Summer Fellowship Report

On

Mapper

Submitted by

Pritam Kumar Sahoo

Under the guidance of

Prof. Prabhu Ramachandran
Deparatment of Aerospace Engineering
IIT Bombay

July 19, 2018

Acknowledgment

I wish to express our profound gratitude to our internship guide prof. Prabhu Ramchandran, Deparatment of Aerospace Engineering, IIT Bombay for his constant support and supervision thorughout the internship.

I am highly indebted to my project mentor Mr. Akshen Doke and my project head Mr. Mahesh Gudi for their continous support, supervision motivation and guidance throughout the tenure of my project in spite of their hectic schedule who truly remained driving spirit in my project and their experience gave me the light in handling this project and helped me in clarifying the abstract concepts, requiring knowledge and perception, handling critical situations and in understanding the objective of my work.

Contents

Chapter 1

Introduction

Mapper is an Web Application for better visualization and processing of user's data. User will be asked to upload their data in .csv format. Mapper will clean the data, modify them, and check and handle errors. Finally, user can take a look at their data statistics plotted in State-wise INDIA Map and 3D Pie-Chart. User can edit their data too.

Chapter 2

Data and Requirements

2.1 Data :: Components

We have mentioned about uploading data. So, what the so called 'data' will contain is :-

- Name of College
- Corresponding State
- Corresponding District
- Address
- International Dial Code, Email-id, etc.

Out of the components mentioned above, first four are of more importance and including these columns is compulsory; as our main focus is to work on them.

2.2 Requirements for this project

This proect has been built upon :-

- Python
- Django

The following libraries have been used for many purpose :-

- Numpy
- Pandas
- Fuzzywuzzy

For developing the UI, we have used :-

- Bootstrap4

2.3 What we have

- College.csv

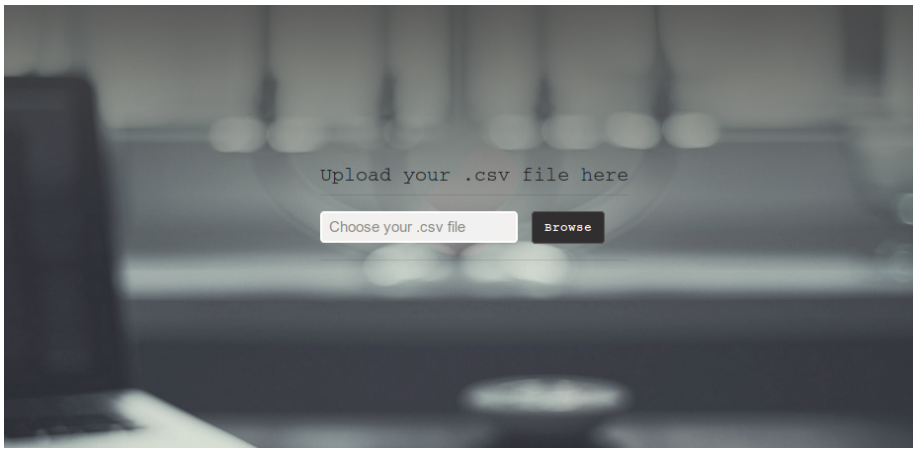
- College.csv contains a good number of clean data about various colleges in INDIA with their corresponding state, district, address, email-id, International Dial Code, etc. It is our refernce data. We all perform all the cleaning operaiton and modification based upon the data this file contains. New clean and modified data which the user will upload; will also be added to this data.

- Bisedes this file, we also have a database which will contain same set of data the 'college.csv' file has. But, when we perform data-fetch operation, we do not go to database and grab the data; we fetch the data from the file instead. After all the work is done, the new set of data which the user uploads will be appended to this database. The database is being maintained because, if, by mistake, we loose the file, we will still have access to the data we created.

Chapter 3

Upload Data

- Home page of our web appliaction will contain a browse button to let users upload their file.



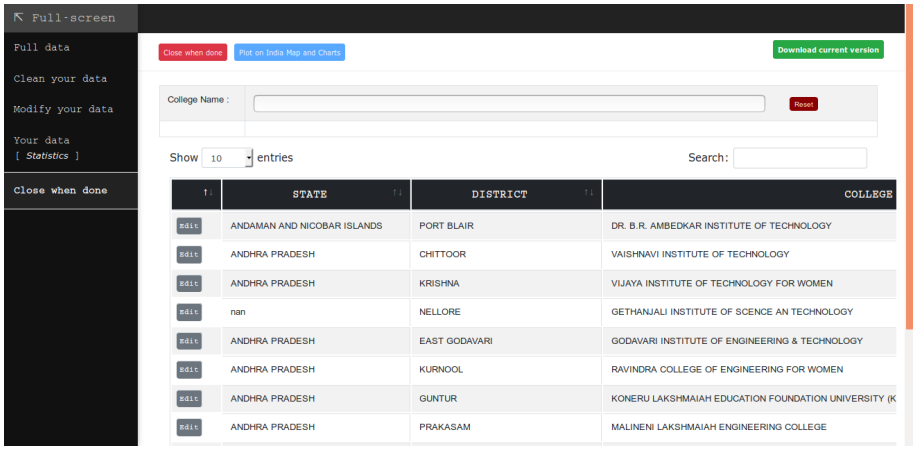
- Only '.csv' files are allowed.
- You can upload only when the file will be selected in correct format.

Chapter 4

Data Processing

4.1 Displaying the data

- After clicking on the 'Upload' button, this page will be shown where user can see their full data in nice format.



- User can edit their data too.
- After completing all work, user can download their clean data in excel format.

4.2 Data cleaning

PANDAS :- We have performed data-processing operations using pandas library. It is an open-source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tool available in Python. We have taken help of pandas Data-Frame object for handling users' data. **Cleaning operation has been done based on two main issues :-**

- **Null Entry (NaN)**
  - If the 'college name' of any record is found to be null; then we can not further process the record anymore. Then it will be treated as wrong entry, and the whole row will be deleted.

```
import pandas as pd
try:
    file = csvfile.objects.get(id=pk)
except csvfile.DoesNotExist:
    raise Http404

'''Reading file contents as a dataframe object'''
data = pd.read_csv(filename)

'''Dropping values for null (NaN) entries for COLLEGES'''
data = data.dropna(subset=['COLLEGE NAME'])
```

- **Duplicate Entry**
  - If more than two rows are found to be holding exactly same set of records, then the first row will be kept and the rest will be dropped.

```
# Fetching all the columns of the dataframe
cols = data.columns.tolist()[1:]

# Dropping duplicate rows, while keeping the first one
data = data.drop_duplicates(cols, keep='first')

#Writing the data back to file
data.to_csv(str(file), index=False)
```

4.3 Modify Data :: Error Handling

FuzzyWuzzy :- We are checking errors for State, District, and College data only with respect to the clean database(college.csv) we already have. For that we need to perform string matching. We have used 'fuzzywuzzy' library, which is run by fuzzy string matching technique. It uses 'Levenshtein Distance' to calculate the differences between sequences in a simple-to-use package. Some of its usage :

```
>>> from fuzzywuzzy import fuzz
>>> from fuzzywuzzy import process

>>> fuzz.ratio("this is a test", "this is a test!")
    97

>>> fuzz.partial_ratio("this is a test", "this is a test!")
    100

>>> choices = ["Atlanta Falcons", "New York Jets", "New York Giants",
    "Dallas Cowboys"]
>>> process.extract("new York jets", choices, limit=2)
    [('New York Jets', 100), ('New York Giants', 78)]
```

Error checking and handling has been done in two ways :-

- **Auto-correction**
  - If any state or district is found to be misspellt; we will auto-correct it for the user. For that, we shall make a list of states or districts; which will be our choices(mentioned above in fuzzy example), and then we'll find the percantage match between the individual choices and the mispellt word; and assign the name with highest percentage to it.

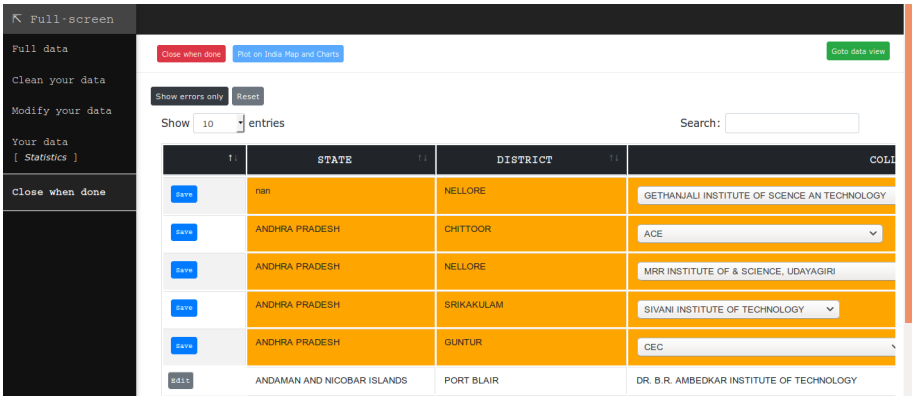
    ```
    >>> match = process.extractOne(misspelt word, choices,
        scorer=fuzz.WRatio)
    ```

  - If State or District name is found to be quite different (e.g.-No such state exists); then we will find the corresponding college in the main database, grab the name of the state, and assign it.
- **Suggestion**
  - Auto-correction for State and District is quite simple and can be easily done, as there are only 29 states in INDIA. But, if there is something wrong with college information, or someone writes a college name in abbreviated form(i.e.-Indian Institute of Technology - IIT or I.I.T.); then it is very hard to auto-correct it, because there may exist many colleges with same abbreviated form. Then we will give some siggestions to user matching with it. If user selects one of the suggestions, then it is nice; otherwise, we will treat the whole record as a completely new one, and append to our existing database.

  The following code performs abbreviation string matching -

```
res = []
for i in range(len(college_list)):
    temp = ''
    for w in str(college_list[i]).split():
        if w != 'AND' and w != 'OF' and w != 'THE':
            temp = temp + w[0].upper()
    temp = temp[:summ+1]
    ratio = fuzz.ratio(name, temp.upper()) # For abbreviation matching
    res.append((college_list[i], ratio))
res = sorted(res, key=lambda x: x[1], reverse=True)
```

  - Here how the 'Modify Data' page looks like :-


Full-screen
Full data
Clean your data
Modify your data
Your data
[Statistics]
Close when done
Close when done
Plot on India Map and Charts
Goto data view
Show errors only
Reset
Show 10 entries
Search:
STATE
DISTRICT
COLL
Save
nan
NELLORE
GETHANJALI INSTITUTE OF SCENCE AN TECHNOLOGY
Save
ANDHRA PRADESH
CHITTOOR
ACE
Save
ANDHRA PRADESH
NELLORE
MRR INSTITUTE OF & SCIENCE, UDAYAGIRI
Save
ANDHRA PRADESH
SRIKAKULAM
SIVANI INSTITUTE OF TECHNOLOGY
Save
ANDHRA PRADESH
GUNTUR
CEC
Edit
ANDAMAN AND NICOBAR ISLANDS
PORT BLAIR
DR. B.R. AMBEDKAR INSTITUTE OF TECHNOLOGY

Chapter 5

Plot the Data

5.1 India Map(State-wise)

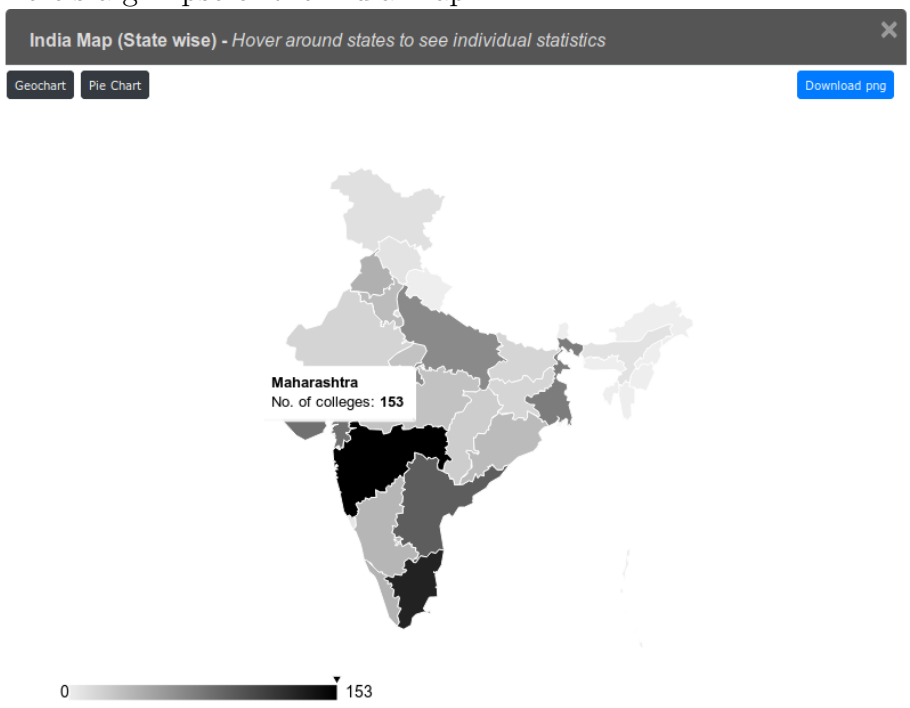
We will plot the state-wise data statistics. For that, we will take advantage of Google Geochart, which gives us a nice Heat map of INDIA, where we can see the statistics of individual states, when hovering on it. It can be done by the following piece of code.

```
google.load('visualization', '1', {'packages': ['geochart', 'corechart']});
google.setOnLoadCallback(drawVisualization);

function drawVisualization() {
  var data = google.visualization.arrayToDataTable(
    {{ state_list | safe }}
  );

  var opts = {
    region: 'IN',
    domain: 'IN',
    displayMode: 'regions',
    resolution: 'provinces',
    datalessRegionColor: 'transparent',
    width: 750,
    height: 540,
    colorAxis: {colors: ['#eeeeee', 'black']},
    backgroundColor: 'white',
    defaultColor: '#f5f5f5',
  };

  var chart_div = document.getElementById('visualization');
  var geochart = new google.visualization.GeoChart(chart_div);

  geochart.draw(data, opts);
```

Particular javascript and CSS files should be included. Now, the data ('state-list') mentioned in the above piece of code looks like :-

```
state_list = [
        ['State Code', 'State', 'No. of colleges'],
        ["IN-AP", "Andhra Pradesh", 121],
        ["IN-AR", "Arunachal Pradesh", 31],
        ["IN-AS", "Assam", 24],
        ["IN-BR", "Bihar", 36],
        ["IN-CT", "Chhattisgarh", 100],
        ["IN-GA", "Goa", 10],
        .
        [...]
        .
      ]
```

Here's a glimpse of the India Map :-

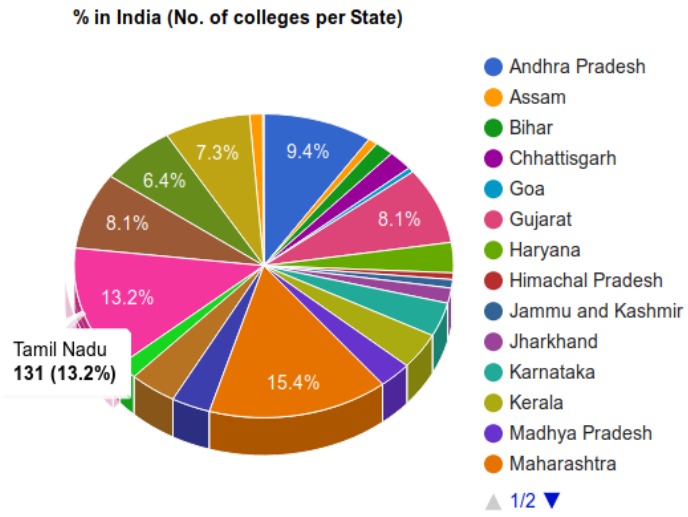


5.2 3D Pie-Chart

Google provides us with Pie-chart too. We can construct the chart with same set of data ('state-list'), but different options (opts).

```
chart_divi = document.getElementById('visualization');
piechart = new google.visualization.PieChart(chart_divi);
piechart.draw(google.visualization.arrayToDataTable({{ pie_list | safe }}),
  {title: '% in India (No. of colleges per State)' , is3D: true});
```

It displays the percentage stats per state.



Reference

- For pandas tutorial - https://pandas.pydata.org/pandas-docs/stable/10min.html
- For more about fuzzywuzzy - https://github.com/seatgeek/fuzzywuzzy
- More about Google geocharts - https://developers.google.com/chart/ interactive/docs/gallery/geochart