M2.851 - Tipología y ciclo de vida de los datos

Gabriel Loja Rodas

Práctica 2

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

Este dataset contiene datos sobre diferentes rojas y blancas del vino portugués vinho verde, un producto de la región noroeste de Portugal. Este tipo de vino abarca el 15% de la producción total en este país. Los datos fueron recolectados desde mayo del 2004 hasta febrero de 2007, los datos psico químicos fueron recolectados por maquinaria especial, mientras que la variable sensorial fue dada mediante el cálculo de la media de tres evaluaciones de catadores profesionales en un rango de 1 al 10.

Este dataset es importante porque permite conocer y explicar la relación entres los atributos psicoquimicos del vino y el sabor del vino, de manera que los fabricantes puedan reconocer patrones y obtengan un mejor vino.

En la presente practica se pretende analizar y limpiar los datos, a partir de esto se obtendrá un modelo que permita predecir la calidad del vino en base a los atributos relevantes también identificados dentro de la presente práctica. Todos estos procesos los realizaremos mediante Python usando las librerías disponibles para análisis de datos.

2. Integración y selección de los datos de interés a analizar.

En esta fase no fue necesario realizar un proceso de integración de los datos, puesto que los datos publicado ya contienen todos los datos necesarios.
Par tener una idea general de los datos que vamos a manejar se muestra un listado de los primeros registros. Aquí podemos observar los atributos psicoquimicos y el atributo sensorial (quality).

```
vinos = pd.read_csv('winequality-red.csv')
vinos.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Podemos ver también que el dataset está conformado de 1599 registros y 12 atributos.

```
vinos.shape
```

```
(1599, 12)
```

De la misma manera para tener una idea clara de los datos se muestra un resumen estadístico de cada atributo. Aquí podemos ver que todos los atributos tienen 1599 registros, también se puede notar que todos son de tipo numérico.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

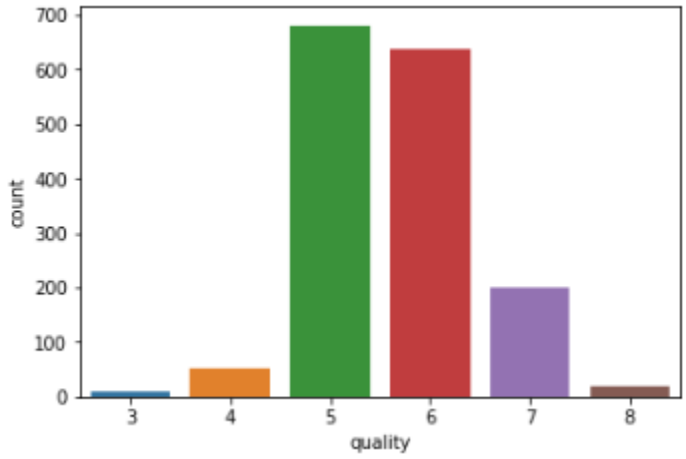
En cuanto a datos duplicados podemos ver que existen 240 registros que se repiten.

```
duplicados = vinos[vinos.duplicated()]
duplicados.shape
```

```
(240, 12)
```

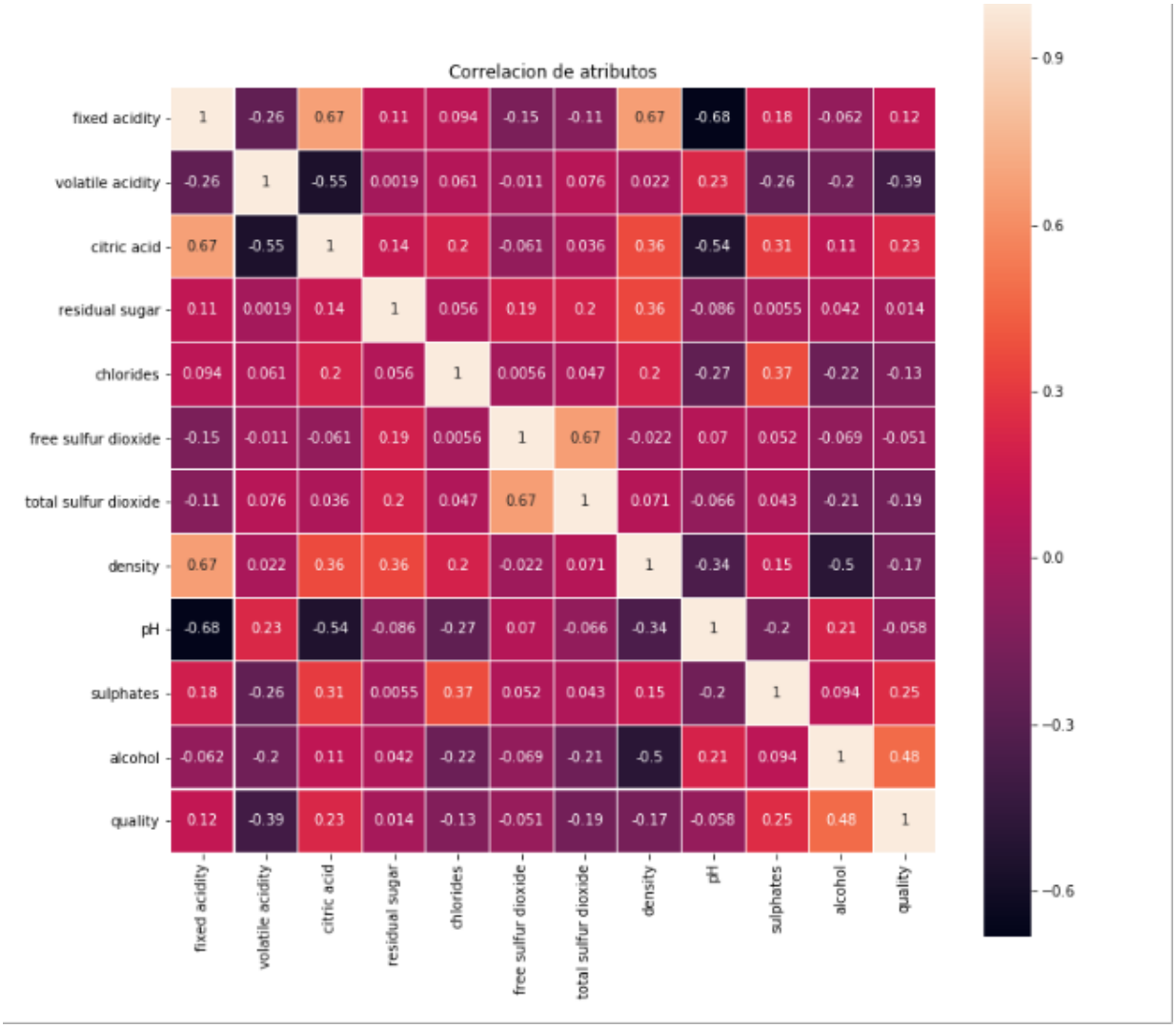
En el presente análisis consideramos que es necesario conservar los registros duplicados debido a que corresponden a evaluaciones iguales de un vino con una misma composición psico química.

Dentro del procesos de selección de datos consideramos que los 11 atributos psicoquimicos del vino son relevantes para el presente análisis, debido a que el objetivo del estudio es identificar cuales son los atributos que mas influyen en el sabor del vino. En lo que tiene que ver con el sabor del vino (quality) tenemos una categoría del 1 al 10 y dentro del dataset tenemos los siguientes valores.



La mayoría de los vinos tienen un sabor (quality) en la categoría media 5 y 6. Hay pocos vinos que saben o muy bien o muy mal.
Para identificar fuertes correlaciones entre atributos realizamos un gráfico de correlación. Del grafico podemos notar que el alcohol es el atributo que tiene la mayor correlación con el sabor del vino.

```
colormap = plt.cm.inferno
plt.figure(figsize=(12,12))
plt.title('Correlacion de atributos')
sns.heatmap(data.astype(float).corr(),
            linewidths=0.1,
            vmax=1.0,
            square=True,
            linecolor='white',
            annot=True)
```



3. Limpieza de los datos.

3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Mediante un procedimiento en Python podemos detectar que ninguna de las columnas del dataset contienen datos faltantes. Si existiesen valores faltantes realizaríamos un proceso de aproximación de los valores por regresión tomando en cuenta valores vecinos.

```
miss_values_count = vinos.isnull().sum(min_count=1)
miss_values_count = miss_values_count[miss_values_count != 0]

print(f"Número de columnas con datos faltantes: {miss_values_count.shape[0]}")
if miss_values_count.shape[0]:
    print("Recuento de valores nulos por columna: ")
    for name, miss_vals in miss_values_count.items():
        p = miss_vals > 1
        print(f"  - A la columna '{name}' le falta{'n' if p else ''} "
              f"{miss_vals} dato{'s' if p else ''}.")
```

```
Número de columnas con datos faltantes: 0
```

En cuanto a valores en 0 tenemos el siguiente análisis que calcula el porcentaje de registros que tienen valores de 0 por cada columna del dataset. Podemos ver que solo la columna tiene valores en 0, específicamente 8.2% de los registros. Para tratar estos valores hicimos una investigación rápida en la que se encontró que el acido cítrico se encuentra en pequeñas cantidades en el vino y que es normal que no lo contenga, debido a esto mantenemos estos valores. Para cada columna de los atributos psico químicos en el caso de presentarse valores en 0 habría que realizar una investigación para determinar si es valor normal o no para un vino. Si el valor en cero se da en el atributo quality lo descartaríamos porque se estableció que el rango de evaluación iba desde el 1.

```
print(vinos[vinos == 0].count(axis=0)/len(vinos.index))
```

```
fixed acidity           0.000000
volatile acidity        0.000000
citric acid             0.082552
residual sugar          0.000000
chlorides               0.000000
free sulfur dioxide     0.000000
total sulfur dioxide    0.000000
density                 0.000000
pH                      0.000000
sulphates               0.000000
alcohol                 0.000000
quality                 0.000000
dtype: float64
```

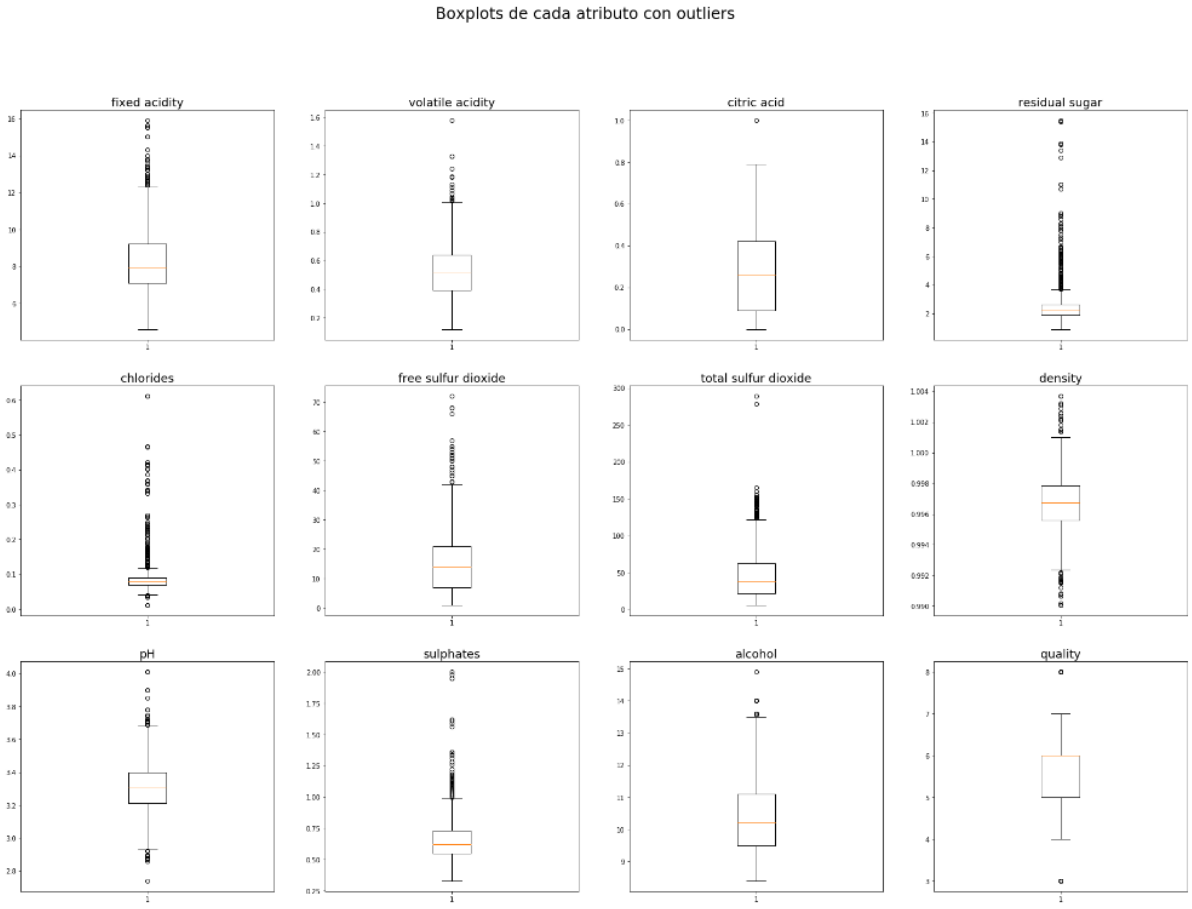
3.2. Identificación y tratamiento de valores extremos.

Para la identificación de valores extremos utilizaremos los gráficos de boxplot, mediante los cuales podemos observar que todos los atributos tienen outliers, denotados por un círculo en los gráficos, debido a que pueden

interferir con el análisis se tomó la decisión de eliminar los registros con estos valores.

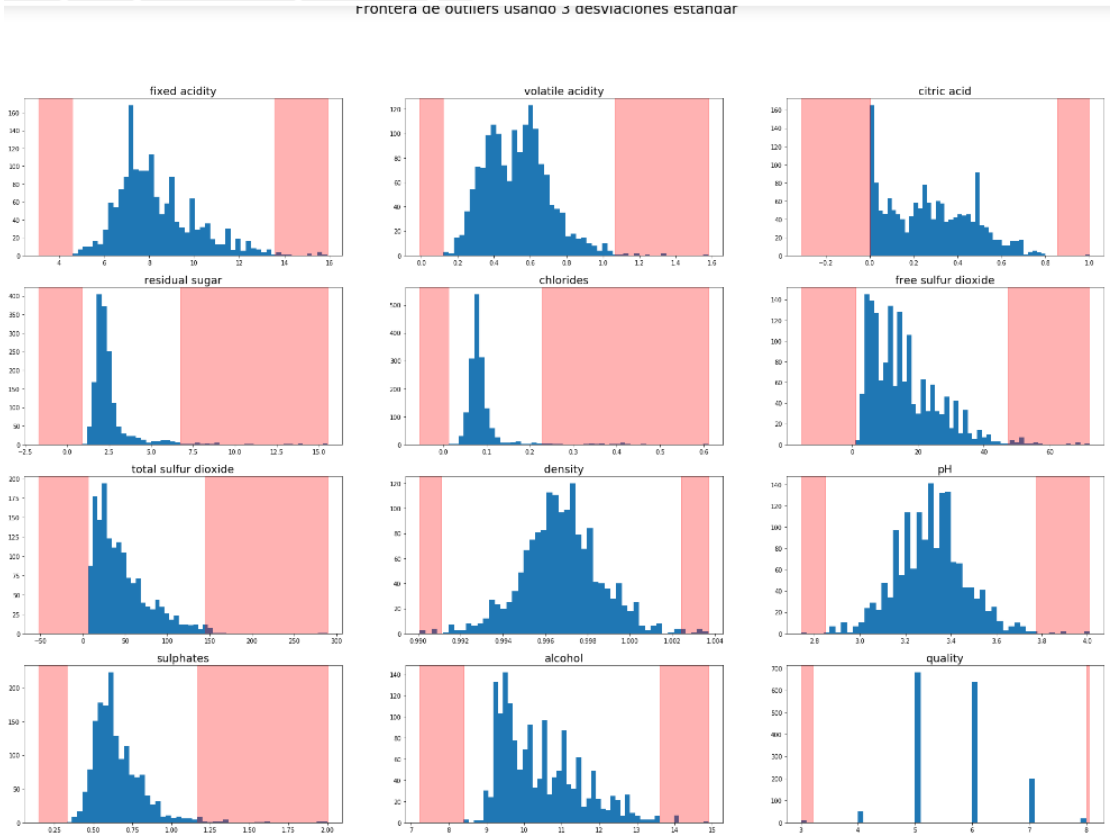
Obtuvimos los boxplots con el siguiente código:

```
plt.figure(figsize=(32,22))
plt.suptitle('Boxplots de cada atributo con outliers',fontsize=24)
for i in range(1,vinos.shape[1]+1):
    plt.subplot(2,6,i)
    plt.boxplot(vinos.iloc[:,i-1])
    plt.title(vinos.columns[i-1],fontsize=18)
```



Entonces se procedio a eliminar los outliers filtrando los valores que esten fuera del rango de 3 desviaciones estandar, en el siguiente grafico se puede observar graficamente los rangos en color rojo de los valores a eliminar. El grafico se obtuvo con el siguiente código:

```
plt.figure(figsize=(32,22))
plt.suptitle('Frontera de outliers usando 3 desviaciones estandar',fontsize=24)
for i in range(1,vinos.shape[1]+1):
    feature = vinos.iloc[:,i-1]
    mean = feature.mean()
    std_3 = feature.std()*3
    lower, upper = mean-std_3,mean+std_3
    plt.subplot(4,3,i)
    plt.hist(vinos.iloc[:,i-1],bins=50)
    plt.title(vinos.columns[i-1],fontsize=18)
    plt.axvspan(feature.min(),lower,color='red',alpha=0.3)
    plt.axvspan(upper,feature.max(),color='red',alpha=0.3)
```

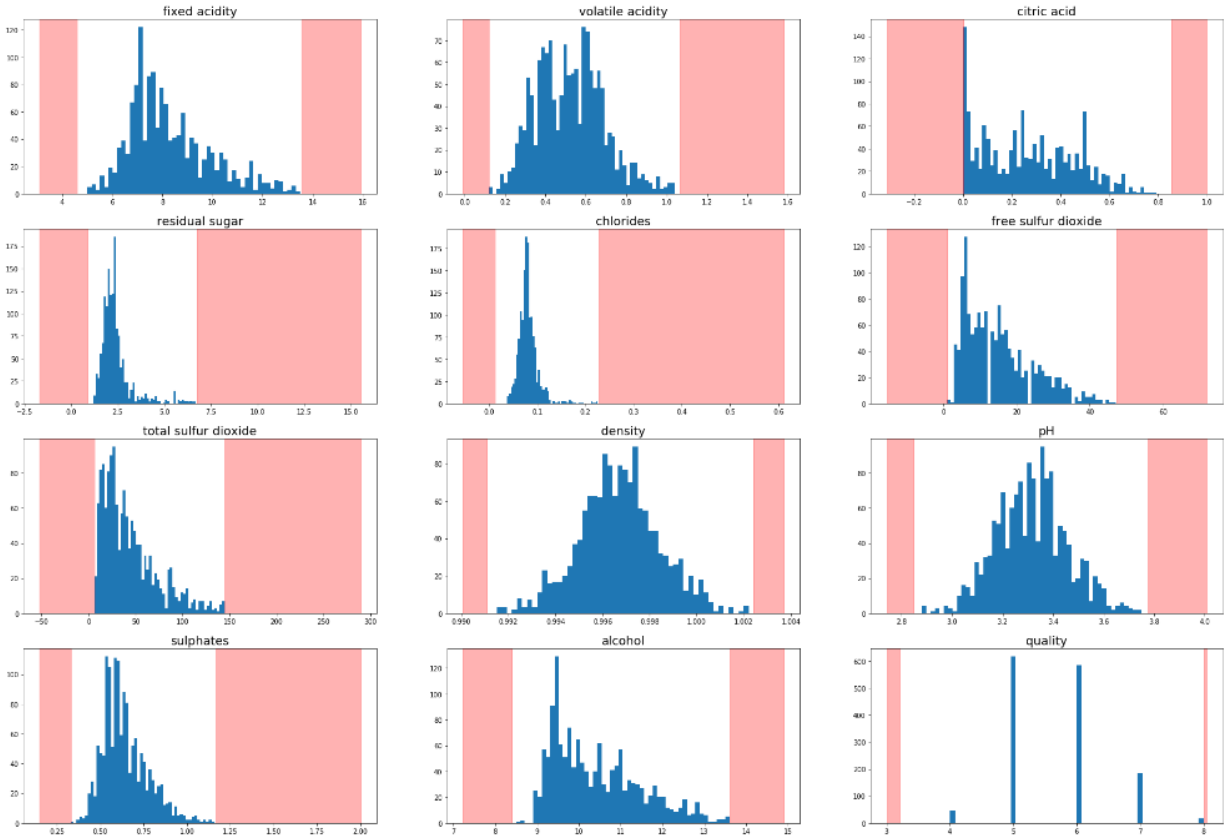


Realizamo el proceso de eliminado de outliers

```
outliers = vinos[np.abs(vinos[vinos.columns]-vinos[vinos.columns].mean())  <=  3 * vinos[vinos.columns].std()]
vinos_sin_outliers = outliers.dropna()
plt.figure(figsize=(32,22))
plt.suptitle('Outliers eliminados usando 3 desviaciones estandar',fontsize=24)
for i in range(1,vinos.shape[1]+1):
    feature = vinos.iloc[:,i-1]
    mean = feature.mean()
    std_3 = feature.std()*3
    lower, upper = mean-std_3,mean+std_3
    plt.subplot(4,3,i)
    plt.hist(vinos_sin_outliers.iloc[:,i-1],bins=50)
    plt.title(vinos.columns[i-1],fontsize=18)
    plt.axvspan(feature.min(),lower,color='red',alpha=0.3)
    plt.axvspan(upper,feature.max(),color='red',alpha=0.3)
```

Podemos ver en el siguiente gráfico como se han eliminado los outliers

Outliers eliminados usando 3 desviaciones estandar



4. Análisis de los datos.

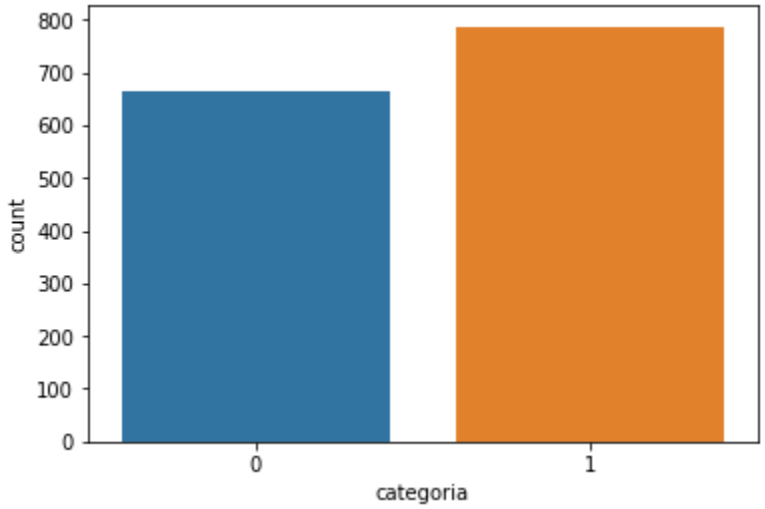
4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

Se procederá a agrupar los datos en dos grupos: el primero será de un buen vino (valor 1) que tiene un valor en quality mayor a 5, el segundo grupo será un mal vino (valor 0) que tendrá valor en quaility menor o igual a 5.

Usamos la siguiente sentencia para crear los grupos, la cual crea un nuevo campo (categoría) de valor 0 o 1:

```
vinos_sin_outliers['categoria'] = pd.cut(vinos_sin_outliers['quality'], bins=[-np.inf,5, np.inf], labels=[0,1])
```

Así quedaría agrupado el dataset:



4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Para la comprobación de la normalidad usamos el paquete pingouin disponible para Python que realiza el test Shapiro-Wilk para cada atributo y nos dice si sigue una distribución normal. En nuestro caso ningún atributo sigue una distribución normal.

```
pg.normality(vinos_sin_outliers)
```

	W	pval	normal
fixed acidity	0.946242	1.389193e-22	False
volatile acidity	0.986117	1.434102e-10	False
citric acid	0.953080	3.752339e-21	False
residual sugar	0.750200	3.050627e-42	False
chlorides	0.841707	9.856487e-36	False
free sulfur dioxide	0.925433	3.118045e-26	False
total sulfur dioxide	0.892024	1.033628e-30	False
density	0.997058	8.058984e-03	False
pH	0.997226	1.196026e-02	False
sulphates	0.951553	1.744444e-21	False
alcohol	0.932599	4.484161e-25	False
quality	0.847175	2.962722e-35	False

Para la comprobación de homeostacidad tenemos los siguientes resultados en relación con el campo categoría creado:

```
==============================
fixed acidity
                 W          pval  equal_var
levene  25.202316  5.801692e-07      False
==============================
volatile acidity
                W      pval  equal_var
levene  1.371825  0.241691       True
==============================
citric acid
                W      pval  equal_var
levene  19.12719  0.000013      False
==============================
residual sugar
                W      pval  equal_var
levene  0.002348  0.961362       True
==============================
chlorides
                W      pval  equal_var
levene  3.130191  0.077065       True
==============================
free sulfur dioxide
                W     pval  equal_var
levene  3.069608  0.07998       True
==============================
total sulfur dioxide
                 W          pval  equal_var
levene  119.840893  7.471414e-27      False
==============================
pH
                W      pval  equal_var
levene  2.853543  0.091388       True
==============================
sulphates
                 W      pval  equal_var
levene  19.029164  0.000014      False
==============================
alcohol
                 W          pval  equal_var
levene  131.527575  3.310753e-29      False
==============================
```

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Debido a que los atributos no están normalmente distribuidos, tenemos que usar pruebas no paramétricas, en este caso usaremos la prueba de Mann-Whitney U para esta prueba se requiere que existe igualdad de varianza, entonces la aplicaremos sobre atributos que cumplan esta condición.

H = No hay diferencia en las medias.
Ha = Existe una diferencia significativa

Para el primer test tenemos que el atributo 'volatile acidity' tiene influencia en que el vino tengo un buen o mal sabor. Debido a que p<0.05 existe una diferencia significativa. Esto también se aplica para el atributo 'free sulfur dioxide'. Mientras que para los demás atributos se acepta la hipotesisi nula, entonces no hay diferencia en las medias para cada categoría.

```
#Relaiza el test de Mann-Whitney U
pg.mwu(vinos_sin_outliers[vinos_sin_outliers['categoria']== 0]['volatile acidity'],
       vinos_sin_outliers[vinos_sin_outliers['categoria']== 1]['volatile acidity'])
```

	U-val	tail	p-val	RBC	CLES
MWU	352949.0	two-sided	9.381647e-31	-0.350825	0.668504

```
#Relaiza el test de Mann-Whitney U
pg.mwu(vinos_sin_outliers[vinos_sin_outliers['categoria']== 0]['residual sugar'],
       vinos_sin_outliers[vinos_sin_outliers['categoria']== 1]['residual sugar'])
```

	U-val	tail	p-val	RBC	CLES
MWU	254785.0	two-sided	0.412807	0.024873	0.484266

```
#Relaiza el test de Mann-Whitney U
pg.mwu(vinos_sin_outliers[vinos_sin_outliers['categoria']== 0]['free sulfur dioxide'],
       vinos_sin_outliers[vinos_sin_outliers['categoria']== 1]['free sulfur dioxide'])
```

	U-val	tail	p-val	RBC	CLES
MWU	278956.5	two-sided	0.026109	-0.067637	0.515137

```
#Relaiza el test de Mann-Whitney U
pg.mwu(vinos_sin_outliers[vinos_sin_outliers['categoria']== 0]['pH'],
       vinos_sin_outliers[vinos_sin_outliers['categoria']== 1]['pH'])
```

	U-val	tail	p-val	RBC	CLES
MWU	268591.5	two-sided	0.358012	-0.027968	0.503965

Para el siguiente análisis se realizó una regresión logística:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score
lr = LogisticRegression()
lr.fit(x_train, y_train)
lr_predict = lr.predict(x_test)
```

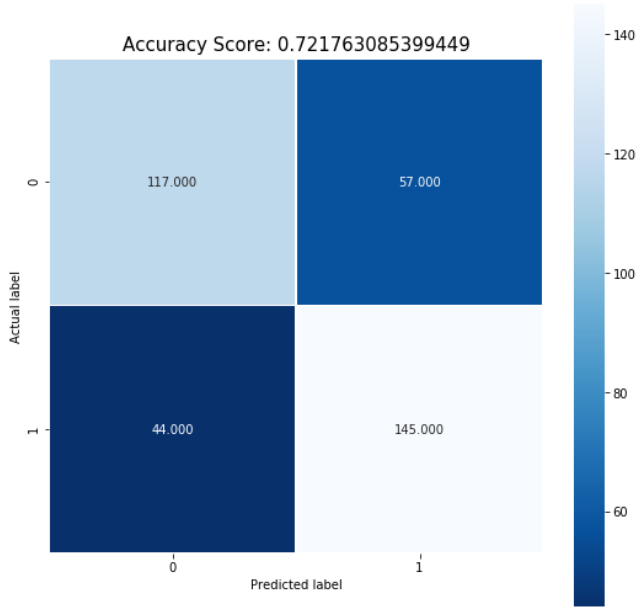
La matriz de confusión y el porcentaje de exactitud del modelo es el siguiente:

```
lr_conf_matrix = confusion_matrix(y_test, lr_predict)
lr_acc_score = accuracy_score(y_test, lr_predict)
print(lr_conf_matrix)
print(lr_acc_score*100)
```

```
[[117  57]
 [ 44 145]]
72.1763085399449
```

Podemos ver que se tiene un modelo con el 72% de exactitud.

La matriz de confusión se representa gráficamente por:

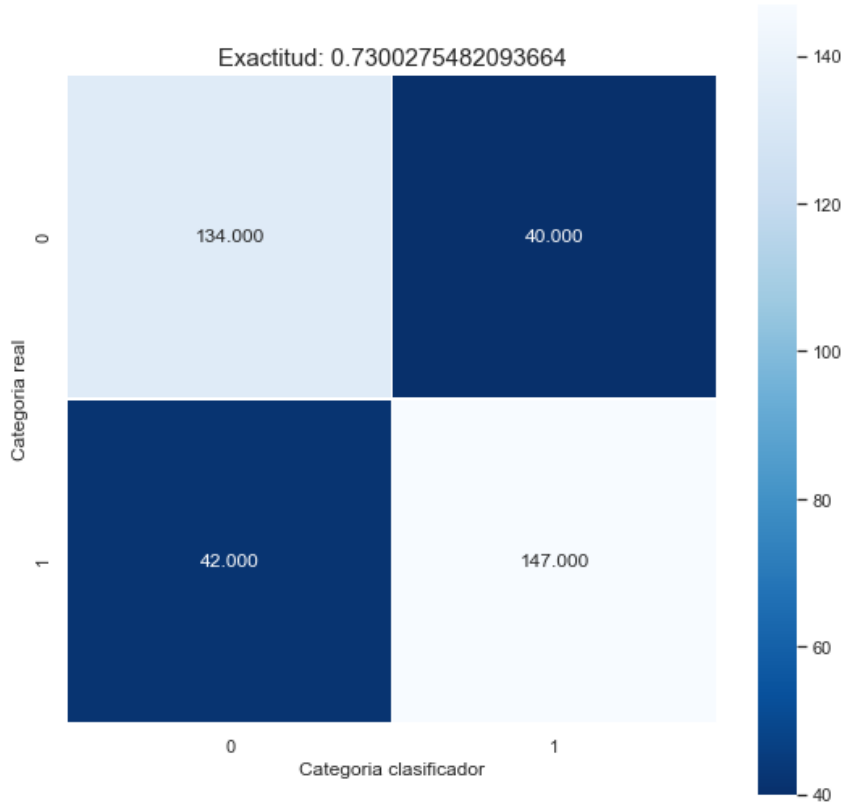


El siguiente clasificador es Random Forest:

```
#Realiza clasifica por árboles aleatorios
from sklearn.ensemble import RandomForestClassifier
RF_clf = RandomForestClassifier(n_estimators = 50)
cv_scores = cross_val_score(RF_clf,x_train, y_train, cv=10, scoring='accuracy')
RF_clf.fit(x_train, y_train)
pred_RF = RF_clf.predict(x_test)
cm = confusion_matrix(y_test, pred_RF)
```

La matriz de confusión y el porcentaje de exactitud del modelo es el siguiente, podemos ver tiene una exactitud del 73%.

```
#Grafica la matriz de confusión
plt.figure(figsize=(9,9))
sns.heatmap(cm, annot=True, fmt=".3f", linewidths=.5, square = True, cmap = 'Blues_r');
plt.ylabel('Categoria real');
plt.xlabel('Categoria clasificador');
all_sample_title = 'Exactitud: {0}'.format(lr_acc_score )
plt.title(all_sample_title, size = 15);
```



Finalmente aplicamos un proceso para obtener los atributos que más peso tienen para la predicción de la categoría, así podemos ver que los atributos mas importantes son: **'volatile acidity', 'total sulfur dioxide', 'sulphates', 'alcohol',** esto concuerda con los tests estadísticos realizados.

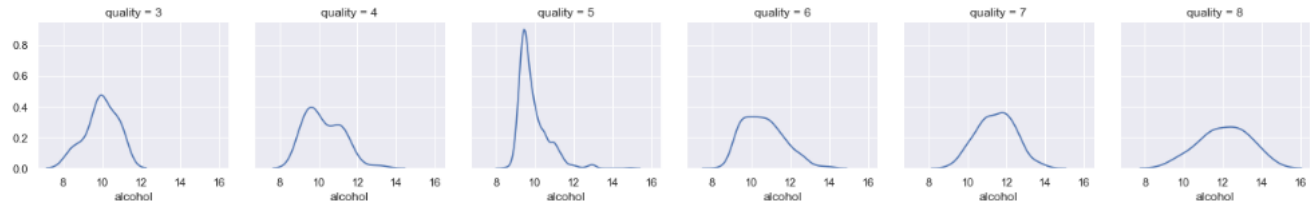
```
from sklearn.feature_selection import SelectFromModel
sel = SelectFromModel(RandomForestClassifier(n_estimators = 100))
sel.fit(x_train, y_train)
sel.get_support()
selected_feat= x_train.columns[(sel.get_support())]
len(selected_feat)
print(selected_feat)
```

```
Index(['volatile acidity', 'total sulfur dioxide', 'sulphates', 'alcohol'], dtype='object')
```

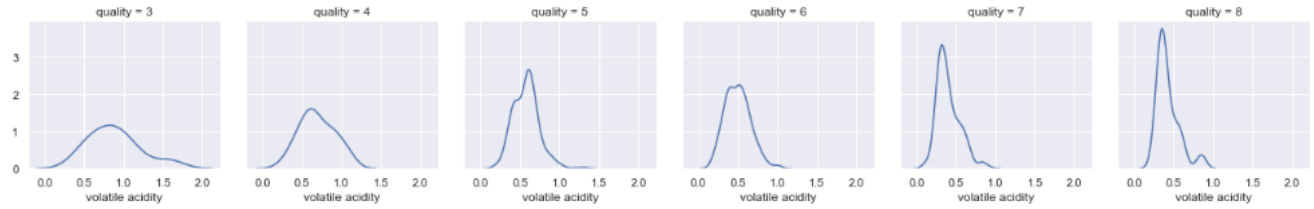
5. Representación de los resultados a partir de tablas y gráficas.

Aquí se representan los atributos que más correlación tienen con la calidad del vino, mediante un diagrama de frecuencia agrupado por las categorías de calidad.

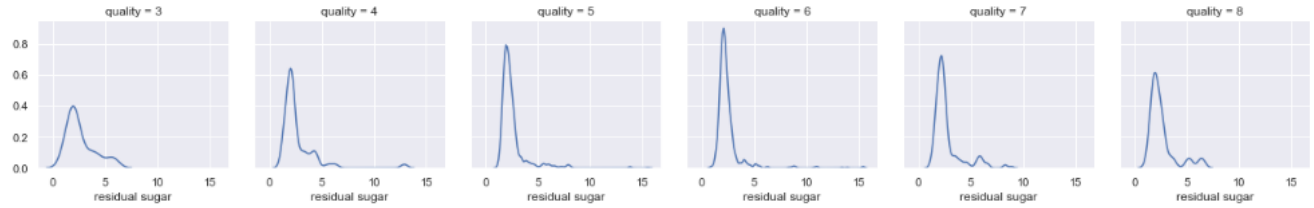
```
#Grafica la distribución de porcentaje de alcohol de acuerdo a la calidad del vino
g = sns.FacetGrid(vinos, col='quality')
g = g.map(sns.kdeplot, 'alcohol')
```




```
#Grafica la distribución de cantidad de sulfatos de acuerdo a la calidad del vino
g = sns.FacetGrid(vinos, col='quality')
g = g.map(sns.kdeplot, 'volatile acidity')
```

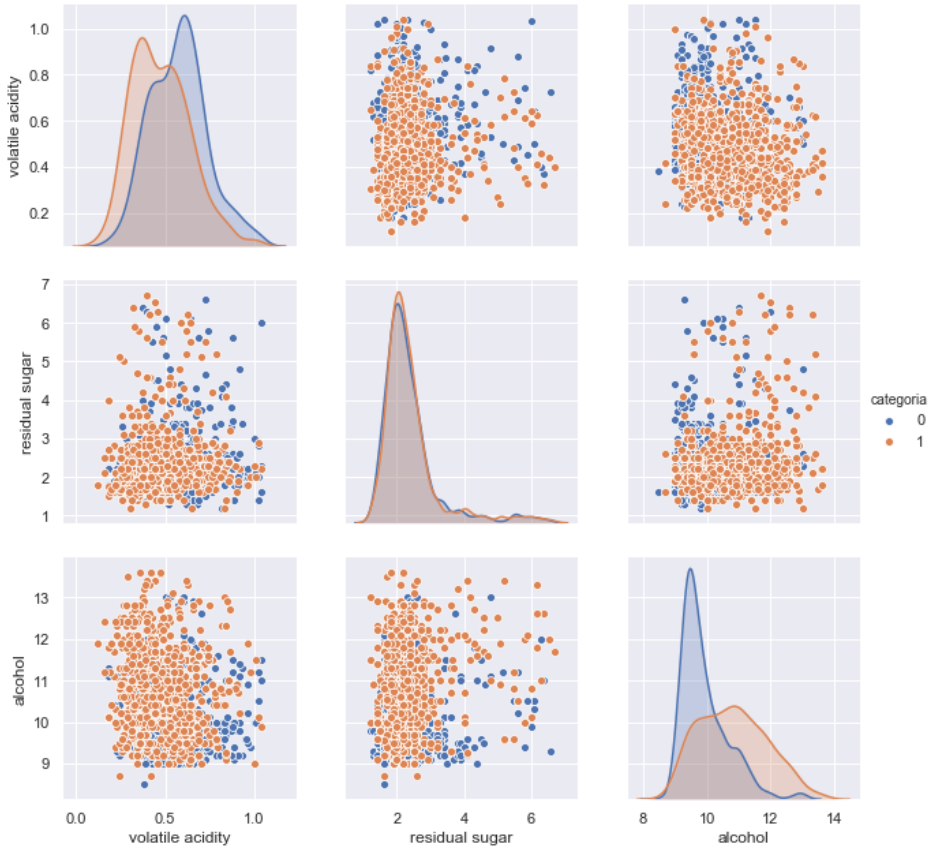



```
#Grafica la distribución de cantidad de acido cítrico de acuerdo a la calidad del vino
g = sns.FacetGrid(vinos, col='quality')
g = g.map(sns.kdeplot, 'residual sugar')
```



A continuación, una gráfica que presenta la relación entre los atributos analizados diferenciados por el color de su categoría.

```
g = sns.pairplot(vinos_sin_outliers, height=3,
                 vars=["volatile acidity", "residual sugar","alcohol"],hue = "categoria")
```



Mediante la siguiente tabla podemos tener una idea general del dataset:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Mediante los diferentes análisis realizados obtuvimos los atributos que mas influyen al momento de definir si un vino es bueno o malo ('volatile acidity', 'total sulfur dioxide', 'sulphates', 'alcohol'), estos resultados están soportados mediante test estadísticos y algoritmos de clasificación, además se pudo comprobar estos resultados en el gráfico de correlación, de la misma manera obtuvimos dos modelos: el primero de regresión logística para predecir la categoría del vino en base a sus atributos con un 73% de exactitud, el siguiente modelo fue de bosque aleatorio en el cual que predice la categoría con un 73% de exactitud, se pueden realizar otros procesos de definición de hiperparametros para lograr un mejor modelo, pero para este trabajo consideremos que eso está fuera del alcance. Podemos concluir que los resultados permiten responder el problema.

Contribuciones	**Firma**
Investigación previa	Gabriel Loja
Redacción de las respuestas	Gabriel Loja
Desarrollo código	Gabriel Loja