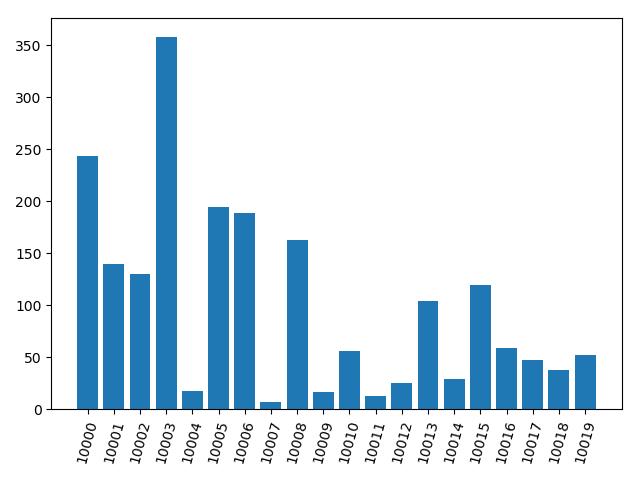
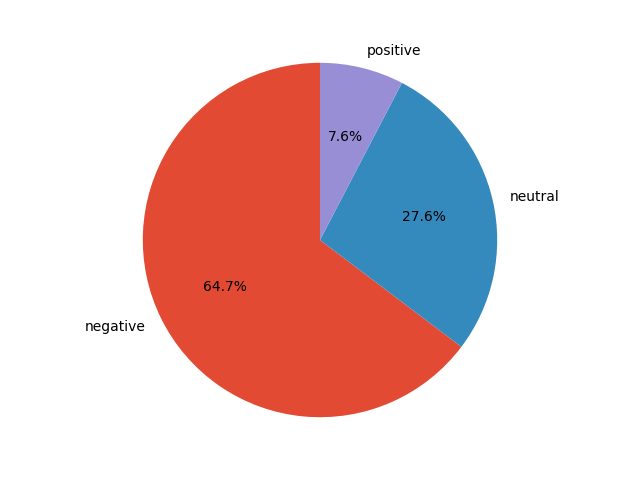
Assignment 2: Opinion Mining

1. **The frequency distributions for the sentiment and topic classes across the full dataset.**



**Figure 1.** The frequency distributions for the topic classes

From the figure we can see that these comments’ topics are not evenly distributed, but are concentrated on several topics, such as 10003 (Economic management), 10000 (corruption / governance), 10005 (healthcare / medicare) and 10006 (Social issues / marriage equality / religion). Those topics are issues that are widely concerned in daily life.



**Figure 2.** The frequency distribution for the sentiment classes

From the figure we can see that most comments are negative, only a few are positive.

1. **The baseline models’ performance and runtime vary with feature dimensions.**

Top N {100, 200, 300} words are selected as vocabulary, and using the tf-idf matrix as features. The metrics and runtime of corresponding model are shown in below (train dataset metrics / test dataset metrics):

Sentiment analysis:

300 words

200 words

100 words

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Recall | Precision macro | Precision micro | F1\_score | Runtime |
| DT | 0.9713/0.65 | **0.9504/0.511** | 0.9769/0.5142 | 0.9713/0.65 | 0.9630/0.5108 | 0.030s |
| BNB | 0.7327/0.684 | 0.5974/0.506 | 0.6724/0.5402 | 0.7327/0.684 | **0.6247/0.5147** | 0.027s |
| MNB | **0.7013/0.696** | 0.4390/0.4061 | **0.8038/0.7477** | **0.7013/0.696** | 0.4494/ 0.4060 | **0.012s** |
|  | | | | | | |
| DT | 0.9913/0.636 | 0.9812/0.4851 | 0.9937/ 0.5002 | 0.9913/0.636 | 0.9873/0.4901 | 0.053s |
| BNB | 0.7527/0.72 | **0.6373/0.5402** | 0.6994/0.5987 | 0.7527/0.72 | **0.6615/0.5575** | 0.032s |
| MNB | **0.732/0.726** | 0.4923/0.4534 | **0.8245/0.7816** | **0.732/0.726** | 0.5246/0.4683 | **0.021s** |
|  | | | | | | |
| DT | 0.994/0.692 | **0.9843/0.5640** | 0.9951/ 0.5861 | 0.994/0.692 | 0.9896/0.5716 | 0.066s |
| BNB | **0.7767/0.736** | 0.6645/0.5576 | 0.7311/0.666 | **0.7767/0.736** | **0.6902/0.5775** | 0.078s |
| MNB | 0.7467/0.734 | 0.5151/0.4664 | **0.8402/0.7959** | 0.7467/0.734 | 0.5556/0.4891 | **0.024s** |

**Table 1.** Sentiment analysis: the metrics and runtime of corresponding

model with different vocabulary size.

From the Table 1, we can see that the Decision Tree (DT) model heavily overfitting on the dataset. When the vocabulary size used in feature processing is 300, the accuracy of 0.994 can be achieved on the training set, but on the test set, the accuracy is less than 0.7. This may be caused by the size of the dataset.

Compared with DT, there is no obvious overfitting in the two models of Bernoulli Naive Bayes (BNB) and Multinomial Naive Bayes (MNB).

When the feature dimension is small (vocabulary size is 100), most metrics of the MNB model are higher than the BNB. But when the feature dimension rises (vocabulary size 200 and 300), the performance of the BNB gradually surpasses the MNB.

From the perspective of runtime, runtime grows as the number of features increases. And the MNB runs much faster than the other two models (approximately a half of runtime).

Topic classification:

300 words

100 words

200 words

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Recall | Precision macro | Precision micro | F1\_score | Runtime |
| DT | 0.822/0.274 | 0.7518/0.1689 | 0.8681/0.1754 | 0.822/0.274 | 0.7858/0.1705 | 0.168s |
| BNB | **0.4693/0.398** | **0.2890/0.2272** | 0.4334/0.2246 | **0.4693/0.398** | **0.3007/0.2207** | 0.048s |
| MNB | 0.428/0.376 | 0.2111/0.1800 | **0.2905/0.2255** | 0.428/0.376 | 0.2152/0.1791 | **0.061s** |
|  | | | | | | |
| DT | 0.8633/0.326 | 0.8008/0.2443 | 0.9042/0.2481 | 0.8633/0.326 | 0.8341/0.2432 | 0.293s |
| BNB | **0.5553/0.474** | **0.3556/0.2694** | 0.5336/0.2969 | **0.5553/0.474** | **0.3765/0.2682** | 0.087s |
| MNB | 0.494/0.412 | 0.2477/0.2015 | **0.3913/0.3492** | 0.494/0.412 | 0.2545/0.2100 | **0.036s** |
|  | | | | | | |
| DT | 0.87/0.37 | **0.813/0.2894** | **0.9090/0.3009** | 0.87/0.37 | **0.8478/0.2892** | 0.429s |
| BNB | **0.6053/0.47** | 0.3845/0.2657 | 0.6011/0.272 | **0.6053/0.47** | 0.4118/0.2690 | 0.037s |
| MNB | 0.5327/0.4 | 0.2708/0.1875 | 0.4099/0.2447 | 0.5327/0.4 | 0.2815/0.1892 | **0.030s** |

**Table 2.** Topic classification: the metrics and runtime of corresponding

model with different vocabulary size.

From the Table 2, we can see the experiment results that similar to the sentiment analysis: overfitting of DT model, more stable performance of BNB and MNB models, and MNB is still the fastest model.

However, the difference is that the all baseline models have much lower metrics on the topic classification task. This should be because of there are 20 topics for all Twitter, and the distribution of topics is imbalance. But the performance of BNB is better than other two models.

In addition, the metric precision micro is equal to the metric accuracy. Check out the scikit-learn API documentation that ‘micro’ parameter for the binary classification task.

**Conclusion:** The DT model is prone to overfitting in the case of small dataset, while NB models avoids overfitting due to the use of prior probabilities.

1. **Evaluate the standard models with respect to baseline predictors.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | DT | BNB | MNB | VADER | Majority class |
| Sentiment  analysis | 0.692 | 0.736 | 0.734 | 0.252 | 0.647 |
| Topic classification | 0.37 | 0.47 | 0.4 |  | 0.179 |

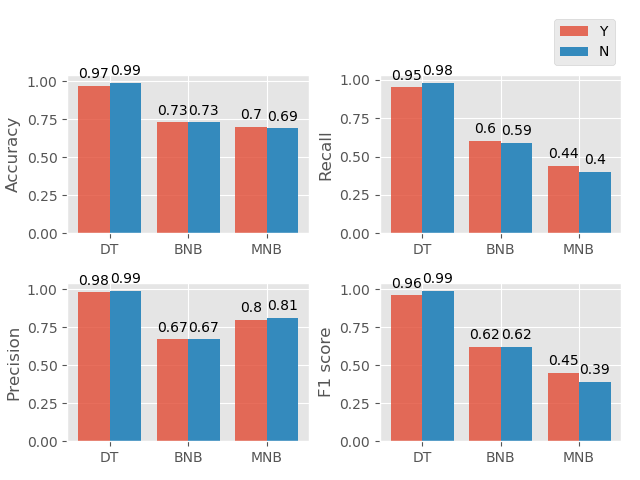
**Table 3.** The test dataset accuracy of standard models and baseline predictors on sentiment analysis and topic classification tasks (with NLTK preprocessing, vocabulary size is set to 300).

Compare to the baseline models, we can find that the standard models outperform the proportion of majority class. That prove the effectiveness of three standard models. However, VADER seems to have some bugs. When using test sets for prediction, the prediction labels for all samples are neutral.

1. **Evaluate the effect that preprocessing the input features.**

Sentiment analysis:

The vocabulary size is set to 100.



**Figure 3.** The metrics on the sentiment analysis training dataset of three baseline models. Where Y presents preprocessing the input features with NLTK, N presents without preprocessing.

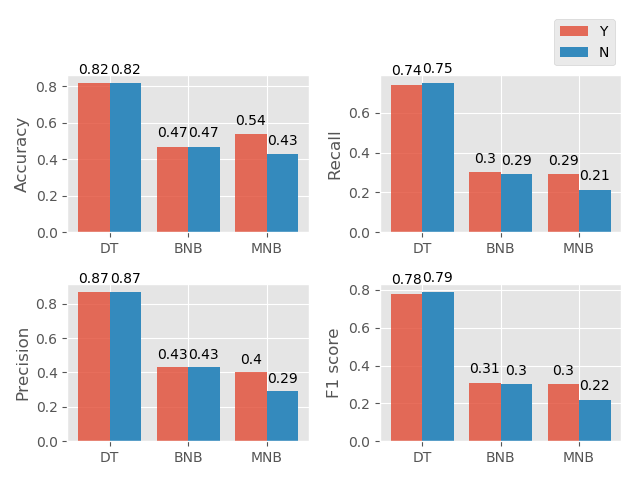


**Figure 4.** The metrics on the test dataset of three baseline models.

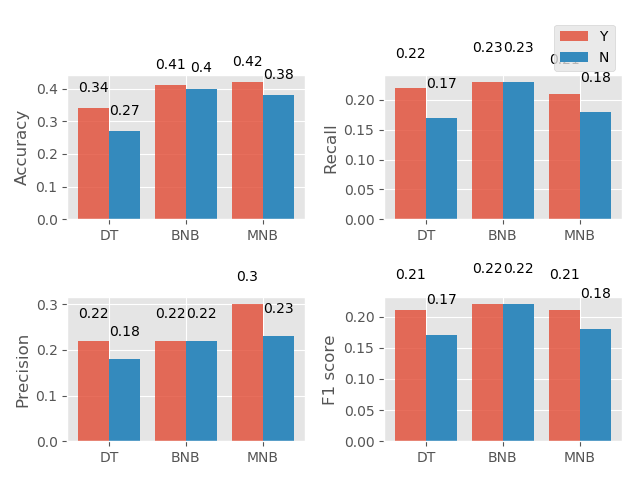
From the Figure 3, we can see that the metrics on the training dataset of baseline models are close with and without preprocessing. However, in the Figure 4, without the NLTK preprocessing, the metrics on the test dataset of DT and MNB drop a little. But the performance of BNB is better.

Topic classification:

The vocabulary size is set to 100.



**Figure 5.** The metrics on the topic classification training dataset of three baseline models. Where Y presents preprocessing the input features with NLTK, N presents without preprocessing.



**Figure 6.** The metrics on the test dataset of three baseline models.

1. **Remove the neutral tweets and evaluate again.**

Sentiment analysis:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Recall | Precision macro | Precision micro | F1\_score | Runtime |
| DT | 1.0/0.9 | 1.0/0.6576 | 1.0/0.7046 | 1.0/0.9 | 1.0/0.6768 | 0.031s |
| BNB | **0.9309/0.9207** | **0.7609/0.6850** | 0.8469/0.7924 | 0.9309/0.9207 | **0.7957/0.7229** | 0.024s |
| MNB | 0.9075/0.9138 | 0.572/0.5536 | **0.9530/0.9564** | 0.9075/0.9138 | 0.6012/ 0.574 | **0.01s** |

**Table 4.** Sentiment analysis: the metrics and runtime of corresponding model

without neutral samples, the vocabulary size is set to 200.

Compare to the Table 1, the metrics on datasets without neutral samples are higher than the original dataset. The accuracy on the test set is about 20% higher than the original data set. Besides, the DT model doesn’t overfit the dataset.

|  |  |  |
| --- | --- | --- |
|  | Negative | Positive |
| DT | 1.0/0.933 | 1.0/0.4762 |
| BNB | 0.9465/0.9377 | 0.7472/0.6471 |
| MNB | 0.9061/1.0 | 0.9129/1.0 |

**Table 5.** Metrics for either class.

Table 4 lists the precision for either class. We can find that DT and BNB have a much higher classification accuracy for negative samples than positive samples, due to sample imbalance.

However, the MNB has a relatively lower classification accuracy rate for negative samples, but the classification accuracy for positive samples can reach 1.

**With and without NLTK preprocessing.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Negative (Y) | Positive (Y) | Negative (N) | Positive (N) |
| DT | 1.0/0.933 | 1.0/0.4762 | 1.0/0.940 | 1.0/0.461 |
| BNB | 0.9465/0.9377 | 0.7472/0.6471 | 0.9506/0.9410 | 0.6952/0.6316 |
| MNB | 0.9061/0.9129 | 1.0/1.0 | 0.9029/0.9097 | 1.0/1.0 |

**Table 6.** Metrics for either class, with and without NLTK preprocessing.

Where Y presents preprocessing, N presents without preprocessing.

Compare the metrics with and without preprocessing, we can find that the classification accuracy of DT and BNB for negative samples increased (both on training and test dataset), but the classification accuracy on positive samples dropped. And the classification accuracy of MNB on negative samples also dropped.

1. **The best model for sentiment analysis and topic classification.**

The best model for sentiment analysis and topic classification is a stacking model using Logistic regression (LR), Gradient boosting decision tree (GBDT), Random forest (RF).

Firstly, using the LR, GBDT and RF model to generate the new features. Then, construct a new LR model to predict the final label of test data, by using the features generated from last layer.

The feature type is ‘TF-IDF’, used vocabulary size of input feature is selected from {100, 200, 300}. The parameters: LR {C: 1, 0.1}, RF {n\_estimators: 100, 200; max\_depth: 3, 6}, GBDT {lr: 0.1, 0.3; max\_depth: 3, 6; n\_estimators: 100, 200}.

The tuned best parameters: vocabulary size – 300; LR C – 1; RF n\_estimators – 200, max\_depth – 6; GBDT lr - 0.1, max\_depth – 6, n\_estimators – 200.

Besides, the stratified k-fold cross validation method is used in training step, to improve the stable performance of model.

The experiment results with best parameters are shown in Table 6.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Accuracy | Recall | Precision macro | F1\_score |
| Sentiment analysis | **0.756** | **0.5641** | **0.7960** | **0.609** |
| Topic classification | **0.48** | **0.3198** | 0.3454 | **0.3238** |

**Table 7.** Metrics for the best model on test dataset, with NLTK preprocessing.

Compare to the standard models and baselines (results in Table 1 and Table 2), the stacking model achieve the best performance (all metrics) on the sentiment analysis task. And on the topic classification task, except the precision metric is the second highest, all of the other metrics are the highest.