

# UIUC Events Recommender System

Repo: [GitHub](#)

## 1. An overview of the function of the code (i.e., what it does and what it can be used for).

This whole project is based on python 3.6 version. We use the most popular and stable framework: MVC, model-view-controller to achieve the whole data interaction which is from back-end to front-end. And the whole project is including three parts:

The first is Django part, used to construct the website which will output the customers' information, like the users' nickname, email and the interested tags. The customized data will be stored in the "subscribe.csv" file.

The second part is text classification part. This part first crawls the events from <http://illinois.edu/resources/calendars.html>. Then it uses a trained KNN text classification model to assign each of the crawled events with a tag.

The third part is integration part. This part uses a crawler to update the events information once a week. The updated events information will be classified with a new tag. Then our systems will send events email to the users according to the new tag classification.

## 2. Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement.

### 1) Sign in interface

Please sign in

Nickname

Email

Sign in

Users can set their own nickname, and input their email for receiving the recommending events later.

2) Click and redirect to choose tags which users are interested in.

- ☒ Ceremony/Service
  - ☒ Conference/Workshop
  - ☒ Exhibition
  - ☒ Festival/Celebration
  - ☐ Health/Fitness
  - ☐ Informational
  - ☐ Lecture
  - ☐ Meeting
  - ☐ Performance
  - ☐ Reception/Open House
  - ☐ Seminar/Symposium
  - ☐ Social/Informal Event
- Save

Here we set twelve types events for users choosing: Ceremony/Service, Conference/Workshop, Exhibition, Festival/Celebration, Health/Fitness, Informational, Lecture, Meeting, Performance, Reception/Open House, Seminar/Symposium, and Social/Informal Event.

- 3) Send recommending events to users according to the tags selected.

Our system will automatically update the events information by crawling the new events from the relevant website once a week, then we will utilize this new events information and the exist events information to train our classification model. By applying this model, we will acquire new tag information (more accurate than initial classification). At last, we will send the respect events information to the specific users by their subscription.

**jiajunc22@gmail.com**

10:01 PM



Coming Events

To: [chenjiajun0627@yahoo.com](mailto:chenjiajun0627@yahoo.com)

### Coming Events

- Conference/Workshop
  - [Allerton Community Yoga](#)
  - [Compass 2g Workshop - Grade Center Drop-in Clinic](#)
- Exhibition
  - [Sousa and Tsar Nicholas II's Birthday: An Unexpected Tour Adventure](#)
  - [Knowledge at Work: The University of Illinois at 150](#)
  - ["From Russia with Love:" John Garvey's Russian Folk Orchestra](#)
  - [Illinois' Anti-Establishment Soundscapes: Troubled Waters in 1970](#)
  - [Pioneer Agricultural Journalists Exhibit](#)
  - [University Primary School Art Show](#)
  - [KAM Exhibition I 2018 School of Art + Design Bachelor of Fine Arts Exhibition](#)
  - [KAM Exhibition I Basel Abbas + Ruanne Abou-Rahme: And yet my mask is powerful](#)
- Festival/Celebration

### 3. Documentation of the usage of the software including either documentation of usages of APIs or detailed instructions on how to install and run a software, whichever is applicable.

- 1) <sup>1</sup>Get the latest official version of the Django:

Option 1:

\$ pip install Django==2.0.5

Option 2:

\$ git clone https://github.com/django/django.git

- 2) BeautifulSoup library install

\$ pip install BeautifulSoup

- 3) Implemented email sending function by setting email parameters in settings.py

### 4. Brief description of contribution of each team member in case of a multi-person team.

1. Xiaokai Cui

Used BeautifulSoup to crawl all the events from UIUC calendar. Trained a text classification model using KNN method to tag the events for future recommendation.

2. Jiajun Chen

Used Django-Crontab to implement the automatically email sending function. Sorted out the data processing progress and connected the crawler with KNN.

3. Lan Li

Using the Django to construct the register interface and subscriber interface, and store users' information into sqlite database. Then output the data file for later use.

[1] <https://www.djangoproject.com/download/>