**Recommended System Document**

**Problem**: Given jobs and candidates data with the fields mentioned below,create a recommendation system that categorizes a given set of candidates across the job pool. For training the model, you have to extrapolate the provided sample data.

Job Fields ()

- Designation

- Required Skills

- Minimum Experience in Months

- Maximum Experience in Months

Candidate Data fields ()

- Skills

- Designation

- Industry

- Experience in Months

- Institute

- Degree

Mapping between jobs and candidates ()

- Job Id and corresponding relevant Candidate Ids

**Assumptions**: Dataset should be in unbiased and proper cleaned form.

**Notes**:

1. **Beautifly** Python Library for web scraping.
2. **Pandas** Python Library for data manipulation and analysis.
3. **Sklearn** Python Library for machine learning algorithms.
4. **Openpyxl** Python library for reading and writing Excel 2010 xlsx/xlsm/xltx/xltm files.
5. **GraphLab** Python Library for implement collaborative filtering algorithm or model.

**Approach**:

Steps which I follow while solving the above problem statement.

1. **Gathering Data:-** At this step I searched on Google for require dataset.I found some on Kaggle, github and merge some of there dataset according to my requirement. But I require some more fields like list of degrees, skills, designation which I found at some websites.To parse that website content I used BeautifulSoup Python Library and then save in my dataset files.At that step I collect all my require dataset files.
2. **Data Cleaning/Preparation of Data:-** At this step I cleaned my data like remove unwanted rows, blank rows, columns.
3. **Splitting Data:-** At this step I splitted my whole datasets in two parts of ratio 70:30 one was train and other in test data.I further split 70% train data in two parts of ratio 60:40 one was train and other was validation data.
4. **Choosing a model:-** At this step I choose a model or machine learning algorithm for my problem. Collaborative filtering could work well for this problem..
5. **Train our model:-** At this step I train my chosen model on training data using GraphLab which is a python library to implement the machine learning algorithm.
6. **Evaluate the model:-** At this step I evaluate or measure model performance of my trained model on the testing data.
7. **Prediction the model:-** At this step I give the users data and predict the recommended jobs for the particular candidate using my trained model.

**Files:**

Information of files.

1. **Candidate.xlsx:-** It contains the all information of candidates descriptions with belows columns:-
   1. CV\_Id
   2. Skills
   3. Designation
   4. Industry
   5. Experience (in months)
   6. Insititude
   7. Degree
2. **Job.xlsx:-** It contains the all information of jobs descriptions with below columns:-
   1. Job\_Id
   2. Designation
   3. Required Skills
   4. Min Exper (in months)
   5. Max Exper (in months)
3. **Candidate\_Train.csv:-** It contains the all information of candidate descriptions with below columns:-
   1. CV\_Id
   2. Skills
   3. Designation
   4. Industry
   5. Experience (in months)
   6. Insititude
   7. Degree
4. **job\_mapping\_train.csv:-** It contains the all mapping of job\_id and and cv\_id with 70% data which I used in training our data.
5. **job\_mapping\_test.csv:-** It contains the all mapping of job\_id and and cv\_id with 30% data which I used in testing our data.
6. **Assignment\_Python\_Code:-** It contains the python class named **JobRecommendedSysetm** code which have the following functions:-
   1. **crawl\_institude\_names**:- This function crawl the institute names and save in the list from the given url using Python BeautifulSoup library.
   2. **crawl\_degree\_names**:- This function crawl the degree names and save in the list from the given url using Python BeautifulSoup library.
   3. **crawl\_job\_title**:- This function crawl the job title names and save in the list from the given url using Python BeautifulSoup library.
   4. **craw\_industries\_name**:- This function crawl the skills names and save in the list from the given url using Python BeautifulSoup library.
   5. **craw\_skills\_name**:- This function crawl the skills names and save in the list from the given url using Python BeautifulSoup library.
   6. **save\_crawls\_fields\_into\_file**:- This function save the all crawled names in the candidate.xlsx file.
   7. **covert\_experience\_field\_yr\_to\_month**:- This function read job.xlsx file experience field which is in year then convert it in to max and min experience field in months and save in two lists named min\_exp\_in\_month and max\_exp\_in\_month.
   8. **save\_max\_min\_experinence\_in\_file**:- This function read min\_exp\_in\_month and max\_exp\_in\_month which is in months then save in the job.xlsx file at F and G column.
   9. **space\_at\_unstructure\_cell**:- This function unstructured data from the job.xlsx.
   10. **slipt\_dataset\_70\_30**:- This function split the dataset in 70:30 ratio and save in train and test.xlsx files.
   11. **convert\_xlsx\_to\_csv**:- This function convert the xlsx files into csv files as our requirement.
   12. **collaborative\_filtering\_algo**:- This function implement the collaborative filtering algorithm on the train and test data and recommend the jobs for the perfect candidate.