NLP With Amazon Food Reviews

Presented by Group 6：
Yuge (Kucina) Li, Chuqian Yin,
Chuyu Chen, Ziwen Ding

The University of Chicago
MSCA 31009
Machine Learning & Predictive Analytics

Agenda

1. Problem Statement & Assumption
2. Exploratory Data Analysis
3. Feature Engineering
4. Modeling Approaches & Results
5. Conclusion & Future Work

Problem Statement

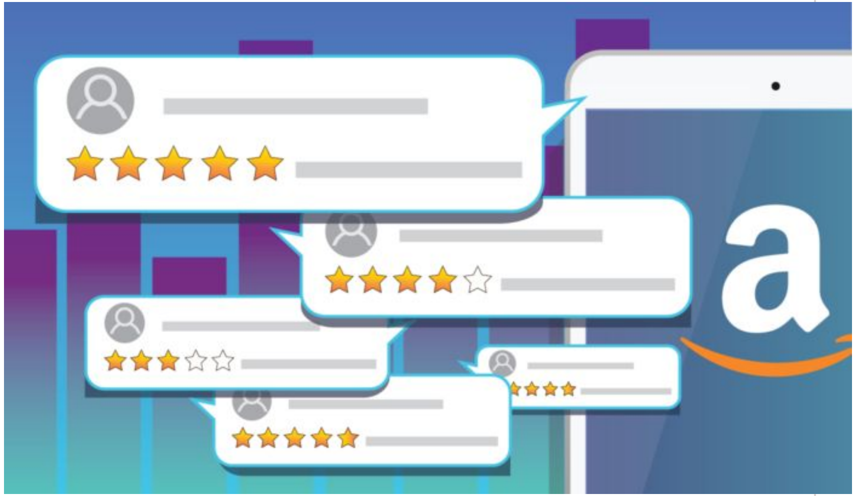
Problem Statement

While shopping online, customers widely look for **reviews** and **ratings**(1-5) for the products, which could help in understanding the reliability of the product before they initiate a purchase.

Questions:

- **Is it possible to determine the sentiment of the review?**
- What words tend to indicate positive and negative reviews?
- What kind of reviews tend to be more helpful?
- How is the word count related to sentiment of a review?

Assumption



Natural Language Processing

- Given a product review, using machine learning models to determine whether the sentiment of the review is **positive** or **negative**.
- Use **F1 Score**, which balances precision and recall, as validation metrics.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

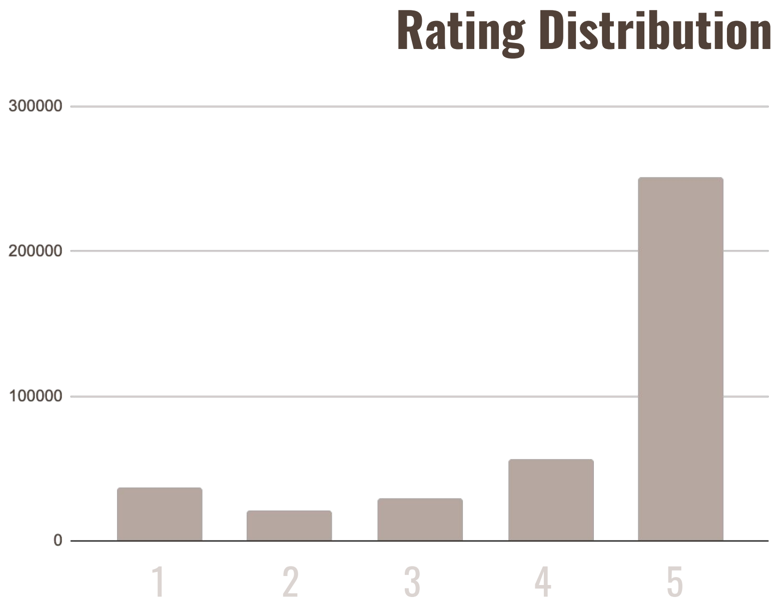
Exploratory Data Analysis

Data Overview

568,454
Amazon Food Reviews

74,258
Number of Products

256,059
Number of Users



Data Preparation

Idex	HelpfulnessNumerator	HelpfulnessDenominator	Text	Score	Label	Usefulness	Word count
318929	1	2	Um.... These are. $1-$1.89 per pouch in stores, or $7-8 per 6 pack. Where do you off selling a 6 pack for $24.99???	1	0	25-75%	23
553775	0	2	The operative thought is bitter. Nuance? There is a salty background. This was a one time purchase. Folger's is even better.	2	0	<25%	22
74408	0	0	Otherwise great. If you're a saltoholic like my husband you'll love these. I got these for the kids' lunchboxes but ended up not using them for that because of the salt.	4	1	useless	31

Exploratory Data Analysis

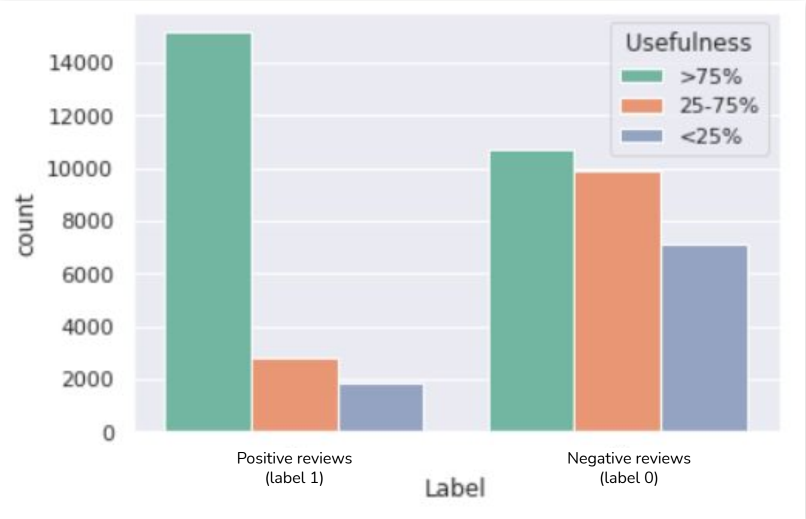
What words tend to indicate positive and negative reviews?

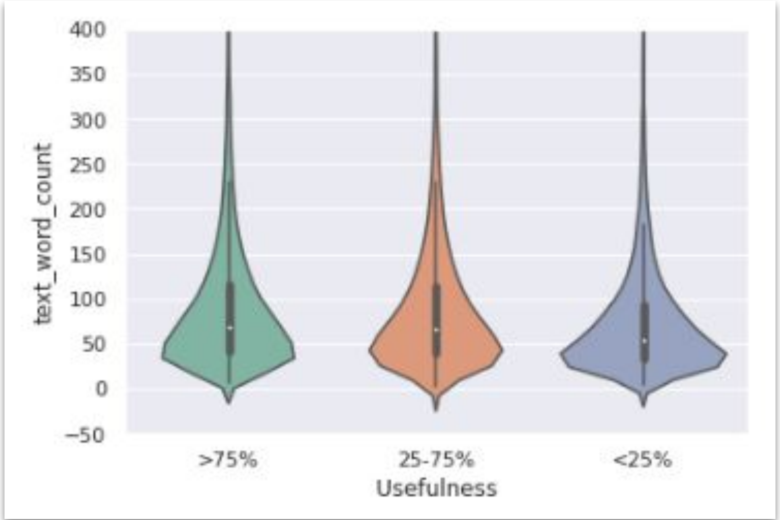
- Positive Review Word Cloud

- Negative Review Word Cloud

Exploratory Data Analysis

What kind of reviews tend to be more helpful?

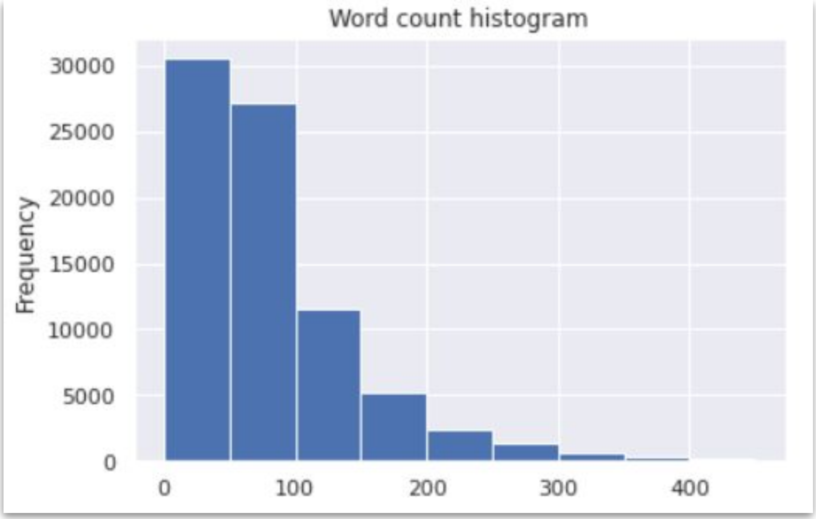


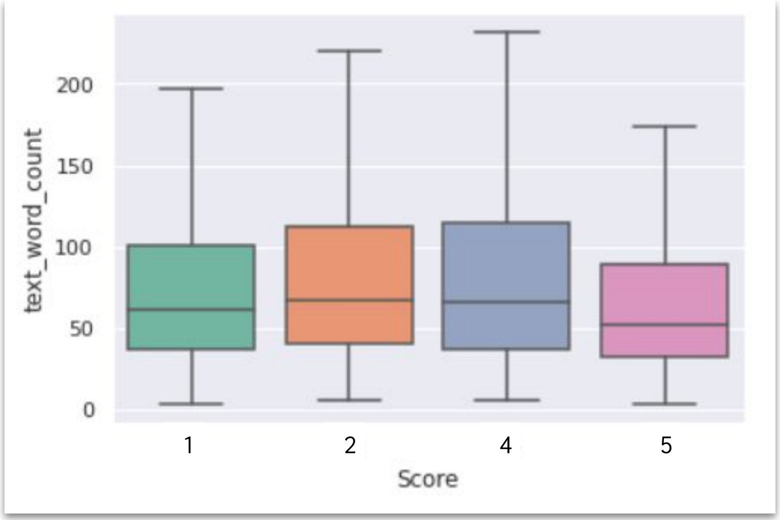


- Usefulness Distribution
- Word Count Distribution

Exploratory Data Analysis

How is the word count related to sentiment of a review?





- Word Count Histogram
- Word Count Distribution

Feature Engineering

Data Cleaning & Feature Engineering

Data Cleaning

- Removed duplicate rows
- Randomly selected 2,500 rows from each rating, except rows with [‘**`Score`**’] = 3 (neutral)
- Determined the target [‘**`Label`**’] and feature [‘**`Text`**’]
- Text Preprocessing
  - Converted all words to lowercase
  - Removed HTML tags and punctuations
  - Removed stopwords (but not changing the sentiment) and stemming
- Data shape (568454,10) → (10000,2)

Feature Engineering

- Average **`Word2Vec`** (learns all the internal relationships between words and return the words in dense vector form)
- Vectorize text corpus into a list of integers using **`Tokenizer`** from keras, and **`pad_sequences`** to the same length

Additional Encoding Methods

- TF_IDF
- Bag of Words
- TFIDF_W2VEC
- FAST_TEXT
- GloVe

Modeling Process

Proposed Approaches

- **Ensemble Models:**
    - Random Forest
    - Gradient Boosting
    - Xgboost

- **Neural Network Models:**
    - Convolutional Neural Network
    - Recurrent Neural Network

Random Forest With Grid Search

Result for Best Model from Grid Search - Overfitting Occurs

	Train Data Prediction	Test Data Prediction
F1 Score	0.72	0.69
Weighted Avg F1	0.69	0.65

Reduce Model Complexity: Decrease max_depth value/Decrease max_features/Increase n_estimators

Result for Model After Hyperparameter tuning - Overfitting mitigates

	Train Data Prediction	Test Data Prediction
F1 Score	0.66	0.65
Weighted Avg F1	0.62	0.60

Gradient Boosting With Grid Search

Result for Best Model from Grid Search - Overfitting Occurs

	Train Data Prediction	Test Data Prediction
F1 Score	0.86	0.74
Weighted Avg F1	0.86	0.74

Reduce Model Complexity: Decrease learning rate/Decrease max_depth/Increase n_estimators

Result for Model After Hyperparameter tuning - Overfitting mitigates

	Train Data Prediction	Test Data Prediction
F1 Score	0.73	0.71
Weighted Avg F1	0.73	0.70

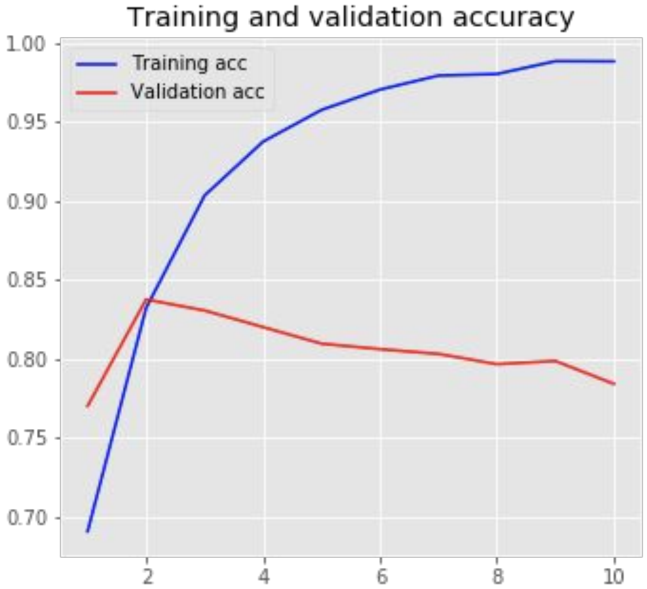
Recurrent Neural Network

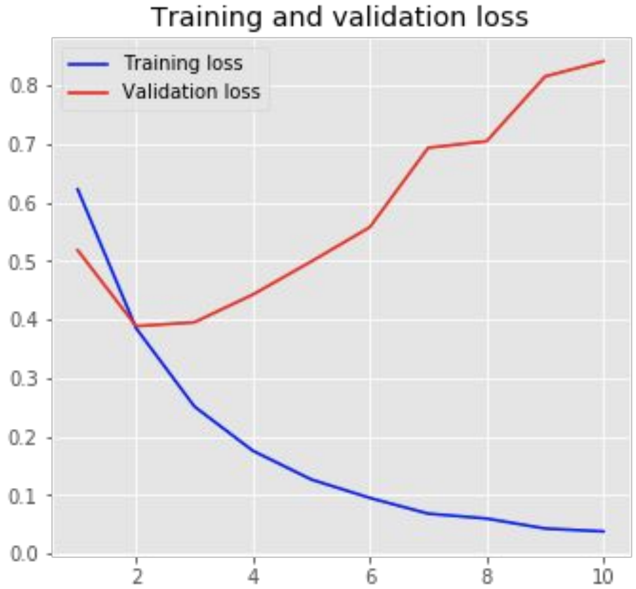
Suitable Model for **Sequence Classification**:

- RNN is basically a sequence of neural network blocks that are linked to each other like a chain
- it exhibit temporal behavior and capture sequential data which makes it a more ‘natural’ approach when dealing with text data.

Base RNN Model with LSTM

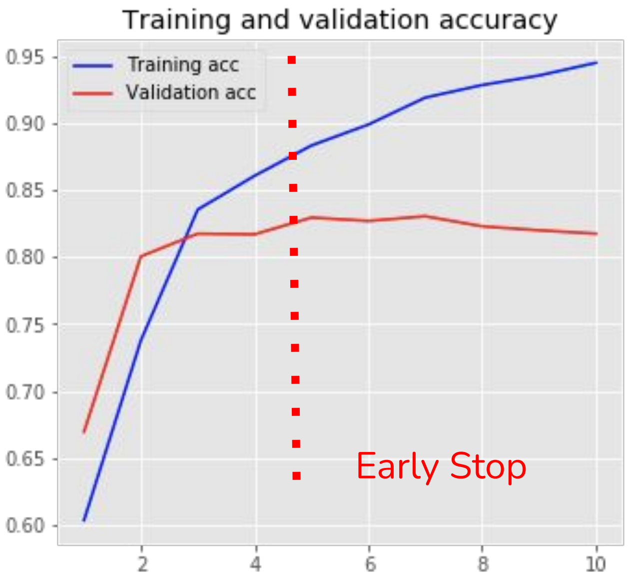
Embedding Layer → LSTM(64) Layer → Dense Layer

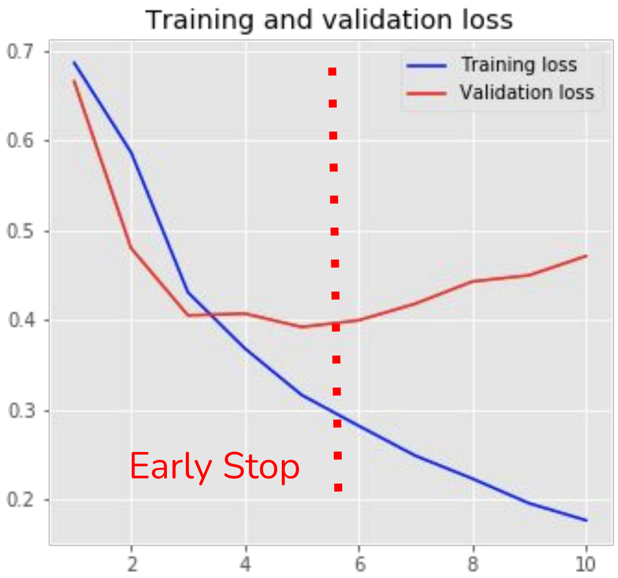




RNN Model with LSTM After Regularization

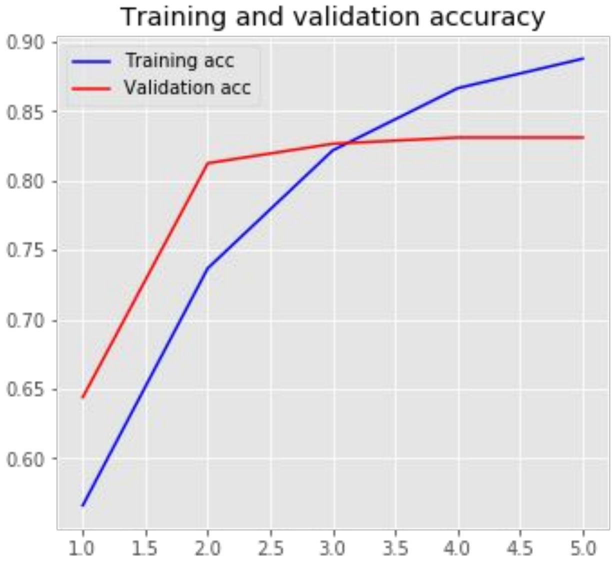
Embedding → SpatialDropout1D(0.5) → LSTM(16) → Dropout(0.5) → Dense

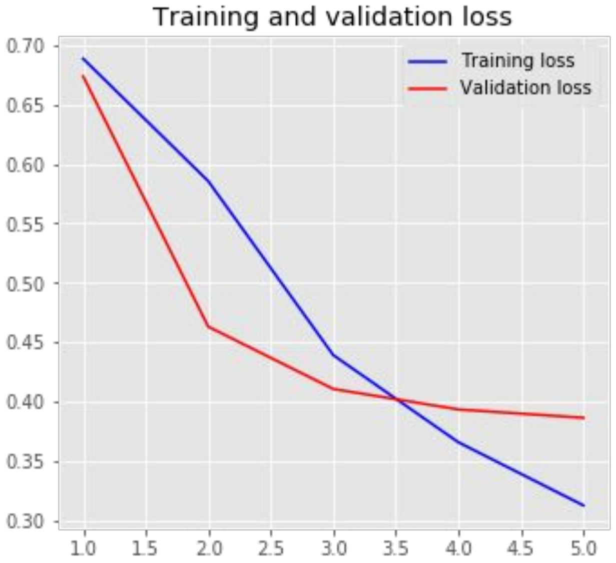




RNN Model with LSTM After Regularization

Embedding → SpatialDropout1D(0.5) → LSTM(16) → Dropout(0.5) → Dense

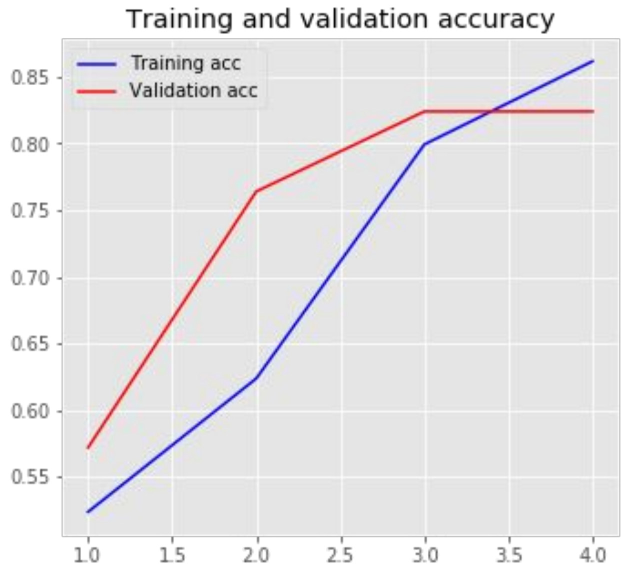


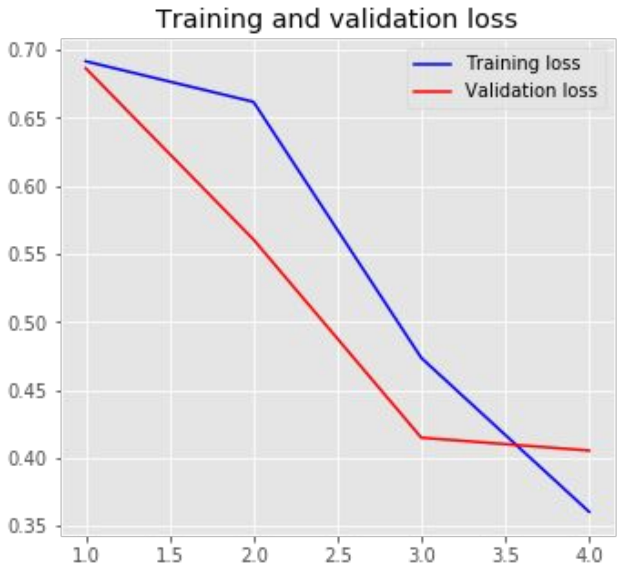


Test F1: 0.83

RNN Model with RNN

Embedding → SpatialDropout1D(0.5) → SimpleRNN (32) → Dropout(0.5) → Dense





Test F1: 0.82

Xgboost

Advantages to solve **text classification** task:

- Good to process structured or tabular data
- Have inbuilt CV and Regularization functions
- Handle missing values well
- Relatively Flexible
- Easily to save and load data

Xgboost With Grid Search & Mutually Tuning

Result for Model from Grid Search - Overfitting Occurs

	Train Data Prediction	Test Data Prediction
F1 Score	0.92	0.75
Weighted Avg F1	0.92	0.75

Change embedding method from Keras default tokenizer to word2vec; increase Gamma value and add another regularization L1 norm: alpha = 0.1; decrease max_depth value and n_estimators

Result for Best Model After Hyperparameter Tuning - Overfitting Mitigates

	Train Data Prediction	Test Data Prediction
F1 Score	0.75	0.72
Weighted Avg F1	0.75	0.72

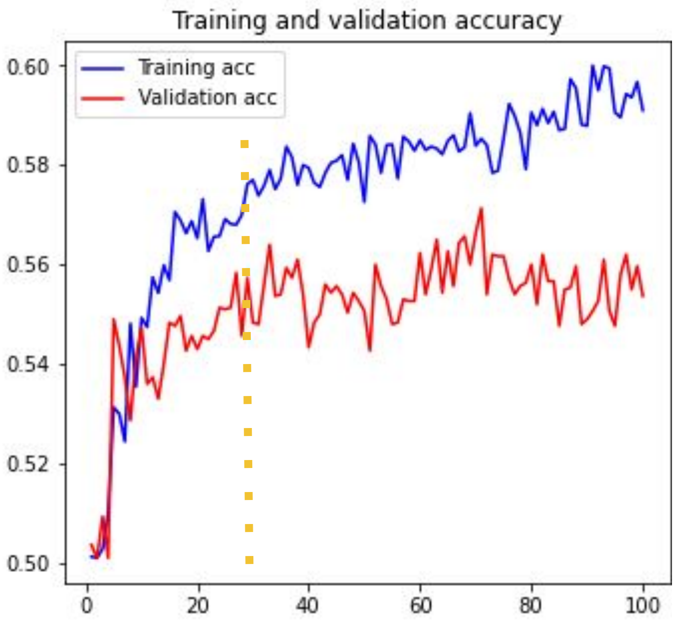
Convolutional Neural Network

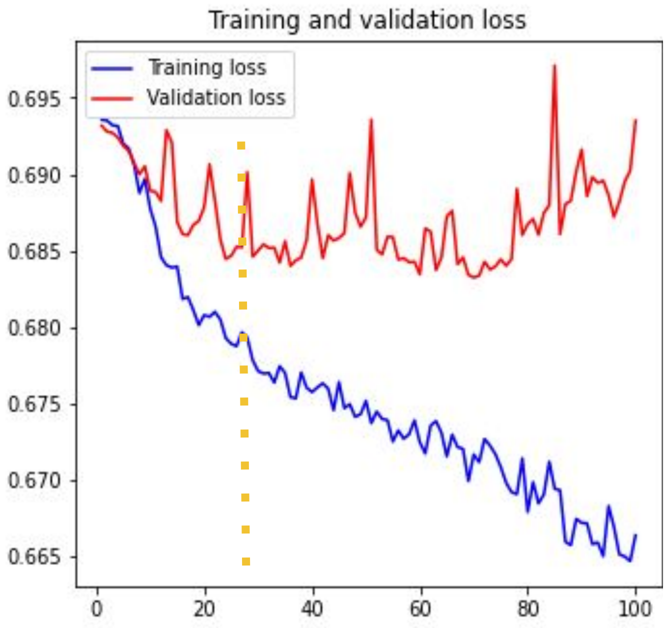
Advantages to solve **text classification** task:

- Extracts sub-structures across matrix space while detecting indicative local and position-invariant patterns with filter/kernel
- Uses feature extractors, convolution and pooling, to reduce dimensional complexity but keep significant information
- Performs very fast on GPUs, and represent large size n-grams in a compact way
- Applies Conv-ND layer to process n-dimensional array representing the sequential text data
- Utilizes dense and drop-out layers to prevent overfitting problem

Base CNN Model

Tokenizer Embedding Layer → Conv1d with Sigmoid → Max Pooling → Dropout → Flatten → Dense





CNN Models Exploration

Result for model trained by data embedded with Kera default tokenizer- performs badly with overfitting problem

	Train Data Prediction	Test Data Prediction
F1 Score	0.59	0.55
Weighted Avg F1	0.59	0.56

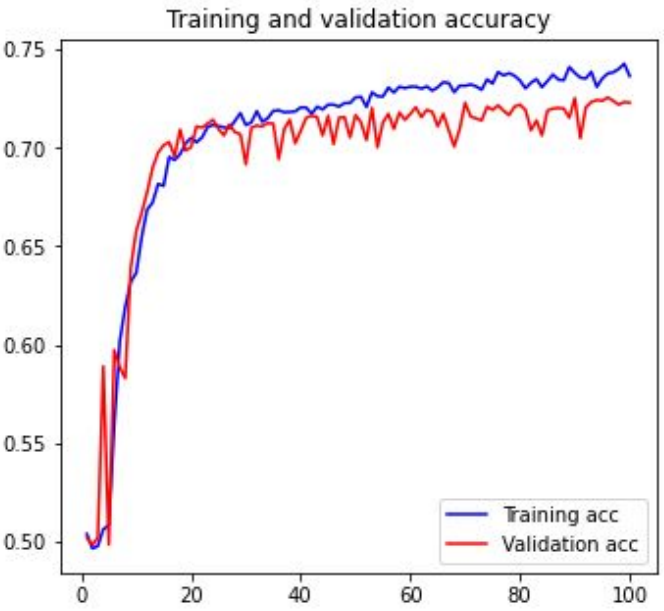
Change embedding method from kera default tokenizer to word2vec; add more Conv1D layers; take off flatten and drop layer; add more Dense layers

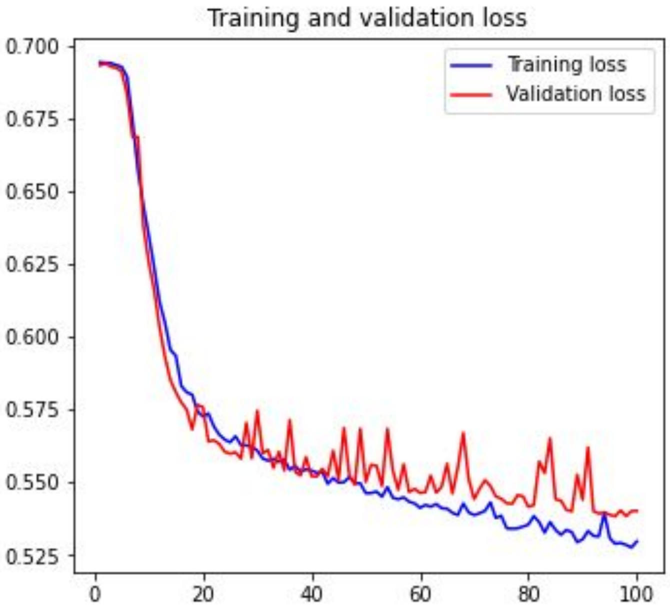
Result for best model after replacing different layers and hyperparameters tuning- performs well with smaller overfitting

	Train Data Prediction	Test Data Prediction
F1 Score	0.74	0.72
Weighted Avg F1	0.74	0.72

Best CNN Model

Word2Vec Embedding Layer → Conv1d with Sigmoid (3) → Global Max Pooling → Dense with Sigmoid (2)





CNN Models Exploration

Result for model trained by data embedded with Kera default tokenizer- performs badly with overfitting problem

	Train Data Prediction	Test Data Prediction
F1 Score	0.59	0.55
Weighted Avg F1	0.59	0.56

Change embedding method from kera default tokenizer to word2vec; add more Conv1D layers; take off flatten and drop layer; add more Dense layers

Result for best model after replacing different layers and hyperparameters tuning- performs well with smaller overfitting

	Train Data Prediction	Test Data Prediction
F1 Score	0.74	0.72
Weighted Avg F1	0.74	0.72

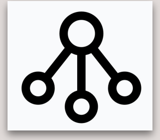
Conclusion
Feature Work

Best Model Performance for Each Category

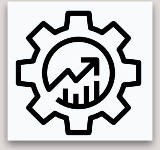
Model/Data	Testing Set Result		Train Set Result	
	F1 Score	Weighted avg F1	F1 Score	Weighted avg F1
Xgboost	0.72	0.72	0.75	0.75
CNN	0.72	0.72	0.74	0.74
Random Forest	0.65	0.60	0.66	0.62
LSTM	0.83	0.83	0.93	0.93
Simple RNN	0.82	0.82	0.91	0.91

Future Work

Apply multiple channels paradigm in text processing



Keep discovering the feasibility of multi-label classification



Evaluate the model in real business cases

THANKS!

Do you have any questions?

Appendix

General Model Exploration

- Applied many embedding methods, such as BOW, TF_IDF, AVG_W2VEC, TFIDF_W2VEC, FAST_TEXT and etc.
- Tried different ways to tune hyperparameters, except for Grid-Search or Randomized-Search methods
- Compare results of models trained by multi-class score data and binary label data

Xgboost Primitive Model Classification Report

Training	Precision	Recall	F-1 Score	Support
0	0.92	0.91	0.92	3503
1	0.91	0.92	0.92	3497
Accuracy			0.92	7000
Macro Avg	0.92	0.92	0.92	7000
Weighted Avg	0.92	0.92	0.92	7000

Validation	Precision	Recall	F-1 Score	Suppor t
0	0.76	0.75	0.75	1511
1	0.75	0.75	0.75	1489
Accuracy			0.75	3000
Macro Avg	0.75	0.75	0.75	3000
Weighted Avg	0.75	0.75	0.75	3000

Xgboost Best Model Classification Report

Training	Precision	Recall	F-1 Score	Support
0	0.74	0.75	0.75	3467
1	0.75	0.74	0.75	3533
Accuracy			0.75	7000
Macro Avg	0.75	0.75	0.75	7000
Weighted Avg	0.75	0.75	0.75	7000

Validation	Precision	Recall	F-1 Score	Suppor t
0	0.72	0.70	0.71	1494
1	0.71	0.73	0.72	1506
Accuracy			0.72	3000
Macro Avg	0.72	0.72	0.72	3000
Weighted Avg	0.72	0.72	0.72	3000

CNN Best Model Classification Report

Training	Precision	Recall	F-1 Score
Accuracy			0.59
Macro Avg	0.59	0.59	0.59
Weighted Avg	0.60	0.59	0.59

Validation	Precision	Recall	F-1 Score
Accuracy			0.55
Macro Avg	0.55	0.56	0.55
Weighted Avg	0.57	0.55	0.56

Base Primitive Model Classification Report

Training	Precision	Recall	F-1 Score
0	0.81	0.71	0.76
1	0.68	0.78	0.72
Accuracy			0.74
Macro Avg	0.74	0.75	0.74
Weighted Avg	0.75	0.74	0.74

Validation	Precision	Recall	F-1 Score
0	0.79	0.69	0.74
1	0.65	0.76	0.70
Accuracy			0.72
Macro Avg	0.72	0.73	0.72
Weighted Avg	0.73	0.72	0.72

Future Work

- Multi-class model
- More channels of texts data in model training
- Additional classifications of helpfulness levels given a product comment
- Identify the tastes of customers and create recommendation system to promote our products precisely
- Develop a recommendation system to promote similar food products based on the score that a customer gave to a certain product