Journal of Modern Technology and Engineering
Vol.8, No.2, 2023, pp.154-164

COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR SENTIMENT ANALYSIS OF TEXT IN AZERBAIJANI AND ENGLISH

Mehdi Rasul*

Landau School, Baku, Azerbaijan

Abstract. The prediction of sentiment of the text within different business spheres has been one of the challenging problems for a variety of linguistics. In this paper, the sentiment analysis of the texts has been carried out using different machine learning (ML) techniques. Various feature extraction techniques and supervised learning algorithms have been employed on the movie review dataset sourced from the Internet Movie Database and translated into Azerbaijani. Specifically, the techniques utilized encompass Support Vector Machines (SVM), Logistic Regression, Decision Trees, Random Forest, AdaBoost, XGBoost, and Naïve Bayes. The proposed models depict the importance of language corpus that Azerbaijani language lacks by comparing the results obtained from both Azerbaijani and English versions of the dataset.

Keywords: Sentiment Analysis, Machine Learning, Logistic Regression, Support Vector Machines, Naïve Bayes, Feature Extraction, Text Preprocessing.

***Corresponding author:** Mehdi Rasul, Landau School, Baku, Azerbaijan, e-mail: *mehdi@rasul.az*

Received: 26 May 2023; Revised: 15 July 2023; Accepted: 23 July 2023; Published: 31 August 2023.

1 Introduction

Sentiment analysis, also known as opinion mining, is an active area of research in natural language processing (NLP) and computational linguistics. It involves using text analysis and classification methods to identify and extract subjective information such as opinions, emotions, and attitudes from text data. Sentiment analysis has many applications across domains such as business, politics, and social media monitoring. Thanks to sentimental analysis of textual data, companies are currently able to evaluate customer feedback, monitor reputation, forecast future user behavior, etc. which leads to driving business institutions further both performance efficiency and monetary wise.

Contemporarily, the studies have shown that machine learning algorithms, particularly ones using supervised learning and deep learning, produce satisfactory results in automation of the textual sentiment analysis. Recent research explicitly illustrates that sentiment analysis of given texts can be automated relying on the combination of computer science and mathematics, while also improving accuracy.

A well-defined English language corpus for model training helps build powerful as well as highly accurate models for texts in English. Given the extensive array of libraries offering diverse natural language processing techniques in English, an automated system for analyzing texts and extracting meaningful insights or summaries from input paragraphs becomes relatively easy to create. This abundance of useful libraries in computer linguistics for English, combined with numerous natural language processing tools, such as word correction, grammar checking, text generation, word tokenization, etc., allows us to build highly accurate sentiment analysis models. In spite of being widely studied for major languages like English, research on sentiment analysis

of under-resourced languages like Azerbaijani is still relatively limited. Azerbaijani is the official language in the Republic of Azerbaijan spoken by around 30 million people in the world and is a language with distinct linguistic characteristics; however, it still lacks labeled datasets and sophisticated language corpus for training machine learning models in sentiment analysis of the texts. Therefore, building sentiment analysis for text Azerbaijani appears to be quite uniquely challenging due to the lack of built-in NLP techniques and language corpus designed specifically for Azerbaijani language.

The study encompasses a comprehensive comparison of machine learning algorithms. These include familiar approaches such as Logistic Regression, Naïve Bayes, and the Support Vector Machines (SVM) classifier as well as ensemble learning methods like Random Forest, AdaBoost, Extreme Gradient Boosting (XGBoost), among others. These algorithms are evaluated to compare their effectiveness in achieving sentiment classification for Azerbaijani. The same algorithms will also be tested on the English version of the dataset in order to address the importance of built-in language corpus as well as advancement of current applicable techniques for NLP tasks in Azerbaijani.

2 Literature Review

The recent studies have shown that the utilization of machine learning techniques, such as supervised and deep learning algorithms, has contributed to significant advancements in the improvement and automation of sentiment analysis for textual data. In 2013, the research by Neethu and Rajasree on sentiment analysis of twitter using machine learning algorithms produced satisfactory results in terms of classification of tweets into positive and negative classes (Neethu & Rajasree, 2013). The authors applied SVM, Naïve Bayes, Maximum Entropy and Ensemble classifiers to classify the 1200 twitter posts. Of the 1200 posts, 1000 were used for training and the remaining 200 posts were used for testing. 90% accuracy was obtained in SVM, Maximum Entropy and Ensemble classifiers, whilst Naïve Bayes produced 89.5% accuracy. Similarly, another study conducted by Chandra and Jana in 2020 applied both machine learning and deep learning algorithms to the scraped data from Twitter API (Chandra & Jana, 2020). The authors mention that LSTM produced 97% of accuracy; however, Logistic Regression, Multinomial Naïve Bayes and Linear SVM classifiers resulted in an average of 82% accuracy. Despite taking longer time for training, deep learning methods are quite useful in prediction of sentiment in the given texts if the number of provided data for training is satisfactorily enough (Chandra & Jana, 2020). In 2017, Baid et al. used 2000 movie reviews from Internet Movie Database, of which 1000 were positive and 1000 negative reviews (Baid et al., 2017). The authors used the StringToWordVector method to extract the features. Machine learning algorithms used were K Nearest Neighbor (KNN), Naïve Bayes and Random Forest. KNN gave the lowest accuracy measured of 55.30%. Moreover, Random Forest Classifier model was able to predict only 78.65% of the whole test dataset correctly, while for Naïve Bayes, this figure was 81.4% (Baid et al., 2017).

In politics, machine learning is also used to determine the sentiment of the texts. In 2016, Heredia et al. collected the political tweets to predict the U.S. 2016 election (Hasanli & Rustamov, 2019). They collected 3 million location-based tweets related to Donald Trump and Hillary Clinton and trained them on the deep convolutional neural network (CNN) to predict the election results and attained 84% of accuracy score. Additionally, in Indonesia, tweets of @jokowi, account of current President of the Republic of Indonesia, have been scraped and fitted into different machine learning models to find the sentiment, specifically positive and negative labels. The results indicated that Sequential Minimal Optimization (SMO) gave 82.7% of accuracy, precision and recall (Wenando et al., 2020).

Furthermore, the sentiment analysis of the texts in Azerbaijani is limited due to lack of the labeled dataset as well as language corpus. Despite this fact, Rustamov and Hasanli applied

Table 1: Classification Accuracy of Different Approaches

Algorithm	**BOW**	**TF-IDF**
Naïve Bayes	94.00	90.00
SVM	93.00	93.00
Logistic Regression	93.00	93.00

various machine learning algorithms, namely SVM, Logistic Regression and Naïve Bayes, to classify the collected and labeled 12000 Azerbaijani tweets (Hasanli & Rustamov, 2019). The collected tweets were from 01.01.2019 to 30.01.2019. The dataset was obtained from twitter API. Firstly, they cleaned the data by removing unnecessary tweets, dropping duplicates, deleting URLs, hashtags, hyperlinks, usernames, etc. In the next stage, emoji change was applied according to a previously generated artificial dictionary which represent each type of emoji and a corresponding value. Feature extraction, Bag of Words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDF) methods have been applied to the tweets in Azerbaijani. The highest result was attained with Naïve Bayes algorithm using BOW feature method. The results are outlined below in Table I (Hasanli & Rustamov, 2019).

Moreover, (Suleymanov et al., 2020) have conducted a research in text classification for Azerbaijani Language using machine learning algorithms. The dataset consisted of 1082844 news reports from 2019. 301224 news reports were dropped from the dataset due to being duplicates. Also, news containing less than 3 sentences were dropped in the cleaning phase of the data. Additionally, the news having more than 100 sentences, less than 30 characters and more than 10000 characters were removed from the dataset. The authors applied TF-IDF and BOW methods to extract the features from the given dataset. The study stated that the TF-IDF method led to better results since it considers the importance of a word in a document using the TF-IDF transformer. According to the shared results, model which used SVM combined with TF-IDF transformer produced the highest accuracy of 93% while the model using Naïve Bayes with BOW feature extraction produced just 56.53% accuracy score (Suleymanov et al., 2020). Based on the results, Artificial Neural Network was the worst in classifying textual news. This contrasts with the results obtained from other studies, where Artificial Neural Network was among the best performing models.

Also, (Mammadli et al., 2019) collected 3000 news from different online websites of Azerbaijani newspapers and tested three machine learning algorithms, namely SVM, Naïve Bayes and Random Forest using frequency based vectorizer and tf-idf by unigram, bigram and trigram methods. The results illustrate that the highest result is achieved with SVM using tf-idf vectorizer and unigram model. Moreover, the study discovered that the Naïve Bayes classifier achieves its optimum outcome of 95.47 percent when paired with a frequency-based vectorizer and a bigram model. Conversely, the random forest attains its maximal F1-score of 93.33 percent when utilizing a tf-idf-based feature extraction and unigram model (Mammadli et al., 2019).

Generally, machine learning algorithms perform well in classification of texts in both Azerbaijani and English. The major algorithms utilized in text classification problems are Support Vector Machines, Naïve Bayes and Logistic Regression which help to identify the patterns in both type and sentiments.

3 Methodology

3.1 Machine Learning

Machine Learning is a sphere that solves complex prediction problems with the help of Mathematics and Computer Science principles. In general, Machine learning consists of two types: supervised and unsupervised. In supervised algorithms, labeled data is required during the

training process, while in unsupervised learning, the models cluster data into groups by finding similarities among data without the need for labels. In contrast to supervised learning approaches, which are employed to solve regression and classification tasks, unsupervised learning techniques are implemented to contend with clustering objectives. The algorithms, namely Logistic Regression, Support Vector Machines, Naïve Bayes, Decision Tree, RandomForest, AdaBoost and XGBoost, belong to supervised learning type of machine learning algorithms.

3.2 Logistic Regression

Logistic regression is a type of supervised machine learning algorithm that is used for classification tasks. Specifically, it estimates the probability that a data point belongs to one class versus another, and the prediction is made by applying a threshold to the estimated probability. Logistic regression calculates the probability of an event occurring using the logistic function, also known as the sigmoid function (EQ 1). This takes any real number and maps it to a value between 0 and 1, which makes it suitable for converting a linear regression model's raw output into a probability prediction.

$$\sigma\left(x\right) = \frac{1}{1 + \ e^{(-x)}}. \tag{1}$$

To train a logistic regression model, an optimization algorithm such as gradient descent is used to iteratively update the coefficients to minimize error by reaching the local minimum of the cost function and attain the optimal coefficients of the features. Once trained, logistic regression takes new data points as input, calculates the weighted sum using the coefficients, passes this through the logistic function to convert to a probability, and outputs a prediction by comparing the calculated probability to the threshold probability. Typically, threshold is 0.5, but it can vary depending on application of the model and preference of the model developer. Some key advantages of logistic regression are that it is fast, straightforward to train and easy to implement. It performs well on many binary classification tasks like medical diagnosis, spam detection, sentiment analysis, etc. However, it may not be a good fit for more complex data, and it is prone to overfitting without regularization. For these cases, more flexible methods like neural networks are preferred.

3.3 Naïve Bayes

Naive Bayes is a probabilistic machine learning algorithm that classifies the data based on the given features. In other words, the probability of the label is calculated with the given features using Bayesian rule.

$$P(A/B) = \frac{P(B/A) \times P(A)}{P(B)}. \tag{2}$$

For a given set of features, $x_1, x_2, \ldots x_n$ and label y, the Bayes rule is applied as following:

$$\begin{aligned} &P(y|x_1,\ x_2,\ ..x_n) \\ &= \frac{P(x_1|y) \times P(x_2|y) \times ... \times P(x_n|y) \times P(y)}{P(x_1) \times P(x_2) \times ...P(x_n)} \end{aligned} \tag{3}$$

where the formula can be re-written as following (EQ. 4):

$$P(y|x_1, x_2, ..x_n) = \frac{P(y) \times \prod_{i=1}^{n} P\left(x_i \mid y\right)}{\prod_{i=1}^{n} P\left(x_i\right)} \tag{4}$$

Since the $\prod_{i=1}^{n} P\left(x_i\right)$ is a constant expression in the EQ 4., the result is taken as maximum value of the probabilities that represents the target value. Thus:

$$result = argmax(P(y) \times \prod_{i=1}^{n} P(x_i|y)) \tag{5}$$

The Naïve Bayes approach first calculates the probabilities of the label with given the input features. Then, the model takes the label which has more possibility to represent the label with a given input feature values. One of the main advantages of the Naïve Bayes algorithm is its simplicity as understanding the mathematics behind the algorithm and implementing it is relatively easy. Besides, the Naïve Bayes model is trained very quickly which is useful time-wise. In addition, considering the greater volume of the text data where dimensionality is very high, Naïve Bayes algorithm usually provides sufficiently good results. Although Naïve Bayes can outperform other alternatives, in case the categorical values are independent, approach called as Naïve Assumption could lead to the suboptimal cases where the features might be dependent. Besides, considering the algorithm treating each feature equally, the failure in modelling is inevitable as in the real-scenario cases, not every feature might be equally treated in modelling process.

3.4 Decision Tree

Decision trees are a type of supervised machine learning algorithm used for classification and regression tasks. The goal is to create a model that predicts the value of a target variable by using decision rules created during the training process from the input features.

The name "Decision Tree" comes from the structure of the model. Model consists of several node types:

1. root node: the first node, where the division begins.

2. leaf nodes: the last node where no further division takes place.

3. internal nodes: the nodes between root and leaf node in which division happens.

The decision tree algorithm is created by continuous division of the data into groups with highest purity, or entropy. Information gain is calculated for several divisions with different parameters and then the one with highest information gain is selected. The branching continues until data is totally pure or the maximum number of nodes were created. Entropy is measure of purity of data and is calculated with the following formula (EQ 6):

$$H\left(S\right) = \sum_{i}^{[labels]} -P\left(i\right) log_2 P(i), \tag{6}$$

where $p(x)$ is the proportion of records belonging to class x.

Information gain is a value calculated for each split and means the change in entropy. It is calculated with the following formula (EQ 7):

$$IG = H(S_p) - \sum \frac{|S_v|}{S} \times H\left(S_v\right). \tag{7}$$

Advantages of decision tree models are that they are easy to understand, explain and are fast to train. Thus, decision trees can be easily visualized that in turn increases its interpretability. Moreover, it can handle both categorical and numerical data without being preprocessed. The way that the decision trees handle the missing values to some extent is considered another advantageous side of the algorithm. However, it is easily prone to overfitting as the tree gets complex. Additionally, in case any subclass dominates among other types of classes, the decision tree will be constructed as a bias tree that the dominating subclass will be given priority in prediction.

3.5 Random Forest

Random forest models are Bagging type ensemble supervised learning models used for both classification and regression problems. They operate by constructing several decision trees independently during training and outputting the class that is the mode of the classes. In other words, the voting classifier method is applied where the final decision in classification tasks is made based on the most occurrences of the decision of sub-samples. All decision trees in random forest have equal weight; thus, a voting method is applied for final prediction. Due to the reason that Decision Tree can easily be overfitted, the bagging method helps to create independently different models and aggregate the results in order to minimize the risk of overfitting.

The main advantage of the Random Forest is that it decreases variance to minimize overfitting risks. Despite being simple to understand and interpret, it can work with a large number of features without dimensionality techniques being applied. However, it is quite slow as it takes several models to be independently running.

3.6 Adaptive Boosting (AdaBoost)

AdaBoost is a boosting type supervised learning algorithm that creates several dependent models to predict the output. The model combines the weak learners to build a strong classifier. Generally, the boosting technique in machine learning is used to decrease the bias in the learning process. The mis-predicted labels are trained in the next phase of the modelling that in turn makes the models be dependent on each other. In case of the Adaptive Boosting algorithm, the model creates a new column in the provided data, which represents 1/n where n is a number of the rows. In the second stage, the model builds the stump for every provided feature. Among the built stumps, the one having the lowest entropy value is selected. After selecting, the performance is measured with Equation 8.

$$PE = \frac{1}{2} log_e \left(\frac{1 - total_{error}}{total_{error}} \right). \tag{8}$$

Having found performance score, the weights should be decreased for the correctly classified ones while misclassified's weights are increased as following:

1. Incorrectly classified:
$$weight_{new} = weight_{previous} \times e^{PE}. \tag{9}$$
2. Correctly classified:
$$weight_{new} = weight_{previous} \times e^{-PE}. \tag{10}$$

After updating the weights, they are normalized by being divided into their mathematical sum value. The dataset is updated to increase the occurrences of the misclassified samples after what stumps are rebuilt for every column. These steps are repeated until there is no misclassified data left or number of iterations reaches predefined number. Based on these stages, AdaBoost algorithm uses boosting technique where every model is built based on the previous model's mistakes.

The AdaBoost algorithm is quite useful in feature selection process. Moreover, it is less prone to overfitting problem, particularly in case of large datasets. In spite of producing more robust predictions, it is sensitive to noise and outliers. Additionally, it is computationally expensive; thus, training process can be very time-consuming.

3.7 Extreme Gradient Boosting (XGBoost)

Another type of boosting algorithms in ensemble learning is XGBoost algorithm. Particularly, Gradient boosting is an extended format of boosting technique that utilizes gradient descent

optimization algorithm on minimizing loss function to generate additively weak learners. Additionally, the random samples are trained in individual trees which leads to the training process being less timely and complex. Besides, XGBoost algorithm applies compressed column-based structure to sort the structure to be able to find the best split of the tree. As a result, training time is decreased dramatically. Based on these strategies, one of the advantageous sides of XGBoost algorithm is being highly optimized to be efficient and maximize hardware usage. Additionally, since the model has built-in Lasso and Ridge, L1 and L2 regularization respectively, overfitting can be prevented during the training process. Being more resistant to outliers and noisy data compared to other tree-based algorithms makes XGBoost an attractive option for many applications. However, even though it provides the feature importance score, the algorithm is hard to be interpreted from business perspective. Furthermore, it should be noted that it is relatively memory-intensive during the training process. Generally, XGBoost is one of the most used machine learning algorithms. However, to achieve optimal performance, crucial processes such as algorithm tuning and hyperparameter tuning should still be implemented.

Compared to AdaBoost algorithm, XGBoost is more regularized and better at handling overfitting problems than AdaBoost. Moreover, AdaBoost does not support parallel processing; therefore, each weak classifier is built sequentially; Also, XGBoost is useful in terms of parallel processing by supporting out-of-core computing. On the other hand, AdaBoost is simpler to implement and understand compared to XGBoost algorithm. In addition, it requires less hyperparameter. Generally, XGBoost is more efficient than AdaBoost; however, due to better interpretability, AdaBoost algorithm is still used more than XGBoost in business applications.

3.8 Support Vector Machines

Support Vector Machines (SVM) are a type of machine learning algorithm used for both classification and regression problems. The main goal of the SVM algorithm is to find the hyperplane which is able to separate the provided data into classes based on feature space. The hyperplane is N-1 dimensional flat affine subspace where N is a dimension of the features. In the case of 3 features that in turn make 3D space, the hyperplane is a 2D plane. In addition, the hyperplane aims to maximize the distance between nearest data points from either class. Particularly, this distance which the hyperplane tries to maximize is called the margin. Moreover, Support Vectors are the data points which are the closest to the hyperplane or define it. In case any of the support vectors is moved, the hyperplane position will change. SVM has three kernels, namely linear, polynomial radial basis function and sigmoid. A kernel is a function type that transforms the input data to a higher dimensional space to be linearly separable. Generally, the algorithm aims to find the hyperplane that has the maximum margin between the end data points of each class. This algorithm is quite effective in case the data is not linearly separable. Furthermore, SVM is effective in terms of building generalized models by maximizing the margin between classes. However, for larger datasets, it is computationally expensive and slow.

4 Discussion

In the paper Python has been used in data cleaning and modeling stages. Specifically, Python built-in library sk-learn was useful by simplifying the use of various ML algorithms that otherwise were mathematically complex. The trained dataset is about movie reviews in English, and it was translated into Azerbaijani. There are 2000 reviews, of which negative and positive review records are distributed accordingly into two 1000 sub-samples. In the first phase of the work, the data was read and cleaned. Using the Pandas library, the dataset was read in python. Having read the dataset, modeling techniques have been applied to both English and Azerbaijani versions of the dataset.

Table 2: Accuracy Scores with Default Parameters

Algorithm	**TF-IDF**	**BOW**
Logistic Regression	86%	84.25%
Naïve Bayes	74%	79%
SVM	85.25%	77.5%
Decision Tree	59.75%	59.25%
Random Forest	74.5%	78.25%
AdaBoost	73%	74%
XGBoost	72%	78.75%

4.1 Azerbaijani Language Processing Approach

4.1.1 Data Cleaning

The words having less than 3 characters were dropped from the dataset. In the next step, only alpha characters were kept because digit or sign-based characters, such as punctuation, do not contribute to the level of sentiment of the texts. Since there is no built-in stop words list for Azerbaijani language in any library of Python language, the dictionary of set of stop-words in Azerbaijani was manually created for stop-words to be dropped from the dataset again due to less impact on sentiment of the texts. After the following methods were applied, all letters in the dataset were converted to lower-case so that the same word in different cases would be considered as a same word. The final step of the data preparation stage is to tokenize the sentences which is implemented by splitting them into the different lists.

4.1.2 Modeling

In order to get better results, different machine learning models, such as Logistic Regression, Naive Bayes, Decision Trees, Random Forest and Support Vector Machines have been experimented. Before training, a variety of feature extraction methods, such as TF-IDF and Countvectorizer, have been applied. Since the size of the data is small, splitting data into train and test sets might not be the right option. Thus, k-fold method was used for model building and evaluation stages. However, to guarantee the obtained results were correct, both approaches have been used to measure the effectiveness of the methods.
Fig 1. Modeling Stages

In the pipeline, during the first stage, the word tokenized text is countvectorized. The selected ngram is (1,2) and analyzer is word. The split of train and test data are 0.8 and 0.2, accordingly. In the next stage, the models with default parameter values are trained. The results indicate that Decision tree combined with Countvectorizer produced 59.7% accuracy and 61% precision, which proves this combination to be not a suitable option. Logistic Regression, on the other hand, produced 84.25% accuracy and 88% precision, which is a much better result (Table II). However, the models are prone to overfitting as the training accuracy values range between 95 and 99.

Thus, regularization is applied to minimize the possibility of overfitting. Overfitting is the phenomenon in machine learning when the model learns the training data too much, and, consequently, the model performs exceptionally well on training data, but poorly on testing data. Therefore, it is preferred to build a model that achieves the performance metrics to be almost close in both training and testing and over the predefined thresholds. After L1 regularization was applied, most models still showed either no or very little improvement in accuracy score. The only model which saw significant increase in accuracy score from L1 regularization was Naïve Bayes with BOW. The results show that Naive Bayes algorithm with BOW produced 86% accuracy, which is a 7% increase from the result with default parameters.

In another experiment, instead of train and test split, k-fold approach was used. K-fold splitting is an approach where the dataset is divided into k equally sized folds. In each iteration,

Table 3: Accuracy Scores using K-fold Approach

Algorithm	**TF-IDF**	**BOW**
Logistic Regression	81%	84.6%
Naïve Bayes	78%	79%
SVM	81%	82%
Decision Tree	64%	59.75%
Random Forest	75%	77%
AdaBoost	73%	74%
XGBoost	77%	78%

Table 4: Accuracy Scores for Sentiment Analysis for Dataset in English

Algorithm	**TF-IDF**	**BOW**
Logistic Regression	87%	78%
Naïve Bayes	85%	81%
SVM	85%	82%
Decision Tree	62%	68%
Random Forest	75%	78%
AdaBoost	75%	76%
XGBoost	77%	76%

a single fold is used as testing whilst k-1 folds are utilized for training. After being trained on different parts of the dataset, the trained models produce different accuracies. By analyzing the standard deviation of the accuracies, the average of the accuracy scores is calculated to find the models' average performance on finding the matches between test features and its labels (Table 3).

4.2 Language Processing on English Dataset

An English version of the same dataset has been used in the modeling process to predict the general sentiment of the reviews. Since Python has enriched built-in libraries for language modeling in English, particularly provided by nltk, word tokenization, stemming, and feature extraction have been implemented without building the steps from scratch. After reading the text files in Python, the data is structured using Pandas library. In the next stage, the stop-words, such as "and", "or", "is", etc., have been removed from the texts. The list of stop-words is taken from the built-in Python library, namely nltk corpus. Moreover, each text has been tokenized using the word tokenizer function. Having passed these stages, the Porter Stemming method has been applied. To be more specific, the method is used to keep the root of different versions of the same word in the dataset. As an example, computers, computation, and computed are kept as "comput" in the texts since their sentiment is same despite being in different versions. In another stage of data preprocessing, the words consisting of at least 4 characters in length have been kept, while the words which consist of 3 or less characters were removed.

In the modeling stage, various machine learning algorithms have been implemented. First, the models are trained with default parameter values. Then, to overcome issues such as overfitting, regularization has been applied to algorithms. As in the previous modeling stage, the data is divided into train and test datasets. However, k-fold approach has been also implemented to measure the model's performance as well as minimize risk of overfitting. Based on default parameter values, Logistic Regression with TF-IDF feature extraction method provides 87% accuracy. Meanwhile, SVM and Multinomial Naïve Bayes with TF-IDF provided 85% of accuracy (Table 4).

However, the Decision Tree algorithm is very likely to suffer from overfitting problem It can be proved by its training and test data accuracy scores: it achieved test accuracy of 67% and training accuracy of 100%. Bagging approach has been used by applying Random Forest algorithm, but the result was not satisfactory even though problem of overfitting was largely

solved.

5 Conclusion

In this research, various ML algorithms have been used to predict the sentiment of the movie reviews in both English and Azerbaijani. The study has indicated the achievements attained in building models with various techniques. TF-IDF and BOW (or Count Vectorizer) have been implemented for feature extraction methods from the texts and tested in different models, namely Logistic Regression, Naïve Bayes, SVM, Decision Tree, Random Forest, AdaBoost and XGBoost.

For Azerbaijani version of the dataset, using TF-IDF feature extraction approach, Logistic Regression and SVM algorithms produced better result compared to other models (81% accuracy, measured as the mean of 10 folds' results). However, decision tree perfomed poorly and produced 64% accuracy. This can partially be attributed to the model suffering overfitting. When BOW feature extraction was used, Logistic Regression produced accuracy of 84% whilst SVM reached 82% accuracy score. Based on both approaches, Decision tree is considered a poor algorithm for this specific problem since it provided only 59.75% accuracy score with BOW method.

The same dataset in English has also been modeled to compare the results with different preprocessing techniques that were not applied in Azerbaijani version, such as stemming, stopwords list, etc. Overfitting was much less of as issue for models based on English dataset. The highest score was attained with BOW feature extraction method using Logistic Regression method. Additionally, SVM and Naive Bayes algorithms performed well with TF-IDF feature extraction method and achieved 85% accuracy score.

The research indicates that the results are similiar for both language models. However, it should be noted that the amount of data is quite small for sentiment analysis. Therefore, the amount of data in the dataset should be increased so that a better comparison can be made.

6 Future work

Python has enriched libraries for building language models in English. These include as list of stopwords, word tokenization, stemming and lemmatization techniques. Particularly, English language corpus is well developed, which helps achieve higher results. However, Azerbaijani language lacks the language corpus which makes it harder to build generalized models. Also, there are not any libraries like NLTK and BERT for Azerbaijani language, which makes creation of accurate sentiment analysis models a lot harder. Therefore, a lot of resources should be dedicated to research and creation of language corpus for Azerbaijani Language, which would help getting higher accuracy scores.

Acknowledgement

I am extremely grateful to Mr. Mammadzada for his guidance and technical support throughout this research experiment. In addition, I am thankful to Dr. Rustamov for providing the essential dataset for the research.

References

Baid, P., Gupta, A., & Chaplot, N. (2017). Sentiment analysis of movie reviews using machine learning techniques. *International Journal of Computer Applications, 179*(7), 45-49.

Chandra, Y., Jana, A. (2020, March). Sentiment analysis using machine learning and deep learning. In 2020 *7th International Conference on Computing for Sustainable Global Development* (INDIACom) (pp. 1-4). IEEE.

Hasanli, H., Rustamov, S. (2019, October). Sentiment analysis of Azerbaijani twits using logistic regression, Naive Bayes and SVM. In 2019 *IEEE 13th International Conference on Application of Information and Communication Technologies* (AICT) (pp. 1-7).

Heredia, B., Prusa, J.D., & Khoshgoftaar, T.M. (2018, May). Location-based twitter sentiment analysis for predicting the US 2016 presidential election. In *The Thirty-First International Flairs Conference.*

Neethu, M.S., Rajasree, R. (2013, July). Sentiment analysis in twitter using machine learning techniques. In 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT) (pp. 1-5).

Mammadli, S., Huseynov, S., Alkaramov, H., Jafarli, U., Suleymanov, U., & Rustamov, S. (2019, September). Sentiment polarity detection in Azerbaijani social news articles. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019) (pp. 703-710). https://doi.org/10.26615/978-954-452-056-4_082

Suleymanov, U., Kalejahi, B.K., Amrahov, E., & Badirkhanli, R. (2020). Text Classification for Azerbaijani Language Using Machine Learning. *Computer Systems Science & Engineering, 35*(6).

Wenando, F.A., Hayami, R., & Novermahakim, A.Y. (2020, October). Tweet Sentiment Analysis for 2019 Indonesia Presidential Election Results using Various Classification Algorithms. In 2020 *1st International Conference on Information Technology, Advanced Mechanical and Electrical Engineering* (ICITAMEE) (pp. 279-282).