**Intelligent Hiring Solution using Big Data, Spark and Machine Learning**

**JPMorgan Chase and Co.,**

**Contributors:**

**Vivek Mishra - Technical Architect**

**Rajesh Rajamanickam – Senior Programmer**

**Owners:**

**Varghese Paul – HR Head**

**Supriya Anand - HR Associate**

Algorithm:

**Step 1**: Feature Extraction - **Delivered**

Sub Algorithm: Parse the resume using open source text parser to filter the job profile and resume into different categories.

Categories: Education background, working experience, skill set, requirement to job/job nature, Current position/Job position

**Step 2:** Semantic Analysers - **Pending**

Sub-Algorithm:

Word2vec: [Loading Data](https://deeplearning4j.org/word2vec#loading-data)

[Tokenizing the Data](https://deeplearning4j.org/word2vec#tokenizing-the-data)

[Training the Model](https://deeplearning4j.org/word2vec#training-the-model)

Evaluating the Model, Using Word2vec

**Code sample:**

Double cosSim = vec.similarity ("Programmer", "Software Engineer");

//output: 0.7704452276229858

**Step 3:** Vector skill analyser using Word2Vec - **Pending**

Use **Step 2** algorithm to find the similarity in Skills

Find the average of probability in order to get the probability of multiple skills

**Step 4:** Converting category into numeric data job skill and job desc - **Pending**

|  |  |  |
| --- | --- | --- |
| **Category** | **Numeric data** | **Job desc** |
| Education level | 4 | 5 |
| Major of Study – **Same as requirement : 1 otherwise 0** | 1 | 0 |
| Industry Experience -– **Same as requirement : 1 otherwise 0** | 0 | 0 |
| Position Name – **Use Step 2** | 0.87 | 0.87 |
| Experience – **In Months** | 60 | 70 |
| Skill set – **Use Step 3** | 0.71 | 0.71 |
| Salary range **– rate algorithm** | 0.5 | 0.3 |
| Work location - **distance** | 200 | 200 |
| Work time **- Same as requirement : 1 otherwise 0** | 1 | 0 |

Use Euclidean distance or cosine similarity to find the probability score/distance and select top 50 to shortlist.

**Code sample:**

distance([4,1,0,3,0.87,60,0.71,0.5,200,1], [5,0,0,5,0.87,70,0.71,0.3,200,0])

// 34.17601498127012

**Step 5:** Using numeric data building a supervised model **[Supriya agreed to give 3000 resumes/300 job desc and list of candidates who cleared/not cleared the interview] - Pending**

Use Logistic regression or Decision tree to check whether candidate will be hired or not.

**Data set sample:**

**Input**

[4,1,0,3,0.87,60,0.71,0.5,200,1 ]-Y

[3,1,0,3,0.77,50,0.31,0.5,100,1 ]-N

[3,1,0,3,0.77,50,0.31,0.5,100,1 ]-Y

Use **Spark ML Lib** Logistic regression to get probability [0-1] out of above selected 50 to select top 20 or less to call for the interview.

**Performance/Memory usage of the model - Pending**