Coronavirus Visualization and Prediction Using Machine Learning Algorithm: Assessing Their Performance

Introduction

COVID-19 is an infection caused by a virus called the coronavirus. It spreads between individuals primarily through small droplets when someone coughs, sneezes, or talks.

Coronavirus affects two groups of people significantly. One group includes children and young individuals, while the other group consists of older adults or people already suffering from serious diseases.

- The majority of people who sign COVID-19 experience mild symptoms, like to a cold or mild flu. They may have a fever, cough, or feel tired. This group often recovers on their own without needing hospitalization.
- However, older adults (e.g., grandparents) or individuals with pre-existing health conditions (such as heart issues, diabetes, respiratory problems, or cancer) may become severely ill if they catch COVID-19. This group might require specialized care in a hospital.

Therefore, while many people recover easily, some are at a greater risk and need specialized care.

The Problem:

In hospitals, medical equipment (such as oxygen, beds, or medicines) was insufficient during the COVID-19 epidemic, and there was no effective plan for distributing them. When needed, it was extremely challenging for healthcare providers to assist everyone.

One key challenge that healthcare providers faced during the COVID-19 epidemic was the shortage of medical equipment and the lack of an appropriate plan for their effective distribution.

Why Prediction is Important/Real World Importance:

At the time of testing positive, prediction helps determine what kind of resources an individual might require. During these tough times, early prediction of necessary resources needed to save a patient's life helped authorities (hospitals and governments) arrange them effectively.

I. Contents:

1. **Dataset Source:**

   The experimental dataset was downloaded from Kaggle[4] and has been provided by the Mexican government[3] for the past two years.

2. **Dataset Size and Structure:**

The dataset structure includes 21 special features, as mentioned in Figure 1, and contains a total of 1,048,576 unique recorded patients. Each feature represents an

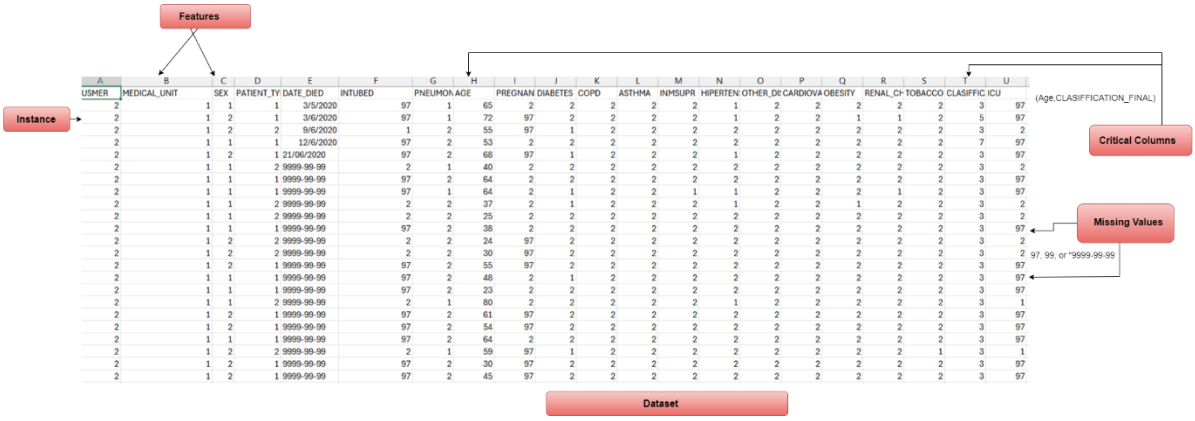


Figure 1:C0VID-19 Dataset CSV File

identifiable detail about a patient, such as age, gender, or medical conditions. Missing values in the dataset are denoted by 97 or 99, as shown in the CSV file screenshot in Figure1.

3. **Machine Learning Problem:**

The experimental dataset consists of a classification problem in machine learning.

The research goal is to predict the patient condition using CSV record the **patient is Covid effected or not**

II. Key Characteristics:

1. Features (Inputs):

As shown in the CSV file, the dataset consists of 21 unique features. These features represent information about patients, such as age, symptoms (e.g., pneumonia, asthma), and medical history (e.g., diabetes, hypertension). In the experiment, machine learning models will use these features as input.

2. Target Variables (Output):

The models will predict the output as a binary label (e.g., 1 = **Covid effected**, 2 = Not effected):

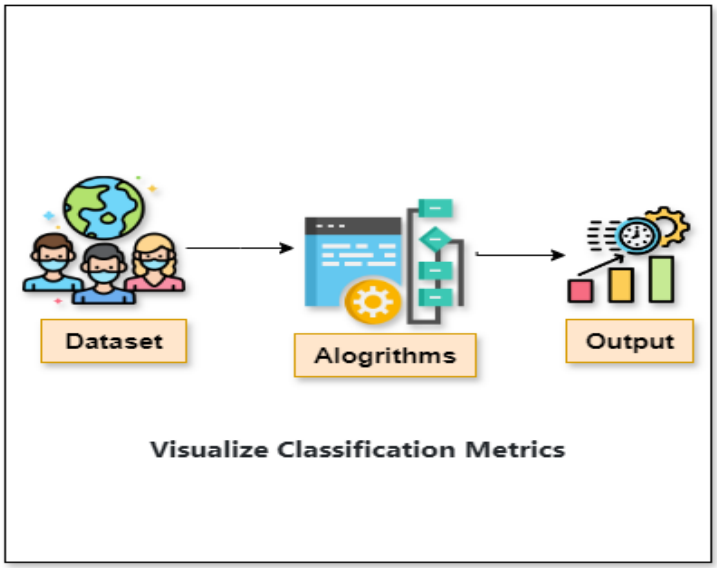
- **1 = Covid effected:** Patients with coronavirus
- **2 =** Not effected**:** Patients without coronavirus

Algorithms Selection and Implementation:

I. Objective:

In the experiment, the four machine learning models are applied to the dataset (COVID-19) and predict the patient condition based on their current symptoms, health status, and medical history that tells weather the patient is COVID-19 effected or not.

The experimental dataset deals with a classification problem. Since the target variable has two classes, it is a binary classification problem (predicting high risk or not). As is shown in visualization classification metrics.



II. Dataset Characteristics:

Based on the characteristics of my dataset mentioned below, I have selected five machine learning models:

- **Size:** Large (over 1 million records)
- **Missing values (Imbalanced dataset):** Present (denoted by 97 and 99).

III. Selected Algorithms:

The four supervised machine algorithms are selected

1. Logistic Regression:
2. Random Forests:
3. Support Vector Machines (SVM):
4. K-Nearest Neighbors (K-NN)
5. Naive Bayes

Logistic Regression:

The most simplest model for binary classification is Logistic Regression. Based on the characteristics of my dataset, it is a well-suited model. Its strengths include being fast to train and easy to implement. However, this may lead to overfitting when number of rows is less than the number of features.

Random Forests:

Random Forest is also a well-suited model based on the characteristics of my dataset, as it performs well with large datasets and handles missing values effectively. This model is an ensemble of decision trees, which improves accuracy. However, it is slower to train compared to a single decision tree.

Support Vector Machines (SVM) :

SVM is not the best-suited model based on the characteristics of my dataset because it struggles with large datasets and imbalanced data. However, I have selected it to check its performance and compare it with other models. This model works well for small to medium datasets with clear margins.

K-Nearest Neighbors (K-NN)

KNN is also not the best-suited model based on the characteristics of my dataset. Since I have a large dataset, this model is slower when there is a lot of data, as it compares every new point to all previous points. However, I have selected it to check its performance and compare it with other models. This model is easy to understand and implement.

Naive Bayes

The Naive Bayes algorithm is also not the best-suited model based on the characteristics of my dataset because it struggles with imbalanced datasets. It tends to favor the majority class, resulting in poor predictive performance for minority classes. This is caused particularly problematic in applications such as fraud detection or disease diagnosis, where the minority class is of greater interest. However, this model is known for its simplicity, efficiency, and effectiveness in handling large datasets. This is a powerful classification algorithm

Results and Discussion/Comparative Analysis:

For the comparative evaluation, the dataset is preprocessed, and feature engineering is performed. As shown in the predictive model construction process, the dataset is first loaded and checked for missing values. This comparative analysis is performed using the Python programming language. The library needed for handling the dataset is: import pandas as pd.

Data Cleaning

After that, dataset cleaning is performed by addressing the missing values. Even if the dataset does not have NaN (missing) values, it contains placeholders such as 97, 98, 99, or "9999-99-99". We

checked our CSV file, as shown in Figure 1. In the dataset contents, it is mentioned that placeholders like 97, 99, or "9999-99-99" need to be handled.

We have handled the other columns (numeric and categorical). The numerical column is addressed by filling in the missing values with appropriate numeric values, while the categorical column is handled by filling in the mode (the most frequent value). Finally, the dataset is checked to ensure that no missing values remain.

Dataset Preprocessing:

In dataset preprocessing, the first step is to **conversion of data types** to ensure all columns have the correct data types (e.g., datetime, category, int, float). Each column in the dataset has a datatype that determines how the program treats that data (e.g., text, number, or date). Sometimes the data is not in the correct format, so it needs to be converted.

Next, we **encode categorical variables** by applying a one-hot encoding scheme. This scheme is used to handle categorical data (e.g., "SEX" or "PATIENT_TYPE"), meaning these columns are considered categories (e.g., "Male" vs. "Female" or different patient types). These labels are not directly understood by machine learning models, as models like linear regression and decision trees only work with numeric values. Therefore, the categorical data must be converted into numbers. Various encoding schemes can be used, but we selected one-hot encoding. This approach creates a new column for each category and marks the presence of each category with a 1 or 0. For example, a column for "SEX" with two categories ("Male" and "Female") would generate two new columns: SEX_Male and SEX_Female.

Feature scaling is another important step in preprocessing, where numerical features are standardized or normalized. Numerical features (e.g., age or blood pressure) are adjusted using feature scaling to bring them to a common scale. This is crucial because some machine learning models (e.g., k-NN, SVM) perform better when features are on the same scale. In our experiment, standardized feature scaling was implemented.

Finally, the data is **splited into train and test sets**. Where 80% of the data is allocated for training and 20% for testing. The step-by-step Process to analyze the COVID-19 dataset

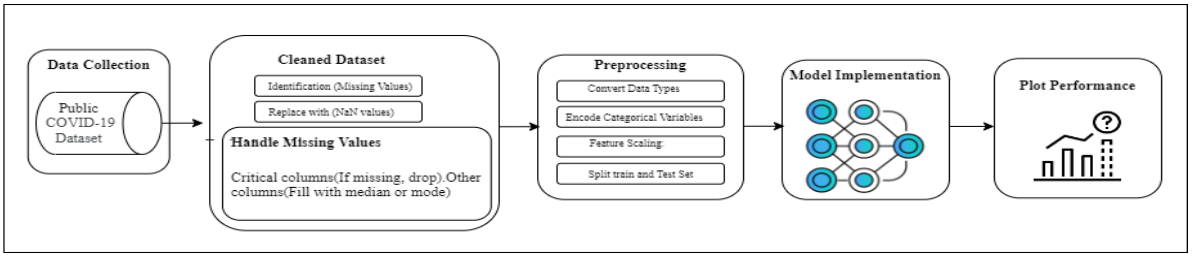


Figure 2:The step-by-step Process to analyze the COVID-19 dataset

Results:

The five supervised machine learning algorithms performance is evaluated on performance metrics. The metrics include Accuracy, Precision, Recall, F1-score (for classification tasks).

Logistic Regression Algorithm result mentioned below where the models will predict output as binary label((e.g., **1 = Covid effected:**, **2 = Not effected**)

- **1 = Covid effected: mean patients with** coronavirus
- **2 = Not effected patients with no** coronavirus

Precision [1] is how many positive predictions calculated by the model correctly. Precision equation is

$$Precision = \frac{\text{True Positives(TP)}}{\text{True Positives(TP)} + \text{False Positive(FP)}}$$

If model calculate high precision mean model makes fewer false positive errors. The model calculated 51% precision for class 1= covid effected and 99% precision for second class 2= not effected mean mostly people have no coronavirus diagnose.

Recall [1] means actual positive cases were correctly identified by the model. Recall is calculated by the equation:

$$Recall = \frac{\text{True Positives(TP)}}{\text{True Positives(TP)} + \text{False Positive(FP)}}$$

F1-score is the combination of both Precision and recall. It depends when recall and precision is high f1-score wiil be high as f1-score for class 1=Covid effected is 0.23(23%) mean low f1-score.Accuracy is not a good metrics f1-score is better than accuracy [2]. It is matheticallly written as

$$f1 - score = \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

It combines the precision and recall scores of a model.

The model determines recall 15% for class 1=Covid Effected. Total support cases 3296 means the model correctly identifies 15 cases only . And for class 2=not effected, model determines 100% causes total support cases 206350.High recall means most of the actual positive cases are identified by the model. Second class has high recall and base on high recall and f1-score model accuracy is 98 % which means model tells most of the people belong to **2 = Not effected patients with no** coronavirus. As our priority is class 1= covid effected detection this model performance (f1-score) is not good for this class so logistic regression model is not best choice for this dataset

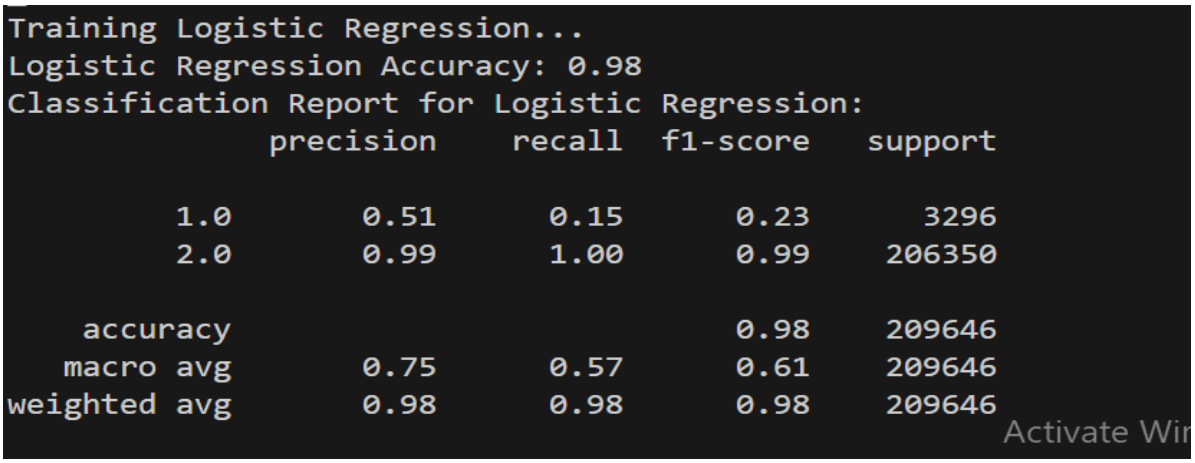


Figure 3:Logistic Regression Algorithm Results

Random Forest Algorithm result is mentioned below where the models will predict output as binary label ((e.g., **1 = Covid effected**, **2 = Not effected**)

- **1 = Covid effected mean patients with** coronavirus
- **2 = Not effected patients with no** coronavirus

The model determines recall 32% for class 1=Covid effected. Total support cases 3296 which means model correctly identifies 32 cases only . And for class 2=not effected, model determines 99% causes total support cases 206350.High recall means most of the actual positive cases are identified by the model. Second class has high recall and base on high recall and f1-score model accuracy is 98 % which means model tells most of the people belong to **2 = Not effected patients with no** coronavirus. If we compare with logistic regression model performance Random Forest improved the detection of Covid effected cases. As recall increased from **15%** to **32%**. **F1-Score** improved from **0.23** to **0.39**. According to these results the Random Forest model is **better at identifying 'Covid effected' cases.**

```
Training Random Forest...
Random Forest Accuracy: 0.98
Classification Report for Random Forest:
              precision    recall  f1-score   support

         1.0       0.50      0.32      0.39      3296
         2.0       0.99      0.99      0.99    206350

    accuracy                           0.98    209646
   macro avg       0.74      0.66      0.69    209646
weighted avg       0.98      0.98      0.98    209646
```

Figure 4:Random Forest Algorithm Results

Naïve Bayes Algorithm result mentioned below where the models will predict output as binary label((e.g., **1 = Covid effected**, **2 = Not effected**)

- **1 = Covid effected mean patients with** coronavirus
- **2 = Not effected patients with no** coronavirus

The model determines recall 98% for class 1=Covid effected. Total support cases 3296 that is a good indication mean model correctly identifies 98 cases But it drop the precision which is 0.09 that means model drop the balance between precision and recall and by considering these two factors model calculate very poor f1-score 16 % only for class 1. Model accuracy is 84% indicates that most of the patients belongs to class **2 = Not effected patients with no** coronavirus. This model is also not a good choice for our dataset

```
Naive Bayes Accuracy: 0.84
Classification Report for Naive Bayes:
              precision    recall  f1-score   support

         1.0       0.09      0.98      0.16      3296
         2.0       1.00      0.84      0.91    206350

    accuracy                           0.84    209646
   macro avg       0.54      0.91      0.54    209646
weighted avg       0.99      0.84      0.90    209646
```


Figure 5:Naive Bayes Algorithm Result

K-Nearest Neighbors Algorithm result mentioned below where the models will predict output as binary label((e.g., **1 = Covid effected**, **2 = Not effected**)

- **1 = Covid effected mean patients with** coronavirus
- **2 = Not effected patients with no** coronavirus

The model determines recall 27% for class 1=covid effected. Total support cases 3296 that is not a good score for class 1=Covid effected and precision is 0.54 which is good scoure for but f1-score is very poor for class 1 (36 %).The model accuracy is 98 % base on class 2=Not effected performance(Precision, recall,and f1-sore).If we compare model performance with other algorithms it improves precision for "Covid effected" (**0.54**), but recall drops to **27%**..To some extent better than Logistic Regression but less effective than Random Forest.

```
Training K-Nearest Neighbors...
K-Nearest Neighbors Accuracy: 0.98
Classification Report for K-Nearest Neighbors:
              precision    recall  f1-score   support

         1.0       0.54      0.27      0.36      3296
         2.0       0.99      1.00      0.99    206350

    accuracy                           0.98    209646
   macro avg       0.76      0.63      0.68    209646
weighted avg       0.98      0.98      0.98    209646
```

Figure 6:KNN Model Performance

According to all models metrics performance for both classes graphical chart is plotted in figure 7 which represents each of four models accuracy (%), training time(Normalized) and complexity(Normalized).Which shows which model is best and why.KNN Model takes 2.6 seconds to train and complexity is 1.00 and its accuracy is 98.49 %. The logistics regression accuracy is 98.44% and training time is 84.42second which is the highest training time as compared to other models and the complexity of model is 1.00. Random Forest model takes 0.16second for training and the complexity of model is 2.0 this model accuracy is 98.41 %.Naive Bayes algorithm accuracy is 83.95% which is lowest accuracy result as compared to other models and its training time is 0.55 seconds and its complexity is 1.00.

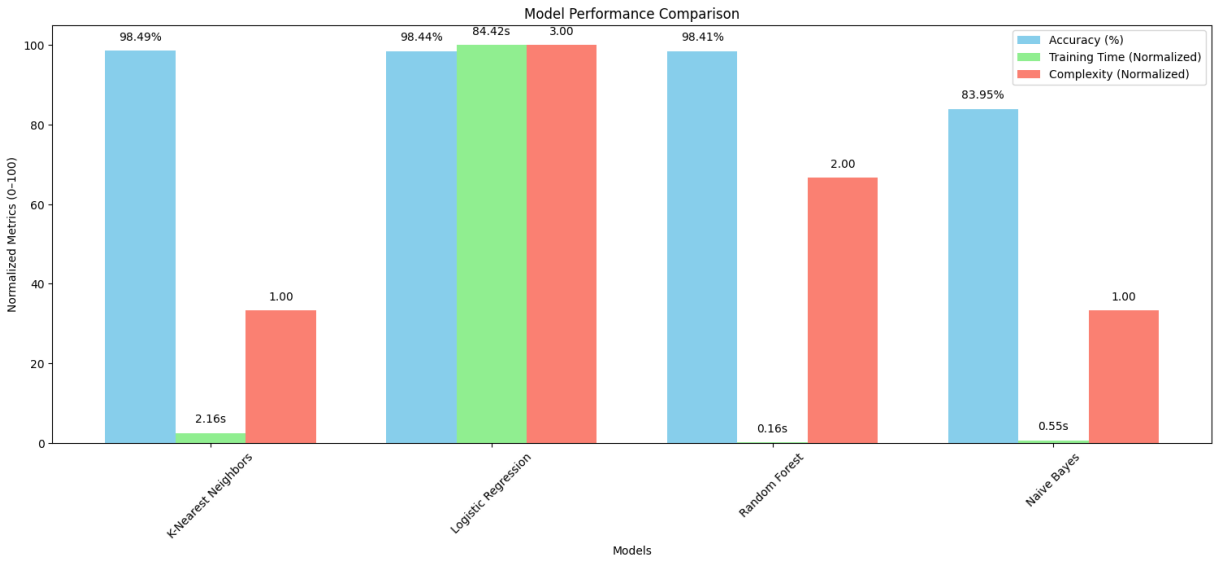


Figure 7:"Model Performance Comparison: Accuracy, Training Time, and Complexity for K-Nearest Neighbors, Logistic Regression, Random Forest, and Naive Bayes

Conclusion:

Random Forest model is the best choice for our dataset because it shows a good balance between performance (accuracy, precision, recall) for **Class 2 (Low Risk)** and illustrate reasonable performance for **Class 1 (High Risk).**However, Random Forest did not have perfect recall for Class 1,on our dataset this model is better at handling imbalanced data as compared to other four models performance across all metrics. If our main priorities were speed and simplicity then **Naive Bayes** can be used, but with lower accuracy. However if our main priorities where training time is not a concern. Then KNN is the best choice.

References:

1. Precision | Definition, Precision Vs Accuracy, Recall, Formula and Example
2. Confusion Matrix, Accuracy, Precision, Recall, F1 Score | by Harikrishnan N B | Analytics Vidhya | Medium
3. Datos Abiertos de México - Información referente a casos COVID-19 en México
4. COVID-19 Dataset