# End-Semester Project Report

**Team Name : Team Gryffindor**
**Project Name : Content-Based News Recommendation System using SVM**
School of Engineering and Applied Science, Ahmedabad University
Machine Learning - CSP523, Winter 2021


1st Jeet Shah
*AU1841006*
*Ahmedabad University*
Ahmedabad, India
jeet.s3@ahduni.edu.in

2nd Manav Vagrecha
*AU1841022*
*Ahmedabad University*
Ahmedabad, India
manavkumar.v@ahduni.edu.in

3rd Parth N. Patel
*AU1841028*
*Ahmedabad University*
Ahmedabad, India
parht.p1@ahduni.edu.in

4th Shreyansh Shah
*AU1841046*
*Ahmedabad University*
Ahmedabad, India
shreyansh.s@ahduni.edu.in



***Abstract*—In this work we have developed a content based news recommendation system. For this we've used two different approaches. First we tried to develop recommendation system using similarity based approach. In this approach user profile vector is calculated from user's previous interaction and it indicates the features of the user. Document vector is calculated from the news dataset and that indicates the features of the news dataset. And then using cosine similarity new articles are recommended to the user. But this approach has a bottleneck that if the news dataset size increases then news dataset features will increase exponentially and that will increase the complexity. To overcome this, we are using classification approach based on support vector machine classifier. Here two classes are defined as: 1)Relevant to the user and 2)Irrelevant ti the user. The classifier will classify news article into these two classed and from the class relevant to the user we'll make recommendation to the user.**



***Index Terms*—News Recommendation System, Content-Based Filtering, cosine similarity, support vector machine**


## I. Introduction & Background

A recommendation system is a system that suggests products, services, information, to users based on analysis of data collected by different processes including user interactivity, explicit user ratings, similarity amongst users, similarity amongst items, etc. Thus, there are, in detail, 6 different types of recommendation systems. [3]

- Content-Based
- Collaborative
- Demographic
- Knowledge-based / Similarity-based
- Hybrid

In this work, we majorly focus on Content-based filtering. A content-based recommendation system recommends based on user's profile, his interests and previous interactions with the system. Thus, in a content-based recommendation system the algorithms used are such that it recommends users similar items that the user has liked in the past or is examining currently. It basically works on user feedback of different articles. There are 2 ways to get user preferences.

- Implicit - Track Click based Activity of user
- Explicit - Requires the user to evaluate viewed articles

Based on the user's preferences build on his/hers interactivity, we will classify all the articles available into 2 categories:

- Relevant to the user
- Irrelevant to the user

We classify on the basis of categories and sub categories of the news articles selected or rejected by the user. We have used Support vector machine classifier to do so. Once we have a hyperplane from svm classifier we predict, on what side of the plane a given article will fall and if it is on the positive side, we recommend it to the user.

## II. Literature Review

### A. Conventional Approach for content based recommendation

We explored various articles on recommendation systems. After a lot of research we finalised the article "Using Content-Based Filtering for Recommendation" [3]. It clearly guides through the entire process of content-based recommendation starting from data cleaning to feature extraction to determining cosine similarity.

### B. Classification Based Approach

We know, that SVM is the the state of the art for classification based problems. The main problem we faced is that our data was categorical. Non-numerical data such as categorical data are common in practice. While some classification methods like KDC are adaptive to categorical variables, SVM can only be applied to continuous numerical data. So, we need to use some encoding schemes like one hot encoding to convert this categorical data into numeric form. We reviewed an article that compared performance of KDC, SVM and KNN on such categorical data [**?**]. SVM showed the best performance and so we decided to use it on our data set.

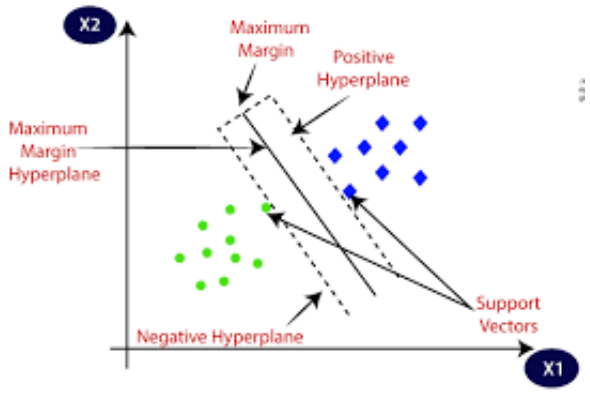



Fig. 1. Linear SVM

## III. Implementation - Classical Approach

### A. Data Information

*1) Description:* Microsoft News Dataset (MIND) is a large-scale dataset for news recommendation research. It was collected from anonymized behavior logs of Microsoft News website. The mission of MIND is to serve as a benchmark dataset for news recommendation and facilitate the research in news recommendation and recommender systems area.

*2) Resources:* Following are few resources from where we can get the dataset

- https://www.kaggle.com/arashnic/mind-news-dataset
- https://msnews.github.io/#about-mind

### B. Data Preprocessing

- Here, initially, we removed Redundant Columns from the news and behavior dataset. We preserved important fields like News-ID, Category, Sub-Category, Title, Abstract in the news Dataset and Index, User-ID, History, Impressions in the behaviour dataset.
- Then we removed the rows containing the NaN values for the Title and Abstract fields from the news dataset.
- Then we removed punctuation, stop words and html words - We have used Natuaral Language Toolkit (NLTK) library for removing stop words. [Stop words are the words which occur a large number of times and don't hold major meaning eg. I, the, a, as,...]

| | News-ID | Cleaned-Data |
|---|---|---|
| 0 | N55528 | shop notebooks jackets royals ca live without brands queen elizabeth prince charles prince phili... |
| 1 | N19639 | seemingly harmless habits holding back keeping shedding unwanted belly fat good worst habits bel... |
| 2 | N61837 | ivan molchanets peeked parapet sand bags front line war ukraine next empty helmet propped trick ... |
| 3 | N53526 | felt like fraud nba wife help fact nearly destroyed nba wife affected mental health |
| 4 | N38324 | seem harmless good reason ignore post get rid skin tags according dermatologist appeared first r... |

Fig. 2. Data after performing Preprocessing

### C. Feature Extraction

After obtaining the preprocessed data, we apply N-grams model to it which helps us to obtain unigrams, bigrams and trigrams of the for each article. Here, we will be

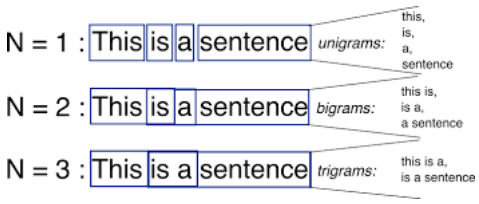



Fig. 3. N-Grams Language Model

```
auto
auto club
average
average price
customer
customer info
data
data reveals
high
high school
lowell
lowell galileo
offers
offers winter
oklahoma
oklahoma deer
roadside
roadside assistance
```

Fig. 4. Feature Extraction

Assigning weight to a given feature for a given article that signifies its importance in the given article using TF-IDF.

$$w_i = TF_i \times IDF_i$$
$$w_i = TF_i \times log(\frac{n}{df_i})$$

where $w_i$ are the weightages for ith term, $tf_i$ = number of occurrences of term $t_i$ in article D, n = number of articles in the dataset, $df_i$ = number of articles in which term $t_i$ appears atleast once

## IV. Results - Classical Approach

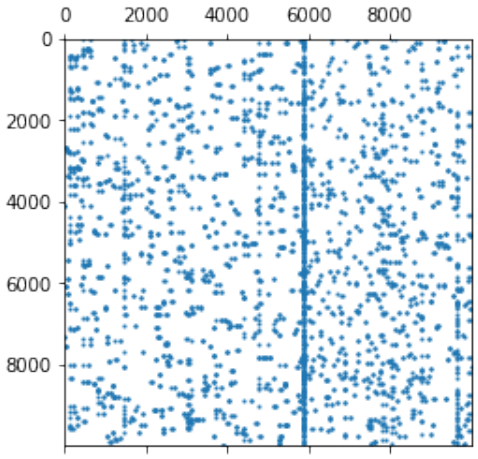



Fig. 5. Visualizing the Feature Matrix

We have got individual feature vector for each article where each feature is assigned a weight as per its significance in the article. Above image shows a cropped portion of the feature matrix whose actual dimensions are 48616x1869854, where 48616 is the total number of articles and 1869854 is the number of features

## V. Implementation - Modern Approach

### A. Data Information

Please refer to the Data Information Section of Implementation - Classical Approach as the dataset is kept the same as it was in the classical approach.

### B. Data Preprocessing

- Here, initially, we removed Redundant Columns from the news and behavior dataset. We preserved important fields like News-ID, Category, Sub-Category in the news Dataset and Index, User-ID, History, Impressions in the behaviour dataset.
- Then we removed the rows containing the NaN values from the news dataset and behaviour dataset.
- Now split the news-ids from the history and impressions field of behaviour dataset for all users and created a separate column for the list of articles and their impressions.

### C. Mathematical Analysis

- Differnt Kernels in SVM
  - Kernels are the main hyper parameter of SVM.
  - They are used to maps the observation into feature space.
  - We have implemented 3 different kinds of kernels
    - Linear Kernel

      $$K(x_i, x_j) = x_i' \, x_j$$

      Here the only hyper parameter is the cost parameter C
    - Polynomial Kernel

      $$K(x_i, x_j) = (r + \gamma . x_i' \, x_j)^d$$

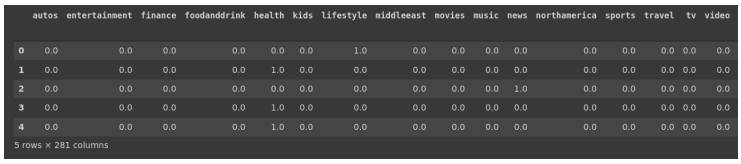
      Usually the parameter r is set to zero and $\gamma$ to a fixed value. Values of d are usually taken between 1 to 10.
    - Radial Kernel

      $$K(x_i, x_j) = e^{\gamma . x_i' \, x_j}$$

      This kernel provides the flexibility of separating observations and due to this flexibility it is the most successful kernel in SVM.
- Regularization parameter(C Parameter)
  It helps in optimization by telling how much we want to avoid miss classifying each training sample.For large value of Regularization Parameter, there will be smaller margin separating hyperplane and mostly all the training points are classified correctly. For small value of parameter, t here will be larger margin separating hyperplane and it misses more points to classify correctly.
- Gamma Parameter
  It defines how far the influence of single training sample reaches. If the value of gamma parameter is low than points far away from hyper plane are considered into calculations for separation line. For high value of gamma only the close points to the line are considered in calculation.

### D. Feature Extraction

As in this approach we will be using the predefined category and subcategory of the article to define the article i.e. all the categories and subcategories will be our features. Thus, After completing analysis over the preprocessed Data, we extracted features from the category and subcategory fields using OneHotEncoding method. Thus, we have a dataframe containing separate columns for each category and subcategory containing 1 value of each article in it's respective category and subcategory and zero elsewhere.



| | autos | entertainment | finance | foodanddrink | health | kids | lifestyle | middleeast | movies | music | news | northamerica | sports | travel | tv | video |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

5 rows × 281 columns

Fig. 6. Feature Extraction using OneHotEncoding

### E. Training - Testing Data Split

For a specific user, we have randomly split the data into training-testing in 8:2 ratio. Thus, we get $X_{train}, X_{test}, Y_{train}, Y_{test}$ from which ($X_{train}$ and $Y_{train}$) can be used for training and ($X_{test}$ and $Y_{test}$) can be used for testing. $X_{train/test}, Y_{train/test}$ contains the vector representation of the articles obtained from the history and impressions field and their impression vector respectively.

### F. Model Training

We have implementated feature-based SVM (linear SVM) and kernel-based SVM (Polynomial and Radial Basis Function SVM). We have tried tuning the hyperparameters manually to fit the training data for each type of SVM. Apart from manual tuning, we have also used GridSearchCV algo right to find the best set of hyperparameters

### G. Model Testing

Now we will give $X_{test}$ as input to the model to predict the classes, i.e. from recommendable class and non-recommendable class. Assume the array of predictions be $Y_{pred}$.

### H. Model Evalution

Using the obtained $Y_{pred}$ and existing $Y_{test}$ we will compute the confusion matrix for the Actual Values and Predicted Values for an article to be recommended. Now, from the confusion matrix, we will get the value of the evaluation metrics accuracy, precision and recall. We have also used the GridSearchCV Algorithm to find the best hyperparamters for the specific testing input. This will automate the process of finding the best set of parameters and also would select the best estimating model amongst 'linear', 'poly' and 'rbf' kernels. It will select the best estimator based on the best score obtained using those specific set of hyperparameters

## VI. Results - Modern Approach

```
Total length to check for best model and parameters:  529
Best Parameters:  {'C': 50, 'degree': 2, 'gamma': 0.005, 'kernel': 'rbf'}
Best Estimator:  SVC(C=50, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=2, gamma=0.005, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
Best Score:  0.7082706766917293
Accuracy: 0.7746478873239436
Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.60      0.60        20
           1       0.84      0.84      0.84        51

    accuracy                           0.77        71
   macro avg       0.72      0.72      0.72        71
weighted avg       0.77      0.77      0.77        71
```

Fig. 7. User 1 : Logs

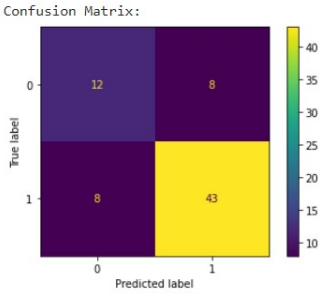



Fig. 8. User 1 : Confusion Matrix

```
Total length to check for best model and parameters:  529
Best Parameters:  {'C': 1, 'degree': 2, 'gamma': 3, 'kernel': 'rbf'}
Best Estimator:  SVC(C=1, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=2, gamma=3, kernel='rbf', max_iter=-1,
    probability=False, random_state=None, shrinking=True, tol=0.001,
    verbose=False)
Best Score:  0.6783559577677225
Accuracy: 0.7692307692307693
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.41      0.55        22
           1       0.76      0.95      0.85        43

    accuracy                           0.77        65
   macro avg       0.79      0.68      0.70        65
weighted avg       0.78      0.77      0.74        65
```

Fig. 9. User 2 : Logs

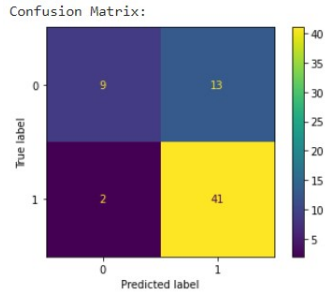



Fig. 10. User 2 : Confusion Matrix

## VII. Conclusion

Amongst 1000 users, we selected 25 random users, and we obtained the average accuracy of the model: 76.25%. Also, we noticed that the accuracy ranged from 67% to 96%. As it is well-known that performance of the model depends on various factors like Number of training points available, Frequency of positive and negative data points in the training data, Distribution of the data, the average accuracy may change upon different set of user's taken.

## References


[1] "Classifying Different Types of Recommender Systems," BluePi, 14-Nov-2015. [Online]. Available: https://www.bluepiit.com/blog/classifying-recommender-systems/. [Accessed: 17-Mar-2021].

[2] "Introduction to MIND and MIND-small datasets," [Online]. Available: https://github.com/msnews/msnews.github.io/blob/master/assets/doc/introduction.md

[3] "Using Content-Based Filtering for Recommendation," R. van Meteren and M. van Someren, [Online]. Available: http://users.ics.forth.gr/~potamias/mlnia/paper_6.pdf. [Accessed: 17-Mar-2021].

[4] "Preprocessing of Categorical Predictors in SVM, KNN and KDC (Contributed by Xi Cheng)." 18 Aug. 2020, https://stats.libretexts.org/@go/page/2488.