MULTI-CLASS TEXT CLASSIFICATION TO IDENTIFY SUBREDDIT OF POSTS

CS 5785 - Applied Machine Learning, Cornell Tech, Fall 2022

James Chu
qjc2@cornell.edu

ABSTRACT

The amount of user generated content is increasing everyday as the general population becomes more digitally active. As a result, users will find it difficult to identify where their content lives, especially for social networks such as Reddit. This analysis aims to develop a multi-class classifier that helps users identify the subreddit that their post belongs to, using classical supervised machine learning algorithms such as Logistic Regression, Linear Support Vector Machines, and Naive Bayes. Our best classifier, trained on 250 subreddit classes was the Linear Support Vector Machine classifier with L2 regularization, resulting in a F1-Score of 0.82576 on our test dataset.

1 INTRODUCTION

Reddit is an online community where users are able to submit content under user-created boards called subreddits. As Reddit grows its user base, the number of subreddits will increase, and each subreddit will become increasingly narrow in its scope of discussions. For example, posts for "Hyundai cars" will spin up a new subreddit dedicated to "Hyundai", rather than being aggregated under a generic "Car" subreddit. Therefore, Reddit users will find it increasingly difficult to locate the appropriate subreddit to post their content. This project will aim to develop a classifier that identifies the subreddit aligned to the content of a user-generated post. We apply several well-known supervised learning algorithms, such as Naive Bayes, Logistic Regression, and Support Vector Machines, on the frequencies of specific words within each post to perform the subreddit classification. The goal is to generate the classifier that results in the highest F1-score.

2 CONTEXT

Previous work on text classification problems have been primarily focused on correctly identifying a small number of classes, for use cases such as sentiment analysis. The problem space for small-class text classification is well studied, and researchers have generated highly accurate classifiers using classical machine learning algorithms such as Logistic Regression and K-Nearest Neighbors, as well as deep learning algorithms such as Recurrent Neural Networks. However, not all classifiers within the small-class domain translate well into the many-class domain. This analysis aims to study the extent learnings from small-class classifiers can be applied to the large-class classifiers, with a focus on classical algorithms due to their to interpretability benefits.

CS 5785, Applied Machine Learning, Cornell Tech
Fall 2022.

3 METHOD

For this project, we apply certain pre-processing steps to 1) reduce the size of the feature space, so to make the experiments computationally efficient and 2) reduce the noise within the features themselves. The pre-processed data will then leverage three machine learning algorithms: Logistic Regression, Naive Bayes Classifier, and Linear Support Vector Machines.

3.1 Data Pre-processing

For this project, we will be using this labelled dataset from Kaggle with 1.013M rows containing the title, content, and subreddit class of each post. The dataset is evenly distributed with 1,000 posts from 1,013 subreddit classes. As this dataset is extremely large, we will only use subsets of the data for experimentation due to hardware constraints. Our initial experiments will leverage a small subset consisting of 25,000 posts that correspond to 25 subreddit classes. We leverage our experiment findings from the smaller subset and apply them to a larger subset with a 10x increase in size: 250,000 posts that correspond to 250 subreddits. For each experiment, we use a 80-20 split for training and testing datasets.

Once the subsets of data have been created, we perform standard pre-processing steps on the posts to reduce noise within the dataset, which includes converting all words to lowercase, removing punctuation and stop words, and lemmatizing all the words to their root words. To reduce the feature space, we also apply a threshold to only include words that appear more than a specific number of times. This is explored further in the "Experiment Evaluation" section. The pre-processed posts are then converted into binary and non-binary feature vectors before being used for training our classifiers.

3.2 Logistic Regression

Logistic regression is a binary supervised learning classification algorithm that produces linear decision boundaries. In the case of multi-class classification, we utilize the "one-vs-rest" approach where we split the multi-class problem into multiple binary classification problems. In its binary form, the model for logistic regression takes the form:

$$f_\theta(x) = \sigma(\theta^\top x) = \frac{1}{1 + \exp(-\theta^\top x)},$$

where

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

is the sigmoid function. We can modify the logistic regression algorithm by applying regularization terms in our objective function to reduce overfitting by penalizing overreliance on any single feature. We will be performing both L1 (Lasso) regularization as well as L2

(Ridge) regularization, which adds the L1 Norm ($\lambda \cdot ||\theta||_1$.) and the L2 Norm ($\frac{\lambda}{2} \cdot ||\theta||_2^2$. to our model's objective function, respectively.

3.3 Naive Bayes Classifier

Naive Bayes is a generative supervised classification algorithm that leverages Bayes theorem and operates under the assumption that the features of the training set are independent. The Bernoulli Naive Bayes model $P_\theta(x, y)$ is defined for binary data $x \in \{0, 1\}^d$. The θ contains prior parameters $\vec{\phi} = (\phi_1, ..., \phi_K)$ and K sets of per-class parameters $\psi_k = (\psi_{1k}, ..., \psi_{dk})$. The probability of the data x for each class equals

$$P_\theta(x|y = k) = \prod_{j=1}^{d} P(x_j \mid y = k),$$

where each $P_\theta(x_j \mid y = k)$ is a Bernoulli(ψ_{jk}). The probability over y is Categorical: $P_\theta(y = k) = \phi_k$. This concept can be extended to a classifier called Multinomial Naive Bayes. Whereas Bernoulli Naive Bayes classifies based on the presence of a single feature, Multinomial Naive Bayes classifies based on the counts of each feature.

3.4 Linear Support Vector Machines

Linear Support Vector Machines (SVM) is a binary supervised classification algorithm that creates a hyperplane that separates the dataset into classes by maximizing the margin. We define a linear model of the form:

$$f_\theta(x) = \theta^\top x + \theta_0.$$

where $x \in \mathbb{R}^d$ is a vector of features and $y \in \{-1, 1\}$ is the target. $\theta^\top$ and θ_0 are the parameters of the model. The geometric margin $\gamma^{(i)}$ with respect to a training example $(x^{(i)}, y^{(i)})$ is defined as

$$\gamma^{(i)} = y^{(i)} \left(\frac{\theta^\top x^{(i)} + \theta_0}{||\theta||} \right).$$

This also corresponds to the distance from $x^{(i)}$ to the hyperplane. We want to maximize the margin such that the distance of $x^{(i)}$ to the hyperplane is at least γ, which leads us to the following optimization problem for the Linear SVM:

$$\max_{\theta, \theta_0, \gamma} \gamma$$

$$\text{subject to } y^{(i)} \frac{(x^{(i)})^\top \theta + \theta_0}{||\theta||} \geq \gamma \text{ for all } i$$

Similar to logistic regression, we can utilize the "one-vs-rest" approach to apply Linear SVMs to multi-class problems. We can also apply L1 and L2 regularization to reduce overfitting by allowing the classifier to identify a separating hyperplane with larger margins.

4 EXPERIMENT EVALUATION - 25 CLASSES

We perform some initial experiments on a smaller subset of data containing 25 classes to gain insight on how to handle the larger subset containing 250 classes.

4.1 Metrics Evaluation

For our experiments, we chose the F1-Score as our evaluation metric. The F1-Score is defined as the harmonic mean of precision and recall:

$$\text{F-Score} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

F1-Score was chosen because both false positives and false negatives are equally undesirable, and our intent is to maximize both precision and recall.

4.2 Word Frequency Threshold

As part of our pre-processing steps, we chose a threshold (M) for the frequency of each word based on its impact on the F1-Score of our test set. This was performed across all of the supervised classifiers used during this initial phase. As shown in the illustrative example below for Logistic Regression, increasing M results in a marginal decrease in the test performance. Therefore, we chose a value of M that reduced our vocabulary to approximately 5-6k words due to hardware constraints that prevented us from processing extremely large feature spaces.

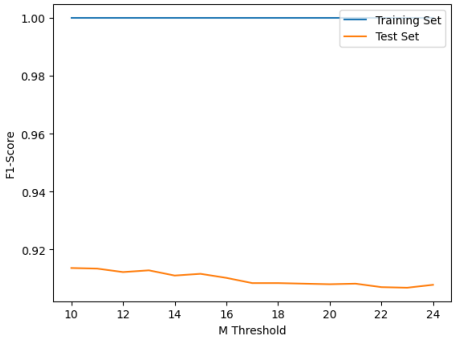

Figure 1: F1-Scores for Logistic Regression with Different M Thresholds for 25 Classes

4.3 Binary vs. Non-Binary Feature Vectors

We first utilized the binary representation of our feature space, which represents the presence of each specific word in the post. Using these binary feature vectors, we ran Logistic Regression with and without regularization terms, as well as Bernoulli Naive Bayes. The regularization strength for Logistic Regression with L1 and L2 regularization terms was set to the default of 1.

Classifier	M	Train F-1	Test F-1
Logistic Regression (no reg.)	20	1.0	0.9204
Logistic Regression (L1 reg.)	20	0.9936	0.9208
Logistic Regression (L2 reg.)	20	0.9998	0.9272
Bernoulli Naive Bayes	20	0.9156	0.8782

Table 1: F1-Scores for Logistic Regression and Bernoulli Naive Bayes with Binary Feature Vectors

We compare this against the same classifiers, but leveraging non-binary feature vectors that take into account the number of occurrences of each specific word in the post. In this case, we are unable

to utilize Bernoulli Naive Bayes as the features are no longer binary, and instead switch over to the Multinomial Naive Bayes classifier.

Classifier	M	Train F-1	Test F-1
Logistic Regression (no reg.)	20	1.0	0.908
Logistic Regression (L1 reg.)	20	0.9959	0.9166
Logistic Regression (L2 reg.)	20	0.9998	0.9226
Multinomial Naive Bayes	20	0.9520	0.9228

Table 2: F1-Scores for Logistic Regression and Multinomial Naive Bayes with Non-binary Feature Vectors

Across all the classifiers, we observe strong performance on the test set with F1-Scores > 90%. However, we also notice a overfitting effect, where the delta between the Train F-1 Score and Test F-1 Score ranges between 3 - 10 percentage points. Overfitting is reduced when L1 and L2 regularization terms are added, which indicates the need for high degrees of regularization when we extend to the larger dataset.

Non-binary features also decreased the Test F1-Score across the board compared to binary features, with the exception of the Multinomial Naive Bayes classifier. This indicates that the presence of a specific word is sufficient to classify the subreddit of a post. Or in other words, the topics of each subreddit were distinct enough that there are unique words only used within each subreddit. As a result, when extending to the larger dataset, we will experiment solely with binary features. Within our binary features experiment, we also decide at this point to discontinue experimenting with the Bernoulli Naive Bayes classifier as it performed the worst on the test set. Furthermore, with additional classes and an expanded feature space with the larger dataset, the independence assumption that underlies Naive Bayes may fall apart.

4.4 (1,2)-Grams

We also experiment with (1,2)-grams to determine if a larger feature space comprising of singular words and pairs of words would offer any benefit to performance.

Classifier	M	Train F-1	Test F-1
Logistic Regression (no reg.)	40	1.0	0.9174
Logistic Regression (L1 reg.)	40	0.995	0.9124
Logistic Regression (L2 reg.)	40	1.0	0.9186
Bernoulli Naive Bayes	40	0.9028	0.8468
Logistic Regression (no reg.)	40	1.0	0.9078
Logistic Regression (L1 reg.)	40	0.9967	0.9104
Logistic Regression (L2 reg.)	40	1.0	0.915
Multinomial Naive Bayes	40	0.9458	0.9122

Table 3: F1-Scores for Logistic Regression and Naive Bayes with Binary and Non-binary Feature Vectors (with (1,2)-grams)

In all cases, Test F-1 Scores decreased, which indicates that singular words were more impactful in identifying a post's subreddit.

5 EXPERIMENT EVALUATION - 250 CLASSES

From our analysis using the 25 classes subset, we proceed with binary features and Logistic Regression as our baseline for our 250 classes subset. We apply varying degrees of regularization to assess if the overfitting effect is reduced. We also experiment with Linear SVMs on the larger dataset as SVMs are generally less prone to overfitting and can also be tuned using L1 and L2 regularization terms.

5.1 Word Frequency Threshold

As with the smaller dataset, we also implement a threshold (M) for the frequency of each word based on its impact on the test F1-Score using a Logistic Regression classifier baseline. In this case, we have decided to implement a singular value of M across all of our classifiers: 100. Again, this was largely driven by hardware constraints and our desire to limit the feature space. Unlike with the smaller subset, we can observe a strong negative linear relationship between increasing M and the test F1-Score. However, the decrease between setting a M threshold of 20 vs. 100 is only 5 percentage points, which validates our previous finding that there are certain frequently used words within each subreddit class that are unique to that subreddit.

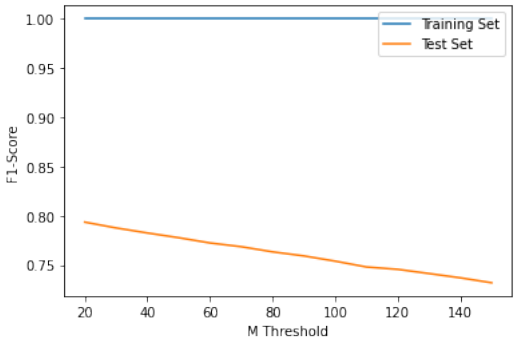

Figure 2: F1-Scores for Logistic Regression with Different M Thresholds for 250 Classes

5.2 Logistic Regression with Regularization

We establish our baseline using Logistic Regression and no regularization terms, which produced a test F-1 Score of 0.75386. We apply varying degrees of L1 and L2 regularization with Logistic Regression. The results are represented in the figure below:

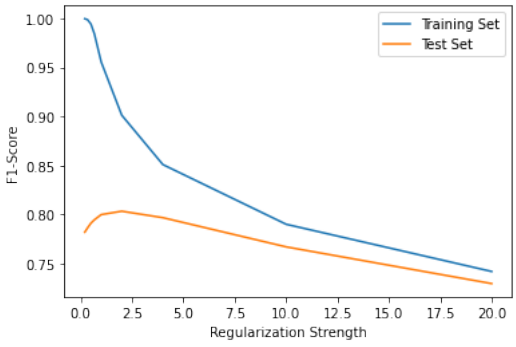

Figure 3: F1-Scores for Logistic Regression with L1 Regularization

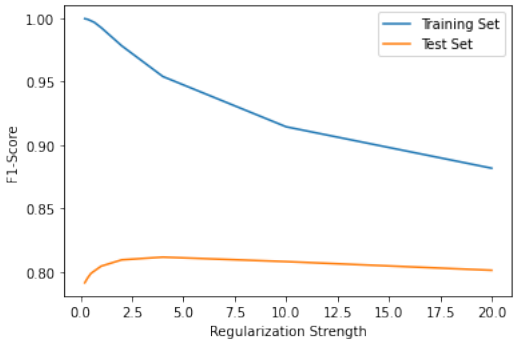

Figure 4: F1-Scores for Logistic Regression with L2 Regularization

As expected, increasing the number of classes dramatically decreases the F1-Score on the test datasets. This is likely because the new subreddit classes that we introduced into the model have overlap in terms of topics and words used, which validates our decision to discontinue the use of Naive Bayes classifiers. The test F1-Score for Logistic Regression with L2 regularization was maximized at 0.81164 when the regularization strength was set to 4, while the test F1-Score for L1 regularization was maximized at 0.80314 when the regularization strength was set to 2. In general, we observe that strong regularization is a necessary component to improving classifier performance. Without regularization terms, these classifiers will over-rely on specific words to identify the subreddits. This produces high training F-1 Scores, but does not generalize well enough for the unseen test data. Furthermore, since L1 regularization generates a sparse solution that reduces the coefficients of certain words to zero, the test performance is marginally worse than that of L2 regularization.

5.3 Linear SVMs with Regularization

We apply Linear SVMs to our dataset, also with varying degrees of L1 and L2 regularization and achieve the following results:

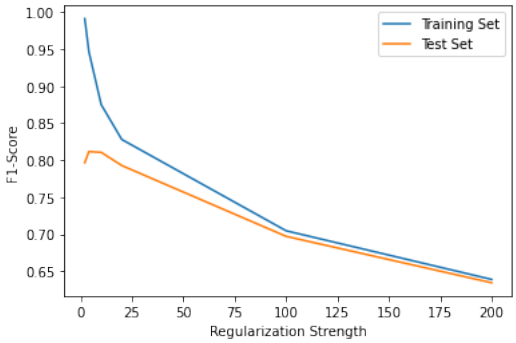

Figure 5: F1-Scores for Linear SVMs with L1 Regularization

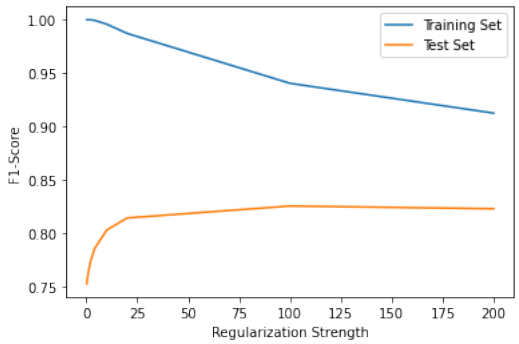

Figure 6: F1-Scores for Linear SVMs with L2 Regularization

Here, the test F1-Score for Linear SVMs with L1 regularization was maximized at 0.81164 when the regularization strength was set to 4, while the test F1-Score with L2 regularization was maximized at 0.82576 when the regularization strength was set to 100. This was our best result yet, and may be explained by the fact that SVMs attempt to produce the optimum decision boundary, while Logistic Regression may produce a sub-optimal decision boundary. We also observe that Linear SVMs were also less computationally expensive when generating a model.

6 CONCLUSION

In our pursuit of creating an accurate subreddit classifier, we first isolated a small subset containing 25 classes. Our initial experiments using Logistic Regression and Naive Bayes on this smaller subset provided guidance on how to proceed with the larger dataset. Notably, binary feature vectors performed better than non-binary feature vectors, and regularization was key to reducing overfitting and increasing test performance. Using these learnings, we applied Logistic Regression and Linear SVMs to the larger dataset, and focused on tuning the regularization strength to improve the model performance. Our best performing model was the Linear Support Vector Machine classifier with L2 regularization, resulting in a F1-Score of 0.82576 on our test dataset.

For future experimentation, we can consider using the entire dataset consisting of 1,013 subreddit classes and determine if our best model is still performant. Within the linear models, we can also experiment with alternative regularization methods, such as elastic net regularization that combines both L1 and L2 regularization. Furthermore, we can continue experimentation with deep learning algorithms, which have had considerable success in the last few years within the text classification space, at the cost of model interpretability.