**Phases of data analysis for DUIT :**

**Phase 1 : Collecting data from LinkedIn**

**Process**: Collected target profiles First Name, Last Name and company name via linkedin CSV downloads and passed the same as input parameter to phantombuser API for scraping.

**Result** : Collected 10,482 records of linkedin users from the connections of Agrim Nautiyal, Prakhar Goel, Aditya Nautiyal and Amritansh Pandey for EDA (Exploratory Data Analysis) and for finding potential leads

**Phase 2 : Analysing Linkedin Data**

**Process**: Created Visualisations using Metabase. This involved the following subphases -

* Shifted the CSV file of 10,482 records to a local postgres Database.
* Installed the .jar file of Metabase to run local experiments on the data and save relevant queries and questions (views)
* Created a google cloud platform instance to store the local postgres DB by importing a .zip file of the local data dump. Credentials for the gcp platform : GCP login - ID : agrim.nautiyal@prakhargoel.com, password : purplefigure123; Database info : Database Instance name : linkedindata2, username : postgres, password : iAxwGO8AasMudkcH, database name : linkedin\_data\_unduplicated, public IP address : 34.93.189.225 (Remember : If you are using an external application to connect to the GCP database, kindly whitelist your application's public IP address under the connections tab/public IP in the GCP-SQL dashboard for a successful connection- currently the accepting whitelist contains 0.0.0.0 so all apps should be able to connect, kindly change if future security needs change).
* Created an AWS EC2 for hosting metabase on an online server that can be shared with other team mates. Credentials for AWS login : ID : agrimnautiyal1999@gmail.com, password : Vasu1999; The AWS RDS instance that stores our metabase workspace's info can be found at : <https://ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#databases:>

The metabase-deployment file used in our use case can be found at : <https://drive.google.com/file/d/1SNUs_GtrTThOkya7FCAVi6iLXZqeqKdc/view?usp=sharing>

* Shifted all the local metabase dashboards to the aws instance. The metabase workspace can be found at : <http://metabasemumbai-env.4fde26kzvr.ap-south-1.elasticbeanstalk.com/> Credentials for logging in and editing questions/dashboard - ID: agrimnautiyal1999@gmail.com, password : vasu1999
* Created a general-analysis dashboard for finding general trends in our data under the linkedin\_data dashboard.
* Based on the above dashboard the next steps are to create other smaller dashboards to go deeper into the general insights and create some hypothesis on the basis of which we will structure our algorithms performance.

**Phase 3 : Scraping company information**

**Procedure**: This step involves scraping company data from glassdoor based on the data obtained from our earlier dashboards to gain further insights into a companies performance and employees point of view. For this we have made use of a simple python script that extracts information into the following schema -

|  |
| --- |
| SCHEMA = [ |
|  | 'date', |
|  | 'employee\_title', |
|  | 'location', |
|  | 'employee\_status', |
|  | 'review\_title', |
|  | 'years\_at\_company', |
|  | 'helpful', |
|  | 'pros', |
|  | 'cons', |
|  | 'advice\_to\_mgmt', |
|  | 'rating\_overall', |
|  | 'rating\_balance', |
|  | 'rating\_culture', |
|  | 'rating\_career', |
|  | 'rating\_comp', |
|  | 'rating\_mgmt', |
|  | ] |

**Status** : Currently the helpful, pros, cons and advice\_to\_mgmt columns are not being scraped due to bugs in the script which will be fixed.

**Results**: All the other information is being scraped and stored in the "Collected\_Data\_Samples" folder as CSVs and will be migrated to postgres DBs once the extraction process is completed.

**Ongoing Phase : Developing the prediction algorithm**

**Procedure**: The algorithm development phase is broken down into the following subphases :

**Design** - The first iteration of this algorithm is very simple and naive. Based on the trends and importance of each feature observed in our visualisations, we will enable a sort of a scoring mechanism on each column of our Database. This scoring can be based on frequency, priority of records and/or relevance of the column(feature) involved. The idea is to calculate the aggregate sum of scores of all the columns and pass the result into a sigmoid function which is denoted by the following -



The beauty of this function is that it takes an input in the domain of (-inf, +inf) and squashes the result into an output in the range of [0,1]; which is perfect for our use case since we have to predict a probability. However we have to first dive a bit deeper into the nature of this function and see how we can normalize our scores to prevent generating skewed/unintelligent results.   
We will also make use of validating our predictions against some sample data (preferably info of existing duit users) instead of just relying on our hypothesis to reduce bias.

**Status** : This approach is still being fine tuned at the moment and different variations of the function and its use is being tried out. Results will be published after a successful first round of generating predictions. Further introspection into data trends is required as enough hypothesis haven't been generated.

*Note: For any queries/clarifications please contact me at : agrim.nautiyal@duit.io/ +91 9159289775*