HR Analytics Predictions

Hilman Bin Zurin

Github link: https://github.com/hilman1998/HR-Analytics

Problem Statement

- Over the years, employee attrition has been a massive problem for companies the world over.
- One paper (Chen 2023) notes that the overall employee turnover rate in 2021 was as high as 53.7%, with many industries experiencing rates near 19%, significantly above the 10% basic standard.
- The aim of this research is to determine the probability that an employee will leave a company.

1. Chen, B., 2023. Factors of Employee Attrition: A Logistic Regression Approach. [online] Available at: Bingzhe Chen,

Why is this important?

To help companies:

- identify important factors influencing attrition.
- make the right steps to keep employees loyal and happy.

Data Source

Data will be from Kaggle.

What are the features?

The features are a mix of employee and employer survey results and general data about the employee such as age, years at company...

Data cleaning

```
generaldata.head()
```

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	JobLevel	JobRole
0	51	No	Travel_Rarely	Sales	6	2	Life Sciences	1	1	Female	1	Healthcare Representative
1	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	2	Female	1	Research Scientist
2	32	No	Travel_Frequently	Research & Development	17	4	Other	1	3	Male	4	Sales Executive
3	38	No	Non-Travel	Research & Development	2	5	Life Sciences	1	4	Male	3	Human Resources
4	32	No	Travel_Rarely	Research & Development	10	1	Medical	1	5	Male	1	Sales Executive

- Many columns were categorical and needed to be changed into numerical.
- Some cells which had empty data were replaced with the mean or mode of the column (depending on the nature of the column)
- One hot encoding was done on some of the columns so that the full extent of the data can be analysed and modeled properly.

Data preparation

```
X_train
```

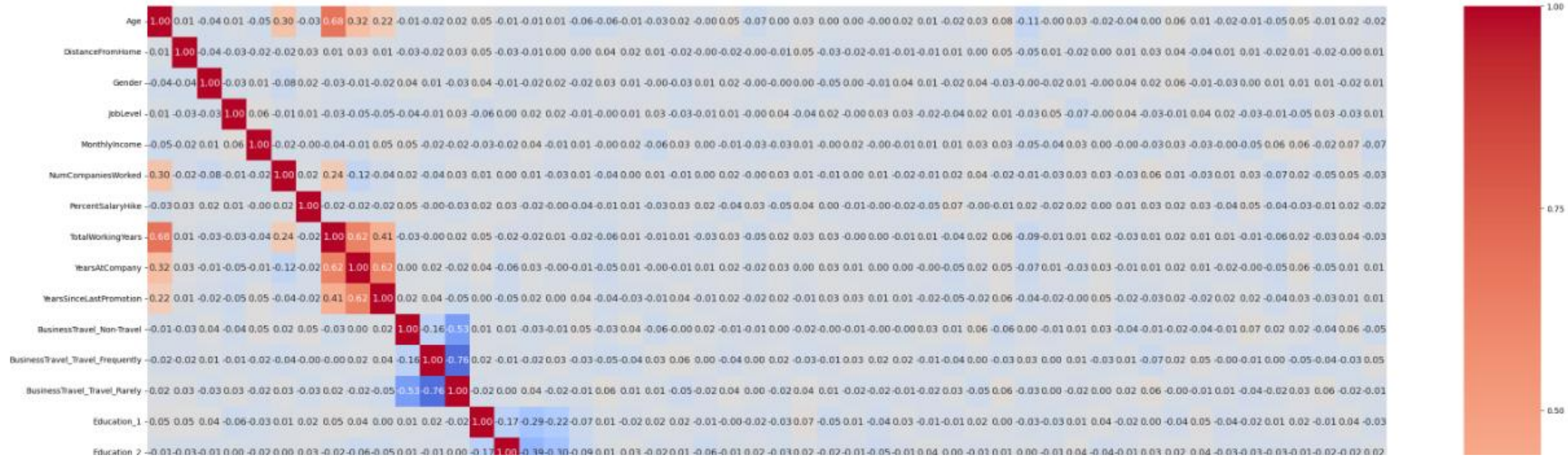
	Age	DistanceFromHome	Gender	JobLevel	MonthlyIncome	NumCompaniesWorked	PercentSalaryHike	TotalWorkingYears	YearsAtCompany	YearsSinceL
2640	40	1	1	2	50710	8.0	17	8.0	1	
3476	28	1	1	2	63470	1.0	15	4.0	4	
4006	28	7	1	1	89660	1.0	16	3.0	3	
1436	38	1	1	4	64720	0.0	12	17.0	16	
3265	40	10	1	2	65670	1.0	13	8.0	8	
...	...	...	...	...	...	...	...	...	...	
3331	37	13	0	3	35640	5.0	11	10.0	5	
71	33	4	1	4	47880	3.0	11	9.0	7	
133	43	10	0	1	46170	1.0	11	25.0	25	
2015	33	9	0	2	46490	0.0	12	4.0	3	
1932	47	18	0	2	55820	1.0	16	9.0	9	

3528 rows × 50 columns

After splitting the data into train and test splits, and after the data cleaning stage is completed, the X_train obtained is shown above.

```python
In [155]: plt.figure(figsize=(35, 35))
          sns.heatmap(X_train.corr(), annot=True, cmap='coolwarm', fmt='.2f', annot_kws={"size": 12}, )

          plt.title("")
          plt.show()
```



A heatmap was generated and some columns were removed as they were shown to have high correlation with other columns. The removed columns were

BusinessTravel_Travel_Frequently and Department_Research & Development

Data modelling

```
Scaler_X = StandardScaler()
X_train_sc = Scaler_X.fit_transform(X_train)
X_test_sc = Scaler_X.transform(X_test)
```

```
logreg = LogisticRegression()

logreg.fit(X_train_sc, y_train)

print(f'Logistic Regression Intercept: {logreg.intercept_}')
print(f'Logistic Regression Coefficient: {logreg.coef_}')
```

```
Logistic Regression Intercept: [-2.04181057]
Logistic Regression Coefficient: [[-0.28362326 -0.04007846  0.08881057 -0.06001248 -0.01787243  0.3546515
   0.05264093 -0.47833252 -0.37469418  0.45347952 -0.43449832 -0.31261611
  -0.02275544  0.08742349 -0.00255252 -0.02755167 -0.07574947  0.10333474
   0.00669526  0.03536886 -0.00102275 -0.04323521 -0.06263693 -0.04582837
  -0.03783218  0.01674263 -0.05988855 -0.15737499  0.13187443  0.03947319
   0.0837167  -0.02140927 -0.17299768 -0.10563411  0.26781622  0.0848534
  -0.07753161  0.01054999 -0.03113248  0.03881248  0.01541005  0.06187383
   0.06823067 -0.07178405 -0.00810698 -0.25222044  0.12786333 -0.09199236]]
```

After the data prep, the data was scaled and fitted into a logistic regression model.

Data evaluation

```
cm = confusion_matrix(y_test,y_pred)

print(accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
```

```
0.8310657596371882
[[720  11]
 [138  13]]
```

```
cm_df = pd.DataFrame(cm, index=["Actual Stay", "Actual Leave"], columns=["Predicted Stay", "Predicted Leave"])
```

```
print(cm_df)
```

```
              Predicted Stay  Predicted Leave
Actual Stay              720               11
Actual Leave             138               13
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.98      0.91       731
           1       0.54      0.09      0.15       151

    accuracy                           0.83       882
   macro avg       0.69      0.54      0.53       882
weighted avg       0.79      0.83      0.78       882
```

Class 0 (Employees Who Stay):

- Precision: 0.84 - When the model predicts an employee will stay, it is correct 84% of the time.
- Recall: 0.98 - The model correctly identifies 98% of the employees who actually stay.
- F1-Score: 0.91 - A high F1-score indicates a good balance between precision and recall for this class.

Class 1 (Employees Who Leave):

- Precision: 0.54 - When the model predicts an employee will leave, it is correct 54% of the time.
- Recall: 0.09 - The model correctly identifies only 9% of the employees who actually leave.
- F1-Score: 0.15 - A low F1-score indicates that the model is not performing well in predicting this class.

Overall Model Performance:

- Accuracy: 0.83 - Overall, the model correctly predicts the status (stay or leave) of 83% of the employees.
- Macro Average: Averages for precision, recall, and F1-score are 0.69, 0.54, and 0.53 respectively, indicating moderate performance.
- Weighted Average: Averages for precision, recall, and F1-score are 0.79, 0.83, and 0.78 respectively, weighted for class imbalance.

Moving Forward

- This model can be used to create an internal tool (application) to help HR departments predict which of their employees will stay or leave.
- This can go a long way to helping companies keep and get the best talent.
- It can also help companies determine the right amount of bonuses by looking at past data and future expectations.

Risks Moving Forward

- Questions may be asked whether using AI is ethical for making big HR-related decisions for a company.

Thank you