



Abstract:

Attrition is one of the many factors affecting the IT world. Some of its effects are loss of revenue, loss of reliable resource etc. This analysis aims at making a detailed study on Attrition on dataset created by IBM data scientists and forecasting the attrition and the ways to reduce it in future.

CAPSTONE PROJECT

IBM HR ATTRITION RATE

Mentored by :
Sravan Malla

Submitted by:
Abhijit
Akhil
Bhavana
Shubham
Sowndharya

Acknowledgement

The study of IBM attrition was motivated to uncover the factors that lead to employee attrition and explore important questions such as 'show me a breakdown of distance from home by job role and attrition' or 'compare average monthly income by education and attrition'. Thanks to the IBM data scientists for creating this dataset. Our mentor and guide Mr. Sravan Malla was excellent in training us and teaching us all the techniques, his support and encouragement was immense. A sincere thanks Ms. Shambavi for her supervision and teaching, without whom we could not have done this project.

We would like to thank all our peers, through whom the learning grew exponentially.

Table Of Contents:

S NO	TITLE	PAGE NO
1.	INTRODUCTION	7
1.1	INDUSTRY REVIEW	7
1.1.1	CURRENT PRACTICES	7
1.1.2	BACKGROUND RESEARCH	7
1.1.3	LITERATURE SURVEY	8
1.2	DATA DICTIONARY AND PREPROCESSING DATA ANALYSIS	9
1.2.1	DATA ATTRIBUTE DETAILS	10
1.2.2	IRRELEVANT COLUMNS	11
1.2.3	SELECTING THE MOST IMPORTANT FEATURES	11
1.2.4	PROJECT JUSTIFICATION	12
2	OVERVIEW OF PROCESSES	12
2.1	DATA EXPLORATION	12
2.1.1	AGE VS ATTRITION	12
2.1.2	BUSINESS TRAVEL VS ATTRITION	13
2.1.3	DEPARTMENT VS ATTRITION	13
2.1.4	DISTANCE FROM HOME VS ATTRITION	14
2.1.5	EDUCATION VS ATTRITION	14
2.1.6	ENVIRONMENT SATISFACTION VS ATTRITION	15
2.1.7	JOB INVOLVEMENT VS ATTRITION	15
2.1.8	JOB ROLE VS ATTRITION	16
2.1.9	JOB LEVEL VS ATTRITION	17
2.1.10	MARITAL STATUS VS ATTRITION	18
2.1.11	GENDER VS ATTRITION	19

2.1.12	NUMBER OF COMPANIES WORKED VS ATTRITION	20
2.1.13	OVER TIME VS ATTRITION	20
2.1.14	RELATIONSHIP SATISFACTION VS ATTRITION	21
2.1.15	WORK LIFE BALANCE VS ATTRITION	22
2.1.16	YEARS IN CURRENT ROLE VS ATTRITION	22
2.1.17	YEARS SINCE LAST PROMOTION VS ATTRITION	23
2.2	DROPPING OF IRRELEVANT COLUMNS	24
2.3	CORRELATION OF DATASET	24
2.4	IMBALANCE DATA DETECTION	25
2.5	TREATMENT OF IMBALANCE DATA	26
2.5.1	OVER SAMPLING	26
2.5.1.1	LOGISTIC REGRESSION	27
2.5.1.2	DECISION TREE	27
2.5.1.3	SVC	27
2.5.1.4	RANDOM FOREST	27
2.5.2	UNDER SAMPLING	28
2.5.2.1	LOGISTIC REGRESSION	29
2.5.2.2	DECISION TREE	29
2.5.2.3	SVC	29
2.5.2.4	RANDOM FOREST	29
2.5.3	SMOTE	30
2.5.3.1	LOGISTIC REGRESSION	30
2.5.3.2	DECISION TREE	31
2.5.3.3	SVC	31
2.5.3.4	RANDOM FOREST	31

3	FEATURE ENGINEERING AND MODEL SELECTION	32
3.1	FEATURE SELECTION	32
3.1.1	BENEFITS OF FEATURE SELECTION	32
3.1.2	BACKWARD SELECTION	32
3.1.2.1	FILTER METHOD	33
3.1.2.2	WRAPPER METHOD	33
3.1.2.3	EMBEDDED METHOD	33
3.1.2.4	CORRELATION MATRIX ANALYSIS	33
3.1.3	UNIVARIATE ANALYSIS	34
3.1.4	LASSO REGRESSION	34
3.1.5	RECURSIVE FEATURE ELIMINATION(RFE)	34
3.2	ASSUMPTIONS	35
3.2.1	REGRESSION	35
3.2.1.1	LOGISTIC REGRESSION	35
3.2.2	CLASSIFICATION	36
3.2.2.1	DECISION TREE	36
3.2.2.2	RANDOM FOREST	36
3.2.2.3	SVM	37
3.3	CLUSTERING	37
3.3.1	PCA	37
3.3.2	K-MEANS CLUSTERING	38
3.3.3	AGGLOMERATIVE CLUSTERING	39
4	MODEL EVALUATION	40
4.1	CROSS VALIDATION	40
4.1.1	SVM LINEAR SEARCH	41

4.1.2	LINEAR REGRESSION	41
4.1.3	NAIVE-BAYES	42
4.1.4	K-NEAREST NEIGHBOR	42
4.1.5	SVM-BF	42
4.1.6	DECISION TREE	42
4.1.7	BAGGING CLASSIFIER	43
4.1.8	ADA BOOST CLASSIFIER	43
4.1.9	GRADIENT BOOST CLASSIFIER	43
4.1.10	RANDOM FOREST	43
5	COMPARISON TO BENCHMARK	44
6	VISUALIZATION	46
7	IMPLICATIONS	46
8	LIMITATIONS	47
9	CONCLUSION	47

1. Introduction

1.1 Industry Review

1.1.1 Current practices:

- Currently IBM artificial intelligence can predict which employees will leave a job with 95 percent accuracy.
- AI, which has replaced 30 percent of IBM's HR staff, can help employees identify new skills training, education, job promotions and raises.
- IBM's bet is that the future of work is one in which a machine understands the individual better than the HR individual can alone
- According to the Consumer Technology Association's Future of Work survey. Yet the tech industry is concerned that school systems and universities have not moved fast enough to adjust their curriculum to delve more into data science and machine learning. As a result, companies will struggle to fill jobs in software development, data analytics and engineering.
- IBM is investing \$1 billion in initiatives like apprenticeships to train workers for what it calls "new collar" jobs.
- The "new collar" jobs could range from working at a call center to developing apps or becoming a cyber-analyst at IBM after going through a P-TECH (Pathways in Technology Early College High School) program, which takes six years starting with high school and an associate's degree.

1.1.2 Background Research:

- Earlier top technology services companies across the world have the problem of high employee turnover in an industry that typically witnesses high attrition rates.
 - However, with the emergence of predictive algorithms and tools that reads data in seconds and provide crucial insights and indicators into employee behaviour And the likes of IBM are investing heavily on such predictive analytical tools, which otherwise is wasted on hires that may not have been the best fit for the company in the first place.
 - targeted at the people who are the most productive and the most likely to stay, because you can save a lot of money on people who have a low probability of staying with you.
IBM found that they could save millions of dollars by targeting our benefits and compensation for the people who are most productive and most likely to stay with IBM.
 - Companies like IBM have, therefore, been forced to find new ways of retaining employees and have had to take a fresh look at traditional technology industry metrics. And the results are starting to show, as companies are now deploying a more strategic approach towards hiring and only investing on talent that is likely to help drive long-term growth.
- Challenges at this point for IBM is to try to find some general models that will bring down the cost of doing this work, so that we can apply it to help people. In the past few years, companies have gathered socioeconomic data from incoming engineers such as the educational qualifications of parents and household incomes. Armed with such information, human resource (HR) departments are able to use algorithms and analytics in recruitment.

1.1.3 Literature Survey - Publications, Application, past and undergoing research

Literature review - A review of the literature survey on employee turnover by Henry Ongori

“Employee turnover” as the term is widely used in business circles. Although most of the researchers focus on the causes of employee turnover but little has been done on examining the sources of employee turnover, effects and advising various strategies which can be used by managers in various organisations to ensure that there is employee continuity in their organisations to enhance organizational competitiveness. This paper examines the sources of employee turnover, effects and forwards some strategies on how to minimize employee turnover in organisations.

-Sources of employee turnover :-Job related factors, Voluntary vs. involuntary turnover, Organisational instability

-Effects of employee turnover:- if employee turnover is not managed properly it would affect the organization adversely in terms of personnel costs and in the long run it would affect its liquidity position. However, voluntary turnover incurs significant cost, both in terms of direct costs (replacement, recruitment and selection, temporary staff, management time), and also (and perhaps more significantly) in terms of indirect costs (morale, pressure on remaining staff, costs of learning, product/service quality, organisational memory) and the loss of social capital.

- Strategies to minimize employee turnover:-Employee turnover attributable to poor selection procedures, employee turnover attributable to wage rates which produce earnings that are not competitive with other firms in the local labour market. Knowledge accessibility, the extent of the organisation’s “collaborativeness” and its capacity for making knowledge and ideas widely available to employees, would make employees to stay in the organisation.

Past and undergoing research:-

Getting rid of the current HR system:

-Traditional human resource departments has been divided between a self-service system, where employees are forced to be their own career managers, and a defensive system to deal with poor performers. to overcome these bringing AI everywhere will get rid of the [existing] self-service system. IBM employees no longer need to decipher which programs will help them upskill; its AI suggests to each employee what they should be learning in order to get ahead in their career.

-Poor performers, meanwhile, will not be a “problem” that is dealt with only by managers, HR, legal and finance, but by solutions groups — IBM is using “pop-up” solutions centres to assist managers in seeking better performance from their employees .many companies have relied on centres of excellence — specialized groups or collaborative entities created to focus on areas where there is a knowledge or skills gap within an organization or community.

-But the new era of AI-centered human resources will improve upon something many human-led HR teams can’t handle as effectively as a machine that can crunch millions of data points and learn in new ways. Recognizing the true resource potential of individuals and serving as growth engines for companies.It is at the individual level. You have to know the individual. Skills are the renewable asset.

-The Fourth Industrial Revolution is underway and it is shaping up to be one of the most significant challenges and opportunities of our lifetime. We are already seeing jobs, policies, industries and entire economies shifting as our digital and physical worlds merge.

According to the World Economic Forum, the value of digital transformations in the Fourth Industrial Revolution is estimated at \$100 trillion in the next 10 years alone, across all sectors, industries and geographies.

-As a result, profound transformation of the workforce over the next five to 10 years as analytics and artificial intelligence change job roles at companies in all industries. IBM expect AI to change 100 percent of jobs within the next five to 10 years.

-IBM is also helping to catalyze a national movement to close the skills gap. IBM and the Consumer Technology Association announced the launch of the CTA Apprenticeship Coalition, to create thousands of new apprenticeships in 20 states in January.

-It provides frameworks for more than 15 different apprenticeship roles in fast-growing fields, including software engineering, data science and analytics, cyber security, mainframe system administration, creative design and program management. New apprenticeships will be modelled in large part on IBM's successful apprenticeship program, which launched in 2017, is registered with the United States Department of Labour and has grown nearly twice as fast as expected.

Its goal is to widen the aperture when it comes to hiring by placing the focus on skills rather than specific degrees. From early-career professionals to mid-career transitions and everything in between, these apprenticeships represent a new pathway to success in 21st century careers. including the growing number of new collar roles where a traditional bachelor's degree is not always required. They also offer an opportunity to build in-demand skills without taking on student debt.

1.2 Data Dictionary and Preprocessing Data Analysis:

Range Index: 1470 entries/ 0 to 1469 Data columns (total 35 columns):

Sr.No	Variables Names	Categorization of Variable	Null values Check
1.	Age	Numerical /Discrete	1470 non_null object
2.	Attrition	Categorical	1470 non_null object
3.	BusinessTravel	Categorical	1470 non_null int64
4.	DailyRate	Numerical /Discrete	1470 non_null object
5.	Department	Categorical	1470 non_null int64
6.	DistanceFromHome	Numerical /Discrete	1470 non_null int64
7.	Education	Categorical	1470 non_null object
8.	EducationField	Categorical	1470 non_null int64
9.	EmployeeCount	Numerical /Discrete	1470 non_null int64
10.	EmployeeNumber	Numerical /Discrete	1470 non_null int64
11.	EnvironmentSatisfaction	Categorical	1470 non_null object
12.	Gender	Categorical	1470 non_null int64
13.	HourlyRate	Numerical/Discrete	1470 non_null int64
14.	JobInvolvement	Categorical	1470 non_null int64
15.	JobLevel	Categorical	1470 non_null object
16.	JobRole	Categorical	1470 non_null int64
17.	JobSatisfaction	Categorical	1470 non_null object
18.	MaritalStatus	Categorical	1470 non_null int64
19.	MonthlyIncome	Numerical / Discrete	1470 non_null int64

20.	MonthlyRate	Numerical/Discrete	1470 non_null int64
21.	NumCompaniesWorked	Numerical /Discrete	1470 non_null object
22.	Over18	Categorical	1470 non_null object
23.	OverTime	Categorical	1470 non_null int64
24.	PercentSalaryHike	Numerical/Discrete	1470 non_null int64
25.	PerformanceRating	Categorical	1470 non_null int64
26.	RelationshipSatisfaction	Categorical	1470 non_null int64
27.	StandardHours	Numerical/Discrete	1470 non_null int64
28.	StockOptionLevel	Categorical	1470 non_null int64
29.	TotalWorkingYears	Numerical/ Discrete	1470 non_null int64
30.	TrainingTimesLastYear	Numerical/Discrete	1470 non_null int64
31.	WorkLifeBalance	Categorical	1470 non_null int64
32.	YearsAtCompany	Numerical/ Discrete	1470 non_null int64
33.	YearsInCurrentRole	Numerical/ Discrete	1470 non_null int64
34.	YearsSinceLastPromotion	Numerical /Discrete	1470 non_null int64
35.	YearsWithCurrManager	Numerical /Discrete	1470 non_null int64

In this dataset , we don't have any null values in the dataset hence dataset is free from null values.

1.2.1 Data Attribute Details :

In the dataset, we encoded all the categorical values into Numerical Values as shown.

Education	1(Below College)/ 2(College)/ 3(Bachelor)/ 4(Master)/ 5(Doctor)
EnvironmentSatisfaction	1(Low)/ 2(Medium)/ 3(High)/ 4(Very High)
JobInvolvement	1(Low)/ 2(Medium)/ 3(High)/ 4(Very High)
JobSatisfaction	1(Low)/ 2(Medium)/ 3(High)/ 4(Very High)
PerformanceRating	1(Low)/ 2(Good)/ 3(Excellent)/ 4(Outstanding)
RelationshipSatisfaction	1(Low)/ 2(Medium)/ 3(High)/ 4(Very High)
WorkLifeBalance	1(Bad)/ 2(Good)/ 3(Better)/ 4(Best)
Gender	1(Male) /2(Female)
Education Field	1(Human Resources)/2(Life Sciences)/3(Medical)/ 4(Marketing)/ 5(Technical Degree)/6(Other).
Job Roles	1(Sales Executive)/2(Research Scientist)/3(Laboratory Technician)/ 4(Manufacturing Director)/5(Healthcare Representative)/6(Manager)/ 7(Sales Representative)/ 8(Research Director)/ 9(Human Resources).
Marital Status	1(Single)/2(Married)/ 3(Divorced)
BusinessTravel	1(Non-Travel)/ 2(Travel Rarely)/ 3(Travel Frequently)
Department	1(Sales) / 2(Research & Development)/3(Human Resources)
Monthly Income	below 5000/between 5000 and 10000/between 10000 and 15000/between 15000 and 20000/above 20000
Overtime	Yes/No
Attrition	1(Yes)/ 0(No)

1.2.2 Irrelevant Columns : There are some features which have irrelevant data or cannot contribute to the target variable. Hence we dropped them

Over18	Since the Over18 feature has all the same values ('Y'), We dropped this feature as it does not contribute to the dependant variable.
StandardHours	since StandardHours is same for all the records, we dropped that column
EmployeeCount	since EmployeeCount is irrelevant to the data set, we dropped that column as well

1.2.3 Selecting the Most Important Features :

1.	'Age',
2.	'BusinessTravel',
3.	'Department',
4.	'DistanceFromHome',
5.	'EnvironmentSatisfaction',
6.	'Gender',
7.	'JobInvolvement',
8.	'JobLevel',
9.	'JobRole',
10.	'JobSatisfaction',
11.	'MaritalStatus',
12.	'NumCompaniesWorked',
13.	'OverTime',
14.	'RelationshipSatisfaction',
15.	'WorkLifeBalance',
16.	'YearsInCurrentRole',
17.	'YearsSinceLastPromotion'

The Important Features are Selected using RFE Method.

1.2.4 Project Justification :

- This is a fictional data set created by IBM data scientists contains employee attrition data
- The Data set is about IBM HR analytics on Attrition.
- This is a classification problem. The dependant variable is **Attrition**.
- We used Classification model algorithms like Logistic Regression, K-NN, Decision Tree, Random Forest, etc.,
- We used bagging and boosting techniques for increasing the accuracy and performance of the model.

- Explore Important questions such as 'breakdown of distance from home by job role and attrition' and 'comparing the average monthly income by education and Attrition'

2. Overview of Processes:

2.1 Data Exploration (EDA):

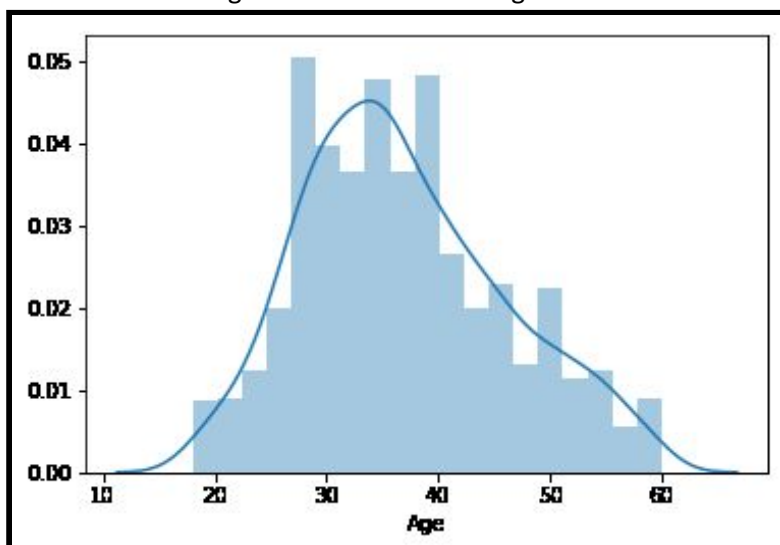
Relationship between the variables:

There are a total of 35 features among which Attrition is the Target variable. Let us see the various visual analytics as follows:

2.1.1 Age vs Attrition:

The distribution of age is continuous. It is spread from 18 to 60. In the age group of 25 to 40, many people are working.

Figure 1: Distribution of age



Attrition rate below age 20 is 58.82%

Attrition rate of age between 20 and 30 is 5.10%

Attrition rate of age between 30 and 40 is 14.23%

Attrition rate of age between 40 and 50 is 9.93%

Attrition rate of age between 50 and 60 is 13.04%

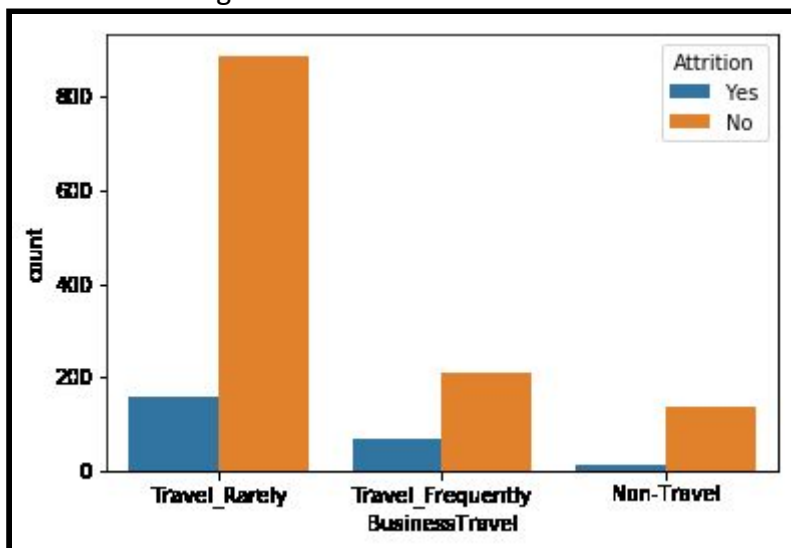
From the above statistics, we can see that the attrition is very high below the age of 20 and very high in the age of 40 to 50.

2.1.2 Business Travel vs Attrition:

This feature describes the type of travel the employee do. There are three categories in business travel. They are:

Travel_Rarely	1043
Travel_Frequently	277
Non-Travel	150

Figure 2: Business travel vs Attrition



Attrition rate when they travel rarely is 14.95%

Attrition rate when they travel frequently is 24.9%

Attrition rate when they dont travel is 8%

2.1.3 Department vs Attrition:

There are three departments namely Sales, Research & Development, Human Resources.

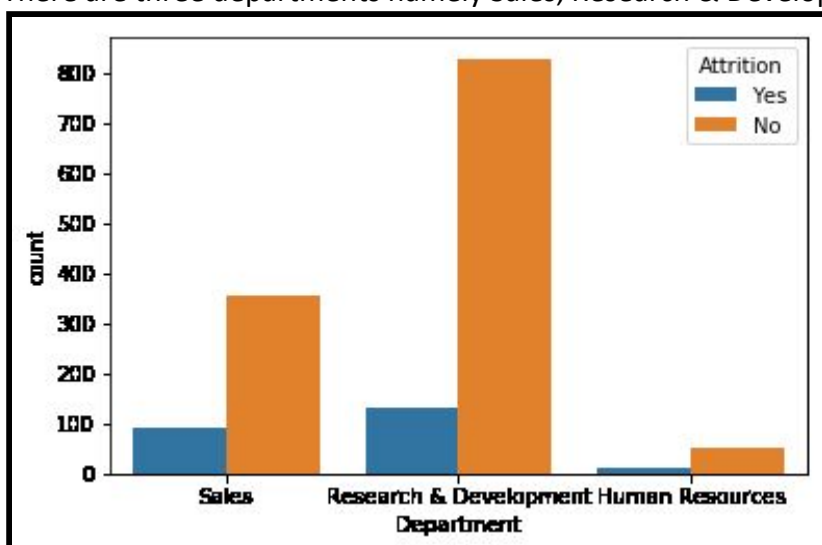


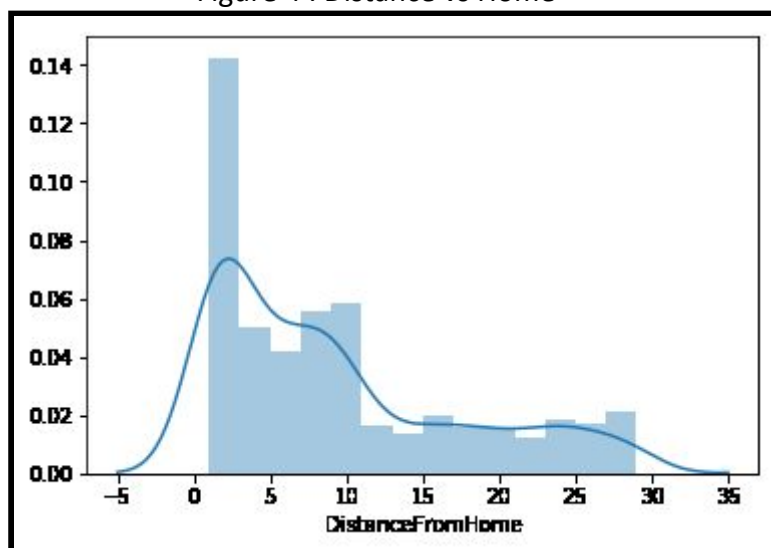
Figure 3: Department vs attrition

Attrition rate Sales is 20.62 %
 Attrition rate Research & Development is 13.83%
 Attrition rate Human Resources is 19.04%

2.1.4 Distance from Home vs Attrition:

This feature depicts the total distance from home to office location. The lowest distance from their home to the workplace is 1 km and the farthest is 29kms. 75% of the people are within 14 kms distance from the workplace.

Figure 4 : Distance vs Home



2.1.5 Education vs Attrition:

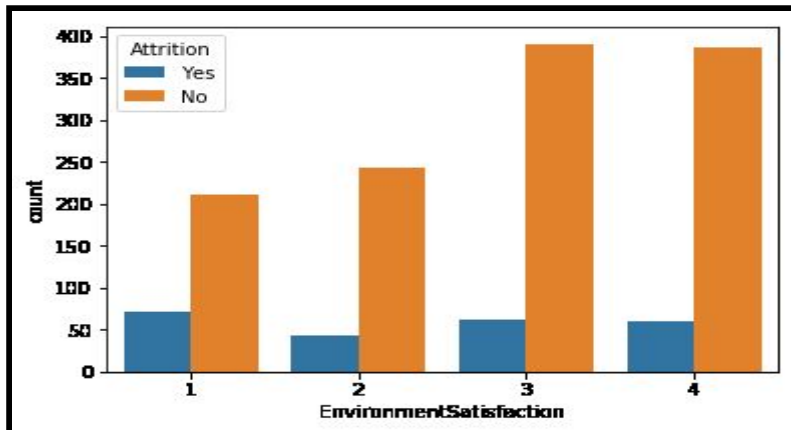
There are five levels in education. They are :

- 1 – High school
- 2 – Diploma
- 3 – Graduate
- 4 – Higher Graduate
- 5 – Doctorate

2.1.6 Environment Satisfaction vs Attrition:

There are four levels in environment satisfaction. 1 being the lowest and 4 being the highest.

Figure 5 : Environment satisfaction vs Attrition

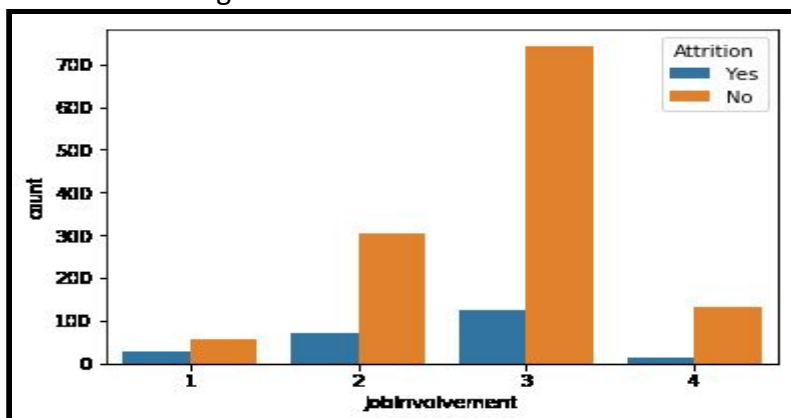


Attrition for Environment Satisfaction level - 1 is 25.35%
 Attrition for Environment Satisfaction level - 2 is 14.98%
 Attrition for Environment Satisfaction level - 3 is 13.68%
 Attrition for Environment Satisfaction level - 4 is 13.45%

2.1.7 Job Involvement vs Attrition:

This feature describes the rate at which the employee involved in their job. There are four levels 1 being the lowest and 4 being 4 the highest.

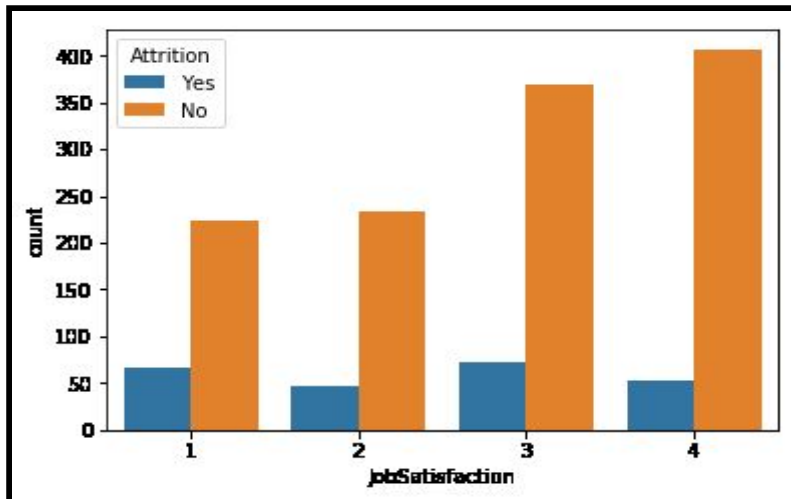
Figure 6: Job involvement vs Attrition



Attrition rate for Job Involvement level 1 is 33.73%
 Attrition rate for Job Involvement level 2 is 18.93%
 Attrition rate for Job Involvement level 3 is 14.40%
 Attrition rate for Job Involvement level 4 is 9.02%
 Job Satisfaction vs Attrition:

The rate at which the employee is satisfied with the job. There are four levels 1 being the lowest and 4 being the highest.

Figure 7: Job Satisfaction vs Attrition



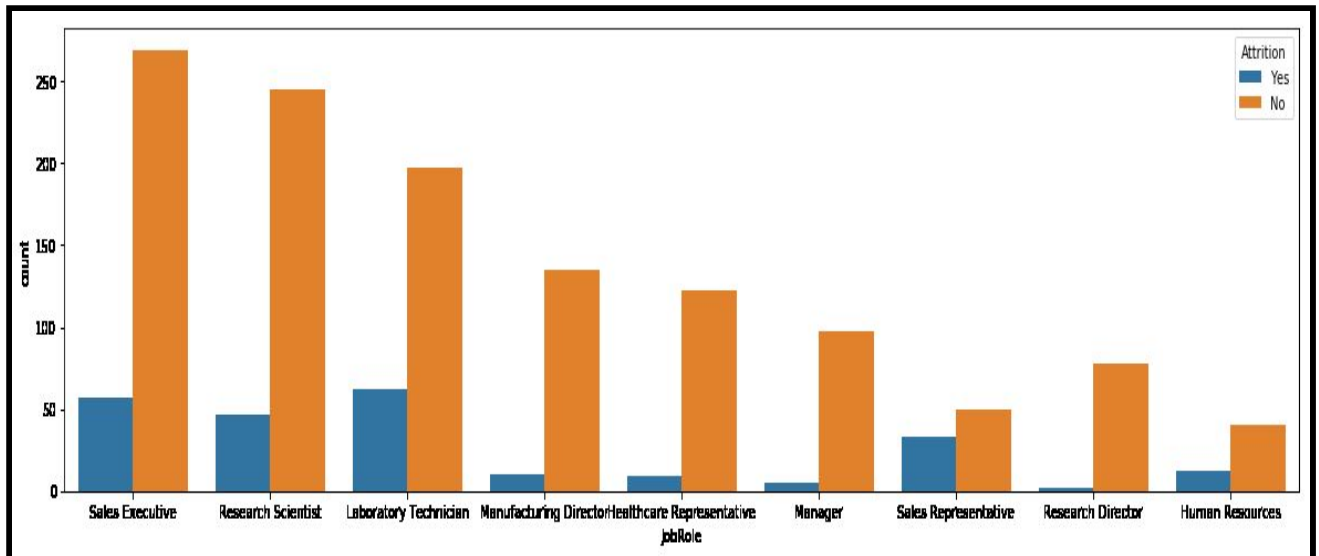
Attrition rate for Job Satisfaction level 1 is 22.83%
 Attrition rate for Job Satisfaction level 2 is 16.42%
 Attrition rate for Job Satisfaction level 3 is 16.51%
 Attrition rate for Job Satisfaction level 4 is 11.32%

2.1.8 Job Role vs Attrition :

There are 9 job roles. They are :

1. Sales Executive
2. Research Scientist
3. Laboratory Technician
4. Manufacturing Director
5. Healthcare Representative
6. Manager
7. Sales Representative
8. Research Director
9. Human Resources

Figure 8: Job role vs Attrition

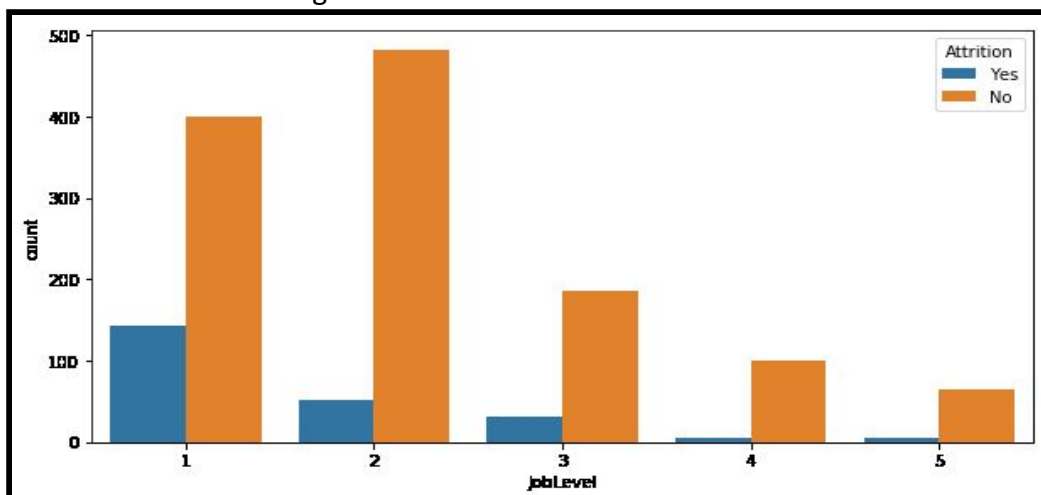


Attrition rate for Job role Sales Executive is 17.48%
 Attrition rate for Job role Research Scientist is 16.09%
 Attrition rate for Job role Laboratory Technician is 23.93%
 Attrition rate for Job role Manufacturing Director is 6.89%
 Attrition rate for Job role Healthcare Representative is 6.87%
 Attrition rate for Job role Manager is 4.90%
 Attrition rate for Job role Sales Representative is 39.75%
 Attrition rate for Job role Research Director is 2.5%
 Attrition rate for Job role Human Resources is 23.07%

2.1.9 Job Level vs Attrition:

There are 5 job levels 1 being the lowest and 5 being the highest.

Figure 9: Job level vs Attrition



Attrition rate for Job Level 1 is 26.33%
 Attrition rate for Job Level 2 is 9.73%

Attrition rate for Job Level 3 is 14.67%

Attrition rate for Job Level 4 is 4.71%

Attrition rate for Job Level 5 is 7.24%

2.1.10 Marital Status vs Attrition :

There are 3 categories in marital status. They are:

Single

Married

Divorced

Figure 10: Marital status vs Gender

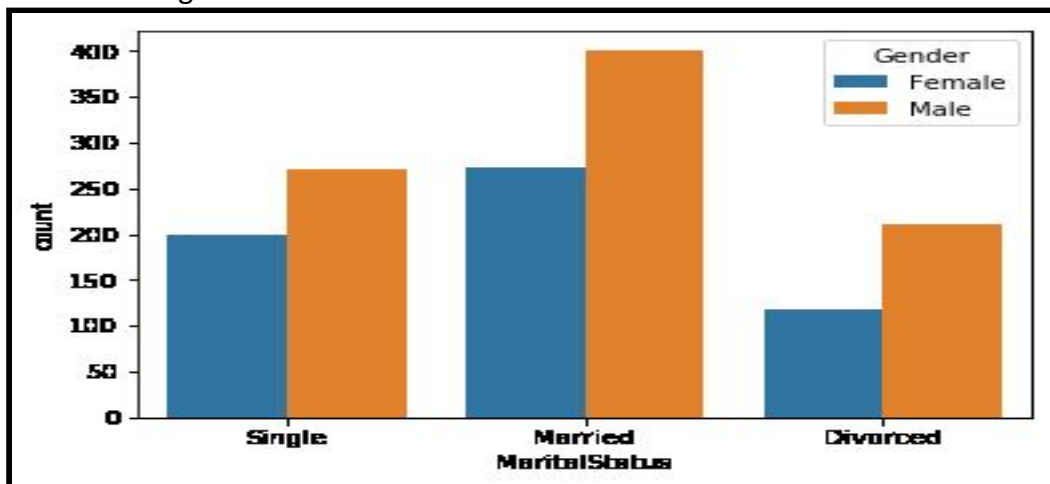
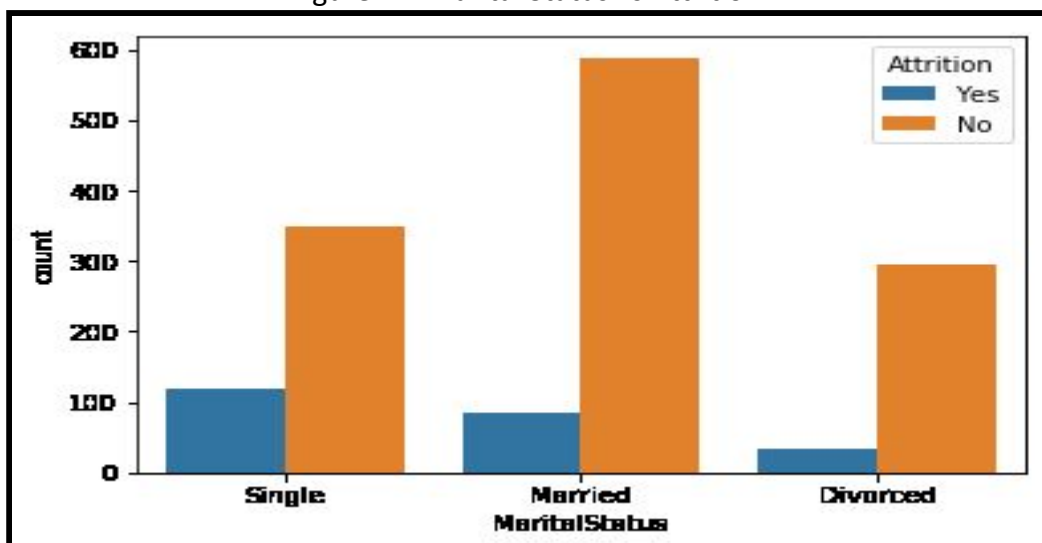


Figure 11: Marital Status vs Attrition



Attrition rate when Marital Status Single is 25.53

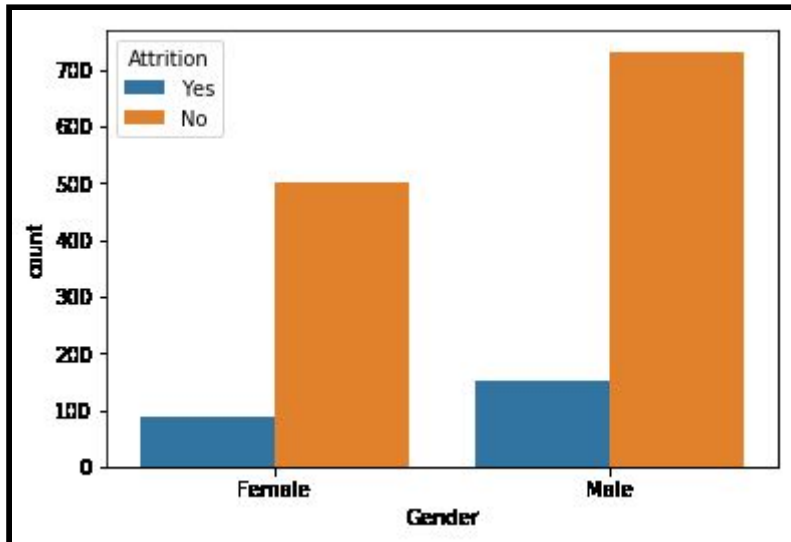
Attrition rate when Marital Status Married is 12.48

Attrition rate when Marital Status Divorced is 10.09

2.1.11 Gender vs Attrition:

This field describes gender of employees.

Figure 12 : Gender vs Attrition



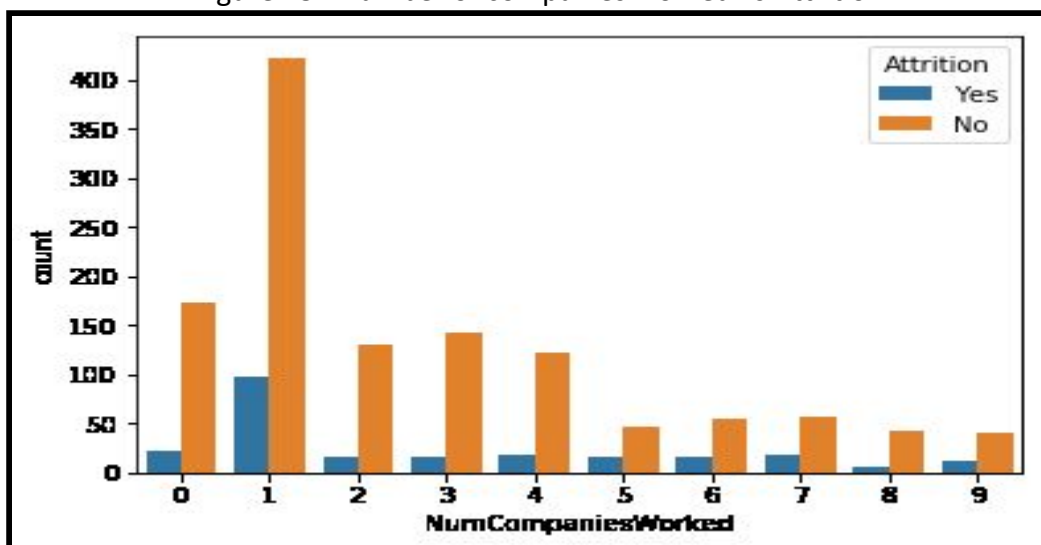
Attrition rate of Female is 14.79 %

Attrition rate of Male is 17 %

2.1.12 Number of Companies Worked vs Attrition:

This field shows the number of companies that the employee worked before. The maximum limit is 9.

Figure 13 : Number of companies worked vs Attrition



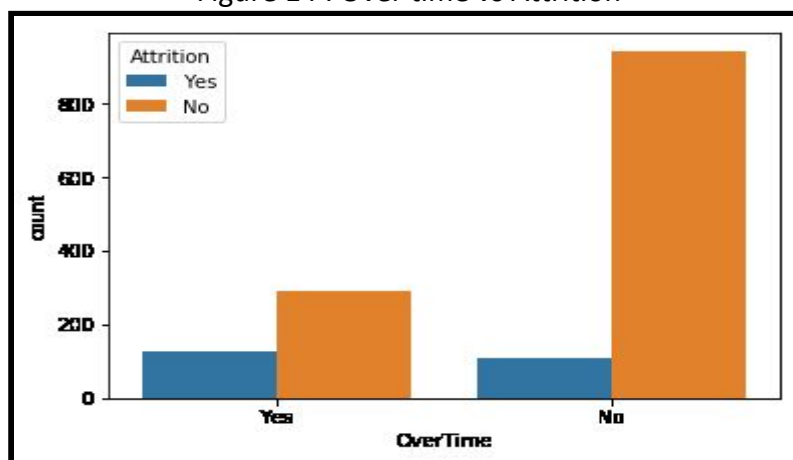
Attrition rate when Number of Companies worked of count 0 is 11.67%
 Attrition rate when Number of Companies worked of count 1 is 18.80%
 Attrition rate when Number of Companies worked of count 2 is 10.95%
 Attrition rate when Number of Companies worked of count 3 is 10.06%
 Attrition rate when Number of Companies worked of count 4 is 12.23%
 Attrition rate when Number of Companies worked of count 5 is 25.39%
 Attrition rate when Number of Companies worked of count 6 is 22.85%
 Attrition rate when Number of Companies worked of count 7 is 22.97%
 Attrition rate when Number of Companies worked of count 8 is 12.24%
 Attrition rate when Number of Companies worked of count 9 is 23.07%

As the number of companies worked previously increases, employees tend to leave the current job.

2.1.13 Over Time vs Attrition:

This is a boolean feature with 'Yes' and 'No'.

Figure 14 : Over time vs Attrition

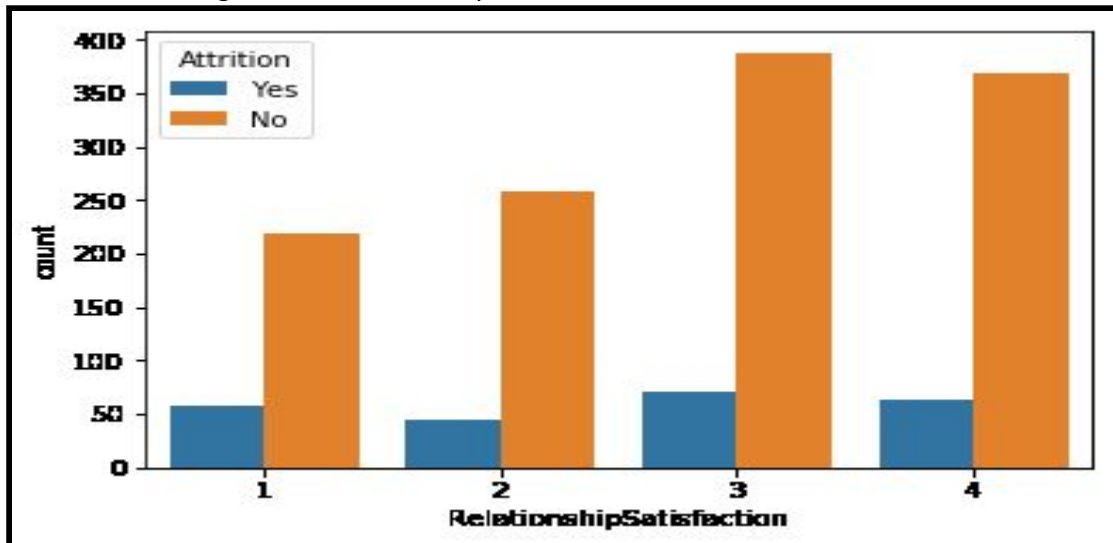


Attrition rate if they have done Overtime is 8.63%
 Attrition rate if they have not done Overtime is 7.48%

2.1.14 Relationship Satisfaction vs Attrition:

This field depicts the level of satisfaction of the employees in their relationships. 1 being highest and 4 being the lowest.

Figure 15: Relationship satisfaction vs Attrition



Attrition rate when Relationship Satisfaction 1 is 20.65

Attrition rate when Relationship Satisfaction 2 is 14.85

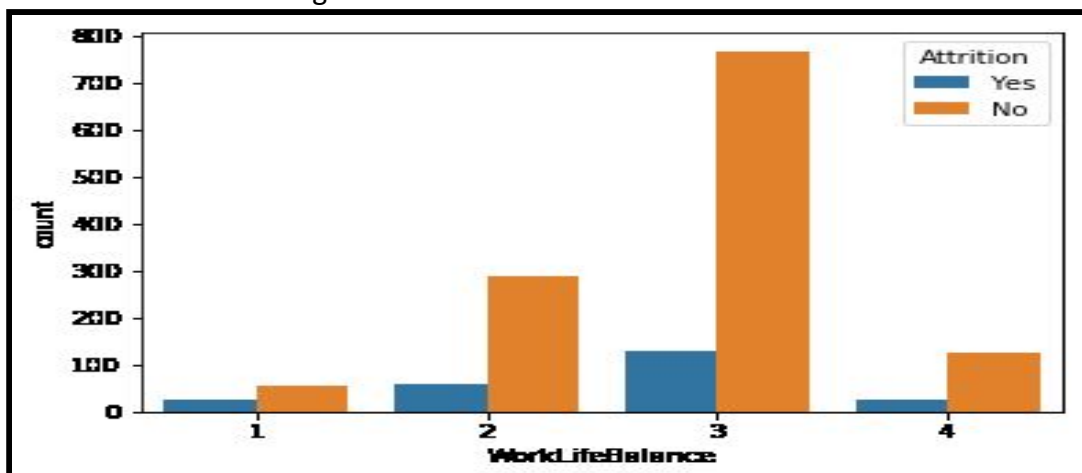
Attrition rate when Relationship Satisfaction 3 is 15.46

Attrition rate when Relationship Satisfaction 4 is 14.81

2.1.15 Work Life Balance vs Attrition :

This field describes the rate at which the employee balances work and life. There are four levels 1 being highest and 4 being the lowest.

Figure 17: Work life balance vs Attrition



Attrition rate when Work life balance is of level 1 is 31.25%

Attrition rate when Work life balance is of level 2 is 16.86%

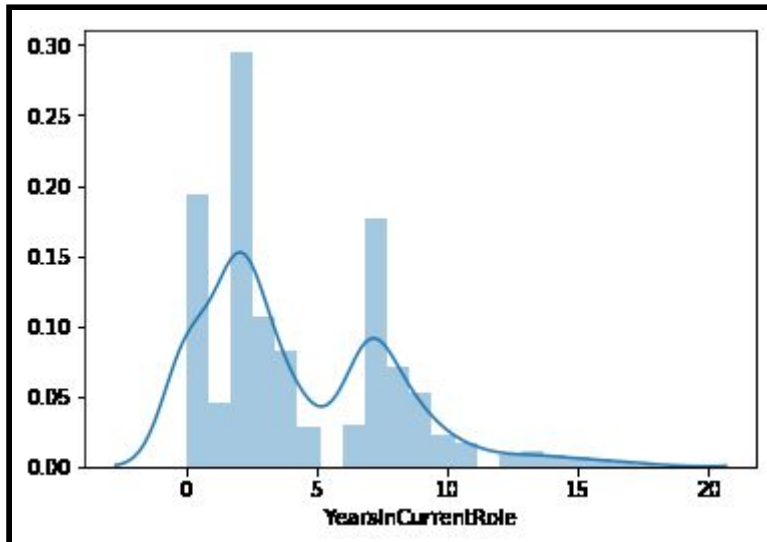
Attrition rate when Work life balance is of 3 is 14.22%

Attrition rate when Work life balance is of 4 is 17.64%

2.1.16 Years In Current Role vs Attrition:

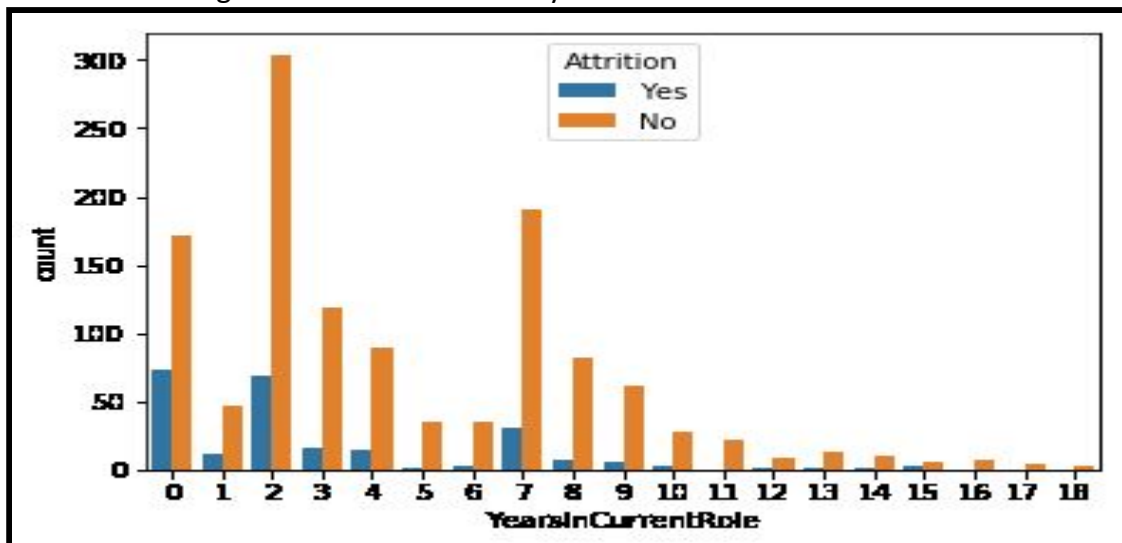
This field describes the number of years the employee is working in the current role.

Figure 18: Years in current role vs Attrition



Attrition rate for working in current role for less than 5 years is 20.06%
 Attrition rate for working in current role between 5 to 10 years is 11.08 %
 Attrition rate for working in current role between 10 and 15 years is 5.26%

Figure 19: Total number of years in current role wise Attrition



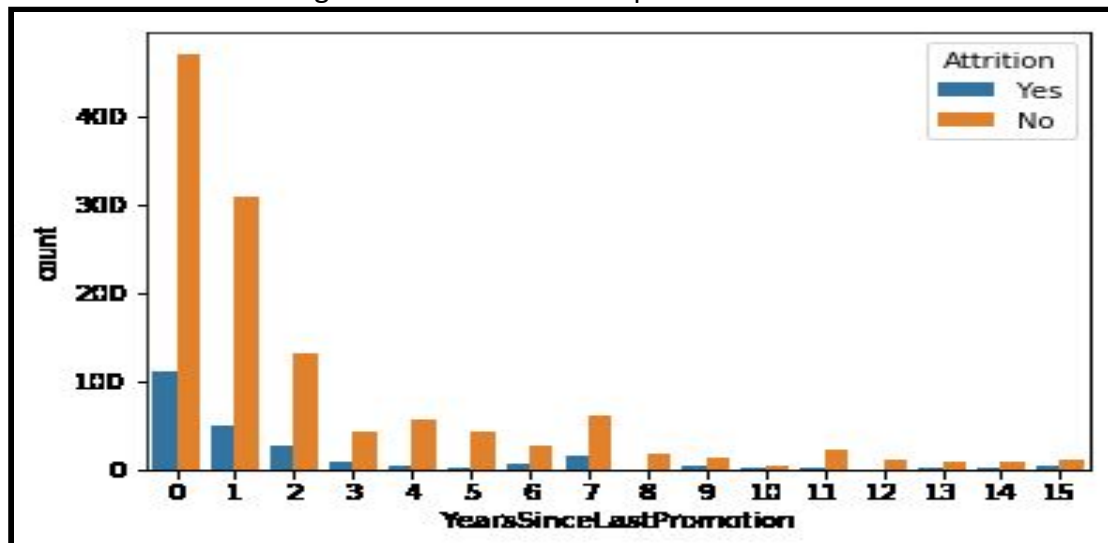
Attrition rate for working in current role for 0 years is 29.91 %
 Attrition rate for working in current role for 1 years is 19.29 %
 Attrition rate for working in current role for 2 years is 18.27 %

Attrition rate for working in current role for 3 years is 11.85 %
 Attrition rate for working in current role for 4 years is 14.42 %
 Attrition rate for working in current role for 5 years is 2.77 %
 Attrition rate for working in current role for 6 years is 5.40 %
 Attrition rate for working in current role for 7 years is 13.96 %
 Attrition rate for working in current role for 8 years is 7.86 %
 Attrition rate for working in current role for 9 years is 8.95 %
 Attrition rate for working in current role for 10 years is 6.89 %

2.1.17 Years Since Last Promotion vs Attrition:

This feature describes the number of years since the last promotion. In the dataset, the years range from 0 to 18.

Figure 20: Years since last promotion vs attrition



Attrition rate for 0 years since last promotion is 18.93 %
 Attrition rate for 1 years since last promotion is 13.72 %
 Attrition rate for 2 years since last promotion is 16.98 %
 Attrition rate for 3 years since last promotion is 17.30 %
 Attrition rate for 4 years since last promotion is 8.19 %
 Attrition rate for 5 years since last promotion is 4.44 %
 Attrition rate for 6 years since last promotion is 18.75 %
 Attrition rate for 7 years since last promotion is 21.05 %
 Attrition rate for 9 years since last promotion is 23.52 %
 Attrition rate for 10 years since last promotion is 16.66 %
 Attrition rate for 11 years since last promotion is 8.33 %
 Attrition rate for 13 years since last promotion is 20.0 %
 Attrition rate for 14 years since last promotion is 11.11 %
 Attrition rate for 15 years since last promotion is 23.07 %

2.2 Dropping of irrelevant Features:

There are some irrelevant Features which were dropped. For example, 'Standard Hours' has the same value for all the record. It has been dropped.

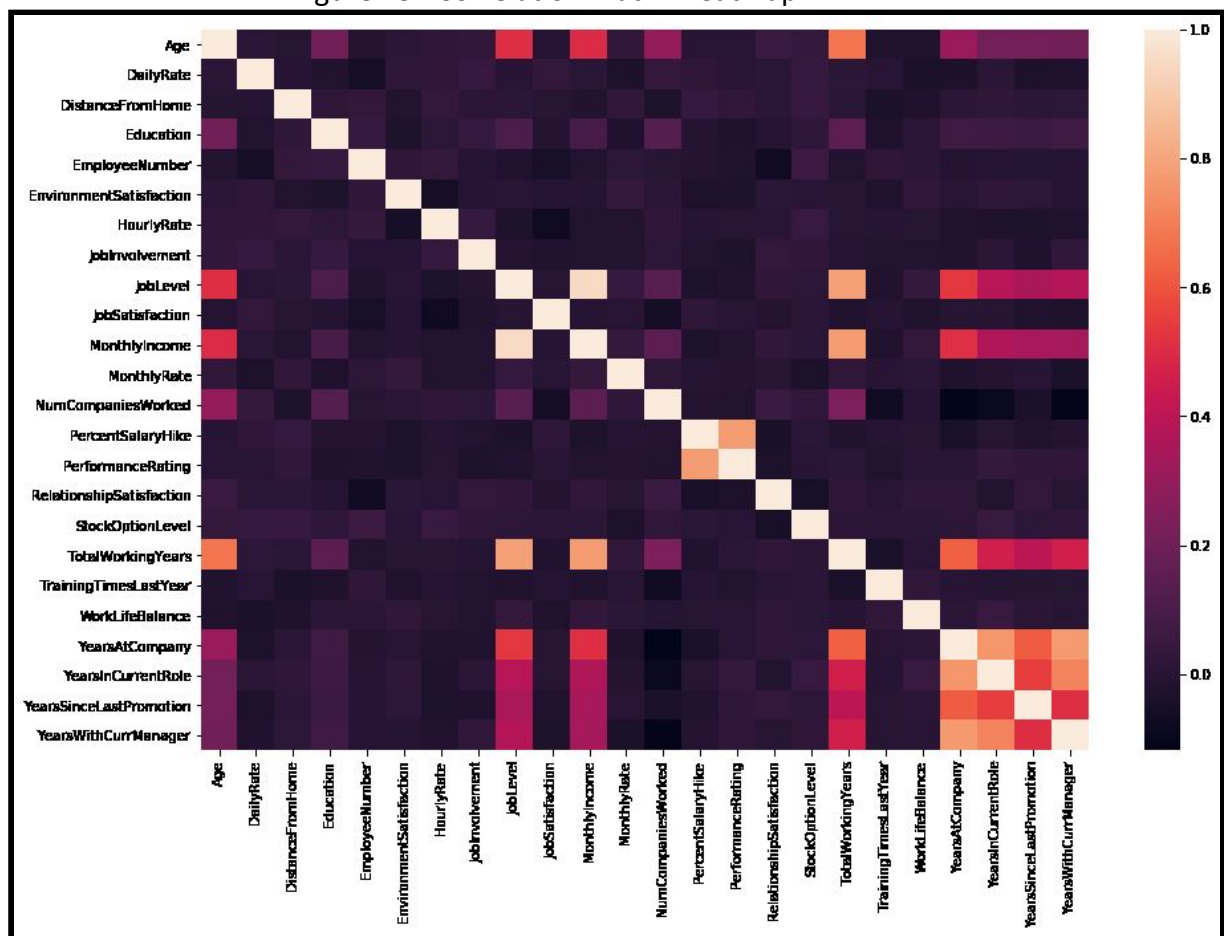
'Employee Count ' is 1 for all the records. Hence , it is dropped.

Since all the employees are above 18 years, 'Over 18' feature is dropped.

2.3 Correlation of the Dataset:

Now we have removed NaN values, outliers, and irrelevant data. The correlation of the full dataset is as below:

Figure 13 : Correlation matrix heatmap



2.4 Imbalance Data detection:

In our dataset, the target feature is 'Attrition'. Attrition field has two values 'Yes' and 'No' , 'Yes' denoting the employee moves out of the company and 'No' denoting that they don't move out. From the records ,let us see the count of the categories:

```
In [522]: ## Attrition =1 percentage
print('Percentage of attrition is ',ibn[ibn['Attrition']==1]['Attrition'].count() / ibn['Attrition'].count() *100)

Percentage of attrition is  16.122448979591837

In [524]: ## Attrition =0 percentage
print('Percentage of no-attrition is ',ibn[ibn['Attrition']==0]['Attrition'].count() / ibn['Attrition'].count() *100)

Percentage of no-attrition is  83.87755102040816
```

It Clearly states that the Data has imbalance. Majority class is of no-Attrition data . The ratio is 83:16.

2.5 Treatment of Imbalance Data:

In treatment of imbalance data, there are popularly three methods. They are :

- Over Sampling
- Under Sampling
- Smote Analysis

2.5.1 Over Sampling:

Over sampling is the process of converting the minority class into majority class. i.e, Expanding the minority class to majority class count.

Oversampling

```
In [532]: from sklearn.utils import resample

# Separate input features and target
y = ibm.Attrition
X = ibm.feature1

# setting up testing and training sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=27)

# concatenate our training data back together
X = pd.concat([X_train, y_train], axis=1)

# separate minority and majority classes
no_attr = X[X.Attrition==0]
attr = X[X.Attrition==1]

# upsample minority
attr_upsampled = resample(attr,
                           replace=True, # sample with replacement
                           n_samples=len(no_attr), # match number in majority class
                           random_state=27) # reproducible results

# combine majority and upsampled minority
upsampled = pd.concat([no_attr, attr_upsampled])

# check new class counts
upsampled.Attrition.value_counts()
```

```
Out[532]: 1    933
          0    933
          Name: Attrition, dtype: int64
```

Here the Class 1 is repeated to match the number of Class 0. After Over sampling let us carry out some of the modelling Techniques:

- Logistic Regression
- Decision Tree
- SVC
- Random Forest

2.5.1.1 Logistic Regression:

After oversampling, we used Logistic regression model to find the accuracy.

```
Accuracy score using Logistic regression after over sampling 0.7717391304347826
-----
F1 score using Logistic regression after over sampling 0.5531914893617021
-----
Recall score using Logistic regression after over sampling 0.7647058823529411
```

Here the accuracy score is 77%.

2.5.1.2 Decision Tree:

After over-sampling , we used Decision tree with Gini as criterion to find the accuracy.

```
Accuracy score using Decision Tree - gini after over sampling 0.779891304347826
-----
F1 score using Decision Tree - gini after over sampling 0.42553191489361697
-----
Recall score using Decision Tree - gini after over sampling 0.4411764705882353
-----
```

Here the accuracy score is 77.9% which is slightly higher than logistic regression

After over-sampling, we used decision tree with Entropy as criterion to find the accuracy.

```
Accuracy score using Decision Tree - entropy after over sampling 0.779891304347826
-----
F1 score using using Decision Tree - entropy after over sampling 0.42553191489361697
-----
Recall score using using Decision Tree - entropy after over sampling 0.4411764705882353
-----
```

Here the accuracy score is similar to Decision tree - Gini.

2.5.1.3 SVC :

After Oversampling, we used SVC to get the accuracy.

```
Accuracy on training set using SVC after oversampling : 1.00
Accuracy on test set using SVC after oversampling : 0.84
```

The accuracy has increased to 84 %

2.5.1.4 Random Forest:

After oversampling, we used Random forest to get accuracy score.

```
1.109375
[[291  9]
 [ 41 27]]
      precision    recall  f1-score   support

      0       0.88      0.97      0.92       300
      1       0.75      0.40      0.52        68

   micro avg       0.86      0.86      0.86       368
   macro avg       0.81      0.68      0.72       368
weighted avg       0.85      0.86      0.85       368
```

Here the accuracy is increased to 88% .

2.5.2 Under sampling:

The wider the gap between the training score and the cross validation score, the more likely your model is overfitting (high variance). If the score is low in both training and cross-validation sets this is an indication that our model is underfitting (high bias). Logistic Regression Classifier shows the best score in both training and cross-validating sets.

```
In [148]: no_attr_downsampled = resample(no_attr,
                                         replace = False, # sample without replacement
                                         n_samples = len(attr), # match minority n
                                         random_state = 21) # reproducible results

# combine minority and downsampled majority
downsampled = pd.concat([no_attr_downsampled, attr])

# checking counts
downsampled.Attrition.value_counts()

Out[148]: 1    169
          0    169
          Name: Attrition, dtype: int64
```

We use four models. They are :

- Logistic Regression
- Decision Tree
- SVC
- Random Forest

2.5.2.1 Logistic Regression:

After Under sampling, we used logistic regression to find accuracy.

```
Accuracy score using logistic regression after under sampling  0.7581521739130435
-----
F1 score using Logistic regression after under sampling  0.5340314136125655
-----
Recall score using Logistic regression after under sampling  0.75
```

The accuracy score is reduced compared to Oversampling.

2.5.2.2 Decision Tree:

After under sampling, we use decision tree with Gini criterion to find accuracy.

```
Accuracy Score using Decision tree - gini after under sampling : 0.7065217391304348
-----
F1 score using Decision tree - gini after under sampling : 0.4375
-----
Recall score using Decision tree - gini after under sampling : 0.6176470588235294
```

We can see that the accuracy score is getting reduced.

After under sampling, we use decision Tree with Entropy as criterion to find accuracy.

```
Accuracy Score using Decision tree - entropy with under sampling : 0.6956521739130435
-----
F1 score using Decision tree - entropy with under sampling : 0.42857142857142855
-----
Recall score using Decision tree - entropy with under sampling : 0.6176470588235294
```

Accuracy of Entropy criterion is similar to Gini in decision tree modelling.

2.5.2.3 SVC:

After under sampling, we used SVC modelling for accuracy rate.

```
Accuracy on training set using SVC with under sampling : 0.99
Accuracy on test set using SVC with under sampling : 0.58
```

The accuracy is reducing very much. It is reducing to 58%

2.5.2.4 Random Forest :

After under sampling, we are using random forest to get accuracy score.

```
0.515625
[[231  69]
 [ 17  51]]
```

	precision	recall	f1-score	support
0	0.93	0.77	0.84	300
1	0.42	0.75	0.54	68
micro avg	0.77	0.77	0.77	368
macro avg	0.68	0.76	0.69	368
weighted avg	0.84	0.77	0.79	368

Here the accuracy to predict 0 is very good but for 1 it has decreased a lot.

2.5.3 SMOTE Analysis :

SMOTE stands for Synthetic Minority Over-sampling Technique. Unlike Random UnderSampling, SMOTE creates new synthetic points in order to have an equal balance of the classes. This is another alternative for solving the "class imbalance problems".

We used four models using SMOTE . They are :

- Logistic Regression
- Decision Tree
- SVC
- Random Forest

2.5.3.1 Logistic regression:

```

SMOTE Analysis

In [153]: from imblearn.over_sampling import SMOTE

# Separate input features and target
y = ibm.Attrition
X = ibm.Feature1

# setting up testing and training sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=27)

sm = SMOTE(random_state=27, ratio=1.0)
X_train, y_train = sm.fit_sample(X_train, y_train)
smote = LogisticRegression(solver='liblinear').fit(X_train, y_train)

smote_pred = smote.predict(X_test)

# Checking accuracy
print('Accuracy Score using Logistic Regression with SMOTE analysis : ',accuracy_score(y_test, smote_pred))
print('-----')
print('Accuracy Score using Logistic Regression with SMOTE analysis : ',f1_score(y_test, smote_pred))
print('-----')
print('Accuracy Score using Logistic Regression with SMOTE analysis : ',recall_score(y_test, smote_pred))

Accuracy Score using Logistic Regression with SMOTE analysis :  0.842391304347826
-----
Accuracy Score using Logistic Regression with SMOTE analysis :  0.6329113924058633
-----
Accuracy Score using Logistic Regression with SMOTE analysis :  0.7352941176470589
  
```

From the above, we can see that the accuracy score is increased to 84%

2.5.3.2 Decision Tree :

We used Decision Tree with Gini criterion after SMOTE.

```

Accuracy Score using Decision Tree - gini with SMOTE analysis:  0.720108695652174
-----
F1 score using Decision Tree - gini with SMOTE analysis 0.38323353293413176
-----
Recall score using Decision Tree - gini with SMOTE analysis 0.47058823529411764
  
```

Accuracy is comparatively less to Logistic regression but increased from under sampling.

We used decision Tree with Entropy criterion after SMOTE.

```
Accuracy Score using Decision tree - entropy with SMOTE analysis: 0.720108695652174
-----
F1 score using Decision tree - entropy with SMOTE analysis 0.38323353293413176
-----
Recall score using Decision tree - entropy with SMOTE analysis 0.47058823529411764
```

Accuracy is comparatively less to Logistic regression but increased from under sampling.

2.5.3.3 SVC:

After applying SMOTE , we are applying SVC for finding accuracy.

```
Accuracy on training set using SVC with SMOTE analysis : 1.00
Accuracy on test set using SVC with SMOTE analysis : 0.81
```

Accuracy is comparatively less to Logistic regression but increased from under sampling.

2.5.3.4 Random Forest :

After applying SMOTE , we tried Random forest for accuracy.

```
0.515625
[[231  69]
 [ 17  51]]
```

	precision	recall	f1-score	support
0	0.93	0.77	0.84	300
1	0.42	0.75	0.54	68
micro avg	0.77	0.77	0.77	368
macro avg	0.68	0.76	0.69	368
weighted avg	0.84	0.77	0.79	368

Accuracy is still increasing for no attrition class but reduced for attrition class.

3. Feature Engineering and Model selection

3.1 Feature Selection:

Feature Selection is one of the main steps of the preprocessing phase as the features which we choose directly affects the model performance. While some of the features seem to be less useful in terms of the context, others seem to equally useful. The better features we use the better our model will perform.

Irrelevant or partially relevant features can negatively impact model performance. Feature selection and Data cleaning should be the first and most important step of your model designing.

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features.

3.1.1 Benefits of Feature selection:

Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.

Improves Accuracy: Less misleading data means modeling accuracy improves.

Reduces Training Time: fewer data points reduce algorithm complexity and algorithms train faster.

Some of the feature selection techniques are as follows:

3.1.2 Backward Selection

We used the Recursive Feature Elimination technique (a wrapper method) to choose the desired number of most important features.

The Recursive Feature Elimination (or RFE) works by recursively removing attributes and building a model on those attributes that remain.

It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute.

We used it directly from the scikit library by importing the RFE module or function provided by the scikit. But note that since it tries different combinations or the subset of features, it is quite computationally expensive.

3.1.2.1 Filter Method :

Filtering our dataset and taking only a subset of it containing all the relevant features (eg. correlation matrix using Pearson Correlation).

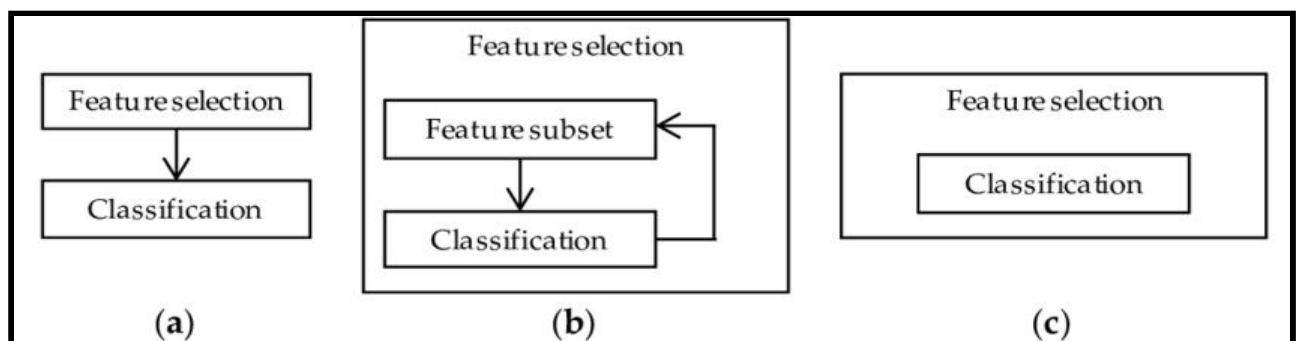
3.1.2.2 Wrapper Method

Wrapper method follows the same objective of the Filter Method but uses a Machine Learning model as its evaluation criteria (eg. Forward/Backward/Bidirectional/Recursive

Feature Elimination). We feed some features to our Machine Learning model, evaluate their performance and then decide if add or remove the feature to increase accuracy. As a result, this method can be more accurate than filtering but is more computationally expensive.

3.1.2.3 Embedded Method

Embedded method is also like the Filter Method also the Embedded Method makes use of a Machine Learning model. The difference between the two methods is that the Embedded Method examines the different training iterations of our ML model and then ranks the importance of each feature based on how much each of the features contributed to the ML model training (eg. LASSO Regularization).



3.1.2.4 Correlation Matrix analysis:

Another possible method which can be used in order to reduce the number of features in our dataset is to inspect the correlation of our features with our labels.

Using Pearson correlation our returned coefficient values will vary between -1 and 1:

- If the correlation between two features is 0 this means that changing any of these two features will not affect the other.
- If the correlation between two features is greater than 0 this means that increasing the values in one feature will make increase also the values in the other feature (the closer the correlation coefficient is to 1 and the stronger is going to be this bond between the two different features).
- If the correlation between two features is less than 0 this means that increasing the values in one feature will make decrease the values in the other feature (the closer the correlation coefficient is -1 and the stronger is going to be this relationship between the two different features).

3.1.3 Univariate Selection :

Univariate Feature Selection is a statistical method used to select the features which have the strongest relationship with our correspondent labels. Using the SelectKBest method we

can decide which metrics to use to evaluate our features and the number of K best features we want to keep. Different types of scoring functions are available depending on our needs:

- Classification = chi2, f_classif, mutual_info_classif
- Regression = f_regression, mutual_info_regression

Chi-squared (Chi2) can take as input just non-negative values, therefore, first of all, we scale our input data in a range between 0 and 1.

3.1.4 Lasso Regression:

When applying regularization to a Machine Learning model, we add a penalty to the model parameters to avoid that our model tries to resemble too closely our input data. In this way, we can make our model less complex and we can avoid overfitting (making learn to our model, not just the key data characteristics but also it's intrinsic noise).

One of the possible Regularization Methods is Lasso (L1) Regression. When using Lasso Regression, the coefficients of the inputs features gets shrunken if they are not positively contributing to our Machine Learning model training. In this way, some of the features might get automatically discarded assigning them coefficients equal to zero.

3.1.5 Recursive Feature Elimination (RFE):

Recursive Feature Elimination (RFE) takes as input the instance of a Machine Learning model and the final desired number of features to use. It then recursively reduces the number of features to use by ranking them using the Machine Learning model accuracy as metrics. Creating a for loop in which the number of input features is our variable, it could then be possible to find out the optimal number of features our model needs by keeping track of the accuracy registered in each loop iteration. Using RFE support method, we can then find out the names of the features which have been evaluated as most important (rfe.support return a boolean list in which TRUE represent that a feature is considered as important and FALSE represent that a feature is not considered important).

3.2 Assumptions

3.2.1 Regression:

3.2.1.1 Logistic Regression:

Logistic regression does not require a linear relationship between the dependent and independent variables. Second, the error terms (residuals) do not need to be normally distributed. Third, homoscedasticity is not required. Finally, the dependent variable in logistic regression is not measured on an interval or ratio scale.

However, some other assumptions still apply.

First, binary logistic regression requires the dependent variable to be binary and ordinal logistic regression requires the dependent variable to be ordinal.

Second, logistic regression requires the observations to be independent of each other. In other words, the observations should not come from repeated measurements or matched data.

Third, logistic regression requires there to be little or no multicollinearity among the independent variables. This means that the independent variables should not be too highly correlated with each other.

Fourth, logistic regression assumes linearity of independent variables and log odds. although this analysis does not require the dependent and independent variables to be related linearly, it requires that the independent variables are linearly related to the log odds.

Finally, logistic regression typically requires a large sample size. A general guideline is that you need at a minimum of 10 cases with the least frequent outcome for each independent variable in your model. For example, if you have 5 independent variables and the expected probability of your least frequent outcome is .10, then you would need a minimum sample size of 500 ($10 \times 5 / .10$).

Accuracy of the model using Logistic Regression without any imbalance data correction technique on test set is 0.8707482993197279				

Accuracy of the model using Logistic Regression without any imbalance data correction technique on train set is 0.8818027210884354				

F1 score using logistic regression without any imbalance data correction technique 0.4722222222222222				

Recall score using Logistic Regression without any imbalance data correction technique 0.3469387755102041				

Classification report using logistic regression without any imbalance data correction technique				
	precision	recall	f1-score	support
0	0.88	0.98	0.93	245
1	0.74	0.35	0.47	49
micro avg	0.87	0.87	0.87	294
macro avg	0.81	0.66	0.70	294
weighted avg	0.86	0.87	0.85	294

3.2.2 Classification :

3.2.2.1 Decision Tree:

The below are the some of the assumptions we make while using Decision tree:

At the beginning, the whole training set is considered as the root. Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model. Records are distributed recursively on the basis of attribute values.

Gini:

Accuracy Score using Decision Tree - gini on train set: 0.8571428571428571				

Accuracy Score using Decision Tree - gini on test set: 0.8605442176870748				

F1 score using Decision Tree - Gini without any technique 0.40579710144927533				

Recall score using Decision Tree - Gini without any technique 0.2857142857142857				

Classification Report for Decision tree - gini				
	precision	recall	f1-score	support
0	0.87	0.98	0.92	245
1	0.70	0.29	0.41	49
micro avg	0.86	0.86	0.86	294
macro avg	0.79	0.63	0.66	294
weighted avg	0.84	0.86	0.84	294

Entropy:

Accuracy Score using Decision tree - entropy on train set: 0.8520408163265306				

Accuracy Score using Decision tree - entropy on test set : 0.8537414965986394				

F1 score using Decision tree - Entropy without any technique 0.2456140350877193				

Recall score using Decision tree - Entropy without any technique 0.14285714285714285				

Classification report using Decision tree - Entropy without any technique				
	precision	recall	f1-score	support
0	0.85	1.00	0.92	245
1	0.88	0.14	0.25	49
micro avg	0.85	0.85	0.85	294
macro avg	0.86	0.57	0.58	294
weighted avg	0.86	0.85	0.81	294

3.2.2.2 Random Forest:

No formal distributional assumptions, random forests are non-parametric and can thus handle skewed and multi-modal data as well as categorical data that are ordinal or non-ordinal.

For Test set				
	precision	recall	f1-score	support
0	0.83	0.98	0.90	364
1	0.45	0.06	0.11	77
micro avg	0.82	0.82	0.82	441
macro avg	0.64	0.52	0.51	441
weighted avg	0.77	0.82	0.76	441
For Train set				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	869
1	1.00	1.00	1.00	160
micro avg	1.00	1.00	1.00	1029
macro avg	1.00	1.00	1.00	1029
weighted avg	1.00	1.00	1.00	1029

3.2.2.3 SVM:

The SVM algorithm is implemented in practice using a kernel.

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra, which is out of the scope of this introduction to SVM.

A powerful insight is that the linear SVM can be rephrased using the inner product of any two given observations, rather than the observations themselves. The inner product between two vectors is the sum of the multiplication of each pair of input values.

```
Accuracy on training set without any techniques : 1.00
Accuracy on test set: 0.83
```

3.3 Unsupervised Learning:

3.3.1 PCA:

When you choose to analyse your data using PCA, part of the process involves checking to make sure that the data you want to analyse can actually be analysed using PCA. You need to do this because it is only appropriate to use PCA if your data "passes" four assumptions that are required for PCA to give you a valid result. In practice, checking for these assumptions requires you to use SPSS Statistics to carry out a few more tests, as well as think a little bit more about your data, but it is not a difficult task.

Let's take a look at these four assumptions:

- **Assumption #1:** You have multiple variables that should be measured at the continuous level (although ordinal variables are very frequently used). Examples of continuous variables (i.e., ratio or interval variables) include revision time (measured in hours), intelligence (measured using IQ score), exam performance (measured from 0 to 100), weight (measured in kg), and so forth. Examples of ordinal variables commonly used in PCA include a wide range of Likert scales (e.g., a 7-point scale from 'strongly agree' to 'strongly disagree'; a 5-point scale from 'never' to 'always'; a 7-point scale from 'not at all' to 'very much'; a 5-point scale from 'not important' to 'extremely important').
- **Assumption #2:** There needs to be a linear relationship between all variables. The reason for this assumption is that a PCA is based on Pearson correlation coefficients, and as such, there needs to be a linear relationship between the variables. In practice, this assumption is somewhat relaxed (even if it shouldn't be) with the use of ordinal data for variables. Although linearity can be tested using a matrix scatterplot, this is often considered overkill because the scatterplot can sometimes have over 500 linear relationships. As such, it is suggested that you randomly select just a few possible relationships between variables and test these. You can check for linearity in SPSS Statistics using scatterplots, and where there are non-linear relationships, try and "transform" these. If you choose to upgrade to our enhanced content, we have SPSS Statistics guides that show you how to test for linearity using

SPSS Statistics, as well as how to carry out transformations when this assumption is violated.

- **Assumption #3:** You should have sampling adequacy, which simply means that for PCA to produce a reliable result, large enough sample sizes are required. Many different rules-of-thumb have been proposed. These mainly differ depending on whether an absolute sample size is proposed or if a multiple of the number of variables in your sample are used. Generally speaking, a minimum of 150 cases, or 5 to 10 cases per variable, has been recommended as a minimum sample size. There are a few methods to detect sampling adequacy: (1) the Kaiser-Meyer-Olkin (KMO) Measure of Sampling Adequacy for the overall data set; and (2) the KMO measure for each individual variable. In the SPSS Statistics procedure later in this guide, we show you which options to select in SPSS Statistics to test for sampling adequacy.
- **Assumption #4:** Your data should be suitable for data reduction. Effectively, you need to have adequate correlations between the variables in order for variables to be reduced to a smaller number of components. Statistics recommends determining outliers as component scores greater than 3 standard deviations away from the mean. The method used by SPSS Statistics to detect this is Bartlett's test of sphericity.
- **Assumption #5:** There should be no significant outliers. Outliers are important because these can have a disproportionate influence on your results. SPSS

```
Cummulative variance explained: [ 15.32229037  23.84317174  31.24774233  37.44058974  43.50260379
49.35076038  55.1647186   60.86337385  66.39183376  71.72696002
76.89648518  81.95894948  86.7665914   90.87339217  94.65479771
97.15098786  99.05168415 100.          ]
```


3.3.2 K-Means clustering:

The clusters are spherical and second that the clusters are of similar size. Spherical assumption helps in separating the clusters when the algorithm works on the data and forms clusters. If this assumption is violated, the clusters formed may not be what one expects. On the other hand, assumption over the size of clusters helps in deciding the boundaries of the cluster. This assumption helps in calculating the number of data points each cluster should have. This assumption also gives an advantage. Clusters in K-means are defined by taking the mean of all the data points in the cluster. With this assumption, one can start with the centers of clusters anywhere. Keeping the starting points of the clusters anywhere will still make the algorithm converge with the same final clusters as keeping the centers as far apart as possible.

Original count:
attrition= 0 : 1233
attrition =1 : 237
KNN :
attrition=1 : 555
attrition=0 : 915

3.3.3 Agglomerative Clustering:

Affinity: Cosine Linkage: Average
Attrition = 0 : 1129
Attrition = 1 : 341
Affinity: Cosine Linkage: Complete
Attrition = 0 : 1294
Attrition = 1 : 176
Affinity: Cosine Linkage: Single
Attrition = 0 : 1469
Attrition = 1 : 1

Affinity: Euclidean Linkage: Ward
Attrition = 0 : 1121
Attrition = 1 : 349
Affinity: Euclidean Linkage: Average
Attrition = 0 : 1402
Attrition = 1 : 68 
Affinity: Euclidean Linkage: Complete
Attrition = 0 : 982
Attrition = 1 : 488
Affinity: Euclidean Linkage: Single
Attrition = 0 : 1466
Attrition = 1 : 4

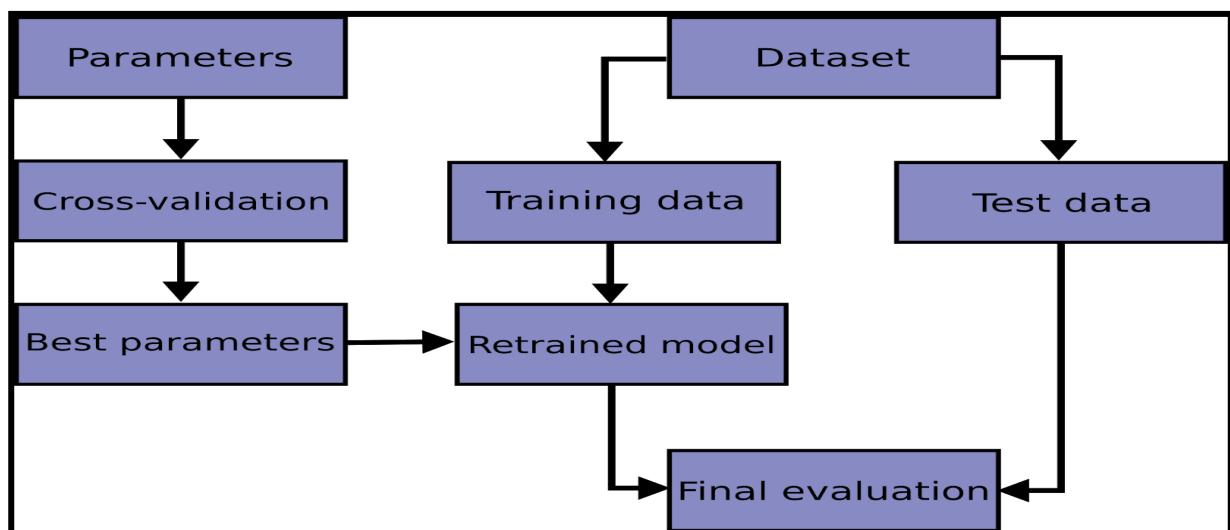
4. Model Evaluation

From the results, we could see that the Imbalance data treatment also didn't help because the accuracies went too low.

We then went to unsupervised learning techniques like K-means clustering and agglomerative clustering. Here also, this couldn't help because, the models performed well even before this. Hence we are going for cross validation and Boosting techniques to see whether the robustness of the model and increase its performance.

4.1 Cross Validation

Learning the parameters of a prediction function and testing it on the same data is a methodological mistake: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data. This situation is called overfitting. To avoid it, it is common practice when performing a (supervised) machine learning experiment to hold out part of the available data as a test set X_{test} , y_{test} . Here is a flowchart of typical cross validation workflow in model training. The best parameters can be determined by grid search techniques.



Cross validation for different models:

4.1.1 SVM-Linear Search:

```
X=ibm_feature1
y=ibm.Attrition.values
from sklearn.model_selection import cross_val_score
from sklearn import svm
svmlnr = svm.SVC(C=1, kernel='linear',gamma=0.2)
svmlnr_scores = cross_val_score(svmlnr,X,Y, cv=10)
svmlnr_scores
```

```
array([0.87837838, 0.86486486, 0.87162162, 0.8707483 , 0.85714286,
       0.85714286, 0.87755102, 0.86986301, 0.89041096, 0.89041096])
```

4.1.2 Logistic Regression:

```
logreg = LogisticRegression()
logreg_scores = cross_val_score(logreg,X,Y, cv=10)
logreg_scores
```

```
array([0.89189189, 0.86486486, 0.87837838, 0.85714286, 0.86394558,
       0.85714286, 0.89115646, 0.87671233, 0.8630137 , 0.90410959])
```

4.1.3 Naive-Bayes:

```
clf = GaussianNB()
NB_scores = cross_val_score(clf,X,Y, cv=10)
NB_scores
```

```
array([0.89189189, 0.88513514, 0.83783784, 0.86394558, 0.88435374,
       0.82312925, 0.84353741, 0.8630137 , 0.83561644, 0.87671233])
```

4.1.4 K-Nearest Neighbor:

```
KNNH = KNeighborsClassifier(n_neighbors= 9 , weights = 'uniform', metric='euclidean')
KNN_scores = cross_val_score(KNNH,X,Y, cv=10)
KNN_scores

array([0.85135135, 0.83783784, 0.83783784, 0.85034014, 0.83673469,
       0.83673469, 0.85034014, 0.84931507, 0.85616438, 0.84931507])
```

4.1.5 SVM-BF:

```
svmrbf = svm.SVC(C=1, kernel='rbf',gamma=0.2)
svmrbf_scores = cross_val_score(svrmbf, X,Y, cv=10)
svmrbf_scores

array([0.83783784, 0.83783784, 0.83783784, 0.83673469, 0.83673469,
       0.83673469, 0.83673469, 0.84246575, 0.84246575, 0.84246575])
```

4.1.6 Decision Tree:

```
dt_model = DecisionTreeClassifier(criterion = 'entropy' )
dt_scores = cross_val_score(dt_model,X,Y, cv=10)
dt_scores

array([0.81081081, 0.85810811, 0.7972973 , 0.82993197, 0.79591837,
       0.78911565, 0.80952381, 0.79452055, 0.79452055, 0.79452055])
```

4.1.7 Bagging Classifier:

```
bgcl = BaggingClassifier(base_estimator=dt_model, n_estimators=19, max_samples=.7)
bgcl_scores = cross_val_score(bgcl,X,Y, cv=10)
bgcl_scores

array([0.84459459, 0.87162162, 0.89864865, 0.87755102, 0.89115646,
       0.82312925, 0.85034014, 0.86986301, 0.8630137 , 0.84931507])
```

4.1.8 ADA Boost Classifier:

```
abcl = AdaBoostClassifier( n_estimators= 50)
abcl_scores = cross_val_score(abcl,X,Y, cv=10)
abcl_scores

array([0.88513514, 0.86486486, 0.87162162, 0.8707483 , 0.89115646,
       0.85034014, 0.87755102, 0.89726027, 0.86986301, 0.89726027])
```

4.1.9 Gradient Boost Classifier:

```
gbcl = GradientBoostingClassifier(n_estimators = 50, learning_rate = 0.05)
gbcl_scores = cross_val_score(gbcl,X,Y, cv=10)
gbcl_scores

array([0.86486486, 0.85135135, 0.87162162, 0.85714286, 0.86394558,
       0.83673469, 0.85034014, 0.84246575, 0.85616438, 0.85616438])
```

4.1.10 Random Forest:

```
rfcl = RandomForestClassifier(n_estimators = 50)
rfcl_scores = cross_val_score(rfcl,X,Y, cv=10)
rfcl_scores

array([0.86486486, 0.85135135, 0.85810811, 0.82993197, 0.86394558,
       0.86394558, 0.85034014, 0.86986301, 0.85616438, 0.85616438])
```

5. Comparison to benchmark:

Different Models vs. accuracy:

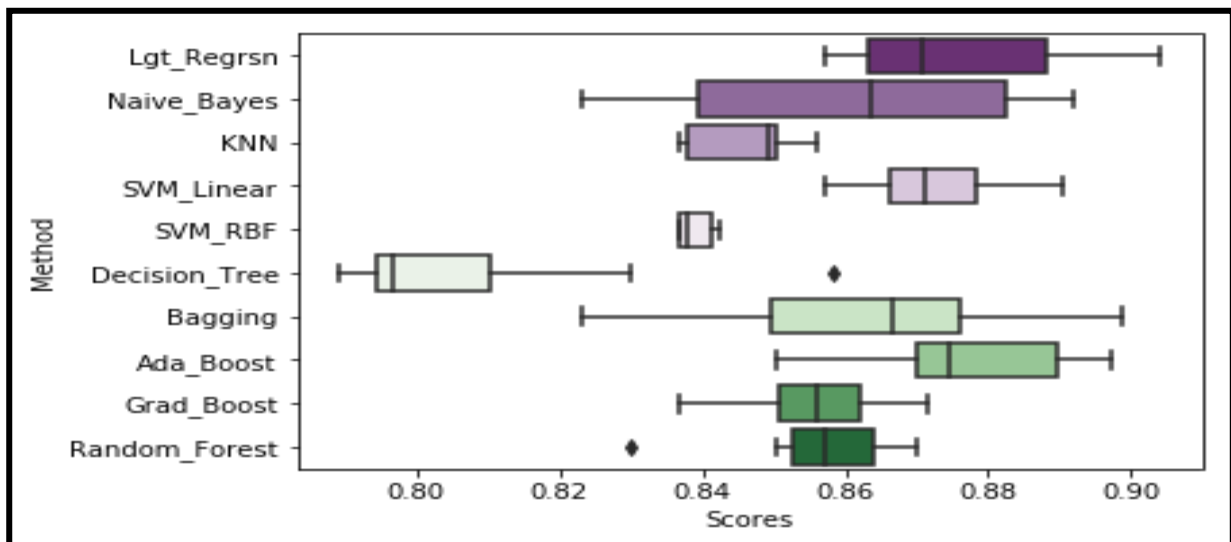
Model	Accuracy on Train set (in %)	Accuracy on Test set (in %)
Before Feature selection		
Logistic Regression	87	87.7
Decision Tree - Gini	85.96	84.69
Decision Tree - Entropy	85.96	84.69
SVC	100	83
Random Forest	100	86
Feature Selection		
Using RFE		
Logistic Regression	87.07	88.18
Decision Tree - Gini	85.71	86.05
Decision Tree - Entropy	85.2	85.37
SVC	94	83
Random Forest	100	86
Using Pearson's Correlation		
Logistic Regression	85.03	86.05
Decision Tree - Gini	85.54	84.69
Decision Tree - Entropy	85.54	84.69
SVC	100	83
Random Forest	100	83
Using K-best method		
Logistic Regression	84.01	83.33
Decision Tree - Gini	83.33	84.01

Decision Tree - Entropy	83.33	84.01
SVC	84	83
Random Forest	83	82
Lasso Regression		
Logistic Regression	84.45	82.5
Decision Tree - Gini	85.51	82.08
Decision Tree - Entropy	85.51	81.63
SVC	100	83
Random Forest	83	82
After Imbalance Data Treatment (Using RFE method)		
Over Sampling		
Logistic Regression	77.1	55.3
Decision Tree - Gini	77.98	42.55
Decision Tree - Entropy	77.98	42.55
SVC	100	84
Random Forest	100	84
Under Sampling		
Logistic Regression	75.8	53.4
Decision Tree - Gini	70.65	43.75
Decision Tree - Entropy	69.56	42.85
SVC	99	58
Random Forest	100	84
SMOTE		
Logistic Regression	84.23	63.29
Decision Tree - Gini	72.01	38.23

Decision Tree - Entropy	72.01	38.23
SVC	100	81
Random Forest	100	85

Before using cross validation, we saw that Logistic regression was giving better result. After cross fold validation, we saw that SVM is giving better results as the results range from approximately 85 to 88.5 % accuracy with 95% confidence interval. With ensemble techniques, we can see that Gradient boosting technique is performing better.

6. Visualization

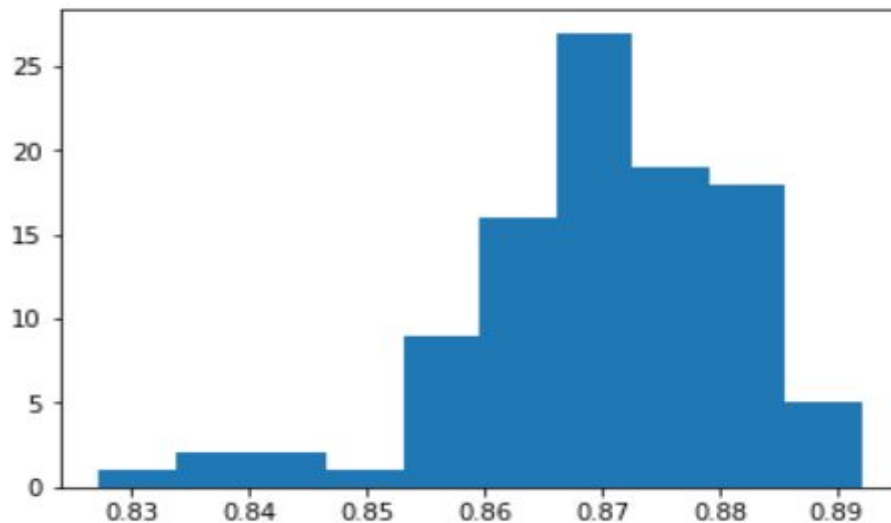


From the above image, we can see that in SVM and gradient boosting, distribution is almost normal and the accuracies are evenly distributed compared to others. Random forest and Decision tree is completely neglected as there are outliers in the accuracy. Based on simplicity, we took our final model as SVM-Linear and carried forward with Bootstrapping to get 95% confidence interval.

7. Implications

Our solution tells whether the employee may/maynot leave the company based on several factors like 'Age', 'BusinessTravel', 'Department', 'DistanceFromHome', 'EnvironmentSatisfaction', 'Gender', 'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus', 'NumCompaniesWorked', 'OverTime', 'RelationshipSatisfaction', 'WorkLifeBalance', 'YearsInCurrentRole', 'YearsSinceLastPromotion'.

After bootstrapping with a 95 % confidence, our model predicted the attrition rate of employees with accuracy for 84.1% to 88.7%



95.0 confidence interval 84.1% and 88.7%

8. Limitations

Since our data set had very few records, we were confined to create a model within that, In future, the dataset can include more detailed data so that we can dive into deep learning and based on the attrition rate, we can give benefits to the employees so that the employee attrition rate gets decreased.

9. Conclusion

With the given dataset, we saw the distribution of all the features affecting attrition using Data visualization techniques and came up with some basic models like Logistic regression, Decision tree, Random forest etc. , We treated the Imbalance data and build models upon them which were not effective. We went to dimensionality reduction using PCA . Since there were less number of attributes and records, PCA was not successful. We went to Unsupervised learning techniques in which K-Means algorithm and agglomerative clustering techniques were used. As the clusters accuracy was also not as desired, we went with feature selection techniques among which we selected RFE and built several models. To increase the accuracy scores and to make the models more optimized in real time, we went for k-fold cross validation. As the final result, we can conclude that without any ensemble techniques, we can use SVM model as it is simpler and simpler model are more efficient.