Proyecto Final

Daniel Espinosa Mireles de Villafranca 136981

Introducción

El objetivo de este trabajo es utilizar tres diferentes modelos de clasificación detallando cada paso de la metodología utilizada para construir cada uno y comparar los resultados de los tres.

El dataset que se utiliza es el Wine Quality Dataset, en el cual hay 11 variables explicativas sobre la información del análisis químico de muestras de vino y una variable target en la cual se califica la calidad del vino del 0 al 10. El problema puede ser resuelto utilizando regresión o clasificación dado que la variable target se puede interpretar como numérica ordinal, como clases distintas o bien establecer un punto de corte para el cual se considera un vino bueno y clasificar un vino como bueno o malo.

En este trabajo se seleccionará un punto de corte que parezca adecuado en la escala de la calidad de los vinos para poder decidir si un vino es bueno o malo.

Las variables que contiene el dataset se describen a continuación:

- 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
- 12 - quality (valor entre 0 y 10) (**OUTPUT**)

(El dataset tiene 4898 observaciones)

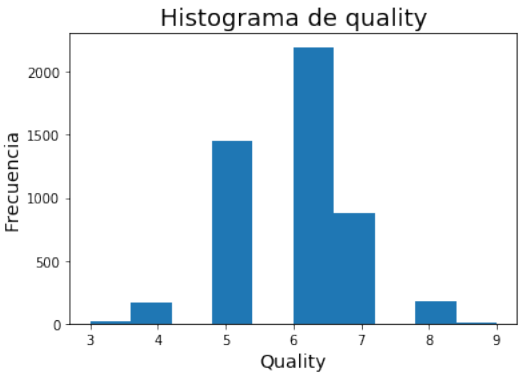
La calidad de un vino es subjetiva y dependerá del gusto de cada persona, sin embargo tomando la escala de la calidad como algo objetivo se debe de poder relacionar con la composición química de cada vino, pues esto afectará el aroma, sabor y cuerpo

Data Profiling

En esta sección se analiza la distribución de las variables para seleccionar el punto de corte en la calificación de calidad del vino con la cual se considerará bueno o malo.

Se analiza también la correlación de Pearson de las variables explicativas respecto a la variable target así como la importancia de las variables según la información que aportan para la clasificación calculada con el *Gini impurity index* para hacer *feature selection* y reducir la dimensionalidad.

A continuación se muestra el histograma de la distribución del la variable **quality**:



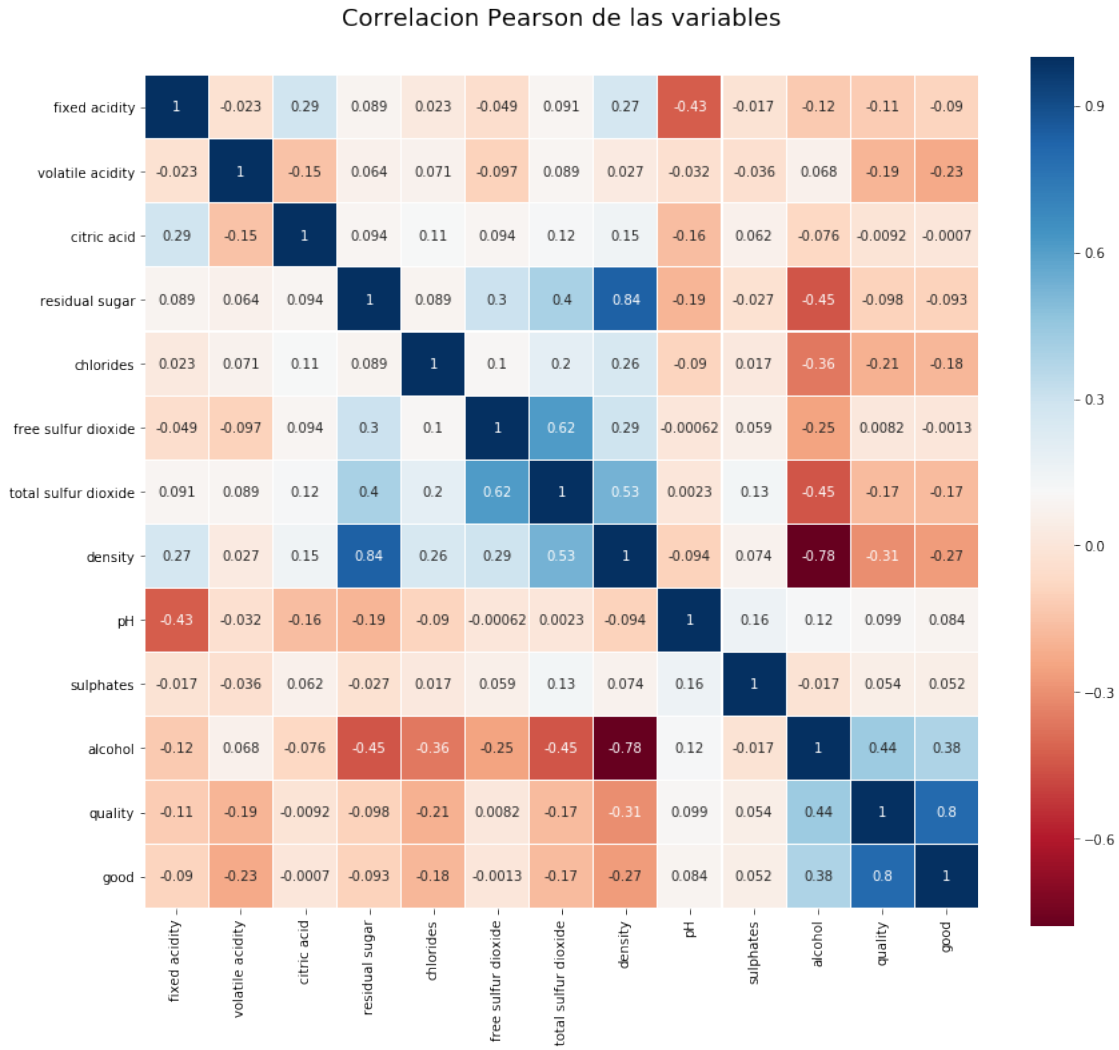
Observando la distribución de la calidad de los vinos, se decide clasificar un vino como bueno si tiene una calificación igual o mayor a 6 y como malo si tiene una calificación menor a 6. Esto se agrega al dataset como la variable **good** que toma el valor de 1 si el vino es considerado bueno (igual o mayor a 6 en calidad) y 0 si es malo (menor a 6 en calidad).

A continuación se presenta la tabla del resumen de las variables en el dataset.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	489
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267	0.489847	10.
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001	0.114126	1.2
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000	0.220000	8.0
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000	0.410000	9.5
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000	0.470000	10.
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000	0.550000	11.
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000	1.080000	14.

alcohol	quality	good
4898.000000	4898.000000	4898.000000
10.514267	5.877909	0.665169
1.230621	0.885639	0.471979
8.000000	3.000000	0.000000
9.500000	5.000000	0.000000
10.400000	6.000000	1.000000
11.400000	6.000000	1.000000
14.200000	9.000000	1.000000

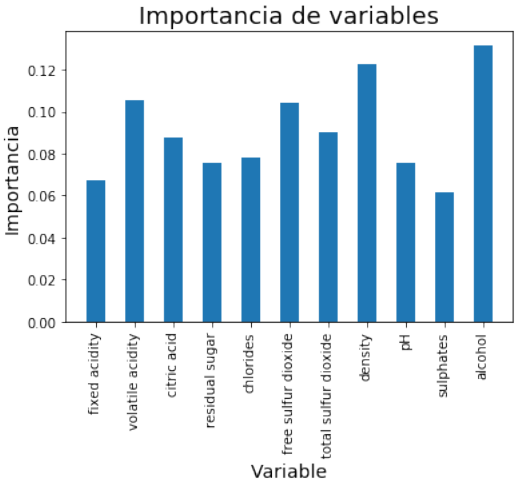
Se calcula la correlación Pearson de las variables para visualizar qué variables tienen correlación más fuerte contra la calidad del vino. A continuación se presenta la gráfica de correlación:



Con esta información se puede hacer *feature selection* básica, por ejemplo si se toman como relevantes las variables que tengan valor absoluto de correlación mayor o igual a 0.1 contra quality, las variables resultantes son:

- fixed acidity
- volatile acidity
- chlorides
- total sulfur dioxide
- density
- alcohol

Sin embargo, para el *feature selection* en este trabajo se ordenan las variables respecto a su importancia calculada al correr un modelo Random Forest, que califica la importancia de las variables según el *Gini impurity index* que tiene correlación con la cantidad de información que aportan las variables respecto a la varaible target. Es importante mencionar que para este punto ya se divide el dataset en 75% para entrenamiento y el restante 25% para pruebas, pues es importante que no haya fugas de información del dataset de pruebas.



Las variables ordenadas por la importancia quedan de la siguiente forma:

- Alcohol
- Density
- Volatile acidity
- Free sulfur dioxide
- Citric acid
- Total sulfur dioxide
- Chlorides
- Residual sugar
- pH
- Fixed acidity
- Sulphates

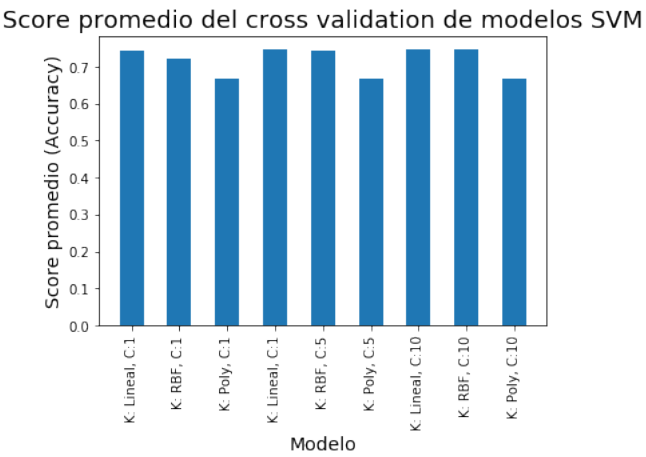
Para hacer reducción de dimensionalidad y quitar variables que puedan estar solamente causando ruido, se decide descartar las tres variables menos importantes (pH, Fixed acidity y Sulphates).

Modelo SVM

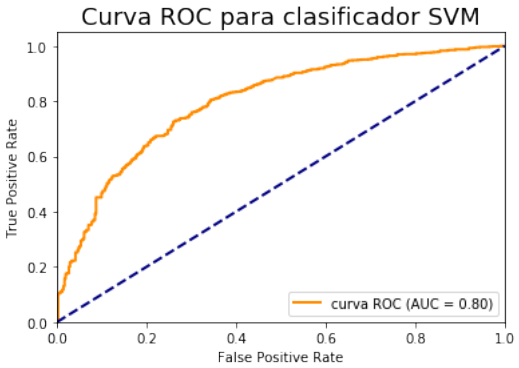
La metodología para seguir con este modelo es la siguiente:

- Escalar los datos de entrenamiento al rango 0 a 1 para que las escalas de las variables no afecten la importancia que se les debe dar.
- Escalar los datos de prueba con la misma transformación aplicada en el paso anterior para que no haya *data leaking* del set de pruebas.
- Hacer K-Fold cross validation para encontrar hiperparámetros óptimos (el tipo de kernel y el valor de C) para este problema.
- Entrenar el modelo con los hiperparámetros seleccionados.
- Correr el modelo sobre el set de pruebas y analizar el desempeño.

El cross validation se realizó con K = 5, se probaron modelos con kernels lineales, *radial basis function* y polinomiales contra diferentes valores de C, el parámetro de penalización de error. A continuación se presentan los promedios de *accuracy* que se obtuvieron con el cross validation de los diferentes modelos.



El mejor modelo de SVM resultó, por un márgen muy pequeño, tener Kernel RBF y C = 10, por lo que se entrenó con todo el set de pruebas y se ejecutó la clasificación sobre el set de pruebas, asignando las probabilidades de pertenecer a cada clase. Con esta información se calcula la curva ROC que se presenta a continuación.



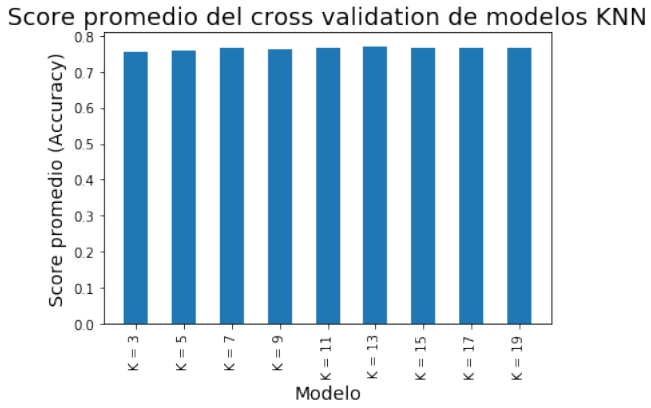
Se puede notar que el AUC de este modelo, el área bajo la curva ROC es de 0.8 lo cual indica que el desempeño no es malo.

Modelo KNN

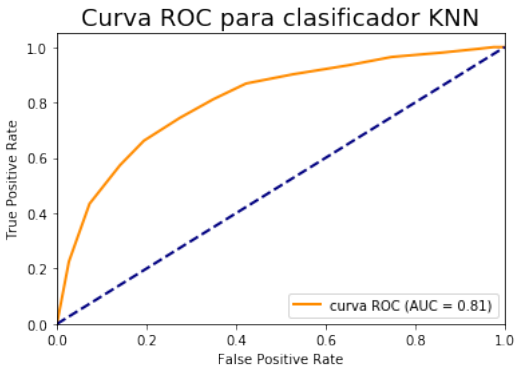
La metodología para seguir con este modelo es la siguiente:

- Escalar los datos de entrenamiento al rango 0 a 1 para que las escalas de las variables no afecten la importancia que se les debe dar.
- Escalar los datos de prueba con la misma transformación aplicada en el paso anterior para que no haya *data leaking* del set de pruebas.
- Hacer K-Fold cross validation para encontrar hiperparámetros óptimos (valor de K, número de vecinos más cercanos a considerar) para este problema.
- Entrenar el modelo con los hiperparámetros seleccionados.
- Correr el modelo sobre el set de pruebas y analizar el desempeño.

Con el modelo KNN se hizo cross validation para seleccionar el número de vecinos más cercanos a la observación a predecir que se deben considerar, un valor muy bajo puede no generalizar suficiente y un valor muy alto puede hacer que haya mucha varianza porque se pueden considerar vecinos que estén lejos en el espacio euclideano.



En la gráfica es difícil notar el modelo con mejor desempeño, pero resultó ser el de K = 13. La curva ROC se muestra a continuación.



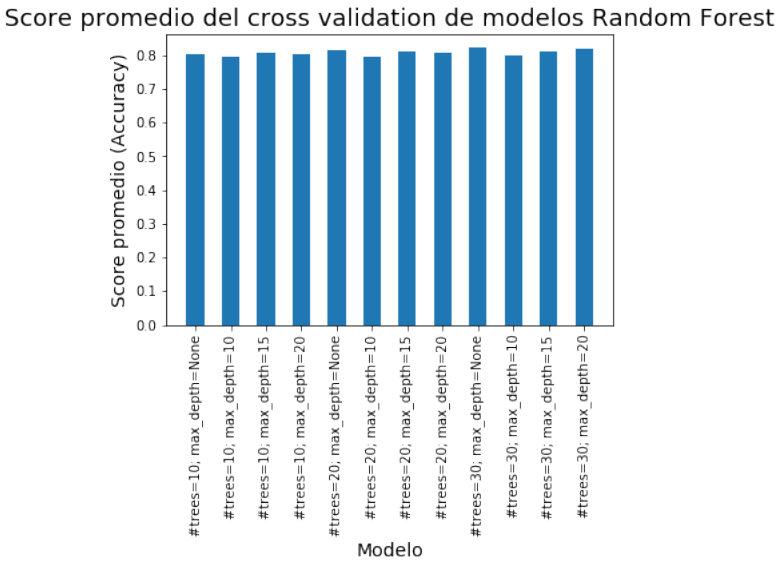
El modelo obtuvo un AUC de 0.81, ligeramente más alto que el modelo de SVM.

Modelo Random Forest

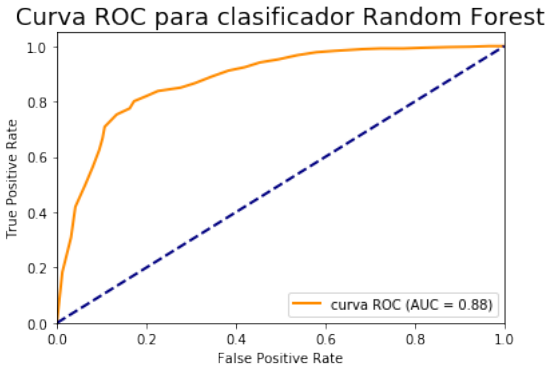
La metodología para seguir con este modelo es la siguiente:

- Hacer K-Fold cross validation para encontrar hiperparámetros óptimos (número de árboles y profundidad máxima) para este problema.
- Entrenar el modelo con los hiperparámetros seleccionados.
- Correr el modelo sobre el set de pruebas y analizar el desempeño.

El cross validation de los modelos Random Forest se hizo cambiando los valores del número de árboles que se utilizan en cada modelo así como la profunidad máxima del árbol. Entre menos profundo más generaliza cada árbol al no poder generar tantas reglas de decisión.

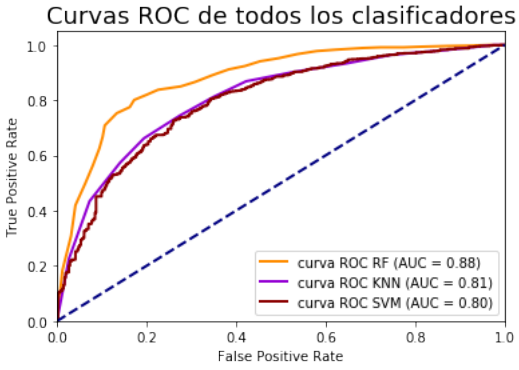


El mejor modelo Random Forest tiene 30 árboles y no tiene profundidad máxima. A continuación se muestra la curva ROC.



El AUC para este modelo es de 0.88, mucho mayor que los de los modelos anteriores.

Resultados



Es bastante evidente que el mejor modelo es el Random Forest, ahora si se impusiera una restricción como regla de negocio, por ejemplo que el FPR debe ser igual o menor a 0.3 pues es caro para un negocio tener falsos positivos, vender un vino malo con precio alto y que los clientes no estén satisfechos entonces se debe buscar un *threshold* en los *scores* de los modelos para que se satisfaga esta restricción.

A continuación se presenta la tabla con la información de los modelos en el punto de corte que satisface la restricción mencionada.

	AUC	**TPR**	**FPR**	**Threshold**
SVM	0.80	0.7515	0.2961	0.6528
KNN	0.81	0.7442	0.2743	0.6923
Random Forest	0.88	0.8499	0.2767	0.5667

Se puede ver que el modelo Random Forest es mucho mejor que los demás pues alcanza un TPR de 0.8499 al cumplir con la restricción.

Conclusiones

Este trabajo comparó exitósamente tres modelos diferentes de clasificación para el problema de clasificar si un vino es bueno o malo con base en su análisis de calidad químico. En este caso en particular el modelo ensamble de Random Forest obtuvo un desempeño mucho mayor que el de Máquinas de Soporte Vectorial así como el algoritmo de KNN o K Vecinos Cercanos.

Future Work

Queda mucho por investigar, algunas de las cosas que se pueden investigar son:

- Si modelos ensamble de SVM y KNN pueden tener mejor desempeño que un Random Forest para este problema.
- Qué otros modelos tienen mejor desempeño para este problema que un Random Forest

Bibliografía

Paulo Cortez, University of Minho, Guimarães, Portugal, http://www3.dsi.uminho.pt/pcortez A. Cerdeira, F. Almeida, T. Matos and J. Reis, Viticulture Commission of the Vinho Verde Region(CVRVV), Porto, Portugal @2009

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.