BigData&DataAnalytics

Summary of the models and its Results

Mauricio Carvajal

IOT ANALYTICS

Table of contents

1. Introduction

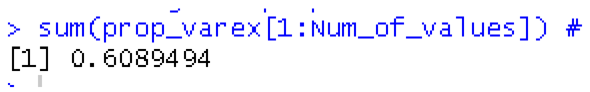
In this section we have the summary of the results that has been obtain thru the project. The main idea of this section is present the results, but the full analysis and details are provide in the full technical document.

2. Dimensionality Reductions

For this project, we have 520 independent variables, so this makes a complex analisys, and also cause a lot of test time. For this purpose, we use Principal Component Analysis (PCA). Actually, the primary purpose of PCA is not as a ways of feature removal. PCA can reduce dimensionality but it won't reduce the number of features / variables in your data. What this means is that you might discover that you can explain 99% of variance in your 1000 feature dataset by just using 3 principal components but you still need those 1000 features to construct those 3 principal components, this also means that in the case of predicting on future data you still need those same 1000 features on your new observations to construct the corresponding principal components.

2.1 Results

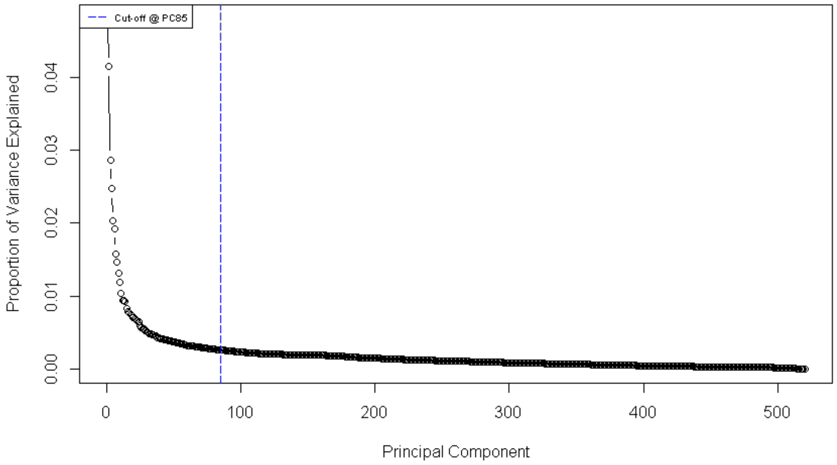
After applying the PCA, we conclude that the number of variables that we require are 85, which give us 60%



```
> sum(prop_varex[1:Num_of_values]) #
[1] 0.6089494
```

2.2 Plotting PCA

In the following plot, is show the proportion of variance, in this case we should up to 85.



3. SVM

In machine learning, support-vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

3.1 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)

3.2 Results

Resampling results across tuning parameters:

C	Accuracy	Kappa
0.25	0.8919043	0.8820412
0.50	0.9124817	0.9044980
1.00	0.9269187	0.9202537
2.00	0.9333158	0.9272575
4.00	0.9385088	0.9329298
8.00	0.9421190	0.9368704
16.00	0.9447175	0.9397052
32.00	0.9466791	0.9418449
64.00	0.9456031	0.9406706
128.00	0.9452239	0.9402552

3.3 Conclusion

Tuning parameter 'sigma' was held constant at a value of 0.01519876. Accuracy was used to select the o ptimal model using the largest value.
The final values used for the model were sigma = 0.01519876 and C = 32.

4. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

4.1 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)

4.2 Results

Resampling results across tuning parameters:

mtry	Accuracy	Kappa
4	0.9532638	0.9490024
5	0.9540666	0.9498794
6	0.9539182	0.9497173
8	0.9545096	0.9503639
10	0.9538129	0.9496036

4.3 Conclusion

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 8.

5 k-Nearest Neighbors

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regrestion. 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.

6 c50model

6.1 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)

6.2 Results

Resampling results across tuning parameters:

model	winnow	Accuracy	Kappa
rules	FALSE	0.8951495	0.8856298
rules	TRUE	0.8956562	0.8861759
tree	FALSE	0.8931240	0.8834259
tree	TRUE	0.8932710	0.8835787

6.3 Conclusion

- Tuning parameter 'trials' was held constant at a value of 1
- Accuracy was used to select the optimal model using the largest value.
- The final values used for the model were trials = 1, model = rules and winnow = TRUE.

7. Test Time

The teatime is a critical factor in machine learning, this is due to amount of data. For instance if we have to an algorimt that has better performance, is not always the case in which this has been seleted.

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

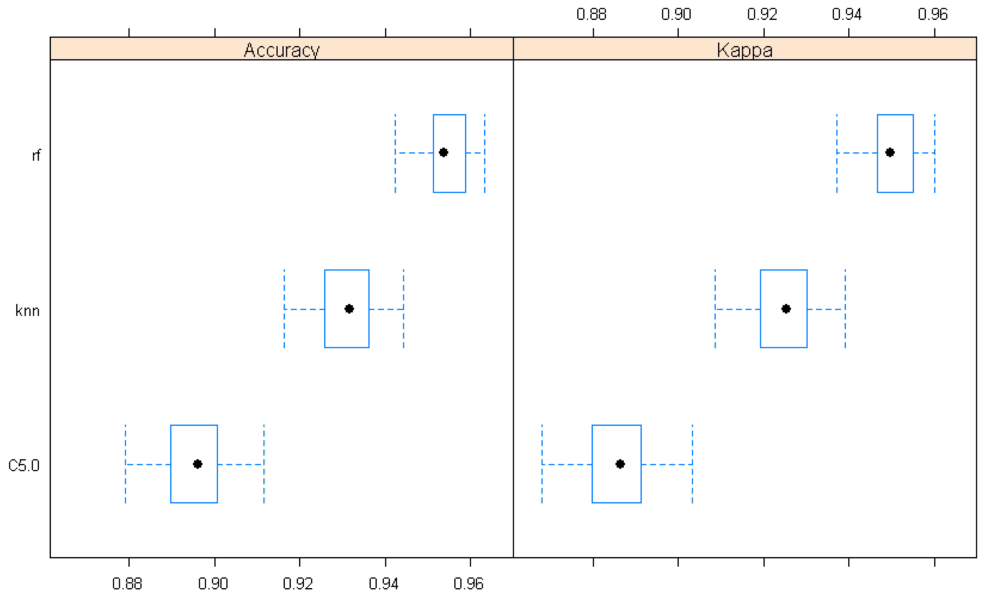
8 Comparing in terms of Resampling

Resampling methods are an indispensable tool in modern statistics. They involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model. For example, in order to estimate the variability of a linear regression fit, we can repeatedly draw different samples from the training data, fit a linear regression to each new sample, and then examine the extent to which the resulting fits differ.

	Algo	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
Accuracy	C5.0	0.8790374	0.8897513	0.8961036	0.8956562	0.9004912	0.9113924	0
	rf	0.9423686	0.9513316	0.9538120	0.9545096	0.9586893	0.9632679	0
	knn	0.9162968	0.9260724	0.9316455	0.9312672	0.9359177	0.9442685	0
Kappa	C5.0	0.8680433	0.8797410	0.8866706	0.8861759	0.8913886	0.9033109	0
	rf	0.9371195	0.9468825	0.9496059	0.9503639	0.9549294	0.9599178	0
	knn	0.9086731	0.9193574	0.9254393	0.9250274	0.9300704	0.9391997	0

8.1 Plot

In this section we plot the different accuracy and kappa of the models that has been evaluate it. Having a visual representation is very useful when there is a need to explain which model is having the best performance.



9. Confusion Matrix

The confusion matrix calculates a cross-tabulation of observed and predicted classes with associated statistics. In this particular case we want to show the relationship between the different algorisms that has been used.

9.1 Accuracy

Accuracy is the percentage of correctly classifies instances out of all instances. It is more useful on a binary classification than multi-class classification problems because it can be less clear exactly how the accuracy breaks down across those classes.

	c5.0	rf	knn
c5.0		-0.05885	-0.05885
rf	< 2.2e-16		0.02324
knn	< 2.2e-16	< 2.2e-16	

9.2 Kappa

Kappa or Cohen’s Kappa is like classification accuracy, except that it is normalized at the baseline of random chance on your dataset. It is a more useful measure to use on problems that have an imbalance in the classes (e.g. 70-30 split for classes 0 and 1 and you can achieve 70% accuracy by predicting all instances are for class 0).

	c5.0	rf	knn
c5.0		-0.06419	-0.03885
rf	< 2.2e-16		0.02534
knn	< 2.2e-16	< 2.2e-16	