Describe the loss function used in training all the classifier that you have selected. De scribe hyper-parameters used in different classifiers and Impact of performing hyper parameter tuning.

- Random forest: It used gini impurity to measure the quality of split
    - Estimators = [2,8,32,128,256,512,1024]
    - max_depth = [1,2,4,8,16]
    - class_weight = [None, 'balanced']
    - min_samples_leaf = [2,4,8,32,128,256]
- Logistic Regression: Logistic Loss i.e Cross entropy loss is used.
    - C = [10**-2, 10**-1, 10**0, 10**1, 10**2]
    - penalty = ['l1', 'l2']
    - class_weight = [None, 'balanced']
- KNN : It used eucliedean distance based voting mechanism
    - n_neighbors = [3,5,9,15,33,65,127,257,513]
    - weights = ['uniform', 'distance']
    - p = [1,2]
- Gradient Boosting Classifier: GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n_classes_ regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. Binary classification is a special case where only a single regression tree is induced.
    - n_estimators = [128,256]
    - max_depth = [1,2,4,8,16]
    - class_weight = [None, 'balanced']
    - min_samples_leaf = [2,4,8,32,128,256]

Mention all the Error metric used to compare the performance of different Classifiers and why we have to use them for this dataset.

- I have used Balanced accuracy metric, f1 score and roc auc score as metric for gridsearch cv.
- Both balanced accuracy and f1 score are suitable for imbalance dataset and very well applicable for corona dataset which is provided.
- Roc_auc metric is calculated just to identify the discriminative ability of the model. A value > 0.5 is necessary for model to be better than simple guess.

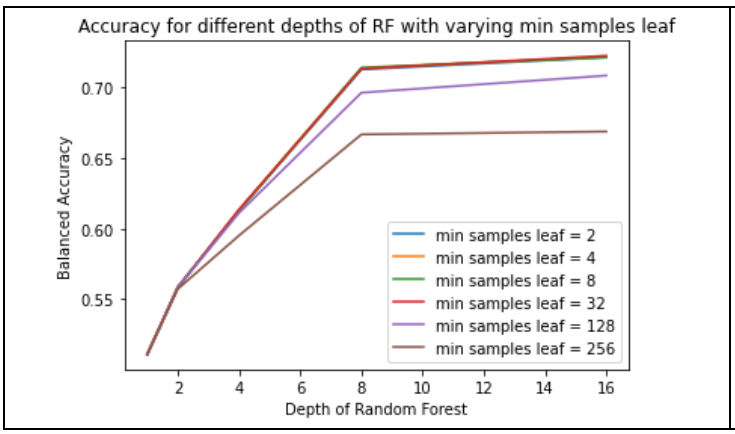
which hyper-parameters seems to influence the performance of each ML model based on your observation. Provide a plot to substantiate your answer. Provide a table for each ML method on the list of hyperparameters that were optimised using gridSearch and the range of values

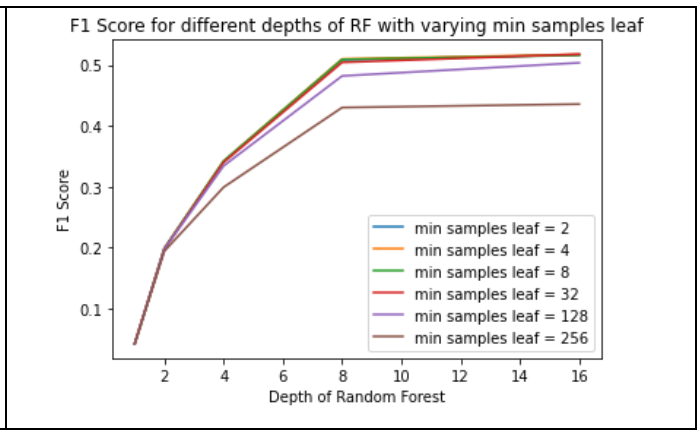
Hyperparameter tuning

Random Forest:

- As Depth increases from 2 to 10, we see linear trend of increased in balanced accuracy. Balanced accuracy takes into account class imbalance and provide weighted accuracy.
- Depth beyond 10, shows steady response in terms of accuracy
- Min samples leaf helps in controlling the overfitting by avoiding making too many splits for small samples support.

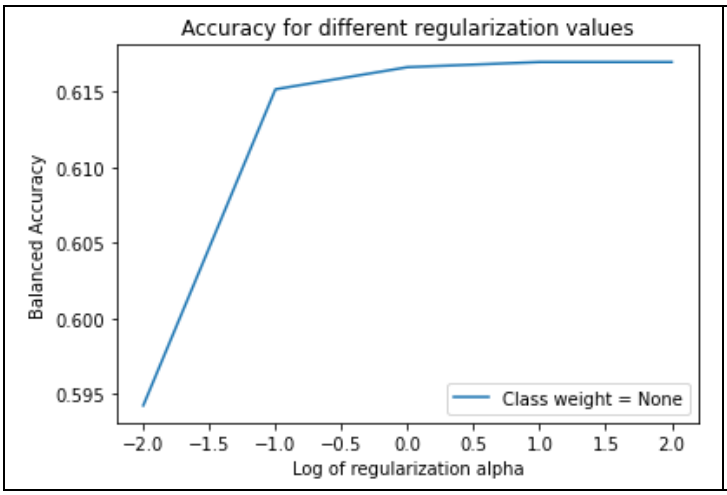
- From graph it is clear that, min samples leaf till 32, do not affect the accuracy. Beyond min samples leaf = 32, this setting provides strong regularization.

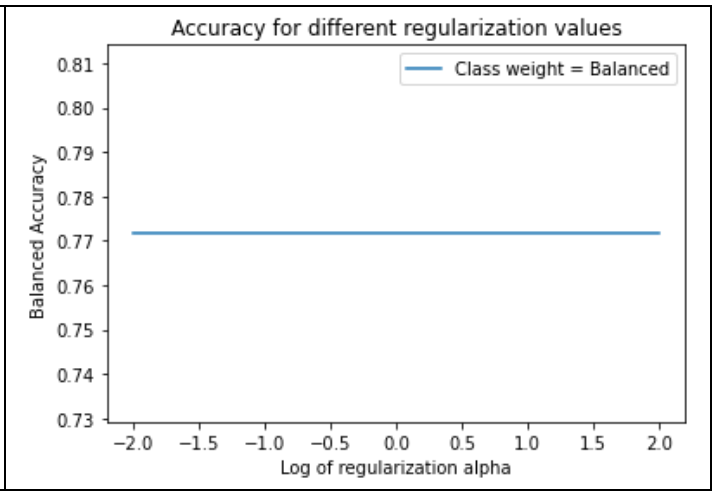




Logistic Regression:

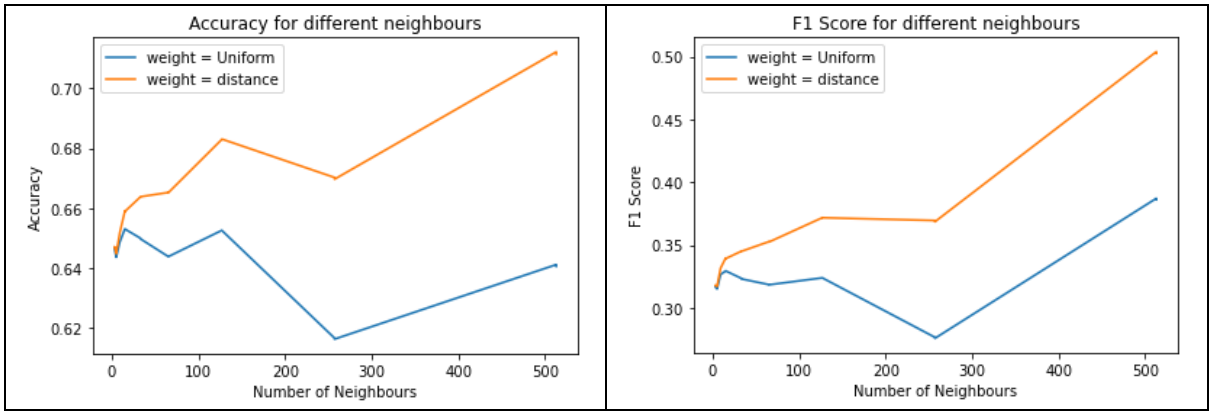
- If class weight is set to None, with increase in regularization, accuracy increases. However, if class weight is balanced, regularization doesn't affect the accuracy. As a comparison, with balanced weight, accuracy is highest as compared to without setting class weights





KNN Classifier:

- On comparison, distance based weighting gives better results compared to uniform based weighting. With 513 neighbours, we get the best results.



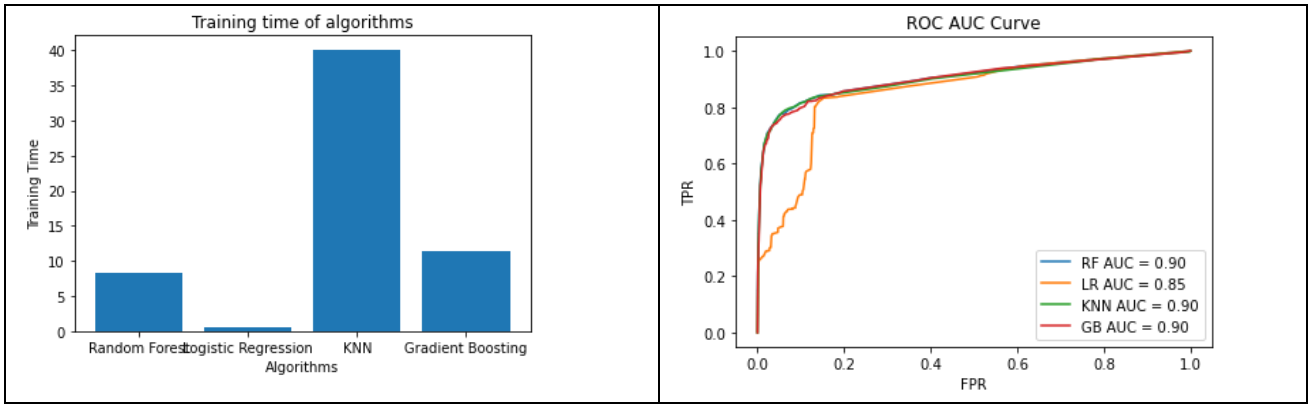
Gradient Boosting Classifier:

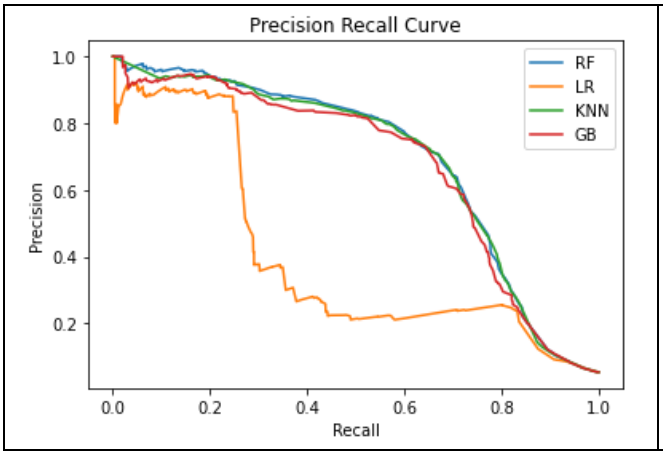
- On comparison, result was similar to random forest classifier. Accuracy increased with increase in depth and then remained constant.

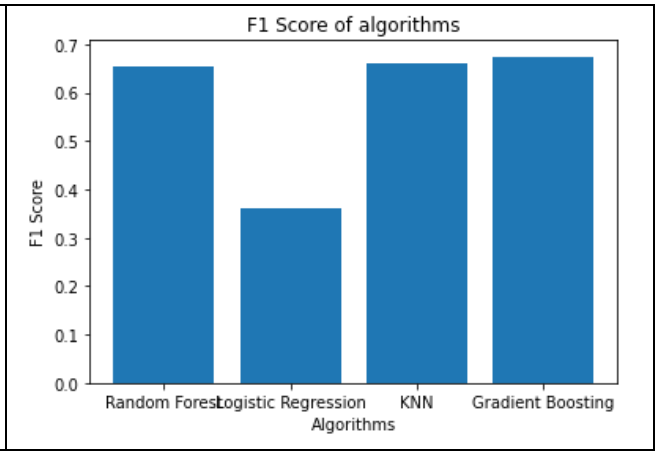
Pick the optimized model for each ML method and perform a visual comparison between them in terms of

- Best model for Random forest has following parameters:
  - Estimators = 128, depth = 8, class weight = None, min samples leaf = 2
- Best model for logistic regression has following parameters:
  - Class weight = Balanced, C = 1, penalty = l1
- Best model for KNN classifier has following parameters:
  - Weight = 'distance', n_neighbours = 513, p =1
- Best model for Gradient Boosting:
  - Estimators = 128, depth = 2, class weight = None, min samples leaf = 2

As shown in below charts, Logistic regression has least training time but also has less f1 score and relatively poor performance in roc_auc curve or precision recall curve. All other methods are equivalent in terms of curve and f1 score except training time. If we have to select the optimal algorithm then it would be random forest







Compare the best performing ML model from above with the error metrics obtained from your ANN model.

- An MLP model with 2 hidden layer is used have 10 units of neuron each. This model performs equivalently to random forest model.
- Only on comparison on time taken for training, here again random forest is the winner.