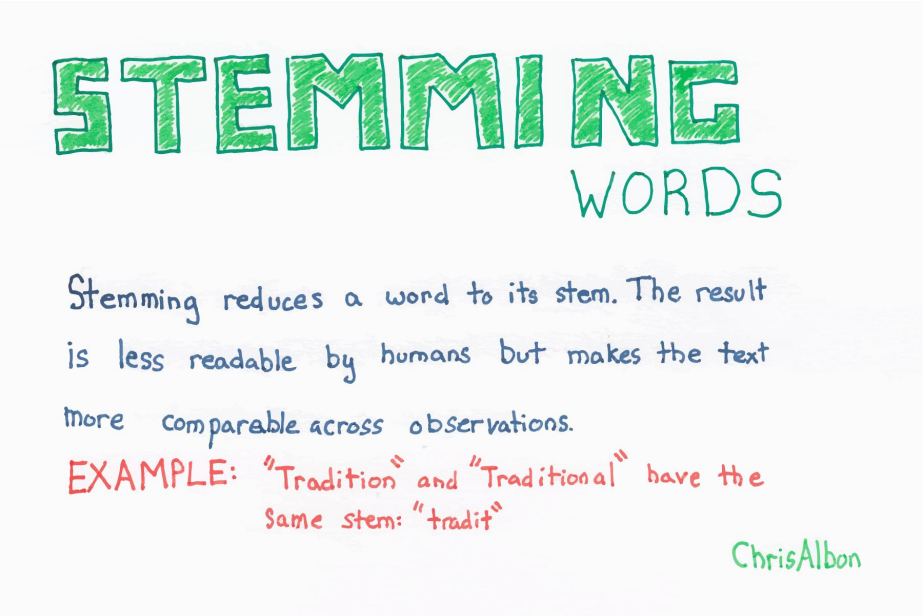
Deliverable 4

Charles Carver, Mingi Jeong, Sam Lensgraf

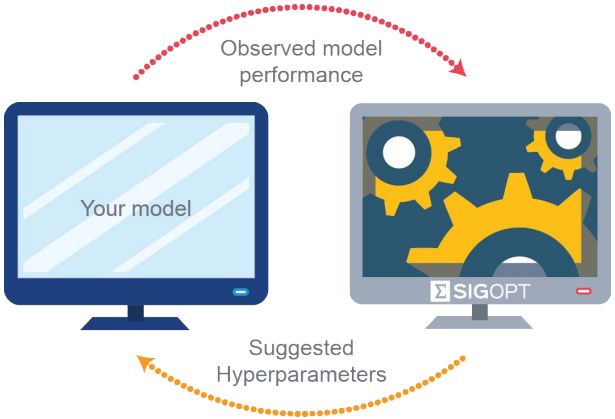
Feature Description: Text-Based Features

- **Old**
  - Found scikit-learn's feature_extraction library & used TfIdfVectorizer
  - Stop words become down-weighted
  - Used with ngram_range = (1, 2)
- **New!**
  - Increased ngram_range to (1, 4)
  - Implemented min_df and set to 2
  - Used PorterStemmer as preprocessor



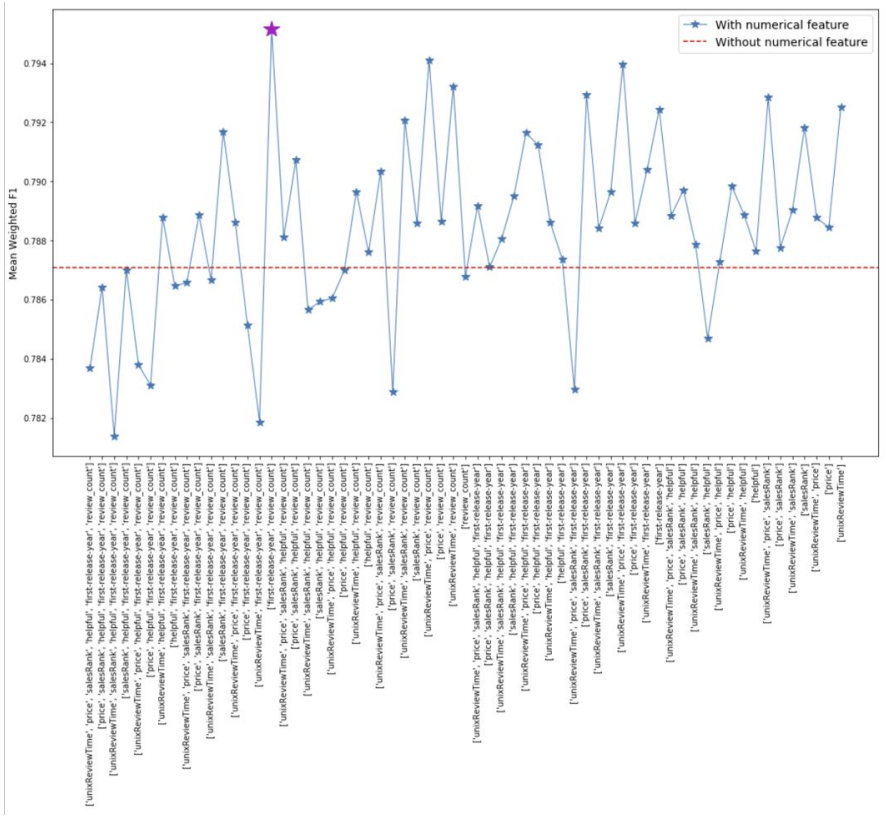
Hyperparameter optimization

- Could not do hyperparameter optimization with large text matrix
- Selected 7000 best features using kbest w/ chi^2 metric.
  - ~2,000,000 columns to 7000!
  - 7000 was sweet spot size. Larger and smaller both reduce performance.
- Trimmed data allowed large grid searches!
  - Logistic Regression (1039 configurations)
  - Gradient Boosting (180 configurations)
  - Support Vector Machine (288 configurations)
  - Random forest (2520 configurations)
- Logistic Regression / SVM were best
- **Logistic Regression**
  - l2 loss function and **C=6.97** were the best choice
- **SVM**
  - Best config: C = 5.5, kernel = linear
  - Performance highly dependent on C



Feature Description: Numerical Features

- **Old**
  - Included numerical fields based on correlation values: **Helpful**
- **New!**
  - Included two new fields: **first-release-year** (if >1990 base year: 1, else: 0), **review_count** (for each product)
  - Found error in preprocessing step:
    - Retained **helpful** column in training, validation phase
    - Accidently retained **UnixReviewTime**, **Price**, **SalesRank** in actual testing phase
  - However, F1 was higher on actual data (0.73 on training / 0.74 on actual test)
  - Led us to reconsider all combinations of textual fields with powerset of numerical features
    - **unixReviewTime**, **price**, **salesRank**, **helpful**, **first-release-year**, **review_count**
    - 2^6 * classifier model * 10-Kfold
  - Found **first-release-year** and **review_count** worked with the best for Logistic Regression in conjunction with textual features



Results: Combined Analysis

- **Old**
  - Random Forest: F1 (10-fold mean) = 0.69
  - K-Nearest Neighbors: F1 (10-fold mean) = 0.58
  - Logistic Regression: F1 (10-fold mean) = 0.73
- **New!**
  - SVM
    - Hyperparameter optimization using grid search
    - C=5.5
    - Best F1 (10-fold mean) = 0.7860
  - Logistic Regression
    - Hyperparameter optimization using grid search
    - C=6.97, class_weight='balanced', max_iter=100000, multi_class='multinomial'
    - Best F1 (10-fold mean) = 0.7905
  - Voting Ensemble
    - Voting classifier (**HARD**): Logistic Regression + SVM
    - Best F1 (10-fold mean) = 0.7891

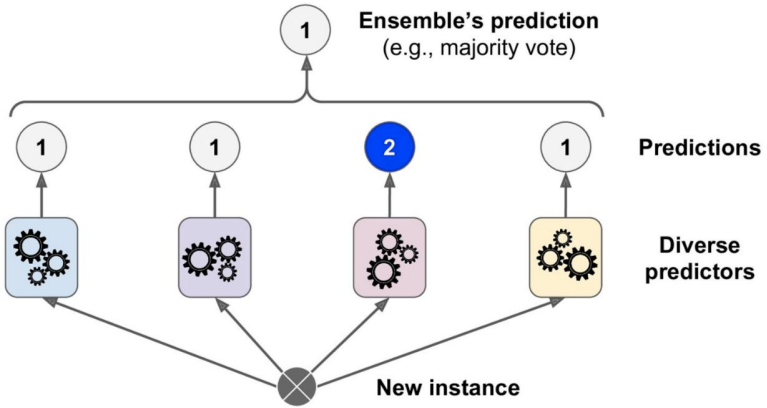


Figure 7-2. Hard voting classifier predictions

Results: Final Implementation

- After hyperparameter optimization, LR achieved best mean 10-fold F1 scores
- Considers the most useful numerical features to be **first-release-year, review_count**
- **KBest** feature selector for classifier
- Mean = 0.790, STDV = 0.0131, 1.7% RSD
- Large improvement from D3 (mean = 0.730 using same model)

Min	0.771
Max	0.810
Mean	0.790

Fold	**F1**
1	0.786
2	0.797
3	0.807
4	0.782
5	0.778
6	0.798
7	0.792
8	0.777
9	0.810
10	0.771

Credits

Charles Carver

- ✅ Slides
- ✅ Deliverable code
- ✅ Had even more fun!

Mingi Jeong

- ✅ Slides
- ✅ Numerical feature analysis
- ✅ Had even more fun!

Sam Lensgraf

- ✅ Slides
- ✅ Text-based feature analysis
- ✅ Had even more fun!