**Project Proposal**

GROUP-16

Employee Attrition Rate using Machine learning

Presented By

Akhila Beemreddy

Neeraja Golla

Fayaz Kulumala

Sai Kiran Reddy Bandi

Spandana Bandaru



Department of Data Science

University of North Texas, Denton, Texas

**Project Roles and Responsibilities**

**Akhila Beemreddy(S.Id-11637565)**

akhilabeemreddy@my.unt.edu

Major: Data Science

**Role**: Python programming, Domain understanding &

Feature Engineering, Exploratory Data Analysis, Model Training & Model Evaluation

**Neeraja Golla(S.Id-11637137)**

neerajagolla@my.unt.edu

Major: Data Science

**Role:** Python Programming, Data pre-processing, Model prediction & Model Evaluation

**Fayaz Kulumala(S.Id-11631146)**

kulumalafayazkulumala@my.unt.edu

Major : Data Science

**Role:** Code Development, Project Documentation, Algorithm selection & Model prediction.

**Sai Kiran Reddy Bandi(S.Id-11651865)**

saikiranreddybandi@my.unt.edu

Major : Data Science

**Role :** Dataset selection, Data pre-processing, Model Training & Model Evaluation.

**Spandana Bandaru(S.Id-11517882)**

spandanabandaru@my.unt.edu

**Role:** Code Development, Project Documentation, References from existing projects

**Workflow:**

Our team meets during after class hours and when everyone are available on campus to discuss the ideas, progress and the tasks that we need to work on. We had a meet in Microsoft teams for online team communication to share our work and make changes in any code for documentation related to the project.

The workflow for employee attrition rate prediction using a machine learning model involves the following steps:

**1)Data Collection:** Gather data on employee demographics, job history, performance metrics, and other relevant factors that can contribute to employee attrition. This data can be collected from various sources, such as HR records, employee surveys, and performance evaluations.

**2)Data Preprocessing:** This involves cleaning the data, removing missing values, and transforming the data into a format suitable for analysis. You may also need to perform feature engineering to create new features that can better predict employee attrition.

**3)Data Exploration:** Perform exploratory data analysis to understand the distribution of the data, identify any outliers or anomalies, and visualize the relationships between different features.

**4)Feature Selection:** Use statistical techniques or machine learning algorithms to select the most important features that are most predictive of employee attrition.

**5)Model Selection:** Choose a machine learning algorithm that is suitable for the problem at hand, such as logistic regression, decision trees, random forests, or neural networks. You may need to experiment with different algorithms and hyperparameters to find the best model.

**6)Model Training:** Split the data into training and testing sets, and train the model on the training set using the selected algorithm. Use techniques such as cross-validation to tune the model and avoid overfitting.

**7)Model Evaluation:** Evaluate the performance of the model on the testing set using metrics such as accuracy, precision, recall, and F1-score. You can also use techniques such as ROC curves to visualize the performance of the model.

**8)Model Deployment:** Deploy the model into production by integrating it with your HR systems and using it to predict employee attrition in real-time. You may need to update the model periodically as new data becomes available.

**9)Model Monitoring:** Monitor the performance of the model over time and retrain or update the model as needed to maintain its accuracy and relevance.

By following this workflow, you can develop a robust machine learning model that can help you predict and mitigate employee attrition.

**ABSTRACT:**

The issue of employee attrition is a major concern for organizations across industries and geographies. To address this challenge, many companies are turning to machine learning classification models to predict whether an employee is likely to quit. These models can help human resources departments intervene in a timely manner and possibly provide remedies to prevent attrition.

In this project, we develop a model to predict employee attrition rates using IBM Attrition Data provided by Kaggle. We use the different techniques to make predictions, and introduce the factors that influence employee attrition rates within organizations. By providing insights into these factors, our study aims to help top management make informed decisions regarding the retention of their workforce.

We have implemented the logistic regression algorithm in this project to predict the employee attrition rate. The reason for opting this model is that Whether an employee is going to stay or leave a company, his or her answer is just binomial i.e. it can be “YES” or “NO”. So, we can see our dependent variable Employee Attrition is just a categorical variable. In the case of a dependent categorical variable, we can not use linear regression, in that case, we have to use “**LOGISTIC REGRESSION**“. So, we have used this model for better accuracy.

The study utilizes the IBM Human Resource Analytic Employee Attrition and Performance dataset, which contains a range of demographic, employment, and performance-related variables for employees. By comparing the performance of these machine learning models, the study aims to identify the most effective approach to predicting employee attrition and help organizations develop better retention strategies.

**Problem Statement:**

Employee attrition is a major cost to an organization and predicting such attritions is the most important requirement of the human resources department in many organizations. In this problem, your task is to predict the attrition rate of employees of an organization.

**Data specification:**

Data set includes 13 features with 1470 rows of employee data for IBM HR analytical employee attrition and performance. At the end of the analysis, we will try to model the employee attrition by predicting the possible attritions using specific machine learning models.

The features are:

1. Age
2. Attrition
3. Department
4. DistanceFromHome
5. Education
6. EducationField
7. EnvironmentSatisfaction
8. JobSatisfaction
9. MaritalStatus
10. MonthlyIncome
11. NumCompaniesWorked
12. WorkLifeBalance
13. YearsAtCompany

**Project Design:**

**Data Collection**: The dataset utilized for this project is IBM Human Resource Analytic Employee Attrition and Performance datasetprovided by **Kaggle.**

Analyzing the dataset using pandas, NumPy, and seaborn: on the selected dataset use pandas to load and clean the data, NumPy to perform calculations and seaborn to create visualizations that help reveal insights and patterns.

Build a machine learning model using scikit-learn: using scikit-learn to train a machine learning model. Use NumPy for data manipulation. Visualize the model's performance using matplotlib and seaborn.

Create a dashboard using matplotlib and seaborn: Use matplotlib and seaborn to create interactive visualizations and plots that can be used to create an informative dashboard. Use pandas and NumPy for data manipulation and cleaning.

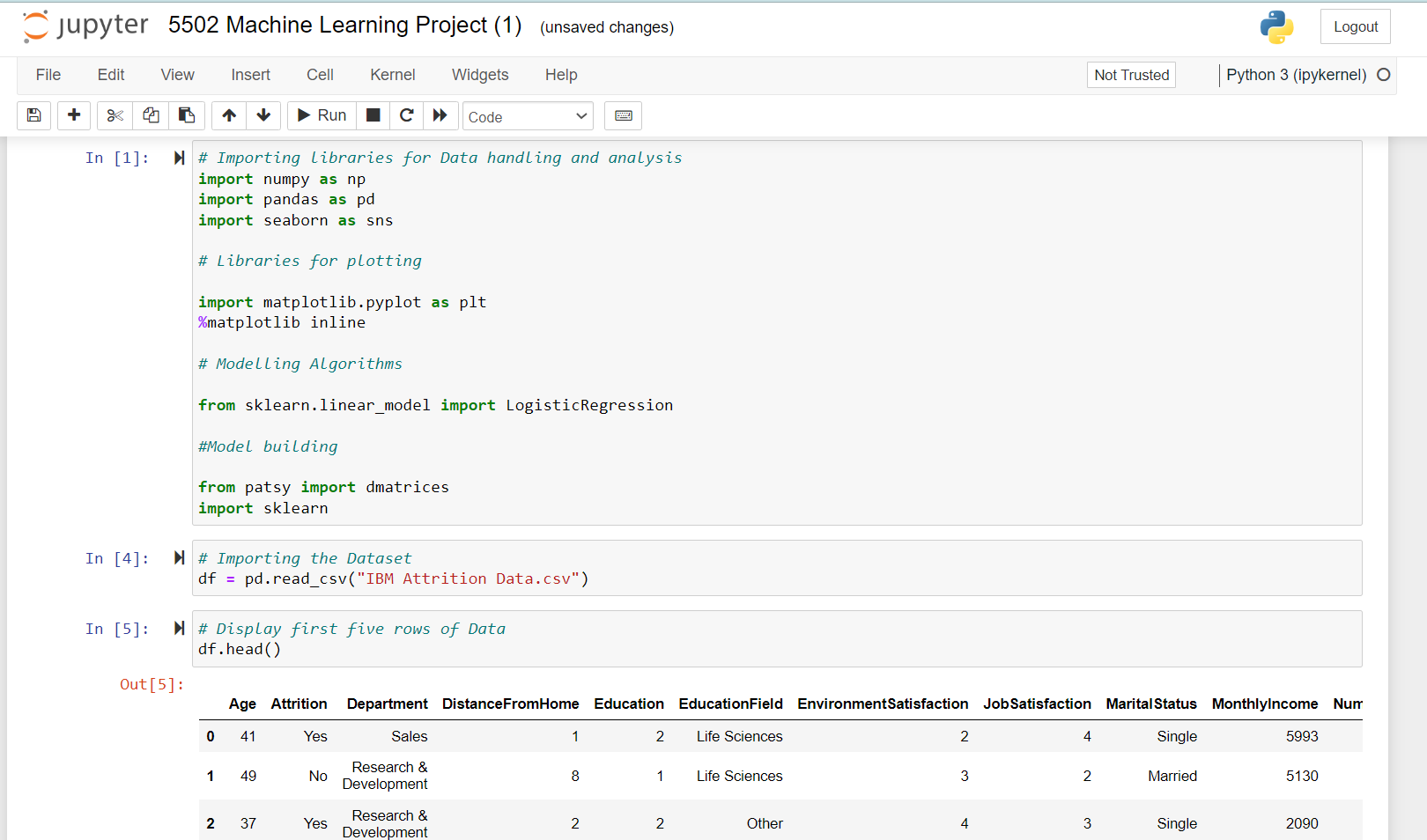
Conduct exploratory data analysis using pandas and seaborn: Loading the dataset into pandas, clean the data, and use seaborn to create visualizations that help reveal insights and patterns. Use NumPy for calculations.

In this project we have implemented Logistic regression which is a type of generalized linear model that uses a logistic function to model the relationship between the predictor variables and the response variable.

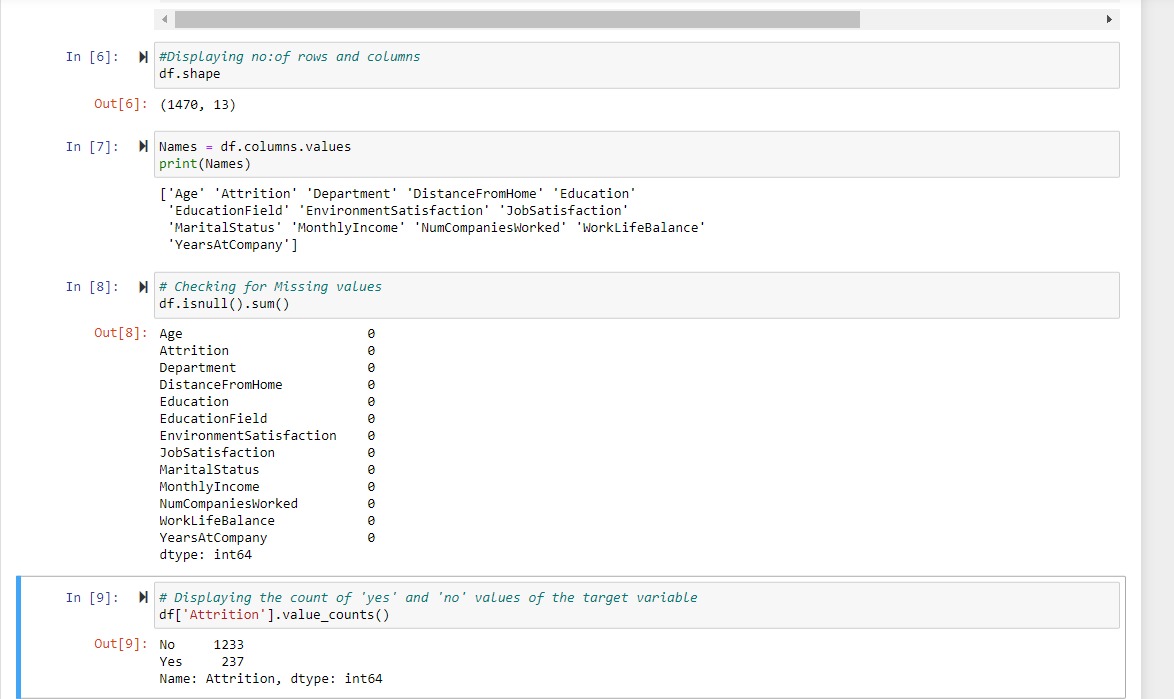
The most important functions and/or methods in our project include the following:

* read\_csv(): This function is used to read in the data from a CSV file and store it in a Pandas DataFrame.
* replace(): This method is used to replace any datafield value with another.
* Dmatrices(): is a function used for preparing data for statistical modeling by constructing design matrices from a formula string and a pandas dataframe.
* corr() is a function used to calculate the pairwise correlation of columns in a pandas dataframe.
* np.ravel(): is a NumPy function used to flatten a multi-dimensional array into a one-dimensional array by concatenating all elements of the array in row-major order.
* np.abs() is a NumPy function used to calculate the absolute value of elements in a given array, element-wise.
* Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome, with the outcome being a binary variable
* train\_test\_split(): This function is used to split the data into training and testing sets.
* fit(): This method is used to train the model on the training data.
* predict(): This method is used to predict the value of y based on the testing data.

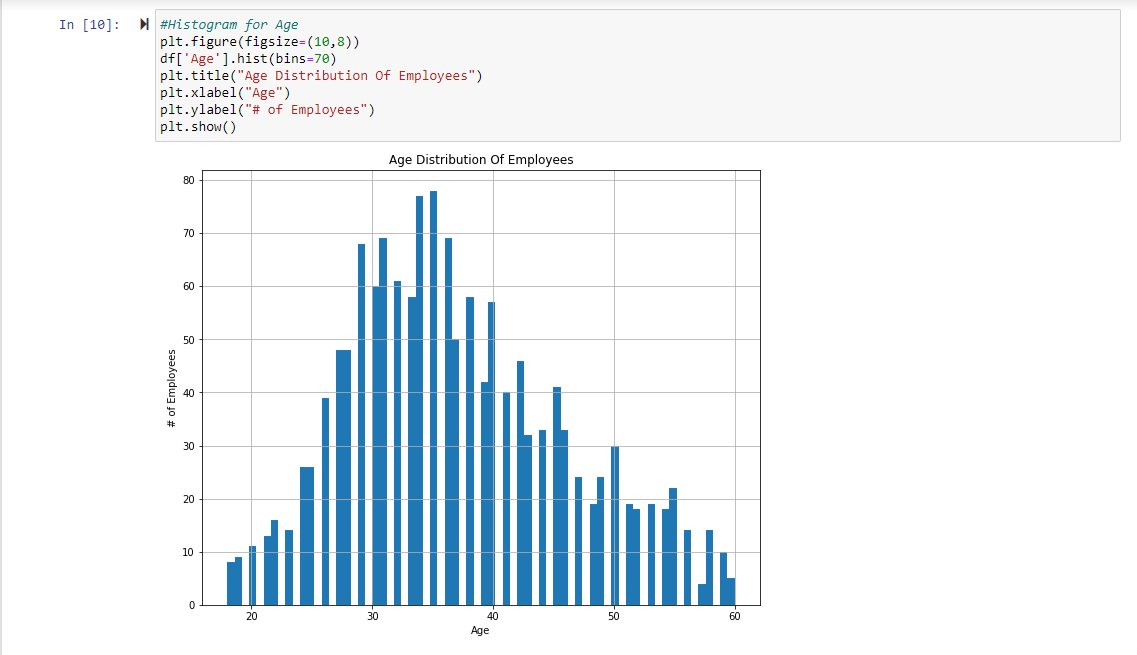
1. Importing libraries such as NumPy, pandas, seaborn, matplotlib, sklearn, and logistic regression. Read data through pandas and created a dataframe and fetches top 5 records using head method.

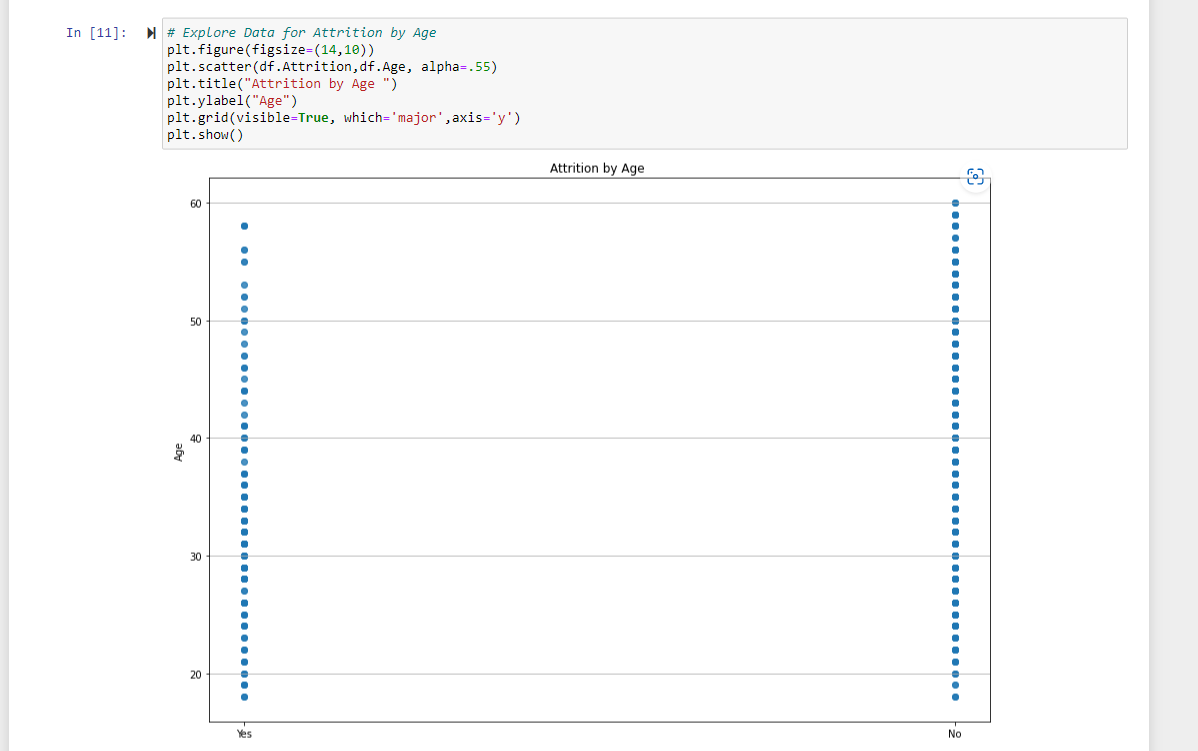


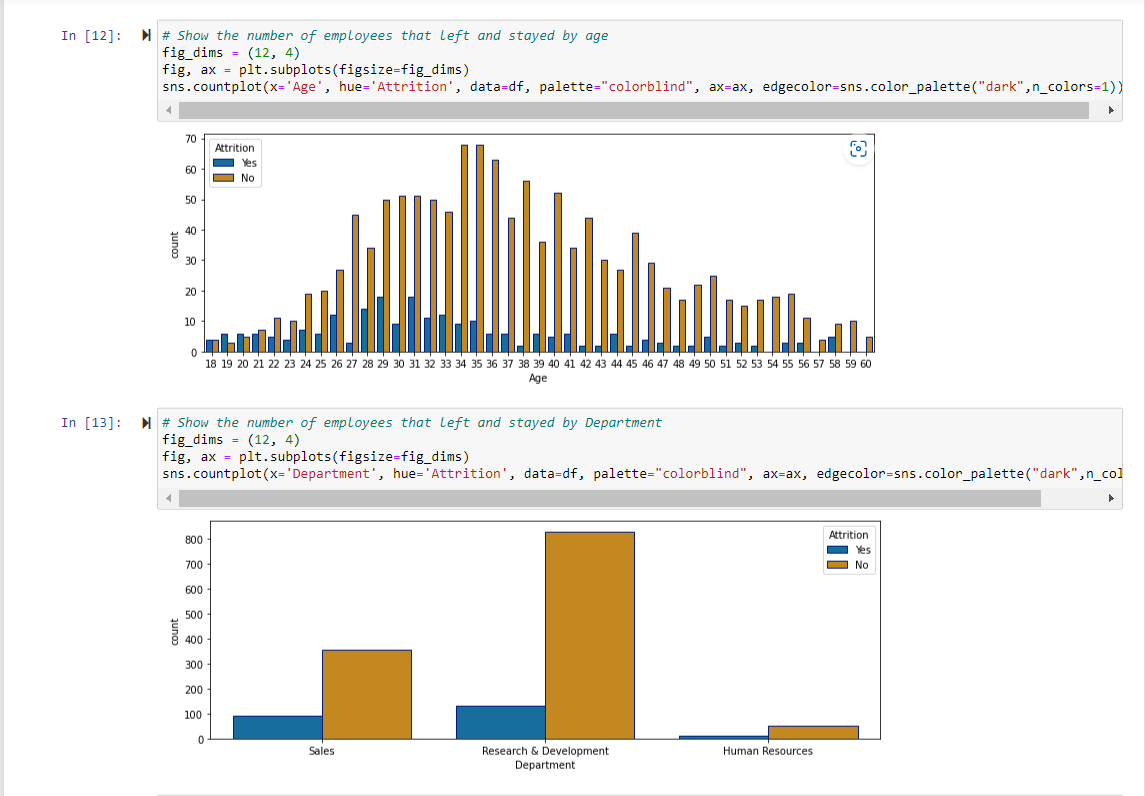
1. Total number of records in our Dataset i.e IBM Attrition Data.csv contains 1470 records and 13 features(columns)
2. Data Cleansing and missing data values within the columns is checked through dataframe



1. Comparing all features with attrition rate and drawn multiple graphs



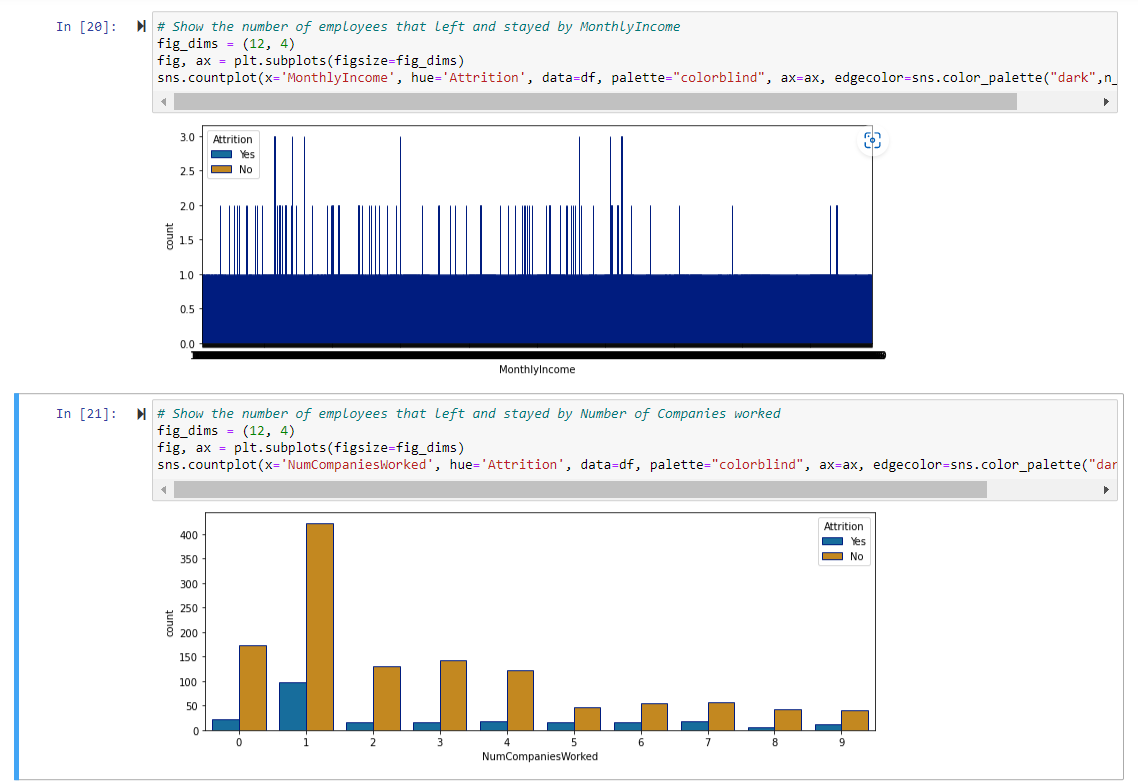




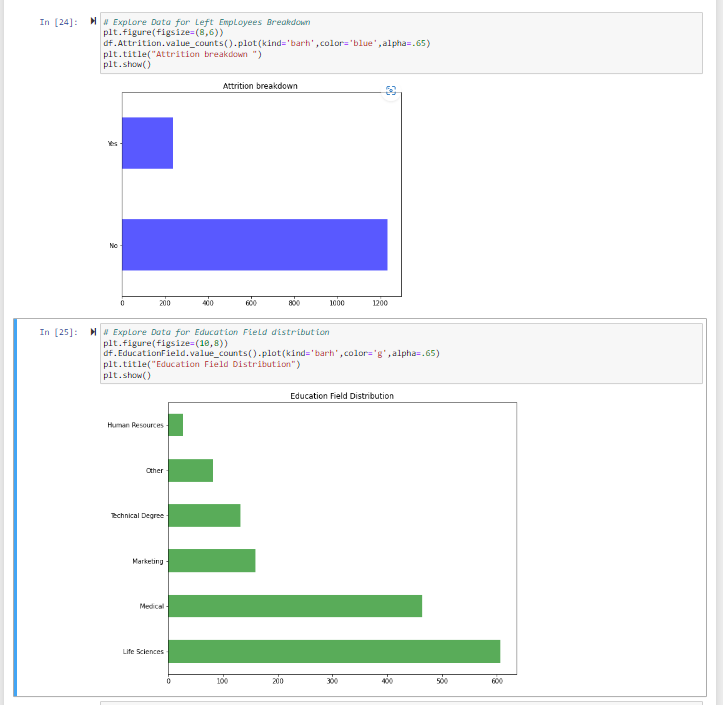


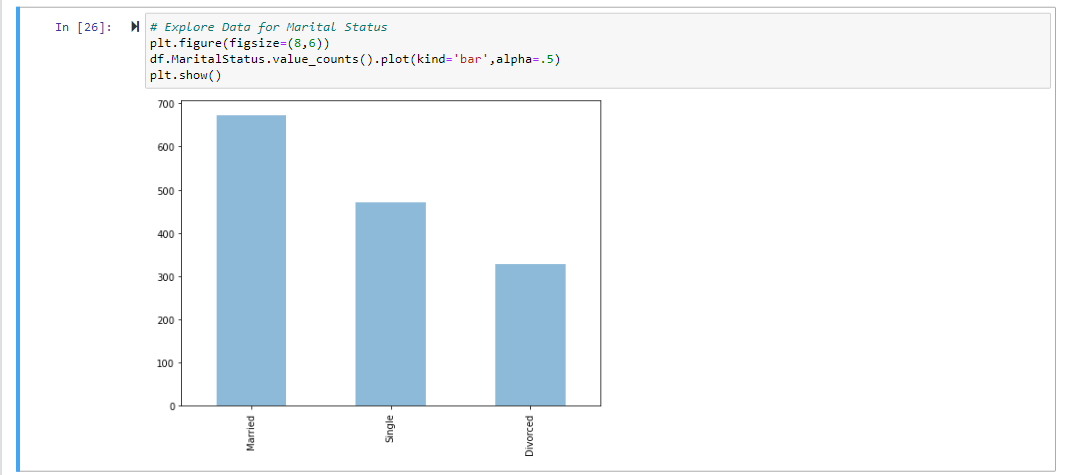




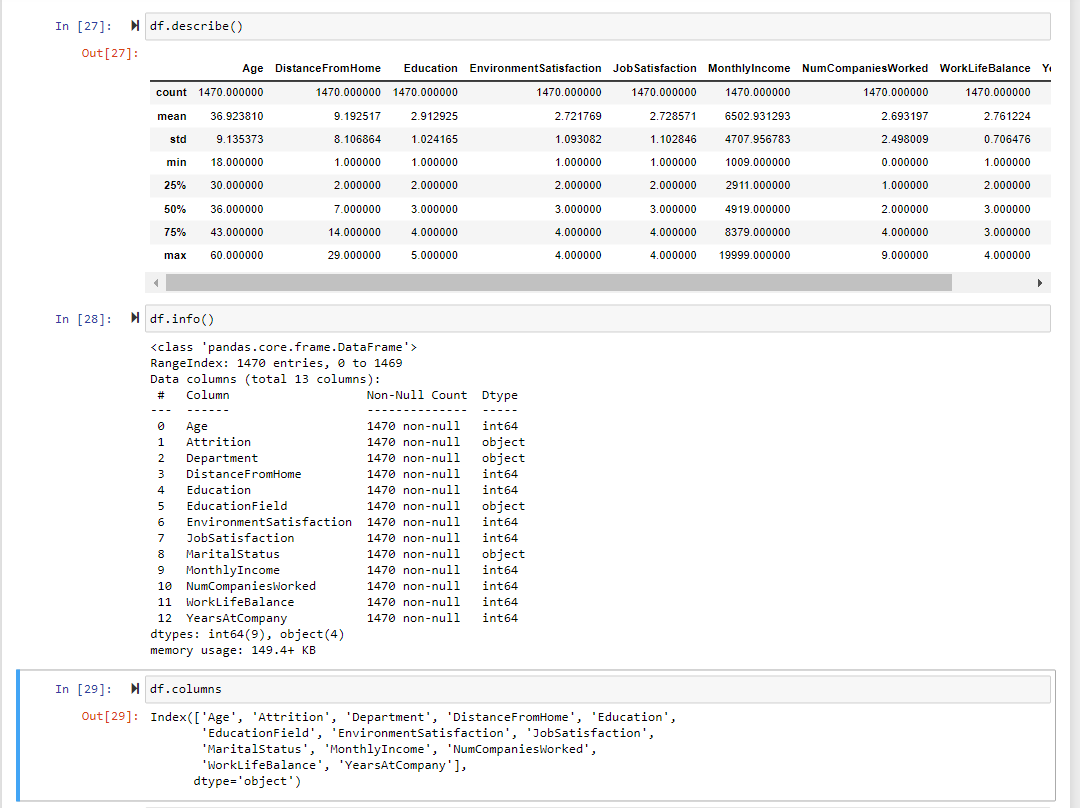




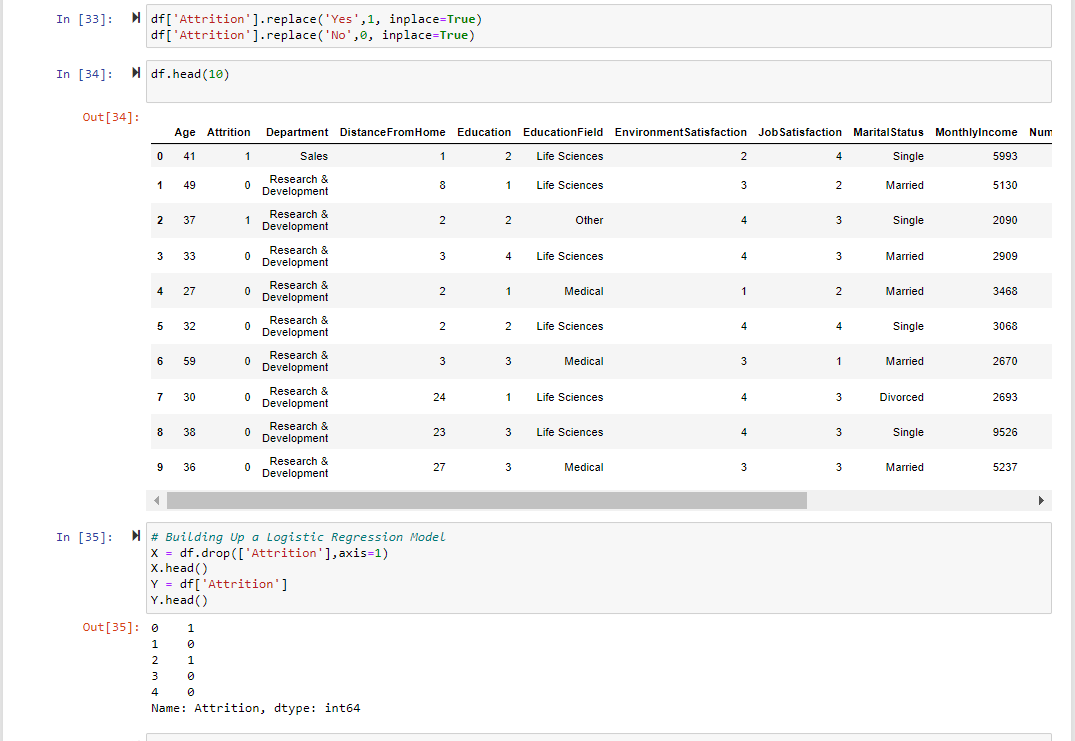


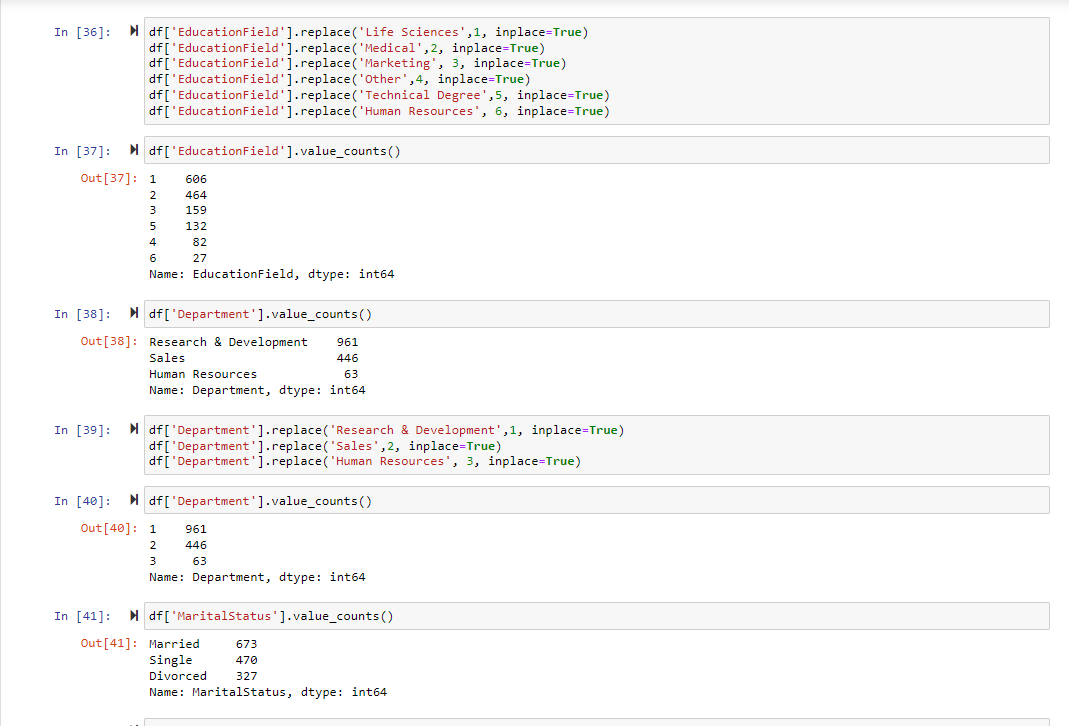


Calculating different statistical measures for all columns using describe method



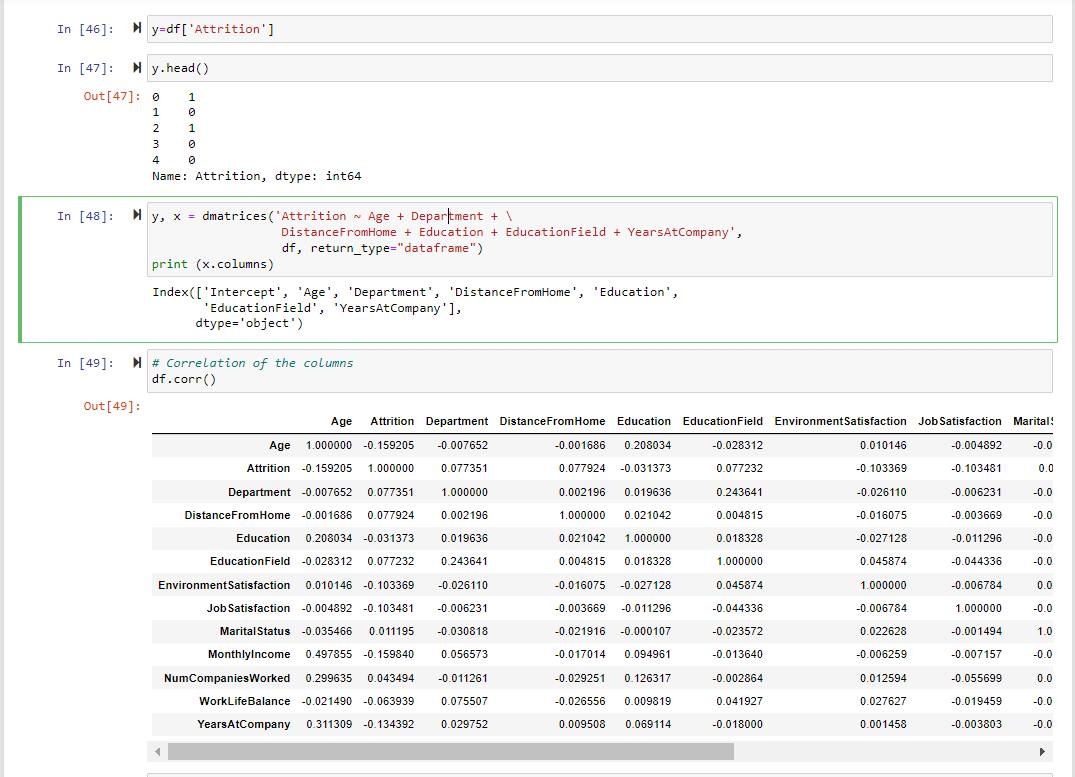
Converting all columns datatypes to integers using replace method

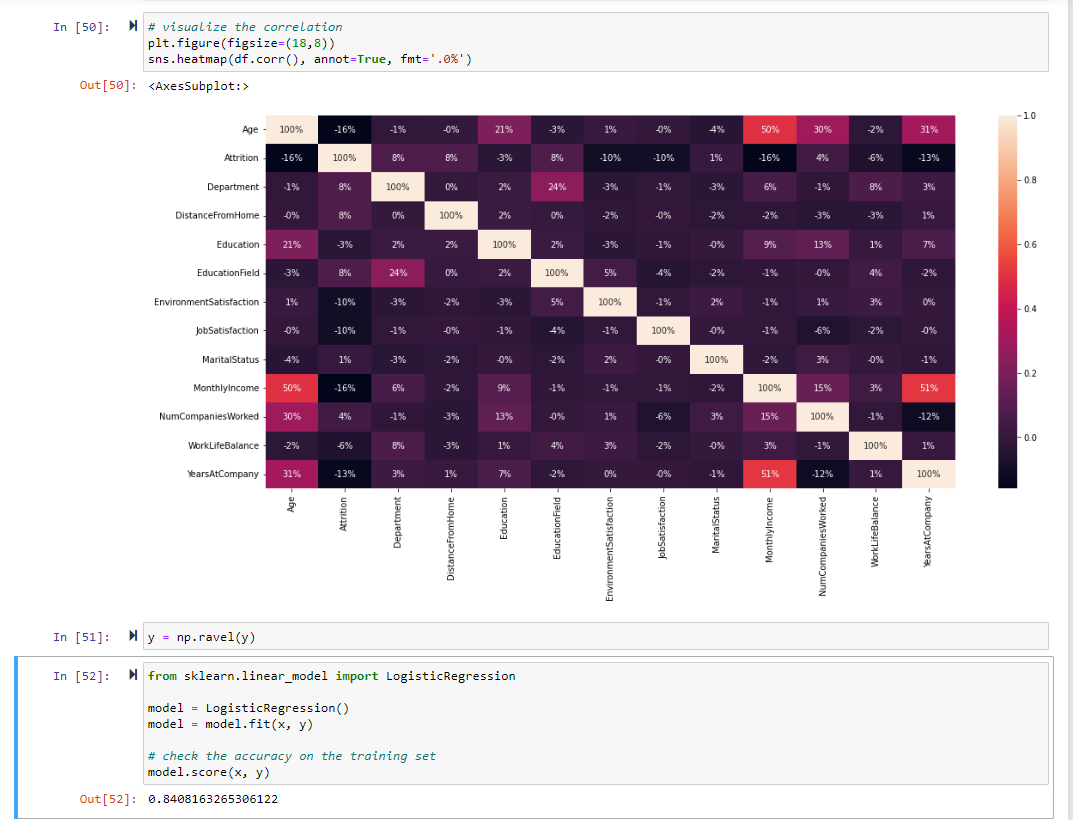




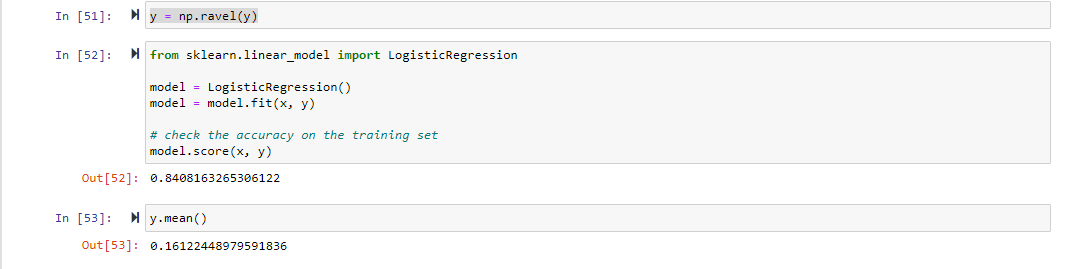


Creating Design matrix x and response vector y can be used to fit the statistical model: [2D array]

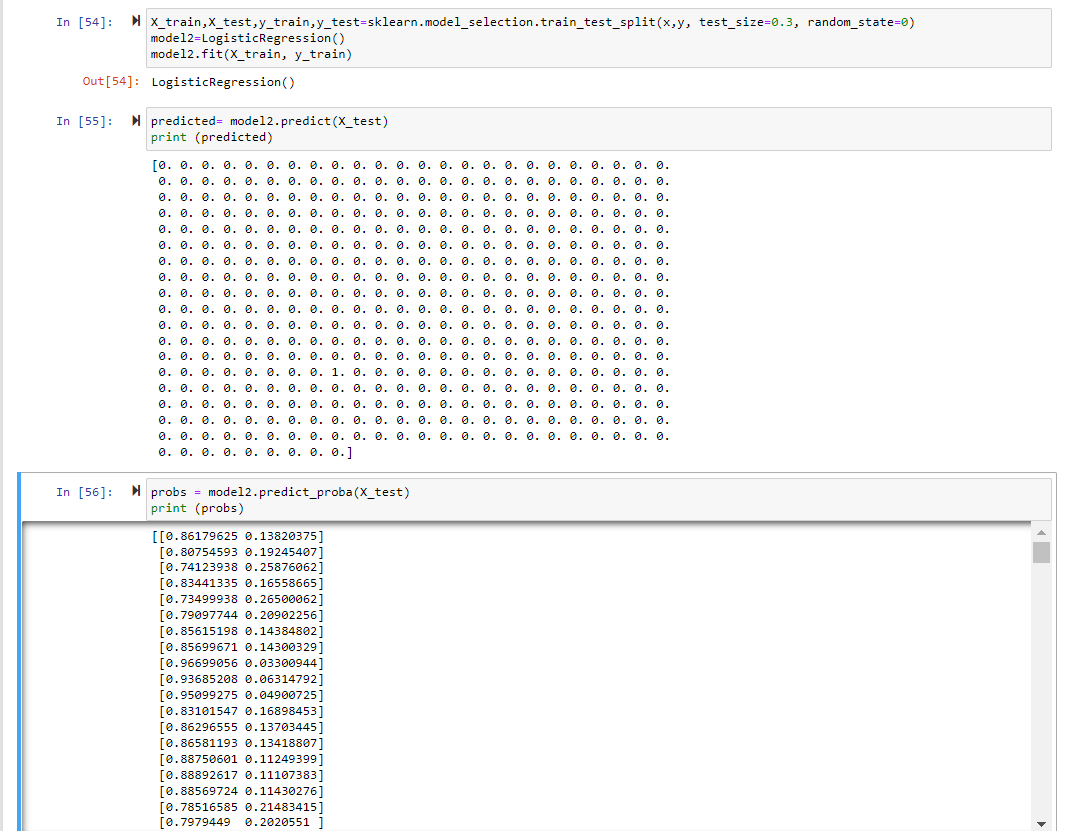




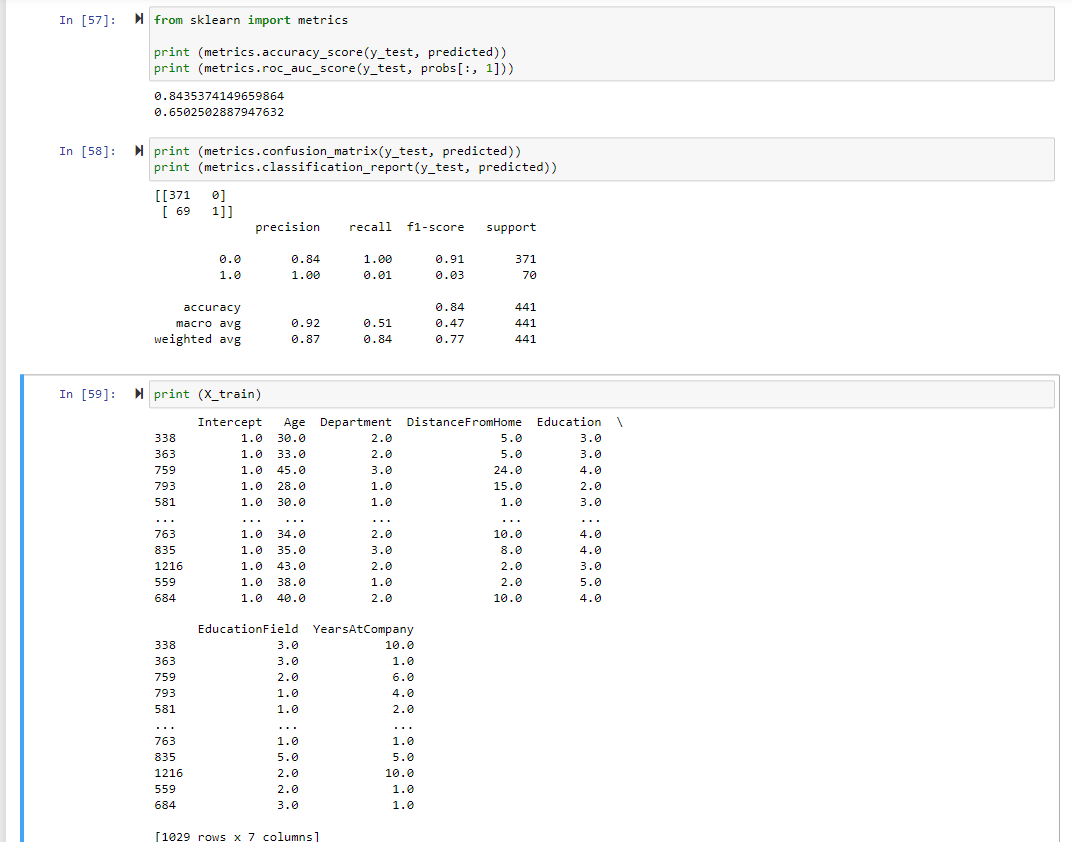
Converting 2D array into 1D array using ravel method and applied logistic regression:



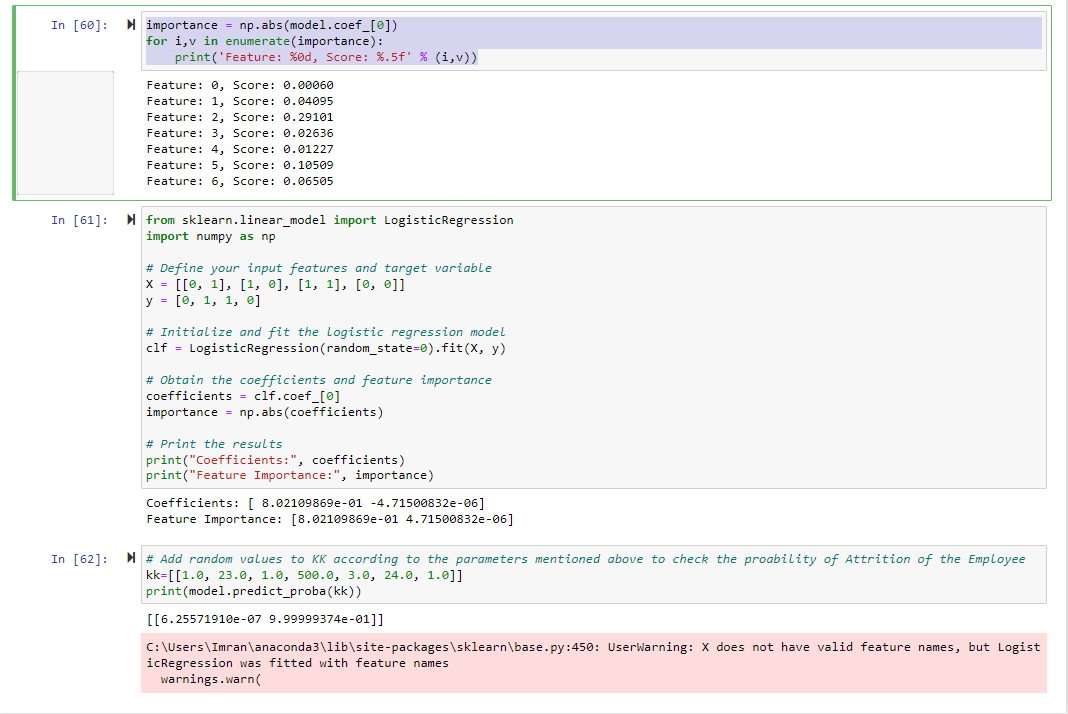
Train the model on test data



Model prediction on test data



Features classification with linear coefficients



**Project Results:**

In the output, the first value represents the accuracy score, which is 0.8435, indicating that the model predicted the true labels with an accuracy of approximately 84%. The second value represents the ROC-AUC score, which is 0.6502, indicating that the model is moderately good at distinguishing between positive and negative classes.

**Repository/Archive:**

Our project is available on Github, where it may be readily collaborated with in terms and by individuals.

https://github.com/Fayazkulumala332/5502\_Project/upload

**Appendix – Code:**

# Importing libraries for Data handling and analysis

import numpy as np

import pandas as pd

import seaborn as sns

# Libraries for plotting

import matplotlib.pyplot as plt

%matplotlib inline

# Modelling Algorithms

from sklearn.linear\_model import LogisticRegression

#Model building

from patsy import dmatrices

import sklearn

# Importing the Dataset

df = pd.read\_csv("IBM Attrition Data.csv")

# Display first five rows of Data

df.head()

#Displaying no:of rows and columns

df.shape

Names = df.columns.values

print(Names)

# Checking for Missing values

df.isnull().sum()

# Displaying the count of 'yes' and 'no' values of the target variable

df['Attrition'].value\_counts()

#Histogram for Age

plt.figure(figsize=(10,8))

df['Age'].hist(bins=70)

plt.title("Age Distribution Of Employees")

plt.xlabel("Age")

plt.ylabel("# of Employees")

plt.show()

# Explore Data for Attrition by Age

plt.figure(figsize=(14,10))

plt.scatter(df.Attrition,df.Age, alpha=.55)

plt.title("Attrition by Age ")

plt.ylabel("Age")

plt.grid(visible=True, which='major',axis='y')

plt.show()

# Show the number of employees that left and stayed by Department

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='Department',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by DistanceFromHome

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='DistanceFromHome',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by Education

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='Education',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by EducationField

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='EducationField',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by EnvironmentSatisfaction

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='EnvironmentSatisfaction',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by JobSatisfaction

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='JobSatisfaction',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by MaritalStatus

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='MaritalStatus',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by MonthlyIncome

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='MonthlyIncome',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by Number of Companies worked

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='NumCompaniesWorked',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by Work life balance

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='WorkLifeBalance',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Show the number of employees that left and stayed by Years worked at company

fig\_dims = (12, 4)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.countplot(x='YearsAtCompany',hue='Attrition',data=df,palette="colorblind",ax=ax, edgecolor=sns.color\_palette("dark",n\_colors=1));

# Explore Data for left Employees Breakdown

plt.figure(figsize=(8,6))

df.Attrition.value\_counts().plot(kind='barh',color='blue',alpha=.65)

plt.title("Attrition breakdown ")

plt.show()

# Explore Data for Education Field distribution

plt.figure(figsize=(10,8))

df.EducationField.value\_counts().plot(kind='barh',color='g',alpha=.65)

plt.title("Education Field Distribution")

plt.show()

# Explore Data for Marital Status

plt.figure(figsize=(8,6))

df.MaritalStatus.value\_counts().plot(kind='bar',alpha=.5)

plt.show()

df.describe()

df.columns

df.std()

df['Attrition'].value\_counts()

df['Attrition'].dtypes

df['Attrition'].replace('Yes',1, inplace=True)

df['Attrition'].replace('No',0, inplace=True)

# Building Up a Logistic Regression Model

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

X.head()

Y = df['Attrition']

Y.head()

df['EducationField'].replace('Life Sciences',1, inplace=True)

df['EducationField'].replace('Medical',2, inplace=True)

df['EducationField'].replace('Marketing', 3, inplace=True)

df['EducationField'].replace('Other',4, inplace=True)

df['EducationField'].replace('Technical Degree',5, inplace=True)

df['EducationField'].replace('Human Resources', 6, inplace=True)

df['EducationField'].value\_counts()

df['Department'].value\_counts()

df['Department'].replace('Research & Development',1, inplace=True)

df['Department'].replace('Sales',2, inplace=True)

df['Department'].replace('Human Resources', 3, inplace=True)

df['Department'].value\_counts()

df['MaritalStatus'].value\_counts()

df['MaritalStatus'].replace('Married',1, inplace=True)

df['MaritalStatus'].replace('Single',2, inplace=True)

df['MaritalStatus'].replace('Divorced',3, inplace=True)

df['MaritalStatus'].value\_counts()

x=df.select\_dtypes(include=['int64'])

x.dtypes

x.columns

y=df['Attrition']

y.head()

y, x = dmatrices('Attrition ~ Age + Department + \

DistanceFromHome + Education + EducationField + YearsAtCompany',

df, return\_type="dataframe")

print (x.columns)

# Correlation of the columns

df.corr()

# visualize the correlation

plt.figure(figsize=(18,8))

sns.heatmap(df.corr(), annot=True, fmt='.0%')

y = np.ravel(y)

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression()

model = model.fit(x, y)

# check the accuracy on the training set

model.score(x, y)

y.mean()

X\_train,X\_test,y\_train,y\_test=sklearn.model\_selection.train\_test\_split(x,y, test\_size=0.3, random\_state=0)

model2=LogisticRegression()

model2.fit(X\_train, y\_train)

predicted= model2.predict(X\_test)

print (predicted)

probs = model2.predict\_proba(X\_test)

print (probs)

from sklearn import metrics

print (metrics.accuracy\_score(y\_test, predicted))

print (metrics.roc\_auc\_score(y\_test, probs[:, 1]))

print (metrics.confusion\_matrix(y\_test, predicted))

print (metrics.classification\_report(y\_test, predicted))

print (X\_train)

importance = np.abs(model.coef\_[0])

for i,v in enumerate(importance):

print('Feature: %0d, Score: %.5f' % (i,v))

from sklearn.linear\_model import LogisticRegression

import numpy as np

# Define your input features and target variable

X = [[0, 1], [1, 0], [1, 1], [0, 0]]

y = [0, 1, 1, 0]

# Initialize and fit the logistic regression model

clf = LogisticRegression(random\_state=0).fit(X, y)

# Obtain the coefficients and feature importance

coefficients = clf.coef\_[0]

importance = np.abs(coefficients)

# Print the results

print("Coefficients:", coefficients)

print("Feature Importance:", importance)

# Add random values to KK according to the parameters mentioned above to check the proability of Attrition of the Employee

kk=[[1.0, 23.0, 1.0, 500.0, 3.0, 24.0, 1.0]]

print(model.predict\_proba(kk))

**References:**

Ajit, P. (2016). Prediction of employee turnover in organizations using machine learning algorithms.

https://github.com/topics/employee-attrition

P. Usha and N. Balaji, "Analysing Employee attrition using machine learning", Karpagam J. Comput. Sci., vol. 13, pp. 277-282, 2019.

https://www.kaggle.com/anupjana/ibm-hr-employee-attrition-analysis-prediction.

S. Ponnuru, G. Merugumala, S. Padigala, R. Vanga and B. Kantapalli, "Employee Attrition Prediction using Logistic Regression“.

Habous, A.; Nfaoui, E.H.; Oubenaalla, Y. Predicting Employee Attrition using Supervised Learning Classification Models.