**Lesson Learned Report**

Course 5 Task 3 is definitely more challenging than previous tasks, here are the problems that I’ve encountered, and solutions found to finally complete the task.

**iPhone**

Random Forest classifier with RFE featured dataset were selected to do predictions on large unlabeled dataset. Functions postResample and ConfusionMatrix were also used to generate the highest accuracy and kappa score with same model and featured dataset.

| RandomForest Model with iphoneCOR/NZV/RFE | | | |
| --- | --- | --- | --- |
| iPhone | iphoneCOR | iphoneNZV | iphoneRFE |
| Time(s) | 55.857 | 35.946 | **58.255** |
| Accuracy | 0.6759888 | 0.7599461 | **0.7764855** |
| Kappa | 0.3247135 | 0.5292822 | **0.5660296** |

| postResample & ConfusionMatrix value with iphoneCOR/NZV/RFE | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| iPhone | RandomForest COR | | RandomForest NZV | | RandomForest RFE | |
|  | postResample | ConfusionMatrix | postResample | ConfusionMatrix | **postResample** | **ConfusionMatrix** |
| Accuracy | 0.6824173 | 0.6824 | 0.6824173 | 0.7762 | **0.7877900** | **0.7878** |
| Kappa | 0.3364190 | 0.3364 | 0.3364190 | 0.5638 | **0.5891535** | **0.5892** |

As the above tables show the highlighted accuracy and kappa score when using RandomForest classifier and RFE dataset yields the highest score: 0.7764855 / 0.5660296; even higher score when using postResample and ConfusionMatrix 0.78779 / 0.5891535 & 0.7878 / 0.5892.

**Samsung**

Same methodology was applied to Samsung dataset. C5.0 classifier with RFE featured dataset were selected to predict large dataset. Functions postResample and ConfusionMatrix were also utilized to generate the best accuracy / kappa score. The time taking for the model to is very similar, so the accuracy is the key for choosing the optimized model and dataset.

| C5.0 Model with samsungCOR/NZV/RFE | | | |
| --- | --- | --- | --- |
| Samsung | samsungCOR | samsungNZV | samsungRFE |
| Time(s) | 8.682 | 6.038 | **10.782** |
| Accuracy | 0.6847871 | 0.7530753 | **0.7673369** |
| Kappa | 0.3020627 | 0.4949592 | **0.5297335** |

| postResample & ConfusionMatrix value with samsungCOR/NZV/RFE | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Samsung | C5.0 COR | | C5.0 NZV | | C5.0 RFE | |
|  | postResample | ConfusionMatrix | postResample | ConfusionMatrix | **postResample** | **ConfusionMatrix** |
| Accuracy | 0.6859267 | 0.6859 | 0.7574162 | 0.7574 | **0.7707381** | **0.7707** |
| Kappa | 0.3109697 | 0.311 | 0.5090933 | 0.5091 | **0.5408266** | **0.5408** |

**Lesson Learned**

The main mistakes that I made I thought it would be good to share with others are:

* Make sure to use stop(cluster) command at end of each task,

I thought I would use this at the very end of this task, like when we finished doing prediction, then we would stop the cluster running. So I had it on all the time, it was a disaster when I was trying to get the RFE dataset, it run forever and wouldn’t return any result.

* Don’t run RFE on the large matrix, as this dataset has no labels, no prediction value for it to run properly.
* Do use the RFE Results to prune the large matrix so that the features are identical to the training dataset; use predictors(rfeResults) to return the features on the large dataset that match the features to the small dataset to be able to do prediction.