**Case Study 02**

**Executive Summary**

DDSAnalytics is an analytics company that specializes in talent management solutions for Fortune 1000 companies. Talent management is defined as the iterative process of developing and retaining employees. It may include workforce planning, employee training programs, identifying high-potential employees, and reducing/preventing voluntary employee turnover (attrition). To gain a competitive edge over its competition, DDSAnalytics is planning to leverage data science for talent management. The executive leadership has identified predicting employee turnover as its first application of data science for talent management. Before the business green lights, the project, they have tasked your data science team to conduct an analysis of existing employee data.

Our team has been given a dataset (CaseStudy2-Data.csv) to conduct exploratory data analysis (EDA) to determine factors that lead to attrition. Our goal is to identify (at least) the top three factors that contribute to turnover. Additionally, the business is also interested in learning about any job role specific trends that may exist in the data set.

We are looking to understand the cause for attrition at DDS Analytics in hopes of retaining talent.

Additionally, salary is hypothesized as one of the main factors in employee attrition, and as such DDS Analytics would like to

* Validate or refute this hypothesis
* See if salary can be predicted from the other employee attributes

Codes

* ## Research & Development department has most attrition followed by Sales
* CS %>% group\_by(Attrition)%>%count(Department)%>% mutate(perc = n/nrow(CS)) -> CS\_perc
* CS\_perc%>% ggplot(aes(x=Department, y=perc))+ geom\_bar(stat="identity") + facet\_grid(~Attrition)
* CS %>% group\_by(Attrition)%>%count(Gender)%>%mutate(perc =n/nrow(CS)) -> CS\_G
* CS\_G%>% ggplot(aes(x = Gender, y = perc)) + geom\_bar(stat = "identity") + facet\_grid(~Attrition)
* ggplot(data = CS, aes(x = MonthlyIncome , y = Gender)) + geom\_boxplot() + coord\_flip()
* CS %>%group\_by(JobSatisfaction) %>% count(JobRole) %>% mutate(perc = n/nrow(CS)) ->CS\_job
* CS\_job %>% ggplot(aes(x =JobRole, y=perc)) + geom\_bar(stat = 'identity') + facet\_wrap(~JobSatisfaction)
* #T test
* CS %>% group\_by(Attrition) %>% count(MonthlyIncome)%>% mutate(perc = n/nrow(CS)) -> CS\_M
* CS %>% ggplot(aes(x = MonthlyIncome, y = Attrition)) + geom\_boxplot() + coord\_flip()
* #T\_Stat = (mean(data$difference)-0)/(sd(data$difference)/sqrt(length(data$difference)))
* t.test(MonthlyIncome ~ Attrition, data= CS, conf.level =0.95,var=F)
* ## between Age and Monthly Icome
* CS %>% ggplot(aes(x= Age, y = Attrition)) + geom\_boxplot() + coord\_flip()
* t.test(Age ~ Attrition, data = CS, conf.level = 0.95, var =F)
* ## Which role has more chance to leave the organization ( Research Scientist and Laboratory technician has more attrition rate)
* CS %>% filter(Department == "Research & Development") %>% group\_by(Attrition) %>% count(JobRole) %>% mutate(perc = n/nrow(CS))-> CS2\_perc
* CS2\_perc %>% ggplot(aes(x=JobRole, y=perc)) + geom\_bar(stat='identity') + facet\_grid(~Attrition)+coord\_flip()
* ggplot(data=CS2\_perc, aes(x = MonthlyIncome, y=JobRole)) + geom\_boxplot() + coord\_flip()
* CS %>% filter(Department == "Sales") ->CS3
* CS3 %>% group\_by(Attrition) %>% count(JobRole) %>% mutate(perc = n/nrow(CS))-> CS3\_perc
* CS3\_perc %>% ggplot(aes(x=JobRole, y=perc)) + geom\_bar(stat='identity') + facet\_grid(~Attrition)+coord\_flip()
* ggplot(data=CS3, aes(x = MonthlyIncome, y=JobRole)) + geom\_boxplot() + coord\_flip()
* CS %>% group\_by(Attrition) %>% count(JobLevel) %>% mutate(perc = n/nrow(CS))-> CS\_perc
* CS\_perc %>% ggplot(aes(x=JobLevel, y=perc)) + geom\_bar(stat='identity') + facet\_grid(~Attrition)+coord\_flip()
* boxplot(MonthlyIncome ~ JobLevel, data = CS, ylab = "Monthly Income")
* #ggplot(data=CS, aes(x = MonthlyIncome, y=JobLevel)) + geom\_boxplot() + coord\_flip()
* #CS %>% ggplot(aes(y=MonthlyIncome, x=JobLevel))+geom\_boxplot()
* ### How
* CS\_perc %>% ggplot(aes( x= Gender, y = perc)) + geom\_bar() + facet\_grid(~Attrition)
* ### Check with the marital Status
* CS %>% ggplot((aes(x= MaritalStatus)))+geom\_bar() + facet\_grid(~Attrition)
* ggplot(data = CS, aes( x= MonthlyIncome, y=MaritalStatus)) + geom\_boxplot() + coord\_flip()
* #### Business Travel department
* CS %>% ggplot((aes(x= BusinessTravel)))+geom\_bar() + facet\_grid(~Attrition)
* ggplot(data = CS, aes( x= MonthlyIncome, y=BusinessTravel)) + geom\_boxplot() + coord\_flip()
* ### Does JobLevel drive attrition ???
* # histogram of MonthlyRate
* hist(CS$MonthlyIncome, main = "Monthly Income")
* # histogram of MonthlyRate
* hist(CS$MonthlyRate, main = "Monthly Rate")
* # histogram of HourlyRate
* hist(CS$HourlyRate, main = "Hourly Rate")
* # histogram of DailyRate
* hist(CS$DailyRate, main = "Daily Rate")
* cor.test(CS$MonthlyRate, CS$HourlyRate)
* # Attrition vs HourlyRate
* boxplot(CS$HourlyRate ~ CS$Attrition,
* ylab = "Dollars",
* main = "Hourly Rate by Attrition")
* # Attrition vs DailyRate
* boxplot(CS$DailyRate ~ CS$Attrition,
* ylab = "Dollars",
* main = "Hourly Rate by Attrition")
* # Attrition vs MonthlyRate
* boxplot(CS$MonthlyRate ~ CS$Attrition,
* ylab = "Dollars",
* main = "Monthly Rate by Attrition")
* # Attrition vs MonthlyIncome
* boxplot(CS$MonthlyIncome ~ CS$Attrition,
* ylab = "Dollars",
* main = "Monthly Income by Attrition")
* # Pairwise scatterplot
* pairs(~ MonthlyIncome + HourlyRate + DailyRate + MonthlyRate,
* data = CS,
* main = "Scatterplot for All Monetary Parameters")
* CS %>% ggplot(aes(x=MonthlyIncome, y=Attrition)) + geom\_boxplot()+ coord\_flip()
* CS %>% ggplot(aes(x=MonthlyRate, y= Attrition)) + geom\_boxplot()+ facet\_wrap(~Gender) + coord\_flip()
* CS %>% ggplot(aes(x=Attrition, y=Age)) + geom\_boxplot()+ facet\_wrap(~Gender)
* CS %>% ggplot(aes(x=MaritalStatus)) + geom\_bar()+ facet\_wrap(~Attrition)
* CS %>% ggplot(aes(x=ID, y=Attrition)) + geom\_boxplot()+ coord\_flip()
* CS %>% ggplot(aes(x=Age, y=Attrition)) + geom\_boxplot()+ coord\_flip()
* is.na(CS)
* ggplot(CS, aes(x = EducationField)) +
* geom\_bar(aes(y = (..count..)/sum(..count..)))+facet\_grid(~Attrition)
* CS %>% ggplot(aes(x=EducationField)) + geom\_bar() + facet\_grid(~Attrition)
* summary(CS)
* table(data$Attrition)
* data <- CS
* #install.packages('ROSE')
* library(ROSE)
* data\_balanced\_over <- ovun.sample(Attrition ~ ., data = data, method = "over",p=0.5,seed=1)$data
* table(data\_balanced\_over$Attrition)
* head(data\_balanced\_over)
* data\_balanced\_over\_scaled <- data\_balanced\_over
* data\_balanced\_over\_scaled[,c("Age","DailyRate","DistanceFromHome","Education","EnvironmentSatisfaction","HourlyRate","JobInvolvement","JobLevel","JobSatisfaction","MonthlyIncome","MonthlyRate","NumCompaniesWorked","PercentSalaryHike","PerformanceRating","RelationshipSatisfaction","StockOptionLevel","TotalWorkingYears","TrainingTimesLastYear","WorkLifeBalance","YearsAtCompany","YearsInCurrentRole","YearsSinceLastPromotion","YearsWithCurrManager")] <- scale(data\_balanced\_over\_scaled[,c("Age","DailyRate","DistanceFromHome","Education","EnvironmentSatisfaction","HourlyRate","JobInvolvement","JobLevel","JobSatisfaction","MonthlyIncome","MonthlyRate","NumCompaniesWorked","PercentSalaryHike","PerformanceRating",
* "RelationshipSatisfaction","StockOptionLevel","TotalWorkingYears","TrainingTimesLastYear","WorkLifeBalance","YearsAtCompany","YearsInCurrentRole","YearsSinceLastPromotion","YearsWithCurrManager")])
* head(data\_balanced\_over\_scaled)
* data\_balanced\_over\_scaled$Attrition <- as.factor(data\_balanced\_over\_scaled$Attrition)
* data\_balanced\_over\_scaled$Gender <- as.factor(data\_balanced\_over\_scaled$Gender)
* data\_balanced\_over\_scaled$BusinessTravel <- as.factor(data\_balanced\_over\_scaled$BusinessTravel)
* data\_balanced\_over\_scaled$Over18 <- as.factor(data\_balanced\_over\_scaled$Over18)
* data\_balanced\_over\_scaled$OverTime <- as.factor(data\_balanced\_over\_scaled$OverTime)
* data\_balanced\_over\_scaled$MaritalStatus <- as.factor(data\_balanced\_over\_scaled$MaritalStatus)
* data\_balanced\_over\_scaled$Department <- as.factor(data\_balanced\_over\_scaled$Department)
* data\_balanced\_over\_scaled$EducationField <- as.factor(data\_balanced\_over\_scaled$EducationField)
* data\_balanced\_over\_scaled$JobRole <- as.factor(data\_balanced\_over\_scaled$JobRole)
* data\_balanced\_over\_scaled$Attrition <- ifelse(data\_balanced\_over\_scaled$Attrition =="No",0,1)
* data\_balanced\_over\_scaled$Gender <- ifelse(data\_balanced\_over\_scaled$Gender== "Male",1,0)
* data\_balanced\_over\_scaled$Over18 <- ifelse(data\_balanced\_over\_scaled$Over18=="No",0,1)
* data\_balanced\_over\_scaled$OverTime <- ifelse(data\_balanced\_over\_scaled$OverTime=="No",0,1)
* library(fastDummies)
* library(magrittr)
* library(dplyr)
* data\_balanced\_over\_scaled <- dummy\_cols(data\_balanced\_over\_scaled, select\_columns = c("BusinessTravel","Department","EducationField","JobRole","MaritalStatus"), remove\_first\_dummy = TRUE)
* ## Removing the column created
* head(data\_balanced\_over\_scaled)
* data\_balanced\_over\_scaled <- data\_balanced\_over\_scaled %>% select(-one\_of(c("BusinessTravel","Department","EducationField","JobRole","MaritalStatus","ID","StandardHours","EmployeeCount","EmployeeNumber")))
* Target <- data\_balanced\_over\_scaled %>% select(one\_of("Attrition"))
* Predictors <- data\_balanced\_over\_scaled %>% select(-one\_of("Attrition"))
* split\_perc = .7
* split\_index <- sample(dim(Predictors)[1], round(dim(Predictors)[1]\*split\_perc))
* Predictors\_Train <- Predictors[split\_index,]
* Predictors\_Test <- Predictors[-split\_index,]
* Target\_Train <- Target[split\_index,]
* Target\_Test <- Target[-split\_index,]
* library(caret)
* library(class)
* classifications <- knn(Predictors\_Train,Predictors\_Test , as.factor(Target\_Train),k=5,prob=T)
* table(classifications, as.factor(Target\_Test))
* cm <- confusionMatrix(classifications,as.factor(Target\_Test))
* ######
* ## Regression
* data\_balanced\_ove\_scaled2 <- data\_balanced\_over
* data\_balanced\_ove\_scaled2$Attrition <- as.factor(data\_balanced\_ove\_scaled2$Attrition)
* data\_balanced\_ove\_scaled2$Gender <- as.factor(data\_balanced\_ove\_scaled2$Gender)
* data\_balanced\_ove\_scaled2$BusinessTravel <- as.factor(data\_balanced\_ove\_scaled2$BusinessTravel)
* data\_balanced\_ove\_scaled2$Over18 <- as.factor(data\_balanced\_ove\_scaled2$Over18)
* data\_balanced\_ove\_scaled2$OverTime <- as.factor(data\_balanced\_ove\_scaled2$OverTime)
* data\_balanced\_ove\_scaled2$MaritalStatus <- as.factor(data\_balanced\_ove\_scaled2$MaritalStatus)
* data\_balanced\_ove\_scaled2$Department <- as.factor(data\_balanced\_ove\_scaled2$Department)
* data\_balanced\_ove\_scaled2$EducationField <- as.factor(data\_balanced\_ove\_scaled2$EducationField)
* data\_balanced\_ove\_scaled2$JobRole <- as.factor(data\_balanced\_ove\_scaled2$JobRole)
* data\_balanced\_ove\_scaled2$Attrition <- ifelse(data\_balanced\_ove\_scaled2$Attrition =="No",0,1)
* data\_balanced\_ove\_scaled2$Gender <- ifelse(data\_balanced\_ove\_scaled2$Gender== "Male",1,0)
* data\_balanced\_ove\_scaled2$Over18 <- ifelse(data\_balanced\_ove\_scaled2$Over18=="No",0,1)
* data\_balanced\_ove\_scaled2$OverTime <- ifelse(data\_balanced\_ove\_scaled2$OverTime=="No",0,1)
* data\_balanced\_ove\_scaled2 <- dummy\_cols(data\_balanced\_ove\_scaled2, select\_columns = c("BusinessTravel","Department","EducationField","JobRole","MaritalStatus"), remove\_first\_dummy = TRUE)
* data\_balanced\_ove\_scaled2[,c("Age","DailyRate","DistanceFromHome","Education","EnvironmentSatisfaction","HourlyRate","JobInvolvement","JobLevel",
* "JobSatisfaction","MonthlyRate","NumCompaniesWorked","PercentSalaryHike","PerformanceRating",
* "RelationshipSatisfaction","StockOptionLevel","TotalWorkingYears","TrainingTimesLastYear","WorkLifeBalance",
* "YearsAtCompany","YearsInCurrentRole","YearsSinceLastPromotion","YearsWithCurrManager")] <- scale(data\_balanced\_ove\_scaled2[,c("Age","DailyRate",
* "DistanceFromHome","Education",
* "EnvironmentSatisfaction","HourlyRate","JobInvolvement",
* "JobLevel","JobSatisfaction","MonthlyRate",
* "NumCompaniesWorked","PercentSalaryHike","PerformanceRating",
* "RelationshipSatisfaction","StockOptionLevel","TotalWorkingYears",
* "TrainingTimesLastYear","WorkLifeBalance","YearsAtCompany",
* "YearsInCurrentRole","YearsSinceLastPromotion","YearsWithCurrManager")])
* data\_balanced\_ove\_scaled2 <- data\_balanced\_ove\_scaled2 %>% select(-one\_of( c("BusinessTravel","Department","EducationField","JobRole","MaritalStatus","EmployeeCount","Over18","StandardHours")))
* head(data\_balanced\_ove\_scaled2)
* split\_perc = 0.7
* split\_index <- sample(dim(data\_balanced\_ove\_scaled2)[1], round(dim(data\_balanced\_ove\_scaled2)[1]\*split\_perc))
* MonthlyIncome\_LM\_train <- data\_balanced\_ove\_scaled2[split\_index,]
* MonthlyIncome\_LM\_Test <- data\_balanced\_ove\_scaled2[-split\_index,]
* linear\_model <- lm(MonthlyIncome~., data = MonthlyIncome\_LM\_train)
* summary(linear\_model)
* Target1 <- MonthlyIncome\_LM\_Test%>% select(one\_of("MonthlyIncome"))
* Predictors1 <- MonthlyIncome\_LM\_Test %>% select(-one\_of("MonthlyIncome"))
* predictions <- linear\_model %>% predict(MonthlyIncome\_LM\_Test)
* RMSE(predictions, MonthlyIncome\_LM\_Test$MonthlyIncome)
* # mjob\_pred\_caret <- train(Predictors\_Train, as.factor(Target\_Train), method = "knn", preProcess = c("center","scale"))
* #
* # mjob\_pred\_caret
* #
* #
* # plot(mjob\_pred\_caret)
* #
* #
* # knnPredict <- predict(mjob\_pred\_caret, newdata = Predictors\_Test)
* #
* # confusionMatrix(knnPredict, as.factor(Target\_Test))
* acc\_table <- data.frame(acc= numeric(100), k = numeric(100), sensitivity = numeric(100), specificity = numeric(100))
* for(i in 1:100){
* classification <- knn(Predictors\_Train,Predictors\_Test , as.factor(Target\_Train), k = i, prob=T)
* table(classification, as.factor(Target\_Test))
* conf\_matrix <- confusionMatrix(classification, as.factor(Target\_Test))
* acc\_table$acc[i] = conf\_matrix$overall[1]
* acc\_table$sensitivity[i] = conf\_matrix$byClass[1]
* acc\_table$specificity[i] = conf\_matrix$byClass[2]
* acc\_table$k[i] =i
* }
* acc\_table
* install.packages("naivebayes")
* library(e1071)
* set.seed(4)
* model <- naiveBayes(Predictors\_Train, as.factor(Target\_Train))
* classification <- predict(model,Predictors\_Test)
* confusionMatrix(table(classification,as.factor(Target\_Test)))