RIT | Rochester Institute of Technology of Dubai

Project Title:

HR Attrition in IBM

Course Name:

ISTE 600-Foundations of Data Mining

Group members:

Ayad Aziz - UID: 381004135

Chrissie Aldo - UID: 781004375

Hameed Al-Obaidi – UID: 355005832

Hind Elessa – UID: 781003285

Tariq Najar – UID:374003643

Toka Khalil – UID: 577008273

Submitted to:

Dr. Michael McQuaid

Table of Contents

Note: Bookmarks are present to reference the code for the visualization, data processing, and data mining techniques sections.

Executive Summary:

RIT Tigers have worked on the dataset extracted from Kaggle to help IBM HR with some critical tasks to optimize the company's running smoothly. Profitability throughout, not limited to the following Firstly, finding out the trends, Secondly, finding out the reasons that cause employee attrition and proposing valuable solutions that HR can implement, Thirdly, helping to study the impact of various factors on employee attrition, which had become a significant part of data/HR analytics and finally building a model to predict if the employee will be attritted or not based on the historical dataset. Our primary purpose of implementing the Data analysis in the business model at IBM is to help them attain their goals of reducing costs by identifying more efficient ways of doing business and by storing large amounts of data.

We have used IBM data mining life cycle CRISP-DM methodology in our project. From the beginning, we had a deep understanding of the problem from the business perspective; then, we had a deep understanding and analysis of our dataset using data mining techniques in terms of data exploration, and visualization of the dataset attributes, specifically our class attribute (employee attrition) vs. essential attributes. Then, we conducted some data preprocessing steps to prepare our dataset, such as data cleaning and dimensionality reduction, so we were able to identify the missing values in attributes and identify outliers to deal with both issues correctly. We also deployed the variance inflation factor (VIF) function in R that helped us in determining which visualizations to do if an attribute is necessary. Fourth, we built various models using a classification data mining technique and a supervised machine learning algorithms such as decision tree, random forest, and Naive Bayes. We have used some data mining/modeling programming (software), such as Weka, R and Python. Fifth, we split our dataset into training and testing (70/30); we have used some evaluation metrics such as accuracy and cost. We found that the random forest had the highest accuracy, 89.62%, and the least cost, $2,735. Our team believes that we are very confident and that our model is ready to be deployed in the company and perhaps further. We had mentioned in the report exciting findings in conclusion and critical lessons learned by the team. For example, the importance of our instructor's deep involvement from the start to the end in optimizing our report quality and the team's time and effort. We have also mentioned the five developmental stages of our team during the project.

Problem Description

International Business Machines Corporation (IBM) is an American multinational technology corporation headquartered in Armonk, New York, with operations in over 171 countries. IBM has a large and diverse portfolio of products and services. As of 2016, these offerings fall into the categories of cloud computing, artificial intelligence, commerce, data and analytics, Internet of things (IoT), IT infrastructure, mobile, digital workplace, and cybersecurity. IBM has one of the largest workforces in the world, and employees at Big Blue are referred to as "IBMers." As there are a large number of IBMers, HR attrition is also inevitable, and IBM needs to work on this aspect to ensure that IBM achieves corporate objectives. Therefore, RIT Tigers decided to help IBM HR implement strategies and preventive actions by building a model that could help in reducing the attrition rate and finally budget on an annual basis.

Employee Attrition

Employee attrition is defined as the process where employees decide to leave the workplace due to a desire to resign for different reasons, such as personal reasons or retirement, and will not be replaced immediately.

What Are the Different Types of Attrition?

There are five types of employee attrition that you need to be aware of:

1. Attrition due to retirement.

2. Voluntary attrition

3. Involuntary attrition

4. Internal attrition

5. Demographic-specific Attrition

What are the costs of employee attrition?

Expenses related to the sourcing and recruiting of talent, including interns, co-ops, and graduates, or even head-hunting someone to fill full-time positions.

Sourcing and HR Processing

Top costs included are

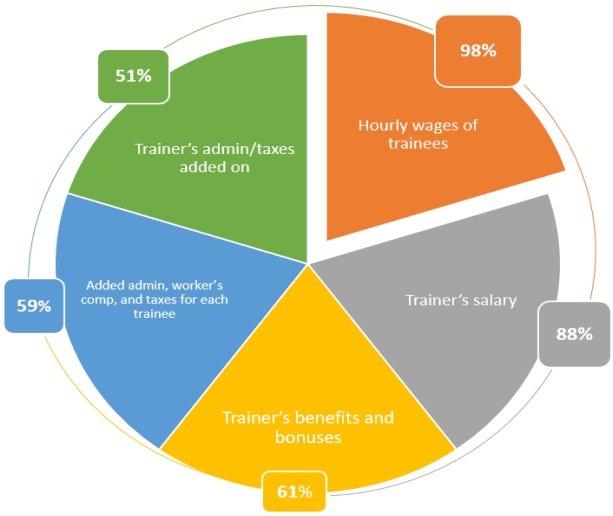
Training

In addition to the cost of filling the vacant positions, The majority of the staff hired will require a minimum period of training and adapting to the new workplace, a duration that varies based on the position filled.

TRAINING

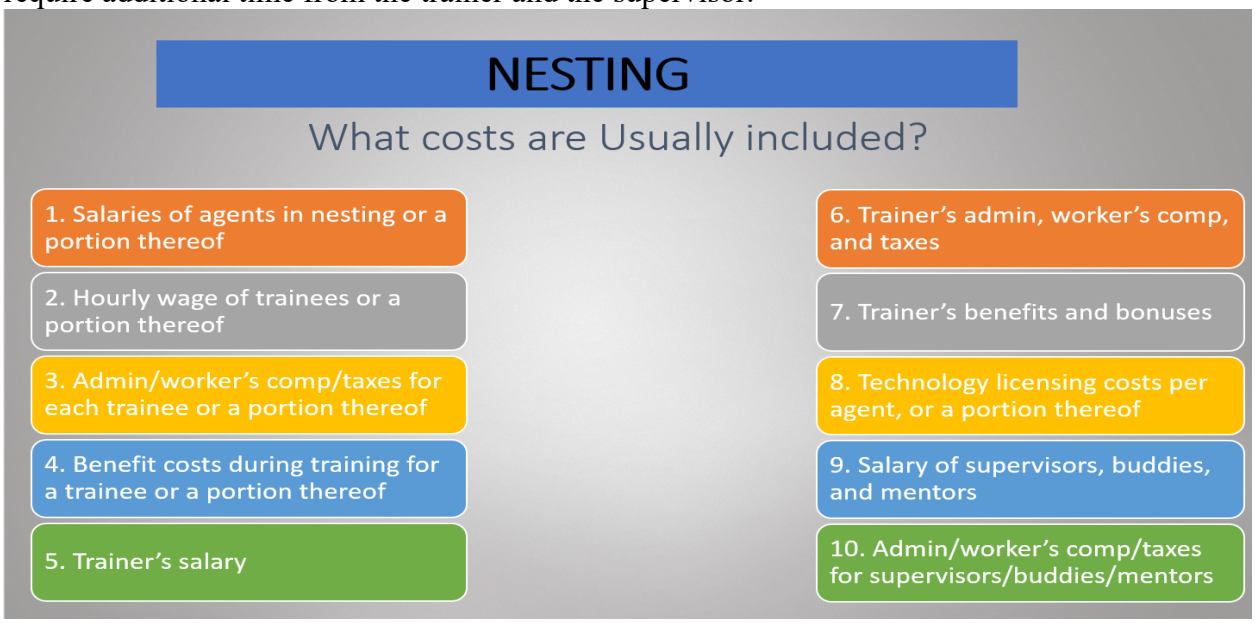
Over 63% have training that lasts close to 2 weeks or more

5 Most Common Training Costs



Nesting

The transition duration of a new hire after the time of his or her training and they start applying their knowledge and skills gained but they still require supervision and observation which will require additional time from the trainer and the supervisor.



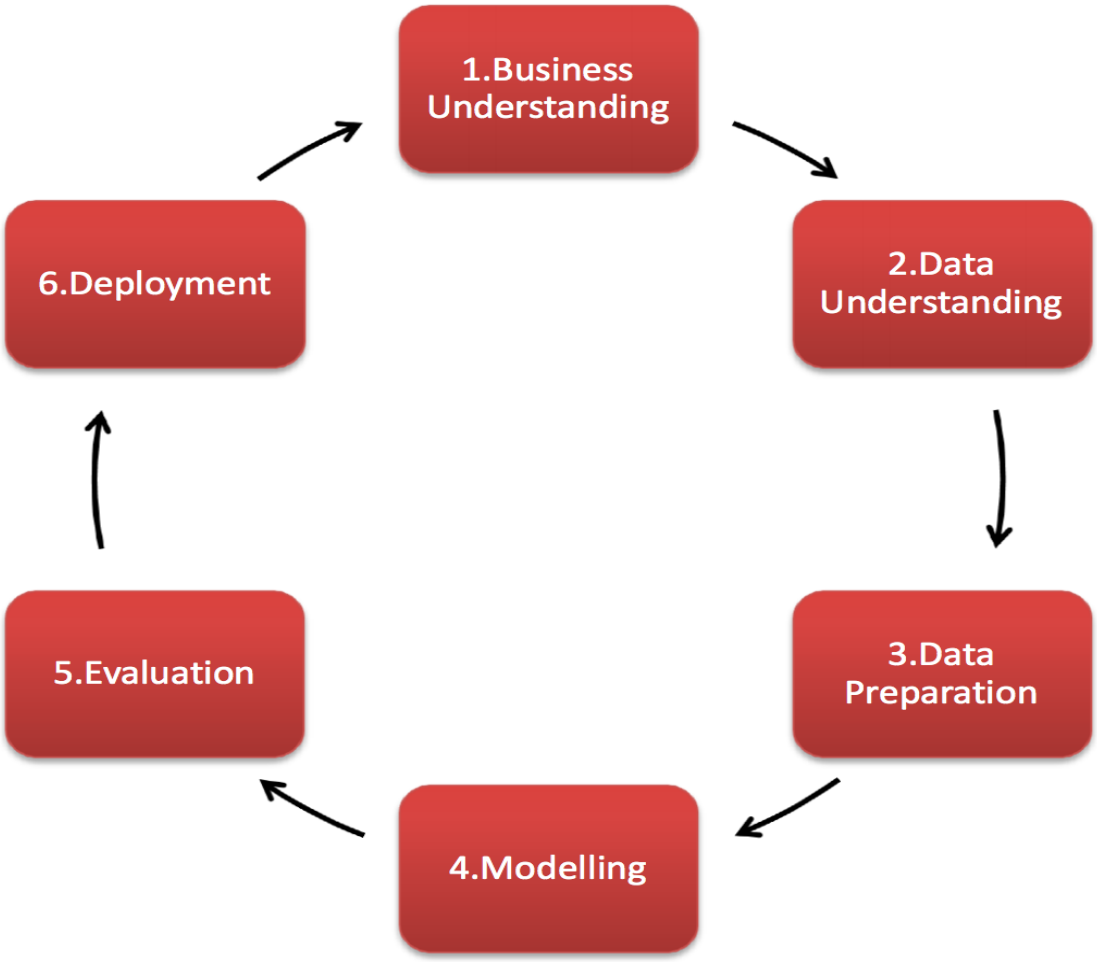
The importance of solving the HR attrition problem at IBM can be summarized as below:

- It is expensive to find, hire and train new talents; therefore, it is cost-effective to keep the people. The company must keep that running smoothly and profitably.
- HR is trying to be proactive and involved in an organization's planning and objectives as this is the core of HR analytics tasks.

The RIT Tigers are planning to propose the possible solutions below to help HR at IBM achieve their departmental objectives.

- Identify the key factors that cause employee attrition and provide them to HR to enhance their process.
- To build a model to predict if the employee will be attrited or not based on the historical dataset.

Data Exploration



((IBM), 2021) The data mining life cycle

Based on IBM , CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, is an industry-proven way to guide your data mining efforts.

- As a **methodology**, it includes descriptions of the typical phases of a project, the tasks involved with each phase, and an explanation of the relationships between these tasks.
- As a **process model**, CRISP-DM provides an overview of the data mining life cycle.

The life cycle model consists of six phases with arrows indicating the most important and frequent dependencies between phases. The sequence of the phases is not strict. In fact, most projects move back and forth between phases as necessary.

- **Source of data:**
  The dataset was obtained from Kaggle (URL Address: HR Attrition data based on IBM Attrition | Kaggle), which is a made-up dataset created by IBM Data Scientists to help them explore and identify factors that could lead to employee attrition. The current usability is 10 and the expected update frequency is Never.

- **Number of records:**
  1,470 observations and 33 attributes.

- **Attribute description**
  The attributes are stated below along with the description of the data:

Data Dictionary of original dataset

No.	Categorical attributes	Description	No.	Continuous attributes	Description
1	Attrition	0 = No 1 = Yes	23	Age	The age of the employee
2	Business travel	Non-Travel = 1 Travel_Frequently = 2 Travel_Rarely = 3	24	Monthly income	The monthly salary of the employee
3	Department	Human Resources = HR = 1 Research & Development = RD = 2 Sales = SL = 3	25	Total working years	Total work experience in years

4	Gender	Female = 1 Male = 2	26	Years since last promotion	Years since the last promotion
5	**JobRole**	Healthcare Representative = 1 Human Resources = 2 Laboratory Technician = 3 Manager = 4 Manufacturing Director = 5 Research Director = 6 Research Scientist = 7 Sales Executive = 8 Sales Representative = 9	27	Years with current Manager	Years with current manager
6	Job involvement	Rated on the scale of 1 to 4, 1 being lowest and 4 being Highest	28	Distance from home	Distance between office and home

7	Job Satisfaction	Rated on the scale of 1 to 4, 1 being lowest and 4 being Highest	29	NumCompaniesWorked	Total number of companies that the employee has worked at
8	Marital Status	Single = 1 Married = 2 Divorced = 3	30	TrainingTimesLastYear	Total number of Trainings that the employee has took provided by the company
9	Overtime	No = 0 Yes = 1	31	YearsAtCompany	TOTAL YEARS WORKED
10	Work life Balance	1 is Bad 2 is Good 3 is Better 4 is Best	32	Absenteeism	Total number of leaves
11	PerformanceRating	1 is Low 2 is Good 3 is Excellent 4 is Outstanding	33	PercentSalaryHike	the amount a salary is increased

12	StockOptionLevel	How much company stocks you own from this company			
13	Higher_Education	12th = 1 Graduation = 2 Post-Graduation = 3 PHD = 4			
14	Status_of_leaving	Better Opportunity = 1 Dept.Head = 2 Salary = 3 Work Accident = 4 Work Environment = 5			
15	Work_accident	No = 0 Yes = 1			
16	Source_of_Hire	Job Event = 1 Job Portal = 2 Recruiter = 3 Walk-in = 4			

17	Job_mode	Contract = 1 Full_Time = 2 Part Time = 3			
18	Mode_of_work	OFFICE = 1 WFH = 2			
19	Date_of_Hire	Date of hire the employee			
20	**Date_of_termination**	Date of hire the employee			
21	**Performance rating**	**1=(Unacceptable)** **2= Needs Improvement,** **3= Meets Expectations** **4=Exceeds Expectations** **5= Outstanding**			
22	**Job level**	Rated on the scale of 1 to 4, 1 being lowest and 4 being Highest			

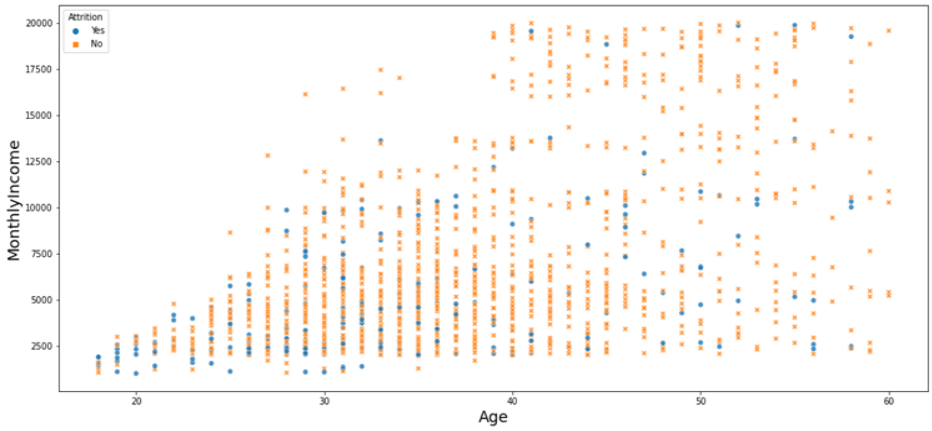
- Missing Values: We have observed missing values in a specific attribute which we have elaborate further in the data preprocessing steps
- Outliers: we suspected that there are some outliers in some dataset attributes and we visualized some of them by boxplot graph. have observed some outliers in some attributes

for instance job role and monthly income, therefore we decided to use models that are not sensitive to outliers such as Random forest and Decision tree

- We explored different programs such as R, Weka, and Python to explore, visualize, and compare the attrition rate vs. each attribute that has some correlation and finally build the model
- We used different visualization graphs such as the Kdplot, boxplot, scatter plot, and bar charts to compare mainly attrition (attribute class) vs. each correlated attribute
- Once we started exploring and understanding our dataset, we thought of some hypotheses that are key factors that are highly correlated with the class attribute:
    1. Lower Monthly income
    2. Working Overtime -
    3. Traveling Frequently
    4. Long distance between Home and work location
    5. Sales pressure for the employees of the sales department is a key reason for this employees department to have a higher rate of attrition

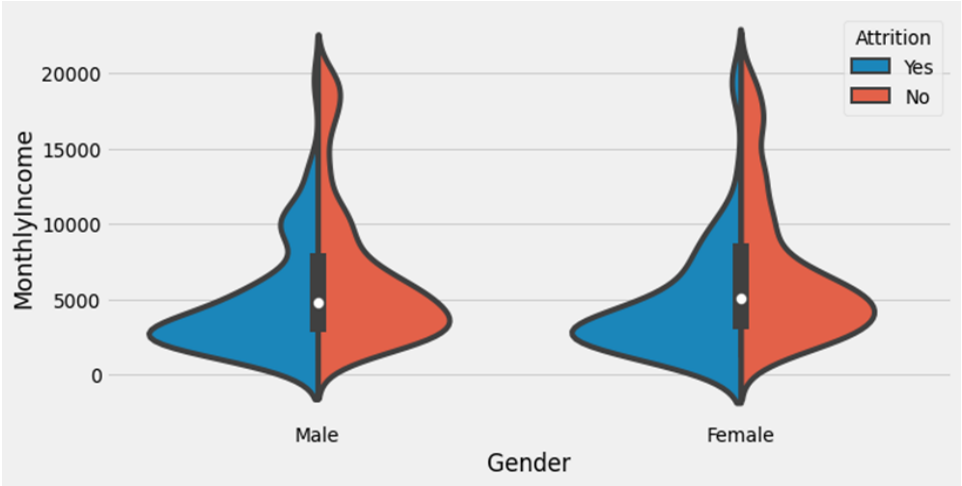
Data Visualization

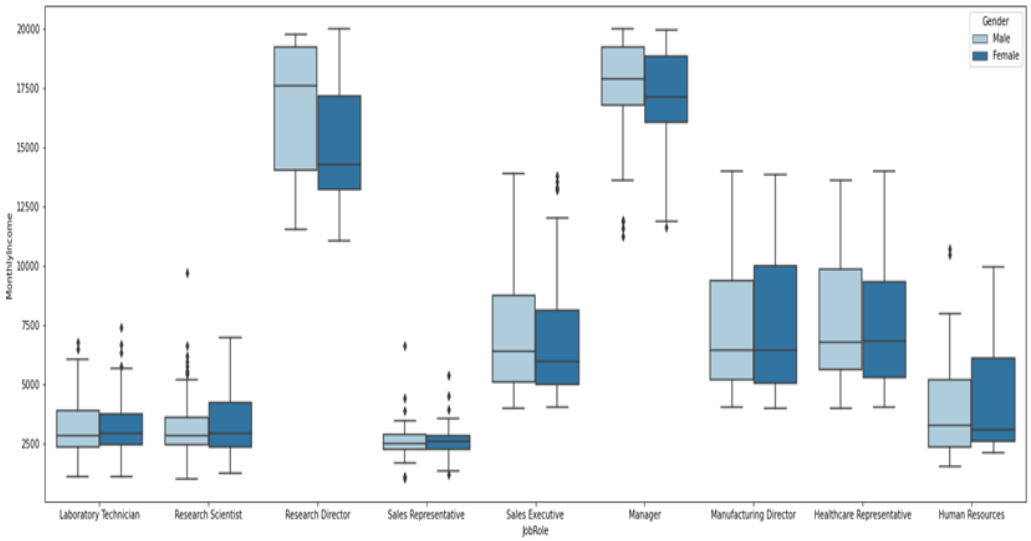
- **Age vs Monthly Income**



- **Insight:**
  From the 16% of employee who left the company, *mostly belonged in the age group b*elow 40

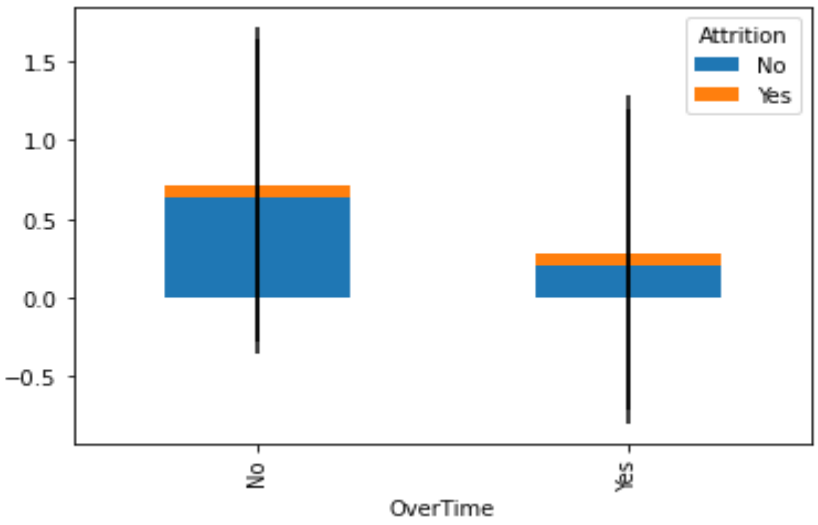
- **Monthly Income vs Gender**





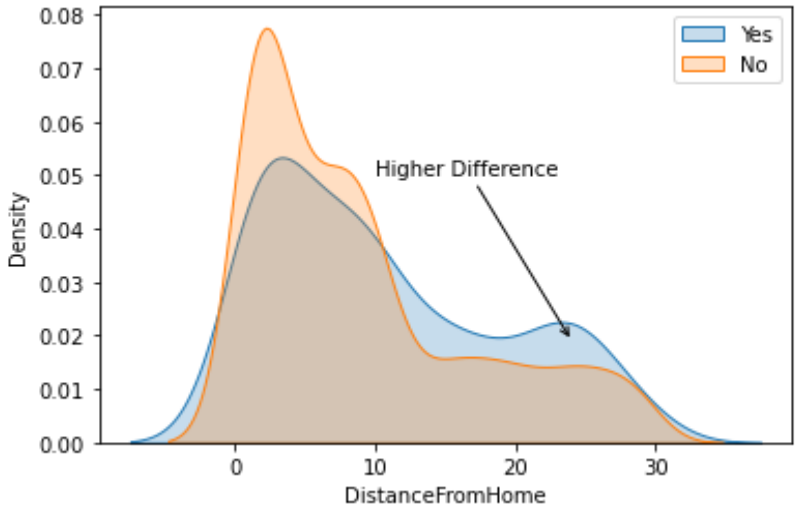
- **Insight:**
  Although there is no overall gender bias in monthly Income category, we can see some bias at Higher income job roles like Research Directors & Managers from this we can conclude a higher attrition rate of female employees at top level jobs

- **Overtime Analysis vs. Attrition**



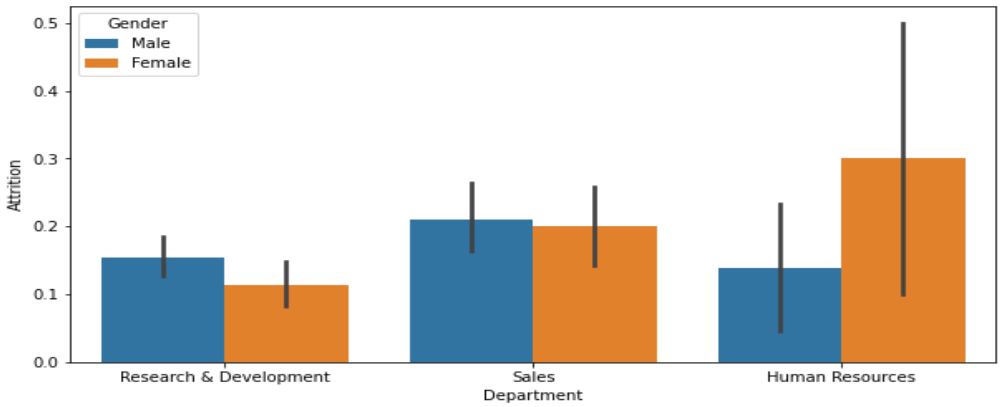
- **Insight:**
  There is no significant difference in attrition of people who did overtime(overtime=yes) and those who did not do overtime(overtime=no)

- **Attrition vs. distance from home**



- **Insight:**
  Most of the employees stay closer to the office location. There's a higher proportion of attrited employees who stay far from the office

- **Attrition Rate of the Departments**



- **Insights:**

- Employees in the R&D department seem to have a lower attrition rate than other department employees with higher attrition in both the Human Resources and Sales departments.
- However, when drilled down by gender, females seem to have almost twice the attrition rate as males in the HR department.

- **Possible causes of attrition:**

- To seek better opportunities

  Younger employees may be leaving the company in search of better opportunities or more suitable job roles
- Less Experience

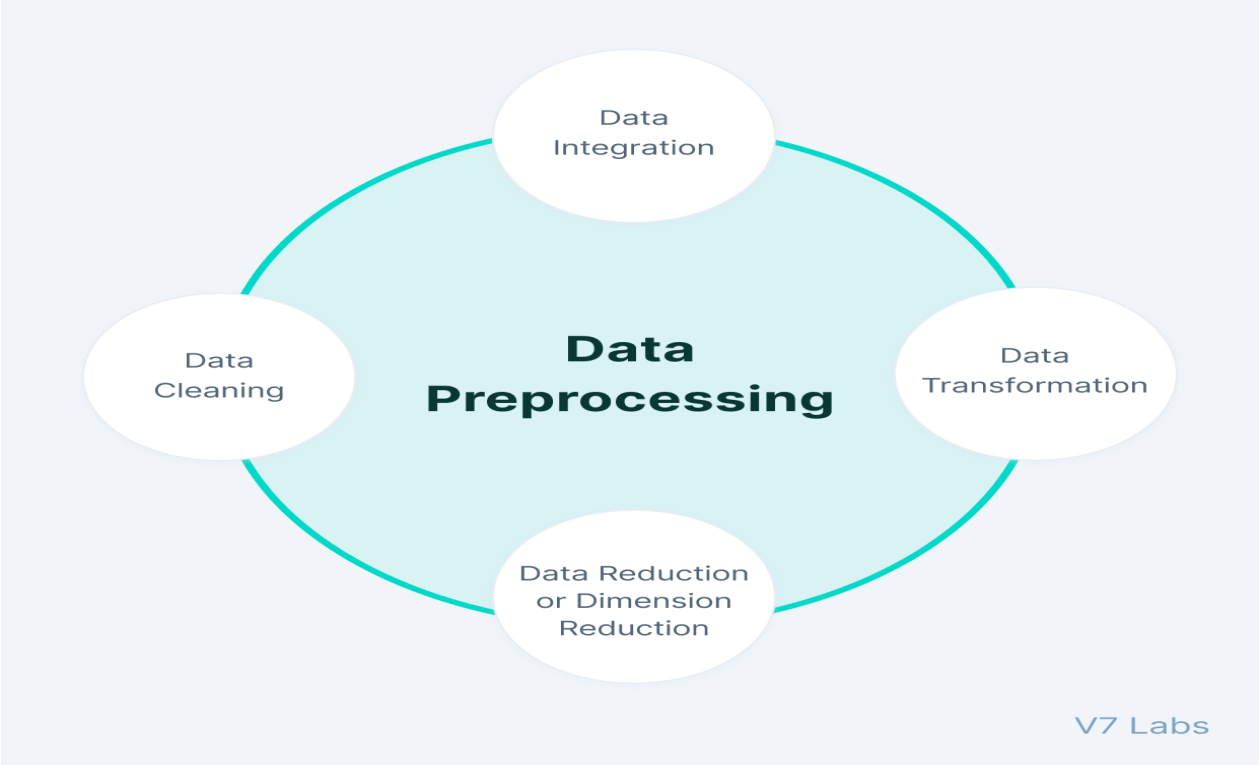
  Employees with less experience tend to churn more as they're still in their early stage of the career & may be looking for a suitable working environment
- Less Salary

  Younger employees may be underpaid with less salary while Females employees at Top positions like Research Director & Manager are paid less in comparison to their male counterparts.

  Note: All visualizations are mentioned in the Appendix

Data Preprocessing

We aimed to enhance the data mining analysis in terms of time, cost, and quality by using some data preprocessing terminologies such as the following:



- Historical thinking:

During our initial dataset exploration and understanding, our thoughts were as follows:

1. To Check the missing and null values to drop that feature.
2. To use Ranker function in Weka, principal component analysis (PCA) or correlation matrix to help us select features, we wanted to drop the features that do not correlate with (the class attribute), which is the employee attrition.
3. To transform our dataset features into continuous features to do the correlation matrix.
4. To discretize some continuous attributes applicable to Random Forest models.
5. To overcome the issue in our dataset, we had a significant imbalance in the class attribute (employee attrition) in the class attribute.

However, we found our instructor's guidance and feedback very helpful as he didn't recommend us using PCA. As PCA reduces the number of dimensions, it doesn't precisely correspond to the original variables.

And he also did not advise us to use a correlation matrix; instead, he advised us to use the **variance inflation factor** (or VIF), which measures how much the variance of a regression coefficient is inflated due to multicollinearity in the model. The smallest possible value of VIF is one (absence of multicollinearity) for the dimensionality reduction that would help us determine which visualizations to do if a variable is necessary. If we want to do a visualization of it, if a variable is going to turn out not to be important, we won't visualize it. Furthermore, he advised us to use SMOTE in R since there was an extreme imbalance dataset, so we needed to artificially increase

the number of the attire employees (up sampling) in the dataset. So, because of how it is right now, if we predict that everybody will not attirate, we will get a good accuracy score because of how few employees' attrite.

The steps of our project data preprocessing are as follows:

1.Data Cleaning:

- We checked the missing values (Date of termination) = 1470 missing values and we have dropped that column.
- We oversampled the class attribute (employee attrition feature) by using smote function. The class attribute before the oversampling: Yes: 237 (16.1%) , No: 1233 (83.9%).

```
'data.frame':   1470 obs. of  31 variables:
 $ Age                     : int  37 21 45 23 22 19 19 28 29 18 ...
 $ Attrition               : int  1 0 0 0 0 1 1 1 0 1 ...
 $ BusinessTravel          : int  3 3 3 3 3 3 2 3 3 3 ...
 $ Department              : int  2 2 2 3 2 3 3 2 3 2 ...
 $ DistanceFromHome        : int  2 15 6 2 15 22 1 2 2 3 ...
 $ Gender                  : int  2 2 2 2 1 2 1 2 2 2 ...
 $ JobInvolvement          : int  2 3 3 3 3 3 1 3 2 3 ...
 $ JobLevel                : int  1 1 3 1 1 1 1 1 2 1 ...
 $ JobRole                 : int  3 7 6 9 3 9 9 3 8 3 ...
 $ JobSatisfaction         : int  3 4 1 1 4 3 1 3 2 3 ...
 $ MaritalStatus           : int  1 1 2 3 1 1 1 1 2 1 ...
 $ MonthlyIncome           : int  2090 1232 13245 2322 2871 1675 2325 3485 6644 1420 ...
 $ NumCompaniesWorked      : int  6 1 4 3 1 1 0 2 2 1 ...
 $ OverTime                : int  1 0 1 0 0 1 0 0 0 0 ...
 $ Percent3Hike            : int  15 14 14 13 15 19 21 11 19 13 ...
 $ PerformanceRating       : int  3 3 3 3 3 3 4 3 3 3 ...
 $ StockOptionLevel        : int  0 0 0 1 0 0 0 0 2 0 ...
 $ TotalWorkingYears       : int  7 0 17 3 1 0 1 5 10 0 ...
 $ TrainingTimesLastYear   : int  3 6 3 3 5 2 5 5 2 2 ...
 $ YearsAtCompany          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ YearsSinceLastPromotion: int  0 0 0 0 0 0 0 0 0 0 ...
 $ YearsWithCurrManager    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Higher_Education        : int  2 2 3 4 4 4 4 3 2 4 ...
 $ Year_of_Hire            : int  2021 2021 2021 2021 2021 2021 2021 2021 2021 2021 ...
 $ Status_of_leaving       : int  3 4 2 4 1 4 4 5 1 5 ...
 $ Mode_of_work            : int  1 2 2 1 2 2 2 2 1 2 ...
 $ Leaves                  : int  4 5 1 1 5 1 2 0 5 5 ...
 $ Absenteeism             : int  2 2 3 0 2 1 2 2 2 2 ...
 $ Work_accident           : int  0 0 0 1 0 1 0 0 0 0 ...
 $ Source_of_Hire          : int  1 3 1 3 1 2 4 4 4 4 ...
 $ Job_mode                : int  1 3 1 2 1 3 1 1 3 2 ...
```

The class attribute after the oversampling: Yes: 1185 (49%) , No: 1233 (51%) :(2418 x 25)
*After using vif function as well

```
'data.frame':   2418 obs. of  25 variables:
 $ Attrition              : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 2 2 1 2 ...
 $ BusinessTravel         : int  3 3 3 3 3 3 2 3 3 3 ...
 $ Department             : int  2 2 2 3 2 3 3 2 3 2 ...
 $ DistanceFromHome       : int  2 15 6 2 15 22 1 2 2 3 ...
 $ Gender                 : int  2 2 2 2 1 2 1 2 2 2 ...
 $ JobInvolvement         : int  2 3 3 3 3 3 1 3 2 3 ...
 $ JobSatisfaction        : int  3 4 1 1 4 3 1 3 2 3 ...
 $ MaritalStatus          : int  1 1 2 3 1 1 1 1 2 1 ...
 $ NumCompaniesWorked     : int  6 1 4 3 1 1 0 2 2 1 ...
 $ OverTime               : int  1 0 1 0 0 1 0 0 0 0 ...
 $ PerformanceRating      : int  3 3 3 3 3 3 4 3 3 3 ...
 $ StockOptionLevel       : int  0 0 0 1 0 0 0 0 2 0 ...
 $ TotalWorkingYears      : int  7 0 17 3 1 0 1 5 10 0 ...
 $ TrainingTimesLastYear  : int  3 6 3 3 5 2 5 5 2 2 ...
 $ YearsAtCompany         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ YearsSinceLastPromotion: int  0 0 0 0 0 0 0 0 0 0 ...
 $ YearsWithCurrManager   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Higher_Education       : int  2 2 3 4 4 4 4 3 2 4 ...
 $ Status_of_leaving      : int  3 4 2 4 1 4 4 5 1 5 ...
 $ Mode_of_work           : int  1 2 2 1 2 2 2 2 1 2 ...
 $ Leaves                 : int  4 5 1 1 5 1 2 0 5 5 ...
 $ Absenteeism            : int  2 2 3 0 2 1 2 2 2 2 ...
 $ Work_accident          : int  0 0 0 1 0 1 0 0 0 0 ...
 $ Source_of_Hire         : int  1 3 1 3 1 2 4 4 4 4 ...
 $ Job_mode               : int  1 3 1 2 1 3 1 1 3 2 ...
```

2.Data integration:

- **We didn't use that.**

3.Data transformation:

- Discretization
  We didn't Discretise
- Sampling
  We didn't use that.
- **Aggregation**
  df2 doesn't have the class attribute, so we concatenated the attrition attribute. We have added it in df3.

4.Data reduction of dimensionality reduction /Feature Selection:

- We changed all the attribute types to integer, then we changed the attrition attribute type to factor.

- We used the variance inflation factor (vif). After vif is applied to the dataset, we removed attributes whose vif value is greater than 5 and we removed attributes whose p value is greater than 0.1 based on our instructor's experience and advice.

Standard errors: MLE

	Est.	S.E.	z val.	p	VIF
(Intercept)	17.48	0.98	17.91	0.00	
BusinessTravel	-0.24	0.09	-2.58	0.01	1.02
Department	-0.26	0.12	-2.22	0.03	1.05
DistanceFromHome	0.03	0.01	3.94	0.00	1.04
Gender	-0.53	0.12	-4.39	0.00	1.03
JobInvolvement	-0.96	0.09	-11.02	0.00	1.06
JobSatisfaction	-0.65	0.06	-11.09	0.00	1.07
MaritalStatus	-0.97	0.12	-8.30	0.00	1.80
NumCompaniesWorked	0.08	0.03	3.06	0.00	1.29
OverTime	1.05	0.13	7.78	0.00	1.07
PerformanceRating	-1.09	0.22	-5.05	0.00	1.01
StockOptionLevel	-0.16	0.10	-1.61	0.11	1.74
TotalWorkingYears	-0.10	0.01	-7.32	0.00	2.63
TrainingTimesLastYear	-0.41	0.05	-7.66	0.00	1.04
YearsAtCompany	0.06	0.02	2.77	0.01	4.60
YearsSinceLastPromotion	0.12	0.03	4.14	0.00	2.12
YearsWithCurrManager	-0.23	0.03	-7.35	0.00	2.50
Higher_Education	-0.22	0.06	-3.75	0.00	1.04
Status_of_leaving	-0.16	0.05	-3.55	0.00	1.05
Mode_of_work	-0.72	0.12	-5.97	0.00	1.04
Leaves	-0.14	0.04	-3.89	0.00	1.03
Absenteeism	-0.46	0.06	-7.81	0.00	1.06
Work_accident	-0.48	0.12	-3.91	0.00	1.03
Source_of_Hire	-0.26	0.06	-4.46	0.00	1.04
Job_mode	-0.74	0.08	-9.09	0.00	1.07

- We removed the attributes from the dataset as follows:
  job level, monthly income, year of hire percent salary hike,job role and age
- We have reached the final dataset df4, with a new description (2418 obs of 25 attributes).

Note: All codes of the data preprocessing are mentioned Appendix

Data Mining Techniques/Algorithms

Our dataset has an attribute called "Attrition," which we decided to use as a class attribute for our classification problem, determining whether or not employees are more likely to be attrited. Yes, and No are the values in the 'Attrition' attribute.

In order to solve the classification problem, we investigated three classification techniques and compared their accuracy and other model evaluation metrics like sensitivity, specificity, cost ,etc. As a result, we can choose the best classifier for our dataset. The dataset was divided into training and testing sets with a split ratio of 70:30.

Note: All codes of Data Mining Techniques/Algorithms are mentioned Appendix

The three algorithms we used are listed below.

1. **Naive Bayes**

The Naive Bayes classification method is based on the Bayes' Theorem and the assumption of predictor independence. A Naive Bayes classifier, in simple terms, assumes that the presence of one feature in a class has no bearing on the presence of any other feature in that class. Naive Bayes has the ability to handle both continuous and discrete data, as well as being highly scalable and requiring little training data.

When building the model, we used Laplace smoothing (laplace = 1) to avoid the zero probability problem, which occurs when some observations are never made and the probability of new observations belonging to that class is 0 due to the conditional probability calculations used in the Naive Bayes method computation. The model contains a set of frequency tables for each of the attributes used in the process, which are then used to calculate the likelihood of any previously unseen observation belonging to a specific class. The following are the results obtained after the model was built and tested with the testing set:

```
Confusion Matrix and Statistics


predictions_NB   0   1
             0 269  70
             1  76 310

               Accuracy : 0.7986
                 95% CI : (0.7675, 0.8272)
    No Information Rate : 0.5241
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.596

 Mcnemar's Test P-Value : 0.679

            Sensitivity : 0.7797
            Specificity : 0.8158
         Pos Pred Value : 0.7935
         Neg Pred Value : 0.8031
             Prevalence : 0.4759
         Detection Rate : 0.3710
   Detection Prevalence : 0.4676
      Balanced Accuracy : 0.7977

       'Positive' Class : 0
```

2. Random forest

Random forest is a classification technique that makes use of ensemble learning, which is a technique for solving complex problems by combining multiple classifiers. One of the most significant advantages is that it reduces dataset overfitting and increases precision.
We built the model with ntree = 500 and mtry = 4 (where ntree is the number of trees and mtry is the number of variables tried at each split) and obtained an OOB (Out-Of-Bag) estimate of error rate of 11.51%, indicating that the model correctly classified 88.49% of the OOB samples. The following results were obtained after the model was built and tested with the testing set:

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 345  46
         1  30 311

               Accuracy : 0.8962
                 95% CI : (0.8718, 0.9173)
    No Information Rate : 0.5123
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.792

 Mcnemar's Test P-Value : 0.08532

            Sensitivity : 0.9200
            Specificity : 0.8711
         Pos Pred Value : 0.8824
         Neg Pred Value : 0.9120
             Prevalence : 0.5123
         Detection Rate : 0.4713
   Detection Prevalence : 0.5342
      Balanced Accuracy : 0.8956

       'Positive' Class : 0
```

3. Decision tree

A decision tree is a classification technique that can be used to make decisions. It divides a dataset into smaller subsets while also steadily developing the decision tree. The final tree is made up of decision nodes and leaf nodes. A decision node has at least two branches. The leaf nodes represent a classification or decision.

The R function ctree() is used to create the decision trees. The ctree() method is a conditional inference tree that uses recursive partitioning to estimate the regression relationship. The following results were obtained after the model was built and tested with the testing set:

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 311  99
         1  81 282

               Accuracy : 0.7671
                 95% CI : (0.7357, 0.7965)
    No Information Rate : 0.5071
    P-Value [Acc > NIR] : <2e-16

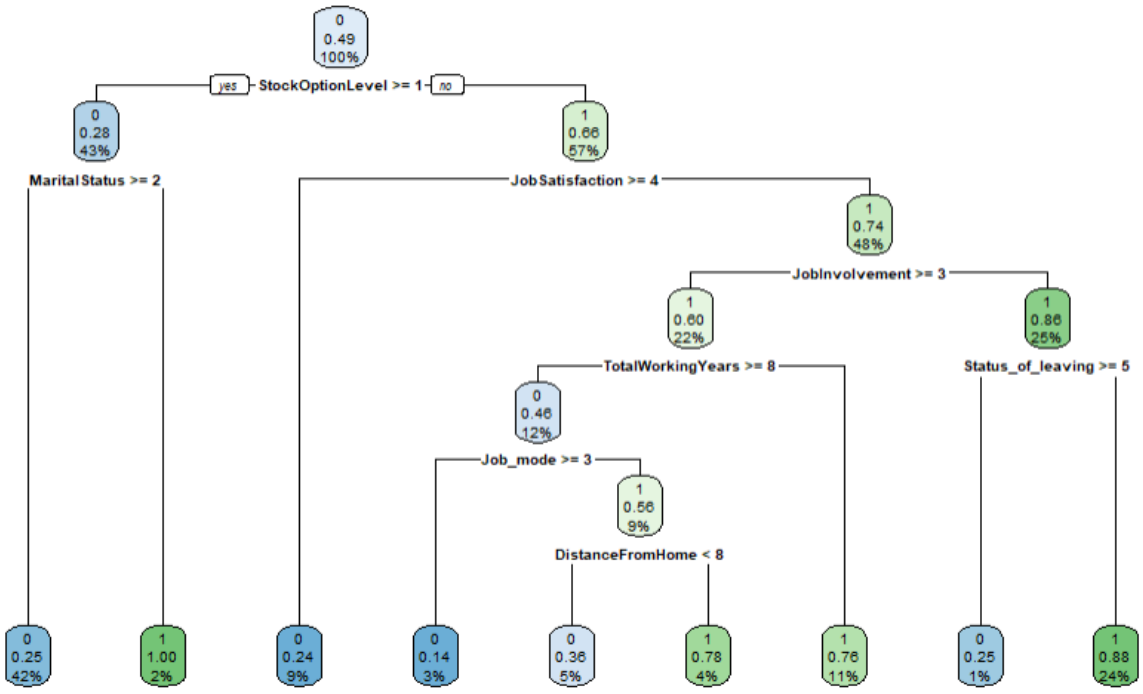
                  Kappa : 0.5339

 Mcnemar's Test P-Value : 0.2051

            Sensitivity : 0.7934
            Specificity : 0.7402
         Pos Pred Value : 0.7585
         Neg Pred Value : 0.7769
             Prevalence : 0.5071
         Detection Rate : 0.4023
   Detection Prevalence : 0.5304
      Balanced Accuracy : 0.7668

       'Positive' Class : 0
```

Decision tree



Results

The three models' accuracy, sensitivity, and specificity are listed in the table below. The model's sensitivity and specificity indicate how frequently it correctly predicts attrition and retention, respectively. As shown in the table, the Random Forest Classifier has the highest accuracy, followed by Naive Bayes and the Decision Tree.

Models	Accuracy	Sensitivity	Specificity
Random Forest	89.62%	92%	87.11%
Naive Bayes	79.86%	77.97%	81.58%
Decision Tree	76.71%	79.34%	74.02%

In order to choose the best prediction model, we'll introduce a cost matrix that computes the cost of each model.

Cost Matrix

	NO (0)	**YES (1)**
NO (0)	TN	FP
YES (1)	FN	TP

Where,
TN (True Negative): indicates that employees will not be attrited and are not attrited.
TP (True Positive): indicates that employees will be attrited and are actually attrited.
FN (False Negative): indicates that employees were not going to be attrited but were.
FP (False Positive): indicates that employees were going to be attrited but were not.

Positive numbers (costs) in a cost matrix can be used to sway unfavorable results. Negative numbers (benefits) can be used to influence positive outcomes because negative costs are perceived as benefits.

False Positive means that the prediction caused the company to spend money on recruiting new workers because of employee attrition. We calculated that each false positive costs $1 as a result. False Negative suggests that the company must pay to keep its employees. We calculated that each false negative costs $100 as a result. Due to the model's goal of assisting the company in determining where to invest more money in employee retention, True Negative indicates that the prediction does not cost the organization, while True positive indicates that the prediction does cost the organization as the employees are attrited. so, -1 is considered as a cost.

Accordingly, below is the proposed cost matrix:

	NO (0)	**YES (1)**
NO (0)	0	1
YES (1)	100	-1

Each model's calculated cost of confusion matrices are given below:

Models	**Accuracy**	**Cost**	**Rank (per Cost)**
Random Forest	89.62%	2,735	1
Naive Bayes	79.86%	7,360	2
Decision Tree	76.71%	7,917	3

Based on the results of the cost matrix mentioned above, we discovered that the Random Forest has the lowest cost among the other models. The Random Forest model exhibits a good level of accuracy (the highest accuracy among the three models). Hence, the Random Forest model is the most effective model to use in this study for solving the classification problem.

However, our instructor advised us to explore using other measures that might work better, like rmse or mae, depending on which techniques we use. But based on our research, we have found that rmse or mae is a model evaluation metric for regression models, not for classification models like in our project. Therefore, we didn't use them.

Note: All codes of Results are mentioned Appendix

Conclusions and Lessons Learned

In conclusion, there are major takeaways from this project in terms of how well the team was able to solve the problem stated above and lessons learned from working on the project together as a team.

To start with summarizing the major takeaways/lessons from the project:

- **The importance of Data analysis is that** it plays a major role in companies, which leads to helping businesses optimize their performances. The implementation of Data analysis in the business model means companies can easily help and attain their goals of reducing costs by identifying more efficient ways of doing business and by storing large amounts of data. Moreover, the below findings have been concluded;
    1. Helps HR to interpret data, find out the trends and help take required steps to keep the company running smoothly and profitability
    2. It helps to study the impact of various factors on employee attrition, which has become a major part of data/HR analytics.

- **Other activities or steps in Data analysis, such as Data cleaning, Data preprocessing, Text mining, Data mining and visualization,** As a team, it was a great experience going through all of these activities and steps, as this helped us solve the problem from scratch with massive data and different codes, so we worked from zero to clean the data, using different programming languages and models such as R Studio, Python, and Weka to generate or analyze data, and lastly, visualize. It was the most important part to reflect all the data so that it could be visually seen. These steps consumed a lot of effort, time, and work from each one of us, but it was very helpful and achievable in terms of the below findings we reached:
    1. There are five top variables impacted by attrition: Monthly income, Overtime, Job level, Total working years, and Job satisfaction.
    2. There are other variables that didn't impact the attrition: job level, monthly income, year of hire percent salary hike,job role, and age.
    3. Below had a high attrition;
        - People who travel frequently
        - People with low satisfaction with the work environment(It is to be expected that job satisfaction is correlated with attrition, but it is nevertheless important to know that it is because the simple act of asking someone about job satisfaction may not be enough to determine job satisfaction. Here we see that it is.)
    4. Job level, Total working years and years at company have the most impact on monthly income.
    5. Leaving the job by employees under 30 is more likely than their older counterparts.

On the other hand, teamwork played an important role in the success of this report. Below are the key learnings and key success factors that the team has deployed and learned through working together on the project:

1- A clear and agile project road map that identifies the expected responsibility matrix for the key tasks, meetings with the course instructor, and the deadlines from course day one.

2- The deep involvement of our instructor, Dr.Mick, during the project ideation and project development, his guidance, and his valuable feedback starting from dataset selection and exploring, project ideation, and project development, selecting the models, coding, and project report generation has strengthened our knowledge and optimized our time and effort.

3- Having a project lead/coordinator and group implement the best practices of project management during the course and continuing following up on meetings and project deliverables to avoid unsuccessful meetings or failing to meet the deadlines.

4- Based on our agile road map, we have found it's essential to have all group members working on key tasks like dataset selection and exploring. At the same time, it's better to distribute the team members for some tasks, for instance, data mining in R, Data mining in Python, and weka. However, we have conducted a knowledge-sharing session to ensure that all team members have access to the exact project details to optimize our resources, time, and workforce and maximize our project's added value and the variety of data mining tools.

5- Creating three work streams to explore and try three different data mining tools to strive for excellence, uniqueness among other groups, and higher grades.

6- Organizing our work by creating a shared drive to store all our documents to be accessible 24/7.

7- Accepting and adapting to our diversity advantages and disadvantages to working with different group members' mindsets since we all come from different cultures, educational backgrounds, and seniority levels.

8- We are lucky that all group members have shown a high level of professionalism, commitment, and work effort toward the project or helping any project team member.

9- Working on this short journey together on this project as a team has taught us many new things and developed our skills and professionalism in many ways. For instance, we have passed through the project team stages like any new team.

a) Forming stage: However, we have to work with some students for the first time and get to know each person; our group consists of 6 students. We think it is considered a large group that could be a challenge in terms of communication and distribution of the workloads. Still, we have turned these challenges into opportunities by having an agile roadmap for

the whole course. It requires deliverables, and we have established ground rules for teamwork. We have pursued the extra mile by creating more work streams for the three data mining software, having a hybrid project work model to fulfill the student's needs and commitments, and completing the project tasks and deadlines.

b) In the storming phase, we had many differences in perspective as we all came from different educational and industrial backgrounds. Thanks to our clear roadmap from the course on day one to deliver this project report.

c) After that, we moved to the norming phase, where we built more on our trust in each other and strengthened our teamwork mindset and skills through several regular project team meetings and communication through our WhatsApp group. As a result, we started helping each other with some tasks, such as analytical thinking and coding. Moreover, we had a tremendous knowledge-sharing session, so each of us explained what was working on in his work stream in detail, and we followed that step by step.

d) We entered the performing stage, where we started seeing our efforts and searching come into reality in performing the models and getting the desired results from the applied models and data mining software.

e) Finally, we are at the adjournment phase where we will celebrate successfully generating this report and our desired highest grades in this course by having a group meeting lunch very soon to celebrate our success story. ProjectManagement.com - The Five Stages of Team Development and the Role of the Project Manager

Appendix

Confusion Matrix

Naive Bayes-

	NO (0)	**YES (1)**
NO (0)	269	70
YES (1)	76	310

Random forest-

	NO (0)	**YES (1)**
NO (0)	345	46
YES (1)	30	311

Decision Tree-

	NO (0)	**YES (1)**
NO (0)	311	99
YES (1)	81	282

R Code

```
---
title: "EmployeeAttritionAnalysis"
author: "RIT Tigers"
date: '2022-06-12'
output:
  word_document: default
  html_document: default
---
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```{r packages, eval=TRUE, echo=TRUE}
library(dplyr)
library(ggplot2)
library(gridExtra)
library(jtools)
library(smotefamily)
```
```

```
library(tidyverse)
library(psych)
library(e1071)
library(caret)
library(randomForest)
library(caTools)
library(party)
library(rpart)
library(rpart.plot)
```
```{r readDataset, eval=TRUE, echo=TRUE}
data1<- read.csv("~/dataset/group project/numAttritionData.csv", header=T)
data1
```

DATA PREPROCESSING

Missing Values:

```{r missingValues, eval=TRUE, echo=TRUE}

missing_values <- sapply(data1, function(x) sum(is.na(x)))

sum(missing_values)

```

```{r removeNullValues, eval=TRUE, echo=TRUE}

df<- subset(data1, select = -Date_of_termination)

names(df)

```

```{r attritionValue, eval=TRUE, echo=TRUE}

table(df$Attrition) #the data is imbalanced

str(df)

```

Oversampling

```{r oversampling, eval=TRUE, echo=TRUE}

new_df <- SMOTE(df[,-2],df[,2],K=7)

str(new_df)
```

```

Concatenating the Attrition Attribute

```{r concatenate, eval=TRUE, echo=TRUE}
#concat attrition attribute
table(new_df$data$Attrition)
df2<-new_df$syn_data
df2<-df2 |> mutate(Attrition=1,.after=Age)
df3<-bind_rows(df,df2)
df3
table(df3$Attrition)
```

Aggregation

```{r eval=TRUE, echo=TRUE }
df3<-select(df3,-class)
df3
```

Feature Selection

```{r vif, eval=TRUE, echo=TRUE}
#vif -> variance inflation factor
df4<-df3 %>% mutate(across(Age:Job_mode,as.integer))
df4$Attrition<-as.factor(df4$Attrition)
df4
m<-glm(Attrition~ .-JobLevel-MonthlyIncome-Year_of_Hire-Percent3Hike-JobRole-Age,data=df4,family=binomial)
summ(m,vifs=TRUE)
```
```

```{r newDataset, eval=TRUE, echo=TRUE}

df4<-select(df4,-
c("Age","JobLevel","MonthlyIncome","Year_of_Hire","Percent3Hike","JobRole"))

df4

#write.csv(df4,"numAttritionData_disc.csv")

```

```{r eval=TRUE, echo=TRUE}

df5 <- df4

df6 <- df4

df7<-df4

```

NAIVE BAYES CLASSIFIER

```{r nbclassifier, eval=TRUE, echo=TRUE}

set.seed(1234)

(emp_att<- length(df5$Attrition))

train_size <- round(emp_att * 0.7) #70% for training

test_size <- emp_att - train_size #rest for testing

emp_indx <-sample(seq(1:emp_att), train_size)

train_sample <- df5[emp_indx,]

test_sample <- df5[-emp_indx,]


classifier_NB <- naiveBayes(

  subset(train_sample, select = -Attrition),

  train_sample$Attrition, laplace = 1)

classifier_NB
```

```
predictions_NB <- predict(classifier_NB,
                          subset(test_sample, select = -Attrition))
table(predictions_NB, test_sample$Attrition)
round(sum(predictions_NB == test_sample$Attrition, na.rm = TRUE)
/length(test_sample$Attrition), digits = 2)

(confusionMatrix_NB <- confusionMatrix(table(predictions_NB, test_sample$Attrition)))
```

RANDOM FOREST CLASSIFIER

```{r rfclassifier, eval=TRUE, echo=TRUE}

df6$Attrition<- as.factor(df6$Attrition)
table(df6$Attrition)

set.seed(2236)
ind<- sample(2, nrow(df6), replace = TRUE, prob = c(0.7,0.3))
train<- df6[ind == 1,]
test<- df6[ind == 2,]

(classifier_RF <- randomForest(Attrition~., data=train, proximity = TRUE))

predictions_RF1 <- predict(classifier_RF, train)
(confusionMatrix_RF1 <- confusionMatrix(predictions_RF1, train$Attrition))

predictions_RF2 <- predict(classifier_RF, test)
(confusionMatrix_RF2 <- confusionMatrix(predictions_RF2, test$Attrition))
```

```

DECISION TREE CLASSIFIER

```{r dtclassifier, eval=TRUE, echo=TRUE}

set.seed(5432)

sample_data<- sample.split(df7, SplitRatio = 0.7)

train_data<- subset(df7, sample_data == TRUE)

test_data<- subset(df7, sample_data == FALSE)


dt_model <- ctree(Attrition~., train_data)

predict_model <- predict(dt_model, test_data)

e_at<- table(test_data$Attrition, predict_model)

e_at

ac_test<- sum(diag(e_at))/ sum(e_at)

ac_test

(confusionMatrix_DT <- confusionMatrix(predict_model, test_data$Attrition))

fit2 <- rpart(Attrition~.,data = test_data, method = "class")

rpart.plot(fit2, extra=106)

```


Python Code - Visualizations

Importing Libraries

```
import numpy as np # mathematical calculation
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns # data visualization
import matplotlib.pyplot as plt # data visualization
from scipy.stats import norm # for stats funtions
%matplotlib inline # data visualization
```

Age vs Monthly Income
```

```
plt.figure(figsize=(18,8))

ax=sns.scatterplot(x="Age",y="MonthlyIncome",data=ea,hue="Attrition",style
="Attrition",alpha=0.80, sizes=(10, 30))

ax.set_xlabel("Age", fontsize=18)

ax.set_ylabel("MonthlyIncome", fontsize=18)
```

Monthly Income vs Gender

a.
```
fig,ax = plt.subplots(figsize=(10,5))
 sns.violinplot(x='Gender', y='MonthlyIncome',hue='Attrition',split=True,data=ea)
```

b.Income vs job role
```
plt.figure(figsize=(24,8))
 sns.boxplot(x='JobRole', y='MonthlyIncome', hue = 'Gender',palette = "Paired",data = ea)
```

Overtime Analysis vs. Attrition

```
pd.crosstab(ea["OverTime"], ea["Attrition"], normalize=True).plot(kind='bar', stacked=True,
yerr=True)
ea.groupby(["OverTime","Attrition"])[["Attrition"]].agg(["count"])
```

Attrition vs. distance from home

```
Attrition_Y = ea[ea['Attrition']==True]
Attrition_N = ea[ea['Attrition']==False]

sns.kdeplot(Attrition_Y.DistanceFromHome,fill=True)
sns.kdeplot(Attrition_N.DistanceFromHome,fill=True)
plt.legend(('Yes', 'No'))

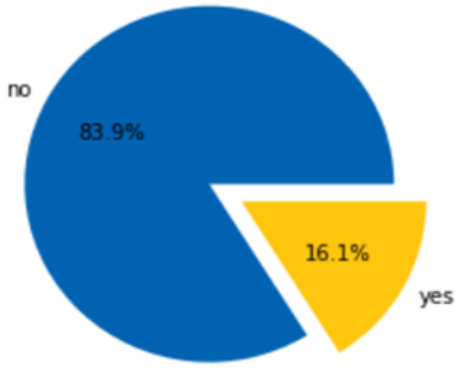
plt.annotate('Higher Difference',
        xy = (24, 0.019),
        xytext = (10, 0.05),
        arrowprops = {'arrowstyle':'->', 'color':'black'})
```

Attrition Rate of the Departments

```
plt.figure(figsize=(10,5))
sns.barplot(x='Department',y='Attrition', data = ea, hue = 'Gender')
```

Some Other Observations found during project implementation.

Employee attrition %



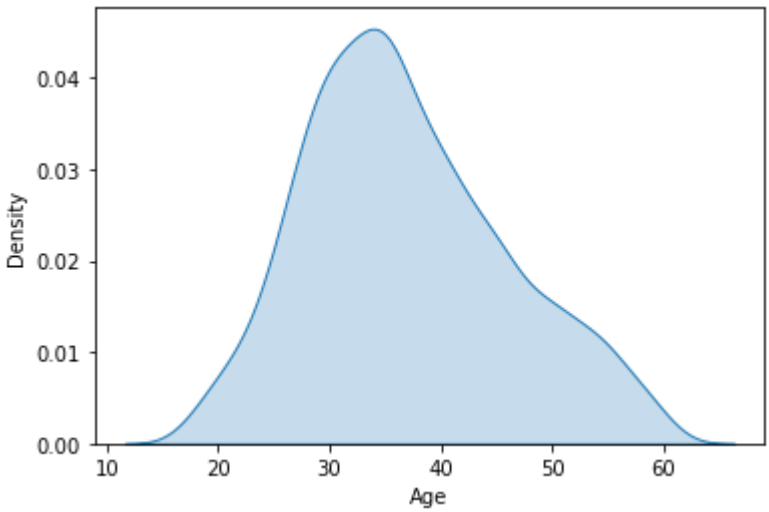
Code:

```
count_attrition = pd.DataFrame(ea['Attrition'].value_counts())
colors = ["#0063B2","#ffc60e"]
plt.pie(count_attrition['Attrition'],labels=['no' , 'yes'] ,explode = (0.2,0),autopct='%1.1f%%',colors=colors)
```

Age Distribution of employees in IBM

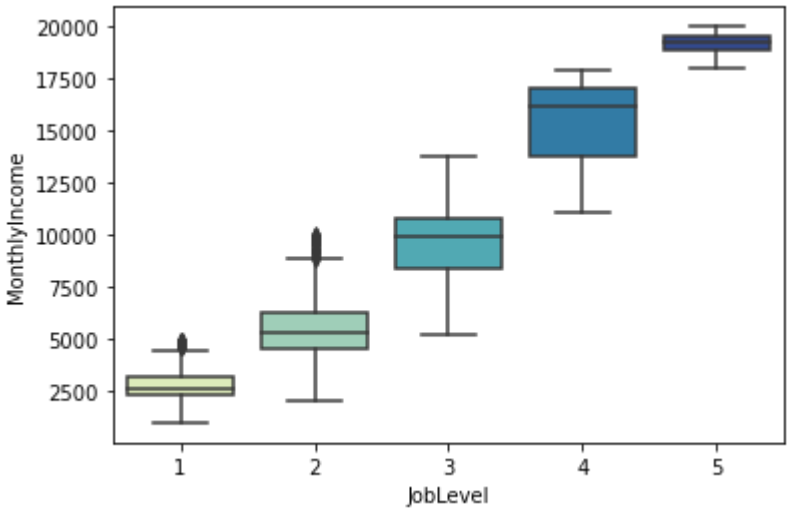
Count 1470.000000	mean 36.923810
std 9.135373	min 18.000000
25% 30.000000	50% 36.000000
75% 43.000000	max 60.000000

Code : ea.Age.describe()



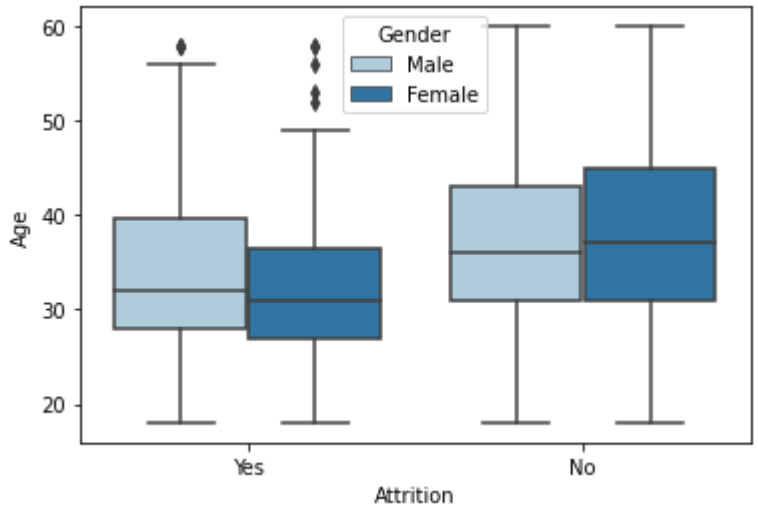
Code : sns.kdeplot(ea.Age,fill=True)

Monthly Income Distribution in relation to Job Level



Code : sns.boxplot(x = "JobLevel",y= "MonthlyIncome",data = ea, palette = "YlGnBu")

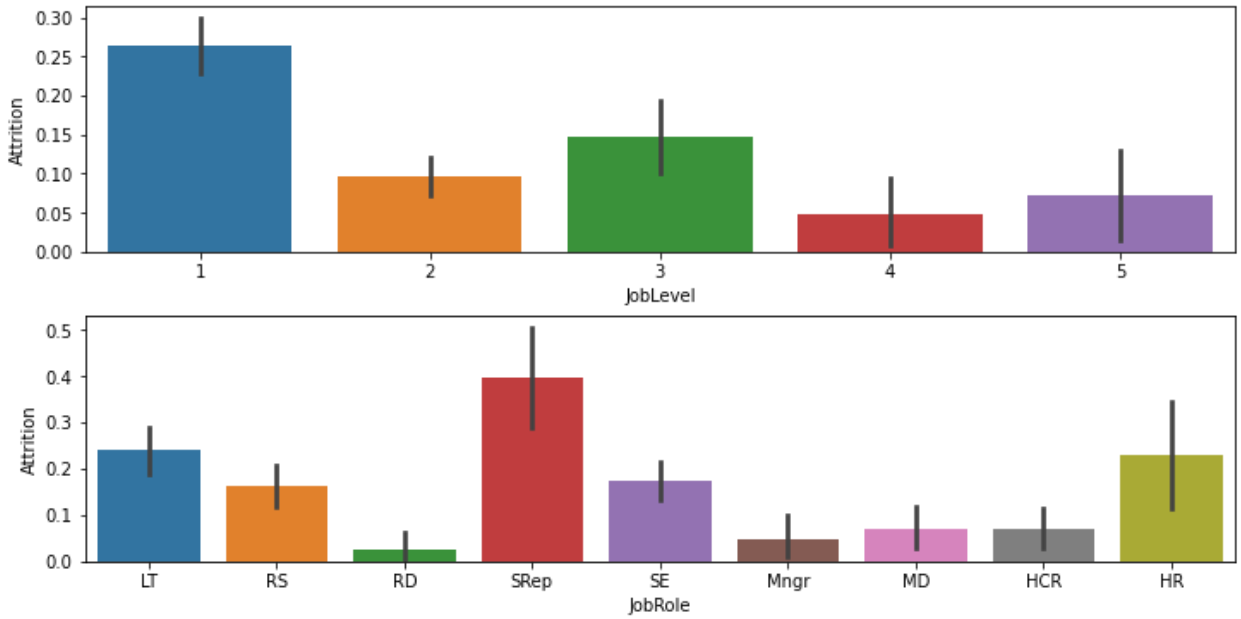
Attrition in relation to Age



Code :

```
sns.boxplot(x="Attrition",y="Age",hue="Gender",data=ea,palette = "Paired")
```

Attrition in relation to JobLevel & JobRole

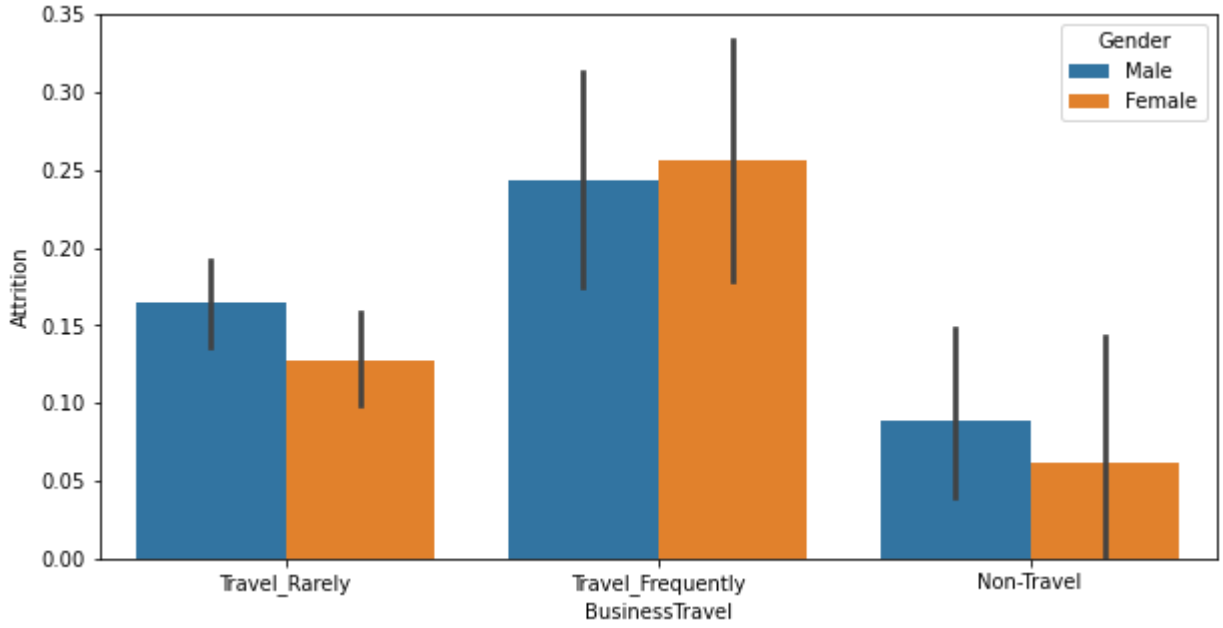


Code :

```
cols = ['JobLevel','JobRole']
fig, ax = plt.subplots(len(cols),1, figsize= (10,5), constrained_layout=True)
for i, col in enumerate(cols):
```

sns.barplot(col,'Attrition', data = replace_colname, ax = ax[i])

Attrition in Relation to Business Travel



Code : plt.figure(figsize=(10,5))
sns.barplot('BusinessTravel','Attrition', data = ea, hue = 'Gender')

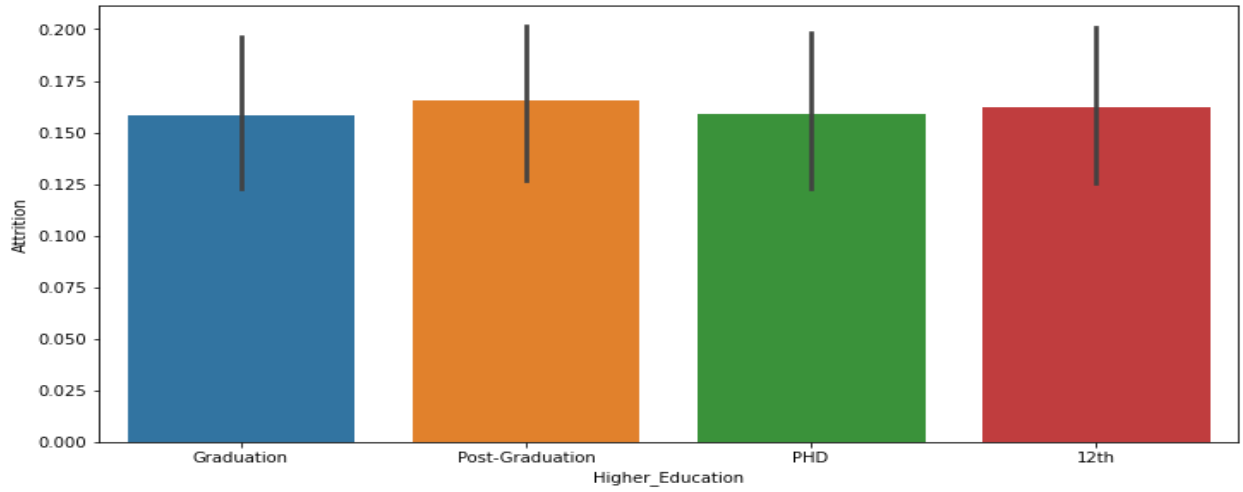
Attrition in Relation to Environmental Factors of IBM

```
Code :cols = ['JobInvolvement','JobSatisfaction','YearsSinceLastPromotion']
fig, ax = plt.subplots(len(cols),1, figsize= (10,5), constrained_layout=True)
for i, col in enumerate(cols):
    sns.barplot(col,y='Attrition', data = ea, ax = ax[i])
```

Attrition in relation to Age, Monthly Income & Years at the company

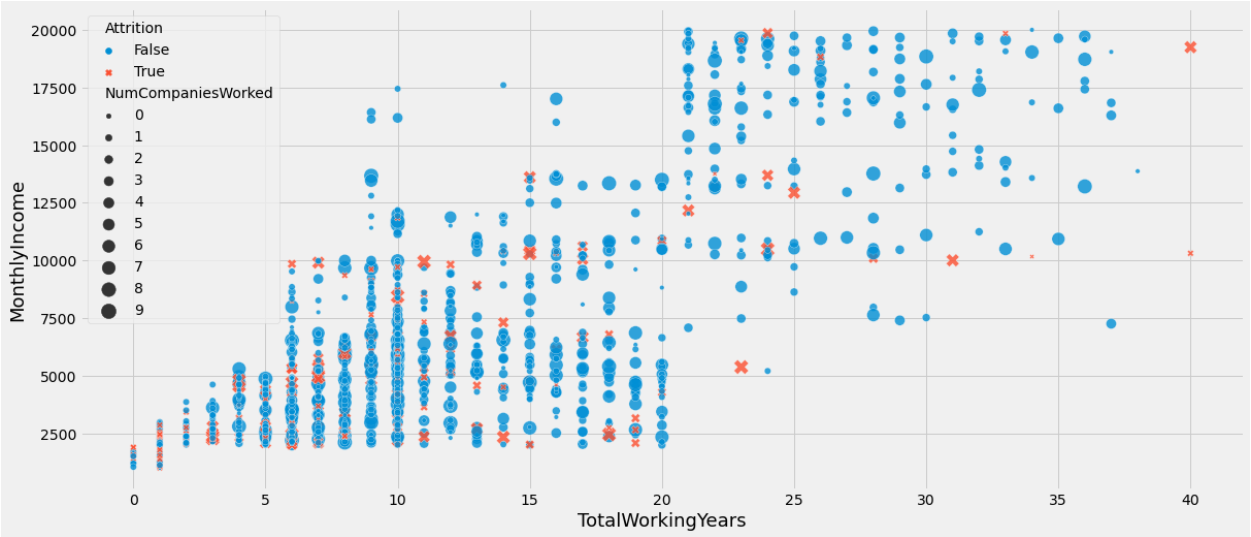

```
Code : plt.figure(figsize = (18, 7))
plt.style.use('fivethirtyeight')
plt.subplot(131)
sns.swarmplot(x="Attrition", y="Age",data=ea,size=5)
plt.subplot(132)
sns.swarmplot(x="Attrition", y="MonthlyIncome", data=ea,size=5)
plt.subplot(133)
sns.swarmplot(x="Attrition", y="YearsAtCompany", data=ea,size=5)
```

Attrition in relation to Higher Education



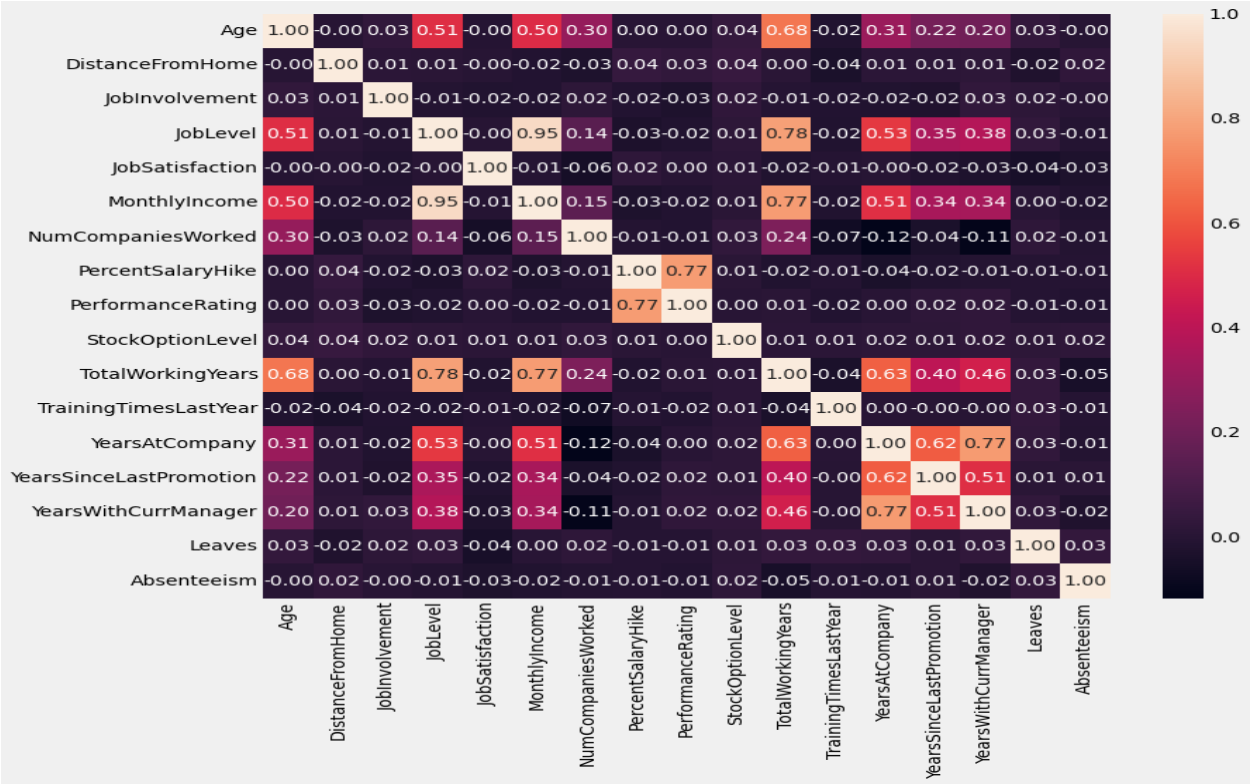
Code : plt.figure(figsize=(10,5))
sns.barplot(x='Department',y='Attrition', data = ea, hue = 'Gender', ci = None)

Attrition in Relation to Monthly Income , Total Working Years & Number of companies worked



Code : plt.figure(figsize=(18,8))
ax=sns.scatterplot(x="TotalWorkingYears", y="MonthlyIncome", data=ea, hue="Attrition",style ="Attrition",alpha=0.8, size=ea["NumCompaniesWorked"], sizes=(20, 200), legend="full")
ax.set_xlabel("TotalWorkingYears", fontsize=18)
ax.set_ylabel("MonthlyIncome", fontsize=18)
Plt.show

Correlation Heat Map



Code : plt.figure(figsize=(18,8))

ax=sns.scatterplot(x="TotalWorkingYears",y="MonthlyIncome",data=ea,hue="Attrition",style="Attrition",alpha=0.8, size=ea["NumCompaniesWorked"], sizes=(20, 200), legend="full")

ax.set_xlabel("TotalWorkingYears", fontsize=18)

ax.set_ylabel("MonthlyIncome", fontsize=18)

Plt.show