Predicting Wine Quality: A Conundrum

Would you like some cheese with that?

Kalbi Zongo, Song Hoa Choi, Gina Shellhammer, Matt Edwards

June 2, 2014

Popping the Cork

Task

Predict the blind taster quality score of a wine based on chemical tests.

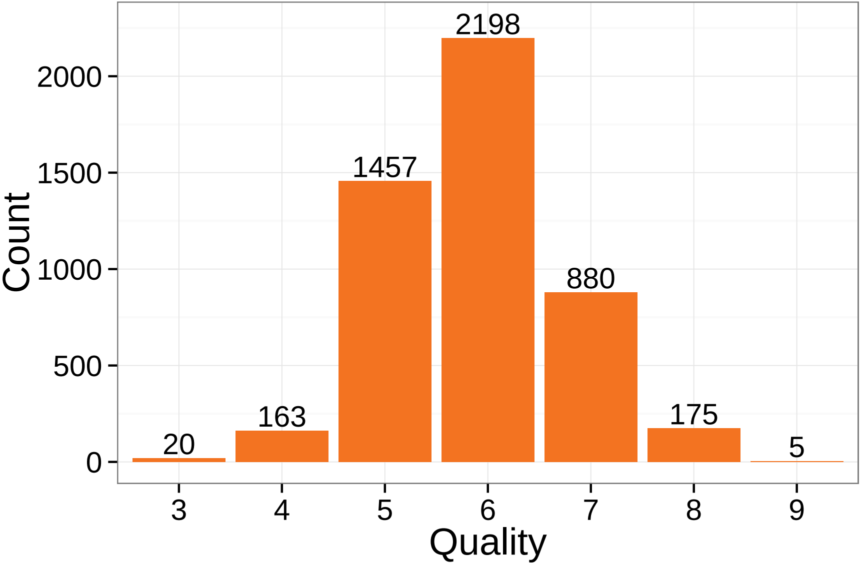
Data

- Two Datasets: Red & White vinho verde wine samples from northern Portugal
- 1599 & 4898 rows, respectively
- Concentrated on White Wine, due to more data

Data

- 11 Explanatory variables: measurements from various phytochemicals in wine
- Response variable "quality" is discrete variable on ordered scale from 0 (worst) to 10 (best)
- Nothing graded as 0, 1, 2, or 10

White Wine Quality Scores



Learning about Wine

Training and Testing Sets

- Training and Testing sets constructed through stratified sampling.
- Quality variable was the strata
- Why: Ensure representation of all quality categories in both Training & Testing datasets.
- How: 37.5% of items (rounded up) in strata were randomly selected to be in the testing set. Remaining 62.5% were the training set.
- Regression and Random method used these training sets, Classification used different set.

Methods

- With "prediction" as the goal, we think regression.
- Forward, Backward and Subset Model Selection done, all resulted in same model.
- Classification Method can also be used to predict.

K-Nearest Neighbor Regression

- Using some measure of distance, find nearest neighbors in dataset
- Order examples by increasing distance
- Find a "optimal" number k of nearest neighbors
- Calculate an inverse distance weighted average with the k-nearest multivariate neighbors
- Used `fit` function from `rminer` package in R. Offers many regression types

Ordinal Regression

Also "Ordered Logistic" Regression

- Estimate seperate binary regression models for all of: $P(score \leq j)/P(score > j)$, for all j
- To get the probability of the score being j: $P(Score = j) = P(score \leq j) - P(score < j)$
- So we can get the probability of each category.

Multiclass Classification

One-vs-All (or One-vs-Rest) Algorithm

- Split problem into n binary classification problems, where n is number of classes.
- Treat class i as "positive" class, everything else as "negative" class
- Train logistic regression classifier $h_\theta^i(x)$ for each class i to predict the probability that $y = i$
- On new input x, evaluate $\max_i h_\theta^i(x)$, so whichever class has the highest probability based on our input, we then predict $\hat{y}$ to be in that class

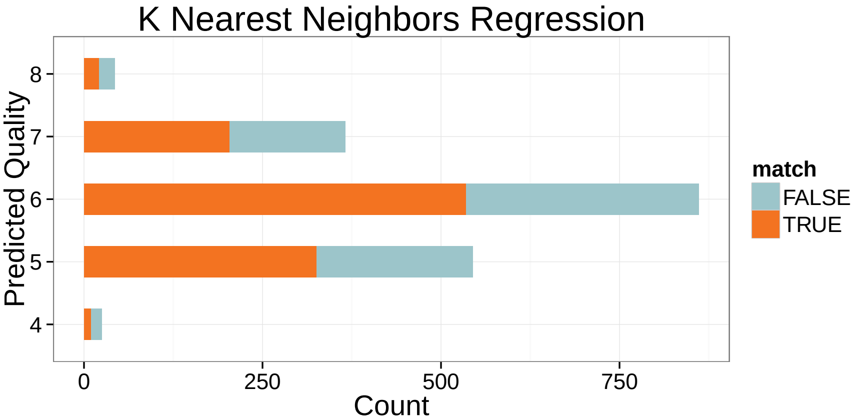
Random Randomness is Random

- 75% of Quality ratings were either 5 or 6.
- Is randomly assigning 5 or 6 to everything as good as, or better than, our other methods?
- Using `rbinom(1,1,0.6014)`, 1s were predicted as quality 6, 0s as quality 5
- Probability of 60.14% because from Training Set, considering only 5s and 6s, 6s were 60.14% of total observations
- Our base line success rate to compare other methods.

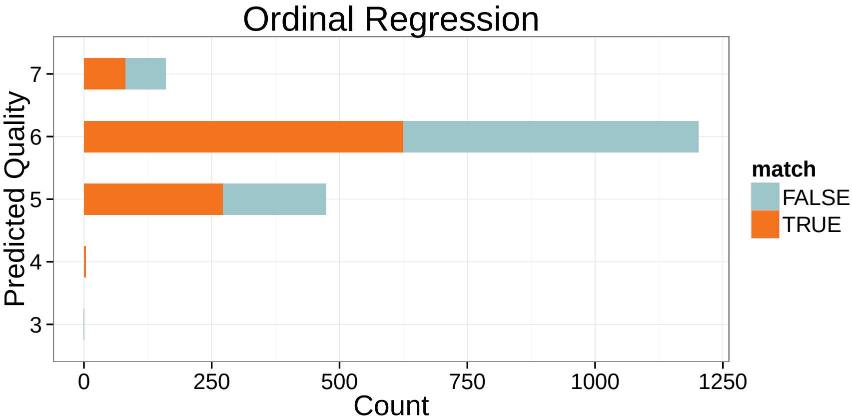
Results


pale yellow core
rich gold core
primary colors
primary colors

K Nearest Neighbors Regression: 59.6% Success Rate



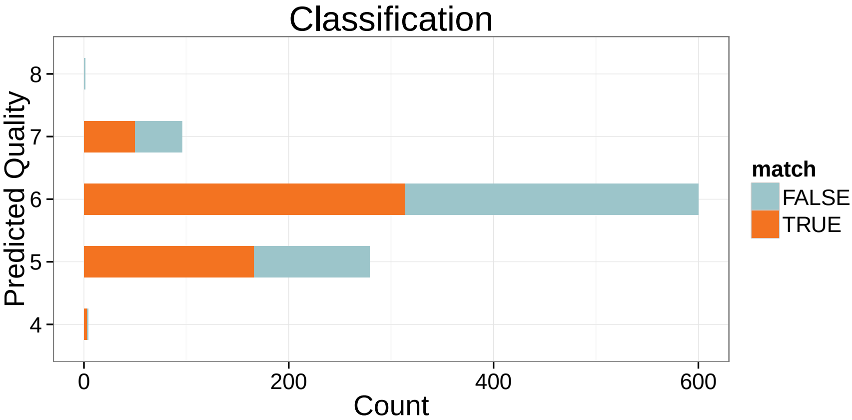
Ordinal Regression: 53.3% Success Rate



Regression Summary

- K Nearest Neighbors
  - Overall 59.6% success rate.
  - No properly allocated 3s or 9s
- Ordinal Regression
  - Overall 53.3% success rate
  - No properly allocated 3s, 8s or 9s.

Classification: 54.4% Success Rate



Classification Summary

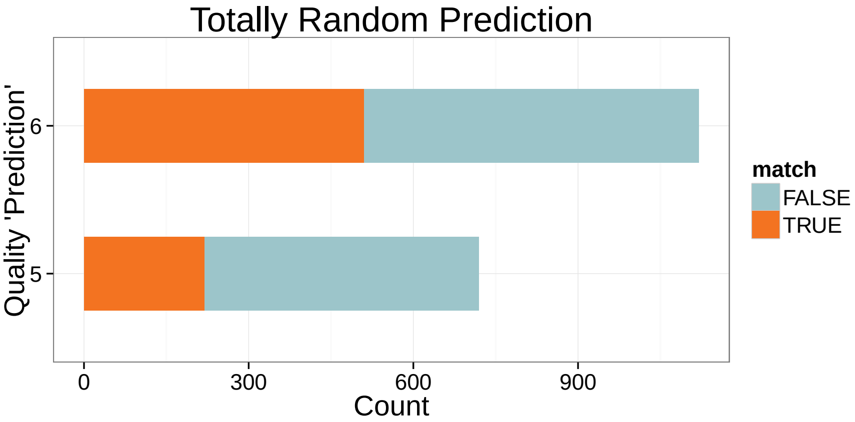
- 54.4% success rate
- No properly allocated 3s or 9s

Comparison: Success by Quality Predicted

	KNN		Ordinal		Classification	
Prediction	% Match	% Fail	% Match	% Fail	% Match	% Fail
3	n/a	n/a	0%	100%	n/a	n/a
4	40.0%	60.0%	100%	0%	75.0%	25.0%
5	59.8%	40.2%	57.4%	42.6%	59.5%	40.5%
6	62.1%	37.9%	52.0%	48.0%	52.3%	47.7%
7	55.7%	44.3%	50.6%	49.4%	52.1%	47.9%
8	48.8%	51.2%	n/a	n/a	0%	100%
9	n/a	n/a	n/a	n/a	n/a	n/a

'N/A' means nothing predicted at that quality.

Random 'Prediction': 39.67% Success Rate



Turns out, that's not really a great 'prediction' method. Who knew?

Discussion


ELKI
LAS MORAS
TORRONTES
2017
LE CHANAY
Hubert Lamy
BOURGOGNE
2009
Finest
2009
MORA

One-vs-All Classification: Assumptions

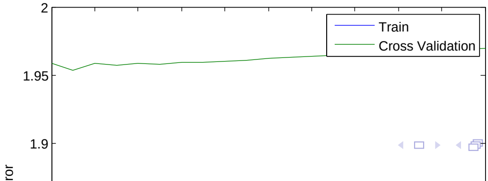
- Need 3 data sets if doing Model Selection.
- Assumes that the the individual log reg are indepent–meaning that the probability of all categories do not sum to one.
- Approach assumes that the category with higher prob is more likely to occur that other categories

One-vs-All Classification: Limitations

- When two or more categories have the same probability of success, then the approach will just pick one.
- The algorithm is computationally expensive. Ran in about 3 mins for this data set.
- Scalability is an issue for the algorithm, larger data sets could cause issues.

Cross-Validation

Lambda is tuning parameter for model. Want to pick the cross validation set with the lowest cross-validation.



Regression: Assumptions

- Multicollinearity is not an issue with prediction. It blows up SE, which is bad for estimation, not so bad with prediction.
- K-Neighbors: Options available for the "search method" for KNN algorithm were not explored. This changes how the hyper-parameters of the algorithm are tuned.
- Cross validation was not explored.

Ordinal Regression Assumptions

Also "Ordered Logistic" Regression

- Because it is likelihood based, need to have "enough" data for modeling.
- Proportional odds - coefficients stay the same, and the intercept value changes. Need to verify. Did not do it.
- To verify, would run each category independently, verifying slopes are the same.
- All explanatory variables have the same weight for all categories. Puts them in possible categories, picks the one with the highest probability.

Regression:Limitations and Scaling

- K-Neighbors: if category distribution is skewed, larger categories can dominate, which is what we see in our results.
- Regression does not always scale well, adding covariates can bog down the number of comparisons, especially with model selection
- Random or stratified sampling of data to get a reasonable set size and model selection to cut down number of covariates could help.

Questions