HR Analytics Project- Machine Learning

Machine Learning to Understand & Predict HR Attrition

Do you want to know what is that one single thing for the grand success of any business Enterprise, startup in the current competitive era, where razor edge like fine technological upgradation take place continuously along with continuous evolving SOP's, business model? Fundamental answers to this question trace back to the core of business which is the need & demand of products.

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

The key to success in an organization is the ability to attract and retain top talents. But where does Machine learning come from? For that we need to go in the background of HR. analytics.

The HR department hires a 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 departments often play a significant role in designing company compensation programs, creating ambiance in the 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 brains. Most organizations run different HR analytics verticals to gather data, analyze data to improve processes within the organization and to make major decisions about human resources.

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

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. For Tech companies' loss of talented brain means loss of domain expertise, knowledge base as for these company's Technological capability exists in domain expertise, intellectual property rights of employees. On the marketing or sales side customers often prefer to interact with familiar people.

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 utilize Machine learning classification techniques to predict whether employees like to leave an organization or stay in an organization. Here I will show you.

HR Analytics Project- Machine Learning

What leads employee's attrition and Prediction of attrition using case study on IBM HR analytics Dataset.

IBM HR Analytics Employee Attrition & Performance Dataset

In this case study we will use the IBM HR Analytics database. This fictional dataset was created by IBM employees and available to download from GitHub and Kaggle. You can also download the 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 the 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.**

Data Preparation: Load, Clean and Format

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

Importing Library:

```
# Pandas is a useful library in data analysis, Numpy library used for working with arrays
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

```
# Pandas read_csv function used for reading the csv file.
df = pd.read_csv("C:\\Users\\prath\\WA_Fn-UseC_-HR-Employee-Attrition.csv")
df.head()
```

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

5 rows × 35 columns

Checking different datatypes in dataset: -

```
# As we have 35 Columns Lets sort Columns by their datatype
df.columns.to_series().groupby(df.dtypes).groups
```

```
{int64: ['Age', 'DailyRate', 'DistanceFromHome', 'Education', 'EmployeeCount', 'EmployeeNumber', 'EnvironmentSatisfaction', 'HourlyRate', 'JobInvolvement', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager'], object: ['Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'Over18', 'OverTime']}
```

We have 9 features with object data types and the rest are Numeric features with int64. Out of all numeric features Education, Environment-Satisfaction, Job-Involvement, Job-Satisfaction, Relationship-Satisfaction, Performance Rating, Work Life Balance are ordinal variables. These ordinal features have a 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 the dataset.

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

```
0
```

```
df.isnull().sum().any() # Presense of Missing values
```

```
False
```

```
df.isin([' ','NA','-','?']).sum().any() # check if any whitespace, 'NA' or '-' exist in dataset.
```

```
False
```

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 '-'.

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

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

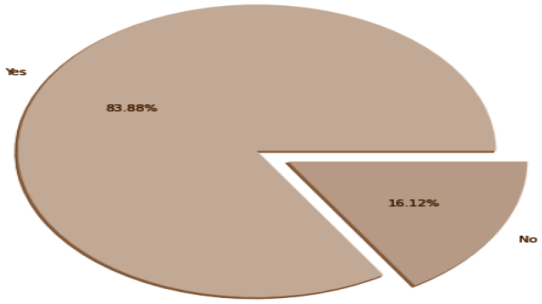
- **Minimum Employee Age is 18 and Maximum age of employee 60.**
- **Average distance from home is 9.1 KM. It means that most employees travel at least 18 KM a day from home to office.**
- **Average performance Rating of employees is 3.163 with min value 3.0. This Means that the performance of most employees is 'Good'. <u>This implies that Attrition of Employee with 'Outstanding' or 5 ratings need to be investigated.</u>**
- **50% of Employees have 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 deviations 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

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

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

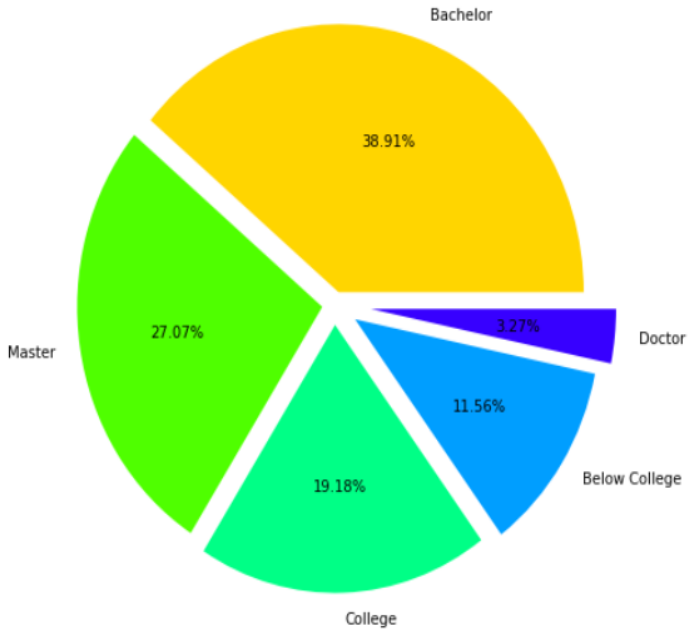
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```



83.88% (1237 employees) Employees did not leave the organization while 16.12% (237 employees) did leave the organization making our dataset to be considered 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 roles & job positions not in alignment with educational background can lead to attrition. Let's investigate this by visualizing these features one by one to gain more insights.

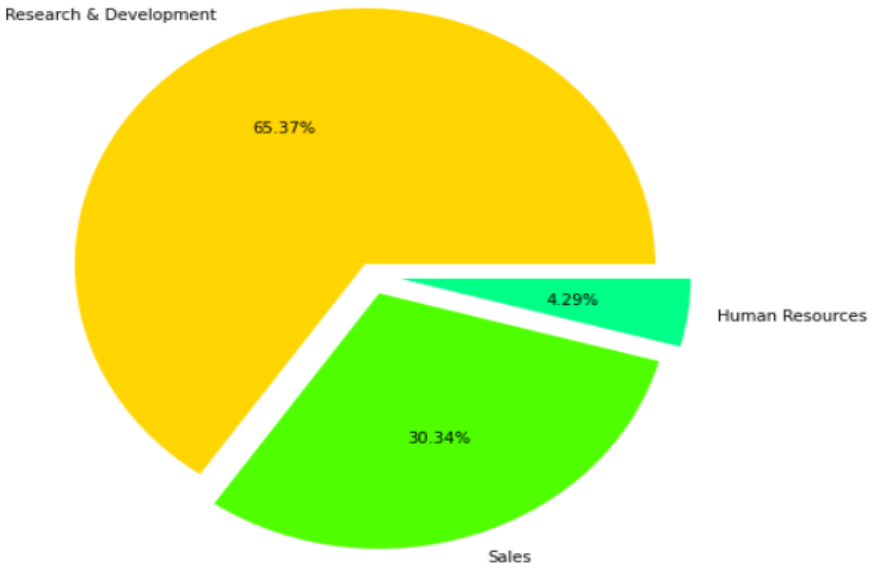
Education level of Manpower available:



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 Manpower:-



Education Level of Employees working in different Department

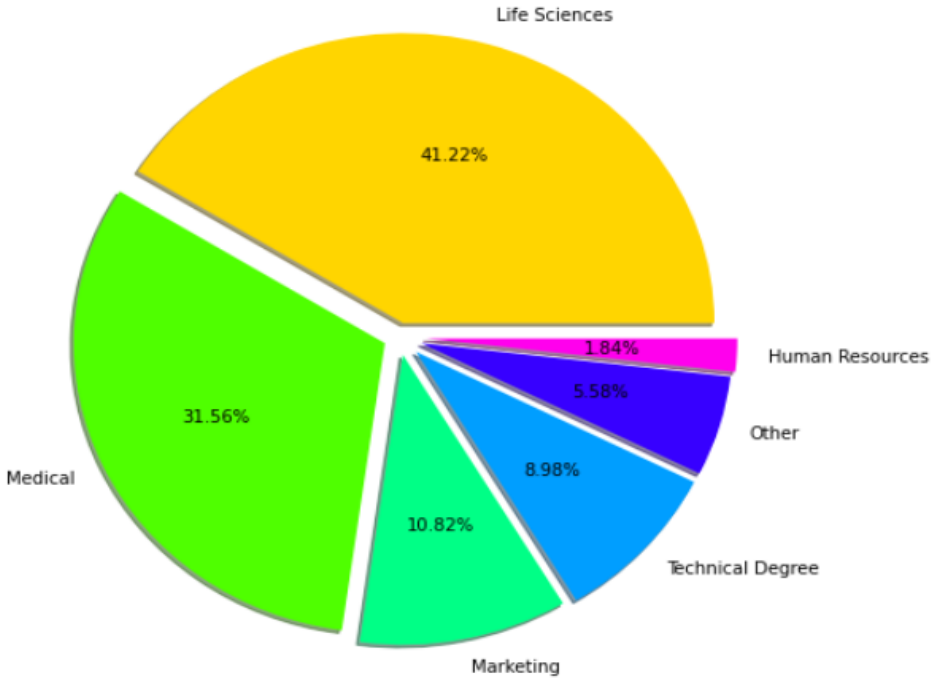
Department Education	Human Resources	Research & Development	Sales	All
1	5	115	50	170
2	13	182	87	282
3	27	379	166	572
4	15	255	128	398
5	3	30	15	48
All	63	961	446	1470

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

1. **65.37% of Employees work inside the Research & Development Department. Out of Total 961 employees, the number of employees with education level of Bachelors, Masters, and Doctor are 379, 255 and 30 respectively.**

2. **Only 63 employees work in the HR department.**

Employee distribution as per education field:



EducationField Department	Human Resources	Life Sciences	Marketing	Medical	Other	Technical Degree	All
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

Key Insights from Pie Plot

1. **Employees belong 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.**

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

Key Insights from above Cross Tab:

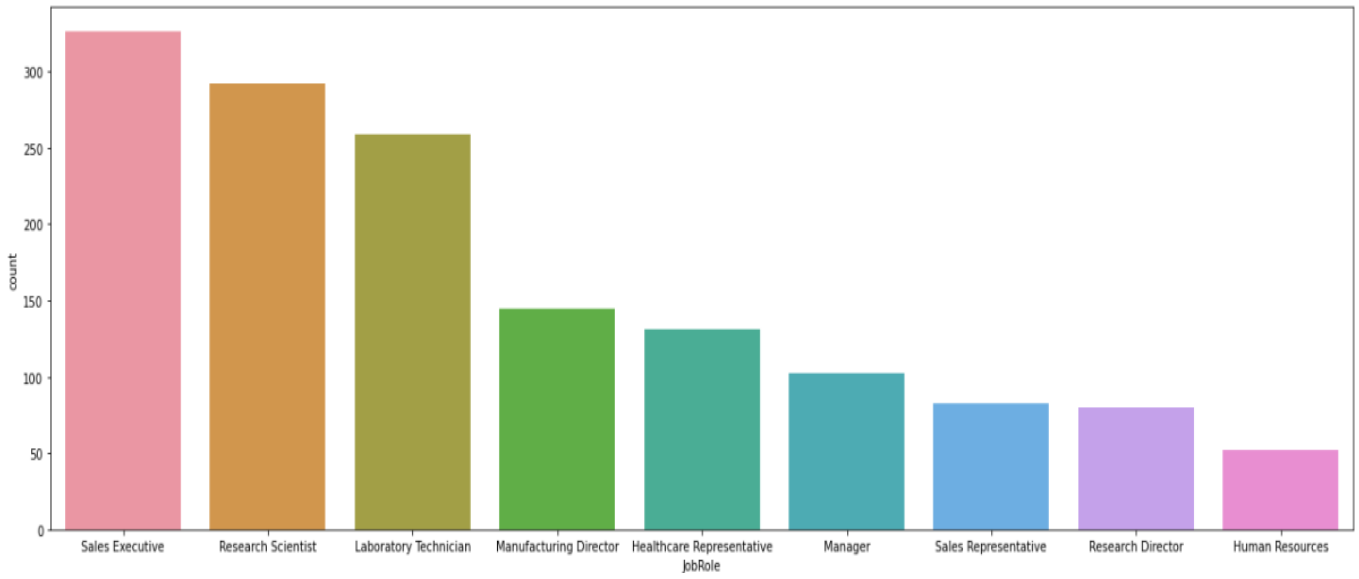
➢ There are only 27 people with HR background and We know that 63 people work in HR Department from previous result. This implies that at least half employee working in the HR department do not have HR background.

➢ R&D department almost everyone comes from domain expertise or technical background except support staff. These employees usually have high salaries, 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 clearly see they are working in a domain to which their educational background does not belong. So, it will be interesting to see the attrition rate in these employees.

Attrition by Job role :

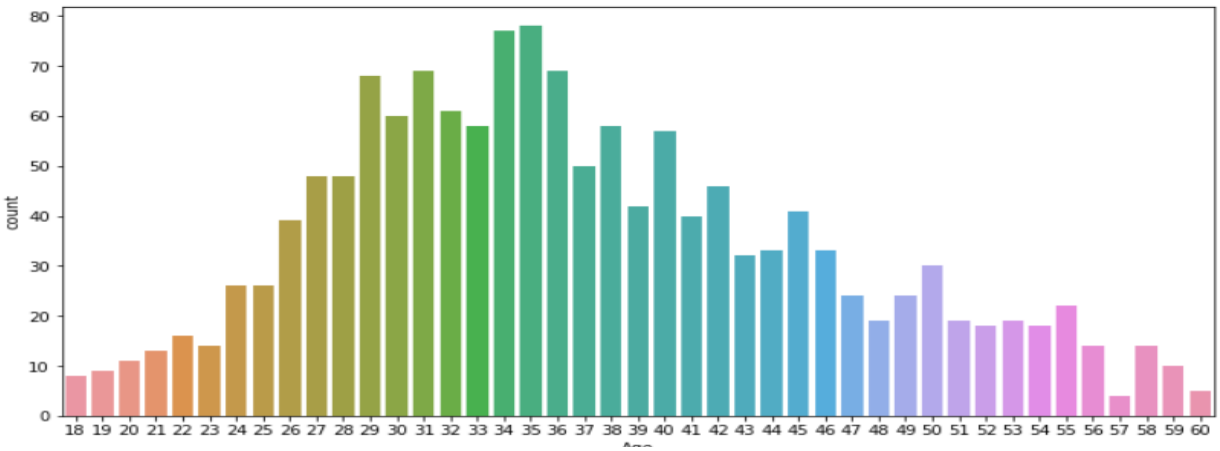


There are 3 job roles in the HR Department, maximum of which are sales Executive with 446 Total Employees.

Human Resources department has 2 Job role

There are 6 different Job roles in the R & D department with a total 961 employees and until now we know that all of them belong to their respective domain background.

which age group attrition rate is high:



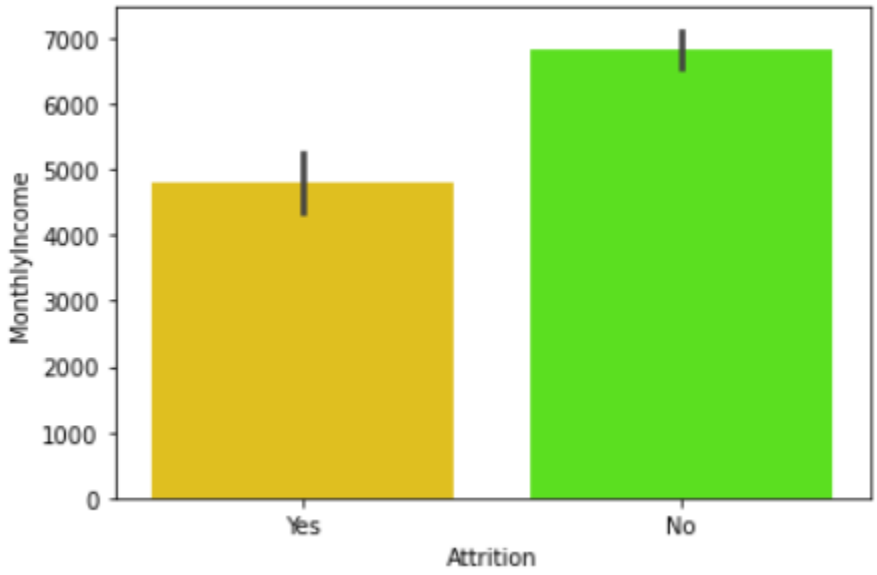
Key Insights from count plot of Age Vs Attrition:

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.

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 the 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 the major reason behind attrition.

➢ To prevent attrition average monthly to be greater than 6900$ is recommended.

Feature Engineering: Data Pre-processing:

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 mentioned 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 Variety of techniques used to achieve the above mentioned 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 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 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. 11 HR Analytics Project- Machine Learning to Understand & Predict HR Attrition**
➢ **Feature selection Techniques:**
✓ **Correlation Matrix with Heatmap**
✓ **Univariate Selection – SelectKBest**
✓ **ExtraTreesClassifier**

1. Dropping unnecessary features

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

```
# Droping unnecessary columns
df.drop(["EmployeeCount", "EmployeeNumber", "Over18", "StandardHours","PerformanceRating"], 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

```
# Ordinal Encoding
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()
```

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

3. Outliers' detection and removal

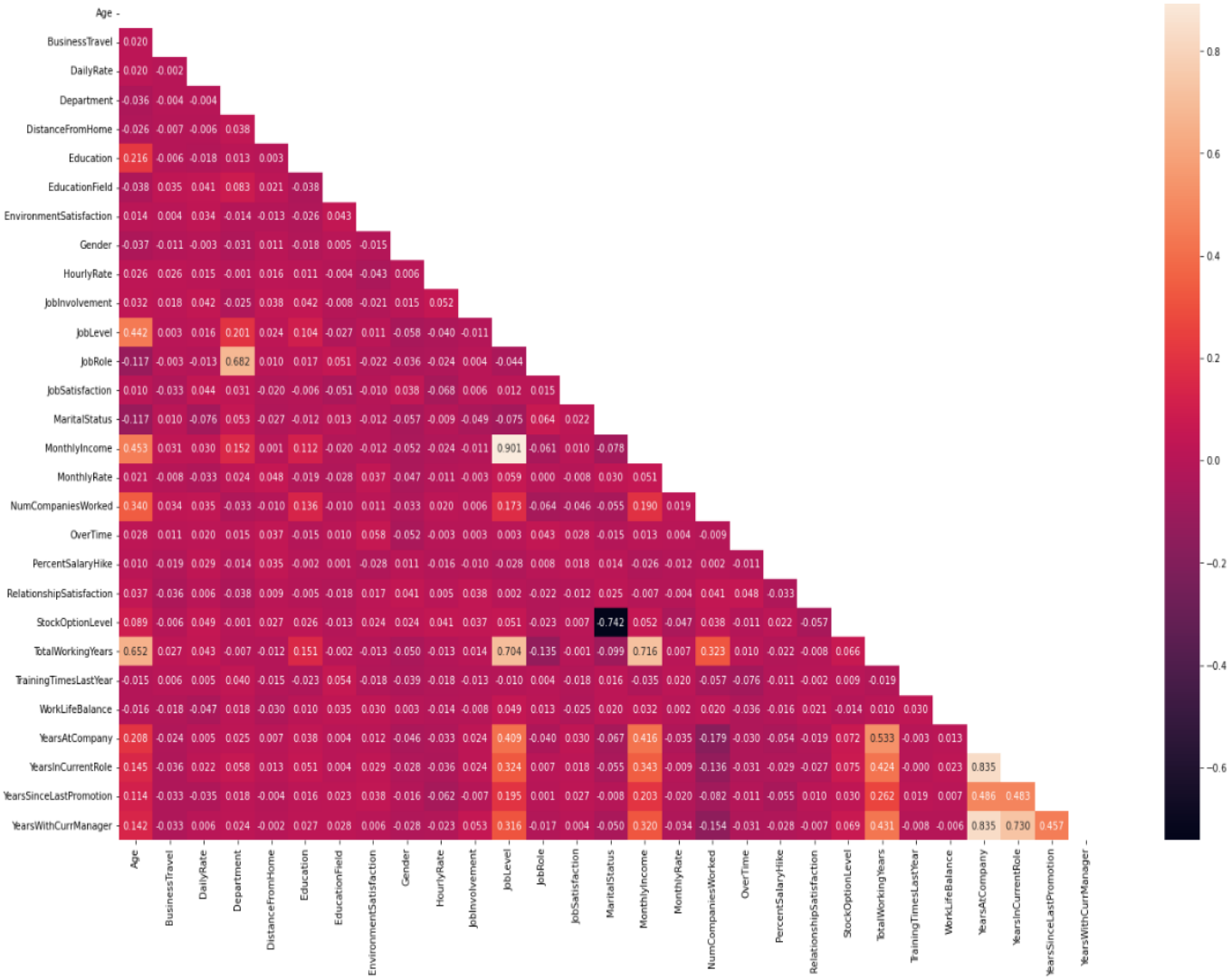
Machine learning algorithms are sensitive to the range and distribution of attribute values. Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models and ultimately poorer results. Outliers can be seen in boxplot of numerical feature. We did not 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. 4. Correlation Heatmap.

```
def threhold(z,d):
    for i in np.arange(3,4,0.01):
        data=d.copy()
        data=data[(z<i).all(axis=1)]
        loss=(d.shape[0]-data.shape[0])/d.shape[0]*100
        print('With threshold {} data loss is {}%'.format(np.round(i,2),np.round(loss,2)))
```

```
#Using zscore method to remove outliers
from scipy.stats import zscore
z=np.abs(zscore(df))
threhold(z,df)
```

4. Correlation Heatmap

Correlation Heatmap shows 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



5. 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

6. 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 overrepresented class. We can resolve this by Synthetic Minority Oversampling Technique (SMOTE) to over-sample the minority class.

```
from imblearn.over_sampling import SMOTE
oversample = SMOTE()
X, Y = oversample.fit_resample(X, Y)
```

```
Y.value_counts()
```

```
1    1158
0    1158
Name: Attrition, dtype: int64
```

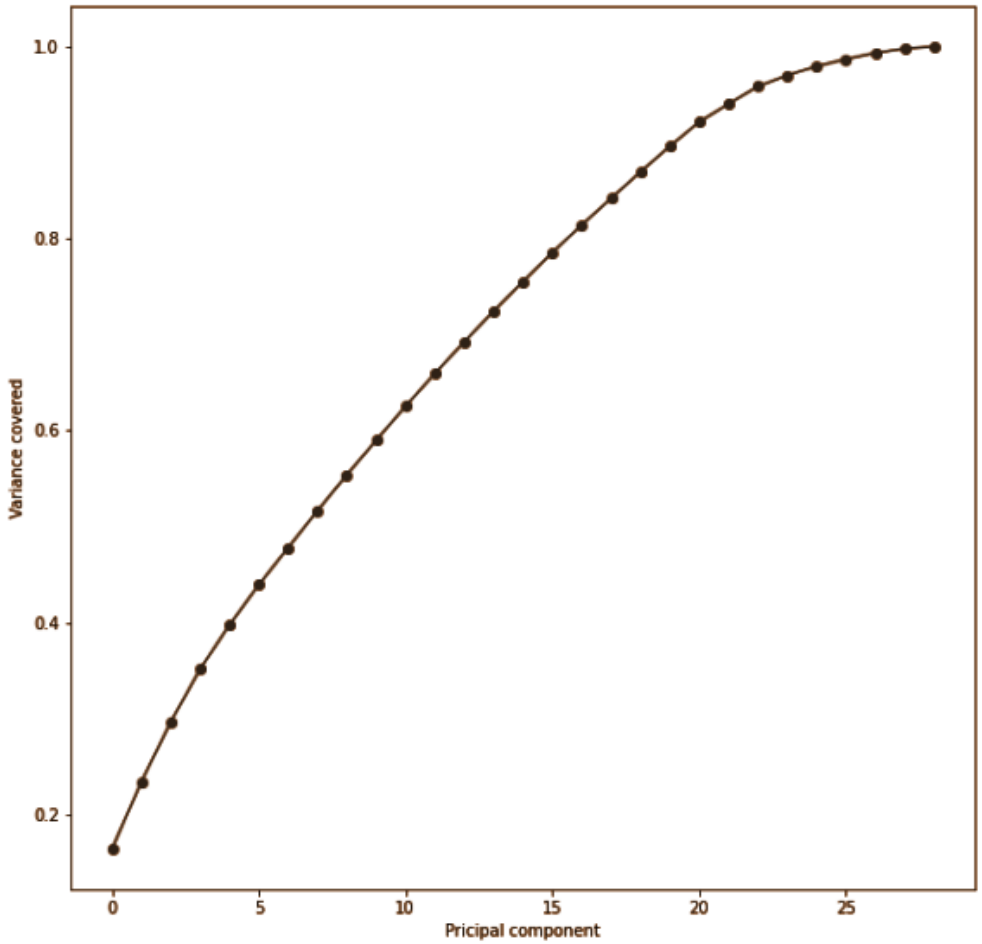
7. Scaling of data using Standard Scalar

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

8. Dimensionality Reduction Using PCA

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

```
plt.figure(figsize=(10,10))
plt.plot(np.cumsum(pca.explained_variance_ratio_), 'ro-')
plt.xlabel('Pricipal component')
plt.ylabel('Variance covered')
plt.show()
```



22 principal components attribute for 90% of variation in the data. We shall pick the first 22 components for our prediction.

```
pca_new = PCA(n_components=22)
x_new = pca_new.fit_transform(X_scale)
```

```
rinciple_x=pd.DataFrame(x_new,columns=np.arange(22))
```

Model Building:

In this section we will build a Supervised learning ML model-based classification algorithm. The objective is to predict attrition in 'Yes' or 'No' leads to fall problems in the domain of classification algorithm. train_test_split used to split data with size of 0.2

```
models =[KNeighborsClassifier(),DecisionTreeClassifier(),
       GradientBoostingClassifier(),RandomForestClassifier()]
```

```
max_accu = 0
maxRS = 0

for rs in range(10,100):
    x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.20, random_state = rs)
    for i in models:
        i.fit(x_train,y_train)
        pred_train = i.predict(x_train)
        pred_test = i.predict(x_test)
        accu_score = accuracy_score(y_test,pred_test)
        if accu_score>max_accu:
            max_accu =  accu_score
            maxRS = rs
            finalmodel = i

print('score :',accu_score,'random state :',rs,'Final model :',i)
print(confusion_matrix(y_test, pred_test))
print('\n')
print(classification_report(y_test, pred_test))
```

```
score : 0.9439655172413793 random state : 99 Final model : RandomForestClassifier()
```

The evaluation matrix along with classification report is as below :

```
score : 0.9439655172413793 random state : 99 Final model : RandomForestClassifier()
[[232   8]
 [ 18 206]]


              precision    recall  f1-score   support

           0       0.93      0.97      0.95       240
           1       0.96      0.92      0.94       224

    accuracy                           0.94       464
   macro avg       0.95      0.94      0.94       464
weighted avg       0.94      0.94      0.94       464
```

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 classifiers to build the final ML Model.

CrossValidation score for Random Forest Classifier:

```
RC= RandomForestClassifier()
```

```
from sklearn.model_selection import cross_val_score
score = cross_val_score(RC, X, Y, cv =5)
print("Score :" ,score)
print("Mean Score :",score.mean())
print("Std deviation :",score.std())
```

```
Score : [0.7262931  0.97192225 0.96760259 0.96976242 0.98272138]
Mean Score : 0.9236603485514262
Std deviation : 0.09882061278268042
```

Hyper Parameter Tuning :

```
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)
```

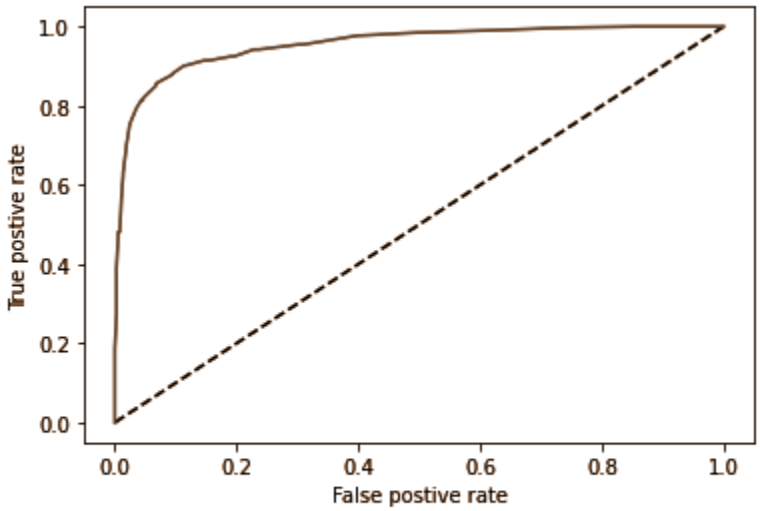
```
GCV.best_params_
```

```
{'bootstrap': True,
 'criterion': 'gini',
 'max_depth': 20,
 'max_features': 'log2',
 'n_estimators': 100}
```

```
Final_mod = RandomForestClassifier(bootstrap=True,criterion='gini',n_estimators= 100, max_depth=20 ,max_features='log2')
Final_mod.fit(x_train,y_train)
pred_test=Final_mod.predict(x_test)
print('Accuracy Score :', accuracy_score(y_test, pred_test))
```

```
Accuracy Score : 0.9504310344827587
```

We can see that Final model with hyper parameter tuning leads to a slight decrease in accuracy score from 0.95 in the original model to 0.94. This is completely possible. We will use a model with default values as our final model. AOC-ROC score of final random forest classifier model is shown below:




```
Auc Score :
 0.8919922516701346
```

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

```
import joblib
joblib.dump(Final_mod,'IBM_HR_Analytics_Final.pkl')
```

```
['IBM_HR_Analytics_Final.pkl']
```

`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 the 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% employees in sales department from different education background. There is possibility of dissatisfaction among them as attrition is high among them.
➢ 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.
[Github link]