# Task 7 Application System Development

## Overview

The impact of any data mining research will eventually have to be realized via an application system. The goal of this last task of the Capstone Project is to leverage what you have learned and developed in the previous tasks to build a small-scale application system that would allow the envisioned end users (i.e., people who will benefit from the results that are generated by your data mining algorithms) to upload a new data set and apply at least one algorithm that you developed or experimented with to mine the uploaded data set using a Web interface. Your application system not only can be used as a demo system to show your accomplishment in this Capstone course, but can also be used potentially by real users. Whenever possible, we will attempt to integrate multiple application systems developed by you into an even more useful (and also more sophisticated) application system for mining review data.

## Abstract

1) A title describing the function of your system
    a. Restaurant recommendation for "Indian Cuisine" based on 2 different approach of search
        i. Average rating based
        ii. Sentiment Based
2) A description of the envisioned application scenario and the envisioned users
    a. Idea of building this application around ways to search differently for a dish name for a cuisine.
3) The design of the interface of your application.
    a. Explained in section above under 'Tool Tip'
4) URL for the application developed
    a. http://161.35.205.51:8080/
    b. Have chosen DigitalOcean for deployment – Ubuntu/4CPU/8GB/160SSD

## Novelty of the application

Based on my research there are many applications available which give users to search for restaurants recommendation. What is unique here is the algorithm and simplicity that I have strived to design. Now simply put, there could be so many things which can make it better as explained in "Future work" section, but this is just a polished slim version of a single task.

## Installation/configuration

- **pip3 install Django**
  - https://docs.djangoproject.com/en/1.8/intro/install/
- check Django installation
  - python -c "import django; print(django.get_version())"
  - should print version ex: 3.0.8
- Use this link to learn how to create a project
  - https://www.tutorialspoint.com/django/django_creating_project.htm
  - https://docs.djangoproject.com/en/1.8/intro/tutorial01/
- cd on terminal to the directory created and "python manage.py help", which should option that you can play around to explore further what you can do with the manage.py.
- To run server, execute "python manage.py runserver"
- Check for -> *Starting development server at http://127.0.0.1:8000/*
- Create an application
  - python manage.py startapp ui
- Create static and template directory under "ui"
- Add "urls.py" for redirect to views for navigation rendering
- Create base.html under templates/ui
- You are all set!! Go on creating views.py, base.html and home.html (follow GitHub for further implementation details)

## Deployment

- Create droplet on AWS or Digital ocean (provision the server)
- ssh root@IP (connect from local machine)

```
sudo apt-get update

sudo apt-get -y install python3-pip
```

- mkdir project
- cd project
- git clone https://github.com/atif-github-venture/data-mining-capstone.git
  - provide credential if needed.
  - In case you have to fetch:
    - git fetch origin
    - git pull origin

- To remove a directory forcefully

    - rm -rf dirname
- cd /data-mining-capstone/Task7
- pip3 intsall -r requirements.txt
- Run the app
    - python3 manage.py runserver 0.0.0.0:8080
- Note: Update the allowed host in the setting.py for the IP you hosting on, before cloning GitHub repo.

## To expose the User interface to outside world use Ngnix

Run the below command on the serve.

- sudo apt-get install nginx
- sudo vi /etc/nginx/sites-available/default
- And replace the content with below

/etc/nginx/sites-available/default

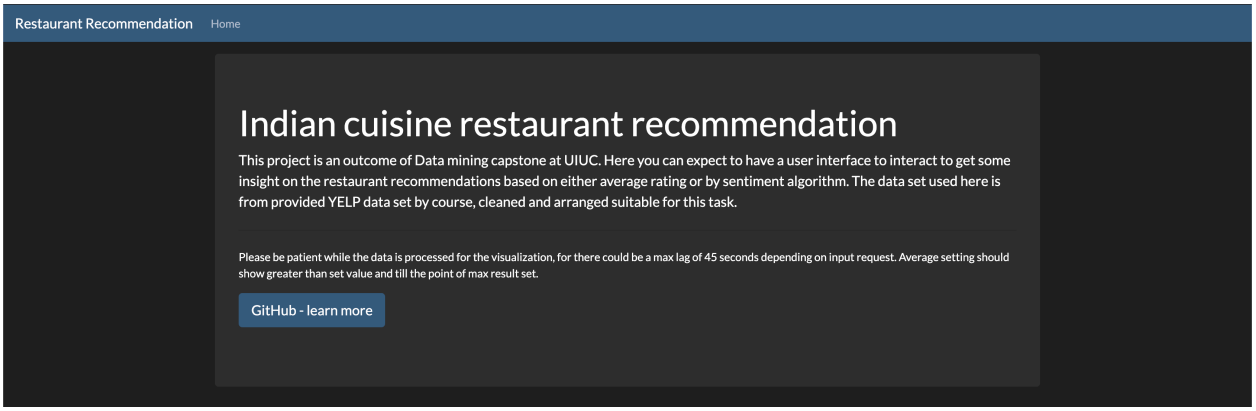
```
server {
    listen 80;

    server_name example.com;

    location / {
        proxy_pass http://APP_PRIVATE_IP_ADDRESS:8080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

If still having issue with port access, try taking reference of below commands:

- sudo ufw allow 8080
- sudo ufw status
- Note: If the firewall is inactive, the following commands will allow OpenSSH and enable the firewall:
- sudo ufw allow OpenSSH
- sudo ufw enable

## Tool tip

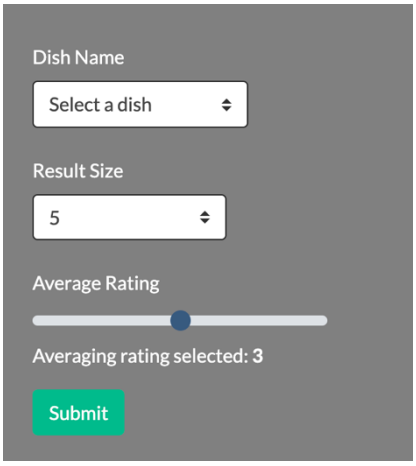
## Landing page




Click on **GitHub - learn more**, to navigate to GitHub repository for this implementation.

## Section below for selection of query type


- On selection of either of radio button, a view will expand further for user input.
- Under either of selection user can provide their inputs as shown below.

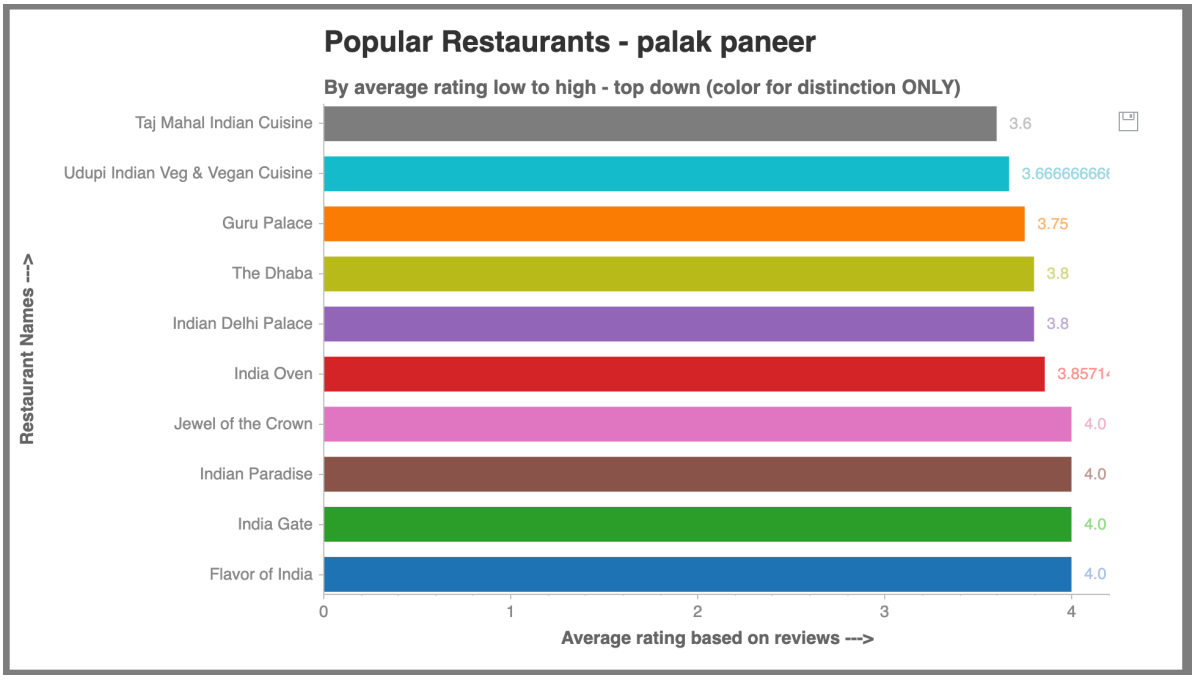



- User can select the desired input from the list of dish names and provide how they would like to filter.
  - For example, result set of 10 with average rating set to 3.6, will provide 10 or lesser result set based on data processing for average rating greater than or equal to 3.6.
- To process the user has to click "Submit" button.
- This should take under 45s in extreme cases based on my learning, otherwise pretty quick.

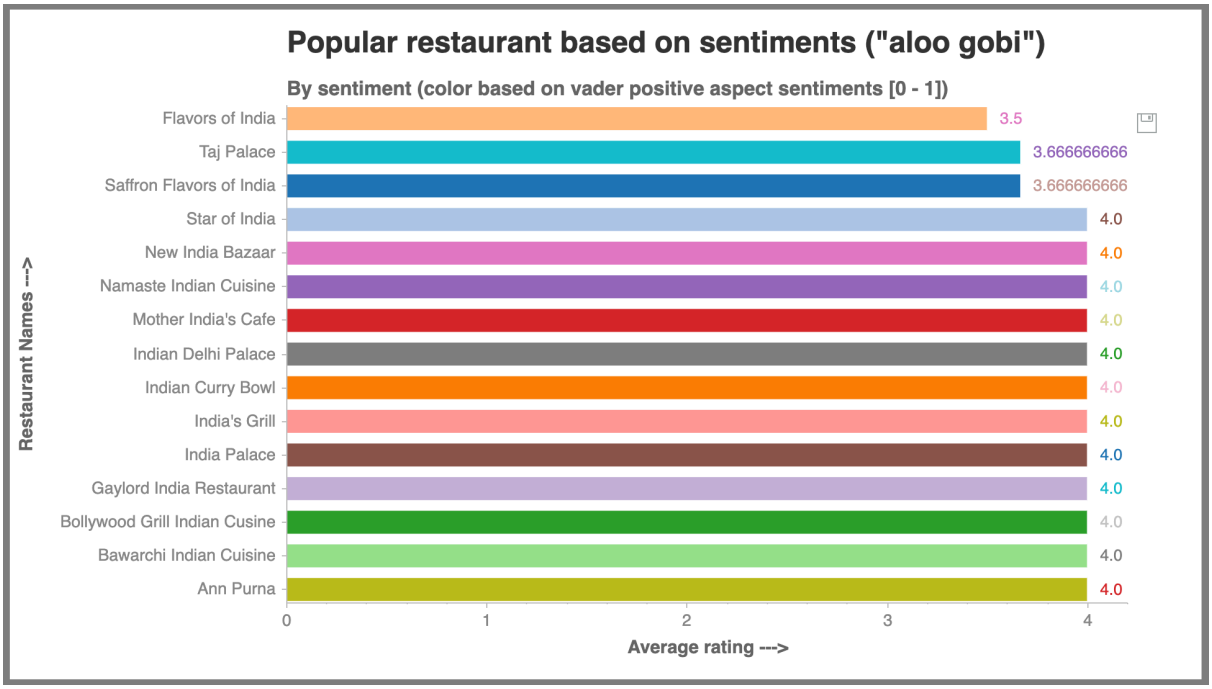
## Error handling

Appropriate error handling will display in case Dish name is not selected.


## Sample result for search by average rating



## Sample result of search based on sentiment analysis




## Additional information based on other factors of sentiments

Additional Info: Based on Vader Sentiment Analysis

| Restaurant | Negative | Neutral | Positive | Compound |
|---|---|---|---|---|
| Flavors of India | 0.011000000000000001 | 0.7829999999999999 | 0.205 | 0.9946 |
| Taj Palace | 0.079 | 0.738 | 0.183 | 0.996 |
| Saffron Flavors of India | 0.081 | 0.833 | 0.086 | -0.3472 |
| Gaylord India Restaurant | 0.0 | 0.713 | 0.287 | 0.9945 |
| Bollywood Grill Indian Cusine | 0.013999999999999999 | 0.868 | 0.11800000000000001 | 0.99 |
| Mother India's Cafe | 0.01 | 0.865 | 0.125 | 0.972 |
| Star of India | 0.035 | 0.767 | 0.19699999999999998 | 0.998 |
| Indian Curry Bowl | 0.0 | 0.894 | 0.106 | 0.9699 |
| India Palace | 0.038 | 0.8109999999999999 | 0.151 | 0.9988 |
| Ann Purna | 0.046 | 0.785 | 0.17 | 0.9981 |
| India's Grill | 0.067 | 0.657 | 0.276 | 0.9908 |
| Bawarchi Indian Cuisine | 0.023 | 0.738 | 0.24 | 0.9988 |
| New India Bazaar | 0.021 | 0.8220000000000001 | 0.157 | 0.9971 |

## Chosen task for the development of application

### Recap of Task 5

*In this task, your goal is to recommend good restaurants to those who would like to try one or more dishes in a cuisine. Given a particular dish, the general idea of solving this problem is to assess whether a restaurant is good for this dish based on whether the reviews of a candidate restaurant have included many positive (and very few negative) comments about the dish. You may choose a target dish or a set of target dishes from the list of "popular dishes" you generated from Task 4 or, otherwise, choose any dishes that have been mentioned many times in the review data (the more reviews you have for a dish, the more basis you will have for ranking restaurants).*

Here I have explained what work I chose and what benefit it fetches and how it has been implemented along with future extensions of this application.

- I have chosen to work on the idea of Task 5.
- I kind of liked the idea of build an application around ways to search differently for a dish name for a cuisine.
- There could be various ways of representing my work and study of Task 5, but I wanted to keep it light and simplified for subtle and crisp presentation.
- This is useful as explained if the user is wanting to know which restaurant to try based on 2 different factors for "Indian" cuisine.
- I have not mentioned of the algorithm as it would be duplicate
    - Refer to my Task 5 implementation or the GitHub repo to check out how I have tweaked to suit to build this application.

### Future work

- Add more cuisines
- Add views and filters based on location/city
- Add geocode enabled view of restaurant's location
- Improvise on algorithm speed using distributed system implementation

Reference:
https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-14-04
GitHub - https://github.com/atif-github-venture/data-mining-capstone