Employee Attrition Control

Team Name: Data Miners
Semester: Spring 2020




Dishti Dave
CS513-B
Computer Science
Graduate Student

Dyuti Dave
CS513-B
Computer Science
Graduate Student

Eman Alofi
CS513-B
Software Engineering
Graduate Student

Vaishnavi Gopalakrishnan
CS513-B
Computer Science
Graduate Student

Instructor : Prof. Khasha Dehnad

Overview

What is Attrition?

- Gradual loss of employees over time
- Leads to high cost for an organization
- Common expenses of losing employees and replacing them
  - ✓ Job postings
  - ✓ Hiring process
  - ✓ Paperwork, and
  - ✓ New hire training
- Additionally, regular employee turnover prohibits your organization from increasing its collective knowledge base &experience over time.
- Customers often prefer to interact with familiar people.
- Errors and issues are more likely if you constantly have new workers.

Problem description

Goal

- To show that companies can recognize the employees that are going to quit
- Help the company to make them stay
- Provide factors to the company with which they can improve the employees' satisfaction

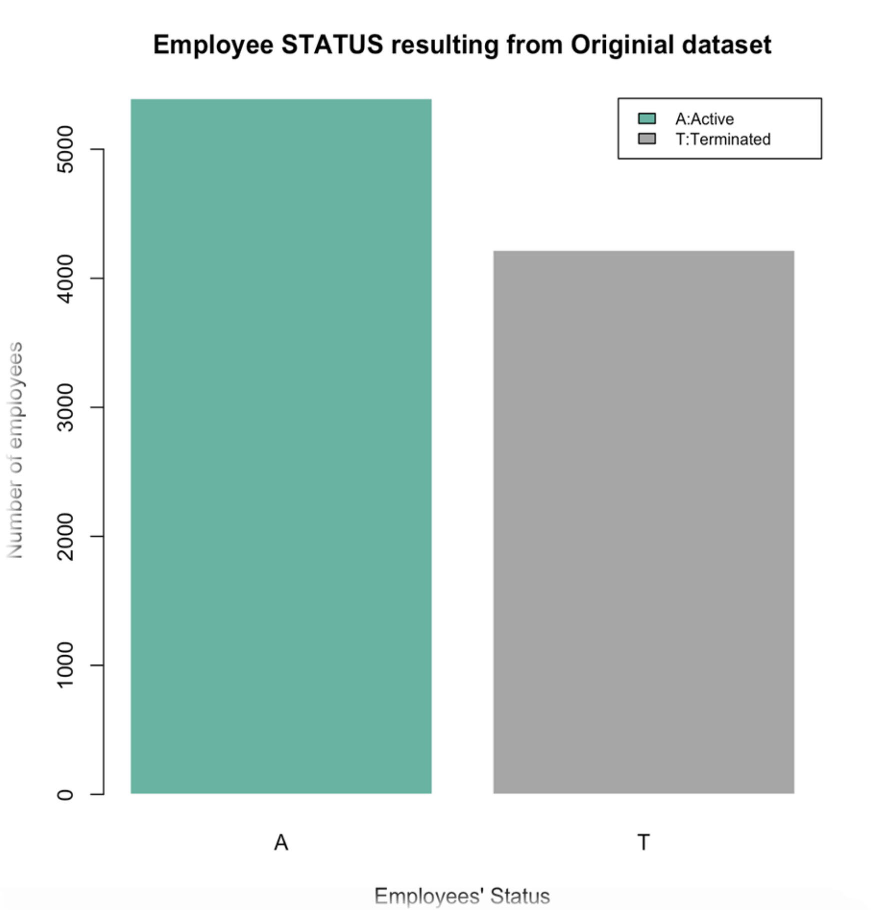
Problem description

Summary

- Predict how many people from the company would quit - Status
- Factors considered
  - ✓ Annual Rate
  - ✓ Hourly Rate
  - ✓ Job code
  - ✓ Job satisfaction score
  - ✓ Age
  - ✓ Performance rating
- Performed data preparation, data analysis & data modelling.
- Developed machine learning models to predict the attrition.
- Only 56.12% of total employees are active in the original dataset.

Analysis of Dataset

Uni Variant Analysis – Bar Graph



- Depicts the status of employees from the Original dataset

A	T
5394	4218

  - **Active – 5394 / 9612 = 56.12%**
  - **Terminated = 4218 / 9612 = 43.88%**

Analysis of Dataset

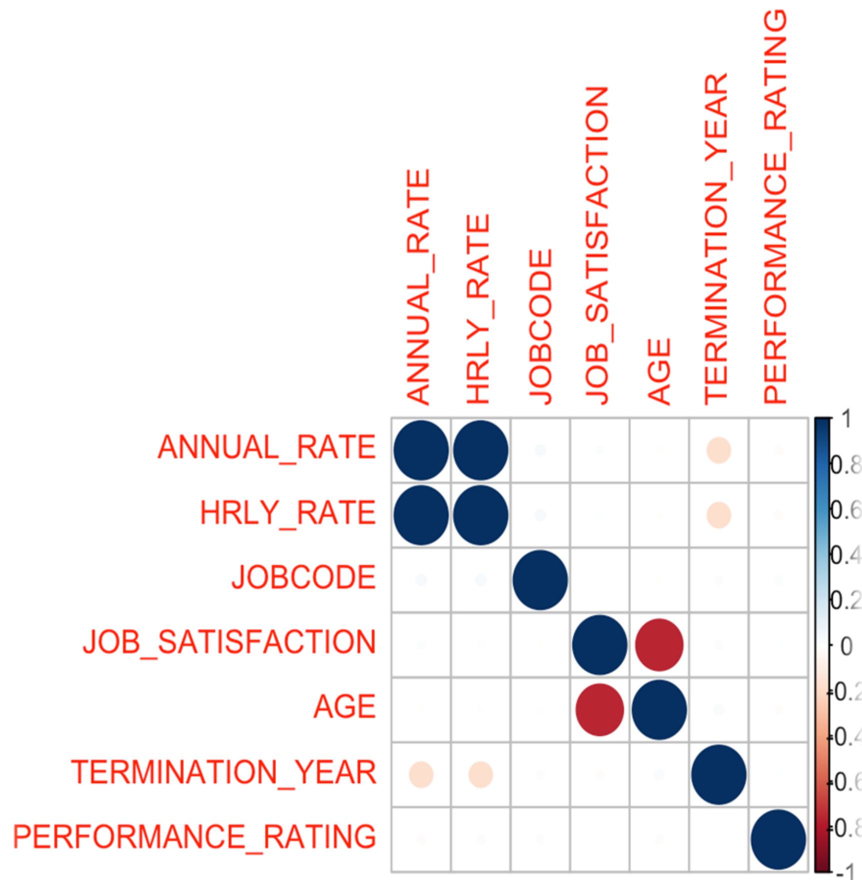
Multi Variant Analysis - Correlation

- Depicts correlation between various factors considered.

  > Annual rate & Hourly Rate

  > Job Satisfaction & Age

- The above factors correlates more with each other



Approaches / Techniques Used

Supervised Learning Techniques

1. a) kNN - Eman
2. b) kkNN - Vaishnavi
3. Naïve Bayes - Dishti
4. Decision Tree - Eman
5. Linear Regression - Vaishnavi
6. SVM - Dyuti
7. ANN – Eman & Vaishnavi
8. Random Forest - Dishti
9. C5_0 - Dishti

Unsupervised Learning Techniques

- K-means - Dyuti
- H-clustering - Dyuti

Solution

- Read the data set
- Analysed the data for null values
- Converted data based on the algorithm requirement
- Normalized the data for certain algorithms
- Applying supervised & unsupervised learning algorithms
- Train the data
- Predict the status of the employees
- Created visualization for some models

Supervised Learning

What?

- Concept of function approximation
- Model relationships and dependencies between the target prediction output and the input features
- We can predict the output values for new data based on those relationships
- Predictive model
- Have labelled data
- Deals with Regression & Classification problems

Supervised Learning

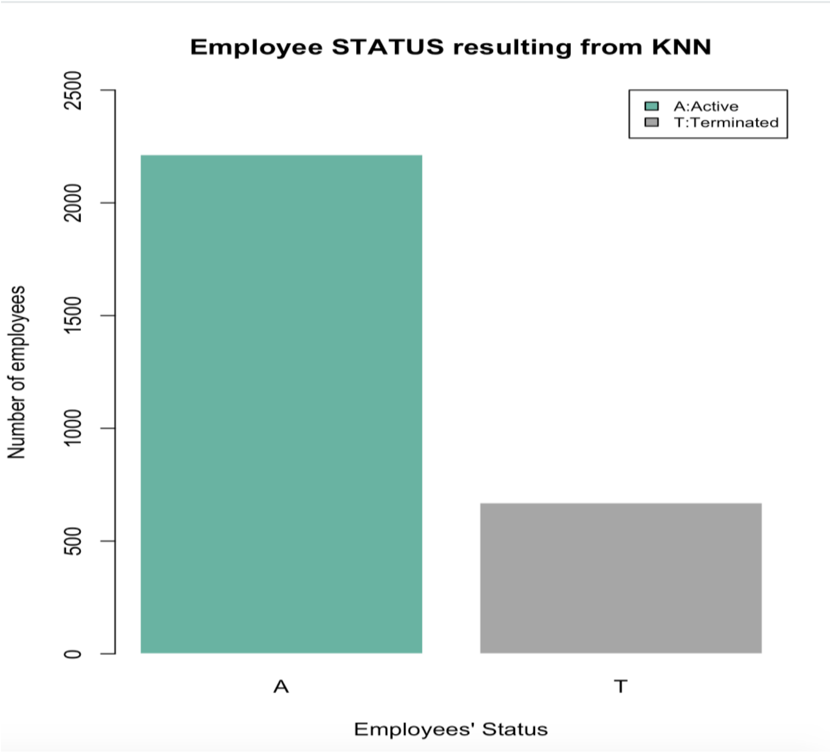
kNN - k-*Nearest Neighbor*

- The data points are predicted based on how its nearest neighbor data are classified.
- The target variable is the "status" of the employee based on the most correlated features which are :
  - ➢ Annual rate & Hourly Rate
  - ➢ Job Satisfaction & Age
- ➢ The value of K is 98 which is the square root of number of rows in the dataset.
  - ➢ 9612^(0.5) = 98.04
- ➢ Accuracy ~ **57%**

```r
normalize <- function(x) {
      return ((x - min(x)) / (max(x) - min(x))) }


# normalize the correlated features
df_n <- as.data.frame(lapply(df[,c(2,3,8,9)], normalize))
```

```r
idx<-sort(sample(nrow(df),as.integer(.70*nrow(df)))) # train 70% of the data
training <- df_n[idx, ] # train 70% of the data
testing <- df_n[-idx, ] # test 30% of the data

train_label <- df[idx, 21] # The target lable for  training dataset is 'STATUS' which is in column #21
test_label <- df[-idx, 21] # The target lable for  testing dataset is 'STATUS' which is in column #21
??knn

# Apply the 'knn'for the train_label which is 'STATUS'
STATUS_test_pred <- knn(train = training,test = testing, cl= train_label,k= 98)
```

Visualization of predicted kNN




```
install.packages("gmodels")
require("gmodels")
library("gmodels")
table <- CrossTable(x = test_label, y = STATUS_test_pred,
                    prop.chisq = FALSE)
# Caculate the accuracy by  posititve (TP), true negative (TN), false negative (FN) and false positive (FP).

tp <- table$t[1,1]

tn <- table$t[2,2]

fp <- table$t[1,2]

fn <- table$t[2,1]
```

- Depicts the status of employees from kNN prediction
- **Total data tested = 2884 (30% of original dataset)**
- **Active employees = 76%**
- **Terminated employees = 24%**

```
STATUS_test_pred
   A    T
2214  670
```

Supervised Learning

kkNN

```
# Train & Predict for testing - unweighted
STATUS_test_pred_k <- kknn(formula=target~.,
                           training, testing,
                           k=98, kernel ="rectangular")
fit <- fitted(STATUS_test_pred_k)

predict_k <- table(testing$STATUS,fit)
```

```
# Train & Predict for testing - weighted
STATUS_test_pred_kw <- kknn(formula=target~.,
                            training, testing,
                            k=98, kernel ="triangular")
fitw <- fitted(STATUS_test_pred_kw)

predict_kw <- table(testing$STATUS,fitw)
```

```
# Find error rate
wrong <- sum(testing[,21]!=fitw)
error_rate2 <- wrong/length(testing$STATUS)
error_rate2

# Accuracy for test
Accuracy2 <- 1-error_rate2
Accuracy2
```

- Accuracy for
  - Unweighted kkNN ~ **99.65%**
  - Weighted kkNN ~ **99.69%**

Supervised Learning

Naïve Bayes

- A Naive Bayes classifier considers each of these features to contribute independently to the prediction, regardless of any correlations between features.
- Accuracy ~ **97.42%**
- It requires less training data and when assumption of independence hold it performs better compared to other models.
- Perform well for categorical input

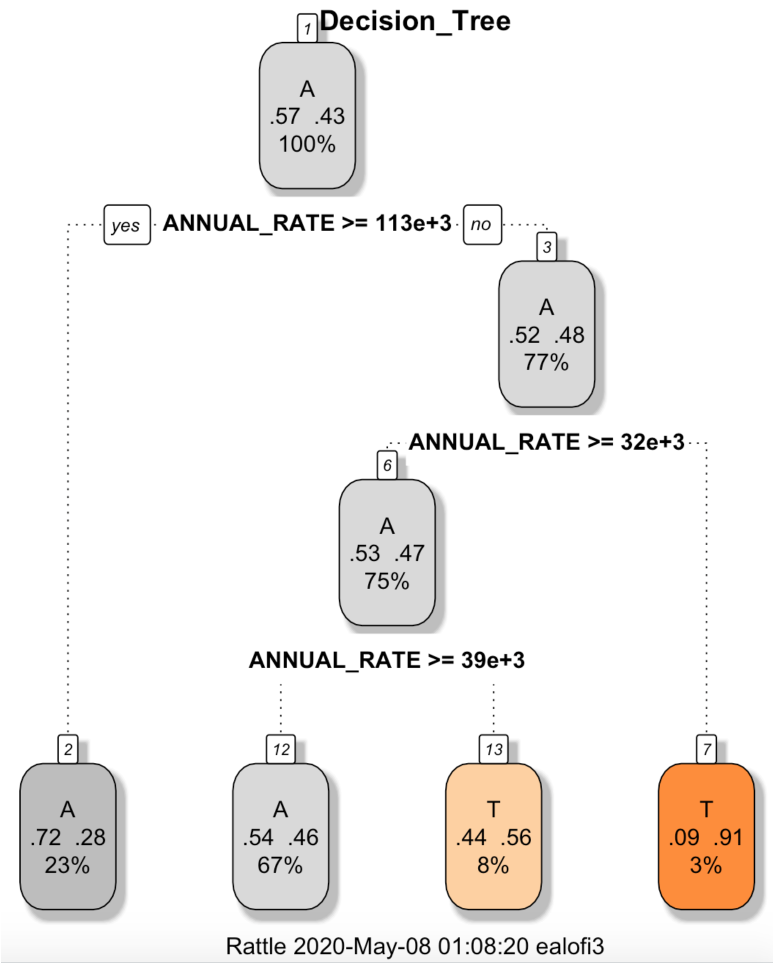
```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        0         1
0.4385405 0.5614595
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  994    0
         1   62 1347

               Accuracy : 0.9742
                 95% CI : (0.967, 0.9802)
    No Information Rate : 0.5605
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9473

 Mcnemar's Test P-Value : 9.408e-15

            Sensitivity : 0.9413
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9560
             Prevalence : 0.4395
         Detection Rate : 0.4136
   Detection Prevalence : 0.4136
      Balanced Accuracy : 0.9706

       'Positive' Class : 0
```

Supervised Learning

Decision Trees

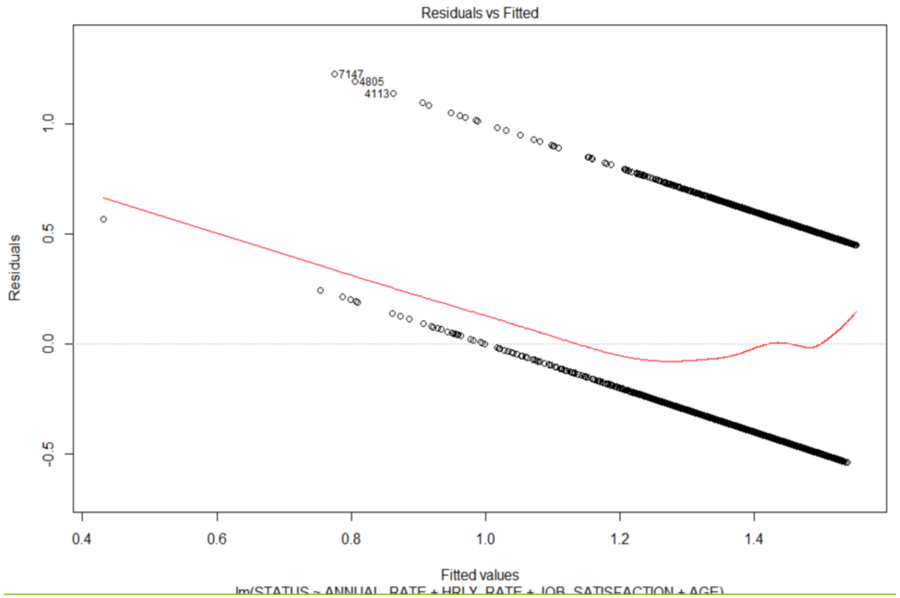
- Classification And Regression Tree
- Tree that represents choices and their results.
- Nodes are choices
- Edges are decisions
- Accuracy ~ **59.1%**



Supervised Learning

Linear Regression

- Predict values for continuous variables based on one or more predictor variables
- Plots are made from the residuals
- Accuracy ~ **96.5%**



Supervised Learning

Support Vector Machines (SVM)

- SVM is a discriminative classifier that takes labelled training data and constructs a hyperplane to categorize new examples.
- It looks at the interaction between future points.
- The idea behind SVM is to optimize the decision boundary that separate classes for prediction
- Accuracy ~ **52%**

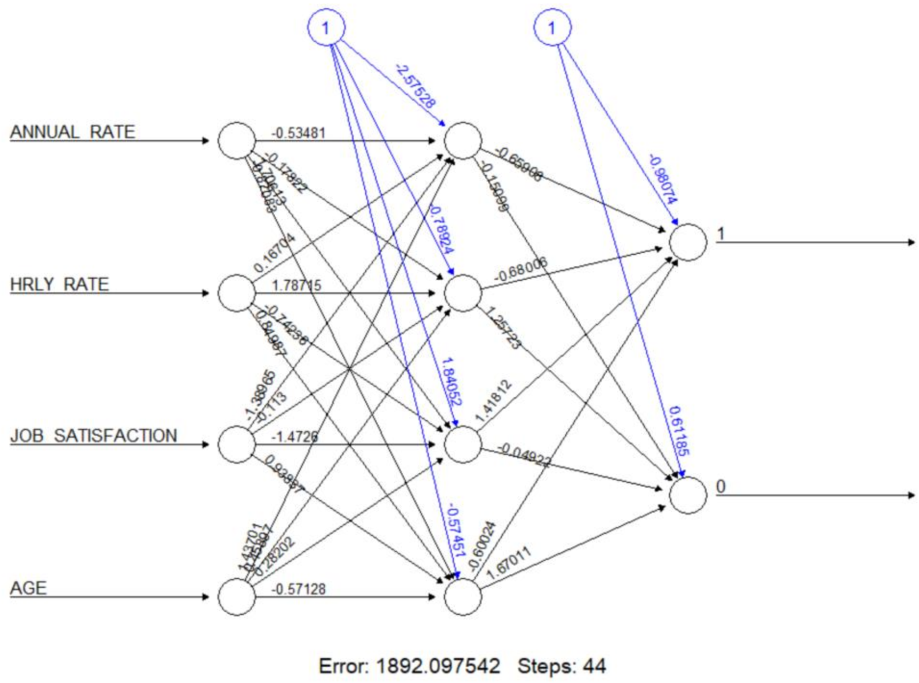
Code Snippet for EDA of Sample

```
dataset$STATUS <- factor(dataset$STATUS, levels =
dataset$ETHNICITY <- factor(dataset$ETHNICITY, levels = c("BLACK","ASIAN","WHITE","HISPA","PACIF","TWO","AMIND","Unknown"),labels = c("1",
dataset$SEX <- factor(dataset$SEX, levels = c("M","F"),labels = c("0", "1"))
dataset$MARITAL_STATUS <- factor(dataset$MARITAL_STATUS, levels = c("Single","Divorced","Married"),labels = c("0", "1","2"))
dataset$REFERRAL_SOURCE <- sub("^$", "Unknown", dataset$REFERRAL_SOURCE)
dataset$TERMINATION_YEAR[is.na(dataset$TERMINATION_YEAR)]= "2030"
dataset$IS_FIRST_JOB <- factor(dataset$IS_FIRST_JOB, levels = c("Y","N"),labels = c("0", "1"))
dataset$TRAVELLED_REQUIRED <- factor(dataset$TRAVELLED_REQUIRED, levels = c("Y","N"),labels = c("0", "1"))
dataset$REHIRE <- factor(dataset$REHIRE, levels = c("TRUE","FALSE"),labels = c("0", "1"))
dataset$DISABLED_EMP <- factor(dataset$DISABLED_EMP, levels = c("Y","N"),labels = c("0", "1"))
dataset$DISABLED_VET <- factor(dataset$DISABLED_VET, levels = c("Y","N"),labels = c("0", "1"))
dataset$EDUCATION_LEVEL <- factor(dataset$EDUCATION_LEVEL, levels = c("LEVEL 1","LEVEL 2","LEVEL 3","LEVEL 4","LEVEL 5"),labels = c("1", "
```

Supervised Learning

Artificial Neural Net

- Stimulate the behaviour of biological system composed of neurons
- Inspired by animal's central nervous system
- Accuracy ~ **50%**

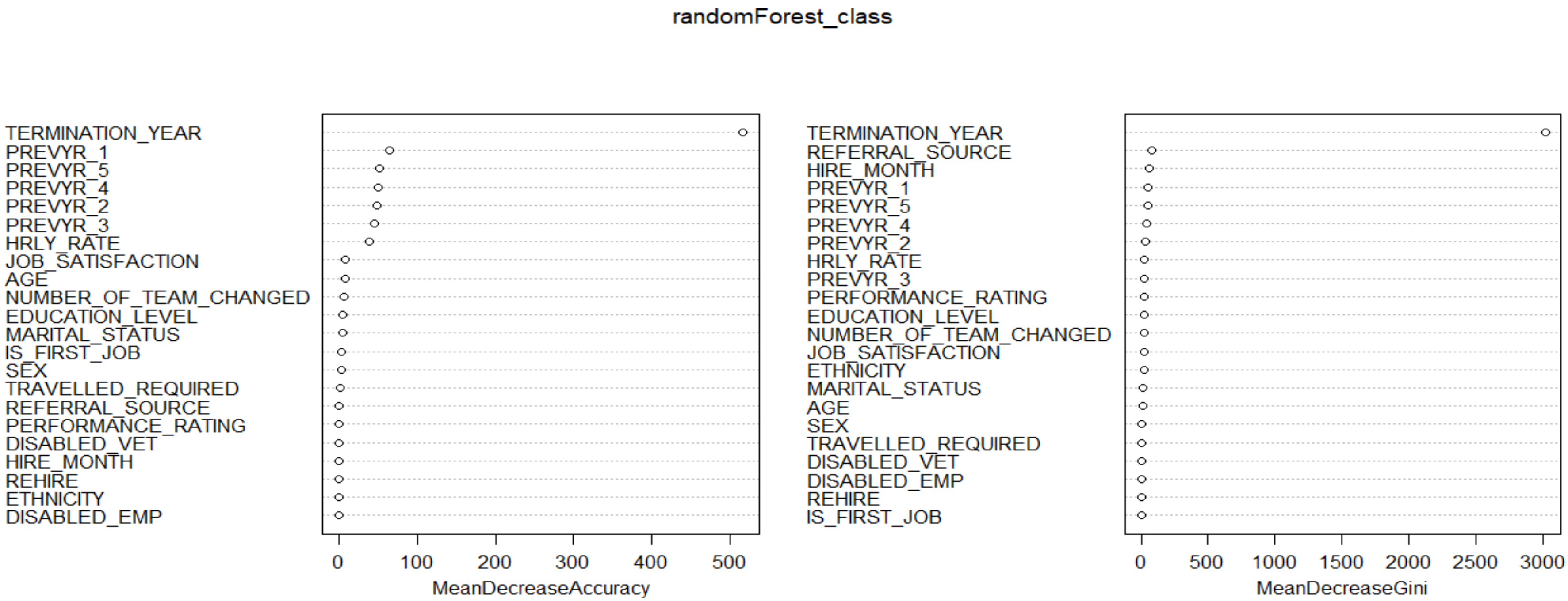


Supervised Learning

Random Forest Feature Importance graph

- It improves on bagging because it decorrelates the trees with the introduction of splitting on a random subset of features.

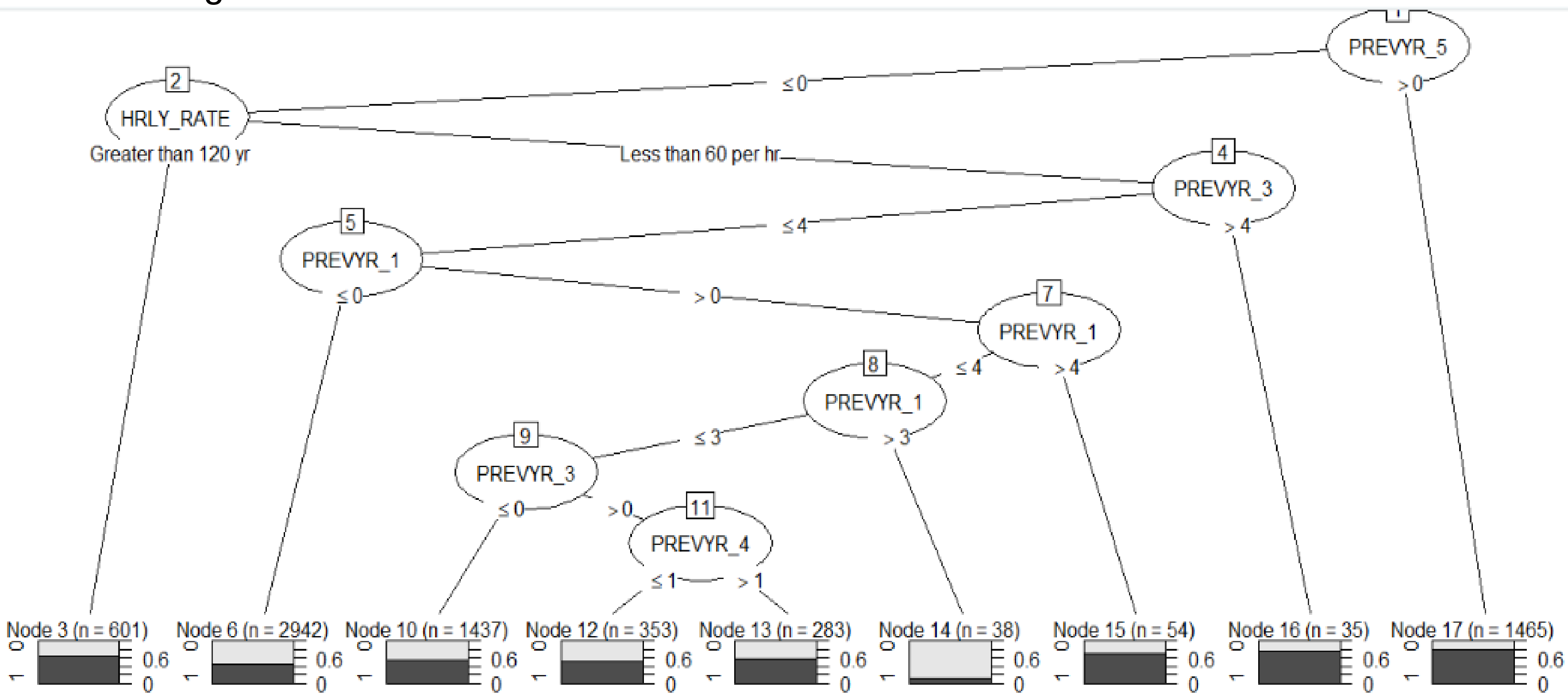
- "RANDOM" because each tree is only trained on a random subset of samples drawn from the training set.
  Accuracy ~ **99%**



Supervised Learning

C5.0

- It works by splitting the sample based on the field that provides the maximum information game.
- C5.0 gives a binary tree or multi branch tree. Accuracy ~ **63.25%**



Unsupervised Learning

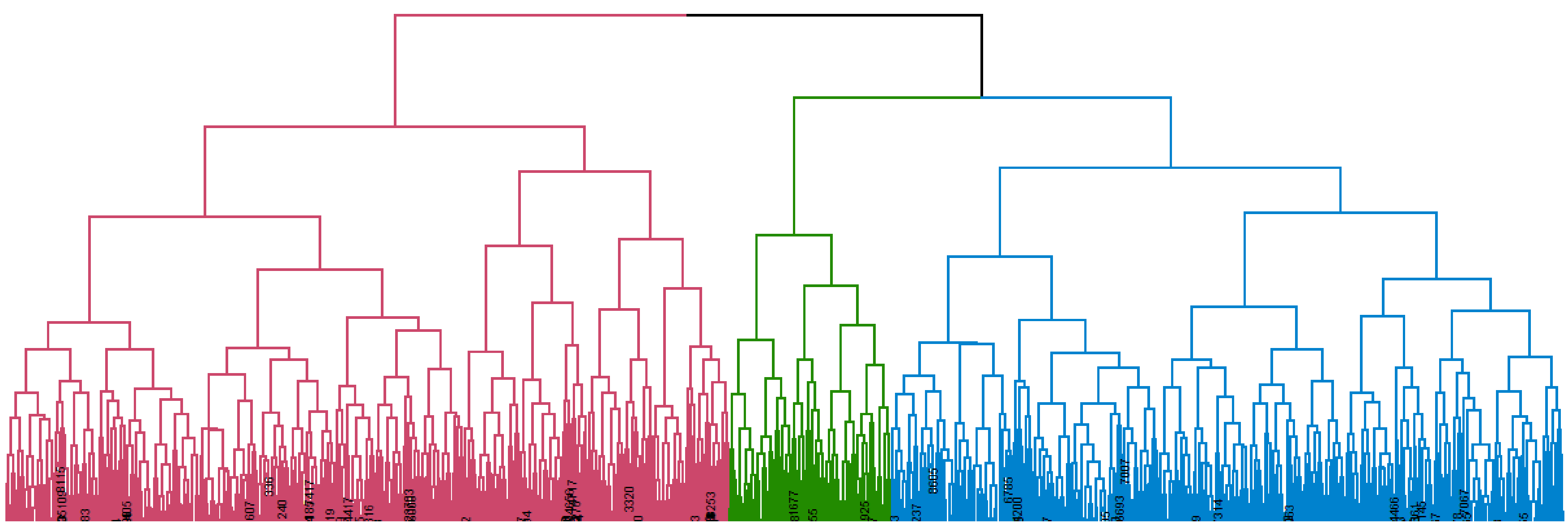
What?

- Concept of pattern detection
- Mine the data for rules, detect patterns, summarize & group the data points
- Trained with unlabelled data
- Descriptive model
- Deals with Clustering & Association rule learning

Unsupervised Learning

h-clustering Dendrogram for cut = 3 obtaining 3 clusters of our dataset solution

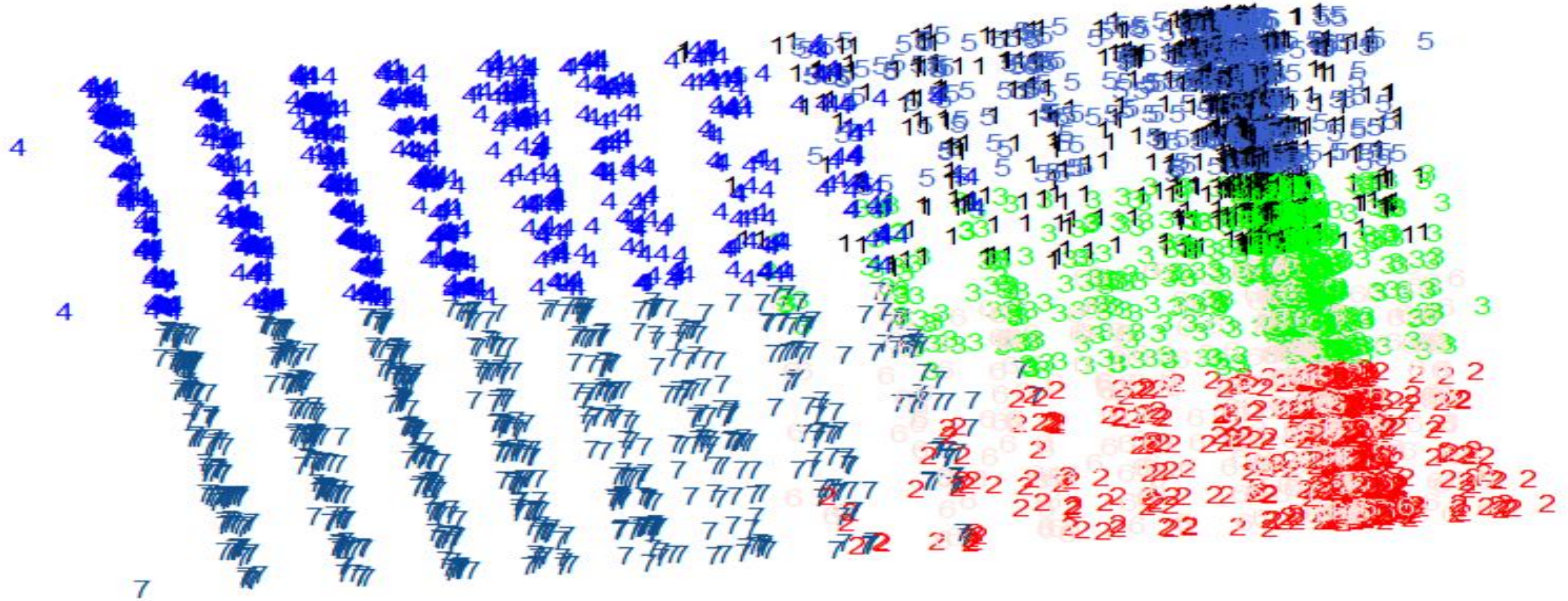
- H-clustering has advantage that any valid measure of distance can be used.
- It uses a powerful technique to build tree structures from data similarity
- It only uses matrix of distances which implies that the observation themselves are not required.
- Accuracy ~ **56%**



Unsupervised learning

k-means cluster plot for k=7

- Used when you have unlabelled data. It becomes a great solution for pre-clustering, reducing the space into disjoint smaller sub-spaces where other clustering algorithms can be applied
- K identifies the number of centroids and then allocates every data point to the nearest cluster.
- Accuracy **48%**



Recommendations

Based on predictions & visualizations

Most of all we found factors which are most important to employees and if they are not fulfilled, it might lead to Attrition.

Based on the predictions, the company must be able to figure out the affecting factors and correct them to Control attrition.

Conclusion & Future Work

- **Conclusion**
- Machine learning models are as good as the data you feed, and more data would strengthen the model.
- While some level of attrition is inevitable it should be kept at the minimum possible level using above solution.
- Based the most important features to the least important features we can identify what are the main causes for attrition.
- It helps to understand the key variable that influence the turnover.
- We have considered every selective group of features to identify what works best for our dataset and analysed our model solution

- **Future Work**
- New machine learning techniques can be applied to business application and especially predictive analytics.
- In this case we can use H2O/LIME to develop and explain sophisticated models that very accurately detect employees that are at risk of turnover.
- Or advanced machine learning models and selective features can help improve our predictive analysis. Using ensemble model to breakdown complex structure into critical features that most related to attrition

stevens.edu

GitHub Repository:

https://github.com/vaishnavimecit/CS513Spring20_DataMiners

Thank You