Introduction

The goal of this experiment is to determine which machine learning model works well for sentence-level sentiment analysis. I am using NLTK, numpy, and scikit-learn and a corpus of text made up of positive and negative movie reviews.

Experimental Procedure

The data was initially preprocessed by a series of nested functions. Punctuation was removed as often, this does not add to the sentiment of the sentence. It also helps to simplify the text we are working with. Similarly, stop words like 'on' and 'the' were removed in order to try and strip the sentence down to its fundamental meaning, by taking away low information words. The choice was made to lemmatize rather than stem to ensure that the processed sentence would remain as close to the original as possible and real words would remain in the dataset. And because time was not a concern in this experiment.
As data was read from the text files, a separate array y was created, assigning a value of 1 to positive sentences and 0 to negative sentences. The collected array of sentences from the original text files was then vectorized and unigram features were extracted using Count Vectorize and labelled X. Here, I tried removing words that didn't occur often, but it didn't appear to make a significant change to my outputs. Both these arrays were then shuffled and split into training (80% of the data) and testing (20% of the data) sets. A validation set didn't appear necessary to complete this task.
The data was then trained on four different models:
Logistic Regression (with linear kernel)
Support Vector Machine
Naïve Bayes
Linear SVM
A Baseline was also used, where a value of 0 or 1 was assigned to sentences with an equal probability. The models were then tested against the testing sets and accuracy was measured to compare the performance of each model.

Range of Parameter Settings Tried

I experimented with using the tfidfVectorizer however this didn't increase the accuracy of naïve bayes, and in most cases, reduced the accuracy score of the models.
I tried lowering the C value (inverse of regularization strength) in the linear regression and support vector models, and whilst this seemed to increase the accuracy score for smaller datasets, it didn't end up being better than Naïve Bayes with the original parameters. Ultimately, after playing around with a few different parameters, in these experiments it ultimately seemed like the final outcome did not fundamentally change, and naïve bayes always came out on top.

Results and Conclusions

Naïve Bayes was the best overall performing model. In these results, the parameters were not changed from the default settings. Linear Regression also seems like in smaller test cases it could perform better than Naïve Bayes, with a low C value.
Data:
Accuracy Classification Score Linear Regression: 0.7613689639006095
Accuracy Classification Score SVM: 0.7524613220815752

```
Accuracy Classification Score Naïve Bayes: 0.7641819034224098
Confusion Matrix for Naïve Bayes:
[[845 234]
 [269 785]]
Accuracy Classification Score Linear SVM: 0.7449601500234412
Accuracy Classification Score Baseline: 0.503047351148617
```