**Employee Attrition**

by Machine Learning using Python

Prerna Thakur, Kalyani Government Engineering College, 10200117039 of 2017-2018

Aditya Kumar Jha, Kalyani Government Engineering College, 10200117076 of 2017-2018

Abhishek Bhuin, Kalyani Government Engineering College, 10200316072 of 2017-2018

Anirban Dey, Kalyani Government Engineering College, 10200218016 of 2017-2018

Arpan Deria, Kalyani Government Engineering College, 10200216063 of 2017-2018

Index

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Contents** | **Page Number** |
| 1 | Acknowledgement | 3-7 |
| 2 | Scope of Project | 8 |
| 3 | Introduction of useful terms | 9-21 |
| 4 | Data Description | 22-43 |
| 5 | Code | 44-51 |
| 6 | Data Analysis | 52-67 |
| 7 | Conclusion | 68-70 |
| 8 | Result | 71 |

# Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty Mr. Titas Roy Chowdhury for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help and guidance given time to time shall carry me a long way in the journey of life on which I am about to embark. I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment. – PRERNA THAKUR

# Acknowledgement

I take this oppourtunity to express my profound gratitude and deep regards to my faculty Mr. Titas Roy Chowdhury for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help and guidance given time to time shall carry me a long way in the journey of life on which I am about to embark. I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

-ADITYA KUMAR JHA

**Acknowledgement**

I take this opportunity to express my profound gratitude and deep

regards to my faculty Mr. Titas Roy Chowdhury for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help and guidance given time to time shall carry me a long way in the journey of life on which I am about to embark. I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

-ABHISHEK BHUIN

**Acknowledgement**

I take this opportunity to express my profound gratitude and deep

regards to my faculty Mr. Titas Roy Chowdhury for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help and guidance given time to time shall carry me a long way in the journey of life on which I am about to embark. I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

-ANIRBAN DEY

**Acknowledgement**

I take this opportunity to express my profound gratitude and deep

regards to my faculty Mr. Titas Roy Chowdhury for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help and guidance given time to time shall carry me a long way in the journey of life on which I am about to embark. I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

-ARPAN DERIA

# Project Scope

Employers generally consider attrition a loss of valuable employees and talent. However, there is more to attrition than a shrinking workforce. As employees leave an organization, they take with them muchneeded skills and qualifications that they developed during their tenure. On the other hand, junior professionals with promising qualifications can then succeed into higher level positions or business owners can introduce more diversity in experience or expertise. Accordingly, there are benefits and disadvantages to attrition. The cost of attrition can be relatively enormous.

# Employee Attrition

Employers generally consider attrition a loss of valuable employees and talent. However, there is more to attrition than a shrinking workforce. As employees leave an organization, they take with them muchneeded skills and qualifications that they developed during their tenure. On the other hand, junior professionals with promising qualifications can then succeed into higher level positions or business owners can introduce more diversity in experience or expertise. Accordingly, there are benefits and disadvantages to attrition.

## Attrition Costs

The cost of attrition can be relatively enormous. Attrition from retirement or resignation diminishes the workforce, demanding additional work hours and dedication from remaining employees. Whereas long-term workers have established bonds with customers and clients, attrition can reduce this rapport, running the risk of losing them to a competitor. Losing clientele affects revenue, profitability and business reputation.

Attrition from retirement or resignation diminishes the workforce, demanding additional work hours and dedication from remaining employees. Whereas long-term workers have established bonds with customers and clients, attrition can reduce this rapport, running the risk of losing them to a competitor. Losing clientele affects revenue, profitability and business reputation.

Despite the negatives of attrition, healthy attrition -- or, desirable turnover -- can positively affect organizations. Losing employees with poor performance records can boost employee morale, employee engagement and productivity among the current workforce. Moreover, attrition can be encouraging to young professionals seeking promotion and upward mobility.

# Attrition Analysis

Employee attrition analysis refers to the employee attrition rate in a company. This analysis helps SaaS companies identify the cause of the churn and implement effective strategies for retention. Gainsight understands the negative impact that churn rate can have on company profits. Named as the "2014 cool vendor for CRM sales" (by Gartner), Gainsight’s customer intelligence and retention process automation technology:

* Gathers available employee behavior, transactions, demographics data and usage pattern
* Converts structured and unstructured data/information into meaningful insights
* Utilizes these insights to predict customers who are likely to churn
* Identifies the causes for churn and works to resolve those issues
* Engages with customers to foster relationships
* Implements effective programs for customer retention Statistics show that acquiring new customers can cost five times more than retaining existing customers. Gainsight performs customer churn analysis to reduce churn, control retention and improve performance.

**How we propose to solve:**

* Analyse the data frame
* Drop the columns which have high multi collinearity and low contribution towards ‘label’.
* Apply different classification models and note down the scores for each models.
* Select the models with the best scores

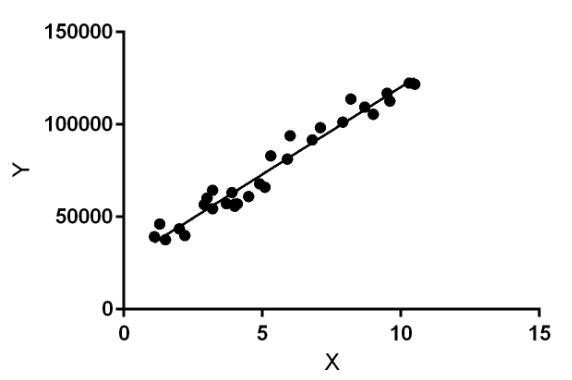
**Some machine learning methods**

Machine learning algorithms are often categorized as supervised or unsupervised.

* **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
* In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.

# Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. It performs aregression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.



Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

Hypothesis function for Linear Regression:

𝑦 = 𝜃1 + 𝜃2𝑥

While training the model we are given:

𝑥: input training data (univariate – one input variable(parameter))

𝑦: labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best 𝜃1 and 𝜃2 values.

𝜃1: intercept

𝜃2: coefficient of 𝑥

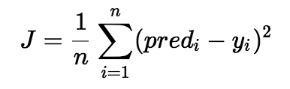
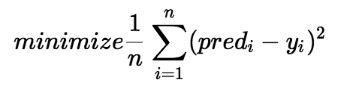
Once we find the best 𝜃1 and 𝜃2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of 𝑦 for the input value of 𝑥.

How to update 𝜃 1and 𝜃2 values to get the best fit line ?

**Cost Function (J):**

By achieving the best-fit regression line, the model aims to predict

𝑦 value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimize the error between predicted 𝑦 value (pred) and true 𝑦 value (𝑦).



Cost function(𝐽) of Linear Regression is the Root Mean Squared Error(RMSE) between predicted 𝑦 value (pred) and true 𝑦 value

(𝑦).

**Gradient Descent:**

To update 𝜃1 and θ2 values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random 𝜃1and 𝜃2 values and then iteratively updating the values, reaching minimum cost.

# Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Sometimes logistic regressions are difficult to interpret; the Intellectus Statistics tool easily allows you to conduct the analysis, then in plain English interprets the output.

**Overfitting-** When selecting the model for the logistic regression analysis, another important consideration is the model fit. Adding independent variables to a logistic regression model will always increase the amount of variance explained in the log odds (typically expressed as R²). However, adding more and more variables to the model can result in overfitting, which reduces the generalizability of the model beyond the data on which the model is fit.

**Assumptions in Logistic Regression:**

* The dependent variable should be dichotomous in nature (e.g., presence vs. absent).
* There should be no outliers in the data, which can be assessed by converting the continuous predictors to standardized scores, and removing values below -3.29 or greater than 3.29.

* There should be no high correlations (multicollinearity) among the predictors. This can be assessed by a correlation matrix among the predictors.

# Decision Tree

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.

A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers the to the question; and the leaves represent the actual output or class label. They are used in non-linear decision making with simple linear decision surface.

# Confusion Matrix

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another).

It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table).

**Accuracy**:

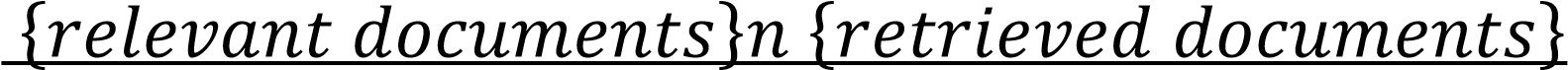
𝐴𝐶 = 𝑇𝑁 + 𝑇𝑃𝑇𝑁 + 𝐹𝑃 + 𝐹𝑁 + 𝑇𝑃𝐴𝐶

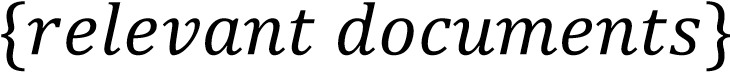
= 𝑇𝑁 + 𝑇𝑃𝑇𝑁 + 𝐹𝑃 + 𝐹𝑁 + 𝑇𝑃

The accuracy is not always an adequate performance measure. Let us assume we have 1000 samples. 995 of these are negative and 5 are positive cases. Let us further assume we have a classifier, which classifies whatever it will be presented as negative. The accuracy will be a surprising 99.5%, even though the classifier could not recognize any positive samples.

**Recall** **or True Positive Rate**:

In information retrieval, recall is the fraction of the relevant documents that are successfully retrieved.

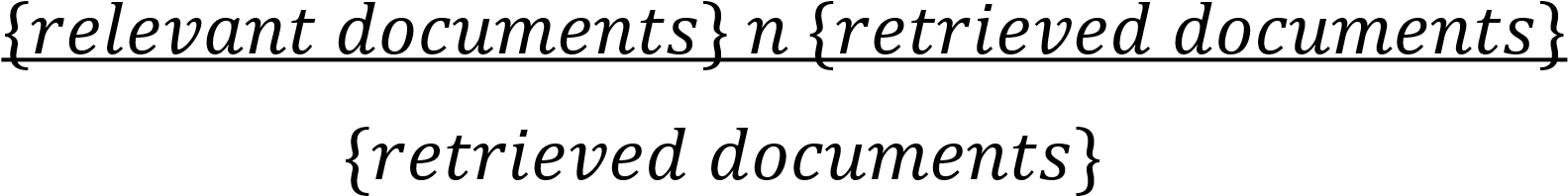
recall =



**True Negative Rate**:



**Precision**:

In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the query: Precision = 

**Multi class case**

To measure the results of machine learning algorithms, the previous confusion matrix will not be sufficient. We will need a generalization for the multi-class case.

# Correlation

Correlation refers to a mutual relationship or association between quantities. In almost any business, it is useful to express one quantity in terms of its relationship with others. For example, sales might increase when the marketing department spends more on TV advertisements, or a customer's average purchase amount on an ecommerce website might depend on a number of factors related to that customer. Often, correlation is the first step to understanding these relationships and subsequently building better business and statistical models.

Correlation is a useful metric because-

* Correlation can help in predicting one quantity from another
* Correlation can (but often does not, as we will see in some examples below) indicate the presence of a causal relationship
* Correlation is used as a basic quantity and foundation for many other modelling techniques

# Data Preprocessing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.

Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Data Preprocessing is needed for-

* For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.
* Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

# MinMax Scaling

An alternative approach to Z-score normalization (or standardization) is the so-called Min-Max scaling (often also simply called "normalization" - a common cause for ambiguities). In this approach, the data is scaled to a fixed range - usually 0 to 1. The cost of having this bounded range - in contrast to standardization - is that we will end up with smaller standard deviations, which can suppress the effect of outliers.

## Random forest classifier

Random forest is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forest creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

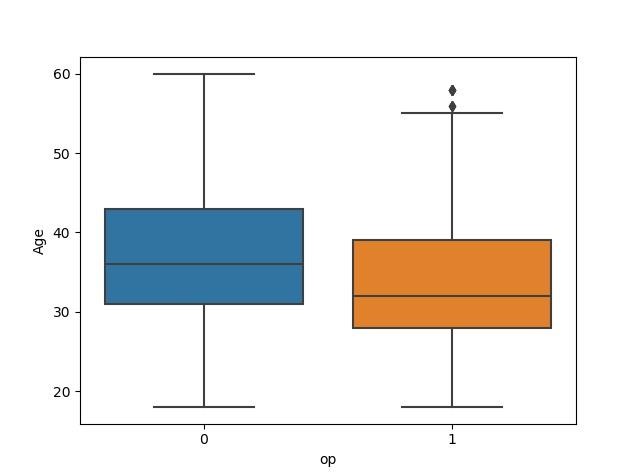
Random forest has a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

### Data Description

#### Boxplot of attrition vs age of employee

𝑥=Attrition

𝑦=Age of employee



Data Type-Integer

Value type: continuous

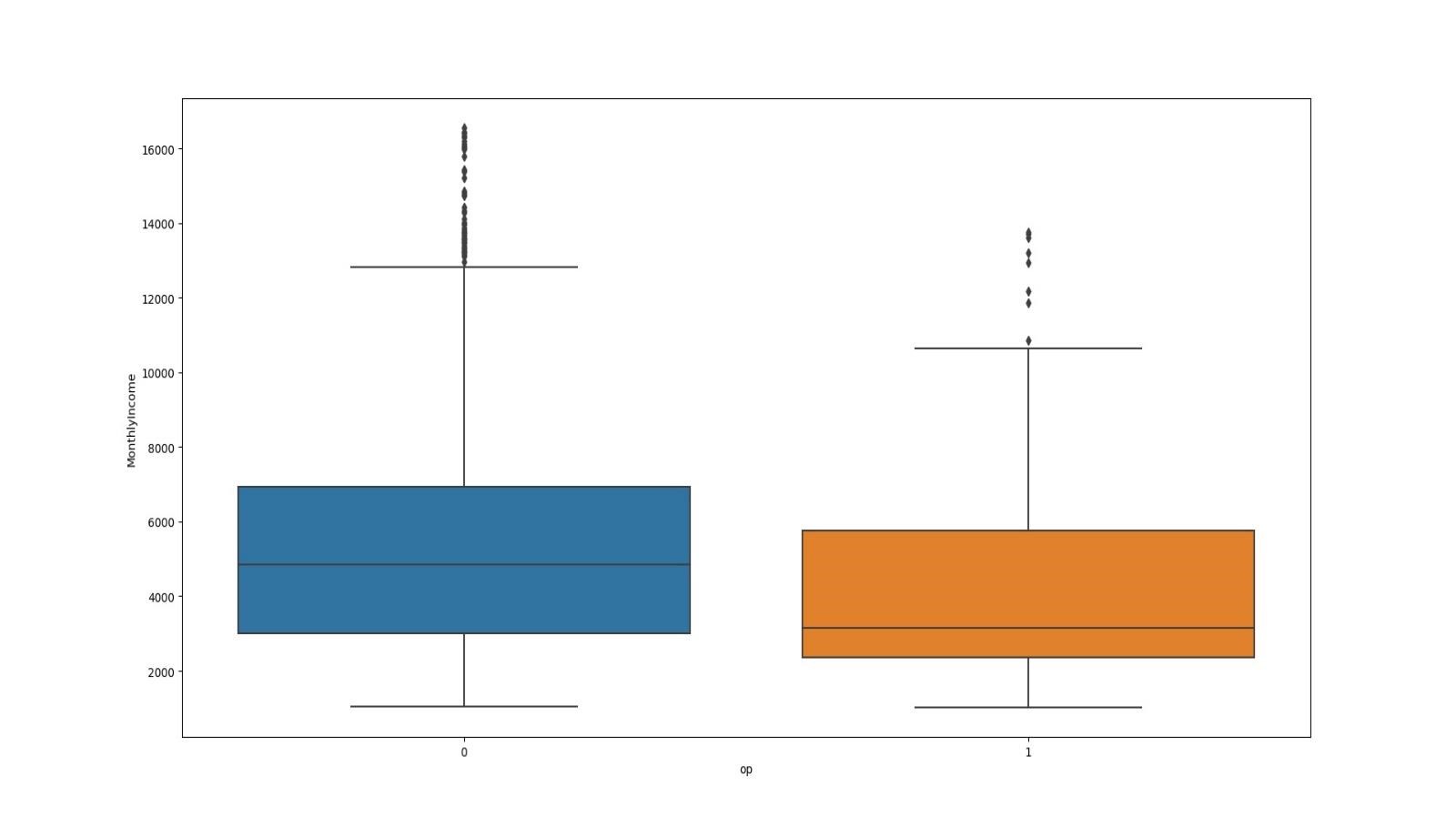
Null percentage: 0

It’s a depending graph and few outliers present

#### Boxplot of attrition vs monthly income

𝑥= Attrition

𝑦=Monthly Income

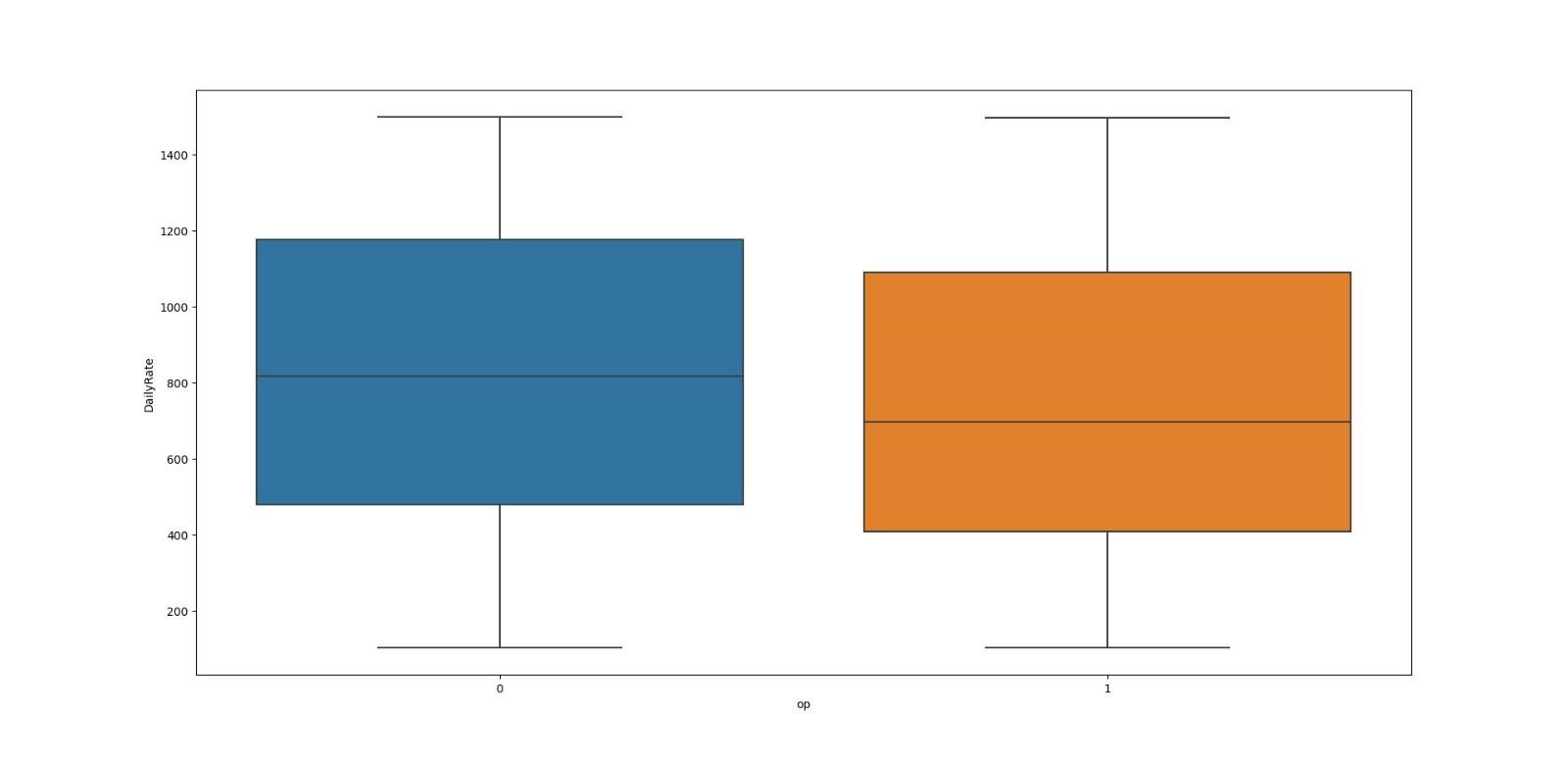


Monthly income is depending but outliers are present.

#### Boxplot of attrition vs daily rate

𝑥=Attrition

𝑦=Daily Rate

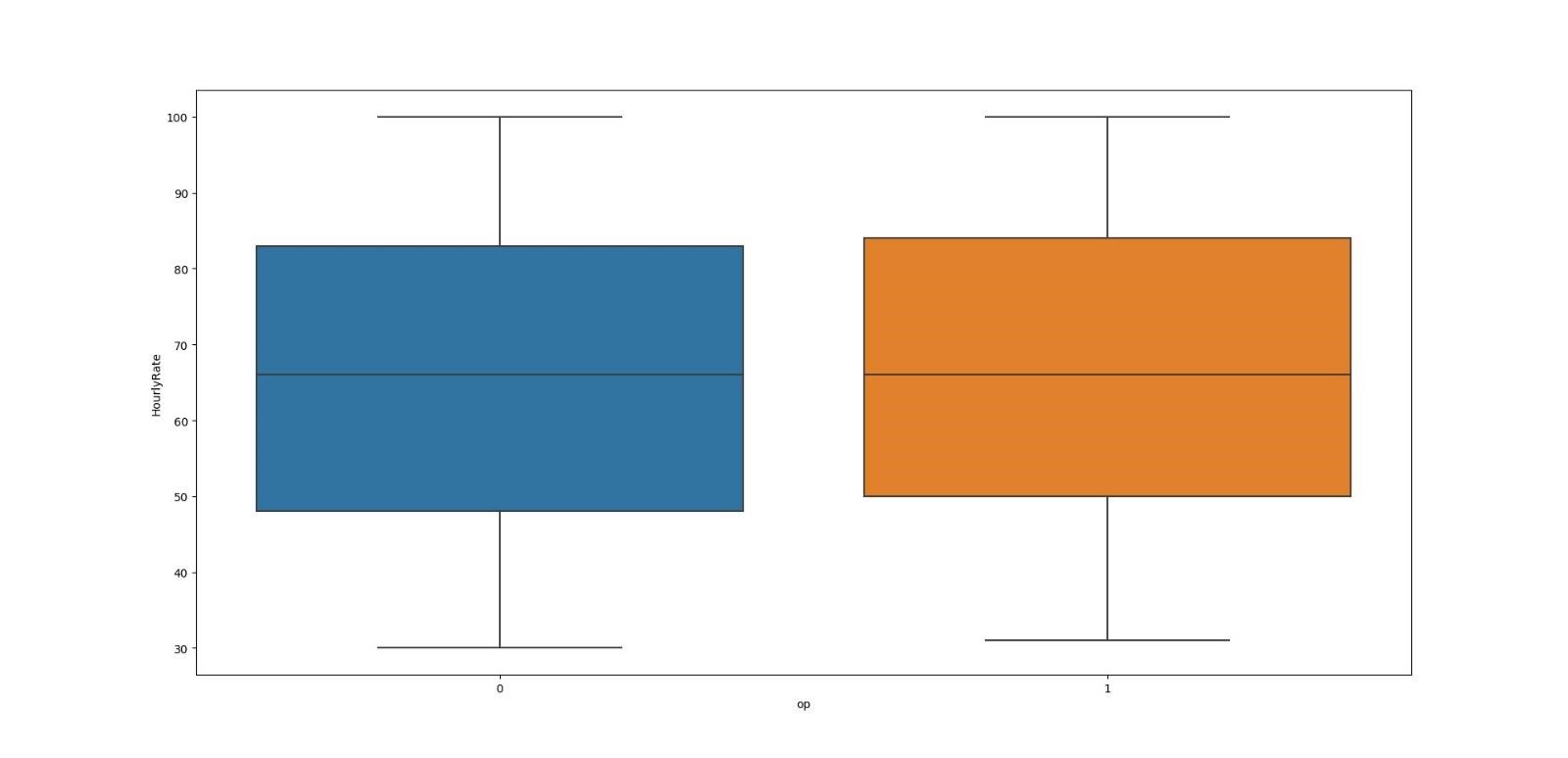


Slight dependencies can be concluded from the graph.

**Boxplot of attrition vs hourly rate.**

𝑥=Attrition

𝑦=Hourly Rate

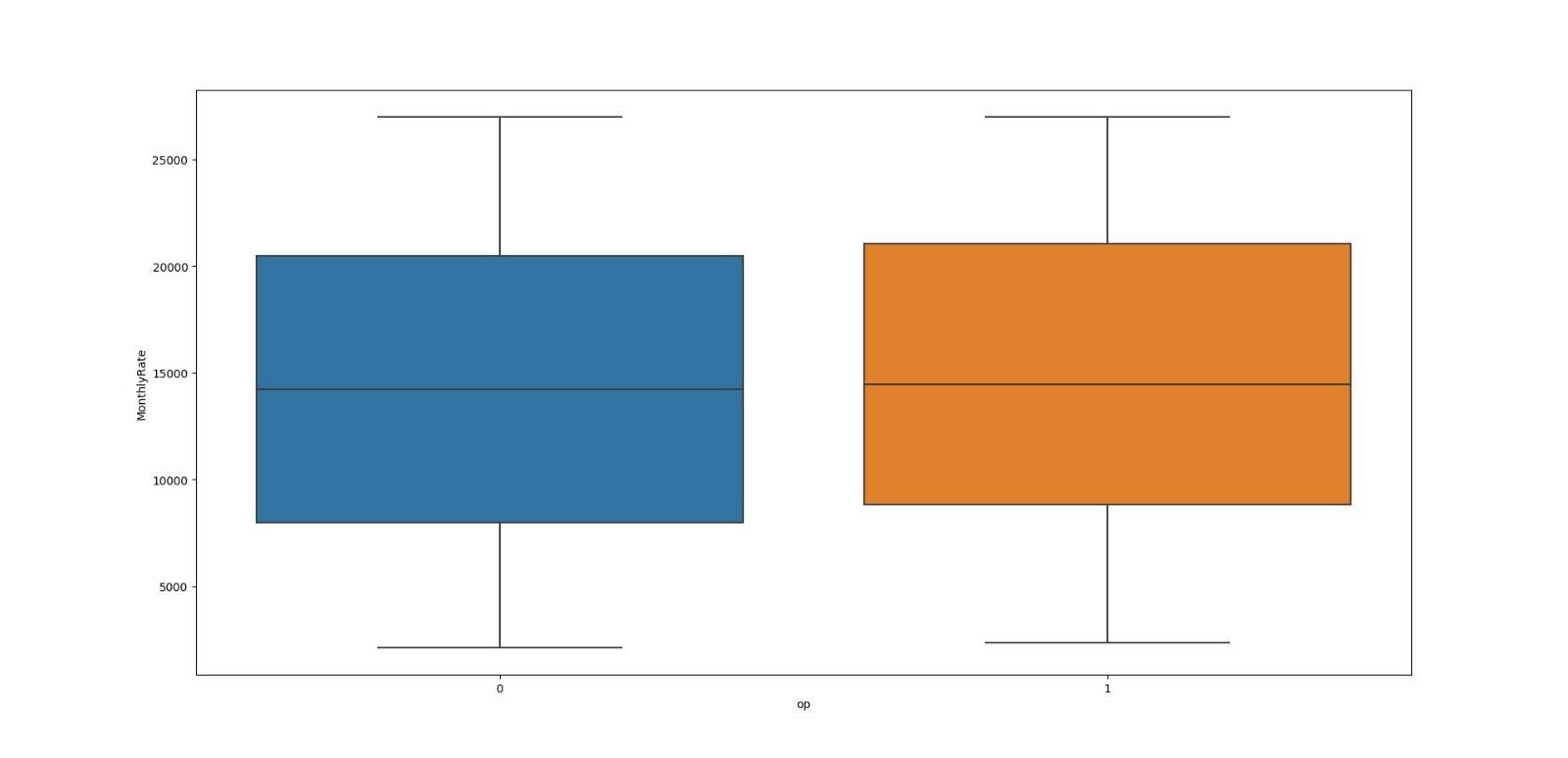


We can conclude that its almost independent.

#### Boxplot of attrition vs monthly rate

𝑥=Attrition

𝑦=Monthly Rate

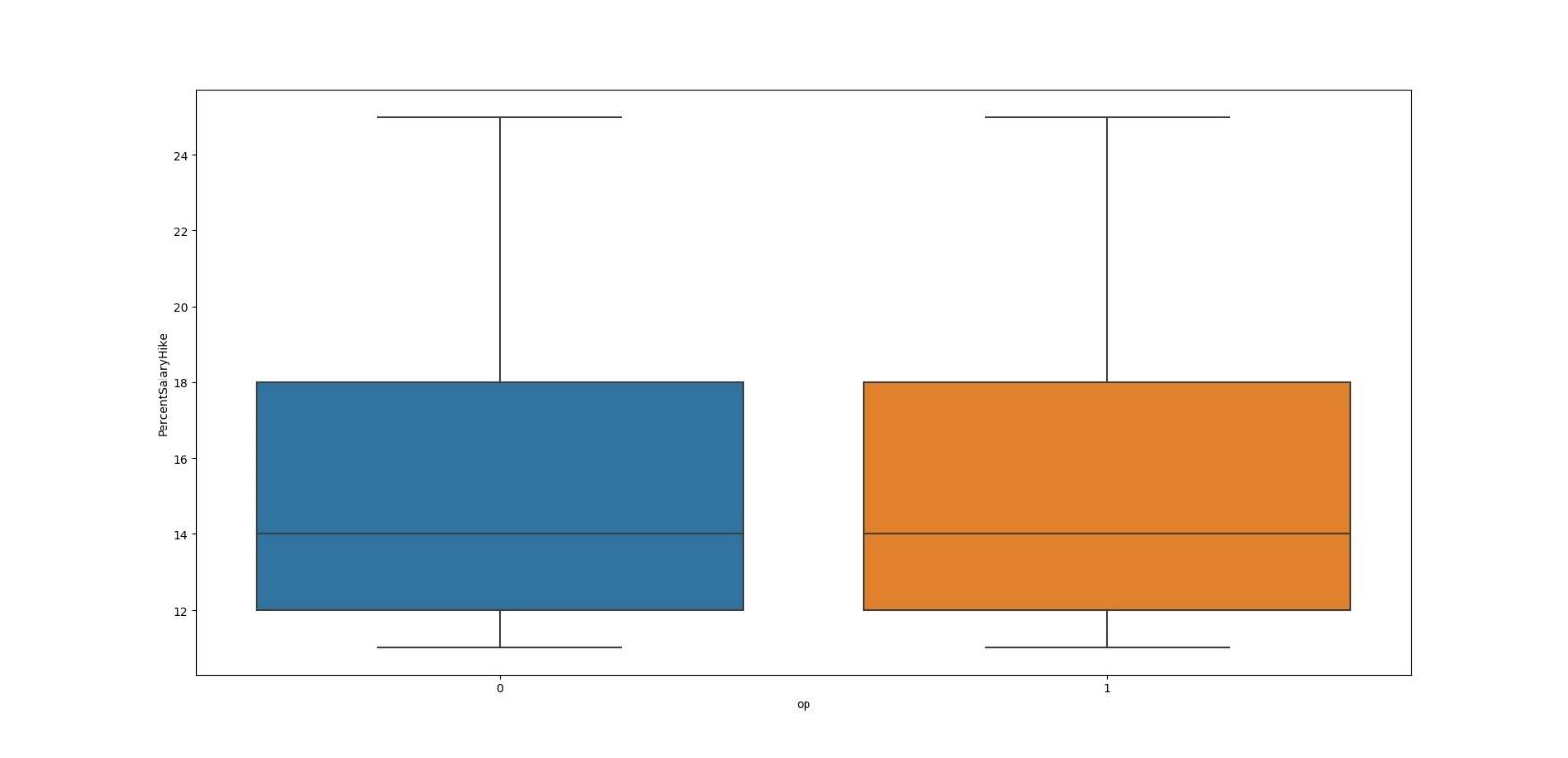


Not much dependent.

#### Boxplot of attrition vs percentage salary hike

𝑥=Attrition

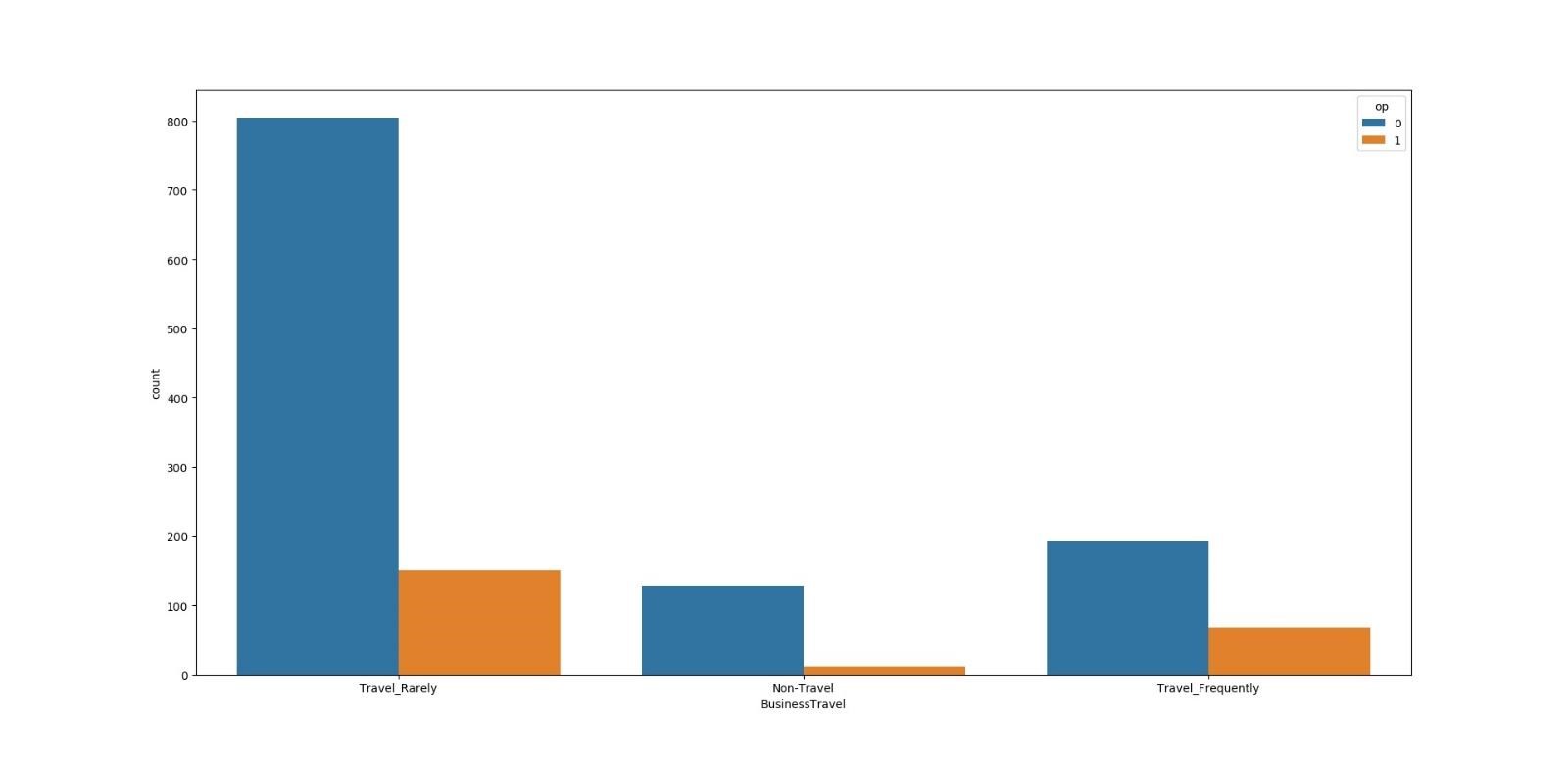
𝑦=Percentage Salary Hike



It’s almost independent.

#### Countplot of attrition vs business travel

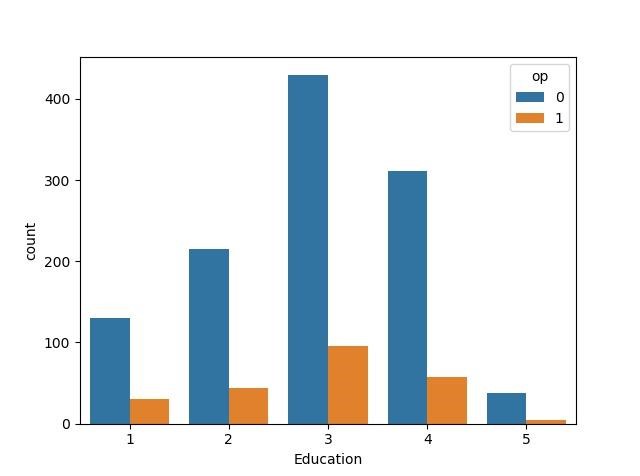
=Attrition y=Business Travel



#### Countplot of attrition vs count

𝑥=Education

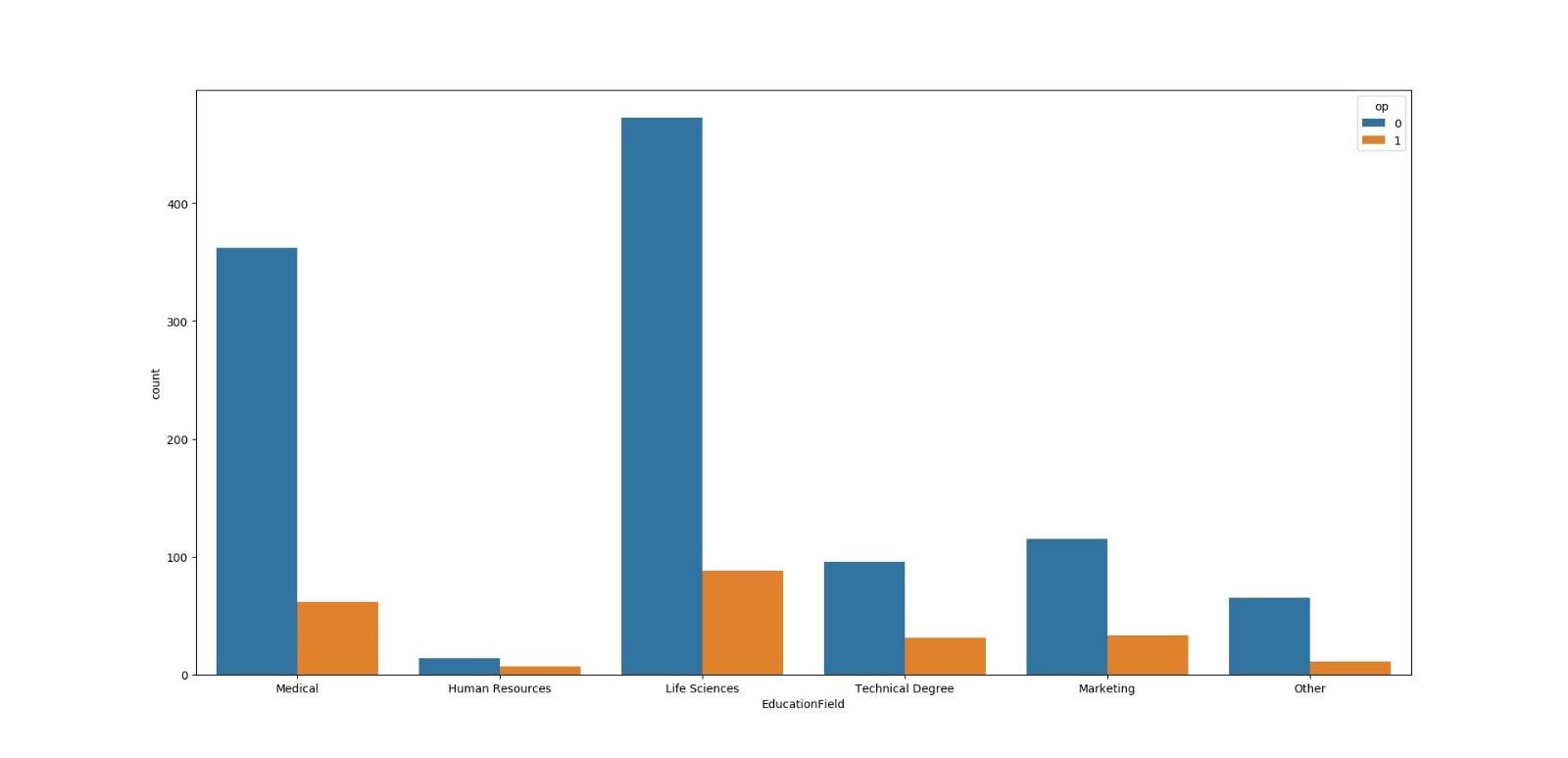
𝑦=Count



#### Countplot of education field vs count

𝑥=Education Field

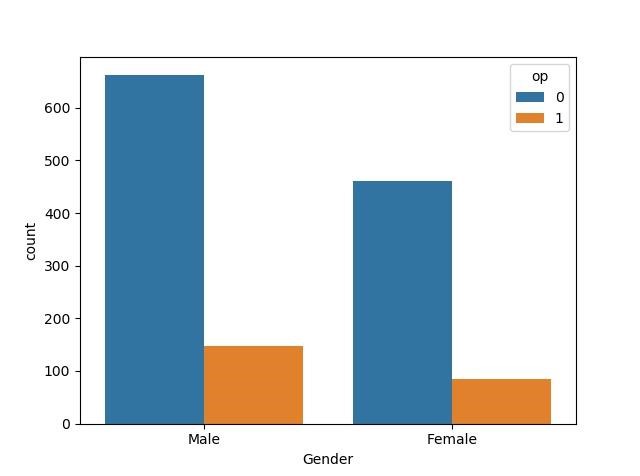
𝑦=Count



#### Countplot of gender vs count

𝑥=Gender

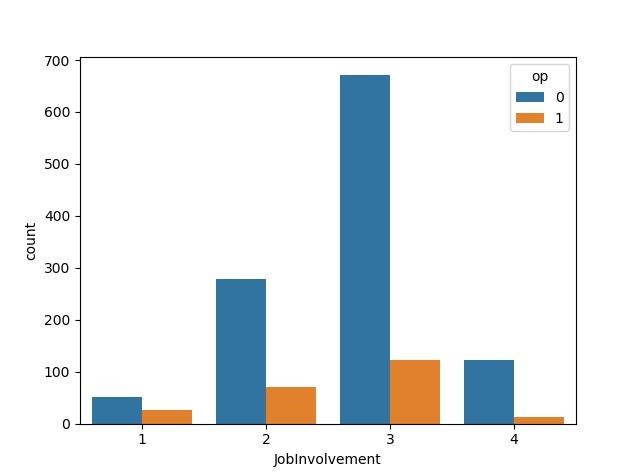
𝑦=Count



#### Countplot of job involvement vs count

𝑥=Job Involvement

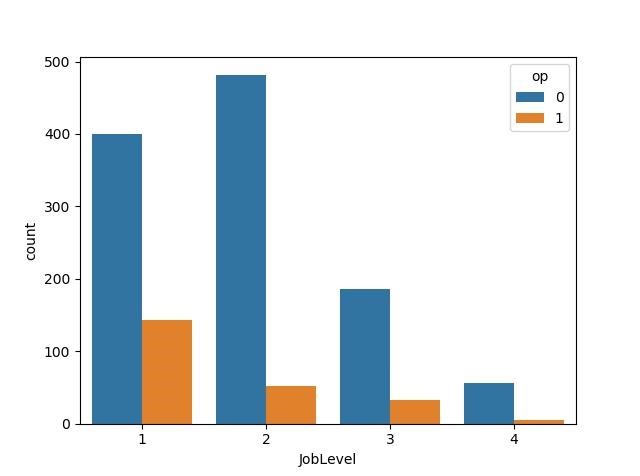
𝑦=Count



#### Countplot of job level vs count

𝑥=Job Level

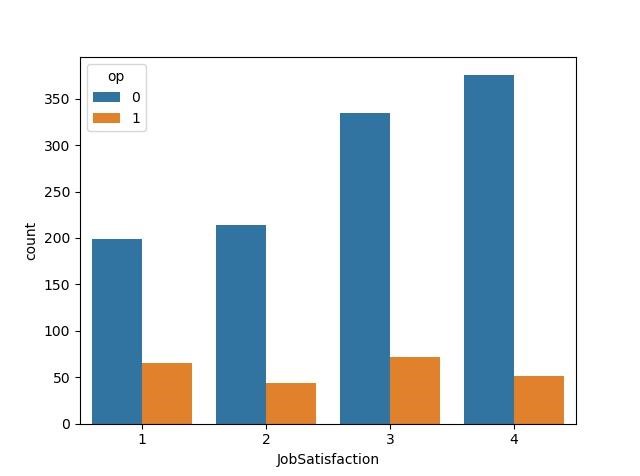
𝑦=Count



#### Countplot of Job Satisfaction vs. Count

𝑥=Job Satisfaction

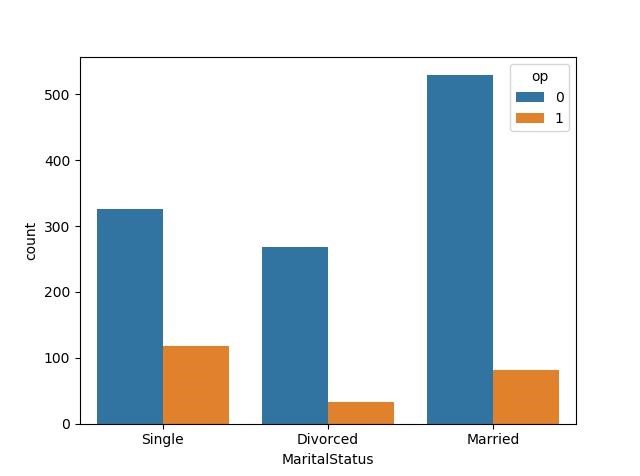
𝑦=Count



#### Countplot of Divorced Martial Status vs count

𝑥=Marital Status

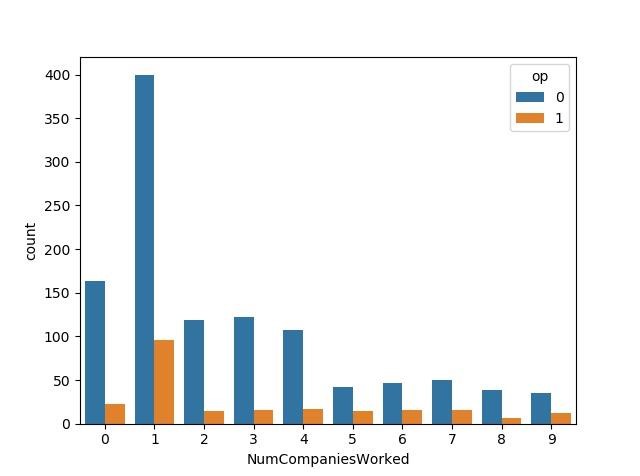
𝑦=Count



#### Countplot of Number of Companies worked vs count

𝑥=Number of Companies worked

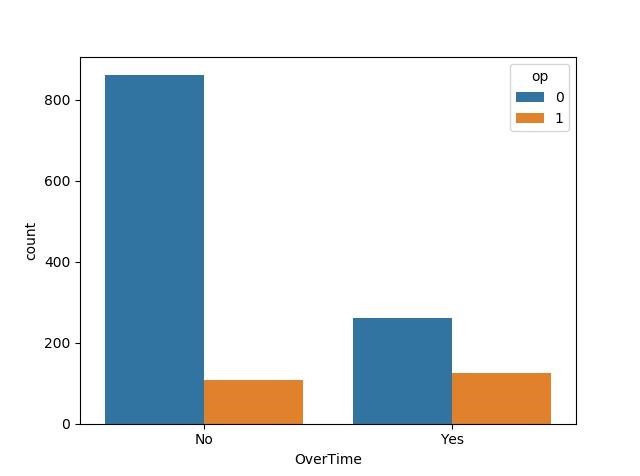
𝑦=Count



#### Countplot of Overtime vs count

𝑥=Overtime

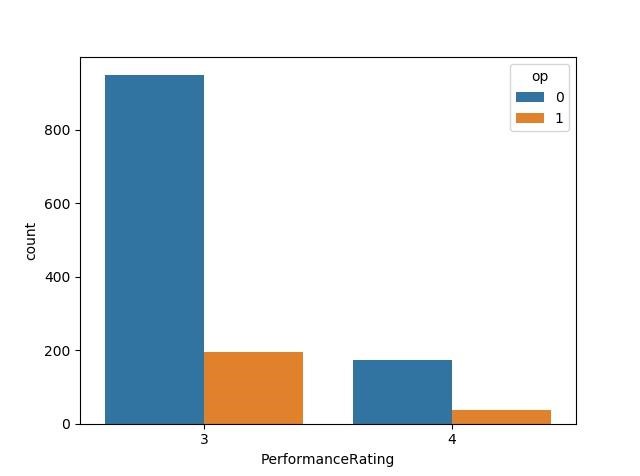
𝑦=Count



#### Countplot of Performance Rating vs count

=Performance Rating

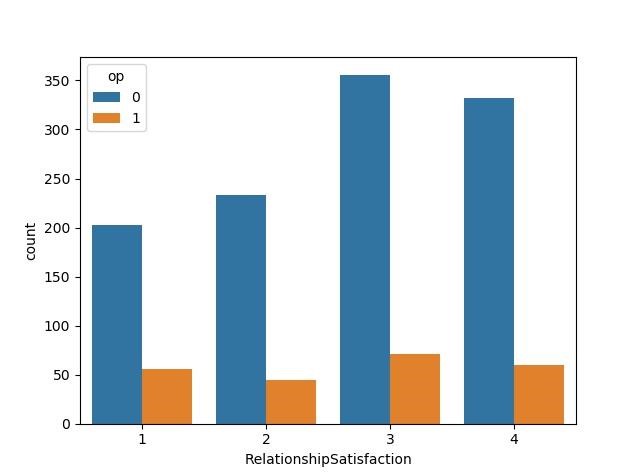
𝑦=Count



#### Countplot of Relationship Satisfaction vs count

=Relationship Satisfaction

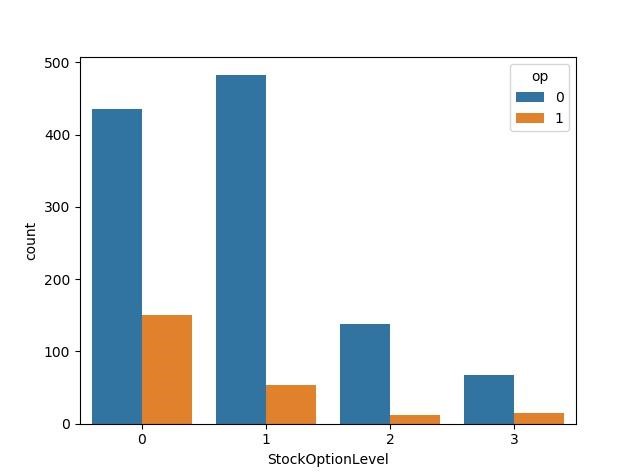
𝑦=Count



#### Countplot of Stock Option Level vs. Count

=Stock Option Level

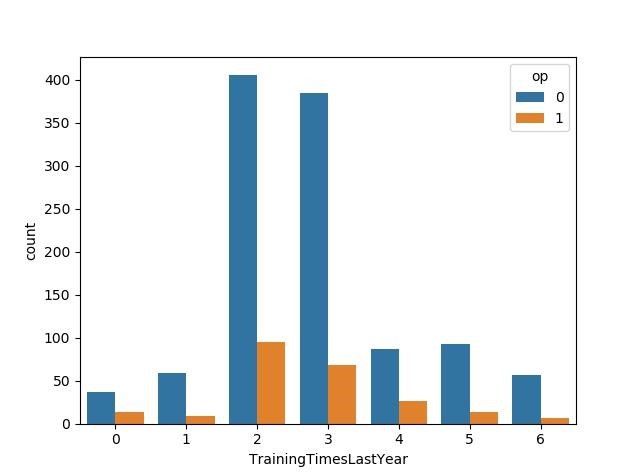
𝑦=Count



#### Countplot of Training Times Last Year vs. Count

X=Training Times Last Year

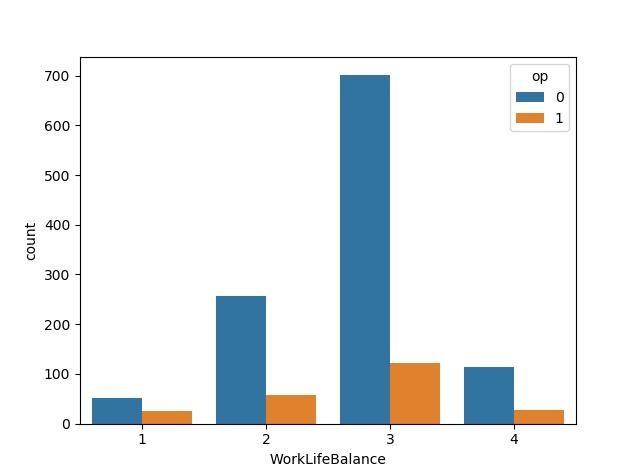
𝑦=Count



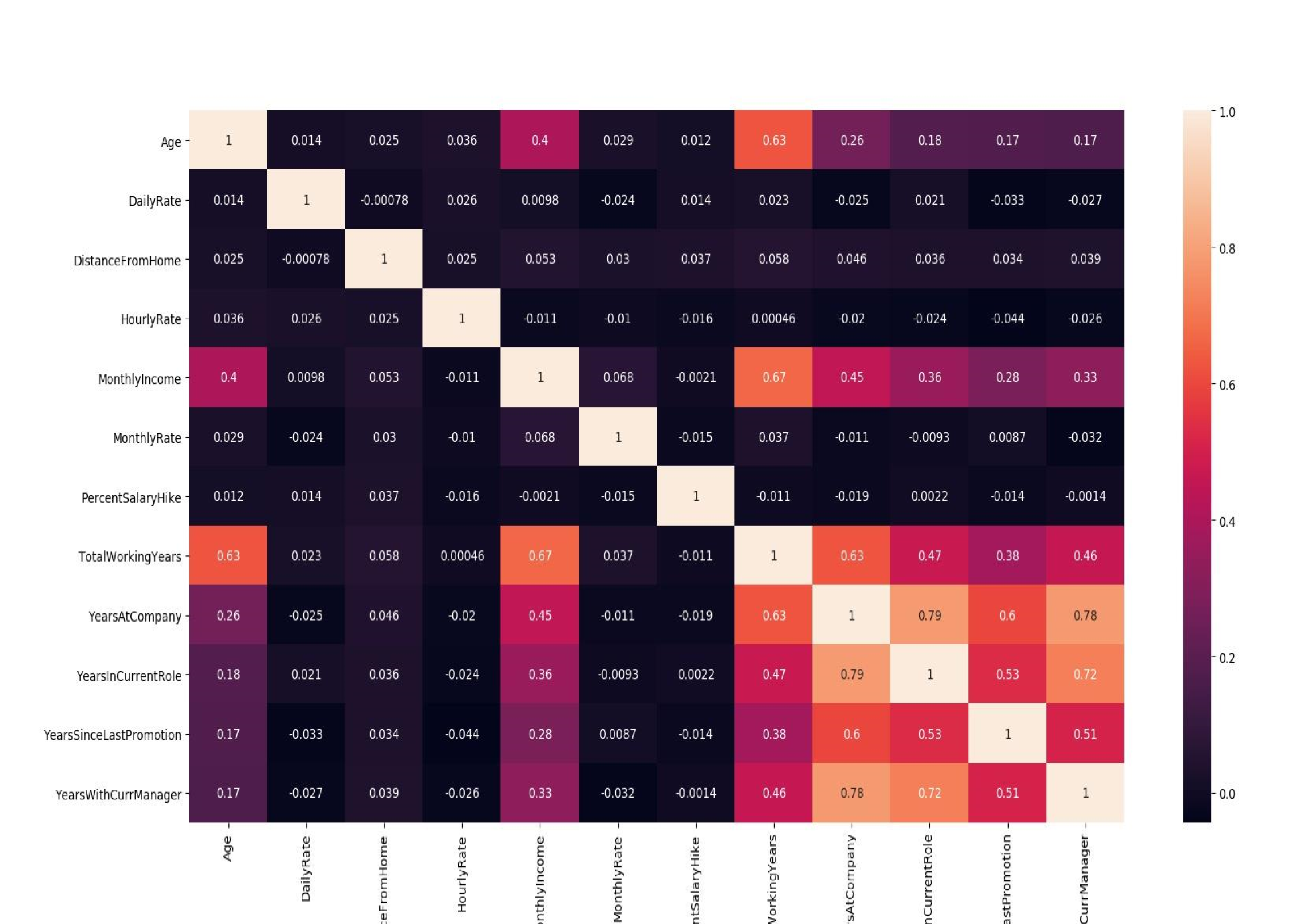
#### Countplot of Work Life Balance vs. Count

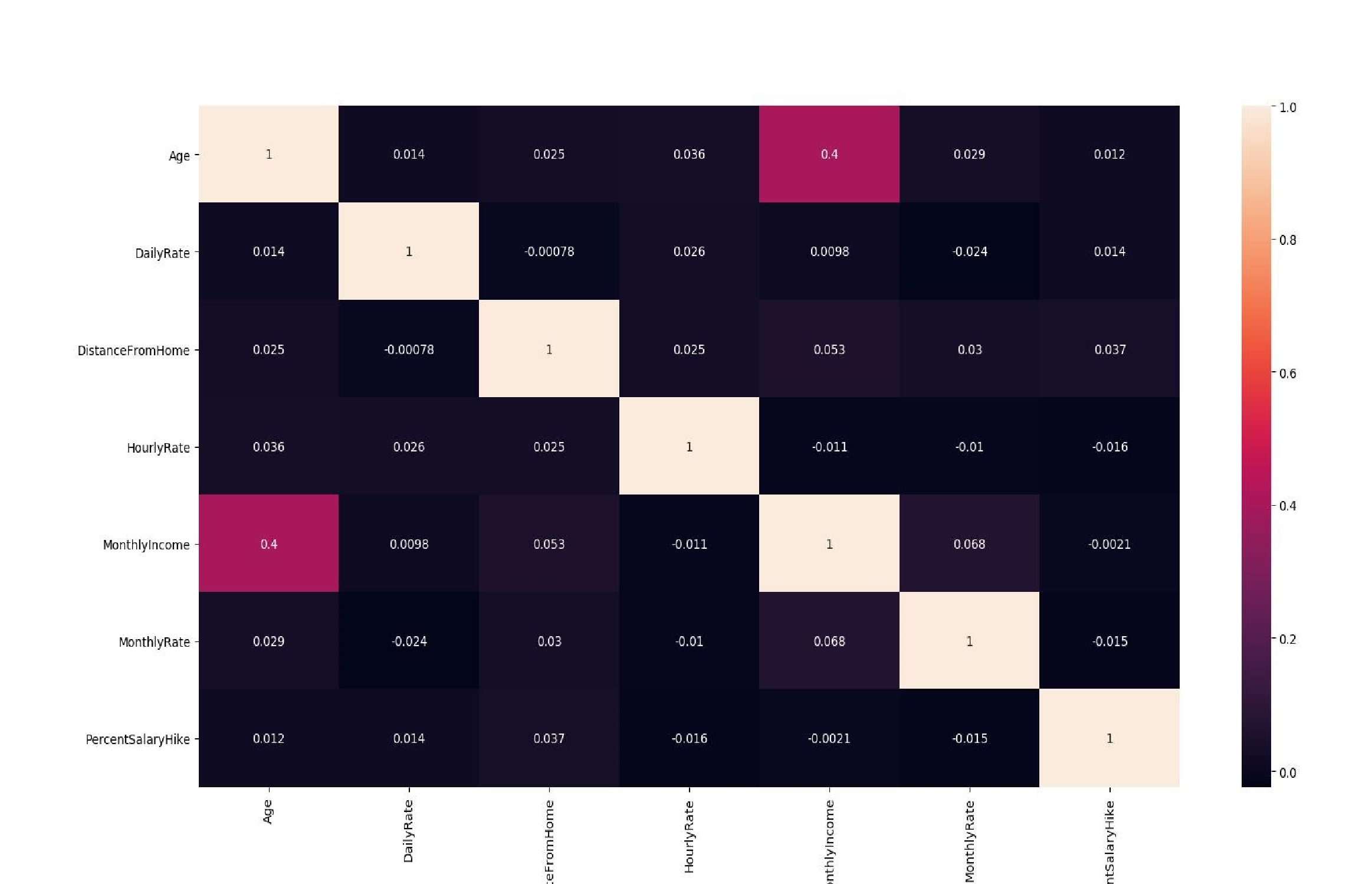
=Work Life Balance

𝑦=Count



**HeatMap before deletion of some columns:**



 **HeatMap after deletion of some columns:**

**Before some columns are deleted, we gather that:**

1.age and total working years are correlated:0.63

2.Monthly income and total working years are correlated:0.67

3.years at company and total working years are correlated: 0.63

4.years at company years, in current role, years since last promotion, years with current manager all 4 are correlated >0.6 coefficient that's why we are deleting 4 continuous columns as

1. they are highly correlated

2.they have too many unique values so we cannot consider them as categorical columns they are:

1. Years at Company
2. Years in Current Role
3. Years Since Last Promotion
4. Years with Current Manager

and converting one continuous column as categorical Years With Current Manager

### Code

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt #ploting, visualisation

import seaborn as sns #ploting

from sklearn import linear\_model

from sklearn.linear\_model import LogisticRegression

from sklearn import tree

from sklearn import neighbors

from sklearn import naive\_bayes

from sklearn import metrics

from sklearn.preprocessing import StandardScaler

from sklearn import model\_selection

from sklearn.feature\_selection import chi2

def modelstats1(Xtrain,Xtest,ytrain,ytest):

stats=[]

modelnames=["LR","DecisionTree","KNN","NB"]

models=list()

models.append(linear\_model.LogisticRegression())

models.append(tree.DecisionTreeClassifier())

models.append(neighbors.KNeighborsClassifier())

models.append(naive\_bayes.GaussianNB())

for name,model in zip(modelnames,models):

if name=="KNN":

k=[l for l in range(5,17,2)]

grid={"n\_neighbors":k}

grid\_obj = model\_selection.GridSearchCV(estimator=model,param\_grid=grid,scoring="f1")

grid\_fit =grid\_obj.fit(Xtrain,ytrain)

model = grid\_fit.best\_estimator\_

model.fit(Xtrain,ytrain)

name=name+"("+str(grid\_fit.best\_params\_["n\_neighbors"])+")"

print(grid\_fit.best\_params\_)

else:

model.fit(Xtrain,ytrain)

trainprediction=model.predict(Xtrain)

testprediction=model.predict(Xtest)

scores=list()

scores.append(name+"-train")

scores.append(metrics.accuracy\_score(ytrain,trainprediction))

scores.append(metrics.precision\_score(ytrain,trainprediction))

scores.append(metrics.recall\_score(ytrain,trainprediction))

scores.append(metrics.roc\_auc\_score(ytrain,trainprediction))

stats.append(scores)

scores=list()

scores.append(name+"-test")

scores.append(metrics.accuracy\_score(ytest,testprediction))

scores.append(metrics.precision\_score(ytest,testprediction))

scores.append(metrics.recall\_score(ytest,testprediction))

scores.append(metrics.roc\_auc\_score(ytest,testprediction))

stats.append(scores)

colnames=["MODELNAME","ACCURACY","PRECISION","RECALL","AUC"]

return pd.DataFrame(stats,columns=colnames)

df\_orig = pd.read\_csv("d:/datasets/employee\_attrition/employeeattrition.csv")

df\_orig.info()

df\_orig.isnull().sum()

col\_drop = []

for col in df\_orig.columns.values :

if df\_orig[str(col)].unique().shape[0] == 1:

col\_drop.append(str(col))

if df\_orig.EmployeeNumber.unique().shape[0] == 1470:

col\_drop.append("EmployeeNumber")

d1={ "Yes":1,"No":0 } # dictionary

d1

df\_orig["Attrition"].replace(d1,inplace=True)

d3={ label:i for i,label in enumerate(df\_orig["BusinessTravel"].unique())} # dictionary

d3

df\_orig["BusinessTravel"].replace(d3,inplace=True)

d4={ label:i for i,label in enumerate(df\_orig["Department"].unique())} # dictionary

d4

df\_orig["Department"].replace(d4,inplace=True)

d5={ label:i for i,label in enumerate(df\_orig["EducationField"].unique())} # dictionary

d5

df\_orig["EducationField"].replace(d5,inplace=True)

d6={ label:i for i,label in enumerate(df\_orig["Gender"].unique())} # dictionary

d6

df\_orig["Gender"].replace(d6,inplace=True)

d7={ label:i for i,label in enumerate(df\_orig["JobRole"].unique())} # dictionary

d7

df\_orig["JobRole"].replace(d7,inplace=True)

d8={ label:i for i,label in enumerate(df\_orig["MaritalStatus"].unique())} # dictionary

d8

df\_orig["MaritalStatus"].replace(d8,inplace=True)

d9={ label:i for i,label in enumerate(df\_orig["Over18"].unique())} # dictionary

d9

df\_orig["Over18"].replace(d9,inplace=True)

d2={ label:i for i,label in enumerate(df\_orig["OverTime"].unique())} # dictionary

d2

df\_orig["OverTime"].replace(d2,inplace=True)

X = df\_orig.drop("Attrition", axis=1)

y = df\_orig["Attrition"]

Xtrain,Xtest,ytrain,ytest = model\_selection.train\_test\_split(X,y,test\_size=.30,random\_state=0)

modelstats1(Xtrain,Xtest,ytrain,ytest)

# MODELNAME ACCURACY PRECISION RECALL AUC

#0 LR-train 0.881438 0.764706 0.389222 0.683010

#1 LR-test 0.879819 0.774194 0.342857 0.661995

#2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

#3 DecisionTree-test 0.798186 0.338983 0.285714 0.590296

#4 KNN(7)-train 0.846453 0.628571 0.131737 0.558328

#5 KNN(7)-test 0.827664 0.250000 0.042857 0.509299

#6 NB-train 0.782313 0.396364 0.652695 0.730060

#7 NB-test 0.791383 0.400000 0.628571 0.725337

corr = df\_orig[:].corr()

corr.style.background\_gradient(cmap='coolwarm')

col\_drop.append("JobLevel")

col\_drop.append("PerformanceRating")

col\_drop.append("YearsAtCompany")

col\_drop.append("YearsInCurrentRole")

col\_drop.append("TotalWorkingYears")

col\_drop.append("PerformanceRating")

#col\_drop.append("Education")

#col\_drop.append("YearsSinceLastPromotion")

col\_drop.append("MonthlyRate")

col\_drop.append("MonthlyIncome")

#col\_drop.append("DailyRate")

#col\_drop.append("HourlyRate")

col\_drop.append("BusinessTravel")

col\_drop = []

df\_orig.drop(col\_drop,axis=1,inplace=True)

cx = df\_orig.drop("Attrition",axis=1)

cy = df\_orig["Attrition"]

chi\_sq = chi2(cx,cy)

chi\_sq

p\_values = pd.Series(chi\_sq[1],index=cx.columns)

p\_values.sort\_values(ascending = False, inplace = True)

p\_values[:].plot(kind="bar")

# MODELNAME ACCURACY PRECISION RECALL AUC

#0 LR-train 0.863946 0.813953 0.209581 0.600150

#1 LR-test 0.859410 0.666667 0.228571 0.603504

#2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

#3 DecisionTree-test 0.782313 0.308824 0.300000 0.586658

#4 KNN(5)-train 0.860058 0.688525 0.251497 0.614728

#5 KNN(5)-test 0.823129 0.318182 0.100000 0.529784

#6 NB-train 0.854227 0.561151 0.467066 0.698150

#7 NB-test 0.863946 0.586207 0.485714 0.710512

X=df\_orig.drop("Attrition",axis=1)

y=df\_orig["Attrition"]

Xtrain,Xtest,ytrain,ytest=model\_selection.train\_test\_split(X,y,test\_size=.2,random\_state=30)

model=linear\_model.LogisticRegression(solver="liblinear",penalty="l1",C=100000)

model.fit(Xtrain,ytrain)

ser=pd.Series(model.coef\_[0])

ser.index=Xtrain.columns.values

ser.abs().sort\_values(ascending=False)

sns.boxplot(x="Attrition",y="Age",data=df\_orig)

sns.boxplot(x="Attrition",y="BusinessTravel",data=df\_orig)

scaler=StandardScaler()

scaled\_df=scaler.fit\_transform(df\_orig[["Age","DailyRate","DistanceFromHome","HourlyRate","MonthlyIncome",

"MonthlyRate","NumCompaniesWorked","PercentSalaryHike","TotalWorkingYears",

"YearsAtCompany","YearsInCurrentRole","YearsSinceLastPromotion","YearsWithCurrManager"]])

scaled\_df = pd.DataFrame(scaled\_df,columns = ["Age","DailyRate","DistanceFromHome","HourlyRate","MonthlyIncome",

"MonthlyRate","NumCompaniesWorked","PercentSalaryHike","TotalWorkingYears",

"YearsAtCompany","YearsInCurrentRole","YearsSinceLastPromotion","YearsWithCurrManager"])

not\_scaled = df\_orig.drop(["Age","DailyRate","DistanceFromHome","HourlyRate","MonthlyIncome",

"MonthlyRate","NumCompaniesWorked","PercentSalaryHike","TotalWorkingYears",

"YearsAtCompany","YearsInCurrentRole","YearsSinceLastPromotion","YearsWithCurrManager"],axis=1)

allcol = not\_scaled.copy()

for col in scaled\_df :

allcol[col] = scaled\_df[col].values

X=allcol.drop("Attrition",axis=1)

y=allcol["Attrition"]

Xtrain,Xtest,ytrain,ytest = model\_selection.train\_test\_split(X,y,test\_size=.20,random\_state=0)

modelstats1(Xtrain,Xtest,ytrain,ytest)

# MODELNAME ACCURACY PRECISION RECALL AUC

#0 LR-train 0.882653 0.765957 0.382979 0.680356

#1 LR-test 0.877551 0.760000 0.387755 0.681633

#2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

#3 DecisionTree-test 0.772109 0.295455 0.265306 0.569388

#4 KNN(5)-train 0.851190 0.724138 0.111702 0.551802

#5 KNN(5)-test 0.826531 0.250000 0.020408 0.504082

#6 NB-train 0.789116 0.401316 0.648936 0.732363

#7 NB-test 0.802721 0.438356 0.653061 0.742857

testcol = allcol.copy()

corr = testcol[:].corr()

corr.style.background\_gradient(cmap='coolwarm')

col\_drop = []

for colm in testcol.columns.values :

if testcol[str(colm)].unique().shape[0] == 1:

col\_drop.append(str(colm))

if testcol.EmployeeNumber.unique().shape[0] == 1470:

col\_drop.append("EmployeeNumber")

testcol.drop(col\_drop,axis=1,inplace=True)

testcol.shape

#col\_drop.append("JobLevel")

col\_drop.append("TotalWorkingYears")

#col\_drop.append("YearsInCurrentRole")

col\_drop.append("YearsAtCompany")

col\_drop.append("PerformanceRating")

#col\_drop.append("BusinessTravel")

#col\_drop.append("Department")

#col\_drop.append("Education")

col\_drop.append("Gender")

col\_drop.append("MaritalStatus")

#col\_drop.append("RelationshipSatisfaction")

pp = testcol.abs()

cx = pp.drop("Attrition",axis=1)

cy = pp["Attrition"]

chi\_sq = chi2(cx,cy)

chi\_sq

p\_values = pd.Series(chi\_sq[1],index=cx.columns)

p\_values.sort\_values(ascending = False, inplace = True)

p\_values[:].plot(kind="bar")

X=testcol.drop("Attrition",axis=1)

y=testcol["Attrition"]

Xtrain,Xtest,ytrain,ytest = model\_selection.train\_test\_split(X,y,test\_size=.20,random\_state=0)

modelstats1(Xtrain,Xtest,ytrain,ytest)

# MODELNAME ACCURACY PRECISION RECALL AUC

#0 LR-train 0.886905 0.839506 0.361702 0.674272

#1 LR-test 0.867347 0.678571 0.387755 0.675510

#2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

#3 DecisionTree-test 0.768707 0.320755 0.346939 0.600000

#4 KNN(5)-train 0.875850 0.862069 0.265957 0.628930

#5 KNN(5)-test 0.829932 0.454545 0.102041 0.538776

#6 NB-train 0.827381 0.467811 0.579787 0.727141

#7 NB-test 0.843537 0.523077 0.693878 0.783673

sns.boxplot(x="Attrition",y="Department",data=df\_orig)

sns.boxplot(x="Attrition",y="Educationfield",data=df\_orig)

sns.boxplot(x="Attrition",y="Education",data=df\_orig)

sns.boxplot(x="Attrition",y="BusinessTravel",data=df\_orig)

sns.boxplot(x="Attrition",y="Gender",data=df\_orig)

sns.boxplot(x="Attrition",y="MaritalStatus",data=df\_orig)

sns.boxplot(x="Attrition",y="RelationshipSatisfaction",data=df\_orig)

**Data Analysis**

|  |  |  |
| --- | --- | --- |
| **Index** | **Column Name** | **Number of unique values** |
| 1 | Age | 43 |
| 2 | Attrition | 2 |
| 3 | Business Travel | 3 |
| 4 | Daily Rate | 886 |
| 5 | Department | 3 |
| 6 | Distance From Home | 29 |
| 7 | Education | 5 |
| 8 | Education Field | 6 |
| 9 | Employee Count | 1 |
| 10 | Employee Number | 1470 |
| 11 | Environment Satisfaction | 4 |
| 12 | Gender | 2 |
| 13 | Hourly Rate | 71 |
| 14 | Job Involvement | 4 |
| 15 | Job Level | 5 |
| 16 | Job Role | 9 |
| 17 | Job Satisfaction | 4 |
| 18 | Marital Status | 3 |
| 19 | Monthly Income | 1349 |
| 20 | Monthly Rate | 1427 |
| 21 | NumCompaniesWorked | 10 |
| 22 | Over 18 | 1 |
| 23 | Overtime | 2 |
| 24 | Percent Salary Hike | 15 |
| 25 | Performance Rating | 2 |
| 26 | Relationship Satisfaction | 4 |
| 27 | Standard Hours | 1 |
| 28 | Stock Option Level | 4 |
| 29 | Total Working Years | 40 |
| 30 | Training Times Last Year | 7 |
| 31 | Work Life Balance | 4 |
| 32 | Years At Company | 37 |
| 33 | Years In Current Role | 19 |
| 34 | Years Since Last Promotion | 16 |
| 35 | Years With Current Manager | 18 |

There are 4 columns which have either only one unique values (Employee Count, Over 18, Standard Hours) and one column having 1470 different values means different values for all columns-Employee Number. These four columns are therefore dropped.

**Distinguishing categorical and continuous columns**

We considered the columns having more unique values than 10 as continuous or else categorical.

Initially, the continuous columns were

* Age
* Daily Rate
* Distance from Home
* Hourly Rate
* Monthly Income
* Monthly Rate
* Percent Salary Hike
* Total Working Years
* Years At Company
* Years In Current Role
* Years Since Last Promotion
* Years with Current Manager and categorical columns were

1. String categorical
   * Business Travel  Department
   * Education Field
   * Gender
   * Job Role
   * Marital Status
   * Overtime
2. Numeric categorical
   * Education
   * Environment Satisfaction
   * Job Involvement
   * Job Level
   * Job Satisfaction
   * NumCompaniesWorked
   * Performance Rating
   * Relationship Satisfaction
   * Stock Option Level
   * Training Times Last Year
   * Work Life Balance

**Zero or negative value checking:**

Further changes and tests are made later like checking for zero or negative values and we find that-

|  |  |  |
| --- | --- | --- |
| **Index** | **Column Name** | **Number of Zeroes** |
| 1 | Total Working Years | 11 |
| 2 | Years At Company | 44 |
| 3 | Years In Current Role | 244 |
| 4 | Years Since Last Promotion | 581 |
| 5 | Years With Current Manager | 263 |

After further investigating zeroes and checking the minimum, maximum and number of zeros in the particular columns which contains zeroes, we get

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index** | **Column Name** | **No. of zeroes** | **Maximum** | **Minimum** |
| 1 | Total Working Years | 40 | 40 | 0 |
| 2 | Years At Company | 37 | 40 | 0 |
| 3 | Years In Current Role | 19 | 18 | 0 |
| 4 | Years Since Last Promotion | 16 | 15 | 0 |
| 5 | Years With Current Manager | 18 | 17 | 0 |

As none of these are odd values, so we do not drop these values

**Outliers checking:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Inde x** | **Column Name** | **Upper**  **Outlier**  **s** | **Lower**  **Outlier**  **s** | **Theoretic**  **al**  **Maximu m** | **Maximu m** | **Theoretic**  **al**  **Minimum** | **Minimu m** |
| 1 | Age | 0 | 0 | 62.5 | 60 | 10.5 | 18 |
| 2 | Daily Rate | 0 | 0 | 2195 | 1499 | -573 | 102 |
| 3 | Distance from  Home | 0 | 0 | 32 | 29 | -16 | 1 |
| 4 | Hourly Rate | 0 | 0 | 137.375 | 100 | -5.625 | 30 |
| 5 | Monthly Income | 114 | 0 | 16581 | 19999 | -5291 | 1009 |
| 6 | Monthly Rate | 0 | 0 | 39083.25 | 26999 | -10574.75 | 2094 |
| 7 | Percent  Salary  Hike | 0 | 0 | 27 | 25 | 3 | 11 |
| 8 | Total  Working Years | 63 | 0 | 28.5 | 40 | -7.5 | 0 |
| 9 | Years At  Compan  y | 104 | 0 | 18 | 40 | -6 | 0 |
| 10 | Years In  Current Role | 21 | 0 | 14.5 | 18 | -5.5 | 0 |
| 11 | Years  Since  Last  Promotio  n | 107 | 0 | 7.5 | 15 | -4.5 | 0 |
| 12 | Years  With  Current  Manager | 14 | 0 | 14.5 | 17 | -5.5 | 0 |

Except Monthly Income columns in other columns which contains upper outliers, those values may not be considered as outliers because they are not anomalies and are totally acceptable values.

So only outliers of Monthly Income values are deleted.

**Checking correlation between continuous data**:

1.age and total working years are correlated:0.63

2.monthly income and total working years are correlated:0.67

3.years at company and total working years are correlated: 0.63

4.years at company years, in current role, years since last promotion, years with current manager all 4 are correlated >0.6 coefficient

That is why we are deleting 4 continuous columns as

1.they are highly correlated

2.they have too many unique values so we cannot consider them as categorical columns

They are:

1. Years At Company
2. Years In Current Role
3. Years Since Last Promotion 4. Years With Current Manager and converting one continuous column as Categorical-Years With Current Manager

Investigating over different categories of categorical columns:

**The percentage of yes and no on total dataset is**

yes:0.1710914454277286

no: 0.8289085545722714

**Checking yes and no for different categories of all categorical columns to find anomaly:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Category name** | **% of yes** | **% of no** |
| Education | 1 | 0.1875 | 0.8125 |
| Education | 2 | 0.1699 | 0.8301 |
| Education | 3 | 0.1825 | 0.8174 |
| Education | 4 | 0.1549 | 0.8451 |
| Education | 5 | 0.1162 | 0.8837 |
| Environment Satisfaction | 1 | 0.2671 | 0.7328 |
| Environment Satisfaction | 2 | 0.1567 | 0.8432 |
| Environment Satisfaction | 3 | 0.1476 | 0.8523 |
| Environment Satisfaction | 4 | 0.1428 | 0.8571 |
| Job Involvement | 1 | 0.3376 | 0.6623 |
| Job Involvement | 2 | 0.2034 | 0.7965 |
| Job Involvement | 3 | 0.1536 | 0.8463 |
| Job Involvement | 4 | 0.0958 | 0.9044 |
| Job Level | 1 | 0.2633 | 0.7366 |
| Job Level | 2 | 0.0973 | 0.9026 |
| Job Level | 3 | 0.1467 | 0.8532 |
| Job Level | 4 | 0.0819 | 0.9180 |
| Job Satisfaction | 1 | 0.2462 | 0.7537 |
| Job Satisfaction | 2 | 0.1705 | 0.8294 |
| Job Satisfaction | 3 | 0.1769 | 0.8230 |
| Job Satisfaction | 4 | 0.1194 | 0.8805 |
| NumCompaniesWorked | 0 | 0.1229 | 0.8770 |
| NumCompaniesWorked | 1 | 0.1935 | 0.8064 |
| NumCompaniesWorked | 2 | 0.1119 | 0.8880 |
| NumCompaniesWorked | 3 | 0.1159 | 0.8840 |
| NumCompaniesWorked | 4 | 0.1370 | 0.8629 |
| NumCompaniesWorked | 5 | 0.2631 | 0.7368 |
| NumCompaniesWorked | 6 | 0.2539 | 0.7460 |
| NumCompaniesWorked | 7 | 0.2424 | 0.7575 |
| NumCompaniesWorked | 8 | 0.1363 | 0.8636 |
| NumCompaniesWorked | 9 | 0.2553 | 0.7446 |
| Performance Rating | 3 | 0.1701 | 0.8298 |
| Performance Rating | 4 | 0.1761 | 0.8238 |
| Relationship Satisfaction | 1 | 0.2162 | 0.7837 |
| Relationship Satisfaction | 2 | 0.1618 | 0.8381 |

|  |  |  |  |
| --- | --- | --- | --- |
| Relationship Satisfaction | 3 | 0.1662 | 0.8337 |
| Relationship Satisfaction | 4 | 0.1530 | 0.8469 |
| Stock Option Level | 0 | 0.2572 | 0.7427 |
| Stock Option Level | 1 | 0.1005 | 0.8994 |
| Stock Option Level | 2 | 0.08 | 0.92 |
| Stock Option Level | 3 | 0.1829 | 0.8170 |
| Training Times Last Year | 0 | 0.2745 | 0.7254 |
| Training Times Last Year | 1 | 0.1323 | 0.8676 |
| Training Times Last Year | 2 | 0.1896 | 0.8103 |
| Training Times Last Year | 3 | 0.1501 | 0.8498 |
| Training Times Last Year | 4 | 0.2300 | 0.7699 |
| Training Times Last Year | 5 | 0.1308 | 0.8691 |
| Training Times Last Year | 6 | 0.0952 | 0.9047 |
| Work Life Balance | 1 | 0.3246 | 0.6753 |
| Work Life Balance | 2 | 0.1841 | 0.8158 |
| Work Life Balance | 3 | 0.1480 | 0.8519 |
| Work Life Balance | 4 | 0.1928 | 0.8071 |
| Years With Current Manager | 0 | 0.3386 | 0.6613 |
| Years With Current Manager | 1 | 01617 | 0.2382 |
| Years With Current Manager | 2 | 0.1501 | 0.8498 |
| Years With Current Manager | 3 | 0.1407 | 0.8592 |
| Years With Current Manager | 4 | 0.1063 | 0.8936 |
| Years With Current Manager | 5 | 0.1333 | 0.8666 |
| Years With Current Manager | 6 | 0.1739 | 0.8260 |
| Years With Current Manager | 7 | 0.155 | 0.845 |
| Years With Current Manager | 8 | 0.08247 | 0.9175 |
| Years With Current Manager | 9 | 0.098 | 0.9019 |
| Years With Current Manager | 10 | 0.1578 | 0.8421 |
| Years With Current Manager | 11 | 0.0 | 1.0 |
| Years With Current Manager | 12 | 0.0 | 1.0 |
| Years With Current Manager | 13 | 0.0 | 1.0 |
| Years With Current Manager | 14 | 0.6666 | 0.333 |
| Years With Current Manager | 15 | 0.0 | 1.0 |
| Years With Current Manager | 17 | 0.0 | 1.0 |
| Business Travel | Non-Travel | 0.0863 | 0.9136 |
| Business Travel | Travel Frequently | 0.2633 | 0.7366 |
| Business Travel | Travel Rarely | 0.15811 | 0.8418 |
| Department | Human Resources | 0.2222 | 0.7777 |
| Department | R&D | 0.1475 | 0.8524 |
| Department | Sales | 0.2137 | 0.7862 |
| Education Field | Human Resources | 0.3333 | 0.6666 |
| Education Field | Life Sciences | 0.1571 | 0.8428 |
| Education Field | Marketing | 0.2229 | 0.777 |
| Education Field | Medical | 0.1462 | 0.8537 |
| Education Field | Other | 0.1447 | 0.8552 |
| Education Field | Technical | 0.2440 | 0.7559 |
| Gender | Male | 0.1541 | 0.8458 |
| Gender | Female | 0.1824 | 0.8175 |
| Job Role | Healthcare Representative | 0.0687 | 0.9313 |
| Job Role | Human Resource | 0.2307 | 0.7692 |
| Job Role | Laboratory Technician | 0.2393 | 0.7606 |
| Job Role | Manager | 0.0714 | 0.9285 |
| Job Role | Manufacturing Director | 0.068 | 0.9285 |
| Job Role | Research Director | 0.0 | 1.0 |
| Job Role | Sales Representative | 0.3975 | 0.6024 |
| Job Role | Sales Executive | 0.1748 | 0.8251 |
| Job Role | Research Scientist | 0.1609 | 0.8390 |
| Marital Status | Divorced | 01609 | 0.839 |
| Marital Status | Married | 0.1748 | 0.8251 |
| Marital Status | Single | 0.3975 | 0.6024 |
| Overtime | No | 0.1113 | 0.8886 |
| Overtime | Yes | 0.3212 | 0.6787 |

**Anomaly from above observations:**

* 1. People who has overtime are more prone to leave the company, the ratio becomes 1:2 in this case
  2. People having job role as research director does not leave company

* 1. People who are working for current manager for 11 12 13 15 16 17 years that means for a long time generally don’t leave the company except current manager year 14 where this ratio is drastically opposite 2:1

* 1. People who has Work Life balance status as 1 generally more prone to leave the company

The ratio is 1:2 for no attrition vs attrition

* 1. people having human resources as education field are more prone to leave the company

The ratio is again 1:2

6.people in manager or manufacturing director post has very less tendency to leave, the ratio is almost 1:10

**Categorical data are one hot encoded. After one hot encoding number of total columns become 105.**

**Feature scaling is applied on the**

**Applying different Tree Classifier and getting feature importance:**

Among 3 applied tree classifiers

1.Decision tree

2.Random forest

3.Extra tree classifier

**Applying different model classification algorithm to get result**

1. Logistic regression
2. Decision tree classifier
3. Random forest classifier

3. K-nearest neighbours 4. Naive bayes **Test 1:**

We applied decision tree classifier for feature importance and passed the highest important 10 columns for these 4 algorithms.

**Output:**

Features passed: 'DistanceFromHome', 'Age', 'OverTime\_No', 'JobLevel\_1',

'MonthlyRate', 'MonthlyIncome', 'YearsWithCurrManager\_0',

'StockOptionLevel\_0', 'DailyRate', 'PercentSalaryHike'

**MODELNAME ACCURACY PRECISION RECALL AUC**

0 Logitic Regression-train 0.886905 0.839506 0.361702 0.674272

1 Logistic Regression-test 0.867347 0.678571 0.387755 0.675510

2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

3 DecisionTree-test 0.768707 0.320755 0.346939 0.600000

4 KNN(5)-train 0.875850 0.862069 0.265957 0.628930

5 KNN(5)-test 0.829932 0.454545 0.102041 0.538776

6 NB-train 0.827381 0.467811 0.579787 0.727141

7 NB-test 0.843537 0.523077 0.693878 0.783673

**Test 2:**

We applied random forest to measure feature importance and again passed 10 highest important columns

**Output:**

Columns: 'MonthlyIncome', 'Age', 'MonthlyRate', 'OverTime\_Yes',

'OverTime\_No', 'DistanceFromHome', 'DailyRate', 'HourlyRate', 'JobLevel\_1', 'BusinessTravel\_Travel\_Frequently'

**MODELNAME ACCURACY PRECISION RECALL AUC**

0 Logitic Regression-train 0.886905 0.839506 0.361702 0.674272

1 Logistic Regression-test 0.867347 0.678571 0.387755 0.675510

2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

3 DecisionTree-test 0.768707 0.320755 0.346939 0.600000

4 KNN(5)-train 0.875850 0.862069 0.265957 0.628930

5 KNN(5)-test 0.829932 0.454545 0.102041 0.538776

6 NB-train 0.827381 0.467811 0.579787 0.727141

7 NB-test 0.843537 0.523077 0.693878 0.783673

**Test 3:::**

**Output:**

**MODELNAME ACCURACY PRECISION RECALL AUC**

0 Logitic Regression-train 0.886905 0.839506 0.361702 0.674272

1 Logistic Regression-test 0.867347 0.678571 0.387755 0.675510

2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

3 DecisionTree-test 0.768707 0.320755 0.346939 0.600000

4 KNN(5)-train 0.875850 0.862069 0.265957 0.628930

5 KNN(5)-test 0.829932 0.454545 0.102041 0.538776

6 NB-train 0.827381 0.467811 0.579787 0.727141

7 NB-test 0.843537 0.523077 0.693878 0.783673

Accuracy and AUC value is high but recall is not much

**Inference from these 3 tests:**

Number of columns are all total 105, so only passing 10 columns will not help incrementing recall, AUC and Accuracy is already pretty high, we need to focus on recall

**Test 4:**

Applying all 4 algorithms on highest 30 columns using feature importance as decision tree classifier Output:

Name of 30 columns those are used:

'DistanceFromHome', 'Age', 'OverTime\_No', 'JobLevel\_1',

'MonthlyRate', 'MonthlyIncome', 'YearsWithCurrManager\_0',

'StockOptionLevel\_0', 'DailyRate', 'PercentSalaryHike',

'HourlyRate', 'WorkLifeBalance\_2', 'WorkLifeBalance\_1',

'JobSatisfaction\_4', 'JobRole\_Sales Executive',

'MaritalStatus\_Single', 'EducationField\_Technical Degree',

'NumCompaniesWorked\_7', 'RelationshipSatisfaction\_1',

'TrainingTimesLastYear\_4', 'WorkLifeBalance\_4', 'JobInvolvement\_1',

'RelationshipSatisfaction\_3', 'RelationshipSatisfaction\_4',

'NumCompaniesWorked\_3', 'NumCompaniesWorked\_0',

'TrainingTimesLastYear\_1', 'JobRole\_Sales Representative',

'JobRole\_Laboratory Technician', 'JobRole\_Human Resources' **MODELNAME ACCURACY PRECISION RECALL AUC**

0 Logitic Regression-train 0.886905 0.839506 0.361702 0.674272

1 Logistic Regression-test 0.867347 0.678571 0.387755 0.675510

2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

3 DecisionTree-test 0.768707 0.320755 0.346939 0.600000

4 KNN(5)-train 0.875850 0.862069 0.265957 0.628930

5 KNN(5)-test 0.829932 0.454545 0.102041 0.538776

6 NB-train 0.827381 0.467811 0.579787 0.727141

7 NB-test 0.843537 0.523077 0.693878 0.783673

**Test 5:**

Applying all 4 algorithms on highest 30 columns using feature importance as random forest

Classifier

**Output:**

Column names: 'MonthlyIncome', 'Age', 'MonthlyRate', 'OverTime\_Yes',

'OverTime\_No', 'DistanceFromHome', 'DailyRate', 'HourlyRate',

'JobLevel\_1', 'BusinessTravel\_Travel\_Frequently',

'PercentSalaryHike', 'JobRole\_Sales Representative',

'MaritalStatus\_Single', 'StockOptionLevel\_1',

'YearsWithCurrManager\_0', 'EducationField\_Medical',

'EnvironmentSatisfaction\_3', 'JobSatisfaction\_1',

'NumCompaniesWorked\_1', 'WorkLifeBalance\_1', 'WorkLifeBalance\_2',

'Education\_4', 'TrainingTimesLastYear\_3',

'EnvironmentSatisfaction\_1', 'WorkLifeBalance\_3',

'TrainingTimesLastYear\_2', 'JobLevel\_2',

'RelationshipSatisfaction\_1', 'Gender\_Male',

'RelationshipSatisfaction\_2'

**MODELNAME ACCURACY PRECISION RECALL AUC**

0 Logitic Regression-train 0.886905 0.839506 0.361702 0.674272

1 Logistic Regression-test 0.867347 0.678571 0.387755 0.675510

2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

3 DecisionTree-test 0.768707 0.320755 0.346939 0.600000

4 KNN(5)-train 0.875850 0.862069 0.265957 0.628930

5 KNN(5)-test 0.829932 0.454545 0.102041 0.538776

6 NB-train 0.827381 0.467811 0.579787 0.727141

7 NB-test 0.843537 0.523077 0.693878 0.783673

**Test 6:**

Using extra tree classifier for feature importance and passing highest 30 columns

**Output:**

Columns:

‘OverTime\_No', 'OverTime\_Yes', 'JobLevel\_1', 'Age',

'YearsWithCurrManager\_0', 'MaritalStatus\_Single',

'DistanceFromHome', 'MonthlyIncome', 'HourlyRate',

'BusinessTravel\_Travel\_Frequently', 'DailyRate',

'EnvironmentSatisfaction\_3', 'MonthlyRate',

'EducationField\_Medical', 'JobSatisfaction\_1',

'EnvironmentSatisfaction\_1', 'WorkLifeBalance\_2',

'RelationshipSatisfaction\_1', 'Gender\_Female',

'RelationshipSatisfaction\_4', 'JobSatisfaction\_2',

'Department\_Research & Development', 'PercentSalaryHike',

'EnvironmentSatisfaction\_4', 'EducationField\_Marketing',

'JobSatisfaction\_3', 'JobInvolvement\_3', 'NumCompaniesWorked\_1',

'Education\_3', 'JobRole\_Research Scientist'

**MODELNAME ACCURACY PRECISION RECALL AUC**

0 Logitic Regression-train 0.886905 0.839506 0.361702 0.674272

1 Logistic Regression-test 0.867347 0.678571 0.387755 0.675510

2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

3 DecisionTree-test 0.768707 0.320755 0.346939 0.600000

4 KNN(5)-train 0.875850 0.862069 0.265957 0.628930

5 KNN(5)-test 0.829932 0.454545 0.102041 0.538776

6 NB-train 0.827381 0.467811 0.579787 0.727141

7 NB-test 0.843537 0.523077 0.693878 0.783673

**Inference from test 4, 5, 6:**

1.Extra tree classifier gives better feature importance than random forest and decision tree classifier

2.Recall is slightly underfitted, so we need to increase number of features a bit, but accuracy and AUC score is not bad

**Test 7:**

Using Extra Tree classifier to get feature importance and passing total 35 columns to get the result

**Output:**

Columns are

'OverTime\_Yes', 'OverTime\_No', 'MonthlyIncome',

'YearsWithCurrManager\_0', 'DistanceFromHome',

'MaritalStatus\_Single', 'HourlyRate', 'JobLevel\_1',

'JobSatisfaction\_1', 'MonthlyRate', 'PercentSalaryHike',

'JobLevel\_2', 'BusinessTravel\_Travel\_Frequently', 'Age',

'StockOptionLevel\_1', 'RelationshipSatisfaction\_1',

'StockOptionLevel\_0', 'TrainingTimesLastYear\_3',

'RelationshipSatisfaction\_3', 'JobRole\_Laboratory Technician',

'EnvironmentSatisfaction\_3', 'EnvironmentSatisfaction\_1',

'JobSatisfaction\_4', 'DailyRate', 'WorkLifeBalance\_3',

'Gender\_Male', 'WorkLifeBalance\_1', 'EducationField\_Medical',

'BusinessTravel\_Travel\_Rarely', 'WorkLifeBalance\_2',

'EnvironmentSatisfaction\_4', 'EducationField\_Life Sciences',

'JobSatisfaction\_3', 'NumCompaniesWorked\_1',

'JobRole\_Sales Representative'

MODELNAME ACCURACY PRECISION RECALL AUC

0 Logitic Regression-train 0.886905 0.839506 0.361702 0.674272

1 Logistic Regression-test 0.867347 0.678571 0.387755 0.675510

2 DecisionTree-train 1.000000 1.000000 1.000000 1.000000

3 DecisionTree-test 0.768707 0.320755 0.346939 0.600000

4 KNN(5)-train 0.875850 0.862069 0.265957 0.628930

5 KNN(5)-test 0.829932 0.454545 0.102041 0.538776

6 NB-train 0.827381 0.467811 0.579787 0.727141

7 NB-test 0.843537 0.523077 0.693878 0.783673

**Inference:**

Naïve bayes showed the best results

### CONCLUSION

1.Used naïve bayes classification algorithm

2.Using extra tree classifier decided highest important features

3.Total 35 columns out of 105 features (that is present after one hot encode of the categorical columns) are used

4.Name of the columns:

1.'OverTime\_Yes'

1. 'OverTime\_No'
2. 'MonthlyIncome'
3. 'YearsWithCurrManager\_0'
4. 'DistanceFromHome'
5. 'MaritalStatus\_Single'
6. 'HourlyRate'

8.'JobLevel\_1',

9.'JobSatisfaction\_1'

1. 'MonthlyRate'
2. 'PercentSalaryHike',

12.'JobLevel\_2'

13. 'BusinessTravel\_Travel\_Frequently'

14.'Age'

15.'StockOptionLevel\_1'

16.'RelationshipSatisfaction\_1',

17.'StockOptionLevel\_0'

18.'TrainingTimesLastYear\_3'

19.'RelationshipSatisfaction\_3'

20.'JobRole\_Laboratory Technician'

21. 'EnvironmentSatisfaction\_3'

22.'EnvironmentSatisfaction\_1',

23.'JobSatisfaction\_4'

24.'DailyRate'

25.'WorkLifeBalance\_3'

26 .'Gender\_Male'

27.'WorkLifeBalance\_1'

28.'EducationField\_Medical'

29.'BusinessTravel\_Travel\_Rarely'

30.'WorkLifeBalance\_2',

31.'EnvironmentSatisfaction\_4'

32.'EducationField\_Life Sciences',

33.'JobSatisfaction\_3'

34.'NumCompaniesWorked\_1',

35.'JobRole\_Sales Representative'

**Results**

1. Accuracy:81%(over test data)
2. Recall score:65.3%(over test data)
3. AUC value:0.74.2(over test data) ROC\_AUC curve:

