

Course Review Analyzer

An NLP approach to analyzing and extracting meaning from course reviews

Developed by Reggie Bain



Background + Motivation

NLP Based Course Review Analysis

- Soliciting and gaining insight from course reviews critical
 - Need to solicit a large number of reviews
 - Draw actionable insights
 - Analyze/summarize without just focusing on hyper positive/negative reviews

Goals

- Train models to predict sentiment, identify nonsense
- Build app that for users to analyze reviews in real time
- Our Targets → **Sentiment, identify gibberish**
- Our Features → **NLP features extracted from reviews**




+



Datasets

Data Sources

- [Coursera Reviews Dataset \(Kaggle\)](#)
- [Amazon Product Reviews \(Kaggle\)](#)
- [Gibberish Text Amazon Data \(Kaggle\)](#)
- [Master Dataset with Features Extracted and Models Saved \(Kaggle - by Reggie Bain\)](#)



Gibberish Text Classification ⋮


John Wackerow · Updated 5 year...

Usability 6.8 · 180 MB

2 Files (CSV)

9

+



Amazon Reviews for Sentiment Analysis ⋮


Adam Bittlingmayer · Updated 5 y...

Usability 6.9 · 517 MB

2 Files (other)

1002

+



Course Reviews on Coursera ⋮


Muhammad Nakhaee · Updated 5 ...

Usability 8.2 · 35 MB

2 Files (CSV)

35

=



Reviews Analyzer Data ⋮

Reggie Bain · Updated 14 days ago

Usability 3.5 · 751 MB

19 Files (other, CSV, JSON)

1

KPIs + Stakeholders

Key Performance Indicators

1. Pipeline can parse unstructured review inputs and predict the sentiment using NLP
2. Models assess positive/negative sentiment better than random guessing (50/50) or other relevant baselines.
3. Filter meaningless entries, create app to allow user input

Stakeholders

1. Course **instructors** to improve their courses or those in “voice of customer” type roles in industry
2. **Administrators** who want to produce broad performance metrics for staff
3. **Students** who may be able to assess the quality of a course or product based on review summaries

Sentiment Analysis



Positive



Negative

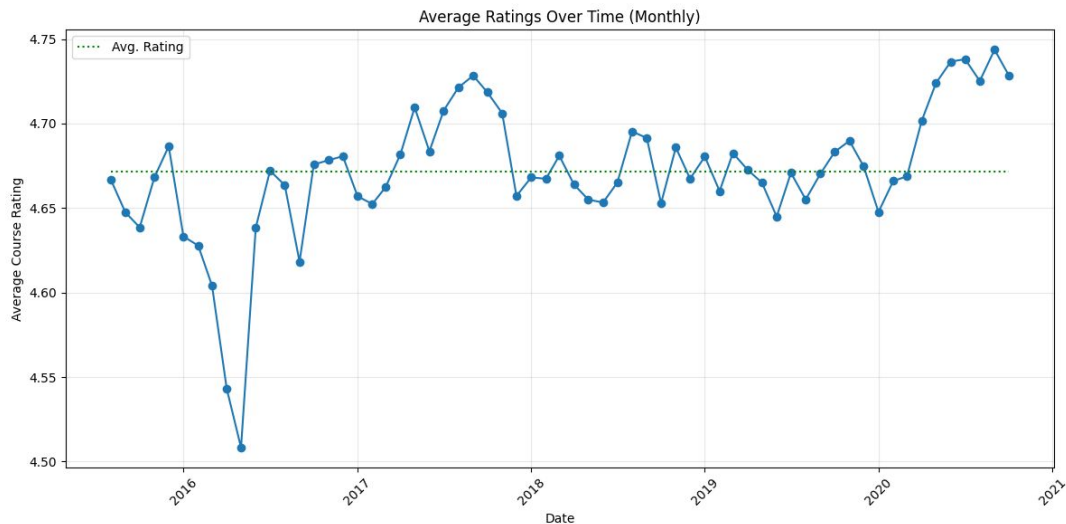
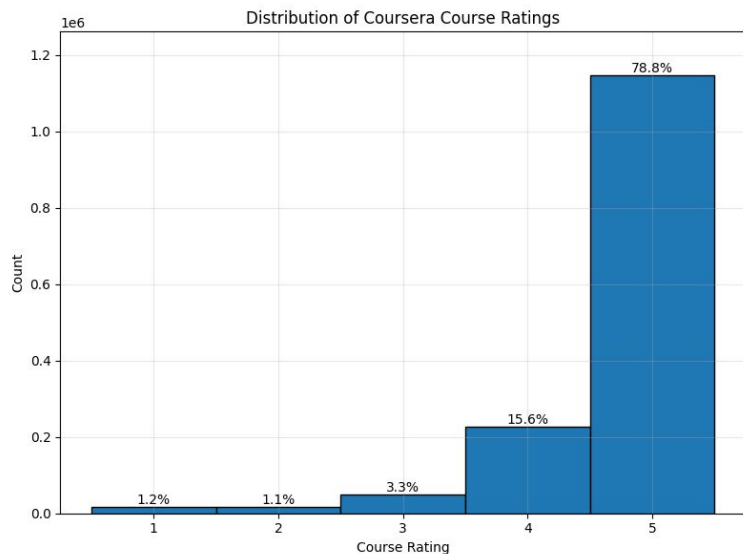


Neutral



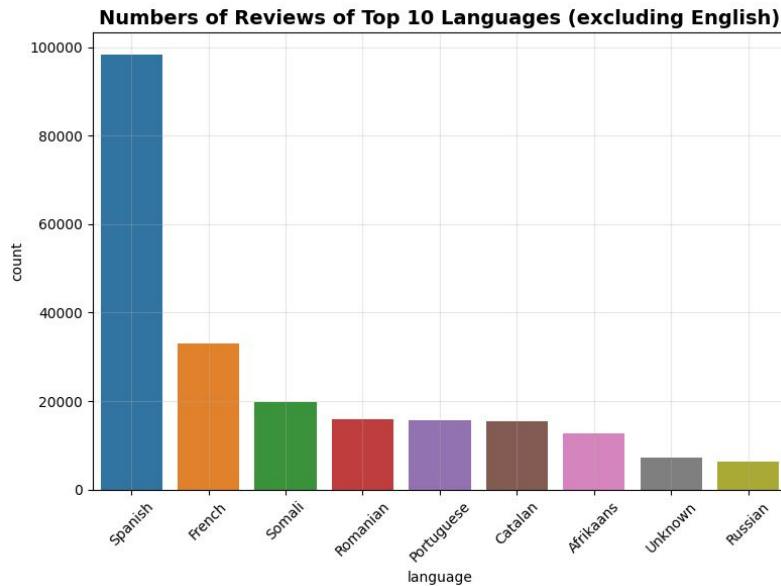
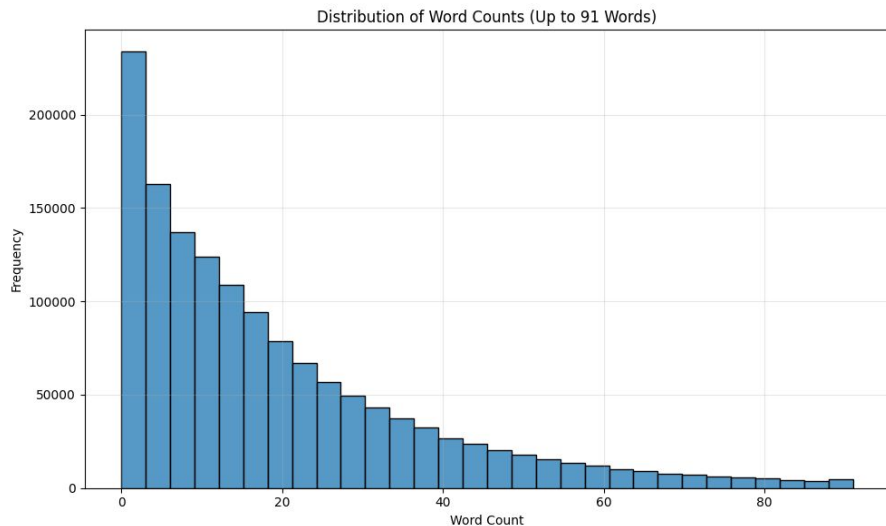
Exploratory Data Analysis (EDA)

Distributions of Course Ratings



Exploratory Data Analysis (EDA)

Distributions of Word Counts & Languages



Feature Engineering + Selection

NLP Features

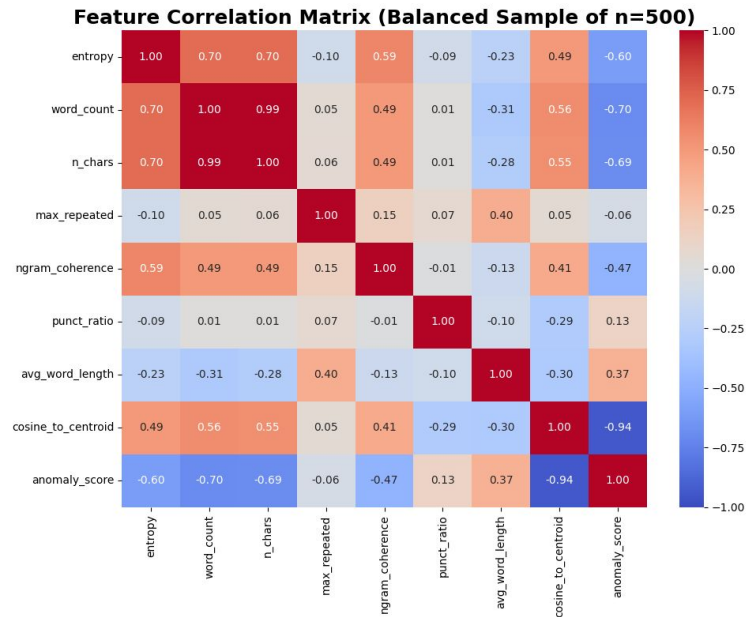
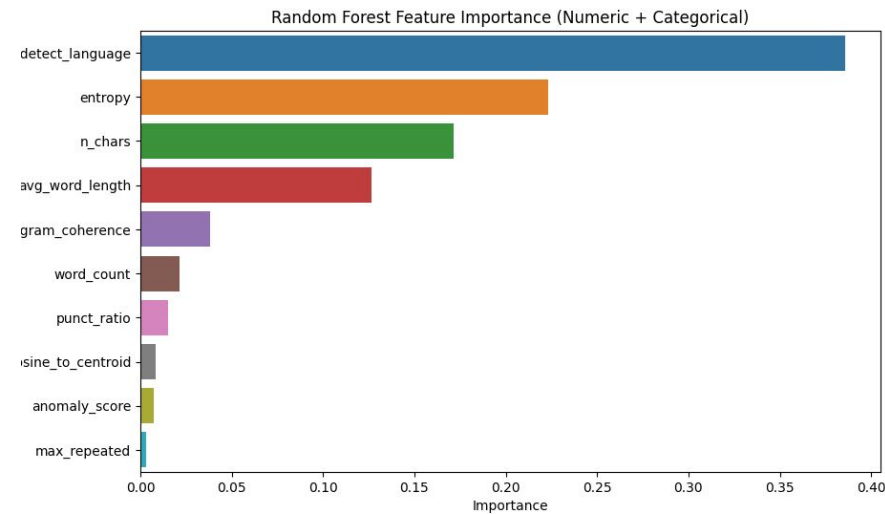
- **Entropy** - use probability a character or word appears in a given language model
- **Cosine-to-Centroid** and **Anomaly Score** - calculate embeddings and find distance from embedding centroid
- **N-gram coherence** - Measures how frequently word sequences appear in natural language to help identify if text uses realistic word combinations of *n*-grams.
- **Polarity** - Sentiment score using VADER lexicon used to capture the tone/intensity of the review
- **Identify the Language** - use Python package to ID language where possible
- Others include: word count, average word length, maximum repeated character, punctuation ratio, exclamation count

$$H(X) := - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

$$\text{N-gram Coherence} = \frac{\text{Number of Valid N-grams in Text}}{\text{Total Number of N-grams in Text}}$$

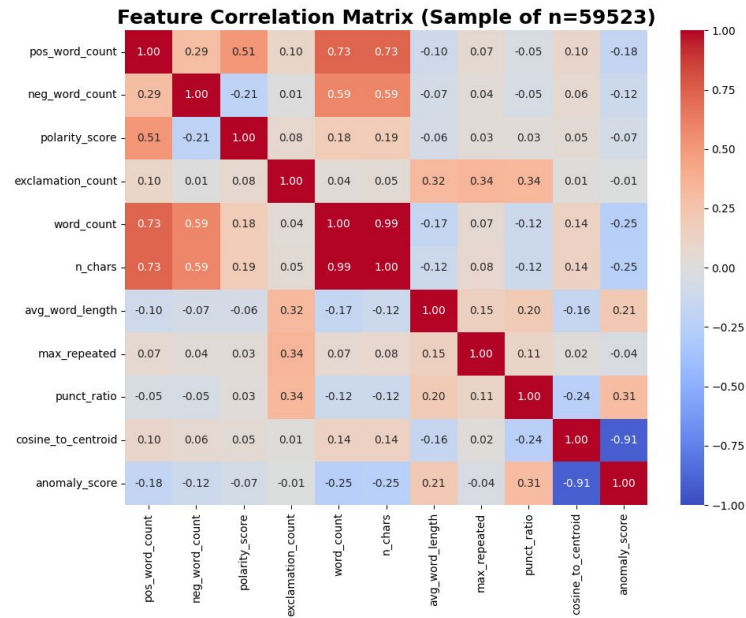
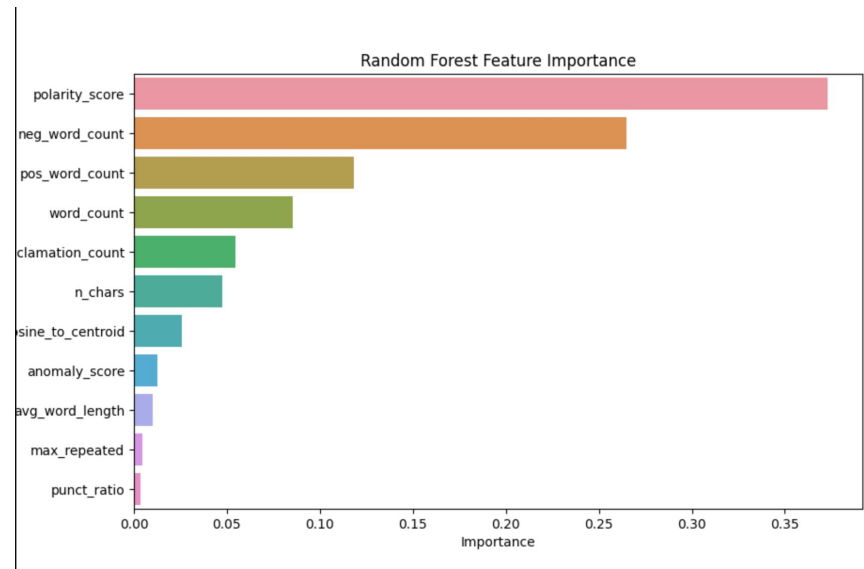
Feature Engineering + Selection

Gibberish Detector Features



Feature Engineering + Selection

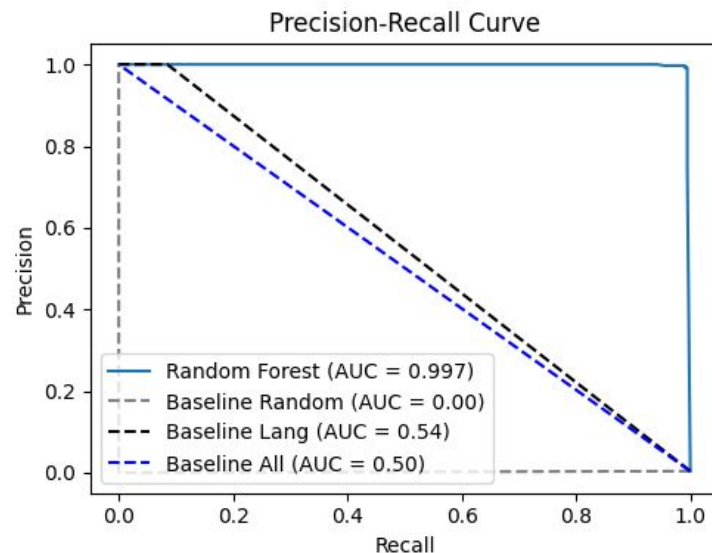
Sentiment Analyzer Features



Results

Performance of Gibberish Detector

Model	Best Parameters	Precision (Gibberish)	Recall (Gibberish)	F1-Score (Gibberish)	ROC AUC	PR AUC
Logistic Regression	{'C': 10, 'solver': 'lbfgs'}	0.981579	0.992021	0.986772	0.998902	0.995229
Random Forest	{'max_depth': 10, 'n_estimators': 50}	0.997319	0.989362	0.993324	0.997338	0.996728
SVM	{'C': 10, 'kernel': 'linear'}	0.98939	0.992021	0.990704	0.999187	0.993797
Gradient Boosting	{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 100}	0.989333	0.986702	0.988016	0.994653	0.982619
Baseline (Random Guess)	Gibberish Prob = 0.00358	0	0	0	0.49825	0.00178649
Baseline (Lang Not Detected)	cannot_detect_language = 1	1	0.0851064	0.156863	0.542553	0.544188
Baseline (None are gibberish)	None Gibberish	0	0	0	0.5	0.501786



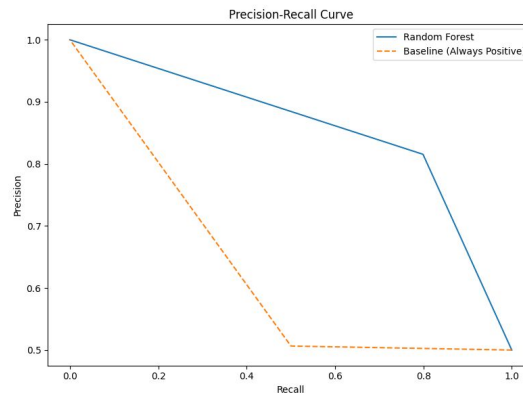
Results

Performance of Sentiment Analyzer

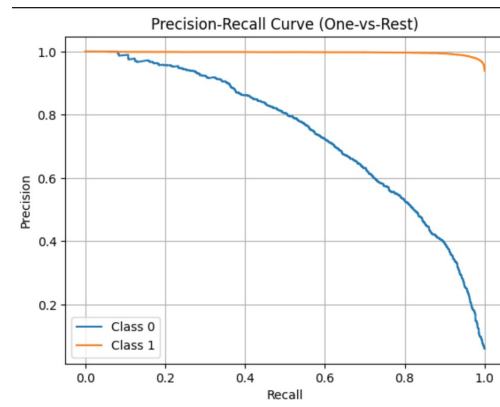
- **Approach 1** - Balance number of positive/negative reviews

Model	Accuracy	F1 Score	Precision	Recall	Best Hyperparameters
Logistic Regression	0.75626	0.755858	0.757963	0.75626	{'C': 0.01}
Random Forest	0.812465	0.812458	0.812512	0.812465	{'max_depth': None, 'n_estimators': 200}
XGBoost Classifier	0.782693	0.782604	0.783161	0.782693	{'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 200}
Baseline (Random Guess)	0.5064	0.506381	0.506401	0.5064	N/A
Distilbert-base-uncased (out-of-box)	0.526989	0.421286	0.600188	0.526989	N/A
Distilbert-base-uncased (fine-tuned)	0.913189	0.913172	0.913509	0.913189	N/A

Classical Features



Fine Tuned NN



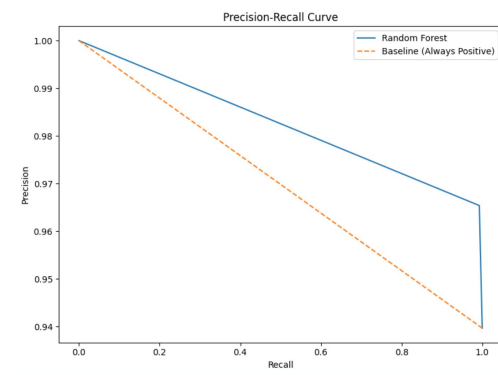
Results

Performance of Sentiment Analyzer

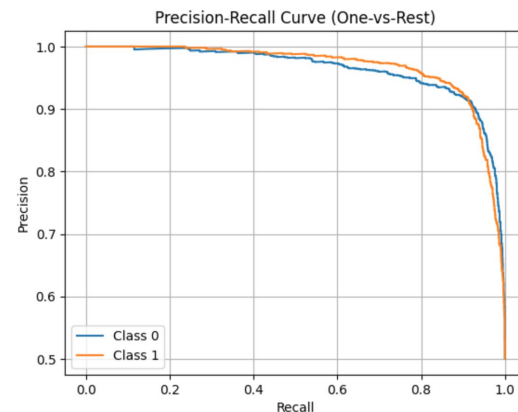
- *Approach 2* - Original highly unbalanced review set

Model	Accuracy	F1 Score	Precision	Recall	Best Hyperparameters
Logistic Regression	0.791821	0.842668	0.929867	0.791821	{'C': 1}
Random Forest	0.959875	0.954458	0.95547	0.959875	{'max_depth': None, 'n_estimators': 200}
XGBoost Classifier	0.827681	0.867041	0.934121	0.827681	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}
Baseline (Always Positive)	0.939662	0.910431	0.882964	0.939662	N/A
Distilbert-base-uncased (out-of-box)	0.250722	0.345742	0.853837	0.250722	
Distilbert-base-uncased (fine tuned)	0.962897	0.959677	0.959309	0.962897	

Classical Features



Fine Tuned NN



Conclusions

Gibberish Classifier

- Showed 500+% increase in F1 over just using language detection, beat all baselines in recall of actual gibberish even when precision was similar
- Raised threshold to reduce false positives on short reviews in deployment

Sentiment Analyzer

- Classical ML models outperformed baseline
 - +60.5% F1 (balanced)
 - +8.2% (unbalanced)
- Fine-tuned DistilBERT performed best
 - +80.4% F1 (balanced)
 - +5.4% (unbalanced)
- 100%+ boost vs. out-of-the-box model
- Achieved with just 1 training epoch

Future Work

Gibberish Classifier

- Stress test/fine tune on edge cases such as long strings of real words arranged in random, meaningless order.
- Obtain or create much larger set of labeled gibberish reviews via hand labeling and/or generating gibberish of specified structure using an LLM.

Sentiment Analyzer

- More training data with larger set of explicitly negative reviews of varying length and structure.
- Continue fine tuning of a pre-trained neural network for many more epochs (50+ or however many needed until validation performance dips).

Web Application

What do students think of your course?

- Try your at: <https://reviews-analyzer-bain.streamlit.app/>
- Enter your own sample course reviews!



Course Review Analyzer



How do students feel about your class?

App Description

This app uses NLP techniques to analyze course reviews. It can track the sentiment, provide a summary, give you an overall score, and provide constructive feedback based on your reviews.

Models Used:

- *Sentiment Analysis:* `nlpTown/bert-base-multilingual-uncased-sentiment` (Hugging Face)
- *Summarization:* `t5-small` (Hugging Face)
- *Rating:* Calculated locally from sentiment results
- *Feedback:* `gpt-3.5-turbo` (OpenAI)

Reviews

Enter reviews (one per line) or use the sample reviews shown

This course was amazing, I learned so much!
Terrible experience, the instructor was unprepared.
It was okay, nothing special but not bad either.
Loved the practical examples, really helpful!
Waste of time, content was outdated.

Actions

Click on an option below. *Note* that summary/feedback should only be run after analyzing sentiment

Analyze Sentiment

Summarize Reviews

Get Overall Rating

Get Feedback