Project Report:

Sentiment analysis on movie reviews with text processing techniques and hyperparameters optimization

Masters in Artificial Intelligence 2024-2026
Husnain Ali(VR526759)
Saman Naeem(VR529199)

Contents

1. Motivation and Rationale

1.1 Context and Problem Statement

Fundamentally for natural language processing, sentiment analysis is a task that determines the sentiment in textual data. As can be seen, it is of great importance to social media monitoring, product reviews, customer feedback analysis and more.

This project aims at customer sentiment classification on IMDB movie reviews. The reviews are labelled as positive or negative with 50,000. The aim is to develop a classification model to determine the sentiment using machine learning techniques.

1.2 Research Theme & Significance

In such domains, the important aspect is sentiment analysis, which is part of natural language processing (NLP), while user opinions affect decision making. Movie reviews are one of the most influential sources of public sentiment, they influence audience preferences, the box office performance and help the content creators know what the reactions of the viewers are. Manual labelling of sentiment is time and effort consuming; therefore, the automation of sentiment analysis is a valuable contributor as it is scalable with large datasets and saves the time and effort. With thousands of reviews to analyze and classify as positive or negative sentiments, machine learning models are more reliable than human annotators as it is a valuable tool for businesses and entertainment industries.

Sentiment classification techniques also have use beyond movie reviews and are employed in a wide range of industries. Beyond social media sentiment, they are successfully used for e-commerce analysis of customer feedback, monitoring brand reputation, or public and political opinion among other things[1]. With the capability to automatically parse sentiments from textual data, companies, policymakers and researchers are able to base their decisions on public perception[2].

The benefits of sentiment analysis exist although its analysis remains intricate at times. The main difficulty that arises from dealing with text suffers from disorder. Reviews and comments written by people frequently contain slang as well as shortened words and spelling mistakes which creates challenges for models to comprehend the intended meaning.

Another tricky issue is sarcasm. People sometimes say opposite things the exact opposite way they wanted. I slept through the whole movie at the end of this review. This movie was so amazing!" On the surface, awesome may be a good word, however, when you read between the lines, it is clearly negative. Common sense and the emotional understanding of the world that humans naturally have means machines have a hard time catching even these subtle cues. The second is big and sparse data. It becomes very easy for models to deal with thousands of reviews, so every unique word is a new feature that model has to process. Since there are far more words than the model can efficiently learn from, the data is absolutely high dimensional. The issue is to not overload the system with the most important words without losing the essence. Models

generally don’t work very well with new types of text. Movie reviews might work well for a system that gets trained with reviews, but it would not be proper for the product reviews or tweets because the language used is different there. A product review that a TV screen is flat is simply a neutral statement; a movie review saying that the acting was flat would be negative. This is why it is so difficult to make sentiment analysis models general enough to handle all domains. For these problems, researchers will use methods such as improved text processing, applied feature selection, and deep learning models, capable of better understanding contextual information. The reality is, though, that there is a lot further to go in getting machines to truly ‘understand’ human emotions in text.

2. State of the Art (SOTA)

The sentiment analysis has advanced at all these years with different methods and approaches being developed to classify text based on the sentiment. The traditional methods mainly rely on machine learning algorithms like Logistic Regression, Naïve Bayes, and Support Vector Machines (SVM), and so on. Often feature extraction techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) or Bag of Words are used to feed these models with a transformation from text into numbers. These classical techniques are simple and inexpensive computational and are thus very apt for basic sentiment classification tasks[3].

RNNs and LSTMs are special kinds of networks designed for inputting data that comes in sequentially, which is exactly the type of data that textual data is made up of. In the end, these models can capture dependencies between the words, and hence can understand the context. In recent times, Transformer based models such as BERT (Bidirectional Encoder Representations from Transformers) have been able to reach new benchmarks in sentiment analysis through their ability to learn the relationship between words in a much deeper way with the use of large amounts of text data.

However, traditional Machine Learning Models such as Logistic Regression are much simpler to train and need less data but the context they understand is very limited hence they may not perform the task well as compared to the context. Since this is a project, where we use SVM (Support Vector Machines) with TF-IDF features was picked because it strikes an acceptable balance between interpretability and efficiency. Support Vector Machines (SVM) function effectively for text classification applications because they provide capable handling of large dimensional feature areas. SVM operates differently from Logistic Regression since its non-linear decision boundary is enabled through various kernel functions including polynomial and radial basis function. Though it is not as lucid as deep learning models to capture the feelings, it offers an easily understandable structure as well as a good enough performance on this dataset, which makes it a practical solution to sentiment classification tasks when computing resources and data are a constraint.

3. Objectives

The main task of this project is to build a sentiment classification model which is able to correctly predict whether a movie review is a positive or negative review. This is because in the plethora of user generated content in the web, an effective model is indispensable in quickly analyzing and categorizing reviews so that both businesses and consumers can understand public opinion. Model operated to process large datasets of movie reviews, produce accurate sentiment predictions that reveal what sentiment the user expressed through their text, and feed to stakeholders to base decision off of user sentiment.

3.1 Specific Objectives

Several specific steps were taken in order to accomplish this general goal. Then the raw textual data was preprocessed so that any unnecessary noises are removed and the text is standardized for better analyzation. This step involved cleaning HTML tags, special characters, as well as all text is in lowercase to eliminate inconsistencies and ready for further processing. Then, the text was converted to numerical features by using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization which captures the importance of each word as compared to the entire dataset. By parsing this, the model gets an understanding of how important each word is when the sentence is being classified in a sentiment.

The text was processed using Count Vectorization (Bag-of-Words) to create numerical features instead of TF-IDF techniques. Through this method the model converts text into a token-count matrix which allows it to determine patterns of word frequencies between different reviews. Count Vectorization handles word occurrence as its core feature while ignoring word significance like TF-IDF so it becomes a recommended method for this classification challenge.

The training data required Support Vector Machine (SVM) model implementation as the central part of the project. SVM became the selected model because it proved effective for text classification of high-dimensional reviews between positive and negative categories. Text classification benefits from SVM because this algorithm supports non-linear decision boundaries using kernel functions unlike simpler models such as logistic regression. The model achieved assessment through accuracy combined with precision and recall joined by F1-score for complete performance classification evaluation.

The optimization of key parameters including the SVM kernel type and the regularization parameter (C) and n-gram range in Count Vectorization occurred through Grid Search Cross-Validation. The model reached its best performance level through the implementation of hyperparameter selection for optimal results.

The feature importance scoring system of SVM differs from Logistic Regression since it does not offer direct assessments. A different approach was taken instead of studying feature coefficients because the study analyzed performance differences from hyperparameter optimization

4. Methodology

4.1 Dataset Description

In this project I have used the dataset IMDB Dataset of 50K Movie Reviews as a large collection of 50,000 labeled movie reviews[4]. Categorized as either positive or negative and thus serving as a binary classification task, each review is represented. Training the model uses 25,000 reviews and the testing is done on 25,000 reviews for dataset. The text can be short, straightforward or longer elaborate review. All reviews are textual comments of different lengths.

4.2 Data Preprocessing

As textual data is inherently unstructured, it needs a lot of preprocessing before it can be used in machine learning model. Thus, the preprocessing steps of this project are as follows:

4.2.1 Data Cleaning:

The initial stage of preprocessing demands cleaning raw text to eliminate unnecessary content and normalize all data points. The following steps were applied:

Removing HTML tags: The majority of reviews contain HTML elements such as
 tags that fail to enhance sentiment classification tasks.

Removing special characters and non-alphabetic text: Foreign language characters and numbers along with special punctuation were removed from text because they offer no value to sentiment analysis.

Converting text to lowercase: The conversion to lowercase ensures that the words Great match with great for processing purposes.

Lemmatization: The lemmatization method transformed words into their dictionary-based formats instead of applying stemming reduction (running becomes run). The meaning remains intact through this procedure which simplifies words for normalization purposes.

Stopword Removal: The algorithm eliminated routine words including is, the, and, this so as to minimize classifying distractions and enhance performance accuracy.

4.2.2 Feature Extraction

The cleaned textual data moved into numerical features through Count Vectorization (Bag-of-Words) processing. By using this approach text data gets transformed into a matrix that shows token frequency patterns but does not establish weight values like in TF-IDF methodology. The model extracted all common words from text content to create them as features so sentiment classification could focus on the most pertinent terms. A single-word approach served as the foundation of the model since it combined simplicity with the key word frequency analysis.

4.2.3 Splitting Data:

It then splits the cleaned and transformed data into two parts, 80 of which is used for training the model and the rest 20 is used for testing the model. This allows the model to be split up in order to evaluate how well the model generalizes to data which has not seen before.

4.3 Model Selection and Hyperparameters

The project implements Support Vector Machine (SVM) as its classification method for sentiment detection. SVM demonstrates excellent capability to handle datasets with many dimensions thus making it an effective solution for text classification assignments that treat words independently. With SVM kernel functions the system transforms data into extended dimensional space to extract complex associations between words contained in text documents. SVM serves as an effective natural language analysis system because it develops flexible decision boundaries to handle data in which word interactions play a vital role in determining sentiment.

4.3.1 Hyperparameter Tuning (Grid Search Optimization)

The SVM model received optimal performance tuning through the application of Grid Search Cross-Validation (GridSearchCV) to its hyperparameters. The optimization required the evaluation of numerous parameters regarding vectorization for feature extraction along with SVM-specific parameters that influenced classifier behavior. Two essential parameters were studied for vectorization:

- The parameter max_df removes common words from the dataset by evaluating its values between 0.1 to 0.7.
- The study assessed sentiment pattern detection by comparing between unigrams (one-word units) and bigrams (two-word units) and trigrams (three-word units) because of their ngram_range adjustments.

The SVM model deployed the best hyperparameters to process the dataset before performing its sentiment classification evaluation. The model proved very effective in binary sentiment classification because it utilized bigrams combined with an RBF kernel alongside appropriate regularization.

5. Experiments & Results

5.1 Evaluation Protocol

The performance assessment of our SVM-based sentiment classification model employed various evaluation metrics for a complete evaluation. Accuracy served as the main evaluation metric to measure which percentage of reviews was correctly identified. The accuracy metric becomes unreliable when analyzing unbalanced datasets consisting of majority and minority class instances. An evaluation of model performance required further analysis through Precision,

Recall and F1-score assessment. Precision determines the frequency of correct positive reviews among all positive predictions through which analysts can monitor false positive misclassifications made by the model. Recall determines how effectively the model detects actual positive reviews which helps identify instances of false negative errors. The F1-score calculates a single score for model reliability through its harmonic mean between precision and recall evaluation. A Confusion Matrix analysis revealed misclassification patterns to check if the model consistently identified positive reviews as negative ones or negative reviews as positive ones. A thorough evaluation of model performance became possible through implementing various evaluation metrics instead of using only accuracy measures.

5.2 Results and Analysis

The SVM model, which was trained using the Bag of Words (BoW) method, attained an accuracy of 87% using the default parameters. The system performed well at breaking down sentiment categories since precision and recall achieved 0.87 metrics for both positive and negative sentiment reviews. The classification ability proved strong based on an F1-score of 0.87 which includes precision and recall. While SVM delivers high performance, it does not offer the same ability as Logistic Regression to interpret the significance of features learned through coefficients. The model bases its classification through support vectors and kernel transformations which find optimal decision boundaries. Numeric values received more attention since they served as essential components for enhancing model accuracy rather than analyzing important class-defining words.

Metric	Positive Sentiment (1)	Negative Sentiment (0)	Overall
Precision	0.86	0.89	0.87
Recall	0.89	0.85	0.87
F1-Score	0.89	0.87	0.87
Accuracy	-	-	0.87

Table 1: Result of Default parameter

To improve Performance, a pipeline was developed to methodically evaluate different parameter values. By Using RepeatedStratifiedKFold(10splits, 3 repeats) for cross validation GridSearchCV determined the optimal combination. The enhanced model produced an accuracy of 88%, only a 1% enhancement over the default model which is just marginally better than the default model, even with this intensive hyperparameter optimization. This shows that while this adjustment may boost model performance, the default values already provide a strong baseline with a high precision score. The best-performing hyperparameter combination was:

- Vect__max_df = 0.4(remove higher frequent terms)
- This model examines both unigram and bigram tokens through vect__ngram_range (1,2) parameter
- SVM__kernel = rbf (Radial Basis Function for optimal decision boundaries)

- SVM__C = The value of C stands at 10 to enable better generalization as a result of regularization techniques.

Metric	Positive Sentiment (1)	Negative Sentiment (0)	Overall
Precision	0.87	0.88	0.88
Recall	0.89	0.86	0.88
F1-Score	0.88	0.87	0.88
Accuracy	-	-	0.88

Table 2: Result of Hyperparameter

Overall, the results show that while training in default parameters, on a small dataset, it may have restricted its capacity to recognize complicated patterns in the text data. However, given a larger dataset,on hyperparameter tuning using GridSearchCV the model may have performed better. Increasing the accuracy to 88%, demonstrating that, while parameter modification helps enhance performance, dataset quantity is critical in deciding the model's overall efficiency. The model worked well but improvements in the future could even include trying deep learning (e.g. BERT or LSTMs) to exploit more meaning and sentence structure of sentences. Moreover, the classification accuracy may reap further by feature engineering further, like including sentiment-specific bigrams and trigrams, to extracting contextual word relationship.

Conclusions

This project was aimed at building a machine learning model that would be able to classify imbd movie review as positive or negative. This objective was successfully done using Support Vector Machine (SVM) with BoW method by achieving the final model accuracy of 87%. This result shows that SVM is very effective for sentiment classification and bigram features helped capture more relevant word combinations, which in addition with optimized hyperparameters helped in making this approach more successful. The most significant advantage of SVM in this project was the fact that it could deal with high dimensional text data very easily. Unlike models like Logistic Regression which assume linear decision boundary, SVM used Radial Basis Function(RBF) kernel to have more flexible and complex decision boundary, helping in improving the classification performance. Also, hyperparameter tuning with Grid Search Verification made a significant difference to the model’s predictive power. One of the main limitations is that sarcasm is not detectable and contextual word dependencies are not possible. For example, a sentence such as 'This movie was so good that I fell asleep' is one in which the model cannot garner the negative sentiment by the presence of terms that are positive. Like this, the model does not differentiate between Slightly positive as “It’s not bad” and Fully negative as “It’s bad”, since the model works with words independently and without taking into account the sentence structure.

Future work has several potential improvements to increase the accuracy of sentiment classification. A key upgrade would be to replace traditional machine learning architectures like predefined machine learning models or something similar by Deep Learning architectures such as Long short term memory networks (LSTMs) or the popular one now: Bidirectional Encoder Representations from Transformers (BERT). This is to overcome the shortcoming of word independent analysis. The second change would be to add a formation of the feature representation using Word Embeddings such as Word2Vec or GloVe, which provides a dense vector representation of words based on their context.

Finally, it successfully proved SVM with best feature extraction and hyperparameter tuning as a strong baseline model for sentiment classification. Nevertheless, traditional machine learning models have some limitations that could be overcame (i.e., to improve sentiment analysis) with contextual and deep learning approaches. Future work can extend into the hybrid models, use SVM for the case of higher dimensional features processing along with deep learning for contextual understanding to achieve even higher accuracy in the sentiment prediction.

References

[1] B. Liu, *Sentiment Analysis and Opinion Mining*, vol. 5, no. 1. in Synthesis Lectures on Human Language Technologies, vol. 5. Morgan \& Claypool Publishers, 2012. doi: 10.2200/S00416ED1V01Y201204HLT016.

[2] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1–2, pp. 1–135, 2008, doi: 10.1561/1500000011.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL-HLT 2019*, 2019, pp. 4171–4186. doi: 10.48550/arXiv.1810.04805.

[4] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2011, pp. 142–150. doi: 10.3115/2002472.2002491.

[5] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011, [Online]. Available: https://jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf