Severity Prediction of COVID-19 on Chest X-Ray Images

Mentor: Dr. Cao

Team Members:

Luke Robbins, Yeyun Xu, Jeremiah Joseph, Aatifa Khan, Justin Pham

Abstract:

Due to the spread of Covid-19, radiologists are consistently tasked with diagnosing Covid-19 lung infection severity on large batches of X-ray images. To do this, some departments use a rating system from 1-16 called the Brixia Score. We have focused on using classical modeling techniques to predict the infection severity to a reasonable degree, and create a load-lightening tool for Covid radiologists. Using U-Net architecture to segment the lungs from the X-ray images, we are able to extract 109 relevant features. We use these features to predict infection severity on a condensed score of light infection (1-5), mild infection (6-11), and severe infection (12-18). The three best performing models were ridge regression (57.6% CV Acc., 53.8% Test Acc.), Random Forest (55.2% CV Acc., 60.1% Test Acc.) and XGBoost (60.1% CV Acc., 58.9% Test Acc.).

TABLE OF CONTENTS

INTRODUCTION

Covid-19 has overwhelmed our healthcare systems in many ways over the last few years, and the increase in patients has led to significantly more chest X-rays related to Covid infection passing through radiologist's hands. Unfortunately, this means that severe cases of infection can often be diagnosed after the ideal window, as the radiologist must sift through X-rays in the order they are received rather than severity. Our group has worked on a model to predict lung severity (Brixia score) based on several features extracted from chest X-rays, and attempts to help the efficiency of the radiologist. Using neural net models for lung segmentation and feature extraction, and several classical modeling strategies, we can predict the severity of a Covid-19 infection. This is a potential solution to the current bottleneck in radiology infection detection, as the radiologist would have severe X-rays brought to their attention for inspection and verification. Currently there is at least one deep learning model implemented in the industry for this purpose, but we aim to test the accuracy of several more classical methods. Generally, deep learning models require more resources to train and upkeep, whereas more traditional methods may be more efficient.

COMMUNICATION PLAN

Our group opted to meet digitally and communicate through GroupMe, MS Teams, and Discord. This allowed us to meet during scheduled times, and separately when needed. We also used Discord and Google Drive for file storage, so that our files would be accessible from anywhere. As a group we met for 1-2 hours every wednesday and friday morning, as well as random times when needed.

We were able to communicate with our mentor, Dr. Yan Cao, through MS Teams. Our scheduled meeting time was every week on Friday at 3:30 pm.

TIMETABLE

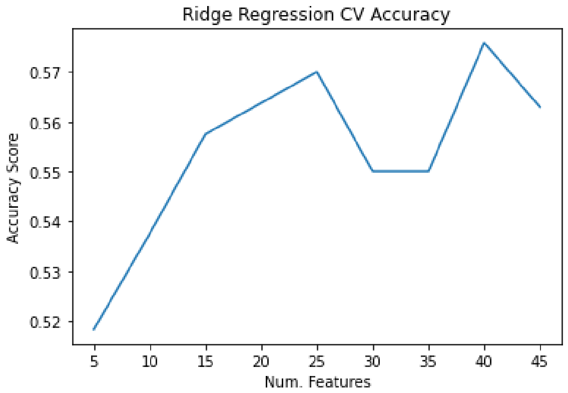
Our major milestones are as follows:

- Create a working, accurate lung segmentation model for data pre-processing (2/25/22)
- Implement feature extraction library on our data for data processing (2/25/22)
- Create T-SNE / PCA visualizations with processed data (3/25/22)
- Select models for further exploration of classification (3/25/22)
- Train models and explore accuracy and results, including improvements as needed (4/1/22)
- Refined model functional and implemented on test data (4/22/22)

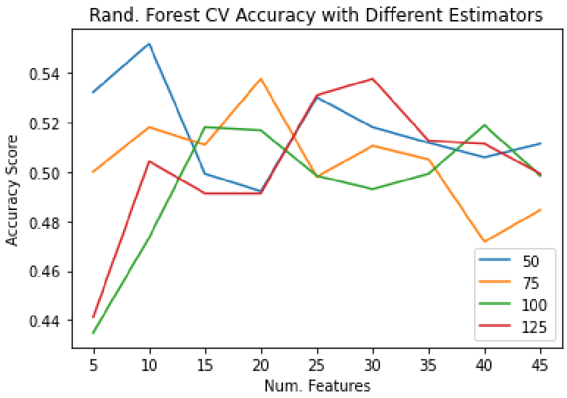
EVALUATION

The success of our models is measured in accuracy of prediction. The measurable sections of our project were lung segmentation and modeling. For the lung segmentation model, we measured success by comparing the output mask to the original X-ray visually, and through a metric called IoU (Intersection of Union). For the modeling stage, we iterated through several parameters and found both CV (Cross Validated) accuracy and prediction accuracy on new test data.

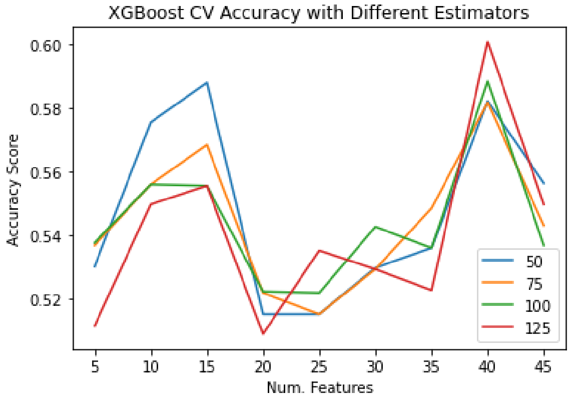
Ridge Regression (57.6% CV, 53.8% Test)



Random Forest (55.2 % CV, 60.1% Test)



XGBoost (60.1% CV, 58.9 Test)



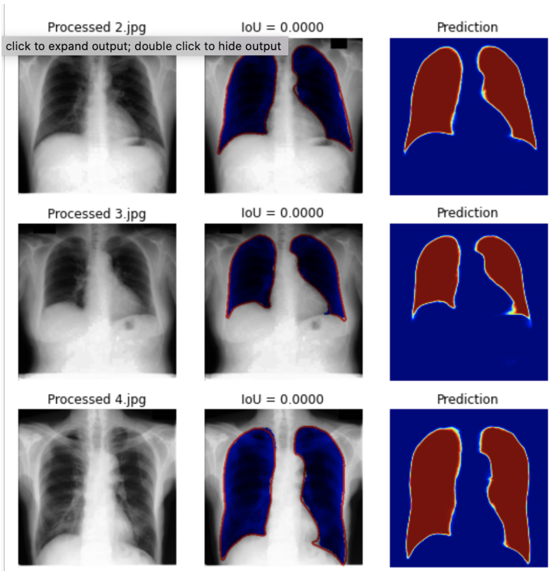
Overall, the XGBoost cross-validated the best, meaning the model will likely generalize better than the others. The XGBoost model also had a high test accuracy score.

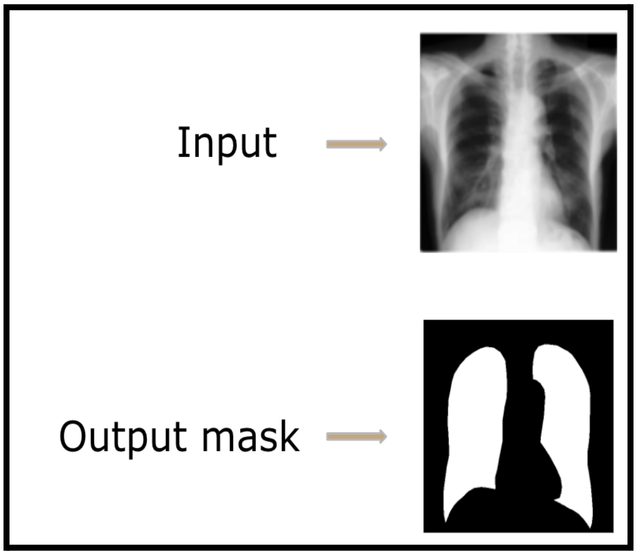
IMPLEMENTATION DETAILS

As mentioned above, the goal of the project was to create a machine learning model capable of predicting the severity of Covid based on a chest xray image. In order to do this, we used labeled chest X-rays. The label score had a range of 0 to 16 that we divided into 3 sub-classes: light (1-5), mild (6-11), and severe (12-16). From there, we used classical machine learning techniques to attempt to predict the correct corresponding class. The implementation consisted of the four following phases:

Lung Segmentation:

The first phase of our project was lung segmentation. When determining the Brixia score, the radiologist looked for certain properties within the lung. Due to this, if we had machine learning algorithms assess an entire chest X-ray it would pick up noise in the data that would not be useful in determining the severity of the infection. This made it extremely important to initially find the part of the chest X-ray that contained the lung, and separate them from the rest of the image. This process involved the use of the U-net Architecture, a convolutional network made for the segmentation of biomedical images. With the help of this specific architecture, we were able to map each pixel in our image to a probability that the image was in the lung. We then took every pixel that had a probability over 50% and used that as the points where we predicted the lungs would be. The below images show a few of the results of this technique. By doing this, we ended up with an output mask for all the X-rays in our dataset that was able to be used in the other phases of our project to determine where the lungs were.



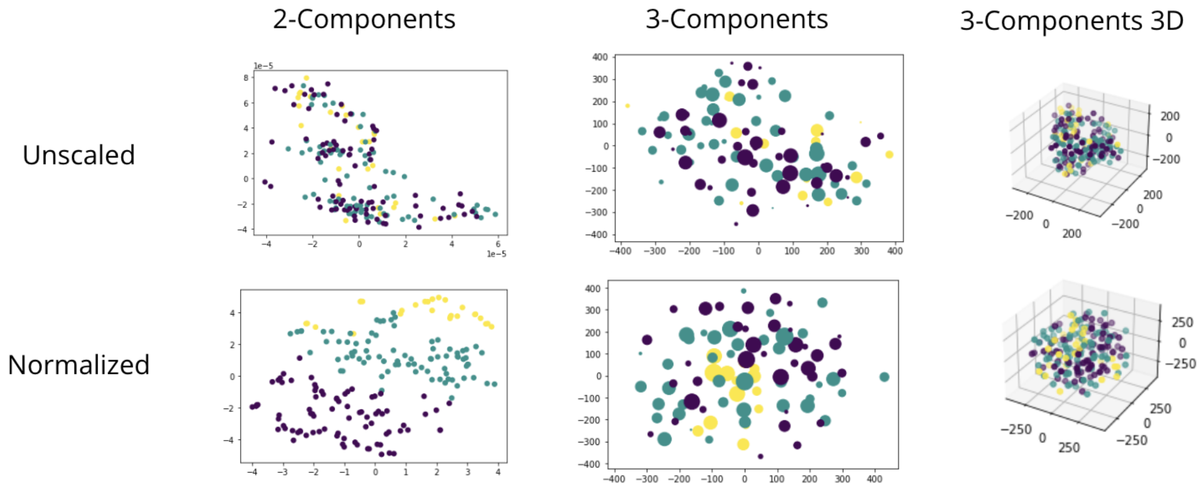


Feature extraction:

The second phase of our project was feature extraction. The purpose of feature extraction was to extract mathematical values from x-ray images and their masks to be used in classification models. The basis for extracting the features that we did extract heavily involves dense mathematical properties that were out of the scope of our project to completely understand. However, a certain useful python library called PyRadiomics was used for the feature extraction process. Pyradiomics uses texture, shape, and grayscale values of the original photo and can perform image transformations, such as logarithmic, square, etc. in order to find more meaningful features. By writing a simple implementation of the tool, we can use imported x-ray images and masks that were created during the lung segmentation process for the Pyradiomics process. We then were able to export a large csv file with 108 features and 192 objects. With this in hand, we have all the tools necessary to create T-SNE visualizations and construct classification models.

T-SNE visualizations:

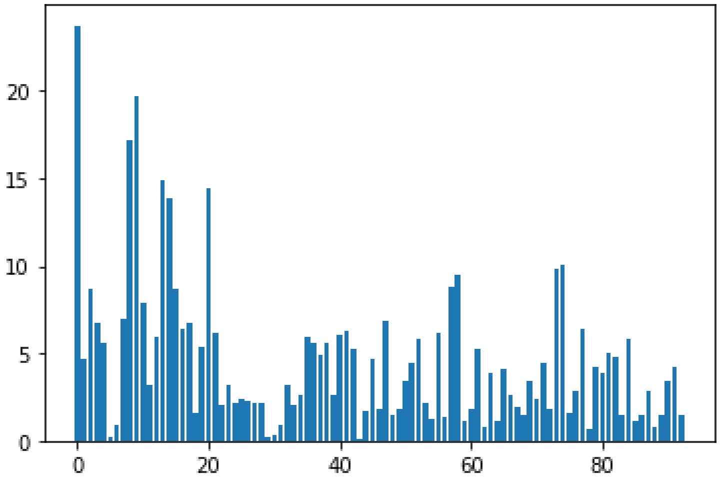
As mentioned earlier, the feature extractor resulted in a dataset with 108 features. With so many features it is extremely difficult to do any exploratory data analysis. T-SNE is an unsupervised technique that is used for data dimensionality reduction. By using a T-SNE plot, we would be able to visualize the data better and see if there were any clear patterns within our data or clustering of our three classes of Covid Severity. When initially running TSNE on our data, we saw there was no clear clustering of the three classes. We then made the observation that some features had very small values while others were much larger. This led us to the idea that normalization might help us better understand the trends in the data. After normalizing, we finally were able to get a graph that showed the clear distinction between the classes with 3 clusters. Below are both the normalized and unscaled T-SNE plots.

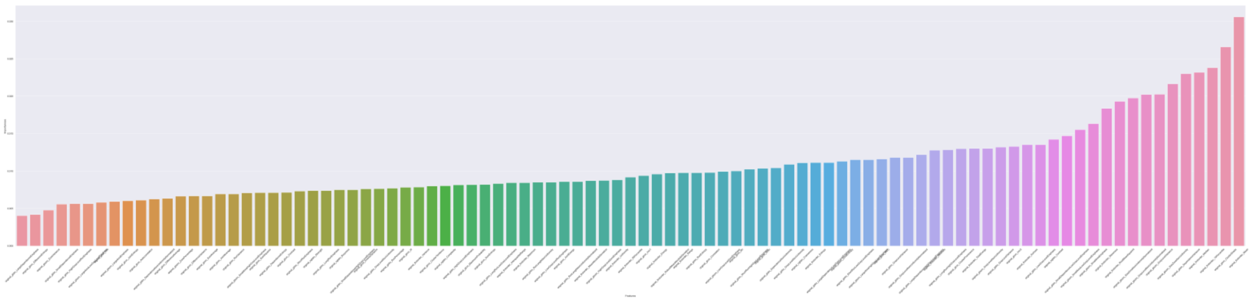


Model construction:

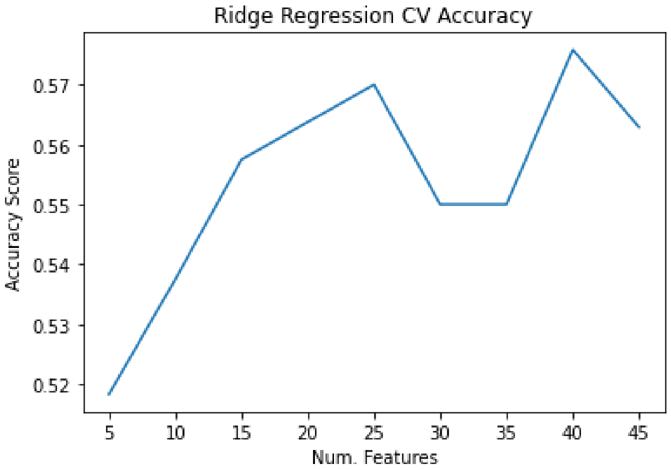
The final phase of the project was to implement the machine learning models. The goal of these models is to accurately classify x-ray images as mild (0), moderate (1), and severe (2) in regards to COVID-19 severity. The type of machine learning used in our models is supervised learning, where the x-ray images have already been assigned correct labels, and we compare our model's classification against the correct labels. We decided, due to the numerous research on deep learning models, to build using classical machine learning models. These models were implemented in python using the sklearn library. We first separated our data into 80% train and 20% test. We also used cross validation in order to evaluate the effectiveness of the models.

The first step we took in our model creation was to do feature selection. As mentioned earlier, the data set created from the feature extractor had 108 features. If there were features that did not have any relation to the COVID-19 severity , these features would add noise to models and thus worsen the results when fed new data. Due to this, we knew we had to find ways to limit the dimensions to where only the most important features were used. The first method we used to do this was ANOVA. By evaluating the variance and using an F test, this method is able to give a score on how important a variable was in determining our target variable. This allowed us to pick the most important features based on that evaluation. Another way that we chose to do feature selection was through random forest. With random forest we measured how each feature decreased impurity. By taking the mean decrease in impurity or Gini Index of each feature, we were able to rank the importance the features had to the target. The final method we used was PCA. PCA or Principal Component Analysis works by applying an orthogonal linear transformation to the data that transforms the data into a new coordinate space. This coordinate space has fewer dimensions, and can represent most of the variance in the data. Below are graphs of the ANOVA (graph in blue) and Random Forest(graph in color) ranking of features.

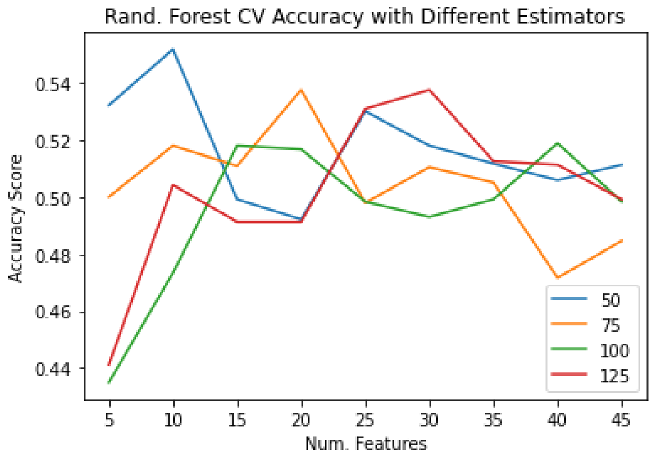




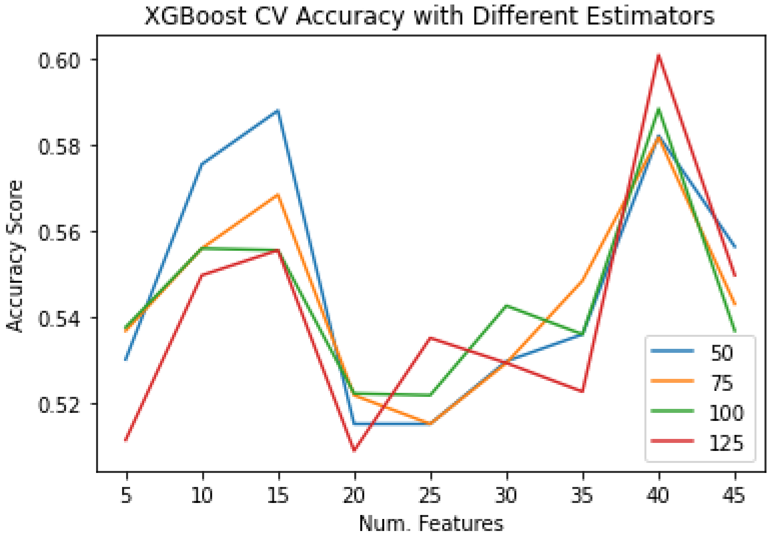
The next step was actually constructing the models. The first models we made were lasso and ridge regression. Both are regularization techniques. Lasso regression also does feature selection within the model as it can set some features to 0. This allows us to get a good benchmark for the other models. The best cross validation accuracy we got with lasso was 52.4%. With ridge regression we decided to use the anova feature selected models. We iterated through the amount of features we used in the model with a for loop. After doing this we saw that when we used the 40 most important features the cross validation accuracy maximized. We plotted this with the matplotlib library in python. The max cross validation accuracy was 57.6% and the test accuracy was 53.8%.



The next model that we chose to run was a random forest. Using for loops we iterated through various depths, number of features, and number of estimators. Our best results came when running a max depth of 6 with 10 features and 50 estimators; the results were a 55.2% cross validation accuracy and 60.1% accuracy on test data.



The final model and the model with the best results we had was XGBoost. XGBoost is very similar to Random Forest. A major difference is that XGBoost uses boosting while Random Forest uses bagging. For this method we also iterated through various numbers of features and different numbers of estimators and plotted the results. The max cross validation accuracy was achieved with 125 estimators and 40 features. This resulted in a cross validation accuracy of 60.1% and a test accuracy of 58.9%. Though the test accuracy was smaller than the random forest, because the cross validation was so much higher we decided to conclude this was the best model.



Here is a link to the github that contains all the code: https://github.com/lukevrobbins/covid_19_severity_prediction

ISSUES AND LESSONS LEARNT

As a group, we faced many hiccups and difficulties throughout the course of the project. Some issues took only a couple hours to fix, while others took many days to figure out a solution. For example, we had many issues with uploading and merging files on GitHub, since none of us had any real experience with it. We also faced many smaller issues like getting accurate masks from the lung segmentation, and getting the feature extractor to export to a readable csv, and general fine tuning of each model to get the highest accuracy possible with our set of data. We worked together to figure out issues in lung segmentation and feature extraction by performing independent research into the issue and trying out different solutions when we met up. For T-SNE visualization and model fine tuning, we sought help from Dr. Cao. She gave us a lot of good tips and examples of how to fine tune our models to get better results as well as helping us interpret our T-SNE visualizations. Overall, many of the issues that we faced are things that we see ourselves learning from and applying it to future work.

ETHICS DISCUSSION

In our project we used code from the documentations of U-Net and the documentation of PyRadiomics for the lung segmentation and feature selection respectively. There was not any hesitation of using it because this code is very prominently used in industry and has been heavily tested and proven to work well. All these sources have been documented in the references and sources.

Another ethical issue about our project occurs in the possible use of it. If there is an inaccurate prediction of the severity of Covid from our model and this result is used to not give resources to a patient who needs it, this could lead to very negative results. To combat this, before the model is used in practice modifications should happen to ensure errors are mostly over predicting severity instead of under predicting it. Also this tool could be used with the aid of humans as well. This could ensure that the mistakes of the model are quickly recognized, while allowing for the benefits of efficiency to still take place.

CONCLUSION

This project has been an overall success. We ultimately had accuracies in the project at around 60%, which is reasonable for classical methods. This accuracy however, is not feasible to be used in an actual medical setting. We believe that others can see trends with the parameters in our machine learning models and use it to conduct research to further enhance the technology. Overall, the implementation cannot be used in the field, but the findings of our project may be useful in creating further work in the field.

FUTURE WORK

As a future work, our results support that our data suggest that we still have a long way to go. The accuracy rate of the best model XGboosting algorithm we get is 60.1%, and we think there is still a lot of room for improvement. The group discussed that we can also improve our accuracy by:

1. increasing the model size - this is very effective when a weak model becomes a strong model through integration learning.
2. modifying the model architecture - in neural networks, different architectures have a greater impact on the results, so we also need to adjust and adapt more specifically to the dataset.
3. reducing or removing regularization - The purpose of regularization (L1, L2, Dropout) is to keep the model from overfitting the dataset, but if the training set is small or the number of model iterations is not enough, the model cannot iterate to a better local minimum, so you should reduce some of the regularization to improve the fit and reduce the bias, but also increase the variance.

CONTACT INFORMATION

Luke Robbins – LVR19000@utdallas.edu – Data Science major, Senior

Jeremiah Joseph – jsj180002@utdallas.edu - Mathematics/Computer Science, Junior

Justin Pham – jhp180004@utdallas.edu, Data Science major, Senior

Yeyun Xu – YXX170004@utdalllas.edu - Data Science major, Senior

Aatifa Khan – AXK180019@utdallas.edu - Data Science major, Senior

RESOURCES & SOURCES

Harvard Medical School(2017), Pyradiomics, *GitHub.*[Resource Code] https://github.com/AIM-Harvard/pyradiomics

imlab-uiip(2020), Lung-segmentation-2d, *GitHub.*[Resource Code] https://github.com/imlab-uiip/lung-segmentation-2d

L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9(Nov):2579-2605, 2008

Raghavian(2020) LungVAE, *GitHub*. [Resource Code] https://github.com/raghavian/lungVAE/

Print Name/Signatures/Date: Company mentor, faculty advisor, and each team member should read and agree by signing this document, and submit an electronic version (PDF/DOC) through elearning.

Luke Robbins 5/4/22	Justin Justin Pham 5/4/22
Jeremiah Joseph 5/4/22	Aatif Khan Aatifa Khan 5/4/22
Yeyun Xu. Yeyun Xu 5/4/22	Yan Cao Dr. Cao 5/5/22