Thinking Machines
Data Science

Machine Learning Researcher Take Home Exam

General instructions

You are given **96 hours** upon receiving the exam to complete the deliverables.

Upload all your deliverables inside the *Exam* folder of the Google Drive folder shared with you after your application. If you are applying for more than one position, segregate your exam deliverables by creating one folder per position.

Index all your **files in one document**, and upload it in the *Exam* folder. For **code and test cases**, we prefer the following methods :

- **upload them in private repositories like Gitlab or BitBucket,**
- or upload them in our **shared *Exam* subfolder**.

Make sure they are **properly linked inside the document**. Other files must be uploaded as well, again with links in the main document.

Remember : **do not upload any parts of this exam or your answers on publicly accessible sites** like the public repositories on Github. This is to ensure that the exam will not be leaked to future applicants.

Upon completion, send an email to your hiring manager with a link to your deliverables to stop the timer. Good luck!

1. Coding exam

Implement the following models in Python:

- Decision tree
- K-means

You are free to use any external libraries or packages to write your code, except for libraries and packages that directly implement the models (for examples, *scikit-learn or gensim/ntlk* in Python). Create the following functions for each model which can be used for testing:

- Decision Tree : `decisionTree(data, max_depth, min_samples_leaf, min_samples_split)`
- k-means : `kmeans(data)`
    - where data is an MxN numpy array
    - This should return
        - an integer K, which should be programmatically identified
        - a vector of length M containing the cluster labels

Provide the seed and other fixed values used when necessary.

For bonus points, you can do any number of the following :

- Implement the Decision Tree model without using any external libraries *(you may use matrix libraries such as numpy*)
- Implement the K-means model without using any external libraries *(you may use matrix libraries such as numpy*)
- Implement the Random Forest model, with the freedom to use external libraries or packages, except those that directly implement it.
    - `rf(train_data,train_labels,test_data)`

2. Exploratory data analysis

Pick one non-English dump from Wikimedia, and perform exploratory data analysis on it. The goal is to surface **at least 3 main insights** from your analysis. Feel free to angle your analysis in any way you see fit. You are also free to use any tool, package, or library to do your analysis. You may even use the code that you created in the first part of this exam.

You are expected to surface interesting insights from your dataset, so please refrain from simply describing its contents and characteristics.

Compile your insights in one slide deck, which you will present to our panel if you move on to the next round.

Compile all that code that you used for your analysis, and send them our way as well.

3. Situational questions

Create a document and answer the following questions :

1. Suppose you are asked to build a model to determine whether a satellite image of an area has low, medium, or high population density. Your labelled dataset contains 10M images with 3 highly unbalanced classes (low, medium, and high). How would you choose your train/dev/test split (or cross validation scheme)? What do you think is the best approach to handling the class imbalance problem?

2. Suppose you've chosen a neural network to classify the images. After training your system, you get the following accuracy scores:
   - Train set accuracy: 80%
   - Dev set accuracy: 40%
   What problems can you identify, and what would be your next steps?

3. Suppose the owner of a well-known restaurant asks you to create a model that can classify customer reviews as positive or negative. He has given you a sample of 1500 client reviews labelled accordingly (75% positive, 25% negative). What would be your step-by-step approach to solving this problem? Be specific!