Project 3: Web APIs and Classification

r/TalesFromTheCustomer and
r/TalesFromRetail

Wee Zi Jian

Table of Contents

Problem Statement

- Reddit threads **r/TalesFromTheCustomer** and **r/TalesFromRetail** describe contrasting retail experiences - one from customers' perspectives and one from retail staff

- These subreddits are **rich in text data** and thus suitable for Natural Language Processing (NLP) modelling using a bag of words approach

Problem Statement

- This project will build a **classification model** to determine if a post is from r/TalesFromTheCustomer or r/TalesFromRetail

- An **interpretable NLP model** could be useful to retail executives who wish to understand more about the experiences of their customers and frontline staff

Methodology

1. Web Scraping

Scrape subreddit APIs to collect text data

2. Pre-Processing

Combine title and post data. Process combined text data

3. Exploratory Data Analysis

Identity most frequent and unique words of each subreddit

4. Screening

Iterate through different shorteners, vectorizers and classification models

5. Grid Search

Grid search parameters to select best performing model

Pre-Processing

927

Rows of data from r/TalesFromTheCustomer

320

Rows of data from r/TalesFromRetail

3

Relevant columns: ‘subreddit’, ‘selftext’, ‘title’

Pre-Processing

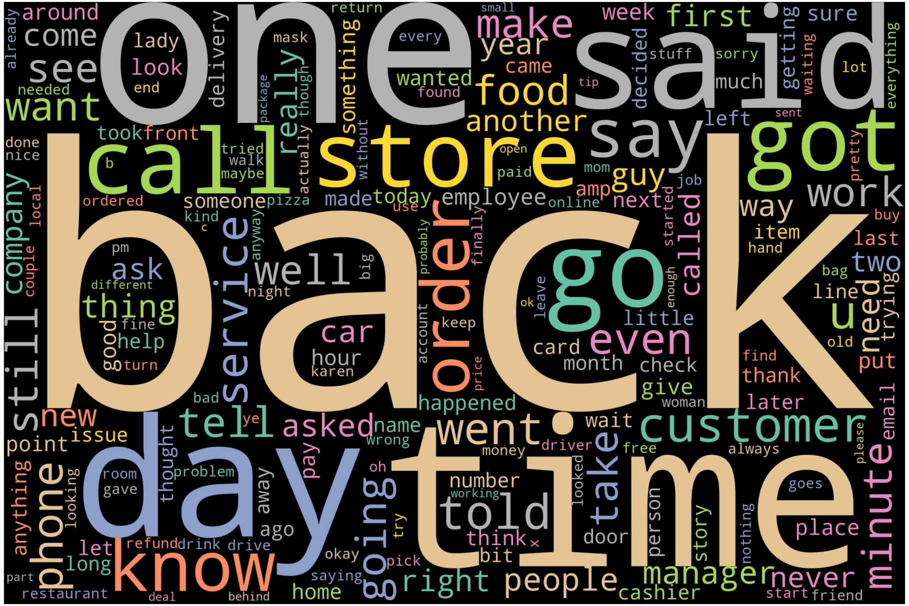
- Create **target** column
- Combine ‘title’ and ‘selftext’ columns into **‘text’ column**
- Create new columns for **lemmatized, stemmed and unshortened text**
- Train test split

	subreddit	selftext	title	target	text	text_lem	text_stem	text_raw
0	TalesFromTheCustomer	Please, for the love of all that is holy, read...	A REMINDER of the one rule that everyone overl...	0	A REMINDER of the one rule that everyone overl...	reminder one rule everyone overlook please lov...	remind one rule everyon overlook pleas love ho...	reminder one rule everyone overlooks please lo...
1	TalesFromTheCustomer	A few weeks ago I went to the large chain phar...	Always check your prescriptions	0	Always check your prescriptions A few weeks ag...	always check prescription week ago went large ...	alway check prescript week ago went larg chain...	always check prescriptions weeks ago went larg...
2	TalesFromTheCustomer	I stopped at a local shoe store last week to f...	She broke the biggest rule...	0	She broke the biggest rule... I stopped at a l...	broke biggest rule stopped local shoe store la...	broke biggest rule stop local shoe store last ...	broke biggest rule stopped local shoe store la...
3	TalesFromTheCustomer	Yesterday I went to get my hair cut. Nothing c...	When they do without checkinf	0	When they do without checkinf Yesterday I went...	without checkinf yesterday went get hair cut n...	without checkinf yesterday went get hair cut n...	without checkinf yesterday went get hair cut n...
4	TalesFromTheCustomer	I was having an awful morning, I'd woke up wit...	This is my private property if I tell you to l...	0	This is my private property if I tell you to l...	private property tell leave aka doorknocker sk...	privat properti tell leav aka doorknock skill ...	private property tell leave aka doorknockers s...

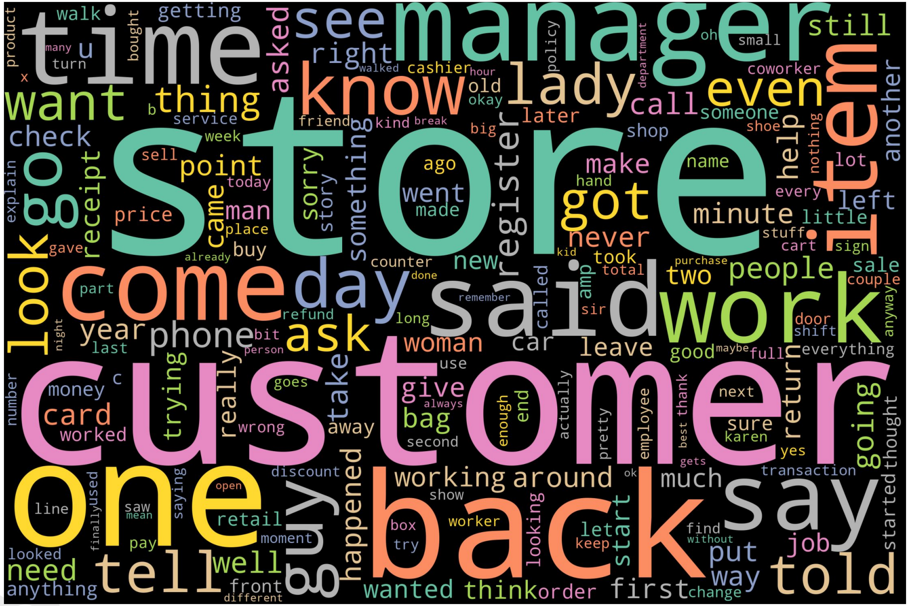
Exploratory Data Analysis

Most Frequent Words

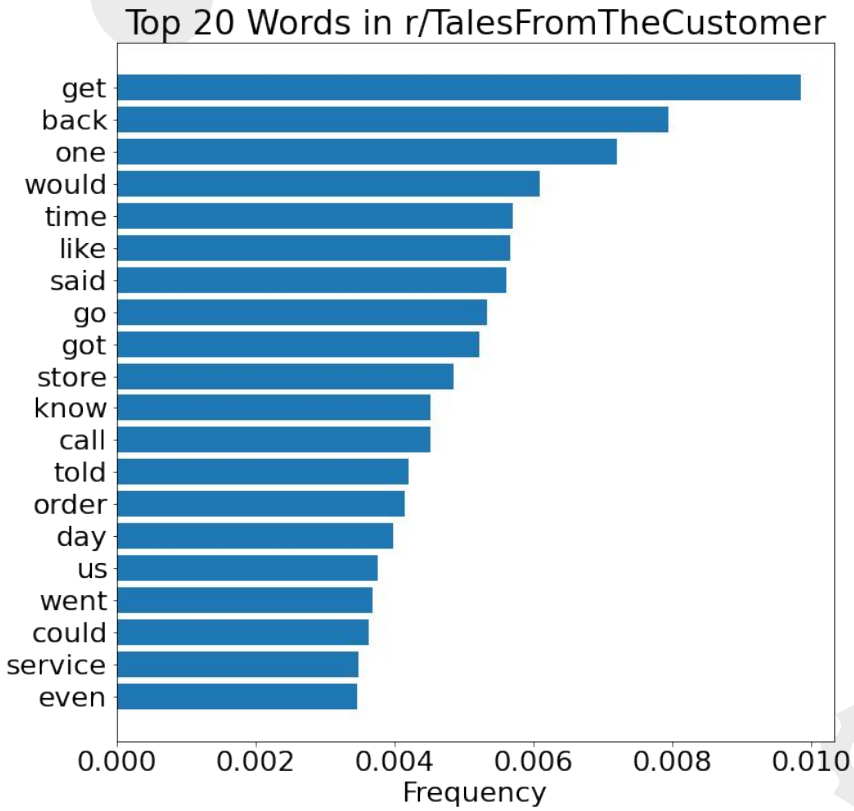
r/TalesFromTheCustomer

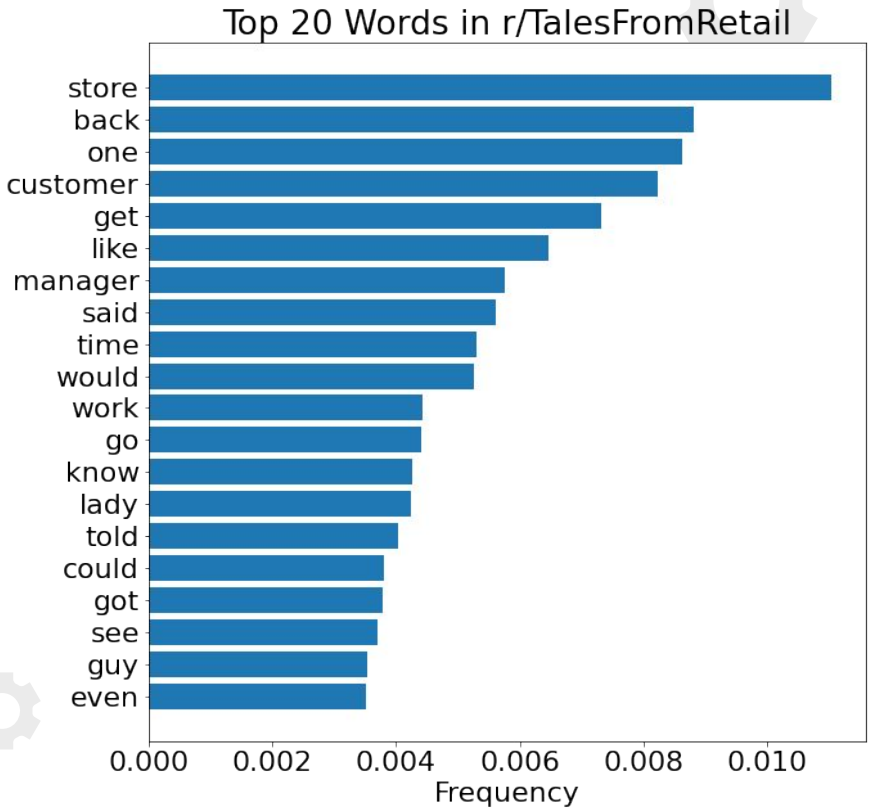


r/TalesFromRetail

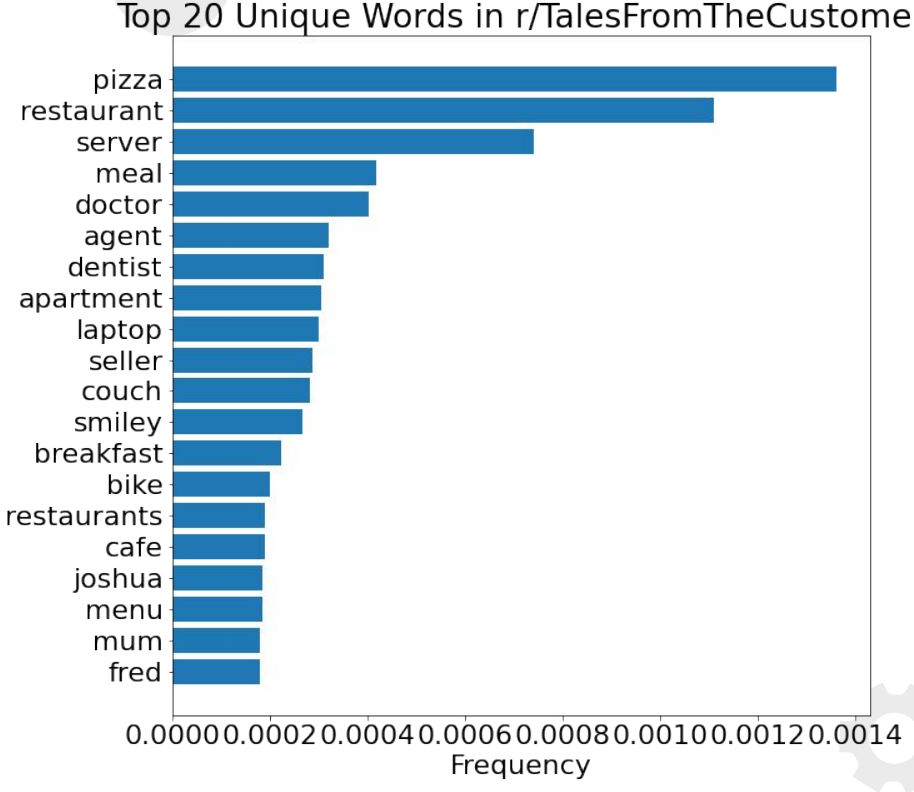


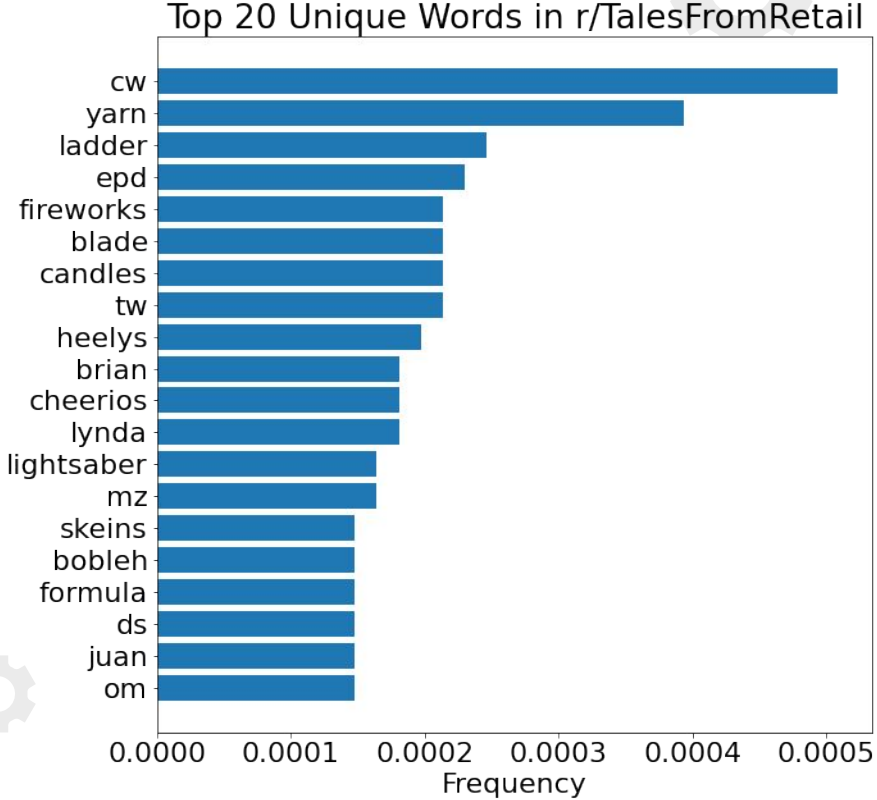
Most Frequent Words





Most Frequent Unique Words





Word Frequency

- Many words are **frequent to both subreddits**, such as 'back', 'store', 'get', 'time' and 'one'
- Given their high frequency in both datasets, these words **may not be useful** for the classification models to distinguish between the 2 subreddits
- For r/TalesFromTheCustomer, the most frequent **unique words** are 'pizza', 'restaurant' and 'server'. Other unique words include **descriptions of service providers** such as 'doctor', 'dentist', 'agent' and 'seller'
- For r/TalesFromRetail, other than **'cw' (coworker)**, the top unique words generally have **very low frequencies** within the entire subreddit corpus. Thus, only 'cw' could be useful to the NLP classification models as it is unlikely for new posts to contain the other words

Modelling

Screening

3 shorteners (lemmatize, stem, unshortened)

2 vectorizers (Count Vectorizer, TF-IDF Vectorizer)

6 models (Logistic Regression, K-Nearest Neighbours, Multinomial Naive Bayes, Random Forest, Extra Trees, Support Vector Machine)

= 36 models

Screening

Top 10 models by test F1 Score:

shortener	vectorizer	model	cv accuracy	cv f1 score	cv roc auc	cv precision	cv recall	test accuracy	test f1 score	test roc auc	test precision	test recall
stem	cvec	log_reg	0.868076	0.716451	0.926153	0.792473	0.667663	0.868932	0.712766	0.791528	0.817073	0.632075
stem	cvec	nb	0.835712	0.685965	0.889784	0.671954	0.704983	0.830097	0.692982	0.802380	0.647541	0.745283
raw	cvec	log_reg	0.859664	0.686896	0.928853	0.787263	0.616168	0.859223	0.691489	0.778826	0.792683	0.613208
lem	cvec	nb	0.836931	0.684520	0.885759	0.680354	0.691251	0.837379	0.691244	0.794950	0.675676	0.707547
raw	cvec	nb	0.847753	0.703241	0.894201	0.712354	0.700664	0.839806	0.679612	0.781169	0.700000	0.660377
lem	cvec	log_reg	0.865645	0.706400	0.925425	0.787567	0.653378	0.851942	0.673797	0.767758	0.777778	0.594340
lem	cvec	svm	0.814148	0.480700	0.917439	0.848663	0.340864	0.837379	0.562092	0.696294	0.914894	0.405660
lem	tvec	knn	0.773350	0.450794	0.797395	0.588519	0.368549	0.798544	0.560847	0.700980	0.638554	0.500000
stem	cvec	svm	0.824940	0.519115	0.920934	0.879326	0.373643	0.832524	0.554839	0.693026	0.877551	0.405660
stem	tvec	knn	0.773364	0.477517	0.804222	0.581337	0.406202	0.781553	0.536082	0.686459	0.590909	0.490566

Screening

- F1 score selected as performance metric due to unbalanced classes
  - Baseline accuracy = 927 / (927 + 320) = 74.3%

- Marginal model performance improvement using lemmatizing/stemming over unshortened text

- Top models are:
  - Logistic Regression (Count Vectorizer)
  - Multinomial Naive Bayes (Count Vectorizer)
  - Support Vector Machine (Count Vectorizer)
  - K-Nearest Neighbours (TF-IDF Vectorizer)

Grid Search

- Grid search (scoring = F1 score) performed to tune parameters of top model/vectorizer combinations

- Choose lemmatized text for simplicity

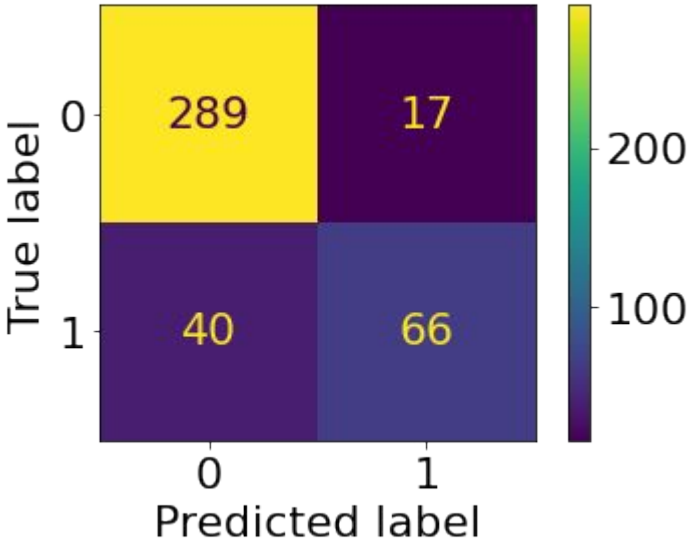
shortener	model	cv best score	best estimator	test accuracy	test f1	test roc	test precision	test recall
lem	log_reg	0.713086	(CountVectorizer(max_df=0.8, min_df=3, stop_wo...	0.861650	0.698413	0.783543	0.795181	0.622642
lem	nb	0.726521	(CountVectorizer(max_df=0.8, max_features=5000...	0.834951	0.723577	0.836478	0.635714	0.839623
lem	svm	0.705447	(CountVectorizer(max_df=0.8, max_features=5000...	0.849515	0.690000	0.784622	0.734043	0.650943
lem	knn	0.524175	(TfidfVectorizer(max_df=0.8, min_df=3, ngram_r...	0.817961	0.539877	0.686305	0.771930	0.415094

Model Analysis

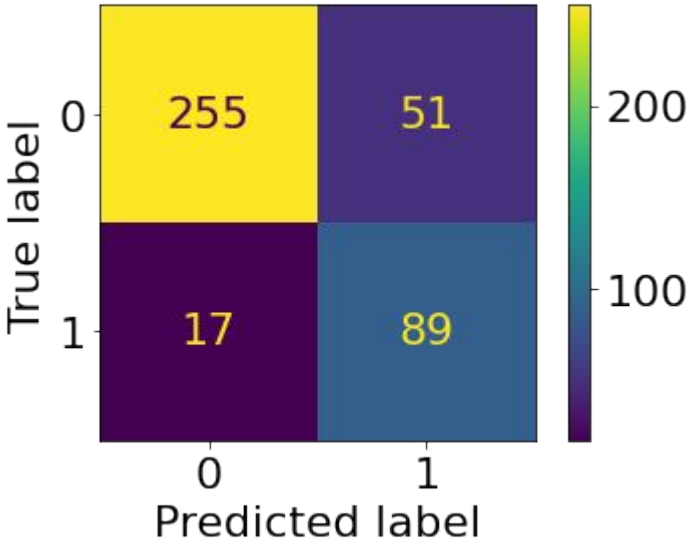
- **Logistic Regression** had the highest test **accuracy** (0.862) and **precision** (0.795)
- **Multinomial NB** had the highest test **F1 score** (0.724), **ROC AUC** (0.836) and **recall** (0.840)
- If we want to maximize precision and minimize false positives, Logistic Regression would be the best model
- If we want to maximize recall and minimize false negatives, Multinomial NB would be the best choice

Model Analysis - Confusion Matrix

Logistic Regression



Multinomial NB



Model Analysis - Top Words

Logistic Regression top positive coefficients (r/TalesFromRetail)

feature	coefficient	abs coefficient
customer	0.886203	0.886203
retail	0.756142	0.756142
register	0.628632	0.628632
come	0.555152	0.555152
sold	0.544022	0.544022
work	0.539236	0.539236
happened	0.438185	0.438185
worked	0.417943	0.417943
looked	0.417407	0.417407
man	0.410243	0.410243

Logistic Regression top negative coefficients (r/TalesFromTheCustomer)

feature	coefficient	abs coefficient
went	-0.702833	0.702833
employee	-0.654883	0.654883
service	-0.626101	0.626101
cashier	-0.554027	0.554027
mask	-0.534601	0.534601
decided	-0.525693	0.525693
local	-0.512941	0.512941
home	-0.504258	0.504258
heard	-0.475553	0.475553
probably	-0.443307	0.443307

Model Analysis - Top Words

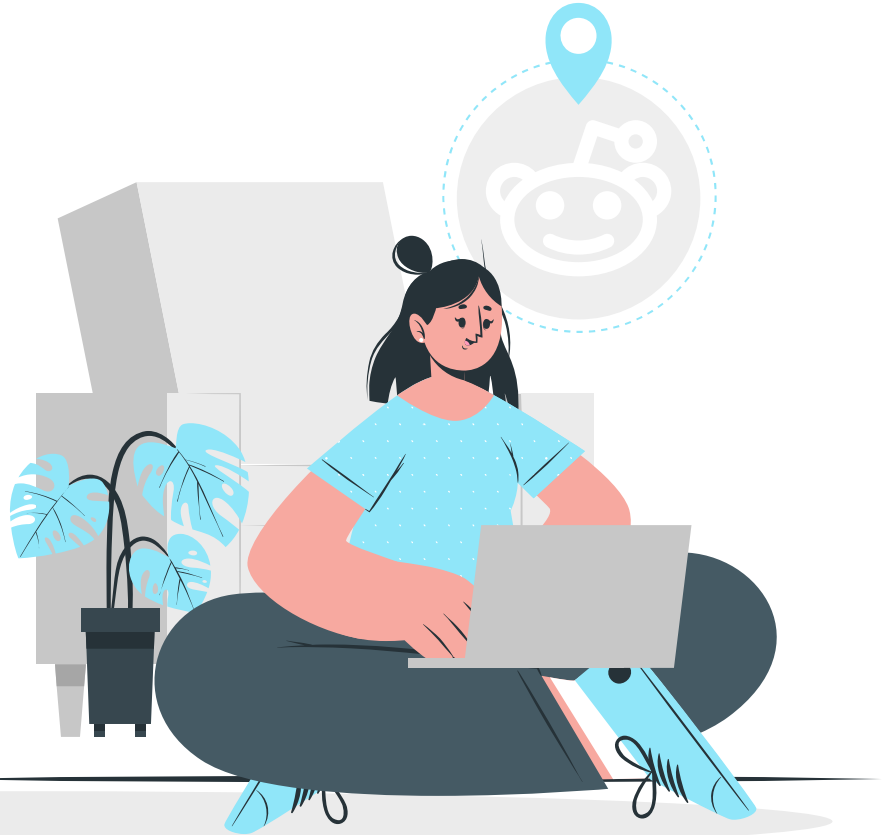
Multinomial NB top positive class probabilities (r/TalesFromRetail)

feature	neg class prob	pos class prob
store	0.006232	0.012572
customer	0.004850	0.012266
time	0.007952	0.006299
like	0.006524	0.006197
manager	0.003047	0.006095
said	0.006378	0.005253
work	0.003761	0.005177
come	0.003203	0.005151
day	0.007074	0.004947
guy	0.003148	0.004641

Multinomial NB top negative class probabilities (r/TalesFromTheCustomer)

feature	neg class prob	pos class prob
time	0.007952	0.006299
day	0.007074	0.004947
like	0.006524	0.006197
said	0.006378	0.005253
store	0.006232	0.012572
got	0.006232	0.003876
know	0.005619	0.004437
say	0.005417	0.004590
order	0.005024	0.002321
told	0.004932	0.004157

Conclusions



Conclusions

- 2 effective and interpretable **classification models** were built to determine if a post is from **r/TalesFromTheCustomer** or **r/TalesFromRetail**:
  - Logistic Regression (maximize precision)
  - Multinomial NB (maximize recall)

- Precision and recall scores could be further improved by creating an **ensemble model**, where misclassified posts are given higher weight in model training, and an aggregate prediction is given by a combination of several models

- These NLP models could be recontextualized to train on **actual customer and service staff feedback** to identify the most frequent words in the actual feedback from customers and retail staff

- Retail executives may want to perform **sentiment analysis** to identify if the feedback from their customers and retail staff are generally positive or negative

Thank You

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, infographics & images by Freepik and illustrations by Stories