Million Song Dataset

Year Prediction, Clustering, and Natural Language Processing

RYAN ELSON

Problem Statement

For marketing purposes on e-commerce sites and song recommendations on music platforms, music analysis is important.

Using a model with the response variable as the year a song was released, businesses can predict if a song is from a certain year or if it has similarities to songs from that year. This information can be used to market to consumers who may like music from that era or to build a better music platform which can cater to a consumer's individual musical preferences.

To predict the year a song came out, I use various models including Linear Regression, Ridge, LASSO, Boosting, and others.

In addition, I use k-means clustering to see if songs fall into natural clusters.

Finally, I use natural language processing (NLP) to find high-value words that may be indicative of a particular decade.

Data Set

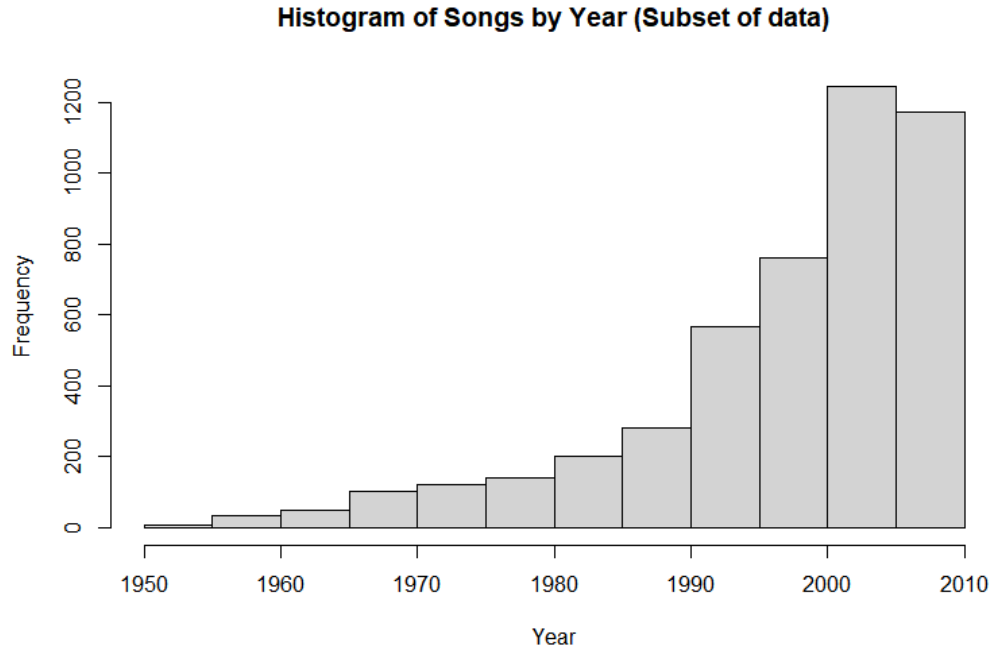
- 10,000 song dataset from the Million Song Dataset (MSD) website (1)
- Observations: 4680 total songs with year available. Other songs did not include year.
- Predictors: 112 features from song files; cleaned/processed to 98 features for year prediction
  - Timbre Average (x12)
  - Timbre Covariance (x78)
  - Artist name, artist hotness, artist familiarity
  - Song title, album title
  - Tempo, loudness, duration
  - Unique IDs…
- Response: Year

Challenges

- Size of dataset and number of features
  - Initially considered using pre-processed Million Song dataset from UCI Machine Repository (2)
  - 500,000+ observations and 90 features had high computational cost, some models could not be built on my local computer
  - Dataset only included year and timbre values so was not appropriate for NLP and did not include other desired features like tempo, loudness, or duration
  - Can address with smaller datasets, cloud-computing, or dimension reduction techniques. Opted to build a smaller dataset that also included song names to be used for NLP
- Music generally does not exhibit large changes from year to year
  - Makes predictions and classification challenging
  - Can attempt to address by using ensemble methods
  - Use regression techniques to predict year and find test error rather than classification for year or decade

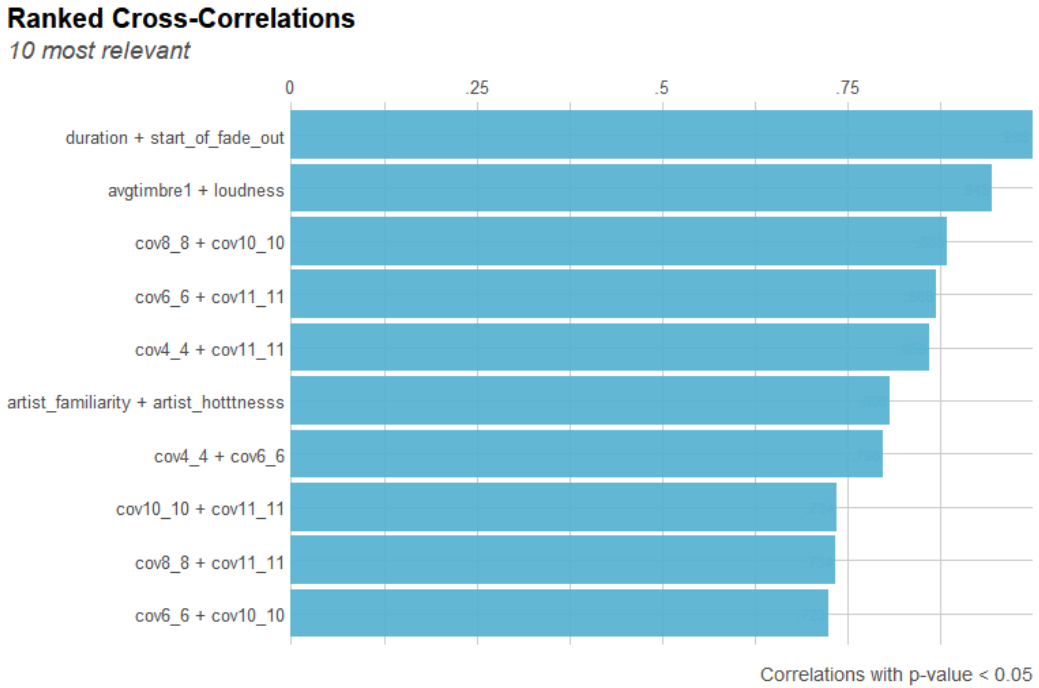
Exploratory Data Analysis

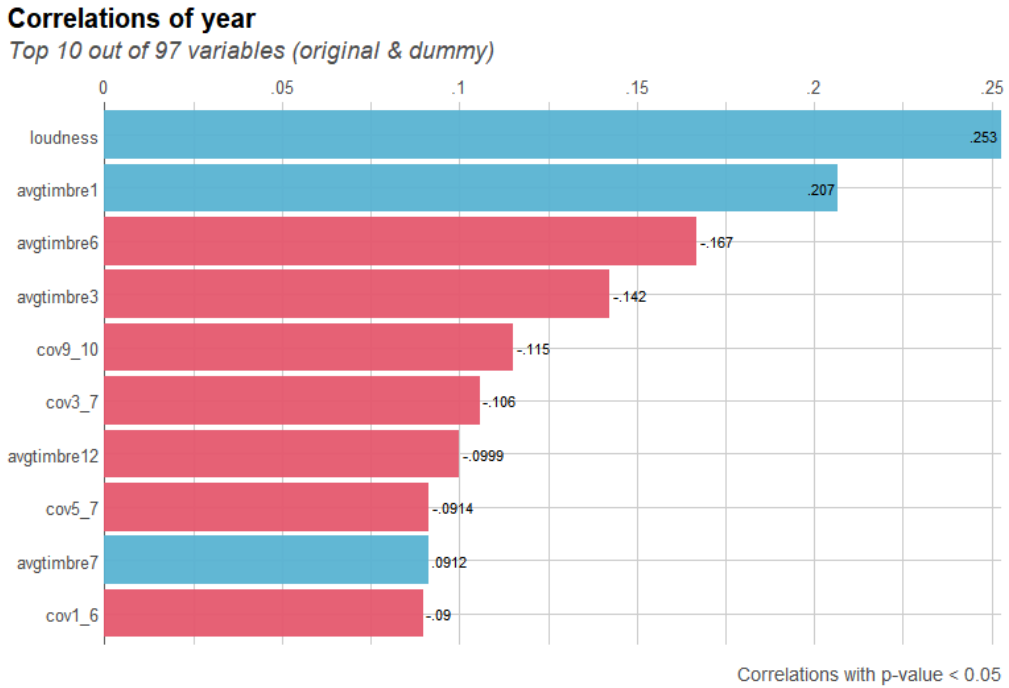
- Year range: 1926 to 2010
- Dataset is unbalanced
  - Without addressing, makes it inappropriate for classification
  - Use regression to predict years instead of classification
    - More meaningful intuitively
    - Example: A one-year error in prediction (1979 vs 1980) vs a nine-year error (1989 vs 1980). Leads to an incorrect classification for a one-year error vs a correct classification with a nine-year error
- Few songs from decades before 1950
  - Would lead to NLP results that aren't meaningful
  - Removed pre-1950 songs in order to use the same dataset for all parts of the analysis



Exploratory Data Analysis

- Variable Correlation
  - Some correlated predictor variables, mainly for certain timbre covariance terms
  - Loudness was the most correlated predictor with year.





Methodology

- Read in data from 10,000 song files
- Keep only songs with years available and from the 1950s onward
- Remove unnecessary features, like unique IDs and features not available for some songs (e.g., Latitude, Longitude). Even genre was not available for all songs, so it was not used either. This may be because some songs can fit multiple categories.
- Process song data to calculate timbre averages and covariance values
- Resultant dataset: 4664 songs, 98 features (for song prediction, not including text-based features)
- Scale feature data before analyzing

Methodology

- Models for Year Prediction:
  - Linear Regression
  - Stepwise Regression (AIC)
  - Ridge
  - LASSO
  - Partial Least Squares (PLS)
  - Random Forest
  - Boosting
- Train/Test Split: 75/25%
- Cross-validation: 30 iterations with random train/test splits when building models
  - Used only 30 iterations due to computational time. Still took approximately three hours to run.
  - Ridge, LASSO: Choose optimal lambda value for each iteration
  - PLS, Boosting: 10-fold CV for each iteration
  - Random Forest: 32 random predictors as candidates for each split, build 500 trees
- Score models using MSE from test dataset.
- Overall model score = average of MSE scores from 30 iterations

Methodology

- K-means Clustering
  - Build 15 models (from k = 1 to k = 15 clusters)
  - Evaluate with elbow method
  - Choose optimal number of clusters using elbow method and silhouette method
  - Visualize clusters and look for groupings by year

- Natural Language Processing (NLP)
  - Term Frequency-Inverse Document Frequency (TF-IDF)
  - Choose high-value words by decade from song titles
  - Method is based on how common a term is within the entire corpus and frequency within a class (decade)

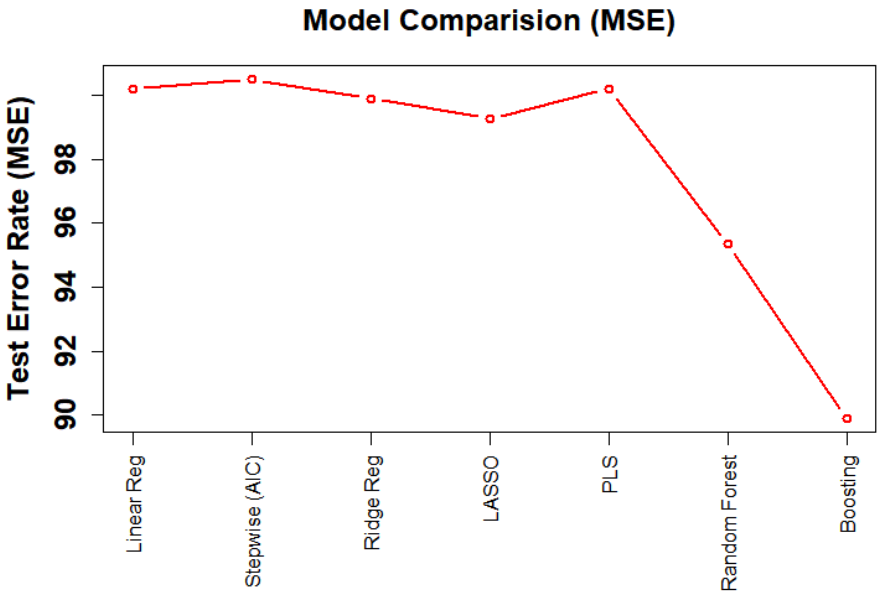
Results

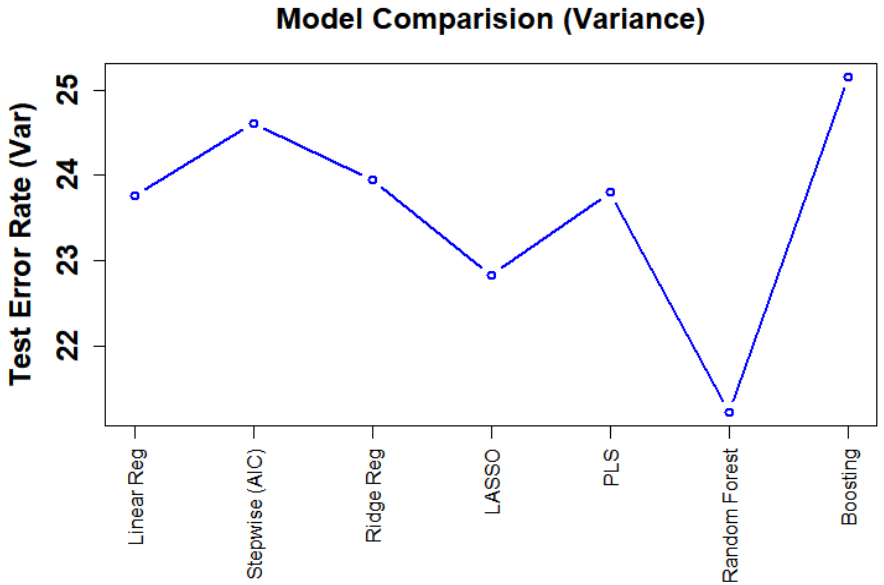
- Analysis completed using R programming language and RStudio software
- Year Prediction
  - Test Error Rates

	Test Error (MSE)	Variance
Linear Regression	100.19	23.76
Stepwise Regression (AIC)	100.49	24.61
Ridge Regression	99.89	23.95
LASSO	99.26	22.84
Partial Least Squares (PLS)	100.19	23.80
Random Forest	95.34	21.23
Boosting	89.90	25.15

Results

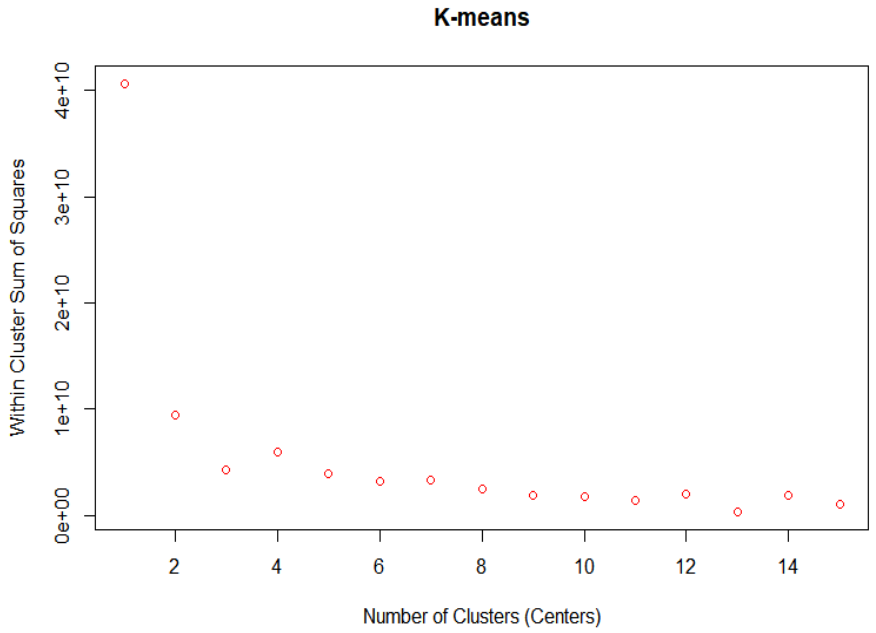
- Boosting had smallest error rate
  - Had largest variance
- Random Forest had smallest variance
  - Second-smallest error rate

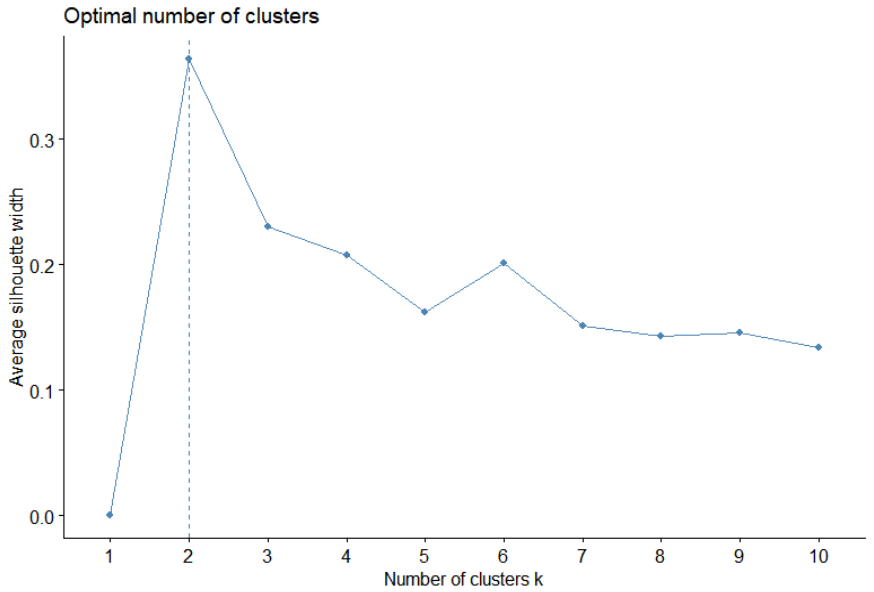




Results

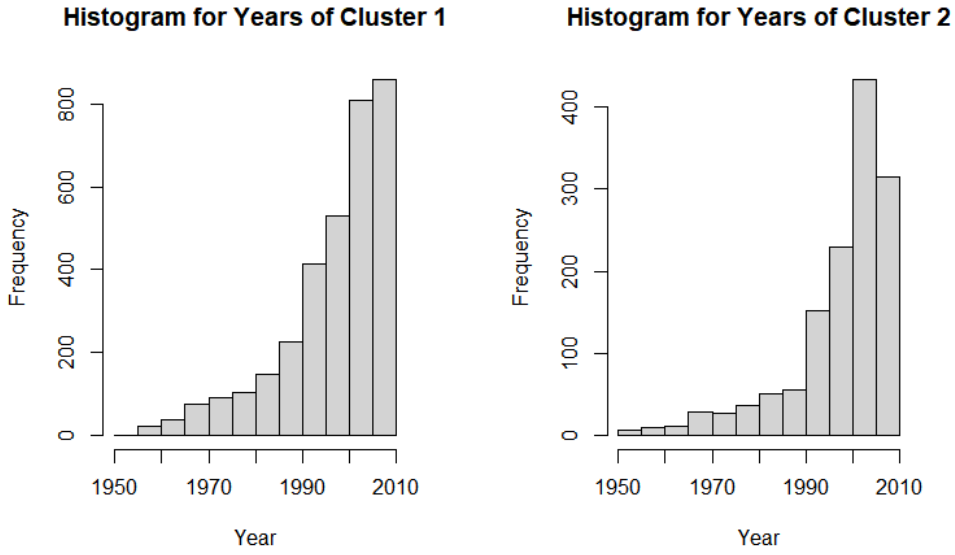
- K-means
  - Optimal clustering at k = 2 clusters
  - Using elbow method and silhouette method

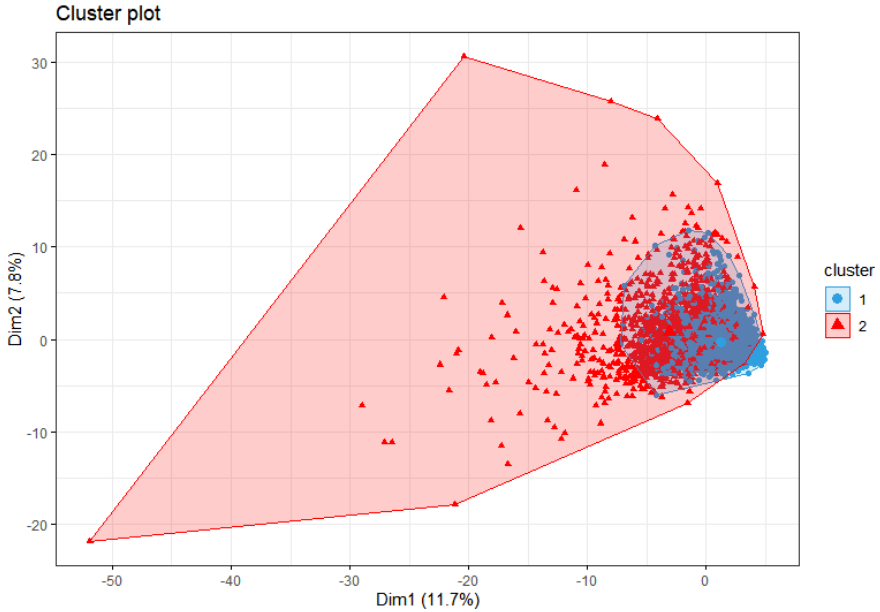




Results

- K-means
  - Clusters overlap
  - Clusters don’t provide a good split by year
  - Doesn’t seem beneficial for finding accurate clusters of songs by year





Results

- Natural Language Processing
  - Selection of interesting top words by decade

1950s – carnaval, cantando, conguitos, faith

1960s – watermelon, quiet, spoken, word, lp

1970s – bit, bully, version, digital, remaster, lp, waste

1980s – start, version, make, night, halloween, windpower

1990s – version, lp, live, explicit, red, album, mix

2000s – remix, version, live, amended, album, explicit, featuring, edit, mix, club

2010s – feat, diamond, cannonball, minion, twins, underworld

Results

- Natural Language Processing
  - Words including version, lp, album, and explicit are top words in multiple decades
    - Opted not to include in stop words list to understand how top words change over time.
  - 1950s and 2010s had too few songs
    - Multiple words tied as top words, even when only present once per decade. Some top terms not very meaningful.
  - Observations
    - 1950s – multiple words from other languages
    - 1970s – “bit,” “digital,” and “remaster” denote a change from analog to digital
    - 1990s – “explicit” and “live” start showing up as explicit language is increasingly included in songs. Last appearance of “lp” as a top term.
    - 2000s – “featuring” and “feat” start showing up in song titles, noting collaboration with another artist
  - Unexpected top words
    - 1960s – “watermelon” ties for top word
    - 1980s – “Halloween” and “windpower” are top 10 words

Conclusion

- Boosting provided the smallest prediction error when predicting a song's year. Although it had the highest variance, it was not much higher than some other models and because of the vastly superior predictive performance, I believe this is the best model.
- Random Forest had the smallest variance and second-smallest test error. Possible choice for best model if putting higher importance on small variance.
- Clustering did not provide meaningful groupings with regard to song year
- Natural Language Processing provided interesting top words, in some cases allowing us to draw conclusions about music trends over time
- Future work includes gathering more complete information for certain features (e.g., Latitude, Longitude, Genre) and seeing if these additional features improve predictive performance or clustering. Can also predict year of all songs on an album and take the mean or median as the final prediction for all songs, again checking to see if this improves prediction error.

References

1. Million Song Dataset, official website by Thierry Bertin-Mahieux, available at: http://millionsongdataset.com/

2. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository. “Year Prediction MSD“ [https://archive.ics.uci.edu/ml/datasets/yearpredictionmsd, http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science.

3. Bertin-Mahieux, Thierry, Daniel PW Ellis, Brian Whitman, and Paul Lamere. "The million song dataset." (2011): 591-596.

4. Jayaprakash Nallathambi. “R Series — K Means Clustering (Silhouette) - CodeSmart - Medium.” Medium. CodeSmart, June 18, 2018. https://medium.com/codesmart/r-series-k-means-clustering-silhouette-794774b46586.

5. Fonseca, Luiz. “Clustering Analysis in R Using K-Means - towards Data Science.” Medium. Towards Data Science, August 15, 2019. https://towardsdatascience.com/clustering-analysis-in-r-using-k-means-73eca4fb7967.