Turkish Journal of Forecasting vol. 06 no. 2 (2022) pp. 46-52

Content list available at JournalPark

Turkish Journal of Forecasting

Journal Homepage: tjforecasting.com

# Topic Modelling and Artificial Intelligence based Method Using Online Employee Assessments to Analyse Job Satisfaction

A. Ozdemir[1], A. Onan[2], V. Cinarli Ergene[3,*]

[1]*Manisa Celal Bayar University, Faculty of Arts and Sciences, Mathematics, Manisa, Türkiye*
[2]*İzmir Katip Çelebi University, Faculty of Engineering and Architecture, Department of Computer Engineering, İzmir, Türkiye*
[3]*Manisa Celal Bayar University, Institute of Natural and Applied Sciences, Mathematics, Manisa, Türkiye*

A R T I C L E  I N F O

*Article history*:

| | | | |
|---|---|---|---|
| Received | 09 | September | 2022 |
| Revision | 10 | October | 2022 |
| Accepted | 12 | October | 2022 |
| Available online | 31 | December | 2022 |

*Keywords*:

Topic modelling
Text classification
Ensemble learning
Artificial intelligence

R E S E A R C H  A R T I C L E

A B S T R A C T

In this study, the performance of the proposed sample selection method was evaluated on some basic classifiers by conducting a basic literature review on the use of topic modelling methods by considering the online evaluations of the employees in order to determine and analyse the job satisfaction factors. In addition, the effectiveness of different representation structures is evaluated in order to represent the data sets effectively and the main results are obtained regarding the use of classification ensemble methods in the field of text mining. In this work it was emphasized that machine learning methods can achieve high performance in classification and work effectively and scalable with large data sets. The dataset used in this study was obtained from www.kaggle.com. A total of 67529 comments collected from people working at Google, Amazon, Netflix, Facebook, Apple, and Microsoft were evaluated. Within the scope of this study, a text mining and artificial intelligence-based method will be developed, and a solution will be brought to text mining with artificial intelligence methods.



## 1. Introduction

Today the importance of the internet in people's access to information is very great. Many different sources such as web pages, news, blogs, social media platforms, e-commerce sites, scientific articles facilitate access to information in every field. As a result, the size of the data stored in internet resources is increasing day by day. Considering that the stored data is mostly text data, automatic analysis of text data becomes very important research problem. So that automated tools have emerged to perform the tasks of searching, understanding and processing large-scale text data. Topic modelling methods to fill this gap; machine learning has begun to be widely applied in natural language processing and information extraction processes. Topic modelling is an unsupervised machine learning method applied to access semantic information from large-scale document collections [1].

*Corresponding author.
*E-mail addresses*: vildan.cinarli@gmail.com (Vildan Çinarli Ergene), acaozdemir@gmail.com (Ali Özdemir), aytugonan@gmail.com (Aytuğ Onan)

https://doi.org/10.34110/forecasting.1173063

## 2. Topic Modelling

It is a kind of statistical model that explores abstract issues within a set of documents in machine learning and natural language processing. It is a machine learning technique, which is unsupervised learning, that can automatically cluster similar expressions with phrases that best characterize the document set [2].

Topic modelling generally gives an idea about what is the topic of a document. Words that occur together can be thought of as a label of the collection. Phrases called topics of documents in a document set are contained within the documents in a confidential and unstructured way. The characteristics of these topics are frequently seen together in the text and generally consist of words that share a common or similar theme [2].

In topic modelling abstract topics are produced by clustering the words that are frequently seen together in the text and the related texts are located in one or more clusters closest to the words they contain [2].

### 2.1. Working principle of topic modelling

Topic modelling is a machine learning method that determines the semantic structure of a document containing text. Expressions that may be close to each other are grouped in the semantic space to form an abstract subject. Then these documents are clustered according to the groups which created and the words they contain. There are many existing topic models developed by researchers in the literature. The Latent Dirichlet Allocation algorithm is the most common algorithm of these topic models. Latent Dirichlet Allocation algorithm learns the topics in the document collection, the possibilities of the words that make up the topics under the topics, which topics are assigned to the words that make up the document for the documents, and the distribution of the topics in this document for each document [3].

The usage areas of topic modelling are as follows [2]:

- Sentiment Analysis
- Bioinformatics
- Chatbots
- Spam Filtering

## 3. Latent Dirichlet Allocation Algorithm

Latent Dirichlet Allocation algorithm is one of the methods of topic modelling, which is shown as a natural language processing research area. It is a generative and probabilistic topic modelling method for extracting relevant topics from documents. The basic idea is based on that documents are a mix of topics, while topics are also a mix of words. It is the method that contributes the most to the popularity of topic modelling methods. Topic modelling method pioneered the studies and made great contributions. Today, there are many new studies on this method. Latent Dirichlet Allocation algorithm discovers hidden themes in documents. To do this, it uses a generative probability model and Dirichlet distributions. Two different of Dirichlet parameters are used. While one of the parameters affects the distribution of the topics in the documents, the other parameter is effective on the distribution of the words in the topics. The high and low Dirichlet values affect the effect of the distributions. This method is based on the Bayesian framework [4].

It does not need any pre-processing information and collects words and processes them. In this process, the placement of the words in the sentence is not important. The coexistence of words is not taken into account. The gensim library (Python) is frequently used for the Latent Dirichlet Allocation algorithm.

The Latent Dirichlet allocation algorithm used in a statistical model that allows clusters of observations to be explained by unobserved groups and explains why parts of the data are similar. The purpose of LDA in topic modelling is to find the topics that a document belongs to based on the words it contains. Topic modelling is one of the most used methods. While using LDA in topic modelling the general purpose of this algorithm is to create fixed and abstract topics. Each abstract topic represents a set of terms. Every term found in the documents is mostly captured by these abstract topics so that all documents match the topic titles. The LDA model is based on two important approaches. Each topic can be described by a term distribution and each document can be described by a topic distribution [2].

LDA is a probability-based topic modelling method. The model generates topics based on word weight from a set of documents. In the working progress of LDA, topics have a probability distribution over words and text documents have a probability distribution over topics. Each topic has a distribution over the word string.

LDA is an unsupervised learning algorithm, and it does not need predefined words. Labels are assigned to the topics according to the classes after the number of topics is determined. LDA assigns a random topic to the words for each document in the document. LDA uses this information to generate various statistics. Local statistics show how many words are assigned to topics in each document, while global statistics show how many times each word is assigned to each topic for the whole document.

After the statistical information is obtained, the topic assignment of each word is performed for each document. Thus, the existing vocabulary information should be updated as much as the number of iterations [5].

## 4. Data sources and preprocessing

The dataset used in this study was obtained from www.kaggle.com. A total of 67529 comments collected from people who works at Google, Amazon, Netflix, Facebook, Apple and Microsoft were evaluated. In order for the model to work correctly, the words were hulled and WordNetLemmatizer module offered by Natural Language Tool Kit (NLTK). Natural Language Tool Kit is a library for Python that processes natural language text.

The parameter settings were run at sampling 50, 100, 150, 200 and 250 iterations for LDA. The number of subjects represented by K was determined as 20. Dirichlet hyperparameters $\alpha$ and $\beta$ were determined as 50/K=2.5 and 0,01. Model parameters are updated by multiplying the number of texts with the total number of words. Each extracted subject was represented by the first ten words.

The resulting text representations are based on five basic classification algorithms, Naive Bayes (NB) algorithm, Support Vector Machines (SVM), k-nearest neighbour algorithm (KNN), Logistic Regression (LR) and Random Forest (RF) algorithms are evaluated. In addition, the activities of ensemble learning methods in text mining have been analysed by using 5 different algorithms, namely AdaBoost algorithm, Bagging, Random Space (RS), Voting and Stacking which are among the ensemble methods. Results are indicated in the Table 1-4 and Figure 1-4 below.

**Table 1.** Accuracy values

| Method \ Number of Subjects | N=50 | N=100 | N=150 | N=200 | N=250 |
|---|---|---|---|---|---|
| KNN | 81.24 | 80.32 | 79.36 | 82.62 | 83.05 |
| SVM | 84.32 | 83.68 | 83.40 | 83.14 | 84.87 |
| LR | 83.48 | 80.69 | 83.24 | 82.71 | 83.90 |
| NB | 84.35 | 84.50 | 83.66 | 83.83 | 85.02 |
| RF | 85.04 | 84.63 | 83.86 | 84.34 | 85.05 |
| AdaBoost (KNN) | 87.05 | 87.02 | 85.17 | 86.17 | 87.74 |
| AdaBoost (SVM) | 87.64 | 87.86 | 85.37 | 87.58 | 89.02 |
| AdaBoost (LR) | 87.64 | 87.80 | 85.26 | 86.84 | 88.76 |
| AdaBoost (NB) | 87.72 | 88.45 | 87.31 | 87.77 | 89.05 |
| AdaBoost (RF) | 87.96 | 88.52 | 87.81 | 88.22 | 90.27 |
| Bagging (KNN) | 86.10 | 85.13 | 83.91 | 85.17 | 85.72 |
| Bagging (SVM) | 86.42 | 86.38 | 84.52 | 85.63 | 86.74 |
| Bagging (LR) | 86.28 | 85.67 | 83.93 | 85.22 | 86.59 |
| Bagging (NB) | 86.68 | 86.40 | 84.88 | 85.70 | 87.21 |
| Bagging (RF) | 86.98 | 86.98 | 85.12 | 85.76 | 87.43 |
| RS (KNN) | 91.87 | 92.64 | 90.62 | 90.99 | 91.22 |
| RS (SVM) | 92.65 | 93.27 | 90.96 | 91.70 | 95.20 |
| RS (LR) | 92.45 | 92.78 | 90.64 | 91.59 | 91.38 |
| RS (NB) | 96.46 | 93.87 | 91.26 | 91.88 | 95.82 |
| RS (RF) | 97.57 | 94.72 | 96.03 | 93.76 | 97.19 |
| Voting (Minimum probability) | 88.85 | 88.61 | 88.06 | 88.28 | 90.47 |
| Voting (Maximum probability) | 89.45 | 89.70 | 88.75 | 88.43 | 90.49 |
| Voting (Majority voting) | 89.70 | 90.01 | 88.87 | 88.89 | 90.52 |
| Voting (Product of probability) | 89.93 | 91.34 | 89.45 | 89.38 | 90.91 |
| Voting (Average of probabilities) | 90.53 | 91.40 | 89.81 | 90.14 | 90.93 |
| Stacking | 91.10 | 92.10 | 89.96 | 90.67 | 91.05 |

**Table 2.** Precision values

| Method \ Number of Subjects | N=50 | N=100 | N=150 | N=200 | N=250 |
|---|---|---|---|---|---|
| KNN | 0.82 | 0.81 | 0.80 | 0.83 | 0.84 |
| SVM | 0.85 | 0.85 | 0.84 | 0.84 | 0.86 |
| LR | 0.84 | 0.82 | 0.84 | 0.84 | 0.85 |
| NB | 0.85 | 0.85 | 0.85 | 0.85 | 0.86 |
| RF | 0.86 | 0.85 | 0.85 | 0.85 | 0.86 |
| AdaBoost (KNN) | 0.88 | 0.88 | 0.86 | 0.87 | 0.89 |
| AdaBoost (SVM) | 0.89 | 0.89 | 0.86 | 0.88 | 0.90 |
| AdaBoost (LR) | 0.89 | 0.89 | 0.86 | 0.88 | 0.90 |
| AdaBoost (NB) | 0.89 | 0.89 | 0.88 | 0.89 | 0.90 |
| AdaBoost (RF) | 0.89 | 0.89 | 0.89 | 0.89 | 0.91 |
| Bagging (KNN) | 0.87 | 0.86 | 0.85 | 0.86 | 0.87 |
| Bagging (SVM) | 0.87 | 0.87 | 0.85 | 0.86 | 0.88 |
| Bagging (LR) | 0.87 | 0.87 | 0.85 | 0.86 | 0.87 |
| Bagging (NB) | 0.88 | 0.87 | 0.86 | 0.87 | 0.88 |
| Bagging (RF) | 0.88 | 0.88 | 0.86 | 0.87 | 0.88 |
| RS (KNN) | 0.93 | 0.94 | 0.92 | 0.92 | 0.92 |
| RS (SVM) | 0.94 | 0.94 | 0.92 | 0.93 | 0.96 |
| RS (LR) | 0.93 | 0.94 | 0.92 | 0.93 | 0.92 |
| RS (NB) | 0.97 | 0.95 | 0.92 | 0.93 | 0.97 |
| RS (RF) | 0.99 | 0.96 | 0.97 | 0.95 | 0.98 |
| Voting (Minimum probability) | 0.90 | 0.90 | 0.89 | 0.89 | 0.91 |
| Voting (Maximum probability) | 0.90 | 0.91 | 0.90 | 0.89 | 0.91 |
| Voting (Majority voting) | 0.91 | 0.91 | 0.90 | 0.90 | 0.91 |
| Voting (Product of probability) | 0.91 | 0.92 | 0.90 | 0.90 | 0.92 |
| Voting (Average of probabilities) | 0.91 | 0.92 | 0.91 | 0.91 | 0.92 |
| Stacking | 0.92 | 0.93 | 0.91 | 0.92 | 0.92 |

**Table 3.** Recall values

| Method \ Number of Subjects | N=50 | N=100 | N=150 | N=200 | N=250 |
|---|---|---|---|---|---|
| KNN | 0.83 | 0.82 | 0.81 | 0.84 | 0.85 |
| SVM | 0.86 | 0.85 | 0.85 | 0.85 | 0.87 |
| LR | 0.85 | 0.82 | 0.85 | 0.84 | 0.86 |
| NB | 0.86 | 0.86 | 0.85 | 0.86 | 0.87 |
| RF | 0.87 | 0.86 | 0.86 | 0.86 | 0.87 |
| AdaBoost (KNN) | 0.89 | 0.89 | 0.87 | 0.88 | 0.90 |
| AdaBoost (SVM) | 0.89 | 0.90 | 0.87 | 0.89 | 0.91 |
| AdaBoost (LR) | 0.89 | 0.90 | 0.87 | 0.89 | 0.91 |
| AdaBoost (NB) | 0.90 | 0.90 | 0.89 | 0.90 | 0.91 |
| AdaBoost (RF) | 0.90 | 0.90 | 0.90 | 0.90 | 0.92 |
| Bagging (KNN) | 0.88 | 0.87 | 0.86 | 0.87 | 0.87 |
| Bagging (SVM) | 0.88 | 0.88 | 0.86 | 0.87 | 0.89 |
| Bagging (LR) | 0.88 | 0.87 | 0.86 | 0.87 | 0.88 |
| Bagging (NB) | 0.88 | 0.88 | 0.87 | 0.87 | 0.89 |
| Bagging (RF) | 0.89 | 0.89 | 0.87 | 0.88 | 0.89 |
| RS (KNN) | 0.94 | 0.95 | 0.92 | 0.93 | 0.93 |
| RS (SVM) | 0.95 | 0.95 | 0.93 | 0.94 | 0.97 |
| RS (LR) | 0.94 | 0.95 | 0.92 | 0.93 | 0.93 |
| RS (NB) | 0.98 | 0.96 | 0.93 | 0.94 | 0.98 |
| RS (RF) | 1.00 | 0.97 | 0.98 | 0.96 | 0.99 |
| Voting (Minimum probability) | 0.91 | 0.90 | 0.90 | 0.90 | 0.92 |
| Voting (Maximum probability) | 0.91 | 0.92 | 0.91 | 0.90 | 0.92 |
| Voting (Majority voting) | 0.92 | 0.92 | 0.91 | 0.91 | 0.92 |
| Voting (Product of probability) | 0.92 | 0.93 | 0.91 | 0.91 | 0.93 |
| Voting (Average of probabilities) | 0.92 | 0.93 | 0.92 | 0.92 | 0.93 |
| Stacking | 0.93 | 0.94 | 0.92 | 0.93 | 0.93 |

**Table 4.** F-measure Values

| Method \ Number of Subjects | N=50 | N=100 | N=150 | N=200 | N=250 |
|---|---|---|---|---|---|
| KNN | 0.82 | 0.82 | 0.81 | 0.84 | 0.84 |
| SVM | 0.86 | 0.85 | 0.85 | 0.84 | 0.86 |
| LR | 0.85 | 0.82 | 0.85 | 0.84 | 0.85 |
| NB | 0.86 | 0.86 | 0.85 | 0.85 | 0.86 |
| RF | 0.86 | 0.86 | 0.85 | 0.86 | 0.86 |
| AdaBoost (KNN) | 0.88 | 0.88 | 0.86 | 0.87 | 0.89 |
| AdaBoost (SVM) | 0.89 | 0.89 | 0.87 | 0.89 | 0.90 |
| AdaBoost (LR) | 0.89 | 0.89 | 0.87 | 0.88 | 0.90 |
| AdaBoost (NB) | 0.89 | 0.90 | 0.89 | 0.89 | 0.90 |
| AdaBoost (RF) | 0.89 | 0.90 | 0.89 | 0.90 | 0.92 |
| Bagging (KNN) | 0.87 | 0.86 | 0.85 | 0.86 | 0.87 |
| Bagging (SVM) | 0.88 | 0.88 | 0.86 | 0.87 | 0.88 |
| Bagging (LR) | 0.88 | 0.87 | 0.85 | 0.87 | 0.88 |
| Bagging (NB) | 0.88 | 0.88 | 0.86 | 0.87 | 0.89 |
| Bagging (RF) | 0.88 | 0.88 | 0.86 | 0.87 | 0.89 |
| RS (KNN) | 0.93 | 0.94 | 0.92 | 0.92 | 0.93 |
| RS (SVM) | 0.94 | 0.95 | 0.92 | 0.93 | 0.97 |
| RS (LR) | 0.94 | 0.94 | 0.92 | 0.93 | 0.93 |
| RS (NB) | 0.98 | 0.95 | 0.93 | 0.93 | 0.97 |
| RS (RF) | 0.99 | 0.96 | 0.97 | 0.95 | 0.99 |
| Voting (Minimum probability) | 0.90 | 0.90 | 0.89 | 0.90 | 0.92 |
| Voting (Maximum probability) | 0.91 | 0.91 | 0.90 | 0.90 | 0.92 |
| Voting (Majority voting) | 0.91 | 0.91 | 0.90 | 0.90 | 0.92 |
| Voting (Product of probability) | 0.91 | 0.93 | 0.91 | 0.91 | 0.92 |
| Voting (Average of probabilities) | 0.92 | 0.93 | 0.91 | 0.92 | 0.92 |
| Stacking | 0.92 | 0.94 | 0.91 | 0.92 | 0.92 |

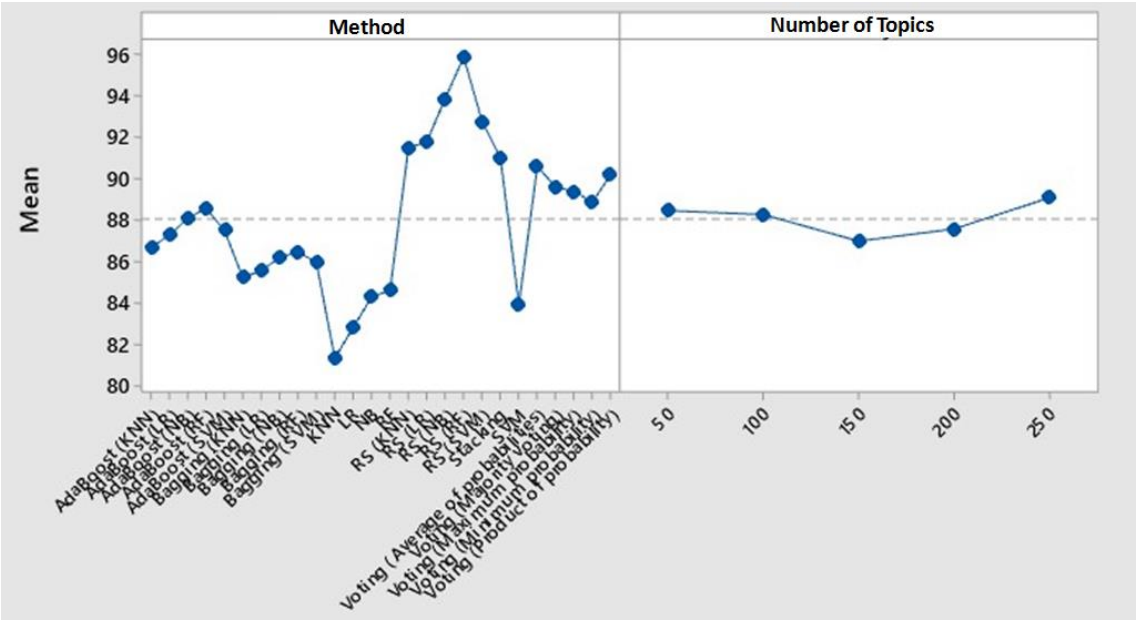


**Figure 1.** Main effects plot for accuracy

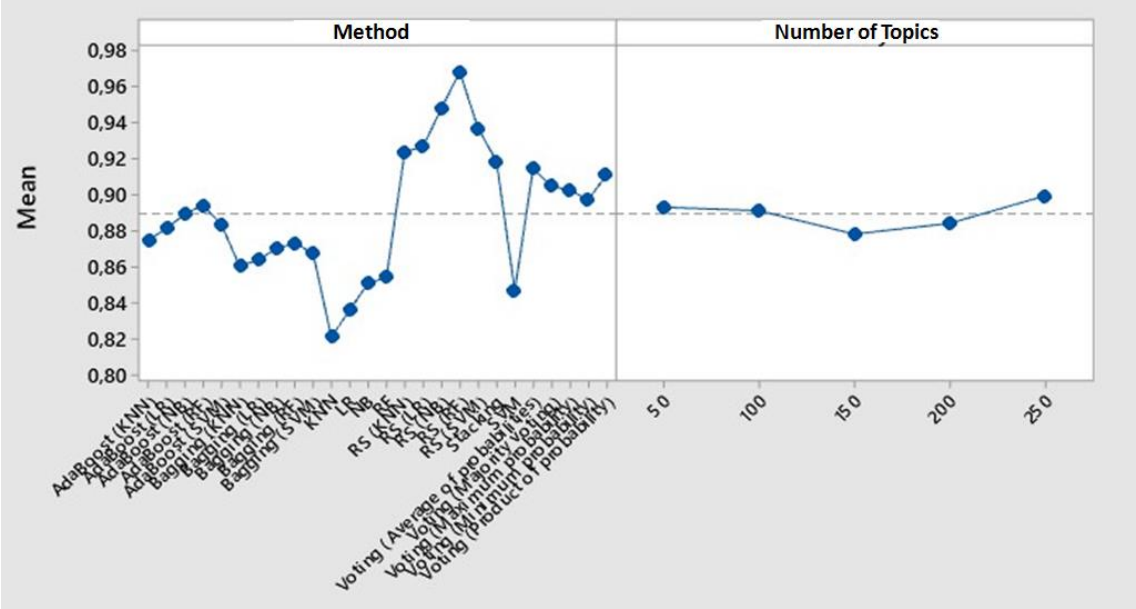


**Figure 2.** Main effects plot for precision

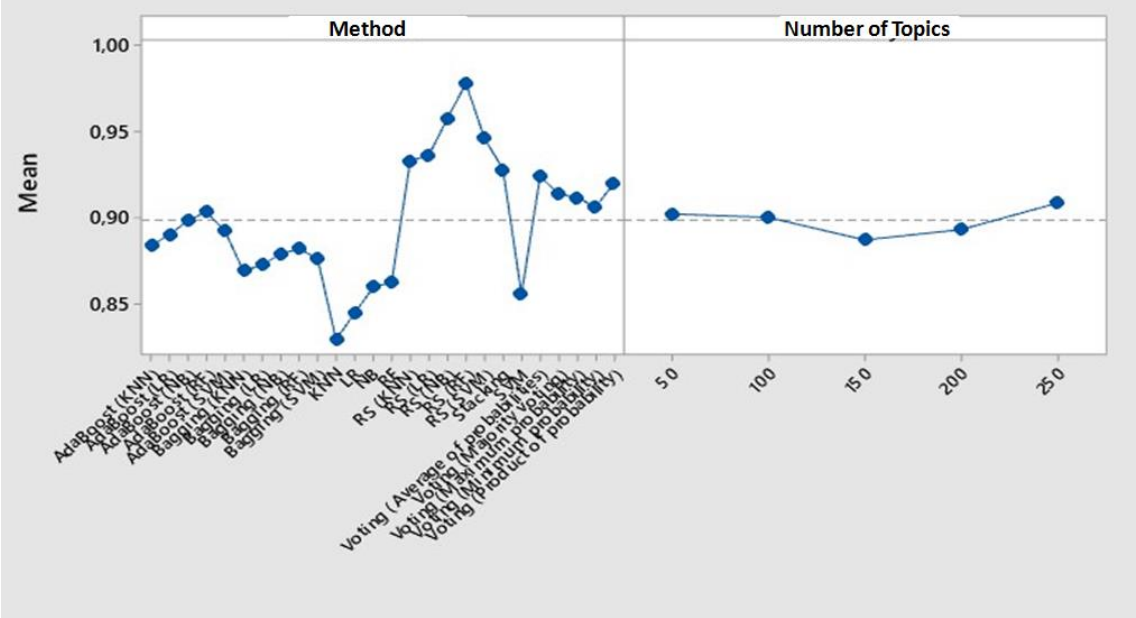


**Figure 3.** Main effects plot for recall

## 5. Result

Random Forest algorithm has got the highest performance among the basic classifiers used in experimental analysis with in the 85.05. The highest performance among the ensemble learning architectures is Random Space algorithm obtained by using Random Forest.

When evaluating the results obtained, the correct classification rate, F-measure, sensitivity and recall criteria were taken into account comparatively. When the results were compared and the performance values were analysed, N=100 iterations were chosen, and the highest success rate was obtained with 94.72% by using Random Space algorithm with Random Forest. The sensitivity value obtained from this experiment was chosen with a success value of 0.99 and the number of iterations N=50; the recall value was found to be 1.00 by selecting N=50 and the F-criterion as 0.99 by selecting N=50 and N=250 iterations. Experimental results show that text mining and machine learning methods are suitable for use in online employee evaluations.

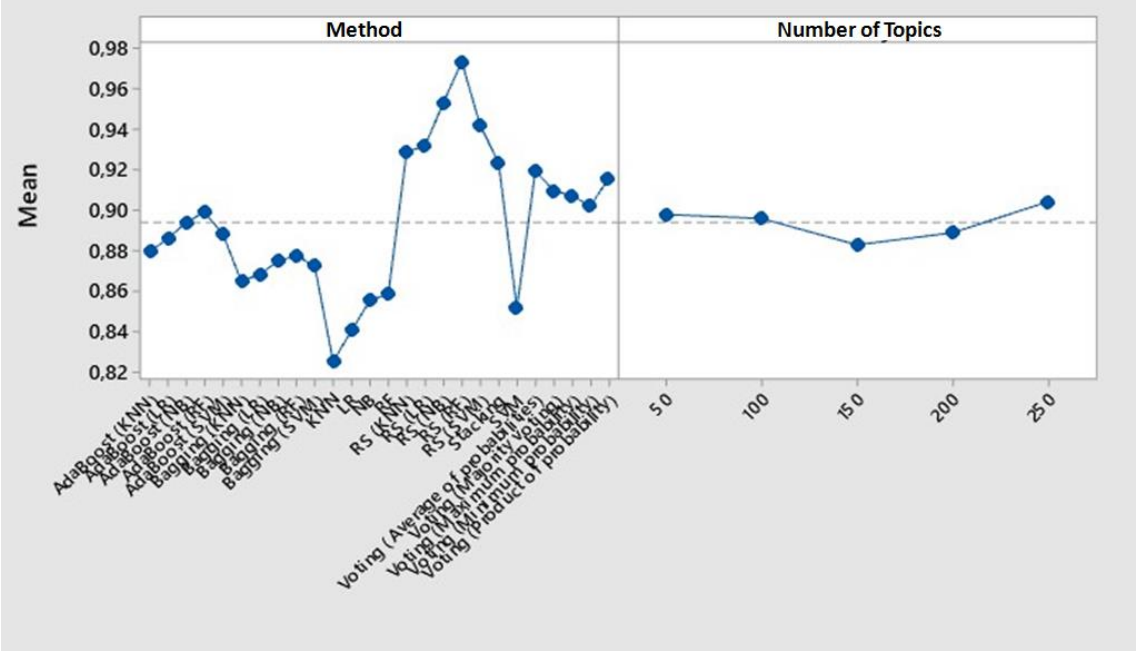


**Figure 4.** Main effects plot for F-measure

## References

[1] Tıp Veri Kümesi için Gizli Dirichlet Ayrımı Latent Dirichlet Allocation for Medical Dataset Ekin Ekinci , Sevinç İlhan Omurca , Elif Kırık , Şeymanur Taşçı 1 DEU FMD 22(64), 67-80, 2020 68.

[2] https://medium.com/@yildizhangocmen/nlp-konu-modelleme-topic-modelling-2852f28bceca

[3] Agrawal, A., Fu, W., Menzies, T. 2018. What is wrong with topic modelling? And how tofix it using search based software engineering, Information and Software Technology, Cilt. 98, s. 74-88. DOI: 10.1016/j.infsof.2018.02.005.

[4] Blei, D. M., Ng, A. Y. 2003. Latent dirichlet allocation. the Journal of machine Learning research , 3, 993-1022.

[5] https://medium.com/@anilguven1055/latent-dirichlet-allocation-lda-algoritmas%C4%B1-13154d246e05.