

Capstone Project - Topic Modeling with /r/PersonalFinance

Joshua Kim

Springboard - Data Science Career Track

Table of Contents

1. Cover Page
2. Table of Contents
3. Introduction
4. Project Goal
5. Data Acquisition
6. Data Dictionary
7. Data Wrangling
8. Most Common Topics
9. Most Common Tokens (Words)
10. Feature Engineering
11. Latent Dirichlet Allocation - Topic Modeling
12. Labeling the Topics and Posts
13. Evaluating the Topics
14. Baseline
15. Linear SVC
16. Logistic Regression
17. Model Results
18. Conclusion - Recommendations
19. Conclusion - Limitations
20. Appendix

Introduction - The Problems



❖ The Problem:

- /r/personalfinance is a subreddit where many people are able to ask for help regarding their financial situations (examples: debt, budgeting, insurance, student loans).
- One of the important features is tagging each post with a flair (topic) in order to categorize and organize different posts by subject. These are usually done manually when you first create the post but you can also do it afterwards.
- Problem #1: Some people do not tag their post with a flair, resulting in a default 'Other' topic. 'Other' is a category for posts whose topics do not fit into one of the 12 main topics. This can result in confusion since some posts can qualify for one of the main topics but are defaulted into 'Other'. Can we build a model that categorizes posts automatically?
- Problem #2: How do we know that the 12 main topics are the best set of topics?
- Problem #3: If a post can fit under multiple topics, how do we decide which topic to use?

Project Goal

❖ The Goal:

- To create a topic model that produces a good variety of topics for /r/PersonalFinance.
- To evaluate how well the new topics fit the posts.

❖ Clients:

- 1) /r/PersonalFinance users: This will help users who don't know which topic to choose as well as those who forget to pick one.
- 2) /r/PersonalFinance moderators: This will result in fewer 'Other' posts, which will create better organization for posts. This means that more people can easily search for older and similar questions, which will reduce the number of frequently asked questions on the subreddit.



Data Acquisition

- ❖ The data was extracted from **pushshift.io** **Reddit API**, using sqlite to store the data.
- ❖ The data was scraped using the request module's built-in JSON decoder.
- ❖ Many pulls were done to extract the data; each pull obtained 500 posts from /r/personalfinance.
- ❖ The **official Reddit API** (PRAW) was used to extract information that is not provided through the pushshift.io Reddit API.
- ❖ Total of 25,000 rows of data and 7 columns.
- ❖ Data was extracted between October 10th and August 10th.

```
"author": "Rohto_Oner",
"author_flair_css_class": null,
"author_flair_richtext": [],
"author_flair_text": null,
"author_flair_type": "text",
"can_mod_post": false,
"contest_mode": false,
"created_utc": 1535255721,
"domain": "self.personalfinance",
"full_link": "https://www.reddit.com/r/p",
"id": "9acuma",
"is_crosspostable": true,
"is_meta": false,
"is_original_content": false,
"is_reddit_media_domain": false,
"is_self": true,
"is_video": false,
```

Data Dictionary

1. Title: Title of each post
2. Date: Date of post submission
3. Time: Time of post submission
4. Upvotes: Number of upvotes (popularity metric)
5. Submission ID: An alphanumeric ID automatically assigned to each Reddit post
6. Flair: The topic (or tag) assigned to each post
7. Self-text: The text information within each post

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23182 entries, 0 to 23181
Data columns (total 7 columns):
title      23182 non-null object
date       23182 non-null object
time       23182 non-null object
upvotes    23182 non-null int64
id         23182 non-null object
topic      23182 non-null object
self_text  23182 non-null object
dtypes: int64(1), object(6)
memory usage: 1.2+ MB
```

	title	date	time	upvotes	id	topic	self_text
0	Ways to make extra side money?	2018-09-19	12:57 PM	1	9h6whn	unknown	
1	(Year UPDATE) Legally blind, going homeless, h...	2018-09-19	12:56 AM	16	9h29g7	Other	
2	19, being kicked out	2018-09-19	12:55 PM	2	9h6vyv	Other	So i just found out last night the home ive be...
3	Online Savings Account?	2018-09-19	12:54 PM	1	9h6vs4	Saving	Hello! Looking for recommendations for an onli...
4	Tools for Managing Incomes and Expenses	2018-09-19	12:52 PM	0	9h6v48	Other	

Data Wrangling

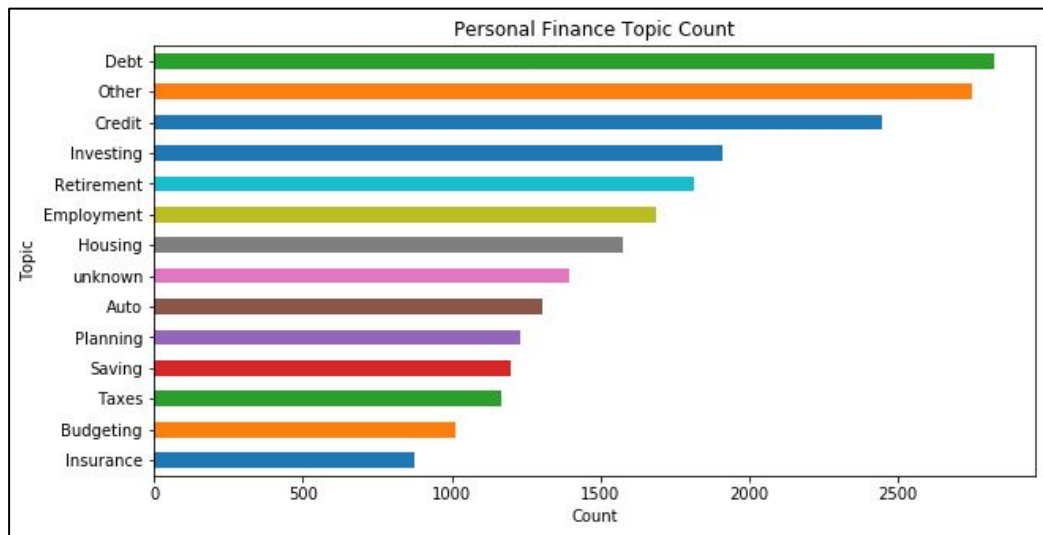
- ❖ Create a new text column by concatenating title and self-text for each post.
- ❖ Convert dates and times into datetime format.
- ❖ Replace missing flairs with 'unknown'.
- ❖ Text Pre-processing:
 - Lowercase the words.
 - Remove numbers/digits.
 - Remove punctuation.
 - Strip whitespace.
 - Remove stopwords (common vocab in the English dictionary).
 - Remove noise.
- ❖ Check for missing text after pre-processing.
- ❖ Set the word limit for each row's text to 150 words.
- ❖ Tokenize the words (turn them into token objects).
- ❖ Lemmatize the words (ex. Studying -> Study).

	text	clean_text
0	Ways to make extra side money?	ways make extra side money
1	(Year UPDATE) Legally blind, going homeless, h...	year update legally blind going homeless one j...
2	19, being kicked out So i just found out last ...	kicked found last night home ive staying going...
3	Online Savings Account? Hello! Looking for rec...	online savings account hello looking recommend...
4	Tools for Managing Incomes and Expenses	tools managing incomes expenses

lemmatized_text
[way, make, extra, side, money]
[year, update, legally, blind, go, homeless, o...
[kick, find, last, night, home, have, stay, go...
[online, saving, account, hello, look, recomme...
[tool, manage, income, expense]

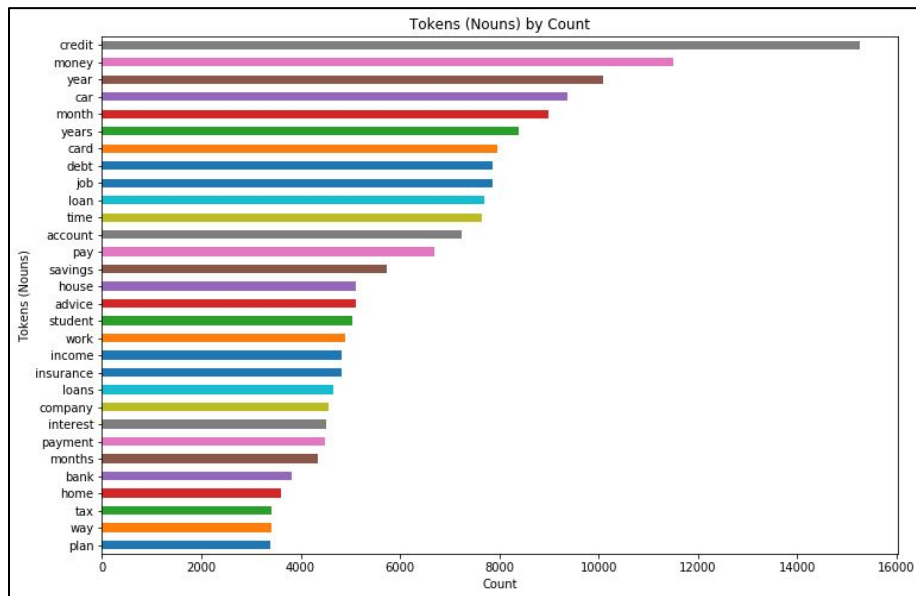
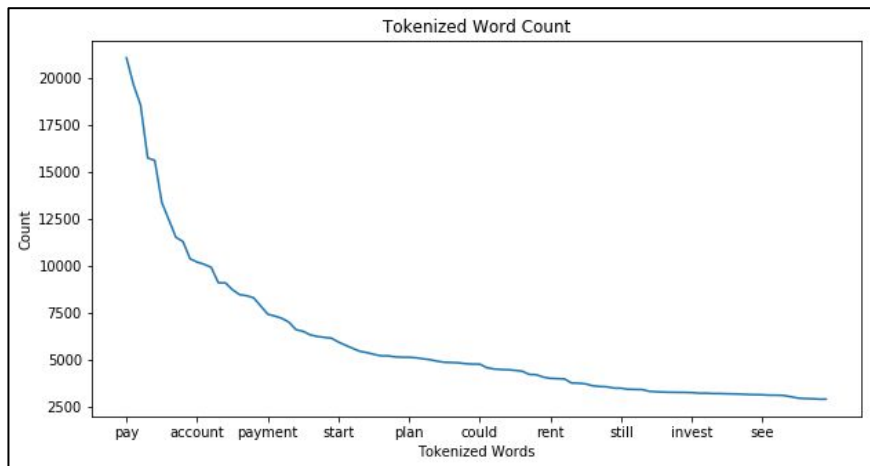
Most Common Topics

- ❖ Debt is the most common topic in the personal finance subreddit, followed by Other and Credit. This indicates that debt is a major concern for many of the redditors (users) and they make submissions in order to seek advice.
- ❖ Other includes posts that either haven't been given a topic or didn't match the main topics. With more data, it could be possible that Other is the most frequent topic instead of Debt.
- ❖ Investment and Retirement are similar in their counts while also being similar in their functions.



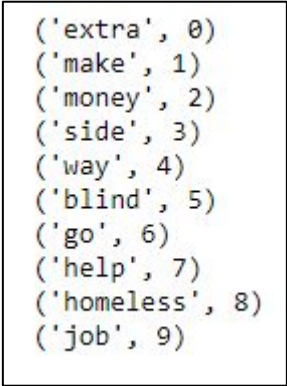
Most Common Tokens (Words)

- ❖ The word 'pay' has been used over 20,000 times in the corpus of text.
- ❖ After the first 10 tokens, the frequency that a word is used decreases very quickly.
- ❖ The words 'account' and 'payment' are also fairly common and transferable among many different topics which is why it is logical for them to have high counts. Towards the end of the graph, we can start to see more topic-specific words such as 'rent' and 'invest'.

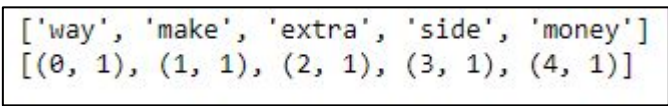


Feature Engineering

- ❖ Before using the Latent Dirichlet Allocation model to create topics, we need to first transform the data so that it appropriately fits into the model.
- ❖ Create a dictionary using all the text from each row (this maps each word to a numeric value).
- ❖ Create a bag-of-words using the dictionary to count the number of times a word appears in a row



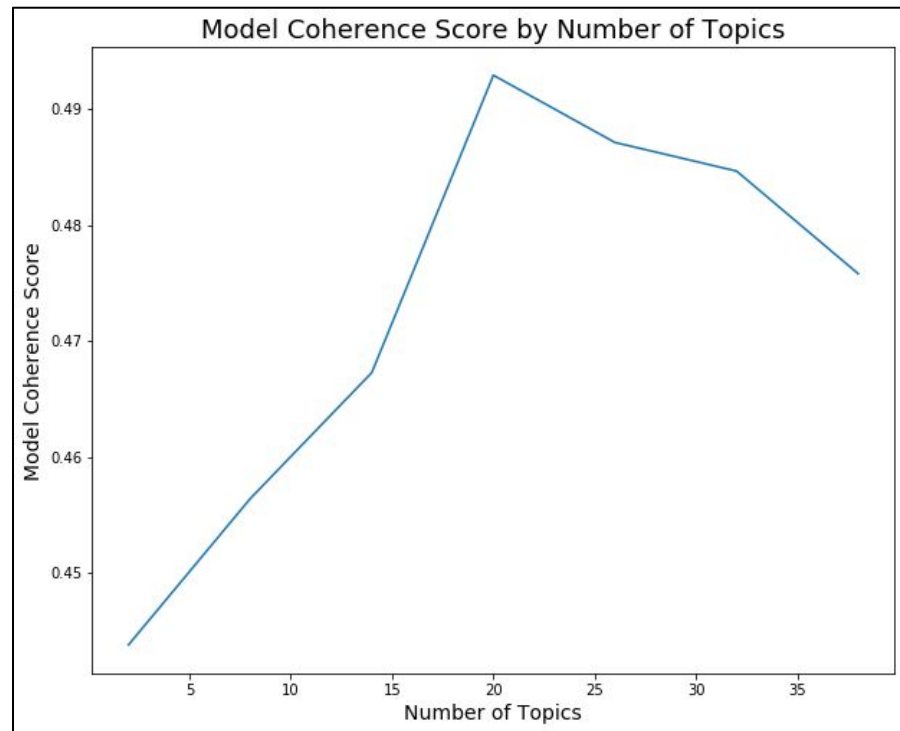
```
('extra', 0)
('make', 1)
('money', 2)
('side', 3)
('way', 4)
('blind', 5)
('go', 6)
('help', 7)
('homeless', 8)
('job', 9)
```



```
['way', 'make', 'extra', 'side', 'money']
[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1)]
```

Latent Dirichlet Allocation - Topic Modeling

- ❖ A corpus (large set of different texts) is represented in matrix form with documents as the rows and the words as the columns.
- ❖ The LDA algorithm creates 2 smaller matrices: mapping topics by words and documents by topics.
- ❖ The most important parameters for hyper-parameter tuning are the number of topics (K) and alpha (α).
- ❖ Coherence (how much a text makes sense given the context of the words) is used to evaluate how “good” the topics are at representing the corpus.
- ❖ Since K=20 provides the highest Coherence Score (0.4929), we will be creating 20 different topics.



Labeling the Topics and Posts

- ❖ Each topic is represented by a number and contains keywords with their respective weights.
- ❖ By doing an eye-test, discernable topics were named by their most common keywords (ex. 7 - 'Auto')
- ❖ Ambiguous topics were labeled 'Other' (ex. 5 - 'Other').
- ❖ Each document is represented by a few topics and their respective weights.
- ❖ We label each document by the topic with the most weight. In the example below, we would label the text with topic 13 (which would be Employment).

```
(5,
 '0.037*"give" + 0.024*"find" + 0.021*"edit" + 0.019*"thing" + 0.018*"guy" + '
 '0.015*"number" + 0.015*"person" + 0.015*"people" + 0.014*"day" + '
 '0.014*"talk"'),
(6,
 '0.278*"credit" + 0.185*"card" + 0.059*"score" + 0.037*"balance" + '
 '0.017*"apply" + 0.017*"limit" + 0.016*"build" + 0.012*"history" + '
 '0.011*"line" + 0.011*"purchase"'),
(7,
 '0.193*"car" + 0.041*"buy" + 0.024*"vehicle" + 0.018*"finance" + '
 '0.018*"drive" + 0.017*"lease" + 0.017*"payment" + 0.015*"repair" + '
 '0.014*"mile" + 0.013*"worth"'),
(8,
 '0.124*"tax" + 0.066*"ira" + 0.050*"roth" + 0.048*"income" + 0.044*"year" + '
 '0.034*"retirement" + 0.030*"contribute" + 0.030*"contribution" + '
 '0.029*"employer" + 0.023*"match"'),
(9,
 '0.066*"question" + 0.033*"make" + 0.031*"personal" + 0.027*"finance" + '
 '0.025*"post" + 0.023*"business" + 0.020*"read" + 0.016*"advice" + '
 '0.015*"understand" + 0.014*"answer"'),
```

Sample Document: Please help me review my budget . . underpaid tech worker in SF. Thanks

Score: 0.09393939393939396

Topic 13: 0.085*"company" + 0.056*"offer" + 0.041*"salary" + 0.034*"job" + 0.026*"position" + 0.025*"current" + 0.024*"work" ear"

Score: 0.06363636363636366

Topic 4: 0.140*"work" + 0.118*"job" + 0.073*"pay" + 0.069*"week" + 0.068*"time" + 0.042*"hour" + 0.034*"day" + 0.030*"make"

Score: 0.06363636363636366

Topic 14: 0.141*"month" + 0.051*"expense" + 0.040*"spend" + 0.036*"budget" + 0.033*"monthly" + 0.029*"income" + 0.021*"rent"

Evaluating the Topics

- ❖ To evaluate how well the topics behave, we performed a classification task and predicted the topics for each row.
- ❖ We intentionally made a bad model assumption: “Most labels accurately represent the post’s subject.” This is to later verify whether the labels work well.
- ❖ Feature Selection: The clean_text column is the only feature. The topic column is the response variable.
- ❖ Pipeline: CountVectorizer transforms the text into a bag-of-words. TF-IDF Vectorizer transforms the word frequencies into a probability that reflects the relative importance of a word in the document.

	clean_text	new_topic
0	ways make extra side money	planning
1	year update legally blind going homeless one j...	housing
2	kicked found last night home ive staying going...	employment
3	online savings account hello looking recommend...	bank account
4	tools managing incomes expenses	budgeting

```
['way', 'make', 'extra', 'side', 'money']  
[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1)]  
[(0, 0.5392389089986213), (1, 0.228596437753383), (2, 0.23277485092437628), (3, 0.6866754165273157), (4, 0.3622951956383253)]
```

Baseline

- ❖ A simple baseline was created using the `DummyClassifier` classifier to compare with the results of the models.
- ❖ 2 primary metrics were used:
 - Overall Accuracy: How accurate was the model in predicting the correct topic?
 - Recall of the 'Other' topic: How accurate was the model in predicting 'Other' for actual 'Other' posts?
- ❖ The models were run using Grid Search CV for hyper-parameter tuning to improve results.
- ❖ Baseline accuracy score: 0.0787
- ❖ Baseline 'Other' recall score: 0.10

Accuracy: 0.0756				
Average 10-Fold Cross-Validation Accuracy Score: 0.0787				
Classification Report:				
	precision	recall	f1-score	support
auto	0.06	0.06	0.06	419
bank account	0.06	0.07	0.06	350
budgeting	0.06	0.05	0.06	293
collections	0.06	0.05	0.05	469
credit card	0.10	0.10	0.10	642
debt	0.02	0.02	0.02	125
employment	0.09	0.09	0.09	593
housing	0.11	0.11	0.11	849
insurance	0.03	0.03	0.03	323
investment	0.06	0.06	0.06	497
loans	0.09	0.08	0.09	550
other	0.10	0.11	0.10	729
planning	0.01	0.01	0.01	253
school	0.05	0.06	0.05	322
taxes	0.08	0.08	0.08	541
avg / total	0.08	0.08	0.08	6955

Linear SVC

- ❖ Linear SVC is great for text classification due to its effective nature in high dimensional spaces (lots of words).
 - We can incorporate bigrams (2-word phrases) and trigrams (3-word phrases) without resulting in overfitting due to this feature.
- ❖ 79.45% of the topics were correctly predicted, which is 10x better than the 7.87% from the baseline!
- ❖ ... however, the predictions show that there was an inconsistency in the original topic labeling for 'Other'. Some 'Other' posts did not belong in 'Other'!
- ❖ These are examples where 'Other' posts were falsely predicted as 'Auto' but they are better suited for 'Auto'.



Accuracy: 0.7896				
Average 10-Fold Cross-Validation Accuracy Score: 0.7945				
Classification Report:				
	precision	recall	f1-score	support
auto	0.87	0.88	0.88	419
bank account	0.79	0.83	0.81	350
budgeting	0.80	0.74	0.77	293
collections	0.80	0.74	0.77	469
credit card	0.89	0.90	0.90	642
debt	0.59	0.35	0.44	125
employment	0.80	0.83	0.82	593
housing	0.77	0.82	0.79	849
insurance	0.81	0.82	0.81	323
investment	0.84	0.85	0.85	497
loans	0.82	0.83	0.83	550
other	0.61	0.60	0.60	729
planning	0.70	0.58	0.63	253
school	0.76	0.78	0.77	322
taxes	0.83	0.88	0.85	541
avg / total	0.79	0.79	0.79	6955

Bought a Used Car and was charged 4000 dollars more than the listed price Hey everyone, yesterday I bought a 2010 Ford Taurus SEL, The vehicle was listed on their website for 8.6 thousand dollars ...

Captial gains on commercial vehicle I just have what I hope will be a quick question regarding how captial gains works when selling a commercial vehicle. I own a two truck dump truck company regi...

Logistic Regression

- ❖ Logistic Regression performed the best with an accuracy of 80.15%! It also had the best precision (80%) and recall (80%).
- ❖ It did well in predicting the main topics, since documents which were labeled with main topics were likely to have had the right label.
- ❖ When we observe the predictions for 'Other', we again see the same problem as with the Linear SVC model: ambiguous words resulted in posts having 'Other' as their topic despite having more in common with one of the main topics.
- ❖ These are examples where 'Other' posts were falsely predicted as 'Loans' despite the subject being about loans.



Accuracy: 0.7976
Average 10-Fold Cross-Validation Accuracy Score: 0.8015

Classification Report:

	precision	recall	f1-score	support
auto	0.87	0.88	0.87	419
bank account	0.83	0.84	0.84	350
budgeting	0.81	0.72	0.76	293
collections	0.82	0.75	0.78	469
credit card	0.88	0.92	0.90	642
debt	0.71	0.28	0.40	125
employment	0.81	0.84	0.83	593
housing	0.77	0.83	0.80	849
insurance	0.82	0.80	0.81	323
investment	0.84	0.87	0.85	497
loans	0.83	0.85	0.84	550
other	0.60	0.63	0.61	729
planning	0.76	0.58	0.66	253
school	0.80	0.77	0.78	322
taxes	0.83	0.89	0.86	541
avg / total	0.80	0.80	0.79	6955

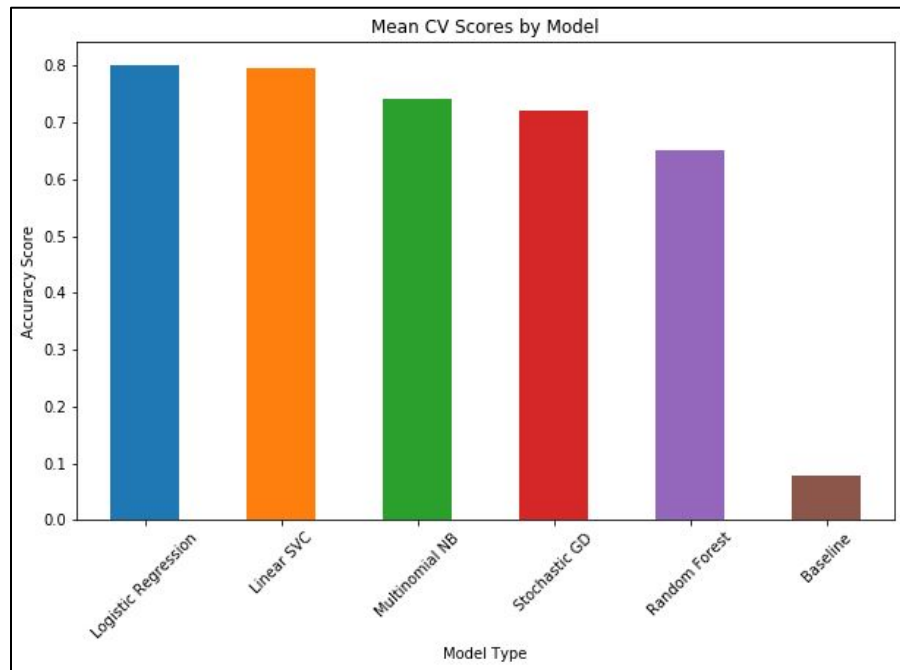
Does the Common Reporting Standard (CRS) share information to the UK Student Loan Company? Does the Common Reporting Standard (CRS) share information to the UK Student Loan Company? Thanks

Can someone please help out with the mortgage math here? Hi! Thanks in advance for your help!\n\nGot a 30-yr mortgage in March (493K @ 450K from my 40...

Loaning a large amount of money for a very short amount of time to a family member.

Model Results

- ❖ The best performing model was Logistic Regression with a 0.8015 accuracy score. It also had the highest overall precision and recall scores.
- ❖ While Naive Bayes is heralded as being one of the better models for text classification tasks, it did not do as well as Logistic Regression.
 - This is because Naive Bayes optimizes a generative objective function which makes it better for dealing with less training data.
- ❖ Most models had high precision and recall rates for their main topics. The eye-test also confirms that the main topics generally fit their respective posts.
- ❖ However, models suffered in the 'Other' topic because it contains posts that overlap with the main topics.



Conclusion - Recommendations

- ❖ 1) Split the Debt topic into multiple categories:

It is a vast topic that can have better search results for users if it were segmented into sub-topics like 'Loans', 'Collections' and 'Credit Card'.

- ❖ 2) Differentiate Credit and Credit Card:

Most posts under the original 'Credit' topic were about credit card debts. The other 'Credit' posts were more geared towards other topics like 'Loans' and 'Housing', so it could be better to simply use 'Credit Card' as a topic.

- ❖ 3) Consider introducing a new School/Student topic:

Reddit is dominated by teens and young adults. 5% of the total posts were clustered under the 'School' topic, which is a sizeable amount. This can also simplify the process of manually flairing for the younger users, since they won't have to worry about which topic to pick.

- ❖ 4) Use Machine Learning to assign topics to untagged posts using a similarity percentage threshold:

Rather than solely labeling posts by whichever topic has the most weight, we can also add a minimum weight percentage. For example, let's use a threshold of 40%. If a post does not have any topics with at least 40% weight, the post will simply be labeled as 'Other'. This is done to maximize subject accuracy.

Conclusion - Limitations

- ❖ Ambiguous topics/words result in mislabeling:

The most obvious limitation is the fact that some of the topics generated by the LDA model were ambiguous.

- ❖ These were labeled as 'Other', which resulted in some posts having 'Other' as their topic despite being better suited for a different topic such as 'Auto' or 'Loans'.

- ❖ This is because those posts had many words with strong weights towards the 'Other' topics.

- ❖ For example, if a post about credit cards contained more words from topic #5 ('Other') than from topic #6 ('Credit Cards'), it would be categorized under topic #5 and labeled as 'Other'.

```
(5,
 '0.037*"give" + 0.024*"find" + 0.021*"edit" + 0.019*"thing" + 0.018*"guy" + '
 '0.015*"number" + 0.015*"person" + 0.015*"people" + 0.014*"day" + '
 '0.014*"talk"'),
(6,
 '0.278*"credit" + 0.185*"card" + 0.059*"score" + 0.037*"balance" + '
 '0.017*"apply" + 0.017*"limit" + 0.016*"build" + 0.012*"history" + '
 '0.011*"line" + 0.011*"purchase"'),
(7,
 '0.193*"car" + 0.041*"buy" + 0.024*"vehicle" + 0.018*"finance" + '
 '0.018*"drive" + 0.017*"lease" + 0.017*"payment" + 0.015*"repair" + '
 '0.014*"mile" + 0.013*"worth"'),
(8,
 '0.124*"tax" + 0.066*"ira" + 0.050*"roth" + 0.048*"income" + 0.044*"year" + '
 '0.034*"retirement" + 0.030*"contribute" + 0.030*"contribution" + '
 '0.029*"employer" + 0.023*"match"'),
(9,
 '0.066*"question" + 0.033*"make" + 0.031*"personal" + 0.027*"finance" + '
 '0.025*"post" + 0.023*"business" + 0.020*"read" + 0.016*"advice" + '
 '0.015*"understand" + 0.014*"answer"'),
```

Appendix

1. <https://www.reddit.com/r/personalfinance/>
2. <https://github.com/pushshift/api>
3. <https://praw.readthedocs.io/en/latest/>
4. <http://pages.cs.wisc.edu/~jerryzhu/cs838/LR.pdf>
5. <https://www.statista.com/statistics/261766/share-of-us-internet-users-who-use-reddit-by-age-group/>

