https://www.sneakyartist.com/blog/reddit-101

Project 3 – Reddit NLP

Hi this is my project 3 on the Reddit NLP

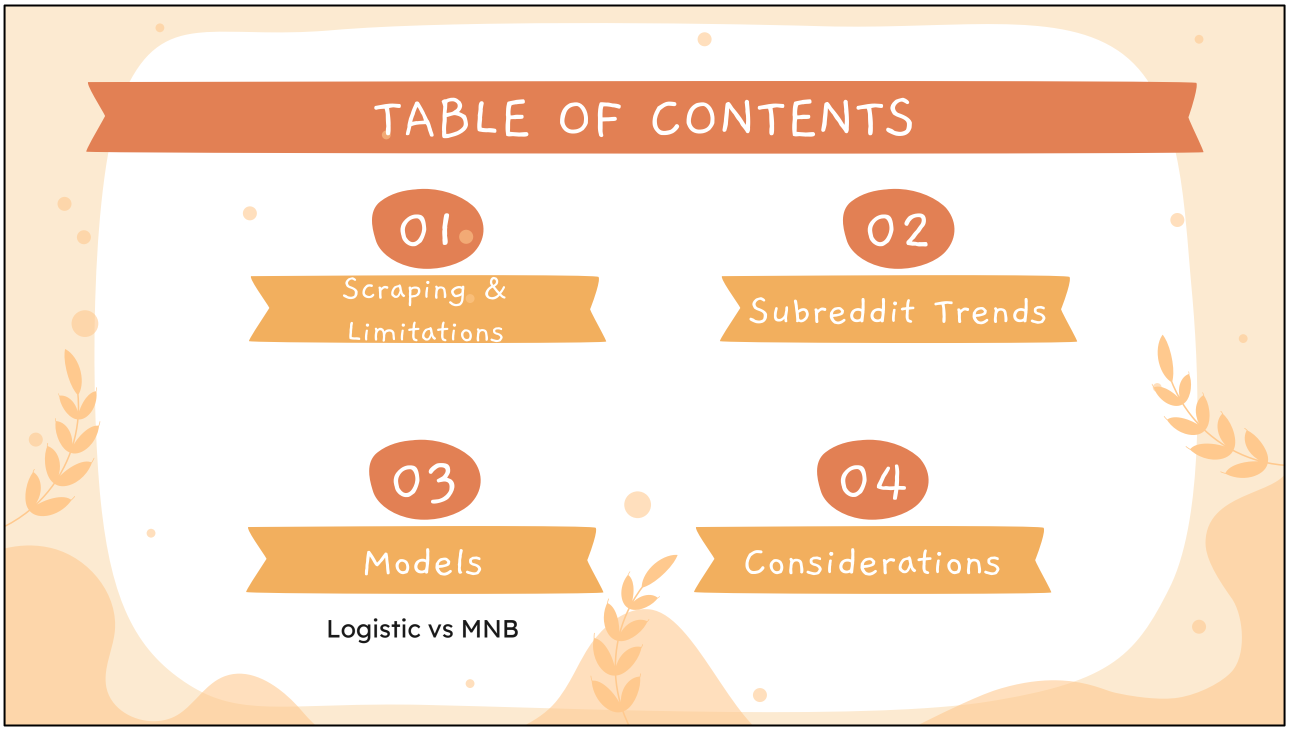


TABLE OF CONTENTS

01 Scraping & Limitations

02 Subreddit Trends

03 Models

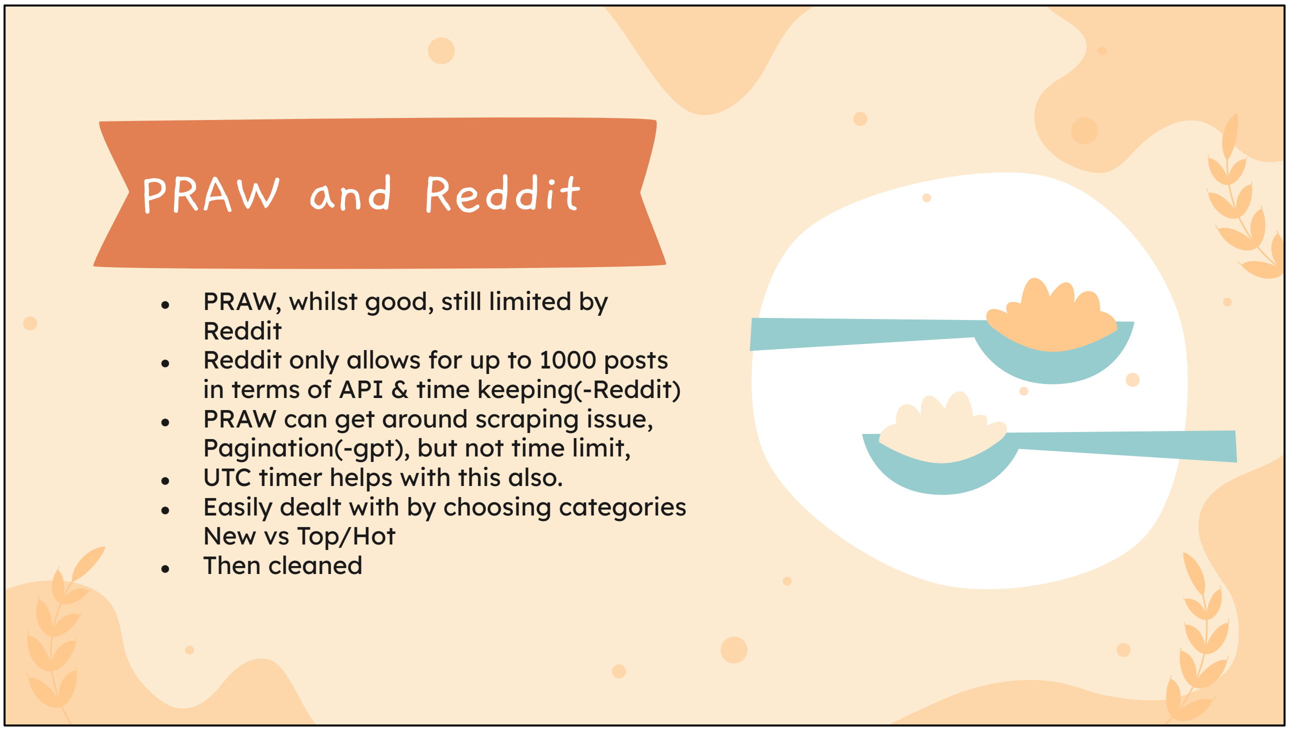
Logistic vs MNB

04 Considerations

Just a quick run through of what I'll be tackling here. Firstly, How I got the data and the problems I ran into whilst doing so, Then, since this is a project on NLP I thought I'd see any noticeable trends within each subreddit before we modelled them, next Building classification models, LR vs MNB, to see if we could build a model that can classify posts accurately to their respective subreddit, which is the aim of this project, and analyse them; and Finally Notable considerations.

01

Scraping and Limitations



PRAW and Reddit

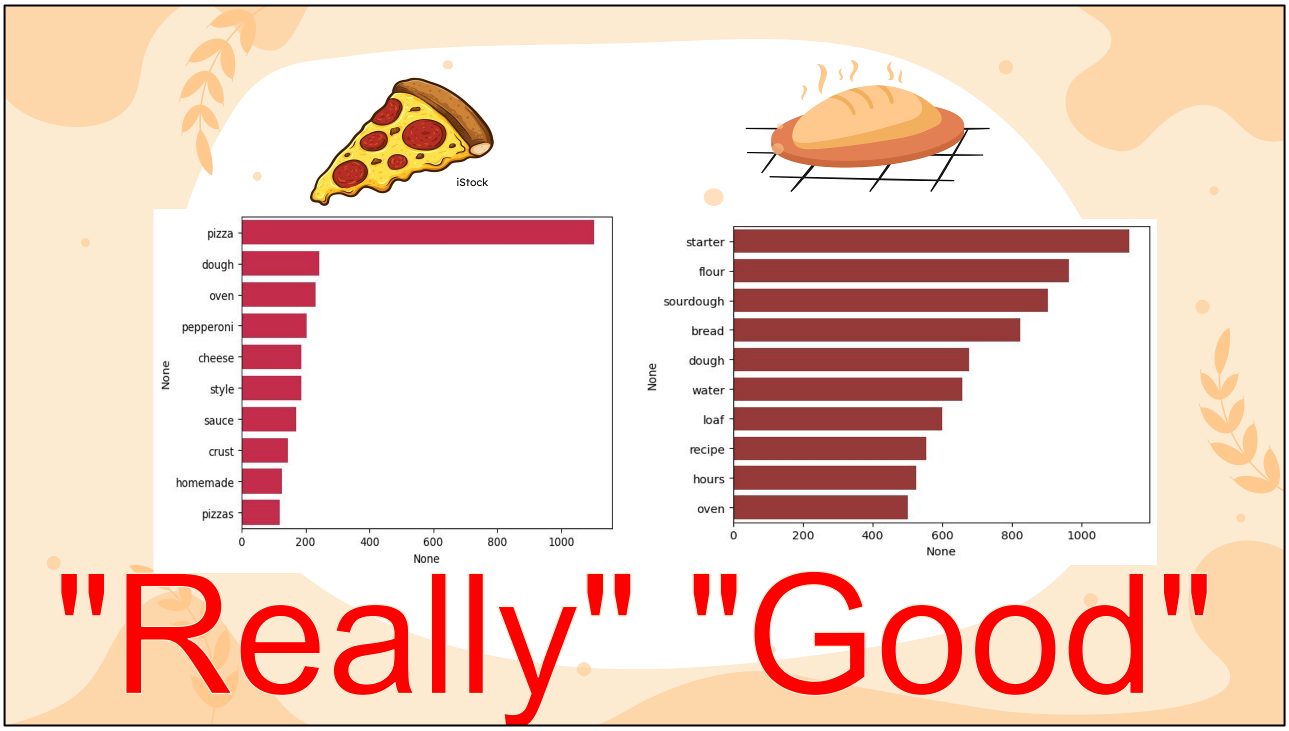
- PRAW, whilst good, still limited by Reddit
- Reddit only allows for up to 1000 posts in terms of API & time keeping(-Reddit)
- PRAW can get around scraping issue, Pagination(-gpt), but not time limit,
- UTC timer helps with this also.
- Easily dealt with by choosing categories New vs Top/Hot
- Then cleaned

The aim of this project is to build a classifier that can accurately predict which posts were from which subreddit, but this is preceded by scraping reddit for posts.

Whilst API scraping is limited by Reddit, PRAW can overcome these issue by sending multiple requests, known as pagination. It works in tandem with other processes, such as adding a UTC timer upon creation, but pagination is the reason why you could scrape 1000 instantaneously, despite Reddit having a 100 posts per request limit. By sending multiple requests that link up right after another, using after and before tokens(they're indicators for APIs used in pagination) they seamlessly combine the multiple chunks into one. But reddit actually has a max 1000 post limit when it comes to it's actual api scraping, as well as in their UI, that you cannot go beyond 1000 posts in time. So you couldn't just send another request beyond the last one separately, unless you used some sort of third party app. However, this is easily overcome, as you could change the categorisation of the posts, ones that didn't have massive overlap such as New and Top, and then remove duplicates whilst concatenating the dataframes. Which is how I was able to scrape easily over 1000 posts.

02

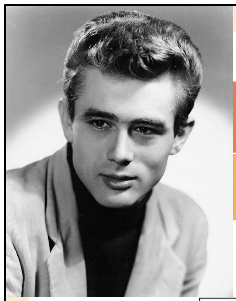
Subreddit Trends



"Really" "Good"

I chose to use the Pizza and Sourdough subreddits as they are distinct yet would definitely have some overlap, as well as I'm a member of each subreddit. We can see that there isn't that much overlap regarding the 2 subreddits, aside from topics of baking, but the spread of each subreddit is interesting to see, that pizza is much more confined to literally pizza, whereas sourdough had a more diverse portfolio regarding keywords, which comes into play much later. Interestingly pizza and pizzas was in the top 10 of most common in pizza words, showing just how much the subreddit had to do with literal pizza, than the working of it, as opposed to sourdough, whose subreddit is a lot more compartmental. The pizza and pizzas is an interesting note, as I decided not to lemmatize the posts, as we were dealing with a social forum, and thought that small nuances and informal language would provide an insight regarding language used.

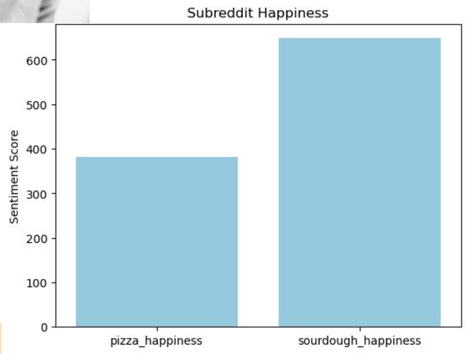
Interestingly, i made a list of shared words in the top 50 which generally just shared vocabulary pertaining to actual baking and the making of each, but included the words "really" and "good" which were the only descriptive terms shared, which we all can understand when it comes to pizza and sourdough, which leads me to my next slide.

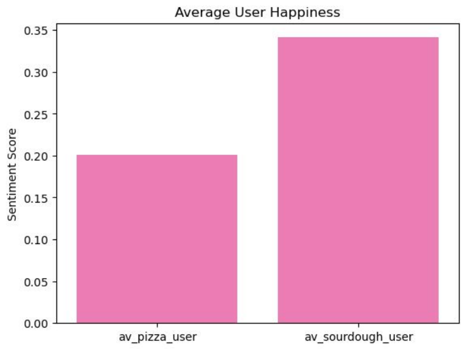


That's Amore?

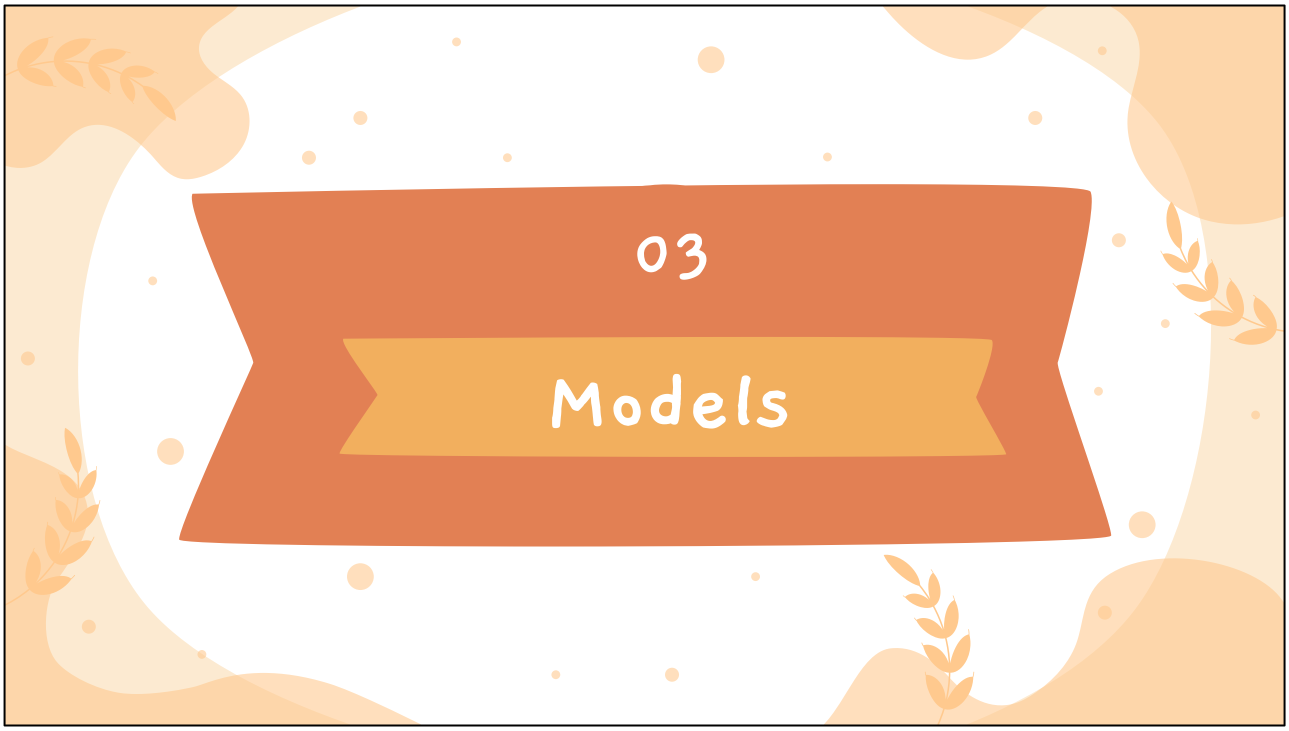
Overall Positivity

Average Positivity





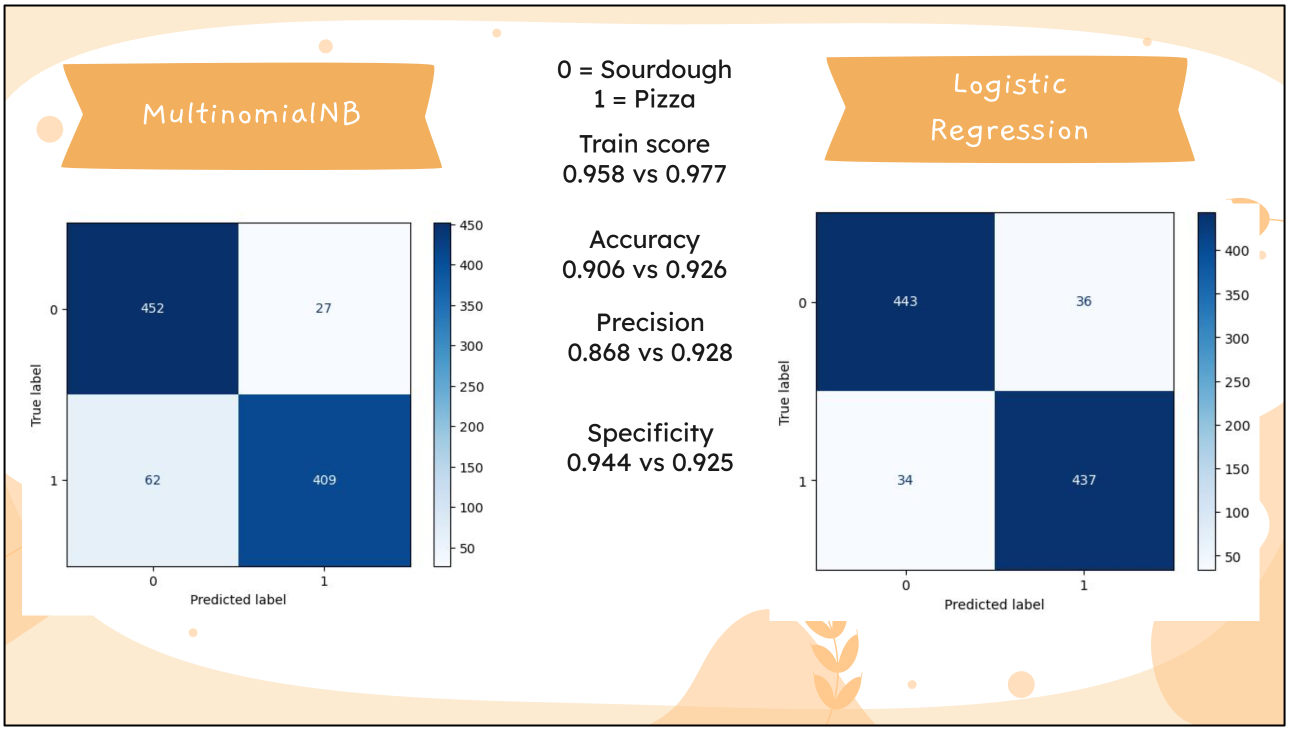
We all know everyone likes bread and pizza, but it's better understood that the whole world loves pizza, so I thought it was worth seeing what kind of sentimental language each subreddit used, via the sentiment analyser. Of the 1900 posts, of each, pizza actually lost out on total positive sentiment, AND average user sentiment, shockingly. the sourdough users seemed to purvey their love of sourdough more ecstatically than pizza users, as well as being generally more happy to talk about sourdough. This may show that although everyone loves pizza, those who love sourdough, are more fanatic about it.



03

Models

On to our classification models



0 = Sourdough
1 = Pizza

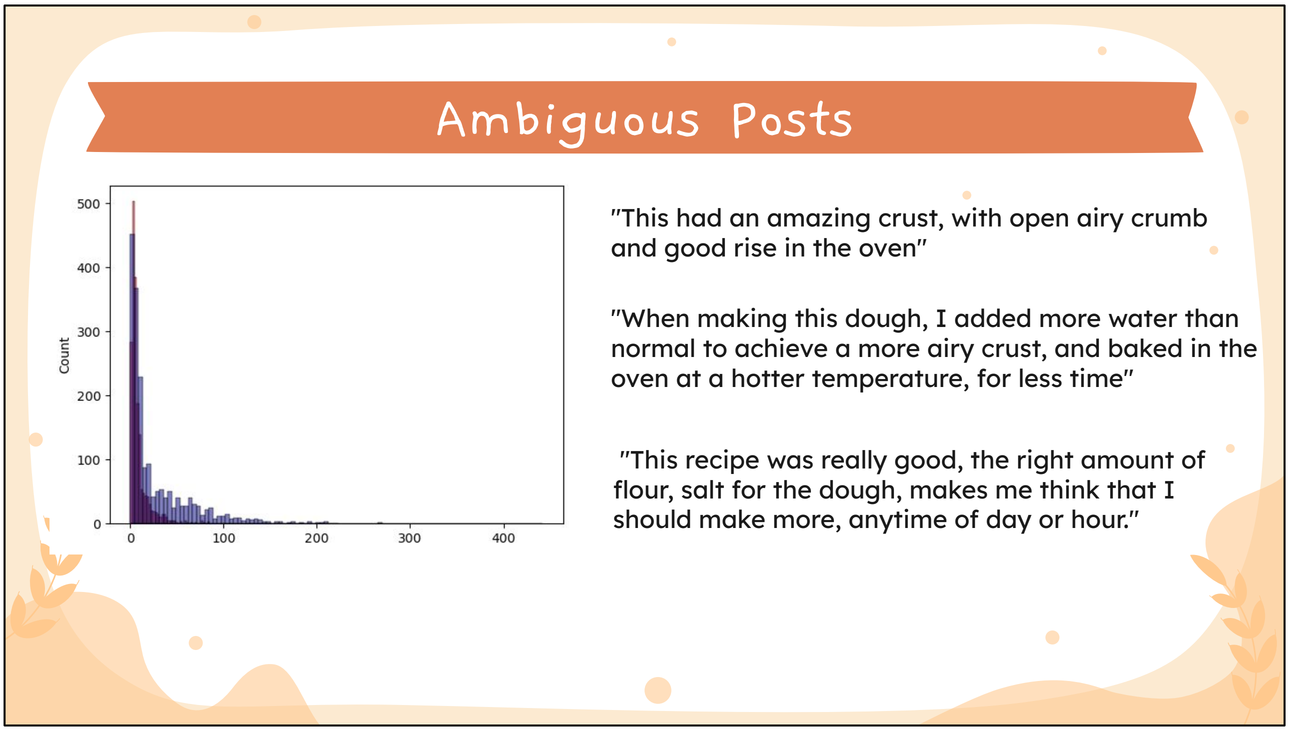
Train score
0.958 vs 0.977

Accuracy
0.906 vs 0.926

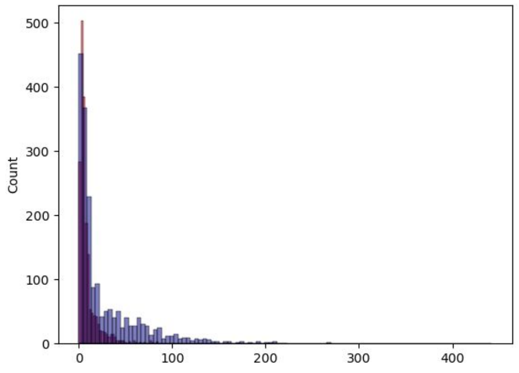
Precision
0.868 vs 0.928

Specificity
0.944 vs 0.925

So we decided to go with Multinomial Naive Bayes, and Logistic Regression when creating our classification models, instantiating a pipeline and gridsearch so that we could fine tune the parameters more easily, and preprocess and fit the model through the multitude of posts in one smooth go. Both actually performed decently well. The baseline model accuracy was 50%, as i literally just set it that way to try and avoid a set imbalance and bias, 1900 posts each. It turns out the best parameters in both cases were just the basic setting, stopwords=english, l2 error, unigram, adding to the idea that social forums work on informal and simple language.
Despite all this, Logistic Regression came out on top, with all it's metrics beating MNB, except for its sensitivity, perhaps indicating that both models were actually better at predicting the 0, sourdough posts, over pizza posts, and this might track. Just to note, in terms of classification, its accuracy is the same as the test set score, as accuracy is based on True predictions.



Ambiguous Posts



"This had an amazing crust, with open airy crumb and good rise in the oven"

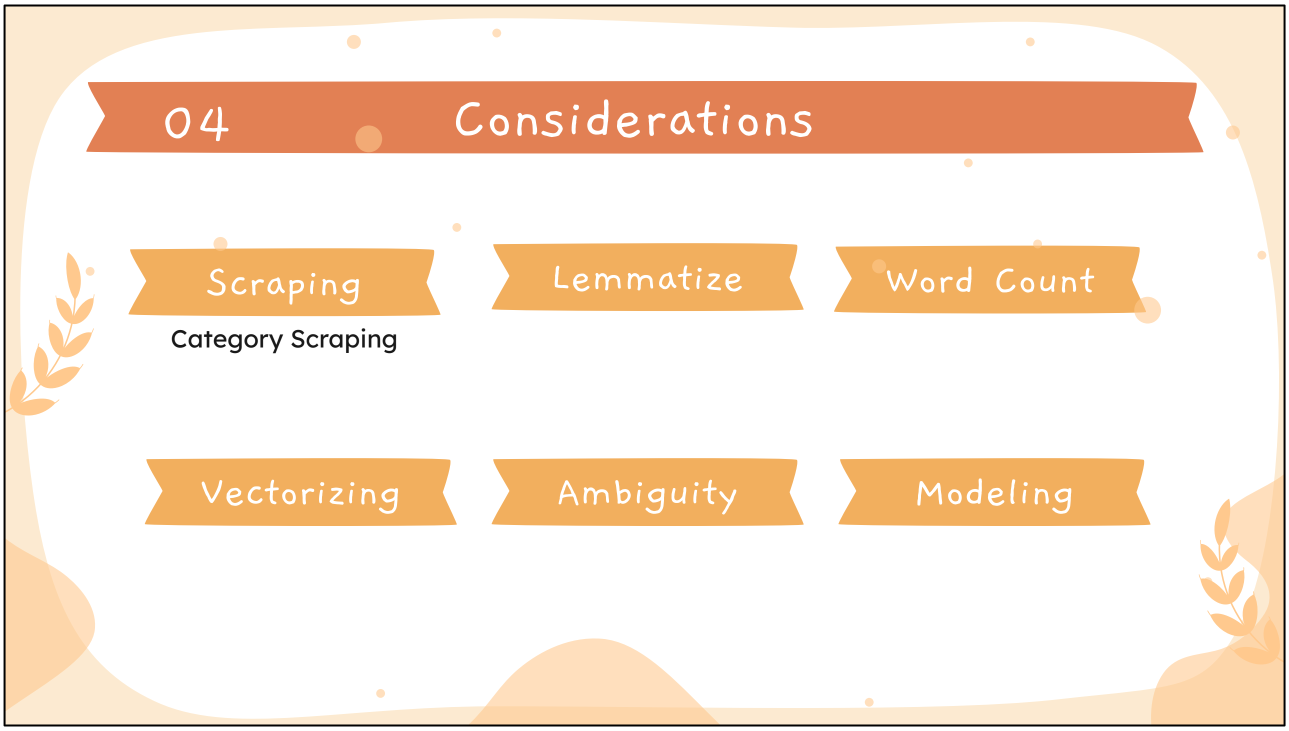
"When making this dough, I added more water than normal to achieve a more airy crust, and baked in the oven at a hotter temperature, for less time"

"This recipe was really good, the right amount of flour, salt for the dough, makes me think that I should make more, anytime of day or hour."

So we mentioned earlier, that the spread of keywords in each subreddit was quite different, as shown in the graph here, although both show a massive right-skew, there is more navy density overall, which is sourdough, than red, which is pizza. We can also see that the pizza popular words drops off much much earlier than the sourdough keywords, which trail on a bit further towards the 200 mark, as opposed to 50 for pizza, showing that our models were probably trained on fewer words for identifying pizza, than for sourdough, which checks out for our specificity scores, which all surpassed our predictive scores. This led me to think that the models would just classify ambiguous posts more towards sourdough, as there were more keywords to identify with, and it would theoretically put less weight on ambiguous words, due to pizza having less variety overall.

The sentences you see are examples I came up with to test the predictive models. The first I made with sourdough in mind, but ambiguous as possible, second was pizza but ambiguous, and the final one being a sentence made up of all the shared terms that were in the top 50 of both subreddits. The model predicted the first 2 correctly, but interestingly predicted the most ambiguous one sourdough, which follows our intuitions regarding model training.

Despite all this, I would say that our models have performed pretty accurately, being roughly 40% points above our baseline accuracy, considering the nature of the subreddits overlap as well.



04 Considerations

Scraping

Category Scraping

Lemmatize

Word Count

Vectorizing

Ambiguity

Modeling

Considerations.

Earlier, when I mentioned overcoming reddits 1000 post limit, this only truly works with massive subreddits, where the category filtering would provide enough difference. In smaller subreddits with less posts, there's bound to be much more overlap, and so this method wouldn't necessarily work.

Perhaps lemmatizing would provide better key word searches, and group together more that have just been abbreviated, given how skewed r/pizza was, may have helped with the modelling, but during our modelling process, we actually encountered errors that indicated that the language was already too simple, and the preprocessing it even further, such as lemmatizing/stemming and raising min-df would not create enough data points to work with.

As we saw that the classifier probably went off of less words for pizza classification, we perhaps would have got better predictive models if there was a more diverse range of words for our classifiers, which would

Just to note, I did countvectorize for the subreddits, and used Tfidf(term frequency inverse document frequency) vectorizing for the model, as indirectly, classification would work better if there were features that linked the text and documents to some target variable, our subreddits, as opposed to just counting and tokenizing, which should have potentially accounted for the small vocabulary variety of r/pizza.

Regarding ambiguity, it may actually be that in our previous example, the shared words actually were just higher on the top50 list for sourdough, than for pizza. As well as the fact that perhaps since I came up with the statements, they weren't truely ambiguous as I thought.

Adding on from the point of ambiguity, perhaps it was just that my model wasn't good enough, and that it was too simple, that there were even more subtle informal nuances that I did not pick up on. the previous graph really tapers off at 50 for pizza. and perhaps we should have scraped more posts, or a subreddit that wrote more, or perhaps didn't use as simple or as informal language. Or, use other classification models such as random forest, or even exploring more potential parameters, but I feel the latter wasn't so much the case here, but the simplicity of the language, which may have made an overfit or too simple a model for us, which the metrics sort of indicate, which works in this scenario, but makes the model in of itself not that adaptable.

RESOURCES

- Reddit - r/Sourdough & r/Pizza
- Lesson Notes

Fin!

Do you have any questions?

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

Please keep this slide for attribution

Thank you for listening, any questions?