Binary Sentiment Classification of IMDB Reviews using Artificial Intelligence Algorithms

A Preprint

Jithendra Bojedla
PSU ID: 946374153

Dileep Kumar Boyapati
PSU ID: 985073057

Monika Kamineni
PSU ID: 920433615

March 22, 2024

Abstract

Sentiment analysis involves categorizing text from customer reviews to extract valuable insights for business improvement. Movie reviews offer valuable insights into films, aiding decision-making on whether to watch them. Analyzing sentiments in reviews automates this process, leveraging Artificial intelligence, Machine learning and natural language processing. This involves extracting attitudes and overall polarity from textual data, quantifying favorability. Despite linguistic complexities, neural networks trained on databases like IMDB Movie Review Database, alongside positive and negative word lists, achieve high accuracy, around 85 percent, in opinion mining.

Keywords: Internet Movie Database(IMDB), Naive Bayes, Random Forest, Decision Tree, Gradient Boost, XG Boost, LSTM, BERT, Accuracy.

1 Introduction

Sentiment analysis is crucial for understanding customer feedback, especially from online reviews. It involves parsing textual data to analyze vast amounts quickly, aiding in strategy formulation based on concise yet impactful reviews. By utilizing NLP and Artificial Intelligence algorithms, sentiment analysis accurately categorizes movie reviews as positive or negative, providing valuable insights. Deep learning techniques handle big data effectively, reducing human intervention and improving sentiment analysis models. This study employs both Artificial Intelligence and deep learning to analyze movie reviews from the IMDB dataset, helping recommend movies and guide effective marketing strategies by determining review sentiments. Its primary objective is to extract valuable insights from a significant corpus of textual data, highlighting the importance of sentiment analysis in understanding user review text.

Motivation:
The main motivation of our project is the challenge that is currently facing by many movie review websites. The motivation stems from the laborious and time-consuming nature of manual sentiment classification. Artificial Intelligence model presents a scalable solution to enhance the speed and accuracy of categorization.

2 AI Task

The research focuses on the AI task of binary text classification. IMDB movie reviews serve as inputs, and the model classifies them into two classes: Positive and Negative.
Example -
Input Narrative Sample: Finally was there released a good Modesty Blaise movie, which not only tells a story, but actually tells the "real" story. I admit that it is a bad movie if you expect an action thriller, but if you stop in your track and remove all your expectations. Then you will notice that it is a story that comes very close to the original made by Peter O'Donnell. You have a cover story just to tell about how Modesty became the magnificent person which she is. It is not a movie to attract new fans, but a movie to tell the real tale. Some things could have been better, but when you

cannot forget the awful movie from '66 then is this a magnificent movie. So are you a fan then sit down relax and just enjoy that the real story is there with a cover story just to make Modesty tell her story.
Output: Positive

3 Dataset

3.1 Data Overview

The dataset consists of 50,000 movie reviews sourced from IMDB, a widely-used online database for film and television-related information. These reviews are categorized as either positive or negative sentiments. IMDB offers extensive information on movies, TV shows, and series, making the dataset valuable for sentiment analysis projects. You can access the dataset through the provided link on the Kaggle platform, where you'll find detailed information about its contents, structure, and accompanying metadata. Dataset from Kaggle can be accessed using the below link:

https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

3.2 Proposed Experiments

For this project, we made experimentation on seven different Artificial Intelligence models. Thier description and mathematical representations are shown below:

1. **Long Short-Term Memory**
   LSTM, or Long Short-Term Memory, is a specialized type of recurrent neural network (RNN) developed to overcome the vanishing gradient problem seen in traditional RNNs. LSTM employs unique cell structures comprising three gates: Forget, Input, and Output, along with a Cell State for internal memory retention. These components work together to selectively retain and update information over extended sequences, enabling LSTM to capture long-range dependencies in sequential data effectively. Trained using backpropagation through time (BPTT), LSTM finds applications in diverse fields like natural language processing (NLP) and time series analysis, thanks to its remarkable ability to manage sequential data adeptly.
2. **Bidirectional Encoder Representations from Transformers**
   BERT, or Bidirectional Encoder Representations from Transformers, is a pioneering NLP model. It's designed to understand language bidirectionally, overcoming limitations of previous models. Built on transformer architecture, BERT processes entire sequences at once, capturing contextual information comprehensively. Through pre-training on vast text data, BERT learns rich representations of words, enabling it to excel in various NLP tasks like sentiment analysis and question answering. Fine-tuning on specific tasks further enhances its performance. BERT's impact extends beyond its original version, inspiring variants like RoBERTa and DistilBERT, each tailored to specific needs, fueling advancements in NLP.
3. **Multinomial Naive Bayes**
   Firstly it is the extension of naive bayes algorithm. Multinomial naive bayes is a Bayesian learning algorithm where it identifies the tag of a text from articles or emails by using bayes theorem. It is mostly used for discrete counts, which is occurrence of a word in a string. The likelihood of the tag of the text is analyzed for a given sample input and outputs the value with the greatest chance of the tag. Based on its features it calculates the probability of the class. Multinomial Naive Bayes can be used in tasks like classification, filtering, and sentiment analysis. It frequently works well in practice, especially when working with high-dimensional and limited datasets like those found in text analysis, despite its "naive" assumption of feature independence.

   The mathematical representation for Multinomial Naive Bayes is:
   $P(y/x1, x2, ..., xn) = P(y) * P(x1/y) * P(x2/y) * ...... * P(xn/y)/P(x1) * P(x2) * ..... * P(xn)$
   Where,
   P(y/x1, x2,....,xn) is the probability of class y with the features x1,x2,x2 and so on.
   P(y) is the prior probability of class y
   P(xi/y) is the conditional probability of xi given class y
   P(xi) is the marginal probability
4. **Random Forest**
   It is an ensemble-learning algorithm widely used in machine learning. It mostly uses decision trees for efficient prediction of data while training and testing of data. It builds multiple decision trees and combines them for stable prediction of data. In a random forest, values are randomly selected to train and test the data. It is known for solving classification and regression problems. Random forest can be used in many fields, which involve health, money and any kind of analysis.

The mathematical representation for Random Forest is:
$MSE = 1/Nsigma(i = 1toN)(fi - yi)^2$
Where,
N is the total number of data points
fi is the returned values by the model
yi is the real value of the data point

5. **Decision tree**
Decision tree is a supervised learning algorithm, which is also mainly used for regression and classification of data. By continually dividing the dataset into subsets according to the most important feature at each node, it produces a decision-making structure that resembles a tree with possible results. The goal of decision trees is to learn a set of rules that can be applied to new, unobserved data.
Components of decision are root node, leaf node, decision node, splitting, feature selection. As this algorithm uses trees for visualization, it makes the decision-making process easy. Decision tree algorithm is mainly used for predictive performance and overfitting of data.
The mathematical representation for Decision tree is:
Classification (Gini Index)
$Gini(D) = 1 - sigma(PK)$
Regression (Mean Square Error)
$MSD(D) = 1/|D|sigma(ibelongsD(Yi - YD)^2)$
Where,
D is the dataset
PK is the proportion of samples in ‘K’ class
Yi is the target value of the sample for regression
YD is the target value of the sample for regression.

6. **Gradient Boost**
A machine learning method called gradient boosting is applied to both regression and classification problems. It is an ensemble learning technique that combines the predictions of several weak models—usually decision trees—to create a strong predictive model. In particular, gradient boosting is predicated on the idea of building trees one after the other, with each tree fixing the errors of the preceding ones.
Because gradient boosting algorithms like XGBoost, LightGBM, and AdaBoost can create extremely accurate models, they have gained a lot of popularity. It is known for being reliable, effective, and able to manage complicated data, non-linear relationships in the data. But when employing Gradient Boosting, adjusting the hyper parameters and avoiding overfitting are crucial factors to take into account.
The mathematical representation for Gradient Boost is:
$X = X - r * d/dxf(x)$
Where,
X is the input
r is the learning rate
f(x) is the output

7. **XGBoost**
The machine learning algorithm known as XGBoost, or eXtreme Gradient Boosting, is a strong and efficient member of the gradient boosting technique family. It has grown in popularity due to its accuracy, speed, and scalability and is used widely for both regression and classification tasks. XGBoost is a state-of-the-art algorithm which includes several features and optimizations over traditional gradient boosting methods. XGBoost's predictive performance and adaptability have made it a popular choice in a number of machine learning competitions and real-world applications. Due to its availability in several programming languages, such as Python, R, and Java, a wide range of users may use it.
The mathematical representation for XGBoost is:
$Fm(xi) = Fm - 1(xi) + *hm(xi)$
Where,
Fm(xi) is the prediction after mth iteration
is the learning rate
hm(xi) is the weak learner with negative gradient

There are several reasons to experiment with the above specific models.

- Each of the models we used belongs to a different category of artificial intelligence algorithms.
- Different models may perform differently on the same dataset. Experimenting with multiple models allows to compare their performance metrics such as accuracy, precision, recall, F1 score, etc.
- Some models may be prone to overfitting or underfitting on certain types of data. By experimenting with various models, we can observe their behavior in terms of overfitting or underfitting and select a model that generalizes well to new, unseen data.
- Random Forest, Gradient Boosting, and XGBoost are ensemble methods that combine the predictions of multiple base models. Ensemble methods often outperform individual models by reducing overfitting and improving generalization.

Therefore, experimenting with multiple models is a prudent approach in artificial intelligence, as it provides valuable insights into the behavior of different algorithms on a specific task.

3.3 Data Preprocessing

Exploratory Data Analysis(EDA):
Exploratory Data Analysis is the most crucial step in any data analysis process. Understanding about the dataset and the relationship between the features are important.

- Displayed basic information about the dataset, such as the number of rows, columns, and data types.
- Checked for missing values in each column.
- Analyzed how changes in the size of the training data impact the performance of artificial intelliegence models.
- Analyzed feature importance for the vectorized text data to understand which words or features contribute the most to the classification of review categories.

Class consolidation:

- Class consolidation, also known as class imbalance correction, is a technique used when dealing with imbalanced datasets.
- In the context of the IMDB dataset, where sentiment analysis is performed on movie reviews, class consolidation aims to address any significant disparity in the number of positive and negative sentiment labels.

HTML Tag Removal:

- The remove html tags function is implemented using regular expressions to remove HTML tags from the text.
- This is essential because text data may contain HTML tags, especially in datasets scraped from web sources like IMDB. Removing these tags ensures that only the raw text content is considered for analysis.

URL Removal:

- The remove urls function is defined to eliminate URLs from the text using regular expressions. URLs often appear in text data, especially in online reviews or comments.
- Removing them helps in focusing on the textual content and avoids any bias introduced by web links.

Lowercasing Strings:

- Text in the ’review’ column undergoes conversion to lowercase using the str.lower() method. Lowercasing standardizes the text by ensuring that all words are in the same case.
- This is important because it prevents the model from treating words with different cases as distinct entities.

Stop words Removal:

- NLTK’s stop words corpus is utilized to remove common English stop words.

- The remove stop words function is created for this purpose.
- Stop words are common words like "the," "is," "and," etc., that occur frequently in text but often carry little semantic meaning.
- Removing stop words helps in focusing on the more relevant content of the text.

Punctuation Removal:

- Punctuation marks are stripped from the text using the remove punctuations function, which employs Python's string module.
- Punctuation marks such as periods, commas, and quotation marks are important for grammatical structure but may not contribute significantly to sentiment analysis or classification tasks.
- Removing them helps in simplifying the text while preserving its semantic meaning.

Digit Removal:

- Regular expressions are employed to remove numerical digits from the text.
- Digits may appear in text data, such as ratings or numeric values, but are usually not relevant for sentiment analysis or classification.
- Removing them ensures that the analysis focuses solely on textual content.

Lemmatization:

- The text's words are lemmatized using NLTK's WordNet resource, facilitated by the lemmatize words function.
- Lemmatization reduces words to their base or root form, which helps in standardizing the vocabulary and reducing word variations.
- This is crucial for improving the effectiveness of text analysis models by treating related words as the same entity.

Tokenization:

- Tokenization involves breaking down text data from IMDB movie reviews into individual words or tokens.
- This process is essential for preparing textual data for analysis and modeling tasks.

Vectorization:

- Vectorization converts the tokenized text data into numerical vectors, making it suitable for algorithms.
- Techniques like CountVectorizer or TfidfVectorizer are used to represent words by their frequency or TF-IDF scores.

The dataset was further split into training and testing sets to evaluate the performance of different artificial intelligence models. These preprocessing techniques aim to enhance the quality and consistency of the text data, preparing it for subsequent analysis tasks such as sentiment analysis or classification.

4 Methodology

The steps involved in the workflow of reviews classification are :

1. Environment setup
   - The main objective is to prepare the development environment for the Artificial Intelligence Project.
   - Install necessary libraries to develop the models.
2. Gather the relevant data from the Internet Movie Database.

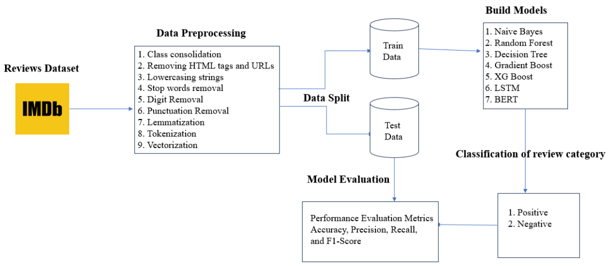


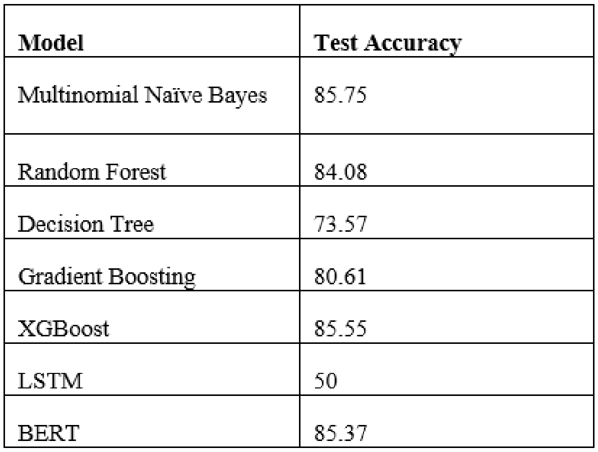
Figure 1: Enter Caption

- Downloaded the IMDB dataset in CSV format which contains 50k rows.

3. Applying various pre-processing techniques to prepare the data.
    - Handle missing data as well as duplicates in the dataset.
    - By using the remove html tags function, removed HTML tags from the text using regular expressions.
    - The remove urls function is used to remove URLs from the text using regular expressions.
    - The text in the 'review' column is converted to lowercase using the str.lower() method.
    - Focus on meaningful words by removal of stopwords.
    - Punctuation marks are removed using the remove punctuations function,
    - Digits are removed from the text using regular expressions.
    - Lemmatization reduces words to their base or root form, which helps in standardizing the vocabulary and reducing word variations.
    - Tokenize the text data to break it into individual words so that the model can understand the data better.
    - Used count vectorization to transform the text into vectors of numbers.
4. Divide the dataset into training and testing sets.
    - Randomly split the preprocessed dataset into a training set and a test set.
    - Based on the size of the dataset, we used an 80-20 split ratio.
5. Build artificial intelligence models for classification.
    - Implemented different types of artificial intelligence models such as Naive Bayes, Random Forest, Decision Tree, Gradient Boost, XG Boost, LSTM, and BERT.
    - Trained these models on the training dataset.
6. Classification of review category.
    - Categorize reviews as either positive or negative based on the model's evaluation.
7. Model Evaluation.
    - Evaluated the model performance using metrics like accuracy, precision, recall, and F1-score.
    - Used a new and unseen dataset to evaluate the model performance.
    - Compared the performance of different models.

5 Results

The evaluation of models in this project was primarily based on their test accuracy. Among the conventional artificial intelligence algorithms, Multinomial Naïve Bayes and XGBoost emerged as the leading performers, achieving accuracies of 85.75 percent and 85.55 percent, respectively. These models demonstrated strong classification capabilities, surpassing Random Forest (84.08 percent), Decision Tree (73.57 percent), and Gradient Boosting (80.61 percent).
In contrast, the deep learning models, LSTM and BERT, displayed differing levels of performance. BERT achieved a competitive accuracy of 85.37 percent, highlighting its effectiveness in natural language processing tasks. However, LSTM lagged significantly behind with an accuracy of 50 percent, indicating a need for further optimization or larger datasets to enhance its predictive capability for the project's objectives.
Overall, the results highlight the effectiveness of ensemble methods like XGBoost and the utility of advanced deep learning architectures such as BERT in achieving high accuracy for classification tasks. However, the choice of model



Model	Test Accuracy
Multinomial Naïve Bayes	85.75
Random Forest	84.08
Decision Tree	73.57
Gradient Boosting	80.61
XGBoost	85.55
LSTM	50
BERT	85.37

Figure 2: Test accuracy obtained for all the seven artificial intelligence models

should consider factors beyond just accuracy, such as computational efficiency, interpretability, and scalability, to ensure suitability for deployment in real-world scenarios. The above figure-2 summarizes the test accuracy for all the seven artificial intelligence models on the same data.

6 Challenges

There are various challenges that we encountered during the project.

- Reviews may include sarcasm that can be challenging for sentiment analysis models to detect accurately.
- IMDB reviews often contain subjective opinions and ambiguous language, leading to variability in sentiment analysis results.
- Negated expressions or comparative statements can significantly alter the conveyed sentiment, requiring models to effectively handle such linguistic constructs.
- Movie reviews often include film jargon and references, making it difficult for sentiment analysis models without domain knowledge to accurately interpret the text.

Conclusion and Future scope

The project introduces a valuable Artificial Intelligence technique that is intended to handle real-world problems. The project included training and testing of seven AI algorithms with Naive Bayes model reaching an impressive 85.75 accuracy. To further improve the performance, advanced approaches such as deep neural networks could be used for the categorization of customer reviews. The project also intends to provide sophisticated modeling tools that are particularly designed for ratings with stars, movie genre and number of people watched, that are part of customer reviews. A systematic process is used to gather user feedback on model predictions, which yields insightful information for ongoing efficiency and improvement in performance. The ultimate goal is better assisting the Internet Movie Database in effectively addressing the user reviews.

References

1. N. T. Thomas, "A LSTM based Tool for Consumer Complaint Classification," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 2349-2351, doi: 10.1109/ICACCI.2018.8554857.
2. Pramod Kumar Naik; Prashanth T; Chandru S; Jaganath S; Sandesh Balan. "Consumer Complaints Classification Using Machine Learning Deep Learning". International Research Journal on Advanced Science Hub, 5, Issue 05S, 2023, 116-122. doi: 10.47392/irjash.2023.S015
3. David Opeoluwa Oyewola, Temidayo Oluwatosin Omotehinwa, Emmanuel Gbenga Dada, "Consumer complaints of consumer financial protection bureau via two-stage residual one-dimensional convolutional neural

network (TSR1DCNN)", Data and Information Management, Volume 7, Issue 4, 2023, 100046, ISSN 2543-9251, https://doi.org/10.1016/j.dim.2023.100046.