#### Group Assignment

#### Task 2.1: Reviews 1 through 5

The Multinomial Naive Bayes model is used for rating prediction with the discrete vocabulary from the text data, and its accuracy is evaluated along with the confusion matrix. A low prediction accuracy of 47.12% is obtained as characterized by its confusion matrix where only the diagonal(TP + TN) was predicted accurately. (Appendix 1a)

The confusion matrix demonstrates that the prediction accuracy is positively associate with the rating, this may due to the larger number of data available for the higher stars with 4-stars having the most at 3526 and 1-star having the least at 749 which is 4 times less. Moreover, many incorrect predictions were between consecutive numbers, such as a rating 3 is easily predicted to be a 4 or 2; This may due to the similar sentiment or word choices of reviews that have a close star rating.

#### Task 2.2: Reviews 1 and 5

After filtered reviews that only contain rating 1 or 5 as the two extremes, the prediction accuracy became 91.87% with a 2x2 confusion matrix. (Appendix 1b). Similar to before the 1-star rating had a lower accuracy at 83.44% and five star at 93.34% which is again due to the much greater number of five-star reviews over one-star reviews.

Via examining the False Negative and False Positive records, length of reviews has a negative impact on model’s prediction accuracy. In particular, only one False Positive record (Actual 5 star but predicted as 1 star) is discovered, which was a brief and concise review that contains a keyword “Not bad”. Yet, since feature extraction split reviews into words, the word ‘bad’ with negative sentiment cause the model to predict this review incorrectly as rating 1.

#### Task 2.3 Token Words

Top 10 token words that are most predictive for either 5 or 1-star reviews are further extracted (Appendix 2). The 1-star to 5- star ratios are applied, as simply going by frequency resulted in conjunction words which did not help differentiate the two.

With ratios, we observed the words most predictive of 5 stars consist of positive emotion words “fantastic, perfect, favorite, outstanding, amazing” followed by food descriptions like “yum, brunch, mozzarella, pasty”. This reveals that most 5-star reviews are focused on food and positive sentiment words. Words that were most predictive of 1-star reviews are composed of negative description words of “disgusting, filthy, unacceptable,” along with ones describing the staff in “staffperson, refused, unprofessional.” Other short emotional words including “ugh and yuck” were also mentioned. This shows that negative descriptions, emotions and staff reviews are predictive of 1-star reviews. Overall*,* most words that are predictive of are either very emotional words or refer to a particular subject like staff or food.

#### Task 3.1 Improving Task 2.2 by using different models

Five binary classifiers are selected to fit data of one and five-star ratings, with each model tuned manually to its best parameters through the use of accuracy graphs and compared with the base Multinomial Naive Bayes model (Prediction Accuracy 91.87%).

* K nearest neighbors classifier: Prediction accuracy= 84.64% (n\_neighbors= 6)
* RandomForestClassifier: Prediction accuracy = 87.38% (n\_estimators=14)
* Log Support Vector Machines: Prediction accuracy= 91.68% (Default Parameter)
* Linear Support Vector Machines: Prediction accuracy= 92.27% (Default Parameter)

The best model is logistic regression model with a prediction accuracy of 92.56% under default parameters (Appendix 1c). However, for predicting the relationship between the text and star rating, model training is limited by the small dataset size and limited features in the data we used. Therefore, logistic regression may not be the prime model for all review datasets, which will be further addressed in task 4.

Using the logistic regression model and filtering by only including 1) reviews longer than 20 words 2) only words longer than 2 letters, brought the best prediction accuracy of 94.87%, we will explain the reasons for these filters in the following task.

#### Task 3.2 Improving Task 2.1 through filtering

Applying different filtering criteria to fit the data of one through five-star ratings and compared with the base Multinomial Naive Bayes model (Prediction Accuracy 47.12%).

Applying following three successful filtering criteria focused on length created improved accuracy of 52.07%:

* Only reviews with more than 10 words were kept. Shorter reviews are presumed to be less probable to include accurate precision words or show enough emotions.
* Remove words shorter than 3 letters as are usually conjunction, or English stop words that create noise.
* Only words that appeared in more than 2 but less than 90% of the total reviews were kept. Words that appear in too few reviews are not useful for testing as they are unlikely to be seen again, high-frequency words like conjunction words might not predictive and do not differentiate between star ratings.

Two methodologies are used based on above however they decreased accuracy.

* Filtering out English stop words, declined accuracy below the base accuracy to 46.92%. (Appendix 1i).
* If the user only wrote one review, they may be less reliable and could be too different from other reviews making it less predictive therefore was filtered out. However, this only resulted in a prediction accuracy of 48.0%. (Appendix 1j)

The best model filtering with the highest accuracy overall gave a result of 53.44% and focused on frequency:

* Assigning low weightings to common and high-frequency words covered all possible noise words like conjunctions words that had low predictive accuracy and would not help differentiate much.
* Limiting the word by frequency to those appearing in more than 2 but less than 90% of the total reviews

#### Task 4.1 Generalized Software for Text-Mining Task

The aim for our task 4 project is to design a generalized python software, that will perform model evaluations and parameter tuning with selected classifiers on a text-mining task. Users can input any text-mining dataset and our software will optimize model parameters with reference to prediction accuracy.

To begin with, two methods are considered to be used for vectorizing texts - Term Frequency (TF) & Term Frequency-Inverse Document Frequency (TF-IDF). Although the model evaluation result reveals that most models perform better without using TF-IDF, it is believed to be caused by limited training data given the size of this dataset. If a larger training dataset is provided, classifiers like Linear SVC that does not depend on frequent but not meaningful words should perform better in predicting review rating.

Regarding the performance of models, Stochastic Gradient Descent(SGD), Multi-layer Perceptron(MLPClassifier) and Linear Support Vector Machines(Linear SVC) perform overall well in 10-fold cross-validation. On average these models have an accuracy of 92% for this dataset. Multinomial Naive Bayes(MNB) also perform relatively well with TF only, but its performance drops significantly when TF-IDF is applied, showing this model largely depends on using common words for prediction, hinders its suitability for this text mining project.

Selected models are then tuned to adopt its best parameters for prediction. Via using GridSearchCV, parameters are chosen based on its accuracy in predicting test data. Due to the limitation of computing power, it is not possible to permute all possible combinations of parameters for each model. In this version of the software, each set of parameters is chosen manually, while this process can be further generalized so the software can choose the best suitable parameter set based on user-chosen key model evaluation indicator (Accuracy, Precision, Recall, …).

To summarize, the Linear SVC model is believed to be the most suitable classifier for text-categorization(with 92.67% accuracy under TF-IDF). One underlying reason is that text is usually linear separable, as most words are highly associated with either positive or negative sentiment. These words will then be granted with higher weight under Linear SVC, which then increases its prediction accuracy. Although the model evaluation reveals that Linear SVC can obtain accuracy as high as 94.5%, this did not imply our prediction accuracy of 92.67% is not good as the previous accuracy is achieved via splitting dataset differently under cross-validation. The model itself should not depend on the way dataset is split, but its ability for predicting unknown records are the indicator we use for evaluating its performance.

### Appendix

#### Appendix 1 Prediction accuracy and confusion matrix

|  |  |
| --- | --- |
| 1a Multinomial Naive Bayes model for 1 through 5 | 1b Multinomial Naive Bayes model for 1 and 5 |
|  |  |
| 1c LogisticRegression | 1d Logistic Regression with filters |
|  |  |
| 1d Filtering by reviews and word length | 1e Filtering by frequency |
|  |  |

#### Appendix 2 Token words

|  |  |
| --- | --- |
| Predictive of Five star | Predictive of One star |
|  |  |

#### Appendix 3 Task 4 Classifier

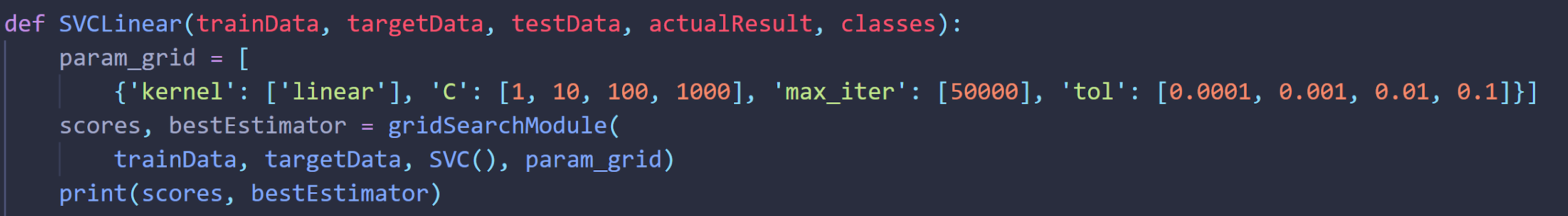
#### 3.1 TF & TF-IDF Difference Explanation, and screenshot examples

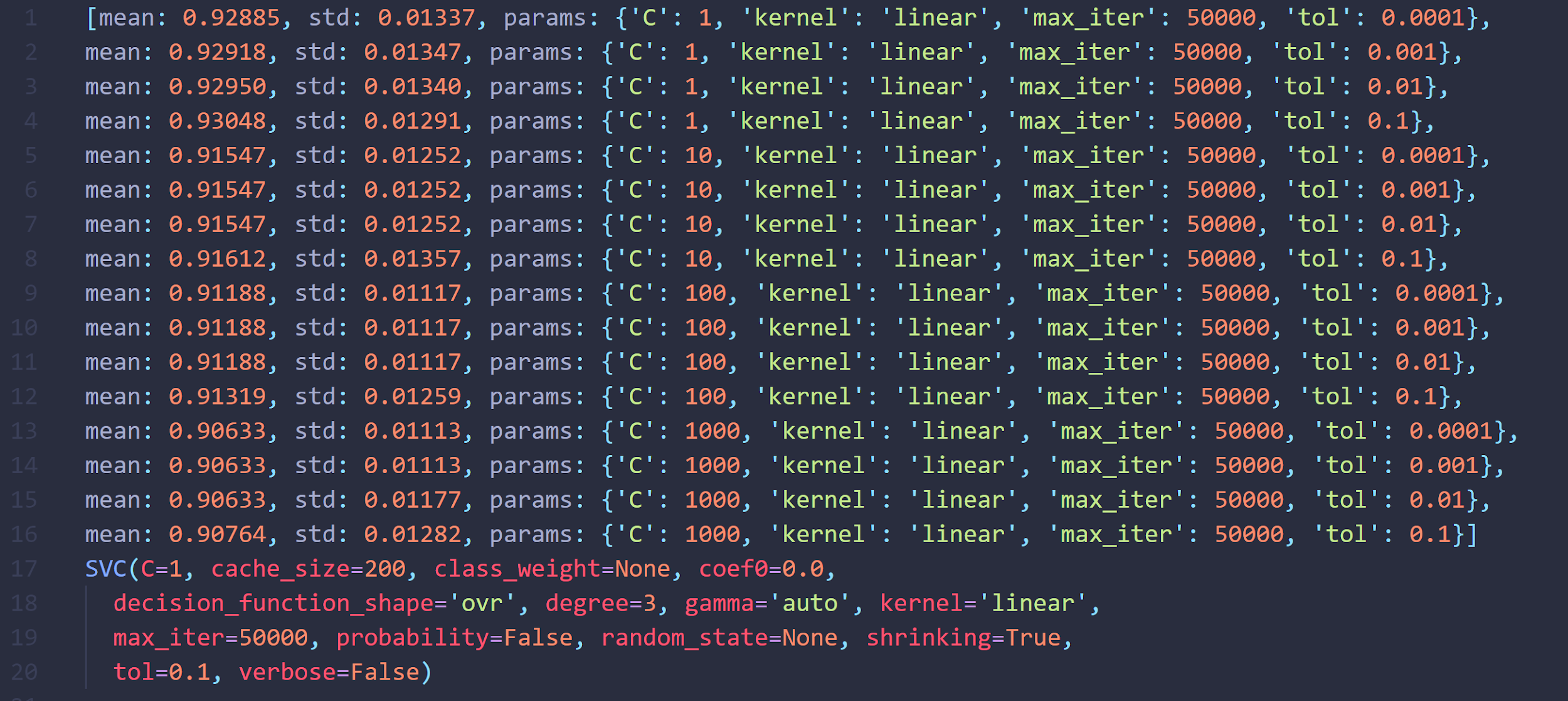
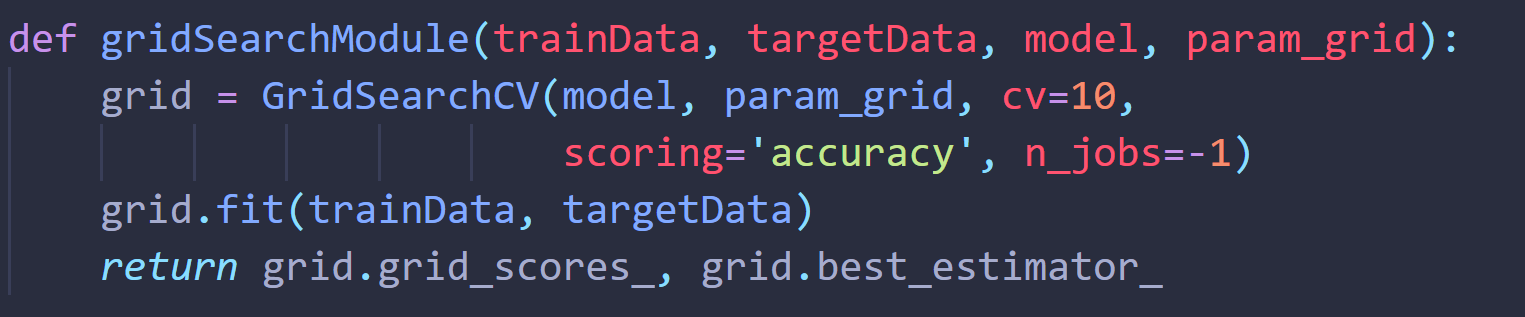
These differ by whether the corpus-frequencies of words are applied a higher weight or not. Using TF means the vectorizer will give common words across all documents a higher weight in prediction even no useful information is contained. Meanwhile, TF-IDF is designed to penalize common words and give higher weight to rare words, as less frequently words are assumed to carry more information.

#### Screenshot Examples

|  |  |
| --- | --- |
| **Figure 1. Model Evaluation Star 1&5 (With TF)** | **Figure 2. Model Evaluation Star 1&5 (With TF-IDF)** |
|  |  |
| **Figure 3. Model Evaluation Star 1-5 (With TF)** | **Figure 4. Model Evaluation Star 1-5 (With TF-IDF)** |
|  |  |

**Figure 5. Sample Parameter Tuning for Linear SVM classifier**

****

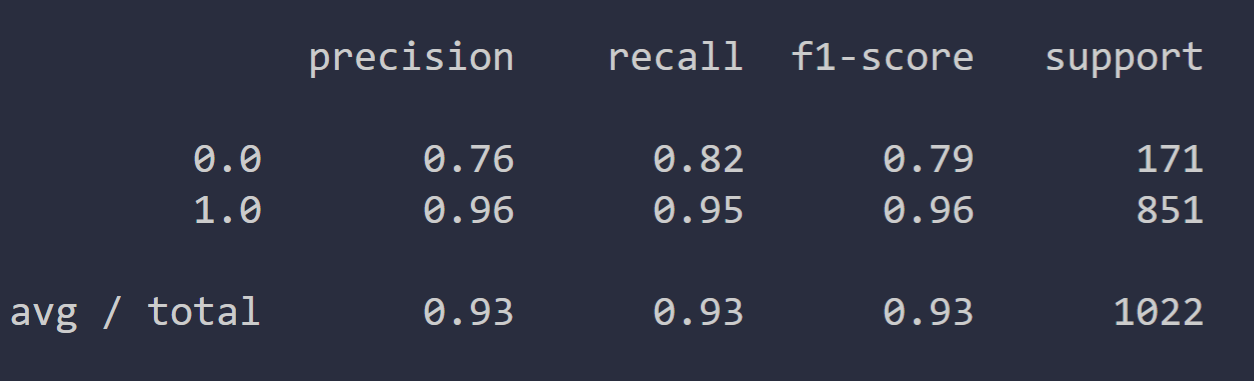
****

**Figure 6. Linear SVM Classifier (With TF-IDF) Parameter Tuning Result & Best Estimator**

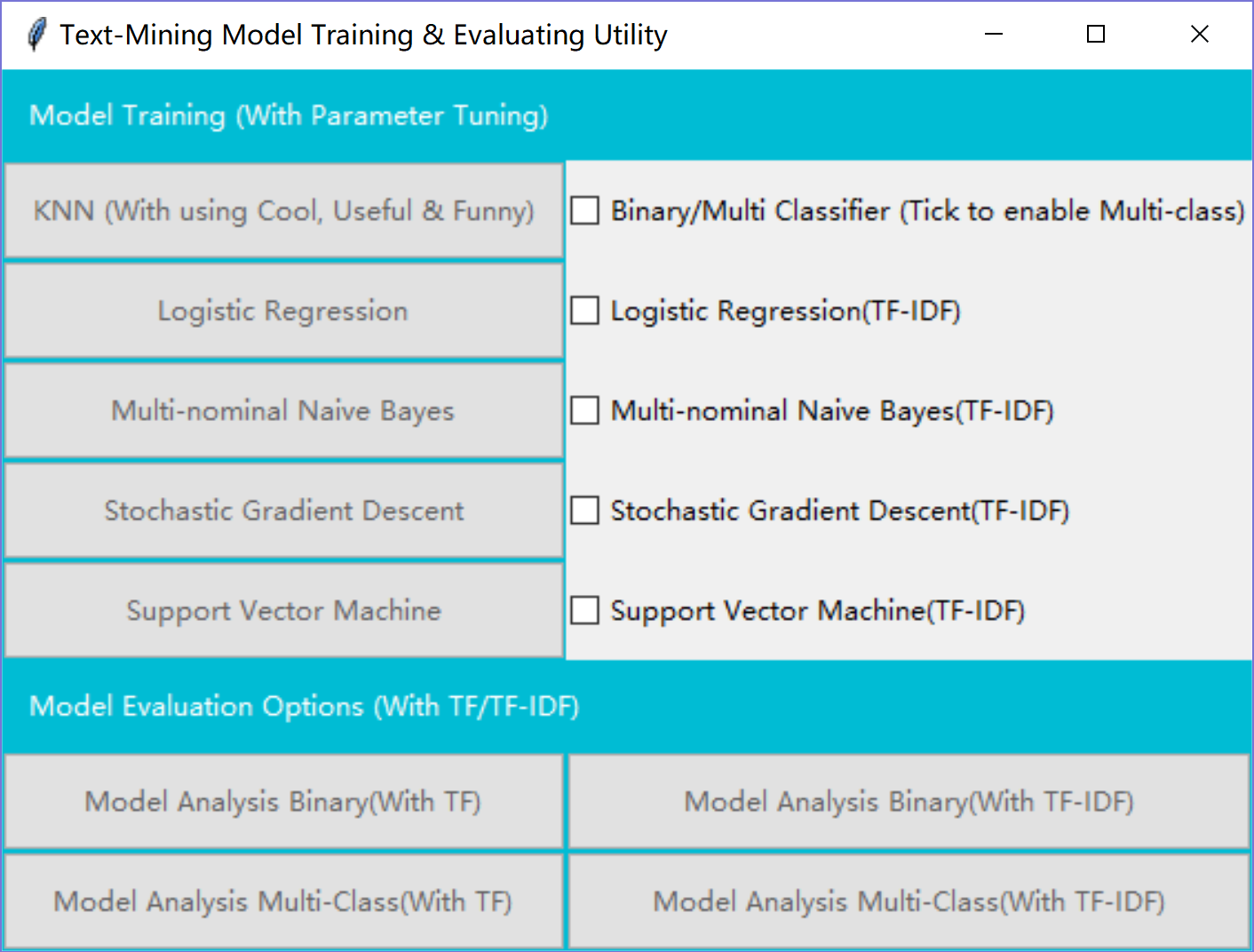
**Explanation:** For example, for Linear SVC classifier, a set of parameters {'kernel': ['linear'], 'C': [1, 10, 100, 1000], 'max\_iter': [50000], 'tol': [0.0001, 0.001, 0.01, 0.1]} is provided for parameter tuning. With the best estimator returned by GridSearch, it will then be used for training via fitting training dataset. A confusion matrix and ROC curve(Only applicable for Binary Classification task) are plotted to provide a visualization of classification result for users’ better understanding.

|  |  |
| --- | --- |
| **Figure 7. Logistic Regression (With Term Frequency Only) Evaluation** | |
|  | |
| **Figure 8. Linear SVM Classifier (With TF-IDF) Confusion Matrix** | **Figure 9. SGD Classifier (With TF-IDF) Confusion Matrix** |
|  |  |

**Figure 10. Linear SVM Classifier (With TF-IDF) Performance Indicator Report**

****

**Figure 11. GUI of Task 4**

****

**Evaluated Model List:**

* RandomForestClassifier()
* LinearSVC()
* MultinomialNB()
* LogisticRegression()
* SGDClassifier(loss='log', max\_iter=100000, tol=1e-3)
* MLPClassifier(solver='lbfgs', alpha=1e-5)
* AdaBoostClassifier()

|  |  |
| --- | --- |
| **KNN based on column [‘Cool’, ‘Useful’, ‘Funny’]** | **Enable/Disable Predicting for Star 1-5 (Uncheck this box means Binary Classification for Start 1 & 5)** |
| **Logistic Regression** | **Using TF-IDF dataset for Logistic Regression Classifier** |
| **Multinomial Naive Bayes** | **Using TF-IDF dataset for Multinomial Naive Bayes Classifier** |
| **Logarithmic Stochastic Gradient Descent** | **Using TF-IDF dataset for Logarithmic Stochastic Gradient Descent Classifier** |
| **Linear Support Vector Classifier** | **Using TF-IDF dataset for Linear Support Vector Classifier** |
| **Evaluate models for predicting Star 1 & 5 via 10-Fold Cross-Validation (With TF Only)** | **Evaluate models for predicting Star 1 & 5 via 10-Fold Cross-Validation (With TF-IDF)** |
| **Evaluate models for predicting Star 1 - 5 via 10-Fold Cross-Validation (With TF Only)** | **Evaluate models for predicting Star 1 - 5 via 10-Fold Cross-Validation (With TF-IDF)** |

**Please be aware that the software will take some time to calculate the result, especially when the Multi-class prediction is enabled (Since the GridSearchCV will select the optimized parameter for the selected classifier, which usually takes several minutes). Be patient, get a coffee, and the prediction result will be ready for you.**