DATA SCIENCE P3 | DAN KIM | 12.21.18

THE PROBLEM

Reddit posts are all in disarray. We need to build a model that will accurately reclassify Reddit posts back to their respective subreddits.

How might we build the best classification model to determine which subreddit a given post came from?

OUR MISSION

To build an accurate classification model to distinguish whether a given post comes from the Nike or Adidas subreddit.

METHODOLOGY:

1. DATA COLLECTION - API APPROACH: NIKE AND ADIDAS

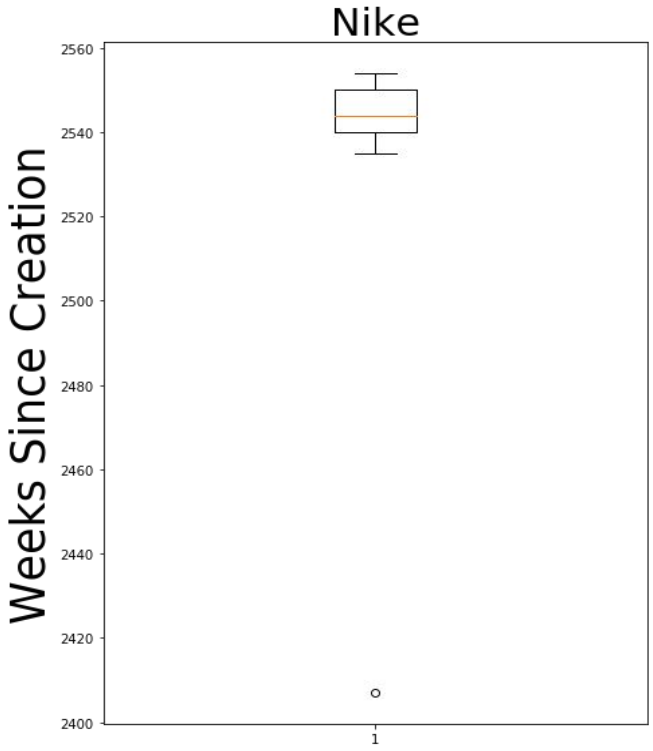
Reddit API → Json Script

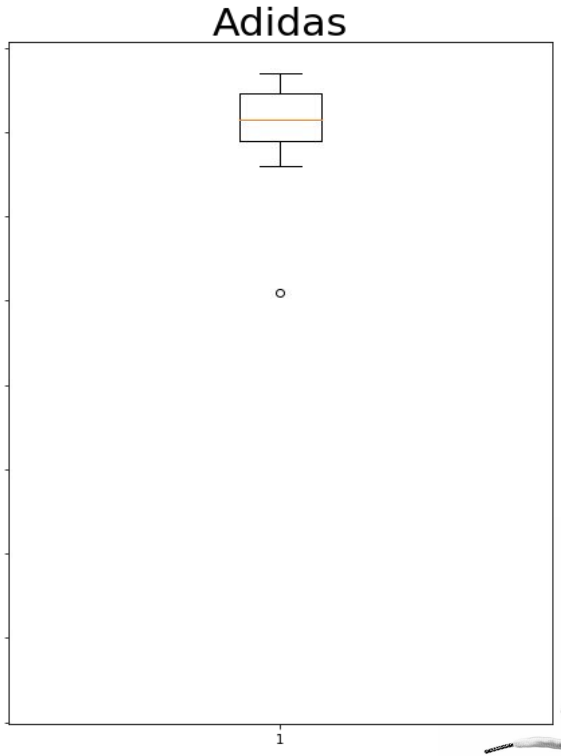
Feature Extraction → DataFrame

METHODOLOGY:
2. DATA CLEANING & EDA

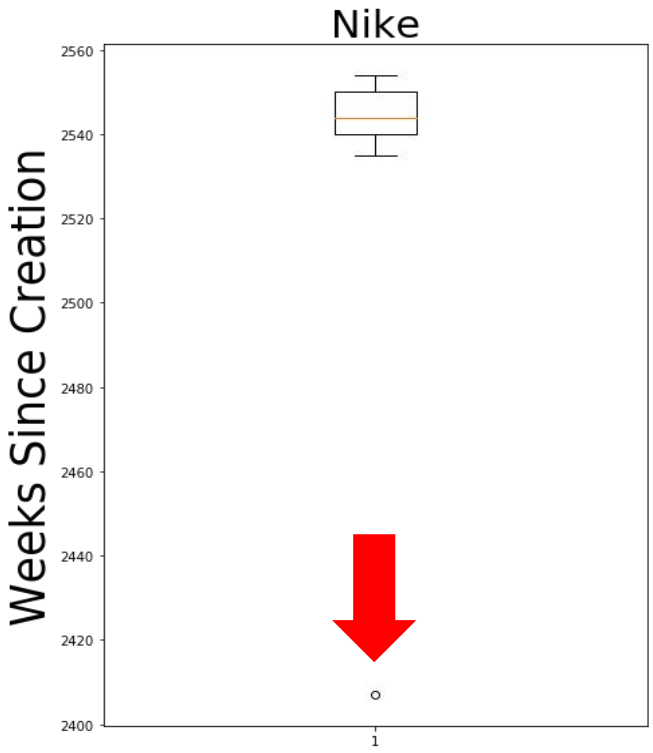
Null Values by DataFrame Column	
author	0
comments	0
created	0
score	0
subscribers	0
text	694
title	0
url	0
subreddit	0

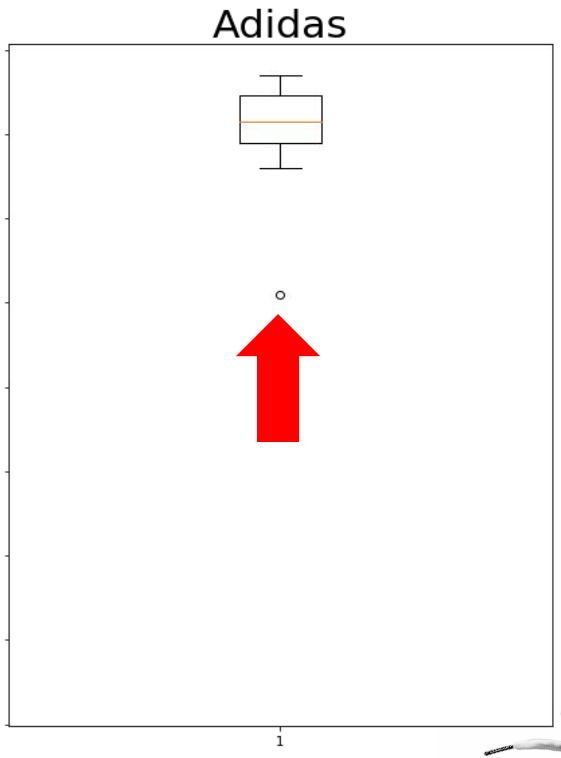
OUTLIERS EXPLAINED





OUTLIERS EXPLAINED





METHODOLOGY:
3. PREPROCESSING

Lemmatize/Stem

CountVectorizer

TfidfVectorizer

Parameters:
stop_words = english
min_df = 2
ngram_range = (1, 5)

METHODOLOGY:
4. MODEL SELECTION

Logistic Regression
Multinomial Naive Bayes
Decision Tree
Bagged Trees
Random Forests
Extra Trees

MODEL PERFORMANCE RESULTS

Best Models:

- Under CountVectorizer:
  - Logistic Regression

- Under TfidfVectorizer:
  - Random Forest

MODEL PERFORMANCE
LOGISTIC REGRESSION

TfidfVectorizer - Logistic Regression Model	Performance (Lasso)	Performance (Ridge)
Training score	0.971	0.968
Testing score	0.754	0.762

CountVectorizer - Logistic Regression Model	Performance (Lasso)	Performance (Ridge)
Training score	0.97	0.972
Testing score	0.768	0.762

Parameters:
Penalty = ‘l1’, ‘l2’
C = 50
tol = 0.0001

MODEL PERFORMANCE
RANDOM FOREST

TfidfVectorizer - Random Forest Model	Performance
Training score	0.974
Testing score	0.766

CountVectorizer - Random Forest Model	Performance
Training score	0.974
Testing score	0.754

Parameters:
n_estimators = 200
criterion = gini
max_depth = 1000

METHODOLOGY
5. MODEL TUNING

The GridSearch gave us lower train and test scores despite selecting the optimal features to use in both our models.

GridSearch Model	Logistic Regression (Lasso)	Random Forest
Parameters Tested	penalty: [l1, l2] tol: [.0001, .001, .00001] C: [1.0, 10.0, 50.0]	n_estimators: [10, 50, 100] max_depth: [50, 100, 150]
Cross-Val Score	0.787	0.773
Train Score	0.935	0.948
Test Score	0.744	0.764

METHODOLOGY
5. MODEL TUNING

The GridSearch gave us lower train and test scores despite selecting the optimal features to use in both our models.

GridSearch Model	Logistic Regression (Lasso)	Random Forest
Parameters Tested	penalty: [l1, l2] tol: [.0001, .001, .00001] C: [1.0, 10.0, 50.0]	n_estimators: [10, 50, 100] max_depth: [50, 100, 150]
Cross-Val Score	0.787	0.773
Train Score	0.935	0.948
Test Score	0.744	0.764

SIGNIFICANT INSIGHTS

Top 10 words that distinguish a given post by Subreddit
(not including ‘nike’ and ‘adidas’)

Nike	Adidas
‘jordan’	‘nmd’
‘air’	‘boost’
‘af1’	‘ultraboost’
‘like’	‘dbz’
‘swoosh’	‘pick’
‘kyrie’	‘pants’
‘lebron’	‘return’
‘kaepernick’	‘yeezy’
‘vapormax’	‘love’
‘sneakers’	‘stripe’

NOTABLE OBSTACLES

Certain phrases and words are post agnostic.

Many posts did not have text in the body of the post. This poses issues if we want to lemmatize and vectorize the text and include it in our model.

CONCLUSION & RECOMMENDATIONS

Since the Logistic Regression model under the CountVectorizer transformer had the highest accuracy, we recommend employing this model going forward.

NEXT STEPS

Include more features:

- Vectorized text body of post
- Dummied authors
- Weeks since creation
- Post score

Remove:

- Vectorized words that are in the title of the Subreddit (i.e. ‘nike’, ‘adidas’)
- Post agnostic words

Q & A