Employee turnover Prediction using Random Forest

In [1]:
```
#import required libraries
import pandas as pd
import numpy as np

#reading the CSV file (HR_comma_sep.csv)
hr=pd.read_csv(r"C:\Users\Asus\Envs\Downloads\New folder\HR_comma_sep.csv")
```

In [2]:
```
hr
```

Out[2]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low
...	...	...	...	...	...	...	...	...	...	...
14994	0.40	0.57	2	151	3	0	1	0	support	low
14995	0.37	0.48	2	160	3	0	1	0	support	low
14996	0.37	0.53	2	143	3	0	1	0	support	low
14997	0.11	0.96	6	280	4	0	1	0	support	low
14998	0.37	0.52	2	158	3	0	1	0	support	low

14999 rows × 10 columns

In [3]:
```
hr.shape
```

Out[3]:
```
(14999, 10)
```

In [4]:
```
hr.columns
```

Out[4]:
```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'sales', 'salary'],
      dtype='object')
```

In [5]:
```
hr.head(20)
```

Out[5]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low
5	0.41	0.50	2	153	3	0	1	0	sales	low
6	0.10	0.77	6	247	4	0	1	0	sales	low
7	0.92	0.85	5	259	5	0	1	0	sales	low
8	0.89	1.00	5	224	5	0	1	0	sales	low
9	0.42	0.53	2	142	3	0	1	0	sales	low
10	0.45	0.54	2	135	3	0	1	0	sales	low
11	0.11	0.81	6	305	4	0	1	0	sales	low
12	0.84	0.92	4	234	5	0	1	0	sales	low
13	0.41	0.55	2	148	3	0	1	0	sales	low
14	0.36	0.56	2	137	3	0	1	0	sales	low
15	0.38	0.54	2	143	3	0	1	0	sales	low
16	0.45	0.47	2	160	3	0	1	0	sales	low
17	0.78	0.99	4	255	6	0	1	0	sales	low
18	0.45	0.51	2	160	3	1	1	1	sales	low
19	0.76	0.89	5	262	5	0	1	0	sales	low

In [6]:
```
# Here we are changing Column Name "sales" to "department" as it's creating confusion.
hr=hr.rename(columns={'sales':'department'})
```

hr

In [7]:
```
hr.dtypes
```

Out[7]:
```
satisfaction_level      float64
last_evaluation         float64
number_project            int64
average_montly_hours      int64
time_spend_company        int64
Work_accident             int64
left                      int64
promotion_last_5years     int64
department               object
salary                   object
dtype: object
```

In [8]:
```
#checking whether null values are there in dataframe or not.
hr.isnull().sum()
```

Out[8]:
```
satisfaction_level      0
last_evaluation         0
number_project          0
average_montly_hours    0
time_spend_company      0
Work_accident           0
left                    0
promotion_last_5years   0
department              0
salary                  0
dtype: int64
```

In [9]:
```
# Adding "technical","support" and "IT" in one column as these are very related to each other & can be taken collectively.

hr['department']=np.where(hr['department']=='support','technical',hr['department'])
hr['department']=np.where(hr['department']=='IT','technical',hr['department'])
```

In [10]:
```
hr
```

Out[10]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	department	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low
...	...	...	...	...	...	...	...	...	...	...
14994	0.40	0.57	2	151	3	0	1	0	technical	low
14995	0.37	0.48	2	160	3	0	1	0	technical	low
14996	0.37	0.53	2	143	3	0	1	0	technical	low
14997	0.11	0.96	6	280	4	0	1	0	technical	low
14998	0.37	0.52	2	158	3	0	1	0	technical	low

14999 rows × 10 columns

In [11]:
```
#one-hot encoding for categorical variables
cat_vars=['department','salary']
for var in cat_vars:
    cat_list='var'+'_'+var
    cat_list=pd.get_dummies(hr[var],prefix=var)
    hr1=hr.join(cat_list)
    hr=hr1
```

In [12]:
```
hr
```

Out[12]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	department	salary	...	department_accounting	department_hr	dep
0	0.38	0.53	2	157	3	0	1	0	sales	low	...	0	0	
1	0.80	0.86	5	262	6	0	1	0	sales	medium	...	0	0	
2	0.11	0.88	7	272	4	0	1	0	sales	medium	...	0	0	
3	0.72	0.87	5	223	5	0	1	0	sales	low	...	0	0	
4	0.37	0.52	2	159	3	0	1	0	sales	low	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
14994	0.40	0.57	2	151	3	0	1	0	technical	low	...	0	0	
14995	0.37	0.48	2	160	3	0	1	0	technical	low	...	0	0	
14996	0.37	0.53	2	143	3	0	1	0	technical	low	...	0	0	
14997	0.11	0.96	6	280	4	0	1	0	technical	low	...	0	0	
14998	0.37	0.52	2	158	3	0	1	0	technical	low	...	0	0	

14999 rows × 21 columns

In [13]:
```
hr.drop(hr.columns[[8,9]],axis=1,inplace=True)
hr
```

Out[13]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	department_RandD	department_accounting	department_hr	departme
0	0.38	0.53	2	157	3	0	1	0	0	0	0	
1	0.80	0.86	5	262	6	0	1	0	0	0	0	
2	0.11	0.88	7	272	4	0	1	0	0	0	0	
3	0.72	0.87	5	223	5	0	1	0	0	0	0	
4	0.37	0.52	2	159	3	0	1	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	
14994	0.40	0.57	2	151	3	0	1	0	0	0	0	
14995	0.37	0.48	2	160	3	0	1	0	0	0	0	
14996	0.37	0.53	2	143	3	0	1	0	0	0	0	
14997	0.11	0.96	6	280	4	0	1	0	0	0	0	
14998	0.37	0.52	2	158	3	0	1	0	0	0	0	

14999 rows × 19 columns

In [14]:
```
hr_vars=hr.columns.values.tolist()
hr_vars
```

Out[14]:
```
['satisfaction_level',
 'last_evaluation',
 'number_project',
 'average_montly_hours',
 'time_spend_company',
 'Work_accident',
 'left',
 'promotion_last_5years',
 'department_RandD',
 'department_accounting',
 'department_hr',
 'department_management',
 'department_marketing',
 'department_product_mng',
 'department_sales',
 'department_technical',
 'salary_high',
 'salary_low',
 'salary_medium']
```

Using Logistic Regression for training and testing data

taking the 10 most important features by seeing coefficients of model above

cols=['satisfaction_level', 'last_evaluation', 'time_spend_company', 'Work_accident', 'promotion_last_5years', 'department_RandD','department_hr', 'department_management', 'department_marketing', 'salary_high', 'salary_low'] X=hr[cols] y=hr['left']

In [15]:
```
#Segregating the input(X) and target(Y) from dataframe:
Y=hr['left']
X=hr.drop(['left'],axis=1)
X.columns
```

Out[15]:
```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident',
       'promotion_last_5years', 'department_RandD', 'department_accounting',
       'department_hr', 'department_management', 'department_marketing',
       'department_product_mng', 'department_sales', 'department_technical',
       'salary_high', 'salary_low', 'salary_medium'],
      dtype='object')
```

In [16]:
```
#importig RandomForest Classifier and splitting the data into training & test data:
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,random_state=100)
rf=RandomForestClassifier()
rf.fit(X_train,Y_train)
```

Out[16]:
```
RandomForestClassifier()
```

In [19]:
```
from sklearn.metrics import accuracy_score
print('Random Forest Accuracy :{:,.3f}'.format(accuracy_score(Y_test,rf.predict(X_test))))
```

```
Random Forest Accuracy :0.993
```

In [20]:
```
from sklearn.metrics import classification_report
print(classification_report(Y_test,rf.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      2830
           1       0.99      0.98      0.99       920

    accuracy                           0.99      3750
   macro avg       0.99      0.99      0.99      3750
weighted avg       0.99      0.99      0.99      3750
```

In [22]:
```
import seaborn as sns
import matplotlib.pyplot as plt
rf_y_pred=rf.predict(X_test)
rf_cm=metrics.confusion_matrix(rf_y_pred,Y_test,[1,0])
sns.heatmap(rf_cm,annot=True,fmt='.2f',xticklabels=['Left','Stayed'],yticklabels=['Left','Stayed'])
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.title('Random Forest')
```

```
C:\Users\Asus\anaconda3\lib\site-packages\sklearn\utils\validation.py:70: FutureWarning: Pass labels=[1, 0] as keyword args. From version 1.0 (renaming of 0.25) passing th
ese as positional arguments will result in an error
  warnings.warn(f"Pass {args_msg} as keyword args. From version "
Text(0.5, 1.0, 'Random Forest')
```

Out[22]: