Indeed - ML Intern - Muhammad_Talha

Firslty, i explore the dataset and check how many rows, columns and dtypes of each columns inside the dataset.
In explore_dataset function we are doing all the things that given below:

- checking shape of dataset
- checking column names that exist in dataset
- checking null values
- checking overall information about dataset using info() method
- checking dtypes of columns

```
================================================================================================
                        EDA Dataset: /content/drive/MyDrive/Assignment Data.csv
================================================================================================
Total Rows: 129971

Total Columns: 17

Columns: ['Unnamed: 0', 'country', 'description', 'designation', 'points', 'price', 'province', 'region_1', 'region_2', 'taster
_name', 'taster_twitter_handle', 'title', 'variety', 'winery', 'Unnamed: 14', 'Unnamed: 15', 'Unnamed: 16']

Total Null Values: 594665
Unnamed: 0                   0
country                     63
description                  0
designation              37465
points                       0
price                     8996
province                    63
region_1                 21247
region_2                 79460
taster_name              26244
taster_twitter_handle    31213
title                        0
variety                      1
winery                       0
Unnamed: 14             129971
Unnamed: 15             129971
Unnamed: 16             129971
dtype: int64
```

After i selected two columns that description and variety that we will use for model training. Description features will be our independent features and variety will be dependent feature.

```
: df = df[['description', 'variety']]
  df.head()
```

:

	description	variety
0	Aromas include tropical fruit, broom, brimston...	White Blend
1	This is ripe and fruity, a wine that is smooth...	Portuguese Red
2	Tart and snappy, the flavors of lime flesh and...	Pinot Gris
3	Pineapple rind, lemon pith and orange blossom ...	Riesling
4	Much like the regular bottling from 2012, this...	Pinot Noir

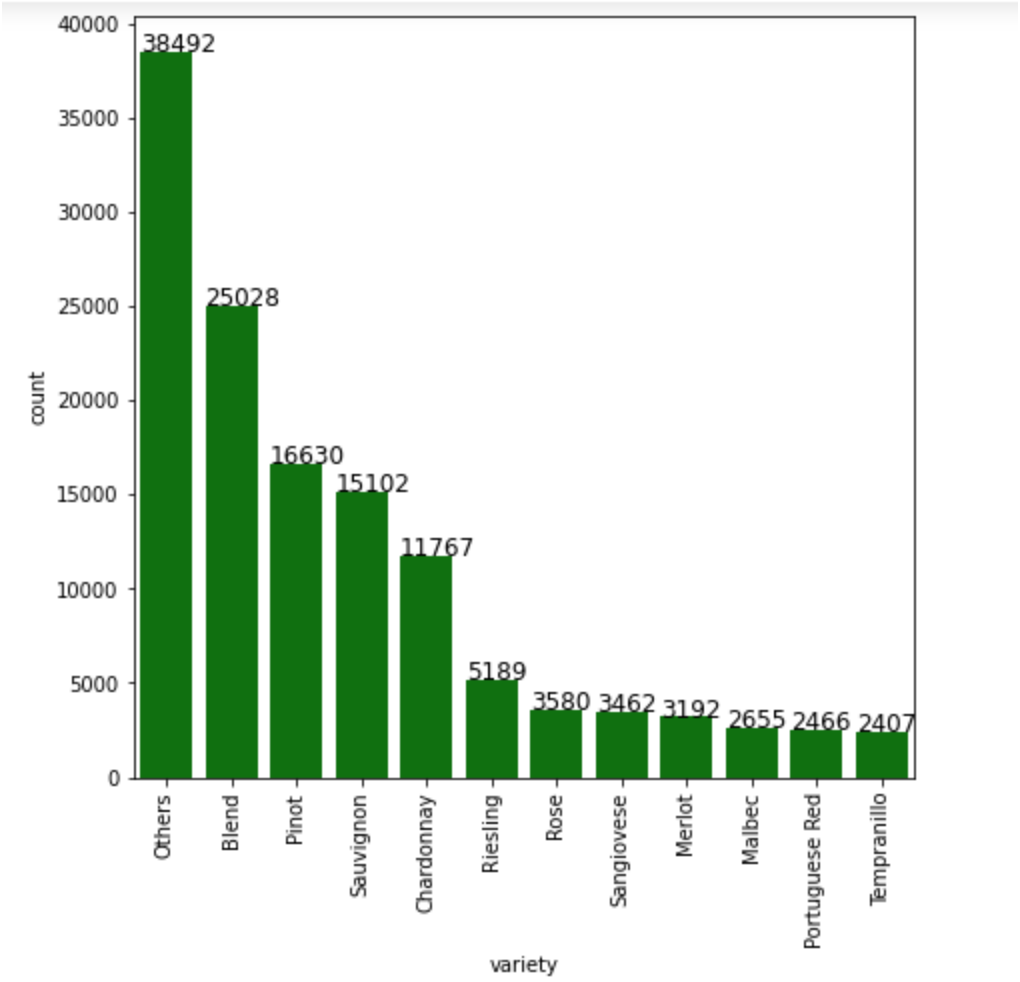
We can see below total 707 variety inside our dataset. So, I combine all the sub varieties into main variety.

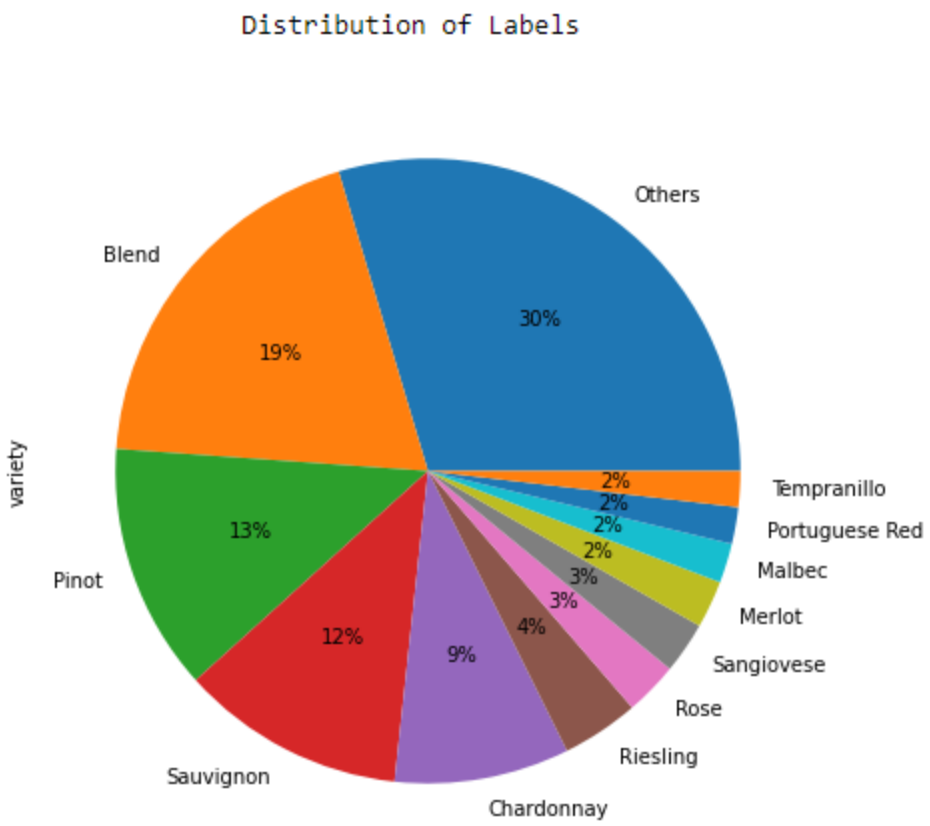
```
v = df['variety'].value_counts().reset_index()
v
```

	index	variety
0	Pinot Noir	13272
1	Chardonnay	11753
2	Cabernet Sauvignon	9472
3	Red Blend	8946
4	Bordeaux-style Red Blend	6915
...	...	...
702	Cabernet Sauvignon-Barbera	1
703	Sauvignonasse	1
704	Forcallà	1
705	Meseguera	1
706	Bobal-Cabernet Sauvignon	1

707 rows × 2 columns

When I combine all the sub varieties into one main variety, So, we can see the distribution of varieties with count and percentage.


40000
35000
30000
25000
20000
15000
10000
5000
0
count
38492
25028
16630
15102
11767
5189
3580
3462
3192
2655
2466
2407
Others
Blend
Pinot
Sauvignon
Chardonnay
Riesling
Rose
Sangiovese
Merlot
Malbec
Portuguese Red
Tempranillo
variety



After i do preprocssing on description features and convert variety categorical feature to numeric form.
In preprocess_text function we will apply all the things that given below:

- removing links
- removing special characters
- removing punctuations
- removing numbers
- removing stopwords
- doing stemming
- transforming in lowercase
- removing excessive whitespaces

```
# creating a dictionary
dic = {'Others':0,'Blend':1,'Pinot':2,'Sauvignon':3,'Chardonnay':4,'Riesling':5,'Rose':6,
       'Sangiovese':7,'Merlot':8,'Malbec':9,'Portuguese Red':10,'Tempranillo':11}
df['variety_label'] = df['variety'].map(dic)
```

```
df.head()
```

	description	variety	description_len	cleaned_description	variety_label
0	Aromas include tropical fruit, broom, brimston...	Blend	172	aroma includ tropic fruit broom brimston dri h...	1
1	This is ripe and fruity, a wine that is smooth...	Portuguese Red	227	ripe fruiti wine smooth still structur firm ta...	10
2	Tart and snappy, the flavors of lime flesh and...	Pinot	186	tart snappi flavor lime flesh rind domin green...	2
3	Pineapple rind, lemon pith and orange blossom ...	Riesling	199	pineappl rind lemon pith orang blossom start a...	5
4	Much like the regular bottling from 2012, this...	Pinot	249	much like regular bottl come across rather rou...	2

I also check which words appears most times in our dataset, that analysis we can see below.

	word	Frequency_distribution
0	wine	83105
1	flavor	70969
2	fruit	63934
3	aroma	41052
4	finish	40465
5	acid	39812
6	palat	38636
7	drink	33970
8	cherri	33590
9	tannin	32980
10	ripe	29143
11	black	29053
12	dri	26516
13	note	25305
14	spice	23546

Data Splition

- 60% for training
- 20% for validation
- 20% for testing

Feature Extraction From TfidfVectorizer

- TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.

Models Evaluation

	accuracy	precision	recall	f1-score
LogisticRegression (TF-IDF)	0.661345	0.721068	0.543703	0.602493
XGBoost (TF-IDF)	0.588174	0.754979	0.463351	0.550382
Random Forest (TF-IDF)	0.647457	0.844718	0.480007	0.566715

From above model evaluation table, we can see random forest giving better precision 84% as compare to other models.

Deep Learning

Model Architecture

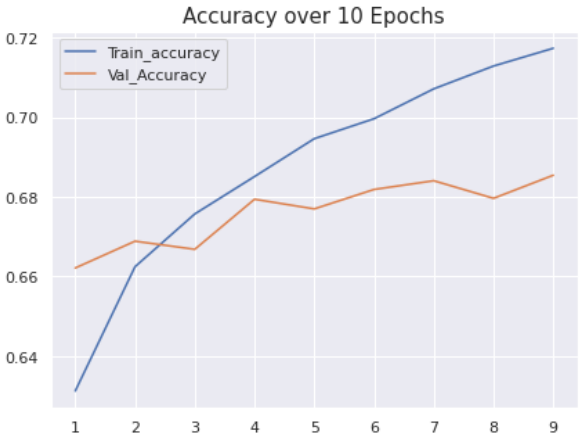
```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_8 (Embedding)     (None, 73, 128)           384000

 spatial_dropout1d_5 (Spatia  (None, 73, 128)          0
 lDropout1D)

 lstm_8 (LSTM)               (None, 256)               394240

 dense_8 (Dense)             (None, 12)                3084

=================================================================
Total params: 781,324
Trainable params: 781,324
Non-trainable params: 0
_________________________________________________________________
None
```



Accuracy on validation set: 0.6749
Precision on validation set: 0.7838
Recall on validation set: 0.5748
F1_Score on validation set: 0.6394

Deep learning giving better results as compare to machie learning models. We can see F1 score is better then machine learning models.