PRAC2 TIPOLOGIA I CICLE DE VIDA DE LES DADES

Oriol Della

1. Descripció del dataset. Perquè és important i quina pregunta/problema pretén respondre? (0,5p)

La base de dades escollida per a la realització de la activitat és Red Wine Quality, la qual prové del repositori de la UCI Machine Learning.

Enllaç

El dataset conté els següents punts d'informació:

Dades descriptives:

1 - fixed acidity
2 - volatile acidity
3 - citric acid
4 - residual sugar
5 - chlorides
6 - free sulfur dioxide
7 - total sulfur dioxide
8 - density
9 - pH
10 - sulphates
11 - alcohol

Etiqueta de classe:

12 - quality (score between 0 and 10)

A la descripció del dataset també podem veure que no falta el valor en cap registre de cap atribut.

Enllaç a la descripció del dataset

Es tracta d'un dels datasets més populars del repositori de la UCI amb un total de 1.538.217 visites. Conté dades que descriuen vins vermells i blancs provinent del nord de Portugal. Els vins estan catalogats segons la seva qualitat.

Concretament el dataset conté els anàlisis químics dels vins junt amb la seva qualitat. Aquest s'ha utilitzat per a modelar sistemes de Machine Learning que puguin catalogar la qualitat del vi atenent a la seva composició química.

2. Integració i selecció de les dades d'interès a analitzar. (0,5p)

Entenem per integració de les dades com al procés de combinació de les dades que provenen de diferents orígens, amb la intenció de crear el dataset que finalment s'utilitzarà per a dur a terme els anàlisis.

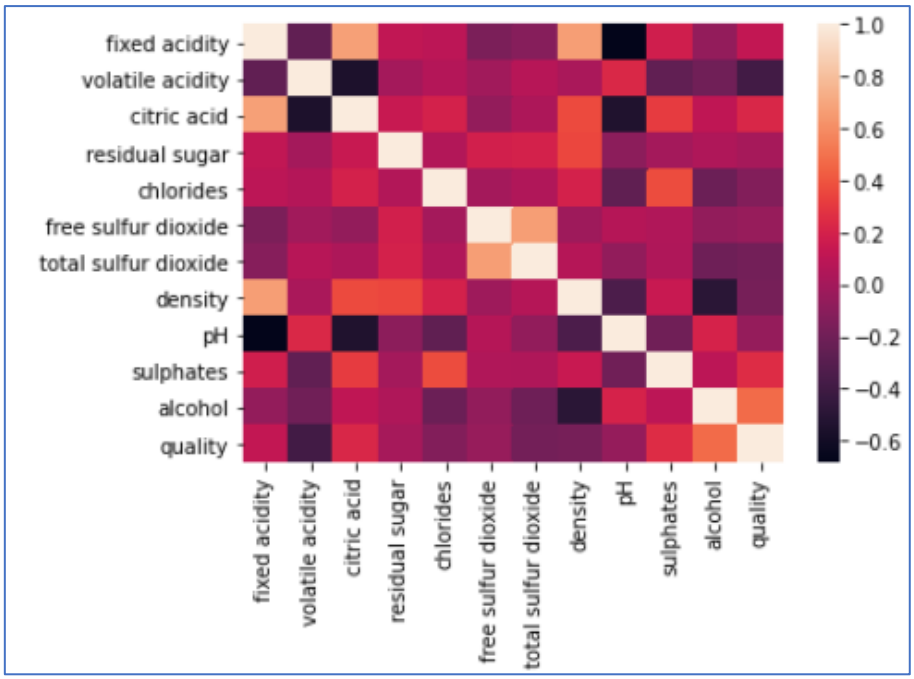
En aquest cas però les dades ja venen ordenades en un sol document en format .csv, de manera que per a disposar de les dades necessàries per a la realització de l'examen només haurem d'importar-les:

```
# Importem les dades

df = pd.read_csv("winequality-red.csv")
print(df.head())
```

```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
```

Per a dur a terme la selecció de les dades a analitzar computarem la correlació entre variables i les mostrarem tant numèricament com en un heatmap:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-0.682978	0.183006	-0.061668	0.124052
volatile acidity	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	0.234937	-0.260987	-0.202288	-0.390558
citric acid	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-0.541904	0.312770	0.109903	0.226373
residual sugar	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.355283	-0.085652	0.005527	0.042075	0.013732
chlorides	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.200632	-0.265026	0.371260	-0.221141	-0.128907
free sulfur dioxide	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.021946	0.070377	0.051658	-0.069408	-0.050656
total sulfur dioxide	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.071269	-0.066495	0.042947	-0.205654	-0.185100
density	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.000000	-0.341699	0.148506	-0.496180	-0.174919
pH	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.341699	1.000000	-0.196648	0.205633	-0.057731
sulphates	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.148506	-0.196648	1.000000	0.093595	0.251397
alcohol	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.496180	0.205633	0.093595	1.000000	0.476166
quality	0.124052	-0.390558	0.226373	0.013732	-0.128907	-0.050656	-0.185100	-0.174919	-0.057731	0.251397	0.476166	1.000000



Si trobéssim correlacions superiors a +-0.8 o +-0.9, hauríem de considerar eliminar alguna de les variables per a eliminar informació redundant i permetre al model centrar-se en aquelles característiques que incorporen més informació.

Com podem veure però, les correlacions més elevades que troben no arriben a +- 0.7, de manera que podem considerar que totes les columnes de les que disposem aporten informació que pot ser rellevant per a l'anàlisi.

3. Neteja de les dades. (2p)

3.1. Les dades contenen zeros o elements buits? Com gestionaries aquests casos?

Per a comprovar, que tal i com la descripció de les dades ens havia anticipat, que no existeixen dades nul·les utilitzarem la funció corresponent de la llibreria pandas:

```
In [7]:  1  df.isnull().values.any()
Out[7]: False
```

Per el retorn de la funció ja sabem que no existeixen dades nul·les, procedim a analitzar si existeixen dades amb el valor 0:

```
In [8]:  1  df.eq(0).any().any()

Out[8]:  True
```

De manera que ja sabem que existeixen valors 0 en les dades.

D'existir valors NaN a les dades, hauríem de prendre alguna mesura correctora, ja que no al passar-les al model ens donaria problemes. Si es tractessin de poques dades, i tinguéssim accés a les dades originals, podríem investigar per a veure si podem cercar-les per a rectificar els espais en buit.

Una altre tècnica seria la de descartar les files que continguessin dades nul·les, de manera que en el dataset nomes hi quedessin les dades complertes.

Existeixen altres mètodes com per exemple omplir els espais buits amb el valor mitjà, medià o moda, tot intentant reduir al mínim l'impacte de les dades perdudes en els nostres anàlisis.

Els mètodes més avançats implicarien aplicar sistemes de imputació de valors mitjançant un model de regressió o algun altre model predictiu.

En tot cas, qualsevol sistema que substitueixi les dades reals per valors predits correrà el perill de reduir la precisió i versemblança del model. Un bon enfocament per al cas que ens ocupa, així com ràpid i efectiu, seria la simple eliminació de les files que continguessin valors nuls.

Sobre l'existència de valors 0 no caldrà aplicar cap corrector. En el dataset que ens ocupa, l'existència de valors 0 no implica l'existència d'errors, ja que perfectament un vi pot no contenir un element específic. Si caldria analitzar amb especial interès el valor 0 si veiem que es tracta d'un outlier, ja que potser ens estaria indicant la presencia d'un error en l'anàlisi o bolcat de dades.

3.2. Identificació i tractament de valors extrems.

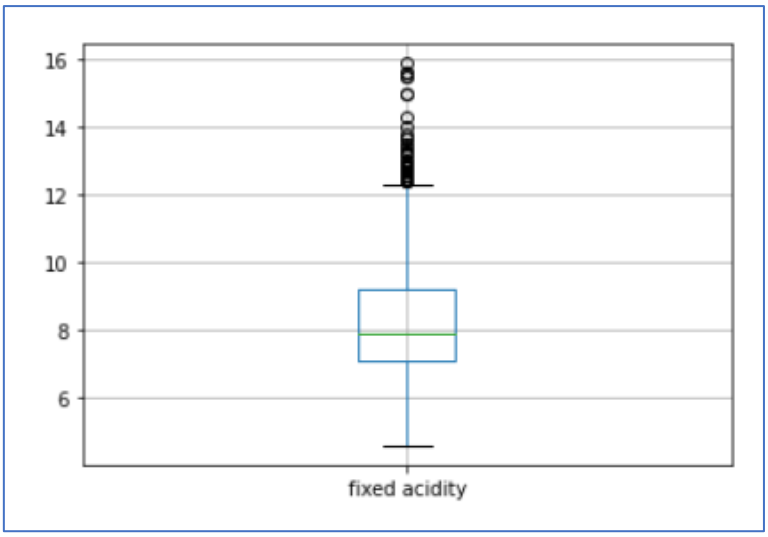
Farem un primer anàlisi de les distribucions de les dades mitjançant la funció describe de pandas.

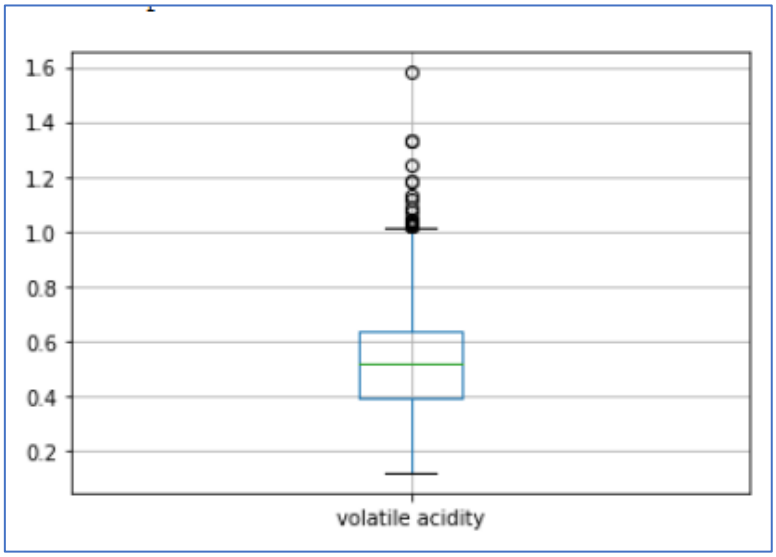
```
1  df.describe()
```

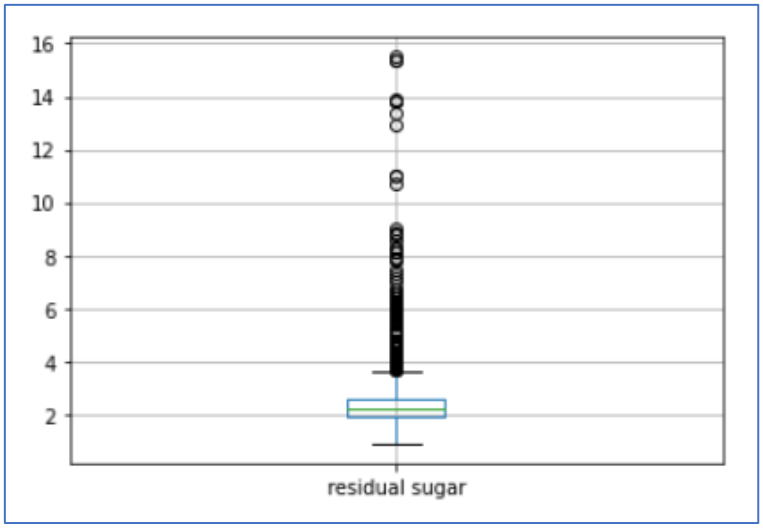
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	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

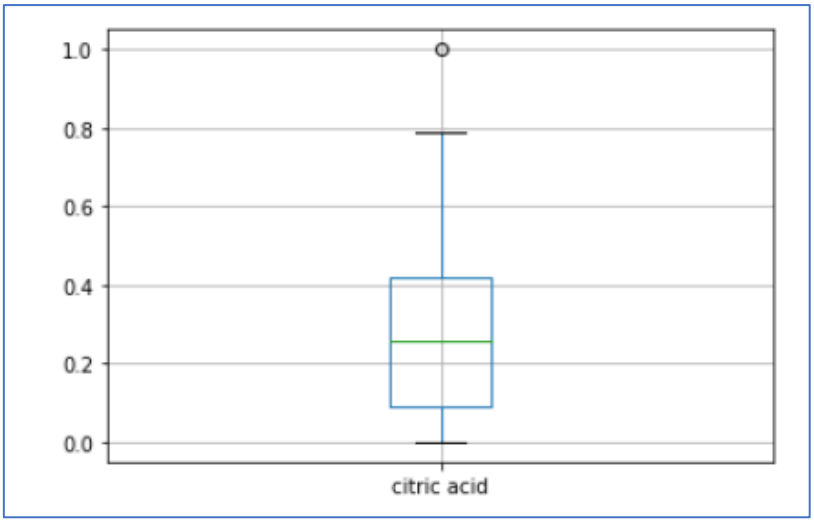
Amb aquest anàlisi inicial de les dades ja podem observar les variables que contenen outliers. Els més notoris els podrien trobar a residual sugar, on amb una mitja de 2.5 i una std de 1.4, trobem un vi amb un calor de 15. Això també succeeix amb les columnes de *sulfur dioxide, chlorides,* d'entre altres.

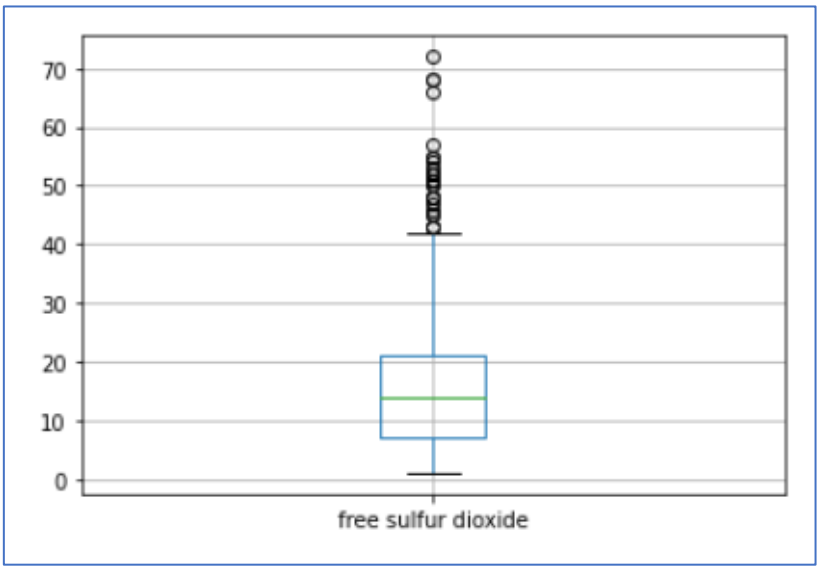
Procedim a visualitzar les distribucions de les columnes amb outliers:

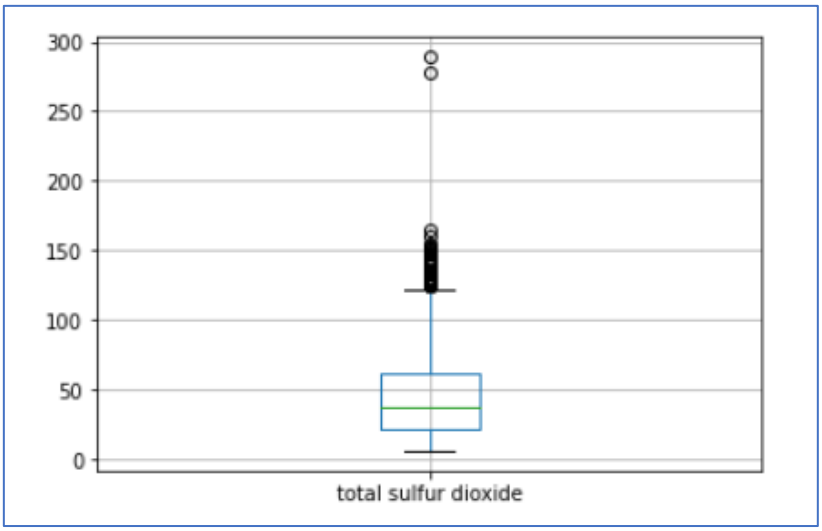


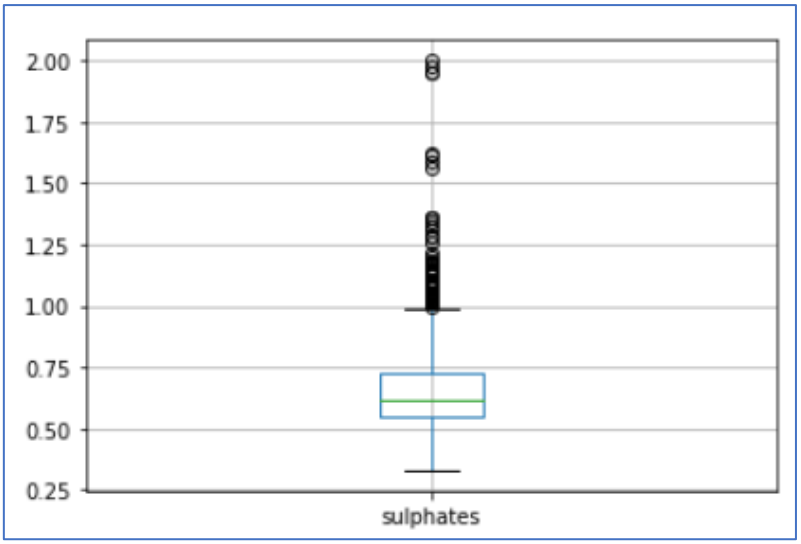





1.0
0.8
0.6
0.4
0.2
0.0
citric acid


70
60
50
40
30
20
10
0
free sulfur dioxide


300
250
200
150
100
50
0
total sulfur dioxide



Dels gràfics realitzats podem veure com les distribucions contenen quantitat de valors extrems. Vista com és la distribució, simplement sembla que els outliers formen part de la distribució natural de les dades, donada la seva quantitat, i no sembla indicar ningun error en les dades.

Si que existeix un cas concret on només existeix un outlier, i és en el cas de l'àcid cítric. Possiblement l'excés d'aquest valor ens indica de que es tracta d'un vi de baixa qualitat. Procedim a mirar l'etiqueta de classe per a veure si és el cas.

```
row = df.loc[df['citric acid'] == 1]
row
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
151	9.2	0.52	1.0	3.4	0.61	32.0	69.0	0.9996	2.74	2.0	9.4	4

La resta de valors semblen normals i la nota qualitativa és 4, essent el mínim un 3. Arribats aquí crec que no tenim arguments suficients per a considerar-lo un error, de manera que el mantindrem a les dades.

4. Anàlisi de les dades.

4.1. Selecció dels grups de dades que es volen analitzar/comparar (planificació dels anàlisis a aplicar).

L'anàlisi de les dades es realitzarà sobre tot el conjunt, per a maximitzar-ne l'extracció d'informació.

Posteriorment separarem les dades en trainset i testset per a entrenar diferents models i poder testejar la seva bondat.

Finalment realitzarem un model de regressió lineal, un arbre de decisió i un random forest, tot amb la intenció de poder preveure la qualitat del vi vista la seva composició química.

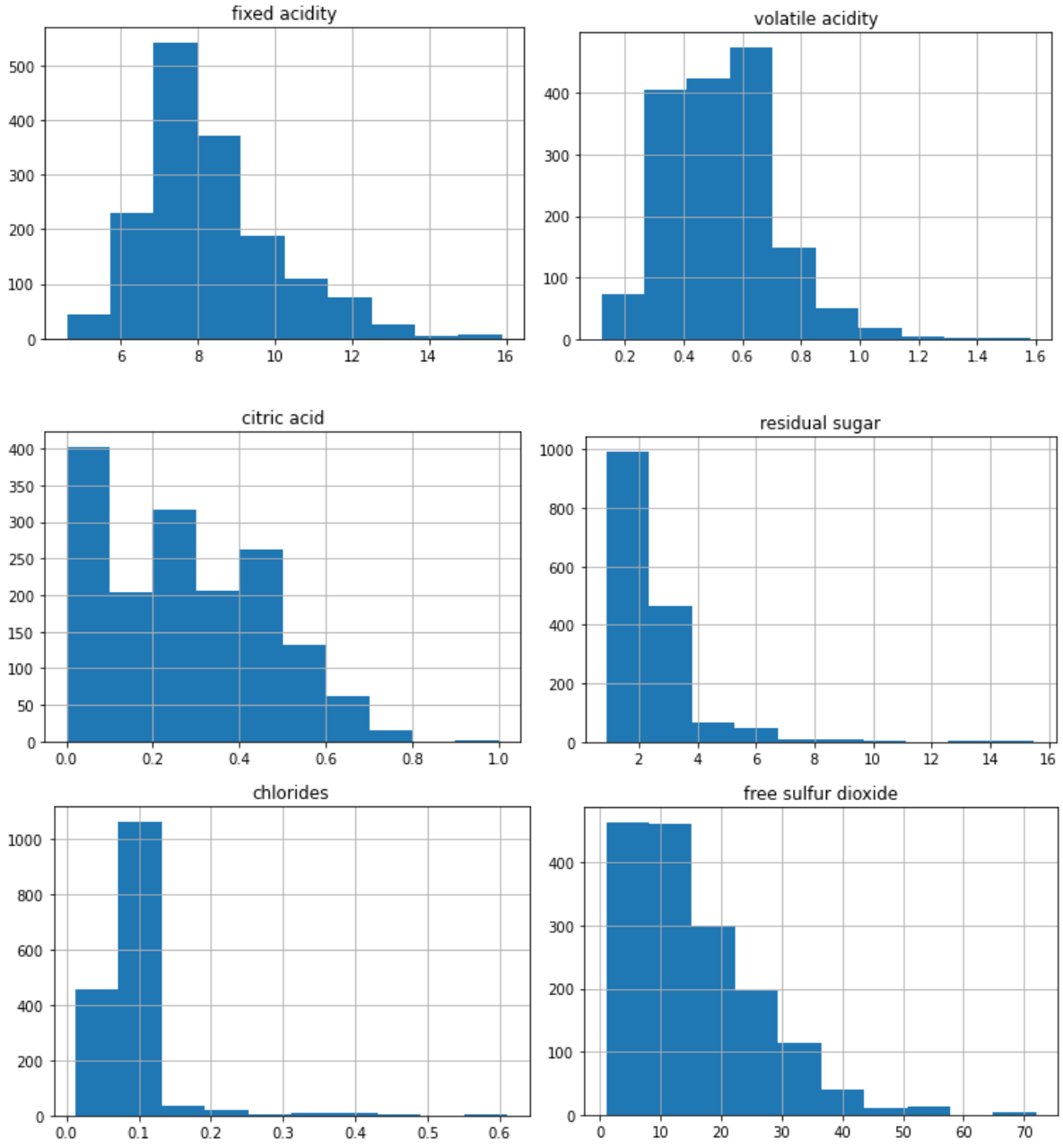
4.2. Comprovació de la normalitat i homogeneïtat de la variància.

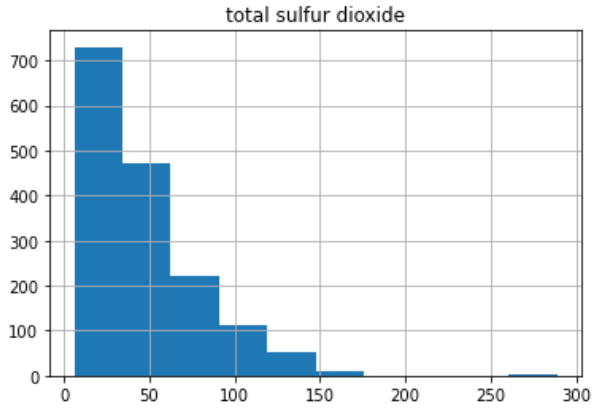
Utilitzarem el mètode visual i el mètode Shapiro-Wilk per a observar la normalitat:

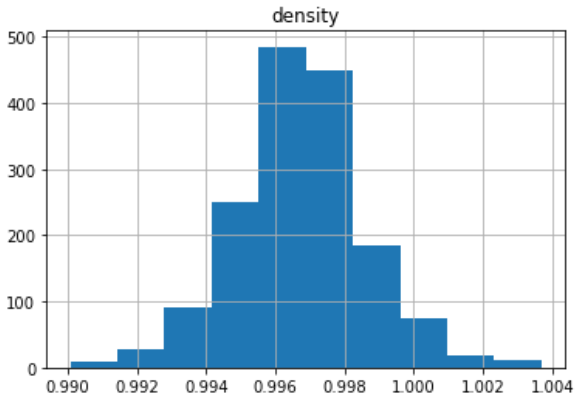
```
for num in df.columns:
    print(num)
    stat, p = shapiro(df[num])
    print('Statistics=%.3f, p=%.10f' % (stat, p))
    alpha = 0.05
    if p > alpha:
        print('Gaussià (no neguem H0)\n')
    else:
        print('no Gaussià (neguem H0)\n')
```

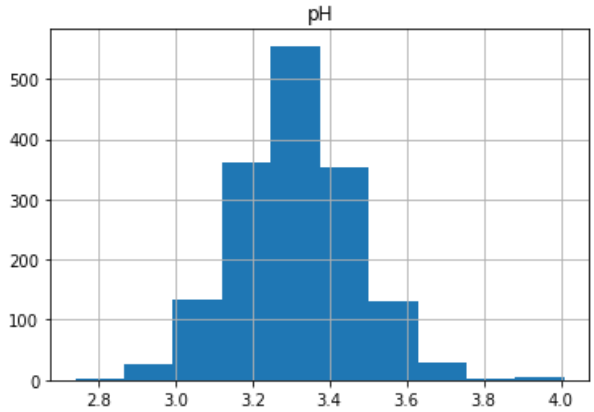
```
fixed acidity
Statistics=0.942, p=0.0000000000
no Gaussià (neguem H0)

volatile acidity
Statistics=0.974, p=0.0000000000
no Gaussià (neguem H0)

citric acid
Statistics=0.955, p=0.0000000000
no Gaussià (neguem H0)

residual sugar
Statistics=0.566, p=0.0000000000
no Gaussià (neguem H0)

chlorides
Statistics=0.484, p=0.0000000000
no Gaussià (neguem H0)

free sulfur dioxide
Statistics=0.902, p=0.0000000000
no Gaussià (neguem H0)

total sulfur dioxide
Statistics=0.873, p=0.0000000000
no Gaussià (neguem H0)

density
Statistics=0.991, p=0.0000000194
no Gaussià (neguem H0)

pH
Statistics=0.993, p=0.0000017137
no Gaussià (neguem H0)

sulphates
Statistics=0.833, p=0.0000000000
no Gaussià (neguem H0)

alcohol
Statistics=0.929, p=0.0000000000
no Gaussià (neguem H0)
```

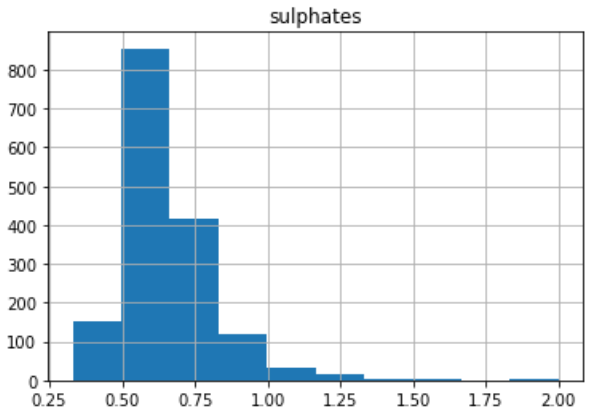
```
quality
Statistics=0.858, p=0.0000000000
no Gaussià (neguem H0)
```

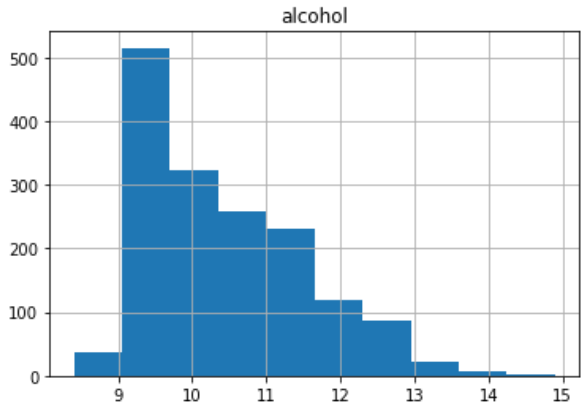


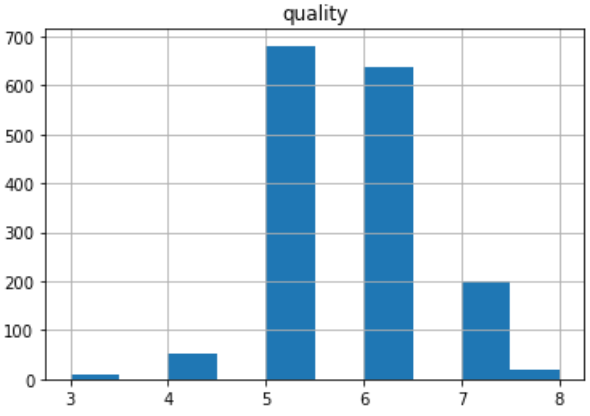












Vist els resultats, podem descartar la normalitat de les dades en pràcticament tots els valors, procedim a investigar la homogeneïtat de la variància:

```
for num in df.columns:
    stat, p = levene(random.sample(df[num].tolist(), 30), random.sample(df[num].tolist(), 29))
    print("Valor de p per a {0}: {1}".format(num, p))
    if p > 0.05:
        print("Variància homogenea.\n")
    else:
        print("Variança irregular entre poblacions.\n")
```

```
Valor de p per a fixed acidity: 0.9524498158038182
Variància homogènia.

Valor de p per a volatile acidity: 0.06855646416066166
Variància homogènia.

Valor de p per a citric acid: 0.81070623926711
Variància homogènia.

Valor de p per a residual sugar: 0.5065934893798065
Variància homogènia.

Valor de p per a chlorides: 0.21465604174794164
Variància homogènia.

Valor de p per a free sulfur dioxide: 0.4762175910147016
Variància homogènia.

Valor de p per a total sulfur dioxide: 0.6339865885054756
Variància homogènia.

Valor de p per a density: 0.42743300481091206
Variància homogènia.

Valor de p per a pH: 0.9599638869710037
Variància homogènia.

Valor de p per a sulphates: 0.14530979329549426
Variància homogènia.

Valor de p per a alcohol: 0.08019831290633714
Variància homogènia.

Valor de p per a quality: 0.732888589312784
Variància homogènia.
```

4.3. Aplicació de proves estadístiques per comparar els grups de dades. En funció de les dades i de l'objectiu de l'estudi, aplicar proves de contrast d'hipòtesis, correlacions, regressions, etc. Aplicar almenys tres mètodes d'anàlisi diferents.

Primerament separem les dades en el set de traint i test, també separant la etiqueta de classe:

```
# Separem les dades en train i test

X  = df.drop( "quality", axis=1)
y = df["quality"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

El primer model que realitzarem serà senzill regressor lineal que utilitzarem com a mètode baseline:

```
# Comprobem la bondat d'un simple regresor lineal, com a model baseline
LR = LinearRegression().fit(X_train, y_train)
```

Comprovem la bondat del model:

```
LR.score(X_test, y_test)
```

```
0.32838876395802086
```

Aplicant aquest mètode de comprovació de la bondat, el model té un resultat mediocre, posteriorment veurem que veritablement el rendiment no ha estat tant dolent.

Procedim a configurar i avaluar l'arbre de decisió:

```
# Intentem la predicció amb un arbre de decisió:
DT = DecisionTreeClassifier().fit(X_train, y_train)
DT.score(X_test, y_test)
```

```
0.6875
```

Obtenim una puntuació de precisió molt superior a l'anterior model, tot i això la score encara és modesta. Intentarem millorar-la utilitzant diferents arbres combinats en el que es coneix com a random forest:

```
# Intentem la predicció amb un random forest.
RFC = RandomForestClassifier().fit(X_train, y_train)
print(RFC.score(X_test, y_test))
```

```
0.709375
```

Hem aconseguit millorar la precisió fins al 70% amb els hiper-parametres per defecte del model. Per a intentar millorar encara més el resultat utilitzarem la funció GridSearchCV per a utilitzar validació creuada i cerca en reixeta:

```
# Afinem els hiperparametres del random forest:
parameters = {"n_estimators": (50, 100, 150, 200),
              "max_depth": range(10, 20)}

clf = GridSearchCV(RandomForestClassifier(), param_grid = parameters, cv=4)
clf.fit(X_train, y_train)

clf.score(X_test, y_test)
```

```
0.740625
```

Hem aconseguit millorar el rendiment del model en 4 punts percentuals només afinant els paràmetres. És una millora molt positiva.

Amb les dades que disposem i els models que hem anat dissenyant hem aconseguit classificar correctament quasi tres quartes parts de tots els registres, de manera que el resultat es prou satisfactori.

També cal considerar que en aquestes puntuacions es considera com a error absolut qualsevol classificació que no correspongui exactament amb la etiqueta de qualitat. Si relaxéssim una mica la exigència i consideréssim l'existència de 3 grups de vins (bons, dolents i estàndard) dins dels quals agrupéssim les diferents etiquetes de qualitat, estaríem possiblement davant d'un model amb rendiment excepcional.

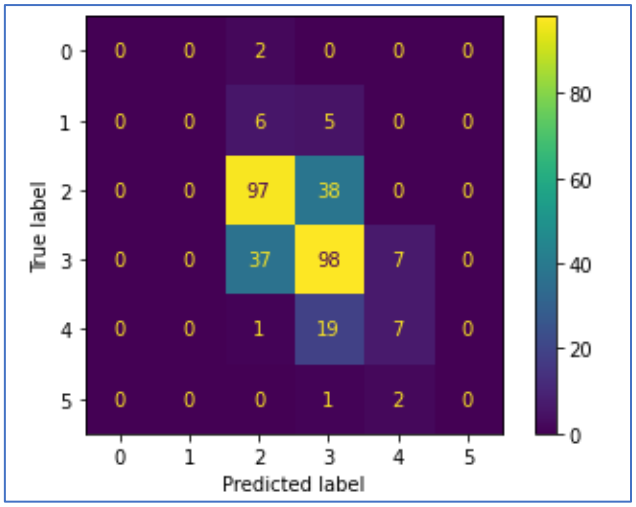
5. Representació dels resultats a partir de taules i gràfiques. (2p)

Procedim a realitzar les matrius de confusió pera veure que tant bones han estat les classificacions dels models anteriors, així com per a veure quines etiquetes s'han aconseguit classificar correctament i quines han donat més problemes:

Comencem amb la matriu de confusió del model de regressió lineal. Per a crear-la arrodonirem els resultats a la unitat, per a situar-los dintre de les etiquetes:

```
y_pred = LR.predict(X_test)
cm = confusion_matrix(y_test, np.round(y_pred))
disp = ConfusionMatrixDisplay(cm)
disp.plot()

total = y_test.size
score = ((97+98+7)*100)// total
score
```

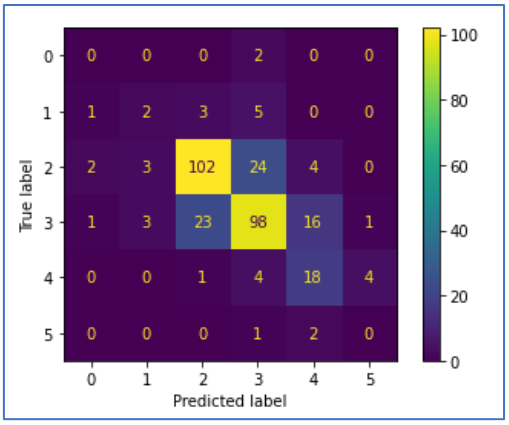
```
63
```



Tot i que la funció score anteriorment ens havia retornat molt mala puntuació, aquí ja podem veure com les classificacions correctes en realitat han estat majoria. Realitzant un petit càlcul podem obtenir la precisió un cop arrodonit y_pred, un 63%

Realitzem la matriu de confusió de l'arbre de decisió:

```
In [93]:
1 y_pred = DT.predict(X_test)
2 cm = confusion_matrix(y_test, np.round(y_pred))
3 disp = ConfusionMatrixDisplay(cm)
4 disp.plot()

Out[93]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1fc8c466be0>
```



I investiguem quins han estat les variables que més han ajudat a la predicció:

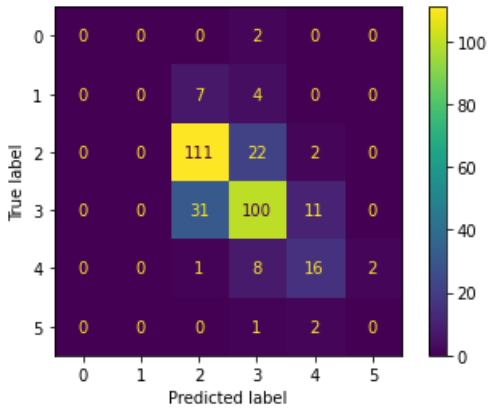
```
features = pd.DataFrame(columns = df.columns.drop("quality") )
features.loc[0] = DT.feature_importances_
features
```

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0.070962	0.10881	0.058911	0.088585	0.080098	0.081809	0.090782	0.078239	0.070054	0.100803	0.170945

L'alcohol és amb diferencia la variable que més determina la qualitat en les nostres dades, seguit dels sulfats i l'àcid volàtil.

Realitzem la matriu de confusió del Random Forest:

```
In [109]:
y_pred = RFC.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(cm)
disp.plot()

Out[109]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1fc8df11ee0>
```

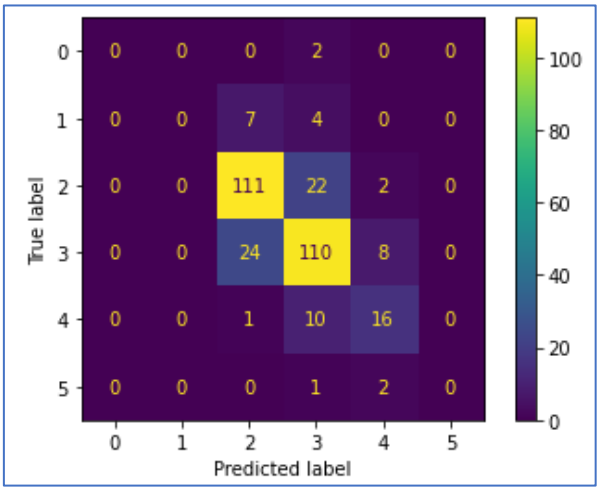


Els resultats segueixen millorant. Observem la importància de les variables en la predicció:

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0.077152	0.106785	0.075251	0.070661	0.078472	0.069912	0.098266	0.093494	0.073574	0.110034	0.1464

Podem observar com el model ara s'ha basat una mica menys en la quantitat d'alcohol i han augmentat les importàncies dels sulfats, la densitat i l'àcid cítric.

Per finalitzar realitzarem la matriu de confusió del random forest i observarem la importància de les variables.



fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0.073472	0.107855	0.074345	0.072009	0.077747	0.068269	0.103239	0.092802	0.075907	0.114824	0.139532

Podem veure com el model ha classificat correctament una gran part de les dades i per la distribució dels error podem veure que quan s'equivoca ho fa per poc, confonent un vi de qualitat x amb un vi de qualitat x+1 o x-1.

També hem pogut acceptar que a mesura que el model es va afinant, es centra menys en la variable alcohol i dona més importància a les altres, tot i que aquesta segueix sent sempre la més determinant.

6. Resolució del problema. A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten respondre al problema? (0,5p)

Classificar els vins per la seva composició química en una escala de l'1 al 10 és una operació complexa. Hi ha molts elements i moltes possibles combinacions que determinen la bondat del vi. Classificar-los numèricament ha resultat complicat per als models que hem implementat, tot i que amb el parameter-tuning hem aconseguit uns resultats més que decents.

En tot cas, crec que la exigència que ens hem auto imposat al intentar classificar-los en 10 etiquetes diferents no era la idònia. Possiblement una estratègia més fructífera hauria sigut la de discretitzar la variable qualitat en 3 o 4 valors diferents (ex: Dolent, Normal, Bo) i procedir l'anàlisi amb aquest enfocament.

Tot i l'exigència de la tasca, amb l'últim model hem aconseguit classificar correctament quasi un 75% dels registres, de manera que considero que el resultat és satisfactori per als models utilitzats i les dades que teníem (les quals estaven desequilibrades, no seguien distribucions normals i tenien etiquetes buides).

7. Codi: Cal adjuntar el codi, preferiblement en R, amb el que s'ha realitzat la neteja, anàlisi i representació de les dades. Si ho preferiu, també podeu treballar en Python. (2p)

Enllaç