Selección de Características - Diego Jafet Garza Segovia

Primero se debe de importar los datos de la base de datos (Vino Tinto.csv) y guardar dentro de una variable (AE). Para esto se ocupara importar la libreria de pandas (pd).

Se mostrara despues las dimensiones del data frame de la base de datos (usando el atributo shape), y las primeras 5 filas de datos de esta misma (usando la funcion head()), mostrando asi sus columnas y tipos de datos que contiene.

In [59]:
```
import pandas as pd
AE= pd.read_csv("Vino Tinto.csv")
print("Dimensiones del data frame: ", AE.shape)
print(AE.head(5))
```

```
Dimensiones del data frame:  (1599, 12)
   acidezFija  acidezVolatil  acidoCitrico  azucarResidual  cloruros  \
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

   dioxidoAzufreLibre  dioxidoAzufreTotal  densidad    pH  sulfatos  alcohol  \
0                11.0                34.0    0.9978  3.51      0.56      9.4
1                25.0                67.0    0.9968  3.20      0.68      9.8
2                15.0                54.0    0.9970  3.26      0.65      9.8
3                17.0                60.0    0.9980  3.16      0.58      9.8
4                11.0                34.0    0.9978  3.51      0.56      9.4

   calidad
0        5
1        5
2        5
3        6
4        5
```

Ahora, se separara el data frame de manera aleatoria en datos de entrenamiento (80% del total -> 1279/1599 datos) y datos de prueba (20% del total -> 320/1599 datos) usando las funciones sample y drop respectivamente.

Posteriormente para comprobar que los datos se repartieron adecuadamente en 80/20, se imprimira en la consola las dimensiones de estos.

In [61]:
```
entre = AE.sample(frac=0.8) #entrenamiento
prueba = AE.drop(entre.index)

print("Dimensiones de datos de entrenamiento: ", entre.shape)
print("Dimensiones de datos de prueba: ", prueba.shape)
```

```
Dimensiones de datos de entrenamiento:  (1279, 12)
Dimensiones de datos de prueba:  (320, 12)
```

Como se puede ver que las dimensiones de los datos de entrenamiento y prueba son correctos, se generara la metodología de selección hacia adelante y se imprimira en la consola los nombres de las variables seleccionadas.

Para esto, hay que importar de la libreria mlxtend.feature_selection la clase SequentialFeatureSelector (SFS) para poder usar esta funcion que nos dara la seleccion hacia adelante, y de sklearn.lineal_model la clase LinearRegression para poder sacar la regresion lineal, el cual es usado dentro del SFS. Se seleccionaran entre 2 y 8 variables.

Adicionalmente, se deberan crear las variables X y Y, las cuales tendran el valor de los datos de las columnas correspondientes, siendo Y la calidad y X el resto de las variables.

Se imprimira despues, los nombres de las variables seleccionadas por la metodología SFS.

In [62]:
```
from mlxtend.feature_selection import SequentialFeatureSelector
from sklearn.linear_model import LinearRegression

X = entre[['acidezFija', 'acidezVolatil', 'acidoCitrico', 'azucarResidual', 'clorur
Y = entre['calidad']

SFS = SequentialFeatureSelector(estimator = LinearRegression(), k_features = (2,8),
SFS = SFS.fit(X, Y)

print("Nombre de variables seleccionadas: ", list(SFS.k_feature_names_))
```

```
Nombre de variables seleccionadas:  ['acidezVolatil', 'cloruros', 'dioxidoAzufreLibr
e', 'dioxidoAzufreTotal', 'pH', 'sulfatos', 'alcohol']
```

Ahora se entrenara un modelo que solamente contenga las variables seleccionadas, para lo cual se usara la funcion fit pero con la variable X actualizada (Xseleccionados).

Despues, se hara una prediccion de la respuesta en las observaciones de prueba. Se deberan generar las variables X y Y (Xprueba, Yprueba) con los datos correspondientes a las pruebas (con los datos de X ya limitados a las variables seleccionadas). Se usara la funcion predict.

Luego, se calculara la R^2 para medir la capacidad de predicción del modelo. Para esto se debera importar de la libreria sklearn.metrics la clase r2_score y poder usar su funcion. Posteriormente se imprimira en la consola el valor de este mismo.

In [63]:
```
Xseleccionados = X[list(SFS.k_feature_names_)]
modelo = LinearRegression().fit(Xseleccionados, Y)

Xprueba = prueba[list(SFS.k_feature_names_)]
Yprueba = prueba['calidad']

prediccion = modelo.predict(Xprueba)

from sklearn.metrics import r2_score
```

```
r2 = r2_score(Yprueba, prediccion)
print("r^2 = ", r2)
```

```
r^2 =  0.3340761448060561
```

Teniendo ahora el R^2 del modelo obtenido usando la metodología de selección hacia adelante, se tomaran las variables seleccionadas y aplicadas en un proceso de selección hacia atrás (SFSatras).

Entre 2 y 5 variables seran seleccionadas, y posteriormente el nombre de estas seran impresas en la consola.

In [64]:
```
SFSatras = SequentialFeatureSelector(estimator = LinearRegression(), k_features = (
SFSatras = SFSatras.fit(Xseleccionados, Y)

print("Nombre de variables seleccionadas: ", SFSatras.k_feature_names_)
```

```
Nombre de variables seleccionadas:  ('acidezVolatil', 'cloruros', 'dioxidoAzufreTota
l', 'sulfatos', 'alcohol')
```

Nuevamente, se entrenara un modelo (modelo2) que solamente contenga las nuevas variables seleccionadas (Xseleccionados2), se realizara una prediccion de la respuesta en las observaciones de prueba (prediccion2) para el cual se usara la nueva variable X de prueba (Xprueba2) con los nuevos datos seleccionados, y se obtendra el valor R^2 correspondiente (r22).

Similarmente a como se realizo anteriormente, tambien se imprimira el valor de R^2 en la consola.

In [67]:
```
Xseleccionados2 = X[list(SFSatras.k_feature_names_)]
modelo2 = LinearRegression().fit(Xseleccionados2, Y)

Xprueba2 = prueba[list(SFSatras.k_feature_names_)]

prediccion2 = modelo2.predict(Xprueba2)

r22 = r2_score(Yprueba, prediccion2)
print("r^2 = ", r22)
```

```
r^2 =  0.33021077280422595
```

Al comparar el primer valor de R^2 (0.3341) obtenido con el segundo valor (0.3302), obtenemos que el primer valor fue mayor que el segundo por 0.0039.

Debido a que el R^2 con un proceso de seleccion hacia adelante fue mayor al R^2 con el proceso de seleccion hacia atras, se puede concluir que el primer modelo de 7 variables es mejor que el segundo modelo de 5 variables.

Sin embargo debido a la poca diferencia que hay entre los valores de R^2, el segundo modelo tambien podria ser viable ya que contiene menos variables, significando que hay una menor probabilidad de que este siendo sobre ajustado a los parametros actuales.