**Project Topic:**

**HR Analytics Project-**

**Understanding the Attrition in HR**

**Project Description:**

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.

There are several steps involved in this project.The steps are

**Step 1: Exploration Of Dataset**

**Step 2: Analysing And Visualisation Of The Dataset**

# **Step 3: Encoding**

# **Step 4: Corelation**

# **Step 5: Spliting The Columns Into Features And Target**

# **Step 6 :Standardization**

# **Step: 7 Variance Inflation Factor**

# **Step: 8 Handling Imbalanced Data**

# **Step: 9 Model selection**

# **Step: 10 Cross Validation**

# **Step: 11 Hyperparameter Tuning**

# **Step: 12 Evaluation**

# **STEP:1 EXPLORATION OF DATASET:**

Starting the given task of HR analytics prediction with machine learning by importing the necessary Python libraries and the dataset:

A screenshot of a computer code

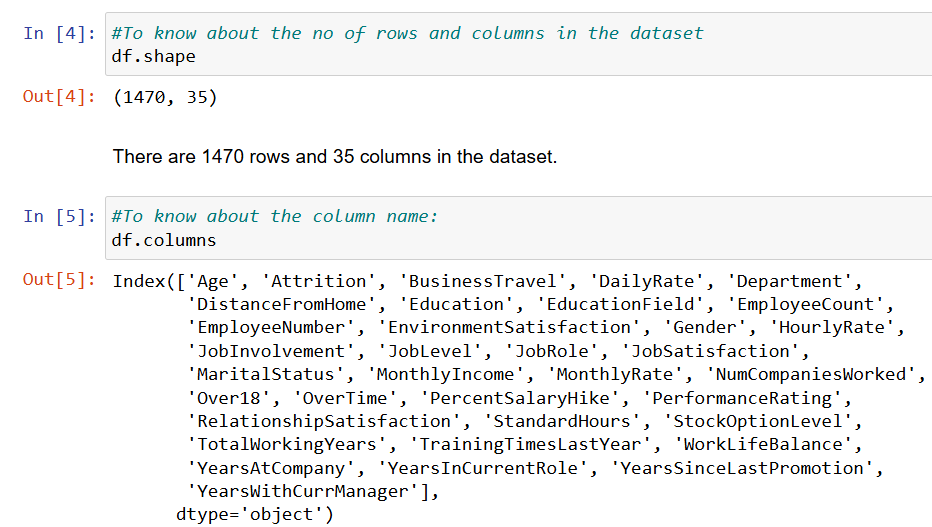
Description automatically generated

Downloading the dataset into workbook for further analysing with the help of pandas.read\_excel. Since the given dataset in the form of excel.

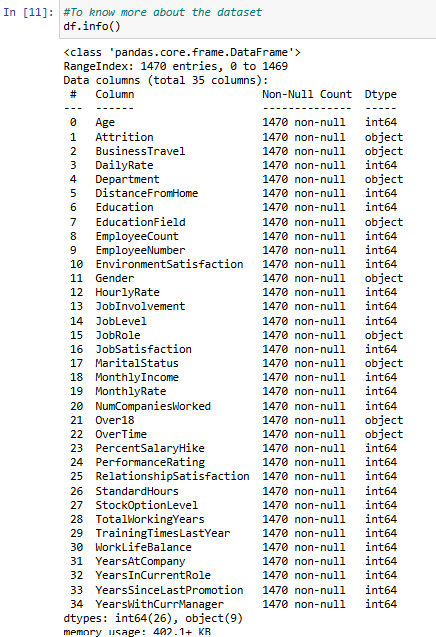
A screenshot of a computer

Description automatically generated

For understanding the given dataset we have to get knowledge about the rows ,columns and their respective datatypes. With the help of df.shape and df.info() we can get information about the shape and datatypes in their corresponding columns.



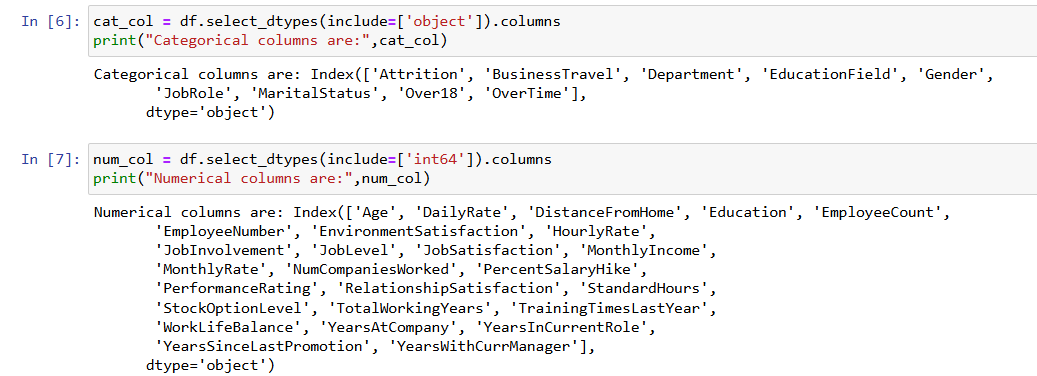
These are the column names of the given dataset and their dtype information.



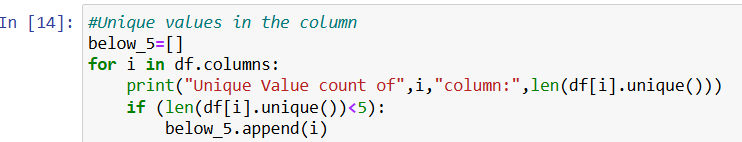
Now we have knowledge about the columns name and datatype stored in them.

For futher deep analysing the columns are separated based the datatype in them.

The columns which has object datatype are considered as the categorical columns and the columns which has int,float datatype are considered as the numerical columns.These columns are separately stored in the list.



From this we can see there 9 columns which has object datatype and balance all the columns out of 35 has int or float values stored in them.



By using the for loop iterating through all the columns to know how many unique values in the columns which also helps in separating the columns into qualitative and quantitative some of the columns have numerical- categories in the columns by using this we can know about those columns also.

The output of the above mention code produce this result:

Unique Value count of Age column: 43

Unique Value count of Attrition column: 2

Unique Value count of BusinessTravel column: 3

Unique Value count of DailyRate column: 886

Unique Value count of Department column: 3

Unique Value count of DistanceFromHome column: 29

Unique Value count of Education column: 5

Unique Value count of EducationField column: 6

Unique Value count of EmployeeCount column: 1

Unique Value count of EmployeeNumber column: 1470

Unique Value count of EnvironmentSatisfaction column: 4

Unique Value count of Gender column: 2

Unique Value count of HourlyRate column: 71

Unique Value count of JobInvolvement column: 4

Unique Value count of JobLevel column: 5

Unique Value count of JobRole column: 9

Unique Value count of JobSatisfaction column: 4

Unique Value count of MaritalStatus column: 3

Unique Value count of MonthlyIncome column: 1349

Unique Value count of MonthlyRate column: 1427

Unique Value count of NumCompaniesWorked column: 10

Unique Value count of Over18 column: 1

Unique Value count of OverTime column: 2

Unique Value count of PercentSalaryHike column: 15

Unique Value count of PerformanceRating column: 2

Unique Value count of RelationshipSatisfaction column: 4

Unique Value count of StandardHours column: 1

Unique Value count of StockOptionLevel column: 4

Unique Value count of TotalWorkingYears column: 40

Unique Value count of TrainingTimesLastYear column: 7

Unique Value count of WorkLifeBalance column: 4

Unique Value count of YearsAtCompany column: 37

Unique Value count of YearsInCurrentRole column: 19

Unique Value count of YearsSinceLastPromotion column: 16

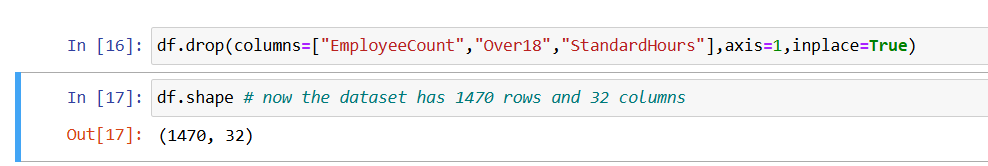
Unique Value count of YearsWithCurrManager column: 18

With this count we can see that Only one unique value in Over18, EmployeeCount, StandardHours column. This is not going to influence much to output so these columns can be removed.

A screenshot of a computer

Description automatically generated

These are columns which has Numerical datatype and it has categories.



Dropping column which has only unique values .

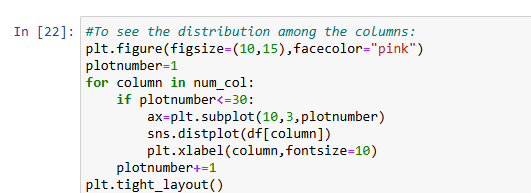
# **STEP 2 :ANALYSING AND VISUALISATION OF THE DATASET**

A screenshot of a computer

Description automatically generated

From this we can say there is no missing and duplicated values in the dataset.

To know about the statistical description of the dataset with the help of pandas df.describe().With this we can analyse the count,Mean,25th percentile,50th percentile,75th percentile ,Maximum and Minimum.

Analysing the distribution of the columns with distplot 

Using this code we can see the distribution of all the columns observed certain facts .The observation are

1] The count of all columns are equal.

2] The mean values is lower than median value- it represents the left skewed data.The columns which may have this are

TrainingTimesLastYear,WorkLifeBalance.

3] The mean values is greater that median value- it represents the right skewed data.The columns which may have this are

Age,DistanceFromHome,YearsSinceLastPromotion,YearsWithCurrManager,TotalWorkingYears,YearsAtCompany,

YearsInCurrentRole,MonthlyIncome,DistanceFromHome,DailyRate

4]The mean value and median value are almost equal and have some symmentric distribution -DailyRate,

A screenshot of a computer code

Description automatically generated

Dropping the EmployeeNumber and checking the shape of the dataset.

# **Step 3: ENCODING**

In the step 3 converting all the object datatype into numerical datatype. Categorical data needs to be converted into a numerical format to be used effectively. This enhances the model performance.

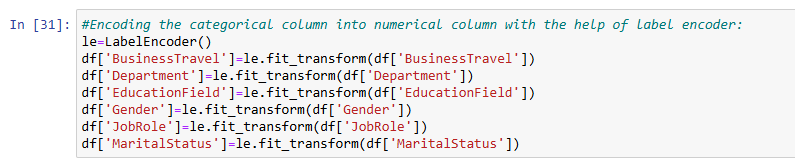
A screenshot of a computer code

Description automatically generated

With the help of label encoder-Encoding the valuesA screenshot of a computer code

Description automatically generated

Columns with only two unique values like Yes or No replacing them with 1 and 0 correspondingly. If it contains more than 2 unique values encoding with the label encoder   where it that a unique integer to each category



With this transformed all the object datatype into into numerical datatype.

# **Step 4: CORELATION**

A statistical tool that helps in the study of the relationship between two variables is known as **Correlation.**It also helps in understanding the economic behaviour of the variables.

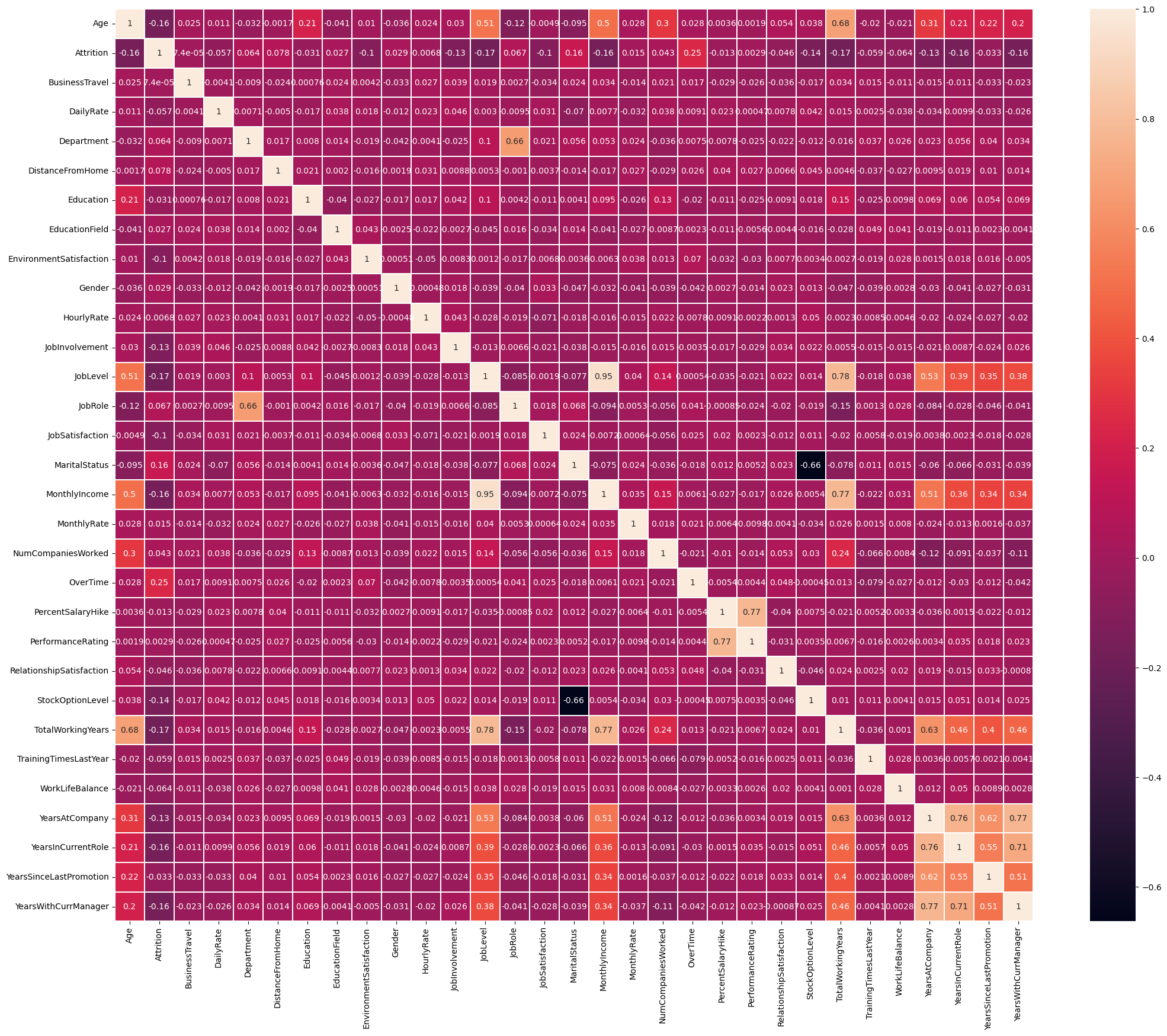
Positive corelation denotes that when two variables move in the same direction, i.e., when one increases the other also increases.

Negative corelation denotes that when two variable move in opposite direction, i.e, when one increases other decreases. Strong negative corelated feature to target variable can degrade the performance so we must identify them,

df.corr() – to know the measure of corelation between the features.

A white rectangular object with black text

Description automatically generated



From this generated heatmap we can observe these things:

\*Attrition column corelated with positively corelated column: Overtime(25%),MaritalStatus(16%)

Negatively corelated column:Age(16%),Environment satisfaction(10%)

All other columns are zero corelated column

\*Age column postively corelated with joblevel(51%),Monthlyincome(50%),TotalWorkingYears(68%),YearsAtCompany(31%),YearsInCurrentRole(21%),YearsSinceLAstPromotion(22%)yearswithcurrmanager(20%)

\*Department column postively corelated with Jobrole(66%)

\*JobLevel column postively corelated with Totalworkingyears(78%),YearsatCompany(53%),

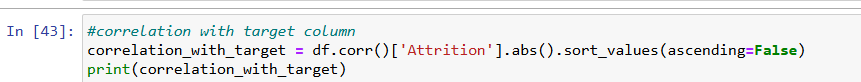
**MULTICOLINEARITY:**

Multicollinearity refers to a situation at some stage in which two or greater explanatory variables in the course of a multiple correlation model.

Years at company and yeartsatCurentrole has strong corelation-76%

Monthly income and joblevel has strong coreltion with 95%.

Yearsatcompany and YearsWithcurrManager with 77%.



From this code we can see the corelation value with target variable.

Attrition 1.000000

OverTime 0.246118

TotalWorkingYears 0.171063

JobLevel 0.169105

MaritalStatus 0.162070

YearsInCurrentRole 0.160545

MonthlyIncome 0.159840

Age 0.159205

YearsWithCurrManager 0.156199

StockOptionLevel 0.137145

YearsAtCompany 0.134392

JobInvolvement 0.130016

JobSatisfaction 0.103481

EnvironmentSatisfaction 0.103369

DistanceFromHome 0.077924

JobRole 0.067151

Department 0.063991

WorkLifeBalance 0.063939

TrainingTimesLastYear 0.059478

DailyRate 0.056652

RelationshipSatisfaction 0.045872

NumCompaniesWorked 0.043494

YearsSinceLastPromotion 0.033019

Education 0.031373

Gender 0.029453

EducationField 0.026846

MonthlyRate 0.015170

PercentSalaryHike 0.013478

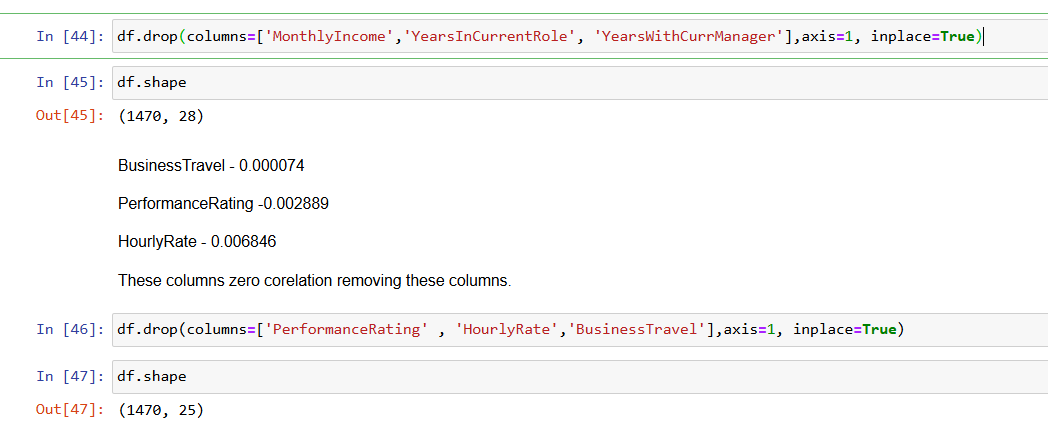
HourlyRate 0.006846

PerformanceRating 0.002889

BusinessTravel 0.000074

Name: Attrition, dtype: float64

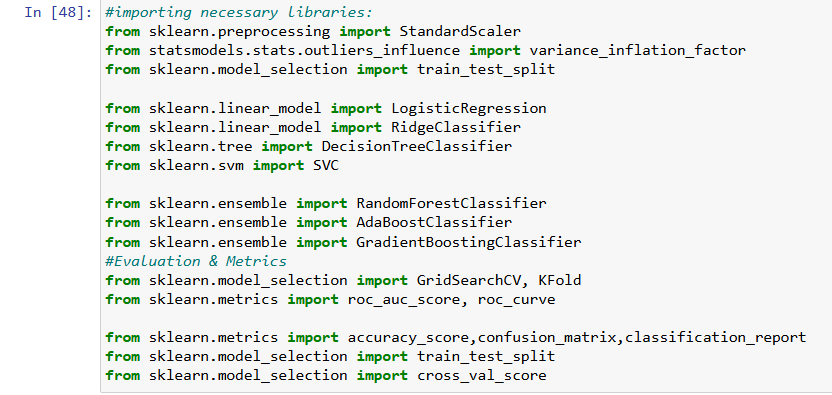
From these above method we can see that these - yearsatcompany,Monthlyincome,yearsatcurrentrole,joblevel,yearswithcurrManager ['MonthlyIncome','YearsInCurrentRole', 'YearsWithCurrManager'] - has negative corelation to the target variable so removing these columns.



Finally after analysing corelation we have 25 columns and 1470 rows for the

Model building.

**Step 5: SPLITTING THE COLUMNS INTO FEATURE AND TARGET**

Before that importing the necessary libraries for the model building.

Splitting the columns into features and target in this specific dataset df[“Attrition] is the target feature stored in the variable Y. All other column are considered as the independent features and they are stored in the variable X.

A white background with red and blue text

Description automatically generated

**Step 6 :Standardization:**

It is the transformation of features by subtracting from mean and dividing by standard deviation. This is often called as Z-score.

Standardization can be helpful in cases where the data follows a Gaussian distribution.

 It translates the data to the mean vector of original data to the origin and squishes or expands the points if std is 1 respectively.

We can see that we are just changing mean and standard deviation to a standard normal distribution which is still normal thus the shape of the distribution is not affected.

Standardization does not get affected by outliers because there is no predefined range of transformed features.

A screenshot of a computer code

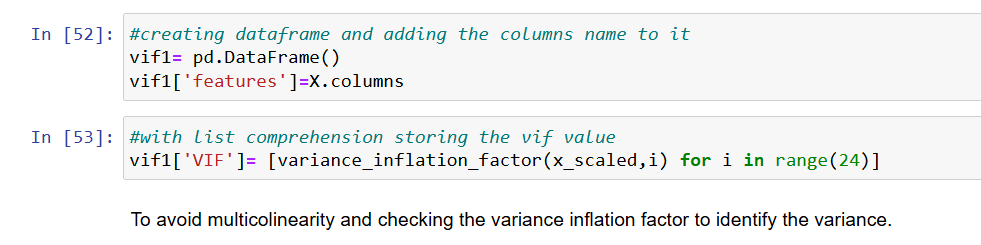
Description automatically generated

**Step: 7 Variance inflation factor**

Multicollinearity occurs when there are two or more independent variables in a multiple regression model, which have a high correlation among themselves. When some features are highly correlated,

we might have difficulty in distinguishing between their individual effects on the dependent variable.

Multicollinearity can be detected using various techniques, one such technique being the Variance Inflation Factor(VIF).



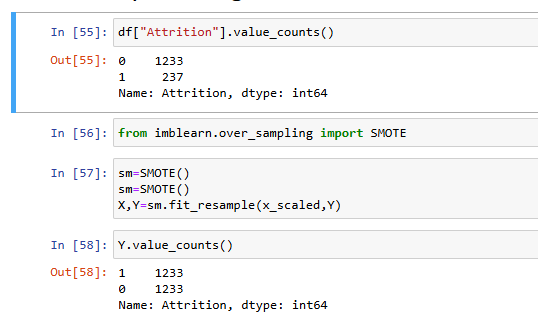
A screenshot of a computer

Description automatically generated

A high VIF (typically above 5 or 10) indicates problematic multicollinearity.

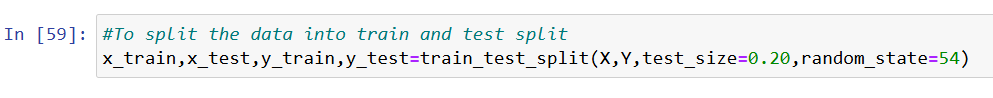
Most of the columns vif values are below 5. Further proceeding the dataset.

**Step 8: Handling Imbalanced Data**

****

Oversampling: Increase the number of instances in the minority class by duplicating or generating synthetic samples. Techniques like SMOTE (Synthetic Minority Over-sampling Technique). Now the dataset looks balanced

**Step 9: Train and test splitting**

****

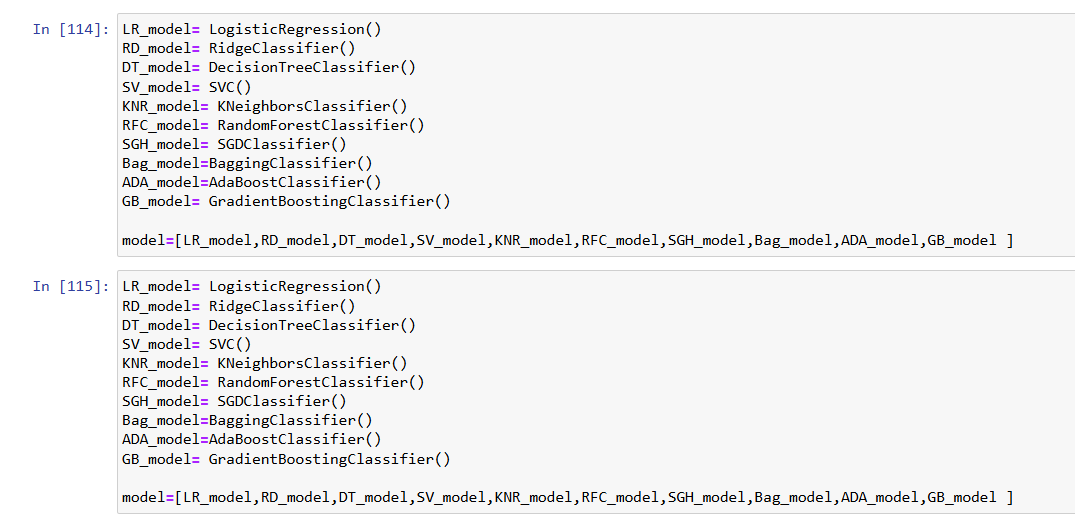
The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

**Step 10: Model Selection**

**Importing the libraries and iterating through the model:**

**A screen shot of a computer program

Description automatically generated**

****

**A computer code with black and red text

Description automatically generated**

**Output:**

Accuracy\_Score of LogisticRegression() is 80.76923076923077

Confusion Matrix of LogisticRegression() is

[[209 46]

[ 49 190]]

precision recall f1-score support

0 0.81 0.82 0.81 255

1 0.81 0.79 0.80 239

accuracy 0.81 494

macro avg 0.81 0.81 0.81 494

weighted avg 0.81 0.81 0.81 494

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy\_Score of RidgeClassifier() is 80.97165991902834

Confusion Matrix of RidgeClassifier() is

[[207 48]

[ 46 193]]

precision recall f1-score support

0 0.82 0.81 0.81 255

1 0.80 0.81 0.80 239

accuracy 0.81 494

macro avg 0.81 0.81 0.81 494

weighted avg 0.81 0.81 0.81 494

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy\_Score of DecisionTreeClassifier() is 87.04453441295547

Confusion Matrix of DecisionTreeClassifier() is

[[218 37]

[ 27 212]]

precision recall f1-score support

0 0.89 0.85 0.87 255

1 0.85 0.89 0.87 239

accuracy 0.87 494

macro avg 0.87 0.87 0.87 494

weighted avg 0.87 0.87 0.87 494

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy\_Score of SVC() is 92.3076923076923

Confusion Matrix of SVC() is

[[238 17]

[ 21 218]]

precision recall f1-score support

0 0.92 0.93 0.93 255

1 0.93 0.91 0.92 239

accuracy 0.92 494

macro avg 0.92 0.92 0.92 494

weighted avg 0.92 0.92 0.92 494

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy\_Score of KNeighborsClassifier() is 84.21052631578947

Confusion Matrix of KNeighborsClassifier() is

[[183 72]

[ 6 233]]

precision recall f1-score support

0 0.97 0.72 0.82 255

1 0.76 0.97 0.86 239

accuracy 0.84 494

macro avg 0.87 0.85 0.84 494

weighted avg 0.87 0.84 0.84 494

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy\_Score of RandomForestClassifier() is 93.7246963562753

Confusion Matrix of RandomForestClassifier() is

[[249 6]

[ 25 214]]

precision recall f1-score support

0 0.91 0.98 0.94 255

1 0.97 0.90 0.93 239

accuracy 0.94 494

macro avg 0.94 0.94 0.94 494

weighted avg 0.94 0.94 0.94 494

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy\_Score of SGDClassifier() is 76.31578947368422

Confusion Matrix of SGDClassifier() is

[[186 69]

[ 48 191]]

precision recall f1-score support

0 0.79 0.73 0.76 255

1 0.73 0.80 0.77 239

accuracy 0.76 494

macro avg 0.76 0.76 0.76 494

weighted avg 0.77 0.76 0.76 494

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy\_Score of BaggingClassifier() is 89.27125506072875

Confusion Matrix of BaggingClassifier() is

[[244 11]

[ 42 197]]

precision recall f1-score support

0 0.85 0.96 0.90 255

1 0.95 0.82 0.88 239

accuracy 0.89 494

macro avg 0.90 0.89 0.89 494

weighted avg 0.90 0.89 0.89 494

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy\_Score of AdaBoostClassifier() is 91.09311740890689

Confusion Matrix of AdaBoostClassifier() is

[[233 22]

[ 22 217]]

precision recall f1-score support

0 0.91 0.91 0.91 255

1 0.91 0.91 0.91 239

accuracy 0.91 494

macro avg 0.91 0.91 0.91 494

weighted avg 0.91 0.91 0.91 494

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accuracy\_Score of GradientBoostingClassifier() is 92.71255060728745

Confusion Matrix of GradientBoostingClassifier() is

[[248 7]

[ 29 210]]

precision recall f1-score support

0 0.90 0.97 0.93 255

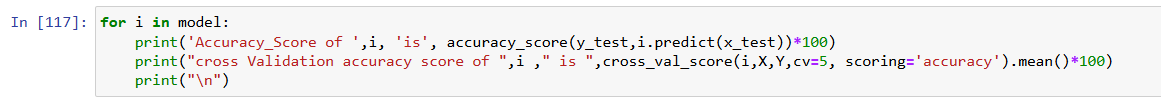
1 0.97 0.88 0.92 239

accuracy 0.93 494

macro avg 0.93 0.93 0.93 494

weighted avg 0.93 0.93 0.93 494

Cross validation:



Accuracy\_Score of LogisticRegression() is 80.76923076923077

cross Validation accuracy score of LogisticRegression() is 78.66979822782108

Accuracy\_Score of RidgeClassifier() is 80.97165991902834

cross Validation accuracy score of RidgeClassifier() is 78.58890868926099

Accuracy\_Score of DecisionTreeClassifier() is 87.04453441295547

cross Validation accuracy score of DecisionTreeClassifier() is 84.31145346593196

Accuracy\_Score of SVC() is 92.3076923076923

cross Validation accuracy score of SVC() is 90.39032281906202

Accuracy\_Score of KNeighborsClassifier() is 84.21052631578947

cross Validation accuracy score of KNeighborsClassifier() is 82.19830665757858

Accuracy\_Score of RandomForestClassifier() is 93.7246963562753

cross Validation accuracy score of RandomForestClassifier() is 93.2745891879019

Accuracy\_Score of SGDClassifier() is 76.31578947368422

cross Validation accuracy score of SGDClassifier() is 74.12774798597367

Accuracy\_Score of BaggingClassifier() is 89.27125506072875

cross Validation accuracy score of BaggingClassifier() is 90.3559960910233

Accuracy\_Score of AdaBoostClassifier() is 91.09311740890689

cross Validation accuracy score of AdaBoostClassifier() is 86.4666464100648

Accuracy\_Score of GradientBoostingClassifier() is 92.71255060728745

cross Validation accuracy score of GradientBoostingClassifier() is 88.98235211996288

**Top model 1:**

Accuracy\_Score of RandomForestClassifier() is 93.7246963562753

cross Validation accuracy score of RandomForestClassifier() is 93.2745891879019

**Top model 2:**

Accuracy\_Score of BaggingClassifier() is 89.27125506072875

cross Validation accuracy score of BaggingClassifier() is 90.3559960910233

**Step 11 :Hyperparameter tuning**

 Hyperparameters are configuration variables that are set before the training process of a model begins.

They control the learning process itself, rather than being learned from the data.

Hyperparameters are often used to tune the performance of a model, and they can have a significant impact on the model’s accuracy, generalization, and other metrics.

A screenshot of a computer code

Description automatically generated

Accuracy increased in random forest classifier finalising the model.

A balance between accuracy and interpretability is needed, the Random Forest Classifier remains a strong model.

RANDOM FOREST CLASSIFIER FOR THE HRANALYTICS PROJECT.

**STEP 12 :Evaluating the model**

For the classification model:

1. Accuracy: the ratio of the number of correct predictions and the total number of predictions.
2. Confusion Matrix: Confusion Matrix is a performance measurement for the machine learning classification problems where the output can be two or more classes. It is a table with combinations of predicted and actual values.
3. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes. From the graph, we simply say the area of the curve ABDE and the X and Y-axis.
4. F1 score: It gives a combined idea about Precision and Recall metrics. It is maximum when Precision is equal to Recall.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

With this Saving the model

A screenshot of a computer code

Description automatically generated