NLP and r/DadJokes: A Difficult Pun-dertaking

Owen Curtis

Executive Summary

Background

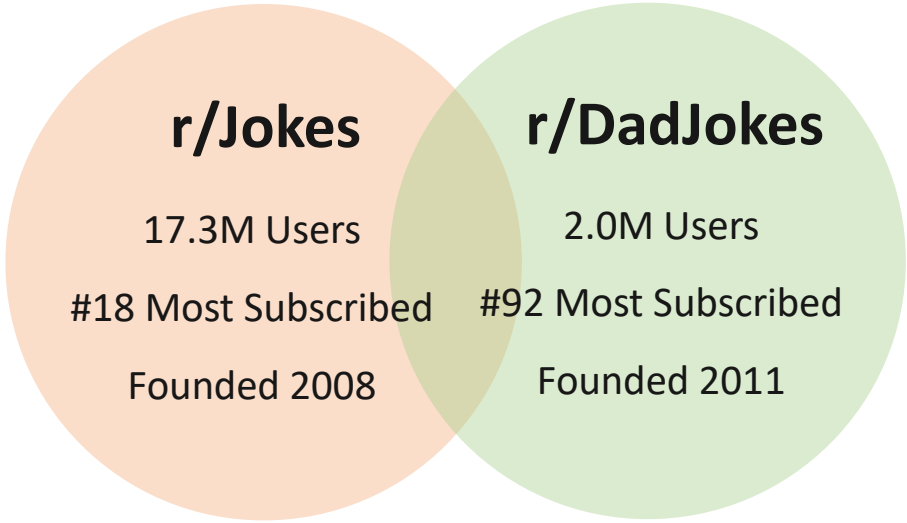
Process

Insights

Wrapping Up

Background: The Subreddits

- **r/Jokes, r/DadJokes** both thriving communities with a significant amount of **user overlap**
- **Goal 1: Prototype classification system to guide posting to appropriate subreddit**
- **Goal 2: NLP v Non NLP?**



Background: The “Dad Joke”

Did you hear about the restaurant on the moon?

Great food, no atmosphere.

What do you call a fat psychic?

A four-chin teller.

Why did the scarecrow win an award?

Because he was outstanding in his field.

Did you hear about the kidnapping at school?

It's fine, he woke up.

Commentary:

Don't ban the "Dad" jokes; preserve and revere them

Surrogate fathers to the rescue

Memorial Day a success

They did a good deed

Halloween Party sale

Looking at language: 'Alphabet soup'

Likes Third Ward plan

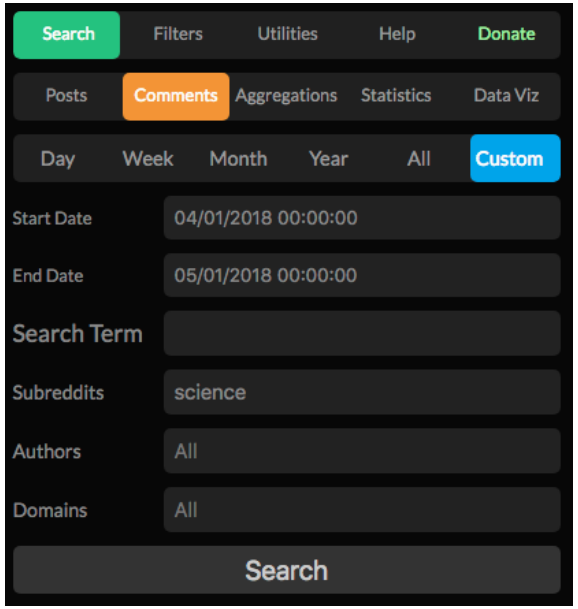
Earliest Known Public Acknowledgement of Joke Type: 1984. *Gettysburg Times*

Approach Summary

Data Processing	Modeling	Insights
Collection	Overview	Differentiating Features
Post/Comment Pruning	**NLP-Based** Model	Misclassification
Feature Engineering	**Non-NLP Based Model**	
Initial EDA		

Data Processing: Collection

Reddit PushShift API



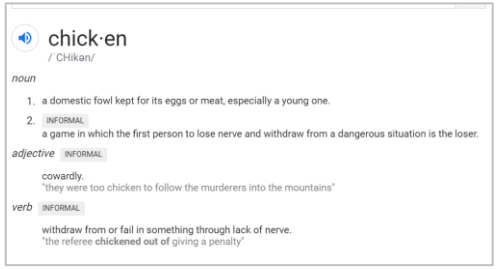
Posts
~12,000

Reddit JSON

Comments
~60,000

Wikipedia

Slay / Sleigh



Homophones/Homonyms
~1,000 Words

Data Processing: Post/Comment Pruning

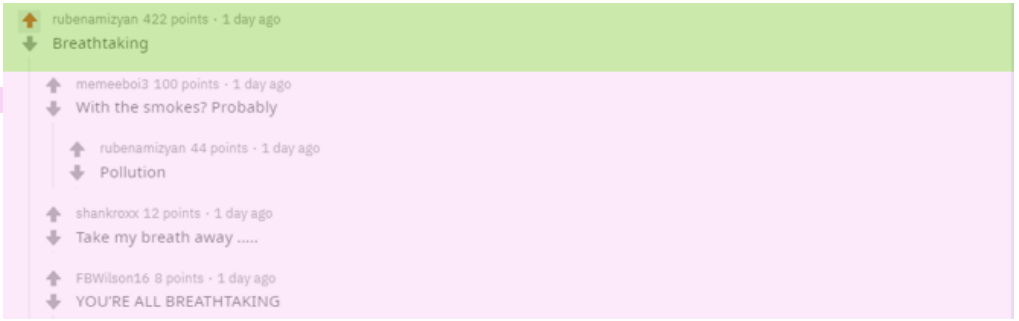
Drop Post if…

- Post has no Title
- Score less than 8
- Title Contains /r
- Images/gif Based
- Inordinately long/short body

Final Count: ~2,000

Drop Comment if…

- Not first tier comment
- Not a top ten comment in high-comment thread



Final Count: ~15,000

Data Processing: Feature Engineering

New Features

Joke Length

Word Count of Title, Body

Profanity Flag

'Shit'
Over_18 Flag

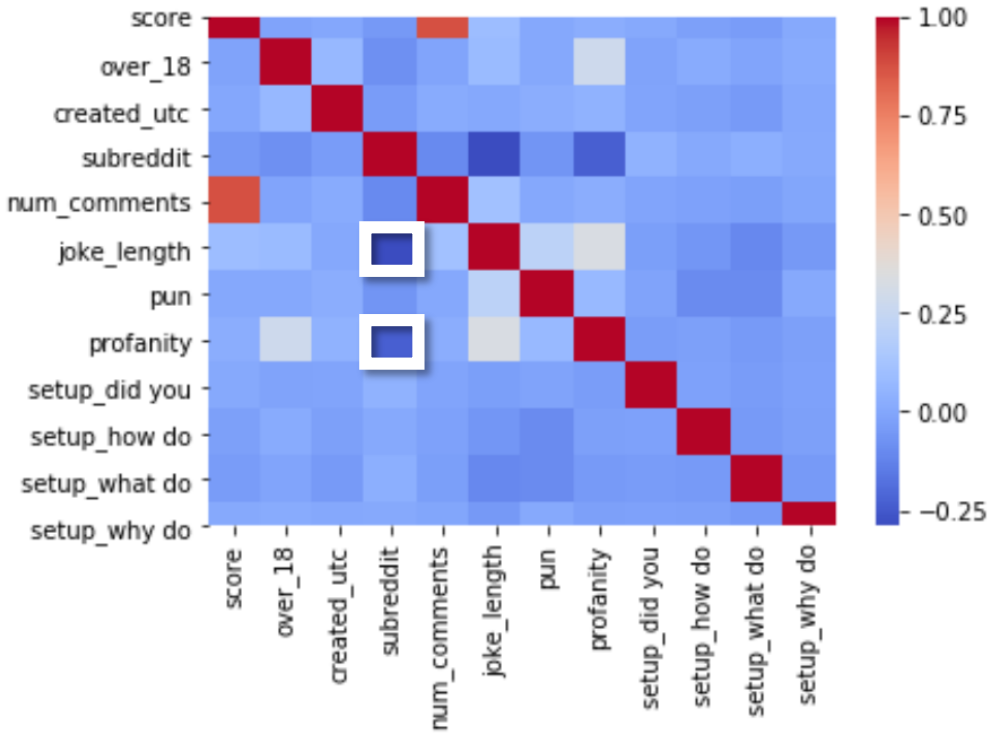
Pun Flag

Binary
Homonym/Homophone
Detection

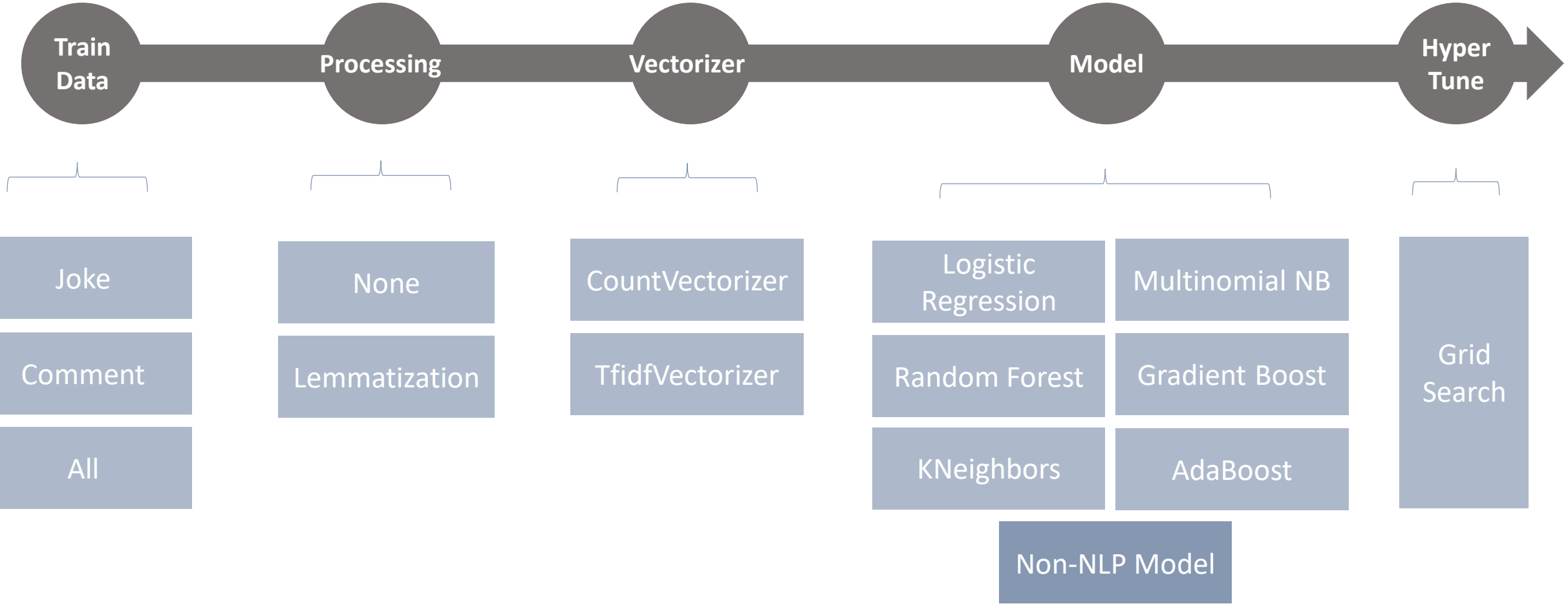
Joke Setup Dummies

"What Do", "Why Do",
"How Do," "Did You"

Feature Correlation



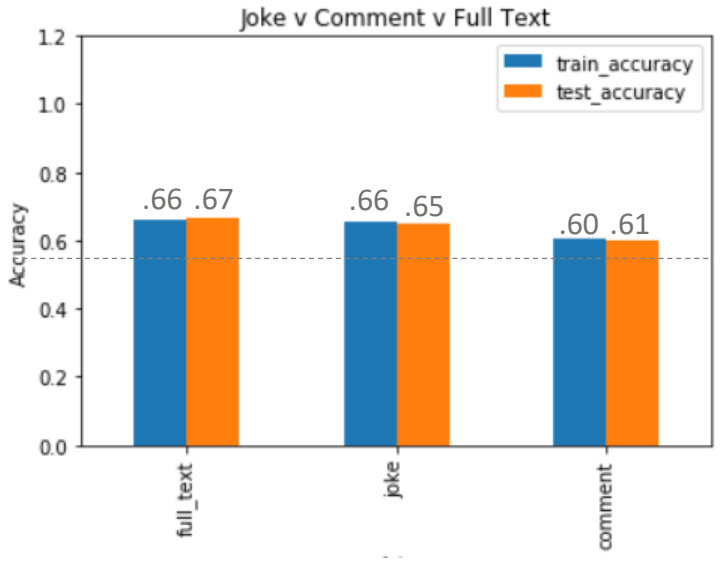
Modeling: Overview



Modeling: Testing Results

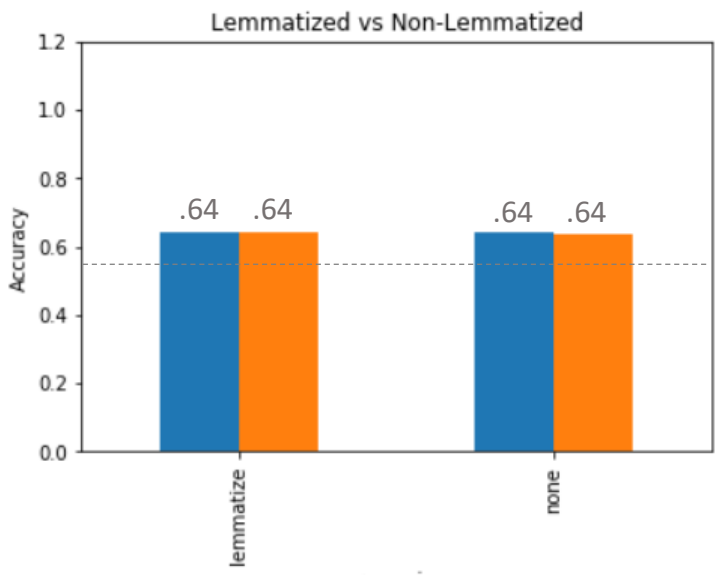
Accuracy Benchmark: 0.58

Training Data



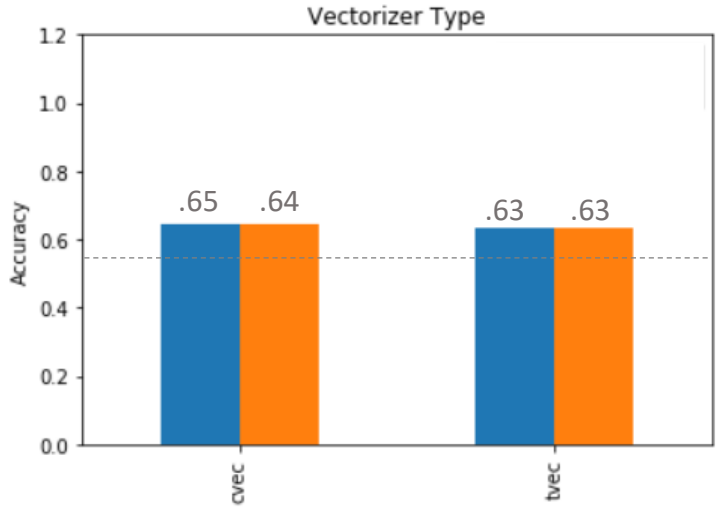
Impact: High

Processing



Impact: Low

Vectorizer



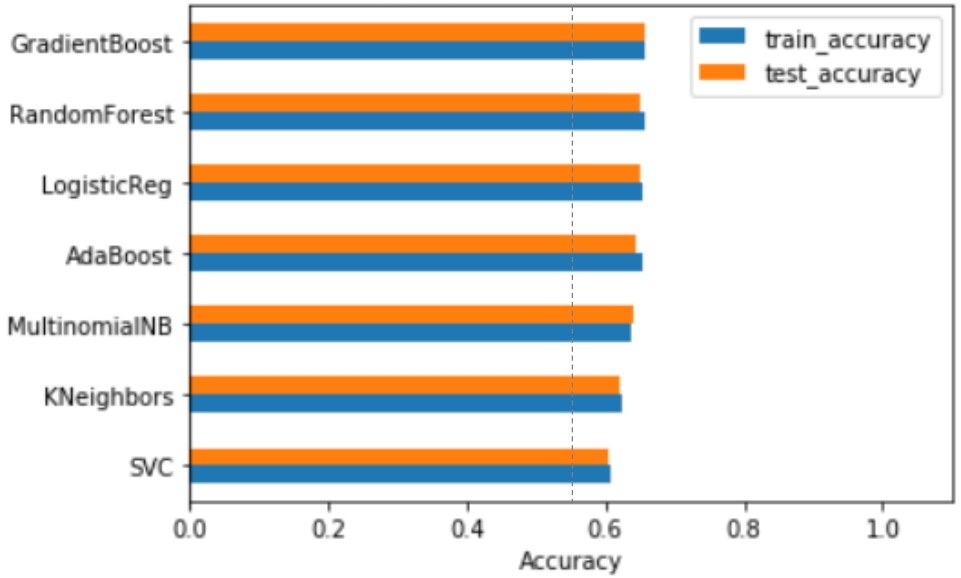
Impact: Low

**Accuracy Benchmark: 0.58*

Modeling: Testing Results

Accuracy Benchmark: 0.58

Aggregate Model Performance



model	train_accuracy	test_accuracy	abs_diff
GradientBoost	0.658	0.656	0.011
RandomForest	0.657	0.650	0.009
LogisticReg	0.653	0.650	0.009
AdaBoost	0.653	0.645	0.014
MultinomialNB	0.637	0.642	0.009
KNeighbors	0.625	0.622	0.013
SVC	0.607	0.604	0.003

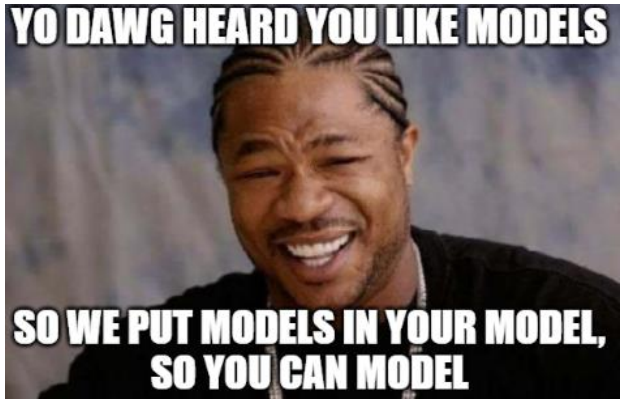
Impact: High

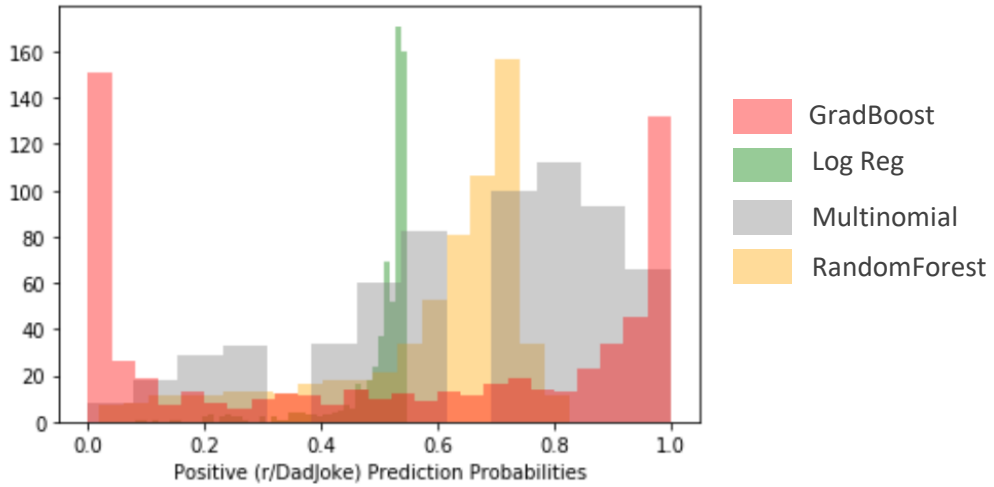
**Accuracy Benchmark: 0.58*

Modeling: Final Recipe (Voting Classifier)

Accuracy Benchmark: 0.58

	NLP
Recipe	**Voting Classifer** (GradientBoost, RandomForest, LogisticReg, Multinomial)
# Features	1300
Train Accuracy	**.69**
Test Accuracy	**.70**
Difference	**.01**





**Accuracy Benchmark: 0.58*

Modeling: How does it compare to simple non-NLP model?

Accuracy Benchmark: 0.58

	NLP	Non NLP
Recipe	**Voting Classifer** (GradientBoost, RandomForest, LogisticReg, Multinomial)	**Voting Classifer** (GradientBoost, RandomForest, LogisticReg, Multinomial)
# Features	1300	8
Train Accuracy	**.69**	**.64**
Test Accuracy	**.70**	**.71**
Difference	**.01**	**.07**

Accuracy score in the **70%** range

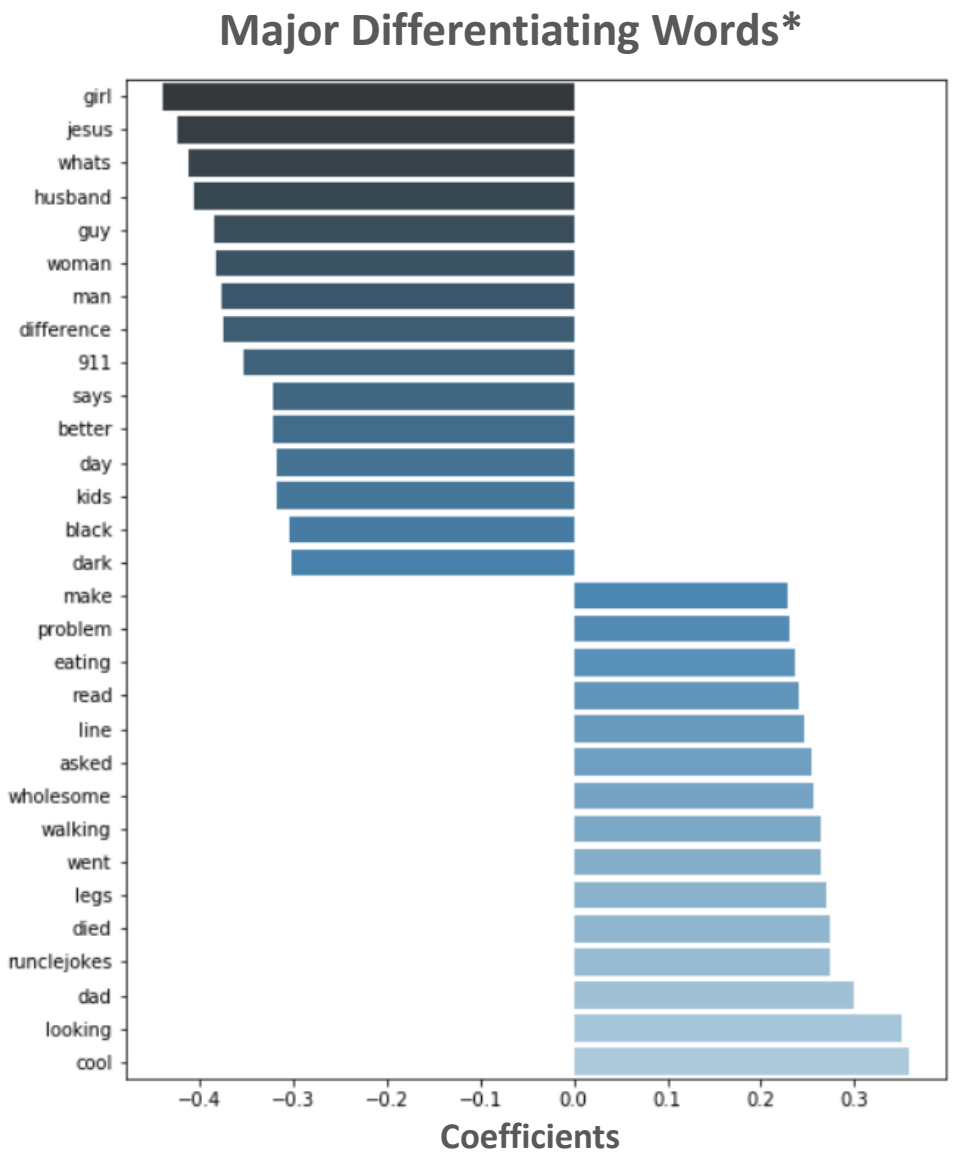
Remembers benchmark is **58%**

**Accuracy Benchmark: 0.58*

Insights: Differentiators

r/Jokes

- Couples/Sex
- Religion
- Tragedies
- Race
- "**General Darkness**"



r/DadJokes

- Wholesome
- Family
- Simple contexts (eating, walking)

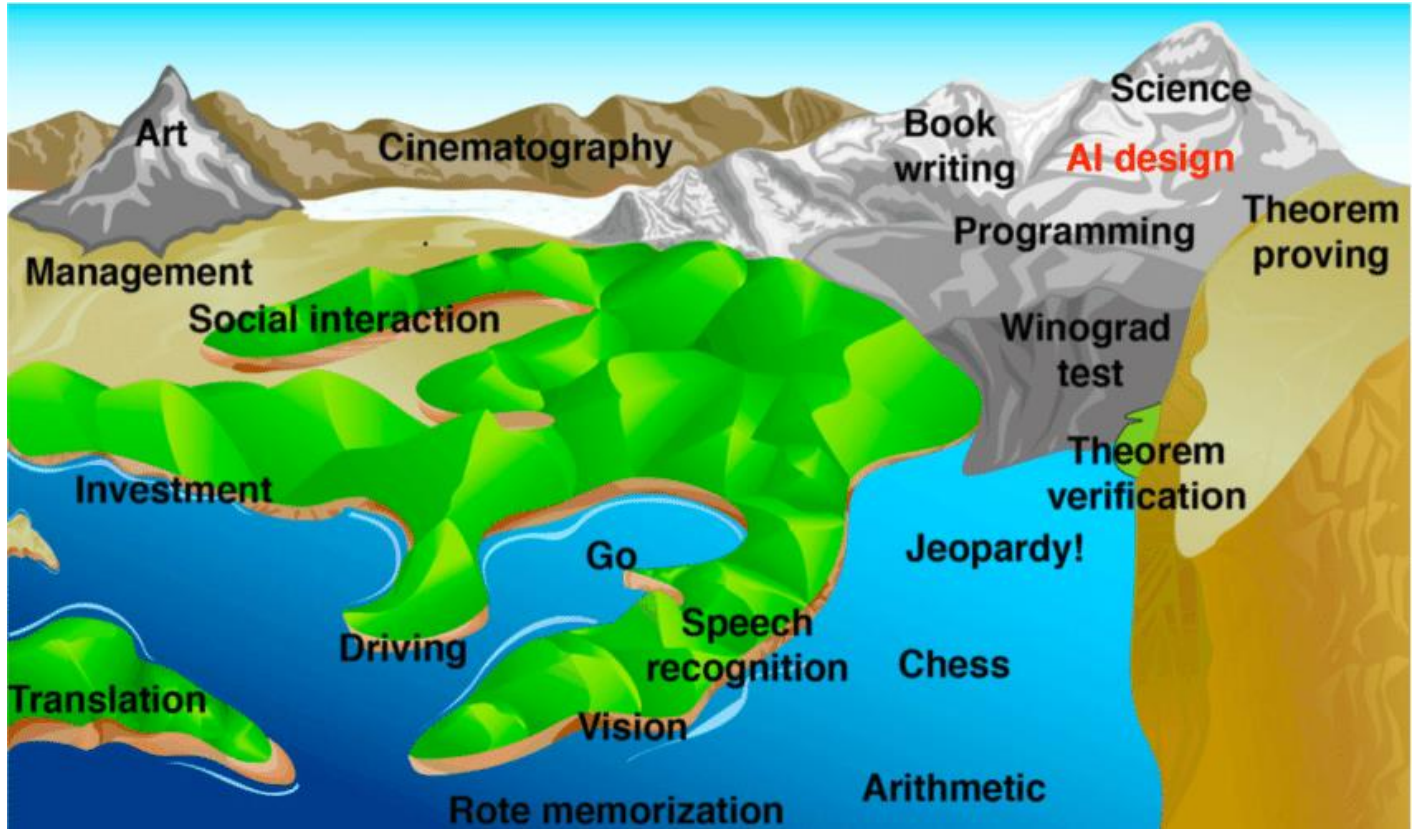
Insights: Very Confident…and Very Wrong

Bucket	In English…	Examples
High Confidence False Positive	**Prediction:** r/DadJokes **Actually:** r/Jokes	• "What's yellow and can't swim? A schoolbus full of children." • "Why is it not good to hear high pitch sounds? It megahertz."
High Confidence False Negatives	**Prediction**: r/Jokes **Actually:** r/DadJokes	• "I have a fear of speed bumps but I'm slowly getting over it." • "son dad can you tell me what a solar eclipse is dad no sun How punny did he tell him **SUN OF A BITCH**"

Sensitivity	Specificity
.90	.38

Prediction Probability Threshold = 0.9

Wrapping Up: The “Art” of The Joke



Source : Max Tegman, Life 3.0

r/SubsimulatorGPT2

Stepping Back: GPT-2 and Reddit

AI Jokes

Posted by u/jokesGPT2Bot 4 days ago

132

What do you call a robot made out of meat?

MIXED

A butcher.

66 Comments Share Save Hide Report 99% Upvoted

r/SubSimulatorGPT2 · Posted by u/jokesGPT2Bot 2 months ago

I've lost my sense of humor

jokes

I can't get out of bed.

63 Comments Share Save Hide Report 100% Upvoted

AI Movie Discussion

What is the best scene in a film that you've ever seen?

MIXED

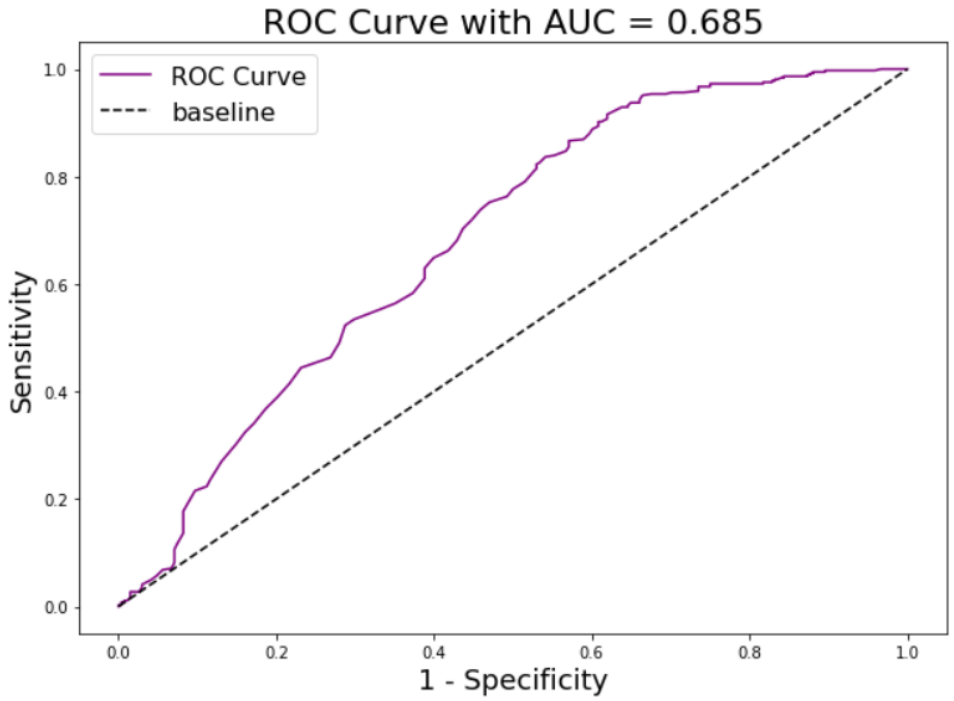
I watched Mad Max Fury Road, and I was surprised by how many great scenes. I always really like how Max's character is the one who makes the most of every situation. Even though Fury Road is a pretty serious movie, I really liked how Mad Max was more comedic and less serious than most action movies. Also, I think one of the greatest scenes in any movie is the time Mad Max is on the roof. The music in the scene is wonderful.

What is your favorite scene? And why?

28 Comments Share Save Hide Report 100% Upvoted

THANK YOU

Insights: ROC

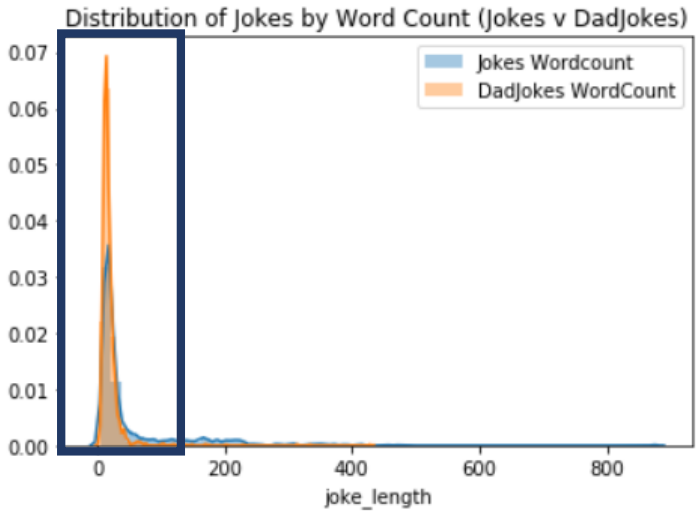


Sensitivity	How effectively model classifies **r/DadJokes**	**.90**
Specificity	How effectively model classifies **r/Jokes**	**.38**

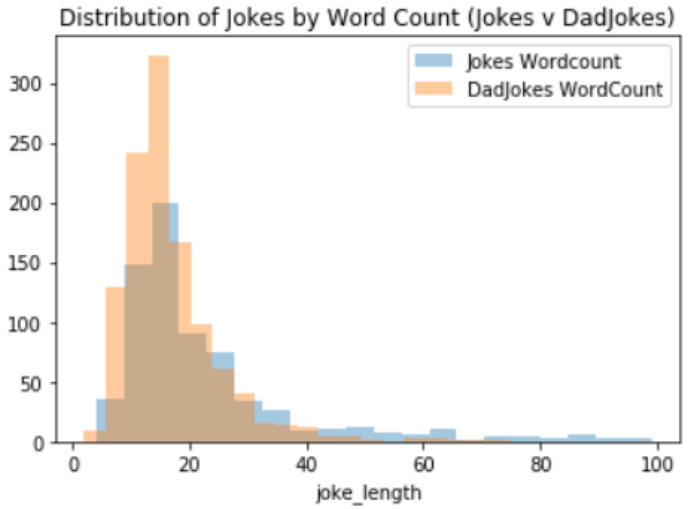
**Performance similar for non NLP model against these KPIs.*

**Logistic Regression Model for interpretability*

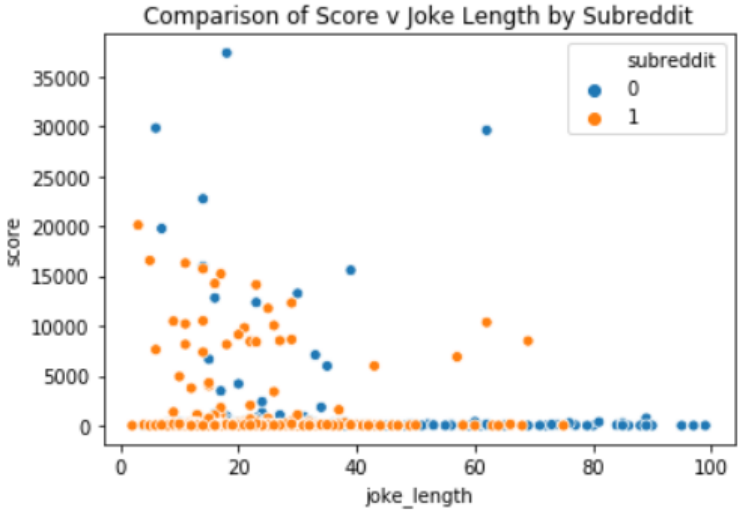
Data Processing: Initial EDA



Distribution of Joke Length is **heavily right skewed for both subreddits**



Isolating for **100 words or less**, we see **r/DadJokes** post numbers climb



Comparing Joke Length vs Score, we see **more shorter, higher scoring dad jokes.**

Data Wrangling: Feature Engineering

Characteristics	Feature Opportunities
They’re mostly short…	Joke Length
Often involves wordplay	Homophone, Homonym Flag
Kid appropriate	Profanity Flag
What Do, Did You…	Setup Flag

Individual Breakout

	data	processing	vectorizer	model	train_accuracy	test_accuracy	abs_diff
10	full_text	lemmatize	cvec	GradientBoost	0.679	0.698	0.019
46	full_text	lemmatize	cvec	RandomForest	0.684	0.696	0.012
47	full_text	lemmatize	tvec	RandomForest	0.699	0.696	0.003
11	full_text	lemmatize	tvec	GradientBoost	0.677	0.694	0.017
22	full_text	lemmatize	cvec	AdaBoost	0.673	0.694	0.021
16	full_text	none	cvec	AdaBoost	0.678	0.691	0.013
23	full_text	lemmatize	tvec	AdaBoost	0.687	0.685	0.002
35	full_text	lemmatize	tvec	MultinomialNB	0.656	0.685	0.029
59	full_text	lemmatize	tvec	LogisticReg	0.672	0.683	0.012
52	full_text	none	cvec	LogisticReg	0.694	0.683	0.010
40	full_text	none	cvec	RandomForest	0.679	0.682	0.003
17	full_text	none	tvec	AdaBoost	0.679	0.682	0.003
4	full_text	none	cvec	GradientBoost	0.683	0.682	0.001
5	full_text	none	tvec	GradientBoost	0.670	0.677	0.007
41	full_text	none	tvec	RandomForest	0.691	0.676	0.016
6	joke	lemmatize	cvec	GradientBoost	0.682	0.676	0.006
58	full_text	lemmatize	cvec	LogisticReg	0.687	0.676	0.012
0	joke	none	cvec	GradientBoost	0.694	0.672	0.021
42	joke	lemmatize	cvec	RandomForest	0.677	0.672	0.005
1	joke	none	tvec	GradientBoost	0.671	0.671	0.000
53	full_text	none	tvec	LogisticReg	0.671	0.668	0.003
54	joke	lemmatize	cvec	LogisticReg	0.671	0.666	0.005
34	full_text	lemmatize	cvec	MultinomialNB	0.649	0.665	0.015
12	joke	none	cvec	AdaBoost	0.670	0.661	0.009