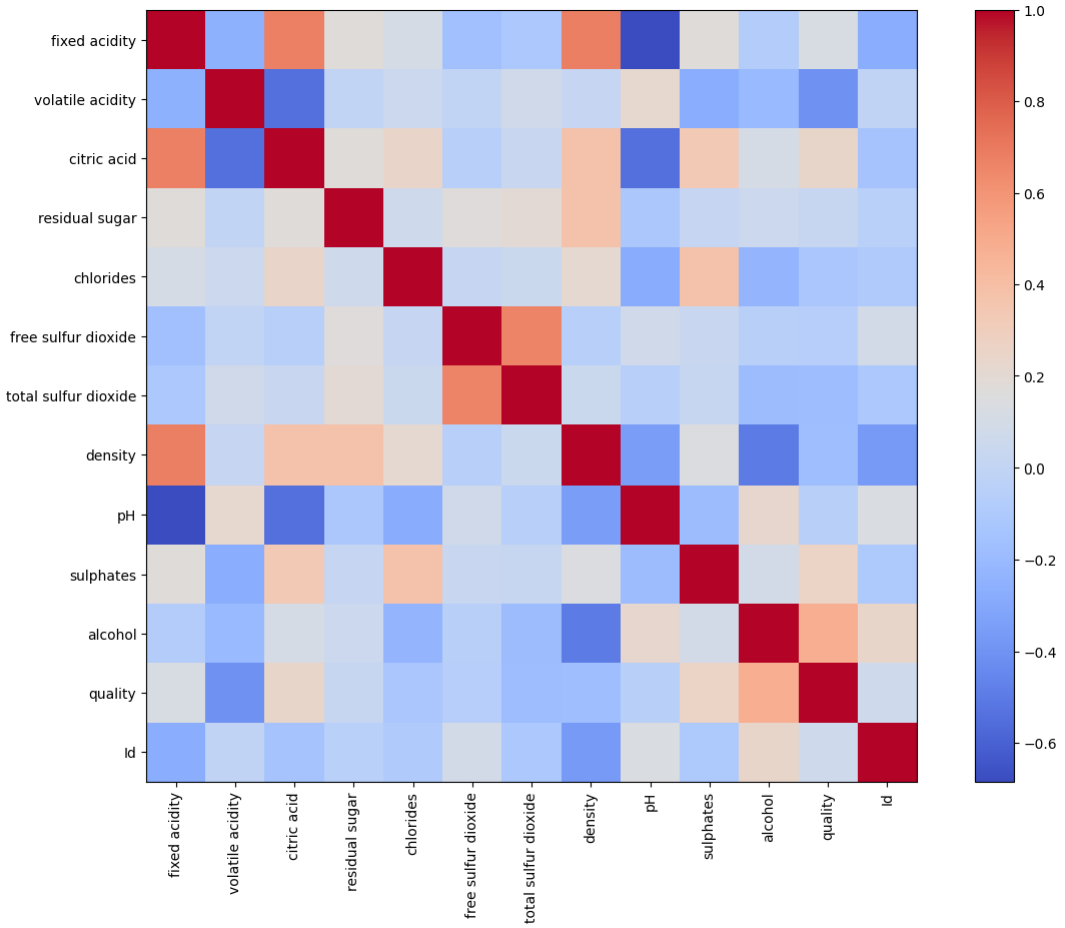
**Predicción de Calidad del Vino (Core)**

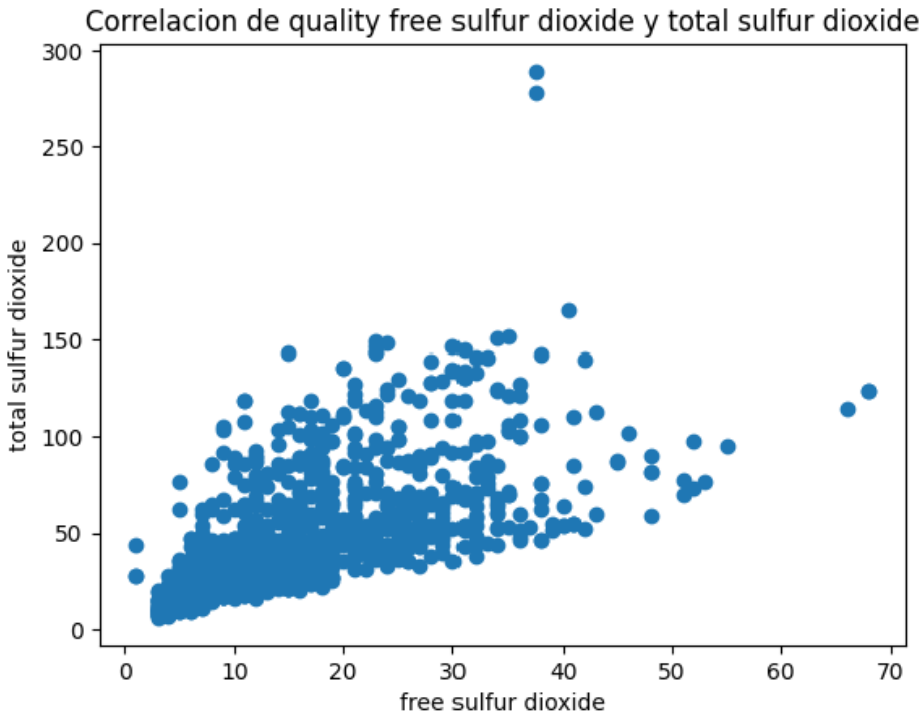
Este proyecto comenzó con la descarga y posterior importación a un Dataframe para su análisis.

Como en todo proyecto realicé un EDAD para analizar la información contenida en el dataset cargado.

Dentro del EDA realizado, obtuve información tales como las estadísticas de las columnas numéricas, hice un grafico de mapa de calor para analizar estas variables y poder ver si existían correlaciones entre ellas. En este gráfico pude ver que existe relación entre algunas variables, como fixed acidity con density, como también entre free sulfur dioxide y total sulfur dioxide

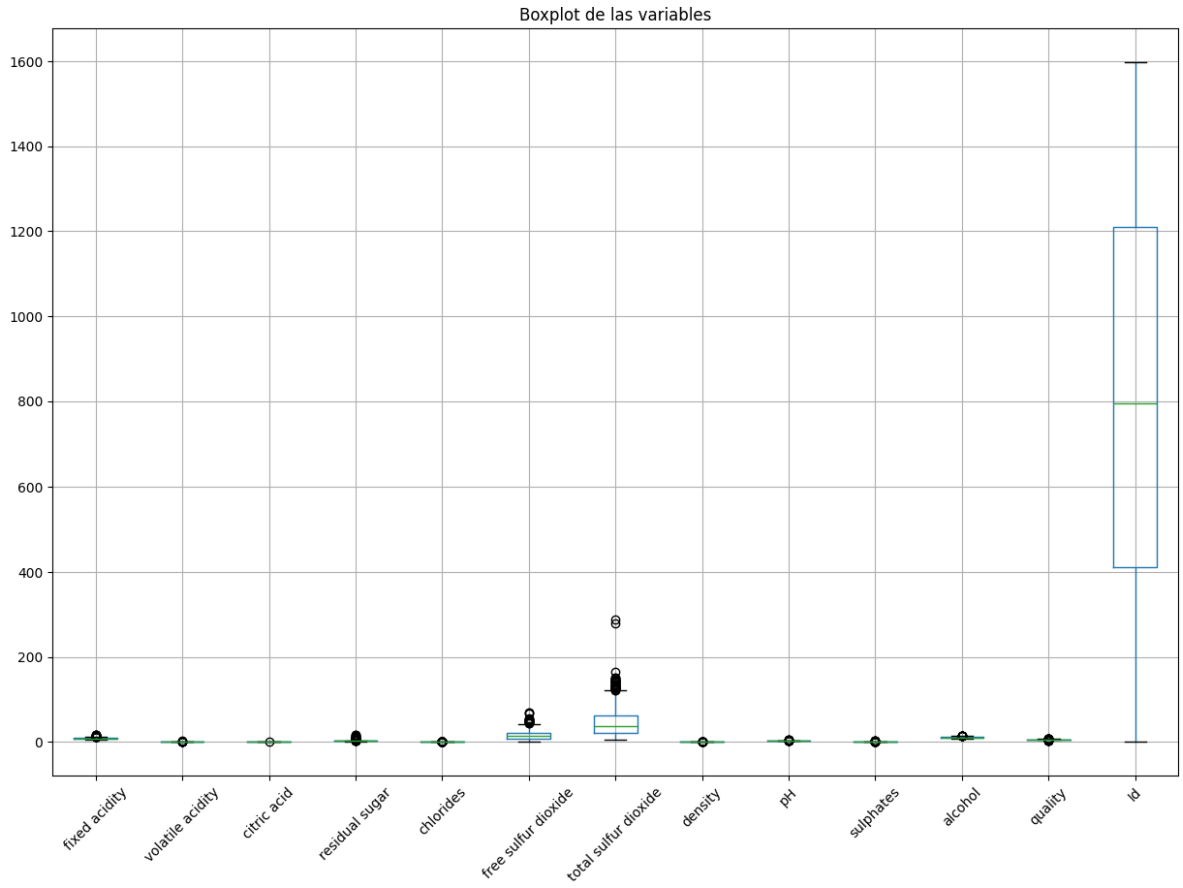


Tome un de estas relaciones y la grafiqué, con esto muestro la relación entre las columnas free sulfur dioxide y total sulfur dioxide

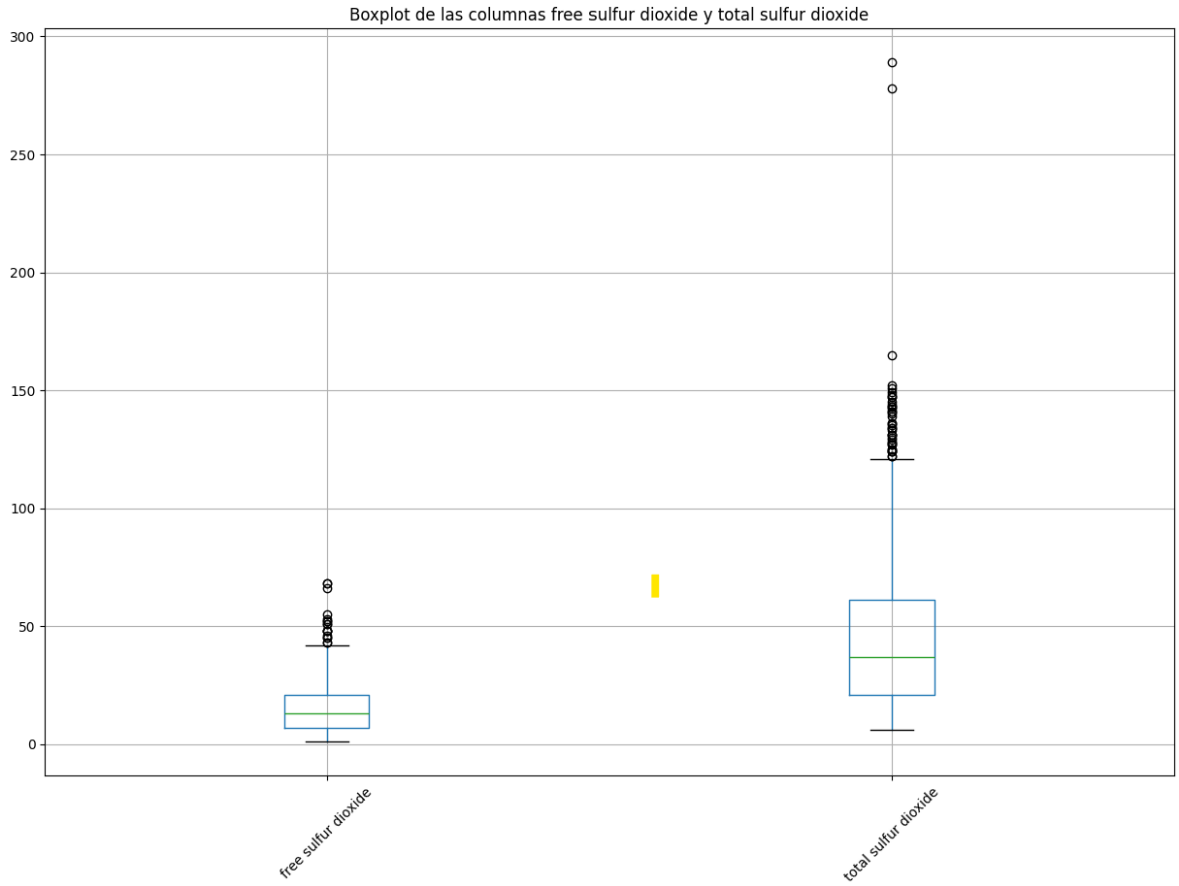



Posteriormente analizo la existencia de nulos para poder trabajarlos, pero no encontré columnas con valores NaN.

Otro paso importante era validar los valores outliers por lo que utilicé un gráfico del tipo Boxplot y pude ver que aparecían 3 columnas con valores atípicos




Por no tener conocimientos respecto de los valores de las columnas en cuestión, y no entender si esos valores son reales o errores, opté por no hacer nada con ellos y los mantuve dentro de mi dataframe




Un entendido en la materia, por ejemplo, un enólogo, podría explicarnos estos valores, quizás sean correctos o errados. Para el caso del ejercicio, estos datos se mantendrán en el dataframe

Una vez analizado los datos comencé la etapa de preprocesar los datos, donde seleccionaba las características importantes del dataframe, revisé las variables por sus tipos de datos para transformarlas y/o escalarlas.

Habiendo trabajado los datos, tuve que aplicar 3 modelos de clasificación, y opte por los siguientes:

- Arbol de Decision
- KNN
- RandomForest

Para cada modelo seleccionado, cree un transformador y su respectivo Pipeline con los cuales entrené los modelos con un 80% de los datos, dejando un 20% para validarlos

Los resultados de las evaluaciones de accuracy fueron los siguiente:

- Accuracy Arbol de Decision: 0.5502183406113537
- Accuracy KNN: 0.5589519650655022
- Accuracy Random Forest: 0.6637554585152838

Según los resultados de los accuracy, el mejor algoritmo resultó ser el Ramdom Forest con un 66% de precisión. Para poder apreciar los resultados antes mencionados, desplegué sus respectivas matrices de confusiones.

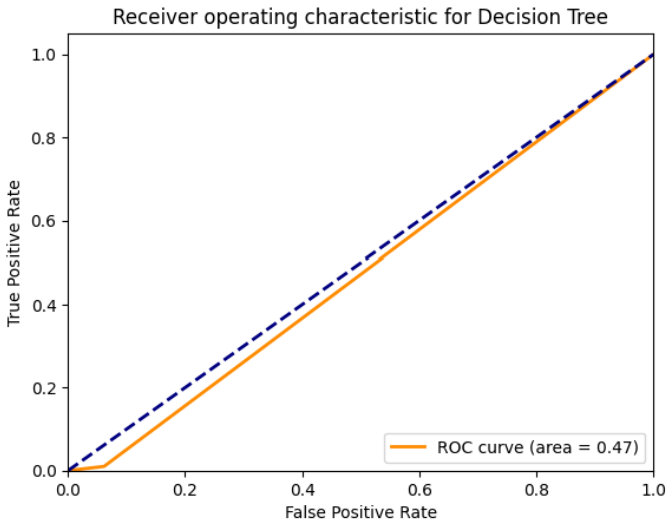
```
Matriz de confusión Arbol de Decision:
 [[ 0  0  0  0  0  0]
 [ 0  1  2  2  1  0]
 [ 1  7 58 30  0  0]
 [ 2  1 33 50 11  2]
 [ 0  0  0 10 16  0]
 [ 0  0  0  0  1  1]]
Matriz de confusión KNN:
 [[ 0  4  2  0  0]
 [ 2 66 28  0  0]
 [ 1 34 55  9  0]
 [ 0  5 14  7  0]
 [ 0  0  2  0  0]]
Matriz de confusión Random Forest:
 [[ 0  3  3  0  0]
 [ 0 75 20  1  0]
 [ 0 31 61  7  0]
 [ 0  0 10 16  0]
 [ 0  0  1  1  0]]
```

Teniendo estos resultados, generé el informe de cada modelo utilizando la función classification_report

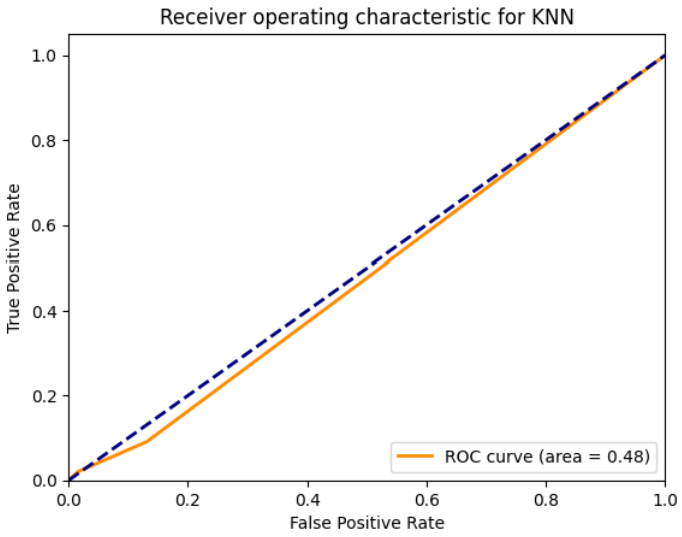
```
Informe de clasificación Arbol de Decision:
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         0
           4       0.11      0.17      0.13         6
           5       0.62      0.60      0.61        96
           6       0.54      0.51      0.52        99
           7       0.55      0.62      0.58        26
           8       0.33      0.50      0.40         2

    accuracy                           0.55       229
   macro avg       0.36      0.40      0.38       229
weighted avg       0.56      0.55      0.56       229

Informe de clasificación KNN:
              precision    recall  f1-score   support

           4       0.00      0.00      0.00         6
           5       0.61      0.69      0.64        96
           6       0.54      0.56      0.55        99
           7       0.44      0.27      0.33        26
           8       0.00      0.00      0.00         2

    accuracy                           0.56       229
   macro avg       0.32      0.30      0.31       229
weighted avg       0.54      0.56      0.55       229

Informe de clasificación Random Forest:
              precision    recall  f1-score   support

           4       0.00      0.00      0.00         6
           5       0.69      0.78      0.73        96
           6       0.64      0.62      0.63        99
           7       0.64      0.62      0.63        26
           8       0.00      0.00      0.00         2

    accuracy                           0.66       229
   macro avg       0.39      0.40      0.40       229
weighted avg       0.64      0.66      0.65       229
```

Después de esto imprimí las curvas de ROC de cada modelo

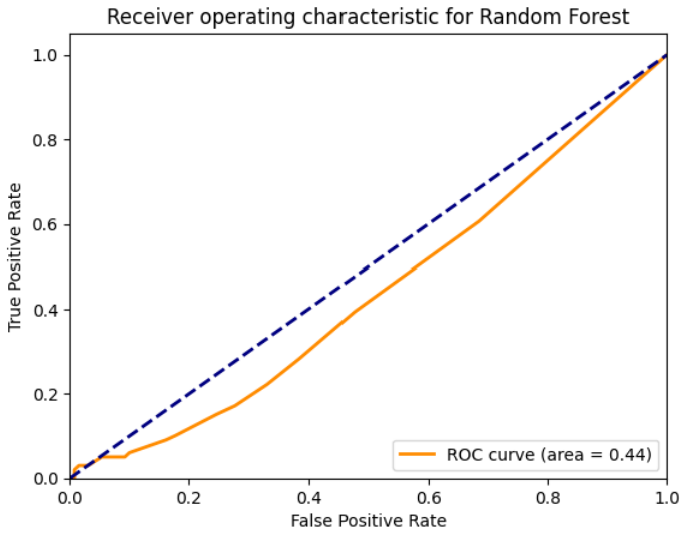
### Decision Tree



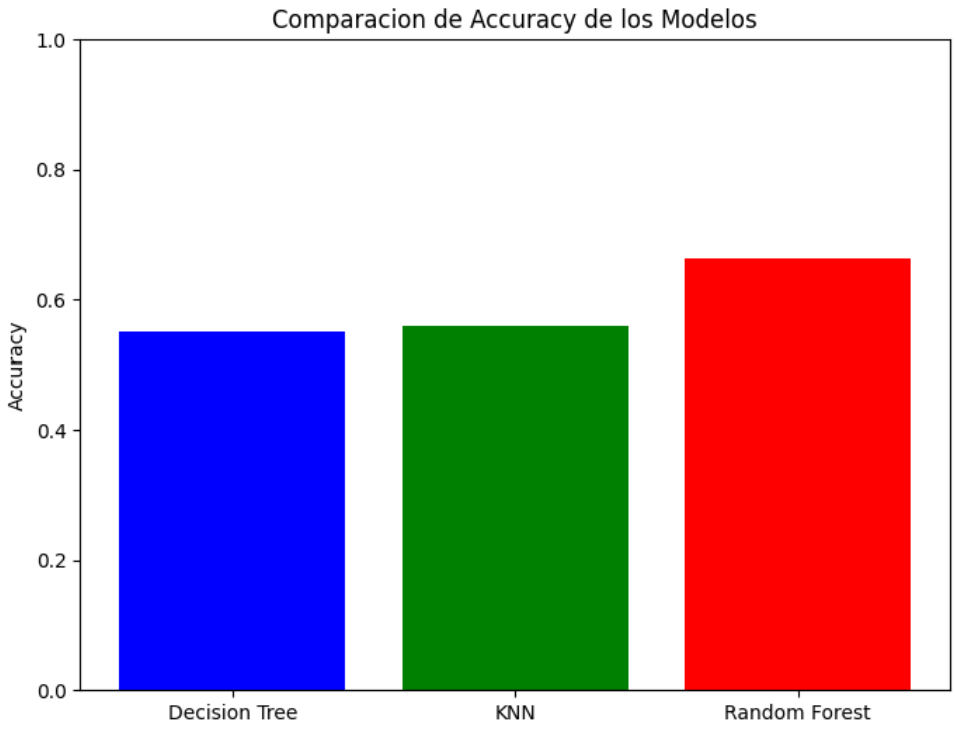
**K-Neighbors**




**Ramdom Forest**




Para terminar presento un gráfico con ls comparación de los accuracy para mostrar el mejor modelo, donde se aprecia que el Ramdom Forest fue el que presentó el mejor rendimiento


Comparacion de Accuracy de los Modelos
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Decision Tree
KNN
Random Forest