2019

REDDIT DATA CLASSIFICATION

NAME: Soumita Chel

3/15/2019

Q1. Part A: Thread Subreddit Classification

- In this problem, the main aim is to predict the correct class of subreddit to which a single thread of discussion belongs to.
- As we know, a thread consists of a number of posts; created a train and test dataframes consisting of subreddit, title, id, URL, author, body.
- Since an author field can be null, we have replaced them in both train and test dataset with a word called 'Blank'.(places where it has null value).
- It is considered the combination of author id, title and body of the post for effectively training the required models.

	subreddit	title	id	url	author	body
0	relationships	How do I [23F] communicate with my self-center...	t1_covzqua	https://www.reddit.com/r/relationships/comment...	Pouritdownmythroat	I think everyone has that one friend who loves...
1	relationships	How do I [23F] communicate with my self-center...	t1_cow04yo	https://www.reddit.com/r/relationships/comment...	WhyFrankWhy	Good point! I definitely wanna keep her as my ...
2	relationships	How do I [23F] communicate with my self-center...	t1_cow4211	https://www.reddit.com/r/relationships/comment...	Pouritdownmythroat	Girl, I know where you're coming from. I have ...
3	relationships	How do I [23F] communicate with my self-center...	t1_cow4esm	https://www.reddit.com/r/relationships/comment...	WhyFrankWhy	Great advice :) I'm pretty optimistic that we ...
4	relationships	How do I [23F] communicate with my self-center...	t1_covz2io	https://www.reddit.com/r/relationships/comment...	eshtive353	Hannah seems like a pretty shitty friend. Frie...
5	relationships	How do I [23F] communicate with my self-center...	t1_covzfrs	https://www.reddit.com/r/relationships/comment...	WhyFrankWhy	Yeah, I've tried talking to her calmly. And sh...
6	relationships	How do I [23F] communicate with my self-center...	t1_covzlik	https://www.reddit.com/r/relationships/comment...	eshtive353	It's up to you to decide how close you want to...
7	summonerschool	What Cherry switch do you recommend for League...	t3_2w8jon	https://www.reddit.com/r/summonerschool/commen...	Blank	I not 100% sure this is the right place to pos...
8	summonerschool	What Cherry switch do you recommend for League...	t1_cooiwv5	https://www.reddit.com/r/summonerschool/commen...	Blank	Post might be removed but I'll answer anyways\...

i. Tokenization And Normalization

Tokenization and normalization on the below fields initially with the help of spacy.

Body & Title- This field contains long sequence of words which initially is not in standard format.

For example, consider the below body text (part of it) from one of the Reddit thread.

Tokenization

```
0        I think everyone has that one friend who loves...
```

This will be broken down into small token like {'I','think','everyone'….} on the tokenization step.

Normalization

It is considered the below normalization steps to process the required fields for training.

The preprocessing would include everything in lowercase, alphanumeric and numeric. **Since the Reddit posts and title consist of such cases, it has been explicitly considered them for better performance of the models.**

ii. Macro Classifier Performance

Below are the results obtained from training the train set with all classifiers as listed below. The row highlighted in yellow is the one with the best performance.

Macro Classifier Performance For All Classifiers					
Name of the Classifier	**Vectorizer/Strategy**	**Accuracy Score**	**Macro-average Precision**	**Macro-average Recall**	**Macro-average F1 measures**
Logistic Regression	**One Hot Encoding**	0.558	0.442	0.562	0.465
Logistic Regression	**TFidfVectorizer**	0.550	0.408	0.651	0.442
SVC Classifier(RBF kernel)	**One Hot Encoding**	0.269	0.057	0.095	0.034
SVC Classifier(RBF kernel)	**TFidfVectorizer**	0.261	0.050	0.013	0.021
MLP Classifier	**One Hot Encoding**	0.605	0.499	0.634	0.518
MLP Classifier	**TFidfVectorizer**	0.431	0.206	0.202	0.171
Dummy Classifier 1	**Strategy =most frequent**	0.261	0.050	0.013	0.021
Dummy Classifier 2	**Strategy =stratified**	0.261	0.050	0.013	0.021

Table1: Macro Classifier Performance for All Classifiers

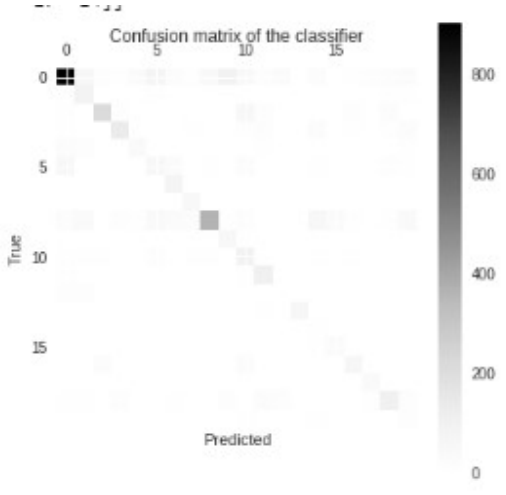

Fig1: Confusion Matrix of the Model with Best Performance

iii. Graphical Representation of the Model with Best Performance

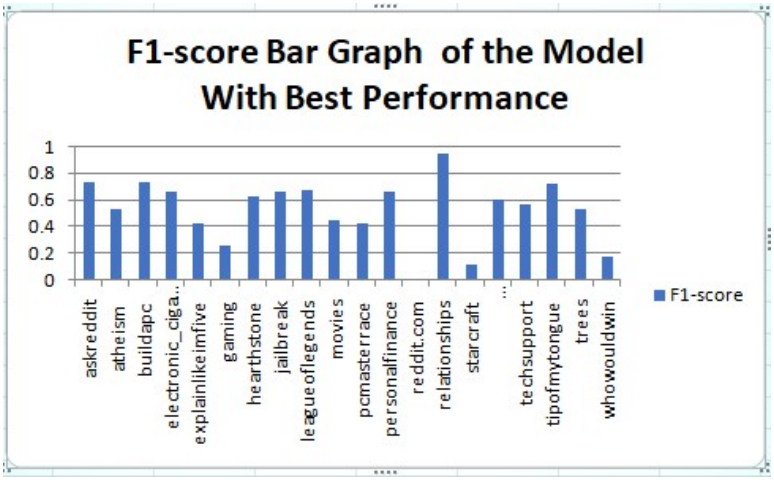

Fig2: F1-score Bar Graph of the Model with Best Performance

iv. Reason for combination of the classifier and encoding of the above model

Multi-layer Perceptron is a supervised learning algorithm that trains its data through backpropagation. Interaction happens only between two neighboring neurons. This algorithm works best on scaled data; the dataset used is also a scaled one. Here have considered categorical data for example the subreddit classes like atheism, movies etc. Many machine learning models cannot interpret these labels. It is required to convert categorical data to numerical values for them, which is done by one hot encoding vectorizer. It is also seen from our experimented data that , the above combination has a good accuracy score and also a good F1-score. Hence it has been considered this to be the model with best performance.

Q2. Part A: Thread Subreddit Classification

i. Best Parameters found for Logistic Regression Model with TF-IDF Vectorization

Type	Parameter Name	Values
Logistic Regression Model	**Solver**	`liblinear`
	Multi_class	`ovr`
	Regularization C	`100.0`
TF-IDF Vectorizer	**sublinear_tf**	`False`
	ngram_range	`(1, 1)`
	max_features	`10000`

Table2: Best Parameters Values for LR Model with TF_IDF

Macro Classifier Performance For Logistic Regression Model (TF_IDF) with Best Features			
Accuracy Score	**Macro-average Precision**	**Macro-average Recall**	**Macro-average F1 measures**
0.561	0.445	0.559	0.463

Table3: Result on Test Data with Best Parameters Values for LR Model with TF_IDF Vectorization

ii. Error Analysis And Findings

- It was found out on the true labels i.e. the test tables and the predicted labels.
- On analysis, it was found that there is total 1762 mismatch among these labels. So total of 1762 out of 4016 test labels were predicted wrong.
- A closer look on few examples yielded in below observations:

```
subreddit                                              whowouldwin
title        Your Favorite Hero Now Has A Healing Factor As...
id                                                      t1_cly3vq5
url          https://www.reddit.com/r/whowouldwin/comments/...
author                                                Wolvenfire86
```

Example 1: Error Analysis

This was predicted by the model to be under 'askreddit' subreddit category. But the true label for this is 'whowouldwin'. In another instance, it was predicted by the model to be under 'astheism' subreddit category. But the true label for this is 'whowouldwin'. Both the titles are same as in correspondence to the subreddit

```
Count of Mismatch labels: 1762
subreddit                                              buildapc
title        [£] What is the cheapest z97 motherboard I can...
id                                                      t1_co4xnzx
url          https://www.reddit.com/r/buildapc/comments/2u4...
author                                                   Blank
Name: 80, dtype: object
```

Example 2: Error Analysis

Going by other few examples, it was observed tokenization was not fully done to get a standardized format, particularly in case of ASCII characters. Also, the title being same, it was ascertained than addition of another new column could do a fair distribution and decrease the number of mismatch labels. Two new features have been introduced as below, which can be further enhanced considering punctuations.

iii. Feature Development

A. Addition of new column called post length in Dataframe
B. Consideration of ASCII characters in Tokenization

Macro Classifier Performance For Logistic Regression Model (TF_IDF) with Best Features and Above Feature Additions			
Accuracy Score	**Macro-average**	**Macro-average Recall**	**Macro-average F1**

	Precision		measures
0.594	0.507	0.611	0.524

Table4: Result on Test Data with Best Parameters Values for LR Model with TF_IDF Vectorization and Feature Additions

Q3.Part B: Comment Discourse Classification

Any null column is replaced by 'Blank'.

i. Macro Classifier Performance For Logistic Regression Model (TF_IDF) with Best Features and Feature Additions For Comment Discourse			
Class Name	**Macro-average Precision**	**Macro-average Recall**	**Macro-average F1 measures**
Blank	0.165	0.159	0.162
agreement	0.177	0.250	0.207
announcement	0.249	0.298	0.272
answer	0.585	0.505	0.542
appreciation	0.505	0.557	0.530
disagreement	0.056	0.105	0.073
elaboration	0.245	0.249	0.247
humor	0.050	0.108	0.069
negativereaction	0.065	0.165	0.093
other	0.077	0.117	0.093
question	0.601	0.588	0.594

Table5: Result on Test Data with Best Parameters Values for LR Model with TF_IDF Vectorization and Feature Additions for Comment Discourse

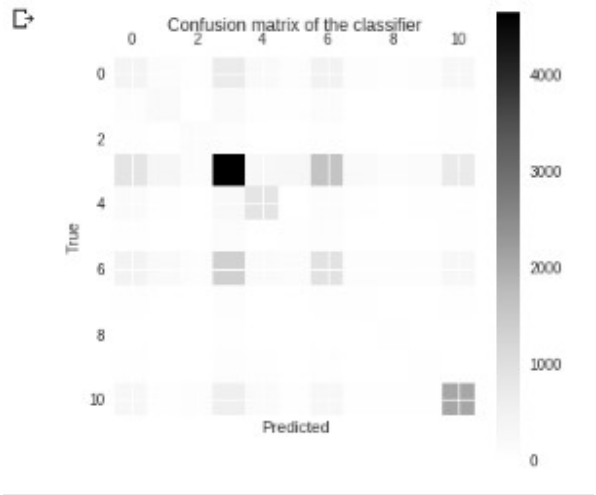

Fig3: Confusion Matrix of the Model

Error Analysis And Findings

Considering the below example, we can see still there is problem with tokenization.

```
* Even in [spoiler] threads, consider using spoiler markup  - [Boba
Fett Spoilers]\(/s "Boba Fett loves to hunt.") = [Boba Fett
Spoilers](/s "Boba Fett loves to hunt.")
```

We can still do more standard tokenization with the basis of punctuations. Also few other columns like subreddit and majority_link as features would help in better classifications.

Q4.Part B: Comment Discourse Classification

Feature Name	Feature Type	Feature Details	Example	Implementation
token.is_punct	Content + Punctuation	It is a Boolean feature of spaCy Tokenizer which tells if a token is punctuation or not.	token=”,” >>token.is_punct() >>True	Implemented under normalization of spaCy.
Reason	We have seen in the error analysis that there is still a chance to tokenize on the basis of punctuations, hence this feature is considered.			
subreddit	Community	It is category of discussion based on specific topic.	whowouldwin	Implemented under feature union of pipeline.
Reason	Subreddit specifies the category of the initial post. Like for the example “whowouldwin” says that the author is asking a question. Thus, this can be related directed to discourse_type, hence this feature is considered.			
author_check	Author	It is a Boolean value column added in the dataframes which says whether the current author is also the author of the initial post	>>author= “vurt” >>in_reply_to=” t3_2v0anq” >>False	Implemented under feature union of pipeline.
Reason	In case of an initial post, there will be only author id and no in_reply_to value. This can either be question, announcement. In case this column has a True value, it means that this good be an agreement or negative reaction. This can help the model train comments correctly.			
post_depth	Structure	It is a numerical column which specifies a number based on the hierarchy of the comment in the thread structure.	Initial Post, if Question, is 0 as it is the starting point, answer could be 1 or 2 depending on the hierarchy level.	Implemented under feature union of pipeline.
Reason	Since the main task here is to classify into discourse type, post_depth helps in this task. By seeing the post_depth, machine can distinguish which type of discourse_type it is.			
majority_link **Reason**	Other	It is 0 if the first post is a question.	0 in case the first post is of question type.	Implemented under feature union of pipeline

Reason	Helps in finding the hierarchy of a comment, in turn helping to identify an discourse type.			
self_post	Metadata	It is a binary value column, 1.0 if the first post in the thread is a self-post.	1.0, if the first post in the thread is a self-post.	Implemented under feature union of pipeline
Reason	Helps in getting the first post reference, which can further help in discourse_type identification.			

Macro Classifier Performance For Logistic Regression Model (TF_IDF) with Best Features and Six Feature Additions				
Feature	**Accuracy Score**	**Macro-average Precision**	**Macro-average Recall**	**Macro-average F1 measures**
token. Is_punct	0.416	0.260	0.281	0.270
subreddit	0.423	0.257	0.291	0.267
is_self_post	0.419	0.258	0.285	0.267
post_depth	0.496	0.317	0.351	0.329
majority_link	0.415	0.306	0.332	0.313
author_check	0.416	0.252	0.282	0.262
Combined	0.518	0.359	0.419	0.378

Feature Importance Graph

y=Blank top features		y=agreement top features		y=announcement top features		y=answer top features		y=appreciation top features		y=disagreement top features		y=elaboration top features	
Weight[?]	Feature	Weight[?]	Feature	Weight[?]	Feature	Weight[?]	Feature	Weight[?]	Feature	Weight[?]	Feature	Weight[?]	Feature
+18.926	x6274	+30.318	x582	+15.937	x30331	+22.123	x28103	+35.824	x8912	+13.612	x6762	+17.027	x5200
+17.410	x453	+23.125	x583	+10.057	x30074	+21.026	x27510	+31.763	x8909	+13.415	x8103	+16.479	x29225
+17.201	x6843	+15.006	x1916	+9.995	x30003	+20.226	x28916	+19.006	x9000	+13.357	x2783	+15.973	x4730
+15.421	x1009	+14.687	x27254	+8.596	x14961	+19.961	x26147	+17.529	x3958	+13.291	x2674	+15.916	x5550
+14.383	x1461	+14.671	x9748	*… 8500 more positive …*		+19.445	x28757	+16.869	x1891	+12.826	x3035	+15.433	x9602
+14.066	x9009	+14.571	x2145	*… 23955 more negative …*		+19.317	x25322	+16.412	x796	+12.148	x6902	*… 13575 more positive …*	
+14.064	x28731	+14.443	x5942	-6.840	x13917	+18.121	x24320	+15.346	x4488	+12.148	x6015	*… 18880 more negative …*	
+13.761	x9425	+14.208	x9930	-6.885	x9840	+16.931	x27952	+14.843	x27305	+12.146	x2259	-15.355	x5871
+13.648	x3940	+14.087	x7499	-6.940	x10851	+15.109	x25970	+14.624	x5030	+12.144	x2921	-15.616	x2744
+13.568	x1989	+14.087	x27307	-7.060	x19698	+14.063	x27085	+14.407	x27617	+11.986	x6178	-15.743	x7677
+13.428	x1182	+14.002	x22496	-7.106	x19932	+13.258	x27347	+14.191	x8380	+11.914	x22890	-15.791	x1480
+13.378	x5687	+13.818	x8723	-7.165	x12893	*… 15586 more positive …*		+14.187	x3348	+11.727	x9308	-15.904	x6835
… 12095 more positive …		+13.679	x9227	-7.186	x14139	*… 16869 more negative …*		+14.176	x9764	*… 8337 more positive …*		-15.951	x498
… 20360 more negative …		+13.231	x7602	-7.304	x16325	-12.831	x28072	*… 9822 more positive …*		*… 24118 more negative …*		-16.687	x9037
-13.429	x6255	*… 9308 more positive …*		-7.544	x751	-12.899	x23733	*… 22633 more negative …*		-11.990	x3782	-17.041	x582
-13.631	x3194	*… 23147 more negative …*		-7.804	x16018	-13.112	x29871	-14.259	x31399	-12.002	x7159	-17.947	x5446
-14.826	x1717	-14.103	x3299	-7.909	x16728	-13.389	x1261	-14.495	x4856	-12.084	x8443	-18.194	x4623
-15.106	x2004	-14.124	x8909	-7.930	x755	-13.497	x29198	-14.712	x10662	-12.202	x8912	-18.461	x5515
-15.179	x8811	-15.366	x1506	-9.451	x17102	-13.770	x8912	-14.962	x6362	-12.982	x7034	-18.591	x1500
-15.259	x2436	-18.821	x352	-10.630	x14275	-13.787	x25283	-15.192	x2258	-13.148	x4753	-18.926	x8912
-16.715	x5724	-21.223	x8912	-28.176	x352	-13.815	x582	-15.349	x98	-13.546	x8517	-20.564	x4404
-19.273	x6162	-22.112	x1314	-28.942	x10499	-19.772	x352	-19.127	x8816	-14.341	x352	-25.672	x352

y=humor top features		y=negativereaction top features		y=other top features		y=question top features	
Weight[?]	Feature	Weight[?]	Feature	Weight[?]	Feature	Weight[?]	Feature
+14.192	x8398	+13.112	x8984	+17.742	x20020	+37.580	x352
+13.819	x527	+12.050	x5635	+13.605	x4700	+20.566	x17522
+13.356	x6952	+11.996	x5856	+12.373	x24358	+19.599	x29196
+11.955	x1038	+11.852	x20221	+11.789	x7218	*... 13529 more positive ...*	
+11.408	x9097	+11.666	x7751	+11.618	x9919	*... 18926 more negative ...*	
+11.358	x2726	+11.379	x3854	+11.407	x4135	-19.239	x29027
+11.344	x6032	+11.270	x3025	+11.139	x27700	-19.984	x31
+11.145	x9459	+11.006	x7029	+11.068	x360	-20.579	x27696
+10.983	x6133	+10.935	x36	+11.062	x8533	-23.320	x24320
+10.941	x2788	+10.706	x7451	+11.004	x7356	-24.117	x16148
+10.808	x8513	+10.698	x5207	+10.607	x2506	-24.743	x25970
+10.781	x1272	+10.692	x3515	+10.489	x6472	-24.743	x14546
+10.766	x8641	+10.521	x1259	+10.481	x24221	-24.768	x8838
+10.737	x1834	+10.299	x9594	+10.446	x27531	-26.508	x22459
... 5293 more positive ...		+10.229	x2498	+10.279	x24757	-26.970	x22755
... 27162 more negative ...		+10.194	x4603	+10.259	x619	-29.457	x27952
-10.651	x9847	*... 5373 more positive ...*		*... 4686 more positive ...*		-29.793	x28103
-10.724	x3927	*... 27082 more negative ...*		*... 27769 more negative ...*		-29.848	x26147
-11.099	x8912	-10.132	x9229	-10.633	x5296	-30.080	x26230
-11.586	x6358	-10.497	x8885	-11.327	x453	-30.522	x27510
-11.928	x4189	-10.525	x1735	-13.593	x352	-30.852	x28916
-15.499	x9399	-10.871	x19	-16.140	x30003	-31.478	x25322

As per the above graph, for features like humor and negative reaction, the model is well trained. For announcement, the model still needs further tuning which is considered for future work.