Text-As-data Course Work Report

Name: SHAMBHAVI SINGH
Student Number: 2711327S

Q1a:

On exploring the dataset and grouping them by "subreddit", we observe that there are 9 different labels namely, Coffee, HydroHomies, NintendoSwitch, PS4, Soda, antiMLM, pcgaming, tea and xbox. For starters, the counts of the "subreddit" columns of all the three different datasets were calculated. The obtained results are displayed below in the form of a table.

	Coffee	Hydro Homies	Nintendo Switch	PS4	Soda	antiMLM	pcgaming	tea	xbox
Train Set	136	134	145	142	102	128	135	146	132
Validation Set	42	38	52	43	43	54	43	48	37
Test Set	56	38	52	48	29	44	47	42	44

Table1: Table representing the distribution of data

It can be seen from the table that the distribution of the labels under subreddit are almost similar ranging from 128 – 145 except that for *Soda* type which is a bit low i.e., 102 so, it might result in poor training for this subreddit type.

Further exploration included plotting of a graph comparing the counts of the subreddits in the train, test and validation datasets. Below is the graph generated:

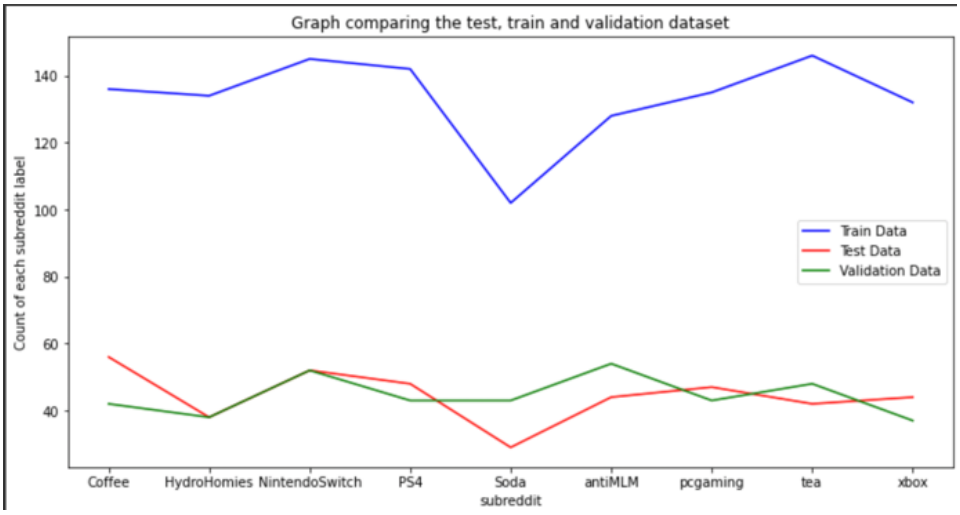


Fig1: Graph comparing the count of subreddits between the train, test, and validation dataset

Q1b:

To begin with, as required, pre-processing of the data was performed and then a common evaluation summary function was created which generates and returns the classification report including the accuracy, precision, recall and F1 scores as well. Following this step, all the given classifiers were trained on the train data and then tested on both the train as well as the test data. Evaluation summary for each of the classifier is calculated along with the entire classification report, confusion matrix and heat maps for both test and train datasets.

	Result on Training Data	**Result on Test Data**
Dummy Classifier with strategy="most_frequent"	Accuracy: 0.122 Precision: 1.000 Recall: 0.122	Accuracy: 0.105 Precision: 1.000 Recall: 0.105
Dummy Classifier with strategy="stratified"	Accuracy: 0.122 Precision: 1.000 Recall: 0.122	Accuracy: 0.107 Precision: 0.114 Recall: 0.107
LogisticRegression with One-hot vectorization	Accuracy: 0.997 Precision: 0.997 Recall: 0.997	Accuracy: 0.743 Precision: 0.746 Recall: 0.743
LogisticRegression with TF-IDF vectorization	Accuracy: 0.994 Precision: 0.994 Recall: 0.994	**Accuracy: 0.770** **Precision: 0.777** **Recall: 0.770**
SVC Classifier with One-hot vectorization (SVM with RBF kernel, default settings)	Accuracy: 0.947 Precision: 0.951 Recall: 0.947	Accuracy: 0.670 Precision: 0.701 Recall: 0.670

Table 2: Result comparison between the classifiers

1) Talking about the model fit, it can be inferred from the accuracy of the models on the training data. Ideally, if a model is trained perfectly then the accuracy obtained should be ~ 99-100%. According to this logic, the model with the combination, Logistic regression and One-hot vectorizer is the best trained model as the resultant **accuracy is 99.7%** on the train data whereas, the models with the Dummy Classifier give an **accuracy of 12.2%** on the train data which indicates the poor training of data. Apart from these, the other model combinations, SVC, and Logistic Regression + TF-IDF vectorizer also perform good in terms of training.
2) Dataset label distribution is uneven, like *Soda* type has comparatively less data than that of other therefore it might lead to poor training of this label type.
3) Dummy Classifier model was used with stratified= 'most_frequent' and 'stratified' with one-hot vectorization; Logistic regression was used with both one-hot vectorization and TF-IDF vectorization; and SVC classifier was used with RBF kernel along with one-hot vectorization. Also, all these classifiers were used in their default settings.

From the above table, it can be concluded that the classifier **Logistic Regression with TF-IDF vectorization gives the best result with an accuracy of 77% on test data and 99.4% on train data.** Also, it has the highest f1 scores under the weighted and macro average. For other classifiers where the f1 scores are low maybe because of the uneven distribution of the uneven data. Below is the bar graph generated with the F1 scores for each class for the best performing classifier i.e., Logistic Regression with TF-IDF vectorizer.

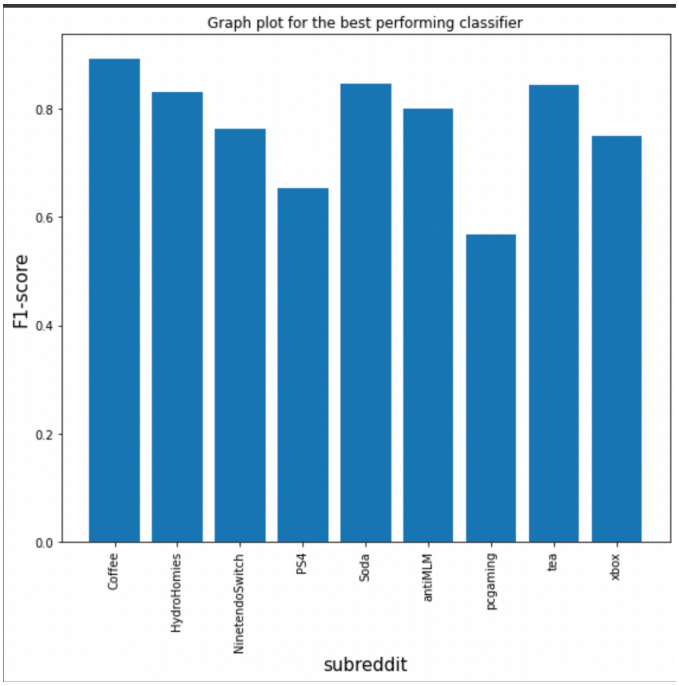
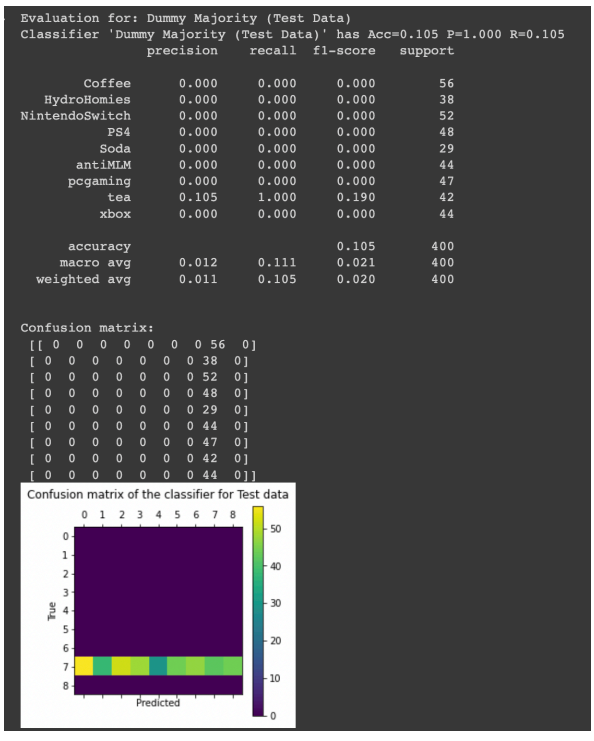


Fig2: Bar graph plot between the F1 score and the different subreddits for the best performing classifier

Below are the snapshots of the results of the different classifiers on the test data:

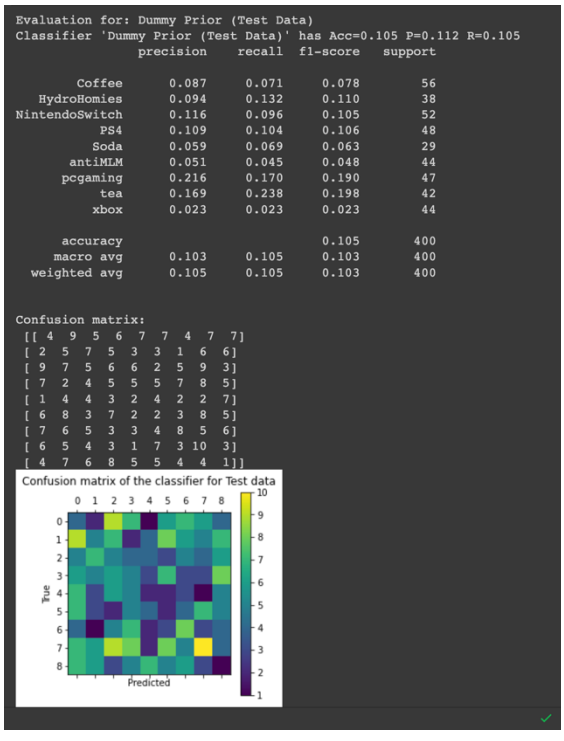


```
Evaluation for: Dummy Majority (Test Data)
Classifier 'Dummy Majority (Test Data)' has Acc=0.105 P=1.000 R=0.105
              precision    recall  f1-score   support

      Coffee       0.000     0.000     0.000        56
  HydroHomies      0.000     0.000     0.000        38
NintendoSwitch     0.000     0.000     0.000        52
         PS4       0.000     0.000     0.000        48
        Soda       0.000     0.000     0.000        29
     antiMLM       0.000     0.000     0.000        44
   pcgaming        0.000     0.000     0.000        47
         tea       0.105     1.000     0.190        42
        xbox       0.000     0.000     0.000        44

    accuracy                           0.105       400
   macro avg       0.012     0.111     0.021       400
weighted avg       0.011     0.105     0.020       400


Confusion matrix:
[[ 0  0  0  0  0  0  0 56  0]
 [ 0  0  0  0  0  0  0 38  0]
 [ 0  0  0  0  0  0  0 52  0]
 [ 0  0  0  0  0  0  0 48  0]
 [ 0  0  0  0  0  0  0 29  0]
 [ 0  0  0  0  0  0  0 44  0]
 [ 0  0  0  0  0  0  0 47  0]
 [ 0  0  0  0  0  0  0 42  0]
 [ 0  0  0  0  0  0  0 44  0]]
```

```
Evaluation for: Dummy Prior (Test Data)
Classifier 'Dummy Prior (Test Data)' has Acc=0.105 P=0.112 R=0.105
              precision    recall  f1-score   support

      Coffee       0.087     0.071     0.078        56
  HydroHomies      0.094     0.132     0.110        38
NintendoSwitch     0.116     0.096     0.105        52
         PS4       0.109     0.104     0.106        48
        Soda       0.059     0.069     0.063        29
     antiMLM       0.051     0.045     0.048        44
   pcgaming        0.216     0.170     0.190        47
         tea       0.169     0.238     0.198        42
        xbox       0.023     0.023     0.023        44

    accuracy                           0.105       400
   macro avg       0.103     0.105     0.103       400
weighted avg       0.105     0.105     0.103       400


Confusion matrix:
[[ 4  9  5  6  7  7  4  7  7]
 [ 2  5  7  5  3  3  1  6  6]
 [ 9  7  5  6  6  2  5  9  3]
 [ 7  2  4  5  5  5  7  8  5]
 [ 1  4  4  3  2  4  2  2  7]
 [ 6  8  3  7  2  2  3  8  5]
 [ 7  6  5  3  3  4  8  5  6]
 [ 6  5  4  3  1  7  3 10  3]
 [ 4  7  6  8  5  5  4  4  1]]
```

Fig 3: Dummy classifier with strategy =" most_frequent"

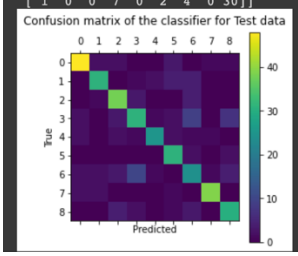
Fig 4: Dummy classifier with strategy="stratified"

```
Evaluation for: LR onehot (Test Data)
Classifier 'LR onehot (Test Data)' has Acc=0.743 P=0.746 R=0.743
                precision    recall  f1-score   support

        Coffee       0.842     0.857     0.850        56
   HydroHomies       0.738     0.816     0.775        38
NintendoSwitch       0.826     0.731     0.776        52
           PS4       0.564     0.646     0.602        48
          Soda       0.735     0.862     0.794        29
       antiMLM       0.886     0.705     0.785        44
     pcgaming        0.511     0.511     0.511        47
           tea       0.886     0.929     0.907        42
          xbox       0.750     0.682     0.714        44

      accuracy                           0.743       400
     macro avg       0.749     0.749     0.746       400
  weighted avg       0.749     0.743     0.743       400


Confusion matrix:
 [[48  0  0  2  2  0  2  2  0]
 [ 2 31  1  0  0  0  2  2  0]
 [ 2  0 38  3  2  0  3  0  4]
 [ 0  1  3 31  1  0 10  0  2]
 [ 0  2  0  1 25  0  1  0  0]
 [ 3  4  0  2  2 31  0  1  1]
 [ 0  4  4  9  1  2 24  0  3]
 [ 1  0  0  0  1  0  1 39  0]
 [ 1  0  0  7  0  2  4  0 30]]
```




```
Evaluation for: LR TF-IDF (Test Data)
Classifier 'LR TF-IDF (Test Data)' has Acc=0.770 P=0.777 R=0.770
                precision    recall  f1-score   support

        Coffee       0.893     0.893     0.893        56
   HydroHomies       0.821     0.842     0.831        38
NintendoSwitch       0.724     0.808     0.764        52
           PS4       0.623     0.688     0.653        48
          Soda       0.957     0.759     0.846        29
       antiMLM       0.783     0.818     0.800        44
     pcgaming        0.610     0.532     0.568        47
           tea       0.792     0.905     0.844        42
          xbox       0.833     0.682     0.750        44

      accuracy                           0.770       400
     macro avg       0.782     0.769     0.772       400
  weighted avg       0.774     0.770     0.769       400


Confusion matrix:
 [[50  1  1  0  0  0  4  0]
 [ 1 32  0  0  0  1  2  2  0]
 [ 1  0 42  3  0  1  3  0  2]
 [ 0  0  5 33  0  1  8  0  1]
 [ 1  3  0  1 22  1  0  1  0]
 [ 1  3  0  0  0 36  0  3  1]
 [ 0  0  8  8  0  4 25  0  2]
 [ 1  0  1  0  1  0  1 38  0]
 [ 1  0  1  8  0  2  2  0 30]]
```

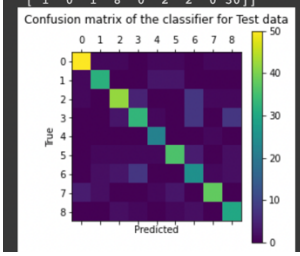


Fig 4: Logistic Regression with One-hot vectorizer

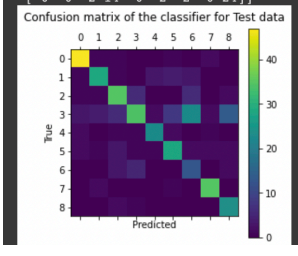
Fig 5: Logistic Regression with TF-IDF vectorizer

```
Evaluation for: SVC onehot (Test Data)
Classifier 'SVC onehot (Test Data)' has Acc=0.670 P=0.701 R=0.670
                precision    recall  f1-score   support

        Coffee       0.904     0.839     0.870        56
   HydroHomies       0.707     0.763     0.734        38
NintendoSwitch       0.686     0.673     0.680        52
           PS4       0.354     0.708     0.472        48
          Soda       0.767     0.793     0.780        29
       antiMLM       0.667     0.636     0.651        44
     pcgaming        0.565     0.277     0.371        47
           tea       0.921     0.833     0.875        42
          xbox       0.889     0.545     0.676        44

      accuracy                           0.670       400
     macro avg       0.718     0.674     0.679       400
  weighted avg       0.715     0.670     0.675       400


Confusion matrix:
 [[47  1  1  3  2  2  0  0  0]
 [ 1 29  1  4  0  1  0  2  0]
 [ 1  1 35  6  1  3  3  0  2]
 [ 0  0  6 34  1  1  5  0  1]
 [ 0  3  0  2 23  1  0  0  0]
 [ 1  4  0  8  2 28  0  1  0]
 [ 0  3  6 23  0  2 13  0  0]
 [ 2  0  0  2  1  2  0 35  0]
 [ 0  0  2 14  0  2  2  0 24]]
```




```
Classifier 'One-hot Naive Bayes (Test Data)' has Acc=0.720 P=0.761 R=0.720
                precision    recall  f1-score   support

        Coffee       0.875     0.860     0.867        57
   HydroHomies       0.763     0.906     0.829        32
NintendoSwitch       0.846     0.620     0.715        71
           PS4       0.667     0.667     0.667        48
          Soda       0.379     1.000     0.550        11
       antiMLM       0.818     0.735     0.774        49
     pcgaming        0.362     0.810     0.500        21
           tea       0.929     0.534     0.678        73
          xbox       0.705     0.816     0.756        38

      accuracy                           0.720       400
     macro avg       0.705     0.772     0.704       400
  weighted avg       0.782     0.720     0.729       400


Confusion matrix:
 [[49  2  0  0  2  1  0  2  1]
 [ 0 29  0  0  1  1  0  1  0]
 [ 0  1 44  9  0  0 14  0  3]
 [ 0  0  2 32  1  1  7  0  5]
 [ 0  0  0  0 11  0  0  0  0]
 [ 0  3  2  1  1 36  4  0  2]
 [ 0  0  1  2  1  0 17  0  0]
 [ 7  3  0  2 12  5  3 39  2]
 [ 0  0  3  2  0  0  2  0 31]]
```

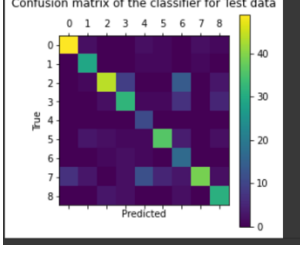


Fig 6: SVC with one-hot vectorizer

Fig 7: Self-selected classifier:Multinomial Naïve Bayes with one-hot vectorizer

Q1c:

The classifier selected is Multinomial Naïve Bayes with TF-IDF vectorizer. After performing several experiments with a combination of classifiers and vectorizers it was noticed that the model with the classifier as Multinomial Naïve Bayes is giving better results than that of others (except that of Logistic Regression) like Dummy, SVC, etc. Below is the table which compares the result with other classifiers:

	Result on Training Data	**Result on Test Data**
Dummy Classifier with strategy=”most_frequent”	Accuracy: 0.122 Precision: 1.000 Recall: 0.122	Accuracy: 0.105 Precision: 1.000 Recall: 0.105
Dummy Classifier with strategy=”stratified”	Accuracy: 0.122 Precision: 1.000 Recall: 0.122	Accuracy: 0.107 Precision: 0.114 Recall: 0.107
LogisticRegression with One-hot vectorization	Accuracy: 0.997 Precision: 0.997 Recall: 0.997	Accuracy: 0.743 Precision: 0.746 Recall: 0.743
LogisticRegression with TF-IDF vectorization	Accuracy: 0.994 Precision: 0.994 Recall: 0.994	Accuracy: 0.770 Precision: 0.777 Recall: 0.770
SVC Classifier with One-hot vectorization (SVM with RBF kernel, default settings)	Accuracy: 0.947 Precision: 0.951 Recall: 0.947	Accuracy: 0.670 Precision: 0.701 Recall: 0.670
Multinomial Naïve Bayes With TF-IDF vectorizer	**Accuracy: 0.978** **Precision: 0.979** **Recall: 0.978**	**Accuracy: 0.720** **Precision: 0.761** **Recall: 0.720**

Table 3: Result comparison after adding self-selected classifier

The TF-IDF vectorizer outdoes one hot encoding as it not only provides the frequency of the terms but also provides the importance of the terms. Using this less important terms can be filtered out before the analysis to make the model less complex with less inputs. Whereas one hot vectorizer works on binary basis and gives equal importance to every term, so the model with one hot vectorizer might give results biased towards the most frequent term. Thus, TF-IDF vectorizer was preferred over the one hot vectorizer.

Q2a:

Before Tuning: f1-macro for Logistic regression with TF-IDF vectorization is **0.772** and accuracy is **0.770.** For tuning of the parameters, GridSearch was applied to find the best

combination of parameters. As per the question, after the tuning below set of parameters were obtained as the best set of parameters:

Best parameters set:
logreg__C: 100.0
logreg__max_iter: 10000
logreg__penalty: '12'
logreg__solver: 'liblinear'
tf-idf__max_features: 10000
tf-idf__sublinear_tf: True

After tuning the parameters, f1-score for the macro average when applied Logistic Regression with TF-IDF vectorization is increased to **0.795** and accuracy has also increased to **0.790** Below is the table comparing the before and after results and a snapshot of the result of the model on the test and train data after the parameter tuning.

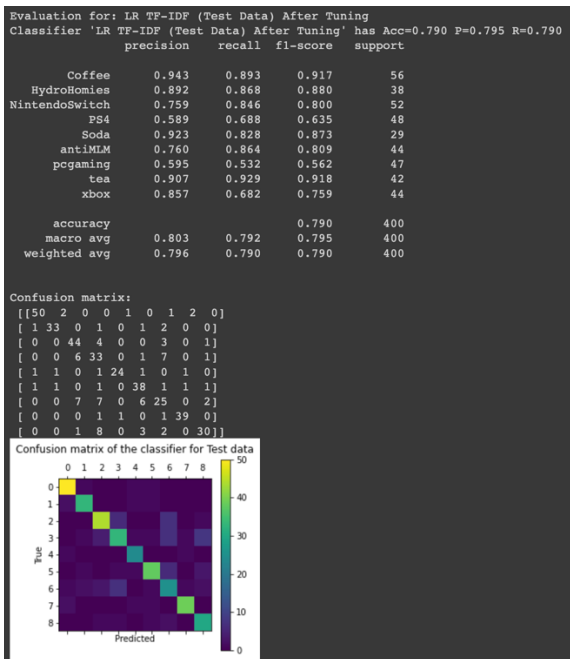


Fig 8: Result of Logistic Regression with TF-IDF. after tuning of the parameters (on test data).

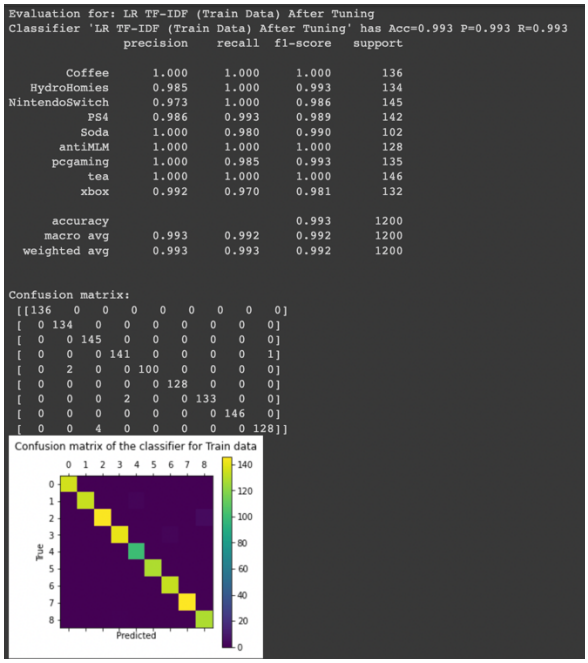


Fig 9: Result of Logistic Regression with TF-IDF after tuning of the parameters (on train data)

LR model with TF-IDF	**Accuracy**	**Precision**	**Recall**	**F1-macro**	**F1-weighted**
Before Tuning	0.770	0.777	0.770	0.772	0.769
After Tuning	0.790	0.795	0.790	0.795	0.790

Table 3: Result comparison before and after tuning of the parameters for LR +TF-IDF model

Q2b: For this part, mismatched labels between the predicted and the actual subreddit labels were calculated and then analysed.

Mismatch has occured for the these labels: ['PS4', 'PS4', 'NintendoSwitch', 'pcgaming', 'NintendoSwitch', 'NintendoSwitch', 'Coffee', 'NintendoSwitch
Count of the labels mimatched: 84

1 to 25 of 84 entries Filter

index	subreddit	predict	body
2	xbox	PS4	I was playing infinite with friends in the party and recorded our gameplay. When I watched my recording, I could not hear myself or my teammates in the video.. what setting can I change so I can hear our voices in future recordings?
3	tea	PS4	Ive only ever tried Ippodo since they have a US based distribution center. Their Kanro gyokuro is the one I've continuously bought and have very much enjoyed it. But I want to try something new to see if I'm missing out on other (possibly higher quality) gyokuros. So do any of you guys know of US based vendors that sell great gyokuro? I'm trying to avoid spending money on international shipping since I only buy 1-2 packages of gyokuro at a time.
8	pcgaming	NintendoSwitch	STAR WARS Jedi Knight - Jedi Academy P5XCR-VY2TY-HLR47 STAR WARS - Knights of the Old Republic CCF78-TBB2B-82W22 Close to the Sun 2B4PL-YLWA5-J9A3F Shiny WZAKR-HAMPG-Y3YM9 GRIP: Combat Racing GAPEV-WGIHN-AYDDF Close to the Sun L4BXE-VZ407-MEC3W STAR WARS™ Jedi Knight II - Jedi Outcast™ 2I7WR-VNCEL-X79EP Codex of Victory 75DBY-R7QJC-YJ0VJ GRIP: Combat Racing WYHAE-8BCYV-B4KWB
13	PS4	pcgaming	Daniel Bloodworth(he reviewed the game for GameTrailers.com) did an awesome Q&A after about 100hrs of gameplay, there's a lot of interesting stuff if you are interested into The Witcher 3 or just can't wait to finally play it yourself. http://www.twitch.tv/gametrailers/v/4972732 Enjoy!
15	PS4	NintendoSwitch	Just heads up if you're interested, not my type of game but I thought it was a good deal if others want it.
16	pcgaming	NintendoSwitch	so basically I'm trying to improve my aim in shooter games, so i started using 3D Aim Trainer, and i was just curious, what are some goals I should try to achieve? like a certain score, avg. hit time, etc
38	antiMLM	Coffee	This is just a small sample. Dreading but also morbidly excited for this party on Sunday that I have to go to - I'll give y'all the deets if I come out alive. Note the 2 unanswered questions, one of which asking what the start up cost is. Dare I say she answered in a...PM 😱
50	pcgaming	NintendoSwitch	I enjoy the Battlefield games a lot. They're my preferred shooter by far. My problem is I tend to suck always dying constantly. I know my biggest problem is positioning and general awareness, but I'm not quite sure how to improve that. I want to try to get better for Battlefield 2042.
56	PS4	NintendoSwitch	I received a code directly from Playstation inviting me to Dragonball fighter z closed beta. I haven't seen another thread about this (hope I searched correctly) so I'm posting here. I will pm the code to anyone who can guess the right number between 1 and 500. If number isn't chosen, I will pick the closest one. Winner will be picked in about 6 hours from now, at around 10-10:30 pm eastern time. Please notify me if there's another thread around with more of these, might make it easier for everyone. Good luck <3 Edit: it seems the code actually gives you more chance to get into beta, info from u/cloudstrifeHY3 . Thanks! ReEdit: I read in the message signing up does not guarantee access, but the code does. So I dunno lol mixed Infos. EDIT: u/Hi-Im-Daisy has won. I sent you a pm so msg me if there are any issues. Congrats :) Number was 379 btw. Google's choice lol.
60	HydroHomies	antiMLM	I'm afraid I talked to my doctor and I'm... Dehydrated
69	Coffee	HydroHomies	My BES870XL has a leaking problem when it tries to make a shot. What parts do I have to replace? Below is the video. https://reddit.com/link/ca7z9c/video/c9lesvwofw831/player https://preview.redd.it/blln3plmlz831.jpg?width=1608&format=pjpg&auto=webp&s=e896f327de8ef788b7dc61250aea8f91eb4a0896 It seems that the part inside red circle is leaking.
70	PS4	pcgaming	I prefer inverted but I feel like all the older FPS games were inverted by default.
73	Coffee	HydroHomies	I'm one of the unlucky Texans going without electricity during a blizzard. I need to make coffee or I'll get a big headache. Right now I have grounds steeping in cool water but I don't think that'll work in time. I need it within about 2 hrs. Hot water from the tap is kinda nasty so I don't want to do that. I thought about using some candles to heat some water just enough for better extraction. I have no tools other than my ceramic dripper since that's the only way I make coffee. Any of yall more experienced people have ideas, what would yall do?
76	pcgaming	antiMLM	Winter-sale20-grogre for 20% off over at greenmangaming.com. I just got Alien isolation for 20$, so it works on items that are already on sale. Wooo!
79	NintendoSwitch	PS4	The Guardian Legend The Magic of Scheherazade Those two are top of my list. Not saying it would or even could happen i just want it to, lol. What old games do you miss?
80	PS4	pcgaming	I was wondering if by now someone had found a way to get full rumble and Sixaxis support to work on a DS4 with PS3 somehow. Maybe something like using a PC as middle man to "emulate" a real DS3. I can't find my DS3 anymore and money is a little short right now, so I want to avoid buying new controllers. And it kinda bothers me that I have a basically fully capable device sitting here but Sony just never bothered to take 5 minutes of their time and add full support for a way superior controller.
82	antiMLM	xbox	Am I the only person that just found this out or am I slow on the uptake?
83	pcgaming	NintendoSwitch	Sonic Forces is probably gonna be a must-get for me, because I love the Unleashed/Colors/Generations gameplay. I could get it on PC...or I could get it on the switch and take it with me. I've liked Binding of Isaac for years despite sucking at it. I never got the expansions. I bought it for the Switch, because I can play it on PC...or I can take it with me. "Take it with me" seems like a powerful fight against my usual "If it's on PC I want it on PC" mindset. Does the portability mean as much to everyone else? EDIT: If your answer is "no I want it for indies and nintendo exclusives" CONGRATULATIONS, your response has been noted. 500 times. You don't have to throw another one into consideration.

Fig 10: Mismatched outputs

Several mismatched labels with the possible reasons are discussed below.

Example 1:-

subreddit	**predict**	**body**
Tea	Coffee	Ive only ever tried Ippodo since they have a US based distribution center. Their Kanro gyokuro is the one I've continuously bought and have very much enjoyed it. But I want to try something new to see if I'm missing out on other (possibly higher quality) gyokuros. So do any of you guys know of US based vendors that sell great gyokuro? I'm trying to avoid spending money on international shipping since I only buy 1-2 packages of gyokuro at a time.

On analysing the above statement, it shows that although the text is talking about the tea but still the model predicts it to be coffee. One of the possible reasons for this might be because of the usage of words like "*Ippodo*" and "*gyokuro*" instead of the word "tea". These are the types/names of Japanese tea. Thus, usage of the common word tea is not there probably because of which the model gets confused and predicts it to be a coffee.

Example 2 :-

subreddit	**predict**	**body**

pcgaming	PS4	I enjoy the Battlefield games a lot. They're my preferred shooter by far. My problem is I tend to suck always dying constantly. I know my biggest problem is positioning and general awareness, but I'm not quite sure how to improve that. I want to try to get better for Battlefield 2042.

On analysing the above statement, it shows that the model predicts the text to be from "*PS4"* label whereas in actual it belongs to the "*pcgaming*" label. The text in the body is talking about a particular game Battlefield. This game battlefield is also available for PS4 gaming consoles so, maybe because of this the model got confused and predicted it to be from PS4 label.

On observing a few more mismatched labels, it can be said that the model is clearly confused whenever there is any text related to gaming.
One of the custom input which was given to predict the subreddit label is given below in the snapshot.

Fig 11: Custom input with predicted result

Q3a:
In the given dataset, apart from "*body*" which contains a small description for every subreddit there are other labels as well which can be used to train the model. So, the features selected for this section of the exercise are *"author"* and *"title"*.

- **"*author*"** is the username of the person who posted. This feature has been selected because there is a possibility of same subreddit type to occur if the user is same.
- ***"title"*** is the title of the post. This feature is selected as this is connected to the "body" feature, so it might help with the better training of model.

Q3b:
For this question, several trials were conducted, and at first single features were added independently with the "*body*" feature just to observe the result analytics. The single features which were added along with the result obtained are discussed below:

- **Added *"title"* only: "**title" feature was added along with "body" feature and then the evaluation result was obtained for Logistic Regression with the tuned parameters and tf-idf vectorizer. The obtained accuracy, precision and recall were 0.810, 0.818 and 0.810 respectively. The f1- score of the model's macro average obtained increased to 0.813 from 0.778

```
title_feature
Evaluation for:
Logistic Regession with TFIDF feature tuning

Classifier '
Logistic Regession with TFIDF feature tuning' has Acc=0.810 P=0.818 R=0.810

                precision    recall  f1-score   support

        Coffee      0.911     1.000     0.953        51
   HydroHomies      0.895     0.850     0.872        40
NintendoSwitch      0.846     0.733     0.786        60
           PS4      0.708     0.596     0.648        57
          Soda      0.897     0.839     0.867        31
       antiMLM      0.841     0.804     0.822        46
     pcgaming       0.574     0.730     0.643        37
           tea      0.929     0.951     0.940        41
          xbox      0.727     0.865     0.790        37

      accuracy                          0.810       400
     macro avg      0.814     0.819     0.813       400
  weighted avg      0.815     0.810     0.809       400


Confusion matrix:
 [[51  0  0  0  0  0  0  0  0]
 [ 1 34  0  0  1  3  0  0  1]
 [ 0  1 44  5  0  0  8  0  2]
 [ 2  1  6 34  0  0  6  0  8]
 [ 2  1  0  0 26  1  0  1  0]
 [ 0  1  0  1  1 37  4  2  0]
 [ 0  0  2  6  0  1 27  0  1]
 [ 0  0  0  0  0  0  2  0 39  0]
 [ 0  0  0  2  1  0  2  0 32]]
```

Fig 12: Results with title feature only

- **Added "*author*" only: "**author" feature was added along with "body" feature and then the evaluation result was obtained for Logistic Regression with the tuned parameters and tf-idf vectorizer. The obtained accuracy, precision and recall were 0.795, 0.798 and 0.795 respectively.

The f1- score of the model's macro average obtained increased to 0.795 from 0.778

```
author_feature
Evaluation for:
Logistic Regession with TFIDF feature tuning

Classifier '
Logistic Regession with TFIDF feature tuning' has Acc=0.795 P=0.798 R=0.795

                precision    recall  f1-score   support

        Coffee      0.893     0.909     0.901        55
   HydroHomies      0.921     0.875     0.897        40
NintendoSwitch      0.808     0.792     0.800        53
           PS4      0.729     0.603     0.660        58
          Soda      0.897     0.867     0.881        30
       antiMLM      0.841     0.787     0.813        47
     pcgaming       0.532     0.625     0.575        40
           tea      0.905     0.905     0.905        42
          xbox      0.682     0.857     0.759        35

      accuracy                          0.795       400
     macro avg      0.801     0.802     0.799       400
  weighted avg      0.802     0.795     0.796       400


Confusion matrix:
 [[50  0  1  0  1  1  0  1  1]
 [ 2 35  0  0  0  3  0  0  0]
 [ 0  0 42  4  0  0  6  0  1]
 [ 0  0  4 35  1  1  9  0  8]
 [ 1  0  1  0 26  0  0  2  0]
 [ 0  1  1  1  0 37  5  0  2]
 [ 1  2  2  7  0  0 25  1  2]
 [ 2  0  0  0  1  1  0 38  0]
 [ 0  0  1  1  0  1  2  0 30]]
```

Fig 13: Results with author feature only

From the results, it can be said that with the addition of a single feature (author or title), it has a good effect on the model's effectiveness as the f1 score of the model for the macro average increases.
Based on these results, if two features are added together, then it might give even better results. Following this idea, both "*title*" and "*author*" were combined with the "*body*" feature and there was a slight increase in the accuracy. The obtained accuracy, precision and recall were 0.818, 0.825 and 0.818 respectively. The f1- score of the model's macro average obtained increased to 0.821 where it was 0.813 and 0.795 when single feature was introduced.

```
combined_feature
Evaluation for:
Logistic Regession with TFIDF feature tuning

Classifier '
Logistic Regession with TFIDF feature tuning' has Acc=0.818 P=0.825 R=0.818

                precision    recall  f1-score   support

        Coffee      0.911     0.981     0.944        52
   HydroHomies      0.868     0.825     0.846        40
NintendoSwitch      0.865     0.726     0.789        62
           PS4      0.688     0.623     0.653        53
          Soda      0.966     0.824     0.889        34
       antiMLM      0.841     0.822     0.831        45
     pcgaming       0.596     0.737     0.659        38
           tea      0.929     0.975     0.951        40
          xbox      0.750     0.917     0.825        36

      accuracy                          0.818       400
     macro avg      0.824     0.825     0.821       400
  weighted avg      0.824     0.818     0.817       400


Confusion matrix:
 [[51  0  0  0  0  0  0  1  0]
 [ 1 33  0  0  0  4  1  0  1]
 [ 0  1 45  5  0  0  8  0  3]
 [ 2  1  5 33  0  0  6  0  6]
 [ 2  1  0  1 28  1  0  1  0]
 [ 0  2  0  1  1 37  3  1  0]
 [ 0  0  2  6  0  1 28  0  1]
 [ 0  0  0  0  0  1  0 39  0]
 [ 0  0  0  2  0  0  1  0 33]]
```

Fig 14: Results with both title and author feature combined

Q3c:
Below table displays the comparison between the performance of the classification with different feature types.

	Accuracy	**Precision**	**Recall**	**F1 macro average**
Single feature added ("title")	0.810	0.818	0.810	0.813
Single feature added ("author")	0.795	0.798	0.795	0.799
Combined feature added (both "title" and "author")	**0.818**	**0.825**	**0.818**	**0.821**

Table 4: Comparison between the result of single and combined features

On comparing the results obtained from that of the single features and the combined features, it can be said that the combination of added features has improved the entire metrics of the model.
It can be observed that one of the feature i.e., "title", leads to increase whereas the other feature

i.e., “author”, leads to decrease in the accuracy as well as the f1-score of the model for the macro average. Thus, due to this uncertainty, combination of both was tried which in result gave a better metrics. Although there is a very slight change observed in the results as of now but to improve such that there is a good change in the metrics, combination of all the features (instead of two) could be made and then tested.

Q4:

Paper Selected: Graph Convolutional Networks for Text Classification:
https://arxiv.org/pdf/1809.05679.pdf

Text classification is one of the oldest problems encountered in natural language processing. To date, there have been several studies geared towards finding a stable and decent solution to this problem. An intermediate step for text classification is text representation, and one of the traditional methods to resolve this issue is by representing the text with handcrafted features. Recent methods to resolve this issue involve convolutional neural networks (CNNs) and recurrent neural networks (RNNs) like LSTM, but there has been a new research direction known as graph neural networks (GCN), which is expected to be a prominent solution to this problem.

Yao L. et al. [1] focused their research on this technique and came up with the idea of a graph neural network. As a part of their research, a large text graph containing word nodes and document nodes was built by them. Their design includes document-word and word-word edges. The weight of each edge was given by the term frequency-inverse document frequency (TF-IDF) of the word in the document and pointwise mutual information (PMI). Following this, the text graph was fed to a two-layered GCN, which provides information exchange access even when there are no document-document edges.

After experimenting with their design model with five different datasets which were all pre-processed by cleaning, tokenization, and removal of stop words, they concluded that Text GCN outperforms every other base model and discussed the effects of parameter sensitivity as well as the size of labelled data on the result.

The first of the two main reasons for the better performance of Text GCN was concluded to be that neural network graphs can learn both document-word and word-word relationships effectively, and the second was Laplacian smoothing, because of which the weighted average of first-order and second-order neighbour information can be used to calculate new features in the GCN model. However, afterwards, it was observed that the text GCN could not outperform CNN and LSTM-based models on the movie review dataset (MR). The prime reason for this turned out to be the ignorance of the sequence of words by the GCN model, whereas, in the CNN and LSTM models, there are explicit word sequences.

The researchers claim that text GCN is capable of learning word embeddings and predictive documents and can give outstanding text classification results. The GCN model is intrinsically transductive, which means that test document nodes are incorporated into GCN training. As a

result, text GCN is unable to swiftly construct embeddings and make predictions for unseen test documents. This is a key disadvantage of this work. Apart from this, another major limitation of their study is that when the data set is large, it generates a very large and complex graph with many edges.

To conclude, apart from generalising the Text GCN model to inductive settings (Hamilton, W et al. [2]), improving classification performance using attention mechanisms (Velickovic et al. [3]) and developing an unsupervised text GCN framework for representation learning on large-scale unlabelled text data are two promising future directions.

References

[1] Yao, L., Mao, C., & Luo, Y. (2019). Graph Convolutional Networks for Text Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 7370-7377. https://doi.org/10.1609/aaai.v33i01.33017370

[2] [Hamilton, Ying, and Leskovec 2017] Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In NIPS, 1024–1034.

[3] [Velickovi ˇ c et al. 2018] Veli ´ ckovi ˇ c, P.; Cucurull, G.; ´ Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. ` Graph attention networks. In ICLR.