SHIKSHAK - E-Mentoring Application

C.V. Ajaykrishnan, Akhil Chandran, M.S. Ananthu Subramanian,
S. Mithun and T. Anjali

Abstract--- *This thesis aims to propose a web-based mentorship application that offers to help persons in need of help in certain areas of life. With social interactions and personal lives being hosted over the internet, a web-based application seems to be an appropriate place to create a safe space for people looking for guidance from professional mentors. This application provides a safe space by offering direct chat user-interface with an expert in the identified field of interest along with avoiding the hectic questionnaires, which are commonly found in other similar applications using text classification of entered queries to identify categories and recommend mentors. This is provided in hopes to give a clean, easy and more personalized user experience.*

Keywords--- *Dataset, NLP, OTP (One-Time-Password), Support Vector Machines (SVM), Sklearn, UX (User-Experience), Vector.*

I. INTRODUCTION[1]

A mentorship is a form of relationship between a mentee and their mentor which would aid the mentee in taking control of their own personal development and work-life under the guidance of a professional mentor. Mentoring helps the mentee to maximize his/her potential and help develop parts of their lives which they may be struggling with and thus can help themselves achieve their personal and professional goals.

In this technology-engulfed world, mentorship web applications can play a vital role in shaping up an individual when he/she enters the modern era. The most common challenges faced by the web application is finding the correct category and field for the mentee to be mentored in.

The user may find difficulty in choosing the right category for which he/she requires mentorship. To overcome this issue, the app provides a User-Interface which instead of forcing the user to select a category or choosing a mentee directly, he/she can type in a question/issue which may be found as difficult to solve, and the app will choose the right category for the proposed question and will direct the person to an appropriate mentor. The interaction between the mentee and the mentor will be via a chat user-interface. In order to provide the best services to the users, each mentee can rate the allotted mentor and can also request a change of mentor under the differing circumstance.

C.V. Ajaykrishnan, Department of Computer Science & Engineering, Amrita School of Engineering, Amrita Vishwa Vidaypeetham, Amritapuri, India. E-mail: ajaykrishnan@am.students.amrita.edu

Akhil Chandran, Department of Computer Science & Engineering, Amrita School of Engineering, Amrita Vishwa Vidaypeetham, Amritapuri, India. E-mail: akhiltrc@am.students.amrita.edu

M.S. Ananthu Subramanian, Department of Computer Science & Engineering, Amrita School of Engineering, Amrita Vishwa Vidaypeetham, Amritapuri, India. E-mail: ananthusubramanian@am.students.amrita.edu

S. Mithun, Department of Computer Science & Engineering, Amrita School of Engineering, Amrita Vishwa Vidaypeetham, Amritapuri, India. E-mail: mithuns@am.students.amrita.edu

T. Anjali, Department of Computer Science & Engineering, Amrita School of Engineering, Amrita Vishwa Vidaypeetham, Amritapuri, India. E-mail: anjalit@am.amrita.edu

DOI: 10.37200/IJPR/V24I5/PR201931
Received: 20 Feb 2020 | Revised: 14 Mar 2020 | Accepted: 28 Mar 2020

II. PROPOSED METHOD

The web application consists of 3 phases. The first phase of the app is the question query-entering phase. For this, a user-friendly search box is provided which then based on text-analysis, directs the user to a recommended mentor. The second phase of the application is the Authentication module. OAuth2.0 authentication procedure will be used in order to authenticate the user. The user needs to enter a mobile number and an OTP that is sent to their mobile phone via the provided mobile number. The server verifies the user's code and creates authentication details for the user.

The third phase of the application is the interaction between the user and the mentor. For this, a chat user-interface was designed which would enable the user to interact with the mentor in a personal manner. In order to increase the market share, the first two questions that have been asked by the user will be answered free. If the user finds the answers satisfactory and the mentor helpful, the user can subscribe for more features. The user can report any kind of issues related to the mentors or the application as well. The user information will not be disclosed to the mentor so as to ensure that there would not be any issues or complications. The mentor will only be able to see the user's requisite information. At last, the user will be requested to rate their experience.

Mentoring Web Client handles all the UI/UX for each user's session and manages their activities.

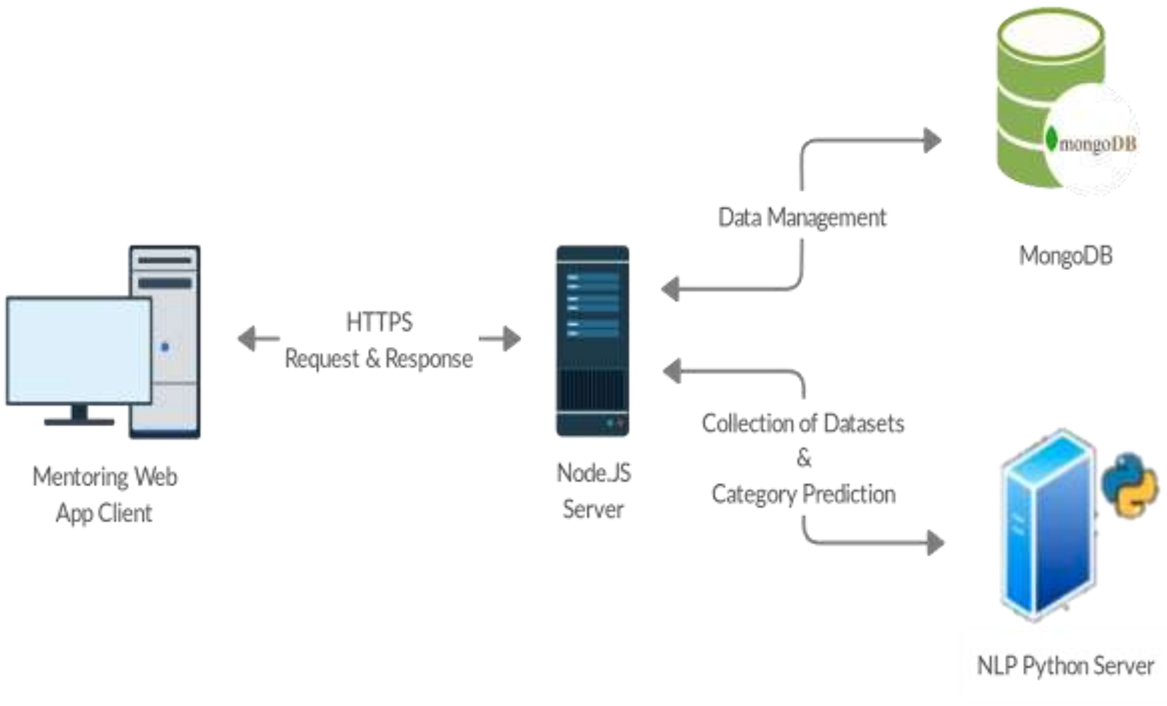


Figure 1: Proposed Method Case Diagram

Node.js server runs as the main backend server. It handles the request of each client with MongoDB to store all the data. Node.js also help the NLP server to learn the category in case it results in error using the API's from other vendors. It deals with User – Mentor Chat Sessions and Authentication.

DOI: 10.37200/IJPR/V24I5/PR201931
Received: 20 Feb 2020 | Revised: 14 Mar 2020 | Accepted: 28 Mar 2020

Node.js also triggers NLP Python server for getting the category of the question by posting question as data. NLP server uses Data Preprocessing and SVM modules to readily give the appropriate category for the question and collects the information for self-learning.

Together with the category as a measure to select the best mentor, Node.js server orders a mentor's list according to the mentor's rating and response time. Mentor rating is a mechanism designed to elevate or deviate mentor rankings based on their past performances.[2][5]

Because the code for Text-Classification was written purely in Python for data integration and training purposes, a separate python server needed to be hosted.

III. Related Works

We followed a research conducted by Laura L. Bierema and Sharan B. Merriam where they had produced the involvement of E-Mentoring playing an important part of mentoring in a person's career path through Electronic Media. They have properly defined each criterion for choosing the right mentor and have studied the situation in which mentors are significant in life. Each and every field requires the right guidance which would help to solve the issues/problems faced in a smarter way. As the above research suggested the ways to have E-Mentoring as a gateway to help the persons in need, we implemented a practical application in guidance with their proposed methodology for choosing the right mentee.[4]

We ran a few rational tests on two of the popular software applications similar to our application. Our observations are recorded in Table 1.

Table 1: Observations based on comparisons with alternative offerings

Alt-Mentoring Web app	*Query Page*	*Recommendation*	*Response time*	*Activities*
Push-far app	No query page, instead offers a to-be manually chosen categorical format	Offers mentor profiles for you to choose yourself.	The response is based on how the customer would work on his/her questionnaire.	Provides only career guidance.
Mentor to go app	No query page but has an extensively long questionnaire session	Recommendation process was done via e-mail	Because the e-mail recommender took approximately 1-2 days for response.	Provides any form of mentorship.

IV. Implementation

As time passes everyone must adapt to technology in the modern technical world. Since mobile websites are more user-friendly than mobile applications, the best way to offer our mentoring help to others is by deploying a mobile compatible web application. The web application contains different modules that make it user-friendly. The modules are listed below.

Authentication Module

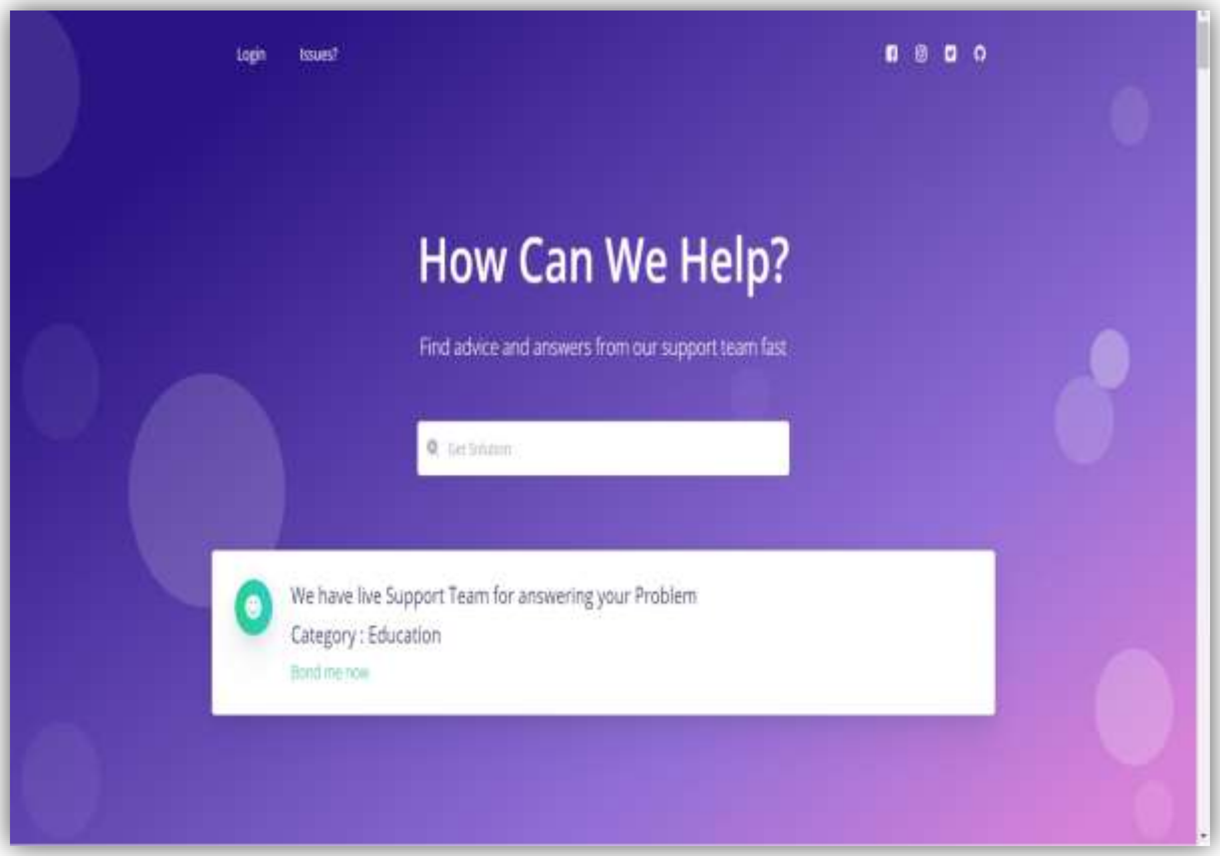
As time passes, everyone must adapt to technology in the modern technical world. Since mobile websites are more user-friendly than mobile applications, the best way to offer our mentoring help to others is by deploying a

DOI: 10.37200/IJPR/V24I5/PR201931
Received: 20 Feb 2020 | Revised: 14 Mar 2020 | Accepted: 28 Mar 2020

mobile compatible web application. The web application contains different modules that make it user-friendly. The modules are listed below.

Subscription Module

Every user can subscribe to receive future updates. The subscribed users will be having the privileges to change the mentor according to the user's wish if the provided mentor is not satisfactory. In addition, the user will be refurbished with the latest news and updates.



How Can We Help?

Find advice and answers from our support team fast

We have live Support Team for answering your Problem

Category : Education

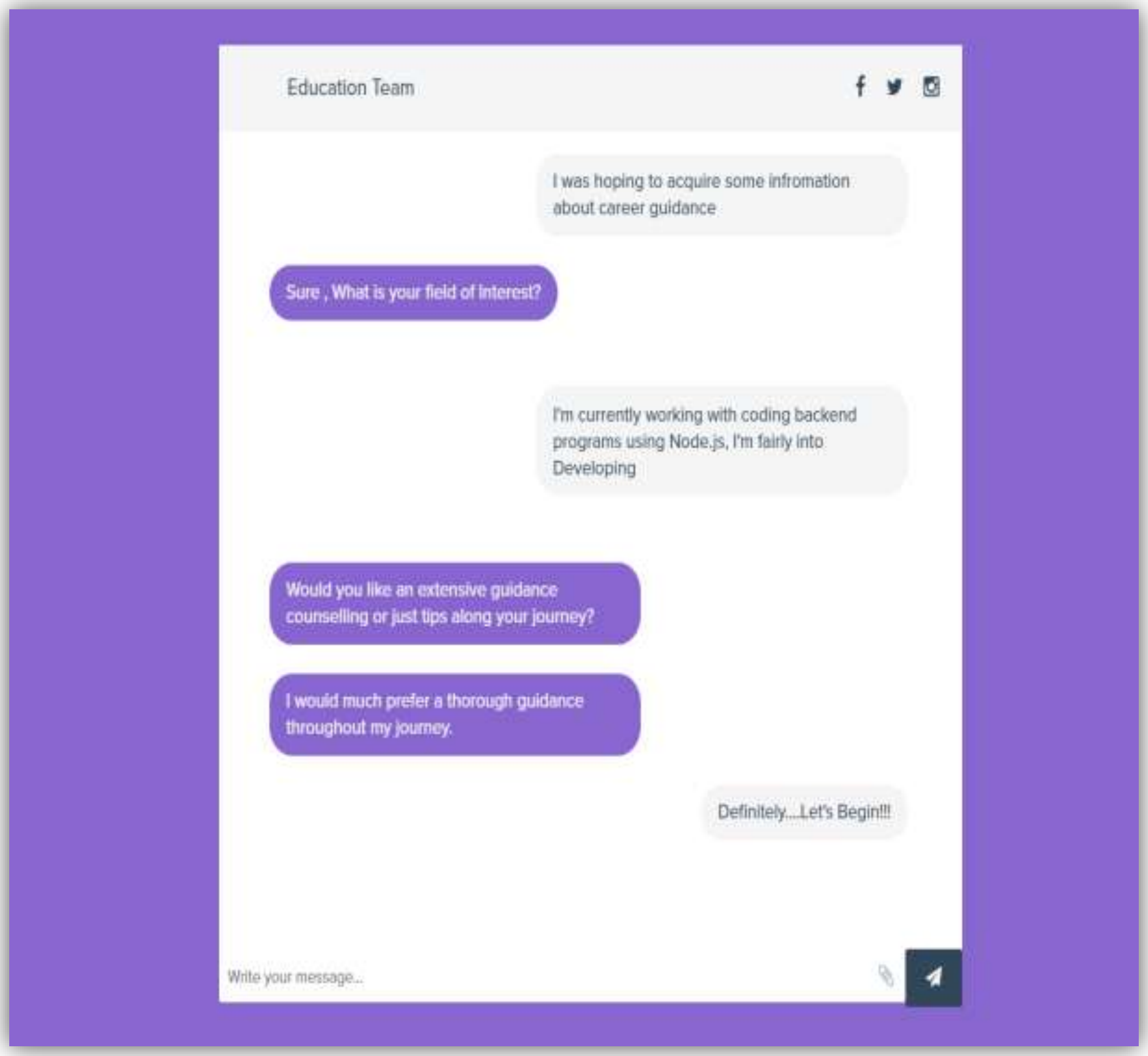
Figure 2: Search Homepage Preview

Search Engine Module

Search engine module act as a middleware between the user and the Search-query Classification Module. It provides the user interface for addressing the issue/problems faced by user to the web application and in return shows the appropriate mentor matched for the user to solve his/her problems at a minimum amount of time.[6]

DOI: 10.37200/IJPR/V24I5/PR201931
Received: 20 Feb 2020 | Revised: 14 Mar 2020 | Accepted: 28 Mar 2020

Mentor Module



Education Team

I was hoping to acquire some infromation about career guidance

Sure , What is your field of interest?

I'm currently working with coding backend programs using Node.js, I'm fairly into Developing

Would you like an extensive guidance counselling or just tips along your journey?

I would much prefer a thorough guidance throughout my journey.

Definitely....Let's Begin!!!

Figure 3: Chat Box Preview

This module is all about the mentors that we provide. This Module helps the users to review the mentors. The Module also contains details about the projects and achievements of our mentors.

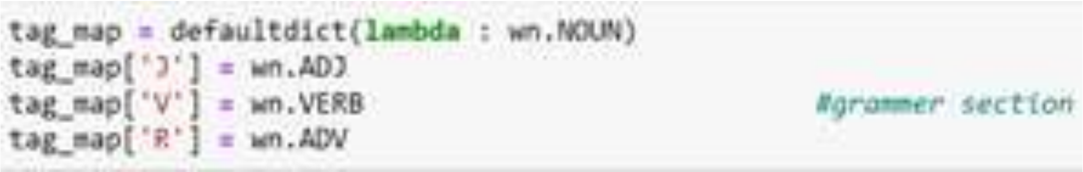
Search-query Classification Module

SVM was used as the Text-Classifying algorithm. Before the collected dataset can be run under the algorithm to generate the most accurate possible prediction of categories so that the result can be fed into the rest of the program to recommend mentors, the text pre-processing stage must be completed. This is done by entering the dataset through multiple stages of breaking down of the features and parameters.

DOI: 10.37200/IJPR/V24I5/PR201931
Received: 20 Feb 2020 | Revised: 14 Mar 2020 | Accepted: 28 Mar 2020

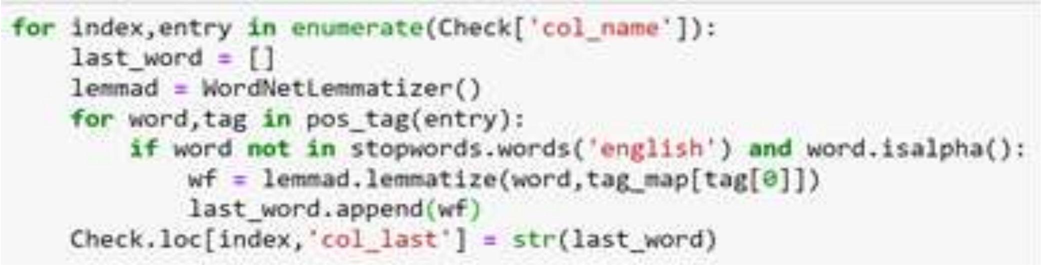
TEXT DATA PRE-PROCESSING STAGE

Each of the separated tokens will be needed to be tagged with the form/type of the word; whether it is an Adjective, Verb or Adverb. So, the forms must be initialized.



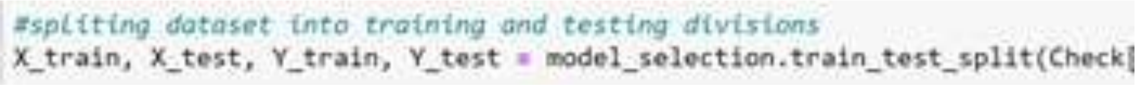
```
tag_map = defaultdict(lambda : wn.NOUN)
tag_map['J'] = wn.ADJ
tag_map['V'] = wn.VERB                                    #grammer section
tag_map['R'] = wn.ADV
```

The further steps include removing blank rows in data, lowering the case of the complete text, then tokenizing the resultant and removing stop words and Non-alpha text. Finally ending it with the Lemmatization process.[8]

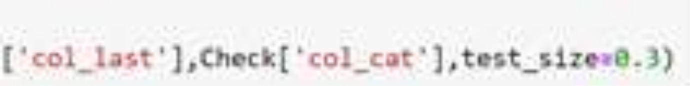


```
for index,entry in enumerate(Check['col_name']):
    last_word = []
    lemmad = WordNetLemmatizer()
    for word,tag in pos_tag(entry):
        if word not in stopwords.words('english') and word.isalpha():
            wf = lemmad.lemmatize(word,tag_map[tag[0]])
            last_word.append(wf)
    Check.loc[index,'col_last'] = str(last_word)
```

Now the set is ready to be split into training and testing sets. The split parameters will be Training data predictors, Training data target, Test data predictors and Test data target.



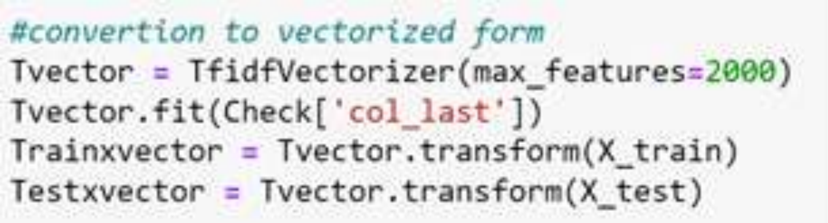
```
#spliting dataset into training and testing divisions
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(Check[
```



```
['col_last'],Check['col_cat'],test_size=0.3)
```

Next encoding process will transform categorical string type data into numerical values.

The next step would be to form numerical feature vectors done using TF_IDF method based on two featured conditions: Summarize Term frequencies and downscale Inverse Document Frequency. This briefly means to identify words that are more of interest to the prediction process.[7]



```
#convertion to vectorized form
Tvector = TfidfVectorizer(max_features=2000)
Tvector.fit(Check['col_last'])
Trainxvector = Tvector.transform(X_train)
Testxvector = Tvector.transform(X_test)
```

The next is the Vectorization stage, which is also designed to limit the feature quantity to a maximum of 2000 features aside from forming a vectorized form of the resultant.

SVM ALGORITHM STAGE

Theory

The theoretical objective of the Support Vector Machine algorithm[1][3] is to find a hyperplane (an N-1 structured space) in an N-dimensional space (in the current case, N is the number of features which is limited to a maximum of 2000). In the SVM algorithm, the purpose is to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss. Moreover, this is the Hinge Loss function for SVM:

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 \quad = 2\lambda w_k$$

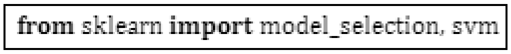
$$\frac{\delta}{\delta w_k} \big(1 - y_i \langle x_i, w \rangle\big)_+ \ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

This is the Gradient conditional function for SVM, which will help reduce nodal values to the most optimal value by forcing the prediction trajectory towards the lowest minimum dip.

$$\boxed{w = w - \alpha \cdot (2\lambda w)}$$

This formula is for the final Gradient update with no Misclassification. This is by which the continuous updation of the prediction value to mark down at the most accurate possible point of prediction based on supplied training data.

The ***sklearn*** package with Python offers built-in SVM functions to work with:

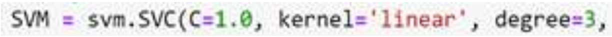


```
from sklearn import model_selection, svm
```

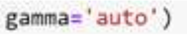
Implementation

SVM algorithm integration is done using in-built functions from various packages.

1. Initializing SVM parameters

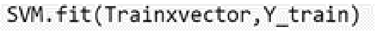


```
SVM = svm.SVC(C=1.0, kernel='linear', degree=3,
```



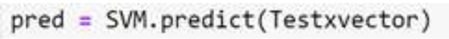
```
gamma='auto')
```

2. Fitting the training set on the classifier:



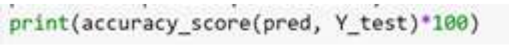
```
SVM.fit(Trainxvector,Y_train)
```

3. Run prediction algorithm:



```
pred = SVM.predict(Testxvector)
```

4. Accuracy testing:



```
print(accuracy_score(pred, Y_test)*100)
```

V. Conclusion

We have majorly focused on providing an easy-to-use interface also concerned with providing security and accuracy when dealing with communication. We have avoided long questionnaire sections compared to the competing applications using text classifying algorithms implemented into our search-query module. This will reduce response time and mentors will be chosen at a faster pace. We have also implemented language inclusion for our foreign-language users. Security will be maintained by providing chat-user interface between the mentor and user and all the information will be provided by the users themselves. Unlike the competition which provides communication platforms that has threads of messages that are exposed to the whole community including all the mentors and users on the platform, we provide closed down personal chat spaces that aims to avoid restrictions during communication.

Acknowledgement

We would like to thank our Chancellor Mata Amritanandamayi for giving us such great opportunity to chase our dream to make this project a reality. Also, would like to thank the IEEE for holding this conference and giving us a platform to offer our work to a wider audience.

References

[1] V. Muralidharan, V. Sugumaran, N.R. Sakthivel "Wavelet decomposition and support vector machine for fault diagnosis of monoblock centrifugal pump, " *International Journal of Data Analysis Techniques and Strategies*, Volume 3, Number 2, p.159-177, 25 April 2011

[2] Sreekanth Narendran, Preeja Pradeep and Maneesha Vinodini Ramesh "An Internet of Things (IoT) based sustainable water management, " *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, 19-22 Oct 2017

[3] K. R. Kavitha ; Ajith Gopinath ; Midhun Gopi, "Applying improved svm classifier for leukemia cancer classification using FCBF", 2017 *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 13-16 Sept. 2017

[4] Laura L. Bierema & Sharan B. Merriam , "E-mentoring: Using Computer Mediated Communication to Enhance the Mentoring Process, " *Innovative Higher Education volume 26, pages211–227(2002)*, March 2002

[5] Stefan Tilkov, Steve Vinoski "Node.js: Using JavaScript to Build High-Performance Network Programs" *IEEE Internet Computing*, Volume: 14 , Issue: 6 , Nov.-Dec. 2010

[6] Nilesh Jain, Priyanka Mangal and Deepak Mehta , "AngularJS: A Modern MVC Framework in JavaScript " *Journal of Global Research in Computer Science*, Volume 5, No. 12, December 2014

[7] Nemati Azize,Karbasi Sheila, "A new approach to improve the accuracy of THE tf_idf ranking algorithm in text Retrieval, " *University of Science and Culture* 2018, Volume 4

[8] Joël Plisson, Nada Lavrac, Dunja Mladenic, "A rule based approach to word lemmatization", *Proceedings of the 7th International Multiconference Information Society, IS-2004, Institut Jozef Stefan, Ljubljana,* 2004.