**HR Analytics Project**

1.Problem Definition

Attrition is a problem that impacts all businesses, irrespective of geography, industry and size of the company. Employee attrition leads to significant costs for a business, including the cost of business disruption, hiring new staff and training new staff. As such, there is great business interest in understanding the drivers of and minimizing staff attrition.

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.

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.**

In this context, the use of classification models to predict if an employee is likely to quit could greatly increase the HR’s ability to intervene on time and remedy the situation to prevent attrition. While this model can be routinely run to identify employees, who are most likely to quit, the key driver of success would be the human element of reaching out the employee, understanding the current situation of the employee and taking action to remedy controllable factors that can prevent attrition of the employee.

This data set presents an employee survey from IBM, indicating if there is attrition or not. The data set contains approximately 1470 entries. Given the limited size of the data set, the model should only be expected to provide modest improvement in indentification of attrition.

While some level of attrition in a company is inevitable, minimizing it and being prepared for the cases that cannot be helped will significantly help improve the operations of most businesses. As a future development, with a sufficiently large data set, it would be used to run a segmentation on employees, to develop certain “at risk” categories of employees. This could generate new insights for the business on what drives attrition, insights that cannot be generated by merely informational interviews with employees.

2.Data Analysis

In order to start with exercise, I have used IBM HR Attrition Dataset The dataset includes features like Age, Employee Role, Daily Rate, Job Satisfaction, Years at Company, Current Role etc. It includes the data of 1470 employees.

(Data source: <https://github.com/dsrscientist/IBM_HR_Attrition_Rate_Analytics>)

| Name | Description |
| --- | --- |
| AGE | Numerical Value |
| ATTRITION | Employee leaving the company (0=no, 1=yes) |
| BUSINESS TRAVEL | (1=No Travel, 2=Travel Frequently, 3=Tavel Rarely) |
| DAILY RATE | Numerical Value - Salary Level |
| DEPARTMENT | (1=HR, 2=R&D, 3=Sales) |
| DISTANCE FROM HOME | Numerical Value - THE DISTANCE FROM WORK TO HOME |
| EDUCATION | Numerical Value |
| EDUCATION FIELD | (1=HR, 2=LIFE SCIENCES, 3=MARKETING, 4=MEDICAL SCIENCES, 5=OTHERS, 6= TEHCNICAL) |
| EMPLOYEE COUNT | Numerical Value |
| EMPLOYEE NUMBER | Numerical Value - EMPLOYEE ID |
| ENVIROMENT SATISFACTION | Numerical Value - SATISFACTION WITH THE ENVIROMENT |
| GENDER | (1=FEMALE, 2=MALE) |
| HOURLY RATE | Numerical Value - HOURLY SALARY |
| JOB INVOLVEMENT | Numerical Value - JOB INVOLVEMENT |
| JOB LEVEL | Numerical Value - LEVEL OF JOB |
| JOB ROLE | (1=HC REP, 2=HR, 3=LAB TECHNICIAN, 4=MANAGER, 5= MANAGING DIRECTOR, 6= REASEARCH DIRECTOR, 7= RESEARCH SCIENTIST, 8=SALES EXECUTIEVE, 9= SALES REPRESENTATIVE) |
| JOB SATISFACTION | Numerical Value - SATISFACTION WITH THE JOB |
| MARITAL STATUS | (1=DIVORCED, 2=MARRIED, 3=SINGLE) |
| MONTHLY INCOME | Numerical Value - MONTHLY SALARY |
| MONTHY RATE | Numerical Value - MONTHY RATE |
| NUMCOMPANIES WORKED | Numerical Value - NO. OF COMPANIES WORKED AT |
| OVER 18 | (1=YES, 2=NO) |
| OVERTIME | (1=NO, 2=YES) |
| PERCENT SALARY HIKE | Numerical Value - PERCENTAGE INCREASE IN SALARY |
| PERFORMANCE RATING | Numerical Value - ERFORMANCE RATING |
| RELATIONS SATISFACTION | Numerical Value - RELATIONS SATISFACTION |
| STANDARD HOURS | Numerical Value - STANDARD HOURS |
| STOCK OPTIONS LEVEL | Numerical Value - STOCK OPTIONS |
| TOTAL WORKING YEARS | Numerical Value - TOTAL YEARS WORKED |
| TRAINING TIMES LAST YEAR | Numerical Value - HOURS SPENT TRAINING |
| WORK LIFE BALANCE | Numerical Value - TIME SPENT BEWTWEEN WORK AND OUTSIDE |
| YEARS AT COMPANY | Numerical Value - TOTAL NUMBER OF YEARS AT THE COMPNAY |
| YEARS IN CURRENT ROLE | Numerical Value -YEARS IN CURRENT ROLE |
| YEARS SINCE LAST PROMOTION | Numerical Value - LAST PROMOTION |
| YEARS WITH CURRENT MANAGER | Numerical Value - YEARS SPENT WITH CURRENT MANAGER |

As here, we can see that shape of data is (1470,35). In which, No of numerical features: 26 , which are: ['Age', 'DailyRate', 'DistanceFromHome', 'Education', 'EmployeeCount', 'EmployeeNumber', 'EnvironmentSatisfaction', 'HourlyRate', 'JobInvolvement', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager'] And No of categorical features: 9 ,which are: ['Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'Over18', 'Overtime']

3.EDA Concluding Remarks

But before we begin any problem, we must understand what the problem actually is there in come the analysis part. Since we have a large volume of data, we must apply statistical analytical tools to understand the various factors at play. This is called exploratory analysis.

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. It is a good practice to understand the data first and try to gather as many insights from it.

By checking the statistics summary ,  
According to dataset : Number of rows in each column are same, means there are no null values in the data set. The mean and median of most of the column are same and the STD and mean are very close to each other. By checking the difference between the 75% and max value there are outliers in some of the column. Unique shows the no of unique values in categorical data.

Sales Department suffers the highest attrition with Sales Representatives. They are closely followed the HR department. Manager and Director positions is the roles with the least loss across all departments. Laboratory Technician role has nearly the same attrition as the HR and Sales Department. Overtime people have 3 times the attrition rate of the non-overtime people. Medical, Life Sciences and Others seem have very close levels of attritions which is among the lowest. Quite Naturally higher Educations seems to be tied with lower attrition, but Doctors and HR people of unusual fields have the highest rate, which is almost equal to below college HR people. As expected, higher the Job Satisfaction less is the attrition.

4.Pre-processing Pipeline

Now preparing the data for modeling and training. For this need to do two steps:

**1.Checking and removing outliers.**

# checking outliers

plt.figure(figsize=(10,5))

df.plot(kind='box',subplots=**True**,layout=(4,8))

plt.show()

# Removing outliers

**from** **scipy.stats** **import** zscore *# removing outliers*

z=abs(zscore(df))

print(df.shape)

df\_final=df.loc[(z<3).all(axis=1)]

print(df\_final.shape)

df=df\_final

(1470, 32)

(1387, 32)

y=df["Attrition"]

dfx=df.drop(columns=['Attrition'], axis=1)

**2.Checking and handling skewness to make data in form to fit in model building**

*#Checking skewness*

dfx.skew()

Age 0.472280

BusinessTravel -1.426774

DailyRate -0.017078

Department 0.183919

DistanceFromHome 0.954752

Education -0.289024

EducationField 0.544868

EmployeeNumber 0.018931

EnvironmentSatisfaction -0.325285

Gender -0.417296

HourlyRate -0.030481

JobInvolvement -0.501401

JobLevel 1.126075

JobRole -0.386843

JobSatisfaction -0.345612

MaritalStatus -0.160952

MonthlyIncome 1.544770

MonthlyRate 0.030596

NumCompaniesWorked 1.037715

OverTime 0.954751

PercentSalaryHike 0.800592

PerformanceRating 1.931566

RelationshipSatisfaction -0.295686

StockOptionLevel 0.962332

TotalWorkingYears 1.034487

TrainingTimesLastYear 0.577614

WorkLifeBalance -0.557100

YearsAtCompany 1.248623

YearsInCurrentRole 0.726675

YearsSinceLastPromotion 1.756335

YearsWithCurrManager 0.694506

dtype: float64

*# handling skewness*

**from** **sklearn.preprocessing** **import** PowerTransformer

pt=PowerTransformer(method='yeo-johnson')

d=pt.fit\_transform(dfx)

d=pd.DataFrame(d,columns=dfx.columns)

x=d

5.Building Machine Learning Models

Now that we are done with the basic pre-processing steps, we can go ahead and build simple machine learning models over this data. We will try 5 models here – **KNeighborsClassifier, SVC, LogisticRegression, DecisionTreeClassifier, GaussianNB to predict the Attrition.**

To compare the performance of the models, we will create a validation set (or test set). Here I have randomly split the data into two parts using the train\_test\_split() function, such that the validation set holds 33% of the data points while the train set has 64%

**Prediction using classification model**

In [36]:

**from** **sklearn** **import** linear\_model

**from** **sklearn.metrics** **import** r2\_score

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=.33,random\_state=42)

In [37]:

print(x\_train.shape,x\_test.shape)

print(y\_train.shape,y\_test.shape)

(929, 31) (458, 31)

(929,) (458,)

In [38]:

maxrscore=0

**for** r\_state **in** range(42,100):

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=.33,random\_state=r\_state)

reg=linear\_model.LinearRegression()

reg.fit(x\_train,y\_train)

y\_pred= reg.predict(x\_test)

r2s=r2\_score(y\_test,y\_pred)

**if** r2s > maxrscore:

maxrscore=r2s

fr\_state=r\_state

print("max r2 score corresponding to ",fr\_state," is ",maxrscore)

max r2 score corresponding to 72 is 0.2810502252374658

In [39]:

*#GRID SEARCHCV*

parameter = {'kernel':('linear', 'rbf','poly'), 'C':[1, 10]}

svc = SVC()

grid = GridSearchCV( estimator=SVC(), param\_grid = parameter)

grid.fit(x, y)

print(grid)

*#print(grid.best\_score)*

print(grid.best\_estimator\_.kernel)

print(grid.best\_params\_)

s=grid.best\_estimator\_.kernel

GridSearchCV(cv=None, error\_score=nan,

estimator=SVC(C=1.0, break\_ties=False, cache\_size=200,

class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3,

gamma='scale', kernel='rbf', max\_iter=-1,

probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False),

iid='deprecated', n\_jobs=None,

param\_grid={'C': [1, 10], 'kernel': ('linear', 'rbf', 'poly')},

pre\_dispatch='2\*n\_jobs', refit=True, return\_train\_score=False,

scoring=None, verbose=0)

linear

{'C': 1, 'kernel': 'linear'}

In [40]:

*#GRID SEARCHCV*

parameter = {'n\_neighbors':(3,5,7,9,11), 'weights':['uniform','distance']}

knn = KNeighborsClassifier()

grid = GridSearchCV( estimator=knn, param\_grid = parameter)

grid.fit(x, y)

print(grid)

*#print(grid.best\_score)*

print(grid.best\_estimator\_.n\_neighbors)

print(grid.best\_estimator\_.weights)

print(grid.best\_params\_)

k=grid.best\_estimator\_.n\_neighbors

w=grid.best\_estimator\_.weights

GridSearchCV(cv=None, error\_score=nan,

estimator=KNeighborsClassifier(algorithm='auto', leaf\_size=30,

metric='minkowski',

metric\_params=None, n\_jobs=None,

n\_neighbors=5, p=2,

weights='uniform'),

iid='deprecated', n\_jobs=None,

param\_grid={'n\_neighbors': (3, 5, 7, 9, 11),

'weights': ['uniform', 'distance']},

pre\_dispatch='2\*n\_jobs', refit=True, return\_train\_score=False,

scoring=None, verbose=0)

5

uniform

{'n\_neighbors': 5, 'weights': 'uniform'}

In [41]:

KNN=KNeighborsClassifier(n\_neighbors=k,weights=w)

SV=SVC(kernel=s)

LR=LogisticRegression()

DT=DecisionTreeClassifier(random\_state=fr\_state)

GNB=GaussianNB()

In [42]:

models = []

models.append(('KNeighborsClassifier', KNN))

models.append(('SVC', SV))

models.append(('LogisticRegression', LR))

models.append(('DecisionTreeClassifier', DT))

models.append(('GaussianNB', GNB))

In [43]:

**from** **sklearn.metrics** **import** classification\_report,confusion\_matrix,accuracy\_score,roc\_curve,auc

In [44]:

Model = []

score = []

cvs=[]

rocscore=[]

**for** name,model **in** models:

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*',name,'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('**\n**')

Model.append(name)

model.fit(x\_train,y\_train)

print(model)

pre=model.predict(x\_test)

print('**\n**')

AS=accuracy\_score(y\_test,pre)

print('Accuracy\_score = ',AS)

score.append(AS\*100)

print('**\n**')

sc = cross\_val\_score(model, x, y, cv=10, scoring='accuracy').mean()

print('Cross\_Val\_Score = ',sc)

cvs.append(sc\*100)

print('**\n**')

false\_positive\_rate, true\_positive\_rate, thresholds = roc\_curve(y\_test,pre)

roc\_auc = auc(false\_positive\_rate, true\_positive\_rate)

print ('roc\_auc\_score = ',roc\_auc)

rocscore.append(roc\_auc\*100)

print('**\n**')

print('classification\_report**\n**',classification\_report(y\_test,pre))

print('**\n**')

cm=confusion\_matrix(y\_test,pre)

print(cm)

print('**\n**')

plt.figure(figsize=(10,40))

plt.subplot(911)

plt.title(name)

print(sns.heatmap(cm,annot=**True**))

plt.subplot(912)

plt.title(name)

plt.plot(false\_positive\_rate, true\_positive\_rate, label='AUC = **%0.2f**'% roc\_auc)

plt.plot([0,1],[0,1],'r--')

plt.legend(loc='lower right')

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

print('**\n\n**')

6.Concluding Remarks:

In this session we will check dissimilarity between different these Five algorithms:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Model** | **Acuracy\_score** | **Cross\_val\_score** | **Roc\_auc\_curve** |
| 0 | KNeighborsClassifier | 86.68 | 84.35 | 58.96 |
| 1 | SVC | 87.77 | 87.16 | 69.80 |
| 2 | LogisticRegression | 87.77 | 87.45 | 71.84 |
| 3 | DecisionTreeClassifier | 79.69 | 78.37 | 62.41 |
| 4 | GaussianNB | 81.22 | 81.61 | 73.49 |

As it can been seen from above comparison table the applied models can successfully predict 87 percentage accuracy by Logistic regression and SVC, KNeighborsClassifier with 86 percentage accuracy , Gaussian Naïve Bayes with 81 percentage accuracy and then DecisionTreeClassifier with 79 percentage accuracy.

Thus, Log Regression is the best model, as it always predicts a high area in accuracy and a better confusion matrix.

**LOGISTIC REGRESSION**

Itis the appropriate regression analysis to conduct when the dependent variable isbinary. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

With an accuracy of 87.77 and a Cross\_val\_score of 87.45 which makes this the most suitable model.

Basically, with the help of our model, the HR department of a firm could sieve the attrition which will hinder the cost of business, just by collecting the needed data then run through the model.

A special thanks to Data Trained for creating the platform to learn Data Science and Artificial Intelligence by creating world-class learning.

Click here for the link to the notebook:

https://github.com/jainnvandana/evaluation-projects/blob/main/project4\_HRanalytics.ipynb