Yelp Bites: Predicting Restaurant Success Through Review Sentiment Analysis

Rhett Burham, Joshua Kravath, and Sophie Shayne

The Problem

Problem: Customer feedback is essential for any business. However, objective sentiment may be blurred by the ambiguity of Yelp's star system.

Cases where stars could be misleading:

- Users with tendencies to leave 5 or 1 stars – skewing average stars for businesses with low frequency of review
- Reviews may be garnered through promotion
- Some reviews may be astroturfed: fake reviews skewing the reviews to be positive or negative.

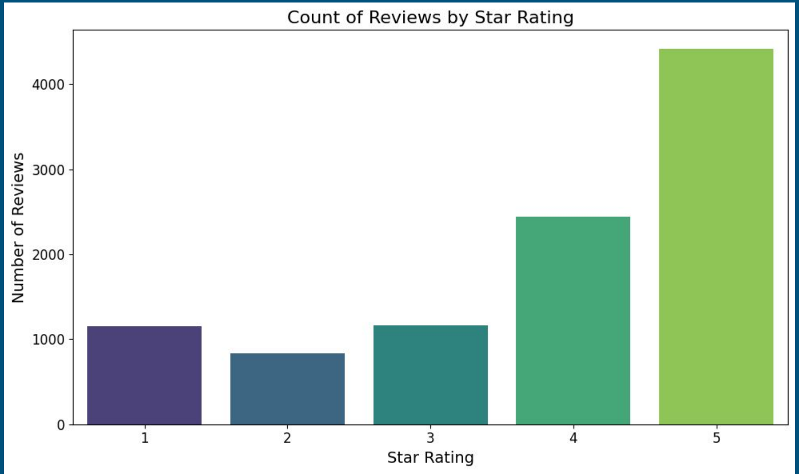
Our goal: Choose a model that predicts Yelp review sentiment (positive, neutral, or negative) using the textual review.

Applications of the model:

Integrity: our model could be used to detect review fraud

Clarity: star selection is discrete going from 1-5 in increments of 1

The Yelp Open Dataset



Raw merged dataset: 6.5 million reviews entries and 150K businesses

Filtered for restaurants in Tampa, Fl: 40,109 review entries, 530 business

Randomly selected 10,000 entries

1. Review text: cleaned and modified into tokenized forms for analysis

   “food”, “place”, “good”, “great”, “service” are examples of **non–stopwords**

2. Stars: What the “true sentiment” of the review.

   Organized into negative (1-2 stars), neutral(3 stars), and positive(4-5)

What are the approaches we will use to explore?

Machine Learning Techniques for NLP:

- Start with simple models: **Naive Bayes** & **Logistic Regression**
- Apply advanced techniques: **LDA**, **QDA**, & **Principal Components Analysis (PCA)**

These techniques provide quick classification through tokenizing and vectorizing the data.

Model Evaluation:

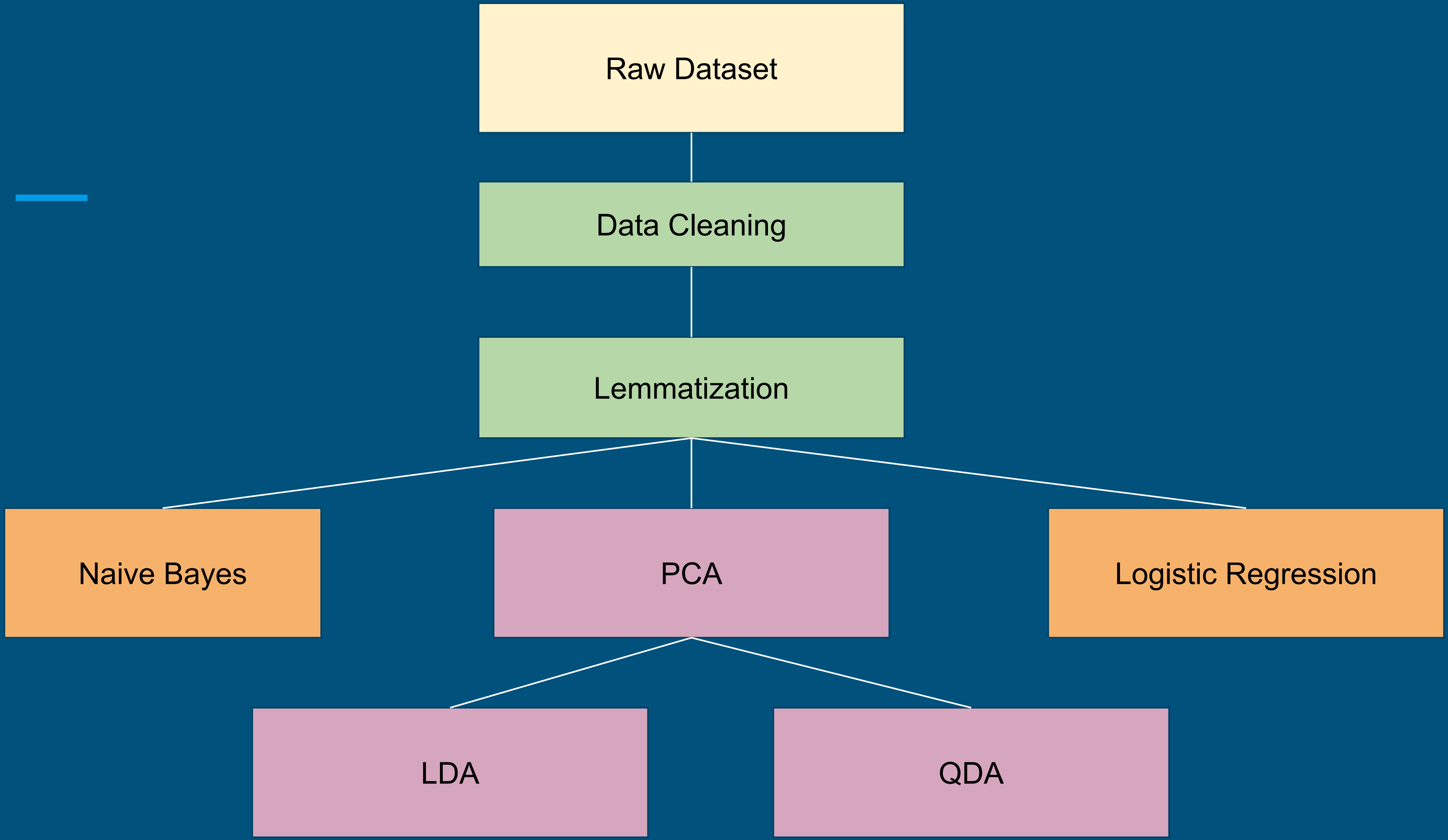
- Evaluate all models based on **performance metrics** like accuracy, precision, F1 score, and recall
- Use **cross-validation** on PCA to select the optimal number of principal components in order to perform advanced techniques like LDA and QDA.

Why are These Approaches Reasonable?

- These are traditional machine learning techniques for text classification tasks.
- Our models quickly convert text to its base form. Also, cross validation give us a robust evaluation.

There **are** challenges we faced from the approaches:

1) There are distributional assumptions in LDA and QDA that might not fit the text data well.
2) Our models may struggle with high dimensional features due to the thousands of features common in NLP tasks.
3) Due to high dimensionality we might also risk *overfitting*.



Raw Dataset

Data Cleaning

Lemmatization

Naive Bayes

PCA

Logistic Regression

LDA

QDA

Raw Data

- Before any work, the Yelp dataset contains reviews just as you may see them on the website

Example of review:

Skip this train wreck if you are looking for dinner. Our server was clueless...like we seriously thought she was lost in the restaurant. Had to ask for setups, water refills, etc. Ordered chicken parmesan and got eggplant parmesan instead with a fish bone in it. The drinks were good.

Data Cleaning

Overall Goal: Prepare the text data in reviews for usage in models.

1) Make reviews lowercase
2) Eliminate punctuation
3) Eliminate “stopwords”
   a) Words removed before text analysis as it doesn’t carry *meaning* in the review. (“a”, “is”, “in”, “that”, etc.)

```
skip train wreck looking dinner server clueless
like seriously thought lost restaurant ask
setups water refills etc ordered chicken
parmesan got eggplant parmesan instead fish
bone drinks good
```

Lemmatization

- Lemmatized all text data post-data cleaning
- Identifies each word’s intended meaning and reduces it down to its base form
  - I.E. Reduces the word “better” into the word “good”

```
skip train wreck looking dinner server clueless
like seriously thought lost restaurant ask
setup water refill etc ordered chicken parmesan
got eggplant parmesan instead fish bone drink
good
```

Data Analysis

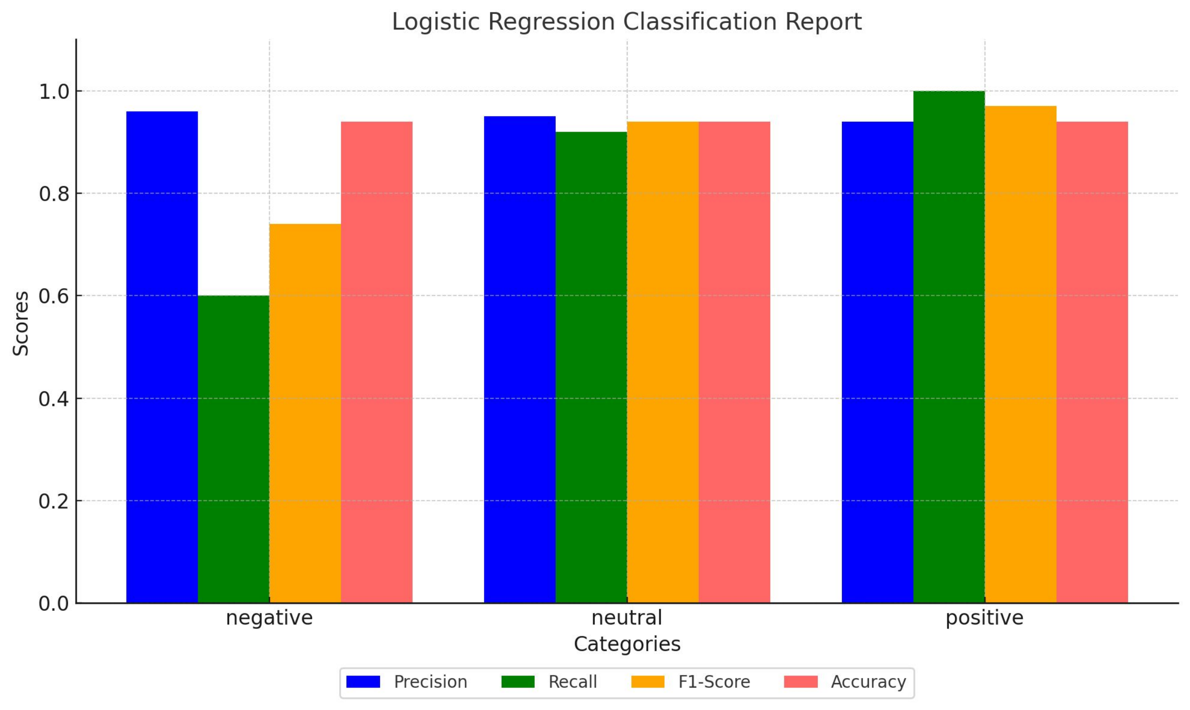
Total Reviews Analyzed:	40,109
Average Word Count:	97.7
Average Character Count:	528
Average Word Length:	4.46
Average Stopword Count:	45.75
Average Stopword Rate:	0.448
Average Stars:	3.804

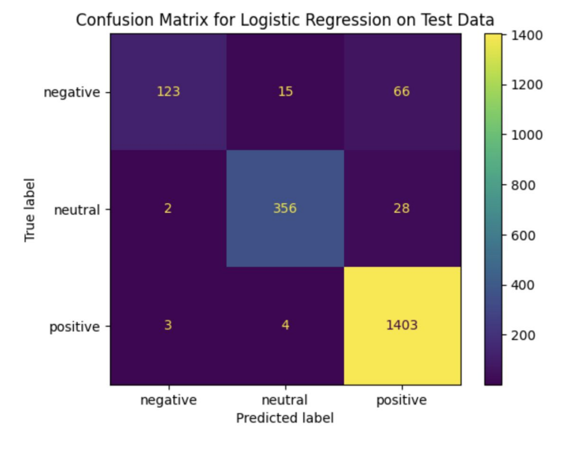
Results

As a reminder:

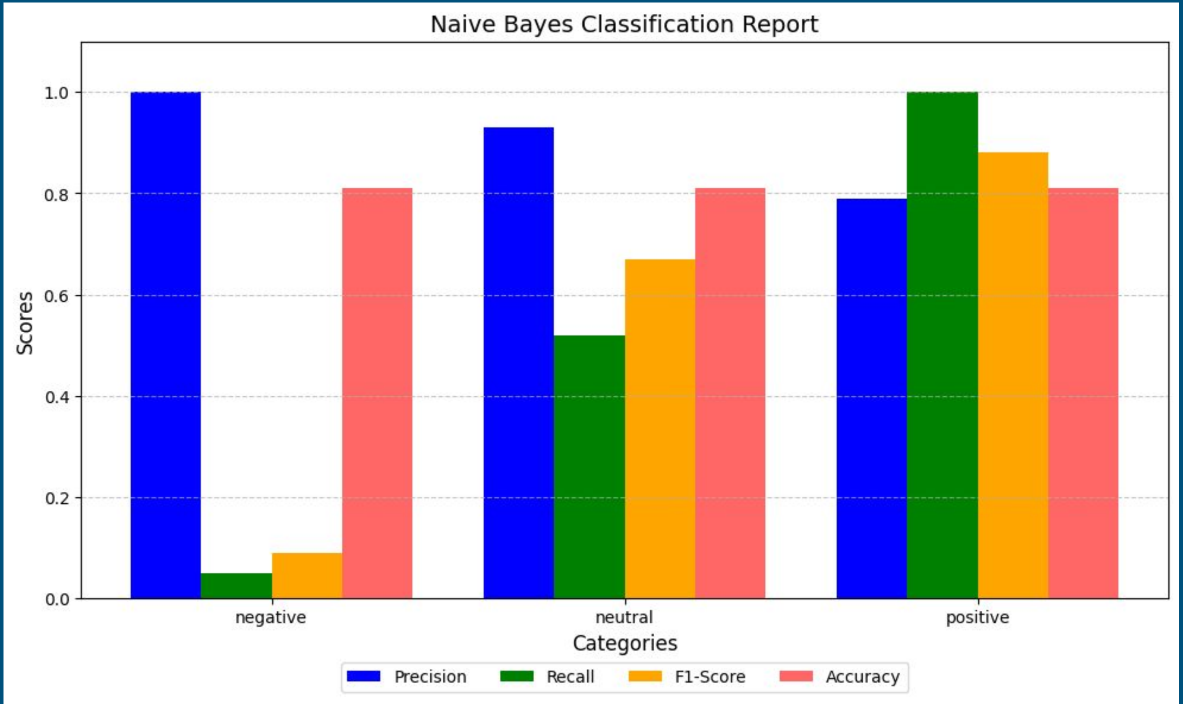
- We first run simple models like **logistic regression** and **naive bayes.**
- We then run PCA to reduce dimensionality in order to run more complex models like LDA and QDA
- We evaluate the efficacy of our models on the performance metrics of accuracy, precision, recall, and F1 score.

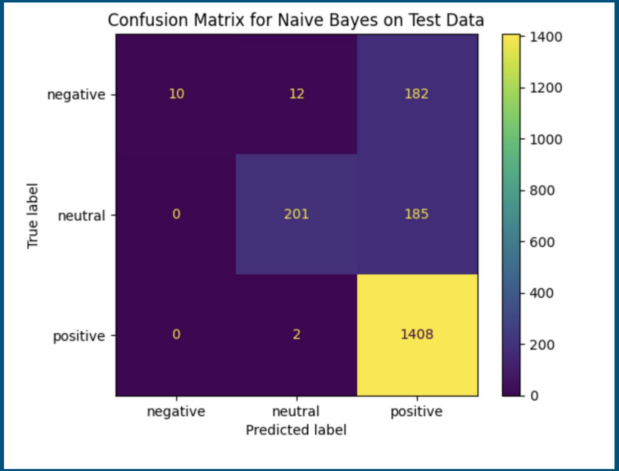
Logistic Regression





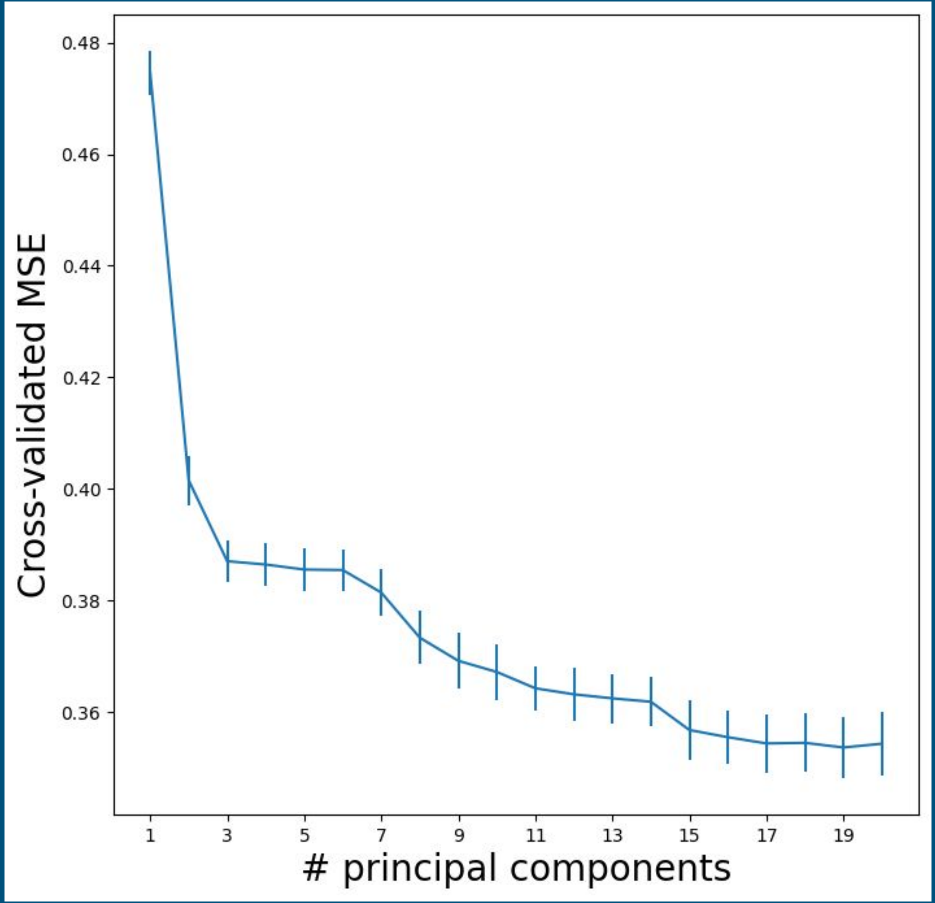
Naive Bayes


Naive Bayes Classification Report
Scores
1.0
0.8
0.6
0.4
0.2
0.0
negative
neutral
positive
Categories
Precision
Recall
F1-Score
Accuracy

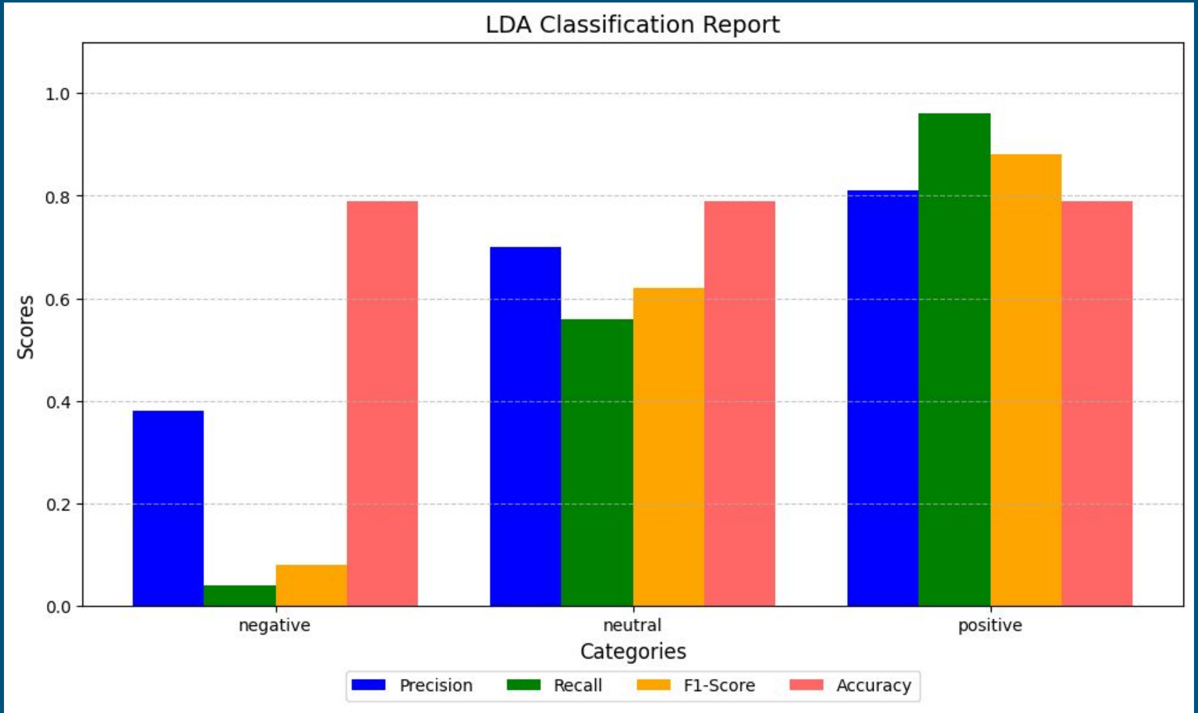

Confusion Matrix for Naive Bayes on Test Data
True label
negative
neutral
positive
10
12
182
0
201
185
0
2
1408
negative
neutral
positive
Predicted label
1400
1200
1000
800
600
400
200
0

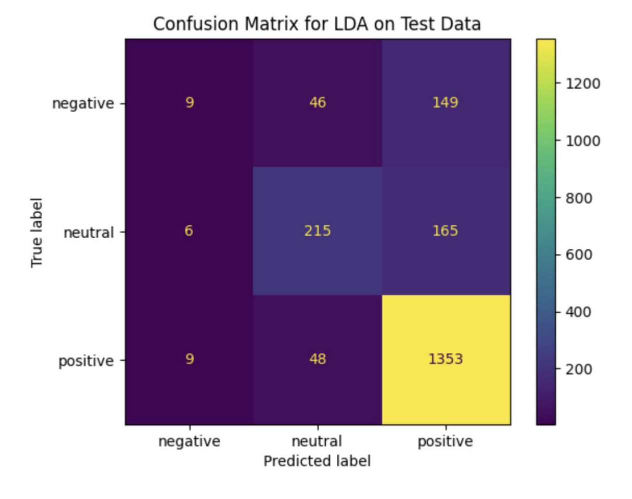
PCA

- We performed PCA before running more complex models in order to reduce the dimensionality of the data.
- **Initial PCA testing:**
  - Number of components: 20
- After using cross-validation to improve the model:
  - **Optimal number of components in the range of 1-21: 19**

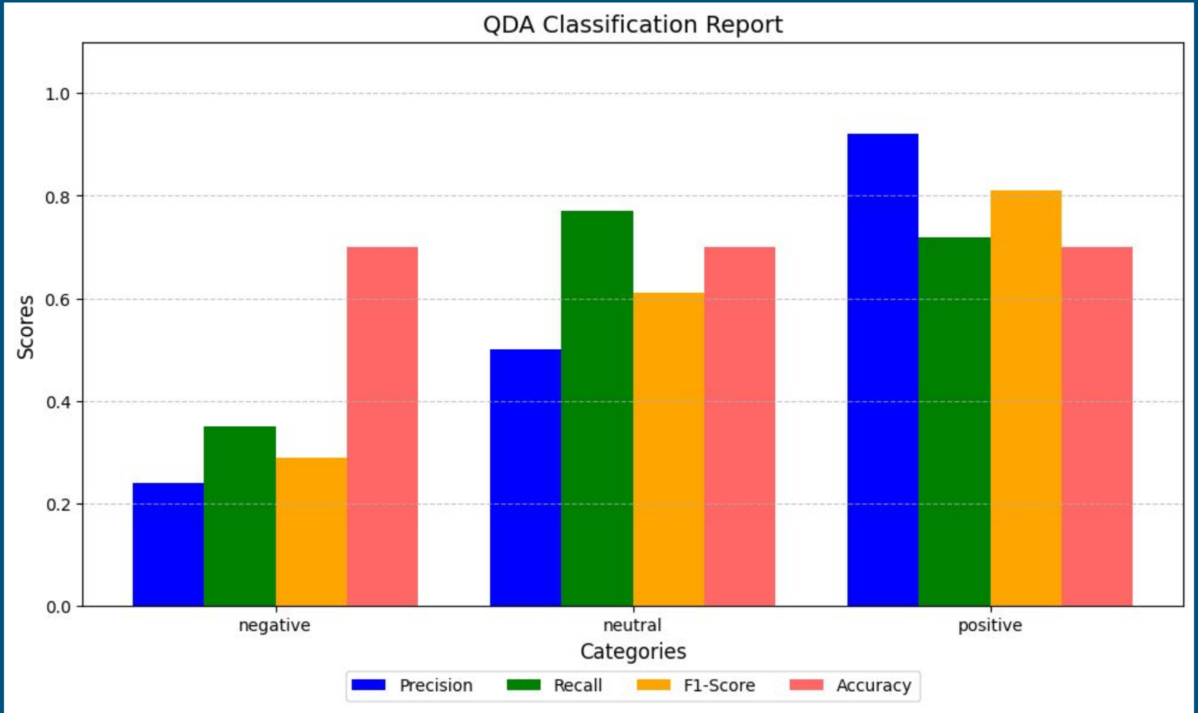


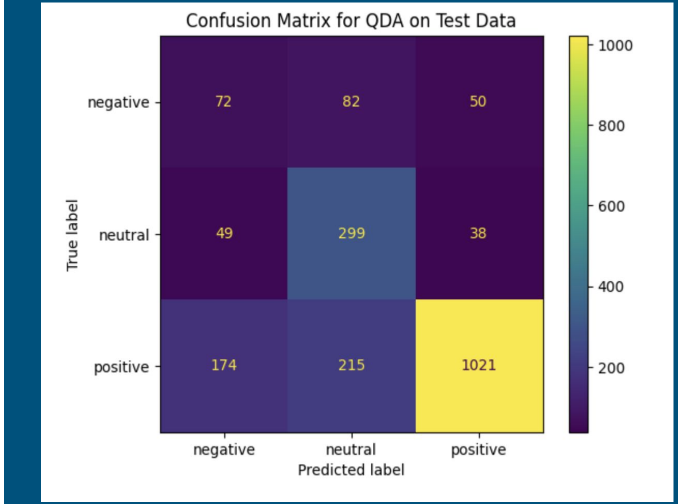
LDA


LDA Classification Report
Scores
1.0
0.8
0.6
0.4
0.2
0.0
negative
neutral
positive
Categories
Precision
Recall
F1-Score
Accuracy


Confusion Matrix for LDA on Test Data
True label
negative
neutral
positive
9
46
149
6
215
165
9
48
1353
negative
neutral
positive
Predicted label
1200
1000
800
600
400
200

QDA

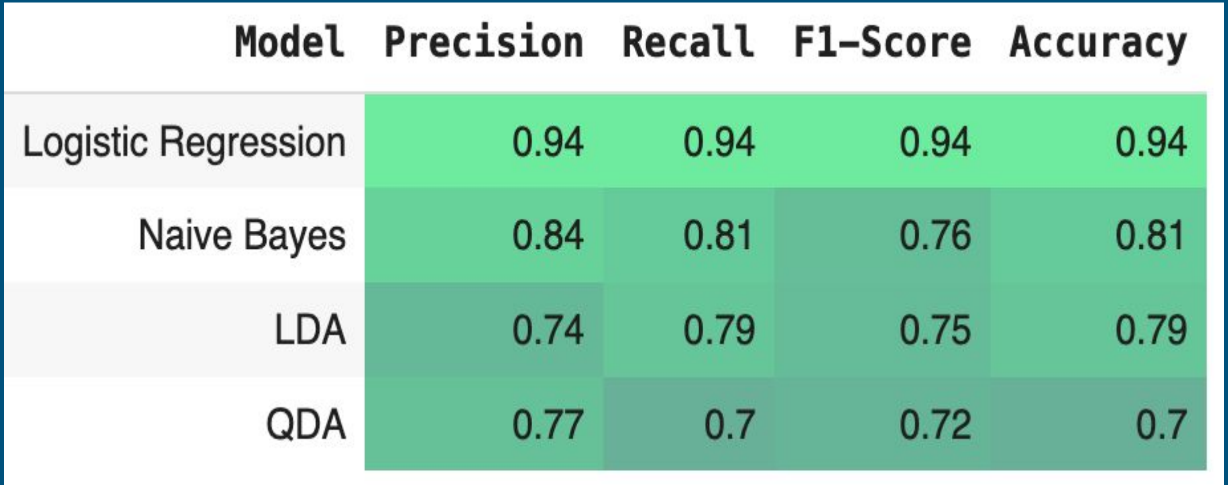



Confusion Matrix for QDA on Test Data

True label \ Predicted label	negative	neutral	positive
negative	72	82	50
neutral	49	299	38
positive	174	215	1021

Weighted Average Comparison Table

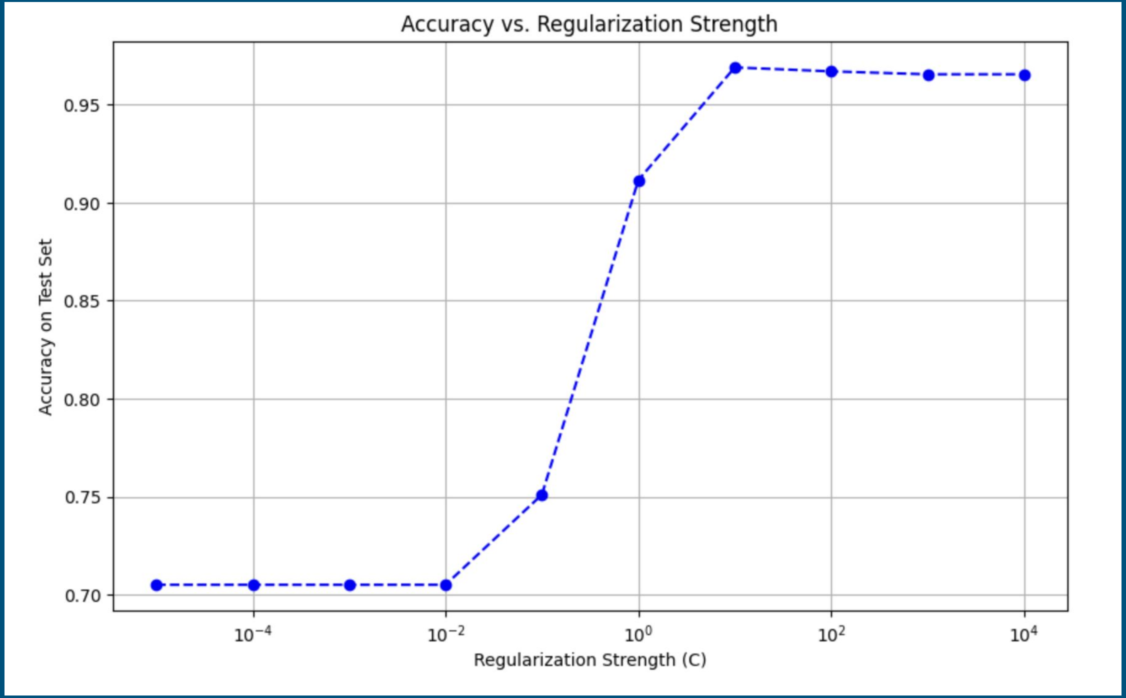
- Across the board, Logistic Regression performs best
- Naive Bayes performs second best
- Complex models are both worse than simpler ones



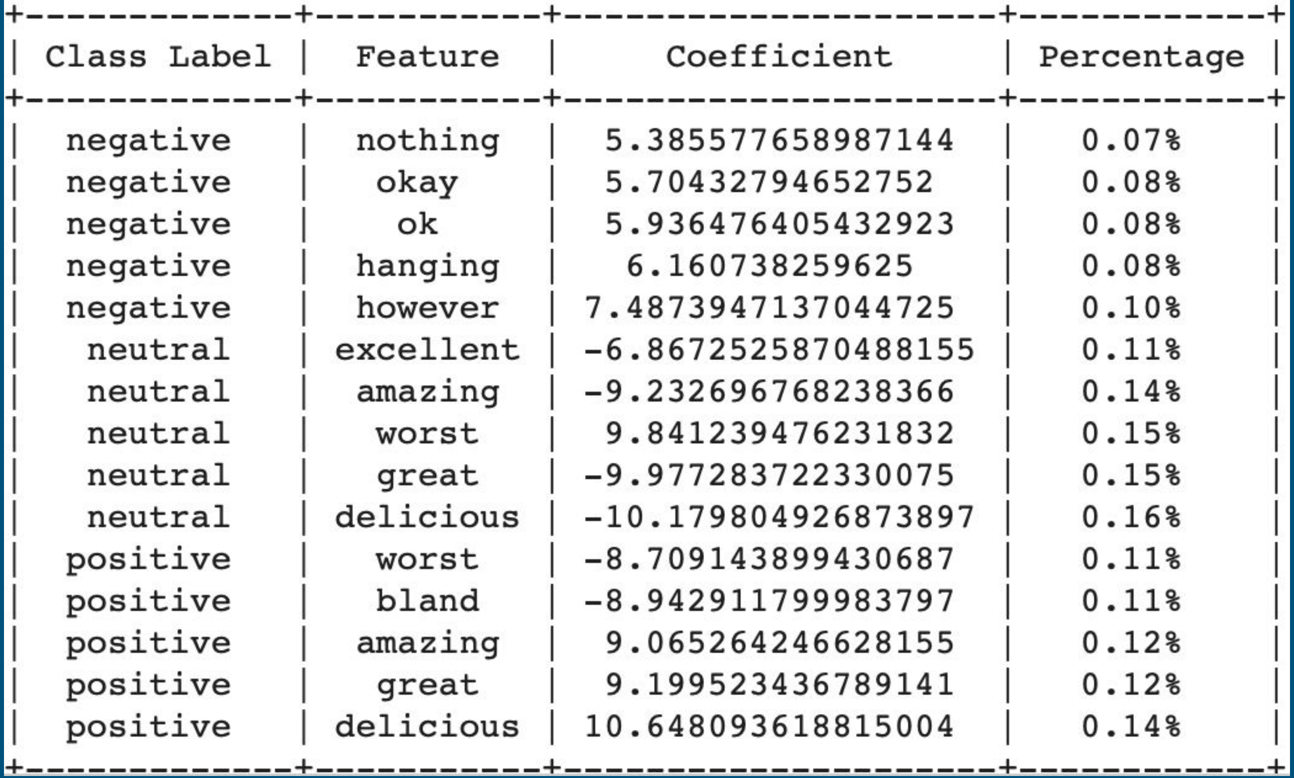
Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	0.94	0.94	0.94	0.94
Naive Bayes	0.84	0.81	0.76	0.81
LDA	0.74	0.79	0.75	0.79
QDA	0.77	0.7	0.72	0.7

Regularization Strength

- We tested regularization strength to see if we can improve the logistic regression model
- Regularization strength of 10 performs the best with 0.967 accuracy
- The model benefits from fitting relevant patterns in the data without too much constraint



Results: Most Predictive Features by Class



```
+--------------+-----------+---------------------+-------------+
| Class Label  |  Feature  |     Coefficient     | Percentage  |
+--------------+-----------+---------------------+-------------+
|   negative   |  nothing  |  5.385577658987144  |    0.07%    |
|   negative   |   okay    |  5.70432794652752   |    0.08%    |
|   negative   |    ok     |  5.936476405432923  |    0.08%    |
|   negative   |  hanging  |   6.160738259625    |    0.08%    |
|   negative   |  however  | 7.4873947137044725  |    0.10%    |
|    neutral   | excellent | -6.8672525870488155 |    0.11%    |
|    neutral   |  amazing  | -9.232696768238366  |    0.14%    |
|    neutral   |   worst   |  9.841239476231832  |    0.15%    |
|    neutral   |   great   | -9.977283722330075  |    0.15%    |
|    neutral   | delicious | -10.179804926873897 |    0.16%    |
|   positive   |   worst   | -8.709143899430687  |    0.11%    |
|   positive   |   bland   | -8.942911799983797  |    0.11%    |
|   positive   |  amazing  |  9.065264246628155  |    0.12%    |
|   positive   |   great   |  9.199523436789141  |    0.12%    |
|   positive   | delicious | 10.648093618815004  |    0.14%    |
+--------------+-----------+---------------------+-------------+
```

Conclusion

Logistic Regression beats the other more complex models

- Logistic regression had the best metrics *across the board*

- It is a simple, interpretable model that can efficiently handle high-dimensional text data through techniques like TF-IDF and regularization. Its ability to make linear predictions and manage imbalanced data with class weighting makes it effective for distinguishing between positive and negative reviews. This is shown with its performance in specificity and accuracy.

Limitations and Future Direction

Limitations:

- The models can be confused by reviews containing typos, preference bias, and temporal changes in slang, abbreviations, emojis, and non-standard grammar.
- Reviews tokenization and vectorization means analysis is done indiscriminate of sentence structure.
- Models trained on english reviews restaurants alone – narrow application.

Future Direction:

- Incorporation of user rating metrics of “funny” and “useful” to understand possible sarcasm or subjectivity.
- Reorganize dataset for equal distribution of sentiment so the models can have a better specificity