31005 Machine Learning - Spring 2023

COVID-19 Mortality Prediction

Assessment Task 2

Haoyang Huang

14351682

2023-10-15

Abstract

In the context of the COVID-19 pandemic, accurately predicting the mortality risk of patients' post-infection is crucial. This project employs machine learning techniques, emphasizing the resolution of the data imbalance issue. This problem often results in subpar prediction performance for minority classes (e.g., mortality cases) in traditional models. Through a comparison of various algorithms, we found that Random Forests not only perform excellently in overall accuracy but also demonstrate significant advantages in handling imbalanced data, especially in the recall of minority classes. Using the feature importance tool of Random Forests, we further identified the key features affecting the mortality risk of COVID-19 patients. In conclusion, this project offers a technically advanced and effective method for predicting the mortality risk of COVID-19 patients, holding significant reference value for public health decision-making.

Content

1. Introduction

Background

Since the outbreak of the novel coronavirus (COVID-19) at the end of 2019, this pandemic has profoundly impacted the global economy, public health, international relations, and everyone's daily life. Against this backdrop, governments and medical institutions worldwide urgently require accurate data and prediction tools to strategize, allocate resources, and safeguard public health more effectively.

Project Objectives

Our objective is to provide medical institutions and governments with precise mortality risk assessments for COVID-19 patients, assisting informed decision-making. To achieve this, we aim to develop an accurate and dependable prediction model, address inherent data imbalances, ensure the model's interpretability in the medical context, and commit to its continuous optimization with evolving data.

Project Process

We consolidate COVID-19 patient data from diverse sources, ensuring its quality and integrity, then extract relevant features for mortality risk. Utilizing advanced machine learning techniques, we develop and validate a predictive model, delve into its results for actionable medical insights, and continuously refine it based on feedback and evolving data, targeting its real-world application in medical and policy decisions.

2. Literature Review

2.1 Clinical Features and Severity Prediction

Since its emergence in Wuhan, China, at the end of 2019, the COVID-19 pandemic has garnered global attention. As the number of cases escalated, understanding the clinical characteristics of COVID-19 and forecasting its severity became a primary task for the public health and medical systems. This review aims to outline three pivotal studies on COVID-19, offering theoretical backing for researchers in model and feature selection.

Firstly, Wang et al. (2020) conducted research on 138 diagnosed COVID-19 inpatients in a Wuhan-based hospital. They unveiled hospital-associated transmissions, wherein 26% of patients necessitated ICU care, with a mortality rate standing at 4.3%. This study exposed the epidemiological and clinical attributes of COVID-19, underscoring the significance of prognosticating its severity. Conversely, Li et al. (2020) retrospectively analysed COVID-19 patients admitted to Wuhan's Tongji Hospital from January 26 to February 5. They discerned that elder patients with conditions like hypertension and elevated lactate dehydrogenase levels were more prone to progress into severe COVID-19. Furthermore, they ascertained that factor such as gender (male), cardiac injuries, elevated blood sugar, and extensive corticosteroid utilization correlated with augmented mortality rates among severe patients.

Regarding model and feature selection, these studies advocate for the incorporation of factors like patient's age, baseline health conditions, biomarker levels (like lactate dehydrogenase), and other risk factors associated with mortality. Moreover, deep learning methodologies, especially in the realm of medical image analysis, have been validated as potent instruments for forecasting the severity of COVID-19. By achieving the objectives, we aspire to furnish health institutions, governments, and policymakers worldwide with a robust tool, facilitating them in making more judicious and timely decisions in their battle against COVID-19.

2.2 Analytical Methods in Medical Research

In the context of medical and clinical research, precise methodologies for data analysis are pivotal for propelling scientific progress and enhancing outcomes for patients. As the magnitude of medical data has burgeoned in recent epochs, the pivotal role of machine learning and statistical methodologies in data elucidation, diagnostics, and prognostication has escalated. This review elucidates three quintessential methodologies frequently deployed in medical explorations.

The Utility of Support Vector Machines (SVM) in Health Assessments

Support Vector Machines (SVM) are renowned as supervised learning paradigms, devised for classification conundrums. Njoki (2022) accentuates that, vis-à-vis other algorithms, SVM proffers superior precision in health data analytics, especially when confronted with non-linear or linearly separable datasets. SVM's prowess lies in its ability to efficaciously navigate high-dimensional realms and its utilization of diverse hyperplanes to demarcate various classes. Additionally, the ease of SVM's deployment, its transparent nature, and its capacity to deliver robust performance in medical evaluations in a succinct timeframe are noteworthy. Njoki (2022) further emphasizes SVM's validated efficacy in pinpointing diverse health conditions, including but not limited to, malignancies, hypertension, and diabetes.

The Implication of Linear and Logistic Regressions in Clinical Investigations

Bzovsky et al. (2022) delineate linear regression as a statistical modality, designed to ascertain the correlation between one or an array of explanatory variables and a continuous dependent variable. Within the medical research ambit, linear regression is ubiquitously employed to discern the impact of determinants such as age, gender, or specific therapeutic interventions on another continuous resultant variable. Concurrently, logistic regression emerges as an alternative modality, aiming at the elucidation of relationships between determinants and a dichotomous outcome variable. Bzovsky et al. (2022) accentuate the instrumental role of these regression methodologies, equipping clinicians with tools to scrutinize and construe the interrelations between predictive determinants and resultant outcomes.

The Pertinence of Random Forests in Clinical Risk Forecasts

Wongvibulsin et al. (2019) underscore that random forests epitomize an ensemble methodology rooted in tree-based paradigms, apt for both classification and regression undertakings. Contrasting with individual decision trees, random forests enhance precision by amalgamating predictions stemming from an array of trees. In the medical research spectrum, random forests have been acclaimed as efficacious apparatuses, prognosticating individual health trajectories, inclinations, and therapeutic responses. This modality amalgamates the strengths inherent to traditional statistical methodologies with the avant-garde virtues of modern machine learning paradigms.

This review highlights three paramount methodologies in the medical research domain: SVM, linear and logistic regressions, and random forests. SVM stands out in scenarios demanding high precision, especially with non-linear data. Linear and logistic regressions serve as fundamental tools in establishing relationships between determinants and outcomes, providing invaluable insights for clinicians. On the other hand, random forests, with their ensemble nature, showcase an adeptness at handling complex datasets, making them a main choice for risk prediction in clinical settings. The integration of these methodologies, grounded in both traditional statistical techniques and modern machine learning, underscores the evolving landscape of medical research, offering promising avenues for future explorations and clinical applications.

3. Preliminary Model Selection

Based on the literature review, to address the prediction task, we have chosen the following three machine learning models for comparison:

Logistic Regression

Logistic regression appears particularly suitable for this project as it can provide a probabilistic interpretation for the contribution of features to the outcome, and our aim is to predict a binary output. However, logistic regression assumes a linear relationship between the dependent and independent variables, which might not always hold in real data.

Decision Trees

Decision trees are versatile and can handle both categorical and numerical data. They are also highly interpretable, allowing for a visual representation of the decision-making process. However, they can sometimes be overfit to the training data if not properly pruned.

Random Forests

Random forests, being non-parametric, excel on high-dimensional and non-linear data. They manage feature interactions well and provide importance scores for each feature. While random forests typically perform well in terms of prediction accuracy, they might not be as interpretable as single decision trees.

SVM

We decided not to proceed with the SVM. Although it's efficient in managing the high dimensionality of data, SVM is not efficient enough that the training complexity of SVM can be at the cubic level of the number of data points in the worst-case scenario. When the dataset is large, this can lead to excessive training time (Claesen, 2013). Furthermore, the computational cost and complexity associated with SVM, especially with large datasets, outweighed the benefits in the context of our project.

After comparing the models, we initially selected logistic regression, decision trees, and random forests for further evaluation in this project.

3.1 Preliminary Selection

After comparing the models, we initially selected logistic regression as the primary model for this project for the following reasons.

Strong Interpretability

Interpretability is crucial in medical decision-making. Logistic regression not only provides prediction results but also explains the contribution of each feature (like age, medical history, etc.) to the prediction. This allows doctors to better understand which factors majorly influence a patient's risk, thereby offering more targeted treatment recommendations.

Simple Parameter Tuning

Logistic regression is easy to tune and train.

Suitability for Binary Classification

Logistic regression is particularly apt for our goal, which is to predict an outcome (whether deceased or not).

4. Dataset Description

Data source: The data set was provided by the Mexican government (link), which is from Kaggle and is in csv format, recording various body data of about 1 million COVID-19 patients This dataset contains an abnormal number of anonymous patient related information including pre conditions The raw dataset consistency of 21 unique features and 1048576 unique patients In the Boolean features, 1 means "yes" and 2 means "no" Values as 97 and 99 are missing data (NIZRI, 2022). The link is:
https://www.kaggle.com/datasets/meirnizri/covid19-dataset

8.2 Attribute description of the dataset

Attribute	Description
sex	1 for female and 2 for male
age	Age of the patient
classification	covid test findings. Values 1-3 mean that the patient was diagnosed with covid in different degrees. 4 or higher means that the patient is not a carrier of covid or that the test is inconclusive.
patient type	type of care the patient received in the unit. 1 for returned home and 2 for hospitalization.
pneumonia	whether the patient already have air sacs inflammation or not.
pregnancy	whether the patient is pregnant or not.
diabetes	whether the patient has diabetes or not.
copd	Indicates whether the patient has Chronic obstructive pulmonary disease or not.
asthma	whether the patient has asthma or not.
inmsupr	whether the patient is immunosuppressed or not.
hypertension	whether the patient has hypertension or not.
cardiovascular	whether the patient has heart or blood vessels related disease.
renal chronic	whether the patient has chronic renal disease or not.
other disease	whether the patient has other disease or not.
obesity	whether the patient is obese or not.
tobacco	whether the patient is a tobacco user.
usmr	Indicates whether the patient treated medical units of the first, second or third level.
medical unit	type of institution of the National Health System that provided the care.

intubed	whether the patient was connected to the ventilator.
icu	Indicates whether the patient had been admitted to an Intensive Care Unit.
date died	If the patient died indicate the date of death, and 9999-99-99 otherwise.

5. Data EDA

Exploratory Data Analysis (EDA) is a crucial step in understanding the data with which we are working. It allows us to uncover patterns, relationships, anomalies, or any useful insights that can be leveraged in the modelling process.

5.1 Distribution of Target Variable

Initially, we presented the gender distribution of the patients and the proportion of deaths versus survivals. While the gender ratio between males and females was balanced, only 7.34% of patients succumbed to the virus, indicating that about 93% of patients survived. This highlights an imbalance in the data, which will be addressed later as part of the challenges.

5.2 Missing Values

While the dataset description mentioned that values 99 or 97 indicate missing data, our initial analysis revealed the presence of a “98” value in some binary attributes, which is illogical. Therefore, we consider 97, 98, and 99 as missing values, excluding the “age” attribute where these might represent real ages. Our analysis found missing values in multiple attributes as depicted in the visual representation.

5.3 Numerical Features Analysis

Understanding feature correlation with the target variable is pivotal. If a particular medical attribute is highly correlated with the outcome, it could play a crucial role in predictive modelling. Correlations between medical features and mortality can offer valuable insights to medical professionals. Additionally, this step aids in making decisions about feature retention or elimination. According to figure 1 By computing the mean and variance of each numerical feature across the two categories of the 'DATE_DIED' attribute, we can infer about the variability of a feature. A significant variance suggests a wide distribution range in the data, potentially aiding in classification. Apart from 'usmer', 'medical_unit', and 'sex', other attributes showed considerable variance differences. In terms of mean values, the average age of patients who died was 61, whereas for survivors, it was 40. This stark difference underscores the increased risk of mortality due to COVID-19 with advancing age.

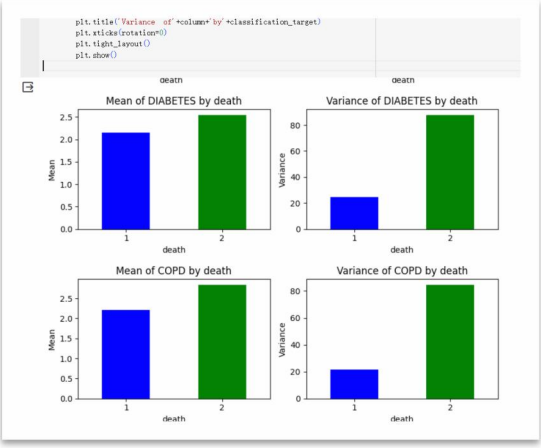


Figure 1

5.4 Heatmap Correlation

Utilizing heatmaps to visualize the correlation between features and the “death” target variable allows for discerning which features have the most significant relationships. This step is instrumental for feature selection, ensuring we use the most relevant predictors in our models.

6. Data Preprocessing

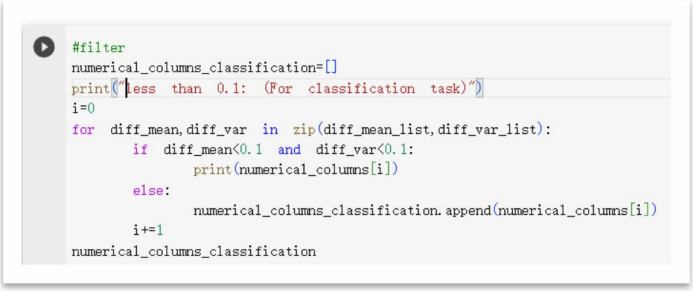
In the process of data preprocessing, several considerations were considered to enhance model efficiency and accuracy:

The features 'intubed,' 'ICU,' and 'pregment' exhibited more than half of their data as missing values. The substantial absence of data in these attributes indicates that they would provide only limited information to the model. Any attempt to impute these missing values, whether through mean, median, or other techniques, might introduce biases, leading to inaccuracies in model predictions. Moreover, it could result in overfitting, diminishing the model's generalizability. Consequently, to prioritize model precision, these three attributes were excluded.

To bolster model efficiency and precision, we computed the difference in mean and variance of data features across the two death categories. Based on these discrepancies, we selectively filtered out features that demonstrated weak associations with the target variable.

We opted to eliminate missing values in this project, a decision grounded in ensuring data accuracy. Such an approach offers several advantages: First, it simplifies the data, enhancing the model's interpretability and reducing predictive errors. Second, it diminishes interference from noise and outliers, thus augmenting analytical accuracy. Furthermore, for a dataset of a million entries, this treatment enhances computational efficiency. Lastly, retaining only values 1 and 2 in the data aids in maintaining high data quality.

According to figure 2, in the data preprocessing phase, we implemented a filter to exclude features that displayed both a difference in mean and a difference in variance less than 0.1. This process helps in refining the feature set, ensuring that only those features with substantial variability and discernible differences between classes are retained for the subsequent classification task.



```
#filter
numerical_columns_classification=[]
print("less than 0.1: (For classification task)")
i=0
for diff_mean, diff_var in zip(diff_mean_list, diff_var_list):
        if diff_mean<0.1 and diff_var<0.1:
                print(numerical_columns[i])
        else:
                numerical_columns_classification.append(numerical_columns[i])
        i+=1
numerical_columns_classification
```

Figure 2

7. Logistic Regression

7.1 Dataset Splitting

Initially, the dataset was divided into training and test sets at a 7:3 ratio. The random state was set to 42 to *ensure consistent and reproducible splits*.

7.2 Fomula

The mathematical formula of logistic regression is used to model binary classification problems. It uses a logistic function (also called a sigmoid function) to estimate the probability that a sample belongs to the positive category. The formula for logistic regression is as follows:

$$P(Y=1|X)=\frac{1}{1+e^{-(\beta_0+\beta_1X_1+\beta_2X_2+\ldots+\beta_pX_p)}}$$

$P(Y=1|X)$ P(Y=1 | X) is the probability that the sample belongs to the positive category (label is 1) given the input feature X. X is the value of the input feature.

The logistic function converts the linear combination into a probability value between 0 and 1, representing the probability of a positive class. If P(Y=1 | X) is greater than or equal to 0.5, the model predicts that the sample belongs to the positive category; if it is less than 0.5, the model predicts that the sample belongs to the negative category. The goal of logistic regression is to find the optimal coefficient β to fit the training data and maximize the prediction accuracy of the model. Logistic regression is used to predict patient survival (positive category is survival, negative category is death). By analysing patient characteristic data, the model can estimate the probability that each patient belongs to the survival category.

7.3 Model Parameter Adjustment

Before training the model, we fine-tuned several key parameters to optimize the model's performance:

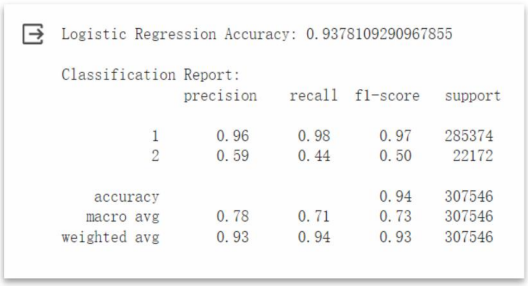
Penalty: We chose 'l2' as L2 regularization typically performs better in practice and doesn't lead to the complete elimination of features. Given that medical data often comprises numerous significant features, we don’t want to discard them entirely through regularization.
Fit_intercept: The model was set to include an intercept term, as even in the absence of any feature input, a baseline prediction is desirable, a standard approach in medical predictions.
solver: 'lbfgs' is a popular optimizer in logistic regression suitable for medium-sized datasets and generally converges well.
Max_iter: We set this to 1000, which is typically sufficient to ensure the logistic regression model converges. Given our dataset might be on the larger side, a higher iteration count ensures the optimizer has ample time to converge, preventing convergence warnings.

7.4 Training Process

The logistic regression model was trained using the partitioned training data. The maximum iteration count (max_iter=1000) ensured the model had sufficient iterations before convergence. Throughout the training, the model iteratively updated its weights to minimize the loss function until convergence was achieved or the maximum iteration count was reached.

7.5 Model Evaluation

The model was assessed using the test dataset. Performance metrics such as accuracy, recall, precision, and the F1 score were computed based on the model’s predictions and the actual labels to gauge its overall performance and ability to recognize positive cases.



Logistic Regression Accuracy: 0.9378109290967855

Classification Report:

	precision	recall	f1-score	support
1	0.96	0.98	0.97	285374
2	0.59	0.44	0.50	22172
accuracy			0.94	307546
macro avg	0.78	0.71	0.73	307546
weighted avg	0.93	0.94	0.93	307546

Figure 3

From the classification results (figure 3), the following observations and conclusions were made:

The model highlighted commendable performance in predicting the survival of patients (Category 1). With a precision of 0.96, a recall of 0.98, and an F1 score of 0.97, it's evident that the model is highly reliable in predicting this category. This is further reinforced by the overall accuracy of approximately 93.78%.

However, when predicting deceased patients (Category 2), the model's performance was relatively weaker. The precision for this category was 0.59, and its recall was 0.44, suggesting that a significant number of actual deceased patients were not correctly identified by the model. The F1 score of 0.50 for this category

further underscores this point, indicating a need for improvement in this area.

The stark contrast between the number of predictions for surviving patients (285,374) and deceased patients (22,172) in the test data suggests a data imbalance. This imbalance likely contributed to the model's bias towards predicting survival (Category 1) over death (Category 2), which could be a primary reason for the relatively lower recall rate for the deceased patients.

Considering the critical importance in the medical field of accurately identifying genuine cases, particularly the deceased patients, the recall metric stands out as a crucial evaluation criterion. Recall effectively measures the model's capability to identify true positive cases. To better address the medical community's needs and provide actionable insights, prioritizing the enhancement of the recall rate for deceased patients in future iterations and refinements of the model is imperative.

7.5 Current Challenge: Data Imbalance Issue

7.5.1 Problem Description

In machine learning projects, when the sample size of one category significantly exceeds that of another, this phenomenon is referred to as data imbalance. In this project, this imbalance is evident as the number of deceased patients is lower than the survivors showed by figure 4. This data skewness can lead the model to have a heightened inclination towards the majority class (surviving patients), while neglecting or misclassifying the minority class (deceased patients).

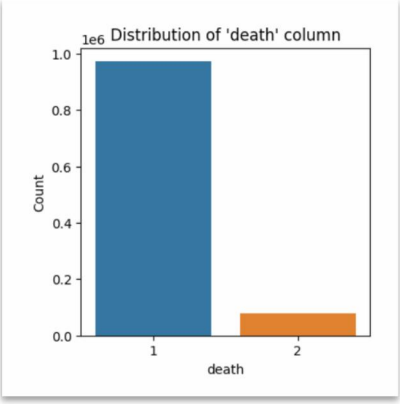


Figure 4

7.5.2 Why It's a Problem

Model Bias: Logistic regression, by its nature, doesn't inherently counteract data imbalance and might lean towards predicting the more prevalent category. This is because the model can effortlessly achieve a higher accuracy rate by doing so. However, this doesn't necessarily imply that the model performs just as well when predicting the minority class.
Decreased Recall: For the minority class, particularly in the medical domain, missing a genuine case can have

severe repercussions. Hence, a high recall is crucial for such problems.
Despite the overall high accuracy (93%), this metric might be misleading because of the inherent data set imbalance, given that most patients survive and only a minority succumb to the disease. In imbalanced datasets, overall accuracy is often dominated by the majority class. As a result, the overall accuracy doesn't genuinely reflect the model's performance, especially regarding the minority class. The logistic regression model, given the current data distribution and target task, doesn't adequately address the data imbalance issue.

7.5.3 Resolution Strategies

To tackle the challenges posed by data imbalance, while we could directly adjust the dataset through methods like oversampling or under sampling, in this project, we prefer to address the issue through model adjustments to better represent the inherent performance of the model. Hence, I decided to experiment with Decision Trees and Random Forests, both of which are widely acknowledged to perform well in the context of imbalanced data. With this approach, I aim to identify a model that can effectively mitigate or even negate the impact of data imbalance.

8. Decision Tree

8.1 Introduction

Based on our experience with the logistic regression model, we observed its sensitivity to imbalanced data. While logistic regression offers an intuitive understanding of the contribution of each feature to the target variable, its linear nature and direct estimation of feature weights might lead to inadequate recall rates when faced with highly imbalanced datasets. Given the importance of recall in the medical field, we believed that a shift to a more flexible model that can directly handle complex relationships between features was necessary. This led us to consider decision trees.

8.2 Information entropy and Formula

Entropy is a measure of the uncertainty of random variables, and the formula for information entropy can be expressed as follows:

$$H(D) = \sum_{i=1}^{c} P(c_i) * \log_2(N)$$

That is:

$$H(D) = \sum_{i=1}^{c} P(c_i) * \log_2(\frac{1}{p_i})$$

To simplify the operation, the standard formula for information entropy is obtained by using the properties

of logarithmic functions for deformation, as follows:

$$H(D) = -\sum_{i=1}^{c} P(c_i) * \log_2(P(c_i))$$

H (D) represents the information entropy of dataset D. C represents the number of categories in the dataset. P () represents the probability of categories appearing in the dataset.

Information Gain:

Information gain is used to measure the degree to which information entropy decreases after selecting a certain attribute for segmentation, that is, the degree to which the purity of the segmented data increases.

The formula for information gain is as follows:

$$IG(D, A) = H(D) - \sum_{v=1}^{V} \frac{|D_v|}{|D|} * H(D_v)$$

IG (D, A) represents the information gain on attribute A, D is the original dataset, which is the sub dataset segmented based on the value v of attribute A, and V is the number of values of attribute A.

CART algorithm and entropy: CART (classification and regression tree) is a decision tree learning technique that can be split using various metrics (CART in Machine Learning, 2022). In this project, CART is utilizing entropy to make decisions. By evaluating information gain based on entropy at each potential segmentation point, CART constructed a binary tree where the feature with the highest amount of information (in terms of entropy reduction) was selected for early segmentation. This method ensures that the top layer of the tree is the most discriminative. By using entropy, CART ensures that the most information can be added to each split, resulting in an efficient and robust decision tree.

8.3 Training Process

The following section provides an in-depth explanation of select parameters employed in the Decision Tree model, specifically highlighting their significance in addressing data imbalance. Each parameter, from 'entropy' as a splitting criterion to the crucial 'class_weight', plays a pivotal role in ensuring that the model not only remains robust against overfitting but also pays due attention to minority class instances. The objective behind these parameter settings is to harness the model's full potential in datasets where one class significantly outnumbers the other. All the parameter is showed by figure 5.

Criterion='entropy': Utilizing 'entropy' as the split criterion can help the model make more informative decisions, especially in the face of imbalanced data.
Max_depth=10: Limiting the tree's maximum depth to 10 prevents the model from becoming overly complex, which could lead it to overfit the majority class and overlook the minority class.
Min_samples_split=10 and min_samples_leaf=5: These parameters ensure that each split in the tree and each leaf node is based on a sufficient number of samples, thus averting overfitting to noise or outliers in the minority class.

```
from sklearn.tree import Decision
from sklearn.metrics import class
clf_tree = DecisionTreeClassifier(
        criterion='entropy',
        splitter='best',
        max_depth=10,
        min_samples_split=10,
        min_samples_leaf=5,
        max_features=None,
        random_state=42,
        max_leaf_nodes=None,
        min_impurity_decrease=0.0,
)
```

Figure 5

In summary, these parameters were chosen to optimize the Decision Tree model's performance, especially considering the challenges of imbalanced data. By fine-tuning the tree's depth, setting minimum samples for splits and leaves, and balancing the classes, the model aims to achieve a good balance between bias and variance, while effectively handling the class imbalance.

8.4 Model Evaluation

From a project perspective, our primary concern is the recall metric. In the medical domain, especially when predicting patient mortality, a high recall signifies the model's ability to correctly identify a greater number of patients genuinely at risk. Overlooking patients who genuinely warrant attention can have severe consequences.

Considering Class 1 (potentially representing patients who survived), the decision tree achieved a recall of 0.99, whereas logistic regression scored 0.98. Although both models performed admirably, the decision tree held a slight edge, identifying a higher number of surviving patients.

More critically, for Class 2 (potentially representing deceased patients), the decision tree had a recall of 0.49, while logistic regression achieved 0.44. This indicates that the decision tree could identify more true instances of Class 2, even though neither model excelled in this respect.
Upon comprehensive evaluation, the decision tree slightly outperforms logistic regression in terms of recall. The significance of a high recall in a medical project is self-evident; we aim to recognize as many genuinely at-risk patients as possible to facilitate timely intervention. In this context, even the modest recall enhancement of the decision tree holds clinical relevance.

	precision	recall	f1-score	support
1	0.96	0.99	0.97	291793
2	0.74	0.49	0.59	22780
accuracy			0.95	314573
macro avg	0.85	0.74	0.78	314573
weighted avg	0.95	0.95	0.95	314573

Figure 6

Impact of Imbalanced Data

Compared to the logistic regression model, the decision tree demonstrated improved recall for the minority class (Class 2), rising from 44% to 49%. A deeper dive reveals that by leveraging information gain as the splitting criterion during parameter tuning, the model prioritizes features that yield higher information gain for splits. This approach aids the model in better distinguishing between imbalanced classes, emphasizing the minority class and thereby enhancing its recall. Given the medical nature of this project and the emphasis on recall, such an approach is commendable.

However, it's regrettable that despite its edge in handling imbalanced data, the decision tree hasn't fully addressed the challenges posed by data imbalance. Specifically, when one class (e.g., deceased patients) is underrepresented compared to another, the decision tree might still exhibit a bias towards the more prevalent class. This predisposition risks misclassifying genuinely high-risk patients (e.g., those at risk of death) as survived, depriving medical teams of the opportunity to provide timely intervention for these critical cases.

As a standalone model, the decision tree's stability and robustness still leave room for improvement. The model can be overly sensitive to slight perturbations in the data, leading to prediction volatility. To mitigate this instability, an ensemble approach is warranted, combining predictions from multiple models to offset the shortcomings of a single model and enhance overall prediction stability and accuracy.

9. Random Forest

9.1 Introduction

Random Forest is a powerful ensemble method that leverages the collective wisdom of multiple decision trees to enhance overall prediction accuracy and stability through a voting mechanism. This approach offers several key advantages. By amalgamating predictions from multiple decision trees, Random Forest mitigates overfitting, significantly enhancing the model's overall accuracy. Moreover, Random Forest exhibits resilience to noise, ensuring stable prediction performance across varying data distributions. Additionally, it provides feature importance scores, aiding in the identification of critical predictors and facilitating feature selection for model optimization. Notably, Random Forest's robustness and diversity enable it to handle missing values effectively. As a result, we re-implemented the preprocessing step without missing value handling.

Random Forest not only yields high accuracy but also excels in addressing data imbalance, notably by improving recall for deceased patients. This means our model is better equipped to correctly identify patients at genuine risk. Given these advantages, we have chosen Random Forest as our next modeling strategy, with the expectation of achieving more accurate and stable prediction results.

9.2 Decision Tree Construction

Random Forest comprises multiple decision trees, each serving as a classification or regression model

tailored to specific tasks. The process of constructing decision trees involves recursively partitioning the dataset into subsets, enabling predictions of the target variable by selecting optimal features and thresholds for splitting.

Introduction of Randomness

A pivotal aspect of Random Forest is introducing randomness to diversify the model. This randomness is achieved through two main mechanisms:

Each tree's construction involves randomly selecting a subset of training data using a technique known as "Bootstrap sampling." This ensures that each tree observes only a portion of the data, enhancing model diversity. What's more, when making split decisions at each node, Random Forest evaluates only a subset of features chosen randomly rather than considering all available features. This strategy ensures that each tree focuses on different features, further enhancing diversity.

Ensemble Prediction

Once multiple decision trees are constructed, Random Forest combines their predictions to make final predictions. In classification tasks, it often employs a majority voting scheme, where each tree "votes" for a predicted class, and the class with the most votes become the final prediction. In regression tasks, it takes the average of predictions from all trees, yielding the final prediction.

These principles underlie the mathematical foundation of Random Forest, making it a versatile and powerful machine learning technique.

9.3 Model Parameter Adjustment

We have set the important parameters as follows,

N_estimators (number of trees): We set 300 trees to ensure the robustness of the model.

Criterion: Like the decision tree, we chose "entropy" as the criterion for evaluating segmentation quality. This is the information gain method, which looks at the increase in purity that each split can bring.

Max_depth (maximum depth of tree): The maximum depth of each tree is set to 10, which helps prevent the model from overfitting.

Max_features (maximum number of features): We chose 'sqrt', which means that each split will consider the square root of the total number of features. This helps ensure model diversity and reduces overfitting.

Bootstrap (whether to use bootstrap sampling): We selected True, which means that when building each tree, it is based on a random subset of the original data.

Class weight: We selected 'balanced', which means that the model will automatically adjust the weight so that minority class samples receive greater emphasis in model training, thereby dealing with the class imbalance problem. The principle behind it is as follows:

$$\text{Weight} = \frac{n_{\text{samples}}}{n_{\text{classes}} * \text{np.bincount(y)}}$$

Among them, n_{samples} is the total number of samples, n_{classes} is the total number of categories, and np.bincount(y) is the number of samples returned for each category. This calculation method ensures that categories with fewer occurrences receive higher weights, while more common categories receive lower

weights. This can help the model pay more attention to categories that appear less frequently.

9.4 Model Performance

When comparing the model evaluation of random forests and decision trees, we can find the following observations:

For category 1 (i.e., surviving patients), both models showed similar performance. The accuracy of both the random forest and the decision tree reached 1.00, which means that both are very accurate in predicting samples as category 1. In terms of recall rate, the two are equally close, about 0.88, which means that among all the samples that are category 1, about 12% of the samples are not correctly predicted.

For category 2 (i.e., patients who died), the random forest showed a higher recall rate than the decision tree, reaching 0.95, which means that the random forest can correctly predict more samples that are category 2. sample. However, both models are less than ideal in terms of accuracy, especially the accuracy of random forest which is only 0.38. This means that among the samples predicted by the model to be category 2, the proportion of samples that are category 2 is relatively low, and there are a large number of misjudgements.

	precision	recall	f1-score	support
1	1.00	0.88	0.93	291793
2	0.38	0.95	0.54	22780
accuracy			0.88	314573
macro avg	0.69	0.91	0.74	314573
weighted avg	0.95	0.88	0.90	314573

Figure 7

9.5 Exploratory Analysis

Obviously, although random forests should theoretically be able to solve some of the inherent problems of decision trees and provide better performance, in this medical project we did not see this clear advantage. This brings us to our attention: why does the random forest not outperform the decision tree as we expected in this case?

There are two reasons. First, the pattern of our data set itself may be relatively simple, that is, the relationships and interactions between data may not be so complex that the performance of a single decision tree is already close to optimal. When the data pattern is simple, increasing the complexity of the model does not necessarily lead to better performance, but may lead to overfitting and unnecessary computational overhead. Secondly, although I have tried different parameter settings to optimize the performance of the random forest, the existing parameters may not be optimal. This may cause the model to perform worse than expected in some respects.

9.6 Hyperparameter Optimization & Feature Selection

To further dig into the root of this problem and find ways to improve the performance of random forests, we decided to turn to a more systematic parameter tuning method: grid search. Grid search is a method of optimizing model performance by iterating through a given combination of parameters and is the best technique (Ayuya, 2021)

According to the documentation of scikit-learn (2019), the sklearn library has encapsulated the grid search function for us, which means that we can easily use it directly for hyperparameter adjustment. Through grid search, we hope to find an optimal set of parameters that optimize the performance of random forests and thereby better exploit its theoretical advantages over decision trees. I consulted with ChatGPT, our advanced AI assistant, which provided us with best practices-based grid search implementation code.

When performing model hyperparameter optimization, a comprehensive grid search involves evaluating every possible parameter combination, which can be computationally very expensive. Especially in the context of large data sets and complex models, a comprehensive parameter space search can lead to unacceptably long computational times. Therefore, considering the limitations of hardware performance, we adopted a more conservative strategy in this project and chose a more limited hyperparameter subspace to explore. Here is the grid of hyperparameters we chose for our random forest in figure 8.

```
# provided by chatgpt
param_grid = {
        'n_estimators': [300],
        'criterion': ['gini', 'entropy'],
        'max_depth': [10],
        'min_samples_split': [5, 10],
        'min_samples_leaf': [2, 5],
        'max_features': ['sqrt'],
        'bootstrap': [True],
        'class_weight': ['balanced']
}

rf = RandomForestClassifier(random_state=42)
```

Figure 8

This selection is intended to serve as an enlightening initial exploration, providing a benchmark and direction for subsequent deeper and more extensive hyperparameter optimization.

Feature selection is a key step in machine learning, which can help us remove redundant or irrelevant features, thereby improving the performance and accuracy of the model. In our project, we exploited the properties of random forests to evaluate the importance of individual features. Random Forest assigns each feature an importance score, which reflects how much that feature contributes to the model's predictions. Based on these scores, we selected the top 50% of the most important features for subsequent model training. Such a strategy can not only reduce the complexity of the model and increase the training speed but may also improve the generalization ability of the model, making it perform better on unknown data.

```
#Select the top 50% of the most important features
n = int(0.5 * len(feature_importances))
top_n_indices = np.argsort(feature_importances)[-n:]

train_x_selected = train_x.iloc[:, top_n_indices]
test_x_selected = test_x.iloc[:, top_n_indices]

# use chosen feature train the model
grid_search.best_estimator_.fit(train_x_selected, train_y)
```

Figure 9

9.7 New Random Forest Evaluation

Parameter selection: After grid search, we obtained the best parameter combination, including using 'gini' as the dividing criterion, the maximum depth of 10, the minimum number of leaf node samples as 2, the minimum number of split samples as 5, etc. These parameter configurations provide better predictive performance for our model.

	precision	recall	f1-score	support
1	1.00	0.88	0.93	291793
2	0.38	0.95	0.54	22780
accuracy			0.88	314573
macro avg	0.69	0.91	0.74	314573
weighted avg	0.95	0.88	0.90	314573

Figure 10

According to figure 10, after selecting parameters through grid search, the performance of the model still did not improve significantly. This may verify the previous conjecture that our data set is relatively simple, and the patterns and relationships in it are very clear and intuitive, resulting in the model failing even with hyperparameter optimization. The performance is also difficult to significantly improve. Nonetheless, we decided to try to further optimize the model through feature selection, hoping that this process would bring additional performance improvements to the model, and evaluate it again.

Based on the feature selection results in Figure 11, we identified the following key features, which will become an important basis for our modeling and analysis: 'USMER', 'HIPERTENSION', 'DIABETES', 'MEDICAL_UNIT', 'CLASIFFICATION_FINAL', 'AGE ', 'PNEUMONIA', 'ICU', 'INTUBED' and 'PATIENT_TYPE'. The selection of these features is performed by the model and they are considered crucial to improve model performance.

```
# Print the selected features
selected_features = train_x.columns[top_n_indices]
print("Selected Features:")
print(selected_features)

Selected Features:
Index(['USMER', 'HIPERTENSION', 'DIABETES', 'MEDICAL_UNIT',
       'CLASIFFICATION_FINAL', 'AGE', 'PNEUMONIA', 'ICU', 'INTUBED',
       'PATIENT_TYPE'],
      dtype='object')
```

Figure 11

Then, we get the model performance combining feature selection + grid search, which showed by figure 12.

Classification Report using Top Features:

	precision	recall	f1-score	support
1	1.00	0.88	0.93	291793
2	0.38	0.95	0.55	22780
accuracy			0.89	314573
macro avg	0.69	0.91	0.74	314573
weighted avg	0.95	0.89	0.91	314573

Figure 12

After using the filtered main features, the overall accuracy of the model improved from 0.88 to 0.89. The f1-score for category 2 (patients who died) improved from 0.54 to 0.55. This means that by reducing the number of features, the model can more intensively capture the most important patterns in the data, thereby improving performance in some respects.

Although feature selection brings a certain degree of performance improvement to the model, the magnitude of this improvement is relatively small. This slight change further confirms our previous conjecture: our data set may have certain limitations, making it difficult for the model to achieve significant performance improvements through traditional optimization methods. This also implies that more complex or in-depth methods may be needed to mine the potential information in the data. Still, feature selection gave us a deeper understanding of key features in the data that played a role in predicting patient death but was clearly not enough to bring the model to the level of performance we expected.

10. Conclusion and Reflection

In this project, we embarked on a mission to harness the predictive prowess of machine learning, with a spotlight on the Random Forest model. Our models exhibited a commendable ability to navigate the intricacies of class imbalances. Despite leveraging a suite of optimization strategies, including grid search and feature selection, the uptick in model performance was subtle, alluding to inherent constraints within our dataset.

A closer inspection of our data revealed that many attributes, such as DIABETES and OBESITY, predominantly possess binary values. This binary nature, while simplifying certain aspects, may also have imposed ceilings on the performance enhancements we aspired for. Through feature selection, we discerned the pivotal role of specific attributes in predicting patient outcomes. However, this revelation didn't necessarily cascade into anticipated improvements in model performance.

Feature selection shed light on pivotal features instrumental in predicting patient fatalities. Yet, this insight didn't translate into significant performance augmentation. Predicting patient survival remains paramount in the realm of medical science, deeply influencing treatment trajectories and the judicious allocation of medical resources. The endeavor underscored the imperative of relentless model optimization and innovative problem-solving when confronted with intricate medical datasets.

Despite encountering a slew of challenges and inherent limitations, the project has been a treasure trove of experiences and insights, particularly pertinent to the design and deployment of predictive models in medical research.

ChatGPT's application does an excellent job of meeting a clear need and is able to efficiently generate the required code snippets that help us achieve our specific goals. Whether providing code examples, explaining algorithms, or performing data analysis, ChatGPT has demonstrated impressive capabilities at these tasks. This tool can save a lot of time and energy, allowing us to focus more on the idea of solving the problem rather than the method. However, it usually does not provide constructive suggestions or in-depth discussions, such as in random forest optimization. This emphasizes the key role of humans in AI applications. AI should be viewed as an auxiliary tool to accelerate and enhance work, but decision-making and judgment are still dominated by humans. When leveraging AI, we need to maintain critical thinking and judgment to ensure that the final decision is informed, meets needs, and is successful in practical applications.

11. Future Work

In light of my evolving journey within machine learning, the horizons extend far beyond just this project. While our immediate attention was directed towards healthcare, it's evident that machine learning's adaptability holds promise across diverse sectors, from finance to robotics. Drawing from the foundational knowledge acquired in this course, I am enthusiastic about diving into a broader spectrum of algorithms. A pivotal aim is to craft machine learning solutions that are tailor-made for specific domains, ensuring a blend of contextual relevance and technical prowess. As I tread further into the intricate realm of machine learning, each endeavor is perceived as a building block, paving the way for more nuanced and sophisticated explorations. The overarching goal remains: to devise holistic strategies that tackle fundamental challenges.

In this project, the depth of analysis was limited due to time constraints. However, I am committed to ensuring that in-depth exploratory analysis remains integral to all my subsequent academic pursuits.

12. Reference List

Ayuya, C. (2021, March 18). Using Grid Search to Optimize Hyperparameters. Engineering Education (EngEd) Program | Section. https://www.section.io/engineering-education/grid-search/

Bzovsky, S., Phillips, M. R., Guymer, R. H., Wykoff, C. C., Thabane, L., Bhandari, M., Chaudhary, V., Sivaprasad, S., Kaiser, P., Sarraf, D., Bakri, S. J., Garg, S. J., Singh, R. P., Holz, F. G., Wong, T. Y., & on. (2022). The clinician's guide to interpreting a regression analysis. Eye, 36(9), 1715–1717. https://doi.org/10.1038/s4143302201949z

CART (Classification And Regression Tree) in Machine Learning. (2022, September 23). GeeksforGeeks. https://www.geeksforgeeks.org/cart-classification-and-regression-tree-in-machine-learning/

Claesen, M. (2013, May 17). Why does training an SVM take so long? How can I speed it up? Artificial Intelligence Stack Exchange. https://ai.stackexchange.com/questions/7202/why-does-training-an-svm-take-so-long-how-can-i-speed-it-up

Li, X., Xu, S., Yu, M., Wang, K., Tao, Y., Zhou, Y., Shi, J., Zhou, M., Wu, B., Yang, Z., Zhang, C., Yue, J., Zhang, Z., Renz, H., Liu, X., Xie, J., Xie, M., & Zhao, J. (2020). Risk factors for severity and mortality in adult COVID-19 inpatients in Wuhan. The Journal of Allergy and Clinical Immunology, 146(1). https://doi.org/10.1016/j.jaci.2020.04.006

NIZRI, M. (2022, October 7). COVID-19 Dataset. Www.kaggle.com. https://www.kaggle.com/datasets/meirnizri/covid19-dataset

Njoki, L. (2022, March 14). Effectiveness of Support Vector Machine in Analyzing Medical Data. Engineering Education (EngEd) Program | Section. https://www.section.io/engineering-education/effectiveness-of-svm-on-health-asessment/

Noble, W. S. (2006). What is a support vector machine? Nature Biotechnology, 24(12), 1565–1567. https://doi.org/10.1038/nbt1206-1565

OpenAI. (2023). ChatGPT (Mar 14 version) [Large language model]. https://chat.openai.com/chat

scikit-learn. (2019). sklearn.model_selection.GridSearchCV — scikit-learn 0.22 documentation. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Shahid, N., Rappon, T., & Berta, W. (2019). Applications of artificial neural networks in health care organizational decision-making: A scoping review. PLOS ONE, 14(2), e0212356. https://doi.org/10.1371/journal.pone.0212356

Shan, F., Gao, Y., Wang, J., Shi, W., Shi, N., Han, M., Xue, Z., Shen, D., & Shi, Y. (2020). Abnormal Lung Quantification in Chest CT Images of COVID‐19 Patients with Deep Learning and its Application to Severity Prediction. Medical Physics. https://doi.org/10.1002/mp.14609

Wang, D., Hu, B., Hu, C., Zhu, F., Liu, X., Zhang, J., Wang, B., Xiang, H., Cheng, Z., Xiong, Y., Zhao, Y., Li, Y., Wang, X., & Peng, Z. (2020). Clinical Characteristics of 138 Hospitalized Patients with 2019 Novel Coronavirus–Infected Pneumonia in Wuhan, China. JAMA, 323(11), 1061–1069. https://doi.org/10.1001/jama.2020.1585

Wongvibulsin, S., Wu, K. C., & Zeger, S. L. (2019). Clinical risk prediction with random forests for survival, longitudinal, and multivariate (RF-SLAM) data analysis. BMC Medical Research Methodology, 20(1). https://doi.org/10.1186/s12874-019-0863-0

APPENDIX:

Partial communication screenshot with ChatGPT：

python Copy code

```python
# ...
# Select the top 50% of the most important features
n = int(0.5 * len(feature_importances))
top_n_indices = np.argsort(feature_importances)[-n:]

# Print the selected features
selected_features = train_x.columns[top_n_indices]
print("Selected Features:")
print(selected_features)

train_x_selected = train_x.iloc[:, top_n_indices]
test_x_selected = test_x.iloc[:, top_n_indices]

# use chosen feature train the model
grid_search.best_estimator_.fit(train_x_selected, train_y)

y_pred_rf_selected = grid_search.best_estimator_.predict(test_x_selected)

print("\nClassification Report using Top Features:")
print(classification_report(test_y, y_pred_rf_selected))
```

For your project, especially in the medical domain, here are some recommendations to optimize the performance of the Random Forest model:

Handling Imbalanced Data:

Continue to use the `class_weight='balanced'` parameter to make the model more attentive to minority classes.
Consider using resampling techniques such as SMOTE to increase the number of samples in the minority class or undersampling techniques to reduce the number of samples in the majority class.
Feature Engineering:

For medical data, some features may have a strong correlation with the target variable. Based on the model's feature importance, further feature selection can be performed to reduce noise and irrelevant features.
Consider creating new interaction features based on medical knowledge.
Adjusting Hyperparameters:

Consider increasing `n_estimators` to boost the number of trees and improve model performance.
Based on the previous results with decision trees, fine-tune hyperparameters such as `max_depth`, `min_samples_split`, and `min_samples_leaf` to prevent overfitting.
Use cross-validation in combination with random or grid search to automatically search for the best hyperparameters.