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

```
[67]: X_train_val.columns
```

```
[67]: Index(['area', 'perimeter', 'length_kernel', 'width_kernel',
             'length_of_kernel_groove'],
            dtype='object')
```

```
[68]: # mean ['area','perimeter','length_kernel','width_kernel','length_of_kernel_groove']
      gnb.theta_
```

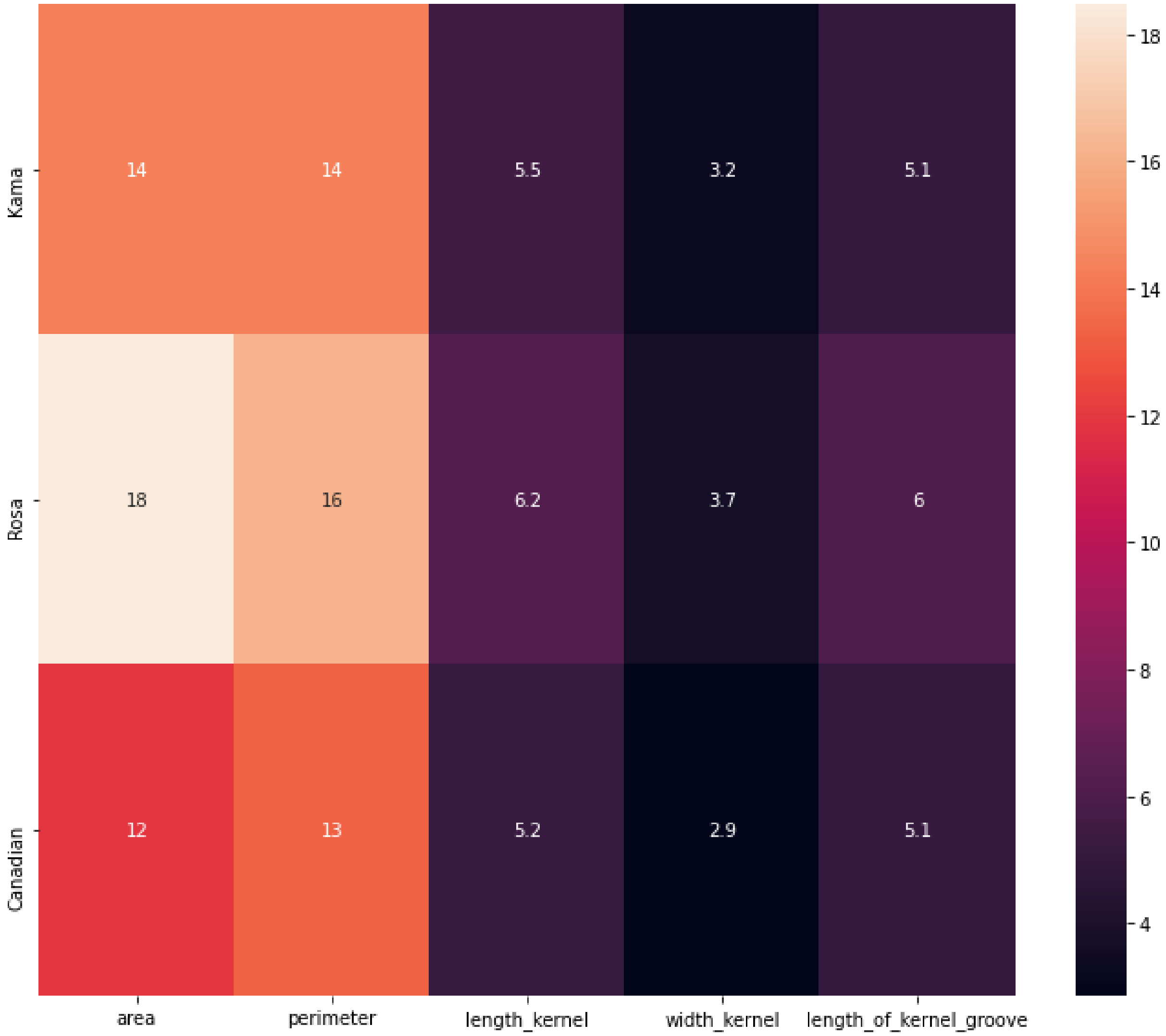
```
[68]: array([[14.31229984, 14.30681863,  5.51372596,  3.24076923,  5.07336077],
             [18.47509091, 16.20672727,  6.18145455,  3.68961818,  6.04552727],
             [11.92980392, 13.26666667,  5.23572549,  2.86598039,  5.11607843]])
```

```
[69]: # variance␣
       ↪['area','perimeter','length_kernel','width_kernel','length_of_kernel_groove']
      gnb.sigma_
```

```
[69]: array([[1.1553264 , 0.2751547 , 0.0466526 , 0.03198691, 0.05833384],
             [1.91794499, 0.33493838, 0.05771701, 0.03350129, 0.04885654],
             [0.47804898, 0.11106144, 0.01841267, 0.01965065, 0.02661811]])
```

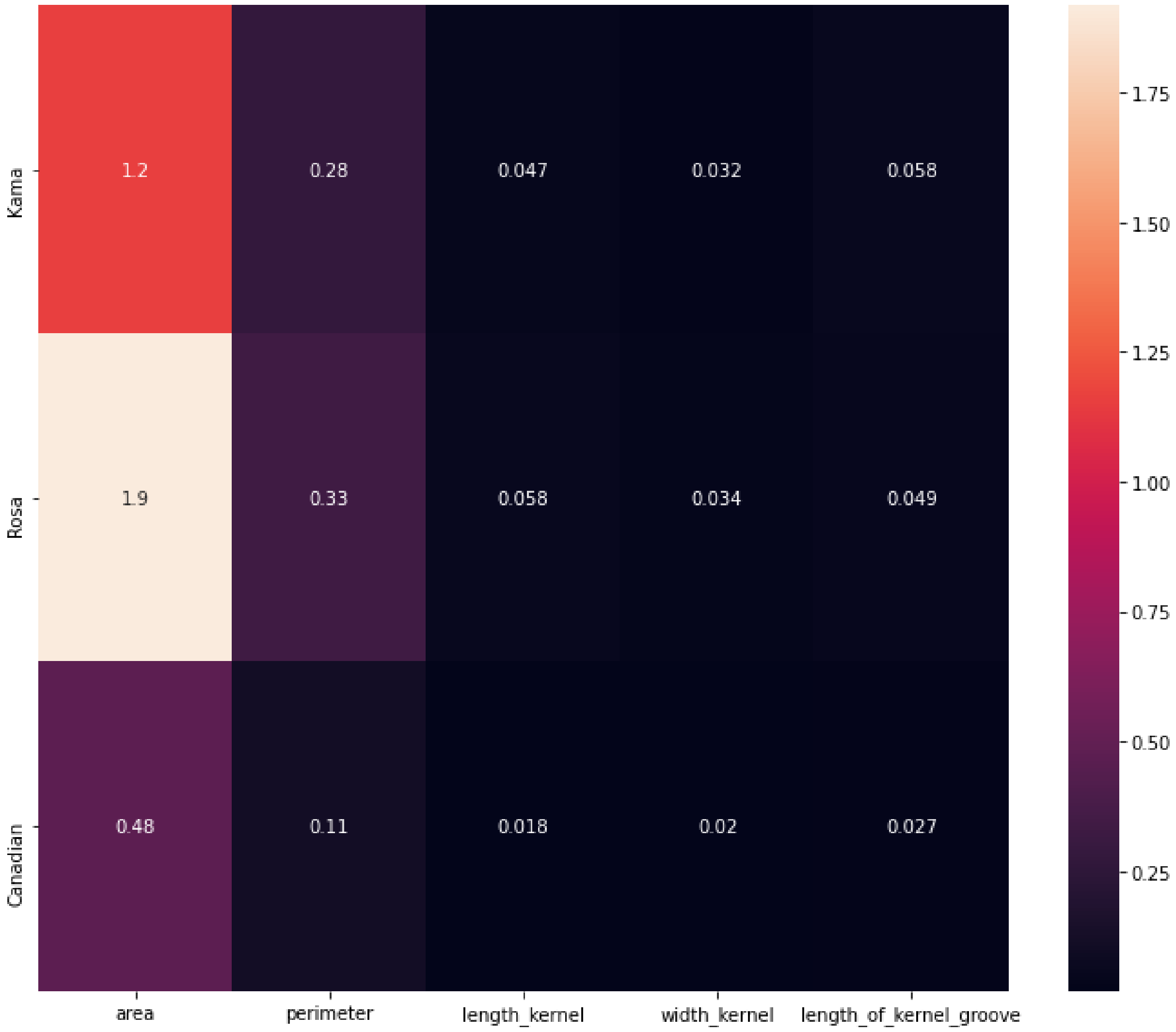
```
[70]: x_axis_labels =␣
       ↪['area','perimeter','length_kernel','width_kernel','length_of_kernel_groove'] #␣
       ↪labels for x-axis
      y_axis_labels = ['Kama','Rosa','Canadian'] # labels for y-axis

      # create seabvorn heatmap with required labels
      fig, ax = plt.subplots(figsize=(12,10))
      sns.heatmap(gnb.theta_, xticklabels=x_axis_labels,␣
       ↪yticklabels=y_axis_labels,annot=True)
```

```
[70]: <matplotlib.axes._subplots.AxesSubplot at 0x20c7435f7f0>
```



```
[71]: # create seabvorn heatmap with required labels
      fig, ax = plt.subplots(figsize=(12,10))
      sns.heatmap(gnb.sigma_, xticklabels=x_axis_labels,␣
       ↪yticklabels=y_axis_labels,annot=True)
```

```
[71]: <matplotlib.axes._subplots.AxesSubplot at 0x20c78e26af0>
```



[CM6]

Seeds Dataset :

In Decision tree, we observe maximum accuracy of 91% at maximum depth of 5 on the training set with 10 fold cross validation. The model gives an accuracy of 92.65% on the test set.

In Random Forest, we observe maximum accuracy of 92.95% at the maximum depth of 5 and 50 independent tress on the training set with 10 fold cross validation. The model also gives an accuracy of 92.65% on the test set.

In Gradient Tree Boosting, we observe maximum accuracy of 94.25% with 150 tress on the training set with 10 fold cross validation. The model also gives an accuracy of 92.65% on the test set.

In Naive Bayes Classifier, we observe maximum accuracy of 90.4% with a variance smoothing parameter 1e-10. The model gives an accuracy of 92.5% on the test set. We notice that the accuracy for variance smoothing parameters 1e-10, 1e-9, 1e-5, 1e-3 is same 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. We see that the smoothing parameter doesn't have a major impact here as all the features are numeric in the dataset.

Comparing the classifiers, we notice that Decision tree approaches has better performance on the training set than Naive Bayes, But we observe similar performance on the test set. This can be attributed to small size of the dataset. Decision tree is a discriminative model, whereas Naive bayes is a generative model. Also, Naive bayes is computationally faster when compared to Decision tree approaches.

From the NB learned parameters of 'theta_' (mean) and 'sigma_' (variance) , and comparing this to the single Decision Tree model, 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 'length_of_kernel_groove' is similar for Class 1(Kama) & 3(Canadian) (5.1) and different from Class 2 (Rosa) (6). Also 'sigma_' (variance) is very low for this feature indicating all the values are spaced close to the mean. Thus this feature 'length_of_kernel_groove' can be used to distinguish Class 2 (Rosa) effectively. This is observed in the decision tree classifier aswell, where the splitting rule 'length_of_kernel_groove' <= 5.57 successfully seperates Class 2 (Rosa) from rest of the dataset.
- Similarly the parameter 'theta_' (mean) of feature 'perimeter' is relatively different for Classes 1(Kama) & 3(Canadian). Also 'sigma_' (variance) is low for this feature indicating all the values are spaced close to the mean. Thus this feature can distinguish between Classes 1(Kama) & 3(Canadian) considerably. This is observed in the decision tree classifier aswell, where the splitting rule 'perimeter' <= 13.80 seperates Classes 1(Kama) & 3(Canadian) to a good extent.
- Similarly the learned parameters of feature 'area' and 'width_kernel' are relatively different for Classes 1(Kama) & 3(Canadian). Thus these feature can distinguish between Classes 1(Kama) & 3(Canadian) decently. This is observed in the decision tree classifier aswell, where the features are used to seperate Classes 1(Kama) & 3(Canadian).

[ ]: