WINE VARIETY PREDICTION FROM WINE REVIEWS

A term project

Presented to

Professor Leonard Wesley

Department of Computer Science

San José State University

In Partial Fulfilment

Of the Requirements for the Class

CS 185C

By

Hardik Kumar (014521039)

April 2020

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

The dataset used in the project is the dataset “winemag-data_first150k.csv”, containing records of wine reviews. The aim of the project is to read the text description of the review of the wine, and then predicting the variety of wine. The project is done in phases:

- **Data Preprocessing**: The data is cleaned to remove unnecessary noise and outliers, making it easier for the machine learning model to learn and give the most accurate predictions.
- **Vectorization:** Using TF-IDF vectorizer and Word2Vec embeddings, each text description is converted into vectors of length 5000 and 100 respectively, for the machine learning model to learn on.
- **Feature Extraction:** The vectors extracted after implementing both TF-IDF and Word2Vec embeddings are used as features to be fed into the machine learning classifier.
- **Machine Learning:** RandomForest, K-Nearest Neighbor and SVM classifiers have been used for the prediction and their performances have been evaluated.

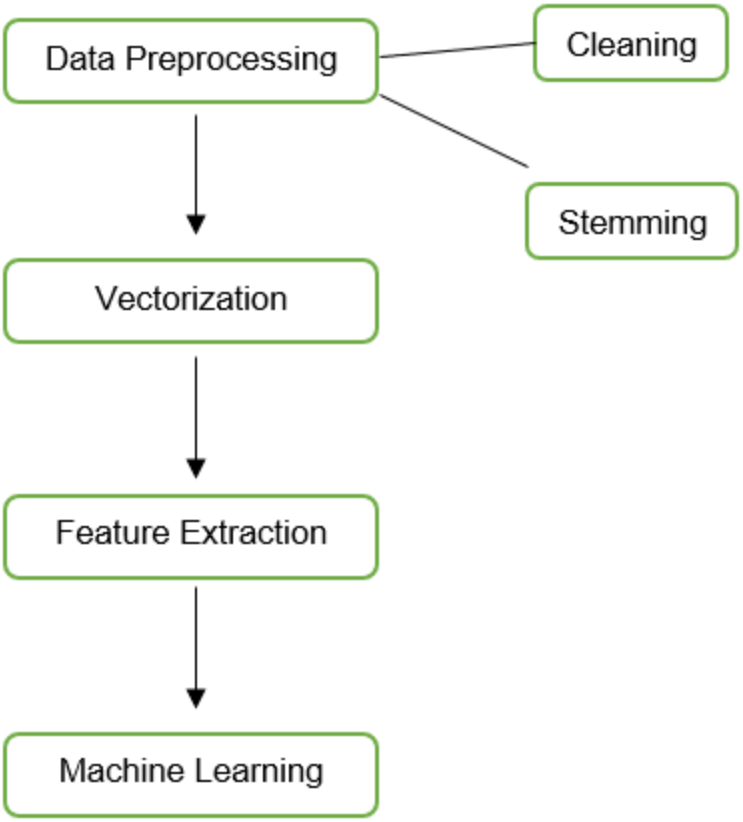

Fig.1 Overview of the Project methodology

DATA COLLECTION

For the project, the dataset was provided by the professor. The dataset used in the project is **“winemag-data_first150k.csv”**. The dataset consists of 150,000 records of wines and their reviews. Some of the fields in the dataset are the country of the wine, points, price, description and variety. The main fields focused and used in this project are the description and the variety. The variety are the label classes which are to be predicted. Examples of varieties: Chardonnay, Pinot Noir, Cabernet Sauvignon, Red Blend, Bordeaux-style Red Blend, Sauvignon Blanc etc. The description is the review written by a connoisseur for a particular wine, and accordingly other information have been provided for that particular review like the winery, region_1, designation, variety etc. The task is to read the description, vectorize it, feed the features into the machine learning model and predict the variety.

Fig.2 is an example of a single record in the wine dataset

country	description	designation	points	price	province	region_1	region_2	variety	winery
US	This tremendous 100% varietal wine hails from Oakville and was aged over three years in oak. Juicy red-cherry fruit and a compelling hint of caramel greet the palate, framed by elegant, fine tannins and a subtle minty tone in the background. Balanced and rewarding from start to finish, it has years ahead of it to develop further nuance. Enjoy 2022â€“2030.	Martha's Vineyard	96	235	California	Napa Valley	Napa	Cabernet Sauvignon	Heitz

Fig.2 Data fields in the wine dataset.

DATA PREPROCESSING

Data preprocessing has been identified as the crucial stage for any machine learning based application. It plays an important role as it focusses on processing the data to make it most suitable for the model. For our problem statement, the data is preprocessed in 4 phases:

A. Alphanumeric removal

Removed punctuations and any other alphanumeric character from the text description. The intuition behind this is that alphanumeric characters do not add much meaning and information while learning. This removes unnecessary noise in the data and makes it less bulky, hence making it better for the machine learning model to learn on.

B. Stop words removal

Stopwords are words like he, she, us, is etc. which do not contribute significantly to the meaning of the text. Hence, removal of stopwords along with punctuations and alphanumeric characters helps removing noise from the sentence and enables the machine to deduce the meaning of the sentence efficiently.

C. Stemming

Words are formed from stems, the core meaning unit, and the affixes which are the bits and pieces that adhere to the stems. Stemming is the process of removing the affixes using heuristics in order to reduce the word to its stem or base. This reduces the complexity of the sentence by reducing word inflections and variant forms, thus making it easier to classify the text into a particular category.

DATA CLEANING

For the data cleaning stage, the 150,930 records were checked for the varieties. For analyzing the outliers, the counts of each wine variety were evaluated. The total wine varieties in the data are 632. Out of them, just a few have enough data for the model to train on and the rest can be considered as outliers, because not enough data is present for them. Fig. 3 gives the count for outlier wine varieties.

	A	B
1	Review Counts	Variety
2	1	Viognier-Valdiguié
3	1	Moscatel Graúdo
4	1	Caprettone
5	1	Vitovska
6	1	Aidani
7	1	Chardonelle
8	1	Malvazija
9	1	Sangiovese Cabernet
10	1	Cabernet Moravia
11	1	Durello
12	1	Premsal
13	1	Pinot Grigio-Sauvignon Blanc
14	1	Chinuri
15	1	Orangetraube
16	1	Muskat
17	1	Garnacha Tintorera
18	1	Rabigato
19	1	Rufete
20	1	Trousseau Gris
21	1	Groppello
22	1	Bombino Bianco
23	1	Jacquez
24	1	Rebula
25	1	Moscofilero
26	1	Merlot-Petite Verdot
27	1	Syrah-Bonarda
28	1	Cabernet Pfeffer
29	1	Mavrotragano

	A	B
466	31	Catarratto
467	32	Rosato
468	32	Roditis
469	33	White Riesling
470	33	Furmint
471	33	Pecorino
472	34	Frappato
473	35	Zibibbo
474	36	Tannat-Cabernet
475	37	Marsanne-Roussanne
476	37	CastelÃ£o
477	37	Claret
478	38	Lagrein
479	39	Tocai Friulano
480	40	Charbono
481	40	Chenin Blanc-Chardonnay
482	40	Traminer
483	42	Alicante Bouschet
484	42	Moscatel
485	43	Cannonau
486	44	Moschofilero
487	44	Malvasia
488	45	Sylvaner
489	46	Cabernet Sauvignon-Cabernet Franc
490	47	Petit Manseng
491	47	Syrah-Grenache
492	47	Austrian white blend

Fig.3 Outlier wine varieties

Therefore, if these varieties were to be considered, the model would not give efficient results as the majority of 632 wine varieties (labels) are outliers. Hence, only the records of top 30 wines are considered. Fig. 4 shows the top 30 wines with the review counts.

Review counts	Variety
14482	Chardonnay
14291	PinotNoir
12800	CabernetSauvignon
10062	RedBlend
7347	Bordeaux-styleRedBlend
6320	SauvignonBlanc
5825	Syrah
5524	Riesling
5070	Merlot
3799	Zinfandel
3345	Sangiovese
3208	Malbec
2824	WhiteBlend
2817	Rosé
2556	Tempranillo
2241	Nebbiolo
2216	PortugueseRed
2004	SparklingBlend
1970	Shiraz
1682	CorvinaRondinellaMolinara
1505	Rhône-styleRedBlend
1365	PinotGris
1365	Barbera
1363	CabernetFranc
1346	SangioveseGrosso
1305	PinotGrigio
1263	Viognier
1261	Bordeaux-styleWhiteBlend
1238	ChampagneBlend
1058	Port

Fig.4 Top 30 wine classes considered

DATA VISUALIZATION

- **Wine Variety Frequency COUNTPLOT**

  Showing the counts of wine variety whose records are atleast **250.** This gives us the information regarding how much data is outlier and cannot be used for prediction. (Generated in Python)

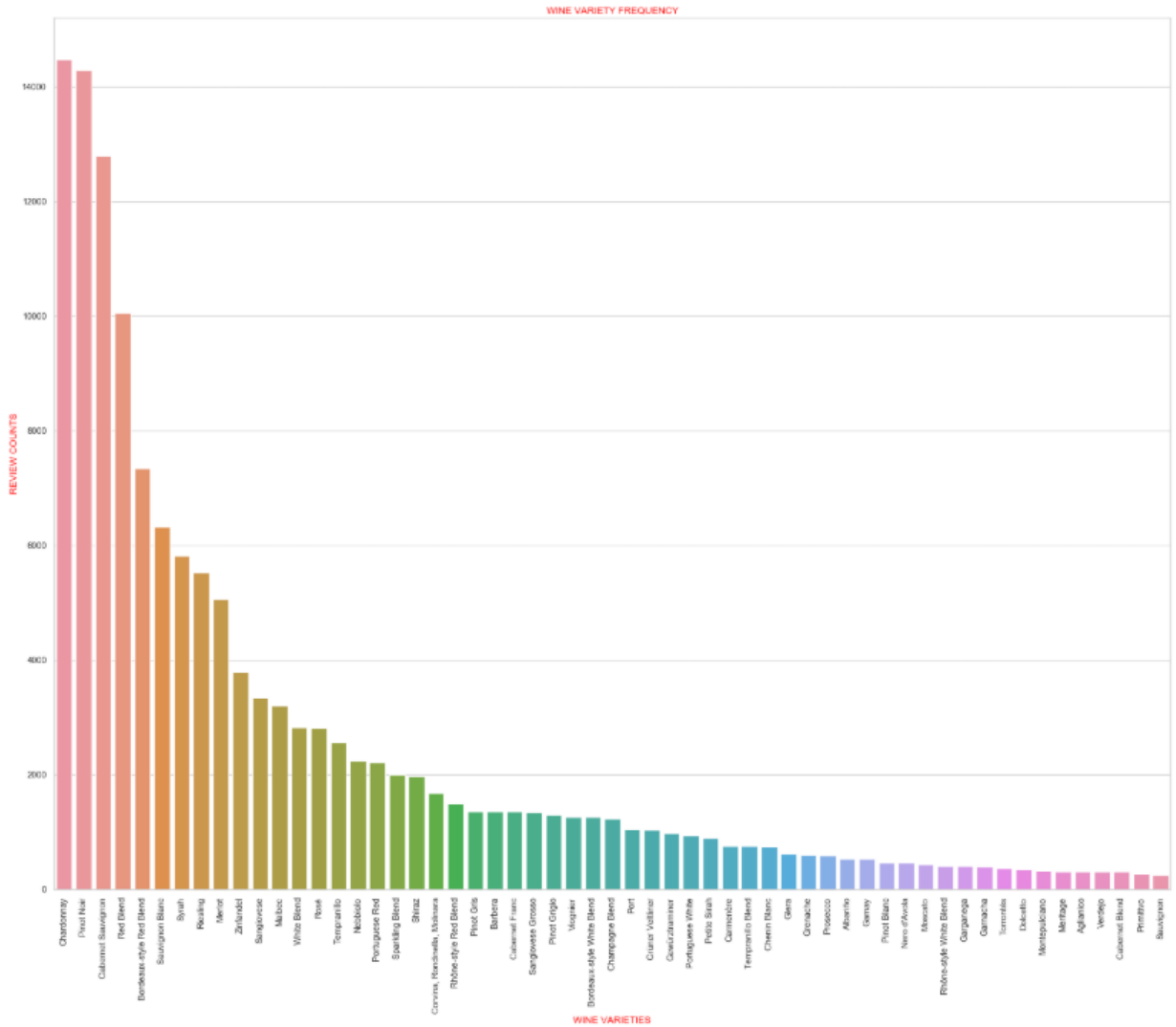

Fig.5 Countplot of wine varieties

- **<u>Top 30 Wine Variety Frequency BAR CHART</u>**

  Generated using Tableau

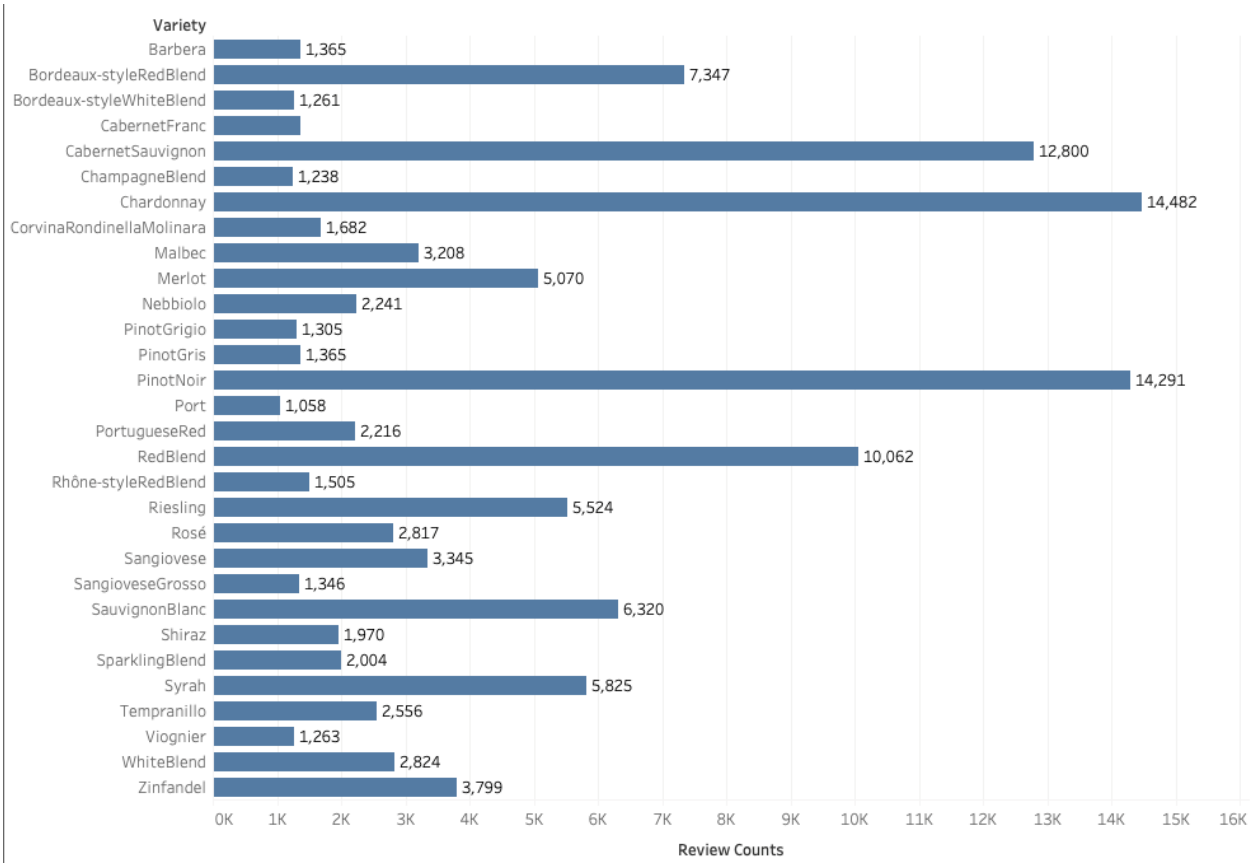

Fig. 6 Top 30 Wine Variety Counts

- **Mean price per Wine Variety VERTICAL BAR CHART**

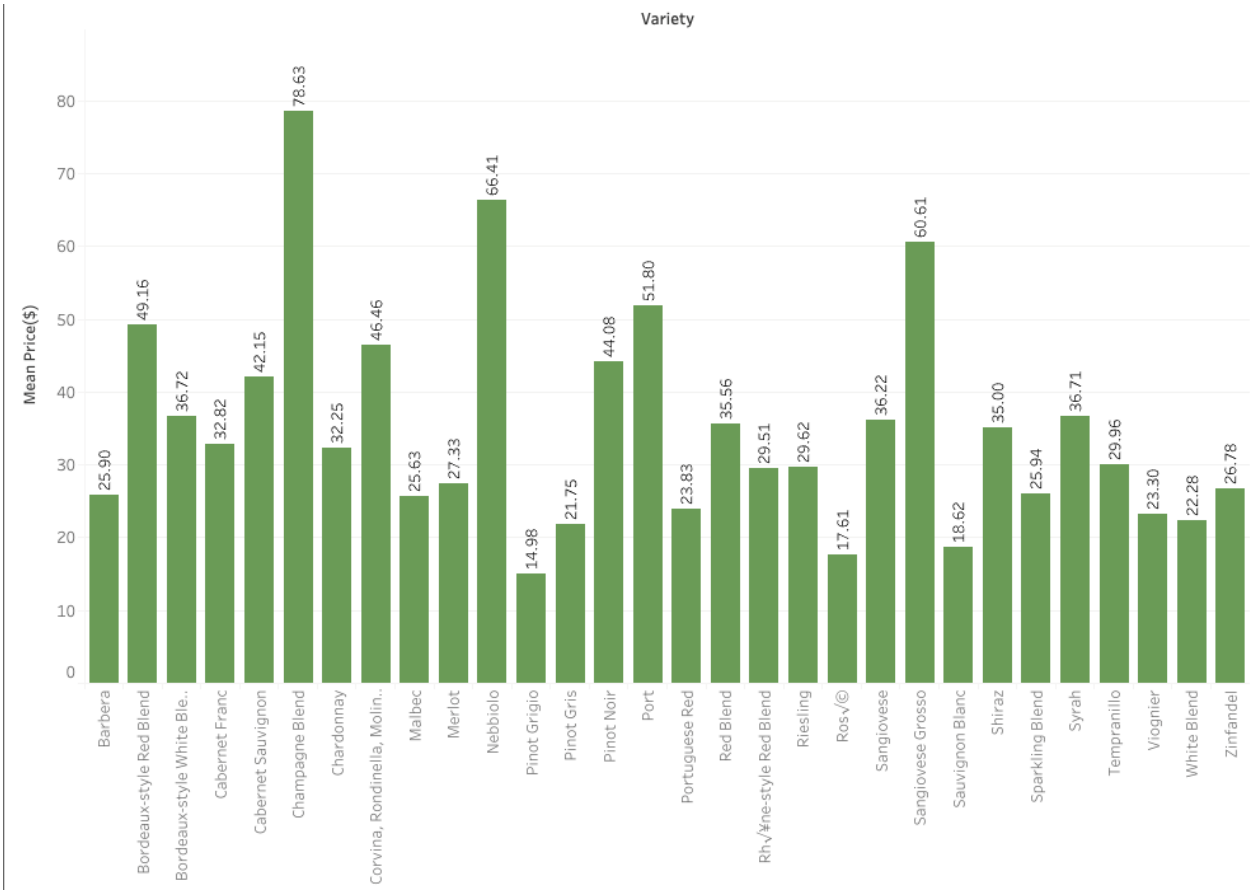

Fig. 7 Mean Price per variety

TEXT VECTORIZATION

Two types of vectorising algorithms have been used for feature extraction:

- **Word Embeddings**

  Word2Vec is a widely used word embedding technique. It converts text into vectors which can be understood by machine learning and deep learning models. Word2Vec can be implemented using two ways-:

  Common bag of Words Model : This takes as input contexts of different words and it produces a target word matching that context. It is faster and works better on big datasets.

  Skip-gram Model : This takes input as target word and produces words that might be of similar context to input word. It works well on small data sets.

  As the dataset is bigger, we use Common bag of words model. The sentences are converted into embedding vectors of length 100. Word2Vec embeddings help in understanding the context of the sentence and can even find similarity in two words on the basis of cosine similarity. Fig. 7 shows an example of this.

```
In [111]: wine2vec.wv.most_similar('melon')

Out[111]: [('nectarin', 0.839788556098938),
           ('citru', 0.7748647928237915),
           ('peach', 0.7630621790885925),
           ('cantaloup', 0.7594323754310608),
           ('honeydew', 0.7553642392158508),
           ('papaya', 0.7529636025428772),
           ('tropic', 0.7404524087905884),
           ('grapefruit', 0.7358745336532593),
           ('banana', 0.732739269733429),
           ('meloni', 0.7305516004562378)]
```

Fig.8 Similarity of word after word2vec

In addition, **normalizing** the 100 sized vectors gave lesser prediction accuracy than the original vectors, so the vectors are not normalized.

- **Term Frequency-Inverse Document Frequency (TFIDF)**

  TF-IDF Vectorizer works on the principle of reflecting the importance of a word in a document, and even in a sentence. It assigns a weight or confidence value to each word in the document according to the following formula :

  $$W_{ij} = tf_{ij} * log(N/df_i)$$

  Where, tf_{ij} = no. of times word i occurs in document j divided by the total number of terms in j

  df_i = no. of documents containing i

  N = total no. of documents

  So, according to the formula, the word which is common throughout the data is given less weightage as it might not be a strong identifier to categorize the review data. Some high frequency words are given higher weightage than other words and **5000** most frequent words are considered. Fig. 8 shows the percentage occurrence of **30** most frequent words.

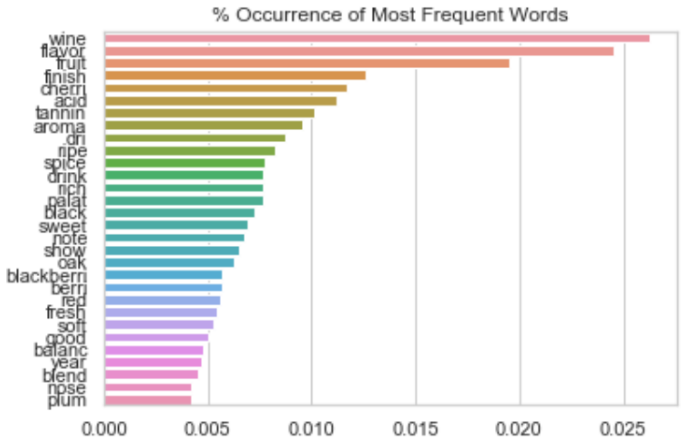

Fig.9 Top 30 most frequent words' occurrence %

In addition to that, the vectors are **normalized** to give better frequency.

MACHINE LEARNING

Machine Learning classifiers are applied on two different feature sets of Word2Vec embeddings and TF-IDF

- ## Word2Vec embeddings

  - ***RandomForest***

    Hyperparameter tuning is performed on the "max_depth" and "n_estimators" parameters of Random Forest classifier.

    ***max_depth**: 10, 25, 50, 100, 150, None*

    ***n_estimators**: 20, 50, 100, 150*

    The best estimators giving the highest accuracy are **max_depth=25** and **n_estimators=50.**

- ***Support Vector Machine (SVM)***

  Hyperparameter tuning is performed on the "kernel", "C" and "gamma" parameters of Random Forest classifier.

  kernel: *linear, rbf*

  C: *0.1, 1, 10, 100*

  gamma: *0.01, 0.1, 1*

  The best estimators giving the highest accuracy are **kernel=linear, C=10** and **gamma=0.1**

- **TFIDF**

  - ***RandomForest***

    Hyperparameter tuning is performed on the "max_depth" and "n_estimators" parameters of Random Forest classifier.

    max_depth*: 10, 25, 50, 100, 150, None*

    n_estimators*: 20, 50, 100, 150*

    The best estimators giving the highest accuracy are **max_depth=50** and **n_estimators=100.**

- ***Support Vector Machine (SVM)***

  Hyperparameter tuning is performed on the “kernel”, “C” and “gamma” parameters of Random Forest classifier.

  kernel: *linear, rbf*

  C: *0.1, 1, 10, 100*

  gamma: *0.01, 0.1, 1*

  The best estimators giving the highest accuracy are **kernel=linear, C=10** and **gamma=0.1**

  The same parameters have been proven to be the best parameters for both Word2Vec and TF-IDF vectors.

PREDICTION RESULTS AND EVALUATION

- **Word2Vec embeddings**

  - ***RandomForest***

Fit Time: 31.006 / Pred Time: 0.552 --------

	precision	recall	f1-score	support
Chardonnay	0.97	0.85	0.90	307
Pinot Noir	0.75	0.80	0.77	1460
Cabernet Sauvignon	0.97	0.62	0.75	255
Red Blend	1.00	0.55	0.71	273
Bordeaux-style Red Blend	0.66	0.82	0.73	2596
Sauvignon Blanc	0.93	0.56	0.70	249
Syrah	0.72	0.95	0.82	2814
Riesling	0.75	0.67	0.71	333
Merlot	0.78	0.68	0.73	650
Zinfandel	0.83	0.61	0.70	1014
Sangiovese	0.82	0.80	0.81	483
Malbec	0.95	0.69	0.80	249
White Blend	0.99	0.54	0.70	278
Rosé	0.77	0.87	0.82	2858
Tempranillo	0.97	0.60	0.74	231
Nebbiolo	0.83	0.67	0.74	455
Portuguese Red	0.70	0.79	0.74	1974
Sparkling Blend	0.87	0.61	0.71	277
Shiraz	0.88	0.83	0.85	1107
Corvina, Rondinella, Molinara	0.81	0.73	0.77	550
Rhône-style Red Blend	0.80	0.65	0.72	696
Barbera	0.77	0.70	0.73	270
Pinot Gris	0.86	0.81	0.83	1251
Cabernet Franc	0.83	0.66	0.74	397
Sangiovese Grosso	0.89	0.71	0.79	399
Pinot Grigio	0.86	0.66	0.75	1150
Viognier	0.68	0.71	0.69	529
Bordeaux-style White Blend	1.00	0.53	0.69	243
Champagne Blend	0.95	0.71	0.82	559
Port	0.86	0.77	0.81	784
accuracy			0.77	24691
macro avg	0.85	0.71	0.76	24691
weighted avg	0.79	0.77	0.77	24691

Getting an accuracy of **77%** with RandomForest classifier, with average precision of **85%** and average recall of **71%.** Following are the extra evaluation metrics:

```
AUC Score: 0.8770829002893473

Sensitivity of classification:  0.7050073780079191

Specificity of classification:  0.9918055266859565
```

- ***Support Vector Machine (SVM)***

	precision	recall	f1-score	support
Chardonnay	0.52	0.69	0.60	245
Pinot Noir	0.69	0.47	0.56	1460
Cabernet Sauvignon	0.30	0.84	0.45	253
Red Blend	0.16	0.41	0.23	287
Bordeaux-style Red Blend	0.74	0.40	0.52	2576
Sauvignon Blanc	0.42	0.69	0.52	256
Syrah	0.90	0.60	0.72	2974
Riesling	0.52	0.77	0.62	338
Merlot	0.35	0.48	0.40	643
Zinfandel	0.30	0.36	0.33	982
Sangiovese	0.63	0.77	0.69	460
Malbec	0.29	0.63	0.40	249
White Blend	0.28	0.47	0.35	258
Rosé	0.88	0.65	0.75	2920
Tempranillo	0.35	0.68	0.46	210
Nebbiolo	0.31	0.82	0.45	452
Portuguese Red	0.72	0.40	0.51	2001
Sparkling Blend	0.42	0.65	0.51	299
Shiraz	0.82	0.71	0.76	1073
Corvina, Rondinella, Molinara	0.55	0.75	0.64	518
Rhône-style Red Blend	0.47	0.37	0.41	663
Pinot Gris	0.55	0.72	0.62	284
Barbera	0.79	0.61	0.69	1257
Cabernet Franc	0.34	0.67	0.45	369
Sangiovese Grosso	0.64	0.72	0.67	414
Pinot Grigio	0.53	0.43	0.48	1191
Viognier	0.30	0.63	0.41	505
Bordeaux-style White Blend	0.28	0.62	0.38	243
Champagne Blend	0.59	0.56	0.57	583
Port	0.62	0.76	0.68	728
accuracy			0.56	24691
macro avg	0.51	0.61	0.53	24691
weighted avg	0.65	0.56	0.58	24691

Getting an accuracy of **56%** with RandomForest classifier, with average precision of **51%** and average recall of **61%.** Following are the extra evaluation metrics:

```
AUC Score: 0.7741467928451865

Sensitivity of classification:  0.6103822085139986

Specificity of classification:  0.9849431195230125
```

- **TF-IDF**

NOTE: For TF-IDF features, the top 15 wine varieties are used for prediction because if we use all 30 wine varieties, the matrix is too large and it was computationally impossible for me to compute the results and the RAM crashed. So, it was reduced down to 15 wine varieties and maximum 5000 features.

➢ ***RandomForest***

	precision	recall	f1-score	support
Chardonnay	0.80	0.84	0.82	1467
Pinot Noir	0.83	0.86	0.84	2624
Cabernet Sauvignon	0.87	0.96	0.91	2927
Red Blend	0.84	0.77	0.80	637
Bordeaux-style Red Blend	0.91	0.76	0.83	1006
Sauvignon Blanc	0.87	0.88	0.87	2864
Syrah	0.83	0.82	0.82	2022
Riesling	0.91	0.88	0.89	1068
Merlot	0.83	0.84	0.83	540
Zinfandel	0.84	0.78	0.81	653
Sangiovese	0.89	0.85	0.87	1271
Malbec	0.90	0.80	0.85	1100
White Blend	0.66	0.83	0.73	477
Rosé	0.92	0.76	0.83	591
Tempranillo	0.89	0.85	0.87	807
accuracy			0.85	20054
macro avg	0.85	0.83	0.84	20054
weighted avg	0.86	0.85	0.85	20054

Getting an accuracy of **85%** with RandomForest classifier, with average precision of **85%** and average recall of **83%.** Following are the extra evaluation metrics:

```
AUC Score: 0.9188899083358034

Sensitivity of classification:  0.8315785822743503

Specificity of classification:  0.9891853211114405
```

- ***Support Vector Machine (SVM)***

	precision	recall	f1-score	support
Chardonnay	0.62	0.77	0.68	1467
Pinot Noir	0.80	0.69	0.74	2624
Cabernet Sauvignon	0.88	0.84	0.86	2927
Red Blend	0.60	0.66	0.62	637
Bordeaux-style Red Blend	0.77	0.59	0.67	1006
Sauvignon Blanc	0.86	0.67	0.75	2864
Syrah	0.78	0.59	0.67	2022
Riesling	0.76	0.88	0.82	1068
Merlot	0.53	0.80	0.64	540
Zinfandel	0.52	0.75	0.61	653
Sangiovese	0.79	0.79	0.79	1271
Malbec	0.81	0.64	0.71	1100
White Blend	0.28	0.83	0.42	477
Rosé	0.73	0.75	0.74	591
Tempranillo	0.80	0.76	0.78	807
accuracy			0.73	20054
macro avg	0.70	0.73	0.70	20054
weighted avg	0.76	0.73	0.73	20054

Getting an accuracy of **73%** with RandomForest classifier, with average precision of **70%** and average recall of **73%**

```
AUC Score: 0.853204567308349

Sensitivity of classification:  0.7341249957630798

Specificity of classification:  0.9804272756411131
```

CONCLUSIONS

After using different combinations of vectorizing algorithms and different machine learning algorithms, it seems that Random Forest turns out to be the best classifier for our data with the TF-IDF features. It gave the best accuracy of **85%** and the specificity and sensitivity values are **0.83** and **0.98**. This means that the proportion of true negatives and true positives are much higher than misclassifications, giving us high precision of **85%** and high recall of **83%** as well. The AUC score is also impressive and the highest with a value of 0.91 which gives the successful classification rate and strength of the classifier. Moreover, the data does not seem to overfit as the estimator trees are also only 100 and the depth is also not infinite (50). Hence, Random Forest performs the best with TF-IDF features of the wine description, and predicts the wine variety with the best results. For future work, more data can be collected for the outlier wine varieties for them to be trained on and to predict for more varieties. In addition to that, deep learning networks like vanilla RNN, LSTM and GRU might work better for our results as these work better for text classification and also on more data.

REFERENCES

[1] W. Zhang et al., “Bridging the gap between training and inference for neural machine translation,” *Int. Conf. on Acoust., Spch., and Sign. Proc.*, vol. 94, no. 12, Jun. 2019

[2] S.A. Alasadi and W.S. Bhaya, “Review of data preprocessing techniques in data mining,” *J. of Engg. and Appl. Sci.*, vol. 12, no. 16, pp. 4102-4107, 2017

[3] Plisson J, Lavrac N, Mladenic D, “A rule based approach to word lemmatization,” *InProceedings of IS 2004*, vol. 3, pp. 83-86

[4] Liu CZ, Sheng YX, Wei ZQ, Yang YQ, “Research of Text Classification Based on Improved TF-IDF Algorithm,” *In2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE) 2018*, Aug 24 pp. 218-222

[5] Rong X. word2vec parameter learning explained. arXiv preprint arXiv:1411.2738. 2014 Nov 11

[6] R.N. Waykole and A.D. Thakare, “A review of feature extraction methods for text classification,” *Int. J. of Adv. Engg. and Res. Dev.*, vol. 5, no. 4, Apr. 2018

[7] B. Xu et al., “An improved random forest classifier for text categorization,” *J. of Comp.*, vol. 7, no. 12, Dec. 2012

[8] Lee JJ, Lee PH, Lee SW, Yuille A, Koch C., “Adaboost for text detection in natural scene,” *In2011 International Conference on Document Analysis and Recognition 2011*, Sep 18 pp. 429-434

[9] Chen D, Bourlard H, Thiran JP., "Text identification in complex background using SVM" *Proc of the 2001 IEEE Comp Soc Conf on Comp Vis and Patt Recog*. CVPR 2001 2001 Dec 8 (Vol. 2, pp. II-II)

[10] S. Xu, "Bayesian Naïve Bayes classifiers to text classification," *J of Inf Sci*, vol. 44, no. 1, pp.48-59, 2018