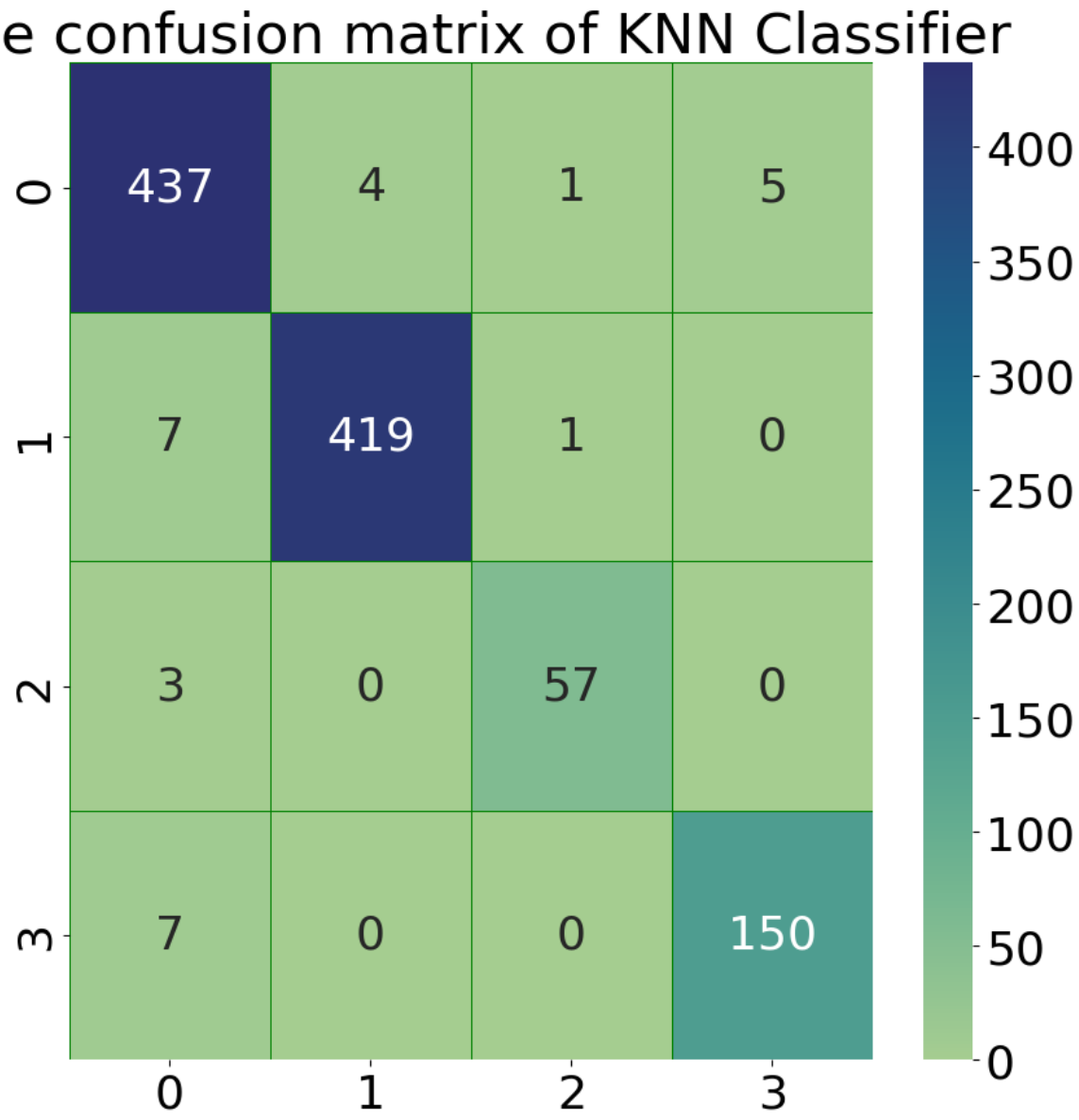
This is a notebook for comparing various models. For each model, we output their correctness in the test set and draw its confusion matrix. The comparison shows the gap between the correctness and stability of different models.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from src.Classifiers import robots_movement_classifier
```

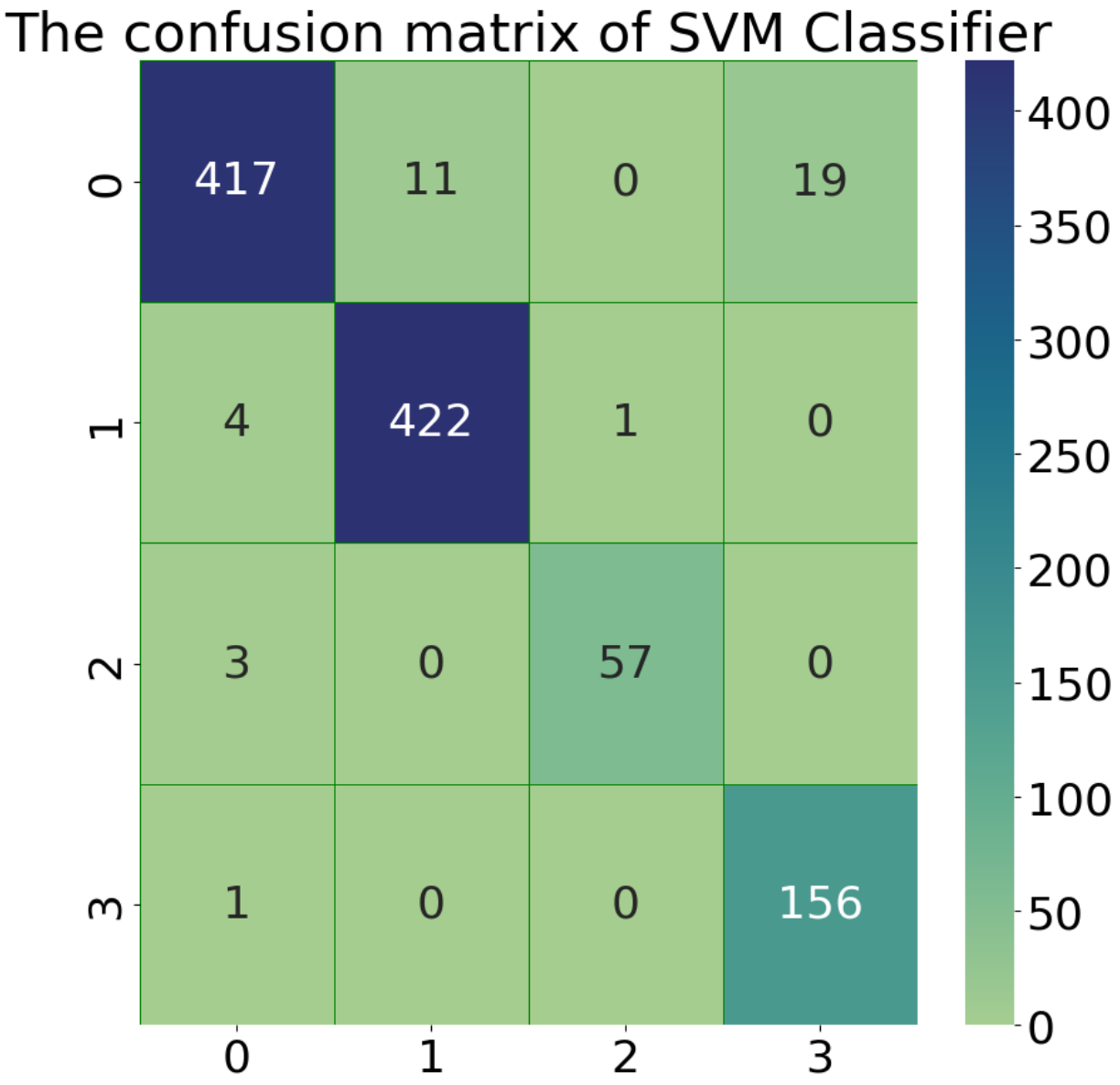
In [3]:
```
Accuracy = []
datasets = pd.read_csv('DATA/sensor_readings_4.csv')
rmc = robots_movement_classifier(datasets)
knn_cm , knn_ac, knnclassifier = rmc.knnclassifier_learning()
Accuracy.append(knn_ac)
print('The accuracy of knn: ', knn_ac)

plt.rcParams.update({'font.size': 26})
f, ax = plt.subplots(figsize =(10,10))
sns.heatmap(knn_cm,cmap = "crest", annot = True,linewidths=0.5 ,linecolor =
ax.set_title("The confusion matrix of KNN Classifier")
plt.show()
```

```
The accuracy of knn:  0.9743354720439963
```



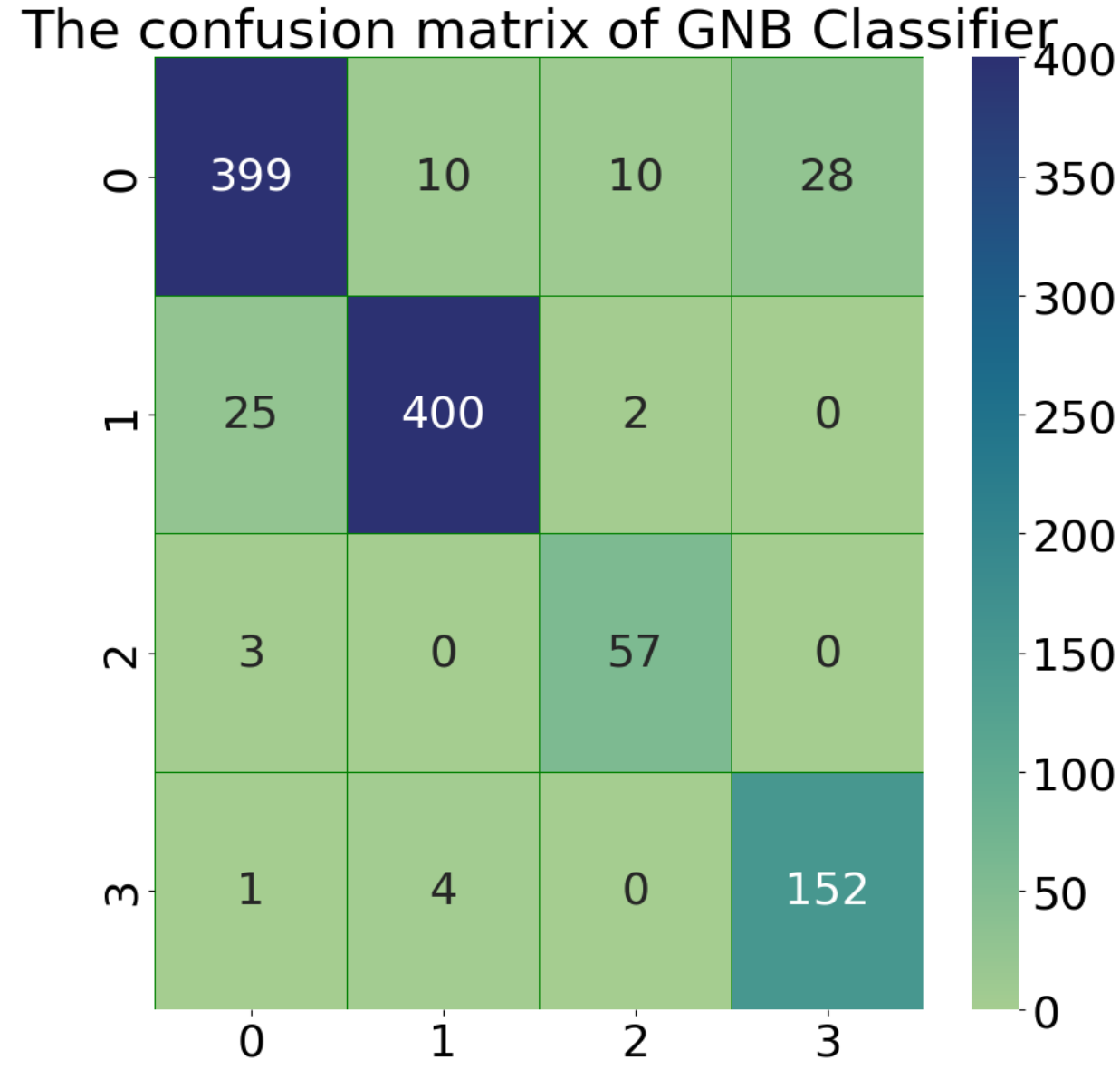
In [4]:
```
SVM_cm, SVM_ac, SVMclassifier = rmc.SVM_learning()
Accuracy.append(SVM_ac)
print('The accuracy of SVM: ', SVM_ac)

f, ax = plt.subplots(figsize =(10,10))
sns.heatmap(SVM_cm,cmap = "crest", annot = True,linewidths=0.5 ,linecolor =
ax.set_title('The confusion matrix of SVM Classifier')
plt.show()
```

```
The accuracy of SVM:  0.9642529789184234
```



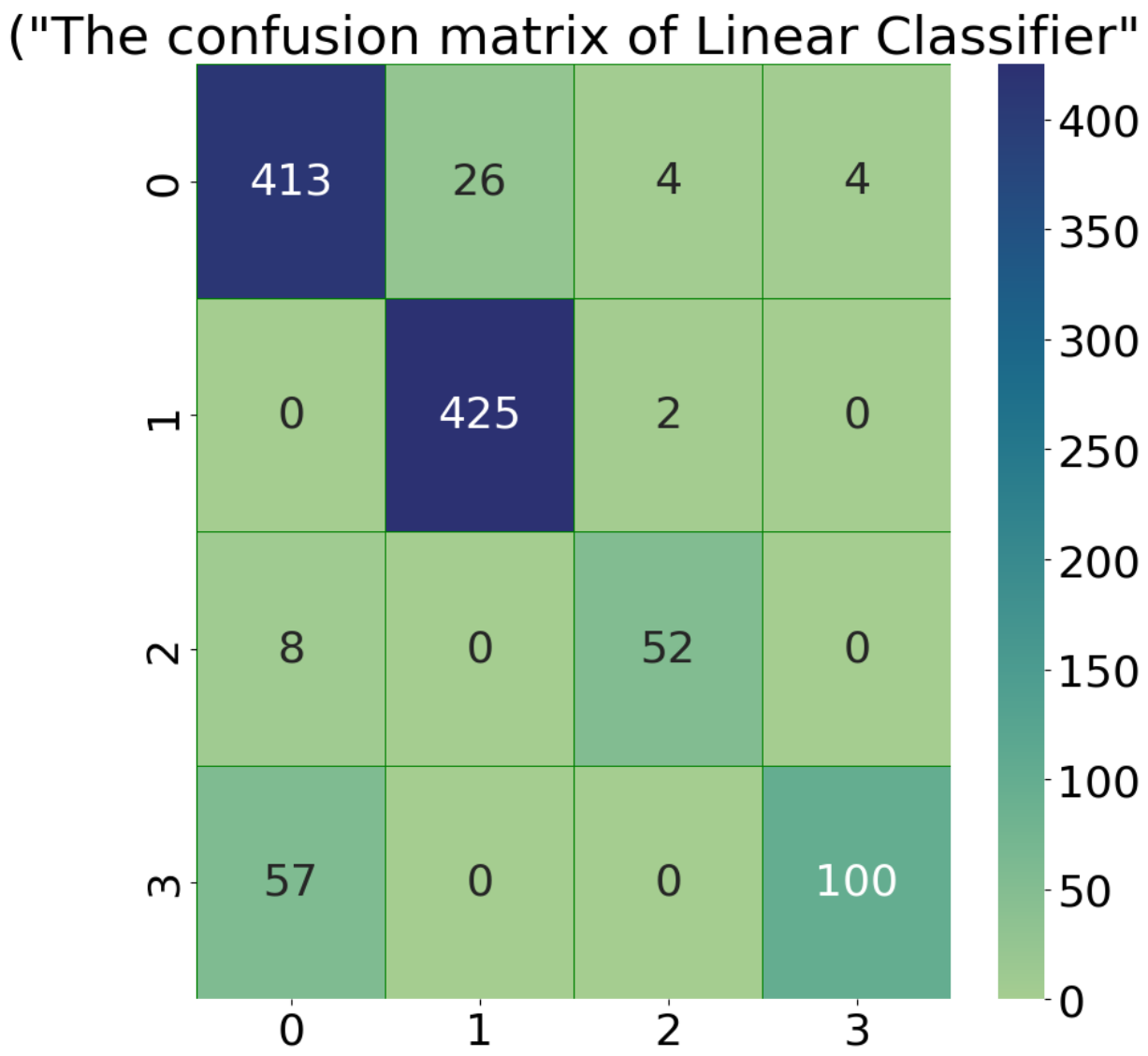
In [5]:
```
GNB_cm, GNB_ac, GNBclassifier = rmc.GaussianNB_learning()
Accuracy.append(GNB_ac)
print('The accuracy of GNB: ', GNB_ac)
f, ax = plt.subplots(figsize =(10,10))
sns.heatmap(GNB_cm,cmap = "crest", annot = True,linewidths=0.5 ,linecolor =
ax.set_title('The confusion matrix of GNB Classifier')
plt.show()
```

```
The accuracy of GNB:  0.923923006416132
```



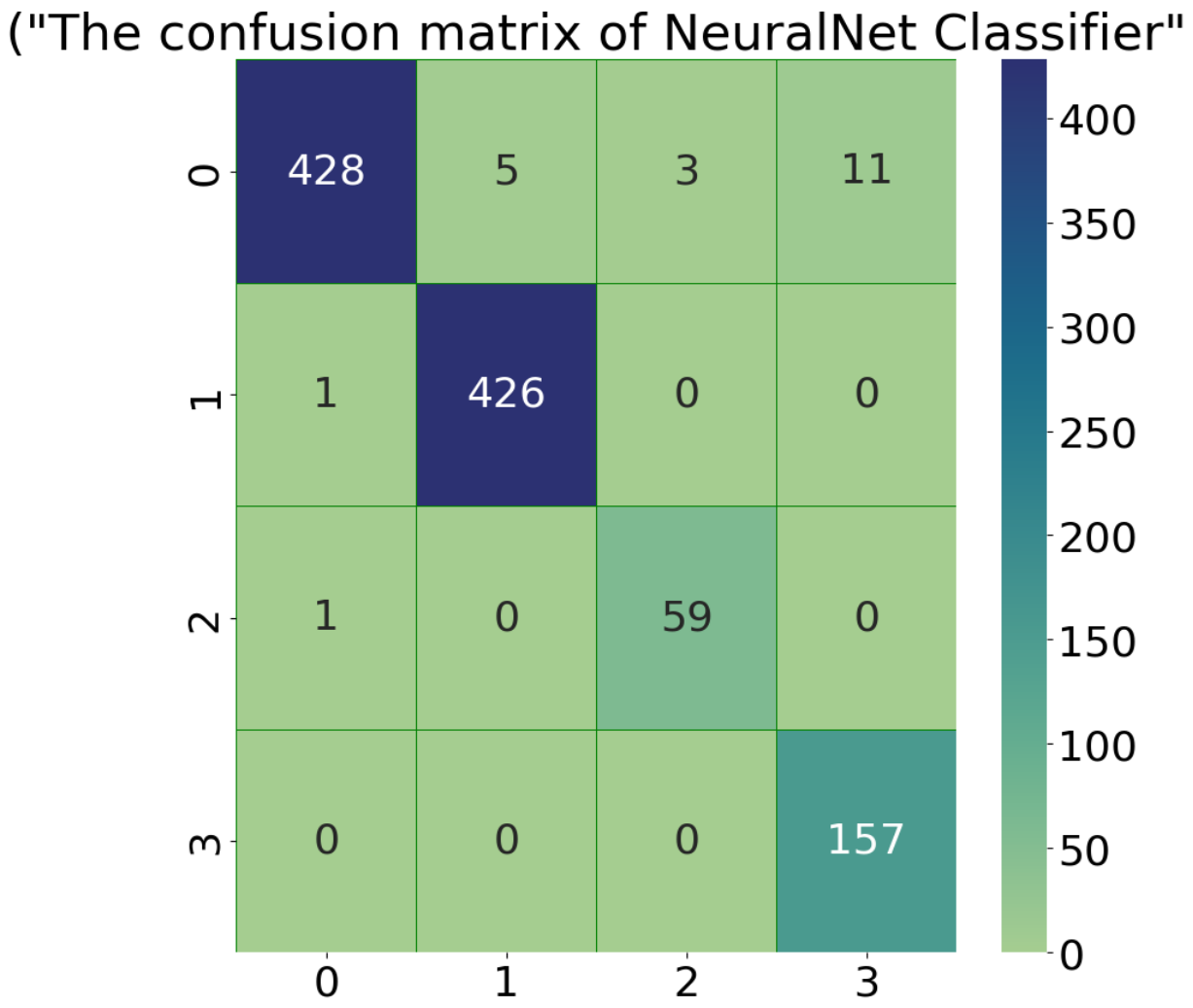
In [6]:
```
Lin_cm, Lin_ac, Linclassifier = rmc.Linear_Classifier()
Accuracy.append(Lin_ac)
print('The accuracy of Linear Classifier: ', Lin_ac)

f, ax = plt.subplots(figsize =(10,10))
sns.heatmap(Lin_cm,cmap = "crest", annot = True,linewidths=0.5 ,linecolor =
ax.set_title('("The confusion matrix of Linear Classifier"')
plt.show()
```

```
The accuracy of Linear Classifier:  0.9074243813015582
```

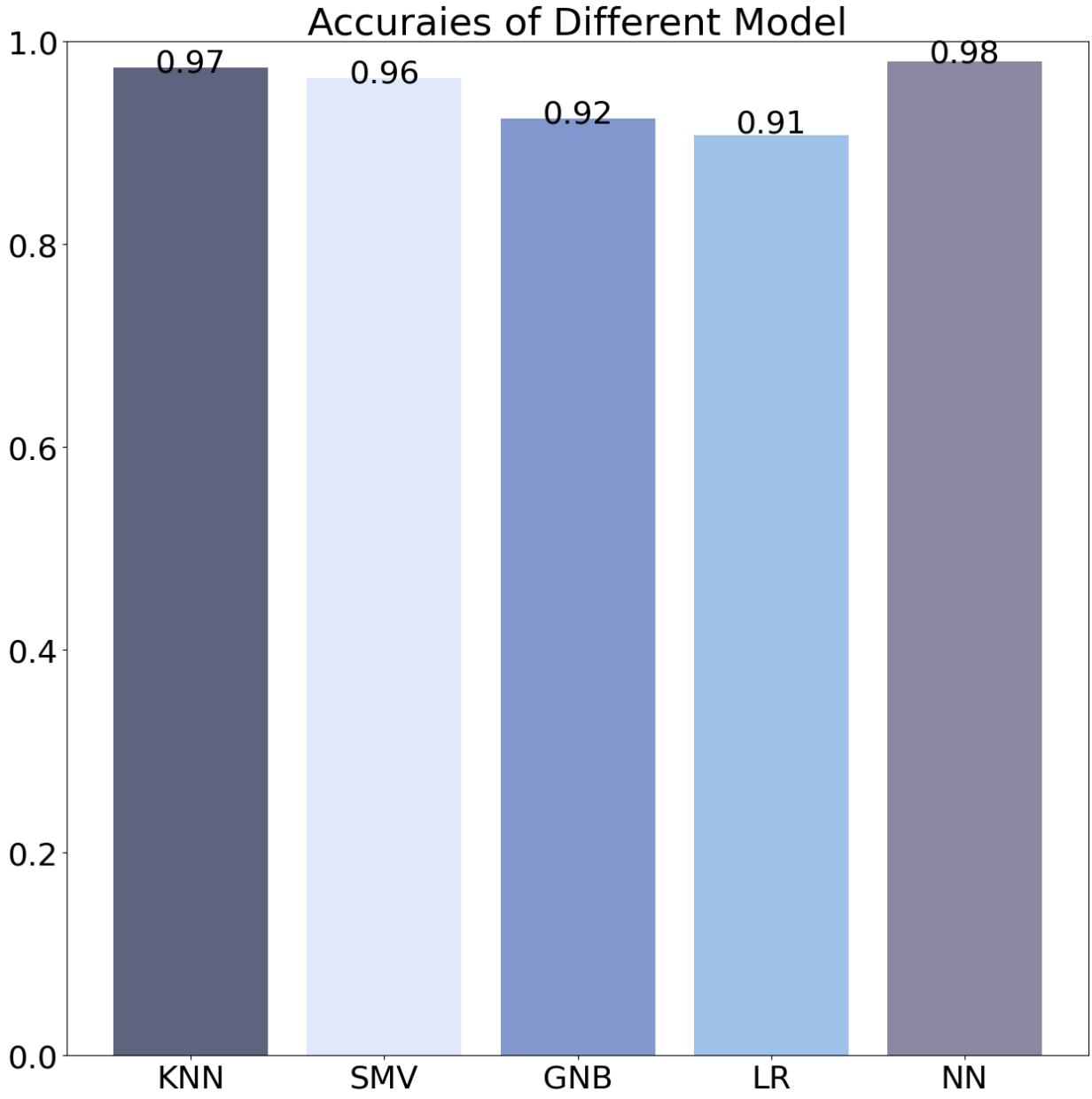


In [7]:
```
NNet_cm, NNet_ac, NNetclassifier = rmc.NeuralNet_Classifier()
Accuracy.append(NNet_ac)
print('The accuracy of NeuralNet: ', NNet_ac)

f, ax = plt.subplots(figsize =(10,10))
sns.heatmap(NNet_cm,cmap = "crest", annot = True,linewidths=0.5 ,linecolor
ax.set_title('("The confusion matrix of NeuralNet Classifier"')
plt.show()
```

```
The accuracy of NeuralNet:  0.9807516040329972

/Users/stlp/opt/anaconda3/envs/env_RMC/lib/python3.9/site-packages/sklear
n/neural_network/_multilayer_perceptron.py:679: ConvergenceWarning: Stoch
astic Optimizer: Maximum iterations (200) reached and the optimization ha
sn't converged yet.
  warnings.warn(
```



In [25]:
```
Models = [ 'KNN', 'SMV', 'GNB', 'LR', 'NN']
acc = [0.97,0.96,0.92,0.91,0.98]
xs = [0,1,2,3,4]
Statistics = pd.DataFrame(Accuracy, index=Models)

plt.figure(figsize = (15, 15))
plt.bar(xs,Accuracy,color=[(87/235,92/235,115/235),
                           (206/235,214/235,231/235),
                           (119/235,139/235,190/235),
                           (145/235,178/235,215/235),
                           (128/235,124/235,147/235)])
def addlabels(x,y):
    for i in range(len(x)):
        plt.text(i,y[i],y[i],ha = 'center')
addlabels(xs, acc)


# HERE tell pyplot which labels correspond to which x values
plt.xticks(xs,Models)
plt.title('Accuraies of Different Models')
plt.ylim((0, 1))
plt.show()
```


Accuraies of Different Model
1.0
0.8
0.6
0.4
0.2
0.0
0.97
0.96
0.92
0.91
0.98
KNN
SMV
GNB
LR
NN