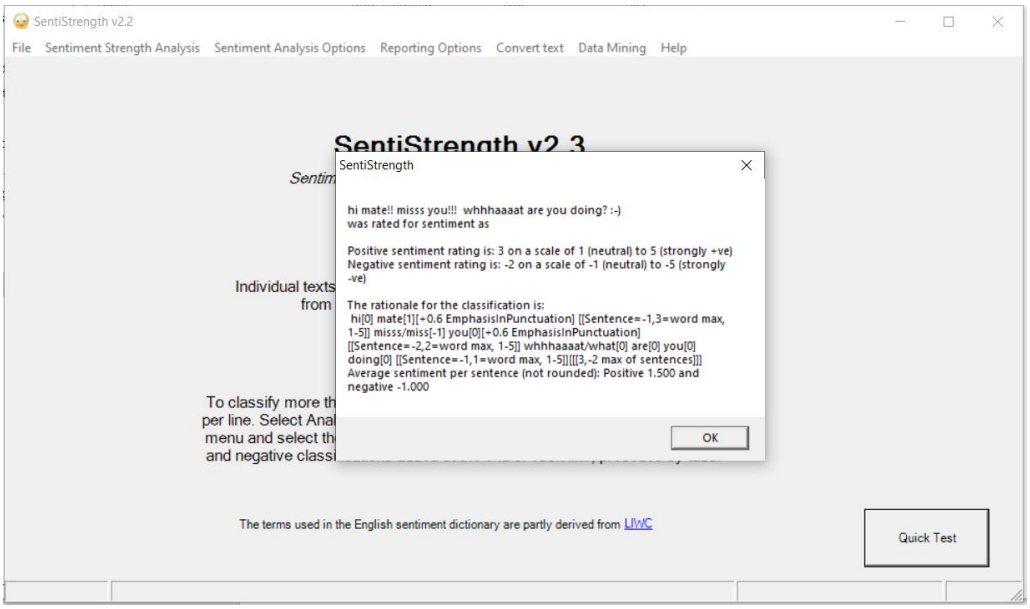
Review-Classifier Model Analysis

A. Pre-Processing of Data:

Upon a through read of the paper published by the author, I understood that multiples binary classifier yields a better precision, recall and accuracy score rather a single multi-class classifier and hence I decided to choose the multiple binary classifier for this project.

The train data is provided the author and is well processed. The test data utilized for this project is a dataset from App-Classifier. This data comprised of some important columns 'Score': indication the score for the app, 'Text': explains the user review. The text column in the data contains junk data, special characters which would restrict us in achieving the best prediction. Hence, the text column was processed in order to remove the unwanted words, characters and transformed into a new column called processed text. Further the test data set also missed few columns that were utilized in building the features (classification techniques) from the train data. Some of these include 'Stop words removal': which removed the stop words from the processed text, 'Lemmatized': lemmatization performed on the processed text, 'Length': indicating the length of words in processed text, 'sentiment score': an average sentiment score and also sentiment score positive and negative: a positive and negative score assigned based on the reviews. We utilized a few libraries from python in order to create these columns in the test dataset. We utilized the stop words English corpus from the "nltk" library to obtain the column without the stop words and similarly utilized the 'wordnet lemmatizer' from the nltk library to obtain column with only the lemmatized words. As suggested by the author, we utilized the "SentiStrength" software in order to obtain a positive and negative sentiment score for the test data. I also took similar approach as the author in order to obtain the single sentiment score which was to select the higher score out of the positive and negative score to check the efficiency.



B. Sampling Size:

The test data has a total of 1048576 records. As we could not manually predict labels for all the records, a small sample set of data was identified using https://www.surveysystem.com/sscalce.htm with a confidence level of 95 and interval of 5. With the 'sample size' set as 384, we obtained the sample data using the sample function in the pandas dataframe. Once this sample dataset was obtained it was manually labelled as described by the author's guidelines. Each review was read and clearly understood and was labeled as either as Bug or Feature or User Experience or Rating. This data which was manually labeled was later used in order to calculate the accuracy of the model which was trained using the author's data.

C. Precision-Recall-F1

1. Bug:

It is to be noted that both the combinations that provided the best accuracy were having stop words removed from the reviews. Both combinations were close to 91% accuracy.

2. Feature:

The prediction metrics for the features also proved that removing the stop words and lemmatizing the words in the review, yields an improved prediction accuracy. The accuracy always turns out to be better when either the entire data is used or when a combination of both the data and the meta data is used. The accuracy is well below when only the meta data is used for predication.

3. User Experience:

While predicting the User Experience, the model performs relatively low by achieving only a maximum of 63% accuracy. This is achieved while performing only bigram of the review, it is to be noted that while removing stop words and lemmatizing the review has deteriorated the perform for this class.

4. Rating:

While predicting the Rating class, we see that the model behaves consistently as it achieves a high accuracy while removing the stop words and lemmatizing the comment. The combination of both the meta data and data has provided a better accuracy for this model.

	Bug			Feature			User Experience			Rating		
Classification Techniques	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
bow-comment	0.877493	0.930514	0.903226	0.942363	0.915966	0.928977	0.641509	0.201183	0.306306	0.962264	0.203187	0.335526
bigram-comment	0.963768	0.401813	0.567164	0.979021	0.392157	0.56	0.487261	0.905325	0.63354	0.778243	0.741036	0.759184
bow-bigram-comment	0.884735	0.858006	0.871166	0.955414	0.840336	0.894188	0.648352	0.349112	0.453846	0.975904	0.322709	0.48503
bow-lemmatized_comment	0.879656	0.927492	0.902941	0.940341	0.927171	0.933709	0.642857	0.213018	0.32	0.963636	0.211155	0.346405
bow-remove_stopwords	0.866848	0.963746	0.912732	0.934783	0.963585	0.948966	0.68	0.100592	0.175258	0.933333	0.111554	0.199288
bow-stopwords_removal_lemmatization	0.869444	0.945619	0.905933	0.935933	0.941176	0.938547	0.666667	0.153846	0.25	0.953488	0.163347	0.278912
bow-bigram-stopwords_removal_lemmatization	0.883382	0.915408	0.89911	0.941176	0.896359	0.918221	0.680556	0.289941	0.406639	0.968254	0.243028	0.388535
rating-comment	0.97037	0.791541	0.87188	0.8306	0.8543	0.963563	0.455556	0.727811	0.560364	0.83	0.86453	0.8421
rating-comment-length	0.895385	0.879154	0.887195	0.931973	0.767507	0.841782	0.57971	0.473373	0.521173	0.90566	0.38247	0.537815
rating-comment-sentiment1-length	0.897516	0.873112	0.885145	0.943262	0.745098	0.832551	0.541935	0.497041	0.518519	0.966667	0.346614	0.510264
rating-comment-sentiment2-length	0.905363	0.867069	0.885802	0.933579	0.708683	0.805732	0.493056	0.420118	0.453674	0.953704	0.410359	0.573816
bow-rating-lemmatized_comment	0.88	0.930514	0.904552	0.940341	0.927171	0.933709	0.634615	0.195266	0.298643	0.963636	0.211155	0.346405
bow-rating-comment-sentiment1	0.876437	0.92145	0.89838	0.942029	0.910364	0.925926	0.692308	0.213018	0.325792	0.962264	0.203187	0.335526
bigram-rating-comment-sentiment1	0.982456	0.507553	0.669323	0.978723	0.386555	0.554217	0.481884	0.786982	0.597753	0.787879	0.7251	0.755187
bigram-rating-stopwords_removal_lemmatization-...	0.953782	0.685801	0.797891	0.977654	0.490196	0.652985	0.471861	0.64497	0.545	0.807692	0.752988	0.779381
bow-bigram-comment-sentiment1	0.884375	0.854985	0.869432	0.955414	0.840336	0.894188	0.651685	0.343195	0.449612	0.97619	0.326693	0.489552
bow-bigram-rating-lemmatized_comment	0.885093	0.861027	0.872894	0.955836	0.848739	0.89911	0.685393	0.360947	0.472868	0.965909	0.338645	0.501475
bow-bigram-remove_stopwords-rating-sentiment1	0.870879	0.957704	0.91223	0.93733	0.963585	0.950276	0.714286	0.177515	0.28436	0.9375	0.119522	0.212014
bow-rating-stopwords_removal_lemmatization-sen...	0.869081	0.942598	0.904348	0.935933	0.941176	0.938547	0.659091	0.171598	0.2723	0.952381	0.159363	0.273038
bow-rating-stopwords_removal_lemmatization-sen...	0.869081	0.942598	0.904348	0.935933	0.941176	0.938547	0.659091	0.171598	0.2723	0.953488	0.163347	0.278912

Figure 1: Accuracy of Classification Techniques using Naive Bayes on App reviews from Apple and Google stores

D. Comparing Author's results

1. Bug

Based on the table below, the comparison between the author's model with that of ours, shows that our model yielded better accuracy with our datasets. A closer look into the metrics, shows the combination of features which has yielded best results for author are the same ones which has yielded us the best result. This clearly indicates that removal of stop words and lemmatizing the words helps in achieving better accuracy for our model.

	Self			Author's		
Classification Techniques	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
bow-comment	0.877493	0.930514	0.903226	0.79	0.65	0.71
bigram-comment	0.963768	0.401813	0.567164	0.68	0.98	0.8
bow-bigram-comment	0.884735	0.858006	0.871166	0.85	0.9	0.87
bow-lemmatized_comment	0.879656	0.927492	0.902941	0.88	0.74	0.8
bow-remove_stopwords	0.866848	0.963746	0.912732	0.86	0.69	0.76
bow-stopwords_removal_lemmatization	0.869444	0.945619	0.905933	0.85	0.71	0.77
bow-bigram-stopwords_removal_lemmatization	0.883382	0.915408	0.89911	0.85	0.91	0.88
rating-comment	0.97037	0.791541	0.87188	0.64	0.82	0.72
rating-comment-length	0.895385	0.879154	0.887195	0.76	0.75	0.75
rating-comment-sentiment1-length	0.897516	0.873112	0.885145	0.69	0.76	0.72
rating-comment-sentiment2-length	0.905363	0.867069	0.885802	0.66	0.78	0.71
bow-rating-lemmatized_comment	0.88	0.930514	0.904552	0.85	0.73	0.78
bow-rating-comment-sentiment1	0.876437	0.92145	0.89838	0.89	0.72	0.79
bigram-rating-comment-sentiment1	0.982456	0.507553	0.669323	0.73	0.98	0.83
bigram-rating-stopwords_removal_lemmatization-...	0.953782	0.685801	0.797891	0.72	0.97	0.82
bow-bigram-comment-sentiment1	0.884375	0.854985	0.869432	0.87	0.88	0.87
bow-bigram-rating-lemmatized_comment	0.885093	0.861027	0.872894	0.88	0.88	0.88
bow-bigram-remove_stopwords-rating-sentiment1	**0.870879**	**0.957704**	**0.91223**	**0.88**	**0.89**	**0.88**
bow-rating-stopwords_removal_lemmatization-sen...	0.869081	0.942598	0.904348	0.88	0.71	0.79
bow-rating-stopwords_removal_lemmatization-sen...	0.869081	0.942598	0.904348	0.87	0.71	0.78

2. Feature

The comparison of metric obtained for Feature with that of the author reveals that features that obtained the best accuracy combination does not match with the best results of the author. Author's models achieved best accuracy with the combination with lemmatized comment, but our model achieved the best accuracy with stop words removed and the lemmatized review.

	Self			Author's		
Classification Techniques	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
bow-comment	0.942363	0.915966	0.928977	0.82	0.59	0.68
bigram-comment	0.979021	0.392157	0.56	0.7	0.99	0.82

bow-bigram-comment	0.955414	0.840336	0.894188	0.87	0.91	0.89
bow-lemmatized_comment	0.940341	0.927171	0.933709	0.9	0.67	0.77
bow-remove_stopwords	0.934783	0.963585	0.948966	0.91	0.67	0.77
bow-stopwords_removal_lemmatization	0.935933	0.941176	0.938547	0.91	0.67	0.77
bow-bigram-stopwords_removal_lemmatization	0.941176	0.896359	0.918221	0.89	0.94	0.91
rating-comment	0.8306	0.8543	0.84	0.74	0.89	0.81
rating-comment-length	0.931973	0.767507	0.841782	0.72	0.82	0.77
rating-comment-sentiment1-length	0.943262	0.745098	0.832551	0.71	0.85	0.77
rating-comment-sentiment2-length	0.933579	0.708683	0.805732	0.67	0.88	0.76
bow-rating-lemmatized_comment	0.940341	0.927171	0.933709	0.9	0.67	0.77
bow-rating-comment-sentiment1	0.942029	0.910364	0.925926	0.92	0.73	0.81
bigram-rating-comment-sentiment1	0.978723	0.386555	0.554217	0.75	0.99	0.85
bigram-rating-stopwords_removal_lemmatization-...	0.977654	0.490196	0.652985	0.75	0.98	0.85
bow-bigram-comment-sentiment1	0.955414	0.840336	0.894188	0.88	0.94	0.91
bow-bigram-rating-lemmatized_comment	0.955836	0.848739	0.89911	0.89	0.94	0.92
bow-bigram-remove_stopwords-rating-sentiment1	**0.93733**	**0.963585**	**0.950276**	**0.87**	**0.93**	**0.9**
bow-rating-stopwords_removal_lemmatization-sen...	0.935933	0.941176	0.938547	0.91	0.72	0.8
bow-rating-stopwords_removal_lemmatization-sen...	0.935933	0.941176	0.938547	0.91	0.73	0.81

3. User Experience

The User Experience prediction in our model yields poor results when compared to that of the author. The highest precision score that is achieved in our model is only 71% whereas the authors model was able to achieve 92%. This could be due to quality of reviews produced from our sample set. However, with recall our model could achieve a maximum of 91%. The maximum accuracy that was achieved in our model was 63% where the author's metrics indicated a maximum of 87%.

	Self			**Author's**		
Classification Techniques	Precision	Recall	F1	Precision	Recall	F1
bow-comment	0.641509	0.201183	0.306306	0.67	0.85	0.75
bigram-comment	**0.487261**	**0.905325**	**0.63354**	**0.91**	**0.62**	**0.73**
bow-bigram-comment	0.648352	0.349112	0.453846	0.85	0.89	0.87
bow-lemmatized_comment	0.642857	0.213018	0.32	0.73	0.91	0.81
bow-remove_stopwords	0.68	0.100592	0.175258	0.74	0.91	0.81
bow-stopwords_removal_lemmatization	0.666667	0.153846	0.25	0.75	0.9	0.82
bow-bigram-stopwords_removal_lemmatization	0.680556	0.289941	0.406639	0.85	0.9	0.87

rating-comment	0.455556	0.727811	0.560364	0.72	0.34	0.46
rating-comment-length	0.57971	0.473373	0.521173	0.7	0.68	0.69
rating-comment-sentiment1-length	0.541935	0.497041	0.518519	0.71	0.66	0.68
rating-comment-sentiment2-length	0.493056	0.420118	0.453674	0.69	0.67	0.68
bow-rating-lemmatized_comment	0.634615	0.195266	0.298643	0.73	0.89	0.8
bow-rating-comment-sentiment1	0.692308	0.213018	0.325792	0.75	0.93	0.83
bigram-rating-comment-sentiment1	0.481884	0.786982	0.597753	0.92	0.69	0.79
bigram-rating-stopwords_removal_lemmatization-...	0.471861	0.64497	0.545	0.92	0.72	0.81
bow-bigram-comment-sentiment1	0.651685	0.343195	0.449612	0.83	0.87	0.85
bow-bigram-rating-lemmatized_comment	0.685393	0.360947	0.472868	0.84	0.9	0.87
bow-bigram-remove_stopwords-rating-sentiment1	0.714286	0.177515	0.28436	0.83	0.89	0.86
bow-rating-stopwords_removal_lemmatization-sen...	0.659091	0.171598	0.2723	0.73	0.9	0.8
bow-rating-stopwords_removal_lemmatization-sen...	0.659091	0.171598	0.2723	0.75	0.9	0.82

4. Rating:

The model while predicting for the rating class yielded good precision, however, has a poor recall score. Hence, this affected the overall accuracy of the model with a maximum accuracy of 77%. This is achieved by the combination bigram with removing the stop words and lemmatizing the words. It is good to note that, the author's accuracy for this class has also been low compared to other classes.

	Self			**Author's**		
Classification Techniques	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
bow-comment	0.962264	0.203187	0.335526	0.76	0.54	0.63
bigram-comment	0.778243	0.741036	0.759184	0.68	0.97	0.8
bow-bigram-comment	0.975904	0.322709	0.48503	0.86	0.85	0.85
bow-lemmatized_comment	0.963636	0.211155	0.346405	0.89	0.65	0.74
bow-remove_stopwords	0.933333	0.111554	0.199288	0.86	0.65	0.74
bow-stopwords_removal_lemmatization	0.953488	0.163347	0.278912	0.87	0.67	0.76
bow-bigram-stopwords_removal_lemmatization	0.968254	0.243028	0.388535	0.86	0.83	0.85
rating-comment	0.83	0.86453	0.8421	0.31	0.35	0.31
rating-comment-length	0.90566	0.38247	0.537815	0.68	0.67	0.67
rating-comment-sentiment1-length	0.966667	0.346614	0.510264	0.66	0.66	0.66
rating-comment-sentiment2-length	0.953704	0.410359	0.573816	0.65	0.72	0.68
bow-rating-lemmatized_comment	0.963636	0.211155	0.346405	0.89	0.64	0.74

bow-rating-comment-sentiment1	0.962264	0.203187	0.335526	0.89	0.6	0.71
bigram-rating-comment-sentiment1	0.787879	0.7251	0.755187	0.71	0.96	0.81
bigram-rating-stopwords_removal_lemmatization-...	**0.807692**	**0.752988**	**0.779381**	**0.7**	**0.94**	**0.8**
bow-bigram-comment-sentiment1	0.97619	0.326693	0.489552	0.85	0.83	0.83
bow-bigram-rating-lemmatized_comment	0.965909	0.338645	0.501475	0.87	0.84	0.85
bow-bigram-remove_stopwords-rating-sentiment1	0.9375	0.119522	0.212014	0.86	0.84	0.85
bow-rating-stopwords_removal_lemmatization-sen...	0.952381	0.159363	0.273038	0.87	0.67	0.74
bow-rating-stopwords_removal_lemmatization-sen...	0.953488	0.163347	0.278912	0.86	0.68	0.76

Observations:

Based on the results that we have obtained, we see that we are able to achieve the same accuracy or better accuracy and precision for particular class like Bug and Feature, however, the User Experience and Rating class have a lower accuracy with a bad precision and recall score respectively. Achieving the same accuracy or precision seems to be possible only for certain classes.

This library does provide a good accuracy in predicting certain class but for other classes accuracy of the model seems to be very low. The model still seems to be robust but, the reproducibility of the model is very difficult. As the author's documentation does not provide a clear idea on what columns are used for this model and how they were created, it is left to the user to investigate and understand the process. Even though the model seems to provide robust results, it is not easily reproducible.

This model provides us with a lot of options for the features however, the paper from the author does not clearly state on how the features are obtained. Manually tracing of the blocks of code is required in order understand the functionality of the library.

This model uses columns like stopwords removed, lemmatized column, stopwords removed and lemmatized column, Sentiment Score, Sentiment Score Positive and Negative, Tense to develop the classification technique. These columns will not be available with most of the review data and needs to be created by the user. The author has provided the details on how to obtain certain column and how they have been used in the algorithm in the paper, however for most of the columns there is no clear documentation of how the column is obtained in the dataset. The paper has also failed to provide us on how the classification technique are formed for the model, it also failed to specify what are the required column in order to obtain a particular classification technique for example the technique Bow-Bigram-Lemmatize utilized the Lemmatized comment column and the technique bow-bigram-stopword-lemmatize utilized the column which is stripped of stop words and has been lemmatized.

Another major set back with the library is utilizing custom data in the model. The model does not provide the user an option to specify where to fetch the test data or how the user could pass a data to classify after training the model. The model defaults to retrieving the data from its own repository and set a 70/30 spilt to train and test and does not let user data to classify. This library still seems to focus only on providing only the precision, recall and accuracy outputs to its user and does not aid them to classify they data.

Thoughts, challenges, learnings, ideas.

This library has a very good idea in classifying the app categories, but it does have a few drawbacks.
Challenges Faced:

1. Understanding the various classification techniques and how they are achieved.
2. Understanding the required column for each classification technique.
3. Lack of feature for the user to input data for classification after training the model.

Learnings:

I learnt the following during this project:

1. How to prepare features for a NLTK classifier
2. How to train and test the NLTK classifies for precision, recall and accuracy.
3. The various combination techniques that could be used to improve model accuracy.

Ideas:

A clearer method for the user to specify the input data to classify, would be helpful in the reproducibility of the model.