PROJECT REPORT

**Estimating COVID-19 Patient Admittance into the ICU**



**TEAM 9**

Gangisetti Spurthy Vahini - 6605

Surisetty Neha Keerthana - 6609

Baja Kusuma - 6613

Pallapu Koushika - 6617

Pranitha Durgam - 6626

Palwai Shiromani - 6647

# INTRODUCTION

The COVID-19 pandemic has caused us to rethink the organization of the health system. It has led to over burdening on many hospitals and they needed better software, better models to make things a little less complicated.

Due to the increasing cases, many people were constantly admitted into hospitals and the intensive care units were full. If the ICUs were handled properly then patients in need could be helped, this pioneered our project.

Approach:

The dataset is taken from the hospitals, which includes all the reports of patients and the record of the patients exiting and entering the hospital.

Using those particular records we predict whether the patient would need a bed or not. We can classify the patients into two classes, the patients who need bed and those who don’t.

Using a classifier the patients are divided into their respective classes.

Relevance to real world:

The classifier would help the hospital to allocate beds for every patient without falling short of them.

# TASK DEFINITION;

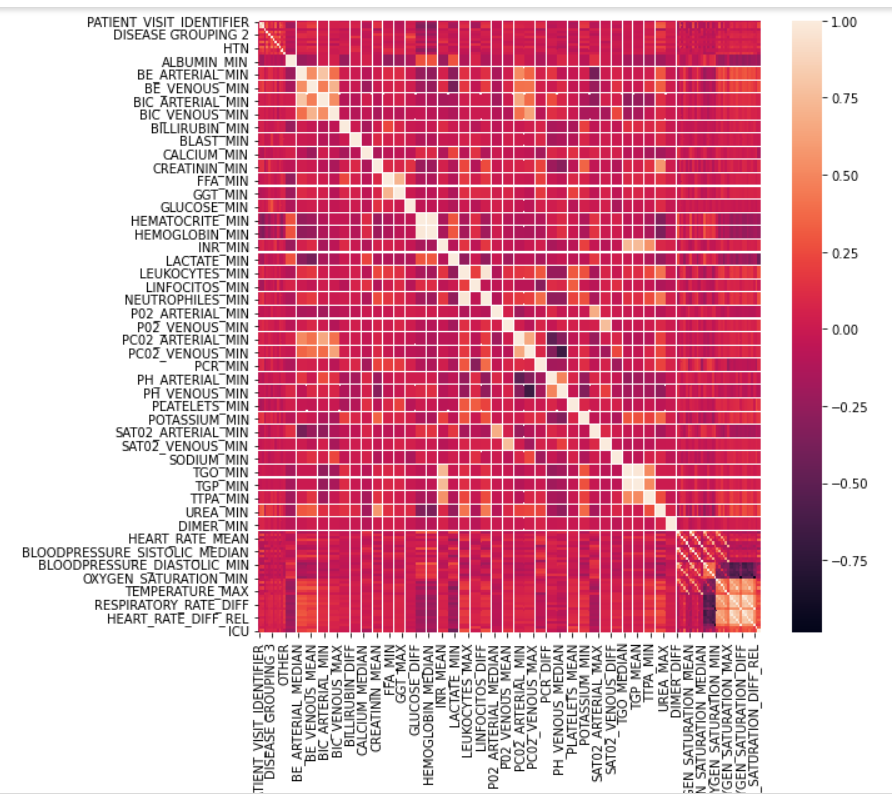
**DATASET**

* Patient demographic information
* Patient previous grouped diseases
* Blood results
* Vital signs
* Blood gasses

In total there are 42 features, expanded to the mean, max, min, diff and relative diff.

# METHODOLOGY

## Data Preprocessing:

* The third step was **FINDING THE CATEGORICAL DATA.** The majority of the data is numerical. So we had to change the rest of the data i.e, the categorical data and the string data into numerical data.
* The next step was to determine the number of nulls in the given dataset. It was identified that we had a total of 223863 nulls in our dataset.There are many ways to deal with nulls including MEAN,MEDIAN,MODE,Forward fill ,Backward fill etc. We have chosen the INTERPOLATE method to handle nulls .The main advantage of using the **INTERPOLATION**  is, with the help of this technique we can replace the missing value with the most accurate value that exists between the two known data points.
* **Constant columns** : If the same value exists within a set of columns, then it is suggestable to drop those constant columns. In order to drop these concol we have used VarianceThreshold where(threshold = 0) which indicates that the values in columns are constant. The constant columns have been appended into a list and then dropped.
* **Quasi Constant columns** : If a same value repeats more number of times within a set of columns, then it is suggestable to drop those quasi constant columns. In order to drop these concol we have used VarianceThreshold where(threshold = 0.01) which indicates that the values in columns are quasi constant. The quasi constant columns have been appended into a list and then dropped.
* The next task was **splitting the dataset**. So this gave us 70 percent of Training Data and 30 percent of Testing Data.
* **Using correlation for dropping columns** – A heatmap was plotted. It was indeed huge, since there were many columns(231). To drop the columns, a function in threshold was considered where a correlation matrix was taken. For every column in the correlation matrix we checked whether the given value is being matched,i.e,whether greater or otherwise. If greater,we are then adding this into a set and then dropping this complete column set.
* Keeping (threshold = 0.6),148 columns have been dropped and the accuracy observed was around 90.29 %
* Keeping (threshold = 0.7),142 columns have been dropped and the accuracy observed was around 90.81 %
* Keeping (threshold = 0.8),129 columns have been dropped and the accuracy observed was around 91.5 %
* Keeping (threshold = 0.9),124 columns have been dropped and the accuracy observed was around 92.02 %
* Keeping (threshold = 0.95),122 columns have been dropped and the accuracy observed was around 91.507 %
* Finally,there are 60 columns remaining based on the value of accuracy obtained. Then we have started Standardizing (fitting the data between 0 to 1).
* Why is Standardizationrequired?
  + Plotting a scatter plot for a non-Standardised dataset will give points that are very distant and different from each other which in turn might confuse the model. So with Standardization, since we are fitting the values between the range of 0 and 1, the data points will be clustered together that would help improve model efficiency. This is termed as **“Feature Scaling”.**

# MODELS USED

## DECISION TREE :

It is a tree-structured classifier, whereinternal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

## RANDOM FOREST :

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

## KNN :

The K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories. It stores all the available data and classifies a new data point based on the similarity. It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

## ADABOOST :

Boosting is an ensemble technique that attempts to create a strong classifier from a number of weak classifiers. AdaBoost is used to boost the performance of decision trees on binary classification problems. It is best used with weak learners. These are models that achieve accuracy just above random chance on a classification problem. The most suited and therefore most common algorithm used with AdaBoost are decision trees with one level.

## NAIVE BAYES :

[Naive Bayes](https://www.upgrad.com/blog/naive-bayes-classifier/) is a probabilistic [machine learning algorithm](https://www.upgrad.com/blog/top-machine-learning-algorithms-for-data-science/) used for many classification functions and is based on the Bayes theorem. Gaussian Naïve Bayes is the extension of naïve Bayes. While other functions are used to estimate data distribution, Gaussian or normal distribution is the simplest to implement as it is needed to calculate the mean and standard deviation for the training data.Typical applications of Naive Bayes are classification of documents, filtering spam, prediction and so on.

## SVM- SVC :

* SVC is a popular supervised learning algorithm. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
* SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.
* There are two types of SVM - linear and non-linear which is based on how linearly separable the data is.
* A plane is drawn through the support vectors which are known as marginal planes. These provide a clearer boundary for the classification to happen, the distance between them is known as margin
* The main objective for the model to work as efficiently as possible is to maximize the margin.
* To do so the data has to be pre-processed correctly and it can also be done using hyper parameter tuning
* The parameters we can tune are -
  + C = The Regularization Parameter tells the SVM optimization how much you want to avoid miss classifying each training example.

If the C is higher, the optimization will choose a smaller margin hyperplane, so training data miss classification rate will be lower.

On the other hand, if the C is low, then the margin will be big, even if there will be miss classified training data examples

* + Gamma = The gamma parameter defines how far the influence of a single training example reaches
  + Margin = Higher the margin, higher the accuracy of model

# RESULTS

| **MODEL NAME** | **ACCURACY FOR TRAIN** | **ACCURACY FOR TEST** |
| --- | --- | --- |
| **Decision Tree** | 100.0 | 82.33 |
| **Random Forest** | 100.0 | 91.16 |
| **KNN** | 100.0 | 89.6 |
| **AdaBoost** | 90.9 | 89.09 |
| **Guassian Naive Bayes** | 78.4 | 79.2 |
| **SVC** | 98.43 | 92.98 |

# 

# DISCUSSION:

Using the decision tree, it was observed that the accuracy for training was perfect but rather less when it was used for prediction

Using Random forest, the accuracy for the prediction was rather improved than the decision tree alone.Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction

KNN gave an 89% accuracy for the training dataset but it was rather less than the others.

The naive bayes algorithm gave the least accuracy for prediction.

Adaboost classifier also could give an accuracy of about 90% after various tuning of hyper parameters.

Using Support Vector Classifier, initially there was a 90 percent accuracy.To improve accuracy, grid search has been used which checks for the accurate hyper parameters to fit in the SVC i.e, the (C,gamma,kernel) values in order to improve accuracy. In this due process, we obtained **C = 10, gamma = 0.01,kernel = rbf**. Putting these values in the SVC, we have achieved **92 percent accuracy** which is the highest accuracy obtained.

The dataset provided by the hospitals is not organized since each patient could have different values and different number of visits to the hospital, thus making it not regularly distributed data. SVC works efficiently with such datasets.

In Classification problems, there is a strong assumption that is Data have samples that are linearly separable but with the introduction of kernel, **Input data can be converted into High dimensional data** avoiding the need of this assumption.  
K(x1, x2)=〈f(x1), f(x2)〉Where K is the kernel function, x1, x2 are n-dimensional inputs and f is a function that is used to map n-dimensional space into m-dimensional space and 〈x1, x2〉is used to specify/indicate the dot product.

This high dimensionality is very useful to find out an optimal hyperplane which can classify data efficiently.

The SVM provides a **very useful technique within it known as kernel** and by the application of associated kernel function we can solve any complex problem.

Although it had a few disadvantages such as underfitting of data which led to less accuracy it performed the best after working on the under fit.

Due to the above reasons SVC was found to give more accuracy than others.

# DEPLOYMENT

Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data

In order to start using a model for practical decision-making, it needs to be effectively deployed into production. If you cannot reliably get practical insights from the model, then the impact of the model is severely limited.

In order to get the most value out of machine learning models, it is important to seamlessly deploy them into production so a business can start using them to make practical decisions.

## DEPLOYMENT USING GRADIO:

There are several methods to deploy a model, one of them being using gradio. Gradio creates an user-friendly web page which can take the input and return the output to the user.

One of its advantages is that it allows you to interact with the web app you are currently developing in your Jupyter or Colab notebook. It has a lot of unique features that can help construct a web app that users can interact with.

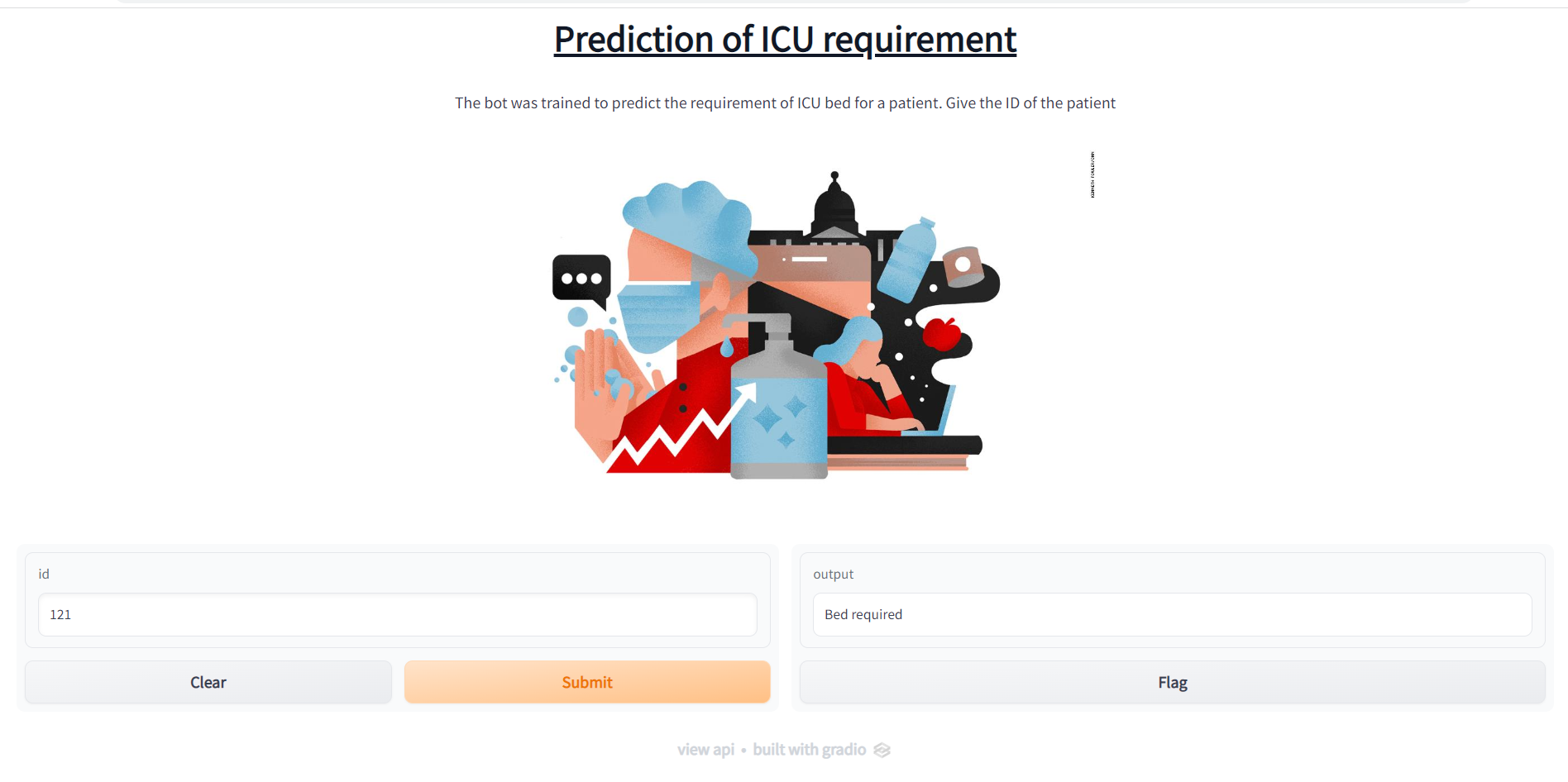
An interface is created using the interface functionality of gradio which is an amalgamation of all the functionalities a user would need to get an output.

#### Approach:

Gradio is designed in such a way that the user inputs the id of the patient from their database and the gradio interface returns if the patient requires a bed or not.

For such a function is written which returns prediction of the model which is pickled. The function is then linked to the interface and an input is taken with the datatype of a number. It returns a text box with the message if the bed is required or not.

Gradio interface:



# CONCLUSION

In conclusion, this kind of interface can be very useful for hospitals which are busy day and night to decide which patient could need an ICU and can see to it that the hospital doesn’t fall short of beds in such a crucial time.

The heavy database which a hospital provides can be efficiently handled by a classifier like Support vector classifier due to its kernel function which helps in deciding on an efficient hyperplane.

# REFERENCES

1. [Working with Missing Data in Pandas - GeeksforGeeks](https://www.geeksforgeeks.org/working-with-missing-data-in-pandas/#:~:text=Checking%20for%20missing%20values%20using,null%20values%20in%20a%20series).
2. [Plot Correlation Matrix and Heatmaps between columns using Pandas and Seaborn. - Knowledge Transfer](https://androidkt.com/plot-correlation-matrix-and-heatmaps-between-columns-using-pandas-and-seaborn/)
3. [sklearn.svm.SVC — scikit-learn 1.1.1 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html)
4. [SVM Hyperparameter Tuning using GridSearchCV | ML - GeeksforGeeks](https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/)
5. [seaborn.heatmap — seaborn 0.11.2 documentation](https://seaborn.pydata.org/generated/seaborn.heatmap.html)
6. [How to Build a GUI Using Gradio for Machine Learning Models](https://www.freecodecamp.org/news/build-gui-using-gradio-for-machine-learning-models/)