# SURVEY ANSWERS PREDICITON

**OBJECTIVES**

Using R language to train the predicting model for incomplete survey answers prediction

**PREPARATORY STAGE**

Initially, we have 2 datasets, one of which we used for the prediction model training and the second one used for prediction purpose.

Most of operations were conducted with “caret” library of R.

Complete processing script is added to the end of this report.

The data preparation steps I proceeded:

1. Data Loading
2. Structure Observation
3. Data types Adjusting
4. Data Sampling
5. Data Partitioning
6. 5 CPU cores activation (to boost the speed of the algorithms)

The results of this step were:

* Training dataset consisted of sampled 75% rows of the initial dataset.
* Testing dataset consisted of the rest 25% rows.

**MODEL BUILDING STAGE**

During this step I built 3 models using Decision Tree C5.0 algorithm and 2 models of Random Forest algorithm. Then, the performance comparison was made.

In order to raise performance, model tuning methods were used.

1. **C5.0 Model #1**

Tuning parameters:

Number of folds = 10

Resampling = Cross-Validation with 1 repeat

Amount of granularity in the tuning parameter grid = 1

Model #1 assessing:

Accuracy: 0.8440

Kappa: 0.6869

1. **C5.0 Model #2**

Tuning parameters:

Number of folds = 10

Resampling = Cross-Validation with 3 repeats

Amount of granularity in the tuning parameter grid = 15

Model #2 assessing:

Accuracy: 0.9250

Kappa: 0.8403

After training the Model #2, I decided to estimate the importance of attributes in prediction using varImp(() function and cut down all unimportant ones.

The result showed that only “salary” and “age” play role in prediction. The rest attributes gave 0 influence.

1. **C5.0 Model #3**

Tuning parameters:

Columns = “salary”, “age”

Number of folds = 10

Resampling = Cross-Validation with 3 repeats

Amount of granularity in the tuning parameter grid = 10

Model #3 assessing:

Accuracy: 0.9232

Kappa: 0.8359

Cutting down the useless attributes gave an increase in neither accuracy nor kappa.

After C5.0 models, I came over to **Random Forest.**

1. **Random Forest Model #5**

Tuning parameters:

Number of folds = 10

Resampling = Cross-Validation with 1 repeat

Amount of granularity in the tuning parameter grid = 1

Model #3 assessing:

Accuracy: 0.9230

Kappa: 0.8367

1. **Random Forest Model #5**

Tuning parameters:

Number of folds = 10

Resampling = Cross-Validation with 3 repeats

Amount of granularity in the tuning parameter grid = 15

Model #6 assessing:

Accuracy: 0.9237

Kappa: 0.8379

All the models were trained and the best one had to be selected. To evaluate each model, I used postResample() function to compare predicted outcomes with testing Y-attributes.

postResample():

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metrics | Model #1 | Model #2 | Model #3 | Model #4 | Model #5 |
| Accuracy | 0.8225546 | 0.9232013 | 0.9227971 | 0.9203719 | 0.9240097 |
| Kappa | 0.6437074 | 0.8354281 | 0.8348752 | 0.8306145 | 0.8385218 |

Thus, Model #5 (Random Forest tuned algorithm) was selected as a production model to apply to Incomplete Survey Dataset.

**FINAL STAGE**

Here I applied the model chosen to incomplete dataset and draw some charts on customers preferences.

After loading and preprocessing the data, I applied the model and run postResample() function to estimate the performance:

|  |  |
| --- | --- |
| Metrics | Model #5 |
| Accuracy | 0.3866000 |
| Kappa | 0.01203978 |

At first sight, this result looked quite unexpected. However, this happened because all missing Y-values in the incomplete dataset were presented by “0” and when the model outputted “1” the mismatch encountered.

Now we got complete survey answers and predicted ones which we can combine and give the final conclusion.

|  |  |  |  |
| --- | --- | --- | --- |
| Answers | Complete | Incomplete | Total |
| Number of “0” | 3744 | 1880 | 5624 |
| Number of “1” | 6154 | 3120 | 9274 |