Introduction

Coronavirus, also known as COVID-19, is an infectious disease that spreads from person to person. It can spread through a COVID-19 patient's cough, sneeze, or respiratory droplets. It moves from one person to another or on a substance through a carrier.Therefore, it is always advised to often wash our hands to avoid contracting this virus. Wear a mask, avoid unnecessary contact with your mouth or nose, and practice social distancing because it enters through those two points.In this age of automation, artificial intelligence and data science are playing a crucial role in the health-care industry. Medical personnel may easily manage their tasks and patient care because these technologies are so interconnected. Every health-care institution is working hard to develop an automated system that can be used to deal with difficulties in this field. Machine learning (ML) is being developed by scientists to create smart solutions for disease diagnosis and treatment. ML can diagnose disease and viral infections more accurately, allowing patients' illnesses can be recognized earlier, hazardous phases of diseases to be avoided, and fewer people to be treated. ML can also help predict COVID-19 infection and estimate future COVID-19 infection counts.

Our goal is to identify the best-performing machine learning model for predicting COVID-19 among Logistic Regression Classifier, Random Forest Classifier, and XGBoost Classifier.

Data Collection & Processing

Dataset Description

The data of all people who received RT-PCR nasopharyngeal swab screening for SARS-CoV-2 were made public by the Israeli Ministry of Health[4].We have used this dataset in our project. In our Dataset, data is collected from March 2020 - November 2021. The dataset has 10 columns. Among these columns symptoms like Cough, Fever, Sore Throat, Shortness of Breath,Headache are the features. Gender, Age 60 and above, Test indication Test date is counted as other features.And Corona Result is the target feature of our work.After omitting rows with missing values and dropping all rows where corona_result = Other to keep the problem as binary classification there is 5861480 rows in our dataset.

Data Prepossessing

Label Encoding

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better

way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.For instance we have encoded positive corona_result as 1 and negative corona_result as 0,female to 0 and male to 1 etc.

One Hot Encoding

One-hot encoding is used in machine learning as a method to quantify categorical data. In short, this method produces a vector with length equal to the number of categories in the data set.We have used one hot encoding in our test_indication column.

Risk Coefficient

We have extracted Risk coefficient from the data. We observed that fever was the most common symptom found among covid positive patients so we gave 0.2 weight for it. Contact with covid patient would directly lead to home quarantine so we gave 0.2 weight for it. Contact with covid patient would directly lead to home quarantine so we gave 0.2 weight for it. Removed Contradictory records using risk coeficient. And the use of risk coeficient is over.

Data Modeling

Correlation using visualization

We can see the correlation map of the features in figure 3.1.None of the features are highly correlated.

Analyzing Target Feature

We have counted positive and negative cases before balancing the dataset and got 162021 covid positive cases, and 5337010 covid negative cases.This dataset is imbalanced and need to be balanced to do a better performance.

Undersampling the Data

To balance the dataset we decided to do Undersampling instead of Oversampling because -data is abundant for negative cases and increasing positive cases by oversampling would be an issue according to the real world scenario.

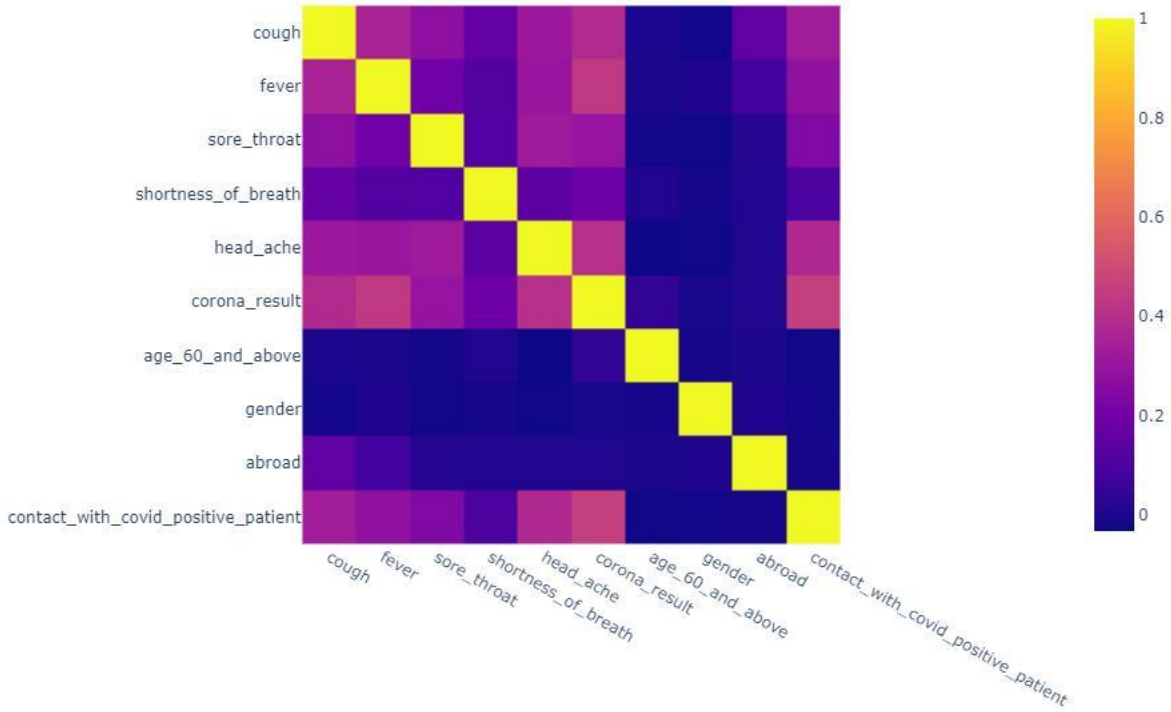


Figure 3.1: Correlation Map

3.3.4 Analyzing Target Feature after undersampling

After undersampling the dataset got balanced. There are 162021 covid positive cases amd 270035 covid negative cases.

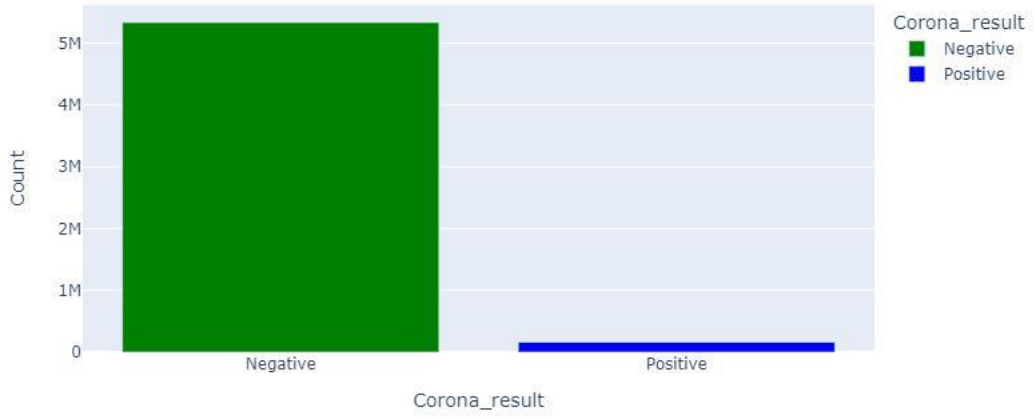


Figure 3.2: Analyzing Target Feature

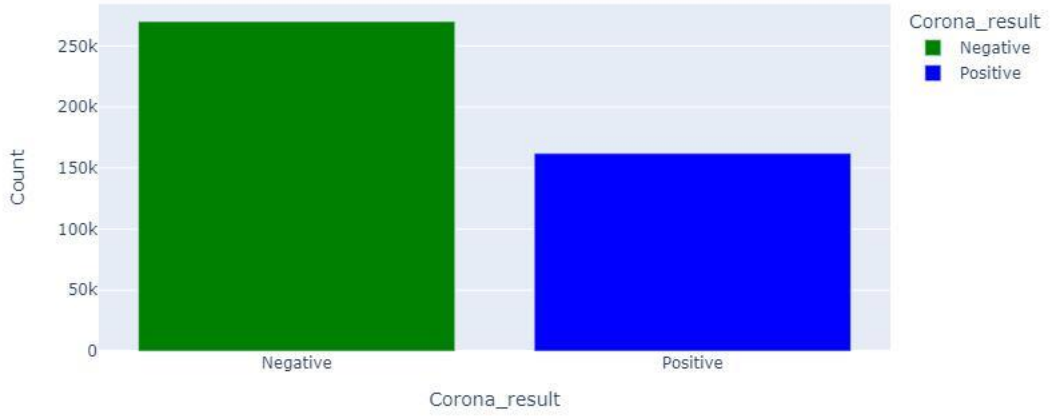


Figure 3.3: Analyzing Target Feature after undersampling

Chapter 4

Methodology

Figure 4.1 shows our working procedure.At first we choose our dataset.Rest of the procedure are described below:

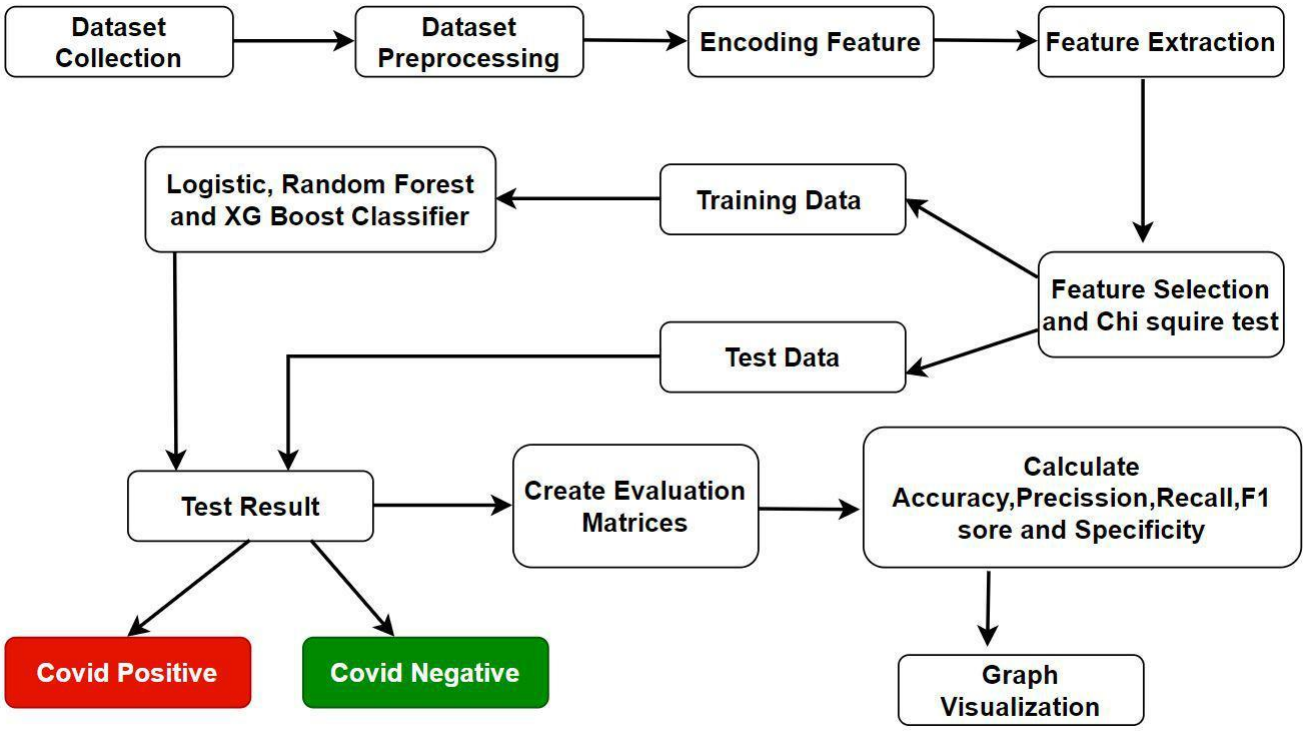


Figure 4.1: Work Flow of Detection positive or negative classes

4.1 Dataset Preprocessing

Our dataset has 5861480 records and 10 features.This is a Binary Classification Problem.Target Feature is Corona_result.Data is completely Categorical except for the test_date feature.First we preprocessed the data.So that it has no null values and has two class to be predicted.

4.2 Encoding The Features

We have used one hot encoding in our test_indication column.After that we observed if any of the features are correlated or not.

4.3 Feature Extraction

We extracted the Risk coefficient from the data. Contradictory records were removed using risk coefficient.When the use of risk coefficient is over this column was dropped.Also data type for all features are converted to integer.

4.4 Feature Selection

Since our Data is completely categorical we used the Chi Square Test to check for all features whether they contribute in detecting covid cases or not.It classified important & unimportant features.And no feature seems unimportant. All the features are contributing towards the detection of covid cases.

4.5 Data Modeling

In our project it is applied for 3 models.We have implement three classifier on our dataset and used gridSearchCv on that models.The classifiers are –

- Logistic Regression Classifier
- Random Forest Classifier
- XGBoost Classifier

GridSearchCv is a process of performing hyperparameter tuning in order to determine the optimal values for a given model.

4.6 Evaluation with Performance Metrics

We have used 5 metrics to evaluate our Models performnace.Besides we have used mean squared error,root mean squared error and Receiver Operating Characteristics (ROC) Curve.

- **Accuracy**
  It's the ratio of the correctly labeled subjects to the whole pool of subjects. Accuracy is the most intuitive one. Accuracy answers how many subjects did we correctly label out of all the subjects.

  $$\text{Accuracy} = (TP+TN)/(TP+FP+FN+TN)$$

  where, numerator is the all correctly labeled subject (All trues) and denominator is all the subjects

- **Precision**
  Precision is the ratio of correctly predicted positive classes to all items predicted to be positive. Or we can say it let us know how many of those whom we labeled as covid affected are actually affected?It is quite normal that for a disease detection there would be a big difference between number of positive and negative class.There will be more negative class than positive.Precision does better in this situation,because it does not include true negative in its calculation and is not effected by the imbalance.

  $$\text{Precision} = TP/(TP+FP)$$

- **F1 score**
  F1 Score considers both precision and recall. It is the harmonic mean(average) of the precision and recall.

  $$\text{F1 Score} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

  It is best if there is some sort of balance between precision and recall in the system. Oppositely F1 Score isn't so high if one measure is improved at the expense of the other. For example, if precision is 1 recall is 0 F1 score is 0.

- **Recall**
  Recall is the ratio of correctly predicted positive classes to all items that are actually positive.

  $$\text{Recall} = TP/(TP+FN)$$

  It let us know about what proportion of actual positives was identified correctly.it is important when we believe False Negatives are more important than False Positives.

- **Specificity**
  Specificity is the correctly -ve labeled by the program to all who are not affected with the virus.It answers out of all the people who are not affected, how many of those did we correctly predict.

$$\text{Specificity} = TN/(TN+FP)$$

- **MSE(Mean Squared Error)**It is the average square difference between the predicted values and the actual values.It is one kind of loss or risk function.It is always a positive value as we squared it.
- **ROC(Receiver Operating Characteristic curve)** ROC shows the performance of a classifier at all classification thresholds.To plot this curve we put false positive rate in x-axis and true positive rate in y axis.It let us know at which threshold point the model gives optimize,best or worst result.

Experiments and Results

Models Applied

Logistic Regression Classifier

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the ‘Sigmoid function’ or also known as the ‘logistic function’ instead of a linear function. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of Logistic Regression. [6]

Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. [7]

XGBoost Classifier

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. It's vital to an understanding of XGBoost to first grasp the machine learning concepts and algorithms that XGBoost builds upon: supervised machine learning, decision trees, en-semble learning, and gradient boosting. Supervised machine learning uses algorithms to train a model to find patterns in a dataset with labels and features and then uses the trained model to predict the labels on a new dataset's features. [8]

GridSearchCv

GridSearchCV is a model selection step and this should be done after Data Processing tasks. It is always good to compare the performances of Tuned and Untuned Models. This will cost us the time and expense but will surely give us the best results. The scikit-learn API is a great resource in case of any help. It's always good to learn by doing. [9]

Chi Square Test

In feature selection, we aim to select the features which are highly dependent on the response.A chi-square test is used in statistics to test the independence of two events. Given the data of two variables, we can get observed count O and expected count E. Chi-Square measures how expected count E and observed count O deviates each other[10]. Steps to perform the Chi-Square Test:

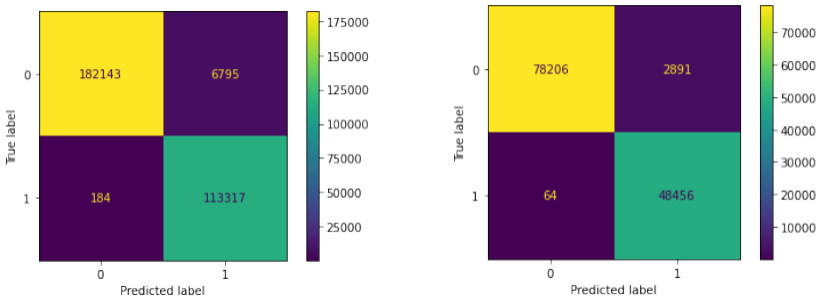
- Define Hypothesis
- Build a Contingency table
- Find the expected values
- Calculate the Chi-Square statistic
- Accept or Reject the Null Hypothesis

Result

In table 5.1 we can see that all the models have similar performance.And train and test results are also quite similar for each model.XGBoost has the maximum test accuracy which is 97.75,it has almost 100% recall also.Precisopn,F1 score for both train and test of all the models are equal upto 1 digit after decimal.

Table 5.1: Performance metrices of models

Classifier	Data	Accuracy	Precision	Recall	Specificity	F1 score
Logistic Regression	Train Test	97.69 97.72	94.34 94.37	99.84 99.87	96.39 96.44	97.01 97.04
Random Forest	Train Test	97.73 97.74	94.32 94.35	99.98 99.96	96.39 96.42	97.07 97.07
XG Boost	Train Test	97.73 97.75	94.32 94.35	99.98 99.97	96.38 96.42	97.07 97.08



(a) Train (b) Test

Figure 5.1: Evaluation Matrix for Logistic Regression Classifier

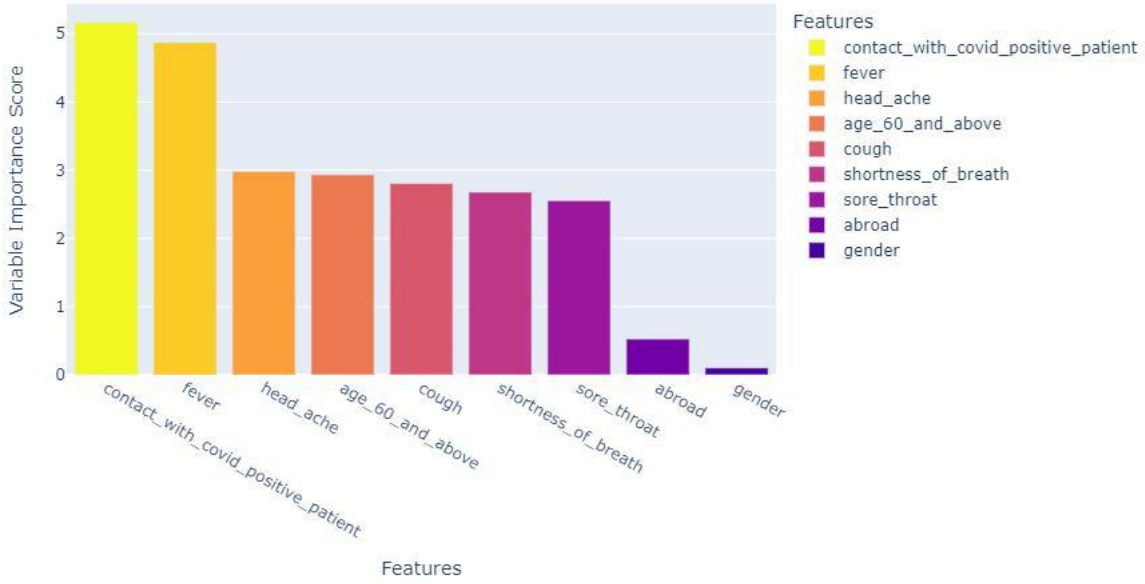


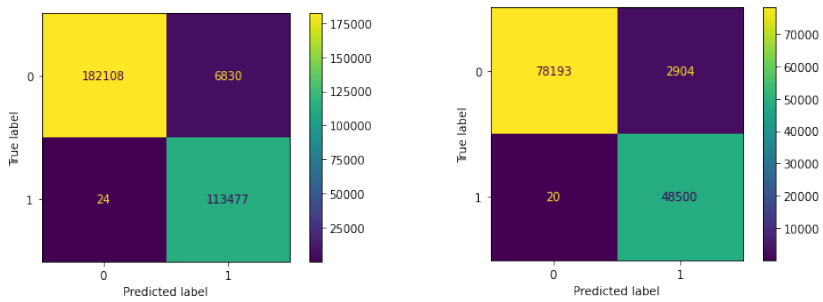
Figure 5.2: Variable Importance in Logistic Regression Classifier

Logistic Regression Classifier

Figure 5.1 shows train and test confusion matrix of logistic regression classifier.It has cor-rectly classified 126662 test samples.Figure 5.2 shows the feature improtance.Contact with covid patient has the highest importance.

Random Forest Classifier

Figure 5.3 shows train and test confusion matrix of random forest classifier.It has correctly classified 126693 test samples.Figure 5.4 shows the feature improtance.Contact with covid patient has the highest importance here also.



(a) Train (b) Test

Figure 5.3: Evaluation Matrix for Random Forest Classifier

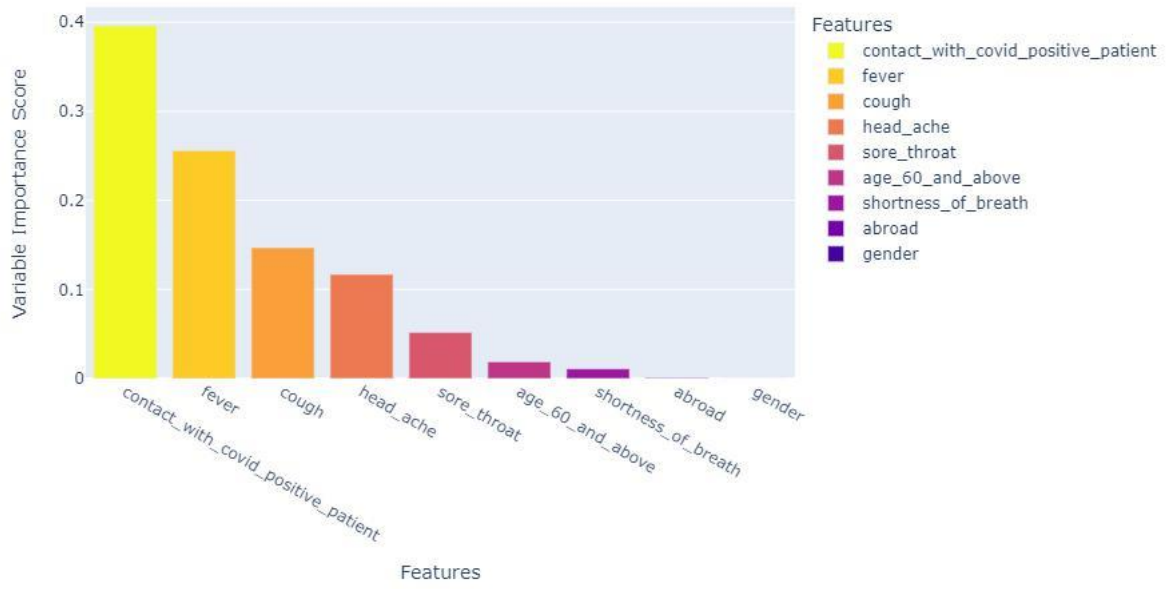
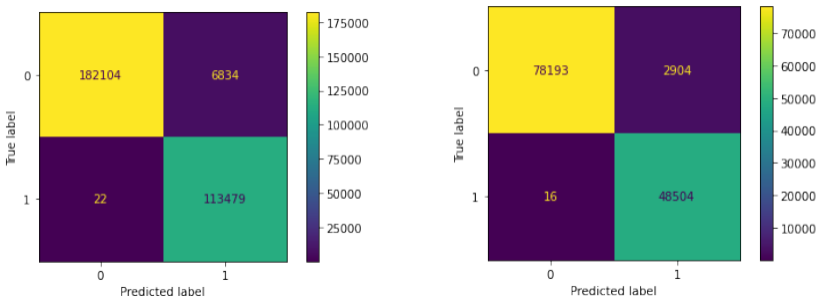


Figure 5.4: Variable Importance in Random Forest Classifier

XGBoost Classifier

Figure 5.5 shows train and test confusion matrix of XGBoost classifier.It has correctly clas-sified 126697 test samples.Figure 5.6 shows the feature improtance.Contact with covid pa-tient has the highest importance.Fever holds the second highest position.Rest of all features have very low importance in this model. []



(a) Train (b) Test

Figure 5.5: Evaluation Matrix for XGBoost Classifier

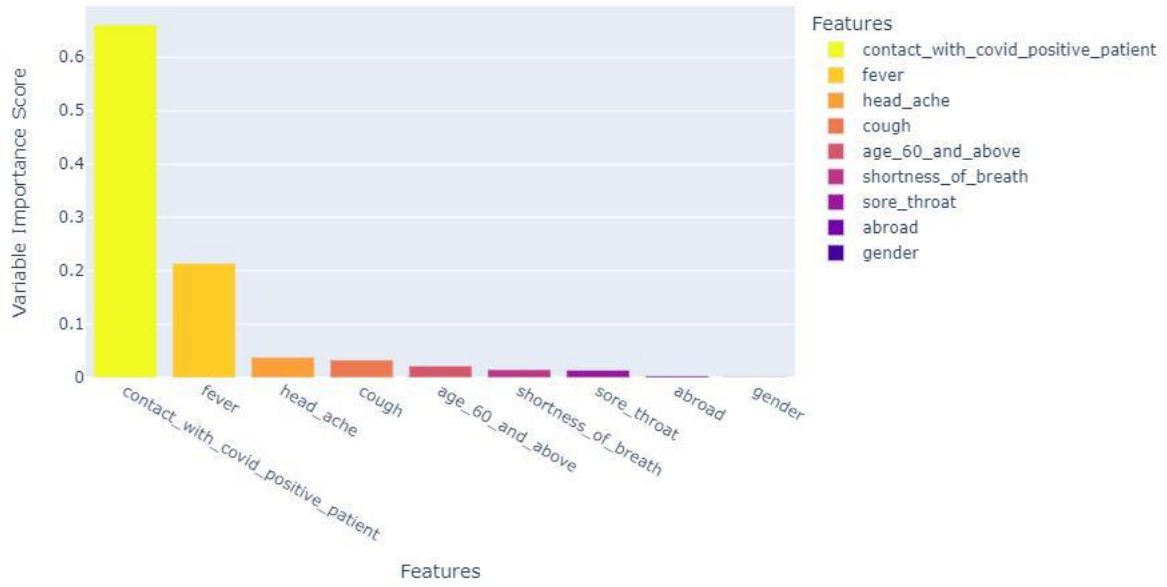
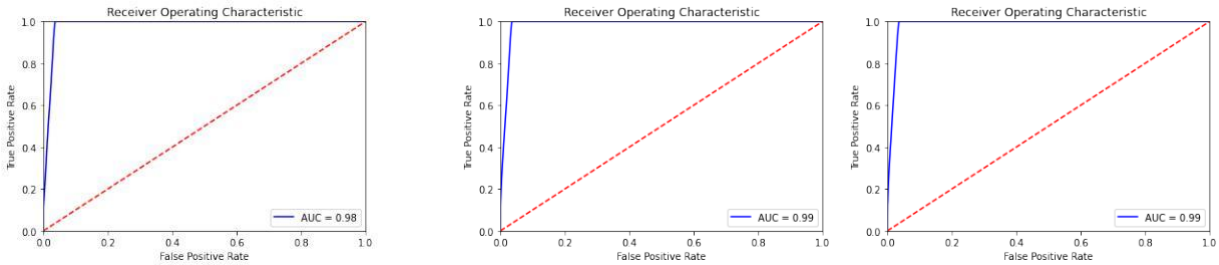


Figure 5.6: Variable Importance in XGBoost Classifier



(a) Logistic Regression (b) Random Forest (c) XGBoost

Figure 5.7: ROC curve of three Classifiers

Table 5.2 shows Mean Squared Error(MSE) and Root Mean Squared Error(RMSE) of three models.From this values we can know the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.Lower MSE or RMSE represent less loss and good accuracy of the model.

Table 5.2: MSE & RMSE of our models

Classifier	MSE	RMSE
Logistic Regression	0.022797935455997283	0.15098985216231348
Random Forest	0.022558769297237245	0.15019576990460565
XG Boost	0.02252790914771982	0.15009300166136935