Wine Recommendation

Capstone #2 Slide Deck

Introduction

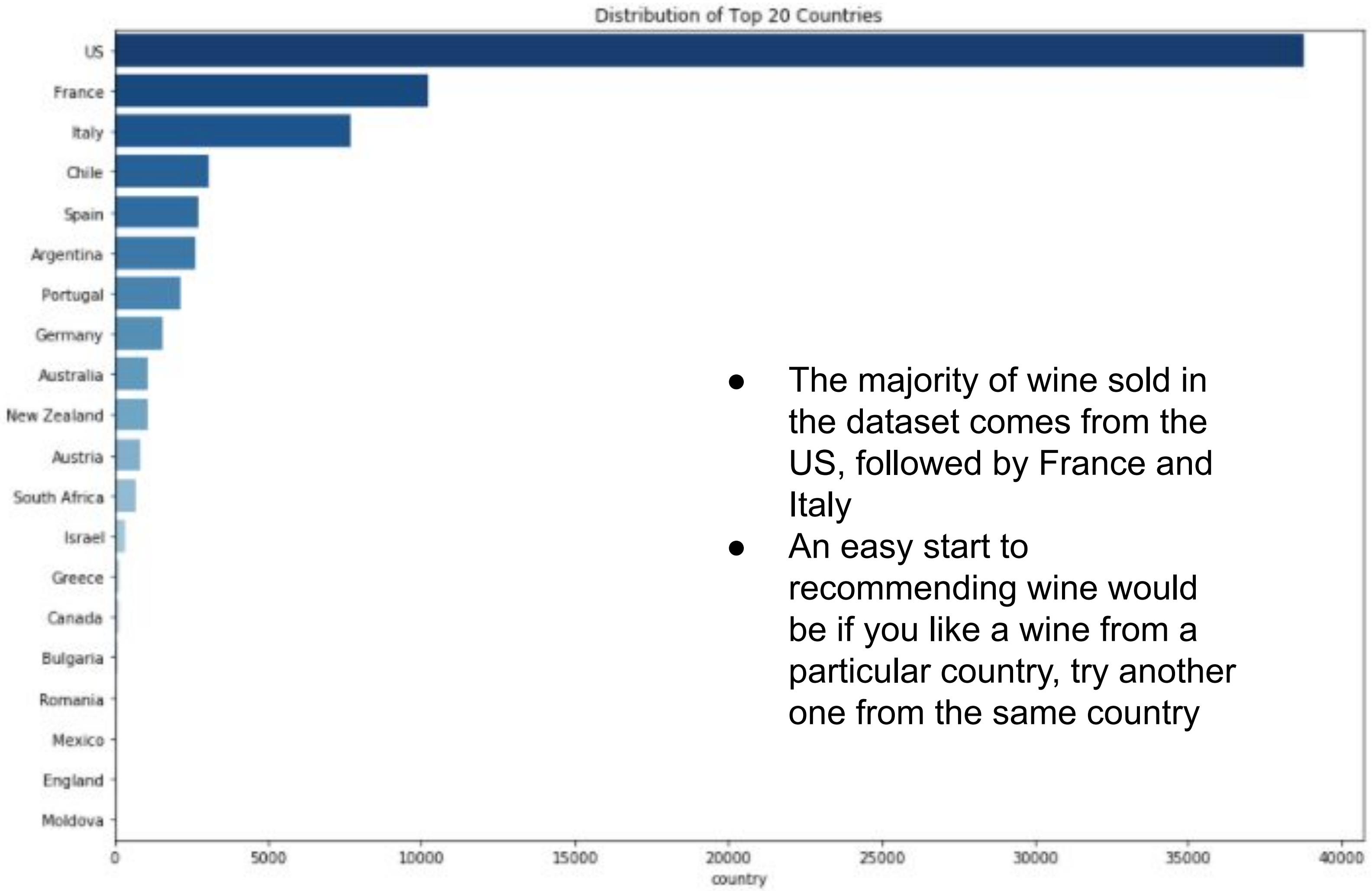
- Wine can be overwhelming to those who don’t consume often
- There are countless wineries, varieties, locations to buy from, etc., all of which have unique characteristics, so finding new wine that you like can be difficult

Data

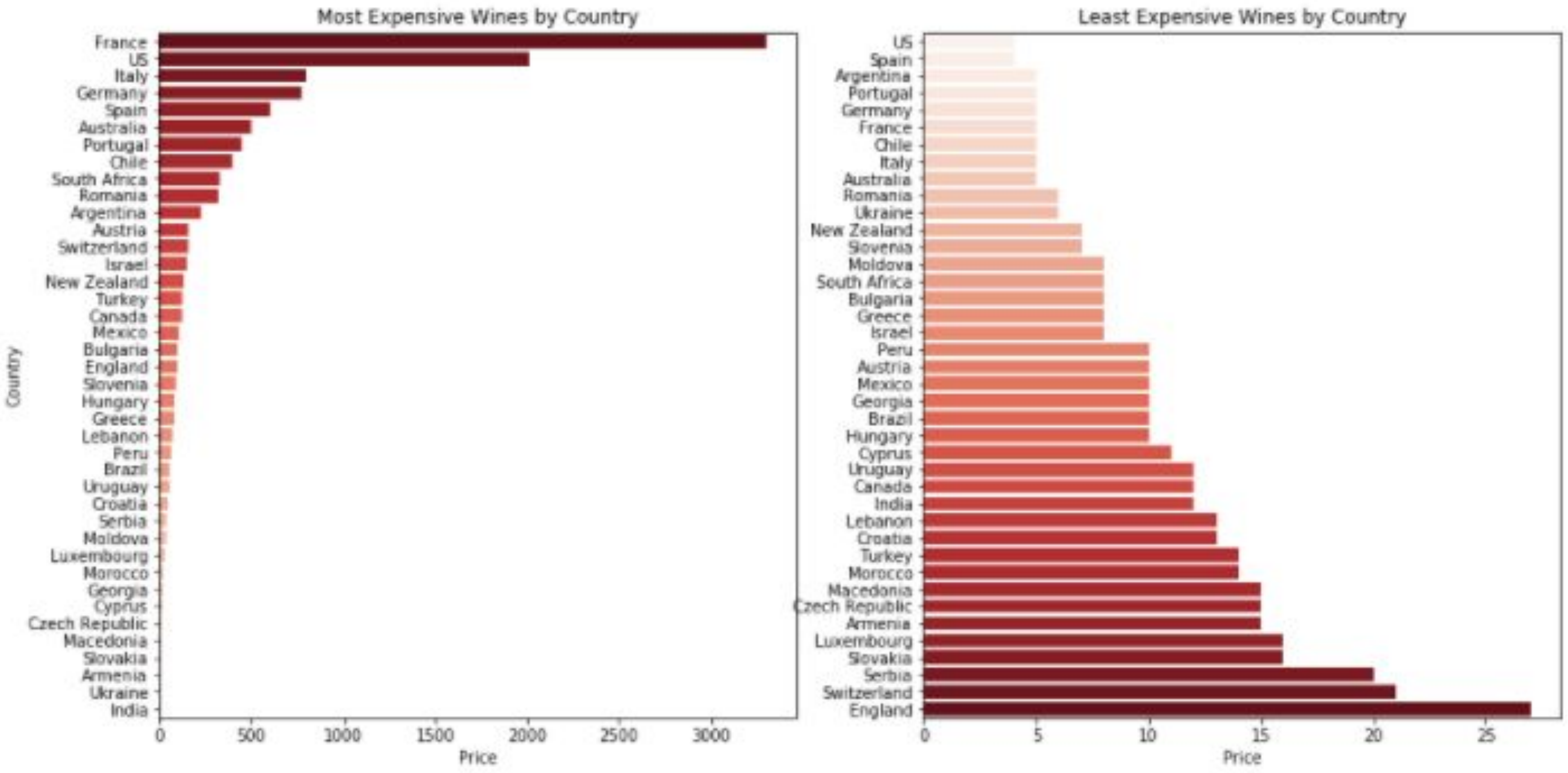
- https://www.kaggle.com/zynicide/wine-reviews
- Contains various attributes of about 280,000 wines, such as price, points, country of origin, variety, vineyard, and a brief sommelier description
- There were a large number of null values in columns like taster name, twitter handle, and region, so I decided to drop those columns
- There were also a large number of varieties in the dataset, so I decided to focus on the top 20 for my analysis

Problem

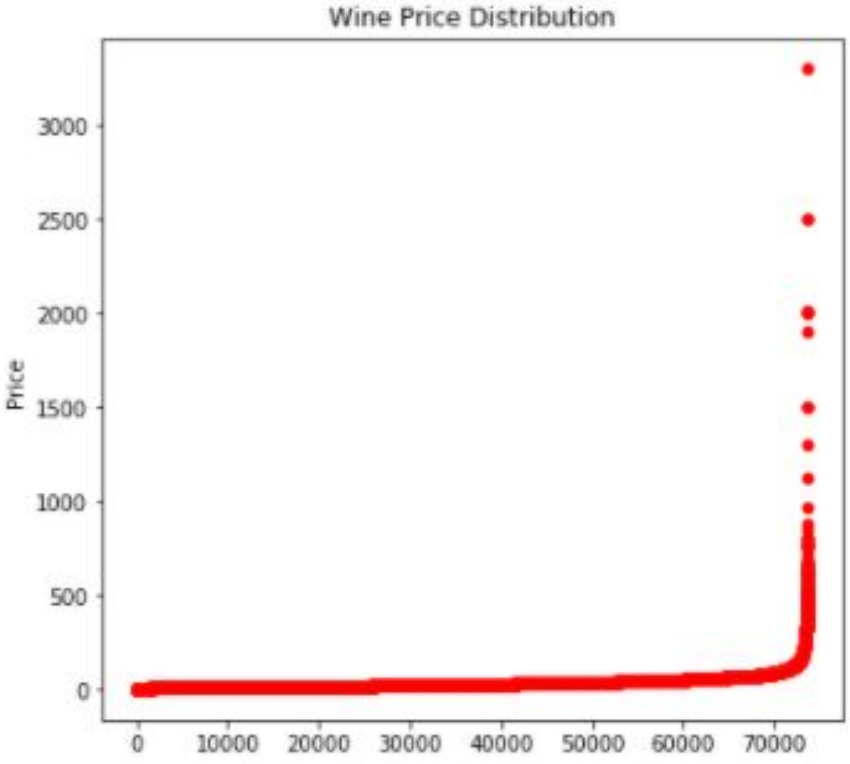
Using the wine data, I will look for trends in order to provide accurate recommendations for users looking to branch out and try different types of wine. I will do this through exploratory data analysis, inferential statistics, and machine learning algorithms to build the recommendation tools.

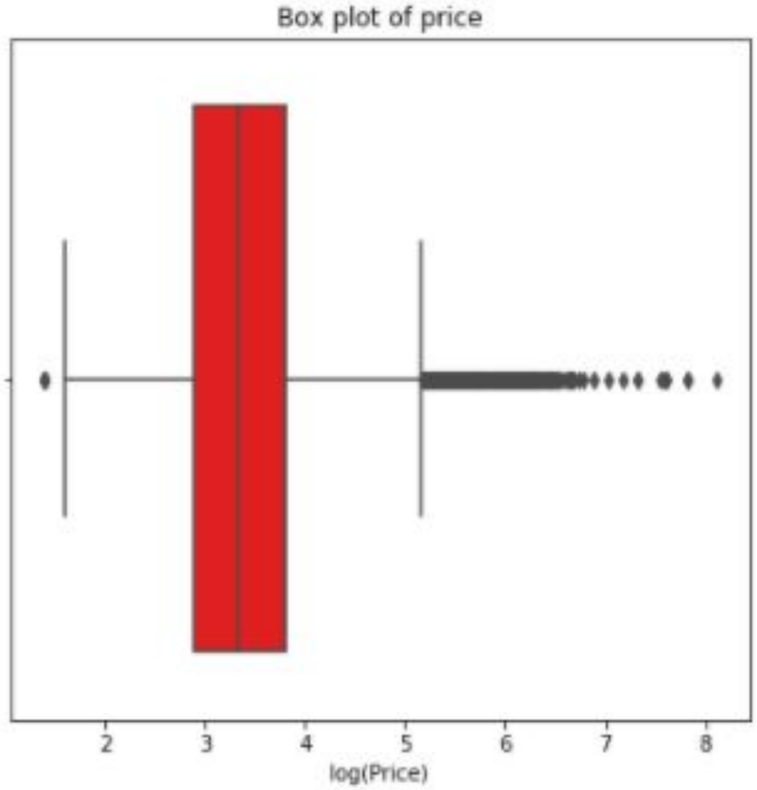


- The majority of wine sold in the dataset comes from the US, followed by France and Italy
- An easy start to recommending wine would be if you like a wine from a particular country, try another one from the same country

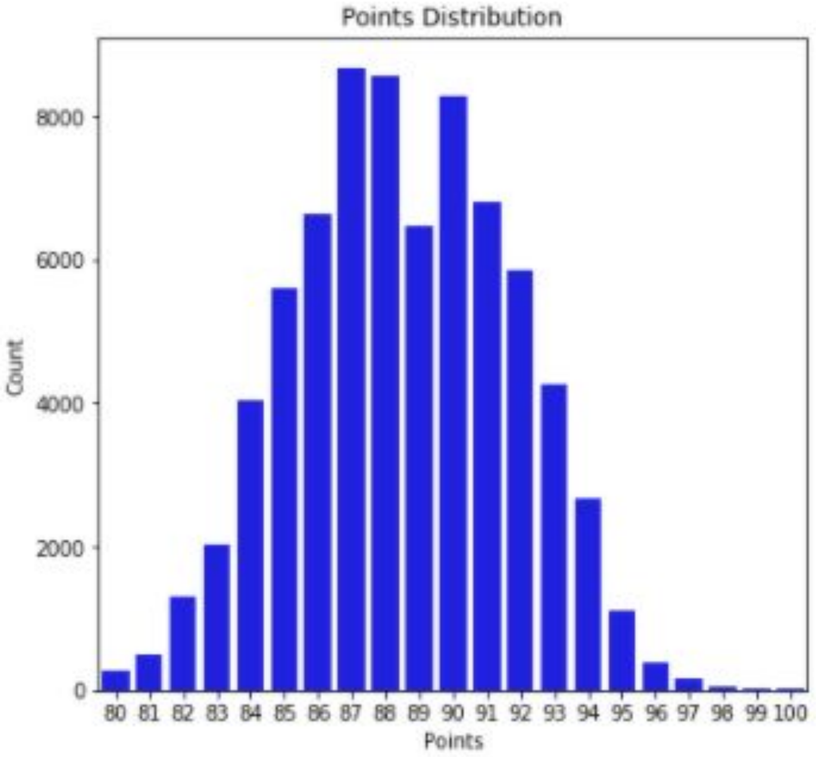


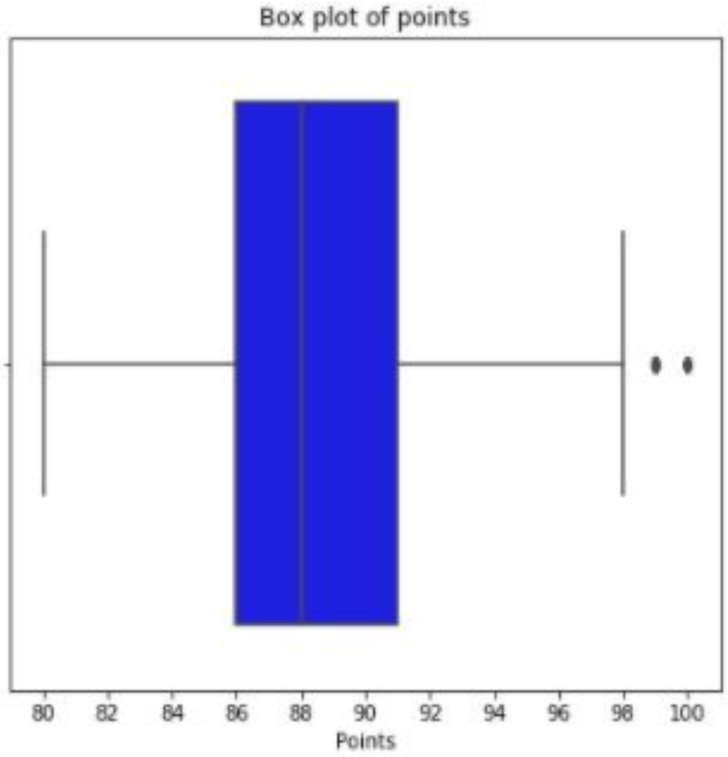
- There are similar groups of countries atop the lists of most expensive wine sold and least expensive wine sold, leading me to believe there is a great price range of wines sold in countries like the US, France, Portugal, Argentina, and Italy





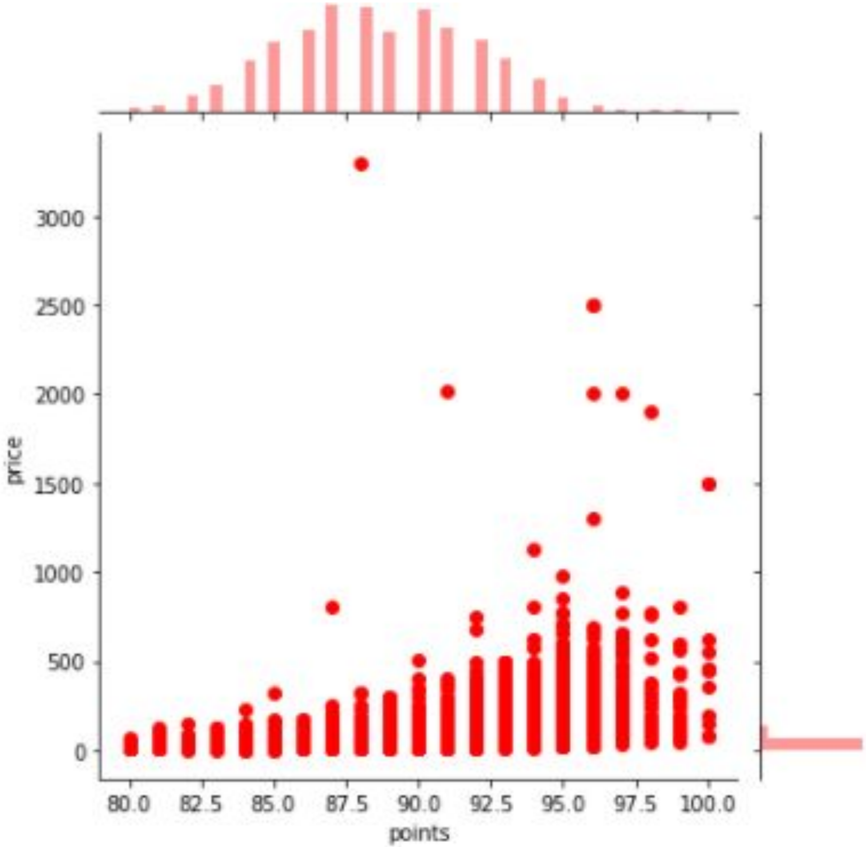
- There are no lower outliers and 751 upper outliers out of 73,691 observations, representing only about 1.03% of the data



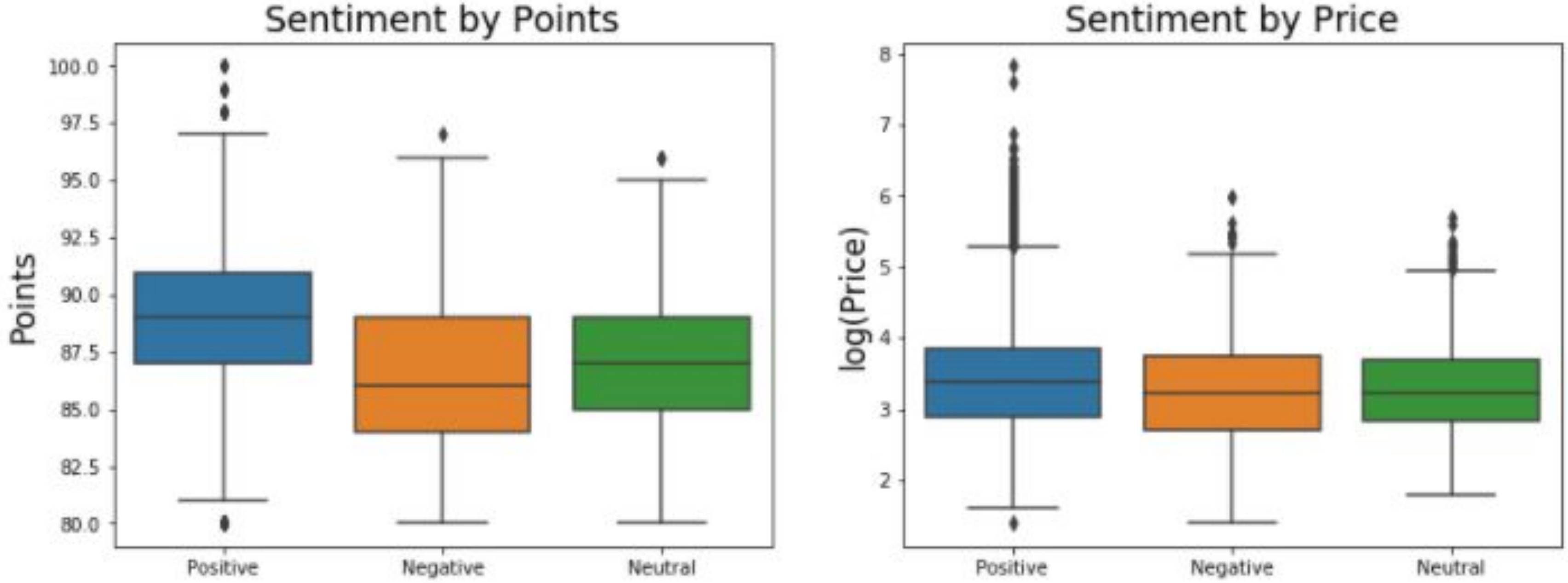


- There are no lower outliers and 35 upper outliers, representing only about 0.05% of total observations

Correlation between Price and Points



- Points and price look to be positively correlated, though there are some exceptions on both ends, with some relatively cheap highly-rated wines as well as some expensive low-rated wines
- Interestingly, the highest priced wine in the dataset at $3,300 was only given a rating of 88, which is around the mean for the points column



- As expected, there is a more positive sentiment among higher rated wines and a more negative sentiment among lower rated wines

- There is less variation on sentiment when grouping by price, further indicating the possibility of finding suitable wines that are also affordable

Chi-Square Test

H_o: Points and Price are independent.

H_a: Points and Price are not independent.

price_range quality	1-30	100+	31-60	61-100
bad	12037	26	1505	184
good	4019	1721	10356	4584
great	7	324	100	219
ok	25327	385	10941	1956

Test statistic = 25,185.857
Critical value = 16.919
P-value = 0.000
Alpha = 0.05

- The test statistic is greater than the critical value, and the p-value is less than alpha, so we reject the null hypothesis that points and price are independent

Machine Learning

- Created a vector representation of the sommelier description using CountVectorizer

- Split the data into a training and test set with the features of the vectors as X and the wine quality categorical variable as y

- Fit a logistic regression model to the training set, and obtained a decent accuracy score of the test set about 71%

- Running this same model, but with grape variety as y yielded an accuracy score of about 68%, so slightly less effective

Tool #1 - Recommending a Specific Wine

- I created a TfidfVectorizer object and fitted it to the description column of the training set
- Then, I used the linear_kernel tool to create a matrix containing the cosine similarities of each description
- I populated an empty dictionary with wine ID as the key and a list of similar wines based on the cosine similarity as values
- Then, I wrote simple functions to extract the wine ID and give recommendations based on the given ID

Tool #1 - Recommending a Specific Wine

```
In [92]: test_id = train.loc[:, 'wine_id'].values[23]
         test_title = train.loc[train.wine_id == test_id, 'title'].values[0]

         get_recommendation(test_id, 8)
```

```
Top 8 recommendations for H. Abrantes Douro Wines 2011 Vargosa Red (Douro):

Château Beaulieu 2011 Château Beaulieu Rosé (Coteaux d'Aix-en-Provence): score
= 0.05
Maison des 3 Ponts 2014 Lepontis Sauvignon Blanc (Charentais): score = 0.04
Sineann 2012 Red (Oregon): score = 0.04
Château Gauthier 2015  Blaye Côtes de Bordeaux: score = 0.04
Olivier Leflaive 2012 Abbaye de Morgeot Premier Cru  (Chassagne-Montrachet): sc
ore = 0.04
Kastania 2012 Jaden and Keira's Cuvée Pinot Noir (Sonoma Coast): score = 0.04
Louis Sipp NV Brut Sparkling (Crémant d'Alsace): score = 0.04
Château Troplong Mondot 2007  Saint-Émilion: score = 0.04
```

Tool #2 - Recommending Variety with KNN

- Dropped all of the columns from the dataset besides province, variety, points, price

- Created a pivot table from the dataframe with variety as the index, province as the columns, and points and price as the values

- Created a csr_matrix from the pivot table, and fit a nearest neighbors model to the matrix

- Using 10 neighbors, I chose a random wine from the pivot table and output a list of recommended varieties based on the distance from the csr_matrix

Tool #2 - Recommending Variety with KNN

In [121]:

```
rand_idx = np.random.choice(wine_pivot.shape[0])
dist, idx = m_knn.kneighbors(wine_pivot.iloc[rand_idx, :].values.reshape(1, -1), n_neighbors = 10)

for i in range(0, len(dist.flatten())):
    if i == 0:
        print('Top recommendations for ' + wine_pivot.index[rand_idx])
    else:
        print(str(i) + ' ' + wine_pivot.index[idx.flatten()[i]] + ' with distance ' + str(dist.flatten()[i]))
```

```
Top recommendations for Nebbiolo
1 Sangiovese with distance 0.5203008843908599
2 Tempranillo with distance 0.6931693010858284
3 Sparkling Blend with distance 0.709833663795427
4 Red Blend with distance 0.7737772438890955
5 Zinfandel with distance 0.7767075492995374
6 White Blend with distance 0.7806166873152927
7 Chardonnay with distance 0.7929535435480267
8 Malbec with distance 0.806316613795247
9 Pinot Gris with distance 0.8148184280321505
```