Wine Predictor

Using Machine learning model

Aman Saini

5/1/20

Data Science

PROBLEM STATEMENT :

Context

- Imagine you are working as a Data Scientist for an Online Wine Shop named “The Wine Land”
- As the name suggests, the online store specializes in selling different varieties of wines.
- The online store receives a decent amount of traffic and reviews from its users.
- Leverage the “reviews” data and draw actionable insights from it.

Things to do :

- Build a predictive model for predicting the wine “variety”.
- A short summary: Model used, features extracted, Model accuracy in train.
  Along with some visualization of data and top 5 actionable Insights from the Data. .

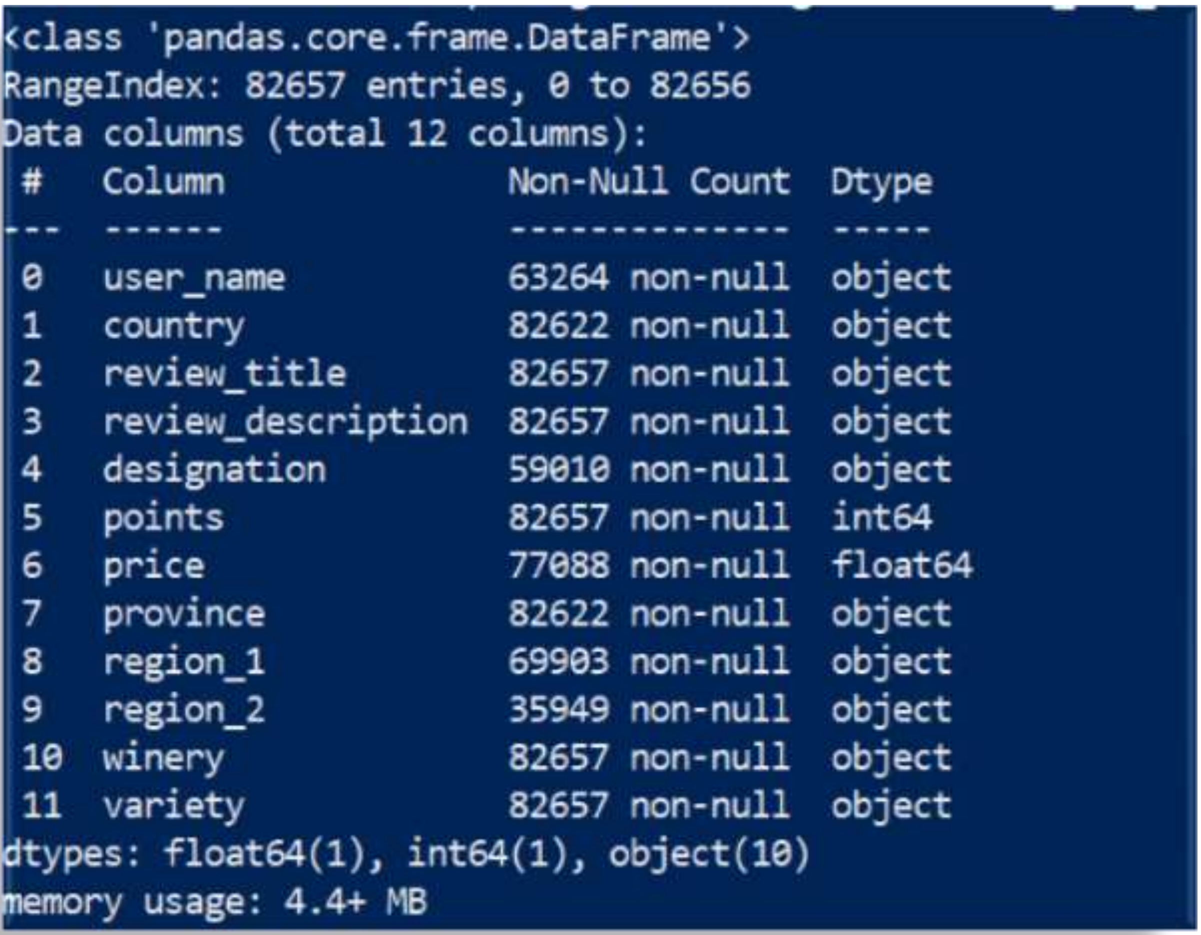
The Data Description is as follows:

- user_name - user_name of the reviewer
- country -The country that the wine is from.
- review_title - The title of the wine review, which often contains the vintage.
- review_description - A verbose review of the wine.
- designation - The vineyard within the winery where the grapes that made the wine are from.
- points - ratings given by the user. The ratings are between 0 -100.
- price - The cost for a bottle of the wine
- province - The province or state that the wine is from.
- region_1 - The wine-growing area in a province or state (ie Napa).
- region_2 - Sometimes there are more specific regions specified within a wine-growing area (ie Rutherford inside the Napa Valley), but this value can sometimes be blank.
- winery - The winery that made the wine
- variety - The type of grapes used to make the wine. Dependent variable for task 2 of the assignment

EDA :

Getting to know the data in hand

An overview of data types involved and the number of data samples usually helps us to let us know what we are dealing with :



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 82657 entries, 0 to 82656
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   user_name           63264 non-null  object
 1   country             82622 non-null  object
 2   review_title        82657 non-null  object
 3   review_description  82657 non-null  object
 4   designation         59010 non-null  object
 5   points              82657 non-null  int64
 6   price               77088 non-null  float64
 7   province            82622 non-null  object
 8   region_1            69903 non-null  object
 9   region_2            35949 non-null  object
 10  winery              82657 non-null  object
 11  variety             82657 non-null  object
dtypes: float64(1), int64(1), object(10)
memory usage: 4.4+ MB
```

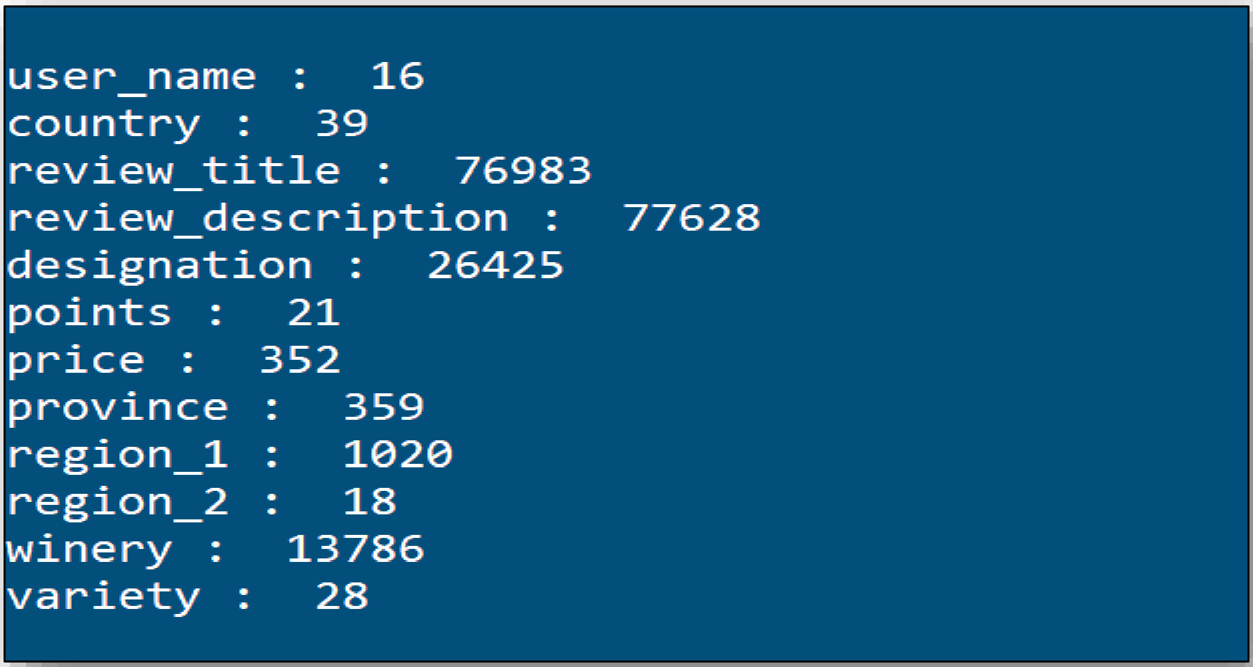
This data set has **82657 data points** and **12 attributes.**

Now we know that 80% of our data contains text with only 2 features of int and float data type.

Features such as user_name, region_2, designation aren't very useful because a large portion of their values are NaN(missing) null values. This gives us an idea of which attributes might serve the purpose of extracting insights from the data.

Price column has few missing values too. This is intuitive that we can extract useful information using this attribute and hence it becomes important to get rid of these null values by a suitable alternative. As the proportion of non-null values is significantly higher, so we can use the present information to substitute for the missing values. Here we'll use the **mean** of all prices to serve this purpose.

If we focus on few attributes individually then we have the following numbers:



```
user_name :   16
country :   39
review_title :   76983
review_description :   77628
designation :   26425
points :   21
price :   352
province :   359
region_1 :   1020
region_2 :   18
winery :   13786
variety :   28
```

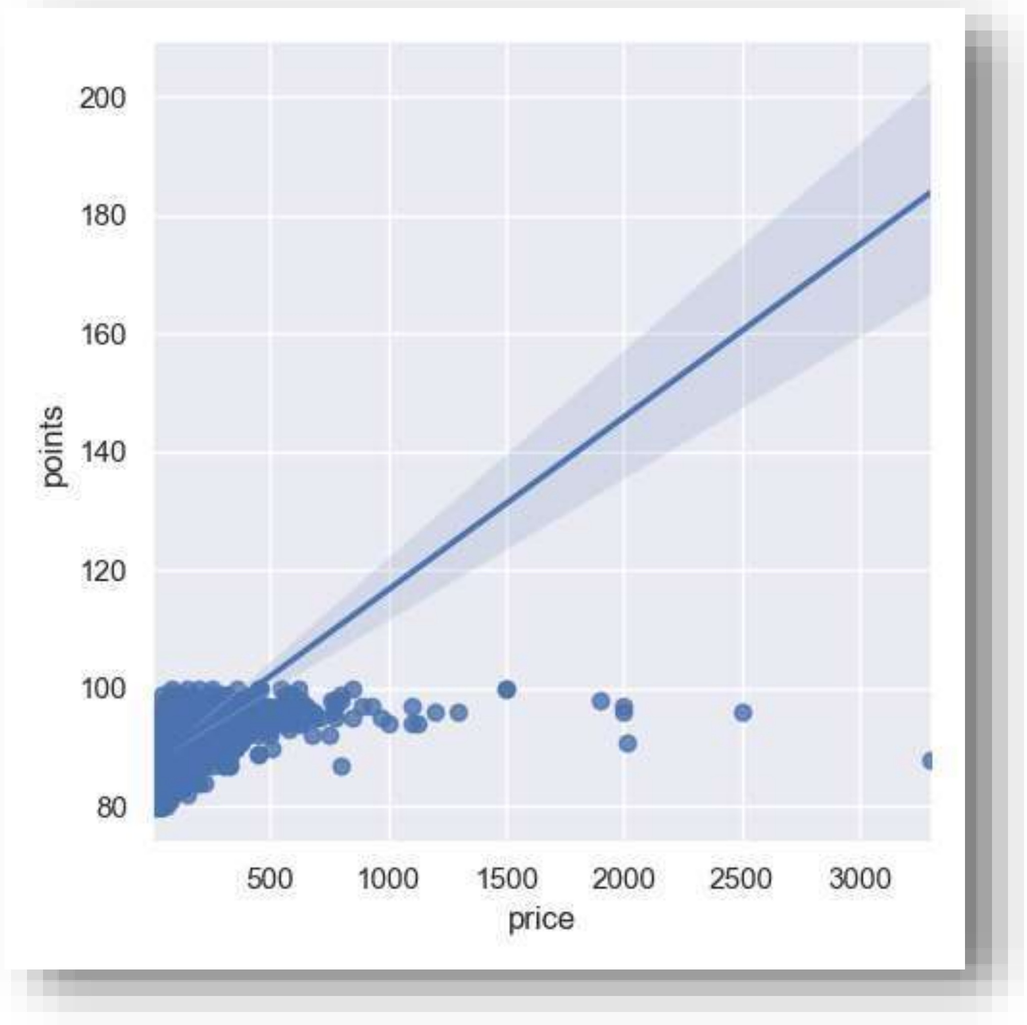
These are the number of **unique values** corresponding to each column. It tells us that the given dataset may not be a practical one as its not possible for 16 users to repeatedly give 60K reviews even if they drink too much. The column on which we are going to invest most our time is “variety”. There are **28 different varieties of wine present** and later on we will predict this attribute using different models.

In any data analysis project the first thing (after cleaning) is exploratory data analysis(EDA).

Following this successful trend, I'll check if relationships occur between various attributes.

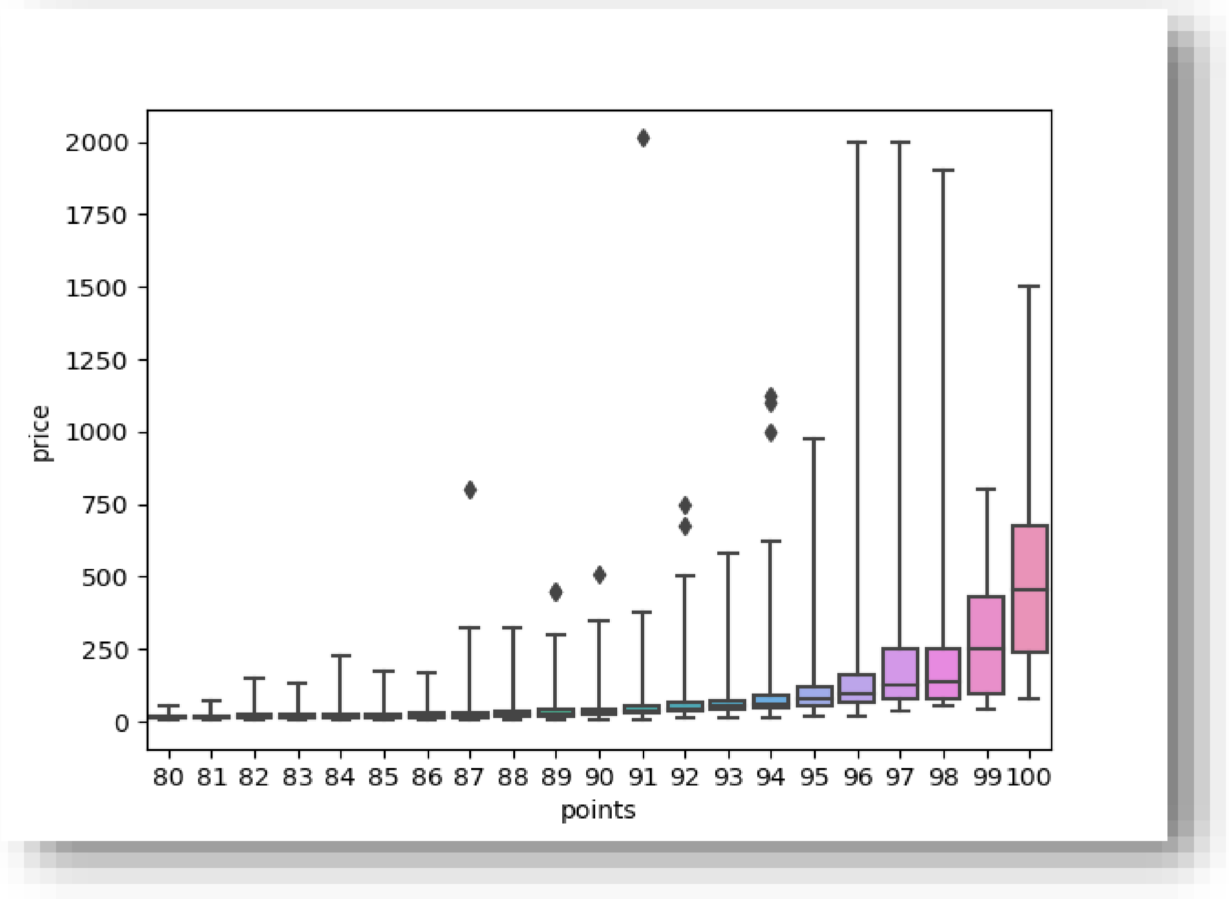
Generally the quality of a product is reflected in the price tag accompanying it. Did costlier wines receive better review than the one with

Price of the wine and points given by the user:



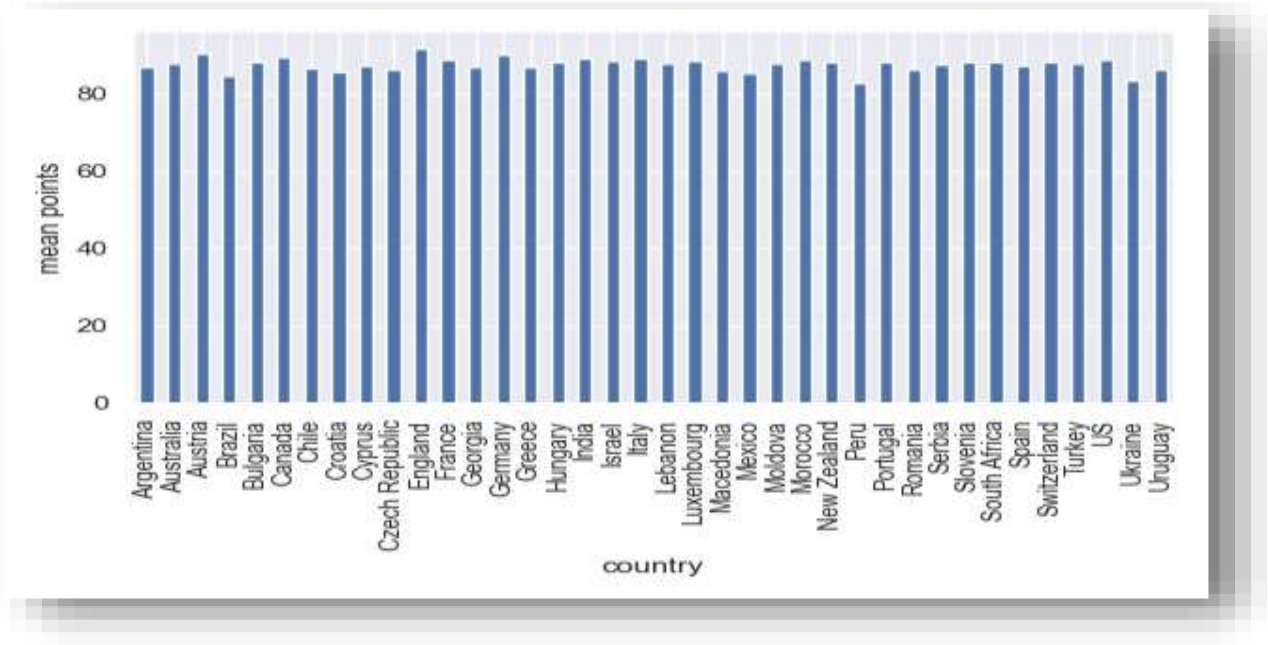
We don't even need Pearson value here!
It is clear that there is a positive relation between price and points. Let us see this distribution of prices according to points using a boxplot.

Median of price distribution corresponding to individual points satisfies what we would have intuitively inferred from our daily life experience. Although it's not true 100% but the quality of a product usually reflected in its price.



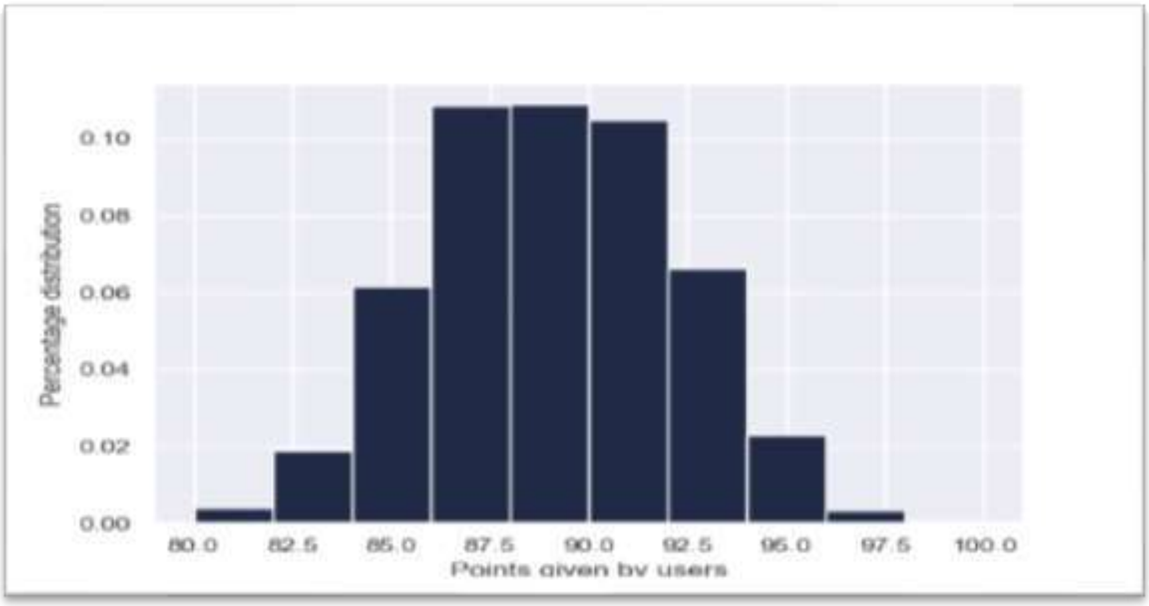
Even though it was obvious the more costlier the wine the better its quality but maybe human psychology playing some miner role as we are bound to think that if something is costly then it's a good choice over the cheaper one.

If wine originated from a particular country more famous in the audience? Here's a bar plot of mean points given by the users to different wines grouped by countries :



It seems like people are not biased towards any wine. Still few over achievers are visible like England and Australia.

Our EDA is so far based on points but are points equally balanced in the data? A histogram of points will tell :



90% of points given by users lie between 85-95. Audience has enjoyed the wine whatever maybe its variety or origin be.

Now if variety is going to be an important feature for us we should have a fair knowledge about other attributes affecting this feature which needs to be incorporated in our analysis later.

This is a list of all countries present in our dataset with the wine variety it has produced the most in terms of units made by that particular country.

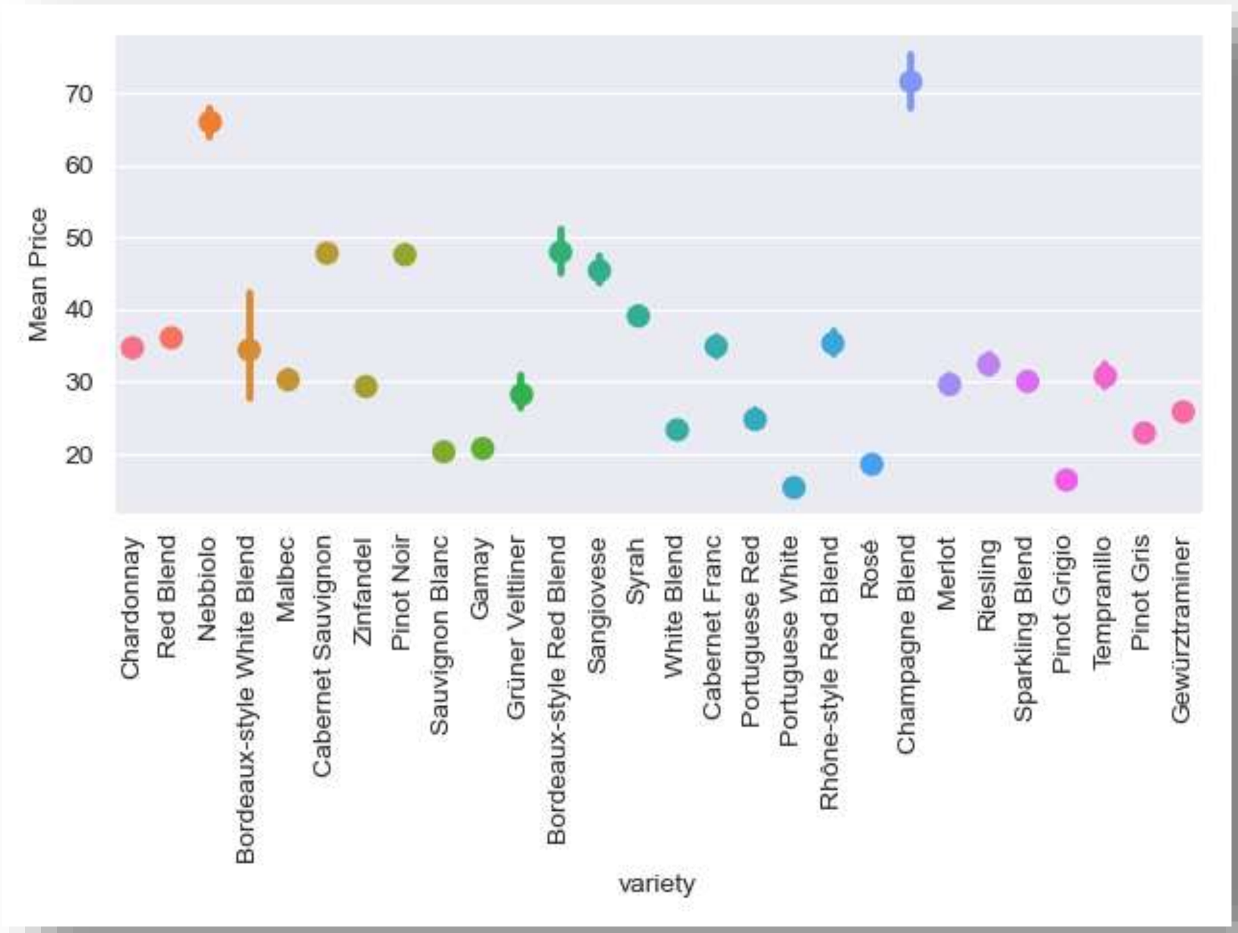
Argentina : Malbec	Australia : Chardonnay	Austria : Grüner Veltliner
Brazil : Sparkling Blend	Bulgaria : Red Blend	Canada : Riesling
Chile : Cabernet Sauvignon	Croatia : White Blend	Cyprus : Red Blend
Czech Republic : Sauvignon Blanc	England : Sparkling Blend	France : Bordeaux-style Red Blend
Georgia : White Blend	Germany : Riesling	Greece : White Blend
Hungary : White Blend	India : Sauvignon Blanc	Israel : Cabernet Sauvignon
Italy : Red Blend	Lebanon : Red Blend	Luxembourg : Riesling

Macedonia : Pinot Noir	Mexico : Red Blend	Moldova : Red Blend
Morocco : Red Blend	New Zealand : Sauvignon Blanc	Peru : Red Blend
Portugal : Portuguese Red	Romania : Pinot Noir	Serbia : Red Blend
Slovenia : White Blend	South Africa : Sauvignon Blanc	Spain : Tempranillo
Switzerland : Pinot Noir	Turkey : Red Blend	US : Pinot Noir
Ukraine : Sparkling Blend	Uruguay : Red Blend	

This is clear while finding the variety of a wine its country of origin plays a significant role with different countries producing different wine varieties.

Different varieties have different origin and quality . This affects the price of varieties to differ from each other. Is this intuition true or not, lets find out.

Here's a distribution of price grouped by varieties using point plot in seaborn



Features extracted from the review description of users

Now, we'll try to find out some interesting patterns from the reviews given by the users.

I have used a **TfidVectorizer** in order to make a **word frequency array** of **review_description** in a the form of a **csr_matrix** which will be easy to processed with **scikit- learn** library

Further I have fed this word frequency array to a **Nonnegative matrix factorization** (NMF) model. This will reduce the features into some top labels. How many top labels we want to extract out of the data depends on us. This is possible due to **interpretable power** of NMF where each component of NMF remembers a top characteristic of the input data sample.

By applying **cosine similarity method** we'll able to group data points based on their **review_description**.

In a simple language we'll be able to **cluster data** based on review_description feature as we may choose.

For instance, we can find out users who gave approximately similar reviews

User Name	Review description given by the user	Similarity
@wineschach	Strident aromas of licorice, herbs, berries and mint make it interesting, while the palate is light and fresh, with raspberry and red currant flavors. There's not a lot of heft, extract or depth to the wine, but it's likable and unhindered by excess oak and such. Enjoyable due to its unchallenging simplicity.	1.000000
@wineschach	Edgy, nervy and funky, with unfamiliar aromas and flavors. The nose is pickled and the palate is tart, linear and green. Loses itself as it opens, turning more herbal, sweaty and strange. Imported by Ecovalley Quality Wine Group and National Refrescos Import Company, LLC.	0.999969
@mattkettmann	Enticing aromas of white lilies, nectarines, white peach, honey-lime water and chalky stone hit the nose on this wine from a new vineyard on Highway 154 in the likely-to-be-named Los Olivos District. The palate is creamy down the center but with tangy edges, offering flavors of lime yogurt and pink grapefruit pith.	0.999903

@mattkettmann	The wine's very reserved nose requires patience to release the asphalt and blackberry jam aromas. Berry flavors show on the palate as does some asphalt and pepper. Everything feels balanced but it could use some more power and punch.	0.999829
@wawinereport	This wine is a unique blend of Tempranillo (62%), Graciano, Cabernet, Merlot and Garnacha. Spice, coffee, tobacco, vanilla and red-fruit aromas are followed by a smooth, elegant palate, with flavors that linger.	0.999625
@vboone	This dense, tannic and concentrated wine needs more time in the bottle. It features aromas of flint and dried herb, with similar flavors on the palate. With pronounced oak, it's expansive and rich. Cellar through 2020.	0.999617

Most of these reviews are positive with first few pointing out little things that can be improved and hence they are more similar than the last ones which sounds to be perfect with no complain.

Similarly we can do this for review_title. We'll find groups of review titles that has similar review descriptions.

Review titles	**Percent matching**
Henry Fessy 2015 Brouilly	1
Adega Vila Real 2015 Colheita Red (Douro)	0.957986
Château Acker-Perreau 2010 Bordeaux	0.956966
Domaine Chevillon-Chezeaux 2014 Bourgogne Hautes Côtes de Nuits	0.949341
Global Wines 2014 Casa de Santar Red (Dão)	0.948380
Château des Bormettes 2016 Cuvée Tradition Rosé (Côtes de Provence)	0.945457
hâteau la Croix du Casse 2013 Pomerol	0.941988
Domaine Reverdy-Ducroux 2014 Beau Roy (Sancerre)	0.940183

This way we can make cluster of any column value based on the review description .

You must have seen those top labels like ‘not-working’ , ‘good product’ , ‘stop working’ , ‘excellent product’ etc. on the review section on Amazon. Well Amazon engineers also used this approach of **unsupervised learning** and extracted **top labels** out the reviews given by users. We are going to do the same with our data using NMF , Again!
Using the **NMF model** we can also find top labels in the review description of **medium reviews**(points < 90) and **good reviews** (points >98). It may seems like 92 points is not an average rating given by users. But from our EDA in initial phase of this project, we found out that most of the reviews lies between 85-95 so the ones going up to 98 and above must be carrying a special feature with them and therefore it’s fair to treat them as good reviews, separately.

Top 15 labels in medium reviews:

wine , finish , aromas , palate , acidity , fruity , wine , drink , flavors , drink , tannins

Top 15 Labels in Good reviews:

wine , acidity , flavors , ripe , fruit , black , tannins , cherry , vintage , years , great

These must look familiar to a wine specialist who would know what type of wine generally perform well in the market.

Variety predictor :

Now the time has come to actually predict the future!
From the EDA we know that there are 28 different varieties of wine which depends on features like **country, price, points, province, regions** .

We are going to use all of these features to predict the wine variety using a classifier model.

Since **review titles , review description, user name** do not give any idea about the variety so they'll be dropped before fitting the model.

Model used: Decision Tree Classifier

Since we can not process text data and numeric data simultaneously due to the limitations of **Scikit- learn** . To overcome this problem we'll make two separate **pipelines**, one for text features and one for numeric features. We'll these two **pipelines** in the end where a classifier will fit both numeric and text features of the **training data**.

This is possible using two very useful utilities provided by **Scikit- learn**, **Function generator**

Which helps us to extract text and numeric data from the dataset separately. Other utility, **Feature union** is used to merge the two separate **pipeline**.

This approach is very useful as it lets us use different model easily without changing any code.

Accuracy metric used for the efficiency of model.

Steps:

1. Two **function generators** used with a **feature union** to create two separate pipelines, one for handling text data and one for handling numeric data.
2. Pre-processing data by **Imputing** missing values in numeric data and **combining all string** data in a single row to a become a single quantity so that it can be tokenize into a word frequency array.
3. Split the dataset in 80-20 proportions and making dummy columns for variety values using pandas **get_dummies** method.
4. Fit and predict the data and measured accuracy which comes out to be **66%**
5. Then predicted labels for test data

6. Merged the predicted labels with the test data and saved the file under the name ‘test_data_pred.csv’