NORTHEASTERN UNIVERSITY  
COLLEGE OF PROFESSIONAL STUDIES



**ALY6020: Predictive Analytics Final Project**

**Covid 19 – Analysis and Predictive Analytics**

Spring 2021 Term

Group A:

Amber Saxena

Mayank Pandey

Nishita Kamal

Instructor: Marco Montes de Oca

June 10, 2021

1. **INTRODUCTION**

Our dataset is retrieved from <https://www.kaggle.com/tanmoyx/covid19-patient-precondition-dataset/download>, which was originally collected by Tanmoy Mukherjee and

uploaded on Kaggle.com. It was originally sourced from

<https://www.gob.mx/salud/documentos/datos-abiertos-152127>  and released by Mexican

government. It is open for any public usage/research while the source of the data is anonymous.

This dataset is related to global pandemic Covid 19. One of the main problem that healthcare

providers face is shortage of medical resources an efficient distribution of these resources to

most needy ones. This dataset can help predicting what kind of resources are required to what

type of patient on the basis of medical history and other historical facts It was last updated on

July 22, 2020. The dataset contains anonymized patient related data to predict features correlated

with covid 19, which is saved in “.CSV” file format. Altogether there are 23 columns and

5,66,602 rows. Many of the columns are ordinal factors that assigns patients medical situation a

label.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Variable** | **Description** | | **id** | Case Identifier number | | **sex** | Gender of Patient | | patient type | Type of care received by patient | | entry\_date | Patient’s admission to care unit | | Date\_symptoms | Date on which patient’s symptom begins | | Date\_died | Identifies the date of Patient died. Unique value assigned if he is alive | | intubed | Identifies if the patient received intubation | | pneumonia | Identifies if patient had pneumonia | | age | Age of the patient | | pregnancy | Identifies if the patient is pregnent | | diabetes | Identifies if the patient has diabetes | | copd | Identifies if the patient has copd | | asthma | Identifies if the patient has asthma | | |  |  | | --- | --- | | **Variable** | **Description** | | inmsupr | Identifies if the patient has immunosuppression | | hypertension | Identifies if the patient has hypertension | | Other disease | Identifies if the patient has any other disease | | cardiovascular | Identifies if the patient has cardiovascular disease | | obesity | Identifies if the patient has obesity | | Renal chronic | Identifies if the patient has Chronic kidney failure | | tobacco | Identifies if the patient has tobacco | | Contact\_other\_covid | Identifies if the patient was exposed to any other covid diagnosed case | | Covid\_res | Identifies the result of sample reported | | icu | Identifies if the patient has immunosuppression | |

Below is a sample format of our dataset in our python repository.

Table

Description automatically generated

Table

Description automatically generated

1. **EXPLANATORY DATA ANALYSIS**

In this section we provided an understanding of the data in the context of exploration and visualizations. Details about variables categories, how we manipulated the datasets will be covered in *III. Feature Engineering* section.

we conducted an initial check on dataset to see the data counts and data types.

Table

Description automatically generated

Table

Description automatically generated Table

Description automatically generated

We checked missing values and found no missing values. Since the dataset is obtained from Kaggle, we find that missing values were already replaced with numerics like 97,98,99.

Graphical user interface, text

Description automatically generated

Lastly, we checked the imbalance and normality of the dataset using seaborn and value\_cpunt function.

Text

Description automatically generated with medium confidence

A picture containing chart

Description automatically generated

This dataset seems highly imbalanced towards class label 1 i.e. there are more cases when patient was alive than dead. In this case, our model may not be able to correctly predict minority class i.e., 0 (death).

We have employed various techniques like random oversampling, random undersampling, class weight re-distribution, SMOTE oversampling.

With this dataset, we are particularly interested in understanding what features had a significant impact on death rate and whether there is any multicollinearity exist among columns. The correlation matrix below shows the correlation coefficients between several variables related to satisfaction. The darker the color is, the stronger the correlation is. The color of red means a positive correlation and blue means the opposite.

A picture containing text, red, tiled

Description automatically generated

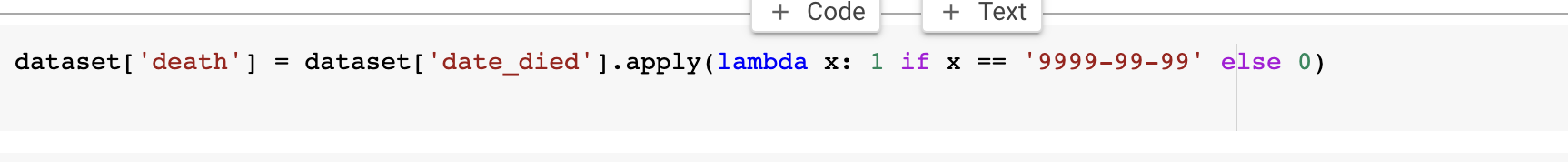
Figure 2.6 Correlation Matrix Heatmap

1. **FEATURE ENGINEERING**

The task here is to prepare all dataset needed from raw data dataset before various modeling techniques processing to predict Covid 19 death cases.

1. *Feature selection*

We derived our target column death from the column date\_died. If provide two class label in our target column death as 0 and 1. 1 is being given to the person who is alive and in death cases 0 was allotted.

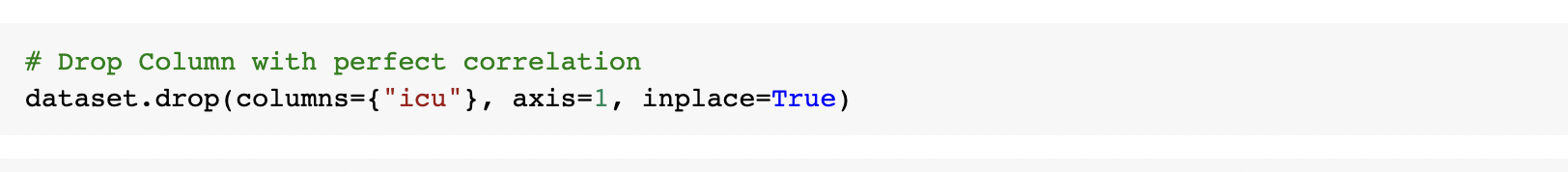


We removed some column intuitively which we felt are not going to help in predicting target column

Text

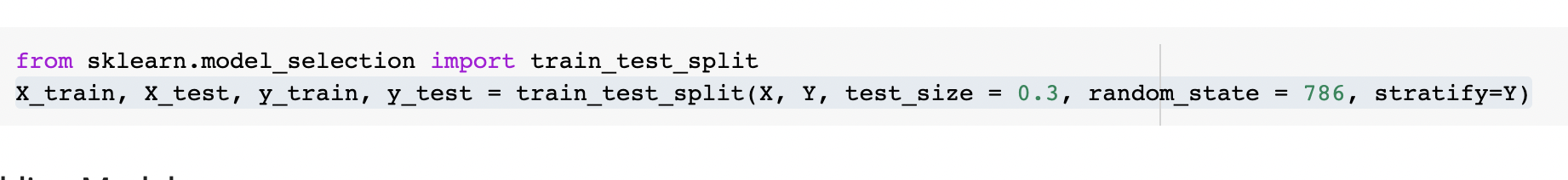
Description automatically generated

Further, we removed those columns which are showing high collinearity with some other columns.



*B. Numerical Binning*

We did numerical binning on the target variable, Alive – 1, Death - 0. These feature and target variables are separated for both the training and testing data in order to display our results.



*C. Imbalanced Data Handling*

We first checked if the data was imbalanced and found 94% of the total dataset has been an alive cases and only in 6% cases death has been reported. We need to make sure that we maintain the same percentage across both training and testing datasets. This kind of splitting the data maintaining the ratio is called "stratification". In this research, the dataset is imbalanced. When the data is imbalanced, there are some popular techniques such as Class weight redistribution, Random Oversample, Random Undersample, SMOTE oversample.

1. **MODELING TECHNIQUES**

After preparing the data, we can implement several tree-based methods and setperformance metrics to evaluate their performance with our passenger data.

*A. Models*

1. *Random Forest*

A random forest model is an ensemble machine learning algorithm that creates multiple uncorrelated decision trees by taking a bagging approach to sampling along. It differs from bagging as it takes a random selection of features as opposed to all of them. The collective results from these trees are then used to make a classification for a given datapoint. Random forests is very accurate dealing with non-linear data as well as outliers and works well with large datasets. Conversely though, the training process is notably slow. This approach can help us determine whether a given patient is at high risk of death or not.

1. *Gradient Boosting Machine*
2. **EXPERIMENT SETTINGS**
3. *Random Forest*

A function is defined to easily call and evaluate our model performance by displaying accuracy information and plots. An example of this is shown below:

Graphical user interface, text, application, email

Description automatically generated

A user defined function has been made to calculate time taken to train the model. It may further help us analyzing performance and model selection.

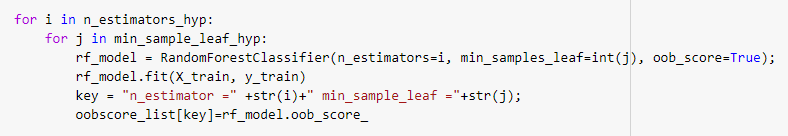
Graphical user interface, text, application

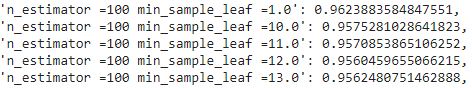
Description automatically generated

The number of trees to use is defined in the variable *n\_estimators\_hyp*, and the minimum number of samples required to split an internal node is defined in the variable *min\_sample\_leaf\_hyp*.

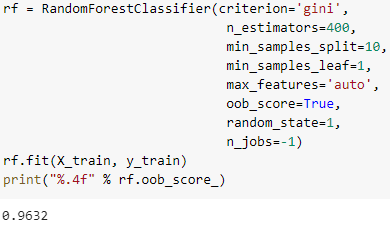


The *RandomForestClassifier* function from the *sklearn* library is called within a nested for-loop to create the models with each of the parameters. For example, a random forest with 100 trees is created with a minimum of 10 leaves. A few iterations of the models are collected and shown below with their Out-of-Bag score.





The OOB score represents the accuracy rate for based on the number of correctly predicted values from the out-of-bag sample. With this information, a random forest model is created using 400 trees and a minimum of sample leaf of 1 is stored. The OOB score for this model is approximately 0.9632.



Calling our custom *eval\_result* function allows us to see the overall accuracy of this random forest model. We got the best result with random forest with random oversampling.The accuracy for the training dataset is approximately 93% and for testing dataset is 91% and the area under the ROC curve for it is about 0.97. However, since it is a highly imbalanced dataset, we have also paid attention to Precision Recall curve and area under PR curve is 0.99 which is encouraging

**Performance Evaluation for Random Forest using Class weight redistribution**

Table

Description automatically generated

Graphical user interface, application

Description automatically generated with medium confidence

ROC Curve for Random Forest using class weight redistribution

Chart, line chart

Description automatically generated

Precision Recall Curve for Random Forest using class weight redistribution

Chart

Description automatically generated

Confusion Matrix for Random Forest using Class weight redistribution

**Performance Evaluation for Random Forest using random oversampling**

**Table

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**A picture containing chart

Description automatically generated**

**Chart, treemap chart

Description automatically generated**

**Performance Evaluation for Random Forest using random undersampling**

**Table

Description automatically generated**

**Graphical user interface

Description automatically generated with medium confidence**

**Chart

Description automatically generated**

**Chart, treemap chart

Description automatically generated**

1. *Gradient Boosting Machine*

1. **RESULT AND DISCUSSION**

The experimentation had been performed on variety of classification models.

*A. Models performance Comparison*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Running time | Accuracy | Precision |
| Random Forest( After Class weight Redistribution | ~173 seconds | 0.80 | 0.22 |
| Random Forest ( Random Oversampling) | ~356 seconds | 0.91 | 0.83 |
| Random Forest( Random Undersampling) | ~23 seconds | 0.88 | 0.79 |
| XGBOOST | ~ seconds |  |  |

The AUC plot shows that the true positive rate is extremely high. This overview indicates that the random forest model was highly accurate in making these predictions. As we noted earlier, random forest models are highly accurate. The training process was considerably slow though, especially since we had to create multiple random forests to find the optimal setting. Thus, it is important to consider the tradeoff between accuracy and processing with a random forest approach.

We have given our attention to PR curve more than ROC curve given the nature of dataset here is highly imbalanced.

1. *Gradient Boosting Machine*
2. **CONCLUSION AND FUTURE SCOPE**

We can use these models to better understand the initial problem of the dataset and how to best determine whether a patient is at high risk of death and need more resources than normal patients.

From the comparison we could see, the Random Forest model with random oversampling gets a high accuracy, lesser time to be trained and high precision as well. It is important to consider the tradeoff between accuracy and the running time, in this project we would recommend predicting the customer airline satisfaction using the Random Forest method under random oversampling.

**REFERENCES**

# Tanmoy Mukherjee. (2020, July 22).COVID 19 patient pre condition dataset, Version 2. Retrieved June 10, 2021 from <https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction>

# Jason Brownlee. Bagging and Random Forest for Imbalanced Classification. Retrieved from https://machinelearningmastery.com/bagging-and-random-forest-for-imbalanced-classification/

# Rocca, J. (2019, April 22). Ensemble methods: bagging, boosting, and stacking. Towards Data Science. Retrieved March 20, 2021 from https://towardsdatascience.com/ensemble- methods-bagging-boosting-and-stacking-c9214a10a205

# Trehan, D. (2020, July 2). Why Random Forest and not Decision Trees. Towards AI. Retrieved March 20, 2021 from https://towardsai.net/p/machine-learning/why-choose- random-forest-and-not-decision-trees

# Yiu, T. (2019, July 12). Understanding Random Forest. Towards Data Science. Retrieved March 20, 2021 from https://towardsdatascience.com/understanding-random-forest- 58381e0602d2