ModelPerformance

Kannu-priya

April 21, 2018

Contents

1. Data import

The cleaned data from logistic regression model development is imported.

```
#Importing employee data cleaned and exported from logistic regression model (model-performance-linear.
dt1 <- read.csv(file = 'D:/Kannu Priya/DataScience/DataSciencePrinciples/AttritionProject/data/dataR.cs
```

---

2. Create Training & testing data set

Create Training & testing data set

```
#Setting a seed before taking random sample so that same sample is reproduced for future reference
set.seed(10000)

# Create training set using sample_frac
train_set <- dt1 %>% sample_frac(.7)
# let's create our testing set using the employee code (per.id) column
test_set <- dt1 %>% filter(!(per.id %in% train_set$per.id))
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

---

3. Decision Tree Model training

Let's fit a model using a classification tree, using the same features that are shortlisted for Logistic regression and plot the final decision tree.

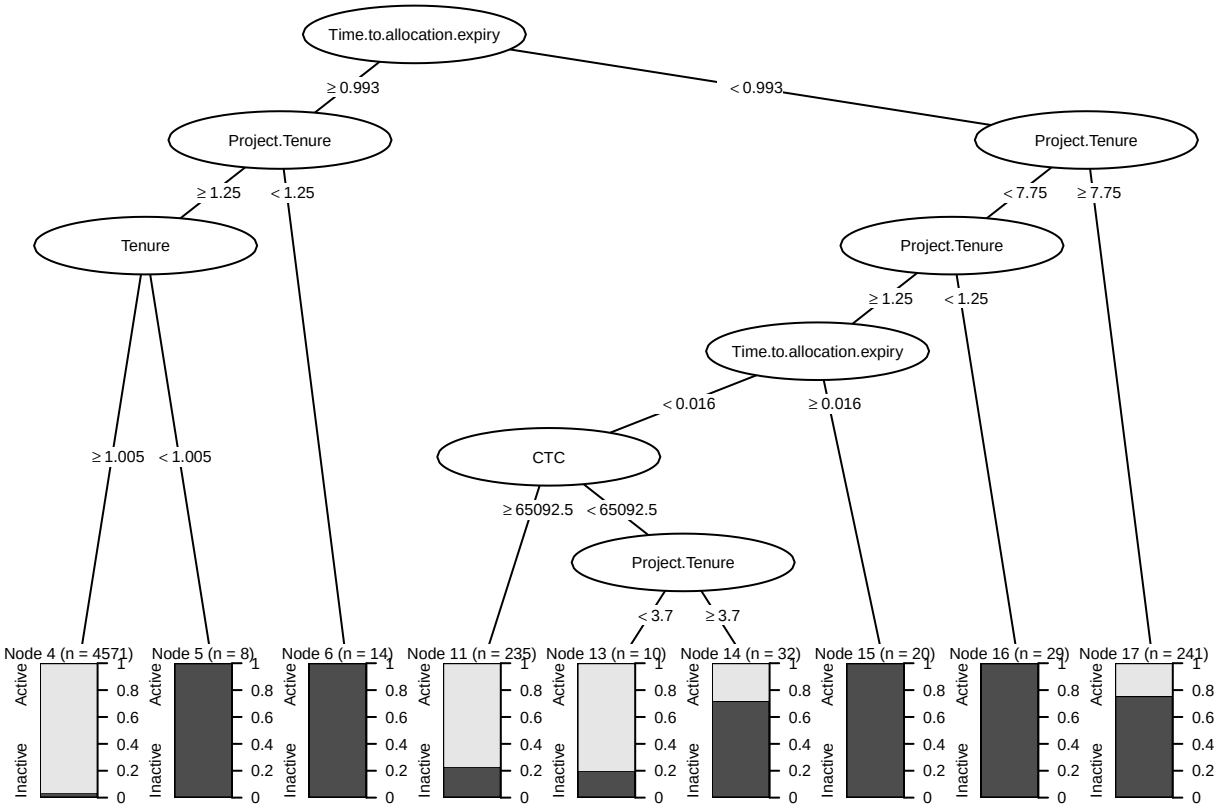
The most important features as per plotted decision tree are Time.to.allocation.expiry, Project.Tenure, Tenure and CTC.

Surprisingly, in logistic regression top 3 significant features were time.to.allocation.expiry, time.to.promotion and time.to.visa.expiry.

Time.to.allocation.expiry is most significant in Decision tree as well but others are not displayed in tree node.

```
#Fit decision tree using shortlisted features
tree_mod <- rpart(Status ~ Age + Tenure + CTC + L1 + Project.Tenure + Time.to.allocation.expiry + Time.

plot(as.party(tree_mod), gp = gpar(fontsize = 6), inner_panel = node_inner, ip_args = list(abbreviate =
```



Observations from the tree:

Most striking is the contrast of Node 4 and Node 5/6: The nodes represent employees whose current project is valid for beyond 0.99 months. There is strong probability of employee staying active if project.tenure >=1.25 AND Tenure > 1.005. On the other hand, if the project.tenure <1.25 or if project.tenure >=1.25 but employee's tenure < 1.005 , then the probability of attrition is very high. From business side, it could mean employees who come out of project and possibly going on bench, have a job insecurity. They are more likely to look for outside job opportunities. Even though they may get a project within company, but once they get a confirmation on external offer, they quit the company (thus low project.tenure <1.25 have high attrition).

Tenure here reflect employee's overall commitment. New employees with just 1 year spent in the company are easy to lose (tenure <1.005). A trigger such as project end, is triggering their attrition decision

---

4. Model Performance Evaluation

Applying the classification tree model object and fitting all independent features of the tested dataset in the model. The test_tree is a vector that holds the predicted status outcomes of employees and we are adding the vector to test data set.

```
#Use predict function to generate predictions for the testing set and add the predictions in test data
test_tree <- predict(tree_mod, newdata = test_set)[,2]
test_set <- test_set %>% mutate(test_tree = test_tree)
head(test_set$test_tree)
```

```
##          1          2          3          4          5          6
## 0.03850361 0.03850361 0.03850361 0.03850361 0.03850361 0.03850361
```

To classify the probabilities to binary outcome - Active or Inactive, lets plot ROC curve for picking probability cut-off value.

4.1 ROC curve plot

The ROC curves and AUC value reveal that the models are fitting the data.

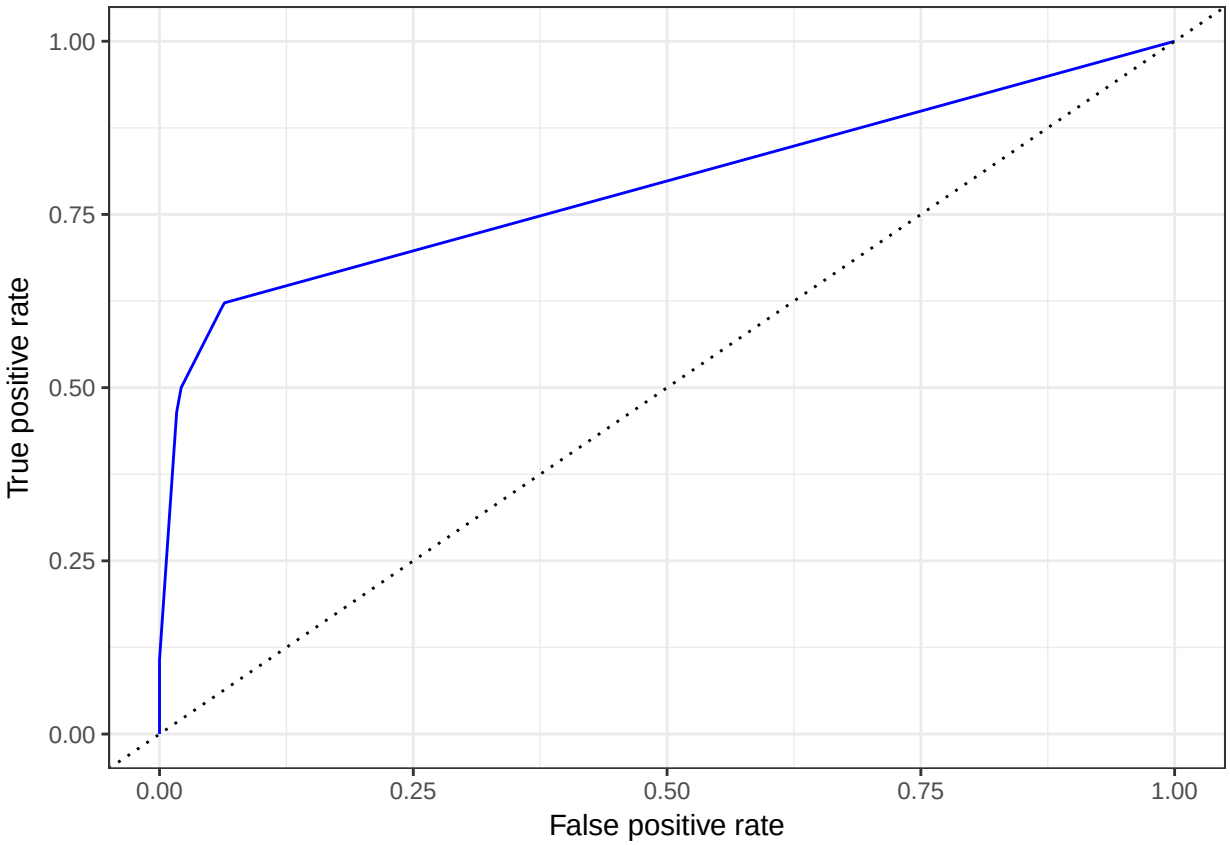
```
#create the prediction object
pred_tree <- prediction(predictions = test_tree, labels = test_set$Status)

# get the FPR and TPR for the tree model
perf_tree <- performance(pred_tree, measure = 'tpr', x.measure = 'fpr')
perf_tree_tbl <- tibble(perf_tree@x.values[[1]], perf_tree@y.values[[1]])

# Change the names of the columns of the tibble
names(perf_tree_tbl) <- c('fpr', 'tpr')

# Plotting function for plotting a nice ROC curve using ggplot
plot_roc <- function(perf_tbl) {
p <- ggplot(data = perf_tbl, aes(x = fpr, y = tpr)) +
geom_line(color = 'blue') +
geom_abline(intercept = 0, slope = 1, lty = 3) +
labs(x = 'False positive rate', y = 'True positive rate') +
theme_bw()
return(p)
}

# Create the ROC curves using the function above
plot_roc(perf_tree_tbl)
```



4.2 AUC

```
# calculate the AUC
auc_tree <- performance(pred_tree, measure = 'auc')

# extract the AUC value
auc_tree@y.values[[1]]
```

```
## [1] 0.7902406
```

The AUC of greater than 0.7 and ROC curve reveal that the model is fitting the data pretty decently.

The AUC value from logistic regression model was 0.78, so as per AUC metric decision tree is coming as better model

4.3 Confusion Matrix

As per ROC curve, will take TPR (true positive rate) greater than 0.62 and FPR (false positive rate) of less than 0.06. The corresponding probability cutoff for classifying the prediction is calculated as below.

```
#create a tibble of fpr, tpr and cutoffs
perf_tbl <- tibble(fpr = perf_tree@x.values[[1]], tpr = perf_tree@y.values[[1]], cutoffs = perf_tree@al

perf_tbl %>% filter(fpr < .125 & tpr > .62)
```

```
## # A tibble: 1 x 3
```

```
##      fpr   tpr cutoffs
##    <dbl> <dbl>   <dbl>
## 1 0.0640 0.622   0.200
```

```
# Taking cutoff value
p_cutoff <- 0.2

#Assign the prediction outcome using above cut-off value
test_set <- test_set %>% mutate(class_tree = case_when(
  test_tree < p_cutoff ~ 'Active',
  test_tree >= p_cutoff ~ 'Inactive'))

#Print confusion matrix
(cnf_mtrx <- test_set %>% count(class_tree, Status) %>% spread(Status, n))
```

```
## # A tibble: 2 x 3
##   class_tree Active Inactive
##   <chr>       <int>    <int>
## 1 Active       1887       74
## 2 Inactive      129      122
```

Kappa Metric

4.4 Performance against business target

RECAP: Model target as agreed with business

(1) Model should accurately predict 90% of actual attributes as high risk

(2) Precision (True positives & true negatives) should be greater than 85%

Lets calculate the performance against these measures.

4.4.1 Hit rate - inactive status

Total inactive employees in test set are 196 and correctly predicted as per confusion matrix are 122.

Inactive prediction accuracy rate for decision tree = 62.2% (target - 90%).

```
#count of actual active and inactive employees in test data
test_set %>% group_by(Status) %>% count()
```

```
## # A tibble: 2 x 2
## # Groups:   Status [2]
##   Status       n
##   <fct>    <int>
## 1 Active    2016
## 2 Inactive   196
```

4.4.2 Precision - True positives & true negatives should be > 80%

Our precision rate as calculated from test_set is 90.8% as compared to the target of 80%.

```
correct_predic = test_set %>% filter(class_tree == Status) %>% count()

total_n = test_set%>% count()
(Precision = correct_predic / total_n * 100)
```

```
##          n
## 1 90.82278
```

---

5 Summary

Overall, decision tree has done better as compared to logistic regression on AUC and Precision metric. The features used as important are different in tree (Except time.to.allocation.expiry) as compared to logistic regression. The tree has overachieved Precision metric but has fallen short by 27.8% from 90% Inactive prediction Hit rate target

The model though promising at initial look, may not be reliable for long term. In the tree terminal nodes, there are multiple nodes with $n < 25$. Thus, change in sample data can bring very high variation to the decision tree outcome. We can overcome this problem through data sampling and bagging techniques.