**PROBLEM STATEMENT-1: DATA EXTRACTION FROM RESUMES (JOBIFY MATCHER)**

**1)PROPOSED SOLUTION:**

* **PROCESS:**  An automated intelligent system is required which can take out required information from hundreds of resumes that were applied for a big corporate company, structure them and specify a job position based on the details collected. The information will be extracted using Natural Language Processing and parsed information may include years of work experience, education experiences and details, technical skills, certifications, and much other information as per the requirement of organization and then a specific job position will be assigned based on the cluster of the parsed information. The parsed information is then stored in a MySQL database for later use and person is assigned on the parsed information using Machine Learning and can be accessed by the Talent Acquisition Team with a comfortable User interface as a web application built in flask. So, the user can upload the resume and can get required information out of resume.
* **TECHNICAL DESIGN:** 
  + pdfminer and doc2text python modules: These modules help extract text from .pdf and .doc, .docx file formats.
  + Sentence Tokenization and Word Tokenization: To extract skills using a technique called tokenization. Tokenization simply is breaking down of text into paragraphs, paragraphs into sentences, sentences into words.
  + nltk (Natural Language Toolkit) module: to load an entire list of stopwords and later on discard those from our resume text.
  + io module: to perform stream and buffer operations in Python
  + Framework: flask
  + Database: MySQL

**2)STAGES OF IMPLEMENTATION:**

* **CHALLENGES FACED, AND RESOLUTION:** Generally, resumes are difficult to parse because they vary in types of information, their order, writing style, etc. Moreover, they can be written in various formats like '.txt',.pdf', '.doc', '.docx', etc. To parse the data from different kinds of resumes effectively and efficiently, the model must not rely on the order or type of data. Each candidate would follow different styles in mentioning their information makes it difficult to parse the resume and get the relevant information and it is easy for us human beings to read and understand those unstructured or rather differently structured data because of our experiences and understanding, but machines do not work that way. Machines cannot interpret it as easily as we can, hence it is a tedious task than we think. To overcome this challenge, we need to use Natural Language Processing to extract the required information about candidates without having to go through each and every resume manually, which ultimately leads to a more time and energy-efficient process.
* **Approach alternatives, and why chosen approach was selected:**
  + **ALTERNATIVES:**
    - Predictive Analytics
  + **APPROACH SELECTED:**

I have chosen keyword-based parsing and also many python libraries were already existing for ease of our intermediate tasks to get done, conversion of data format and parsing the converted data is easier with Natural Language Processing. Named Entity Recognition (NER) using Natural Language Tool Kit (NLTK), where text blocks are classified into segments which are in turn chopped into words and these words are labelled using the Natural Language Tool Kit which labels by assigning tags to each word.

* **How to validate model’s accuracy:**

It can be validated in long term after training the model with different set of resumes for accurate results.

**3)IMPLEMENTATION REQUIREMENTS:**

* **HARDWARE/SOFTWARE REQUIREMENTS:**
  + Python3: install python environment
  + Pip3: to install python packages(pip install pip)
  + Nltk and spaCy for NLP operations
  + Dataset of resumes
  + The importables: csv, re, spacy, pandas, io, pdfminer modules– PDFResourceManager, PDFPageInterpreter, TextConverter, LAParams, PDF page – os, sys, pandas, numpy, en-core-web-sm,
  + Framework: flask
  + Database: MySQL
* **APPROACH TO PRODUCTION DEPLOYMENT:**

After working model is built it deployed as a web app using Django/Flask and retrieved information is stored in database

* **How will the model be kept updated after prod deployment:**

After model is deployed it will be regularly trained with new set of resumes with different styles and will be updated.

**4)LIMITATIONS:**

* Parsing the year of completion will be not more accurate as it always gets collided with date of birth with the candidate if mentioned and also resume may consist many other dates mentioned.
* There will be a doubt in accuracy if there are vague words which may not help the model to detect required information.

**5)RESULT:**

* **ACCURACY OF MODEL:**

Accuracy of the model can be achieved within long term after training the model with different set of resumes.

* **BENEFITS OF MODEL:**
  + It saves time for an organization to filter out the bulk of applications.
  + It helps to eliminate unconscious bias
  + It helps to easily categorize the applicants to suitable job roles
* **OTHER APPLICATIONS:**

The proposed model is not only limited for organization to select applicants but also can be extended for applicants/users in searching of jobs based on the skills mentioned in the resume through an email alert. So, on an overview both the organization and applicant can be benefited.