RESUME TRACKING SYSTEM

FOUNDATIONAL PROJECT – 1

GROUP 5

Anand Pratap Singh	12020084
Francis David Vuppuluri	12020028
Lavanya SN	12020075
Puneet Srivastava	12020026
Sangeeta Thakur	12020088

Table of Contents

1. **Introduction**
   - **1.1. Description**
2. **Business Understanding**
   - **2.1. Business Objectives**
   - **2.2. Business Constraints**
3. **Business Criteria**
   - **3.1. Business success criteria**
4. **Economic Criteria**
5. **Design process and methods**
6. **Process Architecture**
7. **Modelling**
8. **Data Pipeline Architecture**
9. **Evaluating the Model**
10. **Conclusion**

1. INTRODUCTION

Automatic tracking system (ATS) is the model which maps the best possible candidate profiles with the job description based on skills, salary, region and experience. This can quantitatively predict if the candidate will accept and retain the offer from the big IT companies and likewise from all the sectors.

1.1. Description

In our project we focused on getting two sets of data:

a) Job offerings from the organization: The job vacancy we get in various job portals which provide information such as Industry, Skills, Requirements, Role, Salary, and Experience required
b) Candidate's Profile - The LinkedIn profile will contain candidate's profile, the skillset, their work experience.

We will use our Organizational data as standard data to compare it with the Candidates' data to map right candidates to the job openings. Using natural language processing, machine learning and rule-based techniques, this intelligent system creates a bucket of relevant jobs that a candidate is most likely to accept based on above mentioned parameters.

2. BUSINESS UNDERSTANDING

2.1. Business Objectives

- Minimize the hiring costs during pandemic, while ensuring high acceptance rate among most legitimate candidates.

2.2. Business Constraints

- Reduce the time in going through candidate's profile interested in the job openings.
- Minimize the churn rate of the candidates accepting the offer.

3. BUSINESS CRITERIA

3.1. Business Success Criteria

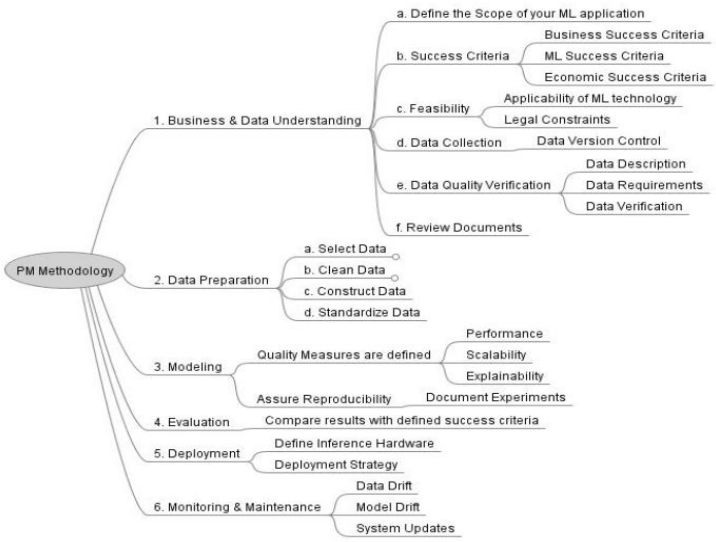
- To get a bunch of employee profiles that are best fit for a job, available to the organization within no time.

4. ECONOMIC SUCCESS CRITERIA

- This system helps to reduce the human involvement and tedious work, hence reducing significant costs.
- This system helps the organizations to get the best suited employee for a job, who would most likely retain that job once offered.

5. DESIGN PROCESS AND METHODS

We follow the PM methodology as shown in the diagram reproduced hereunder:



The system acquires information about candidates and job vacancies from social media like LinkedIn and Naukri respectively and updates the knowledge base.

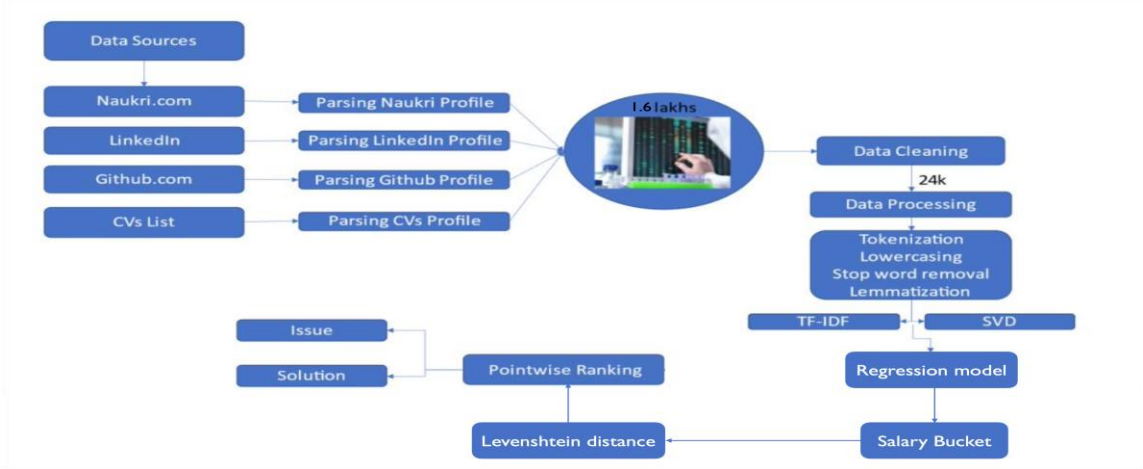
Attributes can be described as under:

1.Title	2.Company	3.Experience	4.Salary
5. Location	6. Skillset	7. Description	

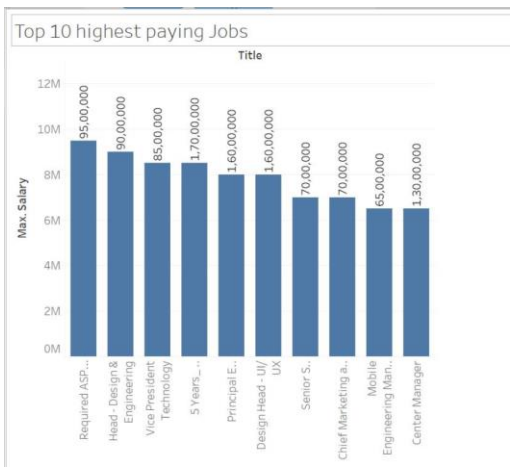
6. PROCESS ARCHITECTURE

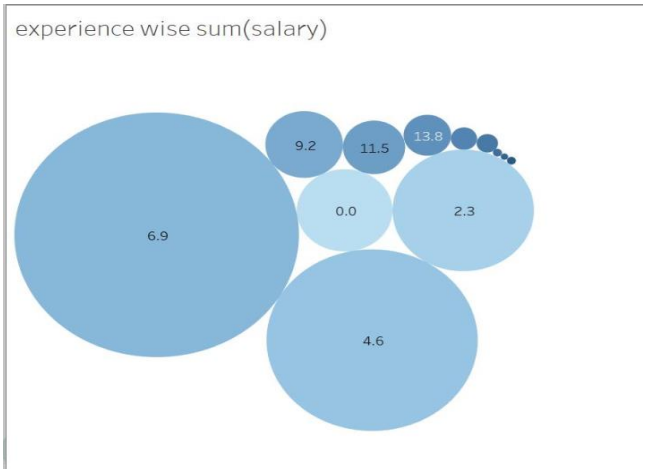
Job opening Dataset:

- Number of Openings: 24,143
- Number of skills in the current dataset: >7000
- Experience: 1 Continuous Numerical Variable
- Regions: Various Cities in India
- Salary: 1 Continuous Numerical Variable



Data cleaning and visualisation helped us to understand our data properly. Below are some findings:





- Once the data is cleaned, processed and explored, it is subjected to feature engineering. Skills are subjected to SVD with 300 components on TF-IDF scores of the skill tokens. Cities are mapped to regions and further one-hot encoded. Maximum Salary is converted to "in lakhs" to standardize all the values. This gives us the final dataframe that can be used for model training.
- Similar steps are done for the features in candidate data as well.

Final Training Dataset:

- Number of Openings: 24,143
- Number of skills: 300
- Experience: 1
- Regions: 4
- Maximum Salary: 1

```
tfidfDf = pd.DataFrame(tfidf_matrix.toarray(), columns = tfidf.get_feature_names())
```

```
tfidfDf
```

	acad	accenture	account	active directory	ad	admin	admissions	adobe	advantum	advisor	...	workforce management	working drawings	wpf	writing skills	written	x	xml	z
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
24158	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
24159	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
24160	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
24161	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
24162	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(

24163 rows × 750 columns

```
Out[478]: 0.8722003236554527
```

```
In [479]: svdTFit.transform(tfidf_matrix)
```

```
Out[479]: array([[ 1.29025638e-02,  7.38121027e-01, -1.06052218e-02, ...,
                  -5.54461872e-05, -8.58228265e-05, -6.00205732e-05],
                 [ 2.00023119e-02,  8.26536689e-01, -2.49871351e-01, ...,
                   1.13792797e-05, -4.95748118e-07,  1.28637428e-04],
                 [ 3.96944470e-03,  6.01603667e-04,  6.03937960e-04, ...,
                   1.32531920e-02,  1.11469740e-02, -2.86807139e-02],
                 ...,
                 [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00, ...,
                   0.00000000e+00,  0.00000000e+00,  0.00000000e+00],
                 [ 1.18947652e-03,  1.67549369e-04,  6.21120246e-05, ...,
                   5.51335909e-02,  4.80197101e-02, -3.32343465e-02],
                 [ 5.15761794e-03,  2.08644736e-02, -9.50065507e-03, ...,
                  -3.82397770e-02,  4.13108220e-02, -3.11453434e-02]])
```

```
In [489]: finalDf = pd.DataFrame(svdTFit.transform(tfidf_matrix))
          finalDf['Experience'] = df['Experience']
          finalDf['North'] = df['Zone'].map(lambda x: 1 if 'north' in x else 0)
          finalDf['East'] = df['Zone'].map(lambda x: 1 if 'east' in x else 0)
          finalDf['West'] = df['Zone'].map(lambda x: 1 if 'west' in x else 0)
          finalDf['South'] = df['Zone'].map(lambda x: 1 if 'south' in x else 0)
          finalDf['Salary'] = df['Salary']/100000
```

```
finalDf
```

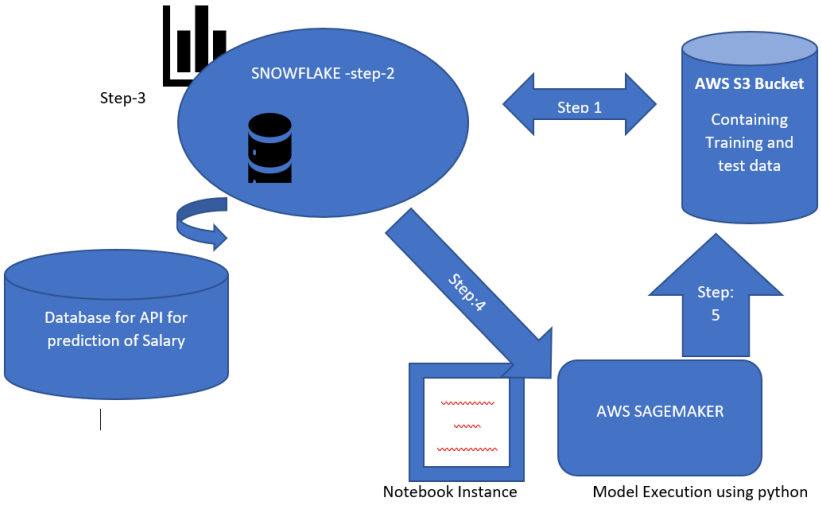
	0	1	2	3	4	5	6	7	8	9	...	296	297	298	2
0	0.012903	0.738121	-0.010605	-0.035483	-0.003779	-0.281745	0.011838	0.608862	-0.015606	-0.001278	...	0.000136	-0.000055	-8.582283e-05	-0.0000
1	0.020002	0.826537	-0.249871	-0.064987	-0.000922	0.005366	-0.006647	-0.498837	0.020246	-0.000294	...	-0.000085	0.000011	-4.957481e-07	0.0001
2	0.003969	0.000602	0.000604	0.000174	0.272257	-0.003045	0.000184	0.000380	0.002689	0.260993	...	-0.031517	0.013253	1.114697e-02	-0.0286
3	0.021081	0.231005	-0.079054	0.018424	0.005994	0.397498	-0.010521	0.075677	-0.066641	0.001903	...	0.017225	0.001121	6.945132e-03	-0.0056
4	0.075304	0.189191	-0.109449	0.091991	0.013674	0.934863	-0.027671	0.200398	-0.074636	-0.003368	...	-0.000006	-0.000015	8.042251e-07	-0.0000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
24158	0.000188	0.000428	0.000122	0.000117	0.000406	0.000758	0.000705	0.000589	0.000162	0.002687	...	-0.044233	-0.050643	5.261549e-02	-0.0561
24159	0.000237	0.000074	-0.000036	0.000103	0.000029	0.000395	0.000859	0.000203	0.000341	0.001320	...	-0.023780	-0.046545	-7.396646e-02	-0.0394
24160	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000e+00	0.0000
24161	0.001189	0.000168	0.000062	0.000074	0.000042	0.000653	0.000413	0.000307	0.000696	0.028511	...	0.082599	0.055134	4.801971e-02	-0.0332
24162	0.005158	0.020864	-0.009501	0.009314	0.001988	0.122799	-0.001755	0.034963	-0.021607	0.005055	...	-0.140455	-0.038240	4.131082e-02	-0.0311

7. MODELING

Once the model(Linear Regression) is trained, it is validated against the validation dataset and results are compared against the actual maximum salaries.

	Title	Company	Experience	Zone	Simplified Skillset	Actual Salary	Predicted Salary
2183	Urgently looking For CPQ Technical Consultant	Virtusa Consulting Services Pvt Ltd	8.5	{'south', 'west'}	communication, java, soap, html, java, cpq, or...	18.0	17.974945
21270	Chief Marketing and Growth Officer For Healthc...	Techs to Suit Inc	17.5	{'north'}	market, tech, digital, market, chief growth of...	3.0	3.140344
832	AWS Senior Cloud Engineer-5-9 Years	Virtusa Consulting Services Pvt Ltd	7.0	{'south', 'west', 'north'}	it, cloud, devops, aws, druva, cloud, cloud, s...	3.0	2.999627
21530	Hiring For British Telecom I UK Voice Process ...	Planet recruiters	3.0	{'north'}	bpo, voice, dell, customer, captive unit, voic...	18.0	21.057374

8. DATA PIPELINE ARCHITECTURE



- Create stages, databases, tables, user, role and warehouses on Snowflake.
- Load data into a table from S3 bucket.
- Using AWS Sage Maker Notebook instance for model preparation and Execution.
- Using Snowflake instance and pull data into a Pandas data frame.
- Visualizing the data and performing basic feature engineering.
- Unload a dataset into S3 and using it to train a machine learning model.
- Run a batch of data through model and load the results back into Snowflake.

Softwares and Versions used:

- Beautiful Soup 4.9. 1
- Python 3.8
- Jupyter Notebook
- Snowflake Enterprise Edition

9. EVALUATING THE MODEL

Once the candidate data is enriched with maximum current salary, it is compared against the jobs data on two parameters:

1. Salary Bucket: Using the candidate dataset, extract Job Openings for each candidate where **Predicted Maximum Salary** lies within the range of (0.7 - 1.2) times the Offered Maximum Salary for that job.
2. Skill Bucket: Skills required for jobs in the salary bucket are compared against the skillset of the candidate based on Levenshtein's Distance similarity score.

In [249]:
```
1 candidateToJob['C33']
```

```
[('J23390', 65),
 ('J25118', 64),
 ('J25123', 64),
 ('J17299', 64),
 ('J17981', 58)]
```

In [250]:
```
1 test[test['Index'] == 'C33']
```

	Index	Skillset	Experience	Zone	Salary LR
31	C33	angular, asp, c, web serv, sql, net, asp, linq, java, jquery, ssrs, ssis	9	{'south'}	14.709382

In [251]:
```
1 train[train['ID'] == 'J23390']
```

	Title	Company	Experience	Salary	Skillset	Zone	Simplified Skillset	ID
23389	Opening with MSys For MEAN Stack developer - Immediate Joiners	MSys Tech India Pvt. Ltd.	5.5	20.0	['AngularJS', 'SQL', 'Hapi', 'Node', 'Mongo DB', 'Web services', 'TDD', 'Javascript']	{'south', 'east', 'west', 'north'}	angular, sql, api, node, mongo, web serv, tdd, java	J23390

10. CONCLUSION

Our system will provide better and efficient solution to current hiring process. This will provide a legitimate candidate to the organization and the candidate will successfully be placed in an organization making it a good cultural fit, which, in turn increases employee retention. This model can be used for all domains after training it for various domain as required.