DUNDER
MIFFLIN, INC
PAPER COMPANY

That’s What Who Said?

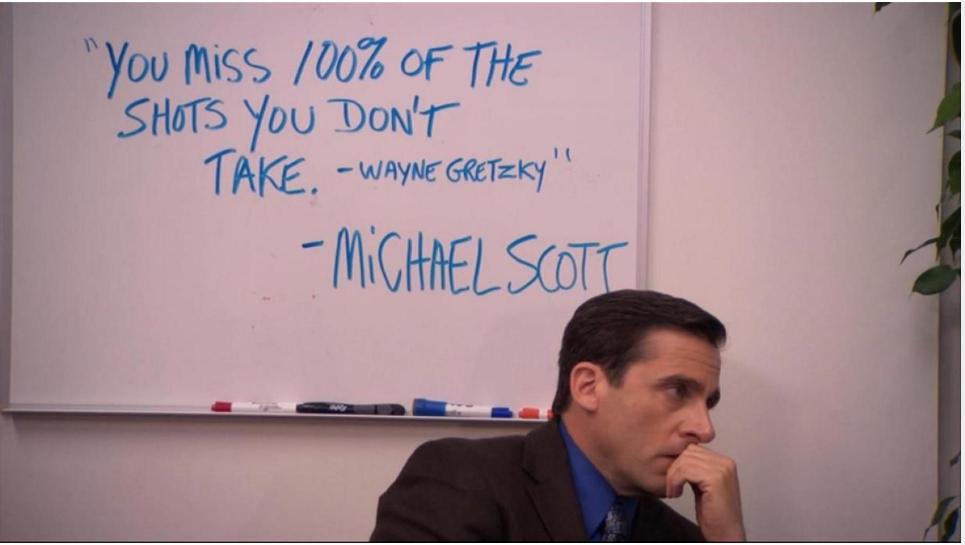
By Levi Hatch

Email: levitravishatch@gmail.com | LinkedIn: /levi-hatch | GitHub: /lthatch

The inspiration for this project- The Office TV Show

I am an Office fanatic! I love all the characters and the crazy situations they find themselves in. I would be embarrassed to say how many hours I have watched, but with that time, I feel as if I can quote just about any character from any noteworthy scene.

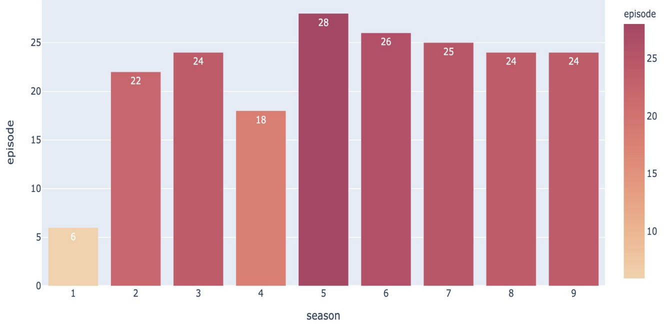
Project Goal



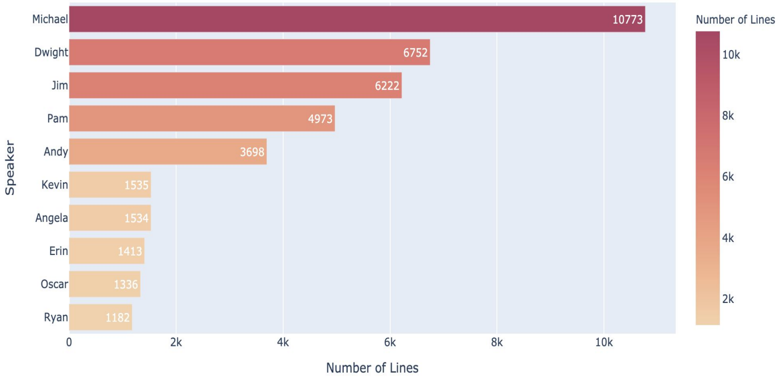
The goal of this project is to build a model that would take input text from a user and identify a character from The Office who would most likely say those words.

Exploratory Data Analysis

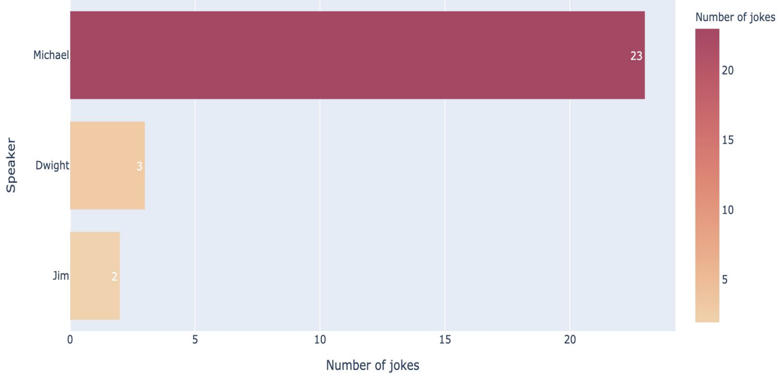
Number of Episodes per Season

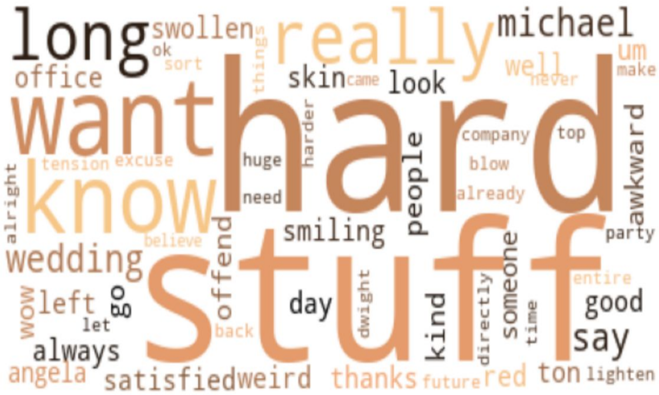


Top 10 Speakers with the Most Lines

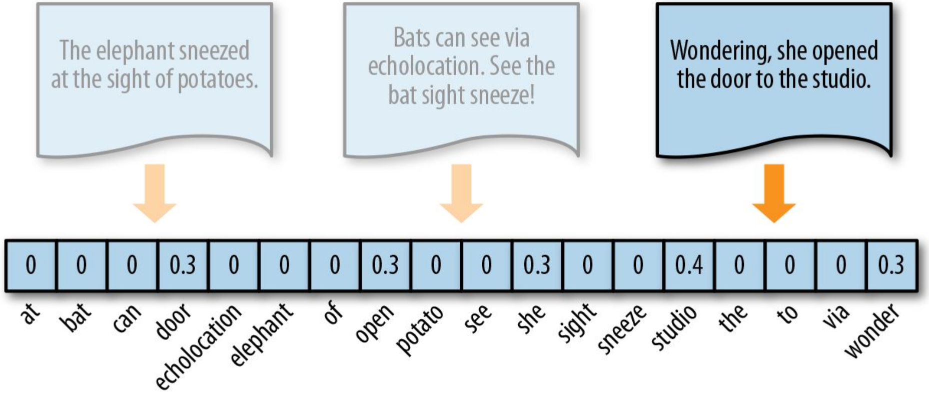


Number of 'That's What She Said' Jokes





Preparing the Data



The elephant sneezed at the sight of potatoes.

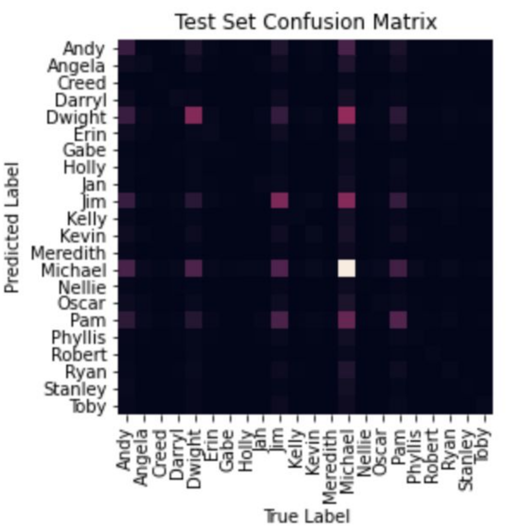
Bats can see via echolocation. See the bat sight sneeze!

Wondering, she opened the door to the studio.

at	bat	can	door	echolocation	elephant	of	open	potato	see	she	sight	sneeze	studio	the	to	via	wonder
0	0	0	0.3	0	0	0	0.3	0	0	0.3	0	0	0.4	0	0	0	0.3

- I limited the amount of classes/characters down to the top 20 speakers
- Using a TfidfVectorizor, I removed stop words and transformed the text into a weighted matrix of numbers
- Once in numeric form, I performed a train, test split and my data was ready to be fed into a model

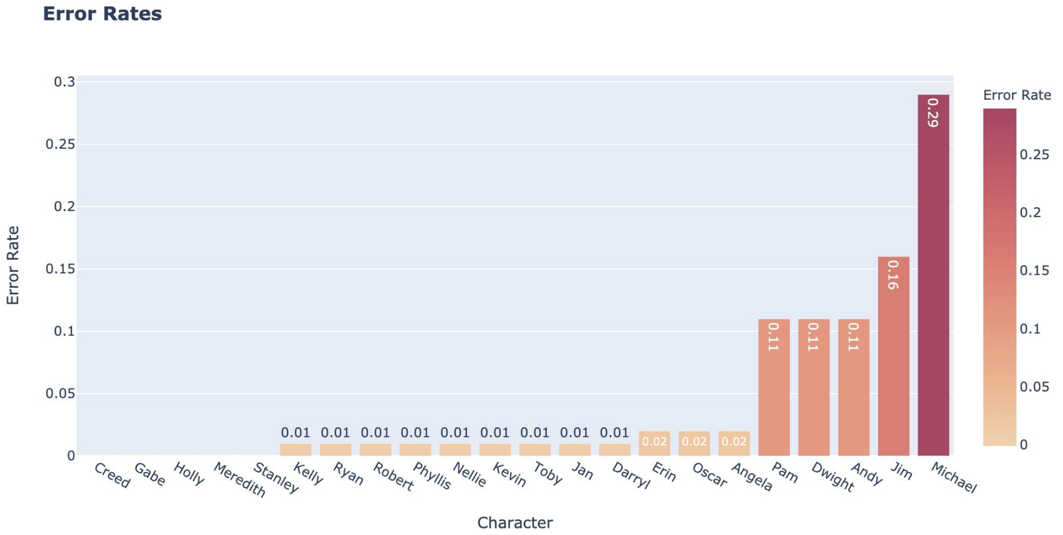
Selecting and Training a Model



Hold out data accuracy: **.24**

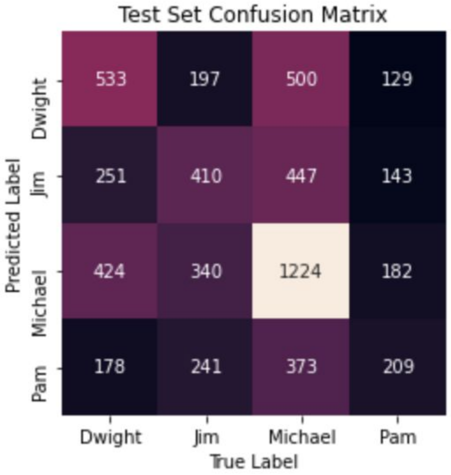
Random guessing expected accuracy: **.05**

I decided to use scikit learn’s ComplementNB model because of the imbalances in my data.



Limiting the Number of Classes

In an effort to improve my accuracy I decided to limit the number of classes the model needed to predict. I chose the top 4 speakers

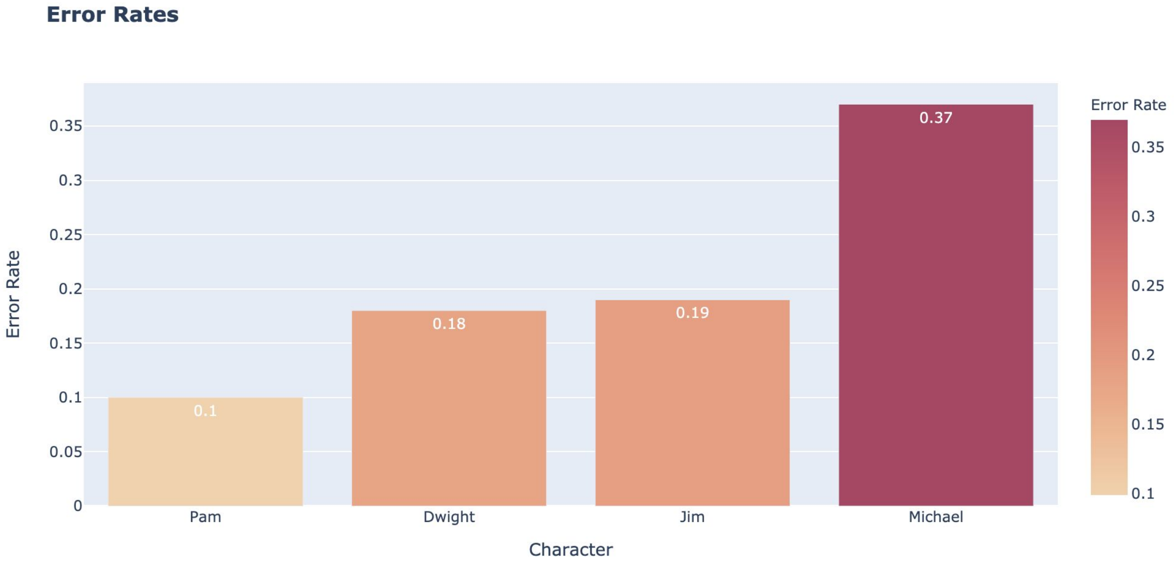


Test Set Confusion Matrix

Predicted Label \ True Label	Dwight	Jim	Michael	Pam
Dwight	533	197	500	129
Jim	251	410	447	143
Michael	424	340	1224	182
Pam	178	241	373	209

Hold out data accuracy: **.41**

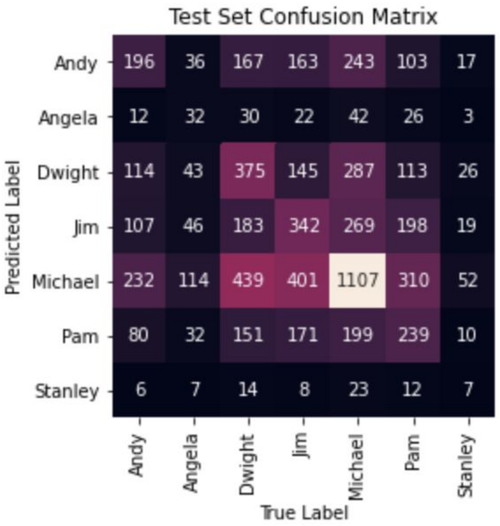
Random guessing expected accuracy: **.25**



Error Rates

Fine-tuning the model

After seeing my accuracy increase I decided to add in 3 more characters. In addition to increasing the number of classifications, I tuned the model using sklearn's Randomized SearchCV

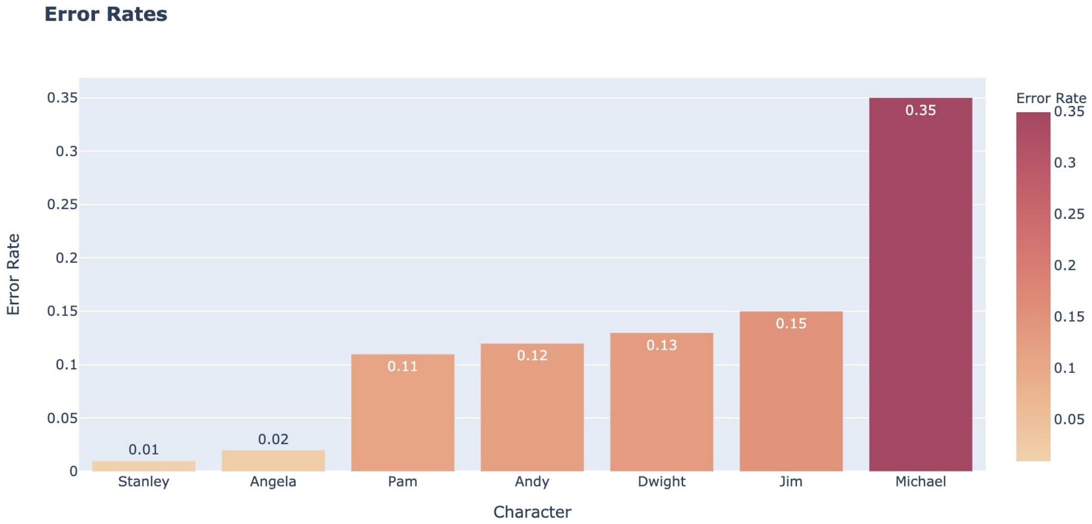


Test Set Confusion Matrix

Predicted Label \ True Label	Andy	Angela	Dwight	Jim	Michael	Pam	Stanley
Andy	196	36	167	163	243	103	17
Angela	12	32	30	22	42	26	3
Dwight	114	43	375	145	287	113	26
Jim	107	46	183	342	269	198	19
Michael	232	114	439	401	1107	310	52
Pam	80	32	151	171	199	239	10
Stanley	6	7	14	8	23	12	7

Hold out data accuracy: **.33**

Random guessing expected accuracy: **.14**



In Conclusion of this Powerpoint

As you can see, the results from the flask app were mostly as expected.

Next steps: Expanding this project into creating a chat bot in order to chat with your favorite characters!

Email: levitravishatch@gmail.com | LinkedIn: /levi-hatch | GitHub: /lthatch

Flask App