



atatrained

LEARN ANYTHING, ANYTIME, ANYWHERE

PROJECT TITLE

**HR ANALYTICS PROJECT- UNDERSTANDING THE
ATTRITION IN HR**

SUBMITTED BY:

MR. KRISHNA TEJA DINAVAHU

ACKNOWLEDGMENT

I would like to express my special gratitude to “Data trained” team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills.

References use in this project:

1. SCIKIT Learn Library Documentation
2. Blogs from towardsdatascience, Analytics Vidya, Medium
3. Andrew Ng Notes on Machine Learning (GitHub)
4. Data Science Projects with Python Second Edition by Packt
5. Hands on Machine learning with scikit learn and tensor flow by Aurelien Geron
6. Sifei Lu, Zengxiang Li, Zheng Qin, Xulei Yang, Rick Siow Mong Goh, "A Hybrid Regression Technique for House Prices Prediction", @2017 IEEE, 2017 IEEE International Conference on Industrial Engineering & Engineering Management , Singapore
DOI:10.1109/IEEM.2017.8289904
7. CH.Raga Madhuri, Anuradha G, M.Vani Pujitha "House Price Prediction Using Regression Techniques: A Comparative Study", IEEE 6th International Conference on smart structures and systems ICSSS 2019
8. Zhongyun, Jiang, Guoxin, Shen, "Prediction of House Price Based on The Back Propagation Neural Network in The Keras Deep Learning Framework", 2019 6th International Conference on Systems and Informatics (ICSA

I2019).

9. Zhen Peng, Qiang Huang, Yincheng Han, “Model Research on Forecast of Second-Hand House Price in Chengdu Based on XGboost Algorithm”, 2019 IEEE 11th International Conference on Advanced Infocomm Technology
10. J.-G. Liu, X.-L. Zhang, and W.-P. Wu, “Application of fuzzy neural network for real estate prediction,” *Advances in Neural Networks - ISNN 2006*, vol. 3973, pp. 1187–1191, 2006.
11. H. Kusan, O. Aytekin, and I. Ozdemir, “*e use of fuzzy logic ” in predicting house selling price,” *Expert Systems with Applications*, vol. 37, no. 3, pp. 1808–1813, 2010.
12. Ayush Varma, Abhijit Sarma, Rohini Nair and Sagar Doshi, ”House Price Prediction Using Machine Learning And Neural Networks”, @2018 IEEE, 2018 Second International Conference on Inventive Communication and Computational Technologies(ICICCT), Coimbatore, India, DOI:10.1109/ICICCT.2018.8473231.
13. Arietta, Sean M., et al. "City forensics: Using visual elements to predict non-visual city attributes." *IEEE transactions on visualization and computer graphics* 20.12 (2014): 2624-2633.

INTRODUCTION

Executive Summary

This report is study of HR Data Analysis using Python and Business Intelligence Tool based on a project that I have successfully completed as Data Analyst.

Every year a lot of companies hire a number of employees. The companies invest time and money in training those employees, not just this but there are training programs within the companies for their existing employees as well. The aim of these programs is to increase the effectiveness of their employees. But where HR Analytics fit in this? and is it just about improving the performance of employees?

HR Analytics

Human resource analytics (HR analytics) is an area in the field of analytics that refers to applying analytic processes to the human resource department of an organization in the hope of improving employee performance and therefore getting a better return on investment. HR analytics does not just deal with gathering data on employee efficiency. Instead, **it aims to provide insight into each process by gathering data and then using it to make relevant decisions about how to improve these processes.**

Attrition in HR

Attrition in human resources refers to the gradual loss of employees overtime. In general, relatively high attrition is problematic for companies. HR professionals often assume a leadership role in designing company compensation programs, work culture, and motivation systems that help the organization retain top employees.

How does Attrition affect companies? and how does HR Analytics help in analyzing attrition? We will discuss the first question here and for the second question, we will write the code and try to understand the process step by step.

Attrition affecting Companies

A major problem in high employee attrition is its cost to an organization. Job postings, hiring processes, paperwork, and new hire training are some of the common expenses of losing employees and replacing them. Additionally, regular employee turnover prohibits your organization from increasing its collective knowledge base and experience over time. This is especially concerning if your business is customer-facing, as customers often prefer to interact with familiar people. Errors and issues are more likely if you constantly have new workers.

The aim in this visualization report is to present the history of the organization how many employee working, gender and salary structure and performance. This visualization showing statically information in a single dashboard in any organization, an employee is more vital for the success and viability of a business. The organization which attracts competent employees, utilize human resources efficiently, manage talent effectively and retain employees is securing the

long term success for the business. Human resource (HR) analytics have the potential to bring great value to the decision making the ability of HR leaders on human and organizational capital. Human resource analytics are useful for improving employee performance and getting an optimal return on investment (ROI) on its human capital. This article examines the application of the emerging discipline of HR analytics for business transformation and success. The relevant literature is reviewed about the integration of HR into business performance measurement. It indented and describes the HR's strategic role, functions and application of HR analytics. Thus, organizations should invest aggressively in this new discipline, link it to the rest of the business, and reskill their teams to bring data to work in every major people related decision

KEYWORDS HR analytics, data-driven HRM, HR professionals, ROI, Organizational transformation

INTRODUCTION In any organization, an employee is more vital for the success and viability of a business. The traditional role of human resource was to collect and keep track of the employee's professional and personal information through manpower inventory, payroll, health & safety and performance management. Now, with the advent of new technology, human resource departments are generating more data than ever before. However, they often struggle to turn their data into valuable managerial insights The organization which attracts competent employees, utilize human resources efficiently, manage talent effectively and retain employees is securing the long term success for the business. The data analysis and its proper utilization i.e. giving meaning to that

data play a more active role in the human resource management. HR managers typically used metrics to measure such data as turnover rates and sick days. There is a difference between HR metrics and HR analytics. Though there has been a proliferation of HR metrics and scorecards in recent years, HR needs the power of analytics to differentiate. HR representatives can now talk with business leaders, and out what they need and hire accordingly. HR analytics enable organizations to align HR metrics with strategic business goals. HR analytics empower organizations to use their treasure of employee data for making better decisions about their workforces and improve operational performance.

IBM HR Analytics Employee Attrition & Performance Dataset

In this case study we will use IBM HR Analytics database. This fictional dataset created by IBM employees and available to download from GitHub and Kaggle. You can also download dataset from [Data set](#). This dataset consists of 1470 rows, 35 features describing each employee's background and characteristics and target variable. Attrition is target variable to be predicted. As target variable is categorical in nature, this case study falls into classification machine learning problem. We have two objectives here:

1. Which key factors result in employee attrition?
2. Building ML Model for predicting attrition.

ANALYTICAL PROBLEM FRAMING

Data Preparation: Load, Clean and Format

Let's begin with importing libraries for EDA and dataset itself.

Importing require library for performing EDA, Data Wrangling and data cleaning

```
import pandas as pd # for data wrangling purpose
import numpy as np # Basic computation library
import seaborn as sns # For Visualization
import matplotlib.pyplot as plt # plotting package
%matplotlib inline
import warnings # Filtering warnings
warnings.filterwarnings('ignore')
```

Importing IBM HR dataset Csv file using pandas

```
df=pd.read_csv('WA_Fn-UseC_-HR-Employee-Attrition.csv')
```

```
print('No of Rows:',df.shape[0])
print('No of Columns:',df.shape[1])
pd.set_option('display.max_columns', None) # This will enable us to see truncated columns
df.head()
```

No of Rows: 1470

No of Columns: 35

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	H
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	3	Male	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	Male	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	4	Female	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	1	Male	

Checking different datatypes in dataset: -

```
df.columns
```

```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
      'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
      'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
      'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
      'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
      'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
      'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
      'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
      'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
      'YearsWithCurrManager'],
      dtype='object')
```


We have 9 features with object datatypes and rest are Numeric feature with int64. Out of all numeric features Education, Environment-Satisfaction, Job-Involvement, Job-Satisfaction, Relationship-Satisfaction, Performance Rating, Work Life Balance are ordinal variable. These ordinal features have unique label for each numeric value.

These Ordinal features come with the following label encoding:

- **Education:** 1- '*Below College*', 2 -'*College*', 3 -'*Bachelor*', 4- '*Master*',5 -'*Doctor*'
- **Environment Satisfaction:** 1- '*Low*', 2- '*Medium*', 3 - '*High*', 4- '*Very High*'
- **Job Involvement:** 1 -'*Low*', 2- '*Medium*', 3- '*High*', 4- '*Very High*'
- **Job Satisfaction:** 1- '*Low*', 2- '*Medium*', 3- '*High*', 4 -'*Very High*'
- **Performance Rating:** 1- '*Low*', 2- '*Average*', 3 -'*Good*', 4- '*Excellent*', 5- '*Outstanding*'
- **Relationship Satisfaction:** 1- '*Low*', 2- '*Medium*', 3- '*High*', 4- '*Very High*'
- **Work Life Balance:** 1- '*Bad*', 2- '*Good*', 3- '*Better*', 4- '*Best*'

Above nomenclature will help in better understanding of data when we perform EDA in this case study.

Data Integrity Check: Dataset can have missing values, duplicated entries and whitespaces. Now we will perform this integrity check of dataset.

Data Integrity Check

Since dataset is large, Let check for any entry which is repeated or duplicated in dataset.

```
] : df.duplicated().sum() # This will check the duplicate data for all columns.  
]  
]: 0
```

Missing value check

```
missing_values = df.isnull().sum().sort_values(ascending = False)  
percentage_missing_values =(missing_values/len(df))*100  
print(pd.concat([missing_values, percentage_missing_values], axis =1, keys =['Missing Values', '% Missing data']))
```

	Missing Values	% Missing data
Age	0	0.0
StandardHours	0	0.0
NumCompaniesWorked	0	0.0
Over18	0	0.0
OverTime	0	0.0
PercentSalaryHike	0	0.0
PerformanceRating	0	0.0
RelationshipSatisfaction	0	0.0
StockOptionLevel	0	0.0
MonthlyIncome	0	0.0
TotalWorkingYears	0	0.0
TrainingTimesLastYear	0	0.0
WorkLifeBalance	0	0.0
YearsAtCompany	0	0.0
YearsInCurrentRole	0	0.0
YearsSinceLastPromotion	0	0.0
MonthlyRate	0	0.0
MaritalStatus	0	0.0
Attrition	0	0.0
EmployeeCount	0	0.0
BusinessTravel	0	0.0
DailyRate	0	0.0
Department	0	0.0
DistanceFromHome	0	0.0
Education	0	0.0
EducationField	0	0.0
EmployeeNumber	0	0.0
JobSatisfaction	0	0.0
EnvironmentSatisfaction	0	0.0
Gender	0	0.0
HourlyRate	0	0.0
JobInvolvement	0	0.0
JobLevel	0	0.0
JobRole	0	0.0
YearsWithCurrManager	0	0.0

Comment: There is no missing data.

Luckily for us, there is no missing data! this will make it easier to work with the dataset.

Dataset doesn't contain Any duplicate entry, whitespace, 'NA', or '-'.

So, Yes Good to Go Further!!!

Statistical parameters like mean, median, quantile can give important details about database. Now is time to look at statistical Matrix of Dataset.

Few key observations from this statistical matrix are listed below: -

- Minimum Employee Age is 18 and Maximum age of employee 60.
- Average distance from home is 9.1 KM. It means that most of employee travel at least 18 KM in day from home to office.
- Average performance Rating of employees is 3.163 with min value 3.0. This Means that performance of most of employee is 'Good'. This implies that Attrition of Employee with 'Outstanding' or 5 rating need to investigate.
- 50% of Employees has worked at least 2 companies previously.
- For Monthly Income, Monthly Rate by looking at 50% and max column we can say outliers exist in this feature.
- By looking at Mean and Median we see that some of the features are skew in nature.
- For ordinal features statistical terminology like mean, median, std deviation are not applicable.
- Standard Hours and Employee Count contain same value for all statistical parameter. It means they contain one unique value.

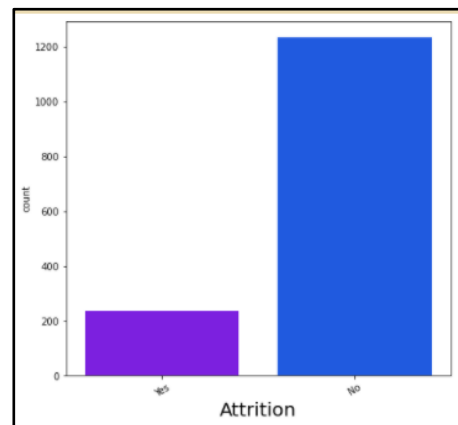
EXPLORATORY DATA ANALYSIS

EXPLORATORY DATA ANALYSIS REFERS TO THE CRITICAL PROCESS OF PERFORMING INITIAL INVESTIGATIONS ON DATA SO AS TO DISCOVER PATTERNS, TO SPOT ANOMALIES, TO TEST HYPOTHESIS AND TO CHECK ASSUMPTIONS WITH THE HELP OF SUMMARY STATISTICS AND GRAPHICAL REPRESENTATIONS.

Let's begin data exploration of Target variable using count plot.

```
df['Attrition'].value_counts()

No      1233
Yes      237
Name: Attrition, dtype: int64
```



83.88% (1237 employees) Employees did not leave the organization while 16.12% (237 employees) did leave the organization ***making our dataset to be consider as imbalanced*** since more people stay in the organization than they actually leave.

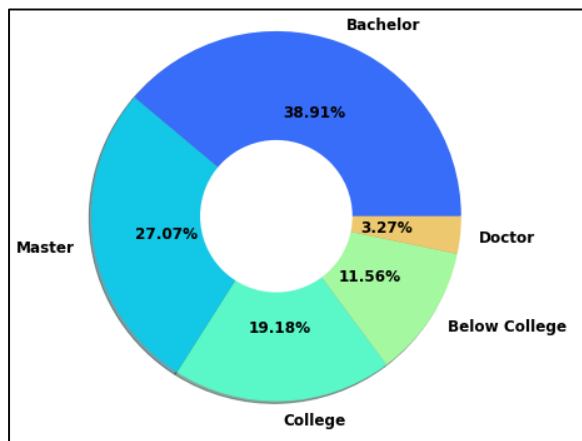
In this dataset we have features like education, department, education field, job role, job satisfaction which are inter related with each other. Job role & job

Key Insights from Pie Plot

1. More than 38 % employees educated at Bachelor level.
2. 30 % of Employees are highly educated which involves master and doctor degree.
3. Almost 19% Employees are educated up to college & 12% are below college.

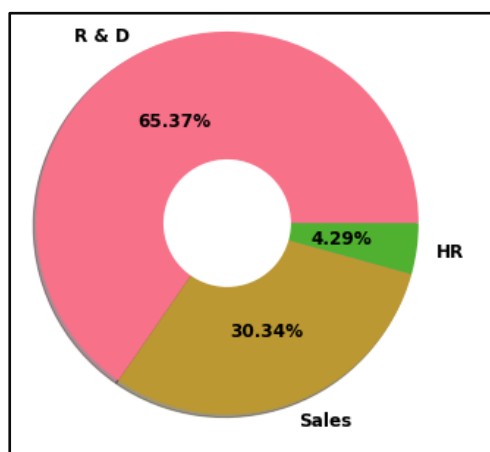
position not in alignment with educational background can lead attrition. Let investigate this by visualisation of these features one by one to gain more insights.

Education level of Man power available:



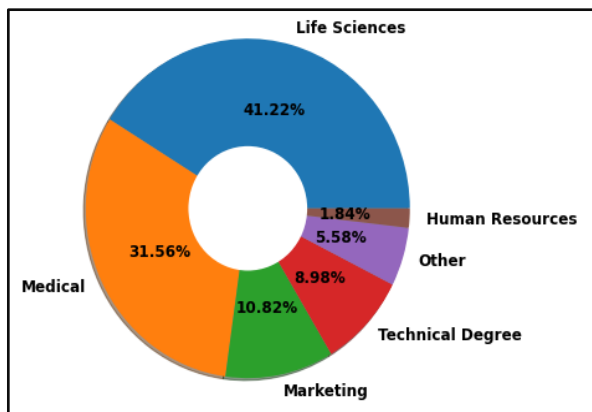
Department wise Distribution of Man power:

Department	Human Resources	Research & Development	Sales	All
Education				
1	5	115	50	170
2	13	182	87	282
3	27	379	166	572
4	15	255	128	398
5	3	30	15	48
All	63	961	446	1470



Key Insights on Department and Education Level of employee in each Department

1. 65.37% of Employees work inside Research & Development Department. Out of Total 961 Employee number of employees with education level of Bachelors, Masters, Doctor are 379, 255 and 30 respectively.
2. Only 63 Employee work in HR department.



Key Insights from Pie Plot

1. Employees belongs to six different domains.
2. 41.22 % Employee comes from Life science background followed by medical profession with 31.56%
3. Least number of Employees comes from HR background.

Employee distribution as per education field:

```
# Let check distribution of department Vs education Field
pd.crosstab([df.Department],[df.EducationField], margins=True)
```

	EducationField	Human Resources	Life Sciences	Marketing	Medical	Other	Technical Degree	All
Department								
Human Resources		27	16	0	13	3	4	63
Research & Development		0	440	0	363	64	94	961
Sales		0	150	159	88	15	34	446
All		27	606	159	464	82	132	1470

The probability of Employees Retention is more when there working domain is in alignment with education background. Let check this with crosstab of department against education field.

We will Analyse Attrition in department according to education background based on above insight further but before that explore Job role in order to include it in further

Key Insights from above Cross Tab:

- There are only 27 people with HR background and We know that 63 people work in HR Department from previous result. This implies that at least half employee working in HR department do not have HR background.
- R&D department almost everyone comes from domain expertise or technical background except support staff. These employees usually have high salary, so it will be interesting to investigate attrition in this category.
- There are 159 Employee with Marketing background and all work in Sales Department.
- 50% Employees in sales department have background of Life sciences & Medical. We can clear see they are working in domain to which their educational background does not belong. So, it will be interesting to see attrition rate in these employees.

attrition analyse. First build matrix of department vs job role which will give us idea about number of employees of different job role across department.

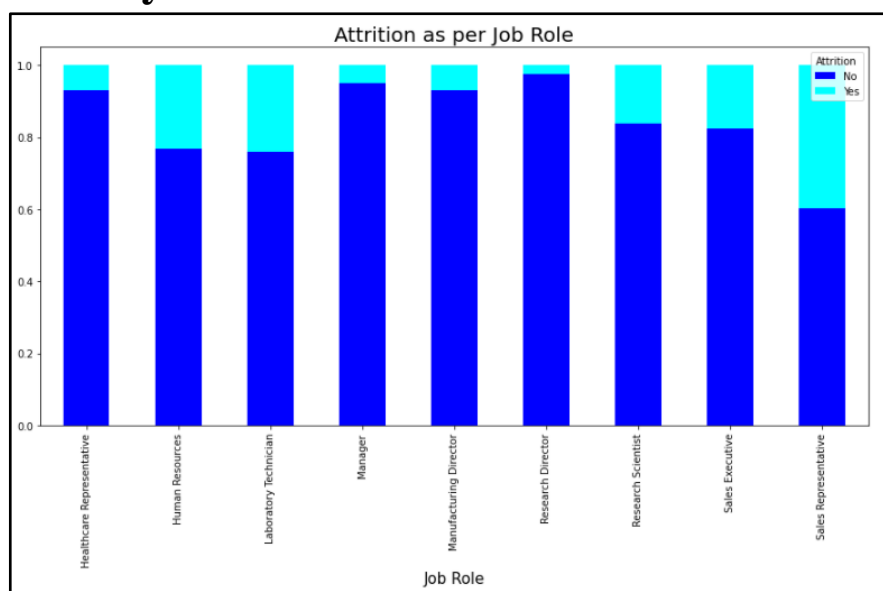
```
pd.crosstab([df.JobRole],[df.Department], margins=True)
```

Department	Human Resources	Research & Development	Sales	All
JobRole				
Healthcare Representative	0	131	0	131
Human Resources	52	0	0	52
Laboratory Technician	0	259	0	259
Manager	11	54	37	102
Manufacturing Director	0	145	0	145
Research Director	0	80	0	80
Research Scientist	0	292	0	292
Sales Executive	0	0	326	326
Sales Representative	0	0	83	83
All	63	961	446	1470

Key Insights from above Cross Tab:

- There are 3 job roles in HR Department, maximum of which are sales Executive with 446 Total Employees.
- Human Resources department has 2 Job role i.e., HR & Manager.
- There 6 different Job role in R&D department with total 961 employees and until now we know that all of them belong to their respective domain background.

Attrition by Job role :



Let's check absolute number matrix of attrition according to job role, again this time using crosstab.

```
pd.crosstab([df.JobRole,df.Department],[df.Attrition], margins=True)
```

		Attrition		
		No	Yes	All
JobRole	Department			
Healthcare Representative	Research & Development	122	9	131
Human Resources	Human Resources	40	12	52
Laboratory Technician	Research & Development	197	62	259
	Human Resources	11	0	11
Manager	Research & Development	51	3	54
	Sales	35	2	37
Manufacturing Director	Research & Development	135	10	145
Research Director	Research & Development	78	2	80
Research Scientist	Research & Development	245	47	292
Sales Executive	Sales	269	57	326
Sales Representative	Sales	50	33	83
All		1233	237	1470

Key Insights from above Cross Tab:

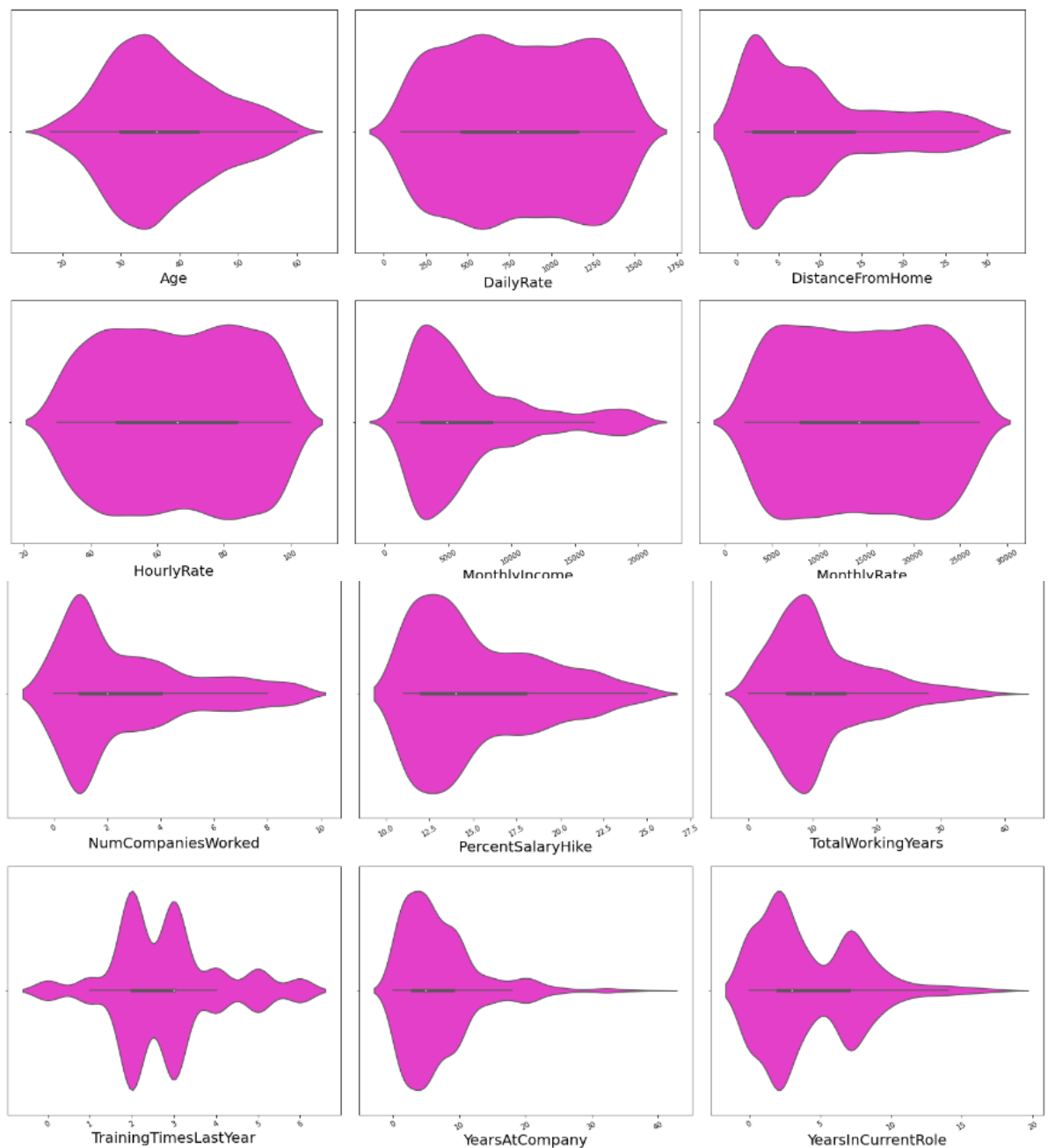
- Percentage of attrition is high in Sales Representative, Laboratory Technician, Human Resources.
- At the Top chart 62 Laboratory Technician has resign from job, followed by 57 sales executive and 47 Research Scientist.
- 16 % attrition rate for Research Scientist, which involve huge investment from company. Company not only loses employee but its knowledge base, expertise & Intellectual property rights in some cases.

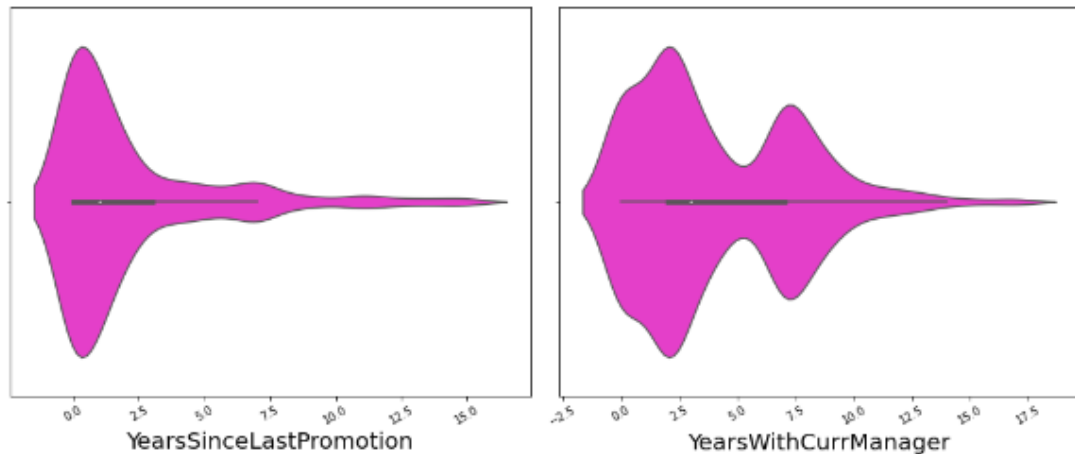
Let's check violin plot of some numerical features to gain more insight.

Key Insights from above Cross Tab:

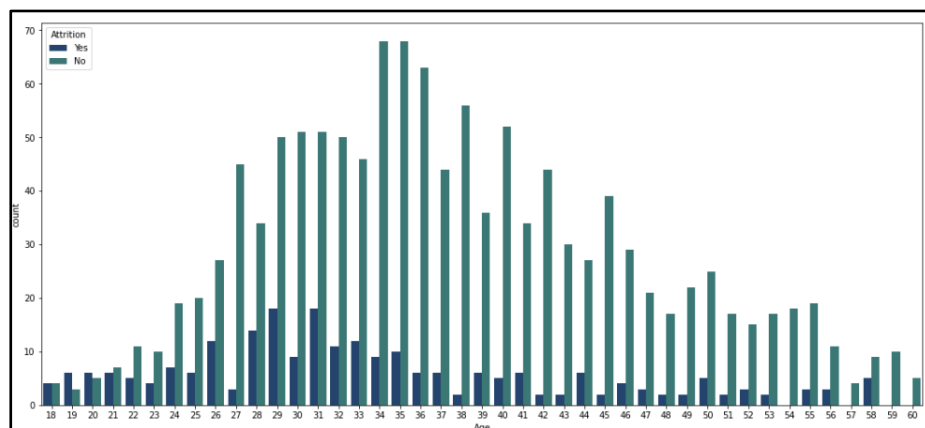
- For Majority of people have spent 3 to 10 years at company.

- Most of people staying company up to 2 years after promotion.
- Majority of people are train 2-3 times in last year
- Majority of people stay in same role for maximum 4 yrs.
- Majority of Employees have salary hike of 10 to 15%.





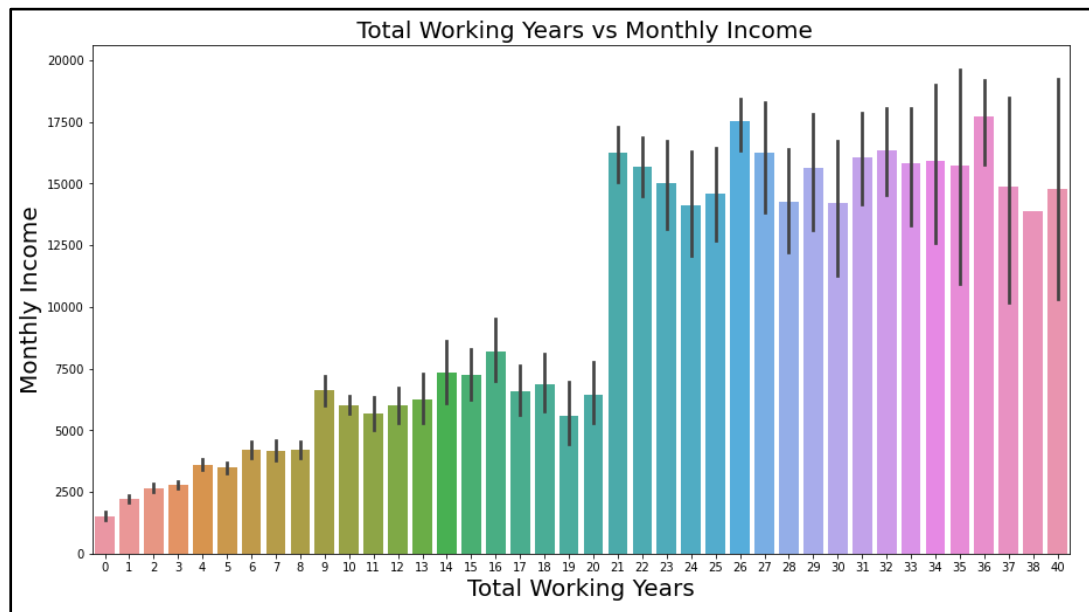
Q. In which age group attrition rate is high?



Key Insights from count plot of Age Vs Attrition:

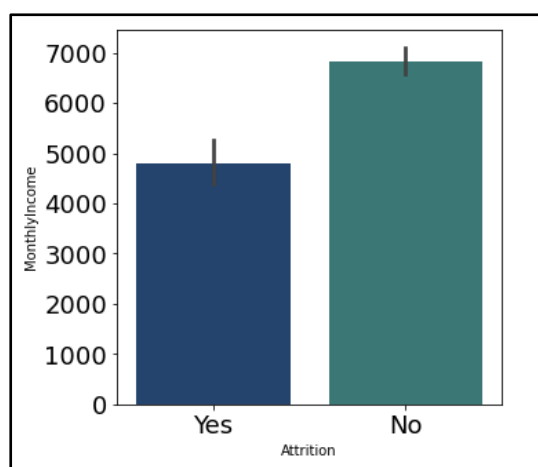
1. The Attrition rate is minimum between the Age years of 34 and 45.
2. The Attrition rate is maximum between the Age years of 29 and 33.

Q. What is variation in monthly income as Total working year



increases.

Monthly Income is higher for the employees with 21 or more number of Total working years. For first 9 years monthly income is less than 5000\$. **But what about attrition, let's bar chart of Monthly income so we can come across some benchmark of average monthly income in both attrition categories.**



Key Insights on Average Monthly Income as per attrition

- We can see that Average monthly income is less in employees who choose to resign compare to rest. Less Monthly Income is major reason behind attrition.
- To prevent attrition average monthly to be greater than 6900\$ is recommended.

FEATURE ENGINEERING: DATA PRE-PROCESSING

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data.

Feature Engineering is very important step in building Machine Learning model. Some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used. In Feature engineering can be done for various reason. **Some of them are mention below:**

1. **Feature Importance:** An estimate of the usefulness of a feature
2. **Feature Extraction:** The automatic construction of new features from raw data (Dimensionality reduction Technique like PCA)
3. **Feature Selection:** From many features to a few that are useful
4. **Feature Construction:** The manual construction of new features from raw data (For example, construction of new column for month out date - mm/dd/yy)

There are Varity of techniques use to achieve above mention means as per need of dataset. Some of Techniques important are as below:

- Handling missing values
- Handling imbalanced data using SMOTE
- Outliers' detection and removal using Z-score, IQR
- Scaling of data using Standard Scalar or Minmax Scalar
- Binning whenever needed
- Encoding categorical data using one hot encoding, label / ordinal encoding
- Skewness correction using Boxcox or yeo-Johnson method
- Handling Multicollinearity among feature using variance inflation factor
- Feature selection Techniques:
 - ✓ Correlation Matrix with Heatmap
 - ✓ Univariate Selection – SelectKBest
 - ✓ ExtraTreesClassifier method

In this case study we will use some of the mention feature engineering Techniques one by one.

1. Dropping unnecessary features

Feature like 'Over18', 'StandardHours' contain single unique value. Features like EmployeeCount, EmployeeNumber are irrelevant from ML model building perspective. We will drop these features.

```
# Dropping unnecessary columns  
df.drop(["EmployeeCount", "EmployeeNumber", "Over18", "StandardHours"], axis=1, inplace=True)
```

2. Encoding Categorical & Ordinal Features

Label Encoding is employed over target variable 'Attrition' while Ordinal encoding employ for rest categorical features.

```
# Using Label Encoder on target variable
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df["Attrition"] = le.fit_transform(df["Attrition"])
df.head()
```

```
# Ordinal Encoding for ordinal variables
from sklearn.preprocessing import OrdinalEncoder
oe = OrdinalEncoder()
def ordinal_encode(df, column):
    df[column] = oe.fit_transform(df[column])
    return df

oe_col = ['BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'OverTime']
df=ordinal_encode(df, oe_col)
df.head()
```

Since now encoding is done we will move towards outliers' detection and removal.

3. Outliers' detection and removal

Machine learning algorithms are sensitive to the range and distribution of attribute values. Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models and ultimately poorer results. Outliers can be seen in boxplot of numerical feature. We did not added boxplot here as it will make this article length, I left it to reader to further investigate. Now we will use Z-score method for outliers' detection.

```
from scipy.stats import zscore
z = np.abs(zscore(df))
threshold = 3
df1 = df[(z<3).all(axis = 1)]

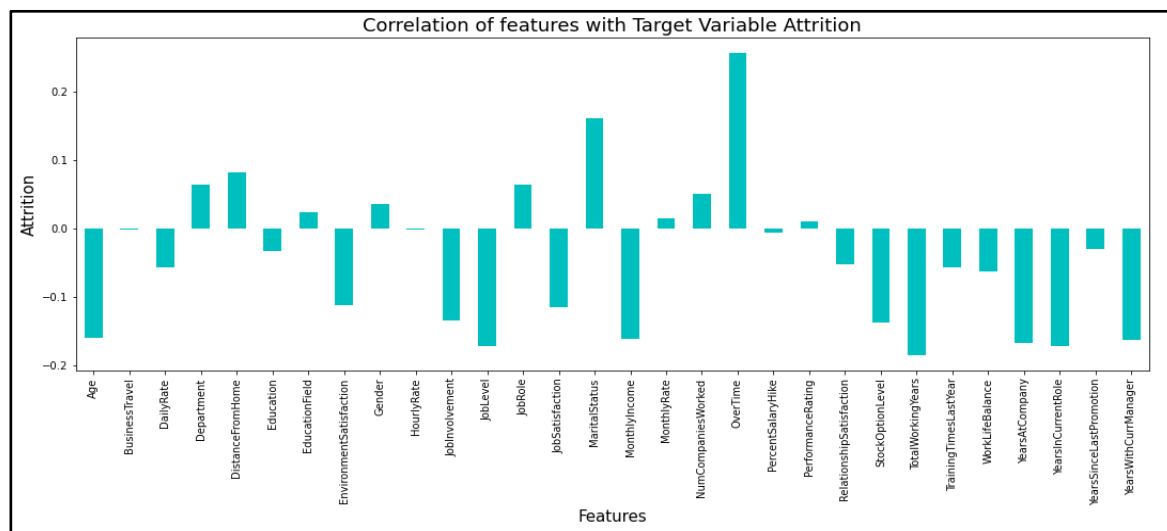
print ("Shape of the dataframe before removing outliers: ", df.shape)
print ("Shape of the dataframe after removing outliers: ", df1.shape)
print ("Percentage of data loss post outlier removal: ", (df.shape[0]-df1.shape[0])/df.shape[0]*100)

df=df1.copy() # reassigning the changed dataframe name to our original dataframe name

Shape of the dataframe before removing outliers: (1470, 31)
Shape of the dataframe after removing outliers: (1387, 31)
Percentage of data loss post outlier removal: 5.646258503401361
```

4. Correlation Heatmap

Correlation Heatmap show in a glance which variables are correlated, to what degree, in which direction, and alerts us to potential multicollinearity problems. The bar plot of correlation coefficient of target variable with independent features shown below



Multicollinearity between features

Variance Inflation factor imported from statsmodels.stats.outliers_influence to check multicollinearity between features.

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif= pd.DataFrame()
vif['VIF']= [variance_inflation_factor(df.values,i) for i in range(df.shape[1])]
vif['Features']= df.columns
vif
```

	VIF	Features
0	1.930457	Age
1	1.014314	BusinessTravel
2	1.025841	DailyRate
3	2.172093	Department
4	1.017385	DistanceFromHome
5	1.065266	Education
6	1.030480	EducationField
7	1.024396	EnvironmentSatisfaction
8	1.024366	Gender
9	1.024189	HourlyRate
10	1.020167	JobInvolvement
11	5.976707	JobLevel
12	2.023213	JobRole
13	1.023909	JobSatisfaction

14	2.298943	MaritalStatus
15	5.842828	MonthlyIncome
16	1.022108	MonthlyRate
17	1.426763	NumCompaniesWorked
18	1.028400	OverTime
19	1.016867	PercentSalaryHike
20	1.022260	RelationshipSatisfaction
21	2.279101	StockOptionLevel
22	4.093506	TotalWorkingYears
23	1.025519	TrainingTimesLastYear
24	1.017093	WorkLifeBalance
25	6.296064	YearsAtCompany
26	3.513852	YearsInCurrentRole
27	1.373189	YearsSinceLastPromotion
28	3.433437	YearsWithCurrManager

We can see that for all features Variance inflation factor is within permissible limit of 10. Multicollinearity does not pose any threat here.

5. Handling imbalanced data using SMOTE

This two-class dataset is imbalanced (84% vs 16%). As a result, there is a possibility that the model built might be biased towards the majority and over-represented class. We can resolve this by Synthetic Minority Oversampling Technique (SMOTE) to over-sample the minority class.

```
from imblearn.over_sampling import SMOTE

# Oversampling using SMOTE Techniques
oversample = SMOTE()
X, Y = oversample.fit_resample(X, Y)
```

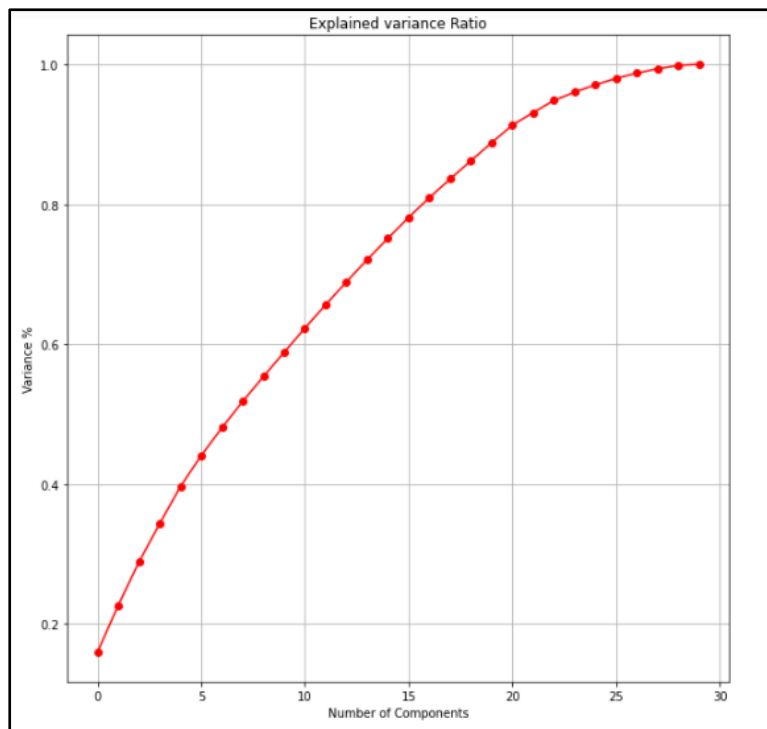
6. Scaling of data using Standard Scaler

```
# Splitting data in target and dependent feature
X = df.drop(['Attrition'], axis = 1)
Y = df['Attrition']

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scale = scaler.fit_transform(X)
```

7. Dimensionality Reduction Using PCA

PCA is used to find patterns and extract the latent features from our dataset.



We can see that 21 principal components attribute for 90% of variation in the data. PCA applied for 21 components.

```
pca_new = PCA(n_components=21)
x_new = pca_new.fit_transform(X_scale)

principle_x=pd.DataFrame(x_new,columns=np.arange(21))
```

MACHINE LEARNING MODEL BUILDING

In this section we will build Supervised learning ML model-based classification algorithm. As objective is to predict attrition in 'Yes' or 'No' leads to fall problem in domain of classification algorithm. train_test_split used to split data with size of 0.33

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(principle_x, Y, random_state=42, test_size=.33)
print('Training feature matrix size:',X_train.shape)
print('Training target vector size:',Y_train.shape)
print('Test feature matrix size:',X_test.shape)
print('Test target vector size:',Y_test.shape)
```

```
Training feature matrix size: (1551, 21)
Training target vector size: (1551,)
Test feature matrix size: (765, 21)
Test target vector size: (765,)
```

First we will build base model using logistic regression algorithm. Best random state is investigated using for loop for random state in range of (0,250).

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, f1_score
maxAccu=0
maxRS=0
for i in range(1,250):
    X_train,X_test,Y_train,Y_test = train_test_split(principle_x,Y,test_size = 0.33, random_state=i)
    log_reg=LogisticRegression()
    log_reg.fit(X_train,Y_train)
    y_pred=log_reg.predict(X_test)
    acc=accuracy_score(Y_test,y_pred)
    if acc>maxAccu:
        maxAccu=acc
        maxRS=i
print('Best accuracy is', maxAccu , 'on Random_state', maxRS)
```

```
Best accuracy is 0.8705882352941177 on Random_state 242
```

Logistics regression model is train with random state 242. The evaluation matrix along with classification report is as below :

```

Logistics Regression Evaluation

Accuracy Score of Logistics Regression : 0.8705882352941177

Confusion matrix of Logistics Regression :
[[324  44]
 [ 55 342]]

classification Report of Logistics Regression
precision    recall  f1-score   support

     0       0.85     0.88     0.87     368
     1       0.89     0.86     0.87     397

 accuracy          0.87          0.87          0.87          765
  macro avg       0.87          0.87          0.87          765
 weighted avg     0.87          0.87          0.87          765

```

As Now base model is ready with f1-score of 0.87, we will train model with different classification algorithm along with k-5 fold cross validation. The final evaluation matrix different classification algorithm is as shown table below:

ML Algorithm	Accuracy Score	CV Mean Score	f-1 Score	Recall	Precision
Logistic Regression	0.8842	0.6869	0.88	0.88	0.88
SVC	0.8864	0.6075	0.89	0.89	0.89
GaussianNB	0.8908	0.7405	0.85	0.85	0.85
DecisionTreeClassifier	0.7882	0.8381	0.89	0.86	0.87
KNeighborsClassifier	0.8537	0.7374	0.85	1.00	0.92
RandomForestClassifier	0.8689	0.9177	0.88	0.99	0.93
AdaBoostClassifier	0.8515	0.8718	0.90	0.93	0.91
GradientBoostingClassifier	0.8799	0.8918	0.89	0.98	0.93
Bagging Classifier	0.8558	0.8856	0.88	0.96	0.92

We can see that Random Forest Classifier gives us maximum f1-score & mean cross validation score. We will perform hyper parameter tuning on random forest classifier to build final ML Model.

```

from sklearn.model_selection import GridSearchCV

parameter = { 'bootstrap': [True], 'max_depth': [5, 10,20,40,50, None],
              'max_features': ['auto', 'log2'],
              'criterion':['gini','entropy'],
              'n_estimators': [5, 10, 15 ,25,50,100]}

GCV = GridSearchCV(RandomForestClassifier(),parameter,cv=5,n_jobs = -1,verbose=3)
GCV.fit(X_train,Y_train)

Fitting 5 folds for each of 144 candidates, totalling 720 fits
GridSearchCV(cv=5, estimator=RandomForestClassifier(), n_jobs=-1,
             param_grid={'bootstrap': [True], 'criterion': ['gini', 'entropy'],
                          'max_depth': [5, 10, 20, 40, 50, None],
                          'max_features': ['auto', 'log2'],
                          'n_estimators': [5, 10, 15, 25, 50, 100]},
             verbose=3)

GCV.best_params_

{'bootstrap': True,
 'criterion': 'entropy',
 'max_depth': 20,
 'max_features': 'log2',
 'n_estimators': 15}

```

Next step is to build final machine learning model over best params in Hyper parameter tuning.

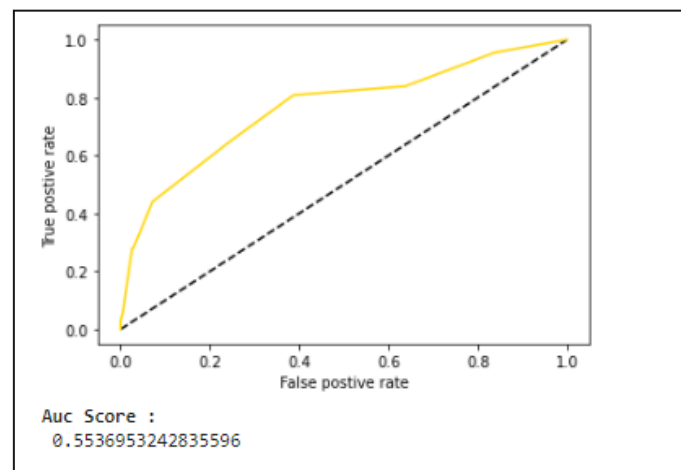
```

Final_mod = RandomForestClassifier(bootstrap=True,criterion='entropy',n_estimators= 15, max_depth=20 ,max_features='log2')
Final_mod.fit(X_train,Y_train)
y_pred=Final_mod.predict(X_test)
print('\033[1m'+Accuracy Score : '+'\033[0m\n', accuracy_score(Y_test, y_pred))

Accuracy Score :
0.8602620087336245

```

We can see that Final model with hyper parameter tuning leads to slight decrease in accuracy score from 0.8980 in original model to 0.8915. This complete possible we will use model with default values as our final model. AOC-ROC score of final random forest classifier model is shown below:



At last, we will save final model with joblib library, so it can be deploy on cloud platform.

Saving model

```
# Saving the model to local file system
#filename = 'finalized_model.pickle'
import pickle
pickle.dump(log_reg, open('reg_model', 'wb'))
```

CONCLUDING REMARKS ON EDA AND ML MODEL

- *Bench mark of 6900\$ monthly income is recommended to Prevent attrition.*
- *Attrition rate is high in age group of 29 to 33. HR needs to keep eye over need & expectation of this age group from company.*
- *Percentage of attrition is high in Sales Representative, Laboratory Technician*
- *16 % attrition rate among Research Scientist and no company afford to lose them.*
- *Almost 50% employs in sales department from different education background. There is possibility of dissatisfaction among them as attrition high among these.*
- *Different feature engineering techniques like balancing data, outliers' removal, label encoding, feature selection & PCA are perform on data.*
- *Random Forest Classifier model gives maximum Accuracy.*

You can get code of this case study from my [GetHub Link](#).