Personal Project Report: 2021 GIDS Data Science Hackathon

Jiahang Wu and Maxim Gorshkov

I. INTRODUCTION

This report documents the process of the working processes for the participating project in University of Rochester Biomedical Data Science Hackathon. The report explains thought processes, data preprocessing, and the training of the data for age prediction.

The goal of the project is to predict the age of patients from the unnormalized counts of 50,000 genes from RNASeq performed on patients, which is the only instructions given to participants.

Part of the resources (images, etc.) used in this report are prepared shortly after the end of the Hackathon.

II. DIMENSION REDUCTION

The goal is to reduce the dimensions of the dataset in order to meaningfully interpret the data. We are restrained by the amount of training data (965 rows) which is vastly outnumbered by features. While we would prefer having less features than rows - as we intuitively consider that a minimum bar of data sparsity -we will only drop features as we can provide a scientific reasoning for the decision.

The data is loaded onto the python notebook and separated into predictor features and age (predicted feature, or Y). The main packages here are pandas and sklearn.

Fifty random rows and the description table of the dataframe are printed (not displayed here). The raw features have vastly different ranges and are thus normalized.

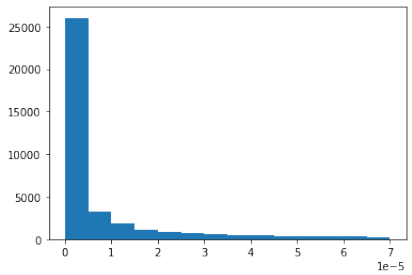

Fig. 1. Histogram of Feature Standard Deviation

As shown, most features of the dataset do not vary in counts. We assume this is because the dataset is the full sequence of human genes, and that most genes are not associated with age but population. Potentially, this could show inter-source biases of our data which is compiled from three different researches; however, we do not possess the information or the background to investigate this path.

We must add that, for our study, the age is actually the causal indicator for the counts of genes rather than what we are attempting now. This is because, logically, gene is usually affected by exposure to elements in the environment, which our age indicates.

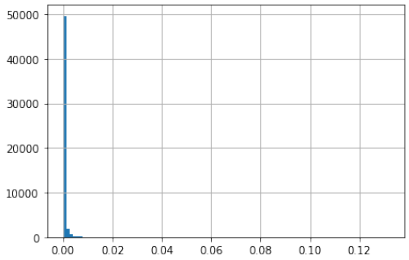

Fig. 2. Alternative Plot of Feature Standard Deviation (pandas default)

TABLE I
DESCRIPTION OF FEATURE STD

count	52935
mean	4.062865e-04
std	2.422413e-03
min	1.134322e-08
25%	5.184202e-07
50%	5.502466e-06
75%	1.420882e-04
max	1.318569e-01

We can expect reasonably that more than 75% of the genes do not vary (>0.0001 std) and are unlikely to contribute to our prediction. We employed the SelectKBest method from sklearn (which measures mutual information between X and y) to remove 75% of the features. We are left with 13,233 features which certainly need further reduction to improve prediction and reduce time and resource cost.

We then calculated the pearson's correlation between each of the remaining features with age.

TABLE II
DESCRIPTION OF X-Y CORRELATION

count	13233
mean	0.081787
std	0.140539
min	-0.317465
25%	-0.053000
50%	0.138972
75%	0.203979
max	0.306376

We removed all features with >0.2 correlation with age. This leaves us with 3,873 features. This selection of features is pickled (X_filtered_4K.pickle) for testing our learning

models.

Because we are working with such few number of samples, we are obliged to attempt further reduction based on the inter-feature correlation. We generated a cartesian product of correlations between the 3,873 features.

If we consider the correlation as edges between features, this dataframe is practically a matrix-represented graph of features' correlation. This is conceptually very helpful.

	ENSG00000000457.13	ENSG00000001461.16	ENSG00000004059.10	ENSG00000004142.11	ENSG00000004478.7	ENSG00000004779.9
ENSG00000000457.13	NaN	0.629971	0.586050	0.583838	0.560616	0.616913
ENSG00000001461.16	0.629971	NaN	0.427006	0.504683	0.566244	0.494109
ENSG00000004059.10	0.586050	0.427006	NaN	0.606935	0.550115	0.777183
ENSG00000004142.11	0.583838	0.504683	0.606935	NaN	0.641047	0.686901
ENSG00000004478.7	0.560616	0.566244	0.550115	0.641047	NaN	0.670017
...	...	...	...	...	...	...

Fig. 3. Snapshot of the Resulting Matrix; the diagonal is removed

	ENSG00000000457.13	ENSG00000001461.16	ENSG00000004059.10	ENSG00000004142.11	ENSG00000004478.7
count	3872.000000	3872.000000	3872.000000	3872.000000	3872.000000
mean	0.577464	0.517657	0.527645	0.530568	0.543261
std	0.256664	0.240921	0.256842	0.225744	0.251979
min	-0.560534	-0.528495	-0.603606	-0.529907	-0.589222
25%	0.574681	0.504684	0.490520	0.535758	0.556207
50%	0.644750	0.583276	0.581144	0.590960	0.609373
75%	0.695956	0.639454	0.664639	0.631819	0.650543
max	0.854063	0.775522	0.886404	0.893947	0.970033

Fig. 4. Snapshot of the Description

It is immediately clear that most features are grouped within clusters of features that are closely inter-correlated. There are different approaches to pruning the graph, but the concern is whether we will lose critical information in the process. For time constraint, we proceeded with a conservative and an aggressive pruning.

The conservative pruning only prunes the leaves of the clusters, as well as one node from any two-feature pairs: these features should ideally represent a minimal information gain for our training. This method removed 175 features leaving 3,698 features. The selection is pickled (X_filtered_3698_leavesPruned).

The (very) aggressive pruning reduce every single clusters and pairs into single features. There isn't a very good way to measure how many (and what) to keep from each cluster given our time constraint and lack of knowledge in the biomedical field. This is largely done as an experiment or reference to the conservative pruning, and it leaves us 840 features, just under the number of sample we have. The selection is pickled (X_filtered_840_noCluster).

III. Prediction Models

In this project, we settled on Neural Network model from sklearn to predict age. From there we tested a few configurations of the neural network, and the layer layout are the follows (in tuples): [100,] (sklearn default), [100,100,100], [300,200,100]. Other less successful models attempted are not included in this report.

The average scores with 80-20 data split over ten times are the following. We do not have access to the actual age of the Hackathon's test dataset.

X Selection	[300,200,100]	[100,100,100]
4K	0.1144577	0.0869418
3698_NoLeaves	0.0568147	0.0477383
840_noCluster	0.0293734	-0.00392742

Unfortunately, this shows that the graph reduction processes were counter-productive. We picked the best performing configuration and used the model to predict the provided test dataset and submitted the result thusly. The actual performance is marginally worse than "random guessing" (MSE <172.15) at 190.2, according to feedback.

IV. Conclusions

While some distance from the winning team's performance, this is nevertheless a great practice experience that solidified our understanding of the data processing and prediction.