Predicting recipe sentiment

from nutrient composition

Mike Touse

4 March 2017

Outline

Introduction

Recipe success is a high-stakes game

- 10% of the American labor force is employed by restaurants operating on razor-thin margins. Offering a recipe that customers will not like wastes precious resources (time and logistics).
- A top recipe website sees 100 million visits every month. Predicting the success of recipes before receiving feedback allows significant advantage in marketing revenue-generating advertisement space.

Goal:

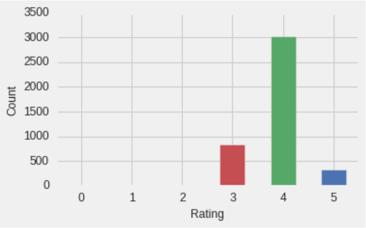
Develop a model to predict recipe success, based only on nutrient quantities, to maximize extensibility across features such as ethnicity and audience.

Data collection and distribution

Process

- Query subsets of recipes using individual ingredients to get ratings and recipe id numbers.
- Concatenate subsets into single list of recipes.
- Query individual recipes from compiled list to capture complete recipe information.
- Parse necessary fields into individual components (nutrients).

Original data distribution:



Unbalanced ratings

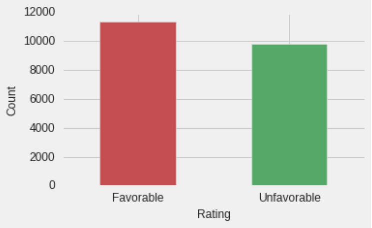
- Initial API responses mostly rated 4

Data collection and distribution

Process

- Query subsets of recipes using individual ingredients to get ratings and recipe id numbers.
- Concatenate subsets into single list of recipes.
- Query individual recipes from compiled list to capture complete recipe information.
- Parse necessary fields into individual components (nutrients).

Final dataset distribution:



Unbalanced ratings

- Initial API responses mostly rated 4
- Adjusted API script to balance dataset
- Grouped ratings in Favorable (4-5) and Unfavorable (0-3)

Recipe response to JSON query

Each includes only nutrients present in the recipe. Example:

```
In [647]: recipes.iloc[0]['nutritionEstimates']

Out[647]: [{'attribute': 'K',
            'description': 'Potassium, K',
            'unit': {'abbreviation': 'g',
             'decimal': True,
             'id': '12485d26-6e69-102c-9a8a-0030485841f8',
             'name': 'gram',
             'plural': 'grams',
             'pluralAbbreviation': 'grams'},
            'value': 0.04},
           {'attribute': 'FLD',
            'description': 'Fluoride, F',
            'unit': {'abbreviation': 'g',
             'decimal': True,
             'id': '12485d26-6e69-102c-9a8a-0030485841f8',
             'name': 'gram',
             'plural': 'grams',
             'pluralAbbreviation': 'grams'},
            'value': 0.0},
           {'attribute': 'PHYSTR',
```

Data overview

- 21,140 recipes
- 113 different nutrients
- Many sparse features
- Scaled to single-serving

Sample descriptive statistics

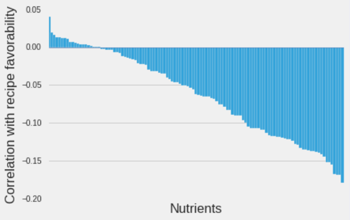
	FAT_KCAL	K	FLD	PHYSTR	VAL_G
count	21140	21140	21140	21140	21140
mean	51.6	0.16	0.0000	0.005	0.19
std	94.8	0.23	0.0000	0.015	0.34
min	0.0	0.00	0.0000	0.000	0.00
25%	4.2	0.01	0.0000	0.000	0.00
50%	23.3	0.08	0.0000	0.001	0.04
75%	62.5	0.21	0.0000	0.005	0.28
max	1810.0	2.95	0.0025	0.630	5.54

Data Exploration

Observations

- Positive and negative ratings are largely overlapped
- No linear discriminants
- Correlations are small and mostly negative

Correlation

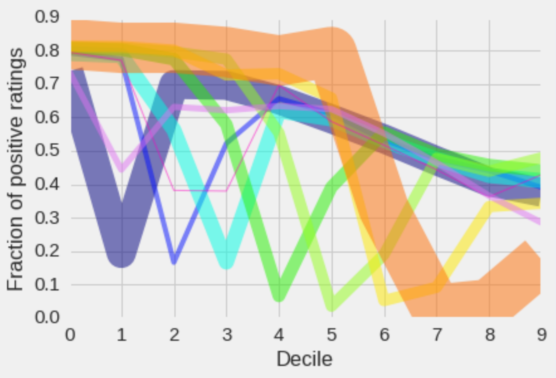


Illustrative example of two most-important nutrients:

Feature positivity

Evaluated portion of recipes rated positively at each decile of each nutrient

Fraction of positive ratings vs. Decile



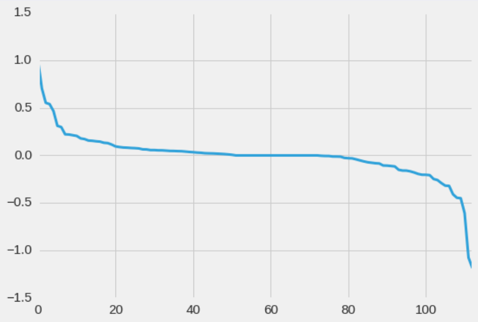
- Most nutrients exhibit single decile with sharp minimum
- Grouped by min decile (line width is number of nutrients)
- Confirms negative correlation

Logistic Regression (LR)

LR Setup

- Scikit-learn StandardScaler
- L1 penalty for sparsity: Zero-weights - little impact
- Minimal sensitivity to regularization parameter
- 80/20 train/test split

Feature Weights

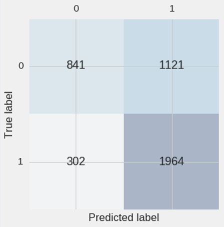


Logistic Regression Results

Performance Metrics

Metric	Value
Accuracy	0.66
Precision	0.64
Recall	0.87
F1	0.73
Area under ROC curve	0.70

Confusion Matrix



Performance observations

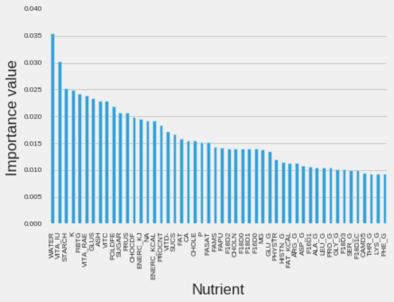
- High False Positive Rate (FPR) causes misleading Recall score (artificially high)

Random Forest

Parameters

- 10,000 estimators
- Gini criterion
- Max features: sqrt
- Unscaled (original per-serving data)

Relative feature importance (top 50)



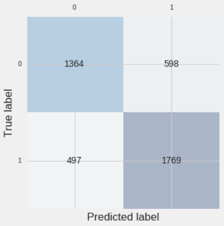
1. Water
2. Vitamin A
3. Starch
4. Potassium
5. Fiber
6. . . .

Random Forest Results

Performance Metrics

Metric	Value
Accuracy	0.74
Precision	0.75
Recall	0.78
F1	0.76
Area under ROC curve	0.80

Confusion Matrix



Performance observations

- Balanced True Positive Rate and True Negative Rate
- 10% increase across all metrics (except Recall) relative to LR classifier

Conclusion

Results

- Correctly predicted positive recipe ratings approximately 75% of the time (depending on metric used) using only nutrients.

Recommendations

- Restaurants should apply predictive model to recipes prior to preparing them for customers and avoid costly logistics (and potential customer turn-off) while collecting feedback.
- Recipe websites should use prior probabilities from model to tune advertisement sales algorithms.
- Recipe developers should use model discoveries to avoid overuse of specific nutrients.
- Despite mediocre predictive performance, favor inclusion of theobromine and caffeine as the nutrients found most positively correlated with favorability (and heavily concentrated in chocolate).