# Amazon Kindle Store Reviews Analysis

**Submitted by**

Team Name: Turing Kings
Team Number: VIT-018

Team Members: -

1) Aditya Deepak Joshi
   Email: adityadeepak.joshi2019@vitstudent.ac.in
   Regno: 19BCE0257
2) Abhinav Gorantla
   Email: abhinav.gorantla2019@vitstudent.ac.in
   Regno: 19BCE0241
3) V L RISHI
   Email: rishi.19bce7553@vitap.ac.in
   Regno: 19BCE7553
4) Atharva Ramgirkar
   Email: atharva.ramgirkar2019@vitstudent.ac.in
   Regno: 19BCE0114

# Introduction

With the rise of technology, user can directly give reviews about products, brands etc. These reviews play vital role in online shopping as well as help people to determine whether a product is good or not. [1]

These reviews or opinions play a significant impact on the success of a business. It helps companies to improve their products based on analyzing the user feedback. Thus it is very important to correctly analyze this huge amount of text in an effective and accurate way.[1]

Huge amount of text data related to user opinions about products and services are generated every day in the world. But it is not feasible to analyze the sentiment of these vast of texts manually. So, an automated process must be applied to mine these text data and analyze the sentiment effectively as the companies need to use these numerous amounts of data to improve their businesses by drawing more effective marketing analysis, product reviews, public relations etc. [1]

# Problem Statement

Amazon Kindle Store is an e-book e-commerce store for all the book reading hobbyists. Online reviews are a category of product information created by users based on personal handling experience. Online shopping websites endow with platforms for consumers to review products and carve up opinions. The problem is most of the comments from customer reviews about the products are contradicted to their ratings. Many customers will post their comments and forgot to rate the product or not engrossed to rate it. [2]

Sentiment analysis is the process of computationally identifying sentiments expressed in a text, especially in order to determine whether the writer's attitude towards a particular topic or product is positive or negative or neutral. Different natural language processing or text analysis techniques are applied for sentiment analysis. [1]

Sentiment mining plays a very important role in business to understand the opinion of customers to improve the products. Customer also depends on the opinion of others who have bought the products already. Reviews or feedback becomes the deciding factor to buy or sell a product. A rating of the products gives a speedy clarification to pact with the product. We will be using Natural language processing to analyse the sentiment (positive or a negative) of the given review. [2]

# Solution

1. Importing Libraries:

A. We started by importing pandas and numpy as they are elementary libraries for any kind of data manipulation/ analysis. For basic text processing, we import nltk and re and use the supplements for stemming, lemmatization and for adopting the 'bag of words' model (CountVectorizer). To check directories,we import os and use train tests as well. Then for establishing neural networks, we import tensorflow and keras and import pickle in order to save the built models.

B. Then we download useful dependencies such as 'wordnet' and 'stopwords'

C. Finally we initialize objects.

2. Reading the Dataset:

A. We then read a dataset called "kindle_reviews.csv" which contains an already existing database of keywords related to reviews.

B. Then we examine the dataset and filter out null values as they are redundant and don't help in the analysis.

3. Data preprocessing:

A. We appropriately delete unnecesssary columns and combine similar ones to improve on the accuracy of our result.

B. Then we encode the *Target* column of the dataset which shows 0 for positive reviews and 1 for negative reviews.

C. Then using text preprocessing techniques, inconsistencies like special characters, lower cases, etc are removed to make data more uniform.

D. Then we read the 'final_reviews.csv' dataset,which will then be used for the model.

E. Then this dataset is balanced by concatenation of one or more datasets.

4. Building models

A. We define target and predictor variables.

B. Then we vectorize the text.

C. After that, we used the train test split.

D. Finally we build the model by adding input layer, hidden layer, output layer and compile,train and save the model.

5. Predictions and testing the model

A. The saved model is then loaded and is then asked to predict the review based on entering a few keywords.

B. The result is finally obtained and verified manually.

C. If the score is nearer to 0, the keywords indicate a positive review.

D. If the score is nearer to 1, the keywords indicate a negative review.

# Literature Survey

Earlier, various rule-based approaches have been used for sentiment analysis. For example, Hutto and Gilbert [1] presented a simple rule-based model for general sentiment analysis and found better performance than the benchmarks used in their study. But the performance of their proposed model was not compared with neural network based approaches. Popular Social Media website like Twitter has also been used for sentiment analysis. Agarwal et al.examined sentiment analysis on Twitter data by introducing Parts of Speech (POS) features.

Later, Kouloumpis et al. [2] investigated the utility of linguistic features for detecting the sentiment of Twitter messages and showed that part-of-speech features is not useful for sentiment analysis in the microblogging domain.

Wilson et al. [3] presented a new approach
to phrase-level sentiment analysis that first determined whether an expression was neutral or polar, and then disambiguated the polarity of the polar expressions.

Different techniques in combination with sentiment analysis algorithms have also been applied. Liu et al. [4] applied sentiment analysis models for predicting the helpfulness of
reviews, which provides the basis for discovering the most helpful reviews for given products.

Reviewers review history was also considered by some researchers. For example, Basiri et al. [5] considered the comment histories of reviewers and found that their proposed system performed better than different algorithms. But they only compared their model with Machine Learning based algorithms and did not compare the performance of their proposed model with neural network-based approaches.

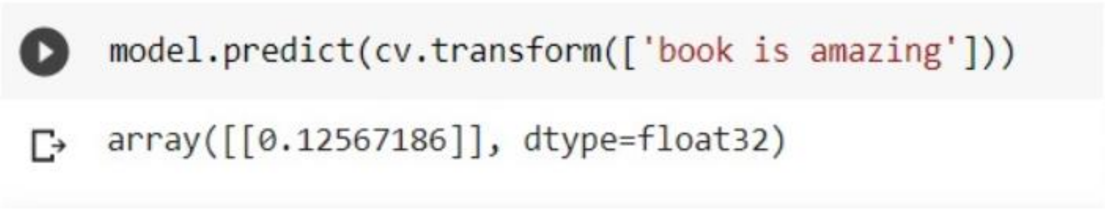
Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. We have used flask for the backend part of our project to work with the model we have developed using NLP (Natural Language Processing).

We have two routes in the project, one GET and one POST. The “/” route renders the index.html file using which the user can submit the book name and review. The post request is sent to the “/predict” route and then the review is put through the NLP model and based on the result either the positive review page or the negative review page is rendered. We also show the image of the book cover on which the user has given their review. This is done using a python package called “isbntools”.

When the user submits the form with the book name, we put it through python isbntools module[6] and use the Google Books API to fetch the book cover image and display it on the results page. We used the google API docs to learn about the Google Books API [7].

# Experimental Investigations

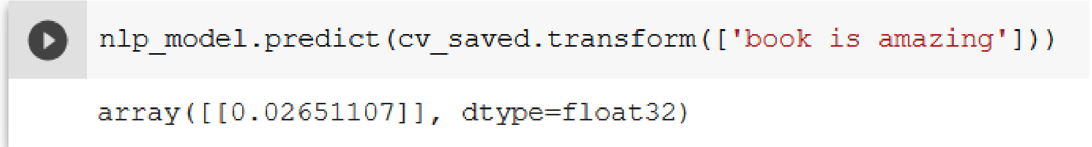
In the project we experimented with different models and made use of the best one.
One example can be seen here when we have only used 10,000 rows to train the model. Logically the prediction should either be wrong or should be less accurate.



```
model.predict(cv.transform(['book is amazing']))
```

```
array([[0.12567186]], dtype=float32)
```

*Model trained with only 10,000 rows*



```
nlp_model.predict(cv_saved.transform(['book is amazing']))
```

```
array([[0.02651107]], dtype=float32)
```

*Model trained with 100,000 rows*

As we can see the model with 100,000 training rows gives a more accurate result.

# Hardware Software Specifications

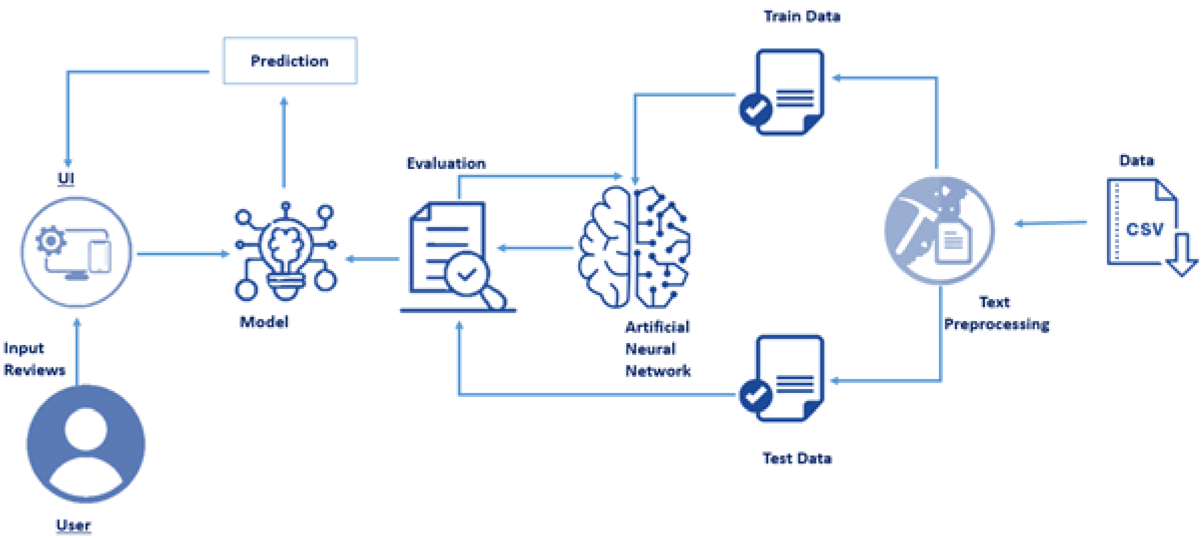
## Hardware specifications used:

1. Processor: Intel® Core™ i5-8250 CPU @ 1.60 GHZ 1.80 GHz
2. Installed RAM: 8.00 GB
3. System type: 64 bit operating system, x64-based processor

## Software specifications used:

1. OS: Windows 10 Home Single Language, build: 19043.1110, version: 21H1
2. For model building: Google Colab(python), Microsoft excel file for datasets, Keras and Tensorflow for making ML based projects.
3. For app building: HTML, CSS used to make webpages, Flask used for building web applications.

# Flowchart

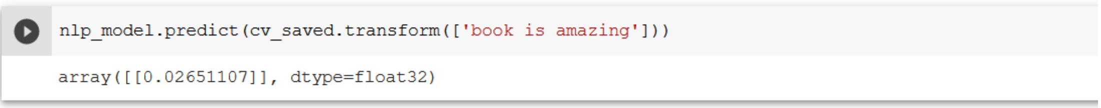

Train Data
Prediction
UI
Evaluation
Data
CSV
Model
Artificial
Neural
Network
Text
Preprocessing
Input
Reviews
Test Data
User

# Conclusion

Thus, the project has been done and the results are as follows:

Case A:

When we use the keywords “book is amazing” we get the result:

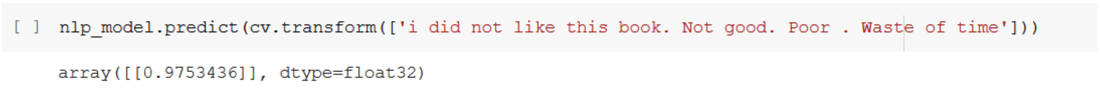


```
nlp_model.predict(cv_saved.transform(['book is amazing']))

array([[0.02651107]], dtype=float32)
```

Case B:

When we use the keywords “i did not like this book. Not good. Poor . Waste of time” we get this result:



```
nlp_model.predict(cv.transform(['i did not like this book. Not good. Poor . Waste of time']))

array([[0.9753436]], dtype=float32)
```

We have previously defined that a score of 0 means the review is positive and a score of 1 means the review is negative.

In Case A, the score generated is closer to 0, thus indicating that the review is positive.

In Case B, the score generated is closer to 1, thus indicating that the review is negative.

Thus, the prediction by the model is right and works properly.

# Future Scope

The basis of this project can be used in larger scale applications handling millions of customer reviews for any product in an online shopping website or even reviews about a particular service like restaurants, car repair shop, etc. The accuracy and range of detection can be expanded upon by including more reviews with keywords and have a broader score range for different levels of customer satisfaction.

# Bibliography

[1] Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In International AAAI Conference on Weblogs and Social Media, AAAI, 216-225.

[2] Kouloumpis, E., Wilson, T. & Moore, J., (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of the Fifth International Conference on Weblogs and Social Media, 538-541.

[3] Basiri, M., Ghasem-Aghae, N., & Naghsh-Nilchi, A. (2014). Exploiting reviewers' comment histories for sentiment analysis. Journal of Information Science, 40(3), 313-328.

[4] Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, ACL, 347-354.

[5] Liu, Y., Huang, X., An, A., & Yu, X. (2008). Modeling and Predicting the Helpfulness of Online Reviews, in Eighth IEEE International Conference on Data Mining, Pisa, 443-452.

[6] https://pypi.org/project/isbntools/

[7] https://developers.google.com/books/docs/v1/using