**Project Summary and Conclusion**

**Note: Link to all our HTML and Text files is as follows:-**

[**https://drive.google.com/drive/folders/11HynY9phRhAaYQP3W-uWX\_x23nVrW7cW?usp=sharing**](#_top)

**Step 1**:

Extract name of restaurant, link to its first page and no. of reviews it contains: -

Table generated: **1 link to restaurants.txt** 

|  |  |  |
| --- | --- | --- |
| Name of Restaurant  *(no header)* | Link to restaurant  *(no header)* | No. of reviews  *(no header)* |

Code written: **1 scrapping links to restaurants.py**

**Step 2:**

Extract html pages of each restaurant till the total no. of reviews exceeds 30,000.

Files generated are stored in the folder **restaurantFiles** with name of restaurant within this and pages within that.

Code written: **2 creating rest html files.py**

**Step 3:**



Extract links to first page of users from restaurant pages extracted in step 2. Please note that link to **all users** were extracted within each restaurant until total no. of reviews within them summed up to 30,000. Hence, not all restaurants were used to extract the link to users.

Table generated: **1 link to users.txt**

|  |  |  |  |
| --- | --- | --- | --- |
| Name of User  *(no header)* | Link to User  *(no header)* | No. of reviews that user has written  *(no header)* | No. of reviews to be counted (100 at max)  *(no header)* |

Code written: **1 scrapping links to users.py**

**Step 4:**

Extract html pages of each user till the total no. of reviews exceeds 30,000. All pages of a user reviews are extracted. Files generated are stored in the folder **userFiles** with name of restaurant within this and pages within that.

Code written: **2 creating user html files.py**

**Step 5:**

****

Within each user extracted in step 4, get the restaurant name, check if it exists in file “1 link to restaurants.txt” and if not then go to that restaurant and extract first page of that restaurant and save it in the folder **restaurantFiles/otherFiles**. Also, add the names and total ratings of these restaurants in file “**3 other restaurants.txt**”.

Format of “**3 other restaurants.txt**” file is as follows:-

|  |
| --- |
| Name of Restaurant  *(restName)* |

Also, we extracted rating provided by the user to that restaurant as well as average rating provided by the user to all the restaurants and saved the details to the file “**3 ratings by user.txt**”.

Format of “**3 ratings by user.txt**” is as follows:-

|  |  |  |  |
| --- | --- | --- | --- |
| Name of User  *(userName)* | Name of Restaurant  *(restName)* | Average rating given by to user to any restaurant  *(avgRatingToAllRestByUser)* | Rating given by user to this restaurant  *(ratingToRest)* |

Code written: **3 extract other restaurants.py**

**Step 6:**

Information about restaurant names, user names and ratings are extracted from files **"1 link to restaurants.txt"** and "**3 other restaurants.txt** " as well as from HTML files we extracted in restaurantFiles folder and otherFiles folder within it.

Average restaurant ratings of all the restaurants are extracted from the HTML files and saved in the text file – “**4 avg rest rating.txt**”

Format of this file is as follows:-

|  |  |
| --- | --- |
| Name of Restaurant  (*restName*) | Overall Rating to Restaurant  (*avgRatingToRest*) |

Use files “**3 ratings by user.txt**” and “**4 avg rest rating.txt**” to create a final dataset containing following information: -

Table name is saved with name: **5 final file.txt**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of User  *(userName)* | Name of Restaurant  *(restName)* | Average rating given by to user to any restaurant  *(avgRatingToAllRestByUser)* | Rating given by user to this restaurant  *(ratingToRest)* | Overall Rating to Restaurant  *(avgRatingToRest)* |

**Note:** *ratingToRest* is our target variable

Code written: **4 final dataset creation.py**

**Step 7:**

Read file “4 final file.txt” in a data frame and changing names of restaurants and users to numbers by assigning them random ids, converting ratings greater than 3 to 1, less than 3 to -1 and equal to 3 to 0. Then on splitting the data set into 70-30% (training-testing) sets and applying different predictive models on them to get maximum accuracy with **Random Forest** model of approx **68%**.

Code written: **5 applying model.py**

**Note: We forgot to mention in the voice over that we created final file for three different cuisines – Indian, Japanese and Italian by running the above steps three times. Hence, as a part of pre-processing in Step7 we have clubbed the “5final file <X>.txt” of each cuisine (where <X> is the name of the cuisine.**

--------------------------------------------------------THE END------------------------------------------------------------------