BigData&DataAnalytics

Analysis of algorithm selected (KNN)

Mauricio Carvajal
IOT ANALYTICS

Table of contents

1. Introduction 3

2. Model Selection 3

2.1 Selection based on performance 3

2.1 Selection based on Test time 3

2.2 Conclusion of the model selected 3

3 k-Nearest Neighbors 3

5.1 Parameters 4

5.2 Results 4

5.3 Conclusion 4

4 Confusion Matrix 5

5 Overall statistics 5

1. Introduction

In this section we analysis the chosen model, in which we present the confusion matrix and the performance measures.

2. Model Selection

In this section, we analyze which is the best model that performance for the wifi localization. It's important to mention that all these models the data has been preprocess using PCA to reduce the test time.

2.1 Selection based on performance

Base on the performance indicator, more specify in accuracy and kappa the Random Forest present the most accurate model base on the accuracy and kappa. In second place we can review that knn and the different is about 2%.

Model	**Accuracy**	**Kappa**
RF	0.9545096	0.9503639
knn	0.9312672	0.9250274
C50	0.8956562	0.8861759
SVM	0.9466791	0.9418449

2.2 Selection based on Test time

In data science another critical indicator in order to choose the model, besides their performance it's the test time that require to build the model. For example, in these case we notice that the KNN is running 8 time faster of the random forest, but when the performance is compare, there is not huge difference it's about 2%.

Algo	**user**	**system**	**elapsed**
RF	12948.79	62.96	13020.36
KNN	1516.97	11.89	1531.29
C50	2711.25	9.33	2721.97
SVM	5689.28	43.93	5738.09

2.3 Conclusion of the model selected

Base on the test time and the performance we conclude that the best model is the SVM, due to it took less than the half of the time than RF to complete and the performance is 2% less, and actually is ~93. So still a very good model.

3 k-Nearest Neighbors

In pattern recognition, the k-nearest neighbors' algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space.

5.1 Parameters

- 15791 samples
- 100 predictor
- 13 classes: '0 0', '0 1', '0 2', '0 3', '1 0', '1 1', '1 2', '1 3', '2 0', '2 1', '2 2', '2 3', '2 4'
- Pre-processing: centered (100), scaled (100)
- Resampling: Cross-Validated (10 fold, repeated 3 times)

5.2 Results

Resampling results across tuning parameters:

k	Accuracy	Kappa
5	0.9312672	0.9250274
7	0.9256106	0.9188583
9	0.9194049	0.9120872
11	0.9140647	0.9062594
13	0.9093572	0.9011250
15	0.9067605	0.8982884
17	0.9010400	0.8920474
19	0.8972617	0.8879243
21	0.8951920	0.8856646
23	0.8925535	0.8827907

5.3 Conclusion

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was k = 5.

4 Confusion Matrix

Calculates a cross-tabulation of observed and predicted classes with associated statistics.

A confusion matrix is a summary of prediction results on a classification problem.

The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix.

The confusion matrix shows the ways in which your classification model is confused when it makes predictions.

It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

	Reference													
		0_0	0_1	0_2	0_3	1_0	1_1	1_2	1_3	2_0	2_1	2_2	2_3	2_4
Prediction	0_0	261	15	3	0	0	0	0	0	0	0	0	0	0
	0_1	19	365	17	5	0	0	0	0	0	0	0	0	0
	0_2	3	9	339	29	0	0	1	0	0	0	0	0	0
	0_3	0	1	43	334	0	0	1	0	0	0	0	0	0
	1_0	0	0	0	0	342	14	1	0	0	0	0	0	0
	1_1	0	0	0	0	6	386	5	0	0	0	0	0	0
	1_2	0	0	0	0	1	5	347	8	0	0	0	0	0
	1_3	1	1	0	0	0	1	14	234	3	0	0	0	0
	2_0	0	0	0	0	0	0	0	5	482	7	1	1	0
	2_1	0	0	0	0	0	0	1	0	5	547	18	5	1
	2_2	0	0	0	1	0	0	0	0	1	9	368	15	0
	2_3	0	0	0	0	0	0	0	1	0	5	20	665	12
	2_4	0	0	0	0	0	0	0	0	0	0	0	1	272

5 Overall statistics

- Accuracy : 0.9401
- 95% CI : (0.9333, 0.9463)
- No Information Rate : 0.1307
- P-Value [Acc > NIR] : < 2.2e-16

	Class : 0_0	Class : 0_1	Class : 0_2	Class : 0_3	Class : 1_0	Class : 1_1	Class : 1_2	Class : 1_3	Class : 2_0	Class : 2_1	Class : 2_2	Class : 2_3	Class : 2_4
Sensitivity	0,92	0,93	0,84	0,91	0,98	0,95	0,94	0,94	0,98	0,96	0,90	0,97	0,95
Specificity	1,00	0,99	0,99	0,99	1,00	1,00	1,00	1,00	1,00	0,99	0,99	0,99	1,00
Pos Pred Value	0,94	0,90	0,89	0,88	0,96	0,97	0,96	0,92	0,97	0,95	0,93	0,95	1,00
Neg Pred Value	1,00	0,99	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00	0,99	1,00	1,00
Prevalence	0,05	0,07	0,08	0,07	0,07	0,08	0,07	0,05	0,09	0,11	0,08	0,13	0,05
Detection Rate	0,05	0,07	0,06	0,06	0,07	0,07	0,07	0,04	0,09	0,10	0,07	0,13	0,05
Detection Prevalence	0,05	0,08	0,07	0,07	0,07	0,08	0,07	0,05	0,09	0,11	0,07	0,13	0,05
Balanced Accuracy	0,96	0,96	0,92	0,95	0,99	0,97	0,97	0,97	0,99	0,98	0,95	0,98	0,98