Group Member Match Based on Data Learning

Geng Song, Zhou Shen, Suli Hu
Department of Electrical and Computer Engineering, Boston University

Introduction

The objective of this project is to design a course-recommendation web system using Django for students in Boston University to both select courses and make friends more easily. Assuming we have bunch of users who want to find proper courses in Department of Electrical and Computer Engineering in Boston University, our web questionnaire system can collect the information of their academic performance and interests, then the back-end will classify each student into an appropriate community based on data learning algorithm.

Back-End System

For the recommendation system, we utilize the k-nearest neighbors algorithm to classify users into different predefined groups based on their answers of the questionnaire. The K-nearest neighbor classification performance can often be significantly improved through (supervised) metric learning. Therefore, the recommendation system put in a ten-element array which contains the user's academic performance and interests, then return a label.

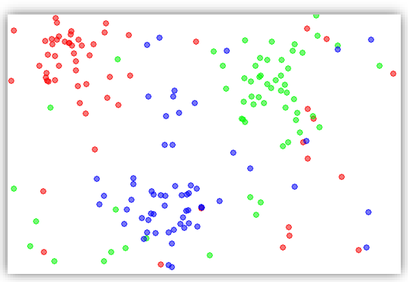


Fig1. Data Set

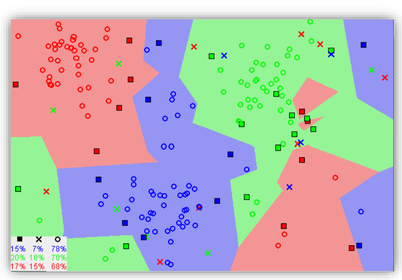


Fig2. KNN Classification Map

In the classification part, we pre-defined nine groups according to the web page of master programs in Department of Electrical and Computer Engineering. Then for each branch, we build three different level community.

- Robotics (Entry Level, Medium Level, High Level)
- Data Analytics (Entry Level, Medium Level, High Level)
- Cybersecurity (Entry Level, Medium Level, High Level)

```python
def knn_classify(inX, dataSet, labels, k):
    dataSetSize = dataSet.shape[0]
    diffMat = np.tile(inX, (dataSetSize, 1)) - dataSet
    sqDiffMat = diffMat**2
    sqDistances = sqDiffMat.sum(axis=1)
    distances = sqDistances**0.5
    sortedDistIndices = distances.argsort()
    classCount = {}
    for i in range(k):
        voteIlabel = labels[sortedDistIndices[i]]
        classCount[voteIlabel] = classCount.get(voteIlabel,0) + 1
    sortedClassCount = sorted(classCount.items(),key=operator.itemgetter(1),reverse=True)
    return sortedClassCount[0][0]
```

Fig3. Code of KNN Algorithm

Front-End system

Questionnaire Web System

we added several academic questions for better course recommendation, also for each question, there will be different response on it; at the end of user finishing questions, the system will show a webpage for loading, at the meantime, our algorithm will match best course for user based on his/her choices, when the algorithm finishes its process, our page will show the result and have a link for user to community.

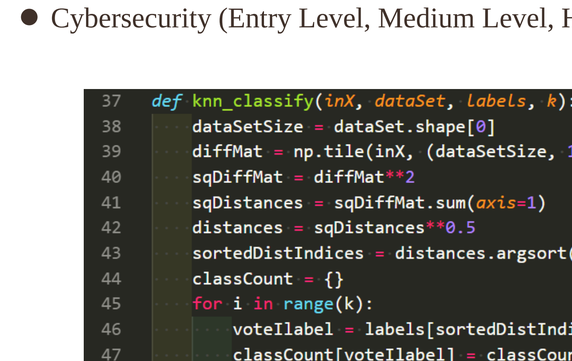


Fig4. Questionaire1

Fig5. Questionaire2

Register/Log-in System

The user register/log-in system aims to provide safety for our users. New users can register for our system with an BU email, then registered users can post their thoughts in communities. It can be generally separated as User/Blog/Media Directories. The Blog Directory contains the templates of the main pages.

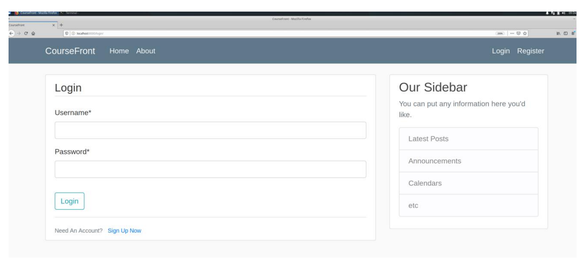


Fig6. Log-in Interface

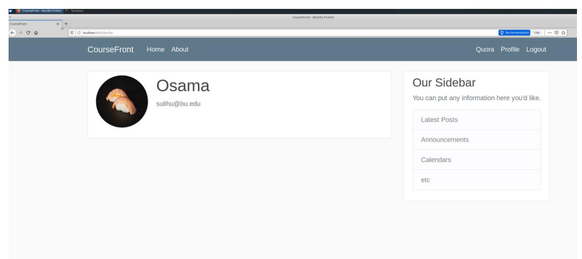


Fig7. Register Interface

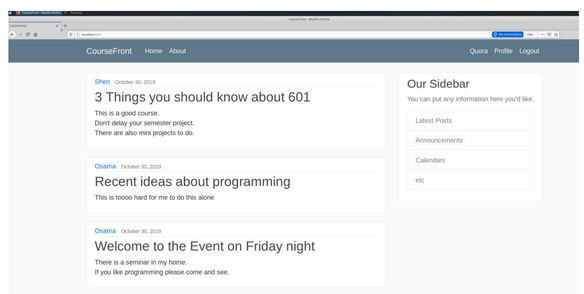
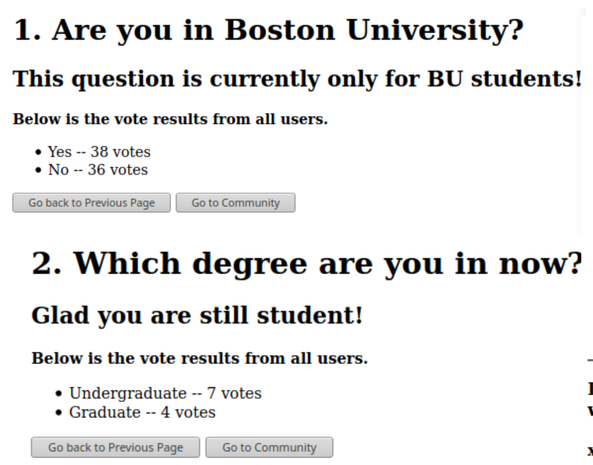
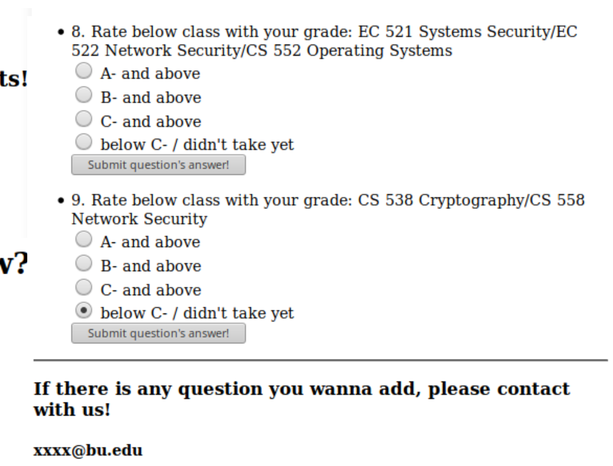


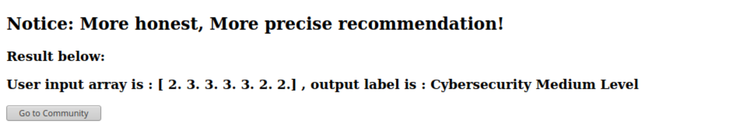
Fig8. Community Post

Result

➢ Here is the main process when users register in our website.







Future Work

For further implementation, our team will focus more on the recommendation of users. Users may want to find classmates or even friends in same interesting course in the community, so we will improve the questionnaire to collect more personal information of users. Private chat will also be provided in our community system, so users can chat through our community system instead of adding social media friends which protect their privacy if they want. We will also open our algorithm as open API for people who are interested in matching and recommendation algorithm