Machine Learning (Text Classification)

Experimentation, Validation and Model Selection

MACHINE LEARNING
FINAL PROJECT (MAY 2024)

Authors: Ioannis Stathakis, Konstantinos Kostis

METHODOLOGY

GOAL : The evaluation and selection of the best performing model, to be used in a text classification task.

MODELS : A total of 5 models were considered: Multinomial Naïve Bayes (MNB), Complement Naïve Bayes (CNB), Support Vector (SVC), k-Nearest Neighbors (KNN) and Neural Network (MLP).

ENVIROMENT : The tests were performed in Python environment, utilizing the scikit-learn library and its useful modules (Pipeline(), GridSearchCV(), etc.).

EXPERIMENTATION : A total of 4 approaches were taken: Grid Search using the initial dataset (for hyper-parameter tuning), Independent Cross Validations (10-fold), Random Sampling of the initial dataset (for class balancing) and finally Up-Sampling of the minority classes (which, combined with Grid Search, proved to produce the most powerful model).

EVALUATION : Evaluation was approached using scikit-learns Classification Reports (Accuracy, Precision, Recall, F1-Measure) and results also explored visually with ROC Curves, Precision/Recall Curves and Confusion Matrices.

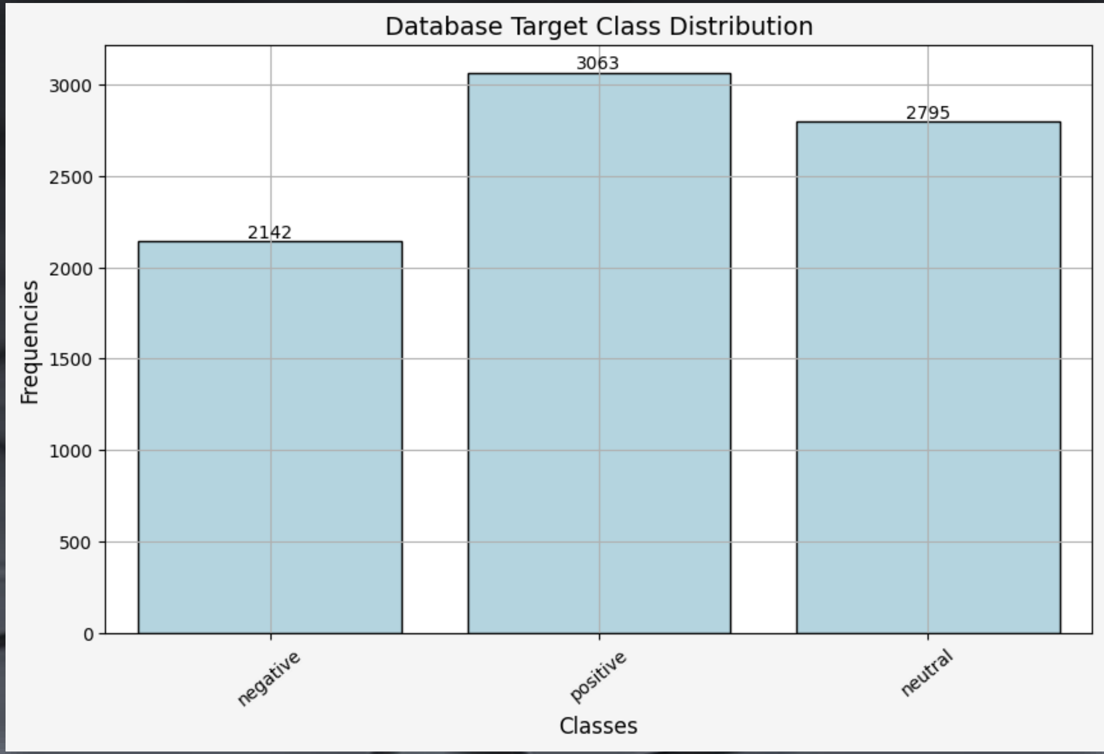
EXPERIMENTATION 1/3

Initial exploration of the dataset showed that the classes were not very noticeably imbalanced, although some differences in their frequencies were obvious.

Experimentation started for all five models using basic sklearn pipelines, utilizing the TF-IDF Vectorizer, with basic English stop words (this yielded better results than the Count Vectorizer in sklearn).

These initial results pointed to the three prevailing models, namely the Multinomial Naïve Bayes, the Complement Naïve Bayes, and the SVC.

These were the models we decided to continue the experiments with, since the other two (the k-NN and the MLP Neural Network) proved rather weak for our purposes.



Further experimentation using Grid Search and various hyper-parameters, improved the performance of all three models, and separated the SVC as the probable best performer of the three. More parameters were refined, such as the use of stop words, which we opted out of. Below are the three resulting Classification Reports for all three models:

Classification Report for the Multinomial Naive Bayes using GridSearchCV():

	precision	recall	f1-score	support
negative	0.56	0.47	0.51	424
neutral	0.55	0.50	0.52	556
positive	0.57	0.68	0.62	620
accuracy			0.56	1600
macro avg	0.56	0.55	0.55	1600
weighted avg	0.56	0.56	0.56	1600

Classification Report for the Complement Naive Bayes using GridSearchCV():

	precision	recall	f1-score	support
negative	0.55	0.57	0.56	424
neutral	0.59	0.47	0.52	556
positive	0.60	0.70	0.64	620
accuracy			0.58	1600
macro avg	0.58	0.58	0.57	1600
weighted avg	0.58	0.58	0.58	1600

Classification Report for the SVC model using GridSearchCV():

	precision	recall	f1-score	support
negative	0.64	0.51	0.57	424
neutral	0.56	0.58	0.57	556
positive	0.64	0.70	0.67	620
accuracy			0.61	1600
macro avg	0.61	0.60	0.60	1600
weighted avg	0.61	0.61	0.61	1600

EXPERIMENTATION 2/3

Separate Cross Validation (10-fold) failed to return better results than the Grid Search (and its internal CV method), as the accuracy levels reached for the three models, the MNB, the CNB and the SVC were 0.547, 0.577 and 0.576, respectively. Below are the complete Cross Validation resulting metrics for this phase:

The MNB Model results are the following:

	Accuracy	Precision	Recall	F1 Measure
Values	0.547625	0.595884	0.514292	0.502319

The CNB Model results are the following:

	Accuracy	Precision	Recall	F1 Measure
Values	0.577375	0.577826	0.568218	0.567771

The SVC Model results are the following:

	Accuracy	Precision	Recall	F1 Measure
Values	0.576125	0.581976	0.561108	0.564108

The next step was to experiment with the balance of the classes, to locate potential model performance differences.

We did this using two different approaches. The first one was to create a new sampled dataset, by randomly selecting 2.000 data points of each class from the original dataset. The Classification Report results of the corresponding Grid Search that followed, are shown to the right.

This approach also failed to return better results than the initial Grid Search, as can be seen in the reports.

Classification Report for the Multinomial Naive Bayes using GridSearchCV() on the sampled dataset:

	precision	recall	f1-score	support
negative	0.56	0.73	0.64	399
neutral	0.58	0.41	0.48	406
positive	0.60	0.60	0.60	395
accuracy			0.58	1200
macro avg	0.58	0.58	0.57	1200
weighted avg	0.58	0.58	0.57	1200

Classification Report for the Complement Naive Bayes using GridSearchCV() on the sampled dataset:

	precision	recall	f1-score	support
negative	0.55	0.77	0.64	399
neutral	0.59	0.37	0.46	406
positive	0.59	0.57	0.58	395
accuracy			0.57	1200
macro avg	0.58	0.57	0.56	1200
weighted avg	0.58	0.57	0.56	1200

Classification Report for the SVC model using GridSearchCV():

	precision	recall	f1-score	support
negative	0.59	0.68	0.63	399
neutral	0.55	0.49	0.52	406
positive	0.60	0.58	0.59	395
accuracy			0.58	1200
macro avg	0.58	0.58	0.58	1200
weighted avg	0.58	0.58	0.58	1200

EXPERIMENTATION 3/3

Our final idea was to use up-sampling to achieve total class balance, and essentially also provide the models with more data, which could prove beneficial performance-wise.

To do this, we used scikit-learns resample() method, to increase the frequencies of the minority classes ("negative" and "neutral"), so that they matched the majority class ("positive"). In this try, we also implemented a custom function for URL elements removal as part of the preprocessing.

This approach proved to be a decisive one, since it yielded an unexpectedly great increase of all performance metrics, for all three models.

Classification Report for the Multinomial Naive Bayes using GridSearchCV():

	precision	recall	f1-score	support
negative	0.75	0.85	0.80	621
neutral	0.75	0.71	0.73	632
positive	0.73	0.68	0.70	585
accuracy			0.75	1838
macro avg	0.75	0.75	0.74	1838
weighted avg	0.75	0.75	0.75	1838

Classification Report for the Complement Naive Bayes using GridSearchCV():

	precision	recall	f1-score	support
negative	0.73	0.86	0.79	621
neutral	0.75	0.70	0.72	632
positive	0.73	0.64	0.68	585
accuracy			0.74	1838
macro avg	0.74	0.74	0.73	1838
weighted avg	0.74	0.74	0.73	1838

To the left, we can see the classification report for the two Naïve Bayes models, the MNB and the CNB. We can easily notice the considerable improvement of all metrics.

Right below, we can see the report for the best performing model, namely the SVC. Again, much higher metrics are evident.

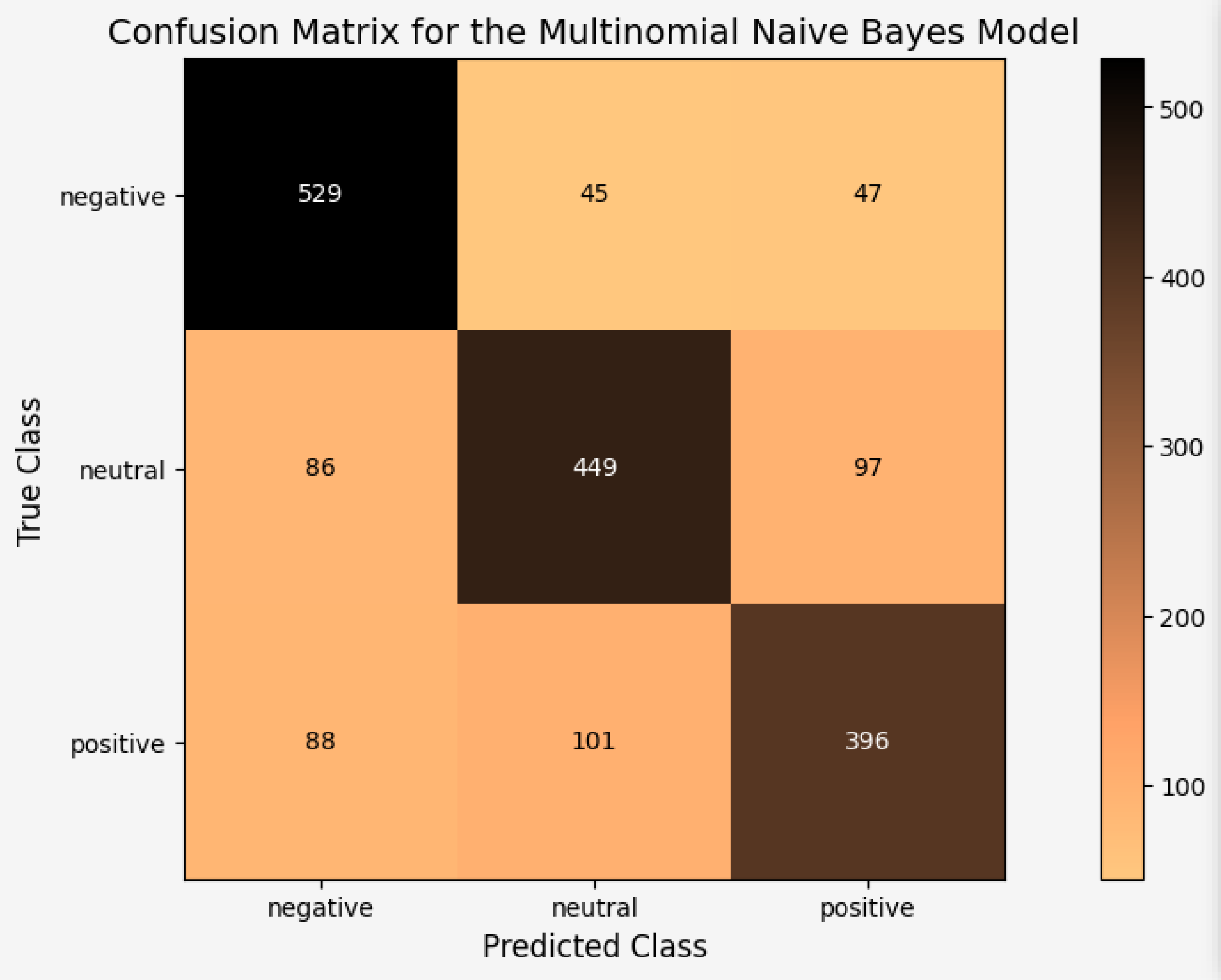
Classification Report for the SVC model using GridSearchCV():

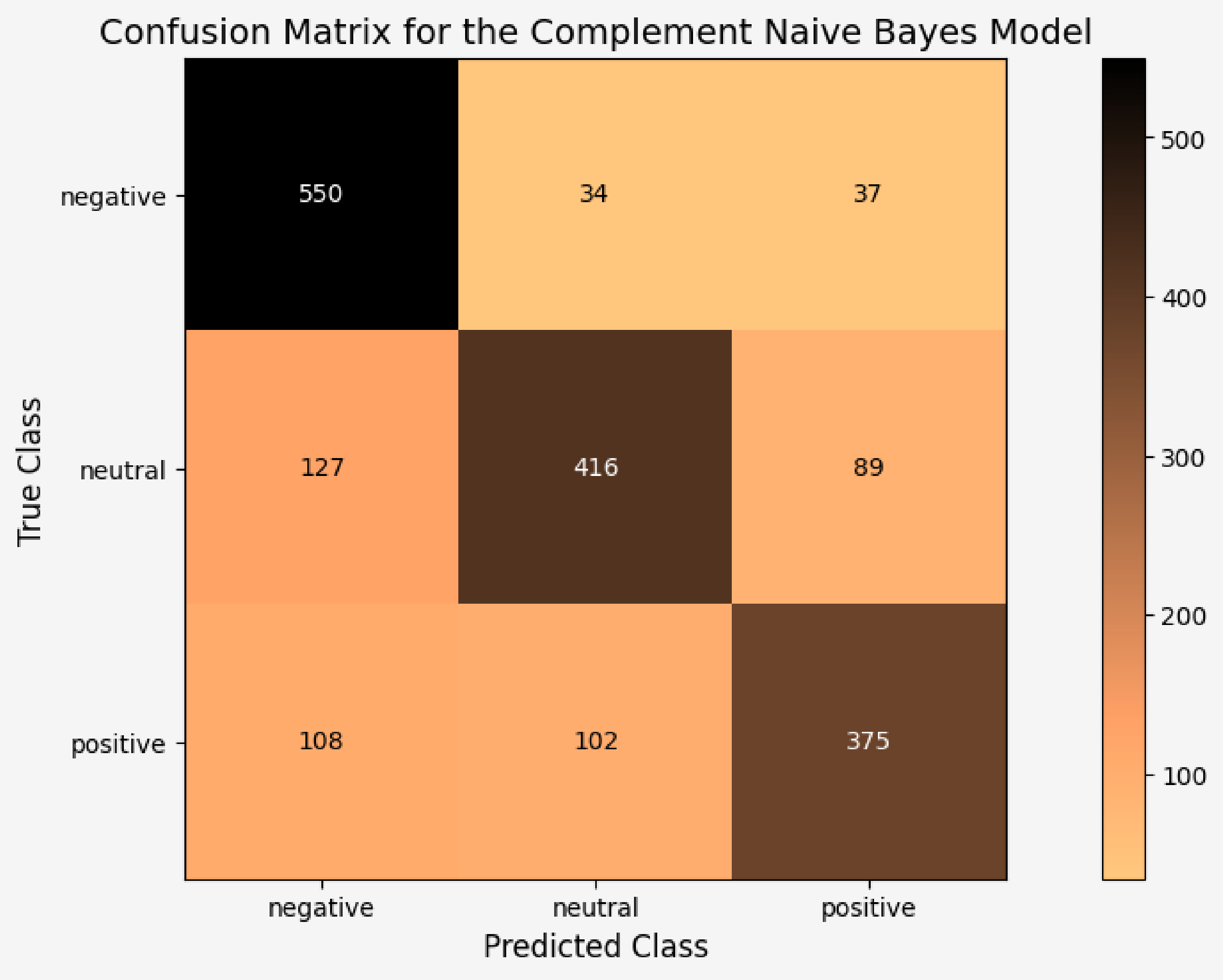
	precision	recall	f1-score	support
negative	0.84	0.85	0.85	621
neutral	0.84	0.76	0.80	632
positive	0.74	0.81	0.78	585
accuracy			0.81	1838
macro avg	0.81	0.81	0.81	1838
weighted avg	0.81	0.81	0.81	1838

MODEL SELECTION 1/2

Further visualizing the results of the last step of the experimentation phase, we can examine Confusion Matrices for the models, starting with the two "runner ups", the MNB and CNB.

We can notice quite good performance on all three classes, especially the "negative" class.





MODEL SELECTION 2/2

Confusion Matrix as well as ROC and Precision/Recall Curves, for the prevailing SVC, also confirm the superiority of this model.

The areas under the ROC Curve show quite satisfactory levels of performance, and the Precision/Recall Curve shows acceptable levels of trade-off between precision and recall. The most problematic class (low precision) seems to be the "positive" class, which means that we expect more incorrect predicted labels, in this class. (although the Precision/Recall Curve is mainly indicative for unbalanced data)

We thus concluded that our model of choice will be the SVC, with regularization parameter C=10, utilizing RBF kernel, and trained on the balanced up-sampled dataset.