TEESSIDE UNIVERSITY

MIDDLESBOROUGH - TS1 3BA

School of Computing, Engineering and Digital technologies

MACHINE LEARNING - CIS4035-N

DATE OF SUBMISSION:08/05/24

Student Name:

Student ID:

Contents

Analysis of HR Employee Attrition Prediction Using Data Analysis

***Abstract*— HR Employee Attrition is one of the crucial concern for the organization for the negative impact in workplace productivity and to improve the strategies for betterment. The main aim of this research paper is to apply the machine learning techniques to predict the employee attrition based on attributes. Here we apply the machine learning models to calculate the employee turnover rate using HR analytics. We have collected the data from Kaggle which is containing samples of 1470 employees. Data cleaning and preprocessing is an important part of the model evaluation where we clean the data and transform it a useful format. Logistic regression, decision tree and xg boost classification model are being used by Python. This paper also introduced the accuracy score that which machine learning model performs better to provide better management in taking a good decision for an organization.**

I. INTRODUCTION

HR Employee attrition is a process of employment in which employee leave the workforce and to leave the workforce, there may be several reasons such as resignation, termination or retirement. Employee attrition is also a not a good thing and it is very important to minimize the employee attrition to improve the organization productivity and higher competitive advantages over its competition. To improve the development of corporation, it is very necessary to understand the behaviour of customer like why they are leaving company, by what reason company terminate the employee and what is the perfect age to retire the employee. To identify the highly talented people in an organization is one of the top challenge and their human resources plays an important role in taking the beneficial decision for company productivity and profits, sometimes it's either good for employee or good for organization.

The main objective of this research analysis is to predict the employee attrition based on various features like education, job role, job satisfaction, department, business travel, overtime, hike and many more features. Organization is all over the world is challenging in talent and facing loss due to employee attrition such as resigning, retirement from the current positions. It only does not affect the organization but also affect the project as well because they have to hire other people and then invest time and money which is very time consuming. The main purpose of this research is to analyses the various attributes such as salary, growth, facilities, appreciation, suggestions, policies, procedures and many more which is helpful to determine the attrition rate for the organizations. This research is also very helpful to understand which employee is more likely to leave the company. Here dataset is collected from the Kaggle and perform the data cleaning techniques. After following the basic steps, machine learning are also applied to analyses the prediction of HR employee attrition.

II. TECHNOLOGIES

A. Data Analysis

Libraries Used

In this HR analytics prediction analysis, the initial step is to import the Python libraries to perform further steps for better data analysis. We have used the pandas and numpy libraries for data manipulation. Seaborn and matplotlib libraries are used for data visualization through which we can plot the graphs and charts attractively by giving labels, colourful markers, titles, etc. By pandas, we can able to perform various operations to manipulate the dataset.

Import Libraries

```
92]: #import libraries
     import pandas as pd
     import numpy as np
     import seaborn as sns
     import matplotlib.pyplot as plt
```

Figure 1: Import Libraries

B. Data Collection

HR Employee attrition dataset is collected from the Kaggle source whose motive to predict which employee left the organization due to departure, termination, or many other reasons based on their work profile history. We read the dataset using the library pandas and stored it in the data variable. We use the head function, we display the top 5 rows of the HR attrition dataset. HR employee attrition dataset containing 35 columns and 1470 rows.

Read Dataset

```
In [70]: #read dataset
         data = pd.read_csv('HR-Employee-Attrition.csv')
         data.head()
```

Out[70]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educatic
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life S
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life S
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life S
4	27	No	Travel_Rarely	591	Research & Development	2	1	

5 rows × 35 columns

Figure 2: HR Dataset

We use the info function to find out the data information such as total column names, null values, and data types.

Data Information

```
1]: #data information
    data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
```

Figure 3: Data Information

We have applied the describe function to find out the description of data. This function provides the count, mean, standard deviation, 25%, 50%, 75%, minimum, and maximum values of the numerical columns.

Data Description

```
In [72]: #data description
         data.describe()
```

Out[72]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumb
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.0000(
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.8653(
std	9.135373	403.509100	8.106864	1.024165	0.0	602.0243(
min	18.000000	102.000000	1.000000	1.000000	1.0	1.0000(
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.2500(
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.5000(
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.7500(
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.0000(

8 rows × 26 columns

Figure 4: Data Description

C. Data Cleaning

Data cleaning is one of the important steps of model evaluation because after cleaning the data, machine learning techniques provide better accurate results so we can achieve our goal. The is null function is used to check the missing values and this dataset does not contain any missing data.

Check Null Values

```
3]: #check missing values
    data.isnull().sum()
```

```
3]: 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
```

Figure 5: Check Null Values

The duplicated function is also used to check the duplicate value and the HR employee attrition dataset does not contain duplicate data as well.

```
[75]: #check duplicated records
      data.duplicated()

t[75]: 0       False
       1       False
       2       False
       3       False
       4       False
               ...
       1465    False
       1466    False
       1467    False
       1468    False
       1469    False
       Length: 1470, dtype: bool
```

Figure 6: Check Duplicated Records

This dataset contains so many attributes of which some are also not useful so we have calculated the count of unique values in all the columns so we can drop them accordingly. After finding the unique values for each column, we have dropped the columns that contain one or 2 values only. So dropped columns are Over18, employee count, employee number, standard hours, and performance rating.

	N-Unique	Unique
Age	43	[41, 49, 37, 33, 27, 32, 59, 30, 38, 36, 35, 2...
Attrition	2	[Yes, No]
BusinessTravel	3	[Travel_Rarely, Travel_Frequently, Non-Travel]
DailyRate	886	[1102, 279, 1373, 1392, 591, 1005, 1324, 1358,...
Department	3	[Sales, Research & Development, Human Resources]
DistanceFromHome	29	[1, 8, 2, 3, 24, 23, 27, 16, 15, 26, 19, 21, 5...
Education	5	[2, 1, 4, 3, 5]
EducationField	6	[Life Sciences, Other, Medical, Marketing, Tec...
EmployeeCount	1	[1]
EmployeeNumber	1470	[1, 2, 4, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16,...
EnvironmentSatisfaction	4	[2, 3, 4, 1]
Gender	2	[Female, Male]
HourlyRate	71	[94, 61, 92, 56, 40, 79, 81, 67, 44, 84, 49, 3...
JobInvolvement	4	[3, 2, 4, 1]
JobLevel	5	[2, 1, 3, 4, 5]
JobRole	9	[Sales Executive, Research Scientist, Laborato...
JobSatisfaction	4	[4, 2, 3, 1]
MaritalStatus	3	[Single, Married, Divorced]
MonthlyIncome	1349	[5993, 5130, 2090, 2909, 3468, 3068, 2670, 269...
MonthlyRate	1427	[19479, 24907, 2396, 23159, 16632, 11864, 9964...
NumCompaniesWorked	10	[8, 1, 6, 9, 0, 4, 5, 2, 7, 3]
Over18	1	[Y]

Figure 7: Unique Values

After checking the data information, some columns are of string or object type so it is very important to convert the column into integer format. Label encoding of sklearn preprocessing is applied to encode the data properly.

Encoding

```
'8]: #Label Encoding

     from sklearn.preprocessing import LabelEncoder
     data1[['OverTime','MaritalStatus','JobRole','Gender','EducationField','Departmen
           'Attrition']] = data1[['OverTime','MaritalStatus','JobRole','Gender','Edu
                                  'BusinessTravel','Attrition']].apply(LabelEncoder(
     data1.head()
```

'8]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Education
0	41	1	2	1102	2	1	2	
1	49	0	1	279	1	8	1	
2	37	1	2	1373	1	2	2	
3	33	0	1	1392	1	3	4	
4	27	0	2	591	1	2	1	

5 rows × 30 columns

Figure 8: Data Preprocessing

D. Data Visualization

Data Visualization is helpful for insights and patterns in the given dataset more effectively. Using these techniques, we can handle complex datasets, and represent information in useful graphs so that clients can understand the relatable information within a second. There are various visualizations we can also use but here we used the Python libraries to plot the histogram, bar plot, scatter plot, etc.

The first graph we explained is a count plot of the attrition and according to the representative graph, the count of NO attrition is more than YES. It means, most of the employees do not leave the organization based on the above features.

Data Visualization

```
In [96]: #count plot of Attrition
         import seaborn as sns
         sns.countplot(x="Attrition", data=data)

Out[96]: <AxesSubplot:xlabel='Attrition', ylabel='count'>
```

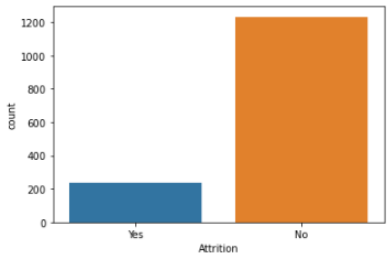

Figure 9: Count Plot of Attrition

The second graph is a histogram of the HR employee attrition dataset which shows the data variation and its range for each numerical column.

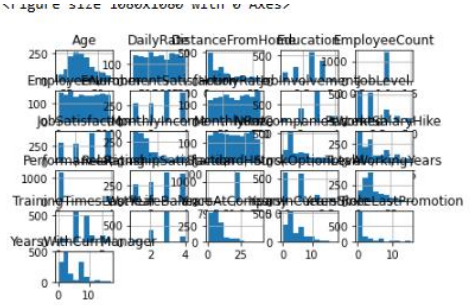

Figure 10: Histogram

The third graph we designed is heat map which is a representation of the correlation matrix of the dataset through which we can identify which variable is highly creatable and useful as compared to other features based on color coding.

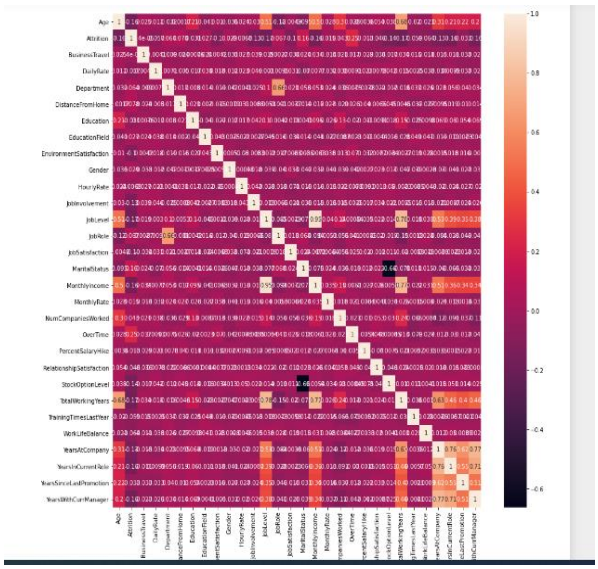

Figure 11: Histogram

E. Techniques Used

We have selected the dependent and independent variables on which we can perform analysis and which depend on other features. Attrition is the dependent variable which is predicted next. 20% of data is taken as test data while the remaining is taken as training data.

Dependent and Independent Variables

```
n [81]: #dependent and independent variables
        x = data1.drop('Attrition', axis=1)
        y = data1['Attrition']
```

Train Test Split

```
32]: #train test split
     from sklearn.model_selection import train_test_split
     x_train,x_test,y_train ,y_test = train_test_split(x,y, test_size = 0.20, random
```

Figure 12: Data Split

3 types of classification-supervised learning techniques are applied to predict the HR employee attrition rate.

Logistic Regression

Logistic regression is a supervised learning technique that is applied to binary variables. This model is applied to true or false, yes or no type values. This model is very useful and highly accurate. This model can work with small and large datasets. In this model, a decision boundary is used to distinguish the attrition classes. This model works with model residuals that are independent and distributed identically.

Logistic Regression

```
[83]: from sklearn.linear_model import LogisticRegression
      log = LogisticRegression(random_state=16)
      log.fit(x_train, y_train)
      y_pred = log.predict(x_test)

      C:\Users\SR\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762:
      ConvergenceWarning: lbfgs failed to converge (status=1):
      STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

      Increase the number of iterations (max_iter) or scale the data as shown in:
          https://scikit-learn.org/stable/modules/preprocessing.html
      Please also refer to the documentation for alternative solver options:
          https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressi
      on
        n_iter_i = _check_optimize_result(

[84]: from sklearn.metrics import confusion_matrix,accuracy_score
      cm = confusion_matrix(y_test, y_pred)
      acc = (accuracy_score(y_test,y_pred))*100
      print("Accuracy Score:- ",acc)
      print ("Confusion Matrix : \n", cm)

      Accuracy Score:-  86.73469387755102
      Confusion Matrix :
       [[255   0]
       [ 39   0]]
```

Figure 13: Logistic Regression

Decision Tree

A decision tree is a supervised learning algorithm whose structure is like a tree in which internal nodes indicate decisions based on feature values and at each internal node, it decides the best features to divide the data based on criteria. Gini impurity is used for the classification dataset. This process continues to subset the data until it does not find better results. An important advantage of this algorithm is its interpretability. This algorithm is very simple and visualizes the decisions based on the decision tree. Sometimes, decision trees suffer the over fitting problem where they are unable to find the patterns from the test data so to handle this pruning techniques are used to handle the over fitting.

Decision Tree

```
In [85]: from sklearn.tree import DecisionTreeClassifier
         clf = DecisionTreeClassifier()
         clf = clf.fit(x_train,y_train)
         y_pred1 = clf.predict(x_test)

In [86]: from sklearn.metrics import confusion_matrix,accuracy_score
         cm1 = confusion_matrix(y_test, y_pred1)
         acc1 = (accuracy_score(y_test,y_pred1))*100
         print("Accuracy Score:- ",acc1)
         print ("Confusion Matrix : \n", cm1)

         Accuracy Score:-  77.21088435374149
         Confusion Matrix :
          [[221  34]
          [ 33   6]]
```

Figure 14: Decision Tree

XG Boost Regression

This is one of the techniques of ensemble learning. The main aim of this technique is to combine multiple weak learners to discover a strong learner and the main focus is to correct the errors faced by previous models. Decision trees are used as weak learners. It is divided into 2 parts: loss function and regularization. It handles the missing values easily and also has the power to handle them during training as well automatically. It is a high-performance model, more scalable, and fast.

XG Boost Classification

```
[87]: from xgboost import XGBClassifier
      xgb = XGBClassifier()
      xgb.fit(x_train , y_train)
      y_pred2 = xgb.predict(x_test)

      C:\Users\SR\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: T
      he use of label encoder in XGBClassifier is deprecated and will be removed in a
      future release. To remove this warning, do the following: 1) Pass option use_la
      bel_encoder=False when constructing XGBClassifier object; and 2) Encode your la
      bels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
        warnings.warn(label_encoder_deprecation_msg, UserWarning)

      [21:23:33] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.
      1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric
      used with the objective 'binary:logistic' was changed from 'error' to 'loglos
      s'. Explicitly set eval_metric if you'd like to restore the old behavior.

[88]: from sklearn.metrics import confusion_matrix,accuracy_score
      cm2 = confusion_matrix(y_test, y_pred2)
      acc2 = (accuracy_score(y_test,y_pred2))*100
      print("Accuracy Score:- ",acc2)
      print ("Confusion Matrix : \n", cm2)

      Accuracy Score:-  87.07482993197279
      Confusion Matrix :
       [[246   9]
       [ 29  10]]
```

Figure 15: XG Boost

III. Conclusion

In this research work, HR employee attrition data is being collected from the Kaggle and implement the data preprocessing. According to the work analysis, it is important to understand the perspective of the employee that why they are leaving company and it depends on various factors such as monthly rate, facilities, policies, etc. We have applied some techniques such as logistic regression, decision tree and xg boost. The accuracy score of logistic regression is 86%. The accuracy score of decision tree is 77%. The accuracy score of xg boost is 87%. So, based on the results, xg boost algorithms performs better and more accurate but one of the disadvantages is it worked with less data so in future it we increase the data size then accuracy score many vary.

Accuracy Score

```
n [91]: import pandas as pd

        df = {'Model Name': ['Logistic Regression', 'Decision Tree', 'XG Boost'],
              'Age': [acc, acc1, acc2]}
        accuracy_table = pd.DataFrame(df)
        accuracy_table
```

ut[91]:

	Model Name	Age
0	Logistic Regression	86.734694
1	Decision Tree	77.210884
2	XG Boost	87.074830

Figure 16: Accuracy Table

IV. Future Scope

The future of HR analytics includes several significances such as helping in data-driven, by using machine learning which tends to provide better workforce management. In the future, we can use deep learning models as well to provide better policies to the employees according to their needs and provide a satisfied workforce.

V. References

[1] Chung, D., Yun, J., Lee, J. and Jeon, Y., 2023. Predictive model of employee attrition based on stacking ensemble learning. Expert Systems with Applications, 215, p.119364.

[2] Kakulapati, V. and Subhani, S., 2023. Predictive Analytics of Employee Attrition using K-Fold Methodologies. IJ Mathematical Sciences and Computing, 1, pp.23-36.

[3] Mozaffari, F., Rahimi, M., Yazdani, H. and Sohrabi, B., 2023. Employee attrition prediction in a pharmaceutical company using both machine learning approach and qualitative data. Benchmarking: An International Journal, 30(10), pp.4140-4173.

[4] Yuan, S., Kroon, B. and Kramer, A., 2024. Building prediction models with grouped data: A case study on the prediction of turnover intention. Human Resource Management Journal, 34(1), pp.20-38.

[5] Speer, A.B., 2024. Empirical attrition modelling and discrimination: Balancing validity and group differences. Human Resource Management Journal, 34(1), pp.1-19.

[6] Kumari, B.K., Sundari, V.M., Praseeda, C., Nagpal, P., EP, J. and Awasthi, S., 2023. Analytics-Based Performance Influential Factors Prediction for Sustainable Growth of Organization, Employee Psychological Engagement, Work Satisfaction, Training and Development. Journal for ReAttach Therapy and Developmental Diversities, 6(8s), pp.76-82.

[7] Madhani, P.M., 2023. Human resources analytics: leveraging human resources for enhancing business performance. Compensation & Benefits Review, 55(1), pp.31-45.

[8] Chowdhury, S., Joel-Edgar, S., Dey, P.K., Bhattacharya, S. and Kharlamov, A., 2023. Embedding transparency in artificial intelligence machine learning models: managerial implications on predicting and explaining employee turnover. The

International Journal of Human Resource Management, 34(14), pp.2732-2764.

[9] Thakral, P., Srivastava, P.R., Dash, S.S., Jasimuddin, S.M. and Zhang, Z., 2023. Trends in the thematic landscape of HR analytics research: a structural topic modeling approach. Management Decision, 61(12), pp.3665-3690.

[10] Dixit, C.K., Somani, P., Gupta, S.K. and Pathak, A., 2023. Data-Centric Predictive Modeling of Turnover Rate and New Hire in Workforce Management System. In Designing Workforce Management Systems for Industry 4.0 (pp. 121-138). CRC Press.

[11] Huang, X., Yang, F., Zheng, J., Feng, C. and Zhang, L., 2023. Personalized human resource management via HR analytics and artificial intelligence: Theory and implications. Asia Pacific Management Review, 28(4), pp.598-610.

[12] Bahuguna, P.C., Srivastava, R. and Tiwari, S., 2024. Human resources analytics: where do we go from here?. Benchmarking: An International Journal, 31(2), pp.640-668.

Hasan, M.R., Ray, R.K. and Chowdhury, F.R., 2024. Employee Performance Prediction: An Integrated Approach of Business Analytics and Machine Learning. Journal of Business and Management Studies, 6(1), pp.215-219.