Instituto Politécnico Nacional

Escuela Superior de Cómputo

Desarrollo de Aplicaciones para Análisis de Datos

Profesor: Ituriel E. Flores Estrada

Alumno: Hugo López Miguel

Examen departamental 3

Fecha: 18 de junio de 2022

Resumen ejecutivo

El conjunto de datos fue obtenido haciendo pruebas a muestras de vino rojo, estos vinos fueron elaborados en Portugal, pertenecen a la variedad denominada “Vinho verde” que hace referencia a la juventud de las uvas.

Esta variedad se produce en el noreste de Portugal, se caracteriza por la moderada cantidad de alcohol, su esencia frutal le da un sabor leve y fresco.

Hay 1599 registros individuales de las pruebas realizadas a los vinos. Existen 12 variables en el conjunto, 11 hacen referencia a pruebas objetivas –tales como las mediciones de pH, azúcar residual, porcentaje de alcohol, etc.– la única variable no objetiva es la de calidad, esta se obtuvo mediante una evaluación del 1 al 10 que realizaron al menos 3 catadores de vino.

El objetivo del presente trabajo es elaborar una regresión lineal para que, dadas ciertas variables, el modelo sea capaz de predecir la calidad del vino que evaluaría un catador.

El modelo de regresión lineal que se aplica es el que proporciona la librería *sklearn* de *Python.*

Primero se aplicó la regresión lineal al conjunto de datos sin aplicar ninguna limpieza. El propio modelo arrojó un *score* de 35%.

Posteriormente, haciendo uso de un mapa de calor de las correlaciones y observando el impacto individual de cada variable en el modelo, se descartaron aquellas variables que no eran relevantes en el modelo. Se encontró que las variables con más significancia eran “volatile acidity”, “sulphates” y “alcohol”.

El modelo obtuvo un *score* de 32.7%.

Finalmente, teniendo las variables más relevantes, para que el modelo sea más cercano a la realidad, a estas variables se les aplicó una normalización.

Y con ese último paso, el modelo nos muestra que el *score* es de 98%.

Wine Quality Data Set

El conjunto de datos se obtuvo al hacer 1599 pruebas a vinos rojos que se obtuvieron de la variedad “Vinho verde”, mismo que proviene del noreste de Portugal.

Hay 11 variables que son obtenidas a través de sensores, son variables objetivas.

La variable de calidad es subjetiva, se obtiene a través de la puntuación que otorgan al menos 3 catadores, esta calificación va del 1 hasta el 10.

A continuación se detallan las variables y se da una breve explicación:

- Fixed acidity
  Mide los ácidos involucrados en el vino, pueden ser fijados o no volátiles.

- Volatile acidity
  Cantidad de ácido acético en el vino, cuando más alto el nivel provoca un sabor similar al vinagre.

- Citric acid
  Este agrega frescura y sabor a los vinos.

- Residual sugar
  Es la cantidad de azúcar encontrada después de que se detiene la fermentación.

- Chlorides
  La cantidad de sal.

- Free sulfur dioxide
  Es el dióxido de azufre libre. Previene el crecimiento de bacterias y la oxidación del vino.

- Total sulfur dioxide
  Es el dióxido de azufre libre y el ligado. Cuanto más grandes es la medición se hará evidente su presencia ante la nariz del bebedor y se notará en el sabor.

- Density
  La densidad se acercará a la del agua dependiendo de la cantidad de alcohol y azúcar.

- pH
  Describe que tan acido o básico es un vino. Va del 0 al 14.

- Sulphates
  Es un aditivo que contribuye a subir los niveles de dióxido de azufre.

- Alcohol
  Porcentaje de alcohol en la botella de vino.

- Quality
  Puntaje entre 0 y 10.

Para la lectura de los datos, se hace uso de la función read.csv y se indica que está separado por el carácter de punto y coma.

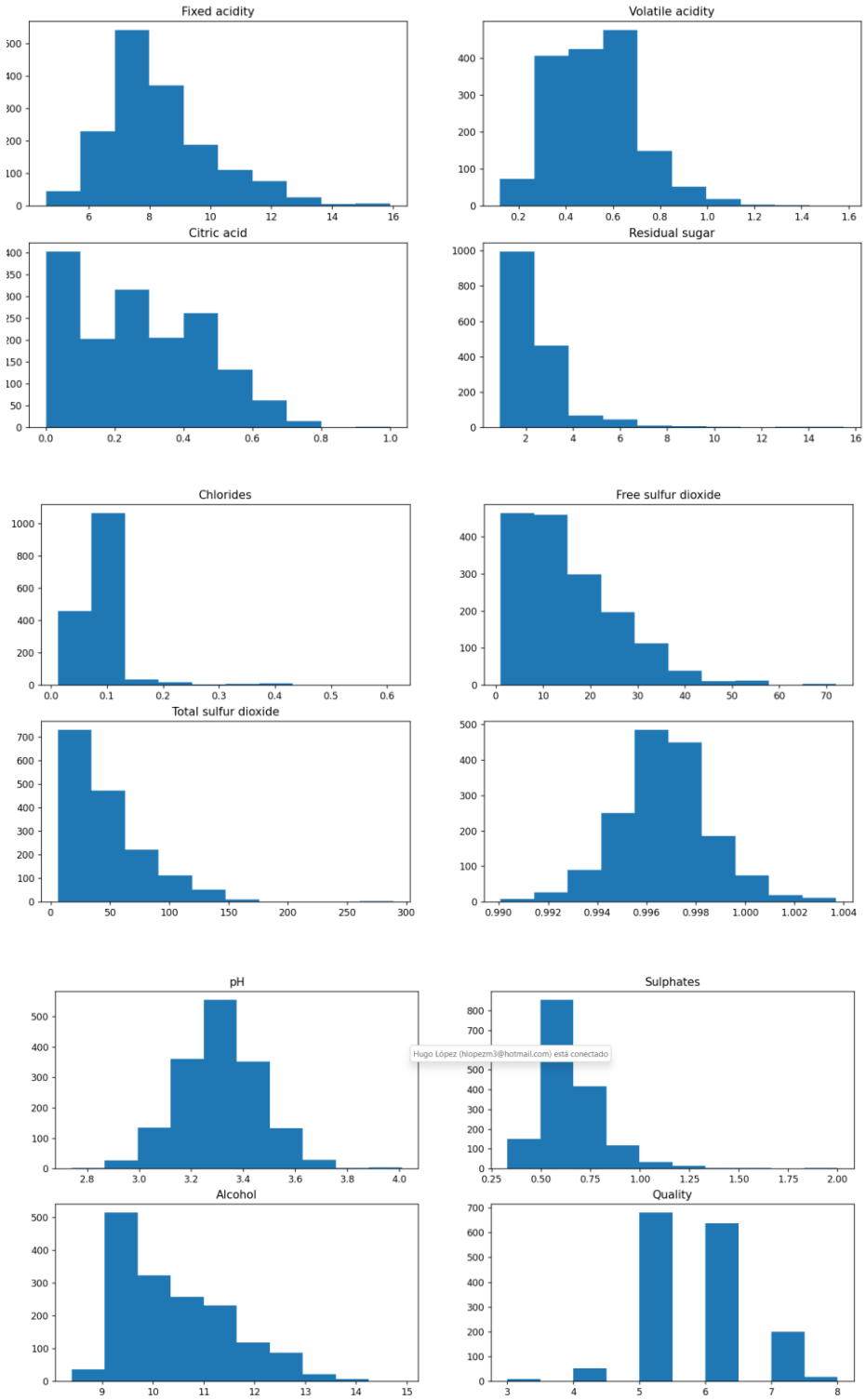
```
nombres = ['fixed acidity','volatile acidity','citric acid','residual sugar',

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

Haciendo uso de la función describe(), se obtuvieron la media, desviación estándar y los cuantiles de los 1599 registros.

	Media	DevStd	Minimo	25%	50%	75%	Máximo
Fixed acidity	8.319	1.741	4.6	7.1	7.9	9.2	15.9
Volatile acidity	0.527	0.179	0.12	0.39	0.52	0.64	1.58
Citric acid	0.27	0.194	0	0.09	0.26	0.42	1
Residual sugar	2.538	1.4	0.9	1.9	2.2	2.6	15.5
Chlorides	0.08	0.047	0.012	0.07	0.079	0.09	0.611
Free sulfur dioxide	15.87	10.46	1	7	14	21	72
Total sulfur dioxide	46.46	32.89	6	22	38	62	289
Density	0.99	0.001	0.99	0.995	0.996	0.997	1.003
pH	3.31	0.154	2.74	3.21	3.31	3.4	4.01
Sulphates	0.658	0.169	0.33	0.55	0.62	0.73	2
Alcohol	10.42	1.06	8.4	9.5	10.2	11.1	14.9
Quality	5.63	0.8	3	5	6	6	8

Posteriormente, haciendo uso de la librería matplotlib, se obtuvieron los histogramas de las 12 variables.



Así, con el conjunto sin ningún tipo de limpieza, se hizo la regresión lineal.

El porcentaje tomado del total para la prueba es 20%

```
#Regresión lineal al DF sin modificaciones
X = df.drop(columns='quality')
y = df['quality']

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2, random_state=3)

lr_multiple = linear_model.LinearRegression()

lr_multiple.fit(X_train, y_train)

Y_pred_multiple = lr_multiple.predict(X_test)

print('Coeficients: ', lr_multiple.coef_)

print('\nIntercepts: ',lr_multiple.intercept_)

print('\nScore: ', lr_multiple.score(X_train, y_train))
```

```
Coeficients:  [ 3.21701286e-02 -1.03467859e+00 -1.53320498e-01  1.23460437e-02
 -1.61715049e+00  5.08258596e-03 -3.32744691e-03 -1.57794200e+01
 -3.84377830e-01  8.10208705e-01  2.88021969e-01]

Intercepts:  19.611580823864532

Score:  0.3530297271672964
```

A continuación, se muestra un resumen del modelo:

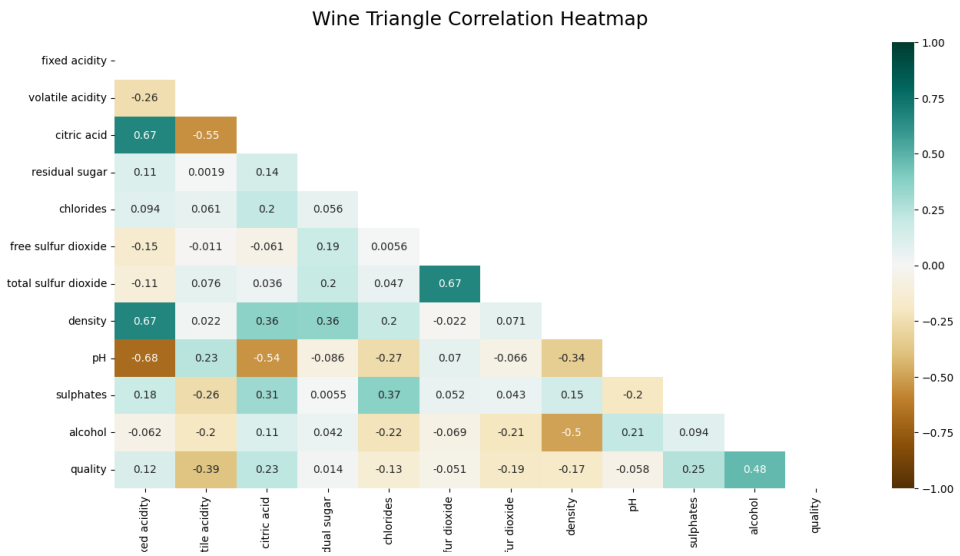
OLS Regression Results

Dep. Variable:	quality	R-squared:	0.353
Model:	OLS	Adj. R-squared:	0.347
Method:	Least Squares	F-statistic:	62.85
Date:	Sun, 19 Jun 2022	Prob (F-statistic):	1.16e-111
Time:	02:33:23	Log-Likelihood:	-1275.2
No. Observations:	1279	AIC:	2574.
Df Residuals:	1267	BIC:	2636.
Df Model:	11		
Covariance Type:	nonrobust		

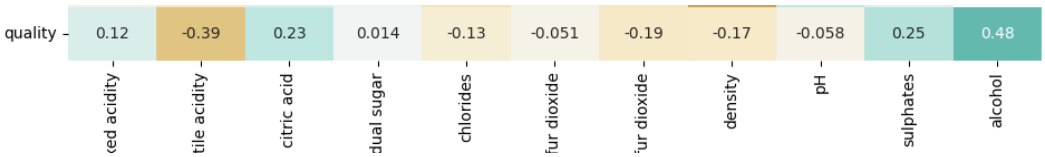
	coef	std err	t	P>|t|	[0.025	0.975]
const	19.6116	24.010	0.817	0.414	-27.492	66.715
fixed acidity	0.0322	0.029	1.102	0.271	-0.025	0.089
volatile acidity	-1.0347	0.136	-7.627	0.000	-1.301	-0.769
citric acid	-0.1533	0.169	-0.910	0.363	-0.484	0.177
residual sugar	0.0123	0.018	0.701	0.483	-0.022	0.047
chlorides	-1.6172	0.490	-3.302	0.001	-2.578	-0.656
free sulfur dioxide	0.0051	0.002	2.066	0.039	0.000	0.010
total sulfur dioxide	-0.0033	0.001	-4.088	0.000	-0.005	-0.002
density	-15.7794	24.488	-0.644	0.519	-63.820	32.261
pH	-0.3844	0.213	-1.809	0.071	-0.801	0.033
sulphates	0.8102	0.128	6.354	0.000	0.560	1.060
alcohol	0.2880	0.030	9.548	0.000	0.229	0.347

Omnibus:	21.196	Durbin-Watson:	1.959
Prob(Omnibus):	0.000	Jarque-Bera (JB):	32.340
Skew:	-0.147	Prob(JB):	9.49e-08
Kurtosis:	3.721	Cond. No.	1.13e+05

Para limpiar el conjunto, se eliminan las variables menos relevantes para determinar la calidad, para eso haremos uso de un mapa de calor de las correlaciones.



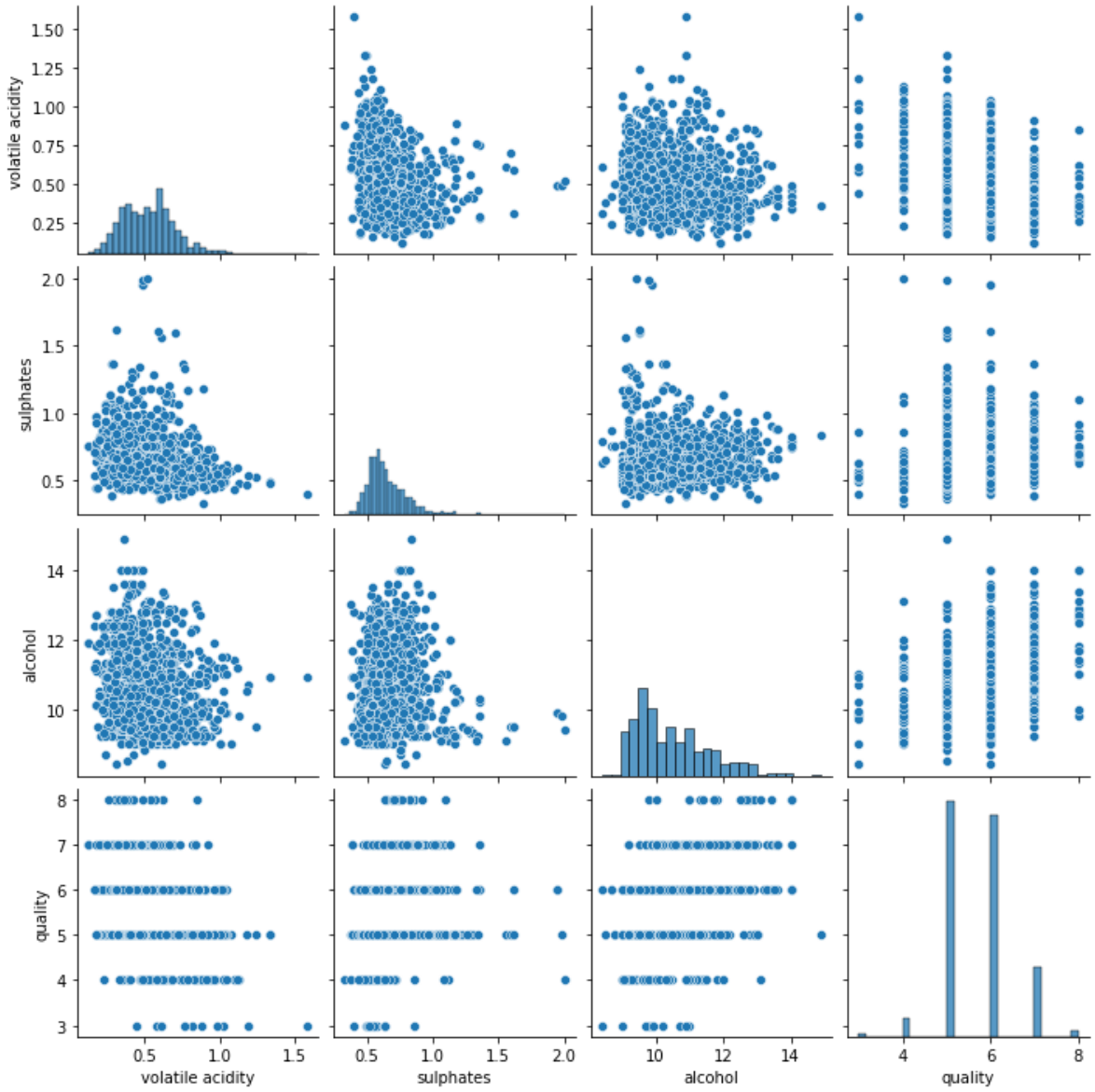
Solo nos interesa la columna de calidad:



Se decidió usar “volatile acidity”, “sulphates” y “alcohol por sus valores más altos en correlación con la calidad.

“Citric acid” y “fixed acidity” no se usaron porque tienen alta correlación con “volatile acidity”.

A continuación, vemos el pairplot de este dataframe:



Lo ingresamos al modelo:

```
#Baja:    ['volatile acidity','sulphates', 'alcohol']

dfC = df.drop(columns=['citric acid','chlorides','total sulfur dioxide','density','fixed acidity','residual sugar','free sulfur dioxide','pH'])

X = dfC.drop(columns='quality')
y = dfC['quality']


X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2, random_state=3)

lr_multiple = linear_model.LinearRegression()

lr_multiple.fit(X_train, y_train)

Y_pred_multiple = lr_multiple.predict(X_test)

print('Coeficients: ', lr_multiple.coef_)

print('\nIntercepts: ',lr_multiple.intercept_)

print('\nScore: ', lr_multiple.score(X_train, y_train))
```

```
Coeficients:  [-1.19262087  0.64498975  0.31509659]

Intercepts:  2.5641414537695177

Score:  0.3297131996335533
```

Sorpresivamente, el puntaje del modelo baja.

A base de prueba y error, se observó que las variables del comentario de arriba, cuando eran retiradas del modelo, este bajaba más.

Cuando se retiraban las otras variables, el puntaje del modelo era similar, es decir, no disminuía si se prescindía de tales variables.

Resumen del modelo:

OLS Regression Results

Dep. Variable:	quality	R-squared:	0.330
Model:	OLS	Adj. R-squared:	0.328
Method:	Least Squares	F-statistic:	209.1
Date:	Sun, 19 Jun 2022	Prob (F-statistic):	2.86e-110
Time:	02:37:25	Log-Likelihood:	-1297.9
No. Observations:	1279	AIC:	2604.
Df Residuals:	1275	BIC:	2624.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.5641	0.223	11.481	0.000	2.126	3.002
volatile acidity	-1.1926	0.110	-10.890	0.000	-1.407	-0.978
sulphates	0.6450	0.113	5.692	0.000	0.423	0.867
alcohol	0.3151	0.018	17.483	0.000	0.280	0.350

Omnibus:	17.977	Durbin-Watson:	1.941
Prob(Omnibus):	0.000	Jarque-Bera (JB):	28.555
Skew:	-0.093	Prob(JB):	6.30e-07
Kurtosis:	3.708	Cond. No.	134.

Finalmente, para maximizar la exactitud del modelo, se procedió a normalizar el dataset.

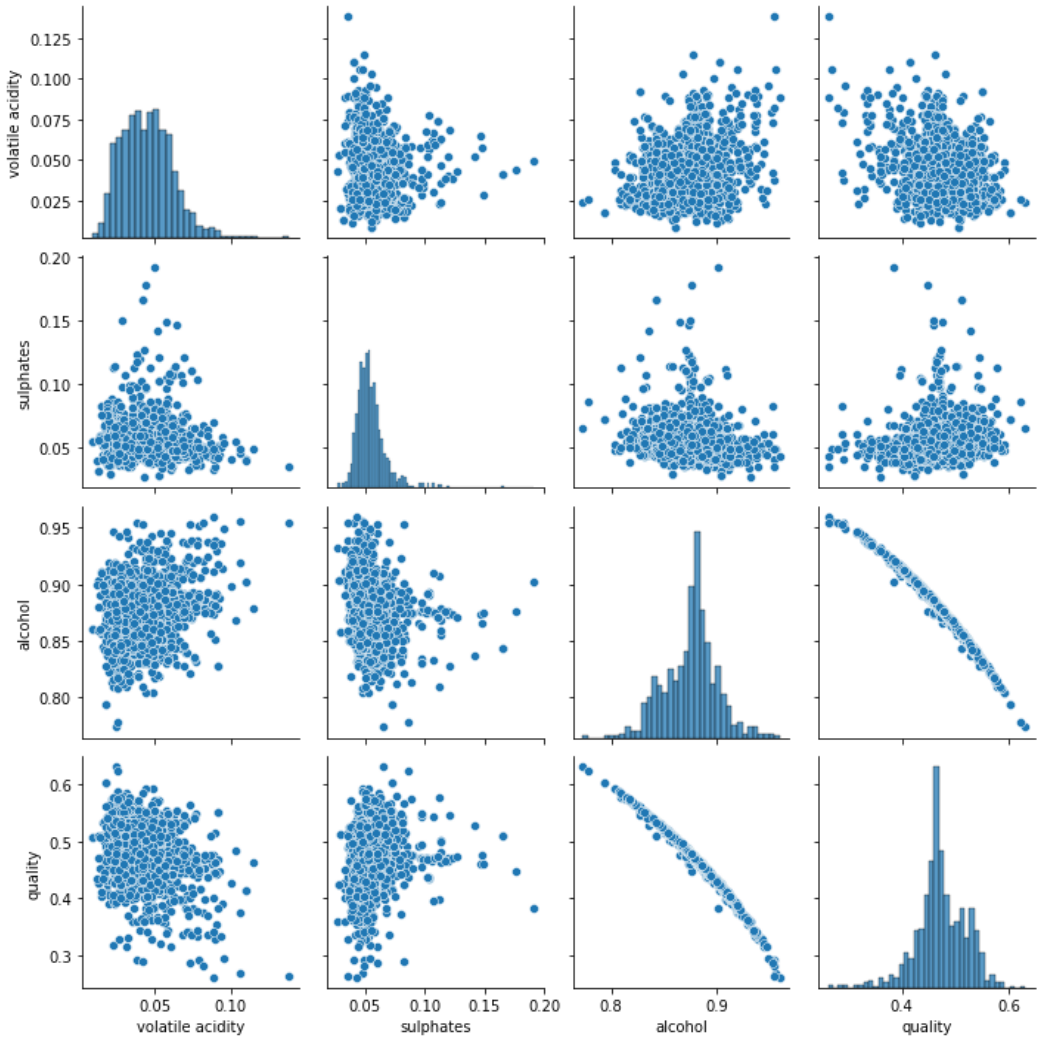
```
from sklearn import preprocessing
d = preprocessing.normalize(dfC)

scaled_df = pd.DataFrame(d, columns= ['volatile acidity', 'sulphates','alcohol','quality'])

print(scaled_df.head())
```

```
   volatile acidity  sulphates   alcohol   quality
0          0.065514   0.052411  0.879760  0.467957
1          0.079581   0.061495  0.886246  0.452166
2          0.068796   0.058839  0.887105  0.452604
3          0.024329   0.050396  0.851515  0.521336
4          0.065514   0.052411  0.879760  0.467957
```

Pairplot:



Y ahora se aplica al modelo:

```
X = scaled_df.drop(columns='quality')
y = scaled_df['quality']


X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2, random_state=3)

lr_multiple = linear_model.LinearRegression()

lr_multiple.fit(X_train, y_train)

Y_pred_multiple = lr_multiple.predict(X_test)

print('Coeficients: ', lr_multiple.coef_)

print('\nIntercepts: ',lr_multiple.intercept_)

print('\nScore: ', lr_multiple.score(X_train, y_train))
```

```
Coeficients:  [-0.13808095 -0.17445751 -1.87470206]

Intercepts:  2.1315084775251694

Score:  0.9858561370317979
```

Resumen:

OLS Regression Results

Dep. Variable:	quality	R-squared:	0.986
Model:	OLS	Adj. R-squared:	0.986
Method:	Least Squares	F-statistic:	2.962e+04
Date:	Sun, 19 Jun 2022	Prob (F-statistic):	0.00
Time:	02:39:17	Log-Likelihood:	4797.3
No. Observations:	1279	AIC:	-9587.
Df Residuals:	1275	BIC:	-9566.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.1315	0.006	365.150	0.000	2.120	2.143
volatile acidity	-0.1381	0.010	-14.228	0.000	-0.157	-0.119
sulphates	-0.1745	0.011	-16.068	0.000	-0.196	-0.153
alcohol	-1.8747	0.007	-284.652	0.000	-1.888	-1.862

Omnibus:	1060.750	Durbin-Watson:	1.886
Prob(Omnibus):	0.000	Jarque-Bera (JB):	26480.458
Skew:	-3.776	Prob(JB):	0.00
Kurtosis:	23.973	Cond. No.	94.5

Nótese como el R-cuadrada y el Score del modelo son valores similares, aproximadamente un 98%.

Recordemos que el valor de R-cuadrada cercano al 0 significa que las variables del modelo no pueden explicar a la variable objetivo.

Un valor de 1 indica que la variable de respuesta se puede explicar perfectamente sin errores mediante la variable predictora.

Nuestro modelo es cercano al 1.

Se obtiene una ecuación:

$$y = 2.1315 - 0.138\,x1 - 0.1744\,x2 - 1.8747\,x3 + \varepsilon$$

Donde x1= volatile acidity,

X2 = sulphates,

X3 = alcohol,

ε = Error