Wine Reviews
--- Text Analytics

Group 6 - Quanxu Pang, Yichen Pan, Ying Hu, Ziyue Zhong

Business Goal

Provide customers with appropriate wine selection strategies and generate business insights of the wine market

Data Description

- The Wine Reviews dataset contains 130k wine reviews with variety, location, winery, price, and description.
- Dataset Source: Kaggle.com
- Data Source: Wine Enthusiast (multichannel marketer of a growing line of wine- and spirits-related products
- Date updated: 11/24/2017
- Columns: country, description, designation, points, price, province, region, taster, variety, and winery

Data sample

	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Unnamed:	country	description	designatior	points	price	province	region_1	region_2	taster_nam	taster_twit	title	variety	winery
2	0	Italy	Aromas inc	Vulkà Bianc	87		Sicily & Sar	Etna		Kerin O'Kee	@kerinoke	Nicosia 201	White Blen	Nicosia
3	1	Portugal	This is ripe	Avidagos	87	15	Douro			Roger Voss	@vossroge	Quinta dos	Portuguese	Quinta dos
4	2	US	Tart and snappy, the fl		87	14	Oregon	Willamette	Willamette	Paul Gregu	@paulgwir	Rainstorm	Pinot Gris	Rainstorm
5	3	US	Pineapple r	Reserve La	87	13	Michigan	Lake Michigan Shore		Alexander Peartree		St. Julian 20	Riesling	St. Julian
6	4	US	Much like t	Vintner's R	87	65	Oregon	Willamette	Willamette	Paul Gregu	@paulgwir	Sweet Chee	Pinot Noir	Sweet Chee
7	5	Spain	Blackberry	Ars In Vitro	87	15	Northern S	Navarra		Michael Sc	@winescha	Tandem 20	Tempranill	Tandem
8	6	Italy	Here's a br	Belsito	87	16	Sicily & Sar	Vittoria		Kerin O'Kee	@kerinoke	Terre di Giu	Frappato	Terre di Giu
9	7	France	This dry and restrainec		87	24	Alsace	Alsace		Roger Voss	@vossroge	Trimbach 2	Gewürztrai	Trimbach
10	8	Germany	Savory drie	Shine	87	12	Rheinhessen			Anna Lee C. Iijima		Heinz Eifel	Gewürztrai	Heinz Eifel
11	9	France	This has gr	Les Nature	87	27	Alsace	Alsace		Roger Voss	@vossroge	Jean-Baptis	Pinot Gris	Jean-Baptis
12	10	US	Soft, supple	Mountain C	87	19	California	Napa Valle	Napa	Virginie Bo	@vboone	Kirkland Sig	Cabernet S	Kirkland Sig
13	11	France	This is a dry wine, very		87	30	Alsace	Alsace		Roger Voss	@vossroge	Leon Beyer	Gewürztrai	Leon Beyer
14	12	US	Slightly reduced, this w		87	34	California	Alexander '	Sonoma	Virginie Bo	@vboone	Louis M. M	Cabernet S	Louis M. M
15	13	Italy	This is dom	Rosso	87		Sicily & Sar	Etna		Kerin O'Kee	@kerinoke	Masseria S	Nerello Ma	Masseria S
16	14	US	Building on 150 years a		87	12	California	Central Coa	Central Coa	Matt Kettm	@mattkett	Mirassou 2	Chardonna	Mirassou
17	15	Germany	Zesty orang	Devon	87	24	Mosel			Anna Lee C. Iijima		Richard Bö	Riesling	Richard Bö
18	16	Argentina	Baked plun	Felix	87	30	Other	Cafayate		Michael Sc	@winescha	Felix Lavaq	Malbec	Felix Lavaq
19	17	Argentina	Raw black-	Winemaker	87	13	Mendoza P	Mendoza		Michael Sc	@winescha	Gaucho An	Malbec	Gaucho An
20	18	Spain	Desiccated	Vendimia S	87	28	Northern S	Ribera del Duero		Michael Sc	@winescha	Pradorey 2	Tempranill	Pradorey

Process Review

Data Preprossing

- Stemming
- POS-Tagging

1. Sentimental Analysis

- Business Insights using sentimental results

2. Topic Modeling

- LDA Model/ LSI Model

3. Keyword Extractor

- Skelearn and word2vestor package from Python

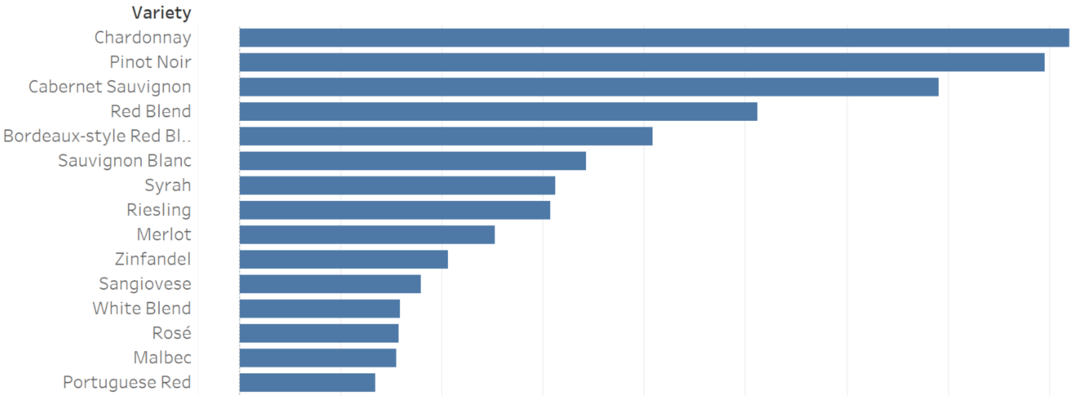
Sentiment Analytics

```
In [37]: scores.head(20)
Out[37]:
```

	compound	neg	neu	pos
0	0.1531	0.000	0.935	0.065
1	0.6486	0.000	0.868	0.132
2	-0.1280	0.053	0.947	0.000
3	0.3400	0.000	0.926	0.074
4	0.8176	0.000	0.805	0.195

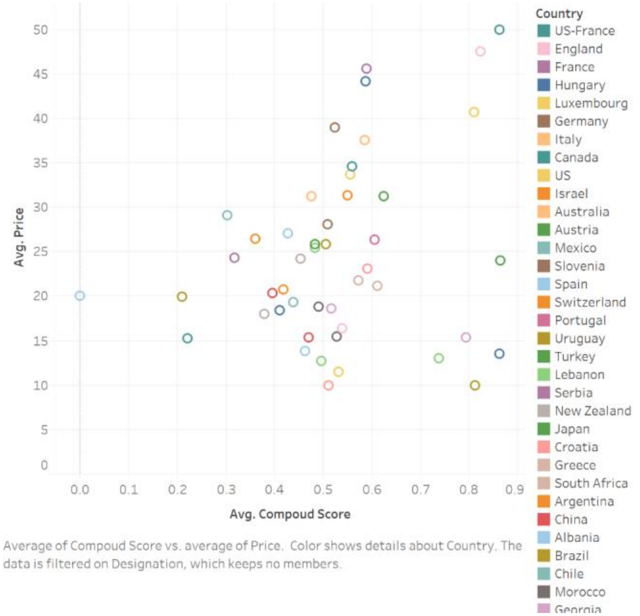
```
In [64]: avg_points = wine.groupby('points')['Compoud_Score'].mean()

In [65]: avg_points

Out[65]: points
80    -0.009855
81     0.043990
82     0.102415
83     0.190330
84     0.333103
85     0.406618
86     0.453449
87     0.492951
88     0.515306
89     0.530699
90     0.580637
91     0.600757
92     0.630100
93     0.661940
94     0.693255
95     0.745709
96     0.737685
97     0.796876
98     0.862329
99     0.881230
100    0.889621
Name: Compoud_Score, dtype: float64
```

Sentiment Analytics



- This chart shows the relationship between wine variety and its accumulative compound score, showing the top 15 variety
- Chardonnay, Pinot Noir, and Cabernet Sauvignon are the top three in the rank
- Ranking higher in this chart is more likely to be welcomed by most customers, and more likely to achieve better sale in market.

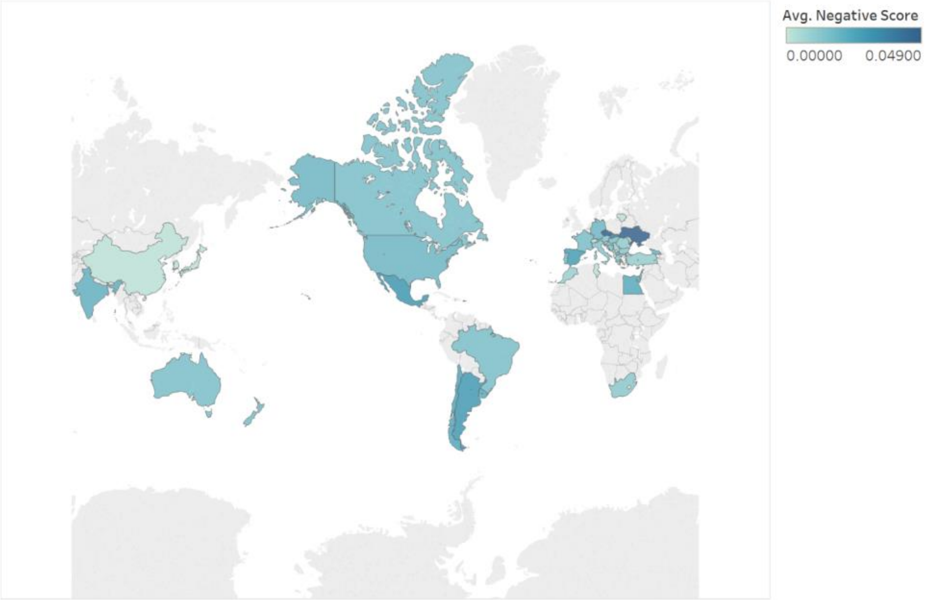
Sentiment Analytics



Average of Compoud Score vs. average of Price. Color shows details about Country. The data is filtered on Designation, which keeps no members.

- This chart shows the relationship between price and the compound score specified by colors according to different countries.
- We can get the cost performance of the wine in each country
- Wine from US-France, England and Luxembourg are of high price and high performance in score
- Wine from Japan, Ukraine, Slovakia, South Korea and Lithuania are of high performance and low price.
- They might be underpriced, there are more space for them to improve the marketing and pricing strategy.

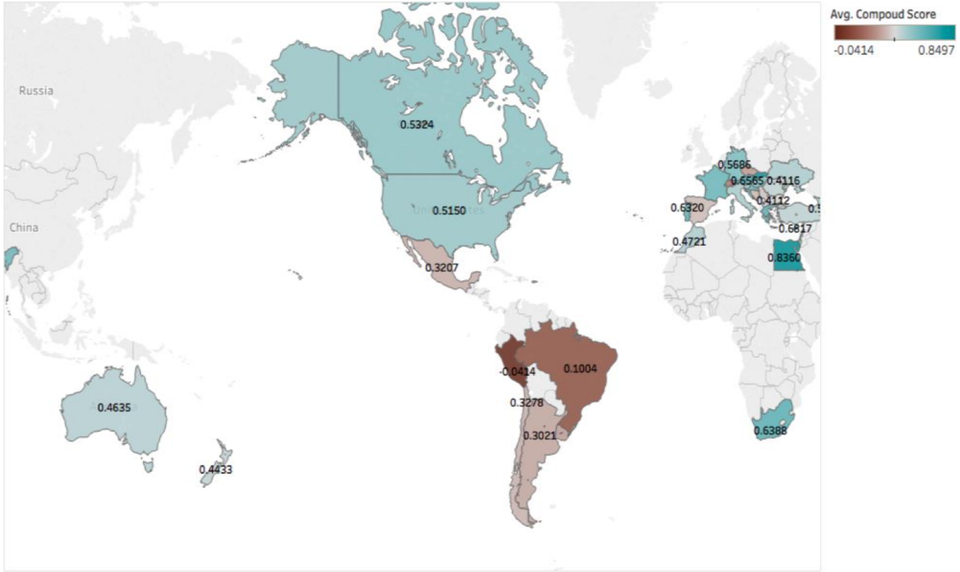
Sentiment Analytics



Map based on Longitude (generated) and Latitude (generated). Color shows average of Negative Score. Details are shown for Country. The data is filtered on Region 1, which keeps 1,237 of 1,237 members.

- This chart shows the negative score worldwide
- China is the region with lowest negative score which reached 0, Czech and Ukraine are the region with the highest negative score, and then follows Mexico and Argentina
- Wine from North America and Australia are a moderate level in Negative score, which can be inferred their wine's review and quality are of a stable level.
- The wine from east Europe and South America requires more attentions when selecting.

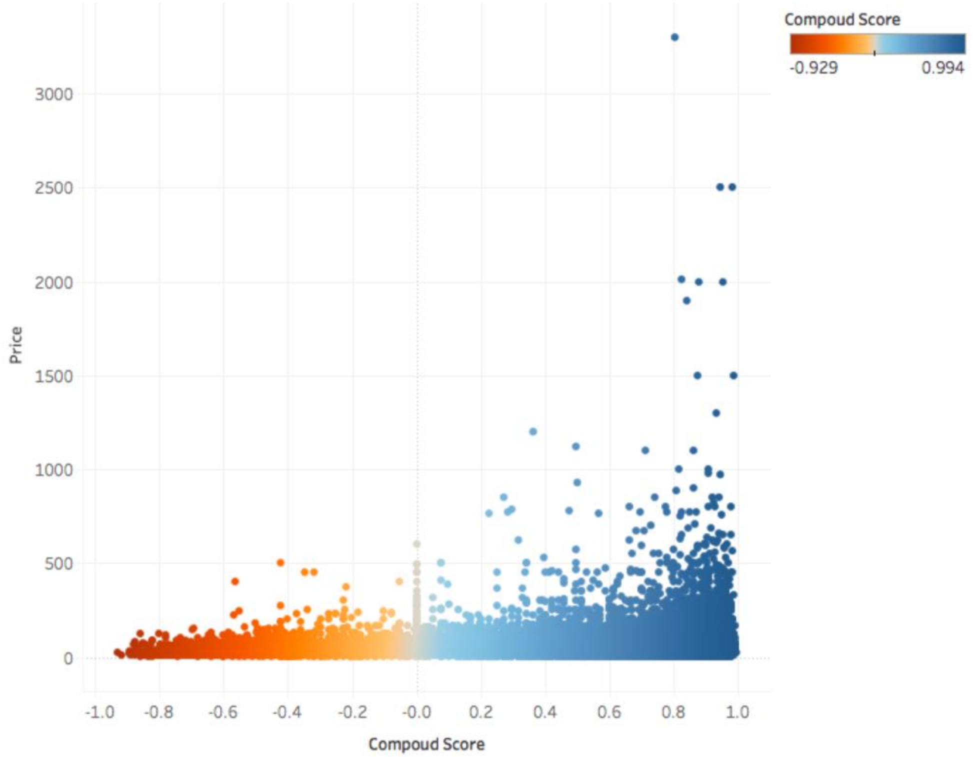
Sentiment Analytics



Map based on Longitude (generated) and Latitude (generated). Color shows average of Compoud Score. The marks are labeled by average of Compoud Score. Details are shown for Country. The view is filtered on Country, which excludes China.

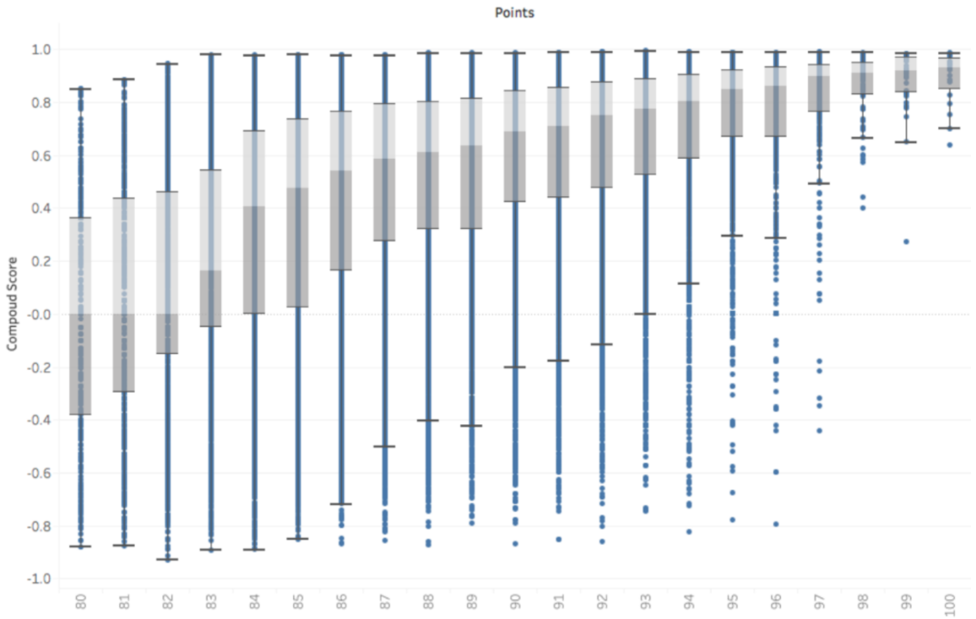
- Average compound sentimental score of wine in different regions.
- Egypt has the highest compound score. South Africa, U.S., Canada and part of European countries also have high-quality wines. But wine from Latin America have relatively low compound scores implying low quality.
- geographic factor can have great impact on the quality of wine.

Sentiment Analytics



- Wine which has a price within the range from 0 to 500 do not have significant association with its quality.
- But for those wine whose price are higher than 500, they all have positive sentimental score.

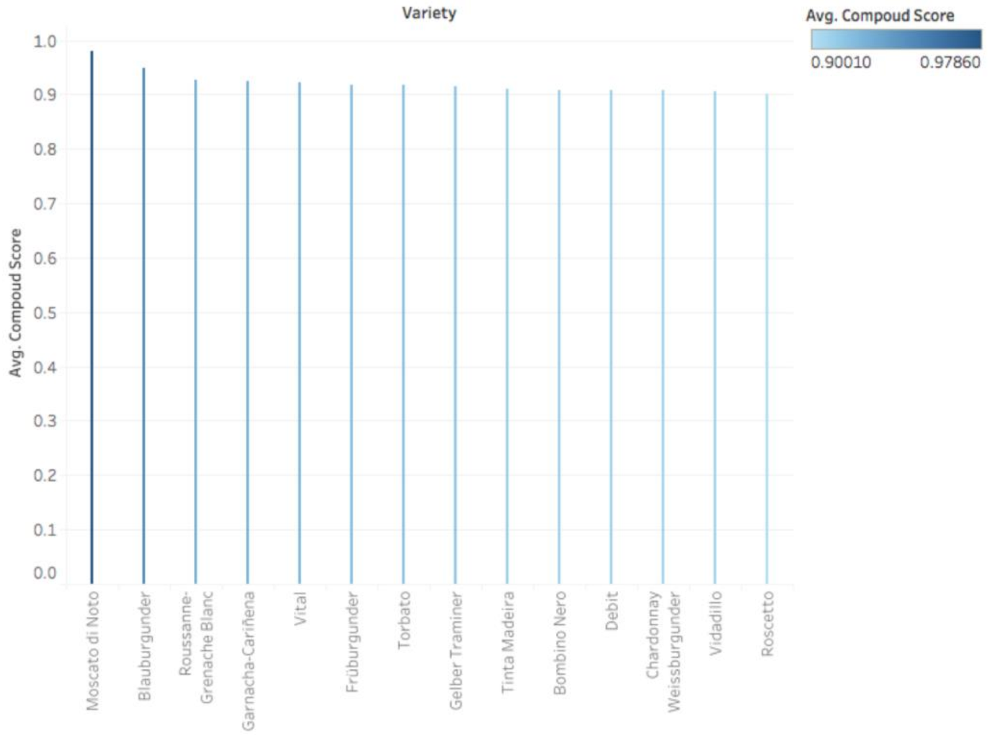
Sentiment Analytics



Compoud Score for each Points.

- There is a strong positive correlation between compound score and points.
- For wine with lower point, their scale is larger compared to wine with high score.
- Business should pay attention to the wine that have low points but high sentimental score, because this may imply a low price with high quality.

Sentiment Analytics



- This figure shows all the wine name that have compound higher higher than 0.9.
- This Ranking can help customers to quickly compare different type of wines.

Topic modeling

a. LDA model

```
In [11]: lda = models.LdaModel(corpus, id2word=dictionary, num_topics=10) #fit lda model

         lda.print_topics(10) #V matrix, topic matrix

Out[11]: [(0,
           '0.020*"flavors" + 0.016*"palate" + 0.013*"wine" + 0.012*"notes" + 0.012*"finish" + 0.012*"aromas" + 0.012*"fruit" + 0.008*"w
         hite" + 0.008*"acidity" + 0.008*"rich"'),
          (1,
           '0.020*"wine" + 0.018*"flavors" + 0.016*"aromas" + 0.016*"fruit" + 0.015*"finish" + 0.012*"acidity" + 0.012*"palate" + 0.010
         *"drink" + 0.010*"sweet" + 0.008*"apple"'),
          (2,
           '0.024*"wine" + 0.023*"fruit" + 0.022*"flavors" + 0.018*"palate" + 0.015*"aromas" + 0.012*"acidity" + 0.011*"finish" + 0.009
         *"cherry" + 0.008*"white" + 0.008*"black"'),
          (3,
           '0.040*"wine" + 0.017*"flavors" + 0.016*"aromas" + 0.014*"acidity" + 0.014*"drink" + 0.014*"fruit" + 0.013*"tannins" + 0.013
         *"ripe" + 0.011*"finish" + 0.008*"palate"'),
          (4,
           '0.017*"wine" + 0.016*"finish" + 0.015*"fruit" + 0.014*"notes" + 0.014*"fresh" + 0.014*"palate" + 0.013*"flavors" + 0.012*"ar
         omas" + 0.011*"acidity" + 0.011*"nose"'),
          (5,
           '0.024*"wine" + 0.019*"drink" + 0.016*"acidity" + 0.015*"palate" + 0.015*"ripe" + 0.014*"flavors" + 0.011*"fruit" + 0.011*"fi
         nish" + 0.010*"aromas" + 0.008*"nose"'),
          (6,
           '0.021*"palate" + 0.017*"cherry" + 0.017*"wine" + 0.014*"aromas" + 0.014*"black" + 0.013*"fruit" + 0.011*"finish" + 0.011*"fl
         avors" + 0.010*"notes" + 0.010*"nose"'),
          (7,
           '0.033*"wine" + 0.016*"acidity" + 0.015*"tannins" + 0.015*"ripe" + 0.013*"drink" + 0.011*"fruit" + 0.011*"palate" + 0.010*"sp
         ice" + 0.010*"flavors" + 0.010*"rich"'),
          (8,
           '0.018*"drink" + 0.016*"berry" + 0.016*"flavors" + 0.015*"aromas" + 0.013*"palate" + 0.013*"tannins" + 0.012*"ripe" + 0.010
         *"red" + 0.010*"finish" + 0.009*"wine"'),
          (9,
           '0.033*"flavors" + 0.023*"wine" + 0.019*"aromas" + 0.017*"fruit" + 0.015*"black" + 0.010*"cabernet" + 0.010*"cherry" + 0.009
         *"rich" + 0.008*"dry" + 0.008*"finish"')]
```

- Although, the content is all about wine, we still can use topic modeling to cluster descriptions.
- Most of topics contain wine, flavors, aromas, and acidity
- There are some difference based on wines’ category among all ten topics such as….

Keyword extractor

In [166]:
```
# show final tfidf feature matrix
display_features(np.round(norm_tfidf, 2), feature_names)
```

```
     100  1200   122    18  1961    20  2006  2015  2020  2022  ...  wine  \
0   0.15  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.13  ...  0.07
1   0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.00
2   0.00  0.00  0.18  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.08
3   0.00  0.00  0.00  0.00  0.00  0.16  0.00  0.00  0.00  0.00  ...  0.00
4   0.00  0.15  0.00  0.12  0.00  0.00  0.00  0.00  0.15  0.00  ...  0.21
5   0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.00
6   0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.00
7   0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.00
8   0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.09
9   0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.09
10  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.00
11  0.00  0.00  0.00  0.16  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.09
12  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.19  0.00  0.00  ...  0.00
13  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.22
14  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.00
15  0.00  0.00  0.00  0.15  0.00  0.00  0.19  0.00  0.00  0.00  ...  0.00
16  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.09
17  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.08
18  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.13  ...  0.21
19  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  ...  0.09
20  0.00  0.00  0.00  0.00  0.21  0.00  0.00  0.00  0.00  0.00  ...  0.00

    winethat  winner  with  wood  year  years  yellow   you  zest
0       0.00    0.00  0.00  0.00  0.00    0.3    0.00  0.00  0.00
1       0.00    0.00  0.08  0.00  0.00    0.0    0.00  0.00  0.00
2       0.00    0.00  0.00  0.00  0.00    0.0    0.00  0.00  0.00
3       0.00    0.00  0.07  0.00  0.00    0.0    0.00  0.00  0.00
4       0.00    0.00  0.06  0.13  0.00    0.0    0.00  0.00  0.00
5       0.00    0.18  0.00  0.00  0.00    0.0    0.00  0.00  0.00
6       0.00    0.00  0.07  0.00  0.00    0.0    0.00  0.00  0.00
7       0.00    0.00  0.00  0.00  0.00    0.0    0.00  0.00  0.00
8       0.00    0.00  0.08  0.00  0.00    0.0    0.00  0.19  0.00
9       0.00    0.00  0.00  0.00  0.00    0.0    0.00  0.00  0.00
```

- We use “feature_extraction” function from Sklearn package and word2vector from genism
- Our target is to get three most meaningful words which stand for a description.
- We calculate TFIDF value of each word. Then we select the top three meaningful words based on TFIDF value.
- This is a demo by using 20 description of wine

Keyword extractor

AP	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ
blackberry	coffee	concentrat	hazelnut	minty	oak	orange	pear	subtle	sweetenec	vanilla
0	0	0	0	3.397895	2.481605	0	0	2.704748	0	0
2.145132	0	0	0	0	0	0	0	0	3.397895	2.299283
0	0	0	3.397895	0	0	2.481605	2.99243	0	0	0
2.145132	2.99243	0	0	0	2.481605	0	0	0	0	0
0	0	2.704748	0	0	0	0	0	0	0	0
2.145132	2.99243	0	0	0	2.481605	0	0	0	0	0
2.145132	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	2.299283
0	0	2.704748	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2.481605	0	0	0	0
0	0	0	0	0	0	0	2.99243	0	0	0
0	0	0	0	0	0	0	0	0	0	2.299283
2.145132	0	0	0	0	0	0	0	2.704748	0	2.299283
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2.704748	0	0
0	0	0	0	0	2.481605	0	0	0	0	0
4.290265	0	0	0	0	0	0	0	0	0	2.299283
0	0	2.704748	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2.481605	0	0	0	0
0	0	0	0	0	0	2.481605	0	0	0	0

- 1. minty, subtle, aged
- 2. vanilla, sweetened, blackberry
- 3. pear, orange, hazelnut
- 4. blackberry, oak, coffee
- 5. concentration, aging, acidity

- Limitation: we only use 20 descriptions. The corpus is not huge enough to generate reliable keywords.

Text Similarity

- This figure shows our codes for calculating Text Similarity basically using ‘gensim’ and ‘numpy’ modules.
- Main thought is transferring documents to vectors in the convenience of calculating.
- Accrossing to change the training text and testing text, we could select specific wines that customers are willing to know.

```python
TaggededDocument = gensim.models.doc2vec.TaggedDocument
puctuation = [',', '.', '?', '!', '@', '%', ':', ';']

def get_dataset():
    with open('training10k.txt','r', encoding = 'utf-8') as wine_review1:
        docs = wine_review1.readlines()
        for p in puctuation:
            docs = [d.replace(p, ' ')for d in docs]

    x_train = []
    try:
        for a, b in enumerate(docs):
            doc_list = b.split('\n')
            l = len(doc_list)
            doc_list[l-1] = doc_list[l-1].strip()
            document = TaggededDocument(doc_list, tags=[a])
            x_train.append(document)
            return x_train
            wine_review.close()
    except UnicodeDecodeError:
        return x_train
        wine_review.close()

def getVecs(model, corpus, size):
    vecs = [np.array(model.docvecs[z.tags[0]].reshape(1, size)) for z in corpus]
    return np.concatenate(vecs)

def train(x_train, size=200, epoch_num=1):
    model_dm = Doc2Vec(x_train,min_count=1, window = 3, size = size, sample=1e-3, negative=5, workers=4)
    model_dm.train(x_train, total_examples=model_dm.corpus_count, epochs=70)
    model_dm.save('model_dmtraining')

    return model_dm

def test():
    model_dm = Doc2Vec.load('model_dmtraining')
    wine_review2 = open('testing10k.txt','r')
    text = wine_review2.readlines()
    for p in puctuation:
            text = [t.replace(p, ' ')for t in text]
    test_text = text[int(input('Please input number of the line you want to test.'))]
    inferred_vector_dm = model_dm.infer_vector(test_text)
##    print (inferred_vector_dm)
    sims = model_dm.docvecs.most_similar([inferred_vector_dm], topn=10)


    return sims
    wine_review2.close()

if __name__ == '__main__':
    x_train = get_dataset()
    model_dm = train(x_train)
    sims = test()
    for count, sim in sims:
        docnum = x_train[count]
        sens = ''
        for sen in docnum[0]:
            sens = sens + sen + ' '
        print (sens, sim)
```

Text Similarity

- If a customer is going to make investment on wine, we could test the descriptions similarity among all kinds of wine. There are two kinds of wine, one is at high price and the other low-price wine’s description shows high similarity with the high-price one.

```
>>>
 RESTART: C:\Users\panch\Downloads\text analytics\PROJECT\wine-reviews\text_proj
ect.py
Please input number of the line you want to test.5
This tremendous 100  varietal wine hails from Oakville and was aged over three y
ears in oak  Juicy red-cherry fruit and a compelling hint of caramel greet the p
alate  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 develo
p further nuance  Enjoy 2022â€"2030   0.8901684880256653
>>>
```

Conclusion and Future Direction

- Based on our findings, customers can accelerate their study on unknowing fields by developing core knowledge. Companies can offer customers more usable tools, reliable information and humanized user interface to improve user experience and improve company reputations.
- For future direction, the most significant part would be gaining usable cleaning data. Generating investing opportunities would be more plausible if we drill down to wine prices. Trying other vector models and testing out their performance is also a big challenge.

Reference

Bansal, Shivam, et al. “Beginners Guide to Topic Modeling in Python.” *Analytics Vidhya*, 29 Aug. 2016, www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/.

“Sklearn.feature_extraction.Text.TfidfVectorizer¶.” *Sklearn.feature_extraction.Text.TfidfVectorizer - Scikit-Learn 0.19.1 Documentation*, scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.