COSC 74 Final Project Write-Up

Kevin King, Spring 2022

Binary Classification

In each of the cutoff values (1-4), I used the following binary classifiers: Logistic regression, Perceptron, SVM (LinearSVC). I performed hyperparameter tuning in each of the cutoffs in order to improve my scores by using feature engineering and cross-validation. The tables below include K Fold averages for the F1 score, accuracy, precision, confusion matrix, and ROC AUC score. I provided code for two types of testing, which can be set by the “submission” variable in the code. When True, it creates the CSV for a Kaggle submission. When False, it prints out the 5-fold cross-validation output.

Feature Engineering

At first, I only used the *reviewText* feature and vectorizing it with *TFIDF vectorizer.* However, my scores were just around the baseline, hovering just above or just below it. I made use of the *summary* feature as well by vectorizing it and then using the *hstack* to combine them. In the below tables, I display my original results based on the *reviewText* feature, whereas the best model incorporates the *summary* feature.

Cutoff 1

Best Model

```
model = LogisticRegression(fit_intercept=False)
```

Results Table

Method	K Fold Accuracy avg	K Fold Precision avg	K Fold F1 avg	K Fold ROC AUC avg	K Fold CM avg
Logistic Regression	0.79	0.76	0.680	0.66	[483.8 707.6] [513.6 4132.8]
Perceptron	0.75	0.67	0.65	0.66	[570.4 621.] [860.6 3785.8]
Linear SVC	0.77	0.71	0.67	0.66	[549. 642.4] [715.2 3931.2]
Best	0.79	0.74	**0.684**	0.67	[528.2 663.2] [592 405.4]

Comments

In this binary classification method, we are attempting to determine whether the *overall* column was either greater than 1 or less than or equal to 1. The best model for this cutoff number ended up being Logistic Regression with the fit_intercept=False (meaning no bias term). Using 5-Fold cross validation, the model was able to get an F1 average score=0.684.

Cutoff 2

Best Model

```
model = LogisticRegression(fit_intercept=False, C=0.4, max_iter=1000)
```

Results Table

Method	K Fold Accuracy avg	K Fold Precision avg	K Fold F1 avg	K Fold ROC AUC avg	K Fold CM avg
Logistic Regression	0.73	0.73	0.723	0.73	[1678.6 704.6] [853.8 2600.8]
Perceptron	0.69	0.68	0.68	0.68	[1614.6 768.6] [1042 2412.4]
Linear SVC	0.72	0.71	0.71	0.71	[1681 702.2] [946.2 2508.4]
Best	0.74	0.73	**0.727**	0.73	[1674.4 708.8] [824.2 2630.4]

Comments

In this binary classification method, we are attempting to determine whether the *overall* column was either greater than 2 or less than or equal to 2. The best model for this cutoff number ended up being Logistic Regression with the fit_intercept=False, C=0.4 (decides regularization strength), and 1000 max iterations taken for the solvers to converge. Using 5-Fold cross validation, the model was able to get an F1 average score=0.727.

Cutoff 3

Best Model

```
model = LogisticRegression(fit_intercept = False, max_iter=100, solver='saga')
```

Results Table

Method	K Fold Accuracy avg	K Fold Precision avg	K Fold F1 avg	K Fold ROC AUC avg	K Fold CM avg

Logistic Regression	0.79	0.79	0.754	0.75	[3208.2 347.4] [887.6 1394.6]
Perceptron	0.74	0.73	0.72	0.72	[2954.6 601] [889.8 1392.4]
Linear SVC	0.77	0.76	0.7408	0.74	[3072.8 482.8] [852.8 1429.4]
Best	0.79	0.78	**0.757**	0.76	[3167 388.6] [845.6 1436.6]

Comments

In this binary classification method, we are attempting to determine whether the *overall* column was either greater than 3 or less than or equal to 3. The best model for this cutoff number ended up being Logistic Regression with the fit_intercept=False, max_iter=1000, and solver='saga'. I experimented with a few different solvers and 'saga' proved to be the best of the options for my model. Using 5-Fold cross validation, the model was able to get an F1 average score=0.757.

Cutoff 4

Best Model

```
model = LogisticRegression(fit_intercept=False)
```

Results Table

Method	K Fold Accuracy avg	K Fold Precision avg	K Fold F1 avg	K Fold ROC AUC avg	K Fold CM avg
Logistic Regression	0.87	0.86	0.743	0.71	[4612.4 97] [637.8 490.6]
Perceptron	0.85	0.76	0.73	0.72	[4369 340.4] [556.8 571.6]
Linear SVC	0.87	0.81	0.74	0.72	[4501.6 207.8] [579 549.4]
Best	0.88	0.84	**0.758**	0.73	[4572 137.4] [580.4 548]

Comments

In this binary classification method, we are attempting to determine whether the *overall* column was either greater than 4 or less than or equal to 4. The best model for this cutoff number ended up being Logistic Regression with the fit_intercept=False. Using 5-Fold cross validation, the model was able to get an F1 average score=0.758.

Multiclass Classification

Best Model

```
model = LogisticRegression(fit_intercept=False, multi_class='ovr', class_weight='balanced')
```

Results Table

Method	K Fold Accuracy avg	K Fold Precision avg	K Fold F1 avg	K Fold ROC AUC avg	K Fold CM avg
Logistic Regression	0.79	0.79	0.754	0.75	[3208.3 347.4] [887.6 1394.6]
Perceptron	0.74	0.73	0.72	0.72	[2954.6 601] [889.8 1392.4]
Linear SVC	0.77	0.76	0.74	0.74	[3072.8 482.8] [852.8 1429.4]
Best	0.79	0.77	**0.763**	0.76	[3035 520.6] [715.8 1566.4]

Comments

In this multi-class classification method, we are attempting to determine whether the *overall* column was either greater than 1 or less than or equal to 1. The best model for this cutoff number ended up being Logistic Regression with the fit_intercept=False, multi_class='ovr' (which makes a binary problem fit for each label), and class_weight='balanced' to adjust weights inversely proportional to class frequencies in the input data. Using 5-Fold cross validation, the model was able to get an F1 average score=0.763.

Clustering

Results Table

Method	Score
Silhouette Score	0.955
Random Score	0.625

Comments

In the clustering method, we are attempting to use k-means clustering with the *category* column as new labels in the model, and using the Silhouette score and Rand index to analyze the quality of the clustering. At first, I only used the *reviewText* feature in the model, at which the output score was just above the baseline but after using the *summary* feature as well, it skyrocketed.