Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022

Technical Coding Research Innovation, Navi Mumbai,
Maharashtra, India-410206

(HR Employee Attrition Analysis)

A Case-Study Submitted for the requirement of
Technical Coding Research Innovation

For the Internship Project work done during

DATA SCIENCE WITH MACHINE LEARNING AND PYTHON INTERNSHIP PROGRAM

by

M. Sai Harsha Vardhan (TCRIB3R89)
Veda Kovvali(TCRIB3R88)
Gayatri Raj Kandala(TCRIB3R86)
Harsha Udutha(TCRIB3R93)

Rutuja Doiphode
CO-FOUNDER &CEO
TCR innovation.

Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022

HR Attrition - Analysis and Prediction Using Python

Name- Veda Kovvali
Department of Computer Science
Gitam University, Vishakapatnam
kovvaliveda@gmail.com

Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022

Abstract –

The hiring process is one of the resource extensive processes for the organizations. Hiring the right talent at the right time is one of the main responsibilities of the HR department. At the same time employees leaving the organization is not good for the organization. The data from the HR department can be used for analysis. It will help us to make necessary decisions. This data-driven process will be more reliable and will help in defining business strategies.

Index Terms -

I. Introduction to dataset

The "HR EMPLOYEE ATTRITION DATASET" consists of the details of an employee like gender, age, business travel, department, education, relationship satisfaction, and many others. Basically, the dataset consists of exactly 2940 employees' data, and employee has 34 features. The dataset consists of both numerical and categorical data.

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

II. Case Study

Title: Analysis of the given dataset and to predict the attrition of the employee from the company.

Objective: To analyze the reason/causes of employee attrition.

Tools used: Jupyter Notebook, python Outcome: Students are able to:

1. Import the dataset and perform preprocessing on it making it suitable for model building.

2. Perform Exploratory Data Analysis and achieve insights from the visualization.

Theory:

Python: Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other

languages use punctuation, and it has fewer syntactical constructions than other languages.

Pandas: Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named NumPy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution.

Matplotlib: Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Seaborn: Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

Techniques used:

Data Preprocessing: Data preprocessing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analysed by computers and machine learning. Raw, real-world data in the form of text, images, video, etc., is messy. Not only may it contain errors and inconsistencies, but it is often incomplete, and doesn't have a regular, uniform design. Machines like to process nice and tidy information – they read data as 1s and 0s. So, calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis.

Data Visualization: Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main goal of data visualization is to make it easier to identify patterns, trends, and outliers in large data sets. The term is often used interchangeably with others, including information graphics, information visualization and statistical graphics. Visualization is central to advanced analytics for similar reasons. When a data scientist is writing advanced predictive analytics or machine learning (ML) algorithms, it becomes important to visualize the outputs to monitor results and ensure that models are performing as intended. This is because visualizations of complex algorithms are generally easier to interpret than numerical outputs.

Model Building:

Random Forest Classifier Building an ML Model requires splitting of data into two sets, such as 'training set' and 'testing set' in the ratio of 80:20 or 70:30; A set of supervised (for labelled data) and unsupervised (for unlabelled data) algorithms are available to choose from depending on the nature of input data and business outcome to predict. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

The Random Forest Classifier

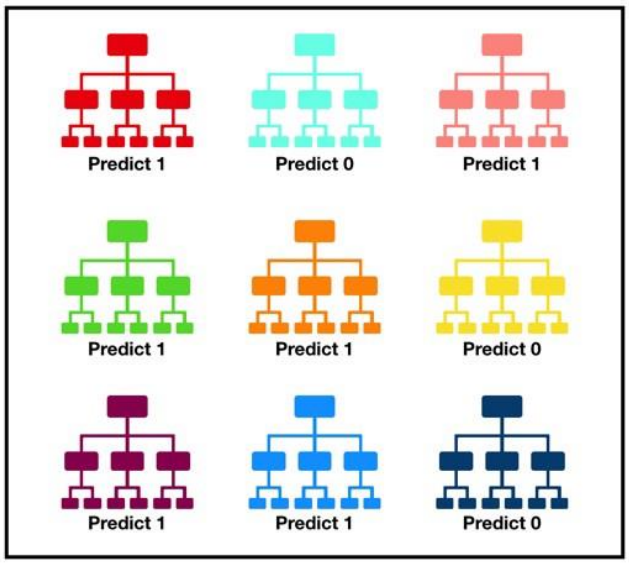
Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don' t constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are:

There needs to be some actual signal in our features so that models built using those features do better than random guessing.
The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.



Tally: Six 1s and Three 0s
Prediction: 1

Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022

NOTEBOOK SCREENSHOTS:

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

In [3]:
```
data=pd.read_csv("HR_Employee_Attrition-1.csv")
```

In [4]:
```
data.head(10)
```

Out[4]:

	EmployeeNumber	Attrition	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	...	Relations
0	1	Yes	41	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	...	
1	2	No	49	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	...	
2	3	Yes	37	Travel_Rarely	1373	Research & Development	2	2	Other	1	...	
3	4	No	33	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	...	
4	5	No	27	Travel_Rarely	591	Research & Development	2	1	Medical	1	...	
5	6	No	32	Travel_Frequently	1005	Research & Development	2	2	Life Sciences	1	...	
6	7	No	59	Travel_Rarely	1324	Research &	3	3	Medical	1		

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [5]:
```
data.describe()
```

Out[5]:

	EmployeeNumber	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EnvironmentSatisfaction	HourlyRate	JobInvolvemen
count	2940.000000	2940.000000	2940.000000	2940.000000	2940.000000	2940.0	2940.000000	2940.000000	2940.00000
mean	1470.500000	36.923810	802.485714	9.192517	2.912925	1.0	2.721769	65.891156	2.72993
std	848.849221	9.133819	403.440447	8.105485	1.023991	0.0	1.092896	20.325969	0.71144
min	1.000000	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	30.000000	1.00000
25%	735.750000	30.000000	465.000000	2.000000	2.000000	1.0	2.000000	48.000000	2.00000
50%	1470.500000	36.000000	802.000000	7.000000	3.000000	1.0	3.000000	66.000000	3.00000
75%	2205.250000	43.000000	1157.000000	14.000000	4.000000	1.0	4.000000	84.000000	3.00000
max	2940.000000	60.000000	1499.000000	29.000000	5.000000	1.0	4.000000	100.000000	4.00000

8 rows × 26 columns

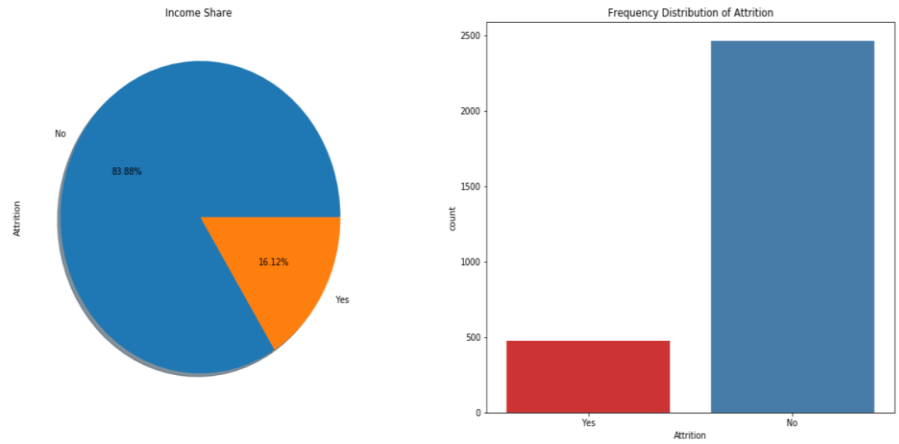
In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2940 entries, 0 to 2939
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   EmployeeNumber            2940 non-null   int64
 1   Attrition                 2940 non-null   object
 2   Age                       2940 non-null   int64
 3   BusinessTravel            2940 non-null   object
 4   DailyRate                 2940 non-null   int64
 5   Department                2940 non-null   object
```

Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022

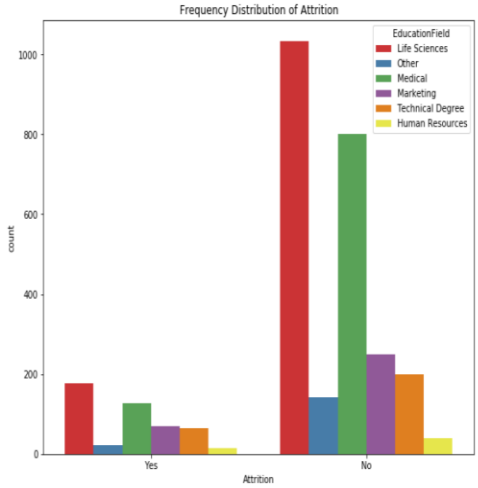
jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

In [16]:
```
f, ax = plt.subplots(1, 2, figsize=(20,8))
ax[0] = df['Attrition'].value_counts(). plot.pie(explode=[0,0], autopct = '%1.2f%%', ax=ax[0], shadow=True)
ax[0].set_title('Income Share ')
ax[1] = sns.countplot(x='Attrition', data=data, palette='Set1')
ax[1].set_title("Frequency Distribution of Attrition")
plt.show()
```



jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

In [17]:
```
f, ax = plt.subplots(figsize=(10,8))
ax = sns.countplot(x='Attrition', hue= 'EducationField', data=df, palette='Set1' )
ax.set_title("Frequency Distribution of Attrition")
plt.show()
```



Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [18]:
```
x1=df.drop(["Attrition"],axis="columns",inplace=False)
y1=df["Attrition"]
x1.drop(["EmployeeNumber","EmployeeCount","Over18"],axis=1,inplace=True)
x1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2940 entries, 0 to 2939
Data columns (total 31 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       2940 non-null   int64
 1   BusinessTravel            2940 non-null   object
 2   DailyRate                 2940 non-null   int64
 3   Department                2940 non-null   object
 4   DistanceFromHome          2940 non-null   int64
 5   Education                 2940 non-null   int64
 6   EducationField            2940 non-null   object
 7   EnvironmentSatisfaction   2940 non-null   int64
 8   Gender                    2940 non-null   object
 9   HourlyRate                2940 non-null   int64
 10  JobInvolvement            2940 non-null   int64
 11  JobLevel                  2940 non-null   int64
 12  JobRole                   2940 non-null   object
 13  JobSatisfaction           2940 non-null   int64
 14  MaritalStatus             2940 non-null   object
 15  MonthlyIncome             2940 non-null   int64
 16  MonthlyRate               2940 non-null   int64
 17  NumCompaniesWorked        2940 non-null   int64
 18  OverTime                  2940 non-null   object
 19  PercentSalaryHike         2940 non-null   int64
 20  PerformanceRating         2940 non-null   int64
 21  RelationshipSatisfaction  2940 non-null   int64
 22  StandardHours             2940 non-null   int64
 23  StockOptionLevel          2940 non-null   int64
 24  TotalWorkingYears         2940 non-null   int64
 25  TrainingTimesLastYear     2940 non-null   int64
 26  WorkLifeBalance           2940 non-null   int64
```

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [19]:
```
df["Gender"].value_counts()
```

Out[19]:
```
Male      1764
Female    1176
Name: Gender, dtype: int64
```

In [20]:
```
df.corr()
```

Out[20]:

	EmployeeNumber	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EnvironmentSatisfaction	HourlyRate	Job
EmployeeNumber	1.000000	-0.005175	-0.025742	0.016464	0.020950	NaN	0.008712	0.017377	
Age	-0.005175	1.000000	0.010661	-0.001686	0.208034	NaN	0.010146	0.024287	
DailyRate	-0.025742	0.010661	1.000000	-0.004985	-0.016806	NaN	0.018355	0.023381	
DistanceFromHome	0.016464	-0.001686	-0.004985	1.000000	0.021042	NaN	-0.016075	0.031131	
Education	0.020950	0.208034	-0.016806	0.021042	1.000000	NaN	-0.027128	0.016775	
EmployeeCount	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
EnvironmentSatisfaction	0.008712	0.010146	0.018355	-0.016075	-0.027128	NaN	1.000000	-0.049857	
HourlyRate	0.017377	0.024287	0.023381	0.031131	0.016775	NaN	-0.049857	1.000000	
JobInvolvement	-0.003552	0.029820	0.046135	0.008783	0.042438	NaN	-0.008278	0.042861	
JobLevel	-0.009020	0.509604	0.002966	0.005303	0.101589	NaN	0.001212	-0.027853	
JobSatisfaction	-0.022970	-0.004892	0.030571	-0.003669	-0.011296	NaN	-0.006784	-0.071335	
MonthlyIncome	-0.007188	0.497855	0.007707	-0.017014	0.094961	NaN	-0.006259	-0.015794	
MonthlyRate	0.006177	0.028051	-0.032182	0.027473	-0.026084	NaN	0.037600	-0.015297	
NumCompaniesWorked	-0.000345	0.299635	0.038153	-0.029251	0.126317	NaN	0.012594	0.022157	
PercentSalaryHike	-0.006685	0.003634	0.022704	0.040235	-0.011111	NaN	-0.031701	-0.009062	
PerformanceRating	-0.010338	0.001904	0.000473	0.027110	-0.024539	NaN	-0.029548	-0.002172	
RelationshipSatisfaction	-0.034827	0.053535	0.007846	0.006557	-0.009118	NaN	0.007665	0.001330	
StandardHours	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [21]:
```
plt.figure(figsize=(15,15))
sns.heatmap(df.corr(),annot=True,cmap='cividis_r',fmt='.0%')
```

Out[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e26668aa88>
```

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [22]:
```
def initial_eda(dataset):
  if isinstance(dataset, pd.DataFrame):
    total_na = dataset.isna().sum().sum()
    print("Total Records", dataset.shape)
    print("Total NA Records", total_na)
    cols = dataset.columns
    dtype = dataset.dtypes
    duniq = dataset.nunique()
    na_val = dataset.isna().sum()
    print("Cols dataset", "Datatype", "unique_records", "null records")
    for i in range(len(dataset.columns)):
      print("%38s %10s %10s %10s"% (cols[i], dtype[i], duniq[i], na_val[i]))
  else:
    print('error in the code ')
```

In [23]:
```
initial_eda(df)
```

```
Total Records (2940, 35)
Total NA Records 0
Cols dataset Datatype unique_records null records
                EmployeeNumber      int64        2940          0
                     Attrition     object           2          0
                           Age      int64          43          0
              BusinessTravel     object           3          0
                     DailyRate      int64         886          0
                    Department     object           3          0
              DistanceFromHome      int64          29          0
                     Education      int64           5          0
               EducationField     object           6          0
                EmployeeCount      int64           1          0
EnvironmentSatisfaction      int64           4          0
                        Gender     object           2          0
                    HourlyRate      int64          71          0
              JobInvolvement      int64           4          0
                    JobLevel      int64           5          0
                     JobRole     object           9          0
```

Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [25]:
```
import category_encoders as ce
encoder = ce.OrdinalEncoder(cols = ['BusinessTravel', 'Department','EducationField','Gender','JobRole', 'MaritalStatus', 'Ove
x1= encoder.fit_transform(x1)
```

In [26]:
```
x1.head(20)
```

Out[26]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	HourlyRate	...	Relations
0	41	1	1102	1	1	2	1	2	1	94	...	
1	49	2	279	2	8	1	1	3	2	61	...	
2	37	1	1373	2	2	2	2	4	2	92	...	
3	33	2	1392	2	3	4	1	4	1	56	...	
4	27	1	591	2	2	1	3	1	2	40	...	
5	32	2	1005	2	2	2	1	4	2	79	...	
6	59	1	1324	2	3	3	3	3	1	81	...	
7	30	1	1358	2	24	1	1	4	2	67	...	
8	38	2	216	2	23	3	1	4	2	44	...	
9	36	1	1299	2	27	3	3	3	2	94	...	
10	35	1	809	2	16	3	3	1	2	84	...	
11	29	1	153	2	15	2	1	4	1	49	...	
12	31	1	670	2	26	1	1	1	2	31	...	
13	34	1	1346	2	19	2	3	2	2	93	...	
14	28	1	103	2	24	3	1	3	2	50	...	
15	29	1	1389	2	21	4	1	2	1	51	...	
16	32	1	334	2	5	2	1	1	2	80	...	
17	22	3	1123	2	16	2	3	4	2	96	...	
18	53	1	1219	1	2	4	1	1	1	78	...	

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [27]:
```
x1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2940 entries, 0 to 2939
Data columns (total 31 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       2940 non-null   int64
 1   BusinessTravel            2940 non-null   int32
 2   DailyRate                 2940 non-null   int64
 3   Department                2940 non-null   int32
 4   DistanceFromHome          2940 non-null   int64
 5   Education                 2940 non-null   int64
 6   EducationField            2940 non-null   int32
 7   EnvironmentSatisfaction   2940 non-null   int64
 8   Gender                    2940 non-null   int32
 9   HourlyRate                2940 non-null   int64
 10  JobInvolvement            2940 non-null   int64
 11  JobLevel                  2940 non-null   int64
 12  JobRole                   2940 non-null   int32
 13  JobSatisfaction           2940 non-null   int64
 14  MaritalStatus             2940 non-null   int32
 15  MonthlyIncome             2940 non-null   int64
 16  MonthlyRate               2940 non-null   int64
 17  NumCompaniesWorked        2940 non-null   int64
 18  OverTime                  2940 non-null   int32
 19  PercentSalaryHike         2940 non-null   int64
 20  PerformanceRating         2940 non-null   int64
 21  RelationshipSatisfaction  2940 non-null   int64
 22  StandardHours             2940 non-null   int64
 23  StockOptionLevel          2940 non-null   int64
 24  TotalWorkingYears         2940 non-null   int64
 25  TrainingTimesLastYear     2940 non-null   int64
 26  WorkLifeBalance           2940 non-null   int64
 27  YearsAtCompany            2940 non-null   int64
 28  YearsInCurrentRole        2940 non-null   int64
 29  YearsSinceLastPromotion   2940 non-null   int64
```

Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

In [28]:
```
cols=x1.columns
from sklearn.preprocessing import RobustScaler
scaler = RobustScaler()
x1= scaler.fit_transform(x1)
x1=pd.DataFrame(x1,columns=[cols])
```

In [29]:
```
x1.head(20)
```

Out[29]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	HourlyRate	...	Rel
0	0.384615	0.0	0.433526	-1.0	-0.500000	-0.5	-1.0	-0.5	-1.0	0.777778	...	
1	1.000000	1.0	-0.755780	0.0	0.083333	-1.0	-1.0	0.0	0.0	-0.138889	...	
2	0.076923	0.0	0.825145	0.0	-0.416667	-0.5	-0.5	0.5	0.0	0.722222	...	
3	-0.230769	1.0	0.852601	0.0	-0.333333	0.5	-1.0	0.5	-1.0	-0.277778	...	
4	-0.692308	0.0	-0.304913	0.0	-0.416667	-1.0	0.0	-1.0	0.0	-0.722222	...	
5	-0.307692	1.0	0.293353	0.0	-0.416667	-0.5	-1.0	0.5	0.0	0.361111	...	
6	1.769231	0.0	0.754335	0.0	-0.333333	0.0	0.0	0.0	-1.0	0.416667	...	
7	-0.461538	0.0	0.803468	0.0	1.416667	-1.0	-1.0	0.5	0.0	0.027778	...	
8	0.153846	1.0	-0.846821	0.0	1.333333	0.0	-1.0	0.5	0.0	-0.611111	...	
9	0.000000	0.0	0.718208	0.0	1.666667	0.0	0.0	0.0	0.0	0.777778	...	
10	-0.076923	0.0	0.010116	0.0	0.750000	0.0	0.0	-1.0	0.0	0.500000	...	
11	-0.538462	0.0	-0.937861	0.0	0.666667	-0.5	-1.0	0.5	-1.0	-0.472222	...	
12	-0.384615	0.0	-0.190751	0.0	1.583333	-1.0	-1.0	-1.0	0.0	-0.972222	...	
13	-0.153846	0.0	0.786127	0.0	1.000000	-0.5	0.0	-0.5	0.0	0.750000	...	
14	-0.615385	0.0	-1.010116	0.0	1.416667	0.0	-1.0	0.0	0.0	-0.444444	...	
15	-0.538462	0.0	0.848266	0.0	1.166667	0.5	-1.0	-0.5	-1.0	-0.416667	...	
16	-0.307692	0.0	-0.676301	0.0	-0.166667	-0.5	-1.0	-1.0	0.0	0.388889	...	
17	-1.076923	2.0	0.463873	0.0	0.750000	-0.5	0.0	0.5	0.0	0.833333	...	
18	1.307692	0.0	0.602601	-1.0	-0.416667	0.5	-1.0	-1.0	-1.0	0.333333		

jupyter TCR INNOVATION PROJECT HR ATTRITION Last Checkpoint: 8 hours ago (autosaved)

In [30]:
```
x_train,x_test,y_train,y_test=train_test_split(x1,y1,test_size=0.3,random_state=42)
```

In [31]:
```
print(x_train.shape," ",x_test.shape," ")
```

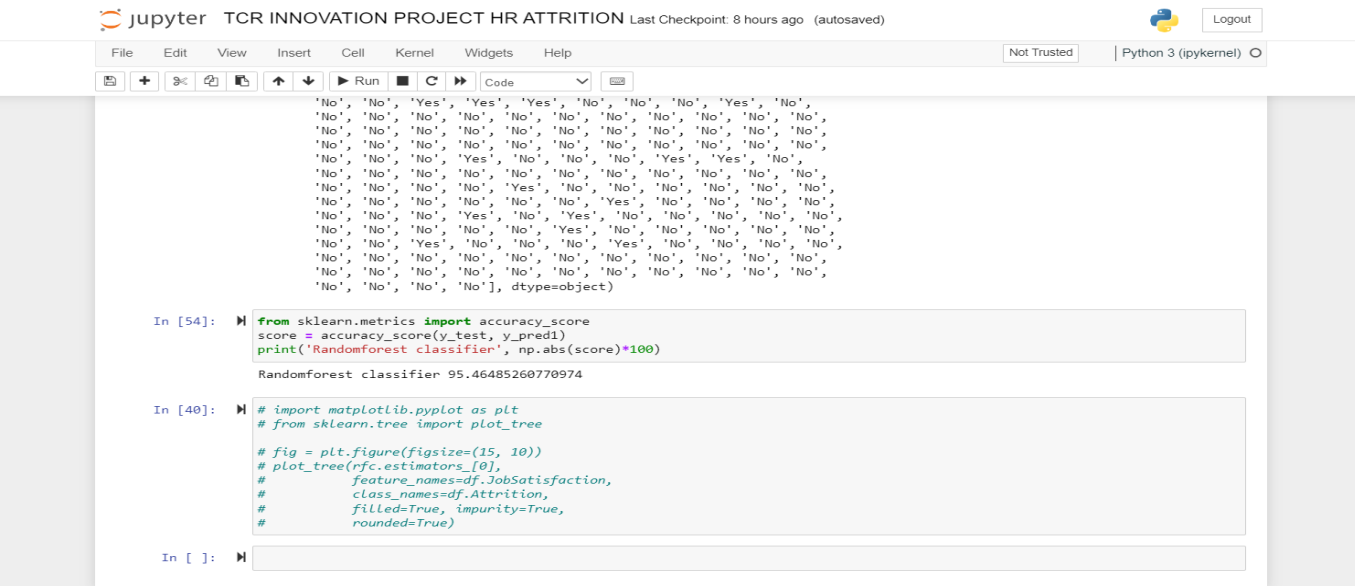
```
(2058, 31)   (882, 31)
```

In [51]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=100, random_state=0)
rfc.fit(x_train, y_train)
y_pred1= rfc.predict(x_test)
```

In [52]:
```
y_pred1
```

Out[52]:
```
array(['No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
       'Yes', 'No', 'Yes', 'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
```

Name- M. Sai Harsha Vardhan
Internship Program-DATA SCIENCE WITH MACHINE LEARNING AND PYTHON
Batch- May 2022- June 2022
Certificate Code- TCRIB3R89
Date of submission-02-07-2022



CONCLUSION:

So, In this project Employee Attrition data was analysed and various insights were given about the reason the employees are leaving the company along with a Random forest classifier model with testing accuracy of 95.46% making it a Best fit.

Randomforest classifier 95.46485260770974

ACKNOWLDGEMENT:

We are pleased to submit this Internship Report as an Intern for 2 months at TCR Innovation. We wish to thank the whole team for providing this great internship opportunity. We would like to thank our trainees for their guidance in the whole program.

REFERENCES:

[Understanding Random Forest. How the Algorithm Works and Why it Is… | by Tony Yiu | Towards Data Science]