Blurb Genre Classification

GitHub : https://github.com/vinceniko/GenreClassification

Vincent Nikolayev	Conner Moore

vn500@nyu.edu	cm4841@nyu.edu

Abstract / Motivation

The motivation for our project is to classify books into genres based on their accompanying blurbs (a short summary used as advertising for the book). We believed this to be possible due to the specific vocabulary often used exclusively, or at a higher rate, within certain genres over others. We researched existing work done on text genre classification and found a dataset compiled by the University of Hamburg to be particularly useful for assessing our hypothesis. Using this dataset as a foundation we used various experimental designs, performed feature selection, and tested several Machine Learning models in order to systematically improve the quality of our analysis, our understanding of the underlying data, and ultimately, our ability to accurately predict on interesting and diverse genre classes.

Data Source

Our dataset is composed of a collection of book summaries (blurbs) and the genres of literature they have been categorized as by the publisher and the group at University of Hamburg. The blurbs and their respective categories were originally taken from the Penguin Random House webpage, and PRH therefore holds the copyrights to the dataset used in our work. These blurbs were compiled into XML format by the University of Hamburg. Using this XML file we converted the data first into a list of books represented as OrderedDict objects (using the xmltodict Python library) and then into a Pandas DataFrame for easier manipulation.

Classes

The group at the University of Hamburg expanded upon the genres originally assigned by the publisher, Penguin Random House, creating a hierarchical class structure. They assigned every blurb one of seven “d0”s, the highest level of the hierarchy corresponding to the most general genre type (Non-fiction, Fiction, Children’s Books, etc), along with one or more “d1”s, for more specific subgenres, and even “d2”s, for subcategories of these subgenres. For the purpose of our project, we stripped away this hierarchy and used a singular class assignment per blurb.

Feature Selection and Feature Extraction

Using a bag of words model coupled with Tf-Idf measures, each unique word counted as a separate input feature to our model. Without feature reduction we originally had far more features than samples. We initially started with 156,170 parameters and only 86,708 samples. Besides the risk of high variance error, having such a large number of parameters proved to be computationally slow and therefore prohibitive for our need for fast project iteration. Thus, our first most pressing issue was to substantially reduce the total number of features.

The first form of feature selection we utilized was the removal of stop-words. There is no universal standard list of English stop words, so we analyzed several of the most frequently used lists, such as Scikit-Learn's and NLTK's. We found the list compiled by Scikit-Learn to be too greedy in our interpretation, including words such as "against" which we thought might be useful in our model, and opted instead to use NLTK's more compact list.

Another form of feature selection we utilized was altering the parameters of our vectorizer. Within this function we set the min_df to 20. This had the effect of removing terms that appeared in less than 20 documents throughout the text corpus. min_df was set in order to remove words that were incredibly sparse throughout the text corpus, and therefore not useful for class prediction. We chose 18,000 as an upper bound by first examining how many pixels would appear in a 128x128 image, and chose a slight overestimation of this value. After the removal of stop words and the min_df restriction, the parameters were reduced to 18,000 in the d0 unbalanced and d1 unbalanced experiments and to 17,035 for the final d1 balanced experiment.

We intended to use PCA to greatly reduce the number of features, and through experimentation, realized we instead needed to utilize the similar Singular Value Decomposition (SVD) algorithm as our feature matrix was very sparse. This is because most features (words) appear in a small subset of the total documents. We found a "sweet spot" of 100 components for SVD, which proved to be a good compromise between accuracy and efficiency. Picking a larger number of components provided a marginal increase to the overall accuracy of our models (a gain of ~1% for 1000 components), but was computationally prohibitive, increasing the execution time linearly (i.e. 1000 components made the models 10 times slower than with 100 components).

Model Selection

The majority of the models we built were SVMs. SVMs provided a good compromise between high accuracy on the nonlinear data, computation speed (if PCA / SVD was performed) and as a result, ease of iteration over several experimental designs. Once we decided on a primary experiment (based on its superior interest and value to the hypothesis) and performed sufficient feature selection with SVMs, we implemented a neural network for a performance comparison.

We also experimented with Complement Naive Bayes on the same dataset. This model was even less computationally intensive than SVM, but proved markedly less accurate, producing an accuracy roughly 4% below the SVM. It did provide a decent benchmark for feature selection changes we had to quickly iterate on such as trying a range of values.

SVM (d0 Unbalanced)

The first model we built was a support vector machine trained on the 7 d0 genres, with the largest being Fiction and Nonfiction. These classes were largely unbalanced, and as a result, produced a highly accurate, albeit less interesting, model. The top 3 classes of nonfiction, fiction, and children's books with a total sample size of ~80,000 total samples dominated the rest of the 4, much smaller classes, that had only 4,000 total samples. The test accuracy for this model was roughly 89%. Although the accuracy was high, it tended to misclassify to the aforementioned top 3 classes, which is likely evidence of overfitting and high generalization error. As seen in the heatmap in figure 1, some common misclassifications include Classics as Nonfiction, Humor as Nonfiction, and Teen & Young Adult as Fiction. This, coupled with the lack of perceived value of classifying on such broad categories as nonfiction or fiction, led us to redesign our experiment to replace them with their respective subgenres.

SVM (d1 Unbalanced)

Our next step in the project was to replace the largest d0 classes, Fiction and Nonfiction, with their respective subclasses. This greatly increased the total number of target classes from the initial 7 d0s, to 35 total classes composed of the 5 other d0s, as well as 30 new d1 classes. These classes ranged largely in size, and were largely imbalanced. This model was expectedly less accurate, with an overall test accuracy of roughly 77.5%. The imbalance of the classes came into play here as well, as classes with the smallest number of samples performed poorly on average. However, interesting observations began to emerge, where we could see misclassifications actually indicating semantically similar genres. For example, Psychology was misclassified as Religion & Philosophy 20% of the time. Another important observation was that classes with small sample sizes performed worse. For example, Military Fiction which had a sample size of 42 had an accuracy of 0% percent and was misclassified as Children's Books 71% of the time.

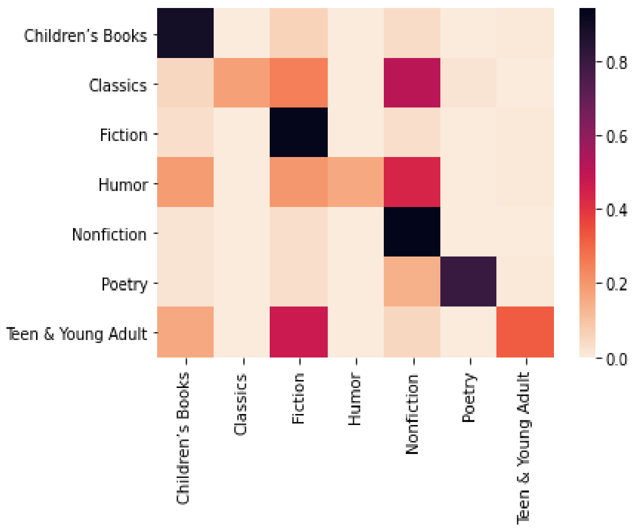


Figure 1: A heatmap visualization of the confusion matrix

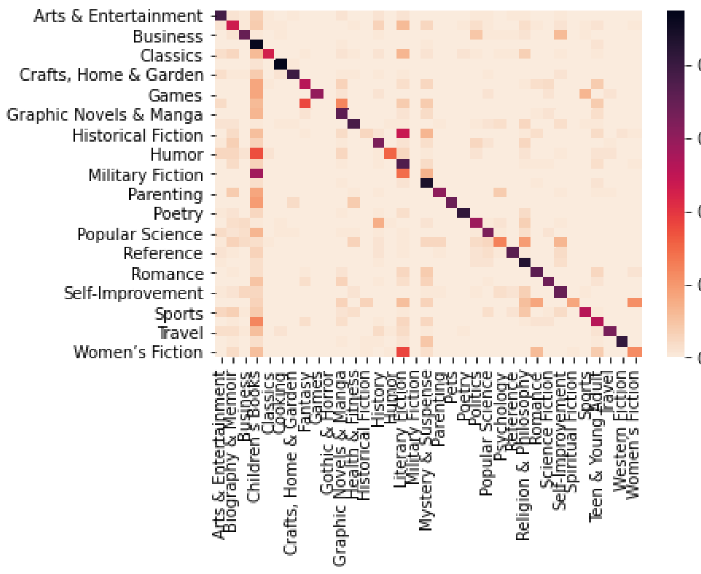


Figure 2: A heatmap visualization of the confusion matrix

SVM (d1 Balanced)

Given our findings in the previous two experimental designs— where unequal class sizes contributed to either disproportionately high accuracies or low accuracies, we decided to balance the class sizes to improve our findings. In particular, the largest class of 19,499 children's books dominated the smaller classes, many of which had sample sizes barely over 1000.

For this reason, for the final SVM and Neural Network experiments, we removed the few classes with the smallest number of samples (below 1,116 samples) and reduced the size of all classes above this threshold to 1,116 samples, completely balancing the remaining 15 classes.

The total accuracy over the test data for this model was 78.69%. The least accurately predicted class was Teen & Young Adult with 60% correctly classified, and the most accurate was Cooking with an accuracy of 96%. For most misclassifications, the model usually appears to have made a reasonable guess at the target, with a sensible degree of similarity existing between the incorrectly guessed class and the true class. For example, Religion and Philosophy's second most predicted class was Self-Improvement—two very related fields. Similarly, Science's Fiction's second most predicted class was Fantasy; Biography & Memoirs' was History; Children's Books' was Teen & Young-Adult; and Graphic Novels & Manga's was Fantasy. Across these pairs the genre's are semantically very similar, and they likely share not only many of the same topics, but also identical words.

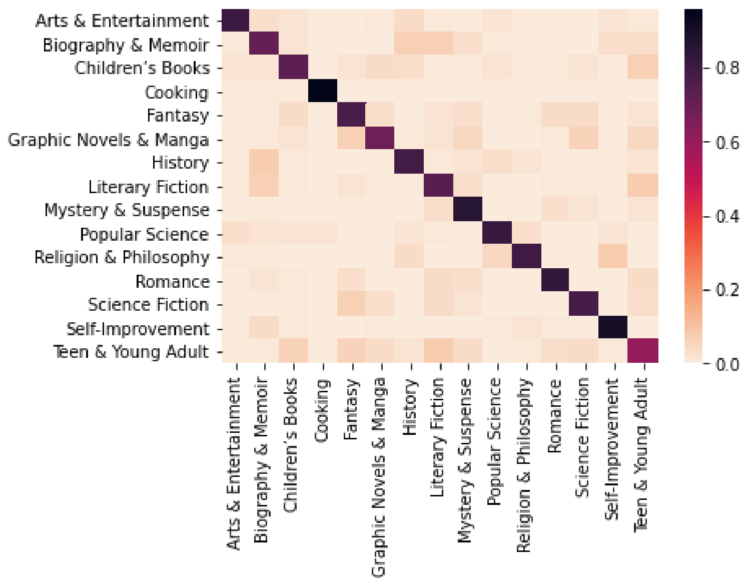
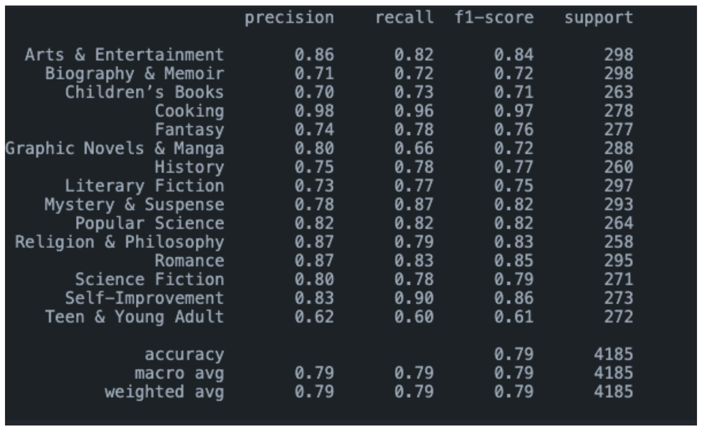


Figure 3: A heatmap visualization of the confusion matrix



```
                        precision    recall  f1-score   support

  Arts & Entertainment       0.86      0.82      0.84       298
    Biography & Memoir       0.71      0.72      0.72       298
      Children's Books       0.70      0.73      0.71       263
               Cooking       0.98      0.96      0.97       278
               Fantasy       0.74      0.78      0.76       277
Graphic Novels & Manga       0.80      0.66      0.72       288
               History       0.75      0.78      0.77       260
      Literary Fiction       0.73      0.77      0.75       297
    Mystery & Suspense       0.78      0.87      0.82       293
       Popular Science       0.82      0.82      0.82       264
 Religion & Philosophy       0.87      0.79      0.83       258
               Romance       0.87      0.83      0.85       295
       Science Fiction       0.80      0.78      0.79       271
      Self-Improvement       0.83      0.90      0.86       273
    Teen & Young Adult       0.62      0.60      0.61       272

              accuracy                           0.79      4185
             macro avg       0.79      0.79      0.79      4185
          weighted avg       0.79      0.79      0.79      4185
```

Figure 4: Scikit Learn Classification Report

Neural Network

After completing analysis on our last SVM model, we decided to experiment with a neural network. The data provided to this simple network was the same as our last SVM model, with 15 equal sized classes composed of 1,116 samples. We utilized the Adam optimizer, and through cross validation found we got the best results with a learning rate of 0.0001. As our project is a multi-classification problem, we utilized the Sparse Categorical Cross Entropy Loss function. The neural network is composed of two dense layers and has 870,015 trainable parameters. The results produced by this neural network varied between 77% and 79% test accuracy, remarkably similar to the results of our SVM performance. More specifically, the neural network made many of the same misclassifications as the SVM model, as can be seen in the heatmap shown in figure 6. We can therefore infer that there is some natural variance that cannot be accounted for in these models (i.e. irreducible error). Please refer to the appendix for further implementation details of the Neural Network.

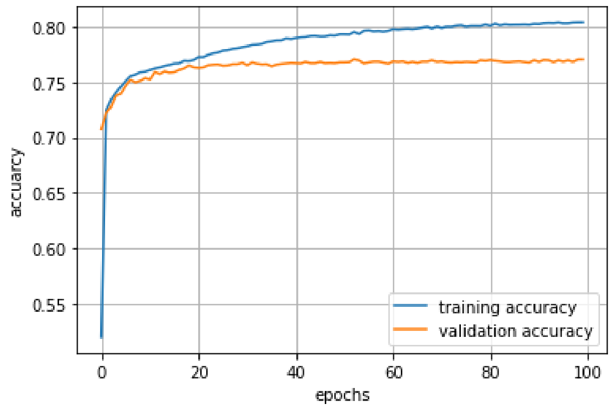


Figure 5: Train and Test Accuracy over 100 Epochs

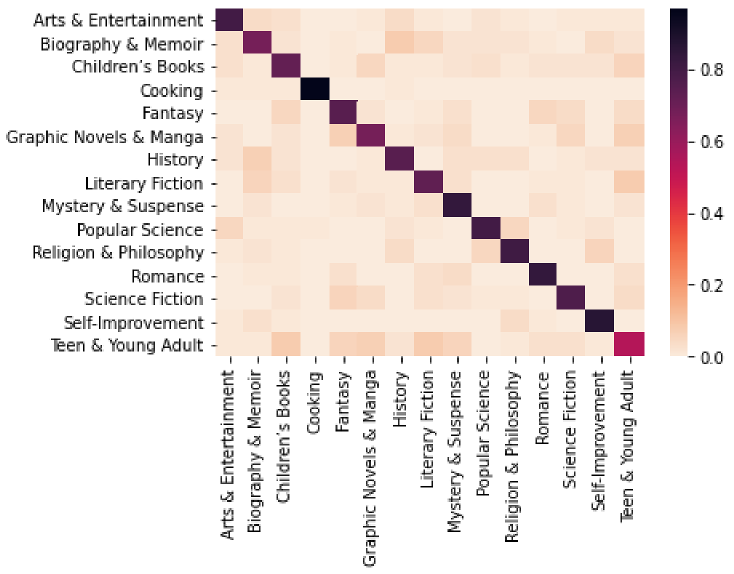


Figure 6: A heatmap visualization of the confusion matrix

Conclusion

Though this was meant as just an extra credit assignment, we believe that there are clear real world ramifications found in the data. Humans assigned genres to the books (and therefore blurbs) that we classified on. The maximum accuracy we achieved using both a SVM and a Neural Network likely indicates an irreducible error that, to some extent, results from the ambiguous choices that are made by humans on whether a textual work belongs to one genre or another. Is it not arbitrary to call Pride and Prejudice a work of literary fiction as opposed to a work of romance? Such genre-blending works can reveal that genres cannot accurately be classified using multi-class classification techniques, and that further experimentation with clustering (using an algorithm like K-Means) may reveal further cross-genre classifications that exist beyond what was defined in the dataset we used.

References

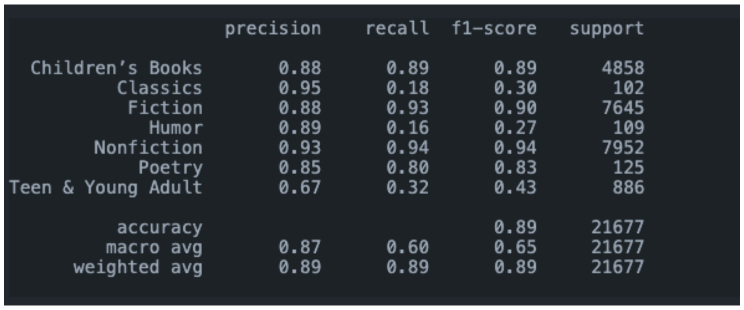
[1] Aly, R. (2018, October 18). Hierarchical writing genre classification with neural networks. https://www.inf.uni-hamburg.de/en/inst/ab/lt/teaching/theses/completed-theses/2018-ba-aly-blurbs.pdf

[2] Rangan, S. (2020). *Introduction to Machine Learning*. Lecture presented at Labs Unit 9, 11, and 12.

Appendix:

SVM Analysis:

SVM D0 Unbalanced:

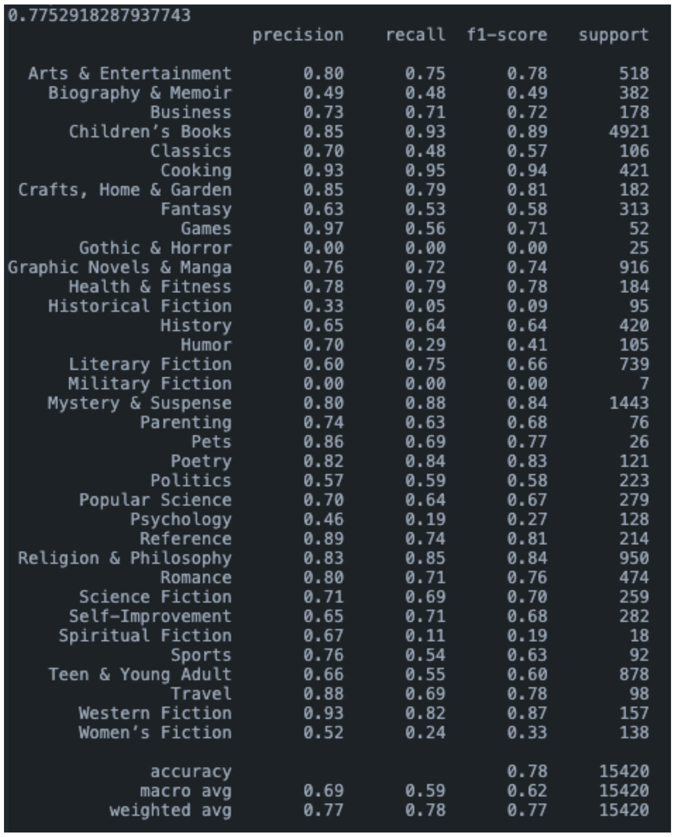


```
                     precision    recall  f1-score   support

  Children’s Books        0.88      0.89      0.89      4858
          Classics        0.95      0.18      0.30       102
           Fiction        0.88      0.93      0.90      7645
             Humor        0.89      0.16      0.27       109
        Nonfiction        0.93      0.94      0.94      7952
            Poetry        0.85      0.80      0.83       125
Teen & Young Adult        0.67      0.32      0.43       886

          accuracy                            0.89     21677
         macro avg        0.87      0.60      0.65     21677
      weighted avg        0.89      0.89      0.89     21677
```

SVM(d1) Unbalanced Classification Report:



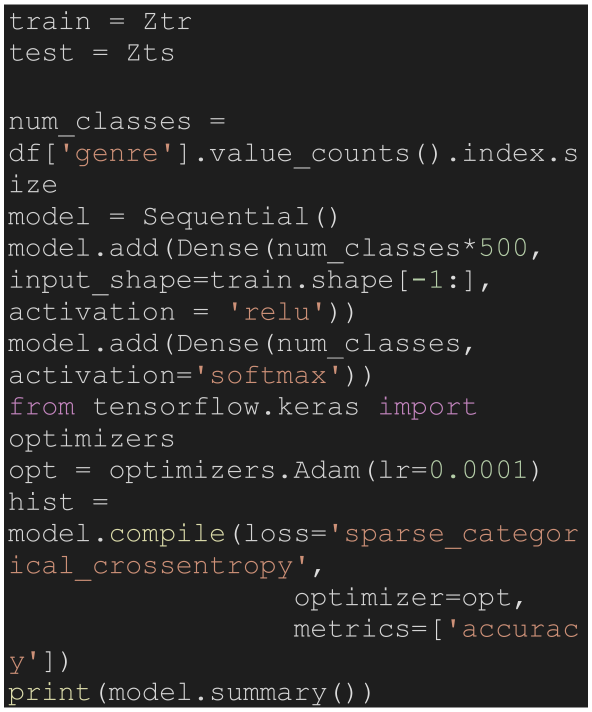
```
0.7752918287937743
                          precision    recall  f1-score   support

   Arts & Entertainment        0.80      0.75      0.78       518
     Biography & Memoir        0.49      0.48      0.49       382
               Business        0.73      0.71      0.72       178
       Children’s Books        0.85      0.93      0.89      4921
               Classics        0.70      0.48      0.57       106
                Cooking        0.93      0.95      0.94       421
  Crafts, Home & Garden        0.85      0.79      0.81       182
                Fantasy        0.63      0.53      0.58       313
                  Games        0.97      0.56      0.71        52
         Gothic & Horror       0.00      0.00      0.00        25
Graphic Novels & Manga         0.76      0.72      0.74       916
       Health & Fitness        0.78      0.79      0.78       184
     Historical Fiction        0.33      0.05      0.09        95
                History        0.65      0.64      0.64       420
                  Humor        0.70      0.29      0.41       105
       Literary Fiction        0.60      0.75      0.66       739
       Military Fiction        0.00      0.00      0.00         7
     Mystery & Suspense        0.80      0.88      0.84      1443
              Parenting        0.74      0.63      0.68        76
                   Pets        0.86      0.69      0.77        26
                 Poetry        0.82      0.84      0.83       121
               Politics        0.57      0.59      0.58       223
        Popular Science        0.70      0.64      0.67       279
             Psychology        0.46      0.19      0.27       128
              Reference        0.89      0.74      0.81       214
  Religion & Philosophy        0.83      0.85      0.84       950
                Romance        0.80      0.71      0.76       474
        Science Fiction        0.71      0.69      0.70       259
       Self-Improvement        0.65      0.71      0.68       282
      Spiritual Fiction        0.67      0.11      0.19        18
                 Sports        0.76      0.54      0.63        92
     Teen & Young Adult        0.66      0.55      0.60       878
                 Travel        0.88      0.69      0.78        98
        Western Fiction        0.93      0.82      0.87       157
         Women’s Fiction       0.52      0.24      0.33       138

               accuracy                            0.78     15420
              macro avg        0.69      0.59      0.62     15420
           weighted avg        0.77      0.78      0.77     15420
```

Neural Network Analysis:

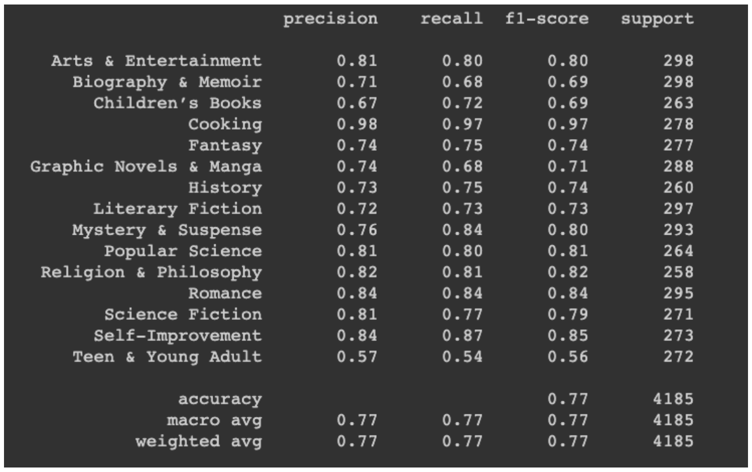
Setup:



```
train = Ztr
test = Zts

num_classes =
df['genre'].value_counts().index.s
ize
model = Sequential()
model.add(Dense(num_classes*500,
input_shape=train.shape[-1:],
activation = 'relu'))
model.add(Dense(num_classes,
activation='softmax'))
from tensorflow.keras import
optimizers
opt = optimizers.Adam(lr=0.0001)
hist =
model.compile(loss='sparse_categor
ical_crossentropy',
                  optimizer=opt,
                  metrics=['accurac
y'])
print(model.summary())
```

Sklearn Classification Report:

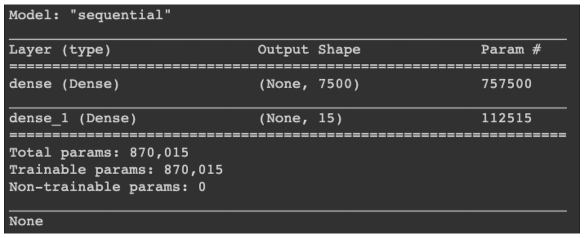


```
                         precision    recall  f1-score   support

   Arts & Entertainment       0.81      0.80      0.80       298
    Biography & Memoir        0.71      0.68      0.69       298
      Children's Books        0.67      0.72      0.69       263
               Cooking        0.98      0.97      0.97       278
               Fantasy        0.74      0.75      0.74       277
 Graphic Novels & Manga       0.74      0.68      0.71       288
               History        0.73      0.75      0.74       260
       Literary Fiction       0.72      0.73      0.73       297
     Mystery & Suspense       0.76      0.84      0.80       293
       Popular Science        0.81      0.80      0.81       264
  Religion & Philosophy       0.82      0.81      0.82       258
               Romance        0.84      0.84      0.84       295
        Science Fiction       0.81      0.77      0.79       271
       Self-Improvement       0.84      0.87      0.85       273
     Teen & Young Adult       0.57      0.54      0.56       272

               accuracy                           0.77      4185
              macro avg       0.77      0.77      0.77      4185
           weighted avg       0.77      0.77      0.77      4185
```

Model Summary



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 7500)              757500
_________________________________________________________________
dense_1 (Dense)              (None, 15)                112515
=================================================================
Total params: 870,015
Trainable params: 870,015
Non-trainable params: 0
_________________________________________________________________
None
```