# ML992: Overview of the Machine Learning Course Project

**Department of Computer Science, University of Tabriz, Tabriz, Iran**

Hamed Babaei Giglou[1]

[1] NLP Researcher and Senior Data Scientist, M.Sc Student
[1] hamedbabaeigiglou@gmail.com

## 1 Introduction

Sentiment analysis is a new topic in research and useful in many other areas. In the modern world, large amounts of textual data are collected through surveys, comments, and reviews on the web. All data collected is used to improve the products and services offered by private and government organizations around the world. Opinion or sentiment can be expressed in one sentence or in multiple sentences as a paragraph. Opinion word orientation determines the orientation of opinion. One single sentence can one or more opinion words.

In general, sentiment analysis is a natural language processing technique used to determine whether data is positive, negative or neutral.

The goal is to determine whatever a given text is positive sentiment or negative sentiment in this task. In this overview, we examined the submitted models and proposed models by participants and lessons learned from them and the evaluation procedure.

The rest of the paper is organized as follows. Section 2 presents dataset. Section 3 describes the proposed method. Section 3 describes the performed baselines. Section 4 describes submitted software. Finally, section 5 presents our conclusions.

## 2 Dataset

The IMDb Movie Reviews dataset is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDb) labeled as positive or negative. The dataset contains an even number of positive and negative reviews. Only highly polarizing reviews are considered. A negative review has a score less than or equal to 4 out of 10, and a positive review has a score higher than or equal to 7 out of 10. No more than 30 reviews are included per movie.

More information about dataset statistics described in the table 1. The overall dataset consists of 50k reviews with 280 tokens on average at each review.

## 3 Baseline Models

In order to compare the proposed methods, we implement 2 baselines as described in below.

| Dataset | Size | Averaged Tokens per Review | # of Positive | # of Negative |
|---|---|---|---|---|
| Train Set | 40000 | 280 | 19981 | 20019 |
| Validation Set | 5000 | 278 | 2514 | 2486 |
| Test Set | 5000 | 280 | 2505 | 2495 |

**Table 1.** Dataset Stats

**NGRAMLR**: The uni-gram representation that contains all words without applying preprocessing and parameter tuning, and LogisticRegression as a classifier with C = 1.

**RANDOM**: a random prediction model predicts 1 if random value is $\in$ [0, 0.5] else 0

## 4 Submissions and Results

We analyzed participant from a different perspective, preprocessing, representations, classification, other approaches, and tools.

### 4.1 Preprocessing

Most of the participants made minor or extensive preprocessing. For example participant [9] [1] [6] [5] [4] [3] used tokenization for spliting reviews into words for more preprocessing. Also [9] [1] [6] [3] are removed stopwords. The [9] [1] [6] [5] are made an extensive preprocessing, such as removals of punctuations, special characters removal, and lowerization. The participant [9] [1] [6] used lemmatization to increase word weights and making them unique for better representations.

Among participants [2] [7] [10] didn't employ any preprocessing. However, among them, [2] got a very promising results.

### 4.2 Representations

The participants [9] [1] [6] [2] are used TF-IDF representations. In addition, the [2] used $sublinear\_tf = True$ parameter while using TF-IDF representation. Also, the [1] [7] [10] [3] used Bag-of-Word (BoW) representations in their experiments too.

The [4] used ensemble feature representations using uni-gram, bi-gram, and TF-IDF representations to enrich the representation of reviews.

### 4.3 Classifications

[1] used different models like SVM, LogisticRegression, RandomForest, Naive Bayes and SGD classifier in their experimentations and they come up good results with SVM whey they are using C=1e-5 and and RBF kernel. Another participand [9] used the same approach and concluded that SVM is the best choice for this task. But participants [6] [2] employed linear SVM and got a bether results too. Partiicpand [7] used LogisticRegression and Naive Bayes and conluded that Naive Bayes works well with BoW

representations. However, [10] relied on the baseline model (LogisticRegression). The [4] used SGD classifier since they believed that it can be fine-tuned like Neural Network models and they made dramatic increase in their results at test time and got place 1 in this project. The author [3] used RandomForest classifier since they it achived good result in their experiments but it performed worss at the end.

4 [3] BK

### 4.4 Other Approaches

Between all of the participants, participant [5] and [8] used some kind of deep learning approach. They used random embedding followed by LSTM or GRU layers to extract higher-level features and a single Fully Connected layer for classification. However, their models come up with overfitting, and they weren't able to fix the issue. Because of this reason they got a worse result in both validation and test sets. It meant that using random embedding representations may not be a good choice, and using pre-trained word embeddings may solve the issue of parameter tuning, and using regularization technique may also solve the issue.

### 4.5 Tools

Most of the participants used scikit-learn library for a classification purpose, and nltk and handcraft approach for preprocessing. Among them only [5] and [8] used TensorFlow, a deep learning library for modeling.

### 4.6 Results

The results presented in table 2. In the table:

- The $*$ represents late submissions
- The  represents people that violate the task rules by sharing Validation-GT outside of the predefined-scope
- The $SScore$ column is the submission score (from 100) based on averaged results
- The $CScore$ column is the code quality score (from 100) and extra mark from 100 points from organizer view for the quality.
- The $RScore$ represents report quality score (from 100)
- The $FinalScore$ is $(SScore + CScore + RScore)/3$ and its value is in range of (0, 100), higher than this is extra mark!

## 5 Conclusion and Analysis

In this paper, we introduced a sentiment analysis task for a machine learning course in fall 2021. With 10 participants we concluded the task. They have been evaluated based on their results, code, and report. In phase 1 [6] achieved the top-performing place, however in the second phase [4] achieved the best performing result and the end best performer in the proposed method is [4]. The participant [1] achived the top ranked score in the final based on their high-quality implementations. The [6] and [9] are the top reporter which they report were in high quality and standard.

| # | ID | ACC (validation) | ACC (test) | AVG | SScore | CScore | RScore | FinalScore |
|---|---|---|---|---|---|---|---|---|
| 1 | danandeh [4] | 0.8902 * | **0.9148** | **0.9025** | 100 | 100 + 80 | 90 | 123 |
| 2 | sasvm [2] | 0.897 | 0.8996 | 0.8983 | 100 | 100 + 10 | 90 | 100 |
| 3 | HAL[6] | **0.8908** | 0.9026 | 0.8967 | 100 | 100 + 30 | **100** | 110 |
| - | NGRAMLR(organizer) | 0.8888 | 0.8864 | 0.8876 | - | - | - | - |
| 4 | wildonion[1] | 0.8628 | 0.9058 | 0.8843 | 100 | **100 + 100** | 90 | **130** |
| 5 | 54rnd [9] | 0.8558 | 0.9018 | 0.8788 | 100 | 100 + 40 | **100** | 114 |
| 6 | yaaghobi[10] | 0.8684 * | 0.888 | 0.8782 | 100 | 100 | 0 | 67 |
| 7 | Sentiment_Analysis [7] # | 0.8902 | 0.49 | 0.6901 | 90 | 90 | 80 | 87 |
| 8 | BK [3] | 0.8524 * | 0.4938 | 0.6731 | 90 | 90 | 60 | 80 |
| - | RANDOM(organizer) | 0.5036 | 0.5046 | 0.5041 | - | - | - | - |
| 9 | garshasbi[5] # | 0.4952 * | 0.5344 | 0.5148 | 80 | 100+10 | 50 | 80 |
| 10 | textclassification [8] # | 0.4952 * | 0.5344 * | 0.5148 | 80 | 100+10 | 50 | 80 |

**Table 2.** Final Results

## References


1. ArefiMoghaddam, M.: wildonion - st999367704. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)
2. Barzegar, S.: sasvm - st99193671001. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)
3. Birang, S., Kangari, E.: BK - st999367101 and st989367401. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)
4. Danandeh, P.: danandeh - st999367102. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)
5. Garshasbi, Z.: garshasbi - st. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)
6. Isazadeh, A.S.: HAL - st9719367151. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)
7. Khosroshahi, P.D.: Sentiment_Analysis - st999367103. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)
8. Kordestani, F.M.: textclassification - st999367701. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)
9. Shams, P.: 54rnd - st999367703. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)
10. Yaaghobi, M.: yaaghobi - st999367106. In: Razmara, J., Giglou, H.B. (eds.) Sentiment Analysis, ML Course Project. ML992 (2021)