Project Structure for COVID-19 Machine Learning Analysis

# Project Objective

“During the entire course of the pandemic, one of the main problems that healthcare providers have faced is the shortage of medical resources and a proper plan to efficiently distribute them. In these tough times, being able to predict what kind of resource an individual might require at the time of being tested positive or even before that will be of immense help to the authorities as they would be able to procure and arrange for the resources necessary to save the life of that patient.’’ (www.kaggle.com, n.d.)

The primary objective of this project is to develop a machine learning model. This model aims to assess the risk level of COVID-19 patients by analyzing their symptoms, current health status, and medical history. Such a predictive tool could be vital in making informed decisions about resource allocation and patient care during the pandemic.

The image illustrates the structured layout of our project, detailing a step-by-step organizational approach. This schematic has been instrumental in guiding the restructuring process, clearly delineating each phase for easier follow-through. The visualization highlights the project's potential, showcasing how systematic planning and decision-making can enhance its development.

This revised version offers a clearer, more concise description. It emphasizes the importance of the schematic in the project's restructuring process and its role in revealing the project's potential.

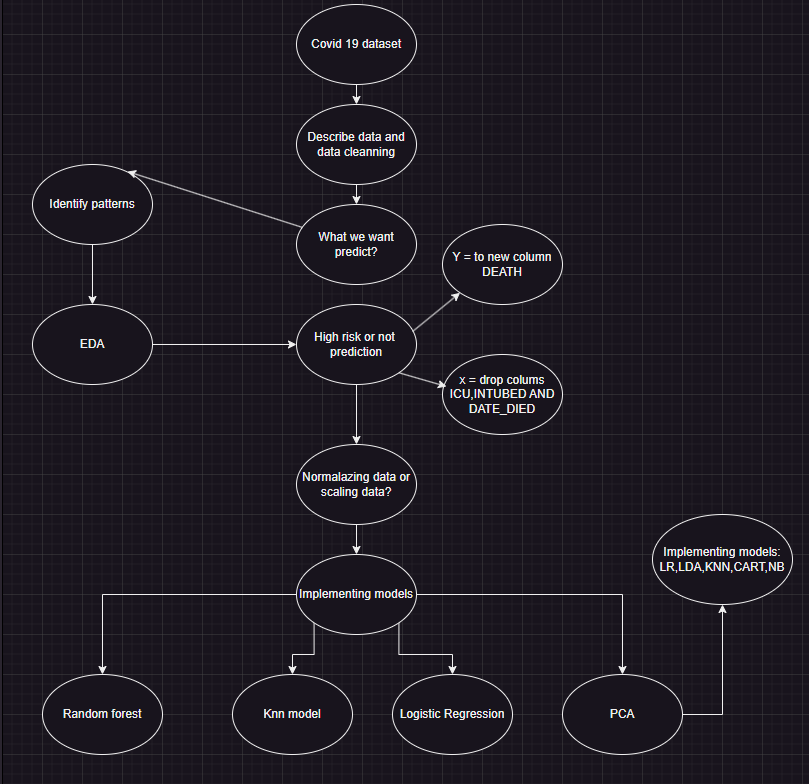


Figure 1 Structure of the project

# Dataset and Data cleaning

Dataset

Utilize a comprehensive COVID-19 dataset, which could include patient information, the data was collected from Kaggle on the following link: <https://www.kaggle.com/datasets/meirnizri/covid19-dataset/data>

## Data cleaning

### Describing the data

First I implementing data visualization and utilizing methods like describe(), shape, and info() to understand its structure and content. This step helps in identifying missing values and verifying the correctness of the data types.For we can identify if the are missing values, or the type of the values is correct,before can start pass to the model and identify a pattern.

### Handling Missing Values

We identify missing values 97 and 98 on the columns PREGNAT, the majority of these records are for males, who cannot be pregnant, we convert this column to a Boolean feature to improve model predictions.

### Drooping and creating a new column

The columns with excessive missing values, like 'ICU' and 'INTUBED', by considering their removal if they are unlikely to contribute valuable information. And I created a new column death and passing the Boolean value because the date 9999-99-99 means that are not death.

# EDA

EDA to understand the dataset deeply. This includes visualizing the distribution of COVID-19 cases with outcomes such as deaths and recoveries. Use plots like histograms, pie charts, and correlation matrix to visualize the proportion of fatalities and survivals. This visualization helps in identifying patterns or anomalies in the dataset.

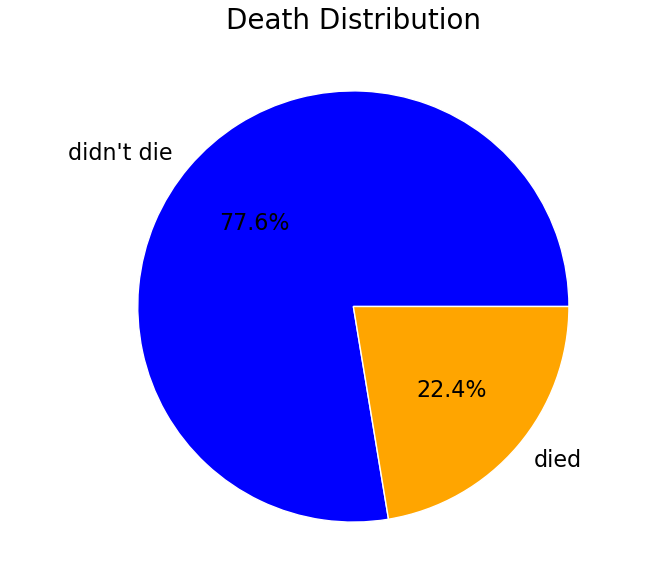


Figure 2 Pie chart of Death distribution

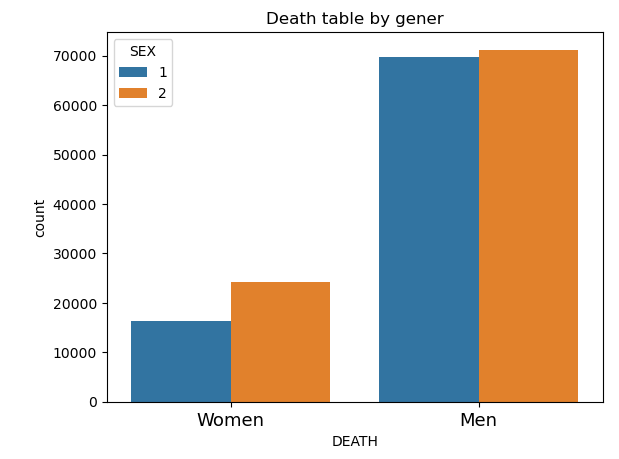


Figure 3 Histogram of Death by gener

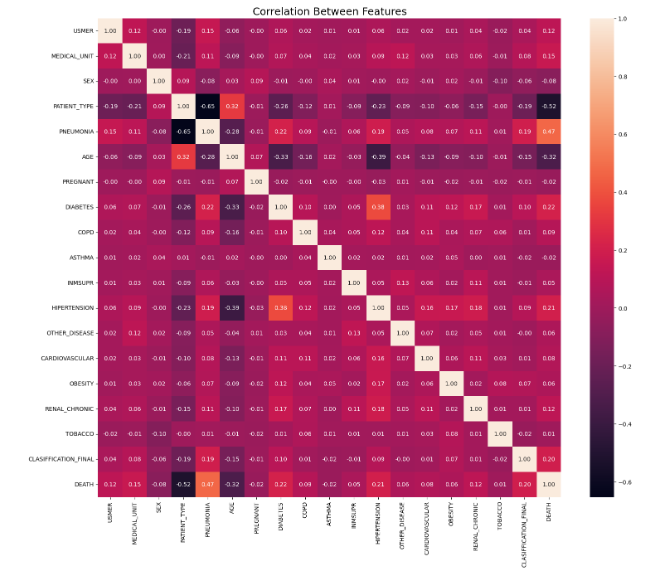


Figure 4 Correlation Between features

# Standardization of Age Data in COVID Dataset

The 'AGE' column in the Covid\_Data dataset was standardized using StandardScaler.

This process transformed the data so that it has a mean of 0 and a standard deviation of 1, which is essential for models sensitive to feature scales.

# Splitting Data into Features and Target:

The dataset was split into features (x) and the target variable (y). The target variable in this case was 'DEATH'.

Train-Test Split:

The dataset was further split into training and test sets using train\_test\_split with 20% of the data reserved for testing and the other 80 % for train.

# Implementing Logistic Regression and sampling the data

A Logistic Regression model was created and trained on the dataset.

Logistic Regression: Good baseline model for binary outcomes (e.g., death vs. recovery).

The model's performance was evaluated using accuracy and F1 score, which are key metrics for classification tasks.

We Address Data Imbalance so we implement the Random Under-Sampling, to balance the dataset by reducing the size of the majority class which is a common technique helps in handling class imbalance.

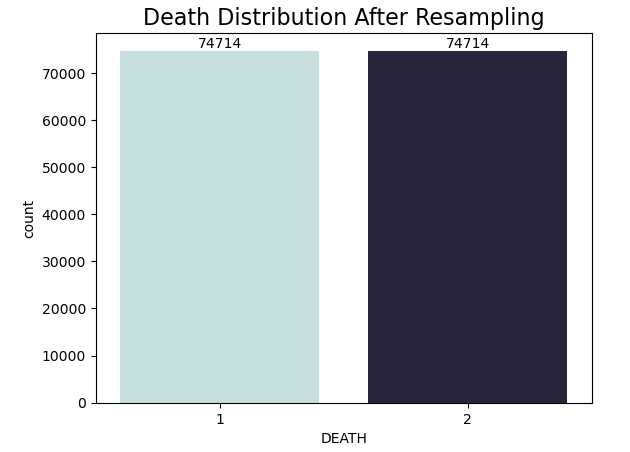
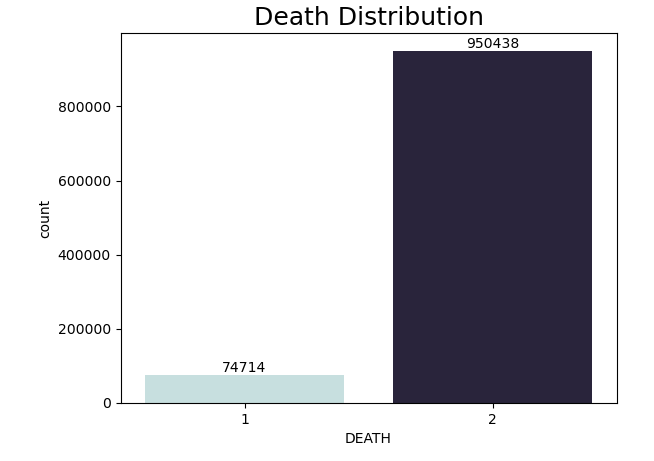


Figure 5 Before and after Resampling the data

We resampling the data, in our analysis of the confusion matrix revealed a significant imbalance, particularly in terms of true negatives. This imbalance was addressed through data resampling, which helps to equalize the representation of different classes. Resampling, the results showed a more balanced distribution of predictions, as evidenced in Figure 6. This adjustment improved the model's ability to accurately predict both classes.

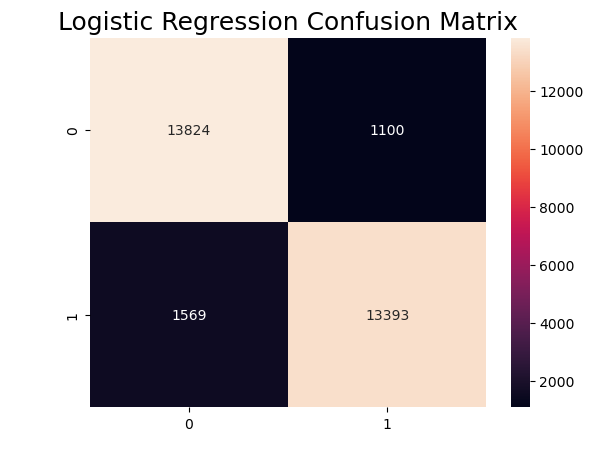
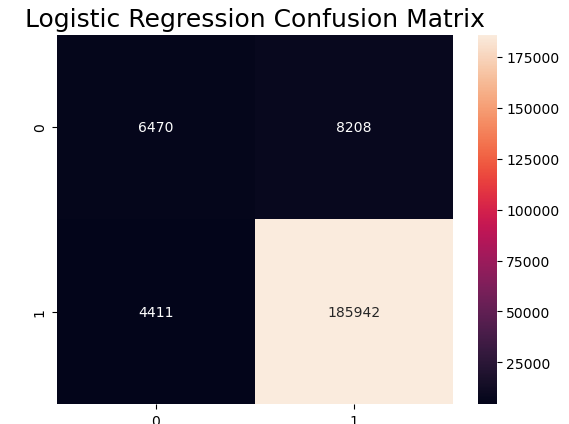


Figure 6 Matrix before and after the resampling the data

# Random Forest

Random Forest is good at figuring out which details about the patients (like their age or health conditions) are really important for understanding how COVID-19 affects them.

A Random Forest model with 100 trees was trained, offering the benefits of ensemble learning.

Random Forest Classifier: An ensemble method that can capture non-linear relationships and interactions between features.

# PCA

Using Principal Component Analysis (PCA) to identify the most significant features.

PCA was used for dimensionality reduction, transforming the dataset into a set with fewer features (components).

The optimal number of components was determined by analyzing the cumulative explained variance.4. Model Development

## Why is dimensionality reduction important in machine learning?

Dimensionality reduction in machine learning is important because reduces the "curse of dimensionality" where too many features make the model complex and less effective and helps models to work faster and more accurately by focusing on the most important features. (Sartorius, n.d.)

A Logistic Regression model was also applied to the PCA-reduced dataset, showcasing a different approach to handling high-dimensional data. Several models, including Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors, Decision Tree, and Gaussian Naive Bayes, were initialized for comparative analysis

Cross-validation was suggested to compare the performance of the models.

The comparison was to be visualized using a boxplot, providing insights into the accuracy and consistency of each model.

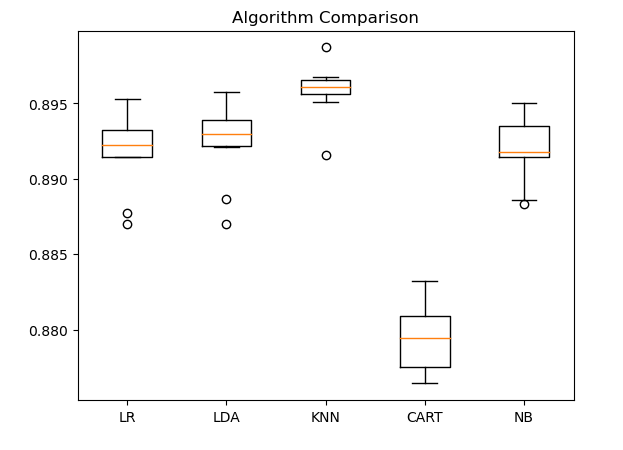


Figure 7 Algorithm Comparison

# KNN Model

KNN is known for its simplicity. It works by looking at the 'k' closest data points (or 'neighbors') to decide the outcome. Using KNN with 3 neighbors looking at the 3 nearest points to make a prediction.

Based on the idea of similarity patients had severe symptoms, it predicts the same for new and similar cases.

# For instance, when inputting data for a new patient into our KNN (K-Nearest Neighbors) model, the algorithm looks to the three most similar patients in the dataset. It examines these three 'nearest neighbors' and their respective outcomes. The prediction for the new patient is then made based on the results observed in these three closest counterparts

# GaussianNB and SVC (Support Vector Classification)

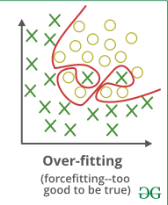
The GaussianNB model looks at each feature of the patient data and calculates the probability of each outcome (death or no death) based on each feature, if the combined probabilities across all features are higher for 'death', then GaussianNB predicts that the patient case will result in death. Otherwise, it predicts survival.

SVC works differently. It tries to find a boundary or a line in the data that separates the death cases from the non-death cases. This boundary is based on the entire dataset. If the patient data falls in relation to this boundary will determine SVC's prediction. If patient data falls on the side with the death cases, SVC predicts death; if it falls on the other side, it predicts no death.

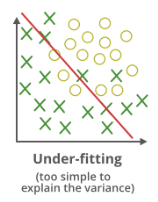
.GaussianNB does this by looking at each feature and calculating probabilities, while SVC looks at how the patient overall data compares to other cases in the dataset.

# Discussing Overfitting/Underfitting/Generalization

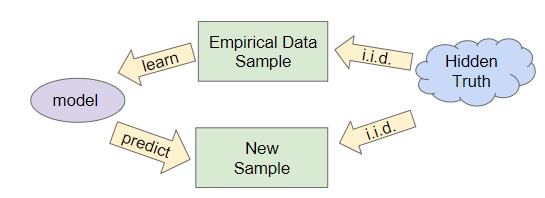
Overfitting: If your model shows high accuracy on the training data but poor performance on the test data, it may be overfitting. Overfitting occurs when a model learns the details and noise in the training data to the extent that it negatively impacts the performance of the model on new data. (Nautiyal, 2017)



Underfitting: If the model performs poorly on both training and test data, it might be underfitting. This means the model is too simple to capture the underlying patterns in the data. (Nautiyal, 2017)



Generalization: The ultimate goal is for the model to generalize well - to perform effectively on new, unseen data. Cross-validation methods are a good way to test for generalization.in the following figure que can observe the data that we have collected. It's called empirical because it's based on observed and measured phenomena and is used to train the model. This data is assumed to be a sample from a larger population or process, capturing the observable aspects of the "Hidden Truth" this is the underlying reality or process that we are trying to understand or make predictions about. It's called hidden because it includes aspects that might not be directly observable or fully understood. The model is trained on the empirical data sample with the goal of uncovering patterns that reflect the hidden truth After the model has been trained, it can be used to make predictions on new samples. (Google Developers, n.d.)



# Conclusion

This project aims to leverage machine learning to extract meaningful insights from COVID-19 data, thereby contributing to the understanding and management of the pandemic. The key is to balance technical rigor with practical significance, ensuring that the models developed are both statistically sound and relevant to real-world applications.

This series of code snippets represents a comprehensive machine learning pipeline, encompassing preprocessing, handling imbalanced data, model training and evaluation, and comparative analysis of multiple models. The approach demonstrates good practices such as standardization, dealing with imbalanced datasets, dimensionality reduction, and robust model evaluation techniques. The use of various models and metrics provides a thorough understanding of each model's performance, guiding the selection of the most appropriate model for the given dataset.

# Reference

www.kaggle.com. (n.d.). COVID-19 Dataset. [online] Available at: https://www.kaggle.com/datasets/meirnizri/covid19-dataset/data [Accessed 25 Nov. 2023].

Sartorius. (n.d.). What Is Principal Component Analysis (PCA) and How It Is Used? [online] Available at: <https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186#:~:text=PCA%20forms%20the%20basis%20of>.

Nautiyal, D. (2017). Underfitting and Overfitting in Machine Learning. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>.

Google Developers. (n.d.). Generalization | Machine Learning. [online] Available at: https://developers.google.com/machine-learning/crash-course/generalization/video-lecture#:~:text=Generalization%20refers%20to%20your%20model.