**Wireless Indoor Localization**

In this dataset, the value of all the feature are already in a good shape, most of them are all in same range.

**The classifier I used:**

1. Naïve Bayes
2. SVM
3. Perceptron
4. OneVsRest
5. KNN
6. Decision Tree
7. Random Forest

**Preprocessing:**

1. Normalization

Only Perceptron classifier result is not stable due to shuffled data, sometimes it comes out with 75% of accuracy, sometimes even 93 % of accuracy.

I tried to use Normalization make it better, but it seems like the same, so this dataset is not necessary to preprocess by Normalization.

All other classifier can always come out with 96% - 98% of accuracy even after shuffling.

1. Standardization

Then I tried Standardization, it can improve Perceptron classifier accuracy to 89% - 94%. All other classifier can also get between 96%- 98%.

**Feature extraction:**

1. PCA

I also tried PCA to preprocess, I used (n\_components =2) to check the variance ratio in all the feature. It shows up that 2 of the feature have ﻿[0.65935397 0.19528077] up to 85% of energy. After I tried it, the results seem the same as before.

And I combined PCA with Standardization, the result of Perceptron can even get like 95% of accuracy.

**Analysis:**

Final report decision is that even with no preprocessing for this dataset, it can already come out with good accuracy.

After **(PCA + Standardization)** can improve the Perceptron Classifier, I think it can also speed up the time and improve a little bit of accuracy for all the classification.

Corss-Validation Accuracy of mean and std, test accuracy, confusion are shown below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ﻿cross\_val\_acc mean | ﻿cross\_val\_acc std | Test Accuracy | Confusion Matrix |
| Naïve Bayes | ﻿0.981 | ﻿﻿0.005 | ﻿﻿97.25 % | ﻿ [[ 97 0 3 0]  [ 0 96 4 0]  [ 1 1 96 2]  [ 0 0 0 100]] |
| SVM | ﻿0.979 | ﻿0.002 | 97.5 % | ﻿ ﻿[[ 97 0 2 1]  [ 0 95 5 0]  [ 1 1 98 0]  [ 0 0 0 100]] |
| Perceptron | ﻿0.959 | ﻿﻿0.029 | ﻿95.75 % | ﻿ ﻿[[ 98 0 2 0]  [ 0 88 12 0]  [ 1 0 97 2]  [ 0 0 0 100]] |
| OneVsRest | ﻿0.979 | ﻿﻿0.005 | ﻿﻿97.0 % | ﻿ [[ 99 0 1 0]  [ 0 93 7 0]  [ 1 1 96 2]  [ 0 0 0 100]] |
| KNN | ﻿0.979 | ﻿0.013 | ﻿97.25 % | ﻿﻿ [[ 98 0 2 0]  [ 0 95 5 0]  [ 1 1 96 2]  [ 0 0 0 100]] |
| Decision Tree | ﻿﻿0.966 | ﻿﻿0.009 | ﻿﻿97.5 % | ﻿ ﻿[[ 96 0 3 1]  [ 0 96 4 0]  [ 1 1 98 0]  [ 0 0 0 100]] |
| Random Forest | ﻿0.977 | ﻿0.007 | ﻿96.5 % | ﻿ [[ 96 0 3 1]  [ 0 94 6 0]  [ 1 2 96 1]  [ 0 0 0 100]] |

All the classifier can predict class 4 in 100%, so I think the data in class 4 must be very clustered.

It’s proved by the visualization of test results below