Predicting Immunosuppression …….

Annie Dai, Charles Fu, Isaac Lucero, Shaan Sheth, Ryan Shikiya, Tanmay Vijaywargiya
University of California, Berkeley

IEOR 142 Final Project

Motivation

In the past two years, the world has endured a pandemic no one could have foreseen coming, and the question of how we could have better prepared has continuously come to mind. The last major pandemic was the Spanish flu which was in 1918 where one third of the world was infected and as high as 100 million people were estimated to have died. Despite COVID happening over 100 years later, and with all the advancements in healthcare and technology, this pandemic showed us how ill equipped we still are at dealing with infectious diseases on this scale. COVID caused huge disruptions all over the world with countries outlawing travel, entire economies shutting down, and the still very apparent deaths that occur everyday from the virus. One of the main health issues that arose very early in the outbreak was the risk pertaining to people who had existing conditions prior which included cancer, diabetes, high cholesterol, etc. The CDC reported that only 6% of all COVID related deaths had the virus as the only medical condition meaning the other 94% had some other medical condition that contributed to their death. In fact, on average, people who died had around three health conditions on their death report. With this project we aim to better understand and predict the likelihood of an individual becoming immunosuppressed with the intention of making it easier to identify individuals who could be at higher risk for potential wide scale diseases in the future.

Immunosuppression is the state in which the immune system is compromised and unable to effectively fight off infections and diseases, which can be caused by a variety of factors, including certain medical conditions, medications, and treatments. Predicting immunosuppression can be important for a number of reasons. For example, individuals who are at high risk of immunosuppression may benefit from regular check-ups and monitoring by a healthcare provider. This can help to catch any potential issues early on and allow for timely intervention. Beyond just predicting an individual's likelihood of being immunosuppressed for potential pandemics, knowing also aids in a multitude of other areas concerning health. Knowing an individual's likelihood of immunosuppression can help healthcare providers to make informed decisions about their treatment and care as well as contributing to help researchers develop new treatments and therapies that can prevent or reverse this condition. By using machine learning to analyze past health data, we can identify patterns and trends that may be relevant to immunosuppression and use this knowledge to improve the health and well-being of individuals at risk on a large scale.

Data

When we began looking for data to help aid in our search to predict immunosuppression we wanted to look for a dataset that was very comprehensive and included many different methods of looking at health. We decided to use The National Health Interview Survey (NHIS) for 2021. The NHIS data is collected through personal household interviews on a broad range of health topics. The system is designed to integrate multiple indicators from many data sources to provide a comprehensive picture of the public health burden of CVDs and associated risk factors in the United States. The data are organized by location (region) and indicator, and they include CVDs (e.g., heart failure) and risk factors (e.g., hypertension). At first, before any cleaning, the data was 29482 rows by 622 columns with many null values. So, the first thing we did was each member of the team took 5 pages from the codebook summary and carefully decided which variables would make the most sense to include in our research. The first few columns were questions like weight, region, and year of survey, but the vast majority of columns were more so questions relating to different aspects of health. These included if the individuals had histories of diseases, if they were testing for certain diseases, and if they engaged in certain behaviors. Once we extracted the most relevant variables to our dependent variable, we eliminated features which contained more than 53% (large jump from 53% to 63% in our features) null values. With the remaining columns that still contained nan values we decided to change them to the "don't know response", which numerically was 9. We also then created dummy variables when necessary. For the undesirable responses "refused", "not ascertained", "don't know" (7, 8, 9) for our y label (value of dependent variable), we changed them to the baseline response of “no” (0). Thus, now the meaning of our dependent variable was now either 0 for not immunosuppressed or 1 for immunosuppressed. Finally, we split our data into X_train, y_train, X_test, and y_test sets with 70% of the data in the training set and the remaining 30% in the test set as well as isolating our dependent variable, 'HLTHCOND_B'.

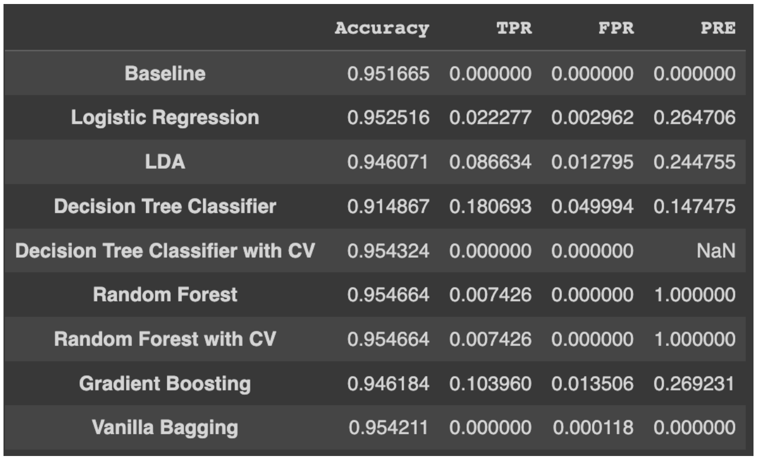
Analytical Models

Since the goal of our analysis is to predict whether or not someone is immunosuppressed, we used a subset of models that could handle classification rather than predicting a range of numerical values. In this section, we will review the several methods used, their results, and how we selected and improved upon the best ones.

We began with a baseline model that predicts the majority value of the dependent variable in the training set. In this case, our baseline predicted 0 (not immunosuppressed) for all records in the test set. This resulted in a baseline test accuracy of 0.9517. While this accuracy is quite high, it’s not entirely surprising because we had a significantly larger number of individuals that were not immunosuppressed in our dataset. Furthermore, while this accuracy is good, simply predicting not immunosuppressed for all individuals is not a good method in this context. This is because it will lead to a very high false negative rate (FNR) which can have serious consequences in a medical setting.

After this, we transitioned to a logistic regression model with a threshold probability of 0.5. This resulted in an accuracy of 0.9525 and a true positive rate (TPR) of 0.0223. Again, in this context, it is important to not only maximize accuracy but also TPR (analogous to minimizing FNR).

We continued training, fitting, and testing various other models to determine which one performed best on the test set. In addition to logistic regression, we fit linear discriminant analysis (LDA), decision tree classifier (CART), CART with cross validation (CV), random forest, random forest with CV, gradient boosting, and vanilla bagging models. To perform cross validation and tune parameters where possible, we made a grid of possible values for the hyperparameter, and set the CV parameter using KFold with 5 splits. Finally, GridSearchCV was used to generate and fit the model. When testing the models, we selected the model with the best_params_ attribute. Figure 1 displays a summary of the test statistics calculated for each of the models we trained.



	Accuracy	TPR	FPR	PRE
Baseline	0.951665	0.000000	0.000000	0.000000
Logistic Regression	0.952516	0.022277	0.002962	0.264706
LDA	0.946071	0.086634	0.012795	0.244755
Decision Tree Classifier	0.914867	0.180693	0.049994	0.147475
Decision Tree Classifier with CV	0.954324	0.000000	0.000000	NaN
Random Forest	0.954664	0.007426	0.000000	1.000000
Random Forest with CV	0.954664	0.007426	0.000000	1.000000
Gradient Boosting	0.946184	0.103960	0.013506	0.269231
Vanilla Bagging	0.954211	0.000000	0.000118	0.000000

Figure 1

After looking at the performance metrics of all our models on the test set, we found our Decision Tree Classification (DTC) model appeared to perform the best given the focus of our problem. Since our focus for the problem is a medical application, we thought it was important to have a balance between high accuracy and a high true positive rate (TPR). This is because when predicting whether someone is immunosuppressed, it seems there is a higher loss associated with not diagnosing a patient with immunosuppression when they actually are immunosuppressed (false negative) compared to diagnosing the patient with immunosuppression when they actually aren't (false positive). So, our DTC model with the highest TPR at 0.1807 and relatively high accuracy of 0.9149 seemed to produce the best performance for our chosen application. Additionally, we thought this was a good model because of its interpretability. With this choice of model, we next decided to determine how confident we were in the model by performing bootstrap validation on its accuracy and TPR performance metrics. From this analysis, we found we could be relatively confident in the aforementioned values because the bootstrap validation appeared to produce low variance in both metrics. In particular, with 500 bootstrap samples, we observed the accuracy only varied from about 0.905 to about 0.923, and the TPR only varied from about 0.11 to about 0.19 as seen in Figure 2.

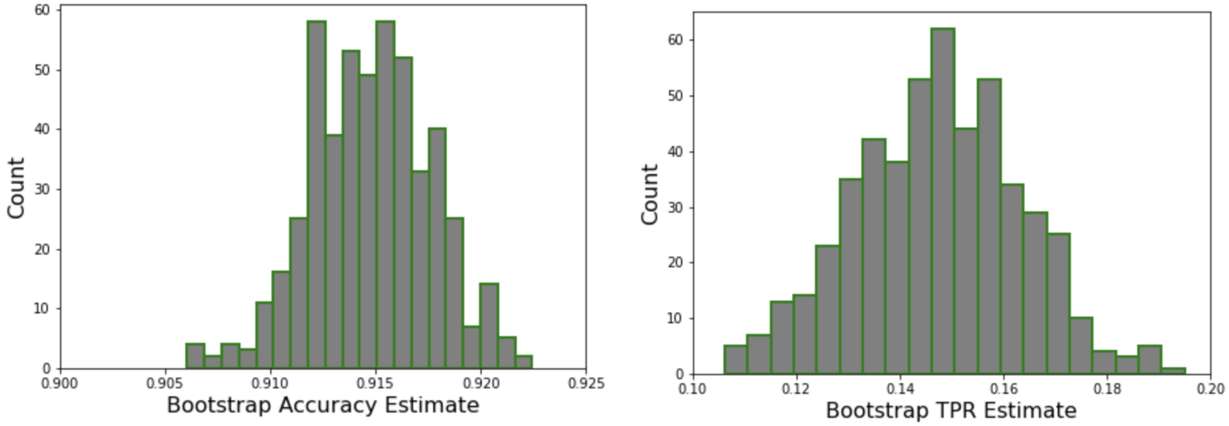


Figure 2

With a model we were confident in and our focus now turned to TPR, we wanted to see if we could improve the TPR of our model without decreasing the accuracy too much. To do this, we first tried cross validation to optimize the ccp_alpha parameter of the DTC for recall (same as TPR) instead of accuracy. The result of this was the same ccp_alpha that was used in the original model was chosen as optimal. With no improvement from cross validation we next tried to maximize the TPR through a method of cost complexity pruning path that one of our group members found online. This method also led to the same DTC as before.

Since we were not able to make any improvements to our DTC model we wanted to see if we could improve another model to be better than our DTC model. Looking back at our table of performance metrics for the models, we thought our logistic regression model made the most sense to try to improve as it had one of the highest accuracies at 0.9525 and was one of the only models that was even relatively close in TPR to our DTC model with a TPR of 0.0223. We first aimed to improve the model to have a higher TPR by decreasing the threshold value. We tested all threshold values between 0 and 0.5 in increments of 0.05 and calculated the accuracy and TPR on the training set associated with each threshold. From this analysis, we found that the threshold of 0.2 seemed to suit our objective the best as it had the best balance between accuracy and TPR with an accuracy of 0.9416 and a TPR of 0.2766 on the training set. We also thought this threshold intuitively made sense as it would mean that the expected loss of a false negative (not diagnosing someone as immunosuppressed when they actually are) is much greater than that of a false positive (diagnosing someone as immunosuppressed when they actually aren't). This improved logistic regression model had an accuracy of 0.9334 and a TPR of 0.1786 on the test set.

The second way we tried to improve the logistic regression model was through feature engineering by taking out any features that had a p-value greater than 0.05. This led to a logistic regression model that had an accuracy of 0.9476 and a TPR of 0.1071 on the test set. Overall, we thought that the model before feature engineering was better for our application because it only had a slightly lower accuracy but a much higher TPR. Using this model, we then wanted to see how confident we were in this model by performing bootstrap validation on the accuracy and TPR performance metrics of the model. From this analysis, we don't think we can be very confident in the aforementioned values because the bootstrap validation appeared to produce high variance in the TPR and moderate variance in the accuracy. In particular, with 500 bootstrap samples, we observed that the TPR varied from less than 0.05 to about 0.37, and the accuracy varied from about 0.915 to about 0.95 as shown in Figure 3.

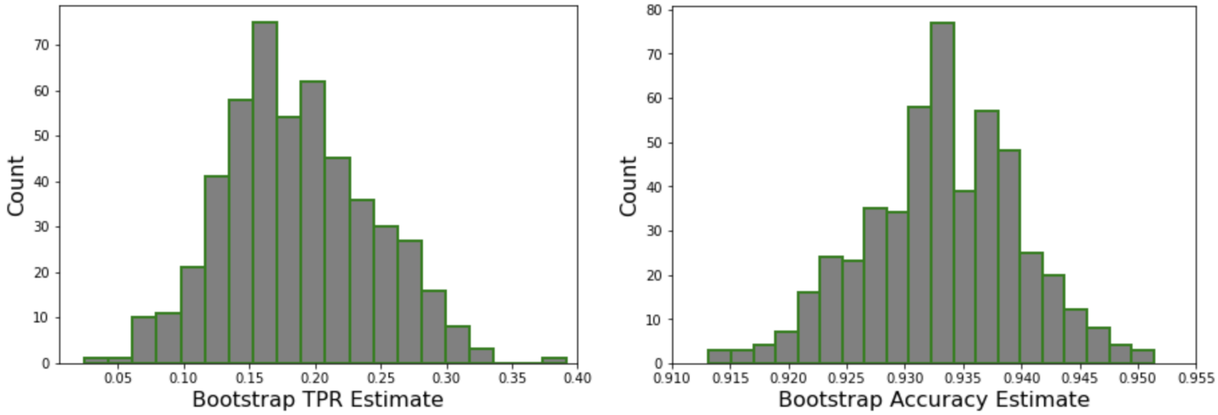


Figure 3

Thus, we concluded that our DTC model was the best fit for our chosen application because it had the highest TPR on the test set, the bootstrap validation showed that we can be confident in the TPR and accuracy metrics, and its accuracy on the test set is not that much less than that of our other models.

We think there are a few ways we can extend our analysis in the future. One way would be to get more data to train our models on to hopefully produce better results. This can either be done by including more of the features from the same data set that we didn't include in this analysis or by finding more data sets with similar information and retraining our models. Another way we can extend our analysis is by trying to build different, and possibly more complex, models. We think that there may be models that are out of the scope of this course that could possibly produce better results than what we found in this analysis. So, it would be interesting to try to find and build other models in the future.

Impact

After running our models and seeing the overall results of each one in predicting immunosuppression our group was able to see the vast potential effects of our project and its applicability to so many different facets of human health. As discussed, the hardest toll that COVID took on the population was on people with other pre-existing conditions, and through our project we were able to classify individuals as either being immunosuppressed or not based on a multitude of factors. Should another pandemic happen, this time around people could be better prepared to take the necessary precautions and in turn have much lower death rates. Of course our model is not perfect and could always have improvements to try and better predict human conditions. One of the ways our model can improve for the future is including more relevant data. This can come in many forms whether it be included in the census as surveys to each individual in households or even expanding the questionnaire from the CDC we used. By including more relevant data, we can possibly make our model more accurate. Another improvement for the future would be in a variable we had decided to omit which was 'HLTHCOND_A'. This variable was slightly different from our dependent variable in that this variable indicated a weakened immune system due to prescriptions. In the world of healthcare a wrong prescription or even side effects from a real prescription run the possibility of worsening the immune system of a person. Including this in the future would definitely aid in figuring out what can cause immunosuppression.

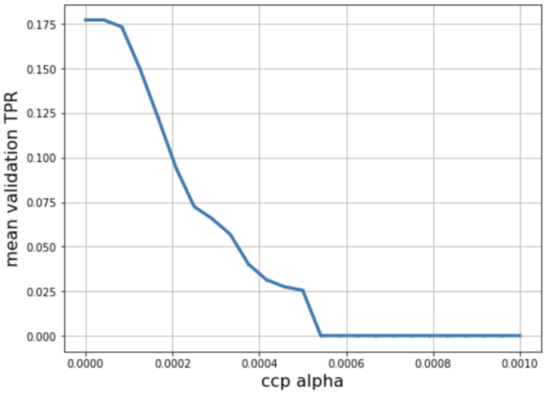
One of the other areas of potential impact was in labeling people as either immunosuppressed or not as it could have a large mental impact. Someone finding out they have a weakened immune system during a global pandemic is something that is sure to cause a lot of panic and stress, which in itself worsens health. Yet, also someone not being labeled when they should have been can also be detrimental. This fine line between the two is why we aimed to find a model that was balanced between accuracy and TPR. However, we more so leaned toward increasing TPR because in medical contexts we want to minimize FNR and maximize TPR to avoid wrongly labeling people as not immunosuppressed and instead error on the side of caution. By focusing on the TPR rates we are still able to make sure the right people are being assigned correctly, but also choosing to error on the not immunosuppressed side rather than the contrary.

References & Appendix

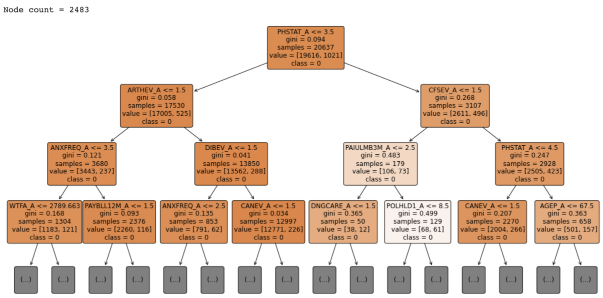
Appendix A: Link to Dataset and PDF with Variables
https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Datasets/NHIS/2021/adultinc21csv.zip
https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/NHIS/2021/adult-summary.pdf

Appendix B: Link to Google Colab (Jupyter Notebook)
https://colab.research.google.com/drive/1QRub_B-1m0QBm3VtrnNlhOQtfSCCIJ-L?usp=sharing

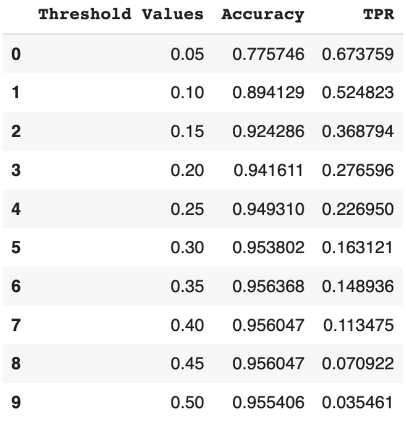
Appendix C: Ccp_alpha vs. Mean Validation TPR with CV on DTC:



Appendix D: Decision Tree Associated with DTC Model:


Node count = 2483
PHSTAT_A <= 3.5 gini = 0.094 samples = 20637 value = [19616, 1021] class = 0
ARTHEV_A <= 1.5 gini = 0.058 samples = 17530 value = [17005, 525] class = 0
CFSEV_A <= 1.5 gini = 0.268 samples = 3107 value = [2611, 496] class = 0
ANXFREQ_A <= 3.5 gini = 0.121 samples = 3680 value = [3443, 237] class = 0
DIBEV_A <= 1.5 gini = 0.041 samples = 13850 value = [13562, 288] class = 0
PAIULMB3M_A <= 2.5 gini = 0.483 samples = 179 value = [106, 73] class = 0
PHSTAT_A <= 4.5 gini = 0.247 samples = 2928 value = [2505, 423] class = 0
WTFA_A <= 2789.663 gini = 0.168 samples = 1304 value = [1183, 121] class = 0
PAYBLL12M_A <= 1.5 gini = 0.093 samples = 2376 value = [2260, 116] class = 0
ANXFREQ_A <= 2.5 gini = 0.135 samples = 853 value = [791, 62] class = 0
CANEV_A <= 1.5 gini = 0.034 samples = 12997 value = [12771, 226] class = 0
DNGCARE_A <= 1.5 gini = 0.365 samples = 50 value = [38, 12] class = 0
POLHLD1_A <= 8.5 gini = 0.499 samples = 129 value = [68, 61] class = 0
CANEV_A <= 1.5 gini = 0.207 samples = 2270 value = [2004, 266] class = 0
AGEP_A <= 67.5 gini = 0.363 samples = 658 value = [501, 157] class = 0
(...)

Appendix E: Accuracy and TPR Values for Different Threshold Values in Logistic Regression



	Threshold Values	Accuracy	TPR
0	0.05	0.775746	0.673759
1	0.10	0.894129	0.524823
2	0.15	0.924286	0.368794
3	0.20	0.941611	0.276596
4	0.25	0.949310	0.226950
5	0.30	0.953802	0.163121
6	0.35	0.956368	0.148936
7	0.40	0.956047	0.113475
8	0.45	0.956047	0.070922
9	0.50	0.955406	0.035461