# PROYECTO DE MACHINE LEARNING - Recomendador de vinos - Memoria -

## INTRO - Presentando y entendiendo el problema…

**Voy a suponer que trabajo para una empresa que ofrece** servicios variados, aplicaciones móvil, etc… y ya me han encargado mi primer trabajo: quieren crear **un recomendador de vinos**, para ayudar y aconsejar al usuario a la hora de elegir un vino para un regalo, una cena en casa de amigos, una ocasión especial…

**La empresa sabe que** cada vez más desde la llegada del covid, va aumentando la costumbre de quedar en las casas para reuniones de amigos, sobre todo cuando aún hay cierto reparo en acudir en grupos muy numerosos a los bares. Y a estos eventos no se suele, o no se debería, acudir con las manos vacías..

La empresa sabe que no se puede generalizar y que una aplicación como ésta no va a tener éxito para todos los públicos, así que quiere centrarse en un mercado específico: gente jóven, pero no demasiado, por ejemplo a partir de los 25 años (los adolescentes no son el perfil que se busca).
También quiere hacer hincapié en la "usabilidad" de la aplicación, no tiene sentido recomendar vinos raros y/o únicos, si luego la persona no puede ir a comprarlos. Tiene que ser algo relativamente fácil de encontrar en el mercado y que no tenga precios exorbitados. Para ello ya haríamos otro modelo, enfocado a una élite más selecta, a coleccionistas por ejemplo, gente que no les importaría gastar más de 100$ en un vino.
Pero esta es otra historia….

**OBJETIVO: crear un recomendador de vinos** que nos indique la variedad a escoger, simplemente dándole unos parámetros de partida, para que la aplicación vaya eligiendo para nosotros y, porque no, nos descubra vinos que, de otra forma sería algo más complicado de encontrar. Está claro que, a pesar de que haya varia gente a la que no le importa recorrer bodegas y sitios especializados en búsqueda de un buen vino, la gran mayoría es más de sacar el móvil y que la aplicación haga el "trabajo sucio" para ellos. Me incluyo.

**¿Y cómo funciona?** Solo habría que insertar unos pocos parámetros como podría ser la región de proveniencia, el viñedo y/o la bodega, el año de producción, un precio alrededor del cual queramos movernos y a lo mejor también una puntuación para saber hasta qué calidad queremos llegar. Una vez hecho esto (podrían ser unas listas desplegables de las cuales el usuario elige los parámetros), es la aplicación la que se encarga de recomendarnos los vinos.

Por estas razones (más adelante en el proyecto lo explicaré) he acotado los datos, descartando los vinos demasiado "difíciles de encontrar"
→ que se haya producido por lo menos 100 de cada variedad
→ que de los viñedos de los que provienen, haya por lo menos 20
→ que haya por lo menos 3 bodegas que los producen

Esto hace que el modelo pueda recomendar vinos relativamente sencillos de encontrar en el mercado, y no que me vaya a recomendar un vino del que solo se produjo una o 2 botellas, no tiene sentido.

Pero ¿qué es lo que le pide específicamente la empresa a Gloria? **Crear un modelo de Machine Learning capaz de predecir el tipo de vino** (la variedad, por ejemplo Chardonnay o Pinot Noir), partiendo de unos parámetros dados.
Éste es su objetivo.
Luego ya se implementará el modelo en una aplicación para móviles y tablets y se lanzará al mercado, pero ésto ya será tarea del departamento de desarrollo de aplicaciones y del departamento de marketing.

## PASOS SEGUIDOS

Para mi proyecto, he podido encontrar en Kaggle un dataset de casi 130000 registros sobre vinos, con información tan variada como el País, provincia y región de proveniencia, unas descripciones hechas por varios sommelier que publicaron reseñas en una web de vinos, el nombre de estos sommelier, los viñedos de donde provienen las uvas, las bodegas de donde se hizo el vino y, por supuesto las variedades de uvas. También estaba la información sobre el precio de estos vinos y la puntuación dada por los varios sommeliers.

### *LIMPIEZA DE DATOS - FEATURE ENGINEERING*

Antes de empezar con los análisis varios, he hecho una búsqueda y limpieza exhaustiva de missing values, valores duplicados, columnas inútiles y outliers más cantosos (resulta que había uno en especial que era claramente un error de tipografía más que un valor de vino muy caro, lo borré directamente).
También he aplicado la técnica del feature engineering en la columna “title”, sacando de allí solo la información del año, que me serviría para mis análisis sucesivos, y prescindiendo del resto de información, al encontrarse repetida en otras columnas del dataframe.
También he detectado y analizado varios outliers, sobre todo en la columna de los precios, pero los he querido mantener durante todo el análisis, borrandolos solo una vez empezada la fase de Machine Learning.

### *EDA*

Una vez obtenido un dataframe limpio, lo he guardado como csv y he podido empezar finalmente con el EDA.
Redacto aquí abajo varias consideraciones que he podido sacar durante el análisis exploratorio.

#### PUNTUACIONES

- La mayoría de las puntuaciones se concentran en la mitad, (de las aprox 6000 valoraciones, más de 3000 han sido con valores de entre el 86 y el 93)
- Entre valoraciones más bajas (80-85) y más altas (94-100), ganan las más bajas, teniendo 122 valoraciones de "80" y solo 6 de "100"
- La puntuación más usada ha sido el "88" (6021 valoraciones), seguido por el "90" (5932)
- Hay que especificar que en general son valoraciones buenas, siendo el valor mínimo un 80 sobre 100. Valoraciones de usuarios de la página web "Wine Enthusiast" (expertos, no aficionados)

**PRECIOS/ PUNTUACIONES**

Hay una cierta relación, se vé como, a mayor precio, más alta es la puntuación. Aunque no es estricta al 100%, de hecho la curva sube gradualmente desde los precios relativos a la puntuación 80 hasta la 91, luego empieza a subir con más fuerza hasta la 97. De allí a la 99 hay subidas mucho más bruscas y ya para de subir, de hecho hay una ligera bajada en correspondencia de los vinos puntuados con 100, el valor máximo de puntos pero no el más alto en cuanto a precios.

En números:

- El vino más caro cuesta 850$ y tiene puntuación 99
- De los vinos puntuados con 100, el más caro cuesta “solo” 617$
- De las 5 puntuaciones más altas (de 96 a 100), el vino con precio más bajo vale 27$, que ya es un precio alto a mi parecer, y en este rango está también el vino más caro (850$)
- De las 5 puntuaciones más bajas (de 80 a 85), el vino de precio más bajo es de 4$ (coincide con el precio mínimo de todo el df), y él más caro no supera los 225$ (que aún siendo muy caro, es casi 4 veces menor que el vino más caro del df)
- En las valoraciones del centro (de 86 a 95), el vino más barato va desde los 6$ hasta los 20, mientras que de los más caros espacia entre 160$ y 800$

**PUNTUACIÓN/PRECIO POR PAÍS**

Quería ver si el País con los vinos más puntuados es también el País con los vinos más caros. Descubro que NO es así:

- US es el País con los vinos más puntuados (de media), aunque las diferencias son mínimas, estando los primeros 5 países en 89 y pico, la diferencia solo está en los decimales.
- Italia es el País con los vinos más caros (de media), aquí se ve algo más marcada la diferencia entre primer y segundo puesto, unos 4$ entre los dos valores medios.

En general, en lo que concierne las puntuaciones, está más nivelada la distribución por países, teniendo todos más o menos el mismo valor medio, mientras que hablando de precios se ve que hay algo más de diferencia entre los países.

Por ejemplo, Italia (el País con los vinos más caros) tiene una media de vinos de casi 47$, contra Argentina (el País con los vinos menos caros), donde la media está poco por encima de los 27$: hay una diferencia de bien 20$!

**PRODUCCIÓN DE VINO POR PAÍS**

Aquí la diferencia entre países se nota muchísimo: US es en absoluto el País que más produce, superando más de la mitad la producción de Francia, que está en la segunda posición y casi 4 veces la de Italia (aunque seguramente la extensión geográfica tenga mucho que ver al respecto).

Canadá, por otro lado, tiene unos valores casi insignificantes respecto al resto de Países ( en el gráfico titulado “PAÍSES PRODUCTORES DE VINO” se ve muy bien esta gran diferencia).

**PRECIOS POR BODEGAS**

De las más de 8000 bodegas, he reunido en 2 gráficos el top 10 de los vinos (de media) más caros y el top 10 de los más baratos.

Los precios se mueven en un rango de 625 a 320$ de media en las 10 bodegas más caras y entre 4 y 7$ en las 10 bodegas más baratas.
Ya se va notando la gran influencia de los outliers

**PUNTUACIÓN POR BODEGAS**

En el top 10 de las bodegas con vinos más puntuados tenemos valores medios muy altos, de 96 a 98.
Por otro lado, las 10 bodegas con puntuaciones más bajas están totalmente igualadas, con una media de 80 para cada una.

**PUNTUACIÓN/PRECIO POR CATADOR**

*Catadores que han puntuado los vinos más caros (primeras 3 posiciones, precios medios):*

- La catadora que puntúa más alto (93.0, por encima del 3º cuartil), ha puntuado los vinos que están en segunda posición en cuanto a los precios medios más altos (también por encima del 3º cuartil, 50$).
- La catadora que está en cuarta posición en cuanto a puntuación (89.58, más cercano a la mediana), es la que ha puntuado los vinos de media más caros (51.27, por encima del 3º cuartil).
- La catadora que está en tercera posición en cuanto a precios (puntuando con una media de 46.9$, que está entre la mediana y el tercer cuartil), en cuanto a puntuaciones está en el sexto puesto (puntuando con una media de 89.19, que corresponde a la mediana).

*Catadores que han puntuado los vinos más baratos (últimas 3 posiciones, precios medios):*

- Los 3 catadores de los vinos con precios más bajos (25.04, 24.74, 21.98) están en la escala de las puntuaciones, respectivamente en última posición(84.69), posición 11(87.60) y 13(87.22) (en torno al primer cuartil de las puntuaciones)

Aunque no sea una relación súper estricta, pero sí vemos una tendencia de que los que puntúan los vinos con precios más bajos también usan unas valoraciones más bajas, mientras que los que puntúan los vinos más caros, suelen dar valoraciones más altas

**RELACIÓN AÑO/PRECIO**

No es exactamente como me esperaba, que a más antiguo el vino, más precio, ya que los 4 años más antiguos tienen un rango de precio bajo respecto al resto (13$ para vinos del año 1919, por ejemplo. Y justo ese año tiene una puntuación de casi 88).
Pero sí es cierto que, quitando los primeros 4 años del dataframe, los años con los precios decisamente más altos (por encima de los 100$) van de 1945 a 1990 (quitando 1985, que ha sido un año "malo") y luego un repunte aislado en 1994, con un precio medio de 125$.

En números:

- Las puntuaciones fluctúan sin mucha lógica en la primera mitad de los años (aprox), hay picos muy altos y picos muy bajos, luego ya se van estabilizando y están alrededor de los 87/88 puntos.
- Luego de repente en 2017 (el último año de estudio) hay una bajada a 84.83
- Las medias más bajas de puntuaciones las tenemos en los años 1978 y 1989
- Las medias más altas de puntuaciones las tenemos en los años 1927 y 1945

#### UVAS/PRECIO

Las 5 variedades más caras (de media) son:

- Francisa
- Sherry
- Cabernet-Shiraz
- Marzemino
- Malbec Cabernet

#### UVAS/CANTIDAD

Las 5 variedades más producidas son:

- Pinots Noir
- Chardonnay
- Red Blend
- Cabernet Sauvignon
- Bordeaux style Red Blend

Si solo consideramos los valores medios, la más producida tendría que ser la Cabernet Sauvignon, pero en la realidad no es así, de hecho en esta clasificación está en cuarto lugar.

Este es otro caso clarísimo de la influencia de los outliers, que pesan mucho más en los valores medios que medianos.

#### MÁS CONSIDERACIONES

*Precio más alto y más bajo de la botella*

- Aunque Italia sea el País con los vinos más caros DE MEDIA, el vino más caro de este data frame es de Australia (580$), pasando Italia a una 6ª posición, con un vino de 800$.
- El vino más caro es un Shiraz del 2010, proveniente del Sur de Australia, del viñedo Grange, y producido en la bodega Penfolds. Lo ha valorado Jose Czerwinski, con 99 puntos.
- Está a la par con otro vino del Sur de Australia, del mismo viñedo, mismo catador, misma bodega y tipo de uva. Única diferencia: es del año 2008.
- DE MEDIA, los vinos más baratos son los de España y Argentina con el mismo precio, 4$, y luego en 5ª posición ya aparece USA, con 5$.

No hay 1 solo vino más barato, sino que son estos 4, que valen 4$ y son respectivamente:

- *proveniencia*: Spain (central Spain), Spain (central Spain), Spain (Levante), Argentina (Mendoza)
- *viñedo*: Flirty Bird, Flirty Bird, Estate Bottled, Red
- *tipo* de uva: Syrah, White Blend, Tempranillo, Malbec-Syrah
- *bodega*: Felix Solis, Felix Solis, Terrenal, Broke Ass
- *catador*: Michael Schachner (el mismo catador para los 4 vinos)

## MODELOS DE MACHINE LEARNING

Después de un análisis bastante exhaustivo de los datos, he podido pasar al problema de Machine Learning.

A lo largo de todo el proceso he tomado varios enfoques, puesto que no conseguía encontrar unos scores aceptables. Pasando de un problema de regresión a otro de

clasificación, y llegando a probar también un modelo de NLP, para ver si la columna de "reviews" podía ser útil en este camino.

*1º ENFOQUE - Problema de Regresión*
Al principio mi idea era la de predecir el precio de los vinos, no el tipo, de allí que en el primer notebook solo haya probado modelos de regresión.
He empezado transformando las columnas categóricas a numéricas y sucesivamente he borrado los outliers.
Desafortunadamente, de todos los modelos que probé, no conseguí sacar un score decente (el mejor RMSE encontrado no bajaba de 21 aprox y el mejor score no superaba el 44%).
Volví atrás varias veces, probando a quitar columnas, a transformar alguna, hasta intenté recurrir a las transformaciones logarítmicas, square root y box cox, pero sin obtener mejores resultados: pasé del peor RMSE con 30.03 al "mejor" con 21.42, pero seguía estando lejos de mis expectativas.
En el notebook 01 (*analisis_inicial.ipynb*) están todos los modelos probados, con sus respectivas métricas. Los listo aquí para que se puedan ver de un vistazo:

- 1º, 2º, 3º, 4º modelos de **REGRESIÓN LINEAL**, con varias pruebas de quitar/poner columnas.
  El mejor de estos resultados ha sido un score de 23.75 y un score de 0.32

- 5º modelo - **REGULARIZACIÓN (RIDGE)**
  - Score del modelo: 0.2855
  - RMSE: 24.4172

- 6º Y 7º modelos - **MODELOS DE RANDOM FOREST**
  El mejor de estos resultados ha sido un score de 22.21 y un score de 0.4

- 8º modelo - **SVM** (el peor de todos)
  He probado las 4 opciones de kernel y, resumiendo, el mejor de los resultados tiene un score de 0.23 y RMSE de 25.29

- 9º modelo - **HIST GRADIENT BOOSTING REGRESSOR**
  - Score del modelo: 0.43
  - RMSE: 21.62

- 10º modelo - **EXTRA TREE REGRESSOR** (parecido al random forest, pero algo peor)
  - Score del modelo: 0.36
  - RMSE: 22.94

- 11º modelo - **ADABOOST**
  - Score del modelo: 0.274
  - RMSE 24.6122

- 12º modelo - **GRADIENT BOOSTING**
  - Score del modelo: 0.4476
  - RMSE 21.4679

- 13º modelo - **XG BOOST**
  - Score del modelo: 0.4446
  - RMSE 21.5267

Resultado de este 1º enfoque: El mejor RMSE es RMSE 21.4679, relativo a gradient boost.
Viendo que no conseguía obtener mejores resultados, intenté acotar los precios, pensando que podría quedarme con los vinos de menos de 50$, y prescindir del resto.

Pasamos así al 2º enfoque del proyecto.

*2º ENFOQUE - Problema de Regresión, acotando más los datos*

Aún acotando precios, seguía teniendo un dataset muy amplio y, al fin y al cabo, para mi propósito de crear un recomendador de vino para mi entorno más cercano, tenía toda la lógica del mundo estar dentro de un rango "realista" de precios.

En el notebook 02 (*02_segundo_enfoque.ipynb*) están enumerados los dos modelos probados.

- 1º modelo - **GRADIENT BOOSTING**
  - Score del modelo: 0.38
  - RMSE 9.0058

- 2º modelo - **XG BOOST**
  - Score del modelo: 0.38
  - RMSE 9.0052

Llegado a este punto, veía que no conseguía subir el score (de hecho había bajado 6 puntos) y si bien el RMSE había bajado mucho, seguía siendo un error muy alto como para aprobar cualquiera de esos modelos. En general seguía sin convencerme esta diferencia entre score y accuray, así que empecé a mirar mi dataframe con otros ojos...

*3º ENFOQUE - Problema de Clasificación (el definitivo)*

Finalmente llegué a la conclusión de que este data frame se adaptaría mejor a un problema de **clasificación**, más que de regresión, así que decidí cambiar el target, pasando a predecir ya no el precio sino las variedades de vino.

Aquí estuve acotando aún más los datos, siguiendo fiel a mi propósito de crear un recomendador de vinos que se adaptara a un público cercano a mi entorno, más que a especialistas en vinos y/o coleccionistas o en general a gente más predispuesta a gastar una auténtica fortuna en una botella de vino y encima buscando entre las variedades más raras y selectas. Lo que a mí me interesa es que las personas a las que se dirige mi recomendador, puedan encontrar fácilmente el vino propuesto y, sobre todo, estando dentro de un rango de precios razonables.

Me quedé con aquellas variedades de las que se produzcan más de 100 botellas, con aquellos viñedos de los que haya más de 20 por variedad y bodegas de las que por lo menos hubiera 3 por tipo de uvas.

Mi dataframe final ya se había reducido considerablemente, llegando a poco más de 5000 registros (aún así, una cantidad más que decente para poder aplicarle un modelo de ML).

En el notebook 03 (*03_tercer_enfoque.ipynb*) están enumerados todos los modelos probados, con sus respectivas métricas.

1º MODELO - **REGRESIÓN LOGÍSTICA**
- Acierto: 56.69 %
- Error: 43.31 %

2º MODELO - **RANDOM FOREST CLASSIFIER**
- Acierto: 52.17 %
- Error: 47.83 %

3º MODELO - **SVC** --> el mejor resultado es con el kernel = sigmoid
- 52.26%

4º MODELO - **HIST GRADIENT BOOSTING CLASSIFIER** --> el PEOR
- Acierto: 12.11 %
- Error: 87.89 %

5º MODELO - **GRADIENT BOOSTING CLASSIFIE**R
- Acierto: 54.23 %
- Error: 45.77 %

6º MODELO - **XGBOOST CLASSIFIER**
- Acierto: 51.87 %
- Error: 48.13 %

7º MODELO - **ADABOOST CLASSIFIER** --> el segundo PEOR, después del HistGradientBoost
- Acierto: 19.59 %
- Error: 80.41 %

8º MODELO - **KNN**
- Acierto: 47.15 %
- Error: 52.85 %

9º MODELO - **NAIVE BAYES**
- Acierto: 43.7 %
- Error: 56.3 %

10º MODELO - **DECISION TREE**
- Acierto: 47.05 %
- Error: 52.95 %

Después de estas pruebas rápidas me quedé con los 3 mejores modelos y fui aplicando un pipeline y gridsearch.

Seguí acotando modelos y cambiando parámetros hasta quedarme finalmente con el **modelo definitivo**: una **regresión logística** que, como resultado, tiene un porcentaje de aciertos del **58%** aprox.
Antes de dar por finalizado mi proyecto, decidí probar una última cosa, que explicaré en el siguiente apartado.

*PRUEBA RÁPIDA NLP*
Quise hacer una prueba utilizando los conceptos de NLP vistos en clase: la idea era, partiendo de unas review, ver cómo de bien predecía mi modelo los tipos de vino.
En el notebook 04 (*04_test_prediccion_con_reviews.ipynb*) se pueden ver las pruebas que hice, atreviendome tambien a utilizar los embeddings, a tokenizar el texto y entrenar la red neuronal con capas convolucionales.
Los resultados no han sido tan malos (51% de accuracy), pero no han llegado a superar el mejor modelo que tenía hasta ahora, por lo tanto (sobre todo ya por la falta de tiempo para seguir investigando), he dejado de lado también este camino.

## CONCLUSIONES
Al principio estaba un poco mosqueada con el resultado, pero reflexionando mejor puedo sacar unos puntos positivos a mi proyecto:

- he partido de unos scores bajísimos cuando era un problema de regresión (un 0.44 de score y en el mejor de los casos el RMSE más bajo era de 21, o sea un error de 21$ sobre los precios, para nada bueno) y los he ido mejorando conforme iba cambiando el enfoque de mi proyecto, pasando finalmente a un problema de clasificación, intentando predecir el tipo de vino (la variedad) en vez que su precio.
- También he ido mejorando poco a poco los scores con el problema de clasificación, pasando de un 43% aprox (con alguna caída hasta el 12% y 19% de aciertos) hasta llegar finalmente a un casi 58% de aciertos (57,67% para ser exactos).
- Está claro que hasta el mejor sommelier, frente a una cata a ciega, no siempre (o prácticamente nunca) llega a un porcentaje de aciertos del 100%, así que me conformaré con mi resultado y puede que, más adelante, lo vaya revisando con más técnicas que vaya aprendiendo por el camino y, quizás, pueda obtener unos resultados mejores.

## ESTRUCTURA CARPETAS

1. src/: carpeta que incluye el proyecto entero
2. src/utils: funciones auxiliares creados para el desarrollo del proyecto
3. src/data/raw: aquí está el csv original, cogido de kaggle
4. src/data/processed: todos los csv que he ido creando y guardando a lo largo del proyecto
5. src/notebooks: notebooks de limpieza, procesado, EDA y modelos de Machine Learning
6. src/train.py: script con todos los pasos del proyecto, desde la limpieza inicial del dataframe hasta el entrenamiento del modelo (solo incluye el modelo principal, las varias pruebas están en los notebooks pero no aquí)
7. src/model: aquí he guardado solo el modelo con el mejor resultado

Nota sobre los varios notebook usados:
Los que hay que tener en cuenta para este proyecto son:

- *01_analisis_inicial.ipynb*
- *02_segundo_enfoque.ipynb*
- *03_tercer_enfoque.ipynb*
- *04_test_prediccion_con_reviews.ipynb*

Otros:

- *00_analisis_datos_borrador.ipynb* (primera toma de contacto, considero varias fuentes, veo qué datos tengo y, finalmente, me decanto por una solución, la que explico en este proyecto)
- *03_mas_tests_otro_pc.ipynb* (Para ir optimizando el tiempo, fui usando también otro pc que tengo en casa, éste es el notebook que he usado para ello

## FUENTES

kaggle.com
winemag.com
Google Slides (para la presentación)