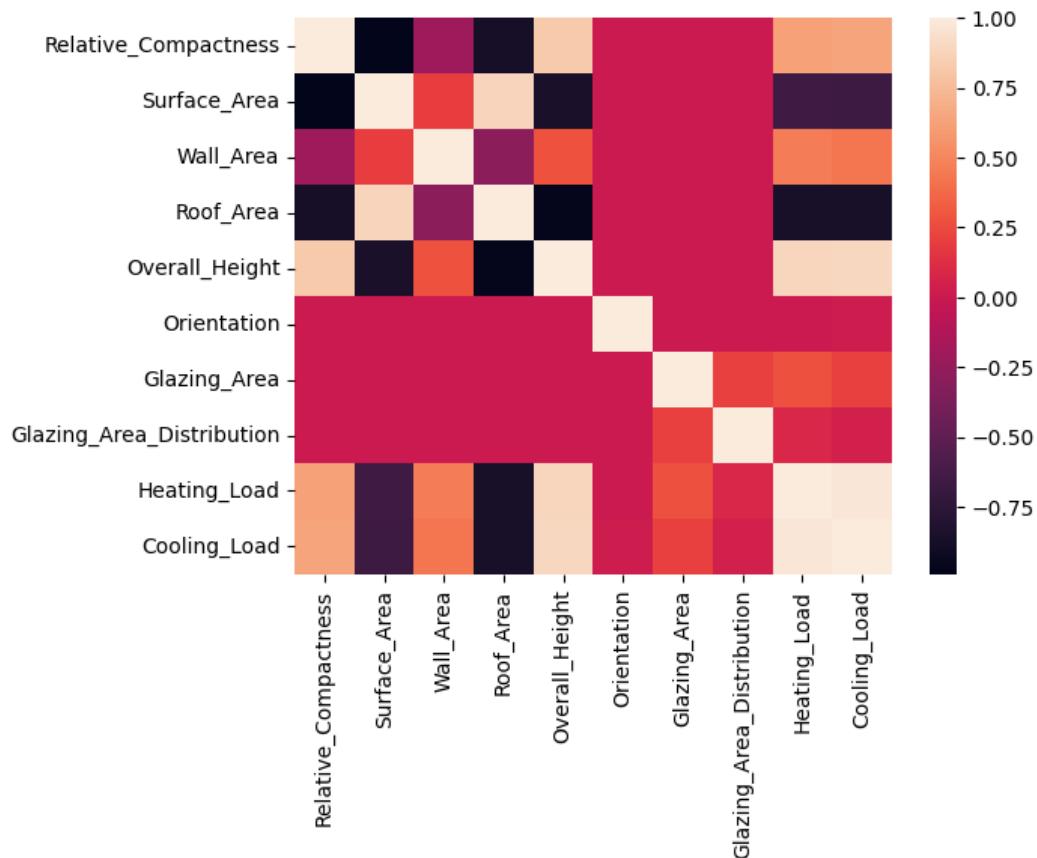
<https://colab.research.google.com/drive/1fbeQcbuEuvqykY6FU7H7AKbvdj6TIPLG?usp=sharing>

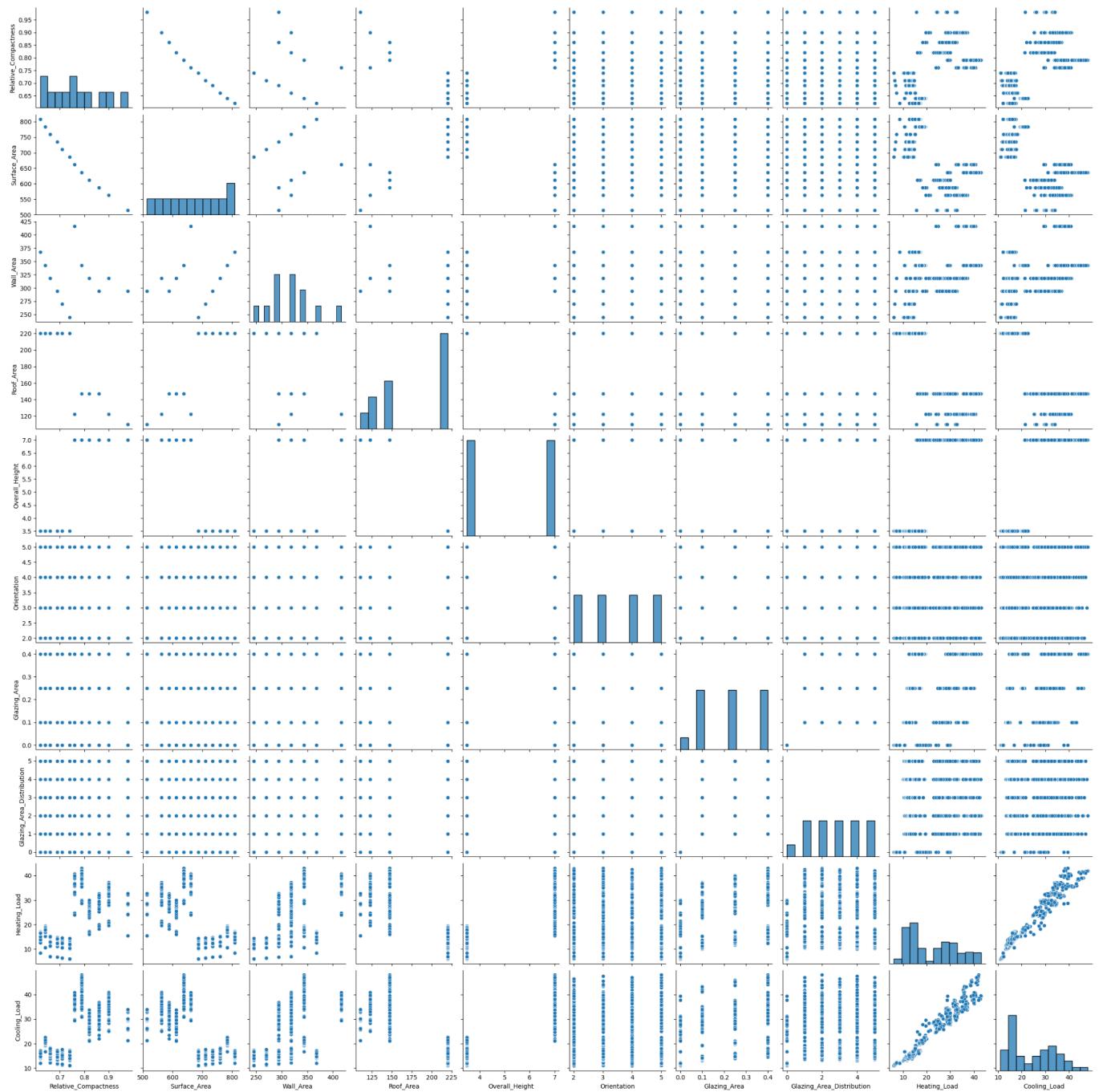
Análisis de estructura

Al analizar los datos se llegó a la conclusión de que existen 8 características a tener en cuenta dentro del set de datos. Estas son:

- Relative_Compactness
- Surface_Area
- Wall_Area
- Roof_Area
- Overall_Height
- Orientation
- Glazing_Area

Matriz de correlación





Identificación y Determinación de Features Relevantes

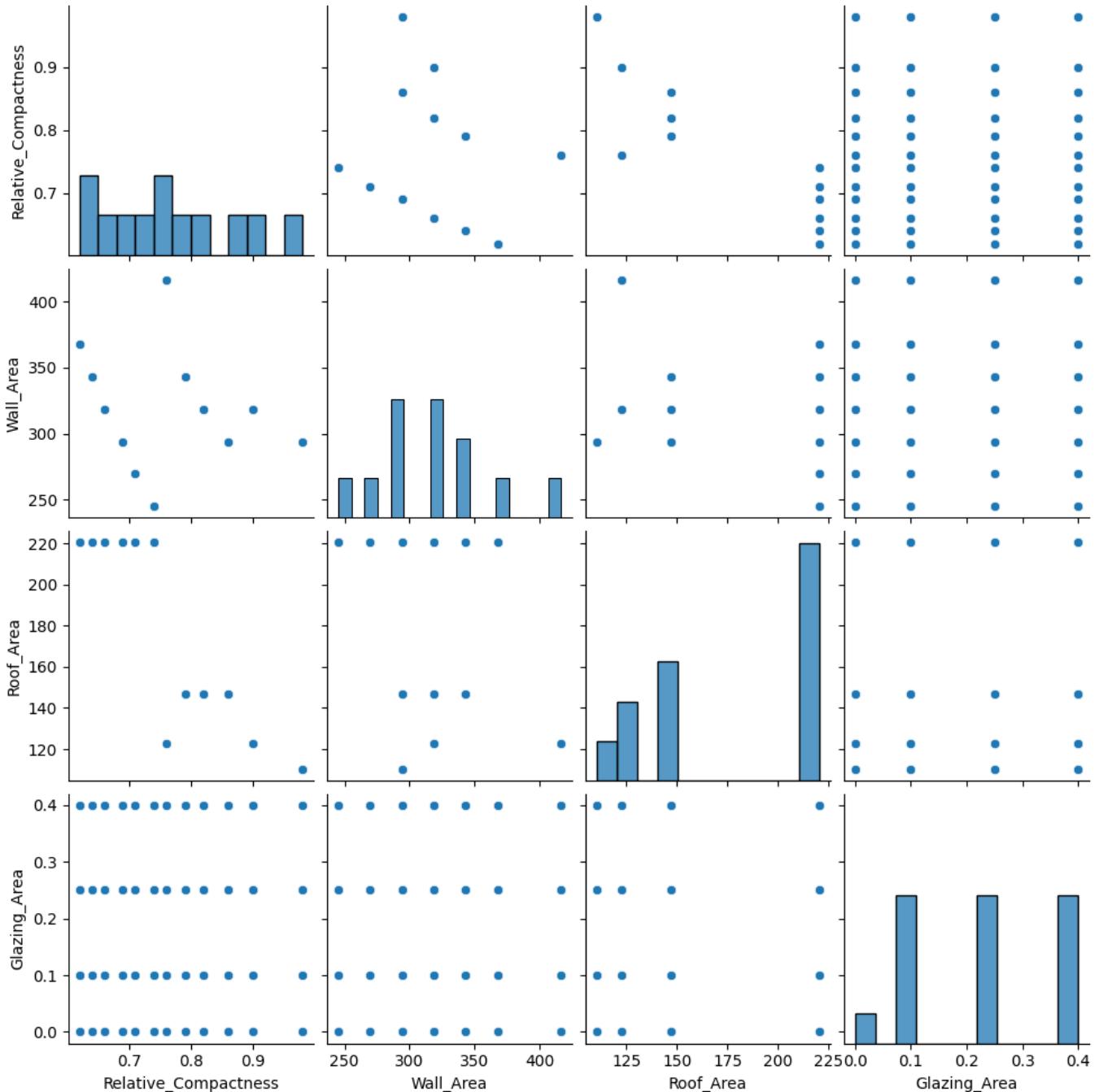
En esta sección, detallaremos los features relevantes que utilizamos tanto para el modelo inicial como para el modelo optimizado

Modelo inicial:

- Relative_Compactness
- Surface_Area
- Wall_Area

Primero decidimos seleccionar los features que tienen un valor de `mutual_info` mayor a 0.5

- Relative_Compactness
- Surface_Area
- Wall_Area
- Roof_Area
- Overall_Height
- Glazing_Area



Como el gráfico no se distingue visualmente los features relacionados, decidimos analizar la matriz de correlación de los features elegidos.

	Relative_Co mpactness	Surface_Are a	Wall_Area	Roof_Area	Overall_Hei ght	Glazing_Are a
Relative_Co mpactness	1.00	-0.99	-0.20	-0.87	0.83	0.00
Surface_Ar ea	-0.99	1.00	0.20	0.88	-0.86	0.00
Wall_Area	-0.20	0.20	1.00	-0.29	0.28	0.00
Roof_Area	-0.87	0.88	-0.29	1.00	-0.97	0.00
Overall_Hei ght	0.83	-0.86	0.28	-0.97	1.00	0.00
Glazing_Are a	0.00	0.00	0.00	0.00	0.00	1.00

Al analizar la matriz de correlación, se elimina Surface_Area ya que presenta una relación casi del 0.9 con respecto a roof_area

Modelo Optimizado:

- Relative_Compactness
- Wall_Area
- Roof_Area
- Overall_Height
- Glazing_Area

¿Si reducimos el número de feature cols a analizar por nuestro modelo obtendremos una mayor precisión al evaluar el mismo?

Si es respecto a los features cols del modelo inicial propuesto por la cátedra, la respuesta es no, ya que incluso en el modelo inicial faltan features para que la predicción sea acertada, ya que el error es muy alto y el score es bajo.

Respecto a los features de la primera iteración si, se analizó el modelo y se detectó que habían dos features muy relacionados entre sí, y al eliminar uno de ellos el modelo aumentó levemente su precisión.

Separación del Set de Datos

¿Aumenta el score y disminuye el error en la etapa de evaluación del modelo?

No, aumenta hasta el 0,3, a partir de este límite si se sigue aumentando el set de datos de pruebas aumenta el error y disminuye el Score ya que la cantidad de datos para entrenar el modelo es baja.

Algunos ejemplos:

```
set_test_size = 0.2
```

```
MSE: 10.739891011706018
regressor.coef_: [-6.06646652 -0.78979964 -7.30362823 2.68907798 7.0707143]
Score: 0.9027415226565492
```

```
set_test_size = 0.3
```

```
MSE: 9.460080231897313
regressor.coef_: [-5.95521524 -0.70816302 -7.24881461 2.65713476 6.98036033]
Score: 0.910012652618275
```

```
set_test_size = 0.4
```

```
MSE: 9.868774920079916
regressor.coef_: [-5.92860657 -0.77547169 -6.78458723 2.58602463 7.52782904]
Score: 0.9020972392414217
```

Normalización

¿Aumenta el score y disminuye el error en la etapa de evaluación del modelo?

Generalmente, normalizar el set de datos lleva a mejoras en el score y disminución del error en los modelos, sin embargo para nuestro modelo el score y error sin normalizar es idéntico al que se obtiene al probar el modelo luego de normalizar los datos.

Problema Regresión Logística

Este dataset lo obtuvimos del siguiente sitio: <https://www.muratkoklu.com/datasets/>

Clasificación de frutos de dátiles en variedades genéticas. En todo el mundo se cultiva una gran cantidad de frutas, cada una de las cuales tiene varios tipos. Los factores que determinan el tipo de fruto son las características de la apariencia externa como el color, la longitud, el diámetro y la forma. La apariencia externa de los frutos es un determinante importante del tipo de fruto. Determinar la variedad de frutas observando su apariencia externa puede requerir experiencia, lo que lleva mucho tiempo y requiere un gran esfuerzo. El objetivo de este estudio es clasificar los tipos de dátiles, es decir, Barhee, Deglet Nour, Sukkary, Rotab Mozafati, Ruthana, Safawi y Sagai, utilizando tres métodos diferentes de aprendizaje automático. Se extrajeron un total de 34 características, incluidas características morfológicas, forma y color.

Link al colab:

<https://colab.research.google.com/drive/1HAZxdme14iA6EII6KLwwb33Io1ZhS9xG?usp=sharing>

Observaciones Modelo Inicial

¿Por qué los valores predecidos por nuestro modelo para el set de datos de prueba nos retorna como resultados de predicción una única clase?

Porque los datos no están normalizados. Esto se ve reflejado ya que los parámetros son muy representativos de la clase 4 y además es una de las que más entradas tiene en el dataset por lo que se produce un sesgo de una contra las demás.

¿Por qué la matriz de confusión solo nos devuelve una columna completa?

La matriz de confusión muestra la relación entre la clase esperada y la clase predecida, y como el modelo predijo siempre la misma clase, en la matriz de confusión se refleja este comportamiento visualizando valores únicamente en la única clase que predijo el modelo.

¿Por qué el score al evaluar nuestro modelo es tan bajo?

Porque para todo el set de datos, el modelo predice una única clase de las 7 clases posibles, entonces la tasa de error es muy alta, eso hace que el score del modelo sea tan bajo.

Selección de Features

En esta sección, detallaremos los features relevantes que utilizamos tanto para el modelo inicial como para el modelo optimizado

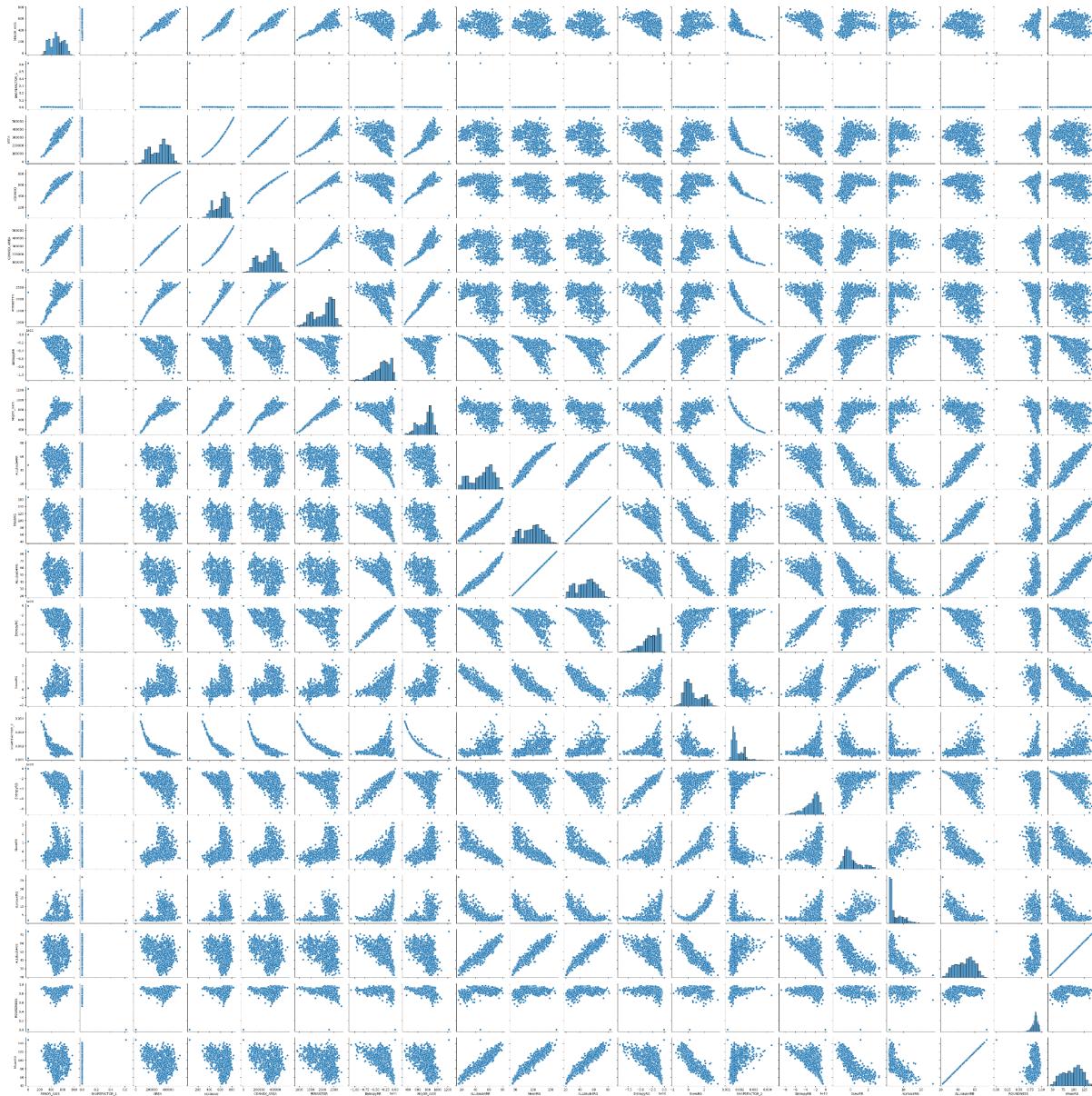
Modelo inicial:

- MINOR_AXIS
- SHAPEFACTOR_1
- AREA
- EQDIASQ
- CONVEX_AREA
- PERIMETER
- EntropyRR
- MAJOR_AXIS

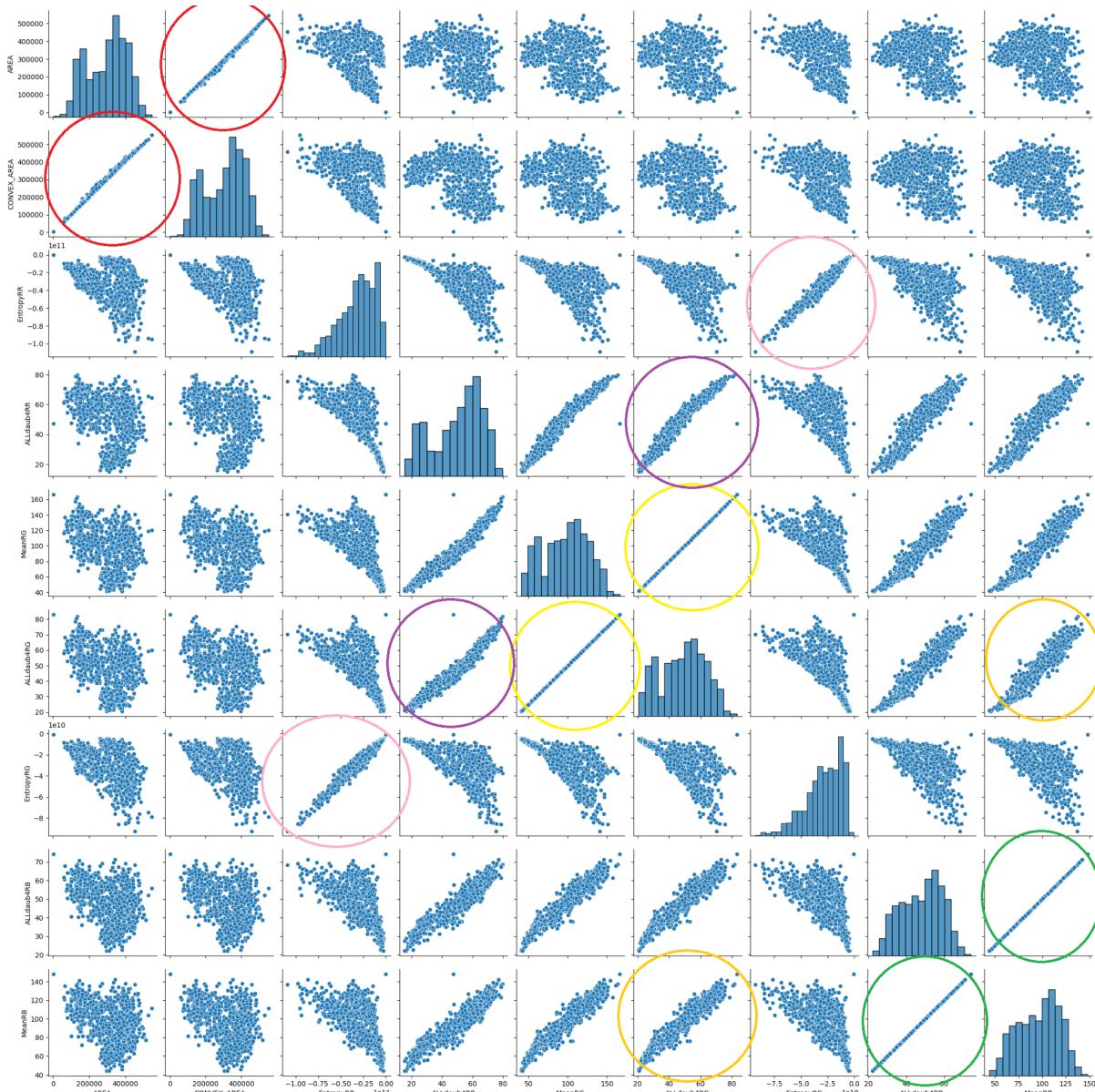
Primero decidimos seleccionar los features que tienen un valor de mutual_info mayor a 0.5

Modelo primera iteración:

- MINOR_AXIS
- SHAPEFACTOR_1
- AREA
- EQDIASQ
- CONVEX_AREA
- PERIMETER
- EntropyRR
- MAJOR_AXIS
- ALLdaub4RR
- MeanRG
- ALLdaub4RG
- EntropyRG
- SkewRG
- SHAPEFACTOR_2
- EntropyRB
- SkewRR
- KurtosisRG
- ALLdaub4RB
- ROUNDNESS
- MeanRB



A partir de este gráfico fuimos eligiendo features que se ven en el gráfico una aproximación a la recta $y=x$, esto significa que entre esas dos features existe una relación muy estrecha, pues el coeficiente de correlación es muy cercano a +1, por lo que se estaría brindando información redundante al modelo, con el riesgo de que genere ruido.



Se seleccionan, comparan y eliminan las features:

- Entre “Area” y “ConvexArea”. Color rojo. Se elimina Area.
- Entre “EntropyRG” y “EntropyRR”. Color rosa. Se elimina EntropyRG.
- Entre “ALLdaub4RG” y “ALLdaub4RR”. Color violeta. Se elimina ALLdaub4RR.
- Entre “meanRG” y “ALLdaub4RG”. Color amarillo. Se elimina meanRG.
- Entre “ALLdaub4RG” y “meanRB”. Color naranja. Se elimina ALLdaub4RG.
- Entre “meanRB” y “ALLdaub4RB”. Color verde. Se elimina meanRB.

Al realizar esto notamos que el score mejoró.

Modelo optimizado:

- MINOR_AXIS
- SHAPEFACTOR_1
- EQDIASQ

- CONVEX_AREA
- PERIMETER
- EntropyRR
- MAJOR_AXIS
- SkewRG
- SHAPEFACTOR_2
- EntropyRB
- SkewRR
- KurtosisRG
- ALLdaub4RB
- ROUNDNESS

¿Si aumentamos el número de feature cols a analizar por nuestro modelo obtendremos una mayor precisión al evaluar el mismo?

Si, al aumentar el número de feature cols se aumenta la precisión siempre y cuando los features seleccionados están relacionados entre sí con el feature que tiene que predecir el modelo. Esto se puede evaluar por ejemplo mediante el coeficiente de correlación. Al aumentar el número de features, se brindan más datos acerca de las clases que se quieren identificar y brinda al modelo más herramientas para clasificar con más precisión las clases. Sin embargo hay que evitar agregar demasiadas features ya que puede llevar al overfitting.

Balancear Set de Datos

Para el balanceo del set de datos se probaron técnicas de Oversampling y Undersampling. Undersampling reduce el número de entradas de todas las clases a aquel de la clase que tiene menos elementos en el dataset:

```
Training target statistics: Counter({0: 43, 1: 43, 2: 43, 3: 43, 4: 43, 5: 43, 6: 43})
Testing target statistics: Counter({0: 22, 1: 22, 2: 22, 3: 22, 4: 22, 5: 22, 6: 22})
```

Con esta técnica obtuvimos el siguiente score en el modelo final:

```
FRUITS DATASET - ACCURACY SCORE BY METRICS: 0.8831168831168831
FRUITS DATASET - SCORE BY REGRESSOR SCORE: 0.8831168831168831
```

Oversampling genera elementos nuevos para que todas las clases tengan un número de entradas igual al de la clase que tiene más elementos en el dataset:

```
Training target statistics: Counter({0: 142, 5: 142, 6: 142, 1: 142, 3: 142, 4: 142, 2: 142})
Testing target statistics: Counter({5: 62, 0: 62, 1: 62, 6: 62, 4: 62, 2: 62, 3: 62})
```

Con esta técnica obtuvimos el siguiente score en el modelo final:

```
FRUITS DATASET - ACCURACY SCORE BY METRICS: 0.8225806451612904
FRUITS DATASET - SCORE BY REGRESSOR SCORE: 0.8225806451612904
```

Como se puede observar, Undersampling arroja mejores resultados, por lo que es la técnica de balanceo que elegimos para nuestro modelo.

Separación de Set de Datos

¿Aumenta el score y disminuye el error en la etapa de evaluación del modelo?

Obtuvimos mejor resultado al separar el set de datos destinando un 30% para test, pasado ese valor el score disminuye.

Algunos ejemplos:

`set_test_size = 0.2`

```
FRUITS DATASET - ACCURACY SCORE BY METRICS: 0.8452380952380952
FRUITS DATASET - SCORE BY REGRESSOR SCORE: 0.8452380952380952
```

`set_test_size = 0.3`

```
FRUITS DATASET - ACCURACY SCORE BY METRICS: 0.8831168831168831
FRUITS DATASET - SCORE BY REGRESSOR SCORE: 0.8831168831168831
```

`set_test_size = 0.4`

```
FRUITS DATASET - ACCURACY SCORE BY METRICS: 0.7783251231527094
FRUITS DATASET - SCORE BY REGRESSOR SCORE: 0.7783251231527094
```

Normalización

¿Aumenta el score y disminuye el error en la etapa de evaluación del modelo?

No solo aumenta el score y disminuye el error sino que es necesaria la normalización para que este modelo con este set de datos y los features elegidos inicialmente prediga de manera correcta.

Esto se observa comparando las matrices de confusión:

Izquierda: modelo con todos nuestros cambios aplicados pero sin normalizar los datos

Derecha: modelo final con todos nuestros cambios y datos normalizados.

