

# PROYECTO FINAL

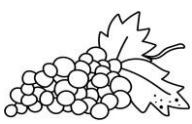
Data Science - Comisión 19155



Luciano Leguizamón  
Simon Escobar  
Oscar Parente

Entrega final: 31/8





# ÍNDICE

## Contenido

## Contenido

Data Science - Comisión 19155.....	1
Entrega final: 29/8.....	1
ÍNDICE.....	1
1. INTRODUCCIÓN.....	2
1.1. Presentación del equipo de trabajo .....	2
1.2. Temática .....	2
1.3. Problemática .....	2
1.4. Objetivo .....	2
1.5. Hipótesis.....	3
1.6. Alcance .....	3
1.7. Stack tecnológico .....	3
1.8. Tabla de versionado .....	3
2. BASE DE DATOS.....	4
2.1. Dataset.....	4
3. SECCIÓN COLABORATIVA.....	6
3.1. Data wrangling .....	6
3.2. Análisis Univariado.....	8
3.3. Análisis Bivariado .....	10
3.5. Algoritmos de entrenamiento .....	14
4. CONCLUSIÓN FINAL .....	20
5. Objetivos a futuro .....	20



# 1. INTRODUCCIÓN

## 1.1. Presentación del equipo de trabajo

Nosotros desde “*Happy Analytics & Colab*” presentamos el siguiente análisis de Proyecto.

Quienes suscriben:

- Luciano Leguizamón
- Simon Escobar
- Oscar Parente

## 1.2. Temática

Una empresa internacional de distribución de vinos con origen italiano llamada “VinTernazionali” presenta una disminución lenta pero sostenida de sus ventas desde hace 10 años.

Luego de realizar una auditoría interna, notaron que la causa de esta disminución es que el sistema de clasificación de vinos que daban a sus clientes no era adecuado, ya que solamente se basaba en la relación calidad-precio. Ese sistema había sido eficiente en el pasado, pero actualmente con la aplicación de nuevas tecnologías, cambios metodológicos en la industria y aparición de nuevas bodegas, la mayoría de los vinos no puede ser clasificado solamente utilizando estas dos variables.

Es por esa razón que actualmente la empresa está intentando aplicar un nuevo sistema de clasificación de sus vinos para retomar el vínculo con sus clientes.

La empresa nos provee un dataset ([https:// www.kaggle.com/ zynicide/ wine- reviews ?select=winemag -data -130k-v2.csv](https://www.kaggle.com/zynicide/wine-reviews?select=winemag-data-130k-v2.csv)) el cual contiene todas las reviews realizadas a los vinos que ellos comercializan.

La consultora “*Happy Analytics & Colab*” trabajará con el dataset provisto e intentará generar ese nuevo sistema de clasificación solicitado por la empresa.

## 1.3. Problemática

El cliente sostiene que la relación calidad precio ya no es un parámetro viable para la clasificación rápida de los vinos, y en consecuencia de eso, sus ventas disminuyen año a año.

## 1.4. Objetivo

Crear un nuevo sistema de clasificación rápida de vinos que pueda ser aplicado por el consumidor final al momento de la compra del producto. Es por ello que este sistema debe utilizar información visible en la etiqueta (bodega, año de cosecha, precio, etc) y no considerar parámetros físico-químicos (acidez, pH, etc) que, para el consumidor final promedio, no tienen ningún significado. Adicionalmente, este sistema debe ser capaz de clasificar nuevos vinos que ingresen al catálogo de la empresa.



## 1.5. Hipótesis

Las hipótesis que se plantean para el trabajo son las siguientes:

- La relación calidad-precio es un buen sistema de evaluación rápida de los vinos.
- Los vinos de alta calidad son los más caros.
- El precio no es la única variable que define si un vino es bueno, regular o malo.

## 1.6. Alcance

El proyecto tiene un alcance internacional, destinado a la empresa distribuidora “VinTernazionali”.

## 1.7. Stack tecnológico

En el proyecto serán utilizadas las siguientes herramientas:

1. Google Colab
2. Jupyter Notebook
3. Paquete Office

## 1.8. Tabla de versionado

ENTREGA	FECHA	VERSIÓN
Primer Entrega del Proyecto Final	21/06/2022	01.2022
Segunda Entrega del Proyecto Final	14/07/2022	02.2022
Tercera Entrega del Proyecto Final	02/08/2022	03.2022
Cuarta Entrega del Proyecto Final	31/08/2022	04.2022



## 2. BASE DE DATOS

### 2.1. Dataset

- **Número de Columnas:** 19
- **Número de Filas:** 129971

#### Modelado del Dataset:

- Se procedió a concatenar 4 columnas accesorias que estaban en .csv que fueron divididas por presentar una "," + Tabulación:
  - **Columna 1:** 2956
  - **Columna 2:** 159
  - **Columna 3:** 12
  - **Columna 4:** 1
- Se procedió a modelar el dataset para poder obtener el dato de género de los Sommeliers para poder usar el algoritmo KNN para problemas de clasificación:
  - Se validó que en la columna taste\_name no tuviera datos NaN, vacíos o corruptos.
  - Se agregó una nueva columna en la que se diferencia el género del Sommelier.
  - Tras concatenar el dato se procedió a confeccionar la nueva columna "winery".
  - Se validó que no hubiera datos numéricos en las columnas adyacentes a "points" y "price".
- Se regularizaron los nombres de las categorías de los vinos para que tuvieran la misma nomenclatura dado que variaban en base al idioma.
- Se procedió a mover un total de 1000 datos.
- Se usó Power Query, la opción de agregar una columna a través del ejemplo para poder traer el dato del año de producción del vino.
- Se validó que en la columna points no tuviera datos NaN, vacíos o corruptos.
- Se generó una Tabla Dinámica para validar el recuento de cada puntaje.
- Se leyó reseñas en las que se obtuvo la conclusión que los puntajes que califican en su mayoría a los buenos vinos eran aquellos que tenían un puntaje de 93 o más.
- En base al último punto se generó un método de clasificación:
  - Los vinos que tuvieran un puntaje de 93 o más se les otorgó el valor 1
  - Los vinos que tuvieran un puntaje de 92 o menos se les otorgó un puntaje de 2
- Se generó una nueva columna con este puntaje denominada: "aprobó"

#### Datos NA:

- **Country:** 43
  - Se completa los registros con el top 12 países
- **Price:** 8996
  - =SI([@price]="";ALEATORIO.ENTRE(15;50);[@price])



- **Taster\_name:** 26284
  - Se completó con el top 15 tester.
- **Year:** hay 4415 que se generó una fórmula aleatoria.
  - Se procedió a limpiar el dato en las celdas corruptas.
  - Se aplica la fórmula =SI(@year="";ALEATORIO.ENTRE(2006;2014);@year), el dato de los años se sacó a través de una Tabla dinámica para poder saber los años con mayor número de puntaje.

## **Descripción de los campos:**

- **Country:** país de origen.
- **Description:** breve descripción del vino.
- **Designation:** viñedo de donde proviene el vino.
- **Points:** puntaje otorgado por el sommelier.
- **Price:** precio internacional del vino.
- **Province:** provincia de donde proviene el vino.
- **Region\_1:** provincia o estado de donde proviene el vino.
- **Region\_2:** región más específica de donde proviene el vino.
- **Taster\_name:** nombre del sommelier.
- **Taster\_twitter\_handle:** cuenta de twitter del sommelier.
- **Title:** etiqueta del vino.
- **Variety:** varietales.
- **Winery:** bodega.
- **Year:** año de producción.
- **Continent:** continente al que pertenece el país de origen del vino.
- **Gender:** género del sommelier que realizó el testing del vino.
- **Aprobó:** en base al puntaje que obtuvo el vino se divide en 1 (aprobó>92 puntos) y 2 (desaprobó<92 puntos).
- **Categoría de calidad:** En base al puntaje, se procedió a clasificar el vino en 3 categorías: A (>=95 puntos), B (90-94,9 puntos), C (<89,9 puntos).



## 3. SECCIÓN COLABORATIVA

### 3.1. Data wrangling

Una vez importado el dataset, se creó el dataframe “df\_wine” y se obtuvo la siguiente información:

```
[ ] df_wine.count()
```

```
country          129971
description       129971
designation       92505
points           129971
price            129971
province         129905
region_1         108722
region_2         50510
taster_name      129971
taster_twitter_handle 121708
title            129968
variety          129967
winery           129968
year             129971
continent        129971
gender           129971
aprobo           129971
categoria_calidad 129971
dtype: int64
```

- El comando “.count()” cuenta la cantidad de datos que hay en una columna.

```
[ ] df_wine.isna().sum()
```

```
country          0
description       0
designation       37466
points           0
price            0
province         66
region_1         21249
region_2         79461
taster_name      0
taster_twitter_handle 8263
title            3
variety          4
winery           3
year             0
continent        0
gender           0
aprobo           0
categoria_calidad 0
dtype: int64
```

- Mientras que “.isna().sum()” nos dice la cantidad de datos nulos que hay en una columna.





```
df_wine.info()
```



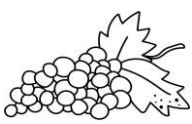
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129971 entries, 0 to 129970
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   country                129971 non-null object
1   description             129971 non-null object
2   designation             92505 non-null  object
3   points                  129971 non-null int64
4   price                   129971 non-null int64
5   province                129905 non-null object
6   region_1                108722 non-null object
7   region_2                50510 non-null  object
8   taster_name             129971 non-null object
9   taster_twitter_handle   121708 non-null object
10  title                   129968 non-null object
11  variety                 129967 non-null object
12  winery                  129968 non-null object
13  year                    129971 non-null int64
14  continent                129971 non-null object
15  gender                  129971 non-null object
16  aprobo                  129971 non-null int64
17  categoria_calidad        129971 non-null object
dtypes: int64(4), object(14)
memory usage: 17.8+ MB
```

```
[ ] df_wine.describe().round(2)
```

	points	price	year	aprobo
count	129971.00	129971.00	129971.00	129971.0
mean	88.45	33.29	2010.64	1.9
std	3.04	24.28	4.03	0.3
min	80.00	4.00	1821.00	1.0
25%	86.00	22.00	2009.00	2.0
50%	88.00	31.00	2011.00	2.0
75%	91.00	42.00	2013.00	2.0
max	100.00	2500.00	2017.00	2.0

- Con “.info()” podemos observar un resumen de las características principales (cantidad de datos nulos y tipos de datos) del dataframe.

- Con “.describe” se puede ver el comportamiento estadístico de las variables numéricas del dataframe.



## 3.2. Análisis Univariado

Consiste en el análisis de cada una de las variables por separado, por lo que se basa exclusivamente en una única variable.

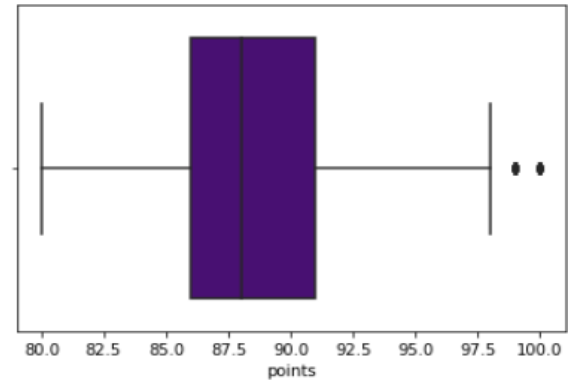
### Tipo de gráfico

### Descripción

### Imagen

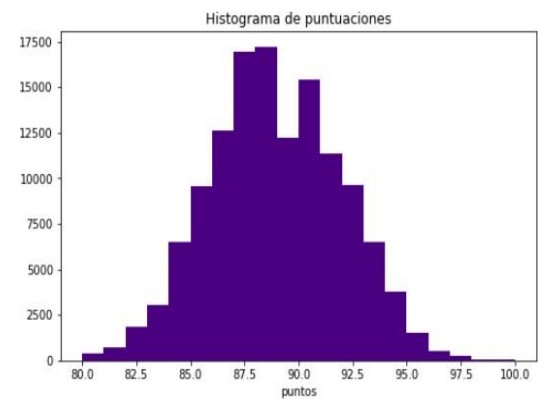
Boxplot

En este boxplot observamos que el puntaje se concentra mayoritariamente en un rango entre 86 y 91 puntos.



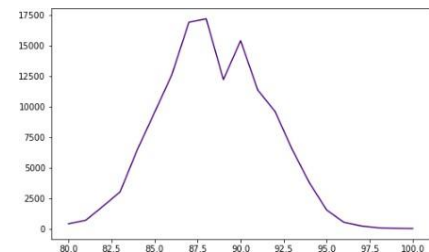
Histograma

Este histograma nos confirma lo mostrado en el boxplot.



Histograma 2

Otra forma de ver la distribución de los datos.



Histograma de años

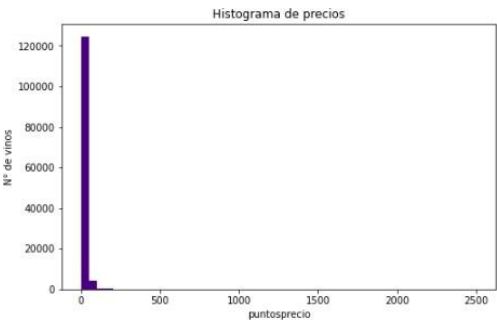
Nos muestra que la gran mayoría de los años de cosecha es posterior al año 2000





Histograma de precios

Observamos que la gran mayoría de los precios tienen un valor bajo.



Boxplot precios

En este boxplot observamos que el precio tiene una variabilidad muy grande. El grueso de los precios son bajos, pero hay algunos que superan los 500 dólares.

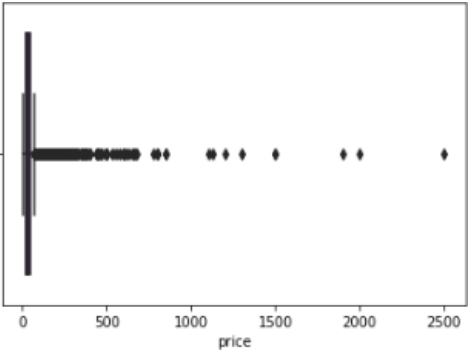
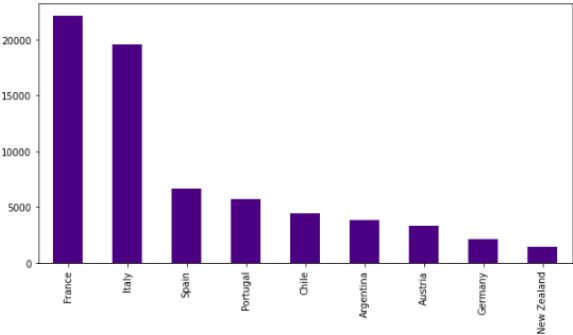
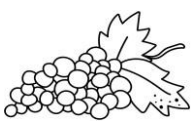


Gráfico de barras

Aquí observamos el país de origen de los vinos que tienen más reviews.





### 3.3. Análisis Bivariado

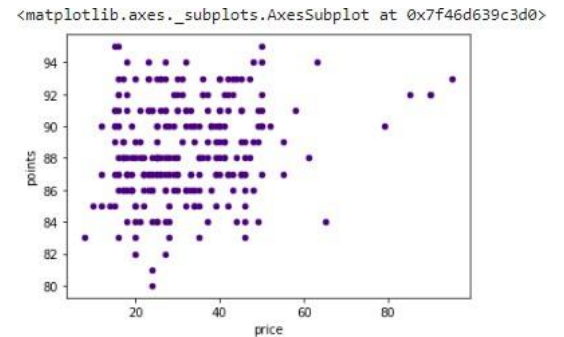
El análisis bivariado es el empleo de dos o más variables , buscando conocer causalidad, efectos o correlaciones.

Tipo de gráfico	Descripción	Imágen
Violin (precios y puntajes)	Comparamos los precios y puntajes de los vinos de los países de américa del sur, se puede distinguir que la variabilidad del puntaje de los vinos argentinos es mayor y que los vinos de Perú tienen un puntaje máximo de 87.5	
Violin (precios)	En lo que respecta a los precios, los vinos de Argentina, Chile y Uruguay tienen un comportamiento bimodal, los precios de Argentina y Chile cuentan con outliers, sin embargo, la media de los precios es similar en los cinco países.	
Scatter plot (puntos y precios < 100 dólares)	La elevada cantidad de puntos no nos permite distinguir ningún rasgo	



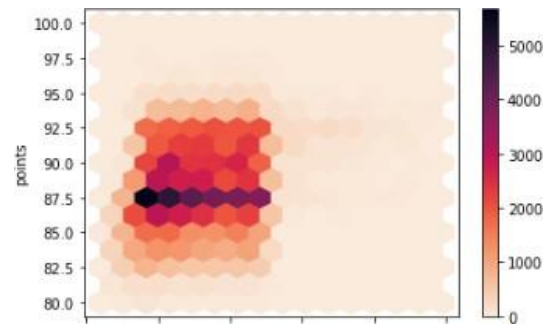
Scatter plot  
(puntos y precios < 100  
dólares)

Haciendo el mismo gráfico pero con una muestra de 250 datos, se observa que en el extremo. Inferior izquierdo hay mayor densidad de datos, también se ve una leve correlación directa.



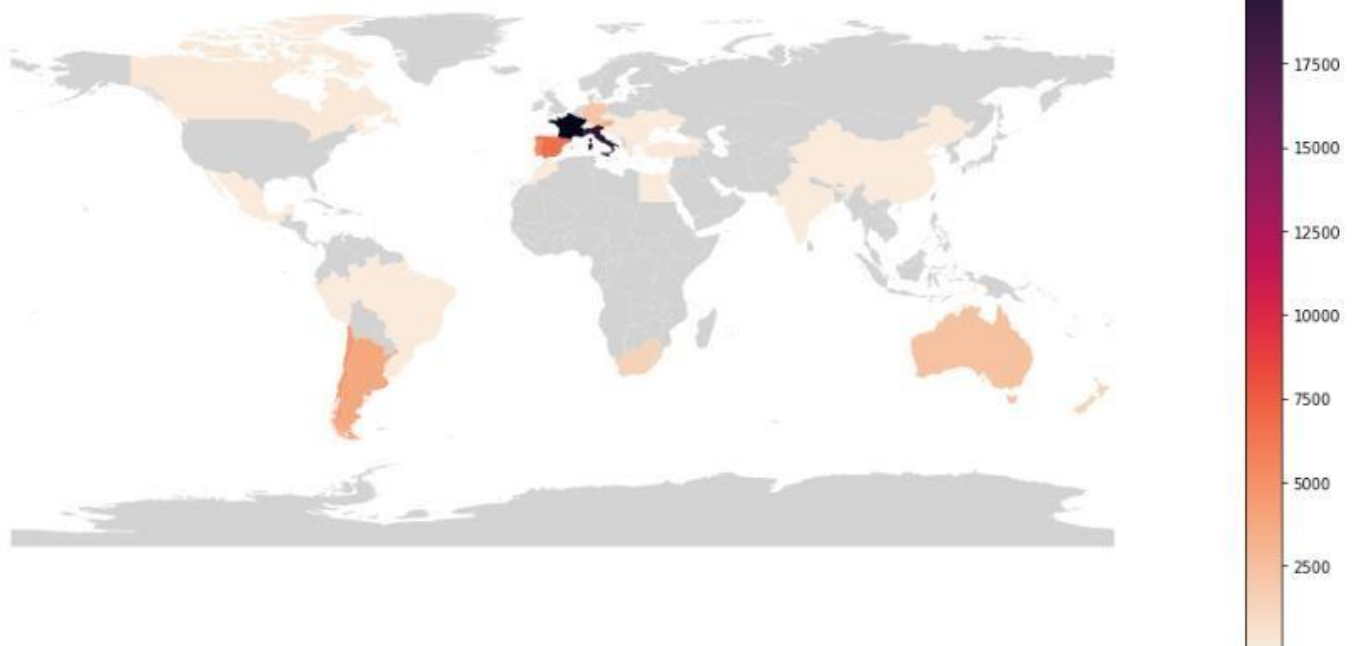
Hexbin

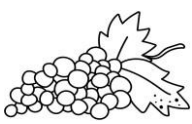
Con un gráfico de dispersión hexagonal (hexbin) es posible notar que la gran mayoría de los vinos tiene puntaje cercanos a 87.5 y un precio que ronda los 20 a 40 dólares.



Generamos un mapa en donde vemos el origen de los vinos que tienen reviews.

Cantidad de vinos analizados por país





### 3.4. Análisis Multivariado

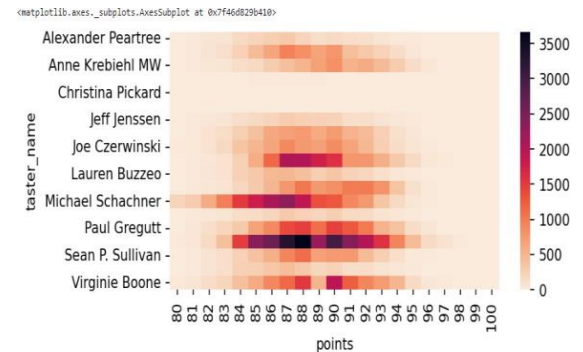
#### Tipo de gráfico

#### Descripción

#### Imagen

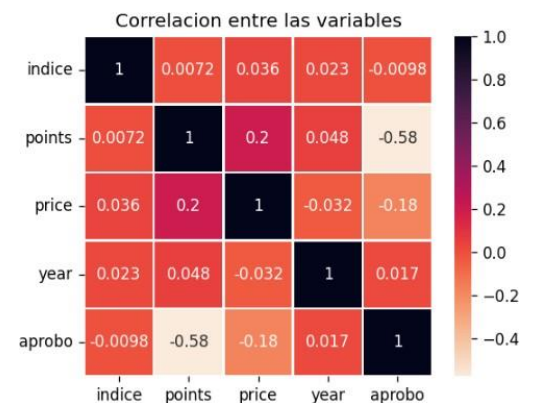
Heatmap utilizando  
crosstab

En este crosstab podemos observar los nombres de los catadores, el puntaje que otorgaron y que tan frecuentemente otorgan ese puntaje.



Matriz de correlación

Generando una matriz de correlación se observa claramente que los valores tienen una baja de adecuación entre sí.

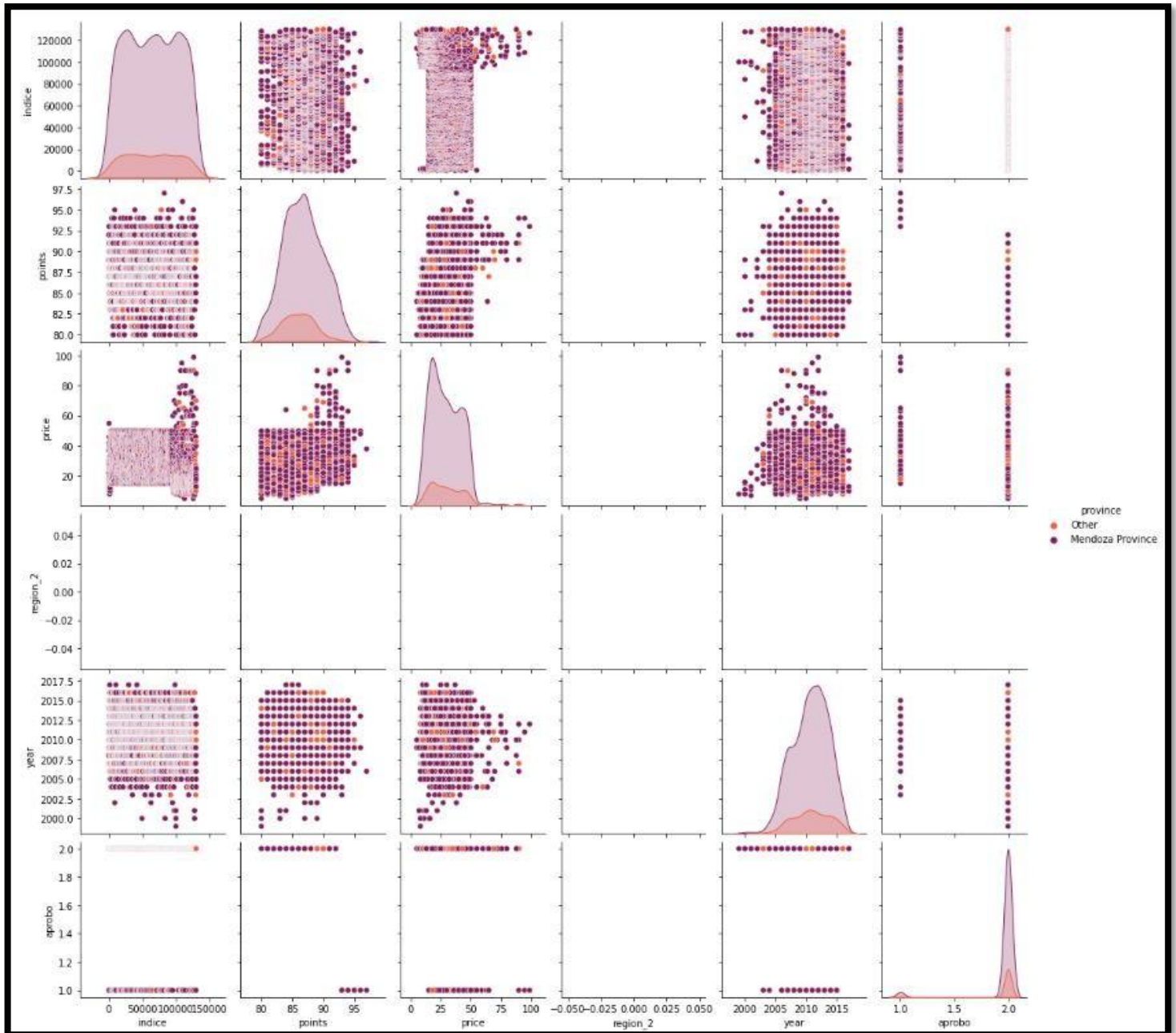
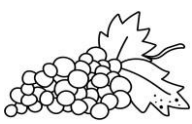


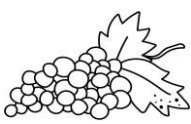
Pairplot

Se crea el dataset “df\_wine\_100usd\_arg”, el cual contiene los vinos con precios más bajos a 100 dólares y provenientes de Argentina. Con ese dataset creamos un pairplot y diferenciamos por provincia de origen, en donde se observa que las variables no tienen una correlación entre sí.

Ver gráfico debajo







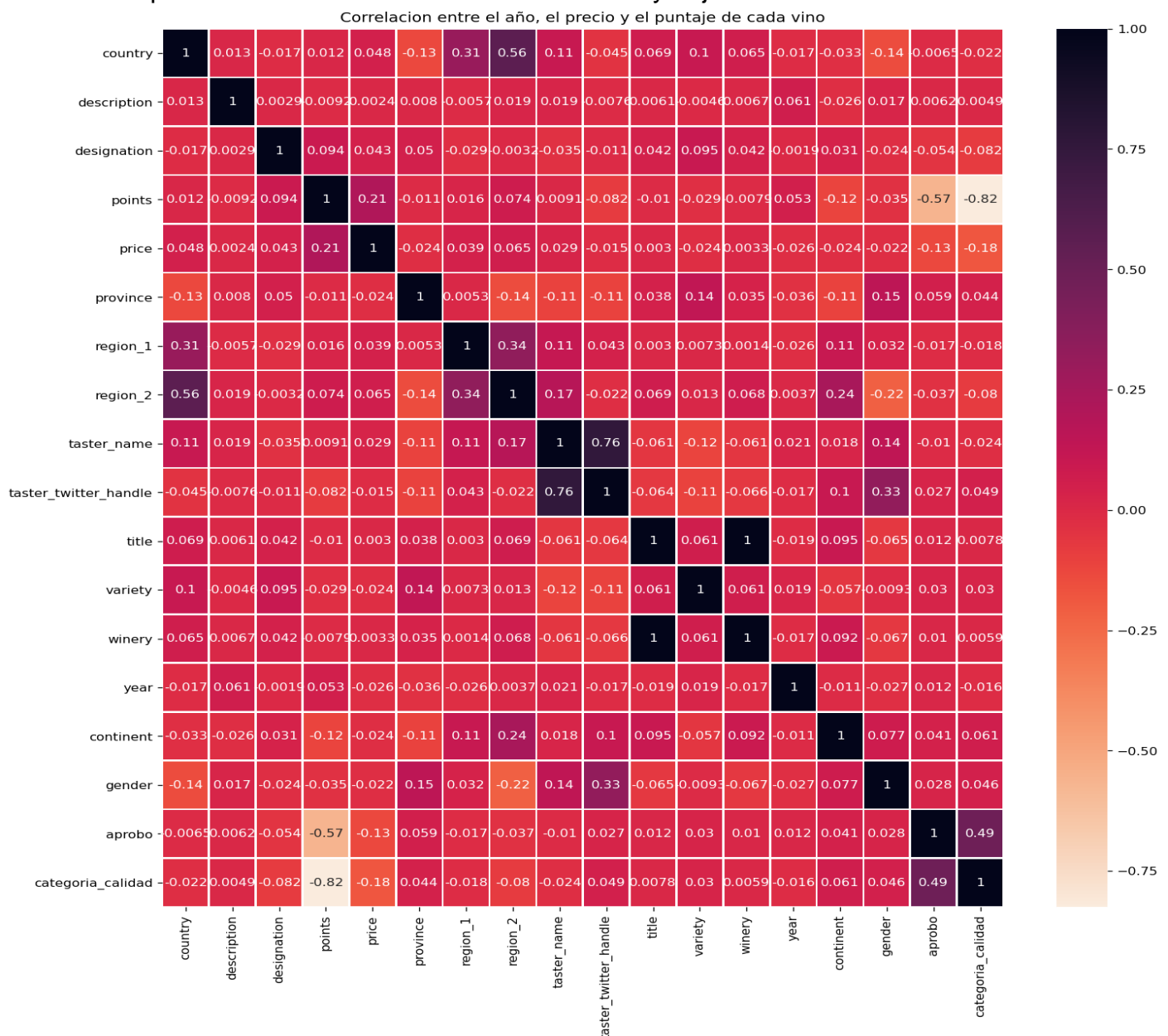
## 3.5. Algoritmos de entrenamiento

Elegimos algoritmos de entrenamiento y preparamos los datos para el proceso de entrenamiento del modelo. También evaluamos los indicadores de desempeño predictivo del modelo y realizamos optimizaciones.

Transformamos todo en variables categóricas para poder entrenar de mejor manera el modelo.

	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	title	variety	winery	year	continent	gender	aprobo	categoria_calidad	
0	22	16239	36529	87	39	331	423	-1	9		4	78794	688	11588	2013	2	1	2	2
1	31	99092	2305	87	15	108	-1	-1	15		10	88539	449	12933	2011	2	1	2	2
2	40	74779	-1	87	14	268	1216	16	14		7	88952	435	12995	2013	3	1	2	2
3	40	59933	27750	87	13	218	548	-1	0		-1	99963	477	14365	2013	3	1	2	2
4	40	55174	36190	87	65	268	1216	16	14		7	101876	439	14596	2012	3	1	2	2

Se observa que la correlación entre las variables es muy baja







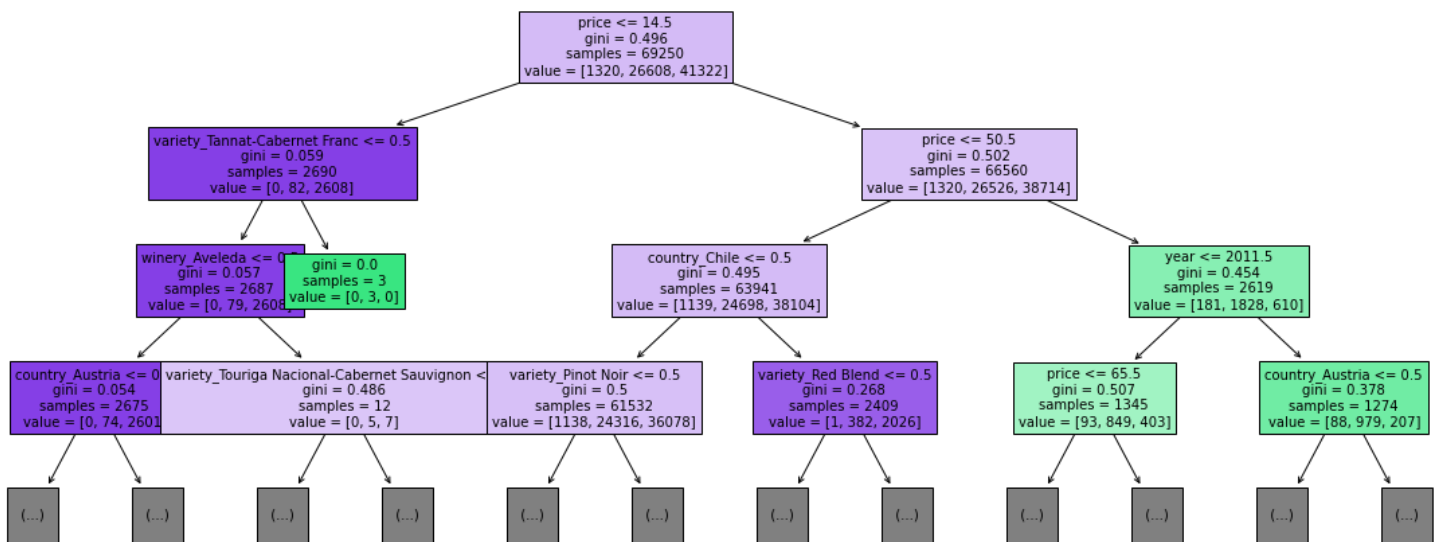
## 3.5.1. Algoritmos de clasificación

En base a los análisis de variables realizados previamente, quitamos del dataset las columnas irrelevantes que no aportan información al modelo. Las columnas que eliminamos son: indice, region\_1, region\_2, taster\_name, taster\_twitter\_handle, description, title, gender.

### 3.5.1.1. Decision Tree

Los árboles de decisión son representaciones gráficas de posibles soluciones a una decisión basadas en ciertas condiciones, es uno de los algoritmos de aprendizaje supervisado más utilizados en machine learning y pueden realizar tareas de clasificación o regresión. Comienza con un único nodo y luego se ramifica en resultados posibles. Cada uno de esos resultados crea nodos adicionales, que se ramifican en otras posibilidades. Esto le da una forma similar a la de un árbol.

Representación gráfica de nuestro árbol de decisión:



## Resultados y optimización de Decision Tree:

Se corrió el algoritmo con las siguientes especificaciones

```
from sklearn.tree import DecisionTreeClassifier
#Creamos el modelo
model_DeTree = DecisionTreeClassifier(max_depth=20, random_state = 47)
```

Obteniendo como resultado

```
% de aciertos en set de entrenamiento: 0.82107
% de aciertos en set de evaluación: 0.65233
```

Posteriormente se realizó la optimización, y los mejores parámetros resultaron ser:

```
Mejores parametros:
{'criterion': 'entropy', 'max_depth': 13}
```



Se corrió el algoritmo con esos parámetros, y se obtuvo como resultado:

% de aciertos en set de entrenamiento: 0.69893  
% de aciertos en set de evaluación: 0.67515

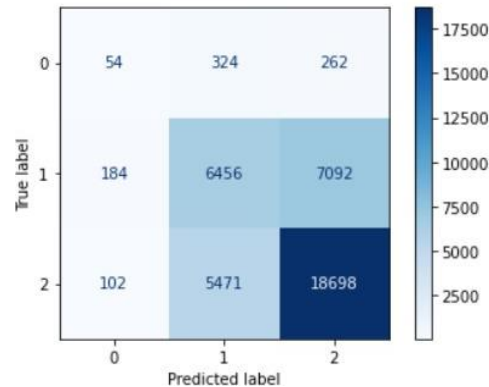
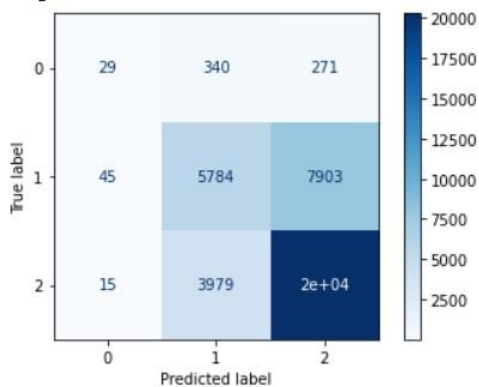
Importancia de los predictores  
en el modelo Decision Tree

	predictor	importancia	
4	winery	0.329235	
1	price	0.229106	
5	year	0.151084	
3	variety	0.144734	
2	province	0.090608	
0	country	0.031767	
6	continent	0.023467	

```
from sklearn.metrics import classification_report
print(classification_report(Yd_test, Yd_test_pred))
```

	precision	recall	f1-score	support
0	0.16	0.08	0.11	640
1	0.53	0.47	0.50	13732
2	0.72	0.77	0.74	24271
accuracy			0.65	38643
macro avg	0.47	0.44	0.45	38643
weighted avg	0.64	0.65	0.65	38643

La matriz de confusión permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Realizamos las matrices de confusión del mismo algoritmo, optimizado y sin optimizar.



**Conclusión:** En la matriz de confusión se puede observar que el algoritmo clasifica de mejor manera los vinos de peor calidad.



## 3.5.1.2. Random Forest

Es una combinación de árboles tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos

- Dividimos nuestra serie de datos en varios subconjuntos compuestos aleatoriamente de muestras, de ahí el “random” de random forest.
- Se entrena un modelo en cada subconjunto: habrá tantos modelos como subconjuntos.
- Se combinan todos los resultados de los modelos (con un sistema de voto, por ejemplo) lo que nos da un resultado final.

```
from sklearn.ensemble import RandomForestClassifier
#creamos el modelo
model_RaFo = RandomForestClassifier(n_estimators = 20 , max_depth = 5)
```

```
% de aciertos en set de entrenamiento: 0.63918
% de aciertos sobre el set de evaluación: 0.64206
```

```
RaFo_Grid.best_params_
```

```
{'criterion': 'gini', 'max_depth': 8, 'n_estimators': 4}
```

```
% de aciertos en set de entrenamiento: 0.65294
% de aciertos en set de evaluación: 0.65342
```

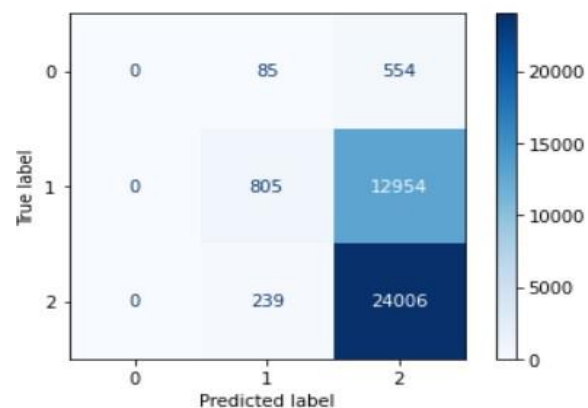
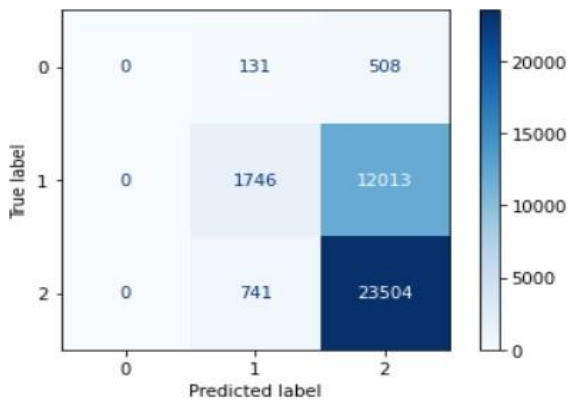
Importancia de los predictores  
en el modelo de Random Forest

-----			
	predictor	importancia	
1	price	0.405862	
6	continent	0.176682	
3	variety	0.121887	
2	province	0.120398	
0	country	0.084190	
5	year	0.083026	
4	winery	0.007957	



```
from sklearn.metrics import classification_report  
print(classification_report(Yr_test, Yr_test_pred))
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	639
1	0.68	0.10	0.18	13759
2	0.65	0.98	0.78	24245
accuracy			0.65	38643
macro avg	0.44	0.36	0.32	38643
weighted avg	0.65	0.65	0.55	38643



**Conclusión:** En la matriz de confusión se puede observar que el algoritmo clasifica de mejor manera los vinos de peor calidad.

### 3.5.1.3. KNN

K-vecinos más cercanos (KNN) es un tipo de algoritmo de aprendizaje supervisado. KNN intenta predecir la clase correcta para los datos de prueba calculando la distancia entre los datos de prueba y todos los puntos de entrenamiento. Luego selecciona el número K de puntos que está más cerca de los datos de prueba.

El algoritmo KNN calcula la probabilidad de que los datos de prueba pertenezcan a las clases de datos de entrenamiento 'K' y se selecciona la clase que contiene la probabilidad más alta.

Supongamos que tenemos una imagen de una criatura que se parece a un gato y un perro, pero queremos saber si es un gato o un perro. Nuestro modelo KNN encontrará las características similares del nuevo conjunto de datos a las imágenes de perros y gatos y, en función de las características más similares, las colocará en la categoría de perros o gatos.

```
from sklearn.neighbors import KNeighborsClassifier  
model_KNN = KNeighborsClassifier (n_neighbors= 5, weights = 'uniform', leaf_size= 10)
```



% de aciertos en set de entrenamiento: 0.65294  
% de aciertos en set de evaluación: 0.64366

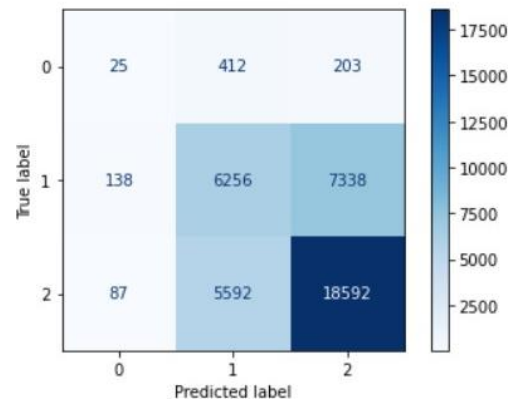
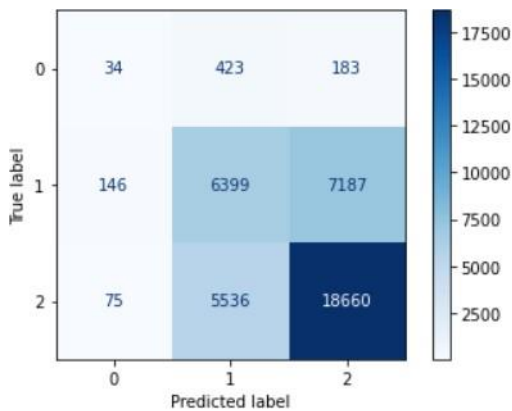
```
KNN_Grid.best_params_
```

```
{'metric': 'manhattan', 'n_neighbors': 5, 'weights': 'uniform'}
```

% de aciertos en set de entrenamiento optimizado: 0.65294  
% de aciertos en set de evaluación optimizado: 0.64935

```
from sklearn.metrics import classification_report  
print(classification_report(Yk_test, Yk_test_pred))
```

	precision	recall	f1-score	support
0	0.18	0.10	0.13	640
1	0.55	0.52	0.53	13732
2	0.74	0.77	0.76	24271
accuracy			0.67	38643
macro avg	0.49	0.46	0.47	38643
weighted avg	0.66	0.67	0.67	38643

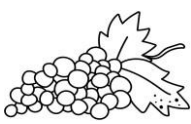


**Conclusión:** En la matriz de confusión se puede observar que el algoritmo clasifica de mejor manera los vinos de peor calidad.

### 3.5.1.4. Conclusión de Algoritmos de Clasificación

Podemos ver cómo han mejorado los modelos con respecto al utilizado en el comienzo del preprocesado.

Si bien no hay tanta diferencia en el resultado entre los tres modelos utilizados (KNN, Decision Tree, Random Forest), podemos indicar que, el que arrojó mejor resultado fue el Decision Tree con Grid Search, que presente un RMSE bajo y, además no presenta Over o Under Fiting.



```
PARAMETROS DEL ALGORITMO DECISION TREE
% de aciertos en set de entrenamiento optimizado: 69.89 %
% de aciertos en set de evaluación optimizado: 67.52 %
Raíz del error cuadrático medio en Train: 0.5678
Raíz del error cuadrático medio en Test: 0.5891
```

```
PARAMETROS DEL ALGORITMO RANDOM FOREST
% de aciertos en set de entrenamiento optimizado: 65.02 %
% de aciertos en set de evaluación optimizado: 64.96 %
Raíz del error cuadrático medio en Train: 0.6264
Raíz del error cuadrático medio en Test: 0.6266
```

```
PARAMETROS DEL ALGORITMO KNN
% de aciertos en set de entrenamiento optimizado: 77.77 %
% de aciertos en set de evaluación optimizado: 67.92 %
Raíz del error cuadrático medio en Train: 0.4841
Raíz del error cuadrático medio en Test: 0.5809
```

### 3.5.1.5. Algoritmo adecuado

En un principio, nuestro enfoque fue realizar tres modelos diferentes esperando obtener resultados distintos entre los mismos. Una vez entrenados y optimizados, notamos que no hay tanta diferencia en los resultados de cada uno de ellos.

## 4.CONCLUSIÓN FINAL

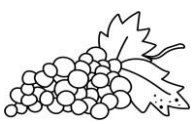
“Happy Analytics & Colab”, defendemos la idea de que no hay relación entre la calidad y el precio en un vino. Ya que, existen vinos muy buenos a muy buen precio y no tienen por qué costar más de 20 dólares para ser considerados excelentes en la relación calidad-precio.

Como es el caso de:

country	point	price	province	variety	winery	year	continent	gender	aprobado	categoría
US	98	16	California	Pinot Noir	Wayfarer	2014	North America	F	1	A
Italy	100	20	Tuscany	Prugnolo Gentile	Avignonesi	1995	Europe	F	1	A
France	100	17	Champagne	Champagne Blend	Krug	2002	Europe	M	1	A
US	98	18	California	Cabernet Sauvignon	Yao Ming	2010	North America	M	1	A
Italy	100	19	Tuscany	Sangiovese	Biondi Santi	2010	Europe	M	1	A
US	98	19	California	Pinot Noir	Williams Selyem	2010	North America	F	1	A
Australia	98	20	South Australia	Shiraz	Penfolds	2008	Oceania	M	1	A
Italy	99	17	Tuscany	Merlot	Tenuta dell'Ornellaia	2004	Europe	M	1	A
France	100	15	Champagne	Champagne Blend	Louis Roederer	2008	Europe	M	1	A
Austria	98	17	Burgenland	Welschriesling	Kracher	2008	Europe	M	1	A
France	98	20	Bordeaux	Bordeaux-style Red Blend	Château Haut-Brion	2010	Europe	M	1	A
France	98	15	Bordeaux	Bordeaux-style Red Blend	Château Mouton Rothschild	2010	Europe	M	1	A

## 5.Objetivos a futuro

Estamos trabajando en la implementación de un dashboard interactivo que permita que la alta gerencia trabaje con los datos fuera de la notebook. Por el momento está modelado para países Sudamericanos



## Dashboard proyecto Coderhouse

### Detalle de los vinos de sudamérica

El propósito de este dashboard es mostrar de forma más amigable algunos datos de vinos sudamericanos y demostrar que es posible mejorar las notebook para hacerlas más interactivas



#### Selector de puntaje

Puntos: 89

