Model Performance

Model Name: ScoreClassificationV1 **Test Date:** 17/03/2022 19:21:00 **Creator:** Tobias Rothlin

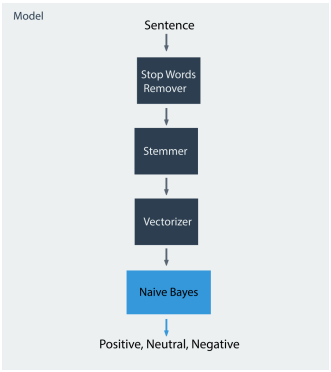
Overview

ML Principle:
Multinomial Naive Bayes

References:

- NultinomialNB Explained
- Stanford NLP Course
- Stanford NLP Lecture
- Englsh Stopwords

Algorithm Description:
The learning algorithm used in this classification is the Multinomial Naïve Bayes. This approach was chosen as it is easy to implement and is computational very efficient. The first step in the classification pipeline is removing all strop words for example 'i', 'me', 'my', 'myself', etc. A list of English stop word is provided by the nltk module. The stop words remover just removes every word that is in the list of stop words. Next the sentence is passed through the stemmer. Stemmers remove morphological affixes from words, leaving only the word stem. This is done with the PorterStemmer class from the nltk module. The final preprocessing step is to vectorize the sentence. This results in a bag of words representation of the sentence. First all the words must be tokenized and then counted. The result will be a numerical feature vector. To generate this vector the CountVectorizer class from sklearn is used. This class implements both tokenization and occurrence counting in a single class. With the sentence now represented in a vector the Naïve Bayes classifier can work with this vector. For the implementation of the Naïve Bayes classifier the MultinomialNB class (sklearn) is used.



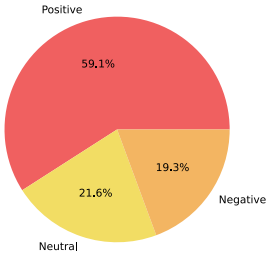
Classification Pipeline

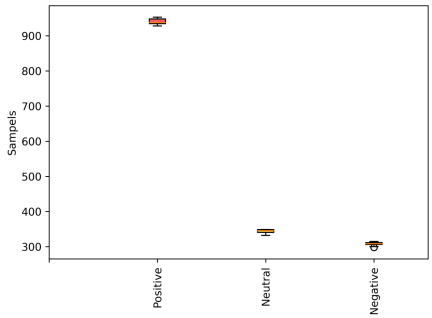
Metrics

Data: ClassifiedDataSetV1.2 with 10 folds cross validation
Split seed: 34.57603

Training Dataset

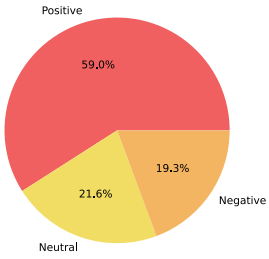
Classes	Number of samples
Positive	940
Neutral	343
Negative	308

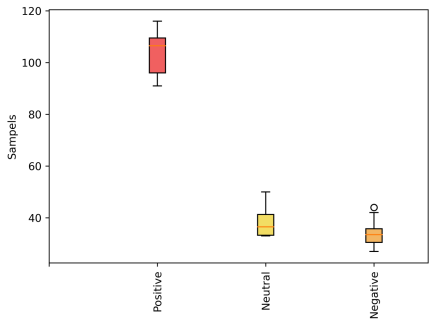


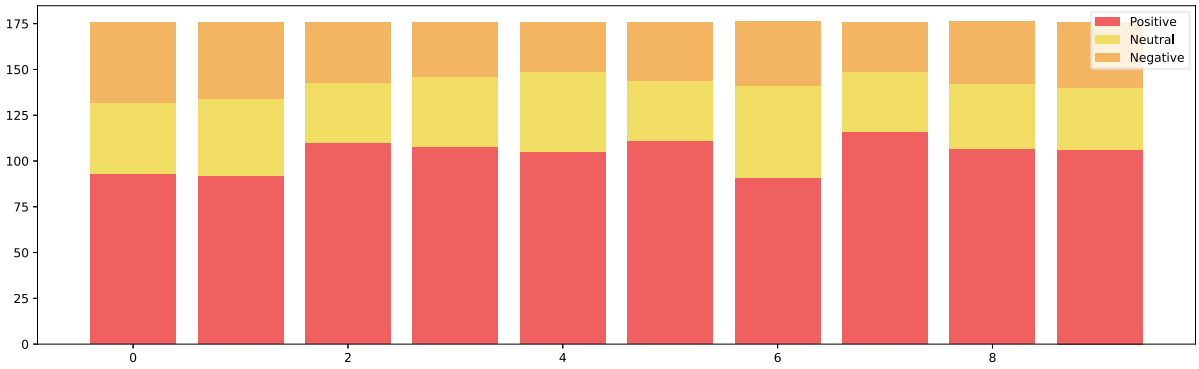


Test Dataset

Classes	Number of samples
Positive	103
Neutral	38
Negative	34

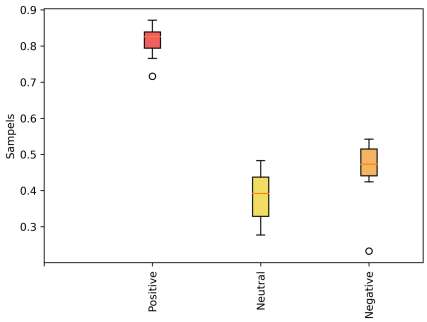




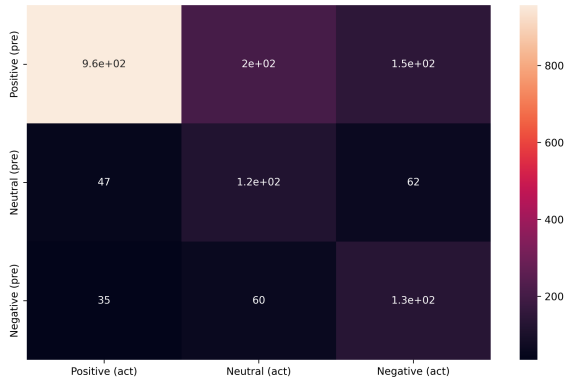


Classification Performance

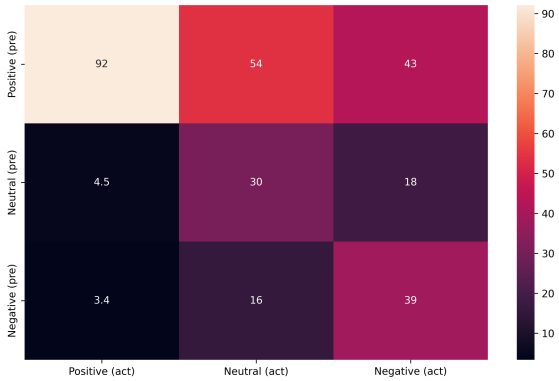
Classes	Precision	Recall	F1 Score
Positive	73.17%	92.11%	81.55%
Neutral	51.56%	30.45%	38.28%
Negative	58.15%	38.82%	46.56%
Accuracy			68.47%
Macro Average	60.96%	53.79%	55.47%
Weighted Average	65.59%	68.47%	65.43%



ConfusionMatrix:



Normalised ConfusionMatrix:



F1 Socre by split: