Assignment 2 Report

1. Evaluating the standard models (DT, BNB, MNB), and present my own model.

In scenario 1

model	*Decision Tree*	*BNB*	*MNB*	*My*
accuracy	0.645	0.628	0.639	0.669
Micro- precision	0.645	0.628	0.639	0.668
Macro- precision	0.472	0.314	0.394	0.326
Micro-recall	0.645	0.628	0.639	0.679
Macro-recall	0.411	0.216	0.369	0.347
Micro-F1	0.645	0.077	0.639	0.219
Macro-F1	0.316	0.054	0.257	0.264

In scenario 2

model	*Decision Tree*	*BNB*	*MNB*	*My*
accuracy	0.532	0.576	0.602	0.67
Micro- precision	0.532	0.576	0.602	0.67
Macro- precision	0.359	0.348	0.351	0.324
Micro-recall	0.532	0.576	0.602	0.67
Macro-recall	0.358	0.302	0.335	0.308
Micro-F1	0.532	0.194	0.258	0.215
Macro-F1	0.268	0.2	0.228	0.204

The above table shows the decision tree and the performance score of the custom model. In scenario 1 the best model is the custom model, followed by the decision tree model. In scenario 2 the best model is the custom model, followed by the MNB model. None of these four models have parameters adjusted, and the default parameters are used. Change them below. The parameters are then evaluated for the model.

2. Develop Decision Tree models for training and testing: (a) with the 1% stopping criterion (the standard model), and (b) without the 1% stopping criterion.

model	*Decision Tree*			
scenario	1		2	
stop criterion	With 1%	Without 1%	With 1%	Without 1%
accuracy	0.638	0.539	0.643	0.546
Micro- precision	0.638	0.539	0.643	0.546
Macro- precision	0.294	0.248	0.472	0.358
Micro-recall	0.638	0.539	0.644	0.546
Macro-recall	0.254	0.237	0.417	0.359
Micro-F1	0.116	0.149	0.643	0.546

Macro-F1	0.147	0.117	0.318	0.269

The stop criterion of the decision tree can effectively cut off those leaves with too few samples to prevent the model from overfitting. According to the data in the above table, cutting these leaves does greatly improve the performance of the model, and the accuracy rate is improved. About 10%.

3. Develop BNB and MNB models from the training set using: (a) the whole vocabulary (standard models), and (b) the most frequent 1000 words from the vocabulary, as defined using scikit-learn Count Vectorizer, after preprocessing by removing “junk” characters.

model	*BNB*				*MNB*			
scenario	1		2		1		2	
vocabulary	whole	1000	whole	1000	whole	1000	whole	1000
accuracy	0.62	0.57	0.63	0.62	0.64	0.6	0.67	0.66
Micro- precision	0.62	0.57	0.63	0.62	0.64	0.6	0.67	0.66
Macro- precision	0.31	0.34	0.39	0.48	0.22	0.35	0.47	0.54
Micro-recall	0.62	0.57	0.63	0.62	0.64	0.60	0.67	0.66
Macro-recall	0.21	0.3	0.36	0.46	0.20	0.33	0.38	0.52
Micro-F1	0.07	0.19	0.63	0.62	0.01	0.25	0.67	0.66
Macro-F1	0.058	0.2	0.25	0.35	0.01	0.22	0.27	0.4

From the evaluation results of the BNB and MNB models, we can know that whether in scenario 1 or in scenario 2, the prediction effect of the model using all vocabulary is better than the prediction effect obtained by using the first 1000 words of frequency, indicating that even if There are words in the top 1000 words that do not represent the buyer's emotional color.

4. Evaluate the effect of preprocessing for the three standard models by comparing models developed with: (a) only the preprocessing described above (standard models), and (b) applying, in addition, Porter stemming using NLTK then English stop word removal using scikit-learn Count Vectorizer.

model	*DT*				*BNB*				MNB			
scenario	1		2		1		2		1		2	
using NLTK	no	yes	no	yes	no	yes	no	yes	no	yes	no	yes
accuracy	0.6	0.6	0.6	0.6	0.5	0.5	0.62	0.60	0.60	0.59	0.66	0.64
Micro-precision	0.6	0.6	0.6	0.6	0.5	0.5	0.62	0.60	0.60	0.59	0.66	0.64
Macro-precision	0.2	0.3	0.4	0.4	0.3	0.3	0.47	0.44	0.35	0.33	0.54	0.47
Micro-recall	0.6	0.6	0.6	0.6	0.5	0.5	0.62	0.60	0.60	0.59	0.66	0.64
Macro-recall	0.2	0.2	0.4	0.3	0.3	0.2	0.45	0.41	0.33	0.29	0.53	0.46
Micro-F1	0.1	0.11	0.63	0.63	0.19	0.17	0.62	0.604	0.261	0.2	0.662	0.644
Macro-F1	0.1	0.0	0.3	0.2	0.1	0.1	0.3	0.31	0.23	0.1	0.4	0.35

Comparing the performance of the three models using NLTK or not using NLTK, it is found that the performance of the model can be improved more or less if NLTK is used. It can be seen that NLTK can remove the words without emotional color in the sentence, so that the model can be trained during training. , when predicting, the features obtained from the existing vocabulary are more accurate, so the performance of the model is improved.

5. Evaluate the effect of converting all letters to lower case for the three standard models by comparing models with: (a) no conversion to lower case, and (b) all input text converted to lower case.

model	*DT*				*BNB*				MNB			
scenario	1		2		1		2		1		2	
lower case	no	yes	no	yes	no	yes	no	yes	no	yes	no	yes
accuracy	0.6	0.62	0.63	0.60	0.62	0.58	0.63	0.62	0.62	0.61	0.63	0.68
Micro-precision	0.6	0.62	0.63	0.60	0.62	0.58	0.63	0.62	0.62	0.61	0.63	0.68
Macro-precision	0.2	0.27	0.45	0.41	0.29	0.37	0.45	0.48	0.29	0.37	0.45	0.55
Micro-recall	0.6	0.62	0.63	0.60	0.62	0.58	0.63	0.62	0.62	0.61	0.63	0.68
Macro-recall	0.2	0.23	0.41	0.39	0.25	0.32	0.41	0.47	0.25	0.35	0.41	0.55
Micro-F1	0.1	0.10	0.63	0.60	0.14	0.23	0.63	0.62	0.14	0.27	0.63	0.68
Macro-F1	0.1	0.08	0.31	0.292	0.12	0.233	0.31	0.357	0.12	0.25	0.31	0.41

When all letters are converted to lowercase, the performance of the model is degraded. In a sentence, only the first letter will be capitalized, so if all letters are converted to lowercase, the sentence will lose the position information of the first letter, and the feature representation is incomplete. , making the model less effective.

6. Describe your chosen "best" method for rating prediction. Give new experimental results for your method trained on the training set of 2000 reviews and tested on the test set of 500 reviews. Explain how this experimental evaluation justifies your choice of model, including settings and parameters, against a range of alternatives. Provide new experiments and justifications: do not just refer to previous answers.

model	*Linear SVC*	
scenarios	1	2
accuracy	0.666	0.674
Micro- precision	0.666	0.674
Macro- precision	0.324	0.385
Micro-recall	0.666	0.674
Macro-recall	0.308	0.315
Micro-F1	0.214	0.217
Macro-F1	0.226	0.216

Similar to SVC with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples. This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.
By comparing the performance of each model, Linear SVC's performance is better than others.