**Project**

**In all sections a code should be written**
**plus a brief explanation of how you did what was required**

Part 1 - Answer according to data from jobs-on-naukricom

For each section provide the numerical answer and briefly explain how you calculated and what the challenges were.

1) What are the 10 most common roles? Note that there may be some similar roles - imagination matrices should be defined and determined. The intent in this section is to produce a table of frequencies, i.e., go over the column of roles, count instances of each Role (or similar roles) and summarize their prevalence.

2) What is the distribution (histogram) of salaries? How many unusual observations are there? Report the key statistics - standard deviation, average, median and quarterly range. Note: Some salaries appear as a range and you must find a way to convert these fields to numbers (by finding the average of that range).

3) What are the 10 most common skills? Is there an overlap of skills between different roles?

4) What are the 10 most common skills in the highest paid positions?

3 + 4 - You are required to break down the sentences into words and move on from there

5) Is there a correlation between seniority and the proposed salary? What is and how did you calculate it?

6) Is there a particular pattern of jobs across geographic locations? If such a pattern exists, is it backed up by an external source?

7) Is there a particular pattern of industries across geographical locations? If such a pattern exists, is it backed up by an external source?

8) Create a cloud of words from job descriptions - is there an interesting pattern?

9) Describe in general how can such information be connected to a system of recommendations for job seekers? What are the significant challenges in this type of system? explanation.

?

part 2 -Answer according to data “section-2-data”

In this section you will work on a forecasting model for customer churn.

All code will be written in PySpark.

1. Key statistics of the columns should be reported, the type of analysis should be matched to the type of data (e.g. continuous variable, discrete variable, etc.).

2. Statistics between variables must be reported - for example, the average wage as a function of companies, etc. Only relevant statistics should be reported.
PySpark libraries can be used such as https://spark.apache.org/docs/2.2.0/ml-statistics.html

3. Is there an interesting connection between salary (estimated) and the presence of a credit card? Does the relationship seem linear? It is advisable to first draw the variables and then decide whether to calculate correlation (You can use https://docs.databricks.com/notebooks/visualizations/index.html )

4. Have interesting patterns been found in sections 1,2? Do these findings support the accepted assumptions in the world of churn prediction?

5. Present a logistic regression model to predict a customer's churn chances. Be sure to divide the data into train / test. Report the classification error as well as the significant coefficients - do the findings "make sense"?

6. Continuing from the previous section - by how much is a customer more likely to stay if he is an active member?

7. Match another model (by choice - knn for example, or decision tree, recommended to use something simple) to the churn classification. Did you get better results than in section 5? If so, explain why?

Part 3 - fake & true files

For each section provide the numerical answer and briefly explain how you calculated and what the challenges were.
1) Read about TF / IDF algorithm for classifying text content.
2) Implement the TF / IDF algorithm in PySpark, it is recommended to implement the TF as a function and the IDF as an additional function and run the functions in order.
The following guide can be used to clear the text:
https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/3923635548890252/1357850364289680/4930913221861820/latest.html
3) Run the algorithm on the text column and every word in each file and report the main findings.
Each word has an appropriate value for each document - sort this table and report the first 20 lines.
The idea is to see that there is an interesting connection between 'rare' words and high values. Note that documents whose majority of the text was common words (which you may have chosen to exclude in section 2) will be given a high value because there are few words left in them, note this bias in the findings.
4) Create a cloud of words from the text. Do the findings match those you found in Section 3?
In this section, create a word frequency table and draw the 40 most common words in the word cloud.
5) Briefly describe the business use-case in which this algorithm will be used.