**Preprocessing and Feature selection:**

**Preprocessing:**

In our datasets integer value 97 and 99 is null or missing value. We dropped all such rows which has missing values in a certain column.

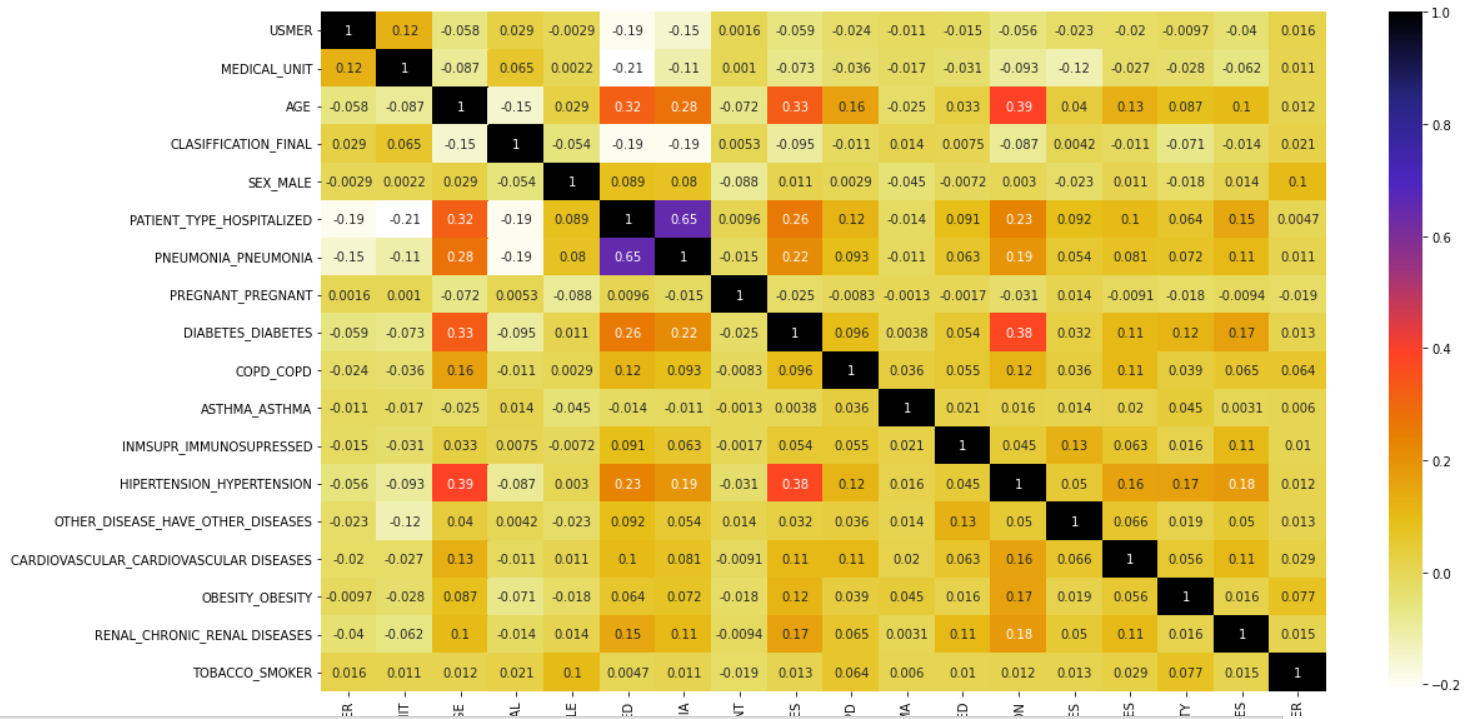
We dropped columns INTUBED, ICU because they had more than 50% missing values.

**One hot encoding:** One hot encoding is a encoding method is numerical value into a boolean value. If a column has 3 different categories, we would create 3 different features each having a Boolean value. Finally, we can drop a feature out of the three to avoid multicolinearity. We used get\_dummies function of pandas library to create dummy columns.

**Feature Selection:**

We used Pearson correlation coefficient to select features for our model training. We dropped all the features which has low correlation with our target columns ‘Classifcation Final’. We also dropped features except one if a number of features has high level of correlation among them.

**Image of Pearson Correlation Index:**



**Algorithm :** We have used three different classification algorithms in our dataset. Those are Decision tree classifier, Random Forest Classifier and Xgboost Classifier. A detailed description of these three algorithms are given below.

**Decision Tree Classifier:** There are two kinds of node in decision tree, Decision node and leaf node. Decision node selects a feature and check which instance of that feature identify the class label completely. If a instance of a certain feature identifies node that has only one type of class label then it reaches the leaf node. If the node has different kinds of labels, then we extend the node further. Choosing the expansion of the right node is important to keep the tree small. We choose the best node at the moment to expand by calculating the information gain. The node with the highest information gain at a given moment gets extended.

**Random Forest Classifier:** Decision Tree Classifier has overfitting problem. It gives good accuracy for data it is trained with but cannot as accurately predict unknown data. This problem is solved with an algorithm called Random Forest Classifier. Random Forest uses bagging approach to solve overfitting problem. Unlike creating a tree from the whole dataset, we create a number of trees, each with a different subset of features. We take samples from our datasets with some duplicates. Each tree would give a class label which we count as a vote. Finally, the label which gets most votes are the predicted label from the model.

**Xgboost Classifier:** Xgboost classifier is a gradient boosting algorithm.

If the dataset has n classification label, it finds out probability by using 1/n. The algorithm creates a separate column resuduel by substracting probability from class label. It would then calculate similarity weight for a feature by following equation,

Weight = \frac{\sum{}(Residuel)\power{2}|\sum{}(Prob(1-Prob))}

It would get **Gain** by adding all leaf nodes value and subtracting parent node value from the summation of leaf nodes. It would give highest priority to the feature which has highest Gain.

**Performance Analysis:**

**Precision:** Precision measures proportion of the true positive predictions in terms of all positive predictions made by model.

**Recall:** Recall measures proportion of the true positive predictions in terms of all actual positive in the datasets.

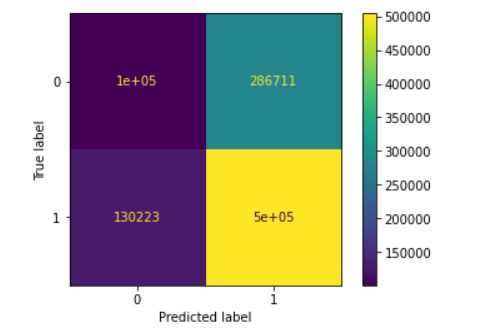
**F1-Score:** F1-score measures how good a model works. F1-score is calculated by using the following formula:

F = (2 \* precision \* recall)/ (precision + recall)

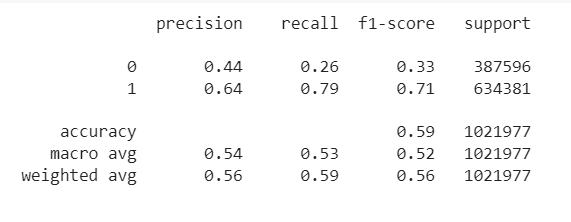
The performance analysis made by our models using different classification algorithms are given below.

**Decision Tree Classifier:**

**Confusion Matrix:**

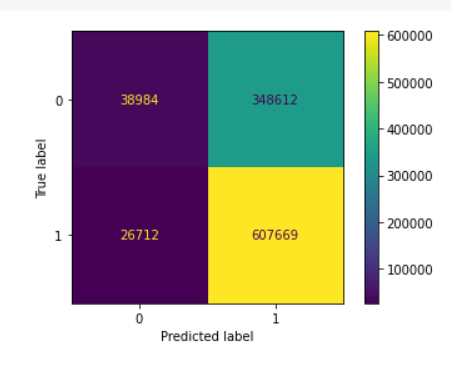


**Overall Performance Report:**

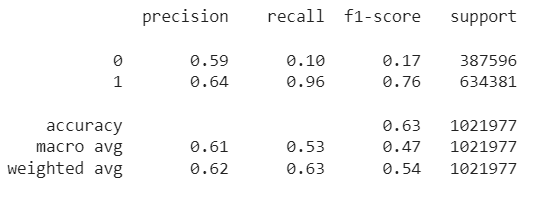


**Random Forest Classifier:**

**Confusion Matrix:**

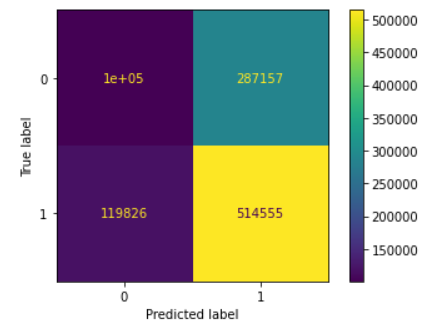


**Overall Performance Report:**

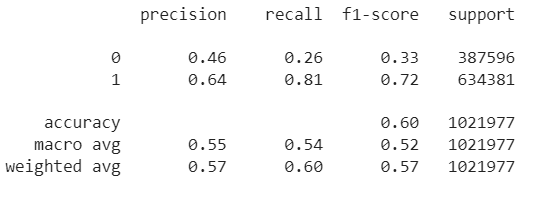


**Xgboost Classifier:**

**Confusion Matrix:**



**Overall Performance Report:**



**Challenges, learning and summary of the project:**

**Challenges:** Since we are new to Machine Learning field, we had to study Machine Learning concisely before building models around those Machine Learning algorithm. We are also comfortable with python programming language. So, we had to take helps from different online blogs to write codes in Python. We are not familiar different ML library and Matplotlib Library. We had to suffer a bit to write codes due to our unfamiliarity with these library.

**Learning:** We got to know about different ML algorithms and how they work while doing this project. We had to do fair share of debugging while doing this project. So, we can say it sharpened our debugging skill a bit. We learned about different graph plotting libraires like Matplotlib and Seaborn. Overall, it was a good learning experience for us.