MACHINE LEARNING TO UNDERSTAND AND PREDICT HR ATTRITION

The one very single thing for grand success of any business Enterprise, startup in current competitive era, where razor edge like fine technological upgradation take place continuously along with continuous evolving SOP's, business model? Fundamental answer to this question trace back to core of business which is need & demand of product.

So next Super Fundamental question comes here: from where product comes? Product comes from innovative ideas & solution from talented minds and effort of domain expertise people making idea into reality, a useful product.

The Key to Success in an Organisation is the ability to attend and retain top talents. But what is the role of Machine Learning here? For that we need to go in background of HR Analytics.

HR department hire lot of employees every year. 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. HR department often play significant role in designing company compensation programs, creating ambiance in work environment with various team activities, imprinting work culture habits on mindset of peoples and training skill systems that help the organization retain smart minds & talented brain. Most of organisation run different HR analytics vertical to gather data, analyse data to improve process within organisation and to make major decision about human resources.

The gradual loss of employees' overtime refers as HR attrition. Attrition can happen due to retirement, involuntary (employee is fired or terminated) or voluntary (resigns from an organization).

Just like several factors contribute towards building a reliable team and organization, similarly, numerous factors contribute to the attrition rate: Improper work-life balance, better job opportunities, salary hike, Lack of growth or work recognition, unhealthy relations with managers. We can gather data about these factors and utilise Machine learning classification techniques to predict whether employee like to leave organisation or stay in organisation. Here I will show you what leads employee's attrition and Predication of attrition using case study on IBM HR analytics Dataset.

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 my GitHub profile here. 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 categorial in nature, this case study falls into classification machine learning problem. We have two objectives here:

DATA PREPARATION: LOAD, CLEAN and FORMAT

First, we have to import libraries for EDA and Dataset itself

In [161]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
from sklearn import tree
from sklearn.metrics import accuracy_score
import codecs
from sklearn.neighbors import KNeighborsRegressor
```

IMPORTING CSV FILE TO JUPYTER NOTEBOOK

In [92]:
```
hr_df=pd.read_csv(r'C:\Users\AMEET\Desktop\Anuja\Data_Trained\FLIP_BOBO_Intership\Evaluation Projects\WA_Fn-UseC_-HR-Emp
hr_df
```

In [4]:
```
print('No of Rows', hr_df.shape[0])
print('No. of Columns', hr_df.shape[1])
```

```
No of Rows 1470
No. of Columns 35
```

In [93]:
```
hr_df.head()
```

Out[93]:

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

5 rows × 35 columns

```
In [96]: 1 hr_df.columns

Out[96]: 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')
```

Checking different Datatypes in Datasets:

```
In [98]: 1 hr_df.columns.to_series().groupby(hr_df.dtypes).groups

Out[98]: {int64: ['Age', 'DailyRate', 'DistanceFromHome', 'Education', 'EmployeeCount', 'EmployeeNumber', 'EnvironmentSatisfaction',
'HourlyRate', 'JobInvolvement', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'Percen
tSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel', 'TotalWorkingYears', 'Tr
ainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrMa
nager'], object: ['Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'Over
18', 'OverTime']}
```

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,

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

Checking Data Integrity

```
In [99]: 1 hr_df.duplicated().sum()

Out[99]: 0

In [5]: 1 hr_df.isnull().sum().any()

Out[5]: False

In [6]: 1 hr_df.isin([' ','NA','-','?']).sum().any()

Out[6]: False
```

There is no missing data, which makes it easier to work with the Dataset.

Dataset doesn't contain any Duplicate Entry, Whitespace , 'NA', or '-'.

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: -

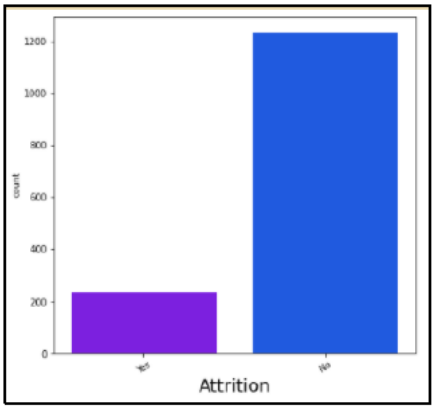
- Minimum Employee Age is 18 and Maximum age of employee 60.
- Average distance from home is 9.1 KM. It means that most of employee travel at least 18 KM in day from home to office.
- Average performance Rating of employees is 3.163 with min value 3.0. This Means that performance of most of employee 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.

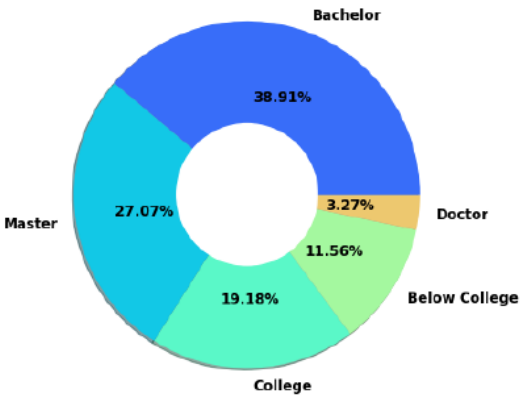
```
In [104]: 1 hr_df['Attrition'].value_counts()

Out[104]: Attrition
          No     1233
          Yes     237
          Name: count, 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 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.

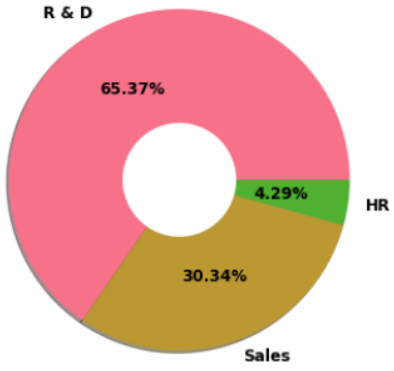
Education Level of Man Power Available:



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.

Department Wise Distribution of Man Power



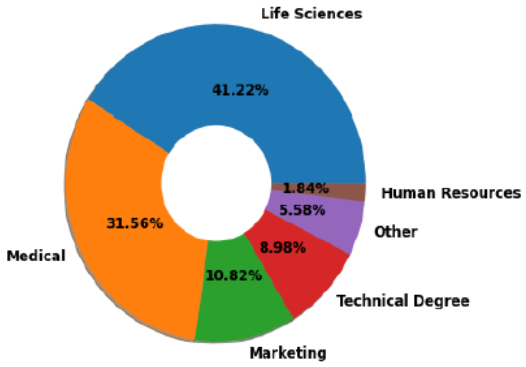
Education Level of Employees working in different Department

Department / Education	Human Resources	Research & Development	Sales	All
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.

Employee Distribution as per Education Field:



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.

```
In [117]: 1 pd.crosstab([hr_df.Department],[hr_df.EducationField], margins=True).style.background_gradient(cmap='summer_r')
```

Out[117]:

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.

Key Insights from above Cross Tab:

- There are only 27 people with HR background and Weknow that 63 people work in HR Department from previous result. This implies thatat least half employee working in HR department do not haveHR 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 interestingto see attrition rate in these employees.

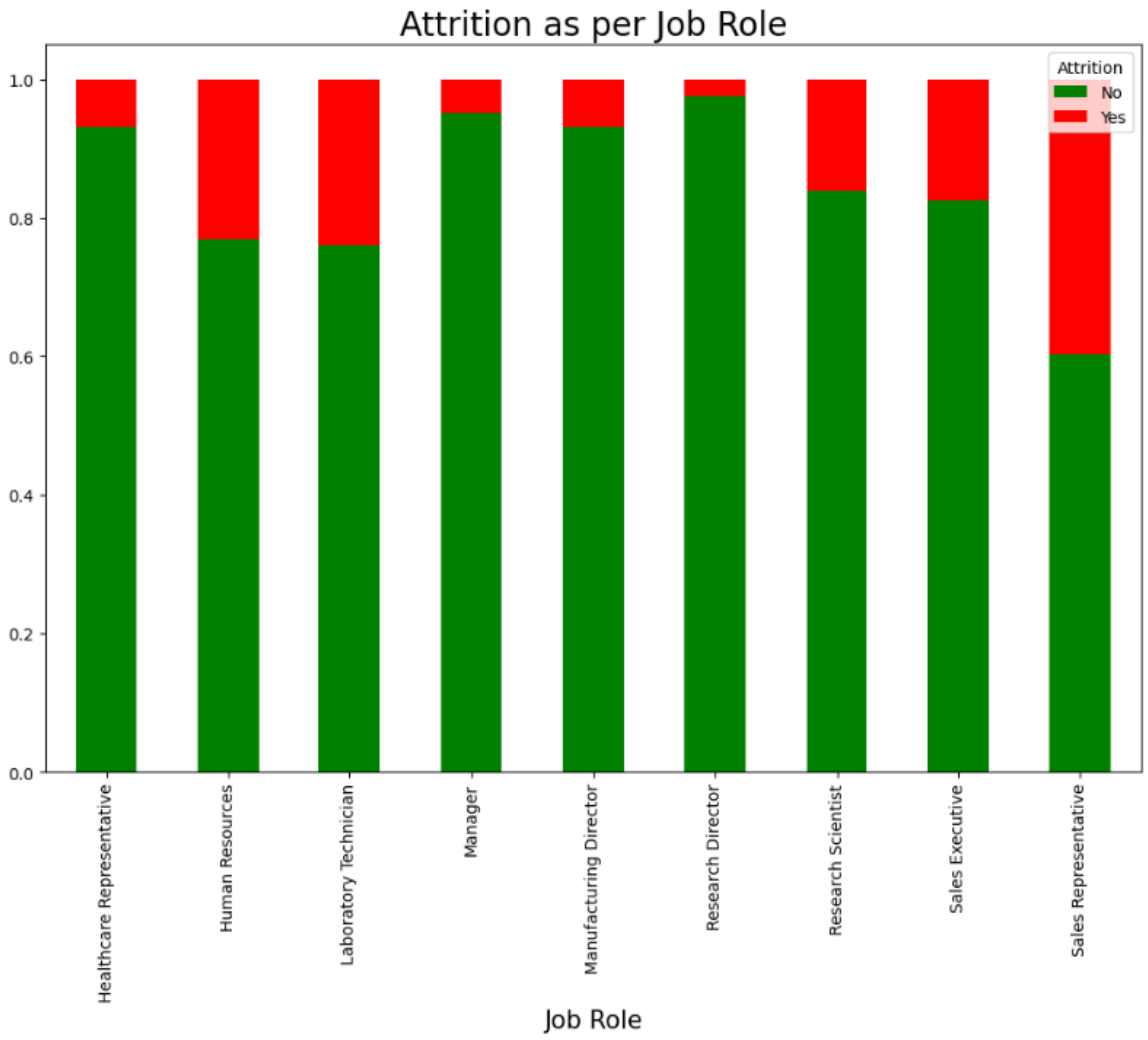
We will Analyse Attrition in department according to education background based on above insight further but before that explore Job role in order to include it in further attrition analyse. First build matrix of department vs job role which will give us idea about number of employees of different job role across department.

Department JobRole	Human Resources	Research & Development	Sales	All
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 job role, again this time using crosstab.

```
In [121]:  1 pd.crosstab([hr_df.JobRole,hr_df.Department],[hr_df.Attrition], margins=True).style.background_gradient(cmap='summer')
```

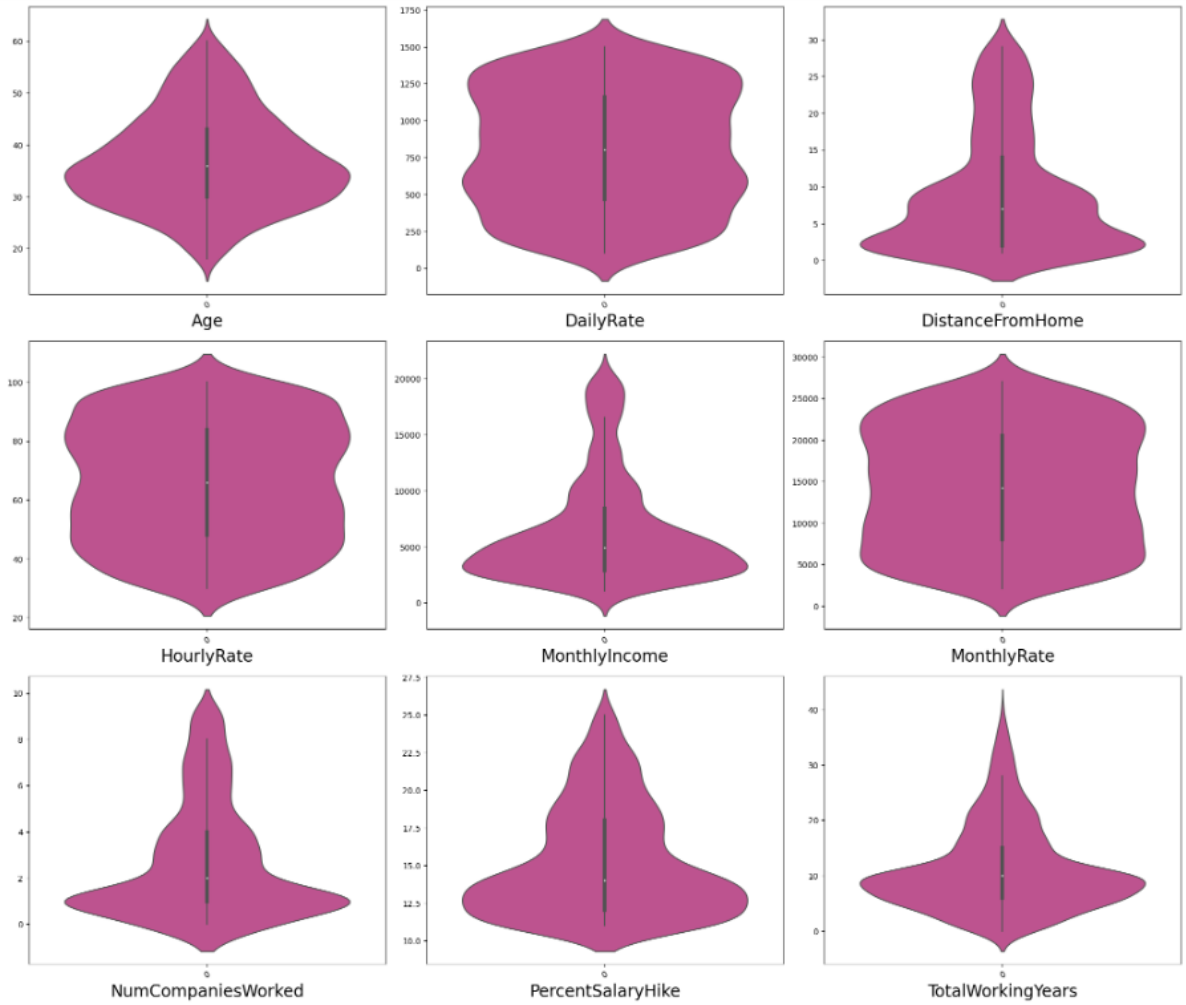
Out[121]:

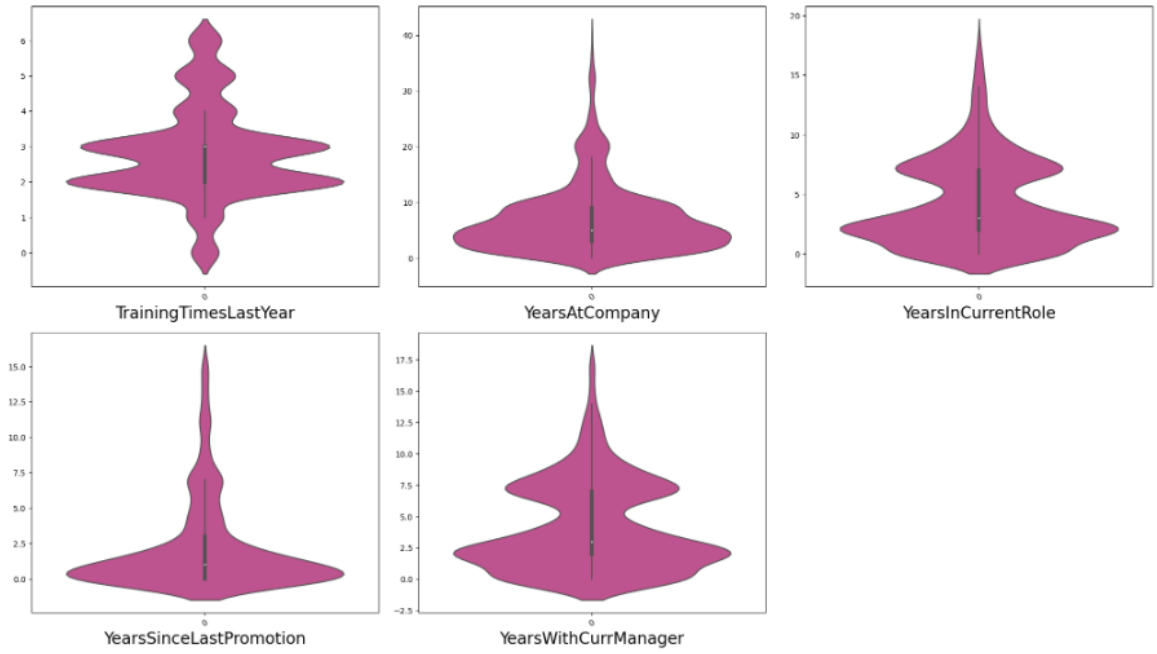
Attrition		No	Yes	All
JobRole	Department			
Healthcare Representative	Research & Development	122	9	131
Human Resources	Human Resources	40	12	52
Laboratory Technician	Research & Development	197	62	259
	Human Resources	11	0	11
Manager	Research & Development	51	3	54
	Sales	35	2	37
Manufacturing Director	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.

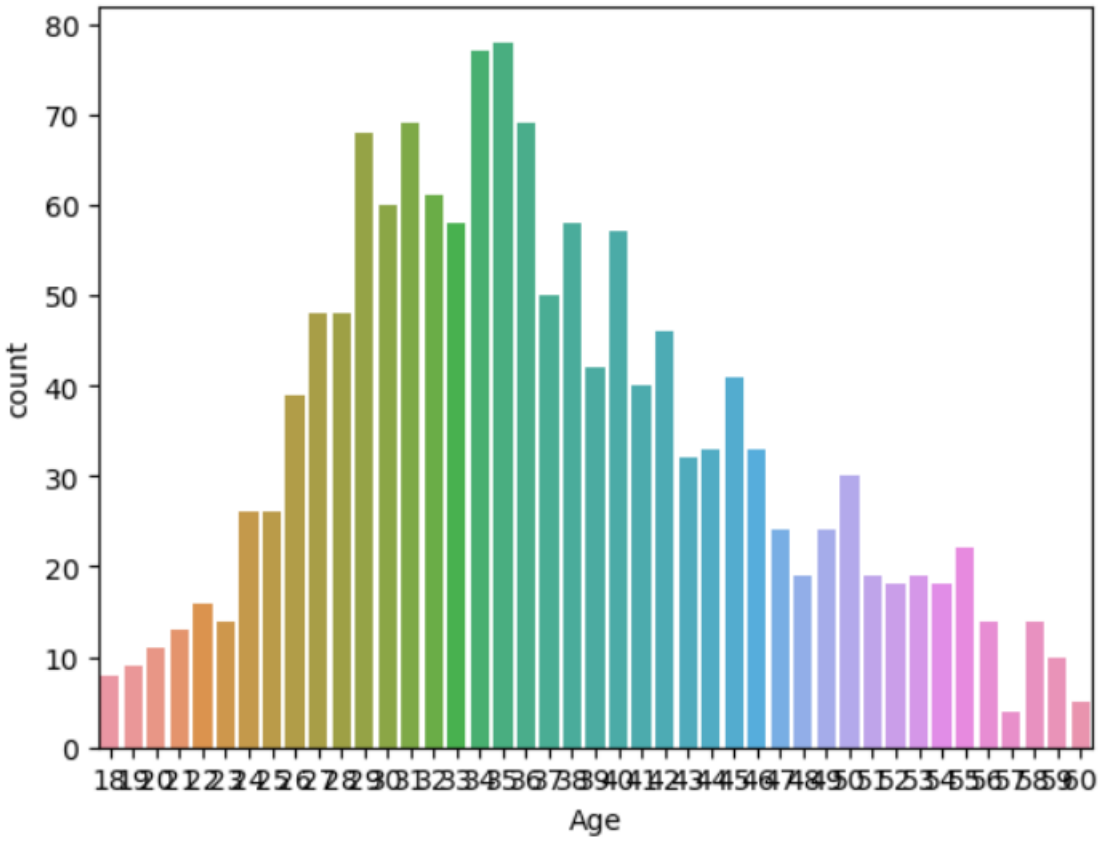




Key Insights from above Cross Tab:

- For Majority of people have spend 3 to 10 years at company.
- Most of people staying company upto 2 years after promotion.
- Majority of people are are train 2-3 times in last year.If employees leaves job then it loss investment for company.
- 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?

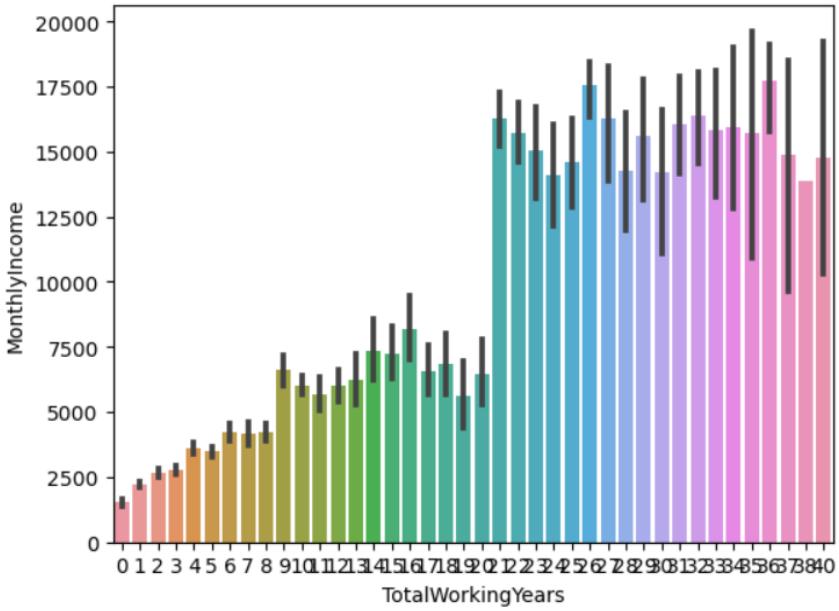


Key Insights from count plot of Age Vs Attrition:

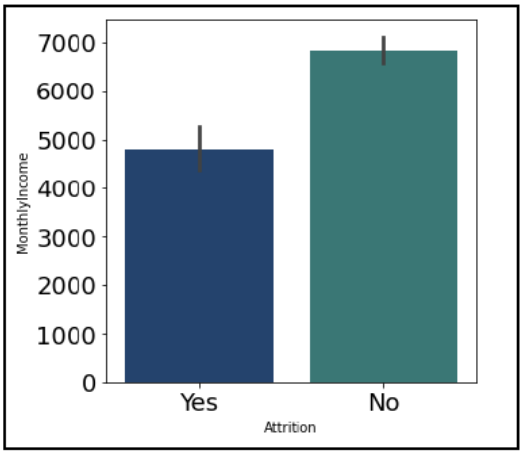
- The Attrition rate is minimum between the Age years of 34 and 45.
- The Attrition rate is maximum between the Age years of 29 and 33.

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

```
In [127]:  1  sns.barplot(x = 'TotalWorkingYears', y = 'MonthlyIncome', data = hr_df)
Out[127]: <Axes: xlabel='TotalWorkingYears', ylabel='MonthlyIncome'>
```



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
- Outlier's 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.

```
In [132]:
1 hr_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.

```
In [131]:
1 from sklearn.preprocessing import LabelEncoder
2 le = LabelEncoder()
3 hr_df["Attrition"] = le.fit_transform(hr_df["Attrition"])
4 hr_df.head()
```

Out[131]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	Relations
0	41	1	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	0	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	1	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	0	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	0	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	

5 rows × 35 columns

```
In [134]:
1 from sklearn.preprocessing import OrdinalEncoder
2 oe = OrdinalEncoder()
3 def ordinal_encode(hr_df, column):
4     hr_df[column] = oe.fit_transform(hr_df[column])
5     return hr_df
6
7 oe_col = ['BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'OverTime']
8 hr_df=ordinal_encode(hr_df, oe_col)
9 hr_df.head()
```

Out[134]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	...	PerformanceI
0	41	1	2.0	1102	2.0	1	2	1.0	2	0.0	...	
1	49	0	1.0	279	1.0	8	1	1.0	3	1.0	...	
2	37	1	2.0	1373	1.0	2	2	4.0	4	1.0	...	
3	33	0	1.0	1392	1.0	3	4	1.0	4	0.0	...	
4	27	0	2.0	591	1.0	2	1	3.0	1	1.0	...	

5 rows × 31 columns

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

3. OUTLIER'S DETECTION AND REMOVAL

Identifying outliers and bad data in your dataset is probably one of the most difficult parts of data clean-up, and it takes time to get right. Even if you have a deep understanding of statistics and how outliers might affect your data, it's always a topic to explore cautiously.

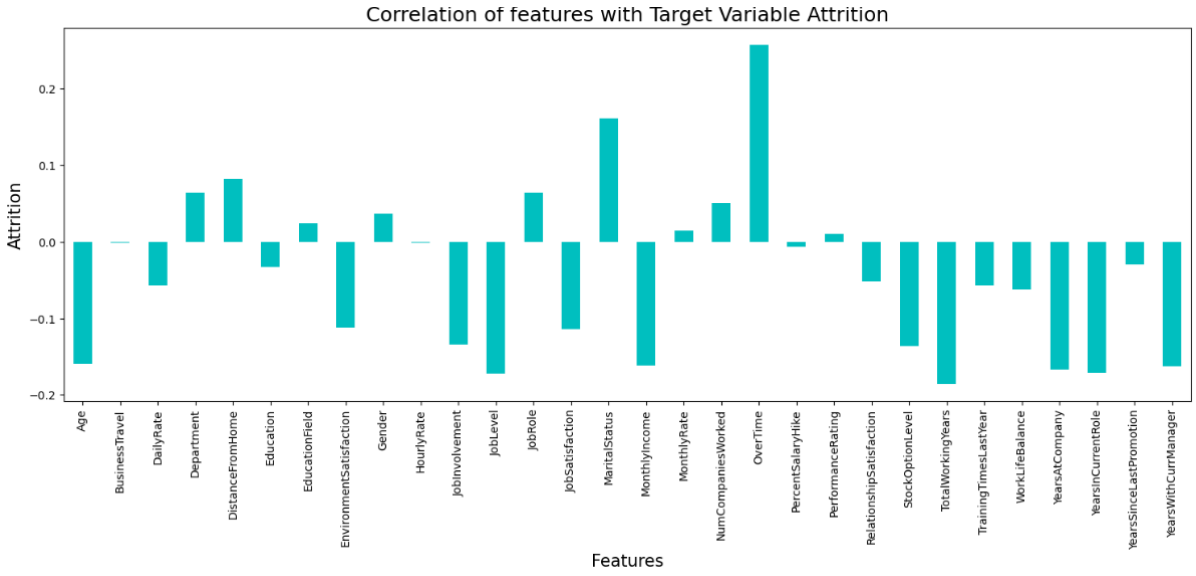
Machine learning algorithms are sensitive to the range and distribution of attribute values. Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models and ultimately poorer results. Outliers can be seen in boxplot of numerical feature. We did not 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 Heatmapshow in a glance which variables are correlated, to what degree, in which direction, and alerts us to potential multicollinearity problems.The bar plotof correlation coefficient of targetvariable with independent features shown below :



5. MULTICOLLINEARITY BETWEEN FEATURES

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

In [144]:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif= pd.DataFrame()
vif['VIF']= [variance_inflation_factor(hr_df.values,i) for i in range(hr_df.shape[1])]
vif['Features']= hr_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 in within permissible limit of 10. Multicollinearity do not pose any threat here.

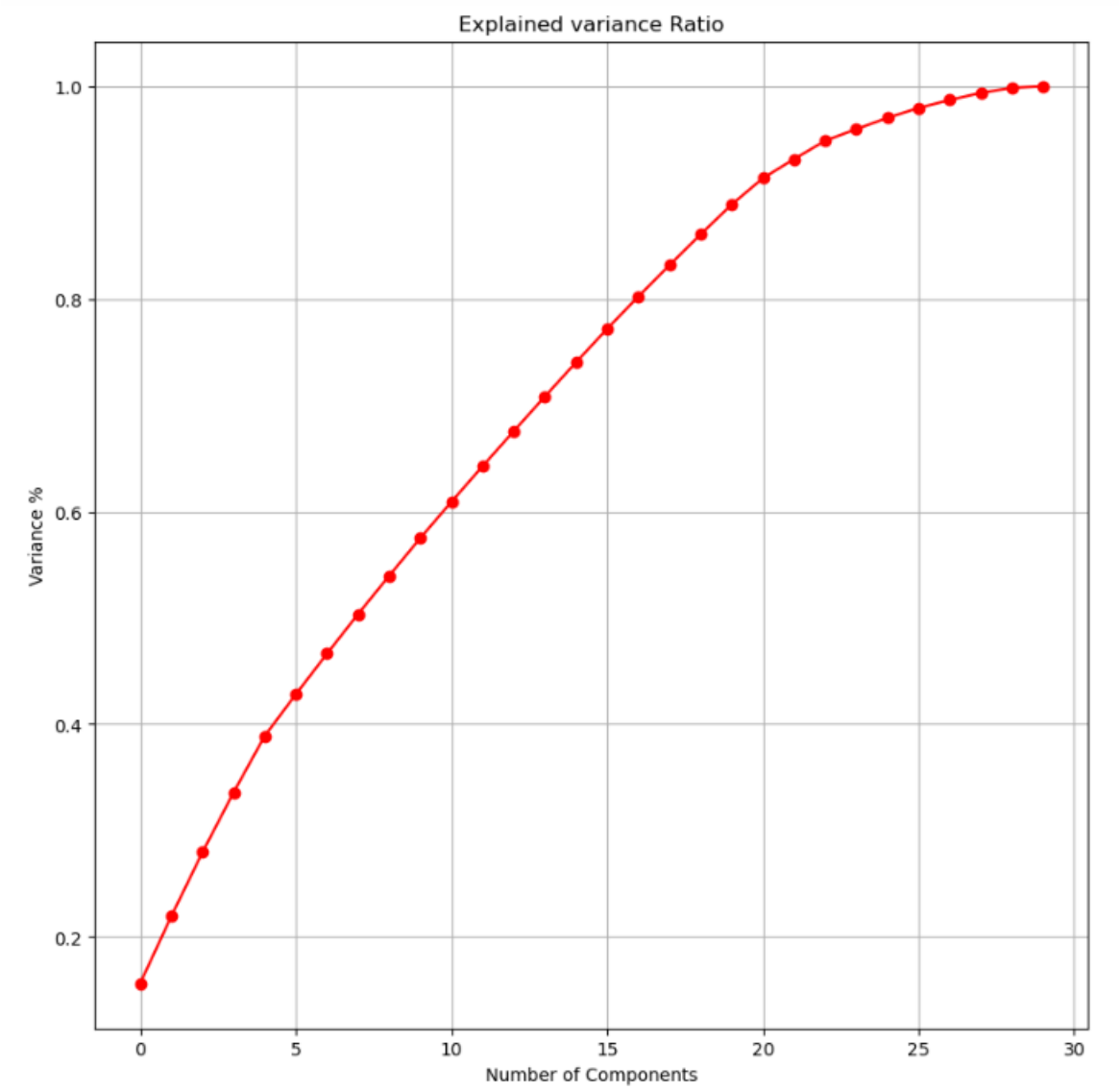
6. SCALING OF DATA USING STANDARD SCALAR

In [148]:
```
from sklearn.preprocessing import StandardScaler
scaler= StandardScaler()
X_scale = scaler.fit_transform(X)
```

In [149]:
```
from sklearn.decomposition import PCA
pca = PCA()
x_pca = pca.fit_transform(X_scale)
plt.figure(figsize=(10,10))
plt.plot(np.cumsum(pca.explained_variance_ratio_), 'ro-')
plt.xlabel('Number of Components')
plt.ylabel('Variance %')
plt.title('Explained variance Ratio')
plt.grid()
```

7. DIMENSIONALITY REDUCTION USING PCA

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



In [150]:
```
pca_new = PCA(n_components=21)
x_new = pca_new.fit_transform(X_scale)
```

In [151]:
```
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

In [154]:
```
from sklearn.model_selection import train_test_spilit
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: (984, 21)
Training target vector size: (984,)
Test feature matrix size: (486, 21)
Test target vector size: (486,)
```

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).

In [155]:
```
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 0.8991769547325102 on Random_state 123
```

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

```
In [156]:
X_train, X_test, Y_train, Y_test = train_test_split(principle_x, Y, random_state=242, test_size=.33)
log_reg=LogisticRegression()
log_reg.fit(X_train,Y_train)
y_pred=log_reg.predict(X_test)
print('\033[1m'+'Logistics Regression Evaluation'+'\033[0m')
print('\n')
print('\033[1m'+'Accuracy Score of Logistics Regression :'+'\033[0m', accuracy_score(Y_test, y_pred))
print('\n')
print('\033[1m'+'Confusion matrix of Logistics Regression :'+'\033[0m \n',confusion_matrix(Y_test, y_pred))
print('\n')
print('\033[1m'+'classification Report of Logistics Regression'+'\033[0m \n',classification_report(Y_test, y_pred))
```

```
Logistics Regression Evaluation


Accuracy Score of Logistics Regression : 0.8662551440329218


Confusion matrix of Logistics Regression :
 [[397   9]
 [ 56  24]]


classification Report of Logistics Regression
              precision    recall  f1-score   support

           0       0.88      0.98      0.92       406
           1       0.73      0.30      0.42        80

    accuracy                           0.87       486
   macro avg       0.80      0.64      0.67       486
weighted avg       0.85      0.87      0.84       486
```

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
Logistics Regression	0.8705	0.6869	0.87	0.87	0.87
SVC	0.9019	0.6075	0.90	0.90	0.90
GaussianNB	0.8470	0.7405	0.85	0.85	0.85
DecisionTreeClassifier	0.8039	0.8381	0.80	0.80	0.80
KNeighborsClassifier	0.8379	0.7374	0.84	0.84	0.85
RandomForestClassifier	0.8980	0.9171	0.90	0.87	0.93
AdaBoostClassifier	0.8457	0.8718	0.85	0.85	0.85
GradientBoostingClassifier	0.8560	0.8831	0.86	0.86	0.86
Bagging Classifier	0.8792	0.8856	0.87	0.87	0.88

(Min Value in column -Green, Max Value in column - Pink Colour)

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.

In [165]:
```
from sklearn.model_selection import GridSearchCV
```

In [167]:
```
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]}
```

In [168]:
```
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
```

Out[168]:
```
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)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [169]:
```
GCV.best_params_
```

Out[169]:
```
{'bootstrap': True,
 'criterion': 'entropy',
 'max_depth': None,
 'max_features': 'log2',
 'n_estimators': 15}
```

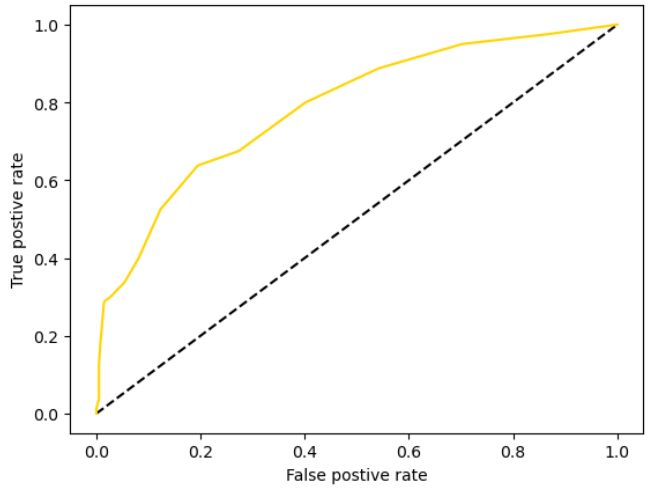
Next step is to build final machine learning model over best params in Hyper parameter tuning.

In [170]:
```
Final_mod = RandomForestClassifier(bootstrap=True,criterion='entropy',n_estimators= 25, 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 :
 0.8518518518518519
```

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:

In [171]:
```
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve

y_pred_prob = Final_mod.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = roc_curve(Y_test,y_pred_prob)
plt.plot([0,1],[0,1], 'k--')
plt.plot(fpr, tpr, label='Random Forest Classifier')
plt.xlabel('False postive rate')
plt.ylabel('True postive rate')
plt.show()
auc_score = roc_auc_score(Y_test, Final_mod.predict(X_test))
print('\033[1m'+'Auc Score :'+'\033[0m\n',auc_score)
```




```
Auc Score :
 0.5600369458128079
```

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

In [174]:
```
import joblib
joblib.dump(Final_mod,'IBM_HR_Analytics_Final.pkl')
```

Out[174]: ['IBM_HR_Analytics_Final.pkl']

CONCLUDING REMARK 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 need to keep an 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 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 GitHub Profile.