Abstract:

In the project, we evaluate several classification methods, including the logistics regression model, multiclass regression, and KNN on two datasets to investigate and evaluate the classification performance. We first process the dataset, construct the methods, and implement them on the dataset to train the model. We investigate the performance of several classification methods on the two datasets, and our key takeaways include follows. Logistics regression performs better on datasets with a binary target variable; multiclass regression performs better on datasets with multiple categories target variable; a large number of features leads to bad performance for KNN in both cases; the size of the training dataset could slightly improve the model performance.

Introduction:

In the assignment, we are assigned the tasks to implement three methods, logistics regression, multiclass regression, and KNN on datasets to train the model and predict the class of the target variables.

We are given two textual datasets: IMDb Ratings and NewGroups4. The first dataset contains 500,00 text reviews, and its target variable is the reviewer's attitude; the second dataset contains texts from four categories, which also works as its target variable.

After investigations and experiments, we have the following findings: 1. For datasets with the binary target variable and a relatively large number of features, the logistics model has higher accuracy and shorter running time than clustering methods like KNN; 2. Implementing multiclass regression for datasets with multiple classes and a large set of features also generates more accurate results in a shorter time than the KNN method; 3. The unsatisfactory performance of the KNN method on both datasets could be due to the large number of features; 4. With the size of the training set increasing (from 20% to 100%), the accuracy rate of the model tends to have a slight improvement.

Dataset:

We are given two datasets in the assignment: IMDb Ratings and NewGroups.

The IMDb Rating training dataset contains 25,000 reviews total, with 12,500 reviews labeled negative and the rest as positive. Every review has its rating scores and textual review contents, where the author assigns the reviews with scores larger than 7 as "positive" and the reviews with less than 4 as "negative". Besides, it contains .vocab files recording the Bag of words. There are 89526 words in the Bag of Words, and each is assigned an index. The dataset also includes a .feat file to record the rating and appearance number of each word in each review. There are 25000 records in the .feat file; each refers to a text review's information. The distribution of the dataset is balanced, with ½ (12500/25000) being positive reviews and the rest being negative reviews.

To process the data, we first use the .feat and .vocab files to vectorize the entire training data. Here we generate a dataframe with columns as the word index and rows names as the

review index. Then we select features(words) that would show in the model. To achieve this, we count the word occurrence in all the reviews and remove the stopwords words by selecting words that only show up in 1%-50% of the total reviews. In this step we have 1774 features left. Then we build a new matrix with only these features, standardize the data, and calculate the z-score for each word feature. We select features with high absolute z-score values. So we have 1173 features after removing those with absolute z-scores less than 1.64 (at a confidence level of 0.05) and select top 500 of them.

The testing dataset also includes 25000 instances but with 89523 features. This could be due to different word appearances within the text dataset, so we filled the dataset with zero columns for the missing features to form the same shape as the training dataset and select the same features.

The most negative feature is “lousy”, which has a z-score of -36.63, and the most positive feature is “love” which has a z-score of 71.22. The higher z-score could imply a better attitude, meaning a higher value in scores.

The second dataset is NewGroups4, containing textual datasets from four categories: sport.hockey, religion.chirtian, comp.graphic and sci.med. The training dataset includes 2377 instances and 30945 features. The testing dataset is separately generated and consists of 1582 instances and the same number of features.

In the data process stage, we applied the same steps as IMDb ratings, building up the dataframe and removing the stopwords and irrelevant words. We are left with 1505 features in this step. Then for each category, we use the Mutual Information method to find the top 100 features with the highest similarity. Then we combine the results, and after dropping the duplicates, we have 319 features in total.

Results:

First, we applied single linear regression to select the most essential features in the IMDb dataset. The features with the most positive and most negative z-scores are shown in the plot **(Task 3.1)**:

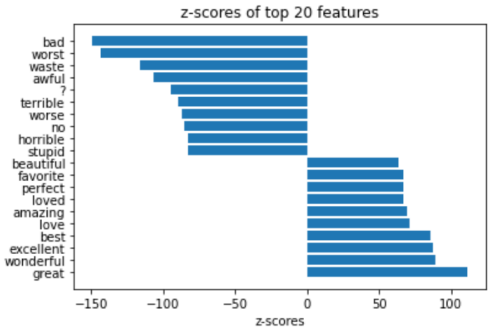

Chart1: top 20 features of the IMDb dataset

After constructing the Logistics Regression model and the Multiclass classification model, we applied them separately to the IMDb score dataset and the NewGroups4 dataset for classification training. We used the gradient descent method on the NewGroup4 to estimate the

coefficients and cross-entropy converge to 0 with 1000 iterations. The learning rate we used here is 0.005 (**Task 3.2**).

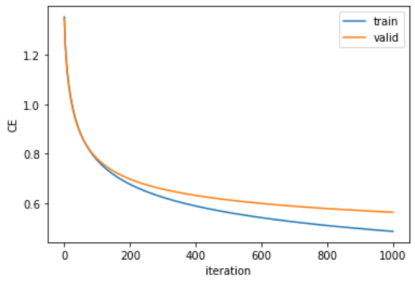

Chart2: Convergence plot for NewGroups4

Also, after constructing the models, we want to observe the top 20 features with the largest coefficients generated by logistics regression using the same visualization method in Task 3.1. This time the most significant features are these **(Task 3.6):**

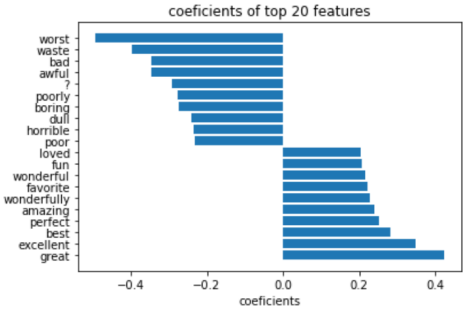

Chart3: Top 20 features of IMDb scores with the largest coefficients

For the NewGroup4 dataset, since there are four categories in the target variable, meaning that we would have four dimensions if plotting barplot as in Task 3.6, we would choose to implement a heatmap to illustrate the top 5 most significant features for each category. The result is below, and we can see that each top 5 features are intuitively closely related to each category (**Task 3.7**).

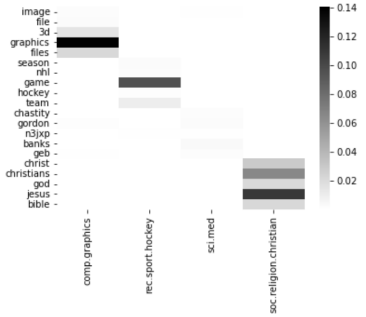

Chart4: Top 5 features for each category in the NewGroup4 dataset

Besides, we would also want to compare how the KNN method could have different performances compared to logistics regression on the IMDb dataset. We implemented the KNN method from the sklearn package and visualized performances using the roc curve **(Task 3.3)**.

The plot is shown below, and we can see that the logistics regression model generates a better result.

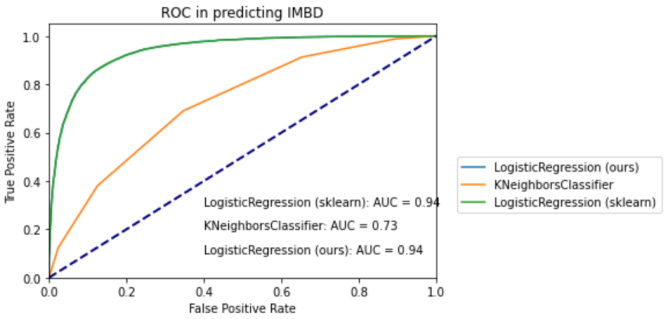

Chart6: Roc Curve of Logistics regression and KNN on IMDb dataset

Next, we experiment with how the training dataset size could influence models' performance, logistics regression, and KNN **(Task 3.4)**. We used the auroc score as the evaluation method, and we constructed the bar plot to see the trend of model performance with the training dataset size increases. The result is shown below, and we can see that the score slightly increases with the training dataset size increases in both methods.

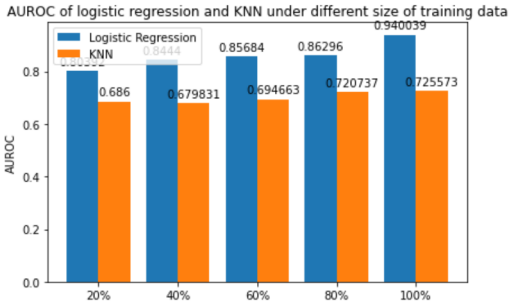

Chart7: AUROC of logistics regression and KNN under different training dataset sizes

We then want to test if the same result (trend) would happen on comparing multiclass regression and KNN on the NewGroup4 dataset. This time we use the accuracy_score from the sklearn package as the evaluation method, and we again construct a bar plot containing the accuracy scores from different training dataset sizes **(Task 3.5)**. We observed that the accuracy gap between multiclass regression and KNN is even larger than in Task 3.4. However, the performances of both models are still improved as the dataset size increases.

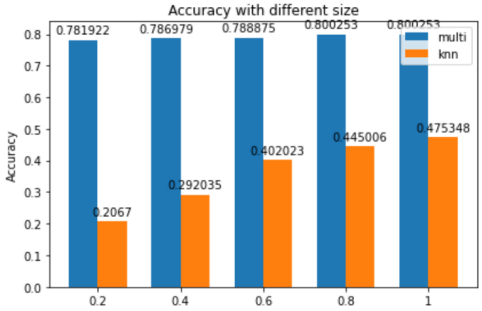

Chart8: Accuracy score of multiclass regression and KNN under different training dataset sizes

Also, for curiosity, we applied the linear regression method on the IMDb rating score, and we received an accuracy rate of 0.4753. Thus we should only consider implementing logistics regression on binary variable prediction.

Discussion and Conclusion:

For the key learnings, we compared both methods with KNN and other classification methods from the sklearn packages on two processed datasets with different sizes. Compared to the datasets in the last assignment, this time, we have fewer target variables, changing from continuous to discrete values with only 2 or 4 categories, and more features (we have over 300 features on both datasets). In this situation, we found that the logistics regression have better performances on datasets with binary target variables, and multiclass regression also shows the better result on the NewGroup4 dataset with four categories. Besides the property of target variables influencing the model performances, we believe that a relatively large number of features also affect the performance of KNN on both datasets. Also, since both cases show a long running time, we should consider other classification methods other than KNN when encountering datasets like IMDb reviews or NewGroup4. Lastly, we found that the training dataset sizes only slightly influence the model's performances on a specific dataset. This means finding a suitable model to train and predict is more important than only focusing on feeding the model with more data.

In the future steps, we will further explore different pre-process dataset methods. In task 3.2, we employed the min-max standardized method rather than regular standard-scaler method to fix the data-overflow question in providing convergence plots of the NewGroup4 dataset. We fixed the problem, and we would further study why the way of standardizing the dataset could affect the convergence process. We are also interested in how to make the model more interpretable. In task 3.6, we noticed that one of the features is “?”, which is hard to interpret for sentiment analysis, and we would like to seek ways to eliminate features similar to this. Lastly, we would also want to test how different learning rates could affect the gradient descent process.

Statement of Contributions:

Kevin and Yujie contributed most part of the coding, and Zhixuan contributed small part of the coding and the report.