NSL-BP: A Meta Classifier Model Based Prediction of Amazon Product Reviews

Pravin Kumar[1] , Mohit Dayal[2], Manju Khari[3]*, Giuseppe Fenza[4], Mariacristina Gallo[4]

[1] Indian Institute of Technology (ISM), Dhanbad (India)
[2] Ambedkar Institute of Advanced Communication Technology & Research (India)
[3] Netaji Subhas University of Technology, East Campus, Delhi (India)
[4] University of Salerno, Fisciano (SA), Italy

Received 11 April 2020 | Accepted 7 September 2020 | Published 6 October 2020

unir
LA UNIVERSIDAD
EN INTERNET

Abstract

In machine learning, the product rating prediction based on the semantic analysis of the consumers' reviews is a relevant topic. Amazon is one of the most popular online retailers, with millions of customers purchasing and reviewing products. In the literature, many research projects work on the rating prediction of a given review. In this research project, we introduce a novel approach to enhance the accuracy of rating prediction by machine learning methods by processing the reviewed text. We trained our model by using many methods, so we propose a combined model to predict the ratings of products corresponding to a given review content. First, using k-means and LDA, we cluster the products and topics so that it will be easy to predict the ratings having the same kind of products and reviews together. We trained low, neutral, and high models based on clusters and topics of products. Then, by adopting a stacking ensemble model, we combine Naïve Bayes, Logistic Regression, and SVM to predict the ratings. We will combine these models into a two-level stack. We called this newly introduced model, NSL model, and compared the prediction performance with other methods at state of the art.

Keywords

Combined Model, Logistic Regression, Machine Learning, Naïve Bayes, Stacking Model, SVM.

DOI: 10.9781/ijimai.2020.10.001

I. Introduction

Nowadays, a large amount of customer reviews, available on every commercial site, provides valuable information about products but also impact the purchase decision of customers.

A recent survey of Ye, Q., Law [1] revealed that about 67.77% of customers are impacted by online reviews when they were making purchase decisions. However, rating prediction is also necessary because searching and comparing text reviews can be a headache for customers [2]. So users' reviews information should be merged, but a large number of reviews and unstructured text formats confuse users, making hard any decision. The star-rating, i.e., a star from 1 to 5 on any commercial site, can give a brief idea of product quality, more quickly than its text content. There are some interesting models that can predict user ratings from the text review [3]. Nevertheless, the rating prediction using reviews' text requires to face several challenges like human errors, vocabulary errors, and so on. The reviews may contain unreliable information increasing the quality of the task results. To get rid of these problems, we can rely on supervised machine learning techniques [4], such as text classification, which allows us to automatically classifying a document into a fixed set of classes according to its meaning. In this context, three different approaches for rating prediction could be applied: binary classification, multi-class classification, and logistic regression. The binary classification classifies a product as good or not, but using multiclass-classification and logistic regression, the customers are also informed about the level of quality of the product by giving a rating (for example, from 1 to 5).

This work proposes a new model (NSL) inspired by ensemble methods, which combine multiple existing models in order to obtain a better prediction result. In particular, adopted classifiers are Naive Bayes, Support Vector Machine (SVM), and Logistic Regression. The combination is made through a two-level stacking. All models have been trained by means of an Amazon dataset. In this sense, the approach also tries to face subsequent challenges:

1. The class imbalance: the dataset is relatively skewed in terms of class distribution.
2. In multi-class case, over-representation of 5-star ratings.

We overcome these issues by applying sampling techniques [6] to even out the class distribution. We dealt with the issue of class imbalance by investing in some balancing techniques [7].

Results show that the two most successful classifiers are Logistic regression and SVM. Still, Logistic regression gives better results than SVM. However, our combined model (the NSL model) gives the best results.

Machine learning algorithms are divided into supervised and unsupervised approaches. The first ones need the labeled data; the latter can be adopted with unlabeled data. Among supervised approaches, we will discuss about the text classification, which has been used in predicting the ratings. In particular, in Fig. 1, we describe the adopted process during text classification: from training data consisting of text documents we extract representing feature vectors

* Corresponding author.

E-mail address: manjukhari@yahoo.co.in

adopted during the training. Labeled training data helps the algorithm in discovering patterns between the input text and the respective rating. Finally, after the same preprocessing aiming to extract feature vectors, the constructed model is adopted on new data (i.e., test set) in order to discover new ratings.

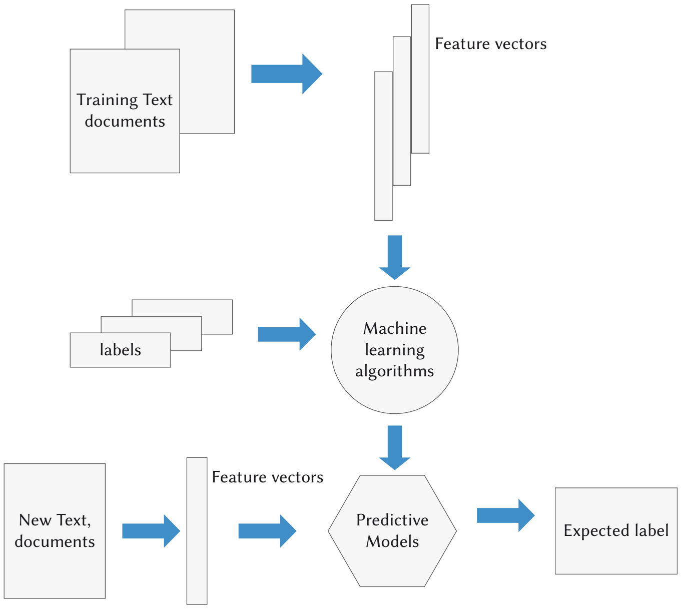


Fig. 1. Supervised Machine Learning.

II. Theoretical Background

A classification task consists in the identification of a predefined class after the learning of a model through training data. The classification could be applied to different types of data. In the context of product ratings, we face a problem of text classification. Formally, the text classification tries to predict the best class c ε C for each document d ε D, where C is a fixed set of classes, and D is a collection of documents. Two types of classifications exist. Basing on the cardinality of C, we can distinguish between binary and multi-class classification (see Fig. 2 and Fig. 3). In particular, when there are only two classes, and each document belongs to one of the two classes, we face with binary classification (e.g. spam filtering in mails). In multi-class classification, there are more than two classes, and each document belongs to one of these classes. This is the case of rating reviews: classes go from 1-star to 5-star, where 1-star is considered the worst review class, and 5-star means the best review class.

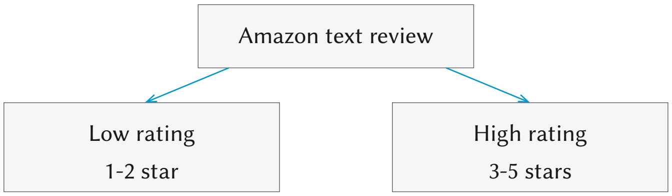


Fig. 2. Binary Classification.

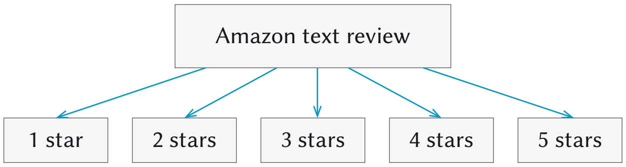


Fig. 3. Multi-class Classification.

Sometimes, due to imperfection in datasets, other important steps (i.e., data cleaning, and resampling) must precede the model training process, as expressed in Fig. 4 and detailed as follows.

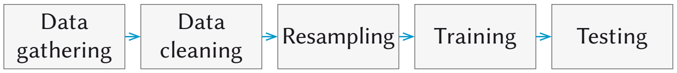


Fig 4. Text Classification Implementation.

Collected data should be cleaned in order to improve its quality: identify incomplete, incorrect, inaccurate and irrelevant parts of the data and then replacing, modifying, or deleting them.

The adopted dataset can be either balanced (Fig. 5 a) or imbalanced (Fig. 5 b). When classes are unequally distributed, the dataset is considered imbalanced.

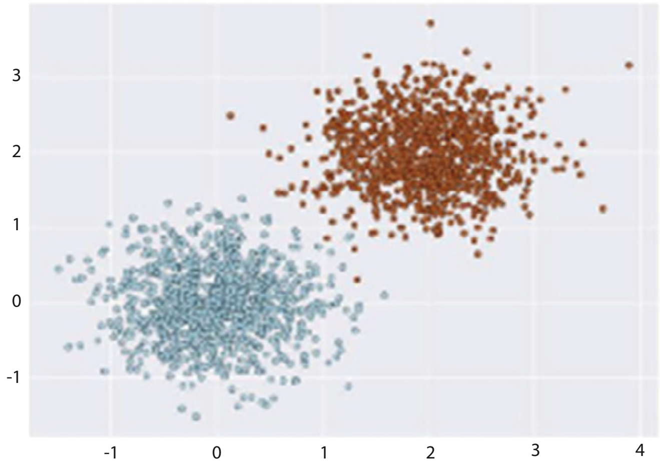


Fig. 5.a. Balanced Dataset

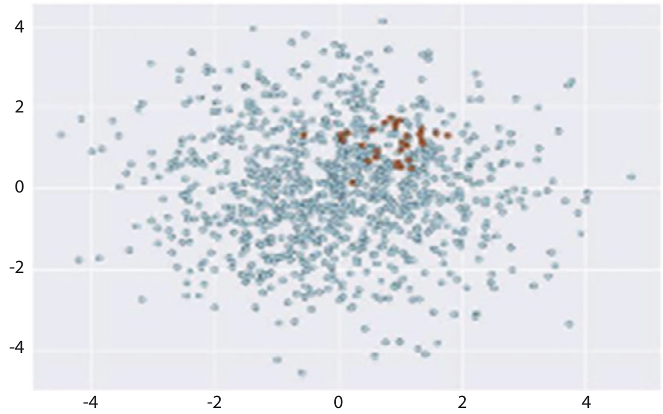


Fig. 5.b. Imbalanced Dataset.

To avoid imbalanced classification problems, various resampling [8], [9] methods can be applied. They aim at balancing data before its adoption. Imbalanced data can be treated by *under sampling* (i.e., reduction of items belonging to the most represented classes) or *oversampling* (i.e., addition of items for under-represented classes) processes.

A. Text Classification Algorithms

Text classification can be made by classification algorithms (i.e., classifiers [23]). Here we present some of the most used classifiers.

1. Naïve Bayes

Naïve Bayes classifiers belong to the family of probabilistic classifiers that apply the Bayes's theorem [32], [33]. Specifically, the classifier calculates the probability by which the document belongs to a particular class. It is based on the MAXIMUM a Posteriori (MAP) estimator [24] that by means of the class prior probability assigns the best class to the document. The mathematical formula of the probability to predict a class c to a document d is defined in Eq. 1.

$$C_{map} = \arg\max_{c \in C} \hat{P}(c) \prod_{1<=k<=n_d} \hat{P}\left({}^{t_k}/_{c}\right) \tag{1}$$

Where, $\hat{P}(c)$ is the class prior probability, the probability that a document belongs to class c, $\hat{P}\left({}^{t_k}/_{c}\right)$ is the probability of a term t at position k in a document d from the class c, and n_d is the number of terms in document d.

2. Support Vector Machine

Support Vector Machines (SVMs) are a class of supervised machine learning algorithms for binary classification problems. The key idea of SVM is to find the hyperplane ∏ that separates the positive points from negative ones as wide as possible. Here W is normal to the plane. W_1 is normal to the plane Π_1 and W_2 is normal to the plane Π_2.

In Fig. 7, we have to reduce the margin of given hyperplanes so that we can be able to find the best hyperplane, which divides the data points accurately. Let us define the distance between the data point and the hyperplane as expressed in the following Equation.

$$y_i(w^T x_i + b)$$

If the distance is *less* than 1, the point is correctly classified; if the distance is *equal* to 1, the point is on the hyperplane; if the distance is greater than 1, the point is misclassified. On the basis of the distance, we can find out the best hyperplane to classify the data i.e depicted in Fig. 6, Fig. 7.

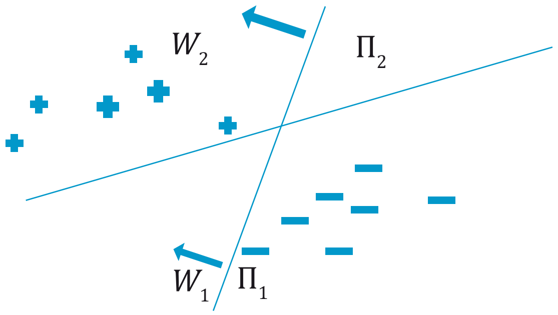


Fig. 6. Hyperplanes to divide the data.

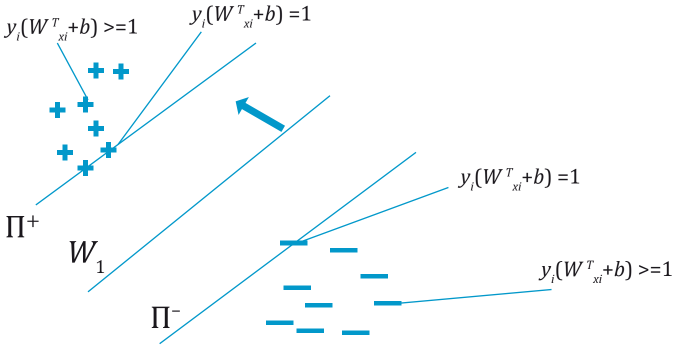


Fig. 7. Margin Hyperplane.

3. Logistic Regression

Logistic regression finds the plane that separates the different classes of data. There could be three interpretations of logistic regression, such as geometry, probability, and loss function, where all the optimization methods are related to each other, with some differences. In logistic regression, we are assuming that classes are linearly separable or almost linearly separable. If classes are not linearly separable, then we have to apply Feature Engineering on data. Feature Engineering consists of mathematical, trigonometry, or logarithmic functions [25].

4. Ensemble Methods

In addition to the described algorithms, there exist approaches that combine multiple base models in order to improve their overall performances. The combination of models can be realized through different aggregation criteria (i.e., bagging, boosting, stacking, and so on).

Our **proposed model is inspired to stacking (or stacked) approach**. It consists in a sequential method where all algorithms to combine are trained by the training set. Then, the new algorithm (the combined one, also considered as meta-classifier) is trained through the prediction outs of the other algorithms.

III. Related Literature

In the area of customers' review classification, numerous solutions are available in the literature.

S. Wararat [10] classifies hotels' customer reviews written as open comments as positive or negative, using a binary classifier (i.e., opinion mining). This model, by adopting the Naïve Bayes technique, gives a prediction accuracy result of 94.37%. Lei et al. calculate each user's sentiment on products and take interpersonal sentimental influence and product reputation into consideration [11]. To make a correct rating prediction, authors fuse three factors into the recommender system [5]. Performance evaluation of the three sentimental factors is conducted on real-world data collected from Yelp. Baccouche et al. proposed a review data pre-processing and subsequent training of different classifiers (i.e., Multinomial Naïve Bayes, Bigram Multinomial Naïve Bayes, Trigram Multinomial Naïve Bayes, Bigram-Trigram Multinomial Naïve Bayes, Random Forest) [12]. In terms of accuracy, the Random Forest approach is the best one. Reddy et al. propose combined collaborative filtering of hierarchical topic models for integrating sentiment analysis [13]. By taking previous reviews, they predict future reviews of a given author. Kawamae introduces a simple supervised learning algorithm for semantic analysis for large text documents [14]. By using pointwise mutual information, the method involves issuing queries to a Web search engine.

Turney applied supervised machine learning classification algorithms to extract the semantic orientation of individual words extracted from a big corpus [15],[34].

To address the sentiment analysis for rating prediction, Kotsiantis et al. proposed graph-based semi-supervised learning algorithms [16]. The task is to give numerical ratings for unlabeled documents based on the perceived sentiment expressed by their text. In this paper, Goldberg et al. combine the LDA model and the association rules to extract the product features and corresponding words of reviews [17]. The authors used cross-validation to prune the extracted result. In this paper, to calculate how much important the word is for review, authors adopted an unsupervised approach and ranked the reviews.

Liu et al. propose a solution showing the meaning of phrases and sentences in vector space [18]. This approach is based on a vector construction through additive and multiplicative functions. Results show that multiplicative models are better than additive alternatives. Mitchell et al. use a vector space framework to represent sentences [19]. Tiroshi et al. propose a graph-based representation of the data in order to generate and self-populate features [20].

IV. Proposed Methods

In terms of review rating prediction, we propose the NSL model inspired to ensemble method solutions that combine multiple classification models. In particular, we combine Naïve-Bayes, Logistic Regression, and SVM in a stacked way. The proposed model is shown in Fig. 8. It works as follows. Let X be the training data having n features. All three existing models are trained on X. The predictive output of each model is converted to a second level data, making each prediction a new feature for this second level. Then, we apply a meta-classifier training on this data. The meta-classifier result will be almost similar to the best of the three models.

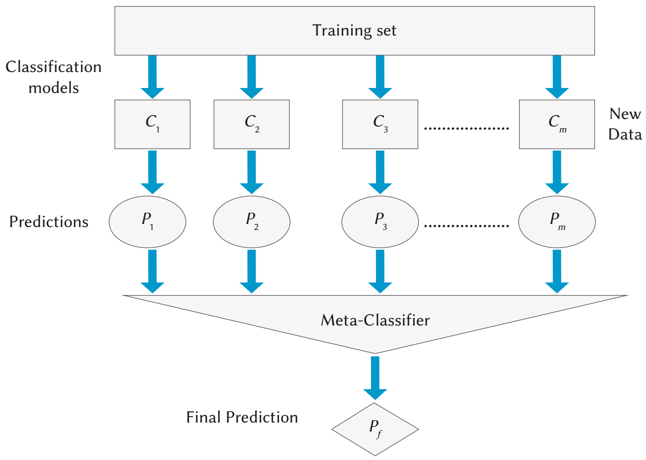


Fig. 8. NSL Model (Combined Model of Naive Bayes, Logistic Regression, SVM).

Inspired to Naïve-Bayes classifiers, we adopt the technique that predicts the best class for a document based on the probability that the terms in the document belong to the class. We took MAXIMUM a Posteriori (MAP) estimator described in Section 2.2.1.

From the Support Vector Machine, we took the minimization of the margin of hyperplane function as:

$$\text{Probabilistic } W^*, b^* = \arg\min_{w,b} \frac{||W||}{2} + c.\frac{1}{n}\sum_{i=1}^{n}\xi_i \tag{2}$$

$$m \arg in\ d = \frac{2}{||W||} \tag{3}$$

Where we have to maximize the margin and to draw the hyperplane that best divides the different data points of reviews.

$$W^*, b^* = \arg\min_{w,b} \frac{2}{||W||} \tag{4}$$

Such that $y_i (w^T x_i^{+b}) >= 1$ for all x_i. It is the hinge loss of SVM. Such that $y_i (w^T x_i^{+b}) > 1 - \xi$, where $\xi >= 0$

From the Logistic regression, we took the geometrical part; in this function, we minimize the distance of every point from the hyperplane and search for the hyperplane, which gives the best results:

$$W * = \arg\min_{\mathrm{W}} \sum_{i=1}^{n} -y_i \log p_i - (1 - y_i) \log(1 - p_i) \tag{5}$$

Where, y_i is +1 or 0: positive and negative points, respectively. Since $p_i = \sigma(w^T x_i)$ is a sigmoidal function, we have to minimize the probabilistic distance function. Then, the result of the meta-classifier will be the final prediction of that data point on the majority voting basis.

In order to preserve the figures' integrity across multiple computer platforms, we accept files in the following formats: .EPS/.PDF/.PS/.AI. All fonts must be embedded or text converted to outlines in order to achieve the best-quality results.

A. Text Preprocessing

Data Cleaning and Data Resampling are two important methods of Text Preprocessing.

The objective of data cleaning consists of: (i) punctuation discarding, (ii) number discarding, (iii) lower-casing of the text (iv) extra whitespace removing, and (v) stop word (like "is", "are", "a", "and", etc.) removing. It simplifies data and makes classification more accurate.

It is observed that the review data has many duplicate entries, so remove duplicates can unbias results in data analysis. Among available methods to remove duplicates, our choice consists of removing multiple reviews of the same user at the same time.

Before starting subsequent steps, we plotted data to recognize resampling needs and adopted suitable solutions in terms of data balancing.

In terms of representation of data, our solution treats the training corpus as a Bag of Words [21] and turned it in numerical feature vectors [22] using the *CountVectorizer* method, described following. So, given text reviews, the objective consists of extracting vectors of d dimension and finding a plane that represents them. Let be:

i) r_1, r_2, r_3......... r_n, the reviews,

ii) v_1, v_2, v_3......... v_i, the vectors of reviews in d dimension space.

iii) If Similarity (r_1, r_2)>Similarity (r_1, r_3) then distance (v_1, v_2) < distance (v_1, v_3).

iv) If r_1 and r_2 are more similar then v_1, v_2 are more closed.

Regarding the value for each feature (i.e., a word), since using only its occurrence could be poor, the TF-IDF [29] Transformer method has been used in order to obtain its TF-IDF value. Let be:

$TF\ (W_i, r_j)$ = Number of times w_i occur in r_j/total number of words in r_j.

$$IDF\ (w_i, D_C) = \log({}^{N}/{}_{n_i})$$

where N is the total number of documents, and n_i the number of documents which contain w_i.

$$n_i \ <= \ N => {}^{N}/{}_{n_i} \ >= 1 \Rightarrow \log({}^{N}/{}_{n_i}) >= 0$$

If w_i is much frequent in the corpus, then the IDF will be very low. If w_i is a rare word, then IDF will be high.

B. Training and Classification

The classifier, during the training phase, learns the mapping between a document and a class. Subsequently, it is able to classify new documents accurately.

A Latent Dirichlet Allocation (LDA) [27] task and a k-means clustering [28] step precede the training process. The LDA divides all reviews based on topic discussed within the text. We span amazon reviews from 1-star to 5-star and we bucket reviews by following criteria:

- Low: 1-2 star (if low > 80%, then 1 star, otherwise 2 stars)
- Neutral: 3 star (if neutral ≅ 50)
- High: 4-5 star (if high ≥ 80%, then 5 star, if high < 80% and high > 60%, then 4 stars)

K-means, based on the TFIDF matrix, groups documents into N clusters. Within each cluster, we count top occurring terms. By using LDA and TFIDF matrix, we attempt to extract N topics from our collection of documents. K-means forces each review to belong to only one cluster, but LDA allows a review to have many topics associated with it.

The algorithm for training of the model and the subsequent classification of new data is resumed in Algorithm 1.

```
Algorithm 1. Construction and Adoption of Classification Model
Inputs Training data D={$x_i$, $y_i$} $i$=1 to m {$x_i^{\varepsilon} R^n$, $y_i \varepsilon \gamma$}
Output An ensemble classifier H
Step1:Training of data
Step2: Using TF-IDF matrix do K-MEANS clustering on the review documents with 9 clusters
Step3: Using TF matrix use LDA for Extract top topics from clusters
Step4: learn first level classifiers
For t ← 1 to T do
    Learn a base classifier h based on D
End for
Step5: construct new data set from D
For $i$←1 to m do
    Construct a new data set that contains
    {$x_i$, $y_i$}, where $x_i$ = {$h_1$ ($x_1$), $h_2$ ($x_2$), ............. $h_i$ ($x_i$)}
End for
Step6: Learn a second level classifier
Learn a new meta-classifier h' based on newly constructed data set with functions
Training of meta-classifier to classify the data
    Step 7: From the Naïve-Bayes  MAXIMUM a Posteriori Function is used
        $Cmap = \arg\max_{c \epsilon C} \hat{P}(c) \prod_{l<=k<=n_d} \hat{P}\left({}^{tk}/_{c}\right)$
    Step 8: From the support vector machine Minimization of margin of hyperplane is used
    Function for Minimization of margin of hyperplane
        $w^*, b^* = \arg\min_{w,b} \frac{\|W\|}{2} + c.\frac{1}{n} \sum_{i=1}^{n} \xi_i$
    Step 9: From the Logistic regression probabilistic part, we minimize the distance of every point from the hyperplane, and search for the hyperplane, which gives the best results. Function for Minimization of distance from hyperplane
        $w^* = \arg\min_{W} \sum_{i=1}^{n} - y_i \log p_i - (1 - y_i) \log(1 - p_i)$
Step 10: predict the class based on the majority of voting
    Return
```

V. Experiment and Result Analysis

The NSL model, generated as described in the previous sections, is evaluated through experimentation on a real dataset as expressed following.

A. Dataset

Experimental analysis has been done on the freely available Amazon dataset "Home and Kitchen"[1]. The dataset contains 500k kitchen product reviews from May 1996 to July 2014. Each review contains product id, user id, profile name, helpfulness rating (e.g., 2/3), time, summary, text.

B. Evaluation

Regarding the validation of the model, we apply the k-fold method. In k-fold validation [26], we divide the training data into many subsets. By dividing our training data into N sets, we hold the Nth set for validation. We have three models, and, as already defined, we obtain the prediction from each one. Obtained predictions are collected in the out-of-sample prediction matrix. This matrix is used as a second level training data to obtain the final prediction. Second level training will select the best of the first level prediction models. By using out-of-sample prediction, we still have large data to train the second-level model. In the meta-classifier, we use various loss functions based on each adopted model, as explained in Section 4.2.

C. Experimental Results

After the creation of the TFIDF matrix, we apply K-means clustering to extract the clusters, as expressed in Fig. 9. The distribution of the topics extracted through the LDA application is shown in Fig. 10. Table I lists the first 10 topics and 15 words per topic associated with them.

The resulting TF-IDF matrix has thousands of attributes, so it is very challenging to show them graphically since we can plot up to 3-dimensional only. We use the Latent Semantic Analysis [30] based on the singular value decomposition [32] to reduce the dimensionality of the matrix. We then use T-SNE [31] to represent our data as best as possible in 2-dimensions.

In Fig. 11, we extract the top 24 words in the "Low" category, 12 of them with red color are not associated, while 12 with green color are associated (i.e., terms more correlated with this type of category). Fig. 12 and Fig. 13 do the same for "Neutral" and "High" categories, respectively.

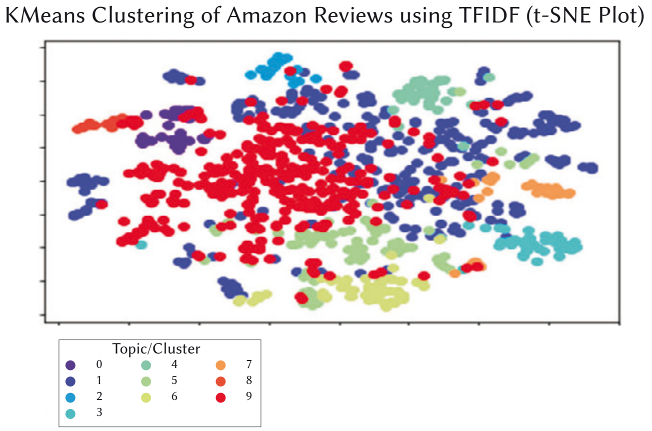


Fig. 9. K-means clustering of Amazon Reviews using TFIDF.

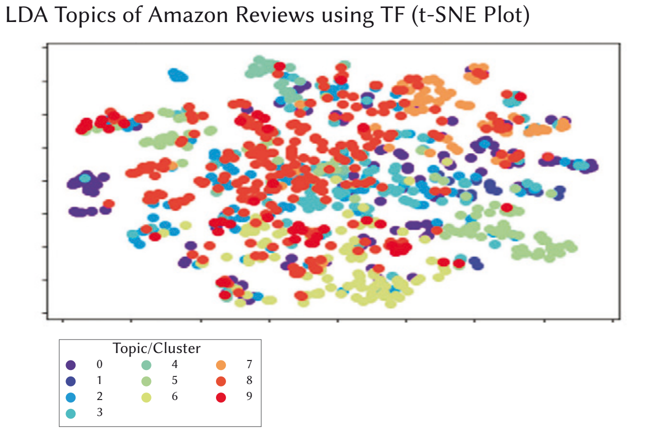


Fig. 10. LDA Topics of Amazon using TF.

[1] http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews

TABLE I. Extracted Topics

Topic	Associated words
Topic #0	use, time, make, machin, work, pot, clean, just, tri, like, unit, onli, juic, turn, blender
Topic #1	unit, fan, air, set, room, high, heat, assembl, low, veri, nois, heater, loud, turn, control
Topic #2	just, work, use, bag, thing, review, like, time, realli, don't, i'm, veri, say, product, read
Topic #3	product, year, replac, amazon, purchas, use, return, new, buy, time, review, just, month, work, did
Topic #4	cut, blade, knife, grinder, grind, use, knive, sharp, edg, steel,handl, slice, veri, good, hand
Topic #5	vacuum, clean, use, floor, bed, like, veri, carpet, mattress, brush, doe, power, attach, cord, cleaner
Topic #6	coffee, cup, water, filter, use, make, glass, hot, tea, brew, maker, drink, like, pour, mug
Topic #7	pan, cook, use, oven, stick, heat, egg, bread, toaster, oil, food, clean, toast, grill, set
Topic #8	use, like, look, veri, just, nice, good, wash, realli, don't, fit, size, hold, make, color
Topic #9	lid, water, use, pillow, open, plastic, size, small, like, fit, bowl, contain, kettl, jar, food

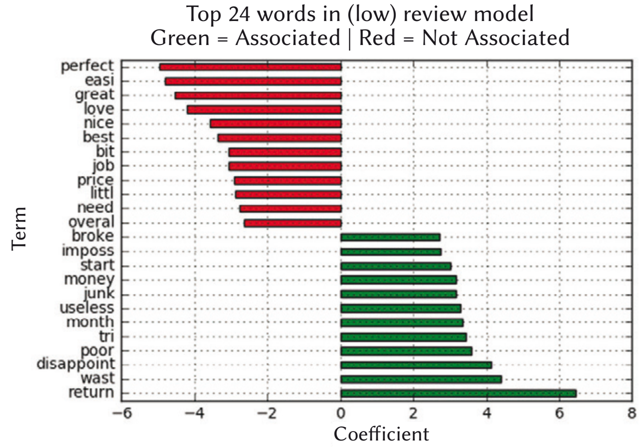


Fig. 11. Top 24 words in (low) review model.

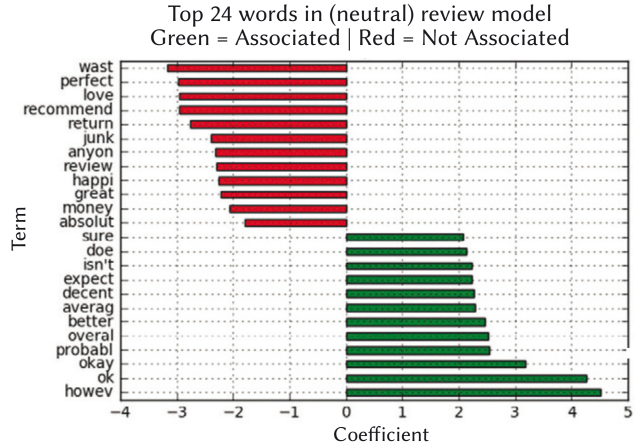


Fig. 12. Top 24 words in (neutral) review model.

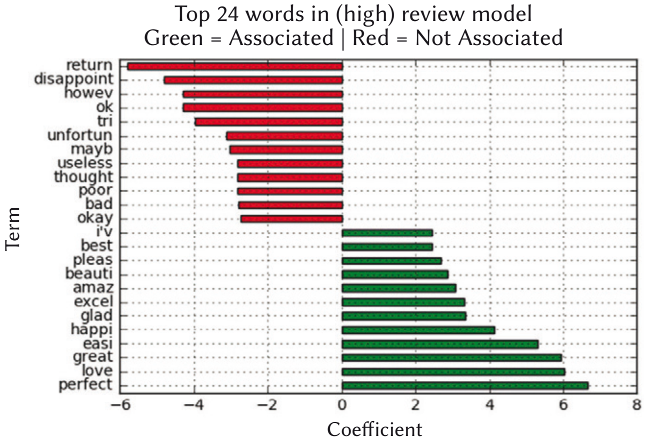


Fig. 13. Top 24 words in (high) review model.

Applying the learning model in new data rows (i.e., data does not used during the training), four categories of results can be encountered:

1. **True Positive (TP)**: prediction accurately predicted as positive
2. **True Negative (TN)**: prediction accurately predicted as negative
3. **False Positive (FP)**: prediction incorrectly predicted as positive
4. **False Negative (FN)**: prediction incorrectly predicted as negative

The prediction Accuracy of the classifier can be calculated as

$$\text{Error Rate-}= \frac{FP+FN}{TP+TN+FN+FP}$$

$$\text{Accuracy} = \text{1-Error Rate}=\frac{TP+TN}{TP+TN+FN+FP}$$

The error rate simply calculates the ratio between the number of wrong predictions made by our classifier and total number of test cases. From Table II to Table V, accuracy is reported for every adopted classifier. Our NSL model gives the highest accuracy with respect to other models. From Fig. 14, Fig. 15, Fig. 16 and Fig. 17, we compare accuracy of all the classifiers with our NSL model.

TABLE II. Accuracy Prediction of LR Model

LR Model	Accuracy
Low rating	75.8 %
Neutral rating	68.6%
High rating	77.8%

TABLE III. Accuracy Prediction of SVM Model

SVM Model	Accuracy
Low rating	74.6 %
Neutral rating	65.8%
High rating	77.4%

TABLE IV. Accuracy Prediction of NB Model

NB Model	Accuracy
Low rating	70.1 %
Neutral rating	66.8%
High rating	69.1%

TABLE V. Accuracy Prediction of NSL (Combined Model)

NSL Model	Accuracy
Low rating	76.2 %
Neutral rating	69.2%
High rating	78.5%

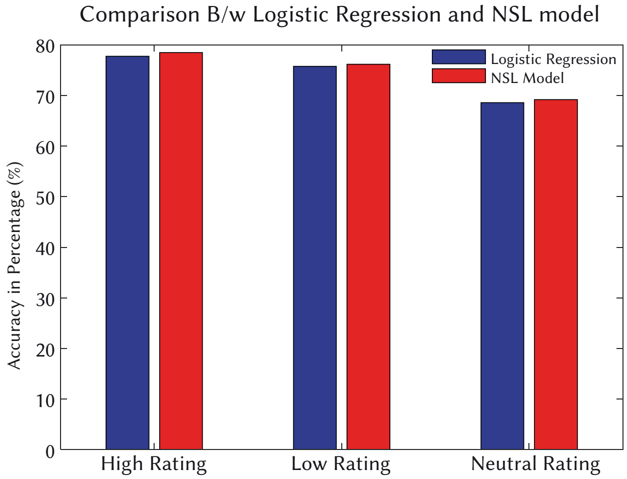


Fig. 14. Combined NSL model and Logistic Regression.

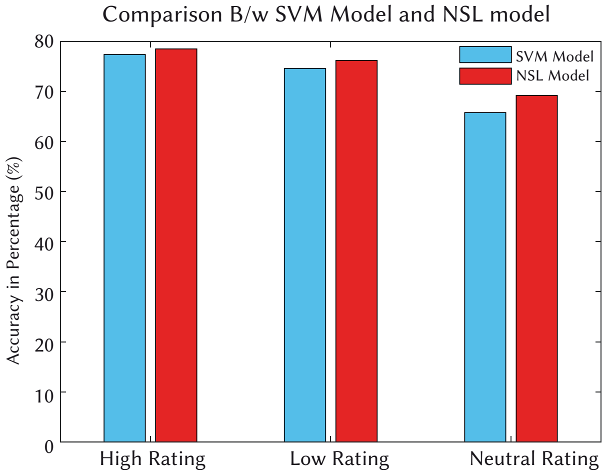


Fig. 15. Combined NSL Model and SVM.

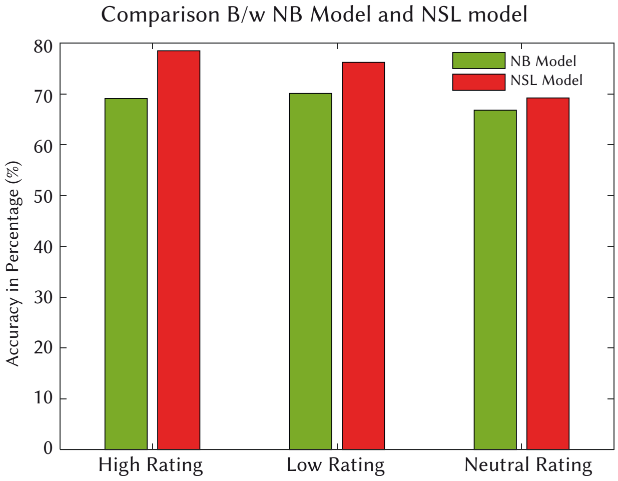


Fig. 16. Combined NSL Model and Naïve Bayes.

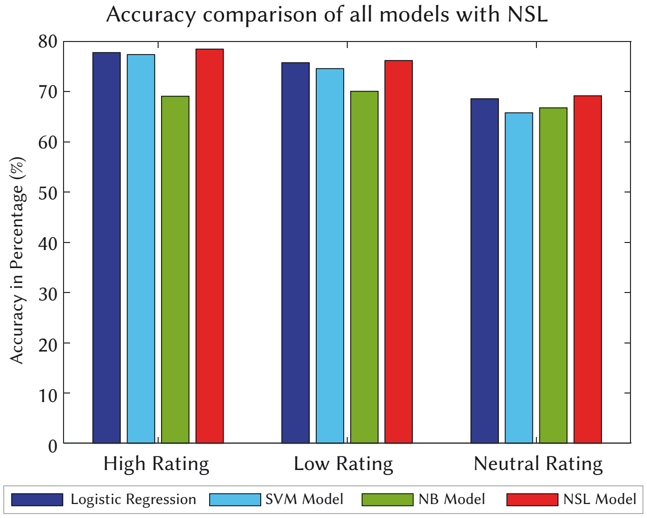


Fig. 17. Accuracy Comparison of all the classifiers.

VI. Conclusion And Future Scope

In this work, we present NSL, a new classification model as a 2-level combination of existing ones (namely, Naïve Bayes, SVM, and logistic Regression classifier). For data preprocessing the underlying method Latent Dirichlet Allocation (LDA and k-means clustering algorithm are used. The LDA divides all reviews based on topic discussed within the text. All the used existing models are combined in stack manner. The predictive output of each model is converted to a second level data, making each prediction a new feature for this second level. After applying the meta classifier on training data, it is observed that the outcome of meta classifier is almost similar to the best underlying model. NSL successfully predicts a user's numerical rating from its review text content. The performance of the classifier is based on the necessity in terms of effectiveness, and it is also concerned with the number of features to be taken care during training and validation phase.

Experimentation has been done by means of "Home and Kitchen" dataset from Amazon. After a preprocessing aiming to balance the dataset, we train and test all four models and compare their performances. The outcomes of this experimental work are based on one type and category of the dataset. Further this classification can be used with other category of the dataset like food product, electronics appliances, delivery rating of product given by users. According to the evaluation metric, SVM gives the best result among multiclass classifiers but Logistic regression gives somehow better result than SVM. However, our proposed model overcomes, in terms of accuracy, all other models. Further scope of the underlying classifier extends for web page classification, electronic-mail classification, detection and classification of unauthorized signatures by combining of Hidden Naive Bayes and NBTree to decrease the Error rate of the classifier. When these enhancements are incorporated in the underlying classification system, it would help further improve the performance and be useful for applications meant for the explicit classification system.

References

[1] Y. Qiang, R. Law, B. Gu, and W. Chen. "The influence of user-generated content on traveler behavior: An empirical investigation on the effects of e-word-of-mouth to hotel online bookings." *Computers in Human behavior* 27, no. 2, pp. 634-639, 2011.

[2] G. Gayatree, N. Elhadad, and A. Marian. "Beyond the stars: improving rating predictions using review text content." *In WebDB*, vol. 9, pp. 1-6. 2009.

[3] B. Stefano, A. Esuli, and F. Sebastiani. "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." *In Lrec*, vol. 10, no. 2010, pp. 2200-2204. 2010.

[4] K. B. Sotiris, I. Zaharakis, and P. Pintelas. "Supervised machine learning:

A review of classification techniques." *Emerging artificial intelligence applications in computer engineering.* Vol. 160, no. 1, pp. 3-14, 2007.

[5] L. Pasquale, M. D. Gemmis, and G. Semeraro. "Content-based recommender systems: State of the art and trends." *In Recommender systems handbook,* pp. 73-105. Springer, Boston, MA, 2011.

[6] C. G. William, 2007. "Sampling techniques". *John Wiley & Sons*, 2007.

[7] T. F. Brian, J. H. Patterson, and W. V. Gehrlein. "A comparative evaluation of heuristic line balancing techniques." *Management science* 32, no. 4 (1986): 430-454.

[8] Y. P. Chaubey, "Resampling-based multiple testing: Examples and methods for p-value adjustment." (1993): 450-451.

[9] D. M. Hawkins, 2004. The problem of overfitting. *Journal of chemical information and computer sciences*, 2004, 44(1), pp.1-12.

[10] S. Wararat. "The analysis and prediction of customer review rating using opinion mining." *In 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 71-77. IEEE, 2017.

[11] L. Xiaojiang, X. Qian, and G. Zhao. "Rating prediction based on social sentiment from textual reviews." *IEEE transactions on multimedia* 18, no. 9 (2016): 1910-1921.

[12] B. Moez, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. "Sequential deep learning for human action recognition." *In International workshop on human behavior understanding,* pp. 29-39. Springer, Berlin, Heidelberg, 2011.

[13] S. C. Reddy, K. U. Kumar, J. D. Keshav, B. R. Prasad, and S. Agarwal. "Prediction of star ratings from online reviews." *In TENCON 2017-2017 IEEE Region 10 Conference*, pp. 1857-1861. IEEE, 2017.

[14] K. Noriaki. "Predicting future reviews: sentiment analysis models for collaborative filtering." *In Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 605-614. 2011.

[15] H. Jiawei, and KC-C. Chang. "Data mining for web intelligence." *Computer* 35, no. 11 (2002): 64-70.

[16] P. D. Turney, and M. L. Littman. "Unsupervised learning of semantic orientation from a hundred-billion-word corpus." *arXiv preprint* cs/0212012 (2002).

[17] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." *Emerging artificial intelligence applications in computer engineering* 160, no. 1 (2007): 3-24.

[18] A. B. Goldberg, and X. Zhu. "Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization." *In Proceedings of TextGraphs: The first workshop on graph based methods for natural language processing,* pp. 45-52. 2006.

[19] L. LiZhen, W. Wang, and H. Wang. "Summarizing customer reviews based on product features." *In 2012 5th International Congress on Image and Signal Processing*, pp. 1615-1619. IEEE, 2012.

[20] M. Jeff, and M. Lapata. "Vector-based models of semantic composition." *In proceedings of ACL-08: HLT,* pp. 236-244. 2008.

[21] T. Amit, S. Berkovsky, M. A. Kaafar, D. Vallet, T. Chen, and T. Kuflik. "Improving business rating predictions using graph based features." *In Proceedings of the 19th international conference on Intelligent User Interfaces*, pp. 17-26. 2014.

[22] Y. Zhang, R. Jin, and Z.H. Zhou. "Understanding bag-of-words model: a statistical framework." *International Journal of Machine Learning and Cybernetics* 1, no. 1-4 (2010): 43-52.

[23] A. Tiroshi, S. Berkovsky, M. A. Kaafar, D. Vallet, T. Chen, and T. Kuflik. "Improving business rating predictions using graph based features." *In Proceedings of the 19th international conference on Intelligent User Interfaces*, pp. 17-26. 2014.

[24] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." *Emerging artificial intelligence applications in computer engineering* 160, no. 1 (2007): 3-24.

[25] D. M. Greig, B. T. Porteous, and A. H. Seheult. "Exact maximum a posteriori estimation for binary images." *Journal of the Royal Statistical Society: Series B (Methodological)* 51, no. 2 (1989): 271-279.

[26] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection." *In Ijcai*, vol. 14, no. 2, pp. 1137-1145. 1995.

[27] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao. "Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey." *Multimedia Tools and Applications* 78, no. 11 (2019): 15169-15211.

[28] A. K. Jain, "Data clustering: 50 years beyond K-means." *Pattern recognition letters* 31, no. 8 (2010): 651-666.

[29] J. Ramos, "Using tf-idf to determine word relevance in document queries." *In Proceedings of the first instructional conference on machine learning*, vol. 242, pp. 133-142. 2003.

[30] T. K. Landauer, P. W. Foltz, and D. Laham. "An introduction to latent semantic analysis." *Discourse processes* 25, no. 2-3 (1998): 259-284.

[31] E. R. Henry, and J. Hofrichter. "Singular value decomposition: Application to analysis of experimental data." *In Methods in enzymology*, vol. 210, pp. 129-192. Academic Press, 1992.

[32] L. V. Maaten, and G. Hinton. "Visualizing data using t-SNE." *Journal of machine learning research* 9, no. Nov (2008): 2579-2605.

[33] L. E. Sucar, "Probabilistic graphical models." *Advances in Computer Vision and Pattern Recognition. London: Springer London.* doi 10 (2015): 978-1.

[34] J. Friedman, T. Hastie, and R. Tibshirani. The elements of statistical learning. Vol. 1, no. 10. *New York: Springer series in statistics*, 2001.

Pravin Kumar

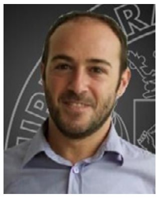
Pravin Kumar is Member of Technical Staff Engineer at Mavenir System, Bangalore. He is currently working in Research and Development Department as a Research and Product development Engineer. Prior to Mavenir he Worked as a senior Research Engineer includes Autonomous Driving, Image processing and Algorithm Design and Analysis for Automotive components to make Driving Intelligent and Safe. He received his master's degree in Computer Science and Engineering from Indian Institute of Technology (ISM) Dhanbad. He holds a bachelor's degree in Computer Science & Engineering from University Of Pune. His research interests include machine learning, Internet of Things, 4G/5G Protocol Stack Development, Algorithm Design And Analysis and Swarm intelligence.

Mohit Dayal

Mohit Dayal is Technical committee Member of IEEE INDIaCom international conference, Delhi and Editorial member of International journal of Recent Advances in Science and Technology. He is currently working in Central University of Haryana, Mahendergarh, Haryana as an assistant professor in computer science and engineering Department. He received his master's degree in Information Security from Ambedkar Institute of Advanced Communication Technologies & Research of Guru Gobind Singh Indraprastha University, Delhi. He holds a bachelor's degree in Computer Science & Engineering from Guru Gobind Singh Indraprastha University, Delhi. His research interests include machine learning, Internet of Things, Big Data, Web Application attacks and information security.

Manju Khari

Manju Khari an Assistant Professor in Netaji Subhas University of Technology, East Campus, Delhi, India formerly Ambedkar Institute of Advanced Communication Technology and Research, Under Govt. Of NCT Delhi affiliated with Guru Gobind Singh Indraprastha University, Delhi, India. She is also the Professor- In-charge of the IT Services of the Institute and has experience of more than twelve years in Network Planning & Management. She holds a Ph.D. in Computer Science & Engineering from National Institute Of Technology Patna and She received her master's degree in Information Security from Ambedkar Institute of Advanced Communication Technology and Research, formally this institute is known as Ambedkar Institute Of Technology affiliated with Guru Gobind Singh Indraprastha University, Delhi, India. Her research interests are software testing, software quality, software metrics, information security, optimization and nature-inspired algorithm. She has 70 published papers in refereed National/International Journals & Conferences (viz. IEEE, ACM, Springer, Inderscience, and Elsevier), 06 book chapters in a springer. She is also co-author of two books published by NCERT of Secondary and senior Secondary School.



Giuseppe Fenza

Giuseppe Fenza is currently an Associate Professor in Computer Science at Department of Management and Innovation Systems, University of Salerno, Italy. He received the Ph.D. degree in Computer Sciences at the University of Salerno, Italy, in 2009. From 2009, his research interests ware mainly focused on Knowledge Extraction from unstructured resources defining intelligent systems based on the combination of techniques from Soft Computing, Semantic Web, areas in which he has many publications. He was deeply involved in several EU and Italian Research and Development projects focused on Situation Awareness, Service Discovery, Enterprise Information Management and e-Commerce. He serves as Associate Editor in international journals, such as: Neurocomputing, International Journal of Grid and Utility Computing, International Journal of Engineering Business Management. He has published extensively about: Fuzzy Decision Making, Ontology Elicitation, Situation and Context Awareness, Semantic Information Retrieval. Recently, he is working in the field of Big Data, Social Media Analytics, and Web Intelligence by proposing novel methods for instance to support microblog summarization, time-aware information retrieval and recommendations extraction. In 2017, he co-founded of Riatlas srl, a company that operates in the field of e-health. From 2018, he is participating in research projects focused in the area of big data analysis and analytics applied to Industry 4.0 and Cyber Physical Systems (CPS), that are: Leonardo 4.0 funded by the national Ministry of Education, Universities and Research; and CPS4EU funded by European ECSEL-IA H2020.

Maria Cristina Gallo

Maria Cristina Gallo received her Master Degree in Computer Science at the University of Salerno, Italy, in 2009. From 2009 to 2017, she collaborates to several research initiatives and projects mainly focused on Computational Intelligence, Data Mining, Ontology Learning e Semantic Information Retrieval in different domain, such as health, e-commerce, and enterprise. Recently, she has worked in the field of social Media Analytics and Semantic Web to study users' interests, characteristics of their posts, and potential cyclic nature of both of them. From 2017 until now, she is a PhD student in Big Data Management at the University of Salerno and she is involved in a project regarding Pattern Recognition and anomaly detection in data streams.