Artificial Intelligence - Final Delivery

Natural Language Processing
Amazon Reviews for SA fine-grained 5 classes CSV

Given the Amazon Reviews dataset, process the data, transforming it into a relevant dataset and use it to predict new reviews in a scale of 1-5 stars. For comparison, different supervised learning algorithms should be used, matching them with different ways of generating the dataset.

Grupo: 91_3D
Bruno Rosendo, up201906334
João Mesquita, up201906682
Rui Alves, up201905853

Problem Formulation

- The aim of this problem is to process Amazon Reviews, using diverse techniques such as *tokenization* and *sentence breaking*, to transform them into an useful dataset that can be used to classify new reviews in a scale of 1-5.
- Additionally, different combinations of pre-processing and supervised learning algorithms should be tested and compared regarding the **performance of the test set**, **confusion matrix**, **precision**, **recall**, **accuracy**, **measure** and **time spent to train/test** the models.

Tools & Algorithms

- Programming Language: Python
- Development Environment: Jupyter Notebook / VSCode for development; Git and GitHub for version control.
- Python libraries: *Pandas*, *NumPy/SciPy*, *Scikit-learn*, *NLTK*, Seaborn, *Matplotlib*, *etc*.
- Pre-processing techniques: *Cleanup*, *Tokenization*, *Sentence breaking*, *Syntax parsing*, *Word normalization*, *Lemmatization*, *Stemming*.
- Machine Learning algorithms: *Naive Bayes*, *Decision Trees*, *Neural Networks*, *K-NN*, *SVM*, *Linear Regression*.

Data Pre-processing

The data pre-processing used in this project followed a bag-of-words approach and consists of the following steps:

- Since the dataset is too large (around 3 million rows), make a sample of about 100 thousand rows, making sure the even distribution of classes is maintained.
- Search for null values and remove them. In this case, there are no null values.
- Normalize each entry with the following process:
  - Remove all non-alphanumeric characters with a regular expression
  - Convert all characters to lower case
  - Remove the stopwords from the text, except negation words (Optional)
  - Apply stemming to all the words in the text
- Save the pre-processed data in a new csv file
- Additionally, generate wordclouds for each of the ratings, for a better understanding of the study case.

Data Pre-processing

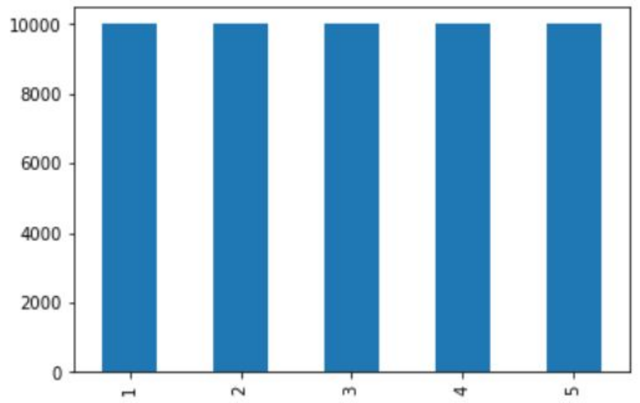


Fig.1 - Distribution of rating classes

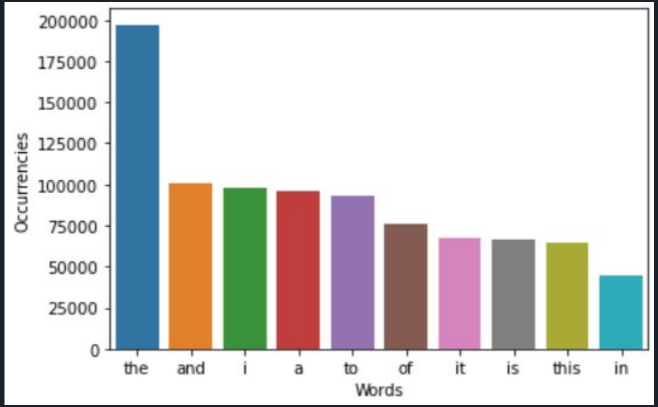


Fig.2 - Total occurrences of words

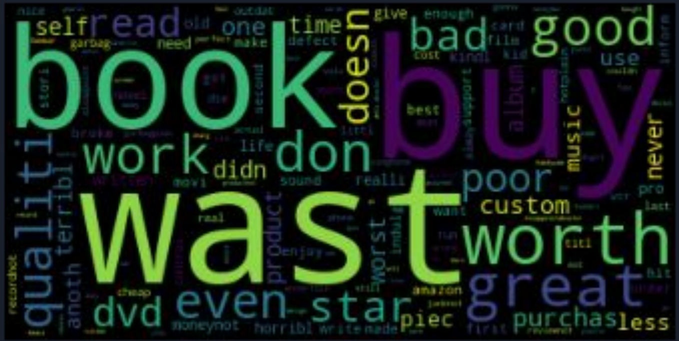


Fig.3 - Wordcloud of 1-star reviews' titles

Developed Models

- In order to try to determine the best model for this case study, we tried different combinations of CountVectorizers, 1-hot Vectors and TF-IDF vectors, together with monograms, bi-grams or tri-grams.
- After constructing the vector, the reviews' titles and texts are concatenated to finally be used by the algorithms.
- There were two different versions of the solver used:
  - Simply run the algorithm with the provided parameters (*predict()*) (More efficient)
  - Run a *GridSearchCV* beforehand to calculate the best parameters for the scenario (*fullPredict()*) (Better Results)
- Since the dataset used is large, the use of tri-grams was difficult due to the lack of computing resources.
- The use of 1-hot Vectors seemed less efficient than the other two models, given its binary view of the text.

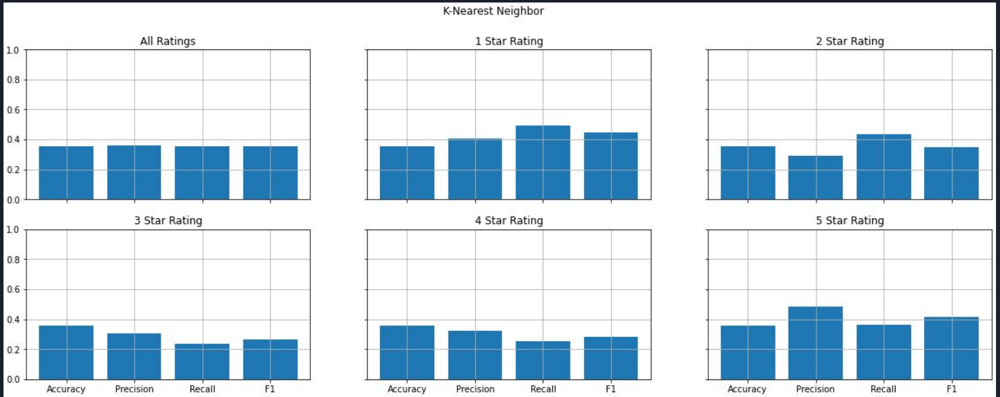


Fig.4 - Accuracy of K-Nearest Neighbor algorithm

Accuracy: 35,525%

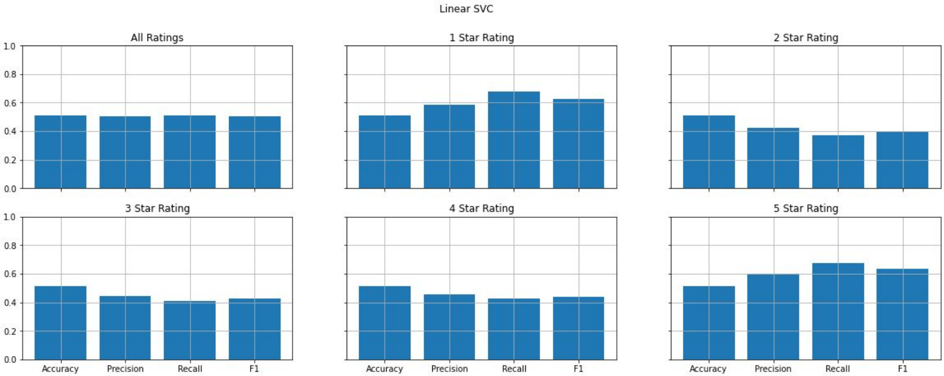
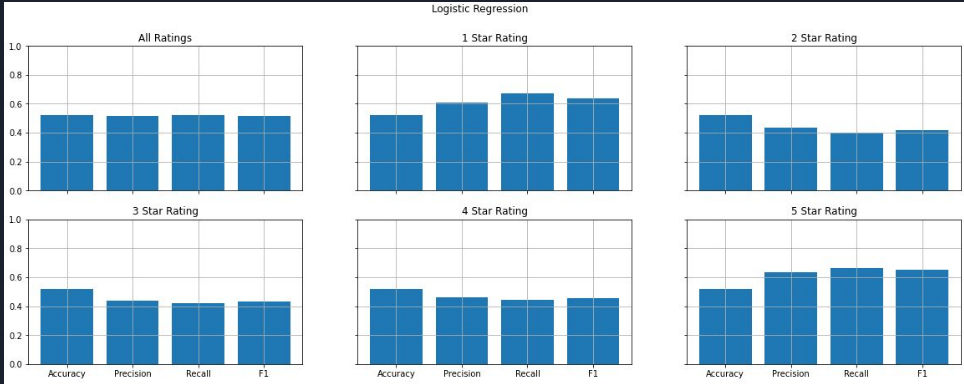


Fig.5 - Accuracy of Linear SVC algorithm

Accuracy:51,1%

Accuracy:52,05%

Fig.6 - Accuracy of Logistic Regression algorithm



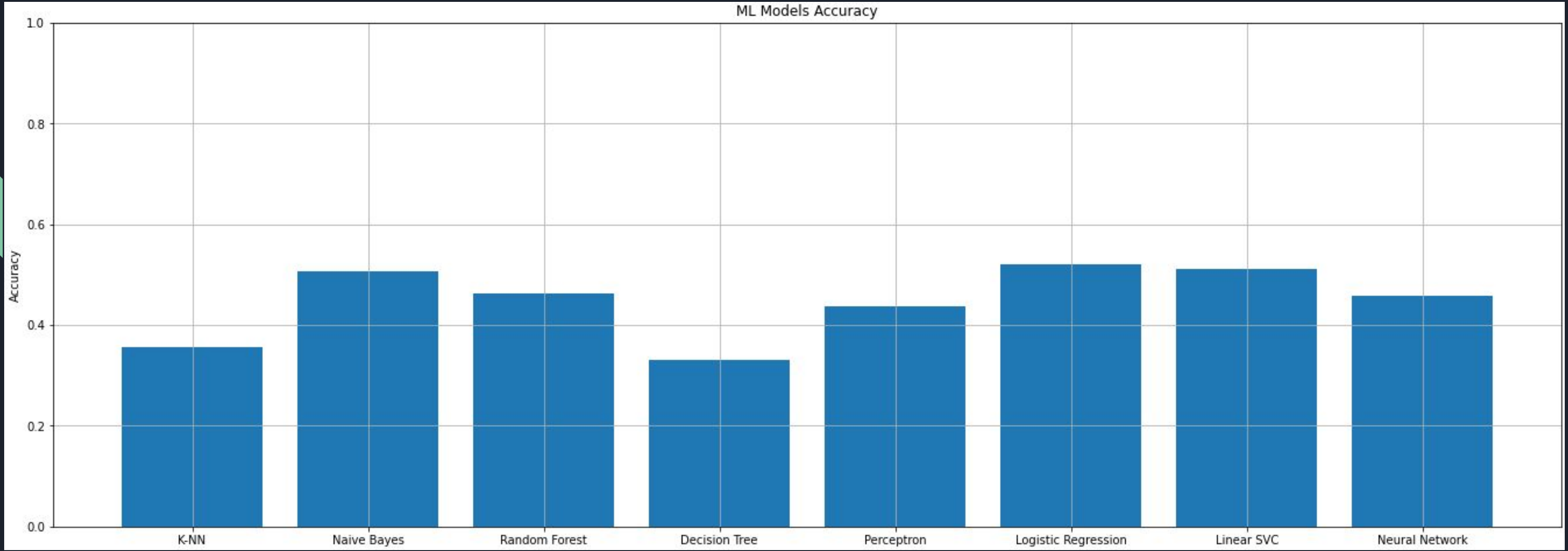


Fig.7 - Models Accuracy

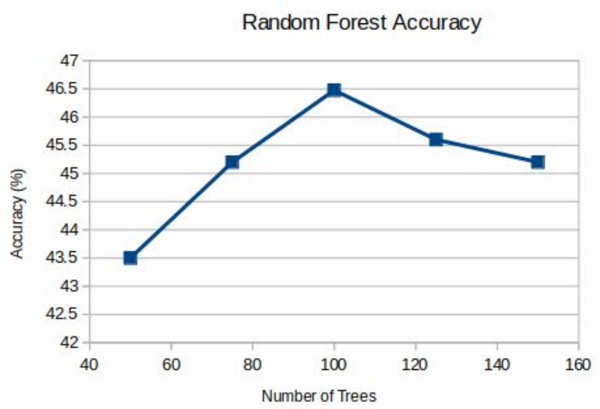


Fig.8 - Random Forest Accuracy

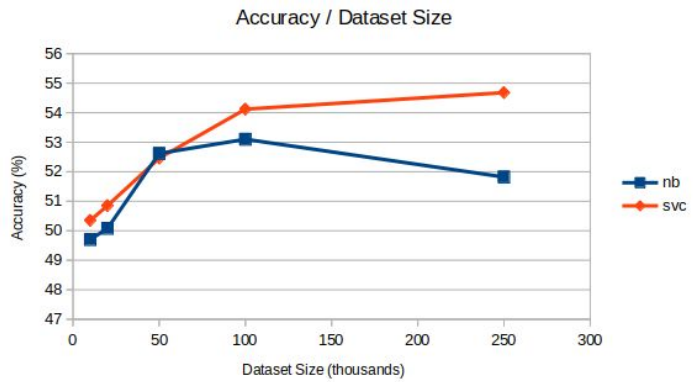


Fig. 9 - Naive Bayes / SVC accuracy for different dataset sizes

Conclusion

- Opposed to our initial expectations, some NLP problems are harder to convert into a high accuracy model, as is the case with this one, since evaluating ratings from 1-5 stars is a task that's not straightforward for humans, hence not exceptionally effective for a computer, because it uses data derived from the former. Consequently, the classes overlap and do not represent opposing ideas.
- Since the dataset used is very large, we expected the Logistic Regression algorithm to perform well. This turned out to be true, it being the algorithm with the best results in the experiments.
- On the other hand, we were not expecting that Naive Bayes and Linear SVC would show results close to the ones gotten from Logistic Regression, since they run much faster than this one.
- The dataset's sample size, word vectors used and the ML model selected have a great influence in the final result. To achieve the best prediction, a combination of these parameters should be used, taking into account the maximum number of features to avoid surpassing the computer's memory capabilities.

Work References

- Natural Language Processing
- Amazon Reviews for SA fine-grained 5 clases
- IART classes’ Slides
- Scikit-learn
- Pandas Documentation
- Treat Negation Stopwords Differently According to Your NLP Task
- NLTK's list of english stopwords