Gene Solutions Data team interview test

(20 points, 4 bonus points)

Instruction

The candidate has 7 days to complete as many of the questions listed below as possible. The completed test must be submitted via email to *hieunguyen[at]genesolutions.vn*. Each answer should be presented in a clear and concise written format. Candidates should also prepare a small presentation to present the answer in an interview seminar with our Data team. This test is designed to evaluate the candidate's knowledge of basic data science, statistics, machine learning and coding skills. The total score is 20 points, with the possibility of earning additional points for solving *Bonus* questions.

While ChatGPT or equivalent platforms are allowed, candidates are encouraged to do the test on their own and should be able to provide detailed explanations for each answer, if required. For any questions regarding the test, please feel free to contact *hieunguyen[at]genesolutions.vn*.

On behalf of the Data team at Gene Solutions, we would like to thank you for your interest in applying for a position in our team and thank you for your time and effort in taking this test. We wish you the best of luck and look forward to receiving your results.

Non-disclosure agreement

The test is confidential. It is made available to you, the candidate, solely for the purpose of the interview process. You are prohibited from disclosing, publishing and reproducing this test in whole or in part, in any form or by any means, verbal or written, electronic or mechanical, for any purpose. By accepting to participate in this test, you agree to be bounded by our interview regulations and these terms and conditions.

Q1. Foundation of Data Science (2 points)

Write a script in your own favorite programming language to randomly generate 1000 points on the surface of a sphere in 3-dimensional space and in 100-dimensional space. Create a histogram of all distances between the pairs of points in both cases. Comments? From such observations, simply explain the concept "The Curse of Dimensionality" and how it affects machine learning model performance. What should one do to mitigate the effect of "The Curse of Dimensionality" in a high-dimensional data set?

Bonus 1: Explain the concept "The curse of dimensionality" with geometrical/mathematical intuitions.

Q2. Basic statistics (2 points)

Explain the meaning of p-value and the concept "Multiple testing corrections". Why and how do we do multiple testing correction?

Q3. Foundation of Machine Learning (4 points)

Q3.1. Underfitting, overfitting (1 point).

Visually explain the concepts "Underfitting" and "Overfitting" (Hint: The variation of bias and variance with model complexity diagram). Bonus 2: *Derive a mathematical formula of the generalization error (expected test error) of a classifier and decompose it to three interpretable terms: Variance, Noise, and Bias.*

Q3.2. Logistic regression (3 points)

A junior team member approaches you with the following question:

"Logistic regression, by its name, seems to suggest it is a regression algorithm. However, it is commonly used as a classification algorithm in practice. Why is it called "regression" if it's always applied to "classification"? Additionally, could you walk me through the derivation of the logistic regression algorithm, including problem formulation, maximum likelihood estimation, and gradient descent? While others on the team can easily implement logistic regression using scikit-learn, no one has explained how to truly understand it to me. Could you please help me out?"

As a senior member of the team, how would you respond?

Bonus 3*: Could you also show our junior members how to implement Logistic regression from scratch in python without using Scikit-learn?*

Q4. Basic coding test (3 points)

Next-generation DNA sequencing data often requires bioinformaticians/data scientists to deal with large text files containing millions of lines. While parsing and processing such files using Python or R is feasible, it is often not optimal in terms of runtime performance. The AWK programming language, a domain-specific language specializing in text processing, offers a faster alternative for data extraction tasks.
In this task, the candidate is required to use AWK and BASH scripting techniques to process the provided *Supplement_data_file.bam* file.

For *.bam file, one can use *samtools (https://www.htslib.org)* to view and convert the file to tab-separated text format *.txt, using

samtools view Supplement_data_file.bam > Supplement_data_file.txt

Perform the following tasks:

- Install *samtools* in your computer and view the *Supplement_data_file.bam,* convert it into .txt file for further downstream processing. Samtools built-in docker/singularity image is also accepted.
- Extract all values on the 9th column of the file, create a histogram of their absolute values (the histogram can be generated by python or R).
- The 6th column is called the CIGAR string, which has the regular expression

  *|([0-9]+[MIDNSHP=X])+

  Remove all lines whose CIGAR string is not [0-9][0-9]M, e.g.: Keep 50M, remove 40M5S5M.
- If the 9th column value is positive, extract the first 4 letters of the string on the 10th column, otherwise, extract the last 4 letters of the 10th column. Create a histogram demonstrating distribution of all these 4-letter sequences.
- Propose a metric to measure the diversity of the above 4-letter sequences in the data. Explain your choice.

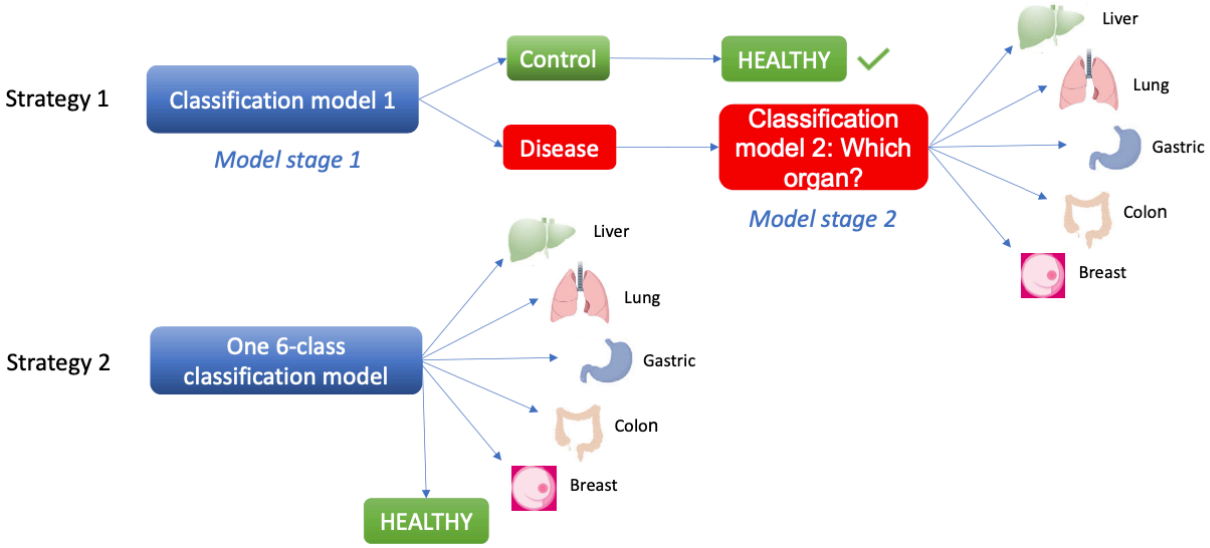
Bonus 4: Perform the above tasks again in python and compare the runtime performance. Draw your conclusions.

For more information on SAM/BAM file format, see

- https://support.illumina.com/help/BS_App_RNASeq_Alignment_OLH_1000000006112/Content/Source/Informatics/BAM-Format.htm
- https://samtools.github.io/hts-specs/SAMv1.pdf

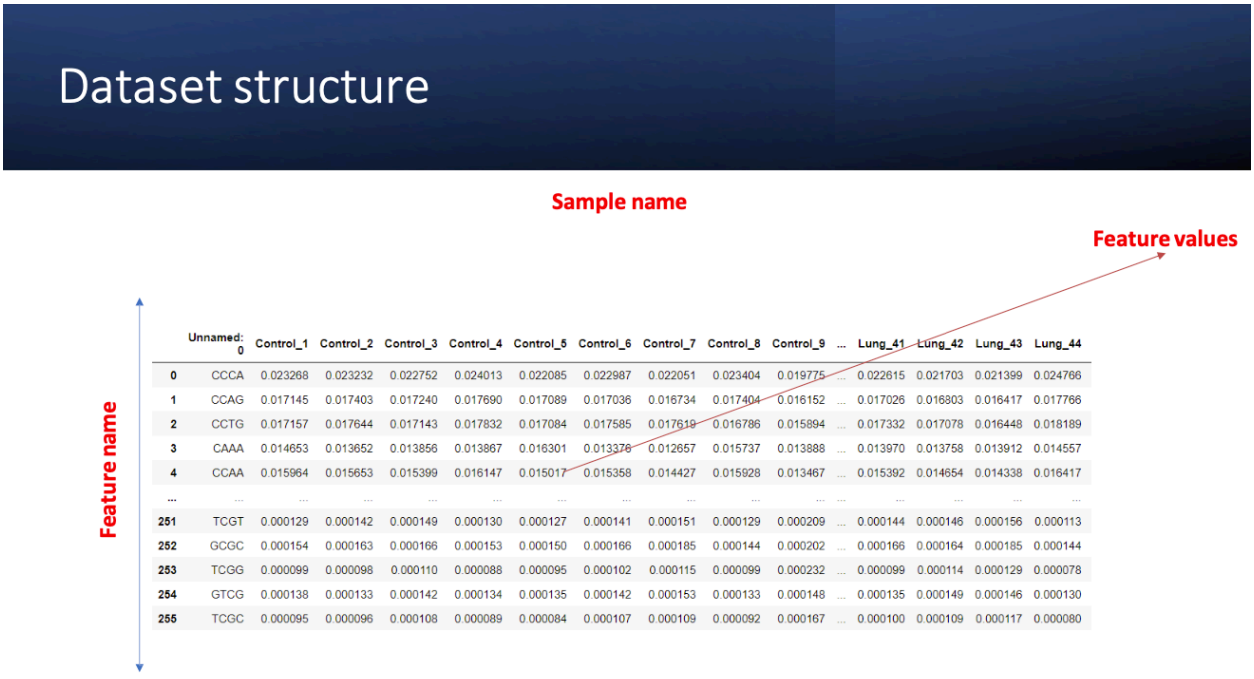
Q5. Data analysis and machine learning model construction (7 points)

Perform exploratory data analysis on the dataset "*Supplement_datasets.zip"* and build machine learning models to classify 6 classes. Candidates could choose one of the following two strategies to implement the model. Candidates should also demonstrate how they do cross-validation, feature selection, model tuning and model selection.


Strategy 1
Classification model 1
Model stage 1
Control
Disease
HEALTHY
Classification model 2: Which organ?
Model stage 2
Liver
Lung
Gastric
Colon
Breast
Strategy 2
One 6-class classification model
Liver
Lung
Gastric
Colon
Breast
HEALTHY

"Supplement_datasets" description:

- ☐ Number of groups of features: 3 (FLEN, EM, NUCLEOSOME).
- ☐ Number of classes: 6 (5 disease classes: CRC, Liver, Lung, Breast, Gastric. 1 healthy class: Control)
- ☐ Number of samples: 50 samples per class (CRC, Lung, Breast, Gastric). For Liver class, we have only 30 samples. For Control class, we have 70 samples.
- ☐ Each group of features (FLEN, EM, NUCLEOSOME) is presented as matrix [*m_features, n_samples]*. The first column of the *.csv file is feature *name:*
  - ☐ FLEN features: {50, ..., 350}
  - ☐ NUCLEOSOME features{-300, ...0, ..., 300}
  - ☐ EM features: 256 combinations of the 4 letters {A, T, G, C}.
  - ☐ {Control_1, ..., Control_70, Liver_1, ..., CRC_1, ...} are Sample IDs.



	Unnamed: 0	Control_1	Control_2	Control_3	Control_4	Control_5	Control_6	Control_7	Control_8	Control_9	...	Lung_41	Lung_42	Lung_43	Lung_44
0	CCCA	0.023268	0.023232	0.022752	0.024013	0.022085	0.022987	0.022051	0.023404	0.019775	...	0.022615	0.021703	0.021399	0.024766
1	CCAG	0.017145	0.017403	0.017240	0.017690	0.017089	0.017036	0.016734	0.017404	0.016152	...	0.017026	0.016803	0.016417	0.017766
2	CCTG	0.017157	0.017644	0.017143	0.017832	0.017084	0.017585	0.017619	0.016786	0.015894	...	0.017332	0.017078	0.016448	0.018189
3	CAAA	0.014653	0.013652	0.013856	0.013867	0.016301	0.013376	0.012657	0.015737	0.013888	...	0.013970	0.013758	0.013912	0.014557
4	CCAA	0.015964	0.015653	0.015399	0.016147	0.015017	0.015358	0.014427	0.015928	0.013467	...	0.015392	0.014654	0.014338	0.016417
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
251	TCGT	0.000129	0.000142	0.000149	0.000130	0.000127	0.000141	0.000151	0.000129	0.000209	...	0.000144	0.000146	0.000156	0.000113
252	GCGC	0.000154	0.000163	0.000166	0.000153	0.000150	0.000166	0.000185	0.000144	0.000202	...	0.000166	0.000164	0.000185	0.000144
253	TCGG	0.000099	0.000098	0.000110	0.000088	0.000095	0.000102	0.000115	0.000099	0.000232	...	0.000099	0.000114	0.000129	0.000078
254	GTCG	0.000138	0.000133	0.000142	0.000134	0.000135	0.000142	0.000153	0.000133	0.000148	...	0.000135	0.000149	0.000146	0.000130
255	TCGC	0.000095	0.000096	0.000108	0.000089	0.000084	0.000107	0.000109	0.000092	0.000167	...	0.000100	0.000109	0.000117	0.000080