**Technical Assessment**

Dataset:

**2010\_W1\_sample.zip:** This sample US Job Market data contains job postings scraped from all job portals of US between 01/01/2010 and 01/07/2010. Each job has multiple fields that correspond to occupation, sector, firm name, geographic location of the posting, etc. The skills mentioned in the job are extracted separately into the field called canon skills. Multiple skills are grouped together under the name skill clusters and multiple skill clusters are grouped to skill cluster families.

Prefered IDE: Jupyter Notebook

Prefered Language: Python

It is advised to create a virtual environment to produce a requirements file.

**TASK 1: Data Preparation (20%)**

1. Unzip and extract the XML file to a CSV without any loss of data. Print the time taken for this conversion in seconds. Display the number of records present and distinct columns.
2. Display the number of missing values for occupation-level classification and sector-level classification (Research on SOC, O\*NET, NAICS codes). Report the ratio of missing values per sector and per 2 digit occupation codes.

**TASK 2: Descriptive Analysis (Optional)**

1. Explore all the columns in the dataset, research on what each field corresponds to. What fields excite you the most?
2. What are the top 5 occupations that are in demand during that time frame?
3. What specific skills were highly mentioned?
4. Are there any sectors that are surprising in terms of the skills asked? (High ask of the skill ‘Music’ in mining? That’s interesting!)
5. Can you think of any other interesting research question that can be answered with the data?

**TASK 3: Keyword Extraction (40%)**

1. Pick any one specific skill from the list of skills and display the number of job postings that mention it. (Ideally a more generic skill like IT with good data subset size to train an ML model).
2. Collect all job descriptions of the skill you selected before (JobText column corresponds to raw job descriptions seen as it is in the job portal). Perform basic NLP cleaning on the text and mention the steps in detail.
3. Extract the distinct unigrams and bigrams from the above dataset and compute their corresponding TF-IDF scores. Display the top 20 unigrams and bigrams. Do these make sense, given the skill mentioned? (Optional)
4. Build a basic binary Naïve bayes classifier that says whether the skill is present or not (You would be required to create a Boolean variable that holds 0 and 1 values indicating the presence and absence of the skill in that job posting and use that as the dependent variable).
5. Display the performance of the model by performing basic cross validation. Report accuracy, F1, precision and recall and share your interpretation of the results.
6. Display the top 10 keywords (unigrams & bigrams) that are influencing the output of the model. Are these relevant to the skill selected? (Optional)
7. Experiment with any other classifier and compare the results with the naïve bayes model created before.
8. TF-IDF scores are not always the best in terms of model performance. Research on other ways to use text in binary classification and mention a few lines on each of them.
9. Pick any one of the other text to numeric conversion strategy and build the same naïve bayes classifier with it. (Optional)
10. Compare the performance of these models and reflect on what works best. (Optional)

**TASK 4: Big Data Adaption (20%)**

1. Most of the current analysis use pandas or classic python dataframes. How can the speed of data processing be improved by using big data techniques?
2. If used pandas dataframes before, create a pySpark data frame and reflect on the improvement in data processing.
3. Does Hadoop work in this case? What are the different nodes of Hadoop and write a brief on YARN. (Optional)

**TASK 5: Mathematical Modeling (16%)**

1. How does a naïve bayes classifier work? Briefly talk about the bayes theorem, inferential and descriptive statistics.
2. How does a TF-IDF score work? If we are to extend the TF-IDF score to a document, what would be the mathematical formulation of it?
3. Without any evidence, what words do you think have higher TF score from the job descriptions?
4. Explain the math behind the text to numeric conversion strategy you picked in Q 3.8.

**TASK 6: Deployment (4%)**

1. Congrats on putting in so much hard work till this task! A good research is when one is able to explain his code and data with utmost clarity. It is crucial to maintain a proper codebase with meaningful folder structure and comments. Create a github repository to store all your code and analysis. Fill in the readme.

NOTE:

1.This assessment is constructed to test your skills and exposure to NLP tools and concepts.

2.You can submit the assessment in a zip folder or share the link to your GitHub Repo.

3.Please export a copy of your code files and output files to PDF for ease of readability.

4.Store all generated results (if relevant) and a well-commented and modularized code and the final output files. You can also add your methodology as a separate PDF file.

5.Do not share the sample datasets shared with you on your GitHub repository or any other online platform. Use a .gitignore file to exclude the data files.

6.You can ask for an extension of 24 / 48 hours if you have valid reasons.

7.You could submit the assessment with partial responses if you could not complete either of the tasks thoroughly. We know that you are in the learning phase, and we will consider your approach to the problem statement and your thought process in the evaluation.