Advanced Recommendation System Using Sentiment Analysis

Jaineel Mamtora, Suraj Chatterjee, Shawn Almeida and Vandana Patil

August 31, 2021

Advanced Recommendation System Using Sentiment Analysis

[1] Jaineel Mamtora
Dept. of Information Technology
St. Francis Institute of Technology
Mumbai, India
jmamtora99@gmail.com

[2] Suraj Chatterjee
Dept. of Information Technology
St. Francis Institute of Technology
Mumbai, India
surajchatterjee5111@gmail.com

[3] Shawn Almeida
Dept. of Information Technology
St. Francis Institute of Technology
Mumbai, India
shawn9920@gmail.com

[4] Vandana Patil
Dept. of Information Technology
St. Francis Institute of Technology
Mumbai, India
vandanapatil@sfit.ac.in

***Abstract*—Recommendation Systems are the heart of E-Commerce. They help consumers with the right contrast of products or items thereby increasing the revenue of the company. The current recommendation systems generally use click-based recommendation models wherein the products, movies, songs or any item, are registered and they are recommended respectively. Some platforms even use the customer star ratings and reviews to recommend the products. But the limitation of that is professional reviewers and general public commonly post their reviews on the product on E-commerce websites. The purpose of Advanced Recommendation System using Sentiment Analysis is to elevate the recommendations provided to the users by gathering the users' sentiments. We are planning to use the sentiments of the people to further provide an advanced understanding of the people's view on the product and suggest that to the future customers accordingly. Our paper will use click-based recommendation system as well as sentiment analysis on product reviews to look for the general sentiment of the item or the product and finally recommend the product to a consumer.**

***Index Terms*—Content-Filtering Approach; K-Nearest Neighbour Algorithm; Recommendation System; Sentiment Analysis.**

I. Introduction

Nowadays, customers generally get the products that they want in short amount of time but often struggle in finding relevant/similar products so as to compare them and get the best choice. This is where Recommendation Systems comes into picture. A Recommendation System refers to a system that is capable of predicting the future preference of a set of items for a user, and recommend the top items. Now, even if people get relevant products, they can't comment on their authenticity. For that, they have to search for reviews and ratings of the product to be purchased. Sentiment Analysis is a Natural Language Processing (NLP) technique used to determine whether data (usually text) is positive, negative or neutral.

The proposed approach is to design a system where the product is recommended based on the similarity of the product previously chosen as well as extract the sentiments from the reviews provided by other customers.

The paper is structured as follows: Section 2 begins with re-evaluating the most relevant work done in Recommendation Systems and Sentiment Analysis. The detailed proposed approach is discussed in Section 3. Section 4 proceeds with discussing the results and the accuracy of the algorithms and techniques used in proposed approach; Concluding remarks and Future Work of the paper are mentioned in Section 5.

II. Related Work

There have been countless publications on recommender systems and sentiment analysis, evidence growing interest in their development and deployment. Publishers have made massive research on the topic and have introduced various advancements in the field of Machine Learning (ML) and Artificial Intelligence (AI). Below are the research papers referred:

A. Literature review related to existing system / methodology

Shipra Narang and Nikita Taneja [1] have used hybrid approach in target domain of movie recommendations based on user features which included user search logs, news article browser history, music browsing history and movie search logs. User features domain was constructed from users' search queries and clicked URLs, represented by a unique user id. To reduce feature dimension, URLs were shortened into domain-level and queries were normalized and split using letter uni-gram features. Finally, the neural network had all the above features to get the recommended products.

Yonas Woldemariam [2] had introduced the implementation and integration of a sentiment analysis pipeline into the ongoing open source cross-media analysis framework. The pipeline included Chat room cleaner, NLP and sentiment analyzer. It likewise analyzed two general classes of sentiment analysis techniques, in particular lexicon based and machine learning approaches. Also, this paper mainly focused on which method was appropriate to detect sentiments from forum discussion posts. This paper worked on the improving accuracy of sentiment analysis and improve performance of the chosen algorithms.

Qian Zhang, Peng Hao, et al. [3] have focused on automatic captures of the semantic relationships between non-identical tags and apply them to the recommendation. The word2vec technique was used to learn the latent representation of tags. Semantically identical labels were then gathered to frame a joint embedding space which included label bunches. That embedding space served as the bridge between domains. By planning clients and things from both the source and target areas into the equivalent embedding space, comparable clients or things across areas were distinguished. Along these lines, the suggestion in an inadequate target domain was improved by transferring knowledge through related clients and items. This paper aimed to exploit the semantic information in tags as a bridge between the source and target domains in cross-domain recommendation. Semantically similar but nonidentical tags were used to overcome data sparsity problem and establish close relationships between two domains.

Sana Nabil, Jaber Elbouhdidi, et al. [4] have presented their approach of recommendation which was based on the analysis of the feelings of the users from consumer reviews on articles. It also compared the various types of recommender systems to observe the performance of these systems in various environments. This paper also presented an approach based on sentiment analysis to build an innovative and powerful recommendation model and its application on twitter in order to improve the functionality of e-commerce platforms.

Jian Yu, Yongli An, et al. [5] have proposed a recommendation algorithm based on the content sentiment analysis and proposed algorithm improved the performance of the traditional product recommendation algorithm based on collaborative filtering. The experimental results showed that the precision of the proposed recommendation calculation was dependent on sentiment analysis which was somewhat higher than the recommendation algorithm based and dependent on collaborative filtering. This paper made use of the sentiment analysis technology and proposed the comprehensive similarity which combined the user's sentiment similarity and the user's score similarity.

B. Literature review related to Algorithms

Shipra Narang and Nikita Taneja [1] have used Deep Structured Semantic Models (DSSM) which matched web search queries and URL based documents. In this model, the input (raw counts of terms in a query or a document without normalization) to the Deep Neural Network (DNN) was a high dimensional term vector and the output of the DNN was a concept vector in a low-dimensional semantic feature space. DSSM was utilized to create latent semantic models that extended substances of various sorts (e.g., questions and documents) into a typical low-dimensional semantic space for an assortment of machine learning tasks such as ranking and classification.

Yonas Woldemariam [2] had incorporated Lexicon-based algorithm to tokenize the sentences and assign polarity, Stanford Sentiment Treebank and Recursive Neural Tensor Network (RNTN) to accurately predict the compositional semantic. Lexicon based approach utilized the sentiment lexicon to describe the polarity (positive, negative and neutral) of a textual content. Recursive neural tensor networks (RNTNs) were used where neural nets were useful for natural-language processing. They had a tree structure with a neural net at each node. Recursive neural tensor networks were used for boundary segmentation, to determine which word groups were positive and which were negative.

Qian Zhang, Peng Hao, et al. [3] have used Flexible Mixture Model (FMM) to factorize the user-item rating matrices in both the source and the target domain, Radial Basis Function (RBF) to perform similarity measurement, Skip-Gram model with Negative Sampling (SGNS) to help speed up training time and improve quality of resulting word vectors. The evaluation metrics used were Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). FMM was used to extend existing partitioning/clustering algorithms for collaborative filtering by clustering both users and items together simultaneously without assuming that each user and item should only belong to a single cluster. Skip-gram Negative Sampling (SGNS) helped to speed up training time and improve quality of resulting word vectors. This was done by training the network to only modify a small percentage of the weights rather than all of them.

Sana Nabil, Jaber Elbouhdidi, et al. [4] have used data analysis using content based, collaborative based and hybrid systems for recommendation and sentiment analysis.

Jian Yu, Yongli An, et al. [5] have implemented Naive Bayes Classification algorithm to classify sentiment polarity, Collaborative Filtering Recommendation algorithm Based on Sentiment analysis (CFRBS) algorithm to improve the accuracy of product recommendation which is based on the traditional collaborative filtering algorithm. CFRBS algorithm was used to calculate the comprehensive similarity between the users through combining the sentiment similarity and the score similarity to receive a score as predict score for product.

III. Proposed System

A. Raw Data Collection

Data (additionally referred to and frequently alluded to as Primary Data) assortment is the beginning stage of any Data analysis. Once when the RD (Raw Data) is gathered, it is processed to transform into Information that can be converted into knowledge further down the analysis track. Searching and contrasting text reviews can be baffling for users. Subsequently we need better numerical ratings framework dependent on the reviews which will make clients purchase effortless.

While making their decision, consumers need to discover helpful reviews as fast as conceivable utilizing the rating system. Therefore, models ready to foresee the client's rating from the text content are of great significant. The consumer's experience can be improved by getting an overall sense of a textual review. Likewise, it can assist organizations with expanding deals, and improve the item by understanding client's requirements.

The Amazon electronics product review dataset was selected. The user's feedback and ratings of various products, as well as reviews about the user's experience with the product(s), are also taken into account. The Amazon reviews and product details make up the electronics dataset. This dataset contains product metadata (descriptions, category information, value, brand, and image features) as well as reviews (ratings, text, and helpfulness votes).

While loading data into Python DataFrame we import CSV (comma separated files) files which is made extremely simple with the read_csv() function in Pandas.

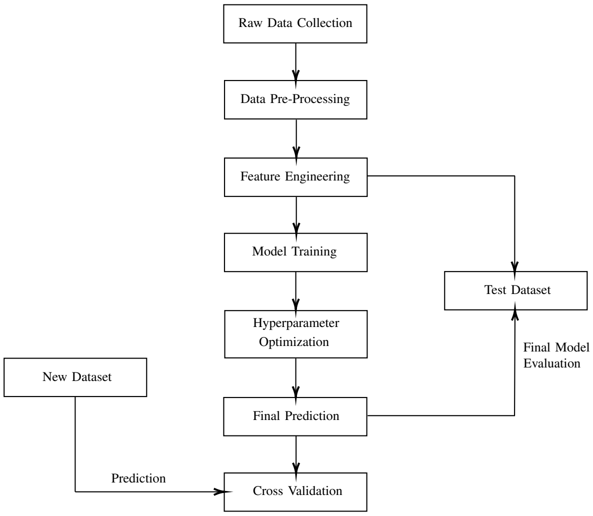

Fig. 1: Process Diagram of Proposed Approach

B. Pre-Processing:

Data pre-processing is a data mining technique for transforming raw data into a format that is both useful and effective. Data transformation and data reduction are the two steps involved in data pre-processing.

In this process, the data is getting filtered with more than 20 reviewers by taking 50 sample reviews of each product. This was done by taking 20 distinct reviews pertaining to a particular product and then sorting it down further.

VADER (Valence Aware Dictionary and Sentiment Reasoner) is now a lexicon and rule-based sentiment analysis platform designed to extract sentiments shared in social media. Under the MIT License, it is completely open-source. A dictionary or a list of words is referred to as a Lexicon. A compilation of all possible positive terms, for example, is called a positive lexicon (like good, amazing, appreciable etc.). A negative lexicon, on the other hand, is a compilation of all possible negative terms (like bad, abolish, ambush etc.)

The Rule based approach of sentiment analysis passes each word of the product review text through certain two dictionaries or lexicons (positive and negative word lists), then counts the amount of positive and negative terms in the text. It is possible to decide whether the review is positive or negative based on that amount.

VADER employs the same strategy, but in a clever manner. Emoji were also used to measure sentiment. Emojis are often used to convey sentiment on social media. As a result, VADER is a fantastic tool for analysing social media sentiment.

After the data is loaded, it is grouped by one or more values, and then some calculations are performed. Grouping chunks of data by product-id is the most effective method of doing so.

The sentiment of the feedback, as well as the product's ratings, are now taken into account. The product's score indicates how good or poor the product is based on the user's experience. It ensures that all unwanted data is removed or cleaned from the system. When the text cleaning process begins, all unnecessary characters and symbols are removed from the results. After that, it's translated to lower case.

C. Feature Engineering

Feature engineering is the method of extracting features from raw data using domain information and data mining techniques. These characteristics can help machine learning algorithms perform better. Applied Machine Learning can be considered as a sub-domain of feature engineering.

Machine Learning algorithms are typically more effective with numbers than with text. CountVectorizer is used to translate the textual feedback into vectors. CountVectorizer is a programme that converts a text into a vector based on the frequency (count) of of word that appears in the text. When a dataset includes several texts of the same set of characters, this is useful. Each unique word is represented by a column in the matrix, and each text sample from the document is represented by a row in the matrix.

Each column's set of numbers will now have different lower and upper limits. As a result, the vectors are scaled using StandardScaler. When the ranges of input datasets vary significantly, or simply when they are calculated in different units of measure, a standardscaler is used. The mean is removed and the data is scaled to the unit variance with StandardScaler. The following is how it works:

We perform Standardization as

$$z = \frac{x - \mu}{\sigma}$$

with Mean as

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (x_i)$$

and Standard Deviation (SD) as

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2}$$

However, outliers have an influence when calculating the empirical mean and standard deviation, which narrows the range of characteristic values. Now when there is abundant rows and columns in the dataset, there is a primary need to reduce it. This process is done by using PCA (Principal Component Analysis).

The number of columns is reduced to a considerable size using Principal Component Analysis (PCA). Principal Component Analysis is basically a statistical procedure that uses an orthonormal transformation which converts a set of correlated variables to a set of uncorrelated variables.

Now the target column is labelled. The target column is mostly the unique id of an entry (product id) which is in String format. Now the string format isn't a preferable format. The numerical data is preferred over it. Therefore to overcome this, it is encoded in a label format. The labels are in numerical format. This process is known as encoding.

D. *Model Training*

A training model is a dataset used to train a machine learning algorithm. It is made up of sample output data as well as the related sets of input data that have an effect on the output. The training model is used to process the input data through the algorithm in order to compare the processed output to the sample output. The precision of the proposed model is determined by the type of problem being addressed. The coefficient of determination, root-mean-square error, mean absolute error, and other related terms are used in regression analysis. Accuracy, precision, recall, F1 ranking, and other similar measures are often used to solve classification problems. The most important part is to understand that unbiased evaluation is needed to properly use these measures, assess the predictive performance of the model and validate the model.

This means that the model's predictive output cannot be evaluated using the same data that was used for training. The model must be evaluated using data that has never been used before by the model. This is done by breaking the dataset before using it. The training set is used to prepare the model for use. During hyperparameter tuning, the validation collection is used for unbiased model evaluation. The test set is needed for a fair assessment of the final model. It isn't used for validation or fitting.

The model can now be trained using a Supervised Learning algorithm since the dataset has been labelled. A supervised learning model accepts a collection of input objects and produces a set of output values. The model is then trained on that data to learn how to map inputs to desired outputs, allowing it to make predictions on data that has yet to be seen. KNN (K Nearest Neighbours) is a Supervised Learning algorithm that searches for the "k" closest labelled data points from a given data point. The majority of the 'k' closest points' labels are then allocated to the data point.

E. *Hyperparameter Optimization*

Hyperparameter optimization in machine learning intends to find the parameters or features of a given machine learning algorithm that deliver the best performance as measured on a validation set. Hyperparameters, in contrast to model parameters, are set before training.

F. *Testing the model*

The ML model is tested on a test dataset after it has been trained using a training set. The test data offers a fantastic opportunity to assess the model. After the ML model has been correctly trained using the training set, the test dataset is used. The test dataset is derived from the same data set that provided the training set. The original dataset is divided into two parts: train data and test data. Data is divided into a 7:3 ratio, with 70 of the data being used for training and 30 for research. The data used in test cases must be carefully chosen. The accuracy of a test case document's findings is largely determined by the test data.

G. *Final Prediction*

This task is basically analysis of numerical prediction. This stage predicts the final output of the system. The final products are predicted and is displayed to the user.

H. *Implementation*

The implementation is divided into three sections:

a) ML Model: The details of the implementation is already mentioned in the section 3 of the paper.

b) Graphical User Interface (GUI): The Home Page, Log In, Sign Up, Contact Us and the Checkout page are created using Hypertext Markup Language (HTML) and styled using Cascading Style Sheets (CSS). The GUI makes it easy for the user to navigate through multiple pages. Various routes are defined for different pages to interconnect and integrate them. Once the Home page is loaded up, random products are generated (for a new user) over the home page systematically. The products, once clicked, loads the user into the product page. The products are mapped to their respective product id which further retrieves details from the data base (MongoDB) and is returned to the website. The factors that play a role in generating recommended products are calculated by the product id i.e. product id with a similar taste (similarity of the product). The reviews and sentiment score are provided too i.e. a user can gain an overall idea of the product based on other's review. The site displays the positive and negative reviews provided by the users. Recommendation is also based on this factor. The checkout page is where the user when done adding all the necessary products makes a final call on the purchases and pays the bill. The user can finally end the session by just logging out.

c) Web Framework: Flask is a great Micro Web Framework written in Python. The ML Model and the GUI are connected using Flask. The model (.sav) first needs to be loaded. Also the data files i.e. sentiment meta data and product meta data which is stored in (.csv) format are also loaded.

This paper uses MongoDB (Mongo Data Base) as a back end database for the login and sign-up into the GUI platform. MongoDB is a NoSQL, cross-platform, document-oriented database program. It helps in storing the credentials of the user along with creating and maintaining the sessions and cookies. PyMongo module is imported in the Flask which provides a set of tools and to establish connection with the

MongoDB. Passlib module is used for the encryption of the user's password and storing the same in the database. It is a comprehensive password hashing framework which supports over 30 hashing schemes.

IV. Results And Discussion

Accurate recommendations of products were made by the model according to product reviews. Sentiment analysis played a great role for analyzing customer's views on products and further enriching the model. Natural Language Processing techniques were used for creating embeddings using customer reviews for the recommendation. A good user experience was provided in the form of personalization and clean GUI.

Below table represents the variation of value of precision over changes in parametric values of the KNN model. Cosine similarity is used to help measure the Precision metric for the model. If the cosine similarity score between the tags of the original and recommended product is higher than the threshold value, a hit is registered and with the help of this, the precision metric of the model is calculated. Furthermore, detailed graphs of various analysis are also included, consisting of the line graphs of rating and sentiment score of the products and a Pearson correlation heatmap depicting the correlation between the rating and the sentiment score. The correlation coefficient of 0.52 as observed in the heatmap helps to justify the need of sentiment score as a parameter for the training of the model.

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|a\|\|b\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}}$$

where $\vec{a}\cdot\vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots\cdot + a_n b_n$ is the dot product of original product and recommended product.

TABLE I: Precision Analysis of the KNN Model

p \n_neighbors	1	5	7
1	84.226 %	78.407 %	78.101 %
2	85.758 %	78.710 %	78.101 %
3	83.307 %	78.866 %	78.407 %
4	83.001 %	78.713 %	78.254 %

p: power parameter for the Minkowski metric
n_neighbors: number of neighbors

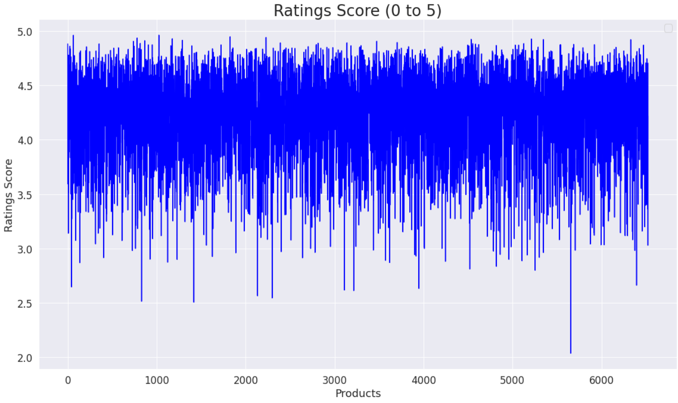

Fig. 2: Line Graph of Rating Score VS Products

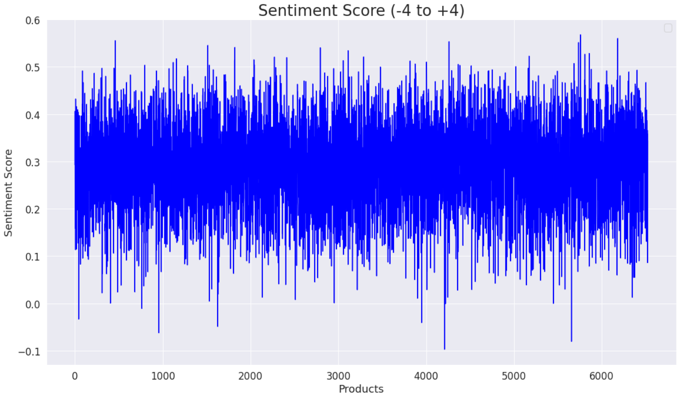

Fig. 3: Line Graph of Sentiment Score VS Products

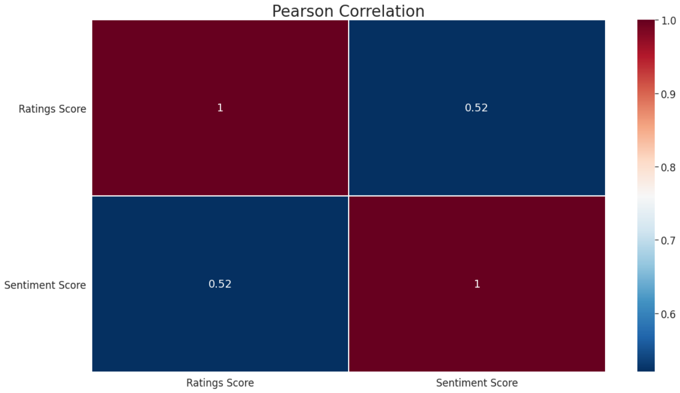

Fig. 4: Pearson Correlation Heat-Map of Rating Score and Sentiment Score

V. Conclusion And Future Work

The advanced recommendation system was successfully implemented which was able to recommend products based on the products previously browsed and/or clicked. We used Sentiment Analysis on reviews provided by other customers who previously bought products and thus provided appropriate recommendations. This system is projected to learn various types of user inputs on a particular product and further down the line it shall recommend the user to what s/he should buy. The proposed system has overall accuracy of above 87 %. The project also aims to further introduce newer Machine Learning models with a better intention to handle larger data. Also, Deep Learning (DL) based models and CNN based models can be implemented to get better personalisation of products. If more computation power were to be provided, the models would become easier to implement using DL based algorithms.

References

[1] Shipra Narang, Nikita Taneja, "Deep Content-Collaborative Recommender System", IEEE International Conference on Advances in Computing, Communication Control and Networking, Faridabad, India, 2018, 110 - 116.

[2] Yonas Woldemariam, "Sentiment Analysis in A Cross-Media Analysis Framework", IEEE International Conference on Big Data Analysis (ICBDA), Hangzhou, China, 2016, 1 - 5.

[3] Qian Zhang, Peng Hao, Jie Lu, Guangquan Zhang, “Cross-domain Recommendation with Semantic Correlation in Tagging Systems”, IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 2019, 1 - 8.

[4] Sana Nabil, Jaber Elbouhdidi, Mohamed Yassin, “Recommendation system based on data analysis – Application on tweets sentiment analysis”, IEEE 5th International Congress on Information Science and Technology (CiSt), Tétouan, Morocco, 2018, 155 - 160.

[5] Jian Yu, Yongli An, Tianyi Xu, Jie Gao, Mankun Zhao, and Mei Yu, “Product Recommendation Method Based on Sentiment Analysis”, Springer International Conference on Web Information Systems and Applications 2018, Tianjin, China, 2018, 488 - 495.