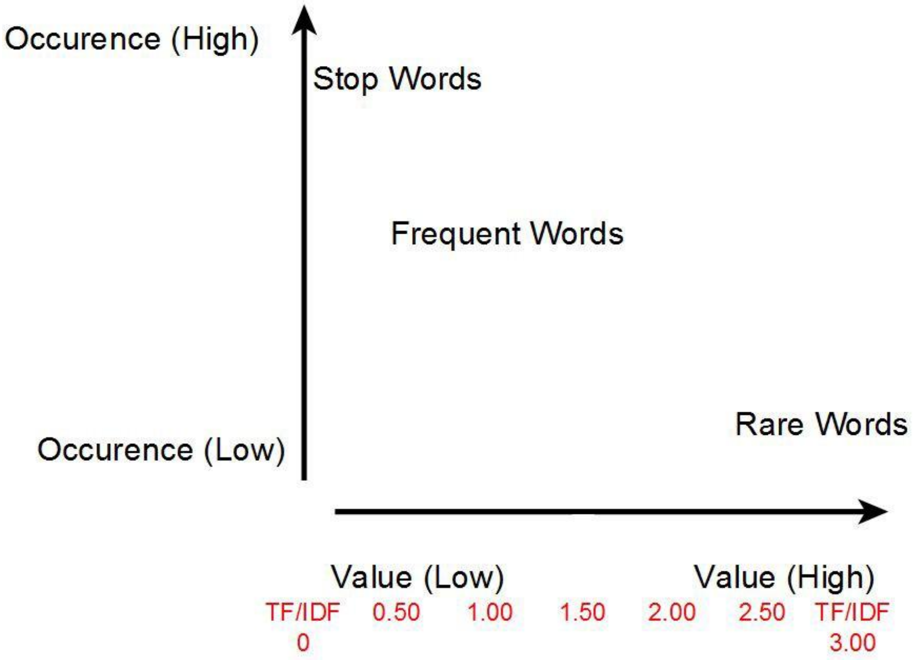
Deliverable 3

Charles Carver, Mingi Jeong, Sam Lensgraf

Feature Description: **Text-Based Features**

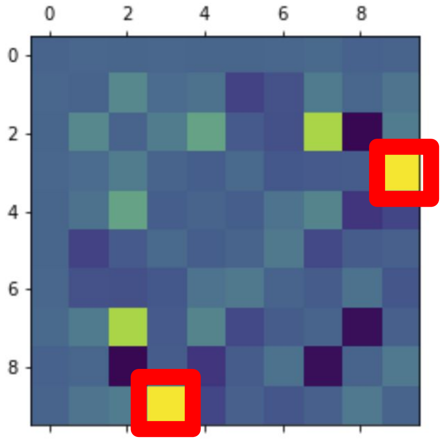
- **Feature:** lowercase concatenation of summary and review data for each product
- Tried hand-implemented Bag of Words
  - Filtered for “common words” to eliminate misspellings
  - Removed stop words
  - Attempted 1,2,3-grams
  - **Poor performance** (runtime and learning results)
- Found scikit-learn’s feature_extraction library & used TfIdfVectorizer
  - Stop words become down-weighted
  - Used with ngram_range=(1,2)
  - **Much better runtime and learning results**!

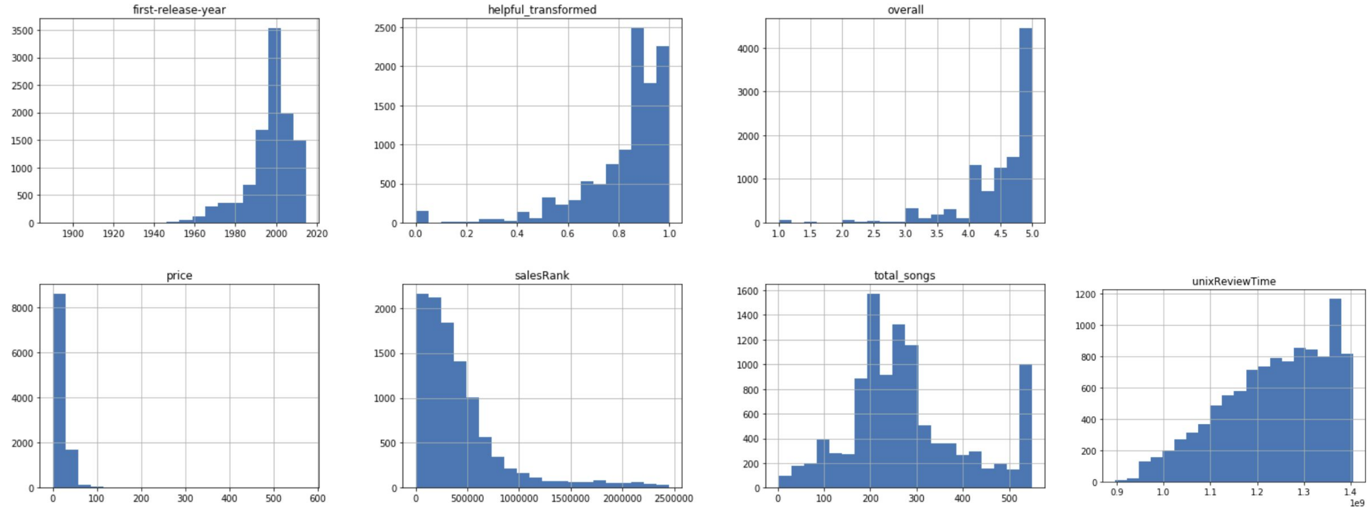


Feature Description: **Numerical Features**

- **Primary Data Processing**

- **Input**: SalesRank, **Helpful**, Price, UnixReviewTime, **first-release-year,** total songs, total related
- Pearson correlation coefficient ***r*** analysis for **‘overall’**
- Imputation (NaN values)
- Categorical variables (root-genre, label)

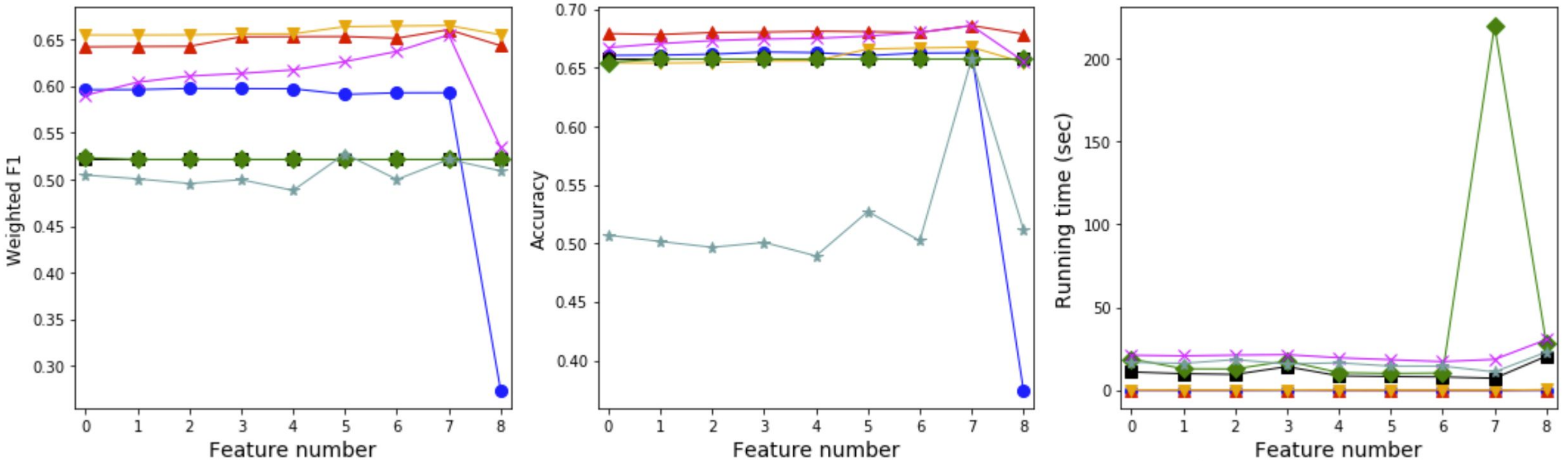


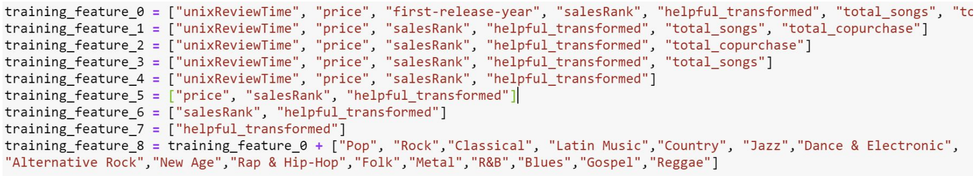


helpful	**0.337780**
unixReviewTime	0.049034
amazon-id	0.014093
title	0.009592
reviewerID	0.003343
price	-0.013251
salesRank	-0.014595
first-release-year	-0.018330
artist	-0.026578

Results: **Numerical Features**

- Different combination of numerical features (9) * classifier models (7)
- Best result: **Random Forest** with “**Helpful**” feature
- **F1** weighted score by k fold: **mean** 0.665 (dropping NaN: over 0.9)





```
training_feature_0 = ["unixReviewTime", "price", "first-release-year", "salesRank", "helpful_transformed", "total_songs", "tc
training_feature_1 = ["unixReviewTime", "price", "salesRank", "helpful_transformed", "total_songs", "total_copurchase"]
training_feature_2 = ["unixReviewTime", "price", "salesRank", "helpful_transformed", "total_copurchase"]
training_feature_3 = ["unixReviewTime", "price", "salesRank", "helpful_transformed", "total_songs"]
training_feature_4 = ["unixReviewTime", "price", "salesRank", "helpful_transformed"]
training_feature_5 = ["price", "salesRank", "helpful_transformed"]
training_feature_6 = ["salesRank", "helpful_transformed"]
training_feature_7 = ["helpful_transformed"]
training_feature_8 = training_feature_0 + ["Pop", "Rock","Classical", "Latin Music","Country", "Jazz","Dance & Electronic",
"Alternative Rock","New Age","Rap & Hip-Hop","Folk","Metal","R&B","Blues","Gospel","Reggae"]
```

Results: **Combined Analysis**

- **Combined Analysis**
  - Process textual data
  - Process numerical data
  - Combine into unified training set
- **Random Forest:** F1=0.69
- **K-Nearest Neighbors:** F1=0.58
- **Logistic Regression:** **F1=0.73**
  - 100,000 max iterations
  - Class weights automatically balanced

Min	0.712
Max	0.745
Mean	0.729

Fold	**F1**
1	0.712
2	0.723
3	0.729
4	0.734
5	0.735
6	0.716
7	0.716
8	0.737
9	0.745
10	0.738

Credits

Charles Carver

☑ Slides

☑ Deliverable code + GitHub

☑ Had fun!

Mingi Jeong

☑ Slides

☑ Numerical feature analysis

☑ Had fun!

Sam Lensgraf

☑ Slides

☑ Text-based feature analysis

☑ Had fun!