HR Analytics Project- Understanding the Attrition in HR

Problem Statement:
Every year a lot of companies hire a number of employees. 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. The aim of these programs is to increase the effectiveness of their employees. But where HR Analytics fit in this? and is it just about improving the performance of employees?

HR Analytics
Human resource analytics (HR analytics) is an area in the field of analytics that refers to applying analytic processes to the human resource department of an organization in the hope of improving employee performance and therefore getting a better return on investment. HR analytics does not just deal with gathering data on employee efficiency. Instead, **it aims to provide insight into each process by gathering data and then using it to make relevant decisions about how to improve these processes.**

Attrition in HR
Attrition in human resources refers to the gradual loss of employees overtime. In general, relatively high attrition is problematic for companies. HR professionals often assume a leadership role in designing company compensation programs, work culture, and motivation systems that help the organization retain top employees.

How does Attrition affect companies? and how does HR Analytics help in analyzing attrition? We will discuss the first question here and for the second question, we will write the code and try to understand the process step by step.

Attrition affecting Companies
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. Additionally, regular employee turnover prohibits your organization from

increasing its collective knowledge base and experience over time. This is especially concerning if your business is customer-facing, as customers often prefer to interact with familiar people. Errors and issues are more likely if you constantly have new workers.

Introduction

In order to start with the exercise, I have used **Employee Attrition & Performance Dataset**. The dataset includes features like Age, Employee Role, Daily Rate, Job Satisfaction, Years At Company, Years In the Current Role, etc. For this exercise, we will try to study the factors that lead to employee attrition.

Let’s get started with my work.

Data Preparation: Load, Clean, and Format

```
df=pd.read_csv(r"C:\Users\Admin\Desktop\Dataset\EmployeeAttrition.csv")
df.head(10)
#top 10 data below
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipS
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	
5	32	No	Travel_Frequently	1005	Research & Development	2	2	Life Sciences	1	8	...	
6	59	No	Travel_Rarely	1324	Research & Development	3	3	Medical	1	10	...	
7	30	No	Travel_Rarely	1358	Research & Development	24	1	Life Sciences	1	11	...	
8	38	No	Travel_Frequently	216	Research & Development	23	3	Life Sciences	1	12	...	
9	36	No	Travel_Rarely	1299	Research & Development	27	3	Medical	1	13	...	

10 rows × 35 columns

#Explore the data types of each column

```
Out[24]: Age                       0
         Attrition                 0
         BusinessTravel            0
         DailyRate                 0
         Department                0
         DistanceFromHome          0
         Education                 0
         EducationField            0
         EmployeeCount             0
         EmployeeNumber            0
         EnvironmentSatisfaction   0
         Gender                    0
         HourlyRate                0
         JobInvolvement            0
         JobLevel                  0
         JobRole                   0
         JobSatisfaction           0
         MaritalStatus             0
         MonthlyIncome             0
         MonthlyRate               0
         NumCompaniesWorked        0
         Over18                    0
         OverTime                  0
         PercentSalaryHike         0
         PerformanceRating         0
         RelationshipSatisfaction  0
         StandardHours             0
         StockOptionLevel          0
         TotalWorkingYears         0
         TrainingTimesLastYear     0
         WorkLifeBalance           0
         YearsAtCompany            0
         YearsInCurrentRole        0
         YearsSinceLastPromotion   0
         YearsWithCurrManager      0
         dtype: int64
```

fortunately, we don’t have any missing values from the above in the HR dataset screenshot, it also looks like we don’t need to format any data.

Data Analysis

Let’s have a look at the dataset and see how features are contributing to the data and in the attrition of employees. We need to first look at the data type of the features, why do we need to do that? Because we can only see that there is only a distribution of numerical/continuous values in a dataset. In order to take a peek into categorical/object values, we have to bind them with a numeric variable and then you will be able to see their relevance to the dataset or we can replace the categorical variable with the dummies.

Df.shape

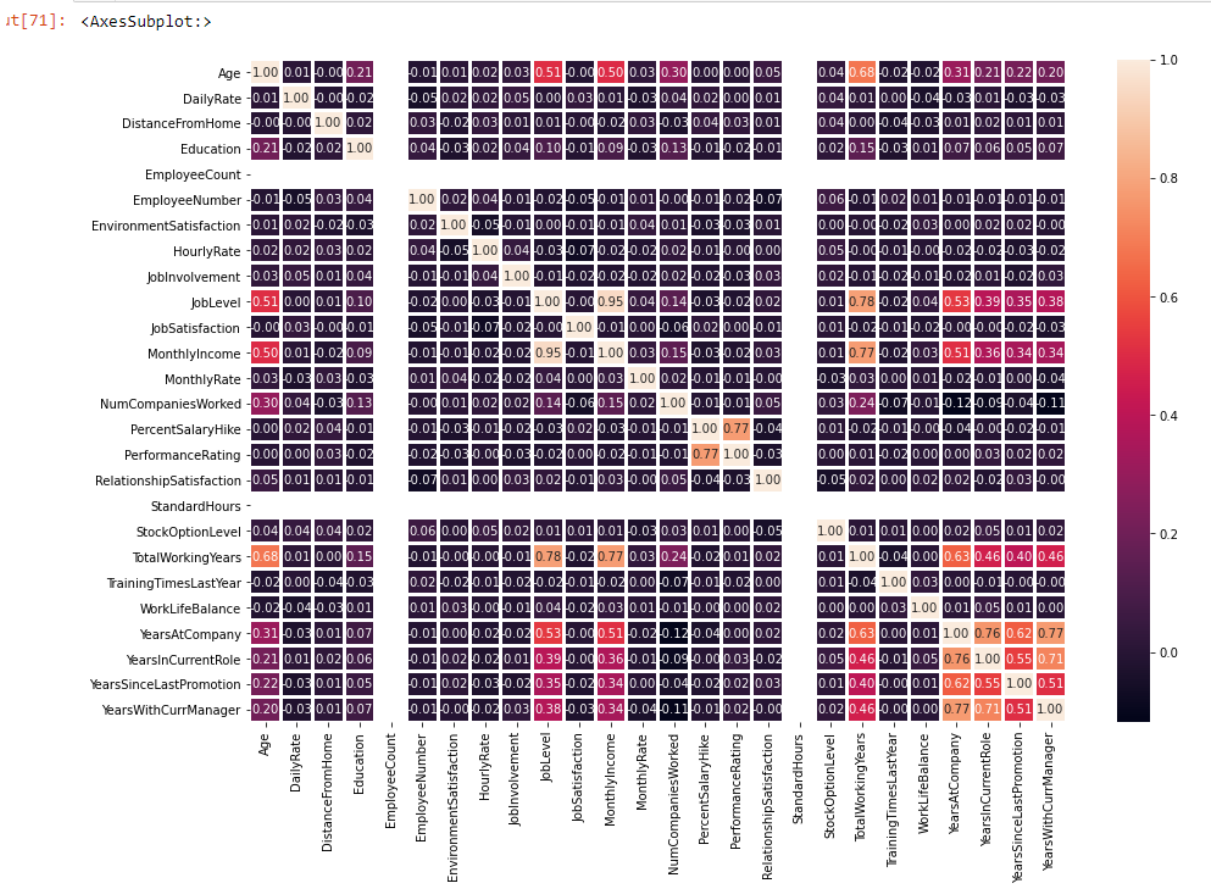
```
Out[13]: (1470, 35)
```

df.dtypes

```
Out[10]: Age                          int64
         Attrition                   object
         BusinessTravel              object
         DailyRate                    int64
         Department                  object
         DistanceFromHome             int64
         Education                    int64
         EducationField              object
         EmployeeCount                int64
         EmployeeNumber               int64
         EnvironmentSatisfaction      int64
         Gender                      object
         HourlyRate                   int64
         JobInvolvement               int64
         JobLevel                     int64
         JobRole                     object
         JobSatisfaction              int64
         MaritalStatus               object
         MonthlyIncome                int64
         MonthlyRate                  int64
         NumCompaniesWorked           int64
         Over18                      object
         OverTime                    object
         PercentSalaryHike            int64
         PerformanceRating            int64
         RelationshipSatisfaction     int64
         StandardHours                int64
         StockOptionLevel             int64
         TotalWorkingYears            int64
         TrainingTimesLastYear        int64
         WorkLifeBalance              int64
         YearsAtCompany               int64
         YearsInCurrentRole           int64
         YearsSinceLastPromotion      int64
         YearsWithCurrManager         int64
         dtype: object
```

For this, my aim is to predict the employee attrition dataset and it is important to see how and which variables are contributing the most to attrition. But before that we need to know if the variables are correlated if they are, we might want to avoid those in the model building process.

There are many continuous variables in the dataset, we can have a look at their distribution and create a grid of pair plots but that would be too much code to see the correlation as there are a lot of variables. Rather, we can create a seaborn heatmap of numeric variables and see the correlation below screenshort. For the variables which are not poorly correlated(ex. correlation value tends towards 0), we will pick those variables and move forward with them and will leave the ones which are strongly correlated(i.e correlation value tends towards be 1).



Let’s first replace “Yes” and “No” in Attrition data with 1 and 0.

```
72]:
df.drop(['EmployeeCount', 'EmployeeNumber', 'Over18', 'StandardHours','EmployeeNumber','Over18','StandardHours','EmployeeCou
```

```
73]:
df.Attrition.replace({'Yes': 1, 'No': 0}, inplace=True)

df.BusinessTravel.replace({'Non-Travel': 0, 'Travel_Rarely': 1, 'Travel_Frequently': 2}, inplace=True)

df.Department.replace({'Sales': 0, 'Research & Development': 1, 'Human Resources': 2}, inplace=True)

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

df.MaritalStatus.replace({'Single': 0,'Married': 1, 'Divorced': 2}, inplace=True)

df.OverTime.replace({'No': 0, 'Yes': 1}, inplace=True)

df.EducationField.replace({'Life Sciences': 0, 'Medical': 1, 'Marketing': 2, 'Technical Degree': 3, 'Human Resources': 4, 'O

df.JobRole.replace({
'Sales Executive': 0, 'Research Scientist': 1, 'Laboratory Technician': 2,'Manufacturing Director': 3,'Healthcare Representa
    'Sales Representative': 6,'Research Director': 7,'Human Resources': 8
}, inplace=True)
```

Now replace other categorical variables with dummy values.

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	...	PerformanceRatin
0	41	1	1	1102	0	1	2	0	2	0	...	
1	49	0	2	279	1	8	1	0	3	1	...	
2	37	1	1	1373	1	2	2	5	4	1	...	
3	33	0	2	1392	1	3	4	0	4	0	...	
4	27	0	1	591	1	2	1	1	1	1	...	

5 rows × 31 columns

Modeling the data

We have our final dataset ready. We now have to start modeling- Predicting the Attrition data. Wait wait? Are you also Confuse like me? We already have the Attrition data then what is it here to predict? well most of the time in Regression and classification problems, you run your model the available values and check the metrics like accuracy of the model by comparing observe values with true values. If you won't have the true values how would you know that the prediction are correct? Now we will realize that how important the training data phase is. We train the model in a way that we can predict(almost) correct result.

In this dataset, we don’t have any missing values from Attrition dataset, we are doing split the data into train and test. We will train the model on training data and predict the results on test dataset.

For this exercise, we will use Random Forest Classifier any many more.

1) Logistic regression

```
In [82]:
lm.fit(x_train, y_train)
```

```
Out[82]: LogisticRegression()
```

```
In [83]:
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score
lm = LogisticRegression()
lm.fit(x_train, y_train)
lm_predict = lm.predict(x_test)
```

```
In [84]:
lm_conf_matrix = confusion_matrix(y_test, lm_predict)
lm_acc_score = accuracy_score(y_test, lm_predict)
print(lm_conf_matrix)
print(lm_acc_score)
```

```
[[386  18]
 [ 70  12]]
0.8189300411522634
```

2)confusion matrix

```
In [87]:
rf_conf_matrix = confusion_matrix(y_test, rf_predict)
rf_acc_score = accuracy_score(y_test, rf_predict)
print(rf_conf_matrix)
print(rf_acc_score)
```

```
[[393  11]
 [ 75   7]]
0.823045267489712
```

3) lasso Ridge

```
In [96]:
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Lasso,Ridge
ls=Lasso()
ls.fit(x_train,y_train)
print(ls.score(x_train,y_train))
alphavalue={'alpha':[1,0.1,0.01,0.001,0.0001,0]}
model=Ridge()
grid=GridSearchCV(estimator=model,param_grid=alphavalue)
grid.fit(x,y)
print(grid)
print(grid.best_estimator_.alpha)
print(grid.best_params_)
```

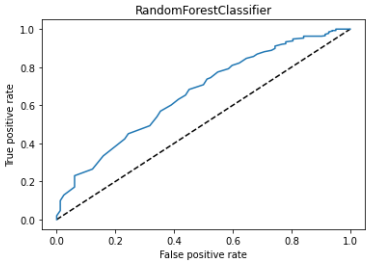
```
0.032642977076592694
GridSearchCV(estimator=Ridge(),
             param_grid={'alpha': [1, 0.1, 0.01, 0.001, 0.0001, 0]})
1
{'alpha': 1}
```

4) roc_auc_score

In [104]:

```
from sklearn.metrics import roc_curve
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score

auc_score=roc_auc_score(y_test,lm.predict(x_test))
y_pred_prob=rf.predict_proba(x_test)[:,0]
tpr,fpr,thresholds=roc_curve(y_test,y_pred_prob)
plt.plot([0,1],[0,1],'k--')
plt.plot(fpr,tpr,label='RandomForestClassifier')
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('RandomForestClassifier')
plt.show()
auc_score=roc_auc_score(y_test,rf.predict(x_test))
auc_score
```



Out[104]: 0.5290690654431296

Summary

Throughout this blog, we can see that Data is important in the Human Resource department(actually in most places it is important). We saw how we can avoid using correlated values and why it is important not to use those while modeling. We used Random Forest and learned how it can be very advantageous over other available machine learning algorithms. Most of all we can find the factors which are most important to employees and if are not fulfilled might lead to Attrition.