In [1]:

```
import pandas as pd
import numpy as np
import pickle
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
data = pd.read_csv('HR_Analytics.csv')
data
```

Out[2]:

	EmployeeID	BusinessTravel	Age	MonthlyIncome	Gender	Department	TotalMonthsOfExp	TotalOrgsWorked	MonthsInOrg	LastPayIncrementBand	AverageFeedback	LastPromotionYears	Attrition
0	1	Travel_Rarely	18	1420	Male	Research & Development	110	4	9	5	4	4	1
1	2	Travel_Rarely	18	1200	Female	Sales	103	3	51	1	4	2	0
2	3	Travel_Frequently	18	1878	Male	Sales	41	4	16	5	4	4	1
3	4	Non-Travel	18	1051	Male	Research & Development	32	4	17	5	2	3	0
4	5	Non-Travel	18	1904	Male	Research & Development	80	3	16	3	4	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	996	Travel_Rarely	40	5473	Male	Sales	44	1	9	3	1	2	0
996	997	Travel_Rarely	40	16437	Male	Human Resources	6	2	5	2	1	1	0
997	998	Non-Travel	40	4069	Male	Research & Development	29	3	1	3	4	2	0
998	999	Travel_Rarely	40	4639	Female	Sales	92	2	7	1	2	4	1
999	1000	Travel_Rarely	40	10435	Female	Research & Development	107	3	76	3	1	4	1

1000 rows × 13 columns

In [3]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   EmployeeID            1000 non-null   int64 
 1   BusinessTravel        1000 non-null   object
 2   Age                   1000 non-null   int64 
 3   MonthlyIncome         1000 non-null   int64 
 4   Gender                1000 non-null   object
 5   Department            1000 non-null   object
 6   TotalMonthsOfExp      1000 non-null   int64 
 7   TotalOrgsWorked       1000 non-null   int64 
 8   MonthsInOrg           1000 non-null   int64 
 9   LastPayIncrementBand  1000 non-null   int64 
 10  AverageFeedback       1000 non-null   int64 
 11  LastPromotionYears    1000 non-null   int64 
 12  Attrition             1000 non-null   int64 
dtypes: int64(10), object(3)
memory usage: 101.7+ KB
```

In [4]:

```
data.shape
```

Out[4]:

```
(1000, 13)
```

In [5]:

```
data.describe()
```

Out[5]:

	EmployeeID	Age	MonthlyIncome	TotalMonthsOfExp	TotalOrgsWorked	MonthsInOrg	LastPayIncrementBand	AverageFeedback	LastPromotionYears	Attrition
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1000.000000
mean	500.500000	31.732000	5090.381000	61.002000	3.056000	31.000000	3.063000	2.539000	2.45400	0.230000
std	288.819436	5.170968	3141.329204	34.818132	1.398148	26.783341	1.413162	1.143585	1.10413	0.421043
min	1.000000	18.000000	1009.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.00000	0.000000
25%	250.750000	28.000000	2719.500000	31.000000	2.000000	9.000000	2.000000	1.000000	1.00000	0.000000
50%	500.500000	32.000000	4306.000000	61.000000	3.000000	23.000000	3.000000	3.000000	2.00000	0.000000
75%	750.250000	36.000000	6347.000000	91.250000	4.000000	48.000000	4.000000	4.000000	3.00000	0.000000
max	1000.000000	40.000000	19626.000000	120.000000	5.000000	116.000000	5.000000	4.000000	4.00000	1.000000

In [6]:

```
data.isnull().sum()
```

Out[6]:

```
EmployeeID              0
BusinessTravel          0
Age                     0
MonthlyIncome           0
Gender                  0
Department              0
TotalMonthsOfExp        0
TotalOrgsWorked         0
MonthsInOrg             0
LastPayIncrementBand    0
AverageFeedback         0
LastPromotionYears      0
Attrition               0
dtype: int64
```

In [7]:

```
data['BusinessTravel'].unique()
```

Out[7]:

```
array(['Travel_Rarely', 'Travel_Frequently', 'Non-Travel', 'TravelRarely'],
      dtype=object)
```

In [8]:

```
data['Gender'].unique()
```

Out[8]:

```
array(['Male', 'Female'], dtype=object)
```

In [9]:

```
data['Department'].unique()
```

Out[9]:

```
array(['Research & Development', 'Sales', 'Human Resources'], dtype=object)
```

In [10]:

```
data1 = data.drop(['EmployeeID','BusinessTravel','AverageFeedback','Age'],axis = 1)
```

In [11]:

```
data1
```

Out[11]:

	MonthlyIncome	Gender	Department	TotalMonthsOfExp	TotalOrgsWorked	MonthsInOrg	LastPayIncrementBand	LastPromotionYears	Attrition
0	1420	Male	Research & Development	110	4	9	5	4	1
1	1200	Female	Sales	103	3	51	1	2	0
2	1878	Male	Sales	41	4	16	5	4	1
3	1051	Male	Research & Development	32	4	17	5	3	0
4	1904	Male	Research & Development	80	3	16	3	2	0
...	...	...	...	...	...	...	...	...	...
995	5473	Male	Sales	44	1	9	3	2	0
996	16437	Male	Human Resources	6	2	5	2	1	0
997	4069	Male	Research & Development	29	3	1	3	2	0
998	4639	Female	Sales	92	2	7	1	4	1
999	10435	Female	Research & Development	107	3	76	3	4	1

1000 rows × 9 columns

In [12]:

```
data1.groupby('Attrition').count()
```

Out[12]:

	MonthlyIncome	Gender	Department	TotalMonthsOfExp	TotalOrgsWorked	MonthsInOrg	LastPayIncrementBand	LastPromotionYears
Attrition								
0	770	770	770	770	770	770	770	770
1	230	230	230	230	230	230	230	230

In [13]:

```
data1.replace({'Gender':{'Male':1,'Female':0}},inplace=True)
data1
```

Out[13]:

	MonthlyIncome	Gender	Department	TotalMonthsOfExp	TotalOrgsWorked	MonthsInOrg	LastPayIncrementBand	LastPromotionYears	Attrition
0	1420	1	Research & Development	110	4	9	5	4	1
1	1200	0	Sales	103	3	51	1	2	0
2	1878	1	Sales	41	4	16	5	4	1
3	1051	1	Research & Development	32	4	17	5	3	0
4	1904	1	Research & Development	80	3	16	3	2	0
...	...	...	...	...	...	...	...	...	...
995	5473	1	Sales	44	1	9	3	2	0
996	16437	1	Human Resources	6	2	5	2	1	0
997	4069	1	Research & Development	29	3	1	3	2	0
998	4639	0	Sales	92	2	7	1	4	1
999	10435	0	Research & Development	107	3	76	3	4	1

1000 rows × 9 columns

In [14]:

```
data1 = pd.get_dummies(data1,dtype=int)
```

In [15]:

```
data1
```

Out[15]:

	MonthlyIncome	Gender	TotalMonthsOfExp	TotalOrgsWorked	MonthsInOrg	LastPayIncrementBand	LastPromotionYears	Attrition	Department_Human Resources	Department_Research & Development	Department_Sales
0	1420	1	110	4	9	5	4	1	0	1	0
1	1200	0	103	3	51	1	2	0	0	0	1
2	1878	1	41	4	16	5	4	1	0	0	1
3	1051	1	32	4	17	5	3	0	0	1	0
4	1904	1	80	3	16	3	2	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
995	5473	1	44	1	9	3	2	0	0	0	1
996	16437	1	6	2	5	2	1	0	1	0	0
997	4069	1	29	3	1	3	2	0	0	1	0
998	4639	0	92	2	7	1	4	1	0	0	1
999	10435	0	107	3	76	3	4	1	0	1	0

1000 rows × 11 columns

In [16]:

```
y = data1['Attrition']
```

In [17]:

```
y
```

Out[17]:

```
0      1
1      0
2      1
3      0
4      0
      ..
995    0
996    0
997    0
998    1
999    1
Name: Attrition, Length: 1000, dtype: int64
```

In [18]:

```
x = data1.drop(['Attrition'],axis = 1)
```

In [19]:

```
x
```

Out[19]:

	MonthlyIncome	Gender	TotalMonthsOfExp	TotalOrgsWorked	MonthsInOrg	LastPayIncrementBand	LastPromotionYears	Department_Human Resources	Department_Research & Development	Department_Sales
0	1420	1	110	4	9	5	4	0	1	0
1	1200	0	103	3	51	1	2	0	0	1
2	1878	1	41	4	16	5	4	0	0	1
3	1051	1	32	4	17	5	3	0	1	0
4	1904	1	80	3	16	3	2	0	1	0
...	...	...	...	...	...	...	...	...	...	...
995	5473	1	44	1	9	3	2	0	0	1
996	16437	1	6	2	5	2	1	1	0	0
997	4069	1	29	3	1	3	2	0	1	0
998	4639	0	92	2	7	1	4	0	0	1
999	10435	0	107	3	76	3	4	0	1	0

1000 rows × 10 columns

In [20]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [21]:

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[21]:

```
▾ LogisticRegression
LogisticRegression()
```

In [22]:

```
ypred=classifier.predict(x_test)
ypred
```

Out[22]:

```
array([0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1,
       1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      dtype=int64)
```

In [23]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,ypred)
```

Out[23]:

```
array([[224,  30],
       [ 19,  57]], dtype=int64)
```

In [24]:

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,ypred)
```

Out[24]:

```
0.8515151515151516
```

In [25]:

```
res=pd.DataFrame(columns=['Attrition','predicted'])
res['Attrition']=y_test
res['predicted']=ypred
res=res.reset_index()
res['ID']=res.index
```

In [26]:

```
res.head()
```

Out[26]:

	index	Attrition	predicted	ID
0	521	0	0	0
1	737	0	0	1
2	740	0	0	2
3	660	1	1	3
4	411	1	0	4

In [27]:

```
new=[[1,22,175,80,25,75,20,10,25,43.5]]
```

In [28]:

```
real=classifier.predict(new)
real
```

Out[28]:

```
array([0], dtype=int64)
```

In [29]:

```
cor_mat=data1.corr()
cor_mat
```

Out[29]:

	MonthlyIncome	Gender	TotalMonthsOfExp	TotalOrgsWorked	MonthsInOrg	LastPayIncrementBand	LastPromotionYears	Attrition	Department_Human Resources	Department_Research & Development	Department_Sale
MonthlyIncome	1.000000	-0.034356	-0.030780	-0.009909	-0.026893	-0.061428	-0.070665	-0.061800	-0.065050	-0.109522	0.13995
Gender	-0.034356	1.000000	-0.038896	-0.044248	-0.012713	0.042922	0.027979	0.047257	0.033995	-0.001202	-0.01295
TotalMonthsOfExp	-0.030780	-0.038896	1.000000	0.053769	0.667504	-0.012453	0.007059	0.019702	-0.005949	-0.036719	0.04030
TotalOrgsWorked	-0.009909	-0.044248	0.053769	1.000000	0.027159	-0.003307	-0.041774	0.008706	0.010409	-0.024405	0.02079
MonthsInOrg	-0.026893	-0.012713	0.667504	0.027159	1.000000	-0.017984	0.009309	0.012605	-0.016980	-0.029331	0.03729
LastPayIncrementBand	-0.061428	0.042922	-0.012453	-0.003307	-0.017984	1.000000	0.094561	0.108528	0.005643	-0.019010	0.01722
LastPromotionYears	-0.070665	0.027979	0.007059	-0.041774	0.009309	0.094561	1.000000	0.765641	-0.003305	0.014814	-0.01387
Attrition	-0.061800	0.047257	0.019702	0.008706	0.012605	0.108528	0.765641	1.000000	-0.011906	-0.010150	0.01542
Department_Human Resources	-0.065050	0.033995	-0.005949	0.010409	-0.016980	0.005643	-0.003305	-0.011906	1.000000	-0.273330	-0.13597
Department_Research & Development	-0.109522	-0.001202	-0.036719	-0.024405	-0.029331	-0.019010	0.014814	-0.010150	-0.273330	1.000000	-0.91582
Department_Sales	0.139958	-0.012954	0.040302	0.020790	0.037298	0.017224	-0.013878	0.015424	-0.135977	-0.915820	1.00000

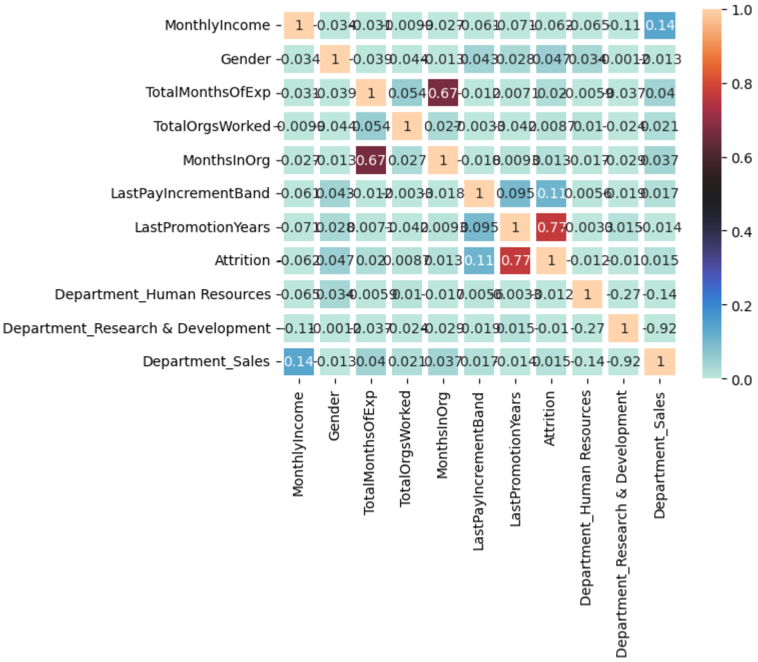
In [31]:

```
import seaborn as sns
```

```
sns.heatmap(cor_mat,vmax=1,vmin=0,annot=True,linewidth=3,cmap='icefire')
```

Out[31]:

```
<Axes: >
```



In [ ]: