Predicting Attrition in Employees and Analyzing Machine Learning Classifiers

Gowtham Ayna Raveendran - gowtayna@indiana.edu

Ramya Nagarajan - ranagara@umail.iu.edu

April 27, 2017

Abstract

Classification is one of the major areas of applications for Machine Learning and Data mining. Data mining does not only help in assisting the domain experts with the knowledge from the data but also provides disruptive insights which helps people to produce unbelievable advancements. The Machine learning advancements help the domain experts in making better decisions in their day to day life. In this project, we are trying to address one such problem of predicting attrition among employees of the company. Attrition in an employee causes reduction in productivity of the employee and impacts on the relationship between the employees. Hence, this becomes an important task to be detected and addresses for a company to retain its employees. We wanted to employ Machine Learning to help the management in predicting whether the employee exhibits attrition or not. Our choice of model was focused towards tree based models. We built a single decision tree model and a random forest model for the same task and compared their performance.

1 Introduction

Attrition is defined as the process of gradually reducing the strength and effectiveness of someone or something through sustained attack or pressure[1]. Any management should take care of their employees in the same way as they would like to take care of their clients. Employees who feel ignored at the workplace seldom contribute to the success of the organization. Analyzing the presence of attrition among employees helps the company to understand more about the mentality of the workforce and to decide on the steps to take to increase their productivity which helps not only the organization, but also gives positive impact on the mentality of the employees.

In this project, we are going to predict the presence of attribution among the employees. For this task, we will be using three different machine learning algorithms individually and finally we will bag them together and use the bagged voting classifier as our fourth classifier. The idea behind this project is to see which algorithm works better and also to understand the effect of bagging different classifiers. We will use the following algorithms for classification - Decision tree, Naive Bayes and Logistic Regression. We will analyze the performance of these individual classifiers and use them to create a voting classifier.

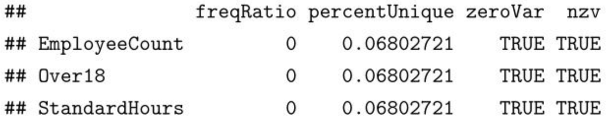
2 Data

The data that we took for this project is from Kaggle. This is a fictional data set created by IBM Data Scientists. The data set has 35 features which is a combination of binary, categorical and numerical features. The dataset has 1470 data points in total. More infromation about this data set can be found at https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

3 Initial Analysis of the Data

3.1 Dimensionality Reduction

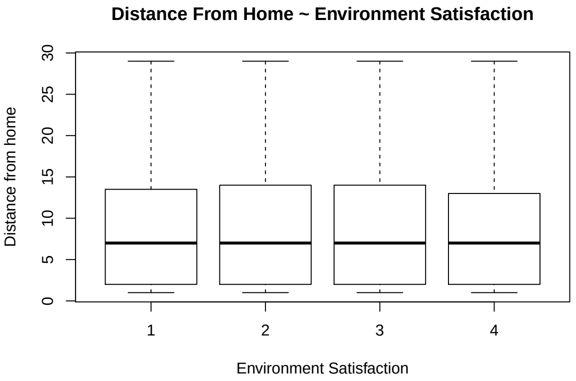
Since the data set had large number of features, we set out initially to apply some domain knowledge to remove unnecessary features that does not provide any additional information and the features that are not relevant to our analysis. On first look, we could see that features such as `EmployeeCount`, `Over18` and `StandardHours` does not provide any valuable information with respect to our prediction task. Hence, we removed these features from our dataset to reduce the number of dimensions. These variables had zero variance which gives zero predictive power for our prediction task.



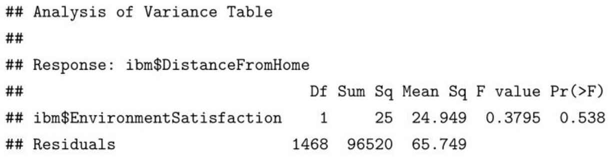
```
##                freqRatio percentUnique zeroVar  nzv
## EmployeeCount          0    0.06802721    TRUE TRUE
## Over18                 0    0.06802721    TRUE TRUE
## StandardHours          0    0.06802721    TRUE TRUE
```

Figure 1: Variables showing near zero variance (zero predictive power)

Next, we focussed on finding the features that are having maximum correlation. We applied our domain knowledge here to get the idea. We checked the relationship between variables `DistanceFromHome` and `EnvironmentSatisfaction` with hope that as the distance from home increases the employee satisfaction decreases. The first variable is a continuous numeric variable while the second one is categorical variable ranging from 1-4. We used ANOVA test and box plot to check whether the groups differ from each other. The results are as seen below.



From the boxplot, our interpretation was that the groups does not differ much. To confirm the same, we went ahead using ANOVA test to compare the means of the 4 groups.



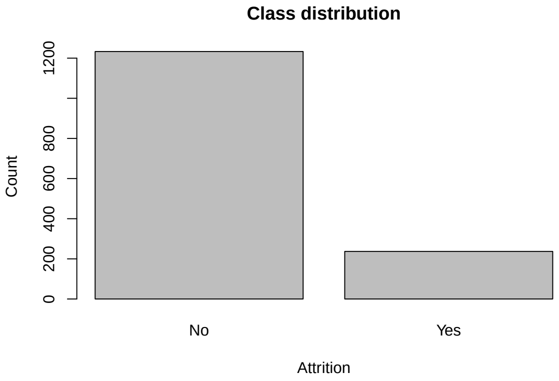
```
## Analysis of Variance Table
##
## Response: ibm$DistanceFromHome
##                                Df Sum Sq Mean Sq F value Pr(>F)
## ibm$EnvironmentSatisfaction     1     25  24.949  0.3795  0.538
## Residuals                    1468  96520  65.749
```

Figure 2: ANOVA Results

Since the p-value of the ANOVA test is high, we could confirm that the continuous variable `DistanceFromHome` does not provide additional information than `EnvironmentSatisfaction`. Hence, we removed the continuous variable and kept only the categorical variable `EnvironmentSatisfaction` in our dataset.

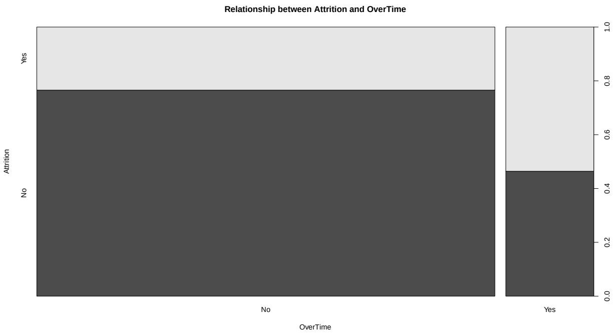
3.2 Class distribution

The data set exhibits skew in the distribution of the class labels. Out of the 1470 data points, there are only 16% of the data that belong to class `Yes` and rest 84% of the data belong to class `No`. Because of this, we chose the train and test split data in an efficient way so that the train data contains equal number of positive and negative class data points. We have tested the performance of the classifiers with both random sampling and undersampling the most frequent class from the data.

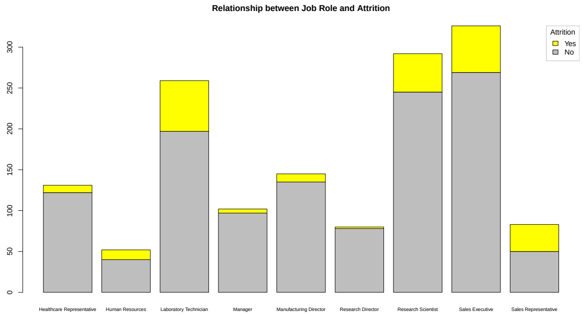


3.3 Variable relationship

Before going ahead with building predictive models for classifying an employee, we performed some exploratory data analysis on the data to get information about the relationship between the predictors. We first checked the relationship between the employees who work overtime and attrition. Our hypothesis was that people who work overtime will exhibit more attrition. Below plot explains the relationship between these two variables. From the plot, we could infer that out of the people who work overtime, around 55% of them exhibit attrition while only around 12% of the people in the other group exhibit attrition.

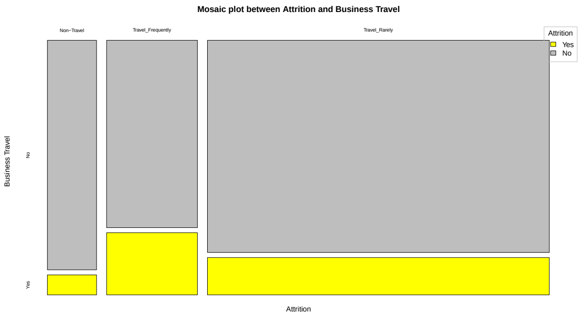


Next, we checked the relationship between the variables Attrition and Job Role.



A majority of the people who work as **Laboratory Technician** exhibit attrition. But the major role that needs focus is the **Sales Representative**. Almost close to half of the people working as Sales Representative exhibit attrition. It is obvious from the plot that attrition is more with employees who work at lower level of jobs i.e., Laboratory Technician, Sales executive and Sales representative. One surprising information from the data for us is that, number of research scientists exhibit attrition. Research scientists may tend to exhibit attrition because of factors like managerial pressure to perform better and come up with results too soon. Also, in some cases, the research which they set out to do may not have fruitful results.

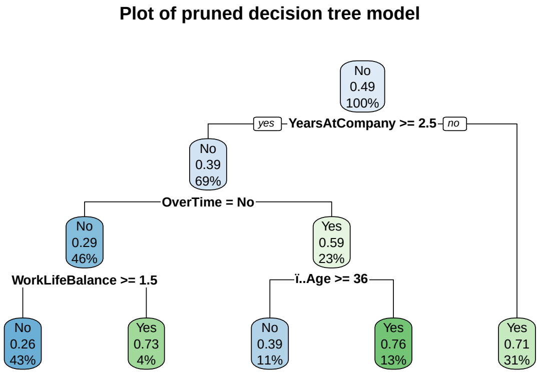
We also checked the relationship between Attrition and BusinessTravel. As we thought, the people who travel a lot are exhibiting more attrition among the three groups. The plot is shown below.



4 Classifiers

4.1 Decision Tree

Decision tree is one of the off the shelf classifier where the algorithm does not require any preprocessing of the data (in most of the cases). There is no need of dimensionality reduction as the model has inbuilt feature selection. This is advantageous because even though when the data set has very large number of features, the algorithm only uses a subset of them which has more predictive power based on information gain. Here, we used rpart package to build the tree model. Two approaches were considered here. First approach used random sampling to generate training data where the class imbalance was not considered. In second approach, we used undersampling to maintain the ratio between the classes around 50%. Based on our results, we could see that under sampling drastically improves the performance of the classifier. We chose the value of the complexity parameter as 0.03 based on the results from the tuning set. The results were pretty decent and the plot of the decision tree is shown below.



4.1.1 Model Interpretation

Models tells us that out of the all the employees in our training data, Attrition is prevalent among early/fresh employees. Further we could see that, employees over age 36 who work overtime also exhibit attrition. Another category is for the employees who do not have good work life balance exhibit attrition. All these decision splits from the tree agree with how things work in real world. The performance of the model on the test data is shown below.

```
##            Reference
## Prediction  No Yes
##        No  865 368
##        Yes  76 161
```

```
## Pos Pred Value Neg Pred Value
##      0.7015410      0.6793249
```

From the above values we can see that the accuracy is around 70% in both the classes. We wanted to improve on the performance of the classifier using ensemble methods.

There is a tradeoff that we need to consider. *If the organization's primary aim is to retain employees then our assumption is that the company doesn't mind spending money on its employees. Therefore, we train our classifier to give more importance to the negative class(Attrition = Yes) over the positive class (Attrition = No).*

4.2 Random Forest

Ensemble methods work on the basis of Wisdom of Crowds. Bagging is one such method in which we build multiple independent classifiers and aggregate the results to make better predictions. For classification, ensemble methods make their final predictions as the majority vote from the individual classifiers.

Random forest is an example of bagged classifier. Here, the individual classifiers are decision trees. Random forests overcome the problem of correlation between the individual classifiers by choosing random subset of features at each split of the decision tree. This makes sure that the individual classifiers are independent.

4.2.1 RF Model & Performance

We built random forest model using package `randomForest`. The tweakable parameters for the random forest model are `mtry` - indicating the number of features that are sampled at every split of the decision tree, `ntree` - represents the number of individual classifiers (trees) to build, `classwt` - allows us to specify the class weight differently for the classes (gives us option to provide more importance to one class over other), `nodesize` - indicates the minimum number of samples at leaf nodes. We chose the `classwt` parameter as a vector with values 1 and 1.25 i.e., we gave **25%** more weightage to the negative class (Attrition = Yes). This informs the classifier that for every mistake that it makes in the negative class, we would penalize it 1.25 times more. This type of class weighting penalizes the classifier more on one class thereby giving preference to one of the classes. We chose the value of `mtry` by using `tuneRF()` on the training set. This gave us a plot of the mtry values with the error rate as shown below.

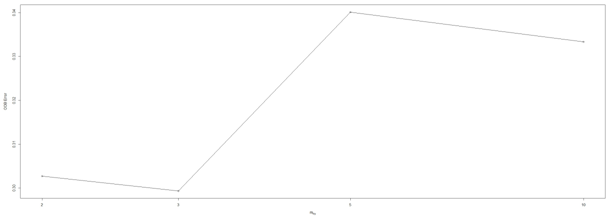


Figure 3: Plot of Error vs Number of features

From this plot, we saw that the value of error for `mtry = 3` was very low and hence we chose this value for building our model by considering this as our optimal choice.

4.2.2 Output of the model

The performance of the Random forest model was better when compared with a single decision tree classifier. Since more weight is given to the class of people who exhibit attrition, our random forest classifier predicted the negative class (Attrition = Yes) with around 80% accuracy with a slight reduction in the accuracy for the other class.

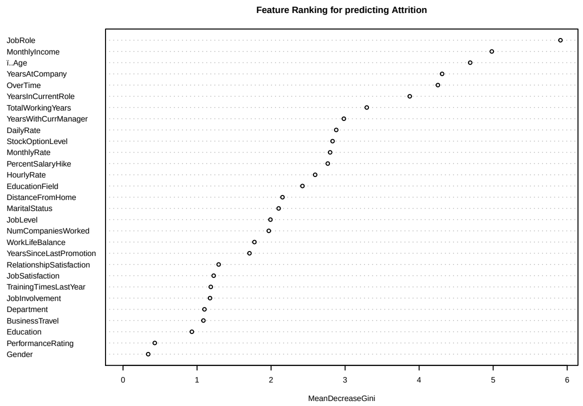
The results are shown below.

```
##           Reference
## Prediction  No Yes
##        No  724 359
##        Yes  19  74
```

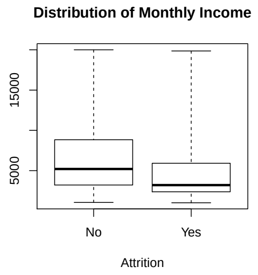
```
## Pos Pred Value Neg Pred Value
##      0.6685134      0.7956989
```

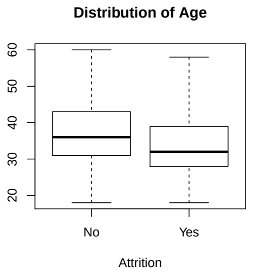
There is an improvement in the performance of the classifier using RF as compared to Decision tree.

The important features used for our prediction is plotted below.



From the variable importance plot, we could see that the features `Job Role` is the most important feature as we discussed before in the pevious sections. The other features that came out to be important are `Age` and `MonthlyIncome`. We checked the distribution of both of these variables with attrition.





We could see that features `Monthly Income` and `Age` does vary across people who exhibit attrition and our random forest model have used those variables as the predictors conforming with the data.

5 Conclusion

Based on the analysis that we performed above, we could see that ensemble methods have performed better than the single decision tree. The improvement in the random forest is because of three parameters - bagging, cost based learning and under sampling. Initially, to handle the class imbalance, we chose the train data to have almost equal proportion of positive and negative class samples. We used cost based learning (class weighting) to make the classifier to give more preference to the negative class (Attrition = Yes) over the other class. And finally, as the random forest model has more number of individual classifier, the predictive power has increased by 10% compared to the single decision tree model. Decision tree had the advantage of feature selection which helped the model to learn only from the important features while not using redundant or irrelevant features.

Since more weight is given to the class of people who exhibit attrition, our random forest classifier predicted the negative class (Attrition = Yes) with around 80% accuracy while slight reduction in the accuracy for the other class.

Acknowledgements

We would like to thank Prof.Christopher Raphael for equipping us with solid foundation in concepts related to Data Mining and Machine Learning. We would also like to thank the AIs for their excellent support in clearing our doubts and making our learning experience better.

REFERENCES

[1] Dataset Information - https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

[2] Decision tree (rpart) - https://cran.r-project.org/web/packages/rpart/rpart.pdf

[3] Random forest package - https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[4] Random forest implementation in `randomForest` - https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm