A Comparison of Naïve Bayes (NB) and Random Forest (RF) on Predicting Employee Departures - Data Exploration

IDM431: Machine Learning – Edward St John

In [2]:

```
# Importing required libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sb
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
```

In [3]:

```
#Loading data
data = pd.read_csv('Employee.csv')
data_num = pd.read_csv('Employee.csv')
```

In [4]:

```
data.head()
```

Out[4]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	Bachelors	2017	Bangalore	3	34	Male	No	0	0
1	Bachelors	2013	Pune	1	28	Female	No	3	1
2	Bachelors	2014	New Delhi	3	38	Female	No	2	0
3	Masters	2016	Bangalore	3	27	Male	No	5	1
4	Masters	2017	Pune	3	24	Male	Yes	2	1

Changing categorical variables to numerical

In [6]:

```
# Renaming Experience column for better readability
data_num.rename(columns={'ExperienceInCurrentDomain':'Experience'}, inplace=True)
data_num.head()
```

Out[6]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	Experience	LeaveOrNot
0	Bachelors	2017	Bangalore	3	34	Male	No	0	0
1	Bachelors	2013	Pune	1	28	Female	No	3	1
2	Bachelors	2014	New Delhi	3	38	Female	No	2	0
3	Masters	2016	Bangalore	3	27	Male	No	5	1
4	Masters	2017	Pune	3	24	Male	Yes	2	1

In [7]:

```
# Converting Education category to numerical
le = LabelEncoder()
Education = le.fit_transform(data['Education'])
print(Education)
```

```
[0 0 0 ... 1 0 0]
```

In [8]:

```
# Same as above for Gender and EverBenched
Gender = le.fit_transform(data['Gender'])
EverBenched = le.fit_transform(data['EverBenched'])
```

In [9]:

```
# Changing dataset to these new numerical variables
data_num['Education'] = Education
data_num['Gender'] = Gender
data_num['EverBenched'] = EverBenched
data_num.head()
```

Out[9]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	Experience	LeaveOrNot
0	0	2017	Bangalore	3	34	1	0	0	0

1	0	2013	Pune	1	28	0	0	3	1
2	0	2014	New Delhi	3	38	0	0	2	0
3	1	2016	Bangalore	3	27	1	0	5	1
4	1	2017	Pune	3	24	1	1	2	1

In [10]:

```
# Creating new numerical city columns, splitting up by city
data_num['Bangalore'] = np.where(data_num['City'] == 'Bangalore', 1, 0)
data_num['Pune'] = np.where(data_num['City'] == 'Pune', 1, 0)
data_num['New Delhi'] = np.where(data_num['City'] == 'New Delhi', 1, 0)
data_num = data_num[['Education', 'JoiningYear', 'Bangalore', 'Pune', 'New Delhi', 'PaymentTier', 'Age', 'Gender'
                     'Experience', 'LeaveOrNot']]
data_num.head()
```

Out[10]:

	Education	JoiningYear	Bangalore	Pune	New Delhi	PaymentTier	Age	Gender	EverBenched	Experience	LeaveOrNot
0	0	2017	1	0	0	3	34	1	0	0	0
1	0	2013	0	1	0	1	28	0	0	3	1
2	0	2014	0	0	1	3	38	0	0	2	0
3	1	2016	1	0	0	3	27	1	0	5	1
4	1	2017	0	1	0	3	24	1	1	2	1

In [11]:

```
#Extracting new numerical data to csv
data_num.to_csv(r'/Users/edward/Documents/City/Machine Learning/Employee_num.csv')
```

Producing correlation heatmap

In [12]:

```
print(data_num.corr())
```

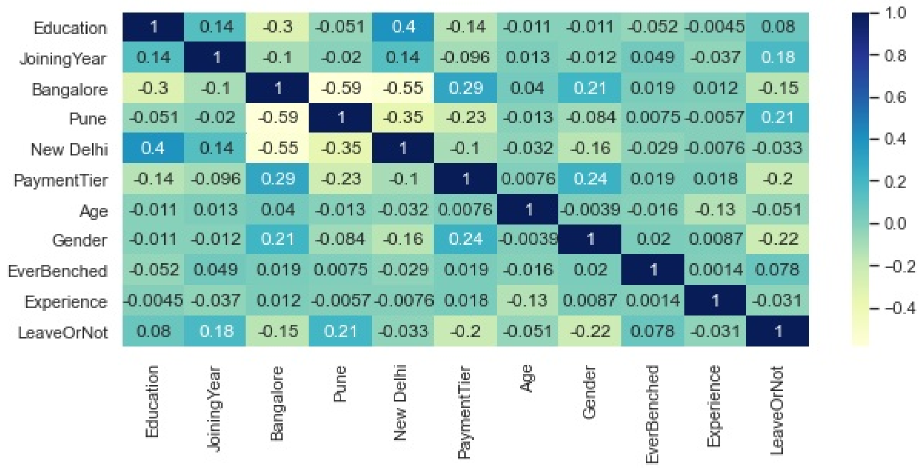
```
             Education  JoiningYear  Bangalore      Pune  New Delhi  \
Education     1.000000     0.142670  -0.298423 -0.051377   0.397825
JoiningYear   0.142670     1.000000  -0.104668 -0.020167   0.141744
Bangalore    -0.298423    -0.104668   1.000000 -0.586654  -0.551420
Pune         -0.051377    -0.020167  -0.586654  1.000000  -0.352096
New Delhi     0.397825     0.141744  -0.551420 -0.352096   1.000000
PaymentTier  -0.140741    -0.096078   0.293730 -0.229910  -0.102642
Age          -0.010611     0.013165   0.039918 -0.013273  -0.032461
Gender       -0.010889    -0.012213   0.209460 -0.083685  -0.155877
EverBenched  -0.052249     0.049353   0.018590  0.007534  -0.029246
Experience   -0.004463    -0.036525   0.011654 -0.005690  -0.007608
LeaveOrNot    0.080497     0.181705  -0.154996  0.206264  -0.033341

             PaymentTier       Age    Gender  EverBenched  Experience  \
Education      -0.140741 -0.010611 -0.010889    -0.052249   -0.004463
JoiningYear    -0.096078  0.013165 -0.012213     0.049353   -0.036525
Bangalore       0.293730  0.039918  0.209460     0.018590    0.011654
Pune           -0.229910 -0.013273 -0.083685     0.007534   -0.005690
New Delhi      -0.102642 -0.032461 -0.155877    -0.029246   -0.007608
PaymentTier     1.000000  0.007631  0.235119     0.019207    0.018314
Age             0.007631  1.000000 -0.003866    -0.016135   -0.134643
Gender          0.235119 -0.003866  1.000000     0.019653    0.008745
EverBenched     0.019207 -0.016135  0.019653     1.000000    0.001408
Experience      0.018314 -0.134643  0.008745     0.001408    1.000000
LeaveOrNot     -0.197638 -0.051126 -0.220701     0.078438   -0.030504

             LeaveOrNot
Education      0.080497
JoiningYear    0.181705
Bangalore     -0.154996
Pune           0.206264
New Delhi     -0.033341
PaymentTier   -0.197638
Age           -0.051126
Gender        -0.220701
EverBenched    0.078438
Experience    -0.030504
LeaveOrNot     1.000000
```

In [18]:

```
# Creating correlation matrix for all variables having coverted to numerical
corr_heatmap = sb.heatmap(data_num.corr(), cmap="YlGnBu", annot=True)
sb.set(rc={'figure.figsize':(10,4)})
```



In [16]:

```
#Finding statistics on all variables
data_num_statistics = data_num.describe()
pd.set_option('display.float_format', '{:.2f}'.format)
data_num_statistics
```

Out[16]:

	Education	JoiningYear	Bangalore	Pune	New Delhi	PaymentTier	Age	Gender	EverBenched	Experience	LeaveOrNot
count	4653.00	4653.00	4653.00	4653.00	4653.00	4653.00	4653.00	4653.00	4653.00	4653.00	4653.00
mean	0.26	2015.06	0.48	0.27	0.25	2.70	29.39	0.60	0.10	2.91	0.34
std	0.52	1.86	0.50	0.45	0.43	0.56	4.83	0.49	0.30	1.56	0.48
min	0.00	2012.00	0.00	0.00	0.00	1.00	22.00	0.00	0.00	0.00	0.00
25%	0.00	2013.00	0.00	0.00	0.00	3.00	26.00	0.00	0.00	2.00	0.00
50%	0.00	2015.00	0.00	0.00	0.00	3.00	28.00	1.00	0.00	3.00	0.00
75%	0.00	2017.00	1.00	1.00	0.00	3.00	32.00	1.00	0.00	4.00	1.00
max	2.00	2018.00	1.00	1.00	1.00	3.00	41.00	1.00	1.00	7.00	1.00

In [19]:

```
data_describe = data.describe()
data_describe
```

Out[19]:

	JoiningYear	PaymentTier	Age	ExperienceInCurrentDomain	LeaveOrNot
count	4653.00	4653.00	4653.00	4653.00	4653.00
mean	2015.06	2.70	29.39	2.91	0.34
std	1.86	0.56	4.83	1.56	0.48
min	2012.00	1.00	22.00	0.00	0.00
25%	2013.00	3.00	26.00	2.00	0.00
50%	2015.00	3.00	28.00	3.00	0.00
75%	2017.00	3.00	32.00	4.00	1.00
max	2018.00	3.00	41.00	7.00	1.00

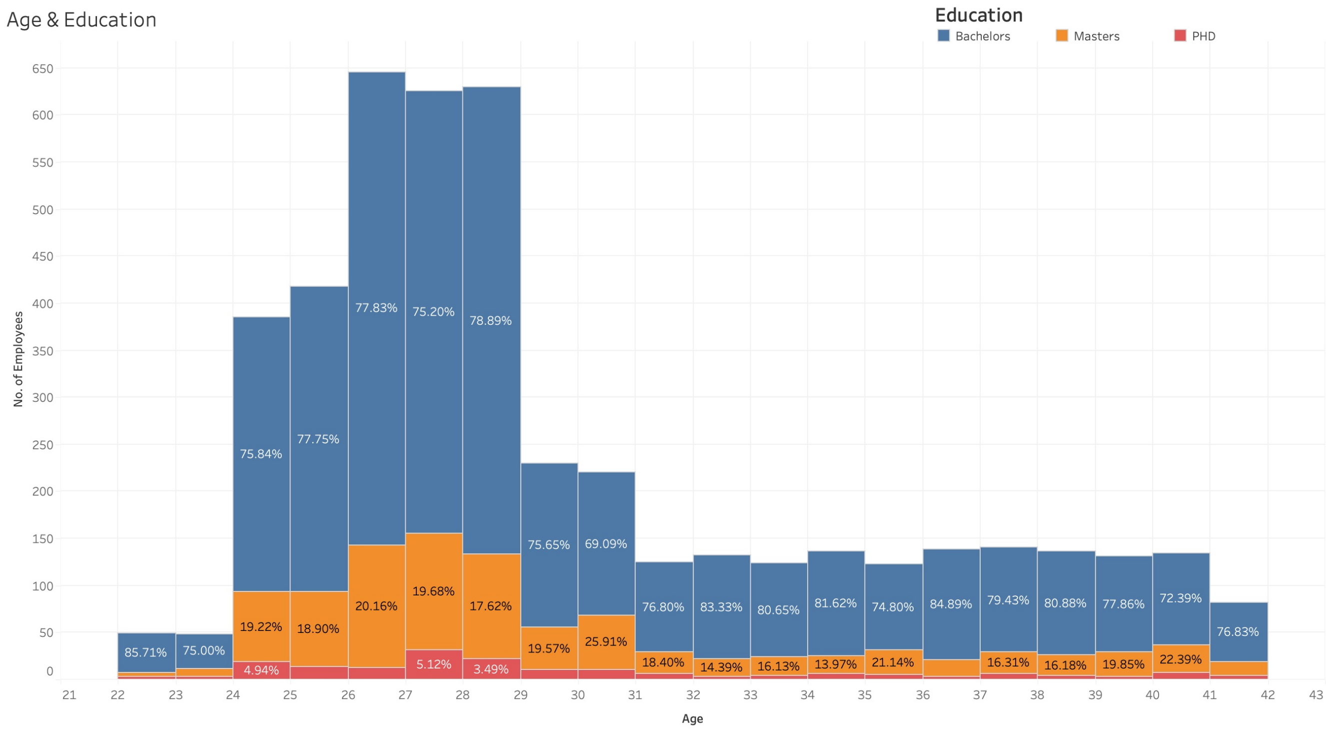
In [20]:

```
#Extracting describe data to csv
data_describe.to_csv(r'/Users/edward/Documents/City/Machine Learning/data_describe.csv')
```

Following visualisations from Tableau

Breaking down Age & Education to see distribution:

Age & Education.png



Breaking down City & Gender to see distribution:

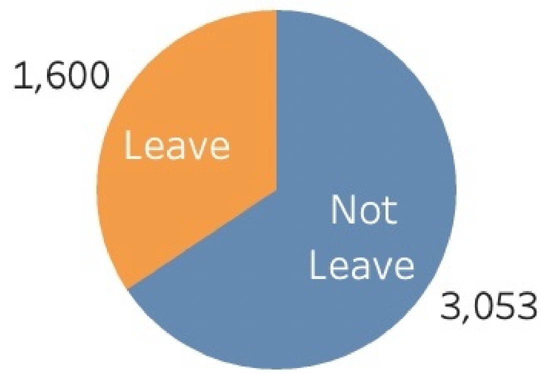
City & Gender

City & Gender

Gender	City: Bangalore	New Delhi	Pune
Male	1,569	537	672
Female	659	620	596

Employees leaving distribution:

Amount of Employees Leaving



Employees Leaving

Employees Leaving

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js