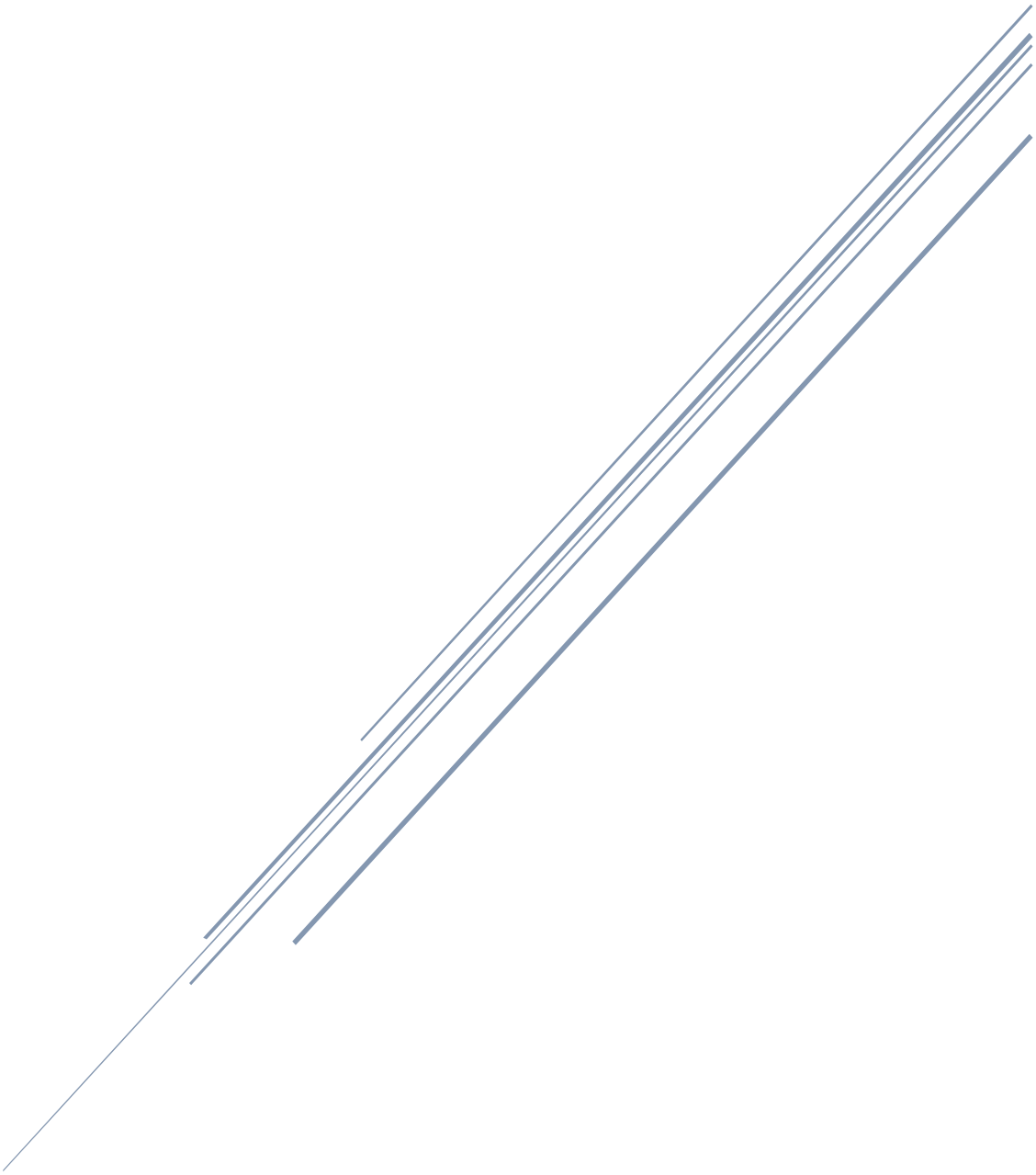
PRÁCTICA 2

Tipología y ciclo de vida de los datos



Suau Ruiz, Alejandro
Máster en Data Science - UOC

Índice

1. Descripción.

1.1. Descripción del dataset.

El conjunto de datos objeto de análisis se ha obtenido a partir de este enlace en Kaggle y está constituido por 12 características (columnas) que presentan 1599 muestras (filas o registros). Entre los campos de este conjunto de datos, encontramos los siguientes:

- **fixed acidity:** la mayoría de los ácidos involucrados con el vino.
- **volatile acidity:** cantidad de ácido acético en el vino.
- **citric acid:** cantidad de ácido cítrico en el vino.
- **residual sugar:** cantidad de azúcar disponible tras detener la fermentación.
- **chlorides:** cantidad de sal en el vino.
- **free sulfur dioxide:** cantidad libre de dióxido de sulfuro.
- **total sulfur dioxide:** cantidad total de dióxido de sulfuro.
- **density:** densidad del vino.
- **pH:** describe cuán ácido o básico es un vino en una escala de 0 (muy ácido) a 14 (muy básico).
- **sulphates:** sulfatos, aditivo para el vino.
- **alcohol:** porcentaje de alcohol contenido en el vino.
- **quality:** variable de salida (basada en datos sensoriales, puntuada del 0 al 10).

1.2. Por qué es importante y qué problema pretende resolver.

La clasificación del vino es una tarea compleja ya que el gusto es el menos entendido de los sentidos humanos. Una buena predicción de la calidad del vino puede ser muy útil en la fase de certificación, ya que, actualmente, el análisis sensorial es realizado por catadores humanos, siendo claramente un enfoque subjetivo.

Un sistema predictivo automático puede integrarse en un sistema de soporte de decisiones, ayudando a la velocidad y la calidad del desempeño del enólogo. Además, un proceso de selección de características puede ayudar a analizar el impacto de las pruebas analíticas. Si se concluye que varias variables de entrada son altamente relevantes para predecir la calidad del vino (ya que en el proceso de producción se pueden controlar algunas variables) esta información se puede utilizar para mejorar la calidad del vino.

2. Integración y selección de los datos de interés.

Todos los atributos que están presentes en el conjunto de datos se corresponden con características relativas a la calidad del vino, por lo que será conveniente tenerlos en consideración durante la realización de los análisis. Sin embargo, cabe la posibilidad de que, en pasos posteriores, mediante algoritmos como PCA (Principal Component Analysis), se descarten componentes *menos relevantes*.

3. Limpieza de datos.

3.1. Missing values (valores nulos, cero o vacíos).

Es habitual utilizar valores vacío, NA o NULL como centinela para indicar la ausencia de ciertos valores. Sin embargo, en el caso de este conjunto, podemos observar como no disponemos de *missing values*.

```
df = pd.read_csv("winequality-red.csv", sep=';')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
fixed acidity           1599 non-null float64
volatile acidity        1599 non-null float64
citric acid             1599 non-null float64
residual sugar          1599 non-null float64
chlorides               1599 non-null float64
free sulfur dioxide     1599 non-null float64
total sulfur dioxide    1599 non-null float64
density                 1599 non-null float64
pH                      1599 non-null float64
sulphates               1599 non-null float64
alcohol                 1599 non-null float64
quality                 1599 non-null int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

```
df.isnull().sum()
```

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

A pesar de ello, si se dispusiese de algún *missing value* podríamos decidir varias maneras sobre cómo tratar con ellos. Una opción, por ejemplo, podría ser eliminar los registros que incluyesen algún valor de este tipo, aunque ello supondría desaprovechar parte de la información. Otra manera, podría ser la utilización de la imputación basada en k vecinos más próximos (*kNN-imputation*), la cual supondría la imputación de valores basado en la similitud o diferencia entre los registros.

3.2. Identificación y tratamiento de valores extremos.

Los valores extremos (*outliers*) son aquellos que parecen no ser congruentes si los comparamos con el resto de los datos. Para identificarlo, podemos llevar a cabo las siguientes acciones:

- Representar un diagrama de caja para cada una de las variables y ver qué valores distan ampliamente del rango intercuartílico.
- Utilizar el método IQR (*interquartile range*), que será el empleado en nuestro caso.

```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1

IQR_df = (df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))
print(df[IQR_df == True])
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	\
0	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	NaN	NaN	NaN	
2	NaN	NaN	NaN	NaN	NaN	
3	NaN	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	NaN	
5	NaN	NaN	NaN	NaN	NaN	
6	NaN	NaN	NaN	NaN	NaN	
7	NaN	NaN	NaN	NaN	NaN	
8	NaN	NaN	NaN	NaN	NaN	
9	NaN	NaN	NaN	6.1	NaN	
10	NaN	NaN	NaN	NaN	NaN	
11	NaN	NaN	NaN	6.1	NaN	
12	NaN	NaN	NaN	NaN	NaN	
13	NaN	NaN	NaN	NaN	NaN	
14	NaN	NaN	NaN	3.8	0.176	
15	NaN	NaN	NaN	3.9	0.170	
16	NaN	NaN	NaN	NaN	NaN	
17	NaN	NaN	NaN	NaN	0.368	
18	NaN	NaN	NaN	4.4	NaN	
19	NaN	NaN	NaN	NaN	0.341	
20	NaN	NaN	NaN	NaN	NaN	
21	NaN	NaN	NaN	NaN	NaN	
22	NaN	NaN	NaN	NaN	NaN	
23	NaN	NaN	NaN	NaN	NaN	
24	NaN	NaN	NaN	NaN	NaN	
25	NaN	NaN	NaN	NaN	NaN	
26	NaN	NaN	NaN	NaN	NaN	
27	NaN	NaN	NaN	NaN	NaN	
28	NaN	NaN	NaN	NaN	NaN	
29	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	

No obstante, si revisamos los datos anteriores, puede darse el caso de que algunos vinos (considerados muy dulces) contengan un valor de 6.1. en el azúcar residual o, 0.170 en

la cantidad de sal. Es por ello por lo que el manejo de estos valores extremos consistirá simplemente en dejarlos tal cual están.

4. Análisis de los datos.

4.1. Selección de los grupos de datos que se desean analizar/comparar.

A continuación, se seleccionan los grupos dentro de nuestro conjunto de datos que pueden resultar interesantes para analizar y/o comparar. Estos grupos distintos son las agrupaciones por calidad de vino: Malos, normales y excelentes.

```
def add_quality_tag(row):
    if row['quality'] >= 1 and row['quality'] <= 3:
        return 'bad'
    elif row['quality'] >= 4 and row['quality'] <= 7:
        return 'average'
    elif row['quality'] >= 8 and row['quality'] <= 10:
        return 'excellent'

bad_wines = df[(df.quality >= 1) & (df.quality <= 3)]
average_wines = df[(df.quality >= 4) & (df.quality <= 7)]
excellent_wines = df[(df.quality >= 8) & (df.quality <= 10)]

df['tag'] = df.apply(add_quality_tag, axis=1)
```

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

Utilizaremos la prueba de normalidad de *Kolmogorov-Smirnov* con la finalidad de comprobar que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente.

Así, se comprueba que para cada prueba se obtiene un p-valor superior al nivel de significación prefijado $\alpha = 0{,}05$. Si esto se cumple, entonces se considera que la variable en cuestión sigue una distribución normal.

```
alpha = 0.05
print("Variables que no siguen una distribución normal:")
for i, col in enumerate(df.columns):
    result = shapiro(df[col])
    if result[1] < alpha:
        print("-", col)
```

```
Variables que no siguen una distribución normal:
- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol
- quality
```

Ahora, continuamos para dar paso al estudio de la homogeneridad de varianzas haciendo uso del *test de Fligner-Killen*. En este caso, estudiaremos esta homogeneidad en los distintos tres grupos seleccionados anteriormente (malos, normales y excelentes) en relación con la cantidad de alcohol que contienen.

```
badw_fligner = fligner(bad_wines.quality, bad_wines.alcohol)
averagew_fligner = fligner(average_wines.quality, average_wines.alcohol)
excellentw_fligner = fligner(excellent_wines.quality, excellent_wines.alcohol)

print(badw_fligner.pvalue > alpha)
print(averagew_fligner.pvalue > alpha)
print(excellentw_fligner.pvalue > alpha)
```

```
False
False
False
```

Puesto que todas ellas presentan un p-valor inferior a 0.05, rechazamos la hipótesis de que las varianzas en los distintos grupos relacionadas con la característica alcohol son homogéneas.

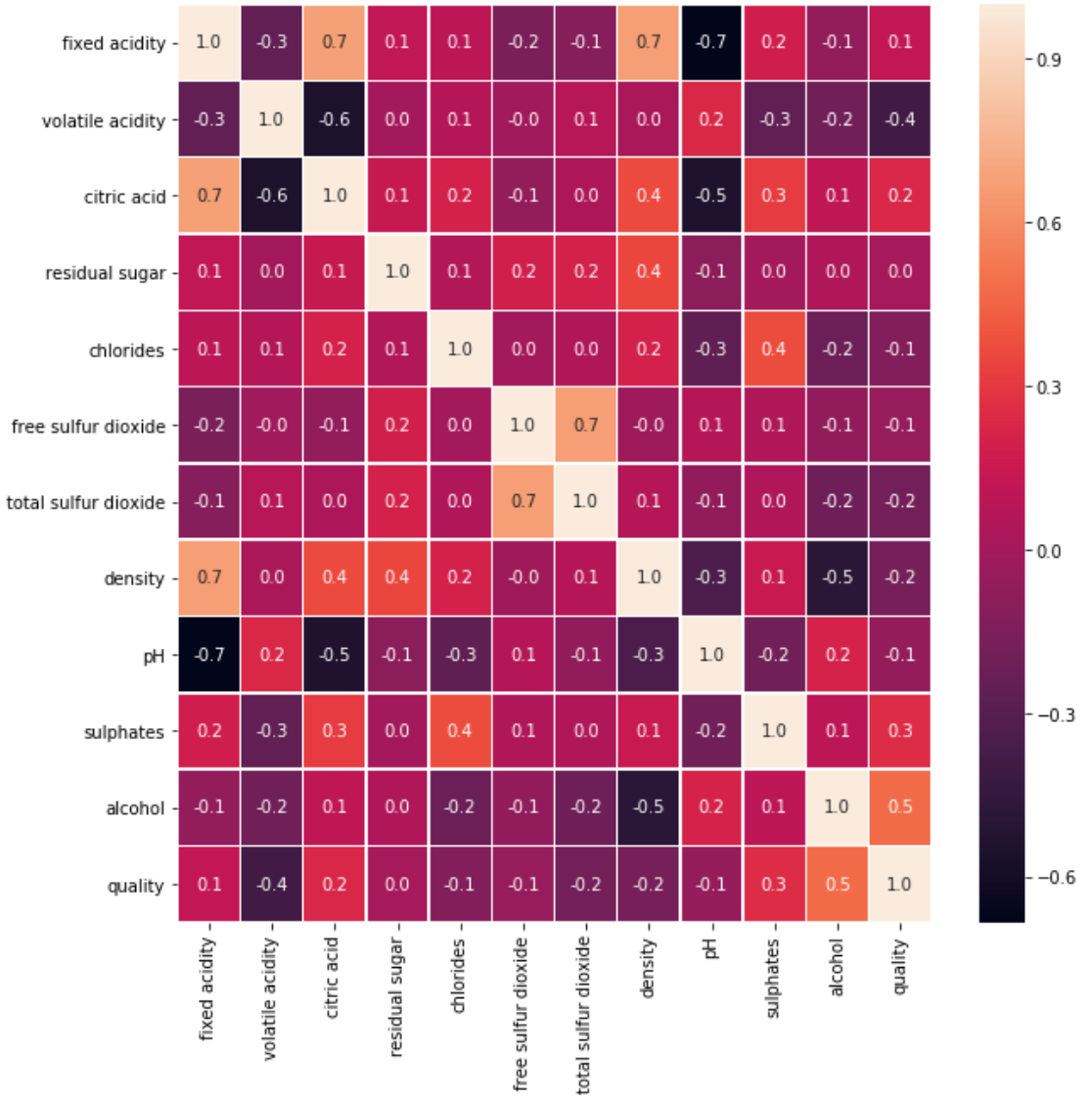
4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos.

4.3.1. Análisis de correlación.

Primeramente, efectuaremos un análisis de correlación entre las distintas variables, para determinar cuáles de ellas ejercen una mayor influencia sobre la calidad del vino. Para

ello, se utilizará el coeficiente de correlación de Spearman, puesto que nuestros datos no siguen una distribución normal.

```
f,ax=plt.subplots(figsize=(10,10))
sns.heatmap(df.corr(),annot=True,linewidth=.5,fmt='.1f',ax=ax)
plt.show()
```



De esta manera, identificamos que los valores más correlacionados con la variable *quality*, con aquellos en los que su valor es más próximo a los valores -1 y +1, tales como *alcohol*, que presenta un 0.5 o *volatile acidity*, con un 0.4.

4.3.2. Contraste de hipótesis.

La segunda prueba estadística que llevaremos a cabo consistirá en un contraste de hipótesis sobre dos muestras para determinar si la cantidad de alcohol es superior en

función de si el vino es media o excelente (obviaremos los de mala calidad por su escasa cantidad de muestras).

En nuestro caso, utilizaremos el test no paramétrico de Mann-Whitney, que resultará más eficiente en nuestro caso.

```
alpha = 0.05
manwhitney_result = mannwhitneyu(average_wines['alcohol'], excellent_wines['alcohol'])
print(manwhitney_result.pvalue < alpha)
```

```
True
```

Debido a que obtenemos un p-valor menor que el valor de significación fijado, rechazamos la hipótesis nula. Por tanto, se puede concluir que, la cantidad de alcohol es superior en el vino si este es de calidad mayor.

4.3.3. Modelo de regresión lineal.

Finalmente, tal y como se planteó en el comienzo de la actividad, resultará realmente interesante poder efectuar predicciones sobre la calidad del vino dadas sus características, con la finalidad de poder mejor dicha calidad según la modificación de dichas características. De esta manera, utilizaremos un modelo de regresión lineal, puesto que no existen variables categóricas y cada una de las variables es numérica.

Definimos una función que nos permitirá obtener aquellas características las cuales su correlación está por encima de nuestro *threshold value* (que lo pasaremos por parámetro).

```
def get_features(correlation_threshold):
    abs_corrs = correlations.abs()
    high_correlations = abs_corrs[abs_corrs > correlation_threshold].index.values.tolist()
    return high_correlations
```

```
features = get_features(0.05)
x = df[features]
y = df['quality']
```

Creamos el conjunto de entrenamiento y el conjunto de test con un 75% y 25% de los registros, respectivamente.

```
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=3)
```

Una vez que se crea el conjunto de datos, es hora de construir su modelo de regresión lineal. Simplemente se puede usar la función incorporada para crear un modelo y luego

ajustarlo a los datos de entrenamiento. Una vez entrenado, coef_ da los valores de los coeficientes para cada característica.

```
regressor = LinearRegression()
regressor.fit(x_train,y_train)

print(regressor.coef_)
```

```
[ 0.01773723 -0.99256049 -0.13962865 -1.59094279  0.00559652 -0.00351973
  0.76859036 -0.43741414  0.81288805  0.30148385]
```

Ahora, predecimos los valores y evaluamos la calidad de la predicción:

```
test_pred = regressor.predict(x_test)
print(test_pred)
```

```
[5.10801475 5.65933623 5.90407267 6.13461179 5.00611866 5.44514691
 5.05735245 6.15497513 5.51919603 5.77259374 5.61809366 5.23616173
 5.23544213 5.31968644 6.47007277 5.043404   5.85287121 5.19427909
 6.07727089 6.34949018 6.42525555 5.51221957 5.8030796  4.93637817
 5.16618356 5.48255293 5.13758624 6.60000969 5.88754763 5.74133915
 6.09716961 6.29379754 4.91269821 5.88611904 5.11007273 5.94574773
 6.80685536 5.04305653 5.25438683 5.88611904 5.17406542 4.84008442
 6.48781656 5.40521715 5.31105571 5.84484462 5.7100681  5.24300809
 5.25021217 5.46398911 5.08740494 5.61369555 6.01375792 6.32497377
 5.47511954 5.36466869 5.09234555 4.92625623 5.21415941 5.08274744
 4.79570013 5.4377645  5.25237771 5.68830391 5.85145609 6.52420079
 5.38691412 5.71775637 5.17641417 5.99156845 5.6445189  5.60892012
 5.74967567 5.21702288 5.97975854 5.51115845 5.41121547 5.6832459
 5.63971524 5.74133915 6.24163428 5.27915822 4.66596769 6.04951743
 5.52401618 5.17823915 5.20672986 5.96322663 5.50411353 5.64866275
 5.70105618 5.6431575  5.72586828 5.3173125  5.37075392 5.394889
 4.82061159 5.46006525 5.47363879 6.54074801 6.13723937 5.61422461
 6.07821503 6.17461539 5.73230665 4.92692198 4.73317591 5.03851027
 5.44868797 5.78432759 6.46608259 5.47530673 6.46876056 5.94466642
 5.43257493 5.20523855 5.34551741 5.20749557 6.19344578 5.61453943
 5.83308923 5.20267759 5.17702922 5.26912156 5.74382704 5.6431575
 6.15450941 5.89677877 5.49186029 5.39047629 5.25848318 5.41150099
 5.70750135 5.68001376 6.58288921 5.89497164 6.37172338 5.72945992
 5.37936908 5.14371952 5.58851063 6.59661777 5.24403336 5.25594627
 5.54721935 5.17243958 5.76990082 6.10847777 6.93985005 4.99562031
 5.01958735 4.68547026 5.82434616 5.01708671 5.21702288 5.70819215
 5.63334181 5.33481542 5.22220103 5.84327644 5.61823023 5.78078643
 5.51830906 6.03898671 5.63808482 5.49193476 5.96787582 4.82363567
 5.26331016 5.6625652  5.73510278 6.59570394 5.02584187 5.9062506
 5.85381667 5.21140744 5.68951564 5.51649995 5.40521715 6.37974889
 6.71642336 4.98858413 5.88413601 5.75553621 5.75508093 5.61677128
 5.71318209 5.40944809 6.05634078 5.58276397 5.88366084 6.51928901
 5.00668628 5.4022397  5.18103224 5.17641417 5.47511954 5.7498414
 5.69035164 4.92770738 5.12908401 4.98756458 6.18242395 5.65384546
 5.45261211 5.56461849 4.99367423 5.8451887  5.31537513 5.48096741
```

5. Representación de los resultados.

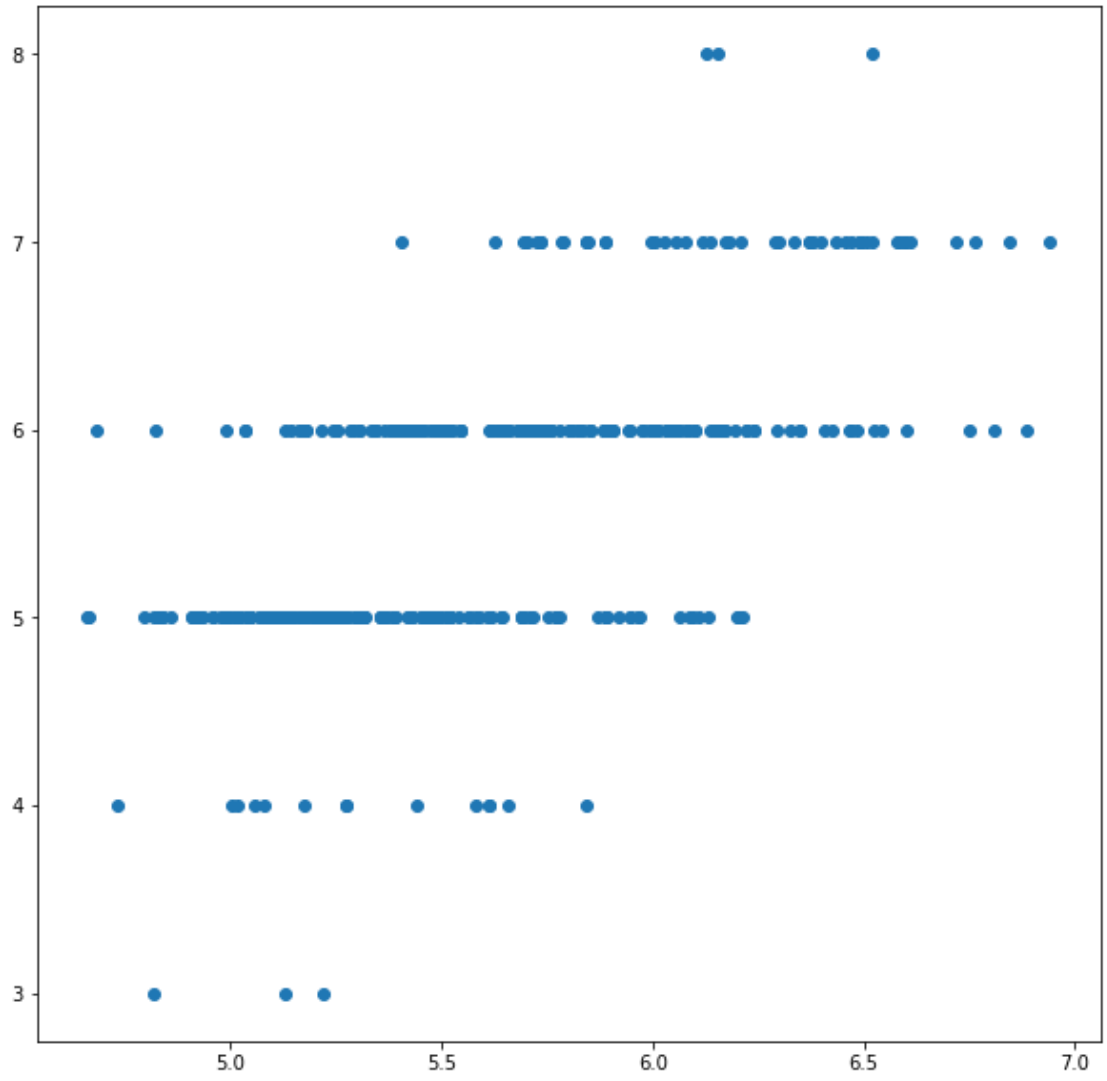
Evaluamos la calidad del modelo mediante la utilización del MSE (mean squared error).

```
rmse = sqrt(mean_squared_error(y_test, test_pred))
rmse
```

```
0.6275381539230979
```

Evaluamos cómo están repartidos los de las predicciones resultantes.

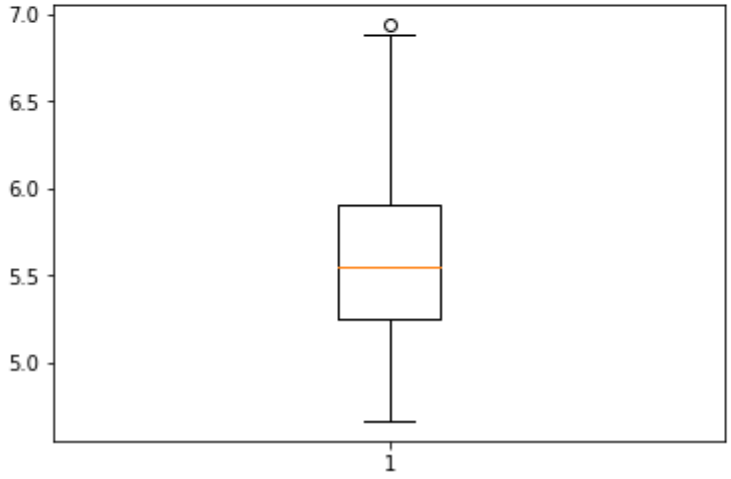
```
plt.figure(figsize=(10,10))
plt.scatter(test_pred, y_test)
plt.show()
```



Utilizamos también un boxplot:

```
plt.boxplot(test_pred)
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x22f26acc240>,
  <matplotlib.lines.Line2D at 0x22f2794fc88>],
 'caps': [<matplotlib.lines.Line2D at 0x22f2794ffd0>,
  <matplotlib.lines.Line2D at 0x22f2795a358>],
 'boxes': [<matplotlib.lines.Line2D at 0x22f2794f550>],
 'medians': [<matplotlib.lines.Line2D at 0x22f2795a6a0>],
 'fliers': [<matplotlib.lines.Line2D at 0x22f2795a9e8>],
 'means': []}
```



6. Resolución del problema, conclusiones y respuestas.

Como hemos podido apreciar, se han realizado tres tipos de pruebas estadísticas sobre un conjunto de datos que se correspondía con variables relativas a la calidad de los vinos. De esta manera, hemos podido conocer cuáles de las variables del dataset ejercían mayor influencias sobre la calidad final del vino, mediante el análisis de la correlación y el contraste de hipótesis. Asimismo, el modelo de regresión lineal creado nos ha facilitado la obtención de predicciones en función de las características proporcionadas por los registros. No obstante, cabe destacar que el porcentaje que se ha obtenido de precisión ha resultado bastante bajo, tan solo de un 62,75% (el cual podría haber sido aumentado con la utilización, por ejemplo, del algoritmo de Random Forest, entre otros).

7. Código.

El código está disponible en el fichero PRAC2, adjunto en la URL de github del proyecto. Se ha llevado a cabo utilizando el lenguaje Python y la herramienta Jupyter Notebooks, para facilitar la claridad.