TEXT AS DATA ASSIGNMENT

**2019**

**3/15/2019**

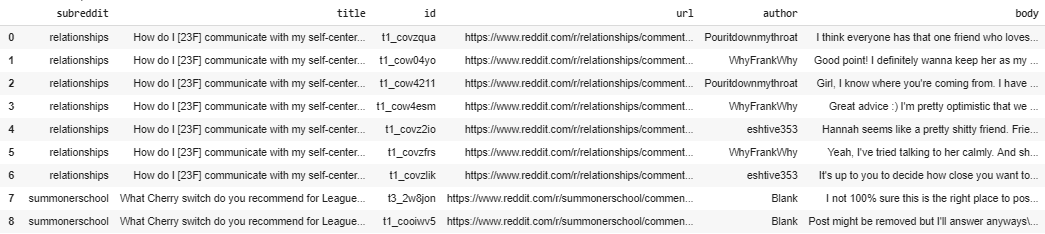
**NAME: Soumita Chel**

**Course: MSC in Data Science**

**ID: 2422493c**

**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.



1. **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**



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.

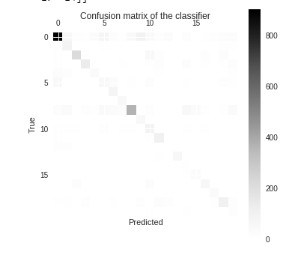
The preprocessing would include everything in lowercase, alphanumeric and numeric. **Since the Reddit posts and title consist of such cases, it has been explicitly considered them for better performance of the models.**

1. **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.

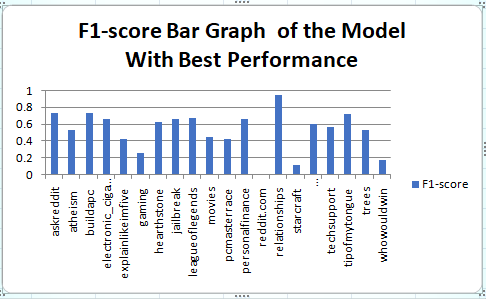
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Macro Classifier Performance For All Classifiers** | | | | | |
| **Name of the Classifier** | **Vectorizer/Strategy** | **Accuracy Score** | **Macro-average Precision** | **Macro-average Recall** | **Macro-average F1 measures** |
| **Logistic 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

1. **Graphical Representation of the Model with Best Performance**

****

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

1. **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**

1. **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

1. **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.

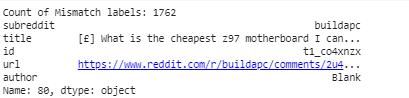
So total of 1762 out of 4016 test labels were predicted wrong.

* A closer look on few examples yielded in below observations:



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



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.

1. **Feature Development**
2. Addition of new column called post length in Dataframe
3. 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 Precision** | **Macro-average Recall** | **Macro-average F1 measures** |
| 0.594 | 0.507 | 0.611 | 0.524 |

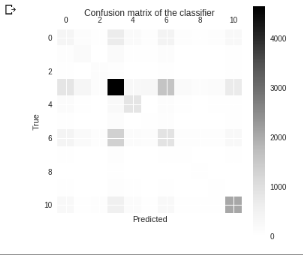
**Table4:** Result on Test Data with Best Parameters Values for LR Model with TF\_IDF Vectorization and Feature Additions

**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.")

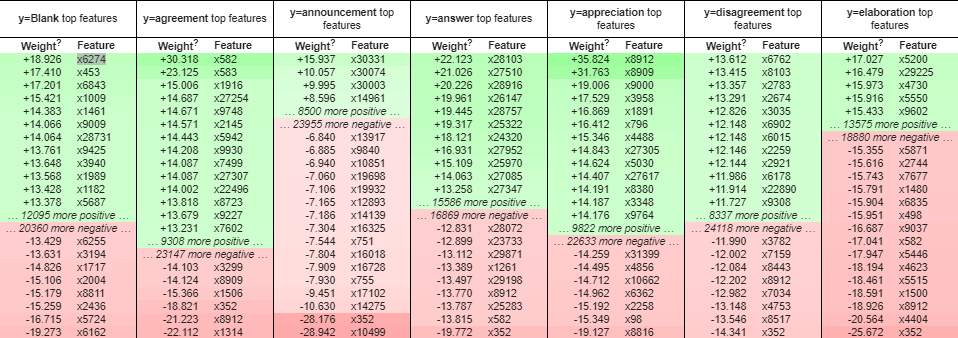
We can still do more standard tokenization with the basis of punctuations. Also few other columns like subreddit and majority\_link as features would help in better classifications.

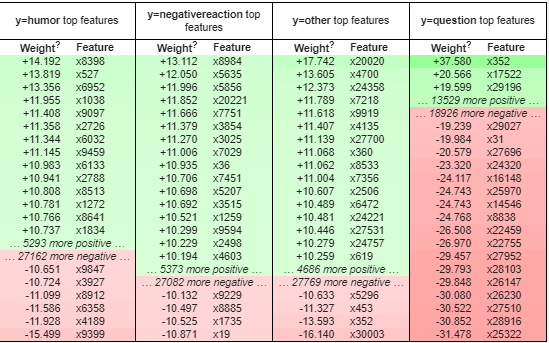
**Q4.Part B: Comment Discourse Classification**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Feature Name** | **Feature Type** | **Feature Details** | **Example** | | **Implementation** | |
| **token.is\_punct**  **Reason** | 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. | |
| 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**  **Reason** | Community | It is category of discussion based on specific topic. | whowouldwin | | Implemented under feature union of pipeline. | |
| 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**  **Reason** | 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. | |
| 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**  **Reason** | 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. | |
| 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**  **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 | |
| Helps in finding the hierarchy of a comment, in turn helping to identify an discourse type. | | | | | |
| **self\_post**  **Reason** | 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 |
| 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**





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.