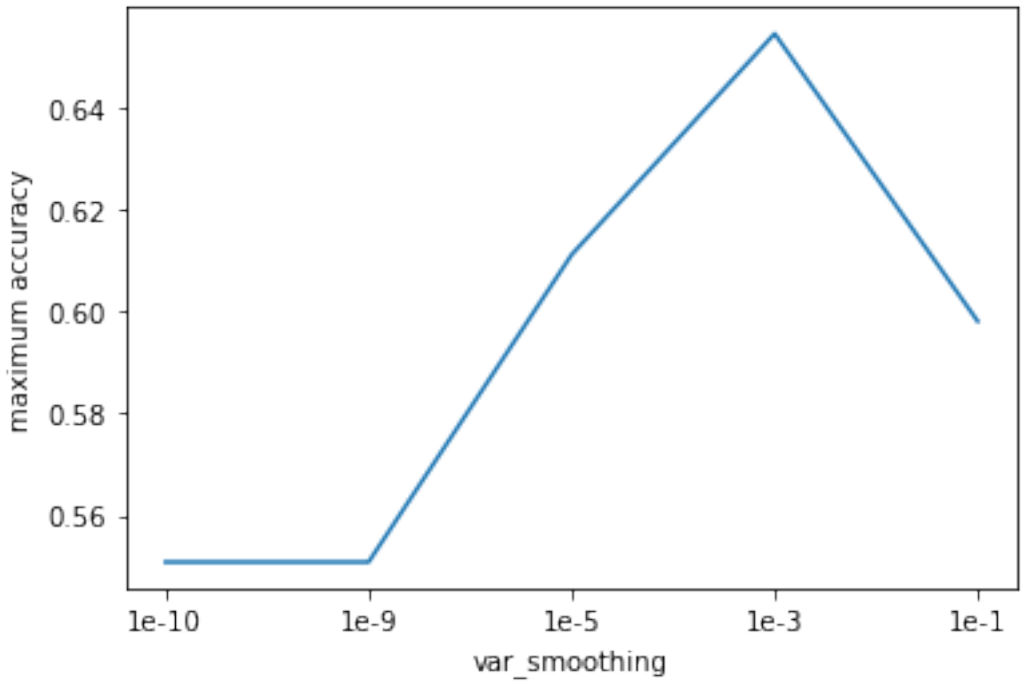


We observe same maximum accuracy for variance smoothing parameters 0.001 (1e-3). Smoothing allows Naive Bayes to better handle cases where evidence has never appeared for a particular category i.e. the problem of zero probability. We observe with increasing smoothing parameter, the accuracy of the model remains constant, peaks at 0.001 and again decreases.

[64]:
```
%%time
# applying the best value of var_smoothing on test set
gnb = GaussianNB(var_smoothing=best_var)
gnb.fit(X_train_val, y_train_val)
gnb.predict(X_test)
accuracy = metrics.accuracy_score(y_test, y_pred)
print('Accuracy: ', accuracy)
f_score = f1_score(y_test, y_pred, average = 'macro')
print('f-score:', f_score)
```

```
Accuracy:  0.6722802290333446
f-score: 0.6624100366524396
Wall time: 22.9 ms
```

NB learned Parameters theta_ (mean) and sigma_ (variance)

[65]:
```
# mean
gnb.theta_
```

[65]:
```
array([[ 3.84338492e+01,  4.37274733e+01, -7.95909554e+01,
         4.97254119e-01,  6.24063904e-03,  1.59760359e-01,
         2.05941088e-01,  1.75736395e-01,  1.72241638e-02,
         9.98502247e-04,  2.09685472e-01],
       [ 4.02877131e+01,  4.37725211e+01, -7.96265344e+01,
         4.90714831e-01,  9.41236327e-03,  3.56652251e-01,
```

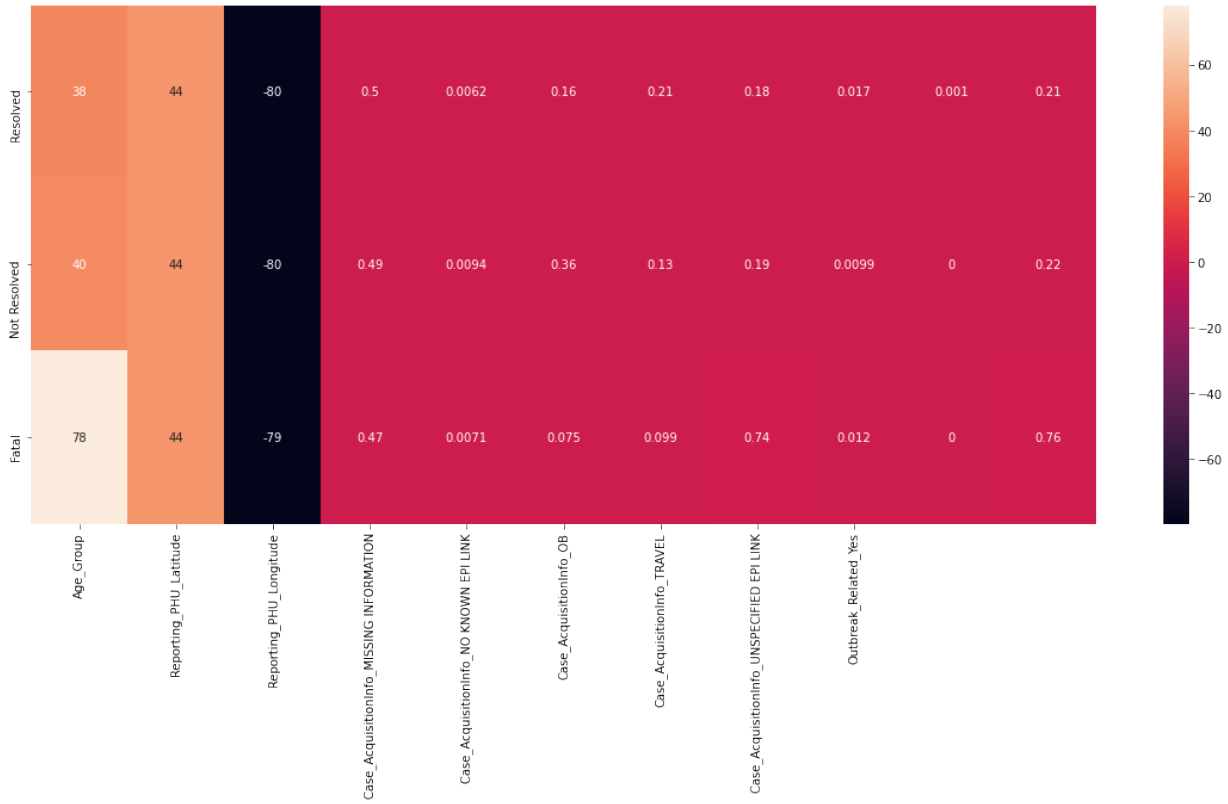
```
          1.32790639e-01,  1.85703383e-01,  9.92113966e-03,
          0.00000000e+00,  2.16229967e-01],
        [ 7.82528561e+01,  4.37284837e+01, -7.94626338e+01,
          4.74739782e-01,  7.10840315e-03,  7.53998477e-02,
          9.92637725e-02,  7.36481340e-01,  1.24397055e-02,
          0.00000000e+00,  7.62122366e-01]])
```

[66]:
```
# variance
gnb.sigma_
```

[66]:
```
array([[359.88055087,   1.13713585,   3.01213355,   0.87992674,
          0.63613597,   0.76417127,   0.79346364,   0.77478739,
          0.64686177,   0.63093178,   0.79565175],
       [390.80383317,   1.2637536 ,   3.10508876,   0.87984807,
          0.63925805,   0.8593857 ,   0.74509156,   0.78115192,
          0.63975699,   0.62993428,   0.79940885],
       [132.03845303,   1.17192068,   3.20262092,   0.8792962 ,
          0.63699215,   0.69964899,   0.71934476,   0.82401086,
          0.64221924,   0.62993428,   0.81122614]])
```

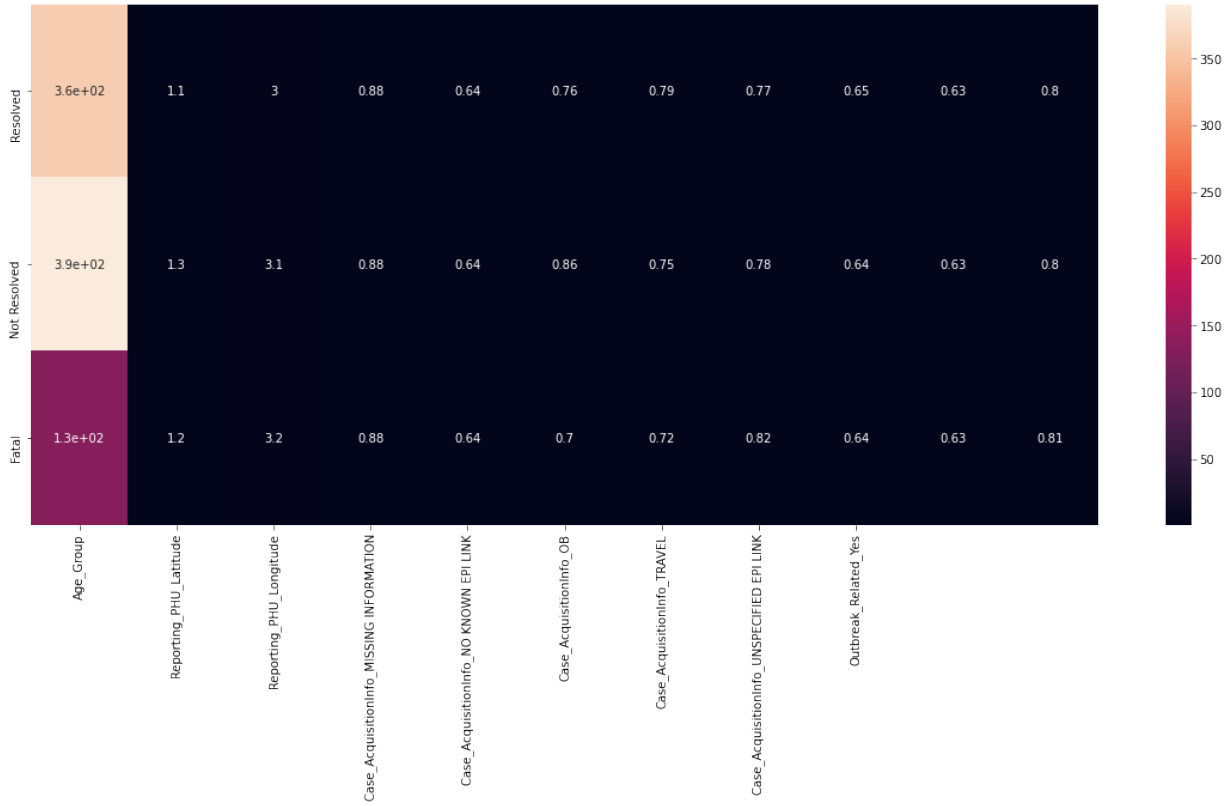
[67]:
```
x_axis_labels = ['Age_Group', 'Reporting_PHU_Latitude', 'Reporting_PHU_Longitude',
       'Case_AcquisitionInfo_MISSING INFORMATION',
       'Case_AcquisitionInfo_NO KNOWN EPI LINK', 'Case_AcquisitionInfo_OB',
       'Case_AcquisitionInfo_TRAVEL',
       'Case_AcquisitionInfo_UNSPECIFIED EPI LINK', 'Outbreak_Related_Yes'] # labels␣
 ↪for x-axis
y_axis_labels = ['Resolved','Not Resolved','Fatal'] # labels for y-axis

# create seabvorn heatmap with required labels
fig, ax = plt.subplots(figsize=(20,8))
sns.heatmap(gnb.theta_, xticklabels=x_axis_labels,␣
 ↪yticklabels=y_axis_labels,annot=True)
```

[67]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x27e04c35730>
```



[68]:
```
# create seabvorn heatmap with required labels
fig, ax = plt.subplots(figsize=(20,8))
sns.heatmap(gnb.sigma_, xticklabels=x_axis_labels,␣
↪yticklabels=y_axis_labels,annot=True)
```

[68]: <matplotlib.axes._subplots.AxesSubplot at 0x27e04e44d90>



[CM6]

Covid Dataset :

In Decision tree, we observe maximum accuracy of 65.7% at maximum depth of 5 on the training set with 10 fold cross validation. The model gives an accuracy of 65.9% on the test set. We see that the accuracy increases from depth 3, maximizing at 5 and further decreases at depth 10.

In Random Forest, we observe maximum accuracy of 66.5% at the maximum depth of 5 and 150 independent tress on the training set with 10 fold cross validation. The model gives an accuracy of 66.4% on the test set.

In Gradient Tree Boosting, we observe maximum accuracy of 66.7% with 200 tress on the training set with 10 fold cross validation. The model also gives an accuracy of 67.2% on the test set.

In Naive Bayes Classifier, we observe maximum accuracy of 65.46% with a variance smoothing parameter 0.001 (1e-3). The model gives an accuracy of 67.2% on the test set. We notice that the accuracy for variance smoothing parameters 1e-10, 1e-9, 1e-5, is same with aspike in 1e-3 and then decreases for 1e-1. Laplace smoothing is a smoothing technique that helps tackle the problem of zero probability in the Naïve Bayes machine learning algorithm.

Comparing the classifiers, we notice that Naive Baiyes has better performance (comparable to Gradient Tree Boosting) on the test set than the Decision tree approaches. Decision tree is a discriminative model, whereas Naive bayes is a generative model. Also, Naive bayes is computationally faster than tree based classifiers.

From the NB learned parameters of 'theta_' (mean) and 'sigma_' (variance) , We observe:

- A feature can be considered good seperator, if the mean of the feature for disctinct classes are far apart, and if the variance of the features are low indicating the values are closer to te mean.

- the learned parameter 'theta_' (mean) of feature 'Age_Group' is similar for Resolved & Not Resolved and different from Class Fatal. Also 'sigma_' (variance) is relatively less for this feature against class Fatal, indicating all the values are spaced close to the mean. Thus this feature 'Age_Group' can be used to distinguish Fatal cases effectively. This is observed in the decision tree classifier aswell, where the splitting rule Age_Group <= 65.00 successfully seperates Fatal cases from rest of the dataset.
- Similarly the feature Outbreak_Related_Yes has a high mean for the class Fatal compared to rest of the classes, indicating most of the Fatal cases are related to Outbreak. This is observed in the decision tree classifier aswell, where the splitting rule Outbreak_Related_Yes <= 0.50 seperates Fatal cases to a good extent.

[ ]: