Wine
Price & Quality Prediction

Thinkful Supervised Learning Capstone

Dataset Overview

- Scraped from Wine Enthusiast Magazine's website by Zach Thoutt, and posted on Kaggle: https://www.kaggle.com/zynicide/wine-reviews
- Thoutt's goal with the dataset was to create a wine identifier based on description.
- Three datasets, one of which is an expanded version of one of the others.
- We will use "winemag-data-130k-v2.json", which includes "title" for the wine reviews.
- Our model will be to predict price and/or rating based on origin, variety, and description of a wine.

Process Outline

- Initial exploratory data analysis
- Data cleaning/feature engineering
  - Normalization of continuous variables
  - Origin/region categorization
  - Variety/type categorization
  - Analysis of collinearity
  - Dimensionality reduction
- Regression Modeling for Price and Points
  - Attempted models
- Classification Modeling for Price and Points
  - Final models
  - Hyperparameter tuning

Limitations/Unavailable Data

- No variable for alcohol by volume. % ABV is also included on wine labels, so that information would be available to the intended users of these models.
- "Points" ratings in this set are all between 80 and 100 points.
- Ratings are all subjective, as are the individual descriptions by the sommeliers.
- The 116603 reviews in the dataset come from only 19 sommeliers – this could lead to bias in the sorts of adjectives used in wine descriptions.

Python Libraries Used

- Numpy, Pandas, Scipy
  - Data manipulation
- Matplotlib, Seaborn
  - Visualization
- Re, Unidecode
  - Text parsing
- Sklearn
  - Preprocessing
  - Regression & Classification Modeling
  - Model validation

Exploratory Data Analysis

Data Columns

- country - The country that the wine is from
- description - A few sentences from a sommelier describing the wine's taste, smell, look, feel, etc.
- designation - The vineyard within the winery where the grapes that made the wine are from
- points - The number of points WineEnthusiast rated the wine on a scale of 1-100 (though they say they only post reviews for wines that score >=80)
- price - The cost for a bottle of the wine
- province - The province or state that the wine is from
- taster_name - Name of the person who tasted and reviewed the wine
- taster_twitter_handle - Twitter handle for the person who tasted and reviewed the wine
- 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
- variety - The type of grapes used to make the wine (ie Pinot Noir)
- winery - The winery that made the wine

Dataframe Head

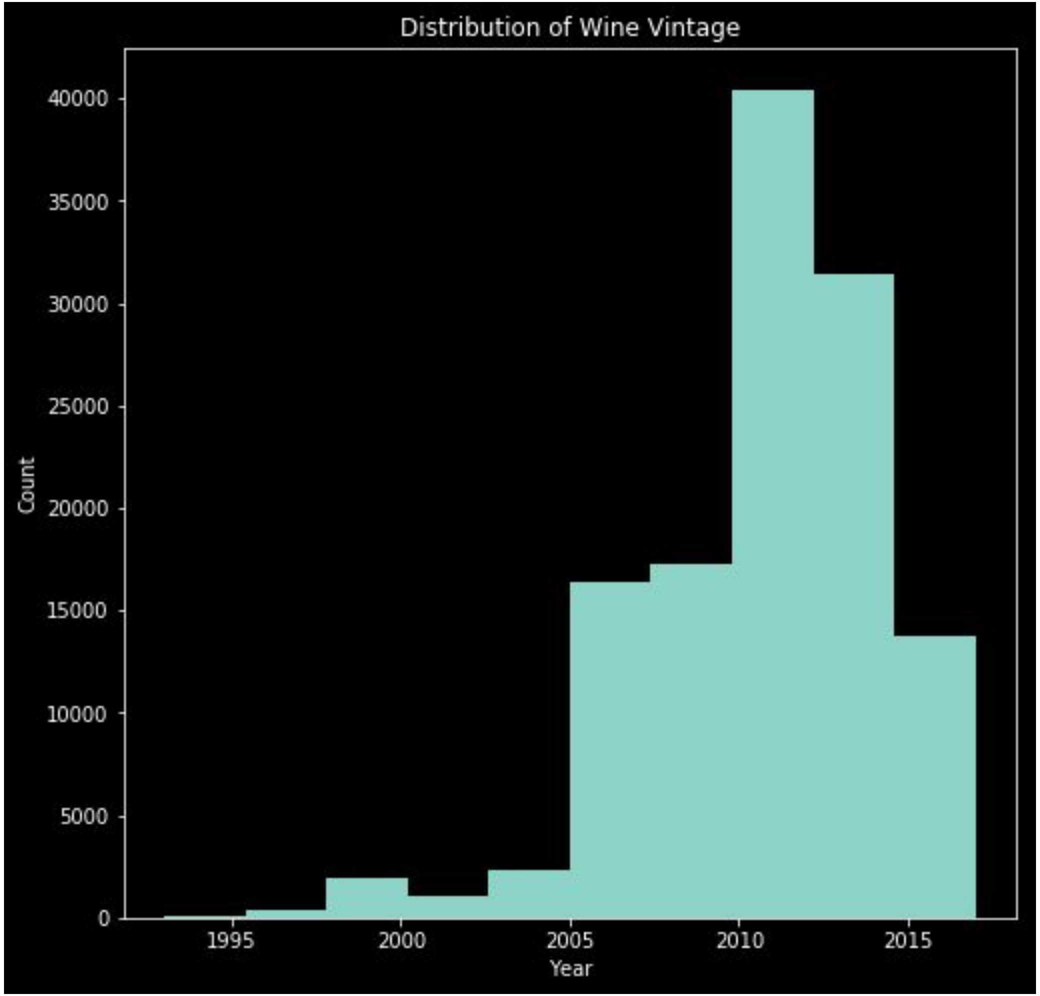
	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	title	variety	winery
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	None	Kerin O'Keefe	@kerinokeefe	Nicosia 2013 Vulkà Bianco (Etna)	White Blend	Nicosia
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	None	None	Roger Voss	@vossroger	Quinta dos Avidagos 2011 Avidagos Red (Douro)	Portuguese Red	Quinta dos Avidagos
2	US	Tart and snappy, the flavors of lime flesh and...	None	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Rainstorm 2013 Pinot Gris (Willamette Valley)	Pinot Gris	Rainstorm
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	None	Alexander Peartree	None	St. Julian 2013 Reserve Late Harvest Riesling ...	Riesling	St. Julian
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Sweet Cheeks 2012 Vintner's Reserve Wild Child...	Pinot Noir	Sweet Cheeks

Excluded Variables

- Region_2
  - Many wine reviews have a NaN value for this field.
  - This is too granular a specification for location.
- Taster_name
  - It’s possible that some reviewers could have biased opinions of particular wines, wineries, countries of origin, etc., but we want our models to be as generally applicable as possible.
  - Reviewers’ names very seldom appear on wine labels.
- Taster_twitter_handle
  - Redundant (effectively the same as taster_name).

Vintage

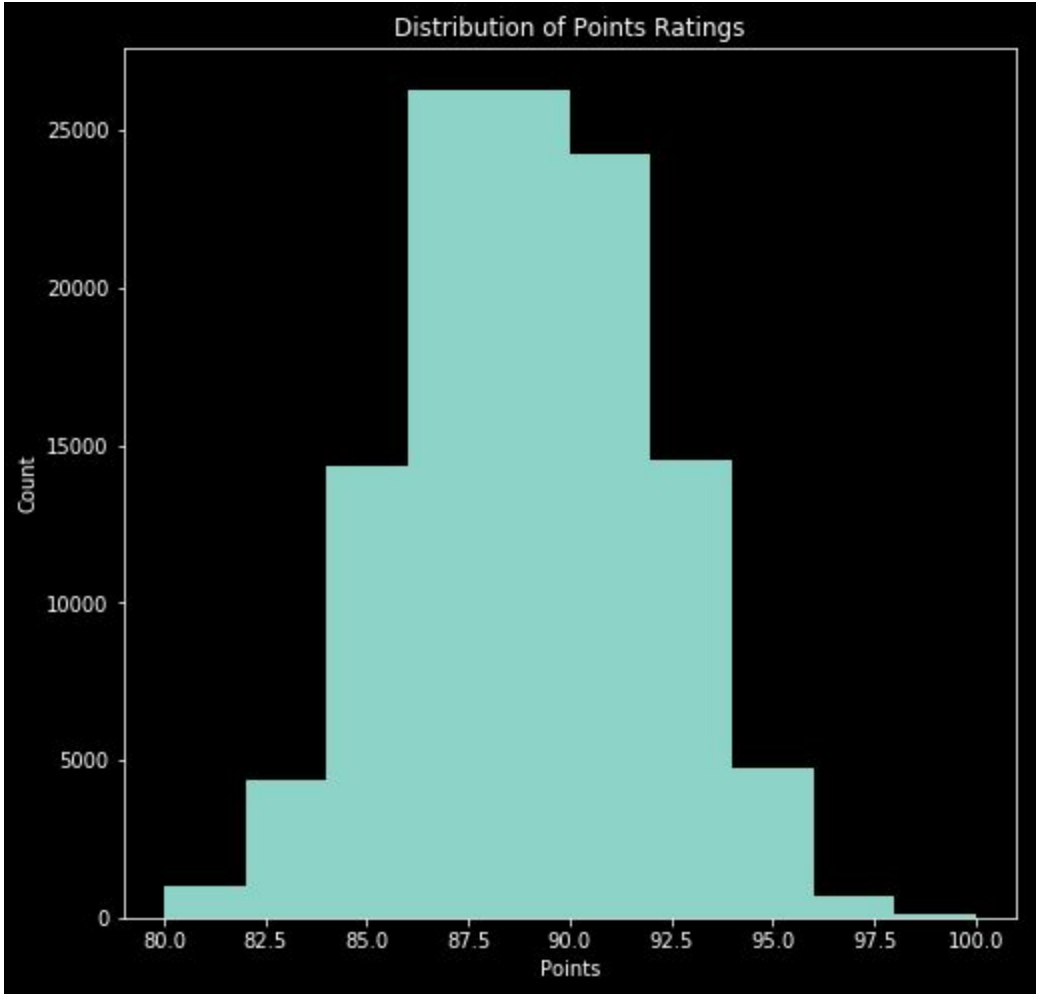
- Vintage isn't a feature on its own; it's usually included in "title."
- Some year information refers to other chronological information:
  - Year winery was founded
  - Age of grape vines
  - Age of barrels for aging sherry/port
- Regex used to pull most recent year from title.



Points

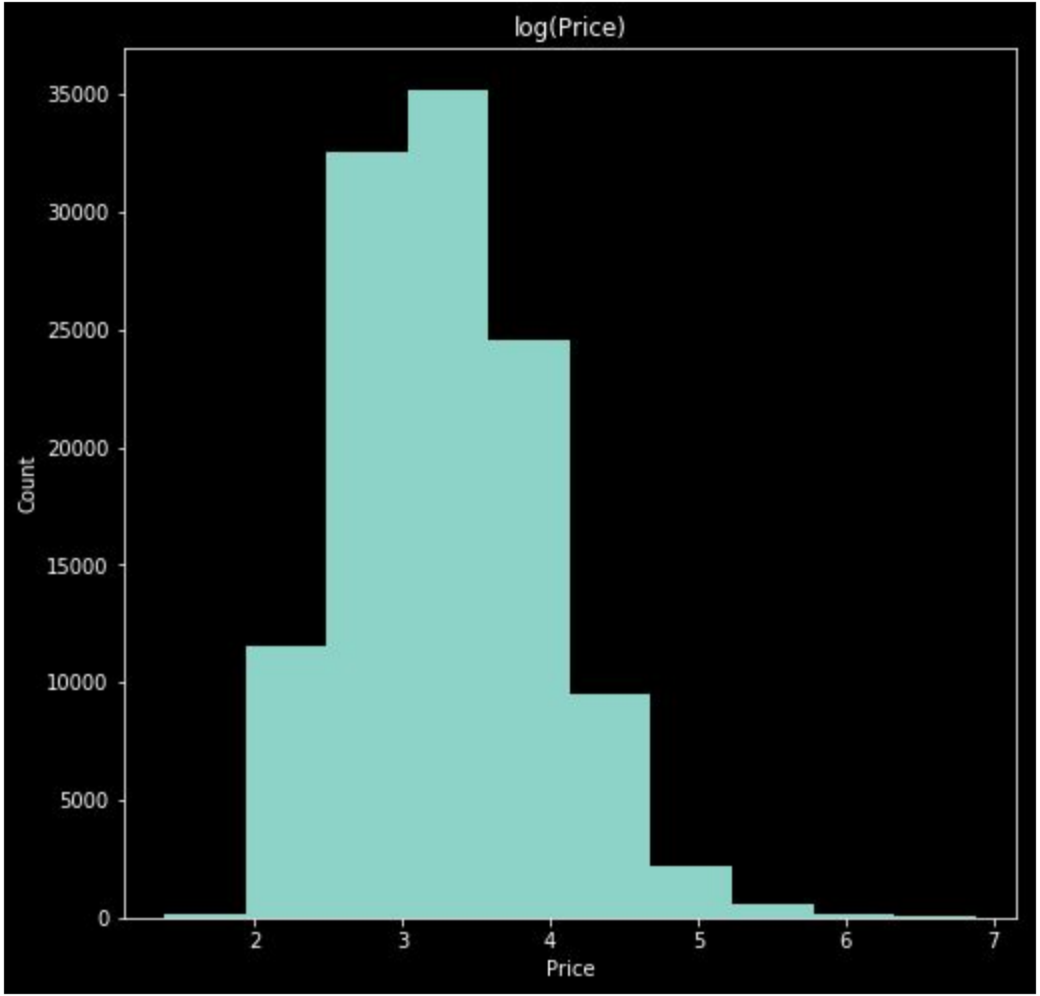
Points appear to be more-or-less normally distributed within the range 80 to 100.

- Discrete values
- Mean is approximately 88.5



Price

- Raw price was not normally distributed
- Entries with price greater than $1000 were dropped, then a logarithmic transformation was applied.

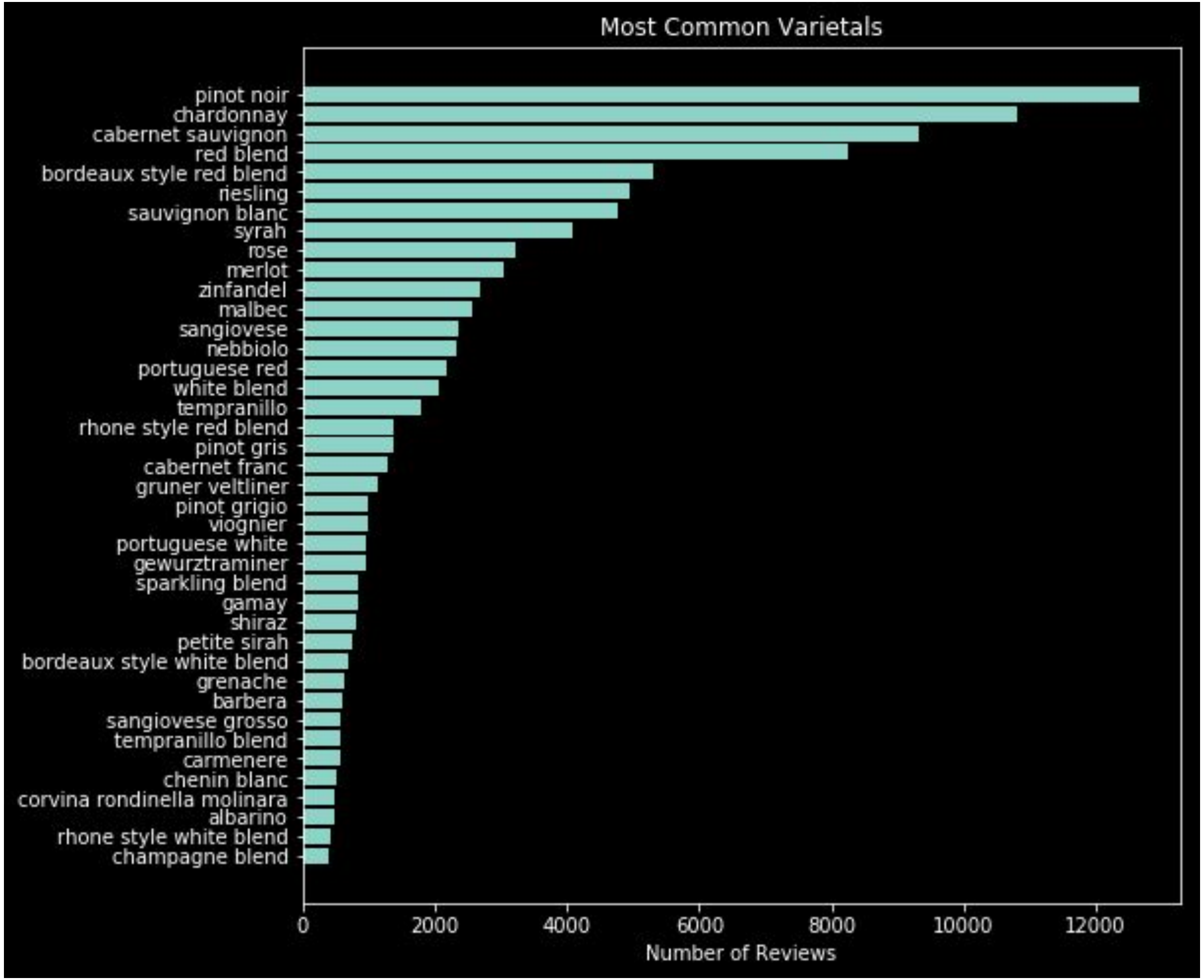


Origin

- Hierarchical variables denoting origin:
  - Country: 42 countries represented
  - Province: 415 provinces
  - Region_1: 1193 regions
  - Winery: 15453 wineries
- US accounts for about half the wines, so country alone isn't granular enough.
- Region_1 and Winery are too specific (there are too many categories in these variables).

Variety

- This dataset has a large number (679) of wine varieties represented, but some of them have only a few entries.
- We will focus on the top 40 varieties.



Designation

- Grab-bag of adjectives, with little consistency.
- Place of origin, descriptions given to the wine by the winemaker.
- Certain terms in this column indicate that the winemaker is presenting the wine as a product of superior quality
  - "reserve," "premier," "select," "grand," etc.
- Create binary feature indicating whether the wine has been labeled by winemaker to denote high quality.

Sample Description:

“There is a select group of under-$20 Malbecs from Argentina that really do the country and the variety proud. Santos is one of them; the 2006 is easy and ripe on the nose, with lovely cola, berry and herb aromas. The palate has a natural feel and bright black-fruit flavors. Not overly complex but a winner for the next year or two.”

- This description is problematic because it contains words that denote variety, origin, and year. If we are going to use these other features, we need to eliminate related terms from the “description” column.

Feature Engineering

Outcomes of Interest

- To predict the price of a bottle of wine, and its relative quality, based on information available on the label of the bottle itself.
- Regression modeling:
  - Could aid wine sellers in setting prices.
  - Could help consumers check for anomalous review scores/prices.
- Classification modeling:
  - Would consumers to make purchasing decisions based on expected price/quality.

Price/Points-Related Features

- Points feature for regression:
  - Points are already normally distributed.
  - Use as is.
- Price feature for regression:
  - Price needs to be transformed to fit a more normal distribution.
  - Natural Log Transformation
- Points feature for classification:
  - “High_rating”: binary feature for >= 90 points.
  - 26.0% of wines in dataset are rated 90 points or higher.
- Price feature for classification:
  - “thirty_or_more”: based on original “price” variable
  - 43.6% of wines in dataset are $30 or more.

Text Cleaning

- Change to lower case, remove punctuation, change diacritic characters to ASCII.
- “Stop words” must be removed manually. (Alternatives: NLTK, SpaCy)
- To help manage dimensionality, descriptor list must be reduced.
    - Arbitrary threshold: term must appear in at least 3500 of 116603 wine reviews (3%)
- Sets of unique words from “description” must be filtered to exclude words matching other features.

General Types of Wine

- Color: Red, White, Rosé.
- Additional categories: Sparkling, Fortified.
- Each grape variety has a type among these that it is most strongly associated with. The wine's general color/type is not included directly in the data set, so this has been input manually, with the aid of winefolly.com, and other sites.
- Some wine varieties do not explicitly refer to a color or type, but to a region. "Meritage," for instance, refers to several grape varieties, all of which are used to make Bordeaux wines.

Wine Types Breakdown

Wines with “type not assigned” are Meritage, Pallagrello, Manzoni, or wine varieties with fewer than 5 reviews.

- Meritage and Pallagrello aren’t enough by themselves to denote a wine’s color.
- Manzoni is a vineyard in Northern California
- Cutoff at 5 reviews is thorough enough, but not exhaustive.

type_red : 0.6241263089285868
type_white : 0.3301115751738806
type_rose : 0.02941605276022058
type_fortified : 0.03040230525801223
type_sparkling : 0.011449105083059612
type not assigned: 0.00452818538116515

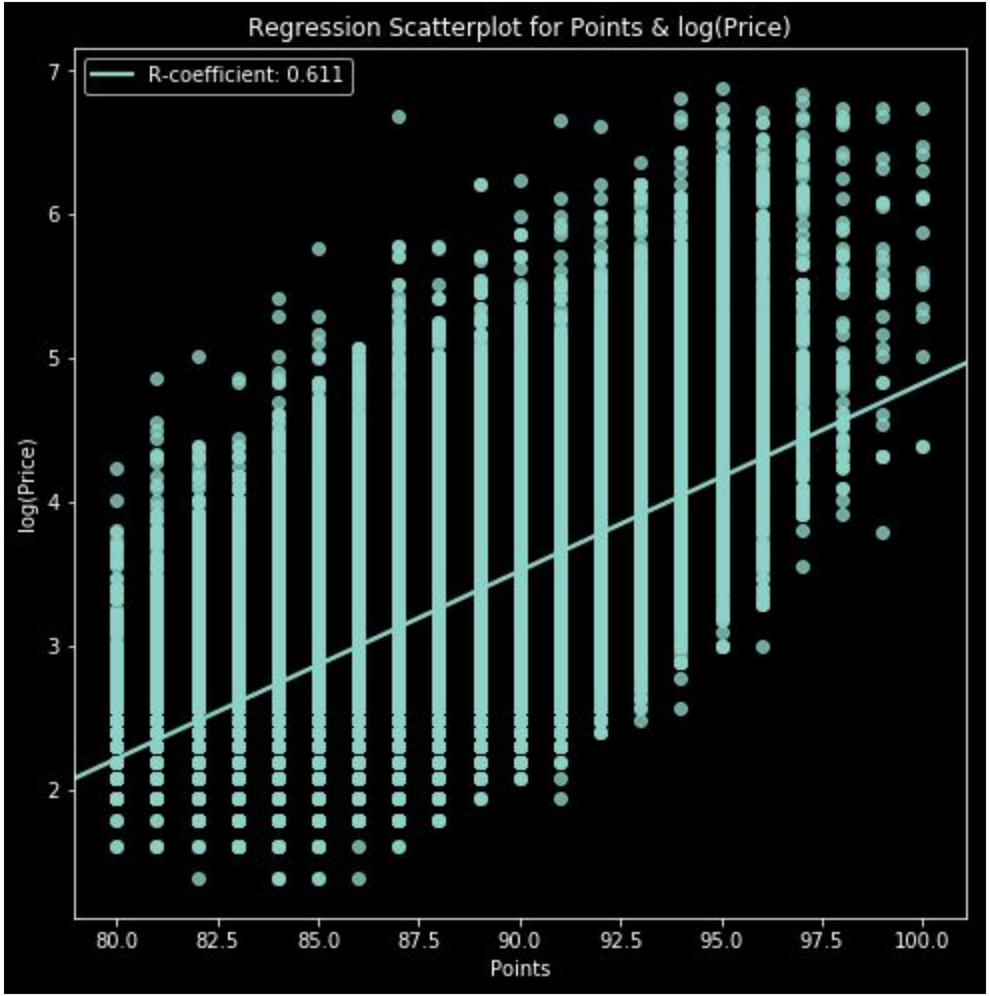
Top Varieties

- Rather than using every single variety of wine as a binary feature, we will perform the same sort of feature engineering we did with origin.
- Take the 40 most common categories, and then create an “other” category for varieties whose count isn’t high enough.
- 100777 wine reviews are top 40 varieties.
- 15826 for other varieties.
- Aggregate remainder into “other red,” “other white,” “other rose,” “other fortified,” and “other sparkling.”

Variable Relationships
&
Feature Reduction

Price-Points

- Fairly strong positive correlation between Log Price and Points.
- Collinearity is not an issue, since we will not be including either in the features for our model; they will only ever be target variables.



Words in Description

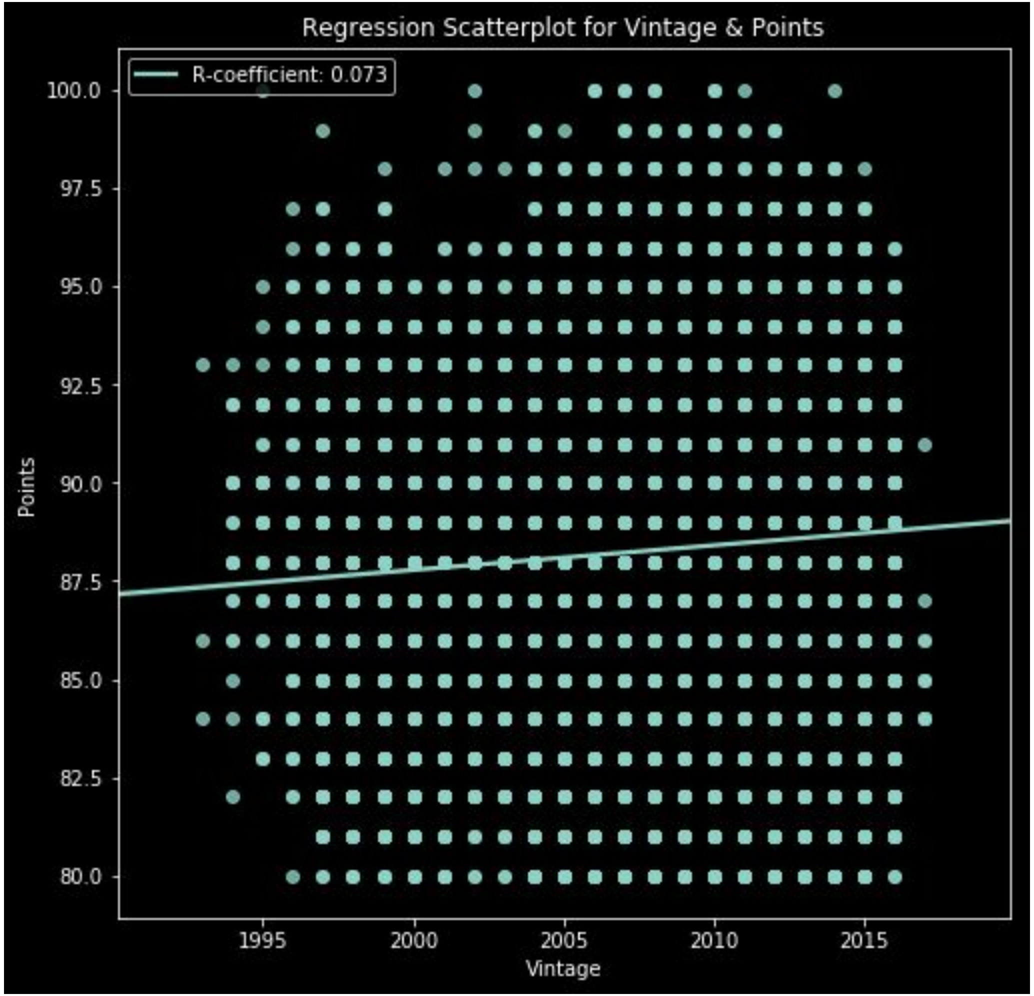
One of the indicators of strong sentiment is how much time the reviewer is willing to devote to it. In this case, since all the wines have positive scores, we can see that the length of the reviewer's description is positively correlated with both price and points given.

- Words in description
- Length in characters

These two features are highly collinear, so we will only use "words_in_description."

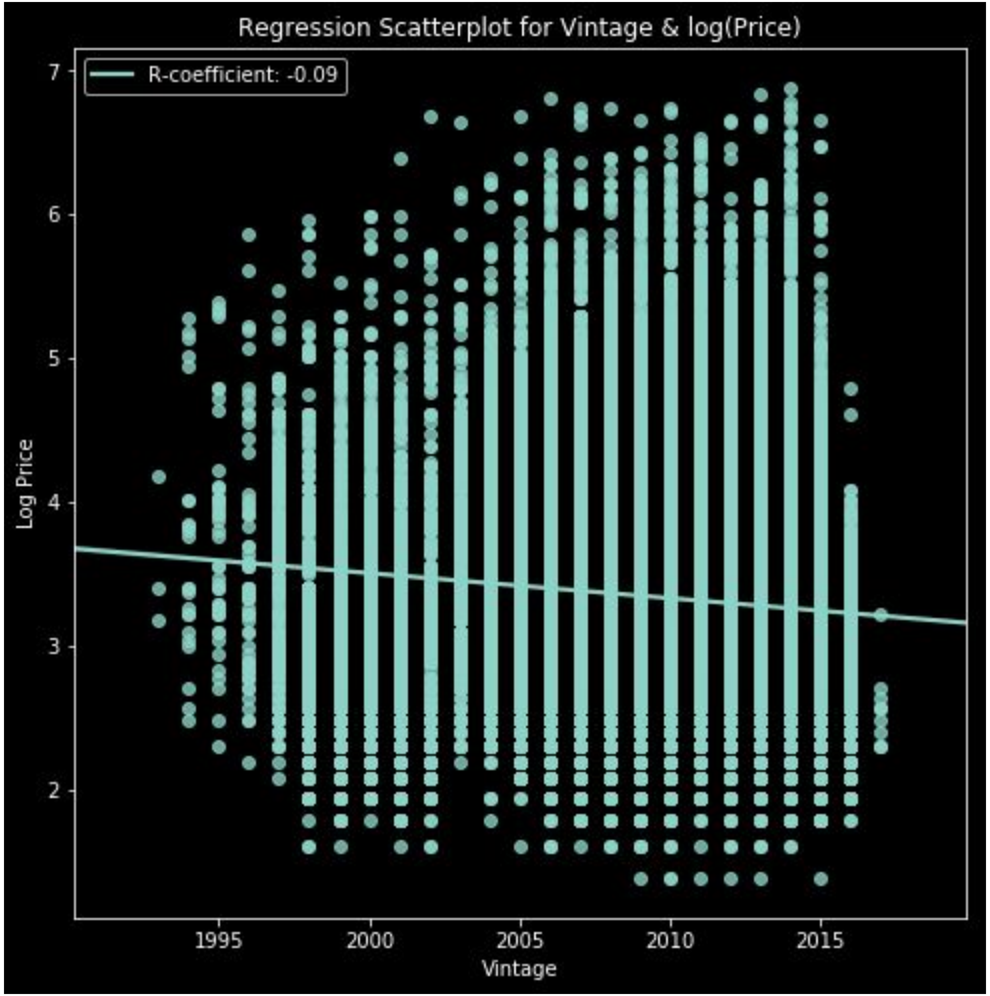
Vintage-Points

- Weak positive correlation with points.
- More recent wines tend to receive slightly higher points ratings overall.



Vintage-Price

- Vintage has a weak negative correlation with log-transformed price.
- Older wines tend to be somewhat more expensive.



Variables of Interest

Regression:

- log(Price)
- Points

Binary classification:

- “thirty_or_more”: 1 if a bottle costs $30 or more, 0 otherwise.
- “high_rating”: 1 if a wine is rated 90 points or higher, 0 otherwise.

Highest-Rated Wines by Variety:

(All numerical values are means for each variety.)

variety	points	price	vintage	logprice	thirty_or_more	high_rating	words_in_description	length_in_characters	descriptor_cherries
champagne blend	91.638821	98.651106	2004.095823	4.345793	0.894349	0.702703	46.132678	258.321867	0.000000
sangiovese grosso	90.801358	65.634975	2006.803056	4.049777	0.881154	0.556876	41.859083	246.176570	0.001698
nebbiolo	90.331188	65.609610	2009.968683	4.048969	0.919348	0.453024	42.969541	257.752038	0.017160
gruner veltliner	90.014047	26.823529	2012.861282	3.165543	0.285338	0.410009	40.267779	231.863038	0.000000
sparkling blend	89.457509	40.807916	2009.472643	3.569238	0.657742	0.344587	43.417928	251.025611	0.008149
riesling	89.449818	32.026103	2011.783286	3.210063	0.325172	0.347025	42.123027	246.831849	0.004047
pinot noir	89.425057	47.258562	2011.168077	3.686754	0.736771	0.405046	43.451633	249.350708	0.112394
syrah	89.296814	39.154167	2010.038725	3.520753	0.640931	0.389706	43.486765	250.933578	0.051225
rhone style red blend	89.149211	34.995696	2010.266858	3.389264	0.522238	0.352224	46.153515	264.469871	0.103300
shiraz	89.089109	41.965347	2008.272277	3.343247	0.422030	0.337871	44.100248	252.777228	0.044554

Most Expensive Wines by Variety:

(All numerical values are means for each variety.)

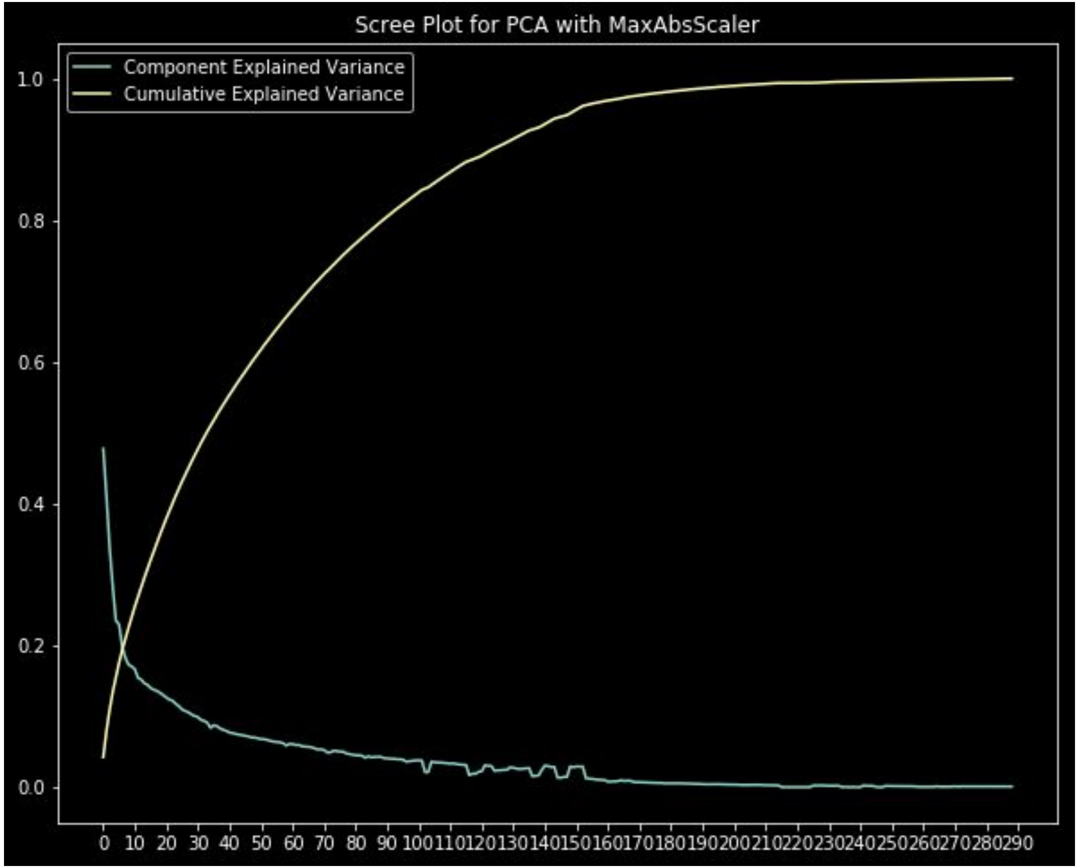
variety	points	price	vintage	logprice	thirty_or_more	high_rating	words_in_description	length_in_characters	descriptor_cherries
champagne blend	91.638821	98.651106	2004.095823	4.345793	0.894349	0.702703	46.132678	258.321867	0.000000
sangiovese grosso	90.801358	65.634975	2006.803056	4.049777	0.881154	0.556876	41.859083	246.176570	0.001698
nebbiolo	90.331188	65.609610	2009.968683	4.048969	0.919348	0.453024	42.969541	257.752038	0.017160
cabernet sauvignon	88.617537	47.972237	2009.629006	3.581547	0.602959	0.305392	42.604352	244.706935	0.060993
corvina rondinella molinara	88.495968	47.907258	2006.951613	3.592306	0.570565	0.225806	38.243952	223.012097	0.000000
pinot noir	89.425057	47.258562	2011.168077	3.686754	0.736771	0.405046	43.451633	249.350708	0.112394
sangiovese	88.614899	45.294192	2009.644781	3.569081	0.563973	0.229798	40.839646	243.609848	0.027357
bordeaux style red blend	88.786709	44.266943	2010.858245	3.485382	0.525414	0.298569	42.934676	242.506589	0.031250
shiraz	89.089109	41.965347	2008.272277	3.343247	0.422030	0.337871	44.100248	252.777228	0.044554
sparkling blend	89.457509	40.807916	2009.472643	3.569238	0.657742	0.344587	43.417928	251.025611	0.008149

Feature Collinearity:

There is some collinearity here between provinces and varieties with protected designation of origin status (PDO), as well as other wines commonly produced in specific regions. For instance, a wine can only be labeled "Bordeaux" if it's from the Bordeaux region of France, otherwise the same variety of grape would be labeled "Meritage." Similarly, Gamay grapes are used to make wine in Beaujolais, France, and Sangiovese is most commonly produced in Tuscany, Italy. Because this is only for a few categories, we will ignore this for now, although further models could implement more stringent filters at this stage.

Dimensionality Reduction

- PCA to determine roughly optimal number of features.
  - The "elbow" on the scree plot was somewhat ambiguous, but 40 looked about right.
- 40 components explain only 54% of variance.
- Separate lists for each target variable, each based on approximate number of features yielded by PCA.
- These lists are features that have highest correlation with each individual outcome variable; top 40 variables for "points" are different from top 40 for "high_rating."



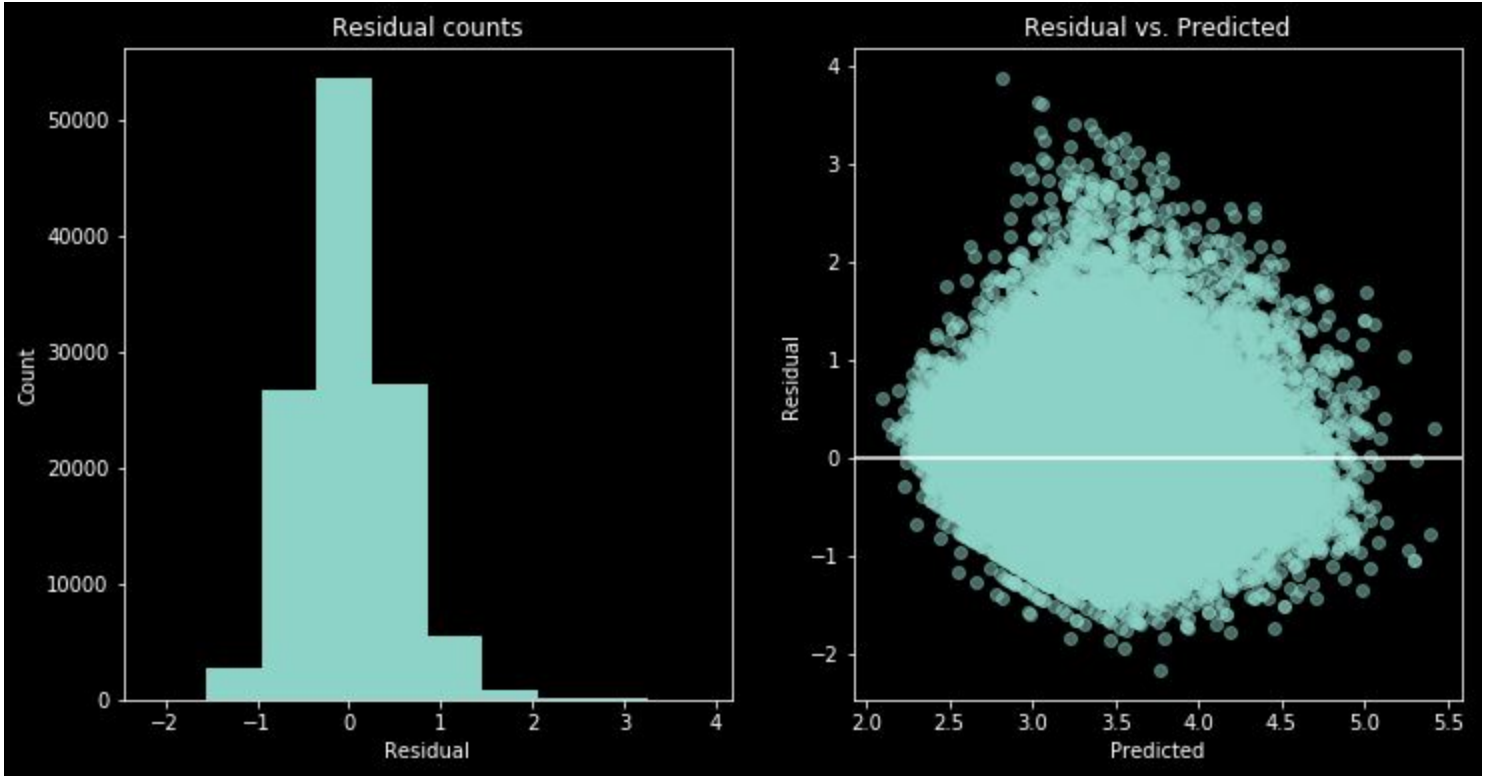
Regression Modeling

Ordinary Least Squares Regression: Price

For starters, we tried an OLS model, to see baseline performance:

- Training accuracy: 0.3538
- Test accuracy: 0.3498
- Mean-squared error: 0.2729

This is rather poor.

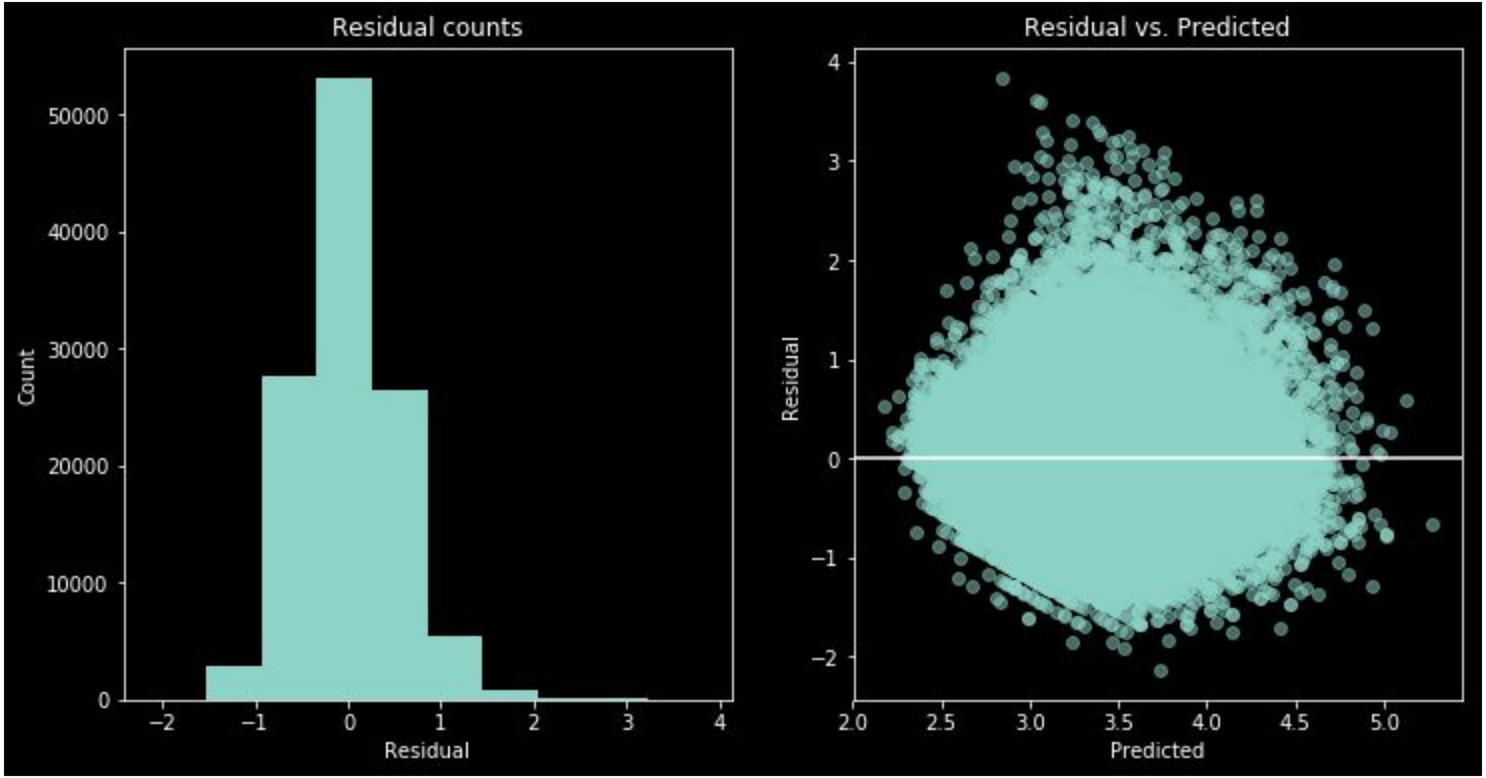


LASSO Linear Regression: Price

Next, we tried LASSO regression, in hopes that the regularization coefficient would help in terms of further reducing our variables:

- Training accuracy: 0.3552
- Test accuracy: 0.3486
- Mean-squared error: 0.2734

Slightly worse.

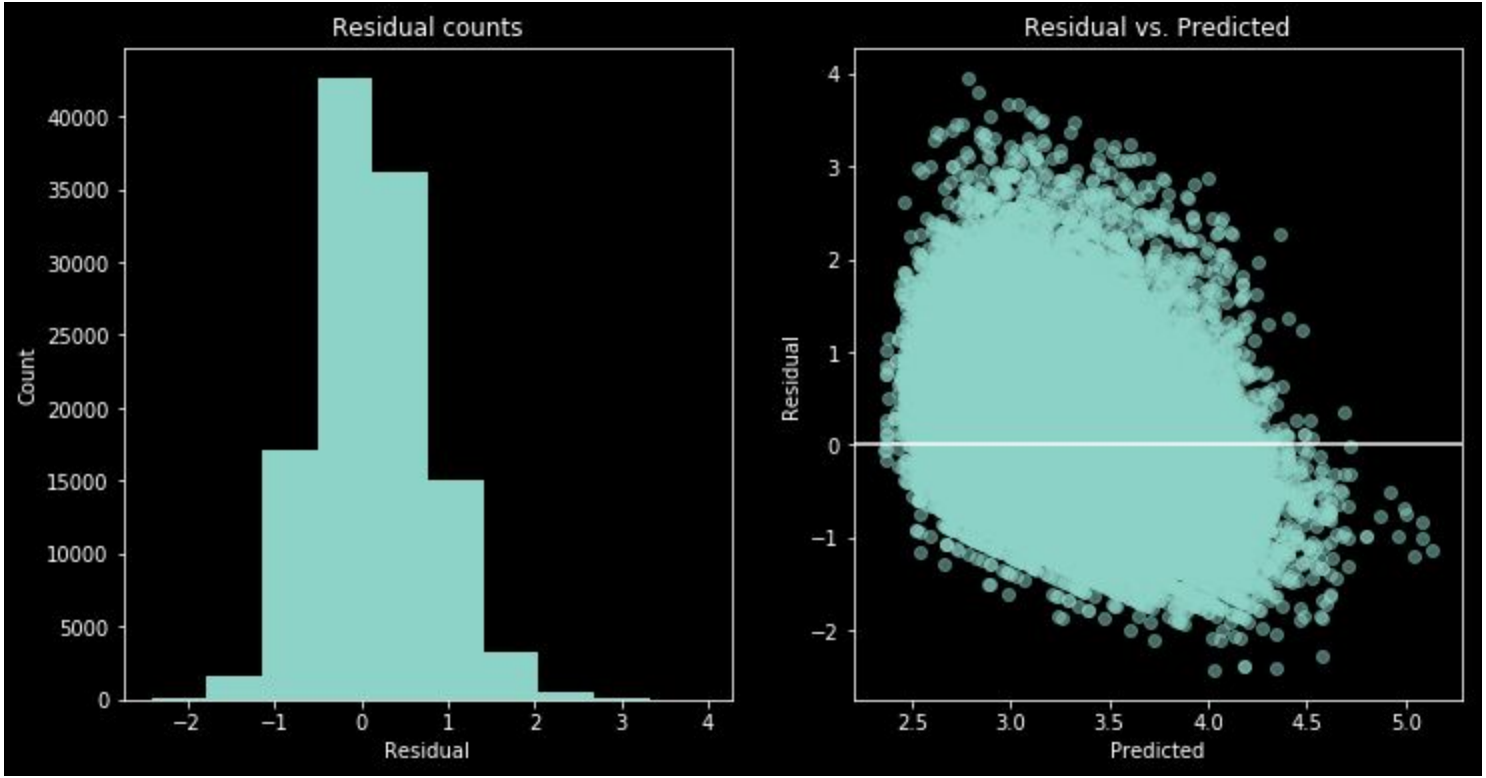


KNN Regression: Price

Next, we will try KNN (K=10), as it is generally resilient to noisy data, and works well with larger datasets:

- Training accuracy: 0.4314
- Test accuracy: 0.3030
- Mean-squared error: 0.2926

Worse performance, slow execution.

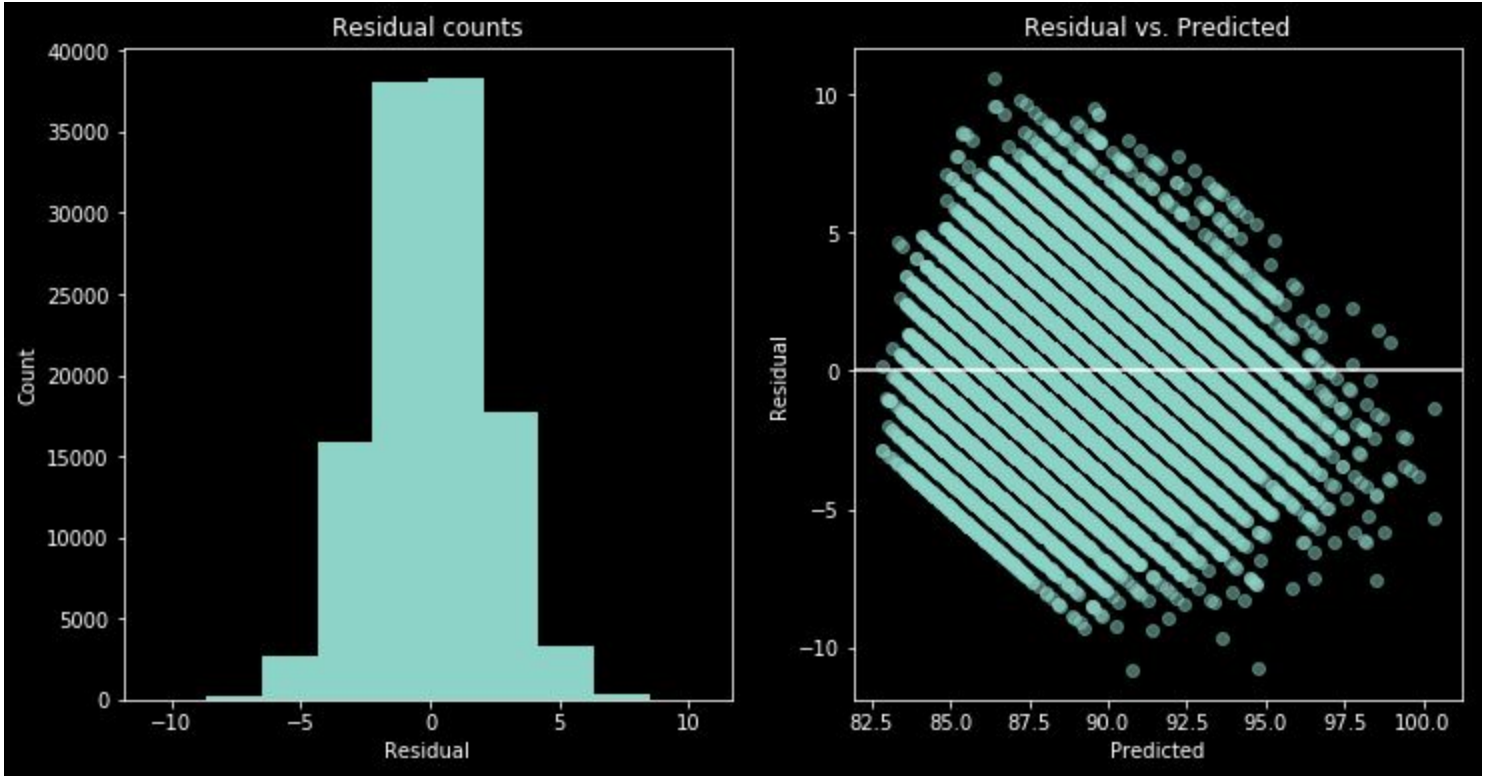


Ordinary Least Squares Regression: Points

OLS on points yielded what looks to be a better score, but the MSE is very high:

- Training accuracy: 0.4454
- Test accuracy: 0.4446
- Mean-squared error: 5.1465

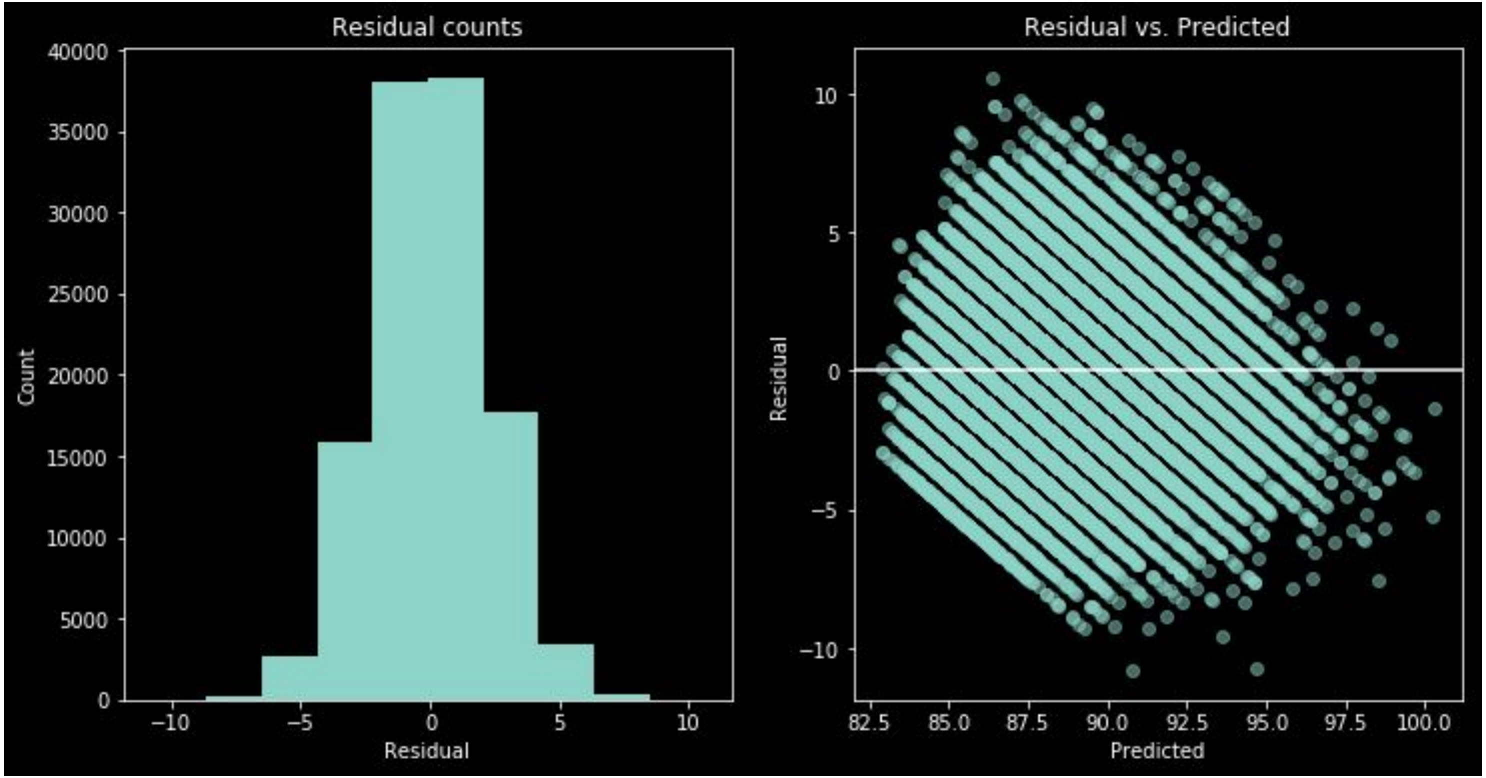
This is not an ideal result.



LASSO Linear Regression: Points

LASSO performed almost as well as OLS:

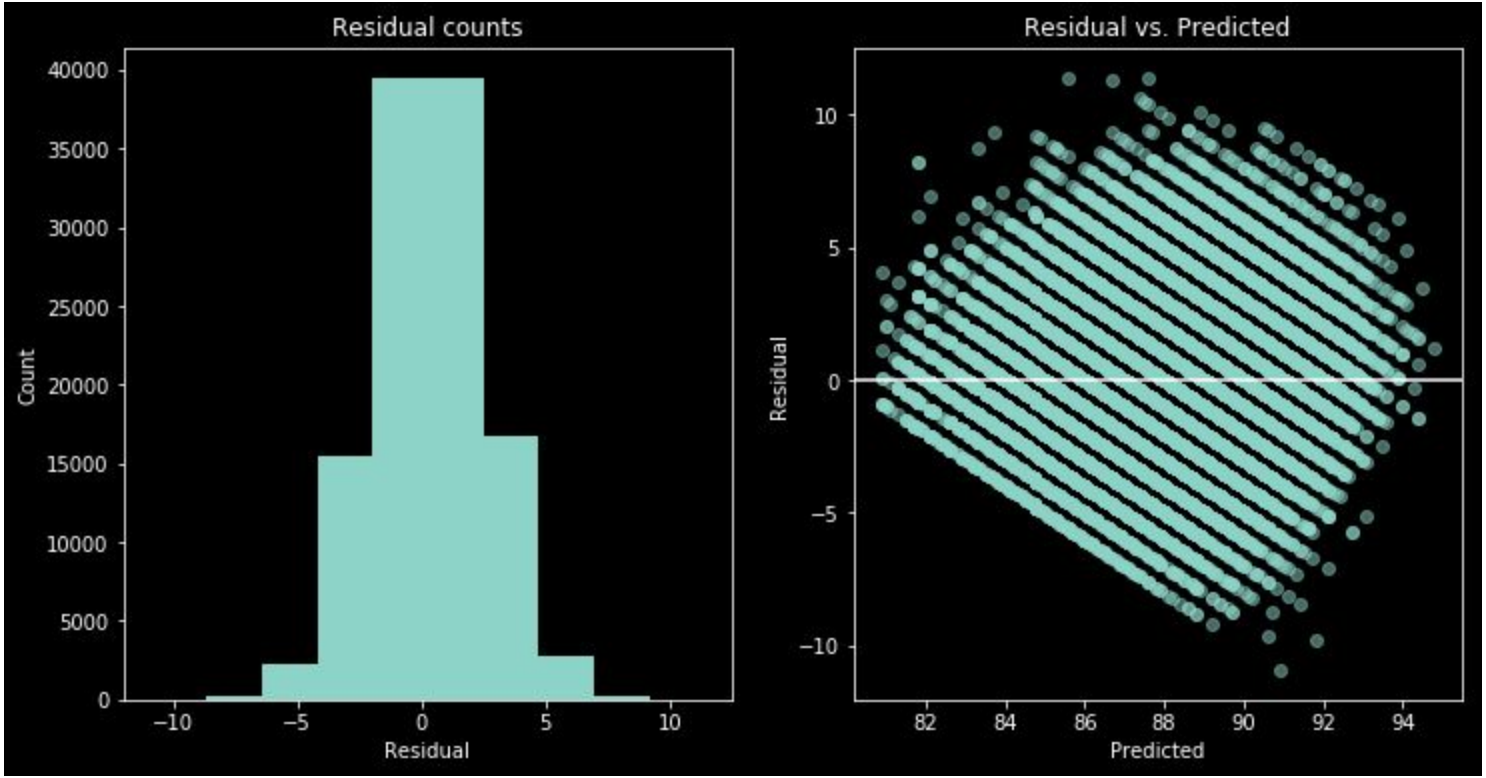
- Training accuracy: 0.4453
- Test accuracy: 0.4445
- Mean-squared error: 5.1475



KNN Regression: Points

KNN Regression performed poorly and slowly once again.

- Training accuracy: 0.4723
- Test accuracy: 0.3654
- Mean-squared error: 5.8809



Regression Results:

- Regression for both price and point score yields heteroscedastic residuals and, particularly in the case of points, abnormally high mean-squared error.
- OLS outperformed other regression methods applied.
- We may achieve better performance by modeling price and points rating as classification problems.

Classification Modeling

Logistic Regression: Price

As with OLS regression, we begin classification using a simple, tried-and-true model as a benchmark.

- Training score: 0.7308
- Test score: 0.7263
- Type I errors: 0.1632
- Type II errors: 0.1106
- Sensitivity: 0.7120
- Specificity: 0.7351

Confusion matrix:

thirty_or_more	0	1	All
row_0			
0	17585	6336	23921
1	4293	10615	14908
All	21878	16951	38829

LASSO Logistic Regression: Price

Next we use LASSO logistic regression, again in hopes of using the L1 parameter to effectively reduce our feature set:

- Training score: 0.7306
- Test score: 0.7260
- Type I errors: 0.1637
- Type II errors: 0.1103
- Sensitivity: 0.7122
- Specificity: 0.7346

Performance was very slightly worse than logistic regression without the L1 coefficient applied.

Confusion matrix:

thirty_or_more	0	1	All
row_0			
0	17597	6357	23954
1	4281	10594	14875
All	21878	16951	38829

Random Forest: Price

Given the binary nature of most of our data, a random forest model may do well for classification:

n_estimators = 100

- Training score: 0.7708
- Test score: 0.7427
- Type I errors: 0.1454
- Type II errors: 0.1118
- Sensitivity: 0.7224
- Specificity: 0.7564

There may be slight overfitting with this model, but it shows improvements over logistic and LASSO regression:

Confusion matrix:

thirty_or_more	0	1	All
row_0			
0	17535	5647	23182
1	4343	11304	15647
All	21878	16951	38829

Random Forest Grid Search:

After running a GridSearchCV, these are the final hyperparameters for Random Forest:

n_estimators = 200, criterion = entropy, class_weight = None, min_samples_split = 50

- Training score: 0.7933
- Test score: 0.7463
- Type I errors: 0.1426
- Type II errors: 0.1111
- Sensitivity: 0.7257
- Specificity: 0.7603
- CVscore (cv=5) mean: 0.7477
- CVscore std: 0.0098

There may be slight overfitting with this model, but it shows improvements over logistic and LASSO regression:

Confusion matrix:

thirty_or_more	0	1	All
row_0			
0	17564	5538	23102
1	4314	11413	15727
All	21878	16951	38829

Gradient-Boosted Decision Tree: Price

Given the success of random forest models, Gradient-boosting on a decision tree may also be worth an attempt.

Initial parameters: n_estimators=100, max_depth=3

- Training score: 0.7380
- Test score: 0.7320
- Type I errors: 0.1636
- Type II errors: 0.1044
- Sensitivity: 0.7233
- Specificity: 0.7372

Performance here is worse than random forest, but could be improved with different parameters.

Confusion matrix:

thirty_or_more	0	1	All
row_0			
0	17824	6353	24177
1	4054	10598	14652
All	21878	16951	38829

GBDT Optimization: Price

A grid search for GBDT yields the following parameters:

Final parameters: n_estimators=200, max_depth=12, loss='exponential', min_samples_split=100.

- Training score: 0.8244
- Test score: 0.7474
- Type I errors: 0.1438
- Type II errors: 0.1088
- Sensitivity: 0.7291
- Specificity: 0.7597
- CVscore (cv=5) mean: 0.7502
- CVscore std: 0.0141

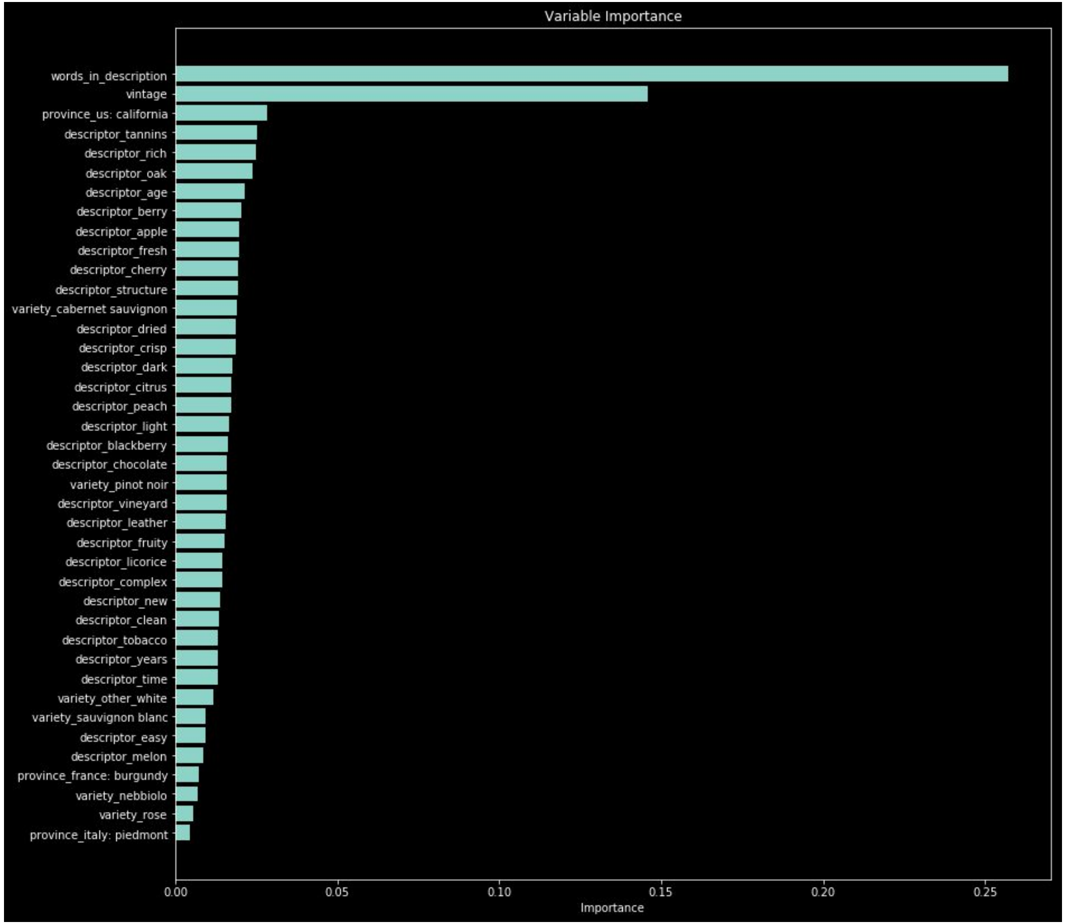
After optimization, the GBDT mostly outperformed the decision tree model, but with a huge cost in execution time, and with a slightly higher standard deviation between scores in cross-validation.

Confusion matrix:

thirty_or_more	0	1	All
row_0			
0	17654	5583	23237
1	4224	11368	15592
All	21878	16951	38829

GBDT Variable Importances: Price

In descending order, the "feature_importances_" ranks for the sklearn gradient-boosted decision tree model for price classification.



Logistic Regression: Points

For modeling points, we will once again start with logistic regression as a benchmark:

- Training score: 0.8037
- Test score: 0.8024
- Type I errors: 0.1416
- Type II errors: 0.0560
- Sensitivity: 0.6802
- Specificity: 0.8283

Relative to price classification, we see much higher accuracy (specificity in particular) from the outset.

Confusion matrix:

high_rating	0	1	All
row_0			
0	26535	5500	32035
1	2173	4621	6794
All	28708	10121	38829

LASSO Logistic Regression: Points

Despite LASSO not outperforming regular logistic regression for modeling price, we will try it here for the sake of thoroughness.

- Training score: 0.8039
- Test score: 0.8029
- Type I errors: 0.1419
- Type II errors: 0.0552
- Sensitivity: 0.6825
- Specificity: 0.8282

This time, LASSO did slightly better. However, improvements were mostly in sensitivity of the model, which isn't as important for our purposes as specificity.

Confusion matrix:

high_rating	0	1	All
row_0			
0	26563	5510	32073
1	2145	4611	6756
All	28708	10121	38829

Random Forest: Points

With random forest having outperformed logistic and LASSO logistic regression for price, we should expect the same pattern for points:

n_estimators = 100, min_samples_split=100

- Training score: 0.8325
- Test score: 0.8099
- Type I errors: 0.1373
- Type II errors: 0.0529
- Sensitivity: 0.7000
- Specificity: 0.8334

This is looking pretty good! Let's tune it some more.

Confusion matrix:

high_rating	0	1	All
row_0			
0	26655	5330	31985
1	2053	4791	6844
All	28708	10121	38829

Random Forest Grid Search:

After running a GridSearchCV, these are the final hyperparameters for Random Forest:

n_estimators = 200, criterion = gini, class_weight = None, min_samples_split = 50

- Training score: 0.8502
- Test score: 0.8132
- Type I errors: 0.0132
- Type II errors: 0.0546
- Sensitivity: 0.7016
- Specificity: 0.8382
- CVscore (cv=5) mean: 0.8056
- CVscore std: 0.0080

This time, gini outperformed entropy in the random forest grid search, but only by about 0.02% accuracy.

Confusion matrix:

high_rating	0	1	All
row_0			
0	26586	5131	31717
1	2122	4990	7112
All	28708	10121	38829

Gradient-Boosted Decision Tree: Points

Once more, we will a try gradient-boosted decision tree model to compare to our random forest.

Initial parameters: n_estimators=100, max_depth=3

- Training score: 0.8124
- Test score: 0.8069
- Type I errors: 0.1440
- Type II errors: 0.0491
- Sensitivity: 0.7037
- Specificity: 0.8274

This time, we see less of a drop in accuracy between the untuned random forest and GBDT models.

Confusion matrix:

high_rating row_0	0	1	All
0	26800	5589	32389
1	1908	4532	6440
All	28708	10121	38829

GBDT Optimization: Points

After running a grid search, we arrive at the same set of parameters we found for price modeling:

Final parameters: n_estimators=200, max_depth=12, loss='exponential', min_samples_split=100.

- Training score: 0.9061
- Test score: 0.8195
- Type I errors: 0.1194
- Type II errors: 0.0610
- Sensitivity: 0.6982
- Specificity: 0.8502
- CVscore (cv=5) mean: 0.8140
- CVscore std: 0.0061

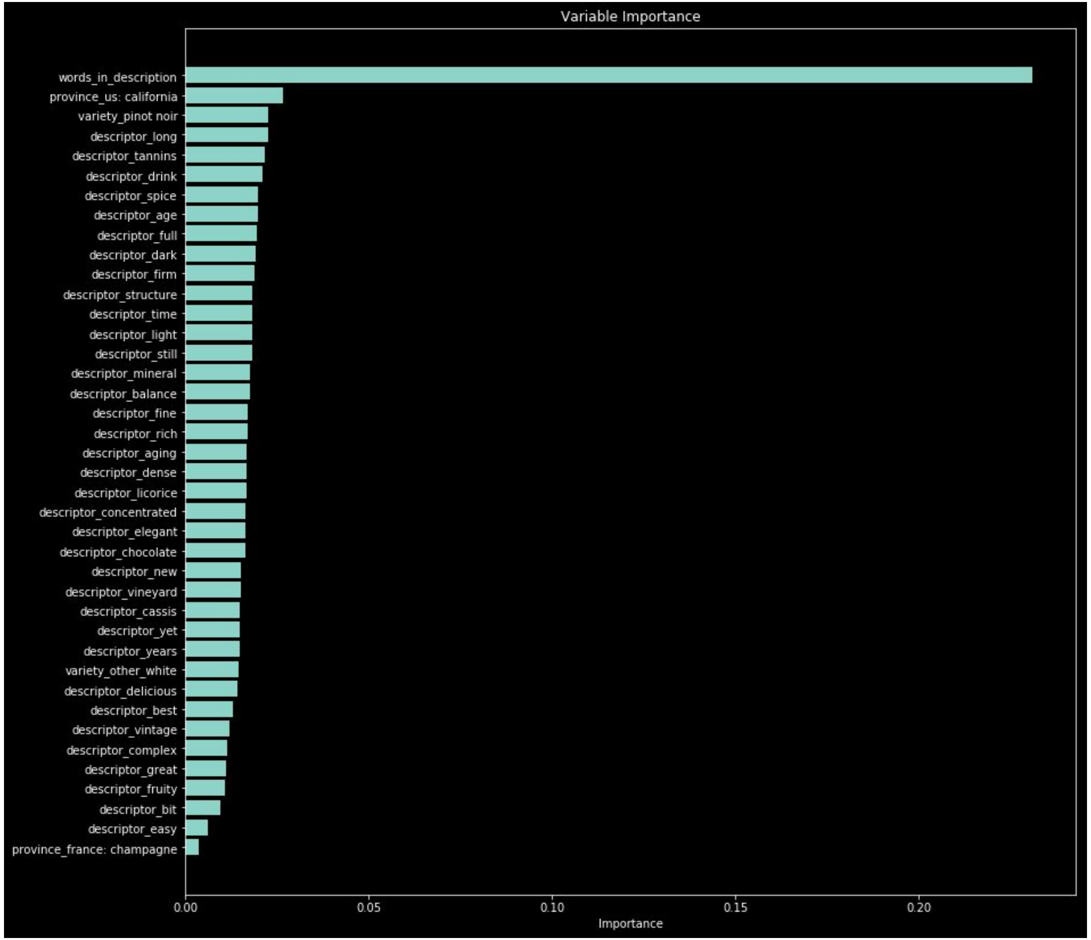
While this model has a slightly lower sensitivity than the tuned random forest model, its specificity is higher by more than a full percentage point, and it also has a higher overall accuracy.

Confusion matrix:

high_rating	0	1	All
row_0			
0	26338	4638	30976
1	2370	5483	7853
All	28708	10121	38829

GBDT Variable Importances: Points

In descending order, the "feature_importances_" ranks for the sklearn gradient-boosted decision tree model for points classification.



Conclusions

Overall results:

Results with regression models were disappointingly low, but it might be possible to improve them by using other modeling techniques for the descriptive terms.

Given that the purpose of a classification model would likely be to aid consumers in making purchasing decisions, high specificity is especially important for most consumers (so as to avoid wasting money on a wine falsely valued at $30 or more, or falsely predicted to be of higher quality), and the gradient-boosted decision tree model achieved the best results as far as specificity was concerned. Gradient-boosted decision trees were able to provide the highest overall accuracy.

However, these models take much longer to run than did the random forests, which achieved similar results, so if time is an issue, it may be better to use a random forest model.

Potential Future Projects:

Further refinement of modeling this dataset would likely include a full-fledged sentiment analysis using NLTK or SpaCy, which could be used for better prediction of rating and/or price.

The same sort of model could be applied to user-submitted information for beer review websites such as RateBeer, BeerAdvocate, and Untappd. Grapes acquire a substantial portion of their flavor characteristics from the soil they're grown in (terroir), and more ingredients go into making beer than go into wine, so this would necessitate a model based more on ingredients than locations of origin:

- Hop variety
- Type of malt
- Adjuncts (fruit, chocolate, coffee, nuts, spices)
- Type of yeast (lager, ale, wild, sour)
- Barrel-aging

Recap

- EDA/Data cleaning
- Initial feature selection
- Feature engineering/selection
- Dimensionality reduction using PCA
- Regression Modeling (Price/Points)
  - Linear Regression (Highest R-squared coefficient)
  - LASSO Linear Regression
  - KNN Regression
- Classification Modeling (>=$30/>=90 Points)
  - Logistic Regression
  - LASSO Logistic Regression
  - Random Forest Classification
  - Gradient-Boosted Decision Tree Classification (Highest accuracy)
- Results

Sources

https://www.kaggle.com/zynicide/wine-reviews/home

https://winefolly.com/

https://www.winespectator.com