

<https://uchicago.box.com/s/vp0ciemgmvof0id5nejgnfhojfwmjquz>

# Online Community Identities:

---

AN ANALYSIS OF ADOPTION-RELATED SUBREDDITS

By Ethan Kozlowski

# Research Question

---

- How do the conversational dynamics and post sentiments within online adoption communities differ between general-purpose adoption subreddits like r/Adoption and adoptee-exclusive spaces like r/Adopted?
- Do adoptees and non adoptees differ in their sentiment towards adoption within these different spaces?

## Supplementary Questions:

- Are the topics discussed in r/Adoption and r/Adopted distinctly different from one another?
- Are the posts from adoptees distinctly different from posts not made by adoptees in r/Adopted and r/Adoption?

# Research Question

---

- How do the conversational dynamics and post sentiments within online adoption communities differ between general-purpose adoption subreddits like r/Adoption and adoptee-exclusive spaces like r/Adopted?
- Do adoptees and non adoptees differ in their sentiment towards adoption within these different spaces?

## Supplementary Questions:

- Are the topics discussed in r/Adoption and r/Adopted distinctly different from one another?
- Are the posts from adoptees distinctly different from posts not made by adoptees in r/Adopted and r/Adoption?

# Research Question

---

- How do the conversational dynamics and post sentiments within online adoption communities differ between general-purpose adoption subreddits like r/Adoption and adoptee-exclusive spaces like r/Adopted?
- Do adoptees and non adoptees differ in their sentiment towards adoption within these different spaces?

## Supplementary Questions:

- Are the topics discussed in r/Adoption and r/Adopted distinctly different from one another?
- Are the posts from adoptees distinctly different from posts not made by adoptees in r/Adopted and r/Adoption?

# Quick Data Overview

---

- Data collection method:
  - scraping [old.reddit.com](https://old.reddit.com)
  - (non dynamic) and reddit archive: [the-eye.eu](https://the-eye.eu)
- Reddits of interest:
  - [r/Adopted](https://www.reddit.com/r/Adopted)
  - [r/Adoption](https://www.reddit.com/r/Adoption)
    - these are the single two largest subreddits related to topic of adoption

# ML: DNN and biLSTM NN

---

Along with other more traditional ML classifiers (decision trees and logistic regression)

I attempted to classify reddit posts into those made by adoptees and those not made by adoptees.

The neural networks performed well but not incredibly so.

DNN

152/152 [=====] - 0s 1ms/step				
	precision	recall	f1-score	support
0	0.68	0.76	0.72	2010
1	0.82	0.75	0.78	2853
accuracy			0.76	4863
macro avg	0.75	0.76	0.75	4863
weighted avg	0.76	0.76	0.76	4863

biLSTM NN

152/152 [=====] - 9s 57ms/step				
	precision	recall	f1-score	support
0	0.63	0.87	0.73	2010
1	0.88	0.64	0.74	2853
accuracy			0.74	4863
macro avg	0.75	0.76	0.74	4863
weighted avg	0.78	0.74	0.74	4863

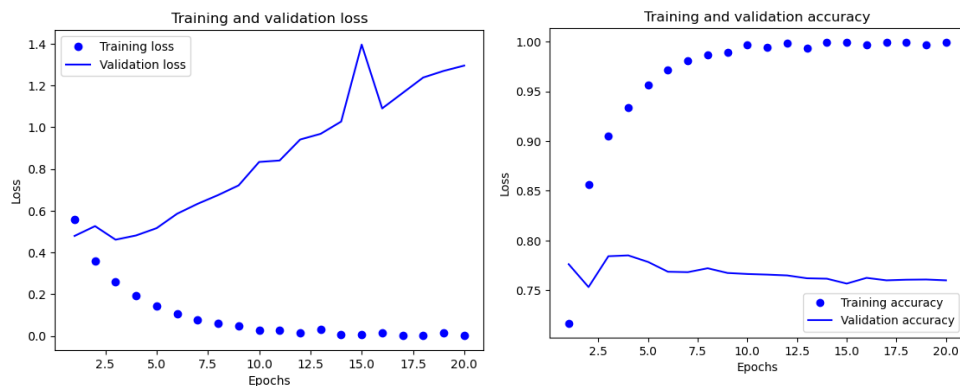
# ML: DNN and biLSTM NN

Along with other more traditional ML classifiers (decision trees and logistic regression)

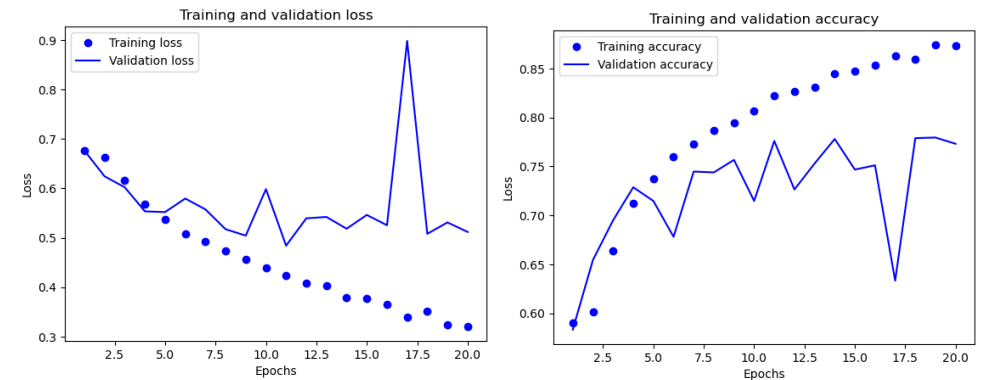
I attempted to classify reddit posts into those made by adoptees and those not made by adoptees.

The neural networks performed well but not incredibly so.

DNN



biLSTM NN



# Word Embeddings (word2vec)

```
1 cosine_sim1 = find_cosine_sim(r_adopted_model, r_adoption_model, "child", "save")
2 cosine_sim2 = find_cosine_sim(adoptee_model, non_adoptee_model, "child", "save")
3 cosine_sim3 = find_cosine_sim(r_adopted_adoptees_model, r_adoption_adoptees_model, "child", "save")
```

✓ 0.0s

Python

```
Cosine similarity between child and save for model1: 0.7254377603530884
Cosine similarity between child and save for model2: 0.06619530916213989
Cosine similarity between child and save for model1: 0.34584587812423706
Cosine similarity between child and save for model2: 0.11032998561859131
Cosine similarity between child and save for model1: 0.9983972311019897
Cosine similarity between child and save for model2: 0.3522150218486786
```

```
1 cosine_sim1 = find_cosine_sim(r_adopted_model, r_adoption_model, "trauma", "adoption")
2 cosine_sim2 = find_cosine_sim(adoptee_model, non_adoptee_model, "trauma", "adoption")
3 cosine_sim3 = find_cosine_sim(r_adopted_adoptees_model, r_adoption_adoptees_model, "trauma", "adoption")
```

✓ 0.0s

```
Cosine similarity between trauma and adoption for model1: 0.6991931796073914
Cosine similarity between trauma and adoption for model2: 0.19732362031936646
Cosine similarity between trauma and adoption for model1: 0.1964179277420044
Cosine similarity between trauma and adoption for model2: 0.26756197214126587
Cosine similarity between trauma and adoption for model1: 0.9992290139198303
Cosine similarity between trauma and adoption for model2: 0.22085809707641602
```



# Word Embeddings (word2vec)

```
1 cosine_sim1 = find_cosine_sim(r_adopted_model, r_adoption_model, "child", "save")
2 cosine_sim2 = find_cosine_sim(adoptee_model, non_adoptee_model, "child", "save")
3 cosine_sim3 = find_cosine_sim(r_adopted_adoptees_model, r_adoption_adoptees_model, "child", "save")
```

✓ 0.0s

Python

```
Cosine similarity between child and save for model1: 0.7254377603530884
Cosine similarity between child and save for model2: 0.06619530916213989
Cosine similarity between child and save for model1: 0.34584587812423706
Cosine similarity between child and save for model2: 0.11032998561859131
Cosine similarity between child and save for model1: 0.9983972311019897
Cosine similarity between child and save for model2: 0.3522150218486786
```

```
1 cosine_sim1 = find_cosine_sim(r_adopted_model, r_adoption_model, "trauma", "adoption")
2 cosine_sim2 = find_cosine_sim(adoptee_model, non_adoptee_model, "trauma", "adoption")
3 cosine_sim3 = find_cosine_sim(r_adopted_adoptees_model, r_adoption_adoptees_model, "trauma", "adoption")
```

✓ 0.0s

```
Cosine similarity between trauma and adoption for model1: 0.6991931796073914
Cosine similarity between trauma and adoption for model2: 0.19732362031936646
Cosine similarity between trauma and adoption for model1: 0.1964179277420044
Cosine similarity between trauma and adoption for model2: 0.26756197214126587
Cosine similarity between trauma and adoption for model1: 0.9992290139198303
Cosine similarity between trauma and adoption for model2: 0.22085809707641602
```

# Word Embeddings (word2vec)

```
1 get_similarities(keywords, r_adopted_model, r_adoption_model)
```

```
{'bond': -0.15175579488277435,  
'seek': -0.2268764227628708,  
'grateful': -0.19217661023139954,  
'transracial': -0.05062051862478256  
'industry': 0.17956453561782837,  
'approval': 0.03199252858757973,  
'adoption': -0.1373712569475174,  
'adoptee': -0.16728758811950684,  
'adopt': -0.2455664873123169,  
'adopted': 0.0359882228076458,  
'trauma': -0.05348401889204979,  
'good': -0.31631654500961304,  
'bad': -0.09666292369365692,  
'child': -0.1786220520734787,  
'save': 0.025004249066114426,  
'love': -0.2885979413986206,  
'hate': -0.07111009210348129,  
'happy': -0.15548314154148102,  
'sad': -0.12258635461330414,  
'angry': -0.05174421891570091}
```

```
1 get_similarities(keywords, adoptee_model, non_adoptee_model)
```

```
{'bond': 0.38437241315841675,  
'seek': 0.16076724231243134,  
'grateful': 0.45100778341293335,  
'transracial': 0.061191461980342865,  
'industry': 0.27541348338127136,  
'approval': 0.6695536971092224,  
'adoption': 0.5073298811912537,  
'adoptee': -0.056700676679611206,  
'adopt': 0.27125269174575806,  
'adopted': 0.6001583337783813,  
'trauma': 0.34761086106300354,  
'good': 0.16712762415409088,  
'bad': 0.303008496761322,  
'child': -0.042299337685108185,  
'save': 0.12295631319284439,  
'love': 0.2635558545589447,  
'hate': 0.2835824489593506,  
'happy': 0.26672235131263733,  
'sad': 0.37462371587753296,  
'angry': 0.2550153434276581}
```

```
1 get_similarities(keywords, r_adoption_adoptees_model, r_adopted_adoptees_model)
```

```
{'bond': 0.3173474967479706,  
'seek': 0.30030035972595215,  
'grateful': 0.3994129002094269,  
'transracial': 0.21451137959957123,  
'industry': 0.2947518527507782,  
'approval': 0.699675977230072,  
'adoption': 0.09742536395788193,  
'adoptee': 0.02635442651808262,  
'adopt': 0.2384576052427292,  
'adopted': 0.6952021718025208,  
'trauma': 0.2110527604818344,  
'good': 0.27848976850509644,  
'bad': 0.3218405544757843,  
'child': 0.20079012215137482,  
'save': 0.40664851665496826,  
'love': 0.2910844087600708,  
'hate': 0.4425756633281708,  
'happy': 0.33523717522621155,  
'sad': 0.4187967777252197,  
'angry': 0.4098033607006073}
```

# Word Embeddings (word2vec)

```
1 get_similarities(keywords, r_adopted_model, r_adoption_model)
```

```
{'bond': -0.15175579488277435,  
'seek': -0.2268764227628708,  
'grateful': -0.19217661023139954,  
'transracial': -0.05062051862478256  
'industry': 0.17956453561782837,  
'approval': 0.03199252858757973,  
'adoption': -0.1373712569475174,  
'adoptee': -0.16728758811950684,  
'adopt': -0.2455664873123169,  
'adopted': 0.0359882228076458,  
'trauma': -0.05348401889204979,  
'good': -0.31631654500961304,  
'bad': -0.09666292369365692,  
'child': -0.1786220520734787,  
'save': 0.025004249066114426,  
'love': -0.2885979413986206,  
'hate': -0.07111009210348129,  
'happy': -0.15548314154148102,  
'sad': -0.12258635461330414,  
'angry': -0.05174421891570091}
```

```
1 get_similarities(keywords, adoptee_model, non_adoptee_model)
```

```
{'bond': 0.38437241315841675,  
'seek': 0.16076724231243134,  
'grateful': 0.45100778341293335,  
'transracial': 0.061191461980342865,  
'industry': 0.27541348338127136,  
'approval': 0.6695536971092224,  
'adoption': 0.5073298811912537,  
'adoptee': -0.056700676679611206,  
'adopt': 0.27125269174575806,  
'adopted': 0.6001583337783813,  
'trauma': 0.34761086106300354,  
'good': 0.16712762415409088,  
'bad': 0.303008496761322,  
'child': -0.042299337685108185,  
'save': 0.12295631319284439,  
'love': 0.2635558545589447,  
'hate': 0.2835824489593506,  
'happy': 0.26672235131263733,  
'sad': 0.37462371587753296,  
'angry': 0.2550153434276581}
```

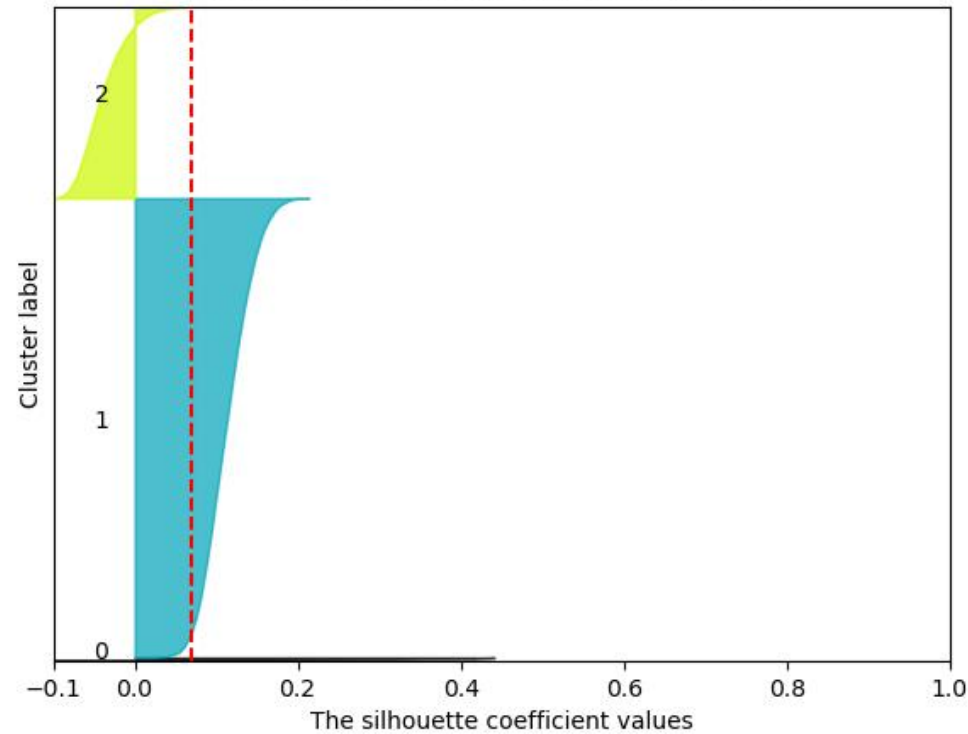
```
1 get_similarities(keywords, r_adoption_adoptees_model, r_adopted_adoptees_model)
```

```
{'bond': 0.3173474967479706,  
'seek': 0.30030035972595215,  
'grateful': 0.3994129002094269,  
'transracial': 0.21451137959957123,  
'industry': 0.2947518527507782,  
'approval': 0.699675977230072,  
'adoption': 0.09742536395788193,  
'adoptee': 0.02635442651808262,  
'adopt': 0.2384576052427292,  
'adopted': 0.6952021718025208,  
'trauma': 0.2110527604818344,  
'good': 0.27848976850509644,  
'bad': 0.3218405544757843,  
'child': 0.20079012215137482,  
'save': 0.40664851665496826,  
'love': 0.2910844087600708,  
'hate': 0.4425756633281708,  
'happy': 0.33523717522621155,  
'sad': 0.4187967777252197,  
'angry': 0.4098033607006073}
```

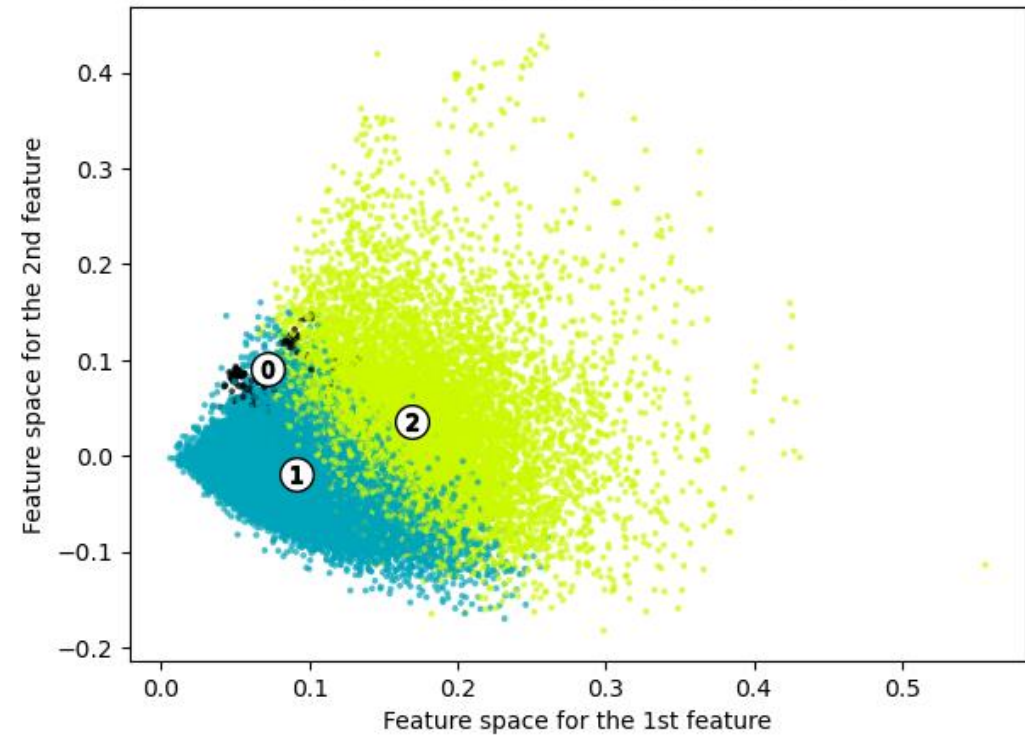
# Clustering and Topic Modeling

## Silhouette analysis for KMeans clustering on sample data with $n\_clusters = 3$

The silhouette plot for the various clusters.

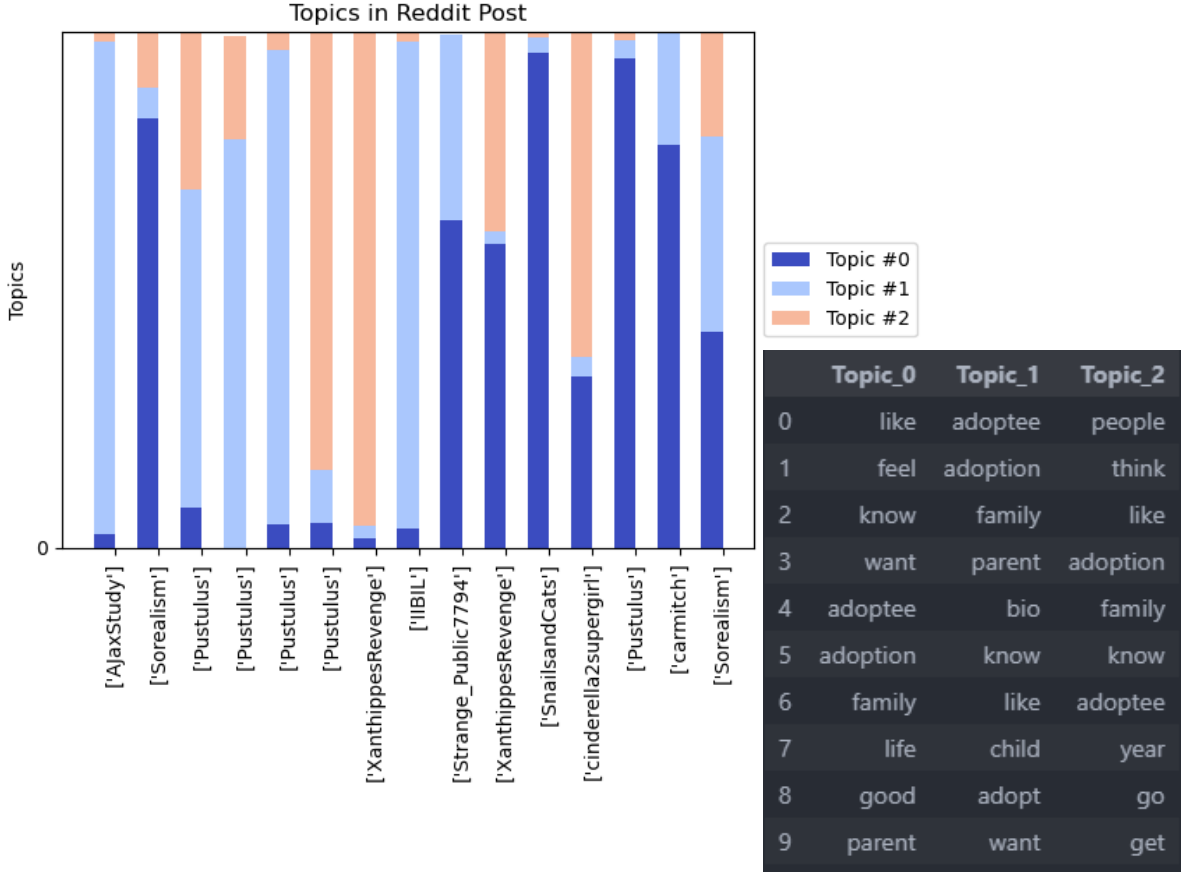


The visualization of the clustered data.

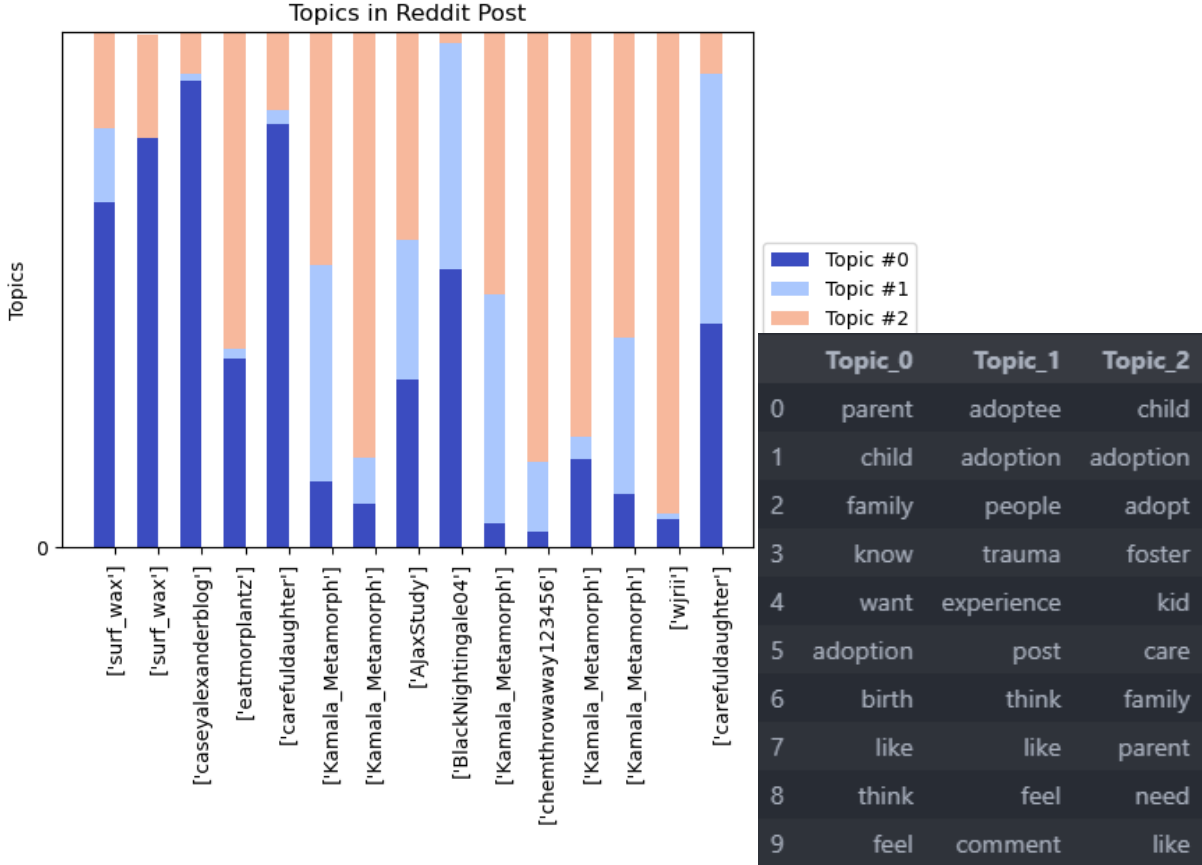


The average silhouette\_score is : 0.069

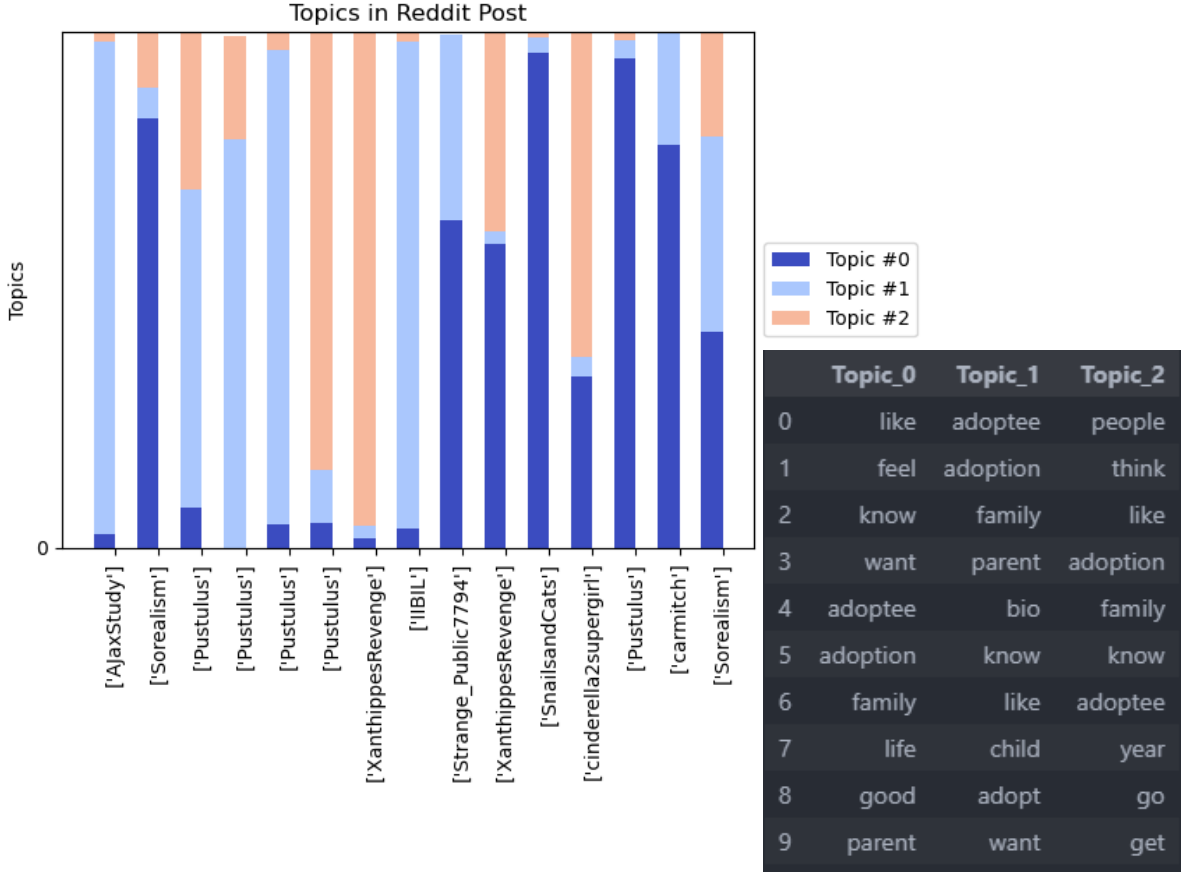
r/Adopted



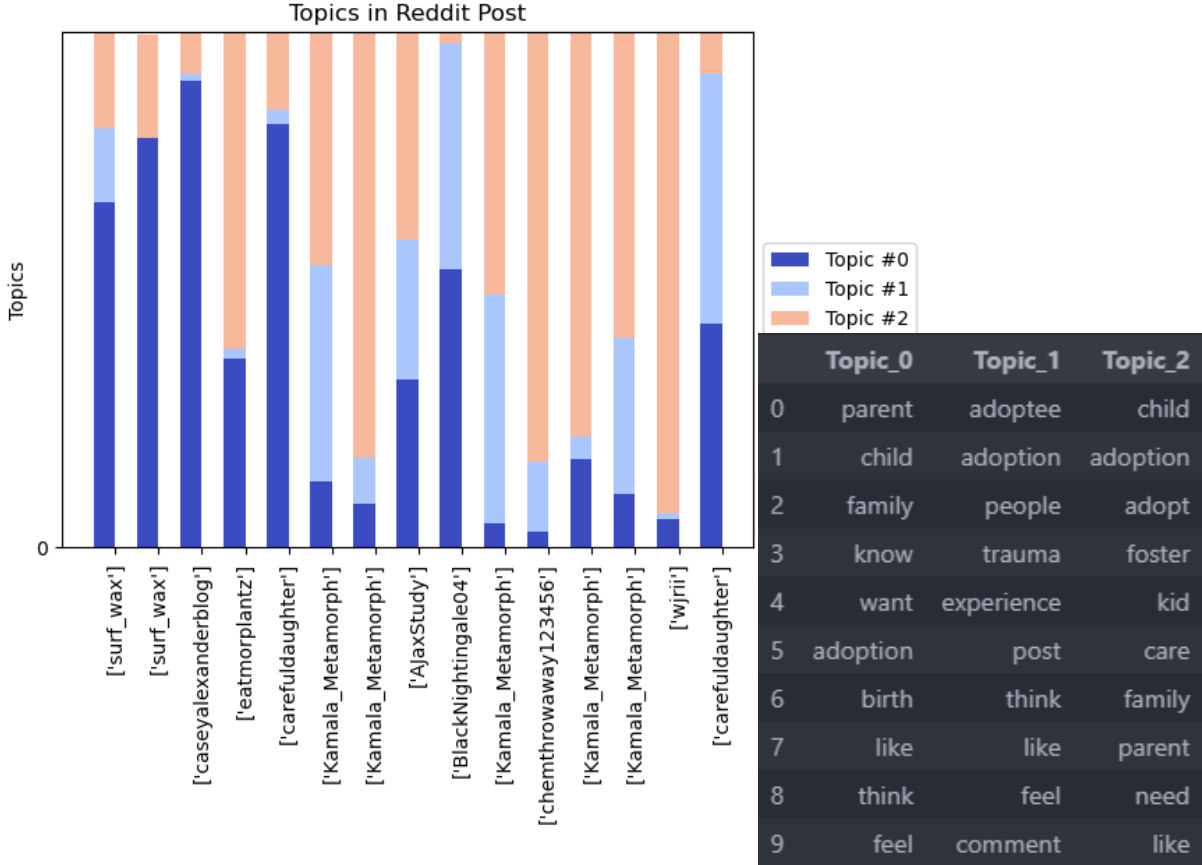
r/Adoption



r/Adopted



r/Adoption



# Sentiment Analysis:

---

No data was prelabeled for sentiment: therefore used VADER to rate the compound sentiment score for all texts.

- VADER is specifically made for social media data and generally performs well (F1: 0.96)

Adoptee Sentiment: 0.39165695604488315

Non-adoptee sentiment: 0.4558886840314298

Running a t-test for sample means we see that means are statistically significantly different from one another at the  $\alpha = .05$  significance level



# Sentiment Analysis: 2 way ANOVA

---

	sum_sq	Degrees of Freedom	F-stat	PR(>F)
is_adoptee	13.540146	1.0	36.756788	1.350233e-09
subreddit	121.939295	1.0	331.022779	1.144487e-73
is_adoptee * subreddit	0.000014	1.0	0.000038	9.950871e-01
Residual	14685.122317	39865.0	NAN	NAN



# Sentiment Analysis: 2 way ANOVA

---

	sum_sq	Degrees of Freedom	F-stat	PR(>F)
is_adoptee	13.540146	1.0	36.756788	1.350233e-09
subreddit	121.939295	1.0	331.022779	1.144487e-73
is_adoptee * subreddit	0.000014	1.0	0.000038	9.950871e-01
Residual	14685.122317	39865.0	NAN	NAN

# Sentiment Analysis: 2 way ANOVA

---

	sum_sq	Degrees of Freedom	F-stat	PR(>F)
is_adoptee	13.540146	1.0	36.756788	1.350233e-09
subreddit	121.939295	1.0	331.022779	1.144487e-73
is_adoptee * subreddit	0.000014	1.0	0.000038	9.950871e-01
Residual	14685.122317	39865.0	NAN	NAN

# Conclusions

---

We see there are measurable differences between the posts written by adoptees and written by non adoptees.

With ML we see that we can predict whether or not someone is an adoptee based off of their post text alone.

Through Topic Modeling we see that r/Adopted has this topic of shared bonding and empathy where r/Adoption has an exclusive topic related to prospective adoptive parents

Through Embedding we see the relationship between specific words varies by subreddit and by adoptee status

We also see that status as adoptee and non adoptee and subreddit are both independently significant predictors of a post sentiment.

# Conclusions

---

We see there are measurable differences between the posts written by adoptees and written by non adoptees.

With ML we see that we can predict whether or not someone is an adoptee based off of their post text alone.

Through Topic Modeling we see that r/Adopted has this topic of shared bonding and empathy where r/Adoption has an exclusive topic related to prospective adoptive parents

Through Embedding we see the relationship between specific words varies by subreddit and by adoptee status

We also see that status as adoptee and non adoptee and subreddit are both independently significant predictors of a post sentiment.