CaseStudyDocument

Lola Awodipe

12/7/2020

#load needed libraries library(plyr) library(tidyverse) library(dplyr) library(formattable) library(mice) library(lattice) library(pan) library(ggplot2) library(scales) library(ggthemes) library(ggplot2) library(class) library(caret) library(naniar) library(e1071) library(Hmisc) library(corrplot) library(caret) library(class) library(dplyr) library(e1071) library(FNN) library(gmodels) library(psych) library(corrplot) library(MASS) library(car)

#Import the dataset EData <- read.csv(file.choose(), header= TRUE) NoAttr <- read.csv(file.choose(), header= TRUE) NoSalary <- read.csv(file.choose(), header= TRUE)

#Check dimensions of dataset

dim(EData) dim(df_val)

#Summary summary(EData)

#convert data to data frame Edata_df = as.data.frame(EData) NoAttr_df= as.data.frame(NoAttr) NoSalary_df = as.data.frame(NoSalary)

#checking to see if any missing variables gg_miss_var(Edata_df) colSums(is.na(Edata_df))

#Count of #Attrition #730 #Approximately 16% of the employees left the organization

LeftCompany <- table(Edata_df$Attrition)$LeftCompany < -as.factor(Edata_df$Attrition)$ summary(LeftCompany)

#Plot barplot(prop.table(table(Edata_df$Attrition)))

#Create variables for Attrition Yes/No

filter_N <- filter(Edata_df, Attrition == "No") filter_Y <- filter(Edata_df, Attrition == "Yes")

shapiro.test(Edata_df$DistanceFromHome)$shapiro.test(Edata_df$Education)$ shapiro.test(Edata_df$Department)$shapiro.test(Edata_df$EnvironmentSatisfaction)$ shapiro.test(Edata_df$PerformanceRating)$shapiro.test(Edata_df$TotalWorkingYears)$ shapiro.test(Edata_df$WorkLifeBalance)$shapiro.test(Edata_df$YearsSinceLastPromotion)$ shapiro.test(Edata_df$JobLevel)

Histogram to show the difference

filter_N %>% ggplot(aes(x = Education)) + geom_histogram(binwidth = 0.5) filter_N %>% ggplot(aes(x = EnvironmentSatisfaction)) + geom_histogram(binwidth = 0.5) filter_N %>%

ggplot(aes(x = DistanceFromHome)) + geom_histogram(binwidth = 0.5) filter_N %>%
ggplot(aes(x = PerformanceRating)) + geom_histogram(binwidth = 0.5) filter_N %>%
ggplot(aes(x = TotalWorkingYears)) + geom_histogram(binwidth = 0.5) filter_N %>%
ggplot(aes(x = WorkLifeBalance)) + geom_histogram(binwidth = 0.5) filter_N %>%
ggplot(aes(x = JobLevel)) + geom_histogram(binwidth = 0.5) filter_N %>% ggplot(aes(x = YearsSinceLastPromotion)) + geom_histogram(binwidth = 0.5)

#Cleaning Data - Removing columns that are not useful in determining Attrition

df_stage <- Edata_df[,!(names(Edata_df) %in% c("ID", "StandardHours", "EmployeeNumber", "EmployeeCount", "Over18", "PercentSalaryHike","NumCompaniesWorked", "TrainingTimesLastYear", "StockOptionLevel", "DailyRate","RelationshipSatisfaction", "PerformanceRating", "EnvironmentSatisfaction","HourlyRate", "WorkLifeBalance"))]

df_stage2 <- NoAttr_df[,!(names(NoAttr_df) %in% c("ID", "StandardHours", "EmployeeNumber", "EmployeeCount", "Over18", "PercentSalaryHike","NumCompaniesWorked", "TrainingTimesLastYear", "StockOptionLevel", "DailyRate","RelationshipSatisfaction", "PerformanceRating", "EnvironmentSatisfaction","HourlyRate", "WorkLifeBalance"))]

#An additional dataframe was created with only numeric values to be read by a correlation #heatmap later in the analysis.

#Return numeric values only df_numeric <- df_stage[, sapply(df_stage, is.numeric)] dim(df_numeric)

df_numeric2 <- df_stage2[, sapply(df_stage2, is.numeric)] dim(df_numeric2)

#Data Exploration

#Scatter plot between Monthly Income, Work life Balance and Attrition
ggplot(Edata_df,aes(Edata_df$MonthlyIncome$, $Edata_af$ WorkLifeBalance, color=Attrition))+geom_point()

#scatter plot between monthly income, JobLevel and attrition
ggplot(Edata_df,aes(Edata_df$MonthlyIncome$, $Edata_af$ JobLevel, color=Attrition))+geom_point()

#boxplot between monthly income and attrition
ggplot(Edata_df,aes(Attrition,MonthlyIncome,fill=Attrition))+geom_boxplot()

#boxplot between monthly income and attrition
ggplot(Edata_df,aes(Attrition,YearsSinceLastPromotion,fill=Attrition))+geom_violin()

#Correlation between Marital status and Attrition
ggplot(Edata_df,aes(Attrition,MaritalStatus,color=Attrition))+geom_jitter()

#Job Roles p1 <- ggplot(Edata_df, aes(x=JobRole), color=JobRole) + ggtitle("Figure A: Job Role") + geom_bar(aes(y = 100*(..count..)/sum(..count..), fill=JobRole), width = 0.5) +

#geom_text(aes(label=100(..count..)/sum(..count..)), vjust=0) + labs(y="Percentage") + coord_flip() + theme_linedraw() + theme(plot.title = element_text(hjust = 0.5)) p1

#Job Role by Percent of Total

p2 <- ggplot(Edata_df, aes(x=JobRole, y = ..prop.., group=1)) + geom_bar() + geom_text(stat = "count", aes(label = round(..prop.., 2), y = ..prop.. + 0.02)) + coord_flip() + ggtitle("Figure B: Job Role by Percent of Total") + theme(plot.title = element_text(hjust = 0.5)) p2

#Job Role vs Attrition Count p3 <- ggplot(Edata_df,aes(x = JobRole,fill = Attrition)) + geom_bar(position = "dodge") + ggtitle("Job Role vs Attrition - Count") + theme(plot.title = element_text(hjust = 0.5)) p3

Department VS Attrition

p4 <- ggplot(Edata_df,aes(x = Department, fill = Attrition)) + geom_bar(position ="dodge") + ggtitle("Department vs Attrition") theme(plot.title = element_text(hjust = 0.2)) p4

#Next, we examine the frequency for gender and how it is distributed #across job roles visually.

demographics <- df_stage[,c("JobRole", "Gender", "EducationField")]

build_table(demographics$Gender, "Gender") build_table(demographics$EduField, "Education") build_table(demographics$JobRole, "Job Role")

theme_set(theme_light())

ggplot(demographics, aes(demographics$JobRole)) + geom_bar(aes(fill=Gender), width = 0.5) + labs(title = "Gender by Job Role", subtitle = "Men & Women in Specific JobRole", x = "Job Role", y = "Frequency") + coord_flip()

ggplot(df_stage, aes(x=Age, y=YearsAtCompany)) + ggtitle("Years at Company vs Age") + theme(plot.title = element_text(hjust = 0.5)) + labs(x= "Age", y="Years") + geom_point(shape=1, col = "purple") + geom_smooth(method = "gam")

ggplot(df_stage, aes(YearsAtCompany)) + geom_density(aes(fill=JobSatisfaction), alpha=0.8) + labs(title= "Years At the Company Density Plot", subtitle="Years at the Company grouped by Job Satisfaction", x="Years at Company", y="Density")

#Correlation Matrix and Heat Map with the Numeric Variables

#Correlation Plotf df_corr <- round(cor(df_numeric),2)

#cor(df_numeric, method = "pearson", use = "complete.obs") #corrplot(df_corr, order="FPC", title="Variable Corr Heatmap")

corrplot::corrplot.mixed (df_corr, lower = "circle", upper ="number", tl.pos = "lt", diag ="l") library(PerformanceAnalytics) chart.Correlation(df_numeric, histogram=TRUE, pch=19)

col<- colorRampPalette(c(“blue”, “white”, “red”))(20) heatmap(x = df_corr, col = col, symm = TRUE)

library(ggcorrplot)

ggcorrplot(df_corr, hc.order = TRUE, type = “lower”, lab = TRUE, lab_size = 4, method=“circle”, colors= c(“tomato2”, “white”, “springgreen3”), title = “Correlation Chart For TalentData”, ggtheme=theme_bw)

#Running Regression Model on Monthly Income vs Age and Gender ggplot(df_stage, aes(MonthlyIncome, Age, color = Gender, shape=Gender))+geom_point()+ggtitle(“Correlation between Monthly income and Ages”) model_AgeIncome <- lm(MonthlyIncome ~ Age+Gender, data = df_stage) summary(model_AgeIncome)

df_stage$Attrition < -as.factor(df_stage$Attrition) df_stage$Attrition < -as.numeric(df_stage$Attrition)

#Logistic Regression Model

lr_mod <- glm(Attrition~Age+BusinessTravel+Department+DistanceFromHome+EducationField+Gender +JobInvolvement+JobLevel+JobRole+JobSatisfaction+MaritalStatus+MonthlyIncome +MonthlyRate+OverTime++TotalWorkingYears+ YearsAtCompany+YearsInCurrentRole+YearsSinceLastPromotion+YearsWithCurrManager, data=df_stage, family=binomial(link='logit'))

summary(lr_mod)

lr_mod2 <- lm(Attrition~Age+BusinessTravel+Department+DistanceFromHome+EducationField+Gender +JobInvolvement+JobLevel+JobRole+JobSatisfaction+MaritalStatus+MonthlyIncome +MonthlyRate+OverTime++TotalWorkingYears+ YearsAtCompany+YearsInCurrentRole+YearsSinceLastPromotion+YearsWithCurrManager, data=df_stage)

summary(lr_mod2)

#K-Nearest Neighbors Model

mjob_outcome <- Edata_df %>% dplyr::select(Attrition)

set.seed(1234) splitpale = .70 samplesplit <- sample(1:dim(df_stage)[1], round(splitpale * dim(df_stage)[1])) # The following code separates the randomly selected values into the 70 to 30 split. trainset <- df_numeric[samplesplit,] testset <- df_numeric[-samplesplit,]

mjob_outcome_train <- mjob_outcome[samplesplit,] mjob_outcome_test <- mjob_outcome[-samplesplit,]

modelknn <- knn(train = trainset, test = testset, cl = mjob_outcome_train, k=25)

put “mjob_outcome_test” in a data frame

mjob_outcome_test <- data.frame(mjob_outcome_test)

merge “mjob_pred_knn” and “mjob_outcome_test”

class_comparison <- data.frame(modelknn, mjob_outcome_test)

specify column names for “class_comparison”

names(class_comparison) <- c(“PredictedMjob”, “ObservedMjob”)

inspect “class_comparison”

head(class_comparison)

mjob_pred_caret <- train(trainset, mjob_outcome_train, method = “knn”, preProcess = c(“center”,“scale”))

mjob_pred_caret plot(mjob_pred_caret)

#Checking classification accuracy

knnPredict <- predict(mjob_pred_caret, newdata = testset)
mjob_outcome_test$mjob_outcome_test <
$-as.factor(mjob_outcome_test$mjob_outcome_test)$

confusionMatrix(knnPredict, mjob_outcome_test$mjob_outcome_test)

mjob_outcome_test$mjob_outcome_test

#The above implementation gave me an accuracy of 0.85 and a Sensitivity of 0.995
###Classifying the the new unknown data set that doesnt contain attrition label.

final_dataset.new = cbind(Edata_df, NoAttr[1]) #Unknown is ready for prediction but we need to tag the observations with the unique ID final_dataset = cbind()

set.seed(200) pred_knn.new <- predict(df_stage, newdata=final_dataset.new, type =“raw”)

#I cannot build a confusion MAtrix because I don’t have an expected label for the dataset.
#confusionMatrix(table(pred_knn.new,as.factor(training$Attrition)))
summary(pred_knn.new)

hist(trsf.new$MonthlyIncome$)hist(final_dataset.new$MonthlyIncome)

prop.table(table(NoAttr.pred$pred_Attrition)) str(NoAttr.pred) head(NoAttr.pred)

no.attrition_df_recode%>%filter (ID == 1171) %>% dplyr::select(MonthlyRate)

#combine the predicted attrition data with the unknown dataset and add unique ID

NoAttr.pred <- cbind(final_dataset.new, pred_Attrition=pred_knn.new) NoAttr.pred <- no.attrition_df.pred %>% mutate(pred_Attrition = as.factor(if_else(pred_Attrition == 0,“No”, “Yes”))) NoAttr.pred1 <- no.attrition_df.pred %>% dplyr::select(ID, pred_Attrition) NoAttr.pred_new <- left_join(no.attrition_df_recode, NoAttr.pred1, by = “ID”) str(no.attrition_df.pred_new)

#Write the result to a csv file and move it to github write.csv(no.attrition_df.pred_new, ‘C:\Users\lawodipe\OneDrive\Documents\Data Science\SMU-Data Science\Doing Data Science\MSDS_6306_DDS\Unit 14 and 15 Case Study 2\OloladeAwodipePrediction.csv’)

###Result evaluation

#Employee’s department and other factors should agree with the top factors selected by linear the model #and trend in our training data. #This is incorrect.

Attrition.Yes <- NoAttr.pred_new %>% filter(pred_Attrition == “Yes”)

head(Attrition.Yes, n=20)

###Examine the employees predicted to leave the company. #Firstly,K-NN predicted result reveals that most of the employees predicted to leave the company work as Lab Tech or sales rep. They rarely travel on business and have low monthly income. This category of employees also have either low or very high total working years suggesting early career with high mobility and late career departing due to retirement. Some of these factors are not included in the top factors identified by the linear model. We should also note that the linear model was only able to #explain about 16% (r = 0.411) of attrition.

#Training the data on salary. describe(training) income_mod1 = glm(MonthlyIncome~., data = training[,c(-4, -11,-18,-21,-20,-31)])

model_summary <- summary(income_mod1)

step<- stepAIC(income_mod1,direction = “backward”,trace=FALSE) summary(step) step$coefficients$step anova

#The final model selected based on the above.

Final.model.sal <- glm(MonthlyIncome ~ BusinessTravel + Education + Gender + JobInvolvement + JobLevel + JobRole + TotalWorkingYears, data = training[,c(-4, -11,-18,-21,-20,-31)])

step_select.r1 <- with(summary(Final.model.sal), 1 - deviance/null.deviance) #r =0.90, r^2 = 0.81

pred_lr.sal <- predict(Final.model.sal, newdata=testing)#, type = “response”) summary(pred_lr.sal)

#Adding the predicted salary to the test data and also back converting the transformed salary response salary.pred_df <- cbind(testing, pred_Monthly_income.log=pred_lr.sal, pred_monthlyincome = (exp(pred_lr.sal)-1)) str(salary.pred_df)

#Write the result to a csv file and move it to github write.csv(salary.pred_df, 'C:\Users\lawodipe\OneDrive\Documents\Data Science\SMU-Data Science\Doing Data Science\MSDS_6306_DDS\Unit 14 and 15 Case Study 2\OloladeAwodipemonthlyincome.csv')

#Residual plots and Cook's D plots to check for assumptions.

par(mfrow = c(2, 1)) p_r1 <- plot(income_mod1$fitted.values, step_select$residuals, main = "Residual Plot for salary prediction") p_c1 <- plot(cooks.distance(income_mod1), main = "Cooks' D for Salary prediction") par(mfrow = c(1, 1))

#There are points that are far from the regression line and there are no random clouds of residuals around -2 to +2. #This shows that the linear model is not a good model to classify the response. #on the positive side the Cooks'D plot did not show any strongly high leverage point,so there are no extreme outlier in the plotted data.

#calculating the RMSE library(qpcR) qpcR::RMSE(income_mod1)

#Youtube - https://www.screencast.com/t/RcCyNXZk