Question 4

[CM4]Random Forest Classifier

1. Hyper parameter tuning is performed using 10-fold cross validation on each label to evaluate the best value for number of trees and Max Depth

Original Features :

```
[ ] DTbase = RandomForestClassifier(max_features = 'auto', random_state = 0)
    param_grid = {
        'n_estimators' : [5, 10, 50, 150, 200],
        'max_depth': [3, 5, 10, None],
    }

    DT_fit = GridSearchCV(estimator=DTbase, param_grid=param_grid, cv = 10, refit='accuracy_score')
    DT_result = DT_fit.fit(Original_data_copy.iloc[:, 3:14], y)

    results_df = pd.DataFrame(DT_result.cv_results_)
    results_df
```

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_max_depth	param_n_estimators	params	mean_test_score	std_test_score	rank_test_score
0	0.017757	0.005407	0.003402	0.000704	3	5	{'max_depth': 3, 'n_estimators': 5}	0.772555	0.034819	20
1	0.024917	0.002625	0.003550	0.000058	3	10	{'max_depth': 3, 'n_estimators': 10}	0.799828	0.032848	19
2	0.110068	0.005244	0.008559	0.000468	3	50	{'max_depth': 3, 'n_estimators': 50}	0.802287	0.028968	18
3	0.326658	0.012556	0.022202	0.002116	3	150	{'max_depth': 3, 'n_estimators': 150}	0.803946	0.028103	16
4	0.426768	0.014316	0.028316	0.003514	3	200	{'max_depth': 3, 'n_estimators': 200}	0.803946	0.028103	16
5	0.014943	0.001144	0.002913	0.000061	5	5	{'max_depth': 5, 'n_estimators': 5}	0.814669	0.031606	15
6	0.027115	0.001237	0.003666	0.000118	5	10	{'max_depth': 5, 'n_estimators': 10}	0.815510	0.032094	14
7	0.122380	0.005092	0.009056	0.001137	5	50	{'max_depth': 5, 'n_estimators': 50}	0.832893	0.042717	5
8	0.367693	0.005441	0.022184	0.001716	5	150	{'max_depth': 5, 'n_estimators': 150}	0.841157	0.036919	1
9	0.481975	0.011075	0.028851	0.002923	5	200	{'max_depth': 5, 'n_estimators': 200}	0.832879	0.040836	6

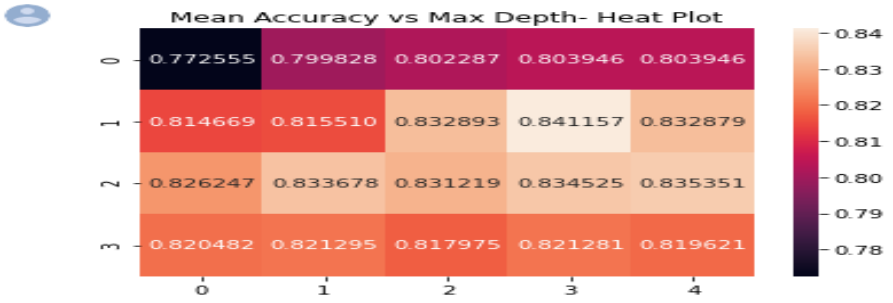
10	0.018383	0.001702	0.003118	0.000062	10	5	{'max_depth': 10, 'n_estimators': 5}	0.826247	0.036205	8
11	0.032452	0.001145	0.003776	0.000147	10	10	{'max_depth': 10, 'n_estimators': 10}	0.833678	0.035825	4
12	0.153483	0.003747	0.009348	0.000380	10	50	{'max_depth': 10, 'n_estimators': 50}	0.831219	0.032581	7
13	0.445450	0.008905	0.024775	0.005772	10	150	{'max_depth': 10, 'n_estimators': 150}	0.834525	0.043197	3
14	0.577283	0.011328	0.029102	0.001252	10	200	{'max_depth': 10, 'n_estimators': 200}	0.835351	0.042152	2
15	0.017089	0.000479	0.002906	0.000062	None	5	{'max_depth': None, 'n_estimators': 5}	0.820482	0.018481	11
16	0.033139	0.002035	0.003952	0.000351	None	10	{'max_depth': None, 'n_estimators': 10}	0.821295	0.025797	9
17	0.150238	0.004513	0.009301	0.000445	None	50	{'max_depth': None, 'n_estimators': 50}	0.817975	0.033474	13
18	0.455255	0.014941	0.023570	0.001712	None	150	{'max_depth': None, 'n_estimators': 150}	0.821281	0.035706	10
19	0.601419	0.009090	0.031274	0.003508	None	200	{'max_depth': None, 'n_estimators': 200}	0.819621	0.037182	12

```
import numpy as np
data = np.array(results_df["mean_test_score"])
data
```

```
array([0.7725551 , 0.79982782, 0.8022865 , 0.80394628, 0.80394628,
       0.81466942, 0.81550964, 0.83289256, 0.84115702, 0.83287879,
       0.82624656, 0.83367769, 0.83121901, 0.83452479, 0.83535124,
       0.82048209, 0.82129477, 0.81797521, 0.82128099, 0.81962121])
```

Heat plot with (5 * 4) mean accuracies for different values of number of trees and maximum depth:

```
data = data.reshape(4,5)
import seaborn as sns
plt.title('Mean Accuracy vs Max Depth- Heat Plot')
ax = sns.heatmap(data, annot=True, fmt='f')
```



We can conclude from the heat map that max depth of 5 with 150 trees provides the best accuracy of 84.11%

PCA features:

```
# RF with PCA
DTbase = RandomForestClassifier(max_features = 'auto', random_state = 0)
param_grid = {
    'n_estimators' : [5, 10, 50, 150, 200],
    'max_depth': [3, 5, 10, None],
}

DT_fit = GridSearchCV(estimator=DTbase, param_grid=param_grid, cv = 10, refit='accuracy_score')
DT_result = DT_fit.fit(pca_features, y)

results_df = pd.DataFrame(DT_result.cv_results_)
results_df
```

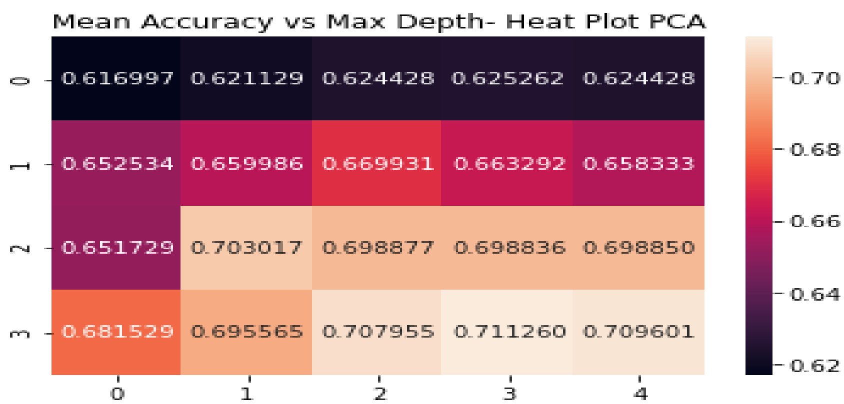
	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_max_depth	param_n_estimators	params	mean_test_score	std_test_score	rank_test_score
0	0.012742	0.001647	0.002435	0.000412	3	5	{'max_depth': 3, 'n_estimators': 5}	0.616997	0.042880	20
1	0.020724	0.000426	0.002933	0.000255	3	10	{'max_depth': 3, 'n_estimators': 10}	0.621129	0.043976	19
2	0.097238	0.005845	0.008004	0.000570	3	50	{'max_depth': 3, 'n_estimators': 50}	0.624428	0.045664	17
3	0.284575	0.006652	0.021331	0.002061	3	150	{'max_depth': 3, 'n_estimators': 150}	0.625262	0.042470	16
4	0.369913	0.007628	0.026685	0.002215	3	200	{'max_depth': 3, 'n_estimators': 200}	0.624428	0.043676	17
5	0.012697	0.000923	0.002388	0.000413	5	5	{'max_depth': 5, 'n_estimators': 5}	0.652534	0.056588	14
6	0.023028	0.001857	0.003090	0.000427	5	10	{'max_depth': 5, 'n_estimators': 10}	0.659986	0.057524	12
7	0.102191	0.001932	0.008162	0.001040	5	50	{'max_depth': 5, 'n_estimators': 50}	0.669931	0.049012	10
8	0.304221	0.003543	0.020213	0.000635	5	150	{'max_depth': 5, 'n_estimators': 150}	0.663292	0.051744	11
9	0.405902	0.006650	0.029259	0.004108	5	200	{'max_depth': 5, 'n_estimators': 200}	0.658333	0.047006	13
10	0.015263	0.002009	0.002439	0.000191	10	5	{'max_depth': 10, 'n_estimators': 5}	0.651729	0.054863	15
11	0.028414	0.001525	0.003149	0.000124	10	10	{'max_depth': 10, 'n_estimators': 10}	0.703017	0.059784	4
12	0.122018	0.002599	0.008878	0.000835	10	50	{'max_depth': 10, 'n_estimators': 50}	0.698877	0.044754	5
13	0.354898	0.004700	0.022204	0.002272	10	150	{'max_depth': 10, 'n_estimators': 150}	0.698836	0.052454	7
14	0.476229	0.007589	0.031783	0.003256	10	200	{'max_depth': 10, 'n_estimators': 200}	0.698850	0.050192	6

19	0.507097	0.007671	0.026216	0.000636	None	200	{'max_depth': None, 'n_estimators': 200}	0.681529	0.061904	9
18	0.395652	0.010129	0.022607	0.000550	None	150	{'max_depth': None, 'n_estimators': 150}	0.695565	0.066476	8
17	0.135405	0.007303	0.009035	0.000270	None	50	{'max_depth': None, 'n_estimators': 50}	0.707955	0.049690	3
16	0.028417	0.000871	0.003956	0.002662	None	10	{'max_depth': None, 'n_estimators': 10}	0.711260	0.054232	1
15	0.015524	0.000956	0.002481	0.000273	None	5	{'max_depth': None, 'n_estimators': 5}	0.709601	0.054029	2

```python
import numpy as np
data = np.array(results_df["mean_test_score"])
data
```

```
array([0.61699725, 0.62112948, 0.62442837, 0.62526171, 0.62442837,
       0.65253444, 0.65998623, 0.66993113, 0.66329201, 0.65833333,
       0.65172865, 0.70301653, 0.69887741, 0.69883609, 0.69884986,
       0.68152893, 0.69556474, 0.70795455, 0.71126033, 0.70960055])
```

```python
data = data.reshape(4,5)
import seaborn as sns
plt.title('Mean Accuracy vs Max Depth- Heat Plot PCA')
ax = sns.heatmap(data, annot=True, fmt='f')
```



We can conclude from the heat map that max depth of None with 150 trees provides the best accuracy of 71.12%

Original Features:

Label	Max_Depth	No. of trees	Accuracy
Confirmed	5	10	96.27%
Recovered	10	10	93.33%
Deaths	3	150	92.39%

PCA Features:

Max_Depth	No.of tress	Accuracy
None	**150**	**71.12%**

Decision Tree is prone to overfitting and ensemble method like Random Forest helps to tackle this issue.

It can be observed that Random Forest has the best accuracy on all the 3 labels compared to all the other tree based classifiers and Naive Bayes on the covid dataset