IMDB Movie Reviews Sentiment Classification Using Supervised Machine Learning Approaches

Daniela Nogueira

May 2022

Abstract

The internet contains enormous amounts of data. Since it contains a large collection of information, platforms like internet forums, review sites, blog sites, and news sources may be used as data sources. These posts may be classified and used to learn more about the preferences of users. A system that acknowledges and parses widespread comments on the internet is being developed using complex techniques. We can rely on Sentiment Analysis techniques to extract those people's ideas and views about a certain issue from organized, semi-structured, or unstructured textual data. Using the International Movie Database (IMDB), a review driven platform about movies and personalities, where millions of users read and write movie critiques which provides a large and diverse dataset for sentiment analysis, we were charged in this project with integrating multiple classification models to forecast the emotion of IMDb reviews, either positive or negative, just by using the information included in each review. The aim is to identify the model with the best generalization and utmost accuracy. We trained a myriad of models with various combinations of text features and hyperparameter values for both the classifiers and the features, which we discovered might have a considerable influence on performance.

Keywords: Sentiment Analysis, IMDb, Supervised Learning, Hyperparameter, Generalization, Accuracy

I. Introduction

The global web has matured into a ginormous Cyber system that houses a tremendous amount of information which is supplied and consumed by people simultaneously. This cyber system may be leveraged as a database where individuals express themselves on social media platforms such as Facebook, Twitter, Rotten Tomatoes, and Foursquare. Opinions expressed in the form of ratings allow for new discoveries to be made about the online community's collective likes and dislikes, as well as its desires and needs. One such review sector is motion picture show reviews, which have an influence on everyone from the public to film critics to the producing company, in which the International Movie Database (IMDB) comes across as the main platform.

Several other studies have been conducted using this dataset, especially the sentiment analysis studies. What makes IMDb a perfect dataset for our study is the fact that the movie reviews on the website are not professional; they come off as extremely casual in a more fluid, fluent, and occasionally unstructured kind of syntax that accurately reflects emotions.

This effective use of emotive phrases to express a rating prompted us to devise a method for classifying a movie's polarity.

This study will analyze the sentiment of opinions from several comments from IMDB website users and classify them using a final model which will be chosen, based on better performance, among Logistic Regression, k-Nearest Neighbors, Decision Trees, Support Vector Machine, Multinomial Naive Bayes, Stochastic Gradient Bosting, Stochastic Gradient Descent, and Random Forest.

II. Related work

Sentiment classification may be a pretty we tend toll-studied task in machine learning. It is often used to check the outcomes associated, make a case for what we learnt and applied from them. In this related work part of the report, we'll explore the use and the efficiency of different Sentiment Classification models with varied necessary studies.

A large variety of works were meted out antecedently on opinion mining and sentiment analysis.

[Nagamma, P., 2015] planned totally different data processing techniques for classification of moving-picture show reviews and it conjointly predicts the box workplace assortment for the movie. Classification accuracy for simulation was improved well by bunch method. The net movie review knowledge collected from IMDB dataset, the box office collection and the success or failure of the movie is foreseen supported the reviews. [Pang, Alexander, 2010] applied the machine learning technique for classification of reviews gift on IMDB movie reviews database, by forming the list of fourteen keywords that are helpful find the baseline for classification accuracy.

[Martineau and Finin, 2009] presented Delta TFIDF, a strategy for effectively coefficient word values before categorization. Whereas vintage TFIDF focuses on common words that do not appear often in the corpus, Delta TFIDF focuses on phrases that appear frequently in one text but are uncommon in the opposite nominal document. We prefer to take it a step farther by doing so. This is accomplished by multiplying the number of words in the performance by the index of the quantitative relationship between positive and negative coaching material including the keyword. The Delta TFIDF model far outperformed in quality.

[Wang and Manning, 2012] for example, projected using Naive Bayes - Support Vector Machines (NB-SVM), an SVM variation that uses NB log-count ratios as feature values. The authors evaluated the effectiveness of NB, SVM, and NB-SVM on seven distinct datasets that enclosed show reviews, client reviews, opinion polarity, and subjectivity. NB-SVM often outperformed different ways in tasks and datasets. This NB-SVM methodology served as the foundation for a few of our most successful models.

III. Proposed work

Dataset

For this study, I used the IMDb Dataset of Movie Reviews, which is available in many formats on public websites such as Kaggle and others. The one used in this work can be found at https://thecleverprogrammer.com/2020/05/25/movie-reviews-sentiment-analysis-binary classification-with-machine-learning/ and it is also included in the folder that I submitted for evaluation on the Goldsmith platform of this course. The dataset consists of 50,000 movie reviews, each of which is labelled as positive or negative and has no additional features.

Data Preprocessing

In order to conduct a sentiment analysis and process my data in the future, it was necessary to apply certain data filtering techniques to convert the raw data into structured format. The preprocessing included several steps:

- **Data Cleaning**
  - Remove HTML tags
  - Remove Non-Alphanumeric characters
  - Convert words to lowercase
- **Tokenization**
- **Removal of Stop Words**
- **Lemmatization**

In terms of data cleaning, first HTML (Hyper Text Markup Language) tags had to be deleted from each review because they were collected from a website, they naturally contain markers, then non-alphanumeric characters like commas, brackets, and symbols had to be also removed as they do not add any interesting information about sentiment, and finally, each word was converted to lowercase to keep the reviews more consistent.

Regarding Tokenization, this is a text normalization process of breaking down the sentence into words called tokens to ensure more effective manipulation of them. It operates by using space or punctuation to separate the words. For example, an algorithm has no interest in whitespace or line breaks. The tokenization technique produces only words and punctuation as a result.

Stop words commonly used words such as “the”, “a”, and “in” which as non-alphanumeric characters, they are not meaningful for our analysis and need to be eliminated.

Finally, about lemmatization and why it was chosen as a method to analyze the meaning behind words instead of the Stemming. The latter uses the stem of the word, it is a process of removing affixes from words to retrieve the base form, for example, the stem of the terms "caring" and "cares" is care. Lemmatization converts a word into its lemma, or root form. For the word “saw”, Stemming would keep just “s”, while Lemmatization would return “see” or “saw”, depending on whether the word was used as a verb or a noun in the sentence. Lemmatization is more precise than stemming since it takes into account the context of the word in the sentence, whereas stemming normally works on a single word without knowing the context. The words are morphologically analyzed during Lemmatization.

- Review before preprocessing

```
In [7]: data['review'][0]

Out[7]: "One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as
        this is exactly what happened with me.<br /><br />The first thing that struck me about Oz was its brutality and unfli
        nching scenes of violence, which set in right from the word GO. Trust me, this is not a show for the faint hearted or
        timid. This show pulls no punches with regards to drugs, sex or violence. Its is hardcore, in the classic use of the
        word.<br /><br />It is called OZ as that is the nickname given to the Oswald Maximum Security State Penitentary. It f
        ocuses mainly on Emerald City, an experimental section of the prison where all the cells have glass fronts and face i
        nwards, so privacy is not high on the agenda. Em City is home to many..Aryans, Muslims, gangstas, Latinos, Christian
        s, Italians, Irish and more....so scuffles, death stares, dodgy dealings and shady agreements are never far away.<br
        /><br />I would say the main appeal of the show is due to the fact that it goes where other shows wouldn't dare. Forg
        et pretty pictures painted for mainstream audiences, forget charm, forget romance...OZ doesn't mess around. The first
        episode I ever saw struck me as so nasty it was surreal, I couldn't say I was ready for it, but as I watched more, I
        developed a taste for Oz, and got accustomed to the high levels of graphic violence. Not just violence, but injustice
        (crooked guards who'll be sold out for a nickel, inmates who'll kill on order and get away with it, well mannered, mi
        ddle class inmates being turned into prison bitches due to their lack of street skills or prison experience) Watching
        Oz, you may become comfortable with what is uncomfortable viewing....thats if you can get in touch with your darker s
        ide."
```

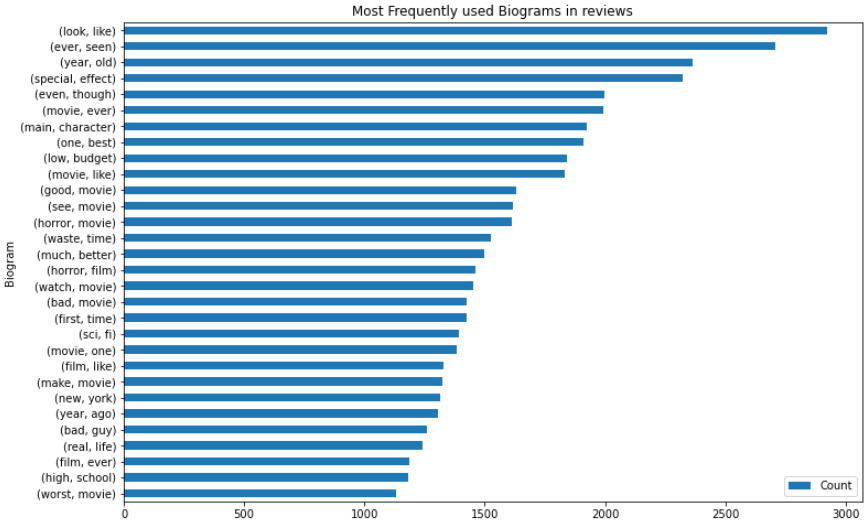
- Review after preprocessing

```
one reviewer mentioned watching 1 oz episode hooked right exactly happened first thing struck oz brutality unflinchin
g scene violence set right word go trust show faint hearted timid show pull punch regard drug sex violence hardcore c
lassic use word called oz nickname given oswald maximum security state penitentary focus mainly emerald city experime
ntal section prison cell glass front face inwards privacy high agenda em city home many aryan muslim gangsta latino c
hristian italian irish scuffle death stare dodgy dealing shady agreement never far away would say main appeal show du
e fact go show dare forget pretty picture painted mainstream audience forget charm forget romance oz mess around firs
t episode ever saw struck nasty surreal say ready watched developed taste oz got accustomed high level graphic violen
ce violence injustice crooked guard sold nickel inmate kill order get away well mannered middle class inmate turned p
rison bitch due lack street skill prison experience watching oz may become comfortable uncomfortable viewing thats ge
t touch darker side
```

Visualization

- Word cloud which illustrates the most frequently used words and the rarely used ones, sized according to their frequency.

- Biograms



Feature Extraction

- **TF-IDF (term frequency-inverse document frequency)**

The method chosen for the selection of features was TFIDF vectorization instead of Bag-of-words. *“TFIDF works by proportionally increasing the number of times a word appears in the document but is counterbalanced by the number of documents in which it is present.*” As a result, terms that appear often in all texts are not given a high score. However, a word that appears frequently in several papers will be given a higher rating because it may be suggestive of the document's context. In other words, Tf-Idf emphasizes rare words while effectively ignoring common ones.

Bag-of-words, on the other hand, is an algorithm that just counts the number of times a word appears in a document without providing any information about the text's meaning. For example, a word bag model would construct the same vectors for the two statements "Student is smart but disorganized" and "Student is disorganized but smart," despite the fact that their meanings are different. TFIDF appears to be more accurate in this regard, as it assesses the relevance of a word to a document in a collection of papers.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Train-Test split and Validation set

Before we can train our algorithms, we must divide the data into two sets: a training set from which the model will learn and a test set from which the model will be tested. However, we should not tweak the model depending on the test set to avoid information leaks; hence we need also set aside a validation set. Setting aside a portion of the training set as the validation set, training on the remaining data, and then evaluating on the validation set. In the test set, we evaluate just once on our final decision model.

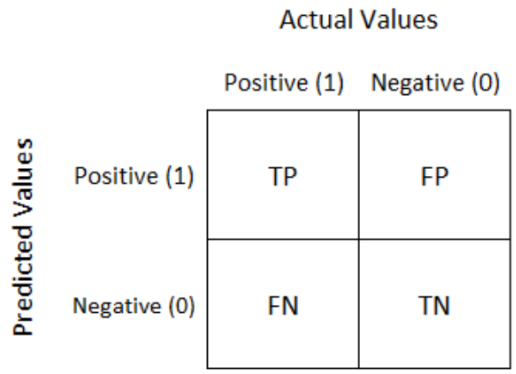
Evaluation Metrics

- **Accuracy**

The **ratio** of correct predictions (true positives and true negatives) among the total number of observed cases. When the target class is well balanced, as it is in our dataset, accuracy is useful.

- **Confusion Matrix**

The confusion matrix produces a matrix that describes the model's performance.



TN, TP, FN, FP are the numbers of instances that are: true negative, true positive, false negative, false positive, respectively.

- **Precision** measures the number of accurately predicted cases that turned out to be positive. It is calculated by dividing the number of genuine positives by the number of predicted positives.
- **Recall** measures how many of the actual positive cases the model properly predicted. It's a valuable metric in fields where False Negative is more problematic than False Positive, such as medicine. Recall is the number of true positives divided by the total number of actual positives.
- **Specificity** is the fraction of accurately predicted negatives over the entire negative prediction produced by the model.
- **F1-score** is the harmonic mean of precision and recall.

- **ROC curve and AUC-ROC**

The **Receiver Operator Characteristic (ROC)** depicts the sensitivity-specificity trade-off displaying the performance of the model across all classification thresholds. Models with curves that are closer to the top-left corner perform better.

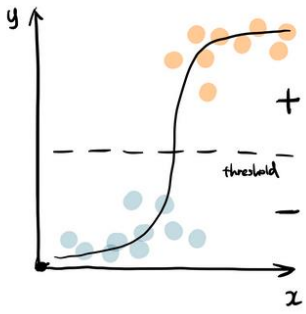
The **AUC (Area Under the Curve)** measures the ability of a model to distinguish between classes. When AUC is 1, the classifier can differentiate all Positive class from all Negative class points accurately. When the AUC is zero, the classifier will forecast all Negatives as Positives and vice versa. The classifier is unable to discriminate between the Positive and Negative classes when AUC equals 0.5.

Classification

- **Logistic Regression**

Logistic regression uses the sigmoid function to return the likelihood of a label. This approach is widely employed when faced with a binary classification task, such as

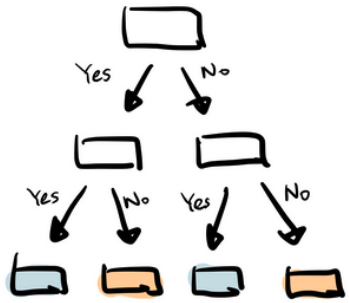
the one we have here. A probability output is generated by the sigmoid function. When the likelihood is compared to a predetermined threshold, the object is assigned a label accordingly.



Hyperparameters turned: penalty, C and solver.

- **Decision Tree**

The decision tree creates hierarchically distributed tree branches, with each branch acting as an if-else condition. Branches form when the dataset is divided into subsets depending on the important features. The decision tree's leaves are where the ultimate classification takes place.



Hyperparameters turned: criterion, max_depth and min_samples_leaf.

- **Random Forest**

Random Forest is an ensemble of decision trees. It brings together the results of numerous predictors. Random forest also employs the bagging technique, which allows each tree to be trained on a random sample of the dataset before getting the majority vote from every tree and making a final decision. It provides more generalization than a decision tree, but because it has more layers, it is less interpretable.

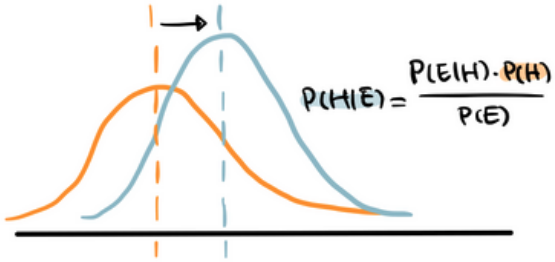
Bagging (bootstrap aggregation) is the process of selecting a random sample of data from a training set with replacement, which means that the individual data points may

be chosen multiple times. To put it another way, bagging is the process of combining numerous randomly selected subsets of the original dataset to train separate learners.

Hyperparameters turned: n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf and n_estimators.

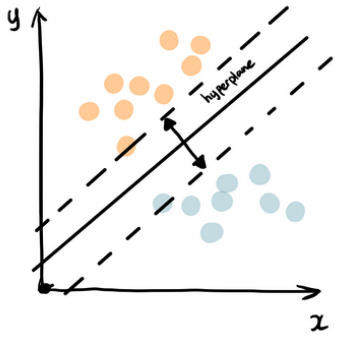
- **Multinomial Naive Bayes**

One of the most basic and effective classification methods is the Naive Bayes classifier. Its foundation is Bayes' theorem, which explains how the probability of an event is calculated using prior knowledge of conditions that may be relevant to the event. The most major advantage of Naive Bayes is that it operates wonderfully even with small amounts of data, whereas most machine learning algorithms require a large amount of training data. The multinomial naive Bayes method is frequently used to classify documents using statistical analysis of their contents. It provides a viable alternative to AI-based semantic analysis and substantially simplifies textual data classification.



- **Support Vector Machine**

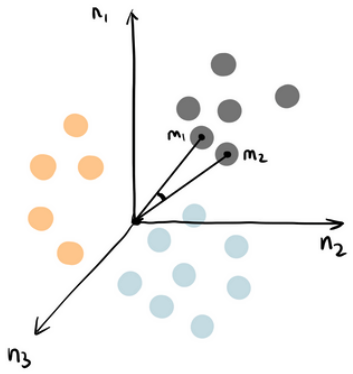
The easiest way to categorize the information supporting the location in relation to a boundary into positive and negative classes is to use the support vector machine. This boundary is interpreted as a hyperplane that maximizes the space between data points from distinct classes.



Hyperparameters turned: Kernel, C, gamma.

- **k-Nearest Neighbors**

The KNN algorithmic method finds the K closest neighbor to a given observation location. Then, using the K points, it assesses the proportions of each type of target variable and forecasts the predicted value with the best ratio.



Hyperparameters turned: n_neighbors, weights and metric.

- **Stochastic Gradient Descent**

A powerful optimization approach for Machine Learning and Deep Learning, Gradient Descent works with a substantial proportion of learning algorithms. A gradient is the slope of a function that measures the degree of change in one variable in response to changes in another variable. A 'stochastic' system or approach is one that is tied to random probability. As a result, in Stochastic Gradient Descent, several samples are picked at random rather than the entire dataset for each iteration.

Hyperparameters turned: alpha, loss, penalty, max_iter and tol.

- **Stochastic Gradient Boosting** (XGBoost)

Boosting is an ensemble learning strategy for minimizing training errors by combining a group of weak learners into a strong learner. A random sample of data is chosen, fitted with a model, and then trained sequentially in boosting, that is, each model attempts to compensate for the shortcomings of its ancestor. With each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule. XGBoost is an implementation of gradient boosting that's designed for computational speed and scale. It makes use of several CPU cores, allowing learning to take place in parallel while training, making it quite fast. XGBoost provides a variety of regularisation penalties to avoid overfitting. It can detect and learn from non-linear data patterns. It is highly flexible and works well with small to medium datasets. Finally, in XGBoost, we can run a cross-validation after each iteration.

Hyperparameters turned: max_depth, n_estimators, subsample, learning_rate, objective and eval_metric.

Hyperparameter Tuning (Cross Validation)

I used two alternative strategies for hyperparameter tuning the models: Grid search and Random search.

For algorithms with more parameters and combinations that would take longer to tune (Stochastic Gradient Boosting and Support Vector Machine), I chose Random search because this method selects hyperparameter combinations at random for a set number of iterations, which in this case was set to 10; for all other cases, I chose Grid search where all possible combinations of the hyperparameters for a given model are used to build models. The performance of each model is tested in both approaches, and the best performing one is chosen.

Experiments and Results

	Model	Accuracy	AUC-ROC
0	Support Vector Machine	0.8161	0.8989
1	Logistic Regression	0.8151	0.8968
2	Stochastic Gradient Descent	0.8147	0.8152
3	Multinomial Naive Bayes	0.8078	0.8078
4	Stochastic Gradient Bosting	0.8064	0.8064
5	Random Forest	0.7992	0.7939
6	k-Nearest Neighbors	0.7376	0.7376
7	Decision Tree	0.6975	0.6975

We can observe that the Decision Tree method performed the worst in the Validation set, with Accuracy and AUC ROC of 0.6975 and 0.6975, respectively. Support Vector Machine and Logistic Regression, on the other hand, had the best results, ordered by higher accuracy, with fairly similar Accuracy and AUC ROC values of around 81 percent and 89 percent, respectively. Both models demonstrated a great capacity to reliably differentiate all positive and negative class points.

Even though these were the two most accurate models, only the Support Vector Machine algorithm was tested and evaluated in the test set, where it achieved an accuracy of 82.28 percent, a very good percentage, indicating that this algorithm provides significantly good results for data never seen before. Additionally, by presenting an Accuracy not so distant and even slightly superior to that observed previously in the Validation set, it is a good indicator of no overfitting and good generalization.

IV. Conclusion and Discussion

According to the conclusions of this study, the goal of analyzing sentiment on the comments of IMDB website users can be accomplished. Furthermore, the accuracy of the SVM classification model used in this study is 82.42 percent. SVM works fantastically on text data. Additional work using deep learning algorithms can be done in future research to try to achieve even higher performance, but for this project, I wanted to investigate the presented classifiers as I had done neural networks in previous projects.

V. References

Ramadhan, N. and Ramadhan, T., (2022). Analysis Sentiment based on IMDB aspects from movie reviews using SVM. Sinkron : Jurnal dan Penelitian Teknik Informatika.

Nagamma, P., H. R. Pruthvi, K. K. Nisha, and N. H. Shwetha. (2015) "An improved sentiment analysis of online movie reviews based on clustering for box-office prediction." In Computing, Communication & Automation (ICCCA), 2015 International Conference on, pp. 933-937. IEEE, 2015.

Pak, Alexander, and Patrick Paroubek. (2010)"Twitter as a corpus for sentiment analysis and opinion mining." In LREc, vol. 10, no. 2010, pp. 1320-1326. 2010.

Wang, S. and Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pages 90–94. Association for Computational Linguistics.

Martineau, J. and Finin, T. (2009). Delta tfidf: An improved feature space for sentiment analysis. In Proceedings of the Third International ICWSM Conference, pages 258–261. Association for the Advancement of Artificial Intelligence.