A Quantitative Test for SARS-Cov-2 Infection

Zefeng Xu

Data Science Institute (DSI), Brown University

GitHub: https://github.com/GunnerForever/1030_project.git

1. Introduction

The world has been plagued by the Coronavirus Disease 2019 (COVID-19) pandemic that originated in Wuhan, China in December 2019 for 4 years now. The disease is caused by infection with a strain of coronavirus known as Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2)[1]. Testing (such as Nucleic Acid Tests on infected secretions[1]) for infections of COVID-19 is critical for treatments and medications used. This project aims to add a quantitative approach to the testing and diagnosis of COVID-19 by using patient gene expression data.

1.1 Data Source

The dataset used in this project is publicly available in the Gene Expression Omnibus (GEO) database supported by National Center for Biotechnology Information (NCBI)[2] with dataset ID GSE212041[3]. This dataset was the supplementary data for the study conducted by Lasalle et al. (2022)[4], and was constructed by Neutrophil Bulk RNA-seq enriched from fresh whole blood drawn from patients at the start of the pandemic[4].

1.2 Data Description

This dataset is longitudinal, including 781 samples from 306 hospitalized COVID-19 positive patients, 78 symptomatic controls[1], and 8 healthy controls from a window of 3 timepoints: Day 0, Day 3, and Day 7 post-hospitalization[3]. Each sample can be identified by a unique ID of format [patient id]_D[time point], with the exception of healthy samples with ID

[1] These numbers are inconsistent with reality. Actual numbers are 304 COVID-19 positive patients, 76 symptomatic controls (386 possible IDs minus 6 patients without any data, which will be mentioned in Section 2.3), and 8 healthy controls.

of format H[1-9]_H. Each sample consists of values for 60640 features where each feature is a unique human gene identified by Ensembl IDs[2], and the corresponding value is its normalized expression level in unit of transcripts per million (TPM). The dataset provides two sets of labels for each sample: covid-19 status and acuity.max, with values shown in Table 1. The target labels of this project are the covid-19 status labels. Since these labels are binary, this project is a binary classification problem.

Patient Group	covid-19 status	acuity.max
COVID-19 Positive	1	1 to 5
Symptomatic Control	0	1 to 5
Healthy Control	0	H

Table 1: Summary of values of label sets. acuity.max labels range from 1 to 5 inclusive and a higher value means more severe symptoms.

2. Exploratory Data Analysis (EDA)

The Exploratory Data Analyses were performed on original dataset without any preprocessing. Below are three major insights from EDA.

2.1 Insight 1 – No Negative Entries

All features of this data are non-negative because all entries are expression levels, which are normalized counts but counts can never be less than zero.

2.2 Insight 2 – Duplicated Samples

This dataset contains three different types of duplicated samples.

- Type 1 Duplication

For some patients, there are two samples (A and B) at a timepoint.
There are 4 such cases in the dataset: e.g., (43_D7A, 43_D7B).

[2] Ensembl IDs: identifiers for genes as per the Ensembl (European Bioinformatics Institute and the Wellcome Trust Sanger Institute) database[9].

- Type 2 Duplication

For some patients, there exists a DE timepoint apart from D0, D3, and D7. Such timepoints correspond to changes in patient status (e.g. intubation, removal from ventilator)[3]. There are 43 such DE samples.

- Type 3 Duplication

This is a combination of Type 1 and 2 duplications. There is only one such pair in the dataset: (302_DEA, 302_DEB).

2.3 Insight 3 – Missing Samples

This dataset exhibits a massive number of missing samples. There are in total 386+8=394 patients[3]. Note that six patients with IDs 141, 142, 183, 202, 269, and 270 do not have any samples. Excluding DE samples, healthy controls, who only have data from one timepoint, and counting A/B samples as one, the information about missing samples is summarized in Table 2. We can see only 121/380 patients have "full" data.

Timepoints with Data	Number of Patients
1	160
2	99
3	121
Total	380

Table 2: Summary of missing samples.

3. Methods

This section focuses on two general topics: data preprocessing, which includes approaches adopted to make the original dataset suitable for machine learning, and the cross-validation pipeline, which includes the models trained and hyperparameters tuned.

[3] Patient IDs range from 1 to 386, including both COVID-19 positive and symptomatic control patients. Healthy patients do not have such an integer ID.

3.1 Preprocessing

- Sample removal

All duplicated samples mentioned in Section 2.2 and healthy controls were removed to avoid potential confusion.

- Missing samples ignored rather than imputed

There are two major reasons: first, the missing samples are also missing corresponding target labels but imputation of target labels is impossible; second, there does not exist any biologically reasonable method to impute genomics data.

- Transformed onto log scale & Dimensionality reduced

All entries were transformed onto the logarithm with base 2 scale by first adding 1 and then take log2. Also, the original number of features (60640) is too many for machine learning. This project extracted the top 300 Highly-Variable Genes (HVG) by adopting methods from Scanpy package[5][6]. HVGs are genes whose expression levels exhibit significant variations across samples. These variations may be indicative of disease states and biological characteristics.

- Columns normalized

Each column was normalized by removing the mean and scaling to unit variance using StandardScaler from scikit-learn package[7].

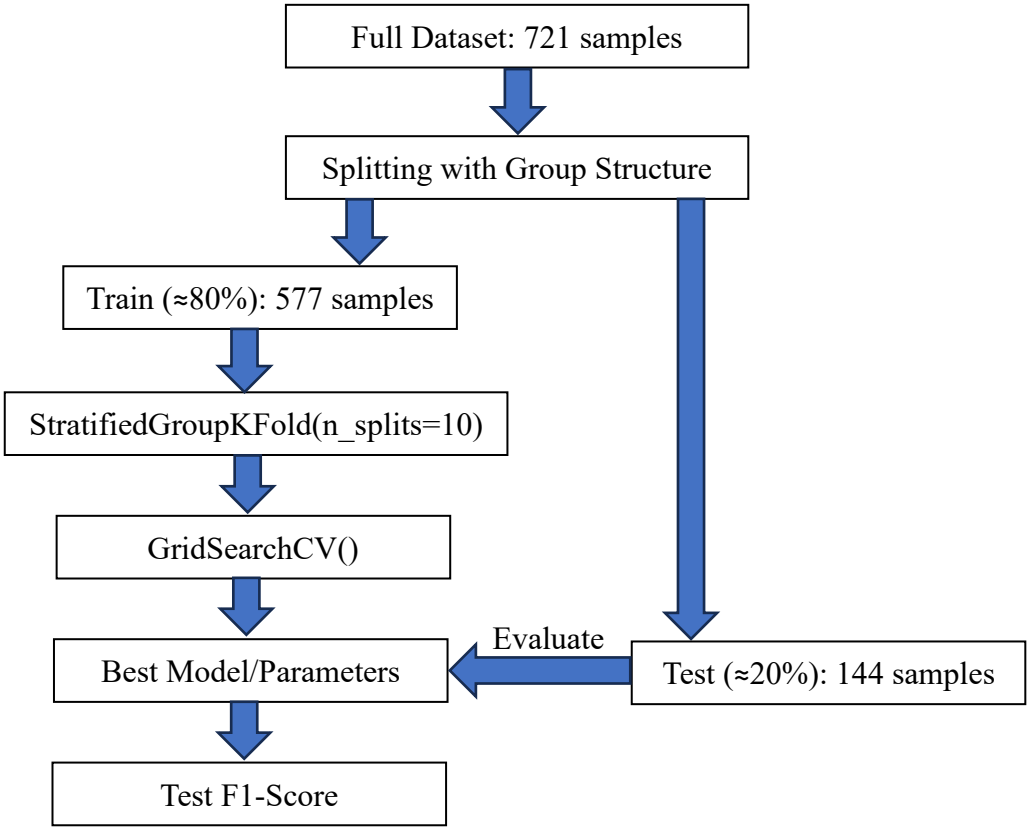
The resulting dataset had 721 samples and 300 columns after preprocessing.

3.2 Cross Validation Pipeline

In order to respect the group structure of patients, group-based splitting strategies from scikit-learn[7] were adopted. F1-Score was used as evaluation metric because this dataset exhibits massive class imbalance[4].

[4] Class 1 has 642 samples. Class 0 has only 79 samples.

Below is a flow chart for the cross-validation pipeline used. The pipeline is repeated with 10 different random states for each machine learning algorithm.



3.3 Models Trained and Hyperparameter Tuned

The models trained and hyperparameters tuned for each model are shown in Table 3.

Table 3 Panel A	K Nearest Neighbor (KNN)
Tuning Round	Tune 1
Parameters Tuned	n_neighbors
Values	n_neighbors: 3 to 14 inclusive

Table 3 Panel B	Logistic Regression	
Tuning Round	Tune 1	Tune 2
Parameters Tuned	penalty C	l1_ratio C

Parameters Controlled	solver='saga'	penalty='elasticnet' solver='saga'
Values	penalty: [None, 'l1', 'l2'] C: [0.01, 0.1, 1, 10]	l1_ratio: [0.1, 0.3, 0.5, 0.7, 0.9] C: [0.01, 0.1, 1, 10]

Table 3 Panel C	Support Vector Machine		
Tuning Round	Tune 1	Tune 2	Tune 3
Parameters Tuned	C	kernel C gamma	C degree gamma
Parameters Controlled	kernel='linear'		kernel='poly'
Values	C: [0.01, 0.1, 1, 10]	kernel: ['rbf', 'sigmoid'] C: [0.01, 0.1, 1, 10] gamma: ['scale', 'auto', 1e-3, 1e-2, 1e-1, 1, 10]	C: [0.01, 0.1, 1, 10] degree: [3, 4, 5, 6, 7] gamma: ['scale', 'auto', 1e-3, 1e-2, 1e-1, 1, 10]

Table 3 Panel D	Random Forest	eXtreme Gradient Boosting (XGBoost)
Tuning Round	Tune 1	Tune 1
Parameters Tuned	max_depth criterion	max_depth learning_rate

Values	max_depth: [5, 10, 15, 20, 25, 30] criterion: [‘gini’, ‘entropy’, ‘log_loss’]	l1_ratio: [0.1, 0.3, 0.5, 0.7, 0.9] C: [0.01, 0.1, 1, 10]

Table 3 Panels A,B,C,D: Hyperparameters tuned for five algorithms

4. Results

4.1 Test F1-Scores

Table 4 summarizes the test F1-Scores. The mean baseline F1-Score over 10 different random states is 0.941829, which is the F1-Score if we always predicted the majority class.

Table 4 Panel A	K Nearest Neighbor (KNN)
Tuning Round	Tune 1
Test F1-Score Mean	0.946617
Test F1-Score Standard Deviation	0.011775

Table 4 Panel B	Logistic Regression	
Tuning Round	Tune 1	Tune 2
Test F1-Score Mean	0.949457	0.951053
Test F1-Score Standard Deviation	0.011064	0.010449

Table 4 Panel C	Support Vector Machine (SVM)		
Tuning Round	Tune 1	Tune 2	Tune 3
Test F1-Score Mean	0.947214	0.950565	0.948160

Test F1-Score Standard Deviation	0.009894	0.007410	0.012846

Table 4 Panel D	Random Forest	eXtreme Gradient Boosting (XGBoost)
Tuning Round	Tune 1	Tune 1
Test F1-Score Mean	0.952527	0.948778
Test F1-Score Standard Deviation	0.014333	0.013943

Table 4 Panels A,B,C,D: Summary of test performance of the five algorithms

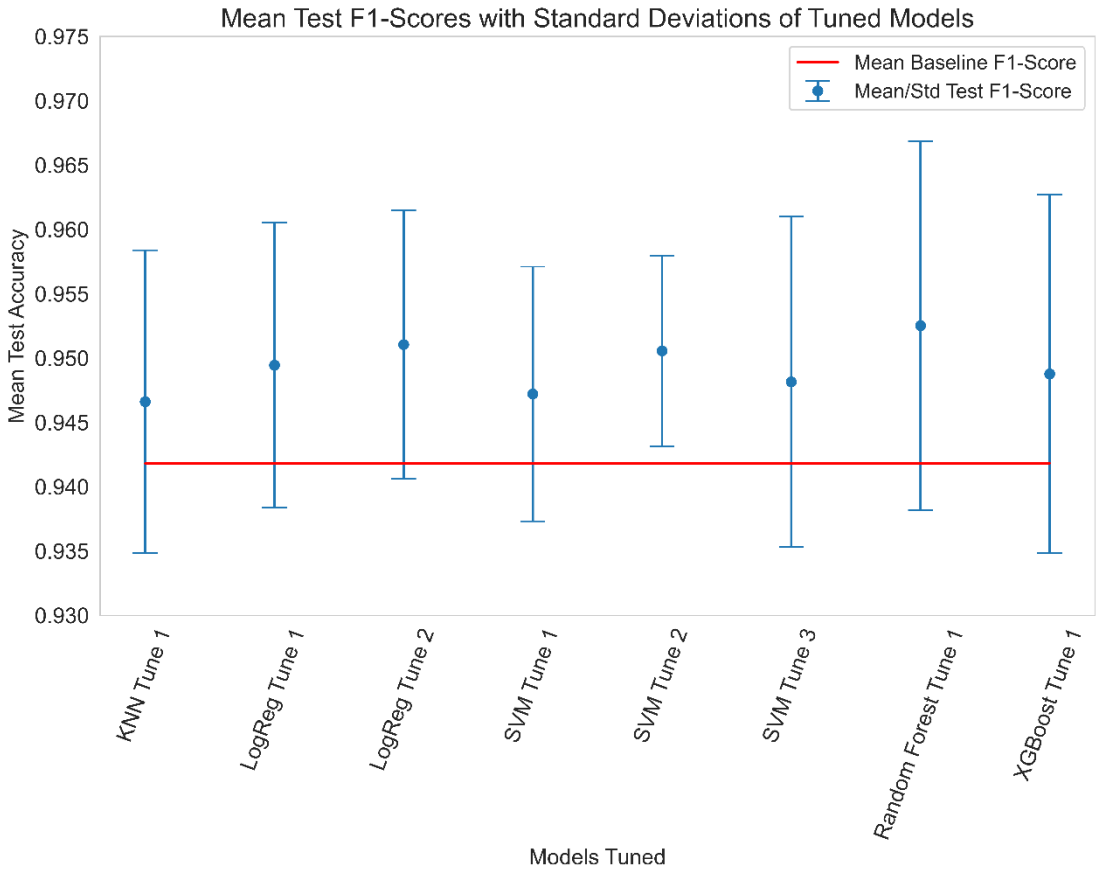


Figure 1: Plot of information in Table 4

As can be seen from Figure 1, all the tuned mean test F1-Scores are higher than the baseline F1-Score with roughly 0.5 to 1 standard deviations.

However, after adding the uncertainties, almost all models can underperform the baseline one. The best-performing model is Random Forest Tune 1 in terms of test F1-Score mean but SVM Tune 2 is the one with lowest standard deviation.

4.2 Global Feature Importance

This section focuses on interpretation of the models. Three different metrics for global feature importance were calculated, as shown below.

4.2.1 Permutation Score

This project experimented with Permutation Scores using Random Forest. The top 20 features sorted by permutation score are shown in Figure 2.

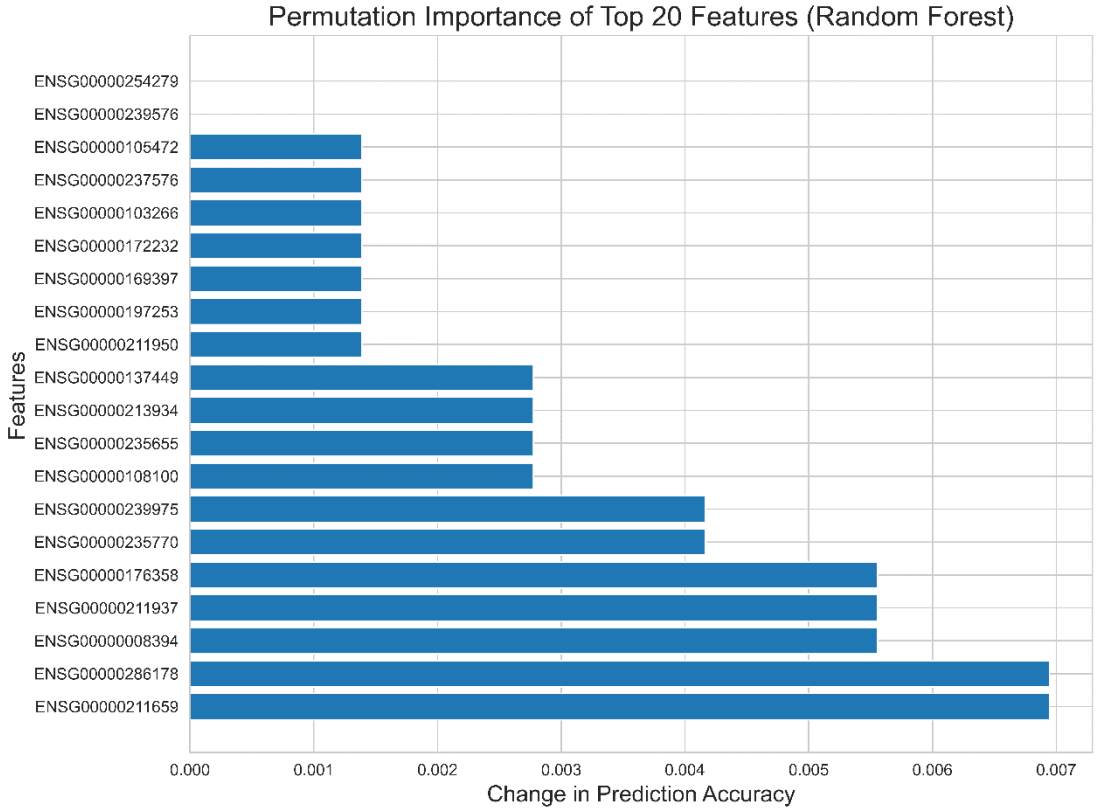


Figure 2: Top 20 features sorted by Permutation Importance

4.2.2. Regression Coefficients

The coefficients of Logistic Regression can also be viewed as a measure of feature importance. Figure 3 shows the top 20 features sorted by magnitude

of coefficient regardless of the sign. A positive/negative coefficient implies a positive/negative impact on the log-odds (and thus the probability) of predicting the positive class.

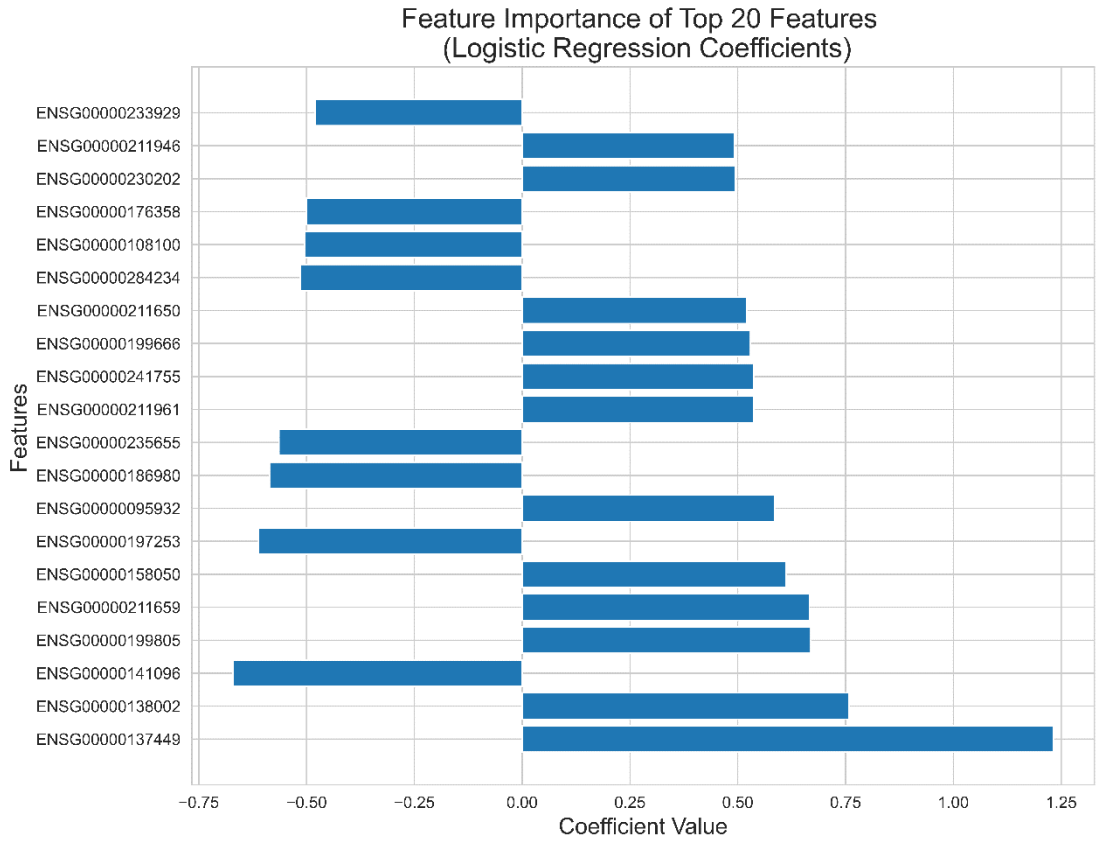


Figure 3: Top 20 features sorted by magnitude of coefficient in Logistic Regression

4.2.3 Random Forest Feature Importance

This project also experimented with the built-in impurity-based feature importance of Random Forest. This metric measures the reduction of impurity, which is a measure of how mixed the classes are in the dataset, when using a specific feature in construction of a random forest. Figure 4 shows the top 20 features.

4.2.4 Implications

Since all features are human genes, a subset of "important" features is a subset of human genes that has some quantitative meaning for the target label, which represents disease status. Particularly important features

(genes) (such as the gene with Ensembl ID ENSG00000211659, ranked as 1st, 5th, 4th in top 20 genes selected by the three metrics respectively) may have closer relationship with the disease. These selected genes, combined by taking union from different selection metrics, can be given to biologists to study their biological relationships with the disease, such as COVID-19 for this project.

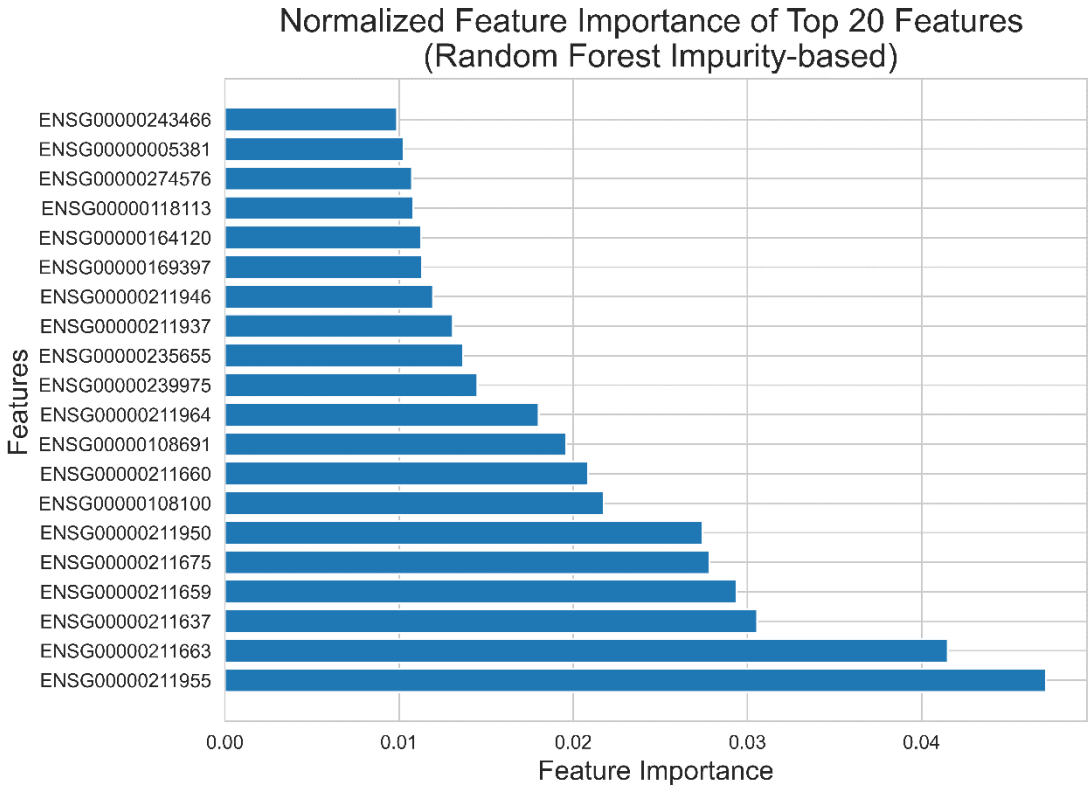


Figure 4: Top 20 feature sorted by Random Forest's feature importance metric

4.3 Local Feature Importance

As opposed to global feature importance, to study the impact of the features on single observations, SHAP[9] values were calculated. Figure 5 shows the top 20 features sorted by SHAP values for the testing sample at index 10 (randomly picked). As can be seen, the particularly important gene (ENSG00000211659) mentioned in Section 4.2.4 is still important for this specific sample (ranked as 11th out of 20).

5. Outlook

The issue of missing/lack of data must be properly addressed since the number of samples can have direct impacts on model performance. Looking for datasets with more, or at least less missing, data could be helpful. In addition, the samples removed in Section 3.1 could have been used, but the way of using them is subject to further consideration. The issue that the models did not well outperform the baseline might be alleviated by proposing a strategy that stratify samples according to target labels while still keeping group structure during initial splitting. Also, there might be some models specifically targeting biomedical noisy data, and experimenting with them could possibly be beneficial.

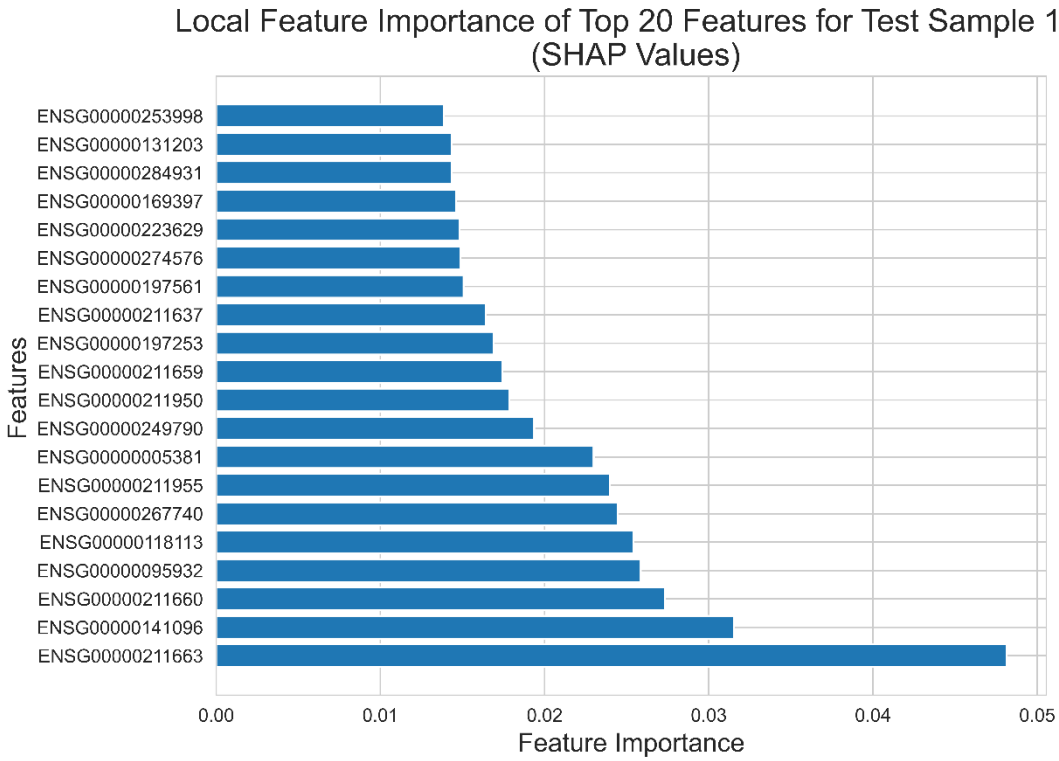


Figure 5: SHAP values of top 20 features for test sample at index 10

6. References

1. COVID-19: https://en.wikipedia.org/wiki/COVID-19
2. The Gene Expression Omnibus database: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4944384/#:~:text=Ge

ne%20Expression%20Omnibus%20(GEO)%20is,studies%20of%20high%2Dthroughput%20gene

3. GSE212041
   https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE212041
4. Original article where this dataset is supplementary data
   LaSalle TJ, Gonye ALK, Freeman SS, Kaplonek P et al. Longitudinal characterization of circulating neutrophils uncovers phenotypes associated with severity in hospitalized COVID-19 patients. Cell Rep Med 2022 Oct 18;3(10):100779. PMID: 36208629
5. Scanpy package: https://scanpy.readthedocs.io/en/stable/index.html#
   Wolf, F., Angerer, P. & Theis, F. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol 19, 15 (2018). https://doi.org/10.1186/s13059-017-1382-0
6. anndata: Annotated data
   Isaac Virshup, Sergei Rybakov, Fabian J. Theis, Philipp Angerer, F. Alexander Wolf
   bioRxiv 2021 Dec 19. doi: 10.1101/2021.12.16.473007.
7. scikit-learn package: https://scikit-learn.org/stable/index.html
   Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
8. SHAP (SHapley Additive exPlanations)
   Lundberg, S., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. https://doi.org/10.48550/arXiv.1705.07874
9. Ensembl Gene ID: https://www.wikidata.org/wiki/Property:P594