EMPLOYEE ATTRITION

The Synthetic Employee Attrition Dataset is a simulated dataset designed for the analysis and prediction of employee attrition. It contains detailed information about various aspects of an employee's profile, including demographics, job-related features, and personal circumstances.

The dataset comprises 74,498 samples, split into training and testing sets to facilitate model development and evaluation. Each record includes a unique Employee ID and features that influence employee attrition. The goal is to understand the factors contributing to attrition and develop predictive models to identify at-risk employees.

This dataset is ideal for HR analytics, machine learning model development, and demonstrating advanced data analysis techniques. It provides a comprehensive and realistic view of the factors affecting employee retention, making it a valuable resource for researchers and practitioners in the field of human resources and organizational development.

FEATURES:

Employee ID: A unique identifier assigned to each employee.
Age: The age of the employee, ranging from 18 to 60 years.
Gender: The gender of the employee
Years at Company: The number of years the employee has been working at the company.
Monthly Income: The monthly salary of the employee, in dollars.
Job Role: The department or role the employee works in, encoded into categories such as Finance, Healthcare, Technology, Education, and Media.
Work-Life Balance: The employee's perceived balance between work and personal life, (Poor, Below Average, Good, Excellent)
Job Satisfaction: The employee's satisfaction with their job: (Very Low, Low, Medium, High)
Performance Rating: The employee's performance rating: (Low, Below Average, Average, High)
Number of Promotions: The total number of promotions the employee has received.
Distance from Home: The distance between the employee's home and workplace, in miles.
Education Level: The highest education level attained by the employee: (High School, Associate Degree, Bachelor's Degree, Master's Degree, PhD)
Marital Status: The marital status of the employee: (Divorced, Married, Single)
Job Level: The job level of the employee: (Entry, Mid, Senior)
Company Size: The size of the company the employee works for: (Small,Medium,Large)
Company Tenure: The total number of years the employee has been working in the industry.
Remote Work: Whether the employee works remotely: (Yes or No)
Leadership Opportunities: Whether the employee has leadership opportunities: (Yes or No)
Innovation Opportunities: Whether the employee has opportunities for innovation: (Yes or No)
Company Reputation: The employee's perception of the company's reputation: (Very Poor, Poor, Good, Excellent)

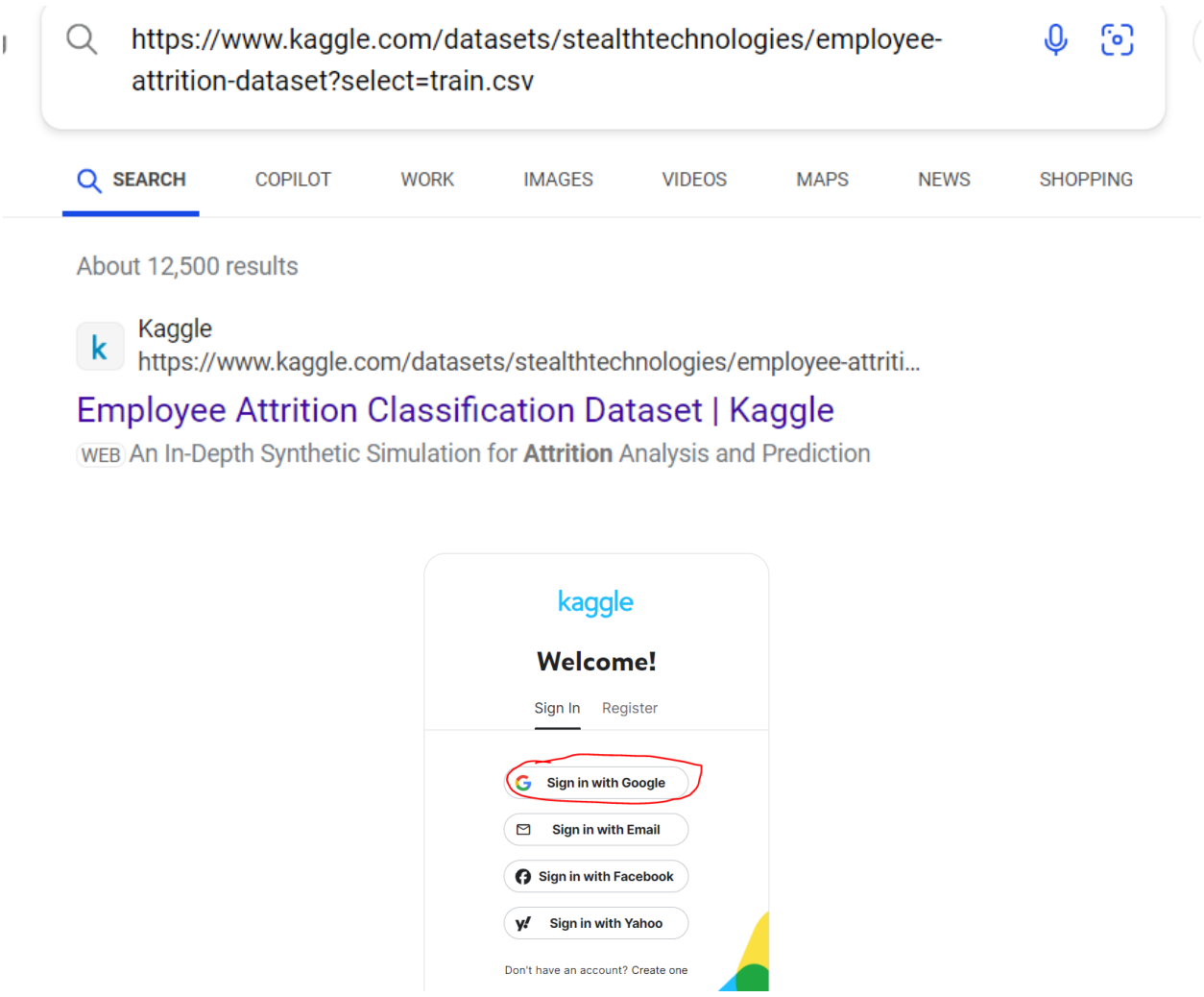
Employee Recognition: The level of recognition the employee receives:(Very Low, Low, Medium, High)

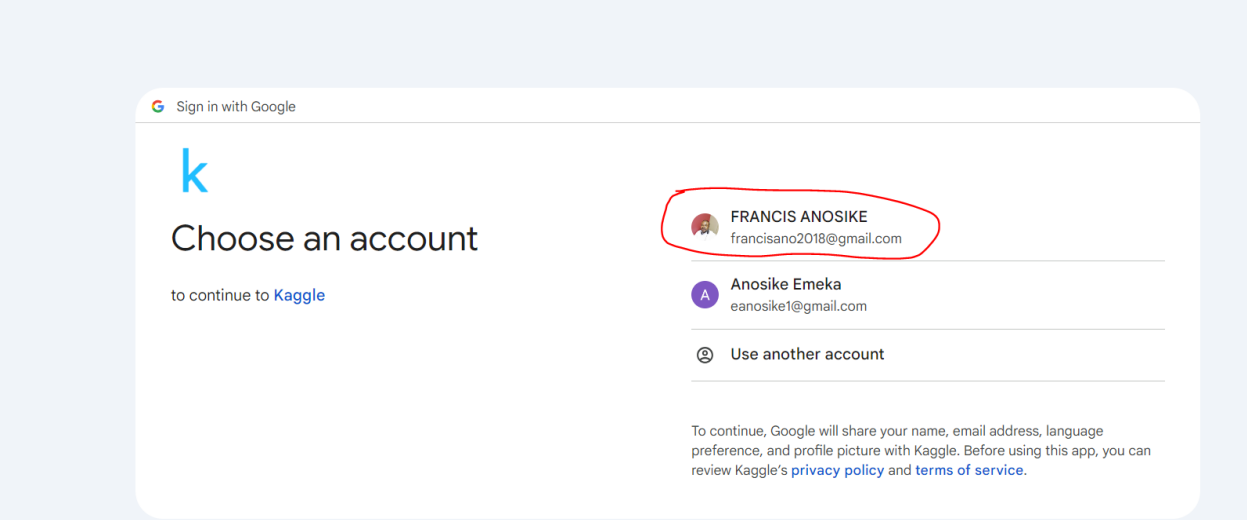
Attrition: Whether the employee has left the company, encoded as 0 (stayed) and 1 (Left).

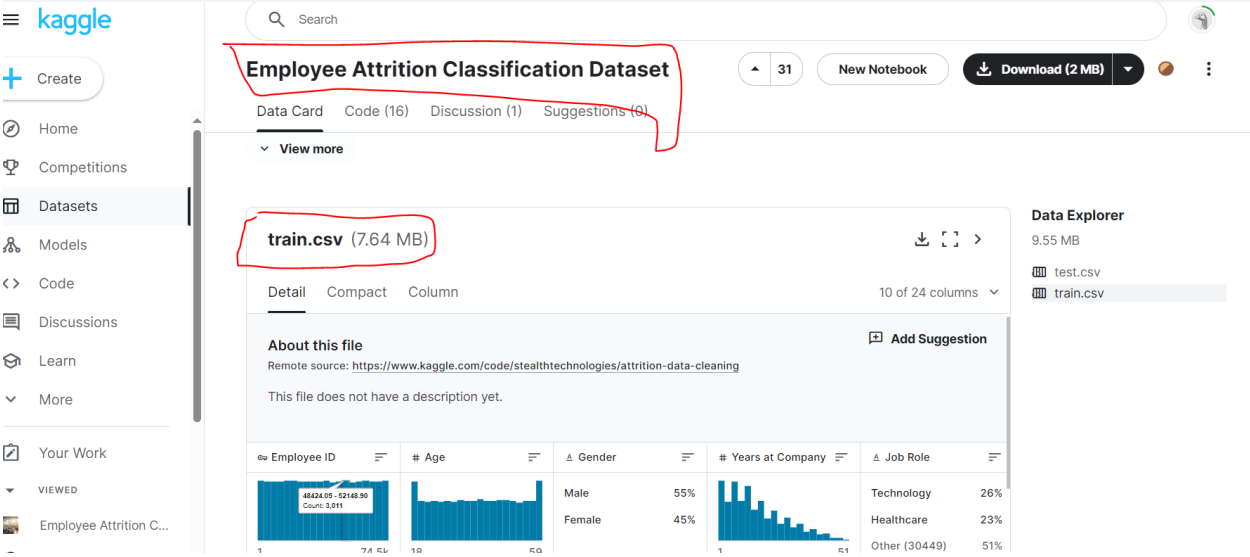
Task 1

LINK to DATASET: https://www.kaggle.com/datasets/stealthtechnologies/employee-attrition-dataset?select=train.csv

1. Share dataset/s used
    - AI Interns to sign up on Kaggle to get datasets




Sign in with Google
Choose an account
to continue to Kaggle
FRANCIS ANOSIKE
francisano2018@gmail.com
Anosike Emeka
eanosike1@gmail.com
Use another account
To continue, Google will share your name, email address, language preference, and profile picture with Kaggle. Before using this app, you can review Kaggle's privacy policy and terms of service.


kaggle
Search
Create
Home
Competitions
Datasets
Models
Code
Discussions
Learn
More
Your Work
VIEWED
Employee Attrition C...
Employee Attrition Classification Dataset
31
New Notebook
Download (2 MB)
Data Card
Code (16)
Discussion (1)
Suggestions (9)
View more
train.csv (7.64 MB)
Detail
Compact
Column
10 of 24 columns
About this file
Remote source: https://www.kaggle.com/code/stealthtechnologies/attrition-data-cleaning
This file does not have a description yet.
Add Suggestion
Employee ID
Age
Gender
Years at Company
Job Role
Male 55%
Female 45%
Technology 26%
Healthcare 23%
Other (30449) 51%
Data Explorer
9.55 MB
test.csv
train.csv

Task 2

2. **Documentation**
    1. **This needs to be reflect the model methodology applied , i.e. Document the entire process, including data preprocessing steps, model selection, evaluation metrics, and any decisions made during development.**
    2. **The key outcome is for the analyst on our end to recreate this model, hence this document needs to be as comprehensive as possible.**

Data quality management was ensured

Data profiler packages was created

Data quality packages were created

Employee Attrition model created

[ ]:
```
#Description:  Training Employee Attrition to find Employee that left or Stayed
```

[13]:
```
#Import the libraries or packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
pd.set_option('display.max_columns',None)
import warnings
warnings.filterwarnings('ignore')
```

```
pip install ydata_profiling
```

```
from ydata_profiling import ProfileReport, compare
```

```
pip install ydata_profiling
```

```
Collecting ydata_profiling
  Downloading ydata_profiling-4.9.0-py2.py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: scipy<1.14,>=1.4.1 in c:\users\francisanosike\anaconda3\lib\site-packages (from ydata_profiling) (1.11.4)
Requirement already satisfied: pandas!=1.4.0,<3,>1.1 in c:\users\francisanosike\anaconda3\lib\site-packages (from ydata_profiling) (2.1.4)
Requirement already satisfied: matplotlib<3.10,>=3.5 in c:\users\francisanosike\anaconda3\lib\site-packages (from ydata_profiling) (3.8.0)
Collecting pydantic>=2 (from ydata_profiling)
  Downloading pydantic-2.8.2-py3-none-any.whl.metadata (125 kB)
     ---------------------------------------- 0.0/125.2 kB ? eta -:--:--
     -------- ------------------------------- 30.7/125.2 kB 640.0 kB/s eta 0:00:01
     ---------------------------------- --- 112.6/125.2 kB 1.3 MB/s eta 0:00:01
     ---------------------------------------- 125.2/125.2 kB 1.2 MB/s eta 0:00:00
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in c:\users\francisanosike\anaconda3\lib\site-packages (from ydata_profiling) (6.0.1)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in c:\users\francisanosike\anaconda3\lib\site-packages (from ydata_profiling) (3.1.3)
Collecting visions<0.7.7,>=0.7.5 (from visions[type_image_path]<0.7.7,>=0.7.5->ydata_profiling)
  Downloading visions-0.7.6-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: numpy<2,>=1.16.0 in c:\users\francisanosike\anaconda3\lib\site-packages (from ydata_profiling) (1.26.4)
Collecting htmlmin==0.1.12 (from ydata_profiling)
  Downloading htmlmin-0.1.12.tar.gz (19 kB)
```

[14]:
```
#loading the data from kagle datasets
df = pd.read_csv('train.csv')
df.head(20)
```

[14]:

ıber of ıotions	Overtime	Distance from Home	Education Level	Marital Status	Number of Dependents	Job Level	Company Size	Company Tenure	Remote Work	Leadership Opportunities	Innovation Opportunities	Company Reputation	Employee Recognition	Attrition
2	No	22	Associate Degree	Married	0	Mid	Medium	89	No	No	No	Excellent	Medium	Stayed
3	No	21	Master's Degree	Divorced	3	Mid	Medium	21	No	No	No	Fair	Low	Stayed
0	No	11	Bachelor's Degree	Married	3	Mid	Medium	74	No	No	No	Poor	Low	Stayed
1	No	27	High School	Single	2	Mid	Small	50	Yes	No	No	Good	Medium	Stayed
0	Yes	71	High School	Divorced	0	Senior	Medium	68	No	No	No	Fair	Medium	Stayed
3	No	37	Bachelor's Degree	Married	0	Mid	Medium	47	No	No	Yes	Fair	High	Left
1	Yes	75	High School	Divorced	3	Entry	Small	93	No	No	No	Good	Medium	Left

[ ]:
```
#EDA
#Exploratory Data Analysis
```

[15]:
```
#view some statistics of the datasets
df.describe()
```

[15]:

	Employee ID	Age	Years at Company	Monthly Income	Number of Promotions	Distance from Home	Number of Dependents	Company Tenure
count	59598.000000	59598.000000	59598.000000	59598.000000	59598.000000	59598.000000	59598.000000	59598.000000
mean	37227.118729	38.565875	15.753901	7302.397983	0.832578	50.007651	1.648075	55.758415
std	21519.150028	12.079673	11.245981	2151.457423	0.994991	28.466459	1.555689	25.411090
min	1.000000	18.000000	1.000000	1316.000000	0.000000	1.000000	0.000000	2.000000
25%	18580.250000	28.000000	7.000000	5658.000000	0.000000	25.000000	0.000000	36.000000
50%	37209.500000	39.000000	13.000000	7354.000000	1.000000	50.000000	1.000000	56.000000
75%	55876.750000	49.000000	23.000000	8880.000000	2.000000	75.000000	3.000000	76.000000
max	74498.000000	59.000000	51.000000	16149.000000	4.000000	99.000000	6.000000	128.000000

:
```
#get the number of rows and columns
df.shape
```

:
```
(59598, 24)
```

:
```
#Get the column data types
df.dtypes
```

:
```
Employee ID                int64
Age                        int64
Gender                    object
Years at Company           int64
Job Role                  object
Monthly Income             int64
Work-Life Balance         object
Job Satisfaction          object
Performance Rating        object
Number of Promotions       int64
Overtime                  object
Distance from Home         int64
Education Level           object
Marital Status            object
Number of Dependents       int64
Job Level                 object
Company Size              object
Company Tenure             int64
Remote Work               object
```

```
#Get a count of the empty vlues for each columns
df.isna().sum()
```

```
Employee ID                 0
Age                         0
Gender                      0
Years at Company            0
Job Role                    0
Monthly Income              0
Work-Life Balance           0
Job Satisfaction            0
Performance Rating          0
Number of Promotions        0
Overtime                    0
Distance from Home          0
Education Level             0
Marital Status              0
Number of Dependents        0
Job Level                   0
Company Size                0
Company Tenure              0
Remote Work                 0
Leadership Opportunities    0
Innovation Opportunities    0
Company Reputation          0
Employee Recognition        0
```

```
#check for any missing /null values in the data
df.isnull().values.any()
```

```
False
```

```
#Get the information about the datasets
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 59598 entries, 0 to 59597
Data columns (total 24 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Employee ID              59598 non-null  int64
 1   Age                      59598 non-null  int64
 2   Gender                   59598 non-null  object
 3   Years at Company         59598 non-null  int64
 4   Job Role                 59598 non-null  object
 5   Monthly Income           59598 non-null  int64
 6   Work-Life Balance        59598 non-null  object
 7   Job Satisfaction         59598 non-null  object
 8   Performance Rating       59598 non-null  object
 9   Number of Promotions     59598 non-null  int64
 10  Overtime                 59598 non-null  object
 11  Distance from Home       59598 non-null  int64
 12  Education Level          59598 non-null  object
```

```
#Get the count of the number of Employeee that stayed or left the company
df['Attrition'].value_counts()
```

```
Attrition
Stayed    31260
Left      28338
Name: count, dtype: int64
```

```
#Print all of the data types and their unique values
for column in df.columns:
    if df[column].dtype == object:
        print(str(column)+ ' : '+ str(df[column].unique()))
        print(df[column].value_counts())
        print('_____________________')
```

```
#Print all of the data types and their unique values
for column in df.columns:
    if df[column].dtype == object:
        print(str(column)+ ' : '+ str(df[column].unique()))
        print(df[column].value_counts())
        print('_____________________')
```

```
Fair         11817
Excellent     5981
Name: count, dtype: int64
_____________________
Employee Recognition : ['Medium' 'Low' 'High' 'Very High']
Employee Recognition
Low          23758
Medium       18033
High         14844
Very High     2963
Name: count, dtype: int64
_____________________
Attrition : ['Stayed' 'Left']
Attrition
Stayed    31260
Left      28338
Name: count, dtype: int64
_____________________
```

```
##Lets convert this attrition to label
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Attrition'] = le.fit_transform(df['Attrition'])
df.head(20)
```

ıber of otions	Overtime	Distance from Home	Education Level	Marital Status	Number of Dependents	Job Level	Company Size	Company Tenure	Remote Work	Leadership Opportunities	Innovation Opportunities	Company Reputation	Employee Recognition	Attrition
2	No	22	Associate Degree	Married	0	Mid	Medium	89	No	No	No	Excellent	Medium	1
3	No	21	Master's Degree	Divorced	3	Mid	Medium	21	No	No	No	Fair	Low	1
0	No	11	Bachelor's Degree	Married	3	Mid	Medium	74	No	No	No	Poor	Low	1
1	No	27	High School	Single	2	Mid	Small	50	Yes	No	No	Good	Medium	1
0	Yes	71	High School	Divorced	0	Senior	Medium	68	No	No	No	Fair	Medium	1
3	No	37	Bachelor's Degree	Married	0	Mid	Medium	47	No	No	Yes	Fair	High	0

```
#the list of classes
le.classes_
```

```
array(['Left', 'Stayed'], dtype=object)
```

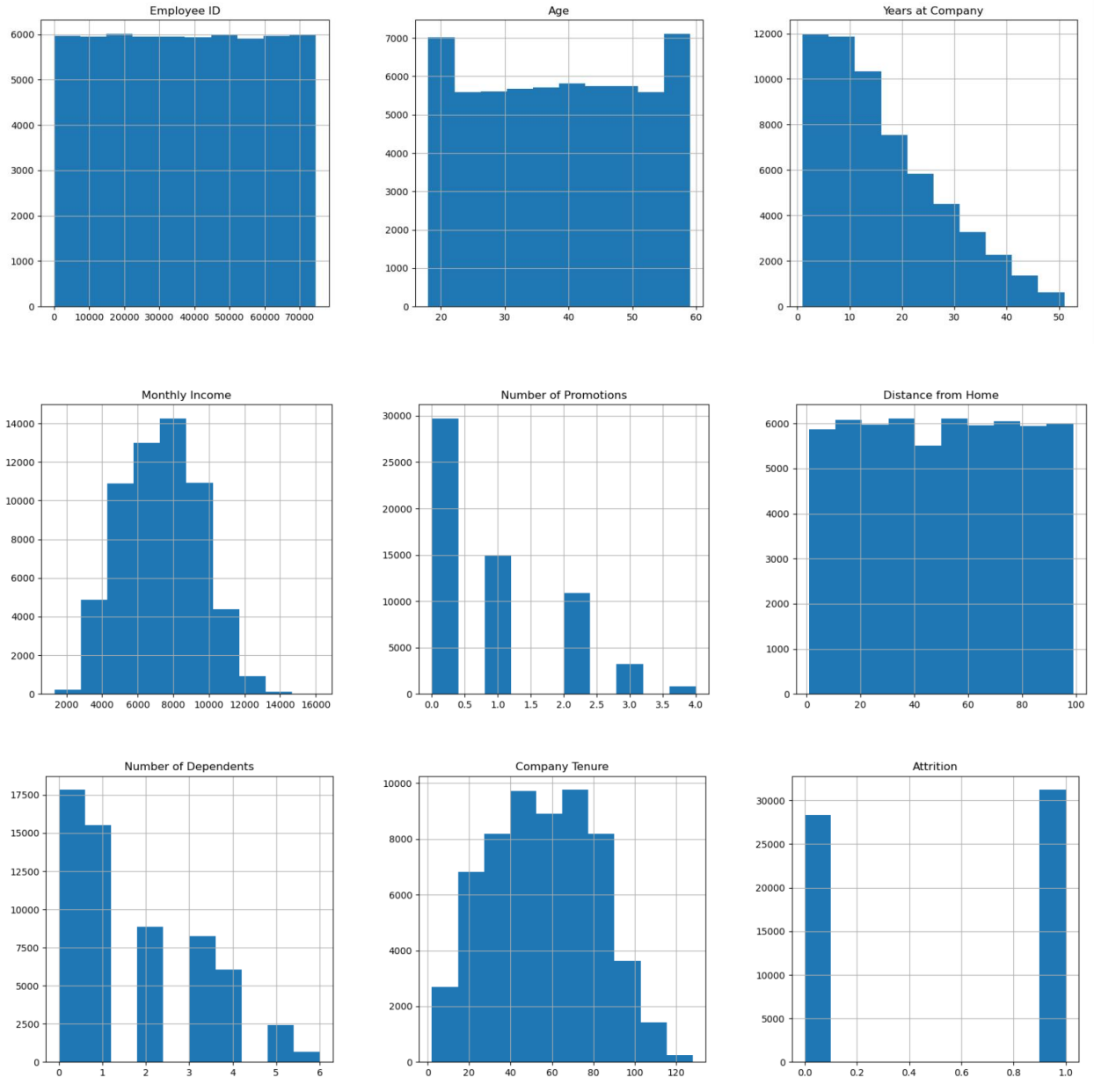
```
#Attrition ={'Stayed': 0, 'Left':1}

#you can assign Attribu=tion number like this
```

```
Attrition_Employee = le.classes_
print(Attrition_Employee)
```

```
['Left' 'Stayed']
```

```
# 0: Employee stayed
# 1 : Employee Left
```

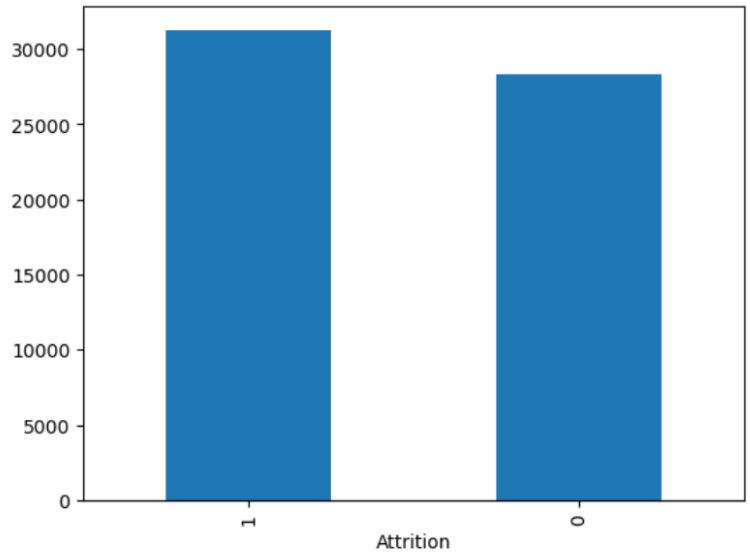
```
#Plotting the data distribution

p = df.hist(figsize =(20,20))
```



```
#checking the balanace of the data by plotting the count of Attrition by their values

color_wheel = {1: "#0392cf", 2:  "#7bc043"}
colors = df["Attrition"].map(lambda x: color_wheel.get(x+1))
print(df.Attrition.value_counts())
p=df.Attrition.value_counts().plot(kind="bar")
```

```
Attrition
1    31260
0    28338
Name: count, dtype: int64
```


```
Attrition
1    31260
0    28338
Name: count, dtype: int64
```


```
#define the columns to be lable encoded
labels_cols = ['Gender','Job Role', 'Overtime','Education Level','Marital Status','Job Level','Company Size','Remote Work'
             , 'Leadership Opportunities','Innovation Opportunities','Work-Life Balance', 'Job Satisfaction', 'Performance Rating', 'Company Reputation'

#initialize lable Encoders
label_encoders = {col: LabelEncoder() for col in labels_cols}

#Apply label Encoding
for col in labels_cols:
    df[col] = label_encoders[col].fit_transform(df[col])
```

Number of Promotions	Overtime	Distance from Home	Education Level	Marital Status	Number of Dependents	Job Level	Company Size	Company Tenure	Remote Work	Leadership Opportunities	Innovation Opportunities	Company Reputation	Employee Recognition	Attrition
2	0	22	0	1	0	1	1	89	0	0	0	0	2	1
3	0	21	3	0	3	1	1	21	0	0	0	1	1	1
0	0	11	1	1	3	1	1	74	0	0	0	3	1	1
1	0	27	2	2	2	1	2	50	1	0	0	2	2	1
0	1	71	2	0	0	2	1	68	0	0	0	1	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	1	66	1	2	2	2	2	35	0	0	1	3	3	0
3	0	42	0	2	0	2	1	73	0	0	0	1	2	0
0	1	34	3	1	2	0	2	29	0	1	0	2	2	1
0	0	62	1	2	0	0	0	9	0	0	0	2	1	0
0	1	20	3	1	3	1	1	81	0	0	0	2	1	1

```
#split our datasets into 2 groups
#training datatsets and testing datasets
#we use bthe training datasets to train our model
#we use the testing datasets to test our model
```

```
#Model Training#

from sklearn.model_selection import train_test_split


#train 70%
#test 30%

x= df.drop(columns=['Employee ID', 'Attrition'])
y= df['Attrition']
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.30)
```

```
x_train.shape
```

```
(41718, 22)
```

```
x_test.shape
```

```
(17880, 22)
```

```
#Logistic Regression
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

```
#model Training
model.fit(x_train, y_train)
```

▾ LogisticRegression

LogisticRegression()

```
#Make predictions
expected = y_train
predicted = model.predict(x_train)
predicted

#this is the predicted Attrition for the Features since we have represented the Attrition in 0, 1
```


```
#Make predictions
expected = y_train
predicted = model.predict(x_train)
predicted

#this is the predicted Attrition for the Features since we have represented the Attrition in 0, 1
```


```
array([1, 1, 1, ..., 1, 1, 1])
```

```
#to better understand the predicted Result we can use the confusion matrix to give insight of the prediction
```

```
from sklearn import metrics
```

```
print (metrics.classification_report(expected, predicted))
```

```
              precision    recall  f1-score   support

           0       0.67      0.63      0.65     19834
           1       0.68      0.71      0.70     21884

    accuracy                           0.67     41718
   macro avg       0.67      0.67      0.67     41718
weighted avg       0.67      0.67      0.67     41718
```

```
print(metrics.confusion_matrix(expected,predicted))
```

```
[[12545  7289]
 [ 6282 15602]]
```

```
import seaborn as sns
```

```
from sklearn import metrics
```

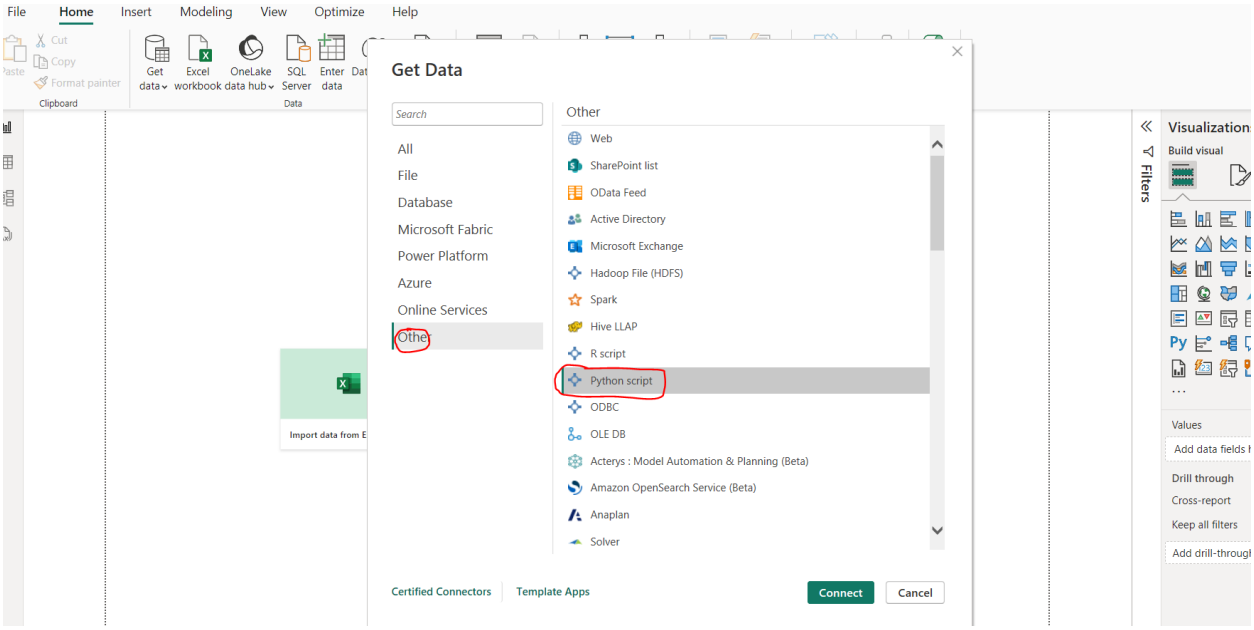
```
#ax = sns.heatmap(metrics, annot=True, fmt='d', cmap ='Reds', xticklabels=Attrition_Employee, yticklabels=Attrition_Employee)
#ax.set_title('confusion matrix for Employee Attrition')
#ax.set_xlabel('prediction', fontsize=14)
#ax.set_ylablel('actual', fontsize=14)
```

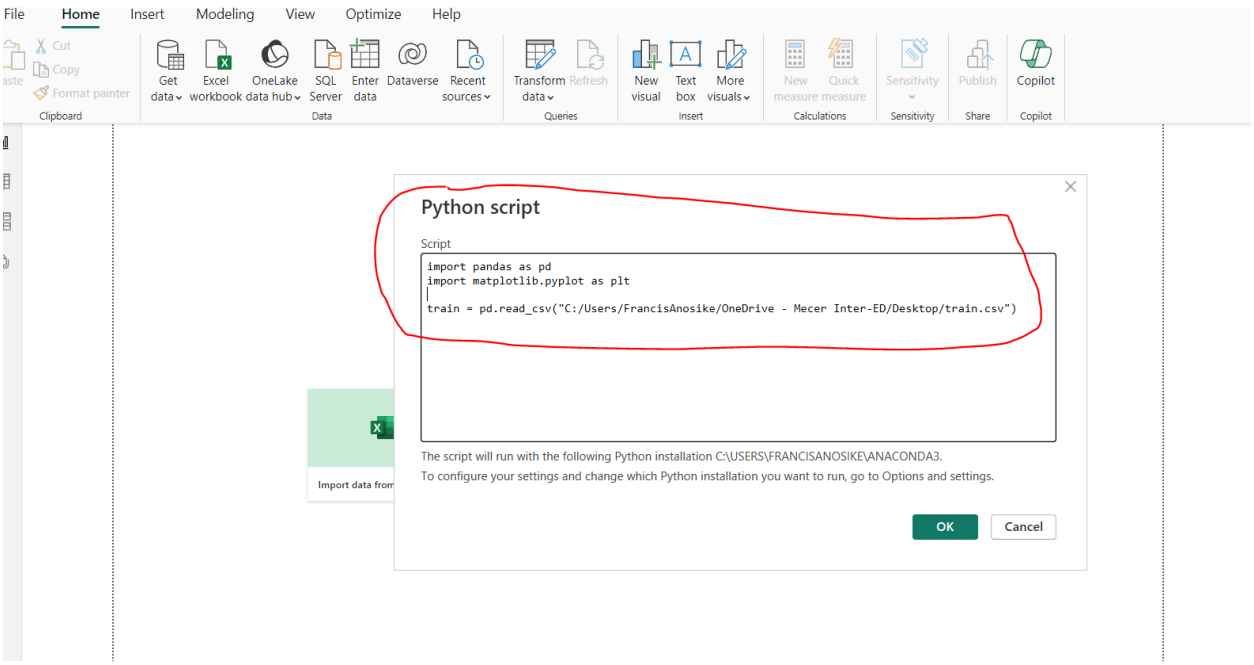
```

```

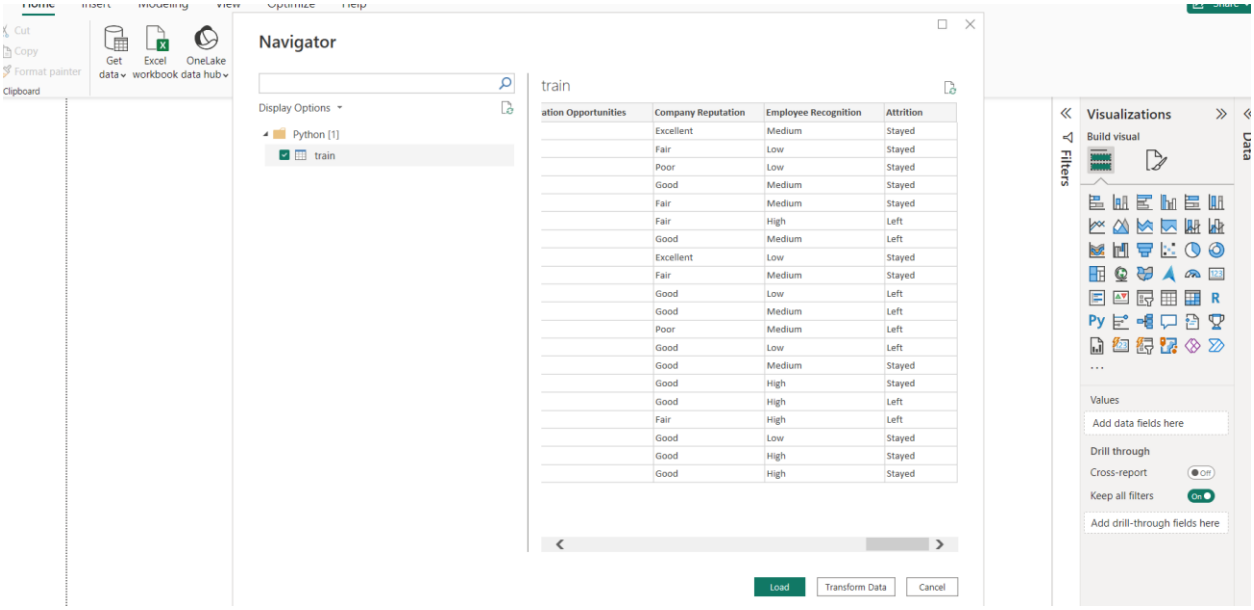
Task 3

Create visuals : Using Python Scripts embedded in PowerBI

Visualizing the dataset using Power Bi

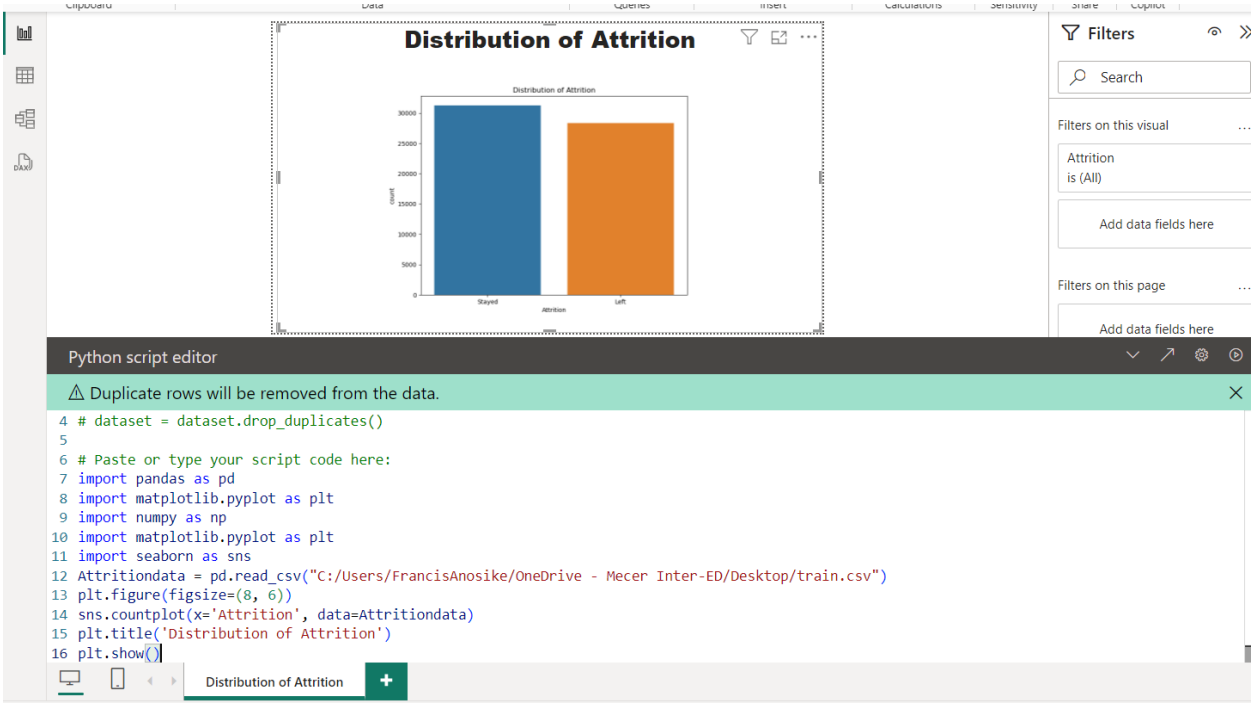


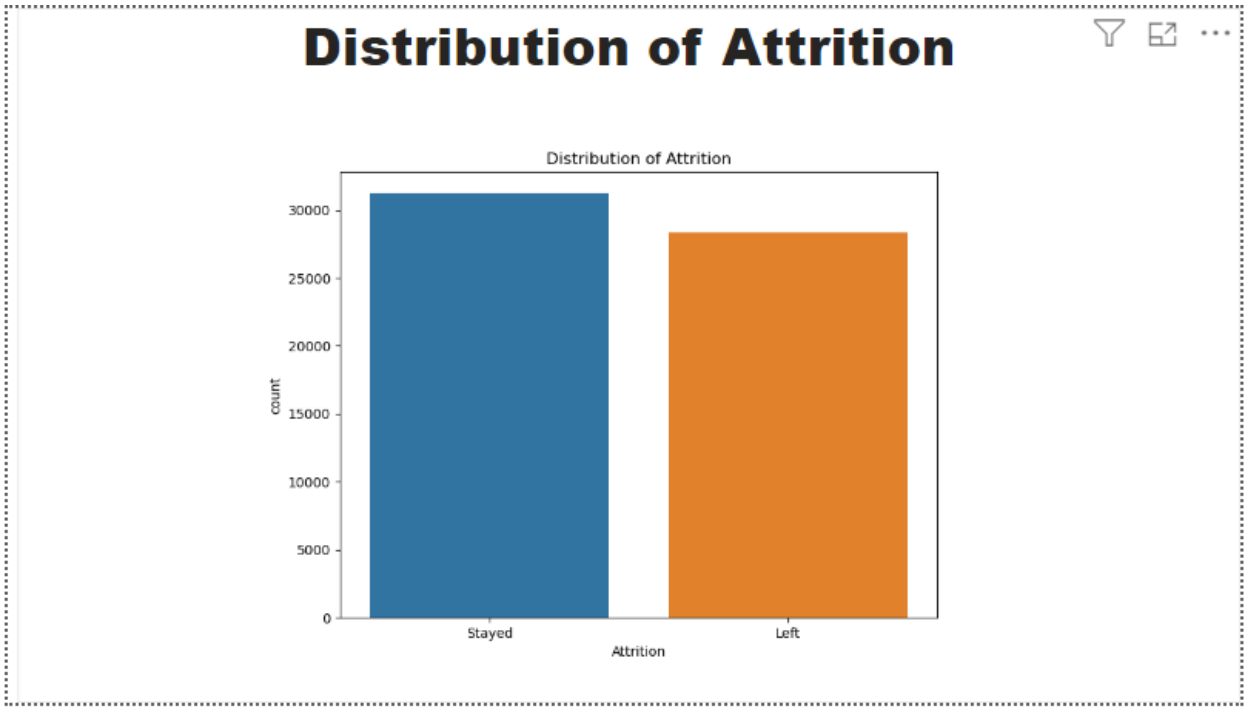


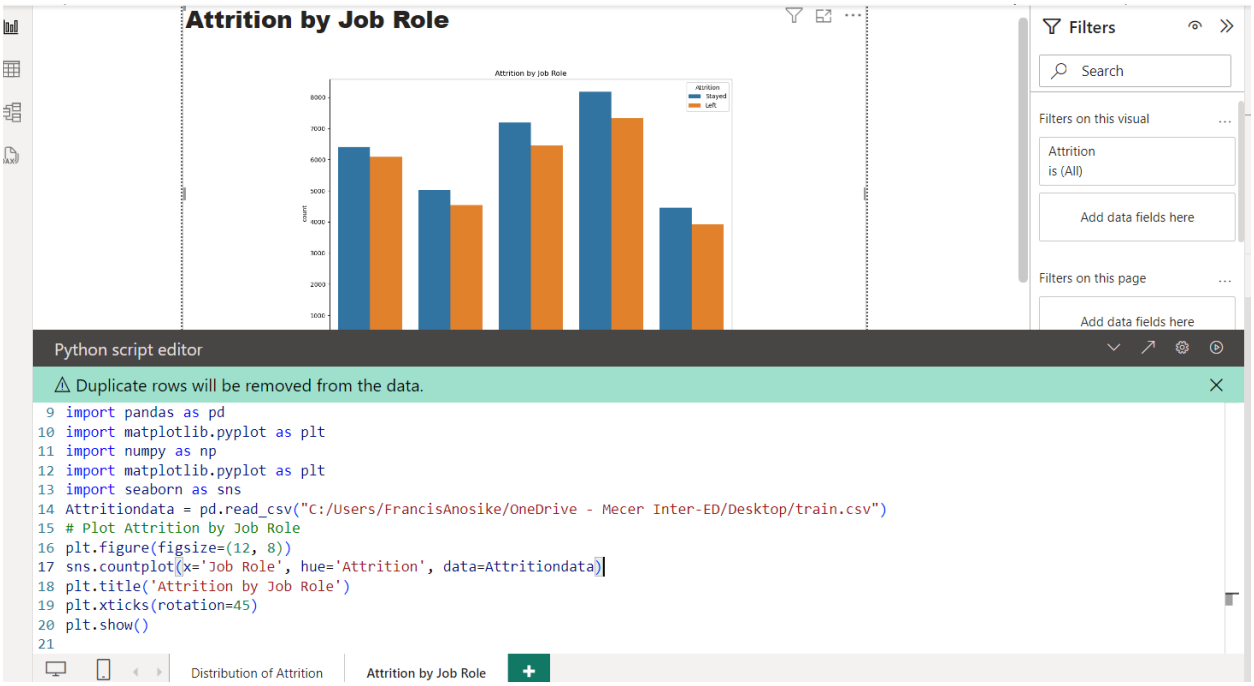


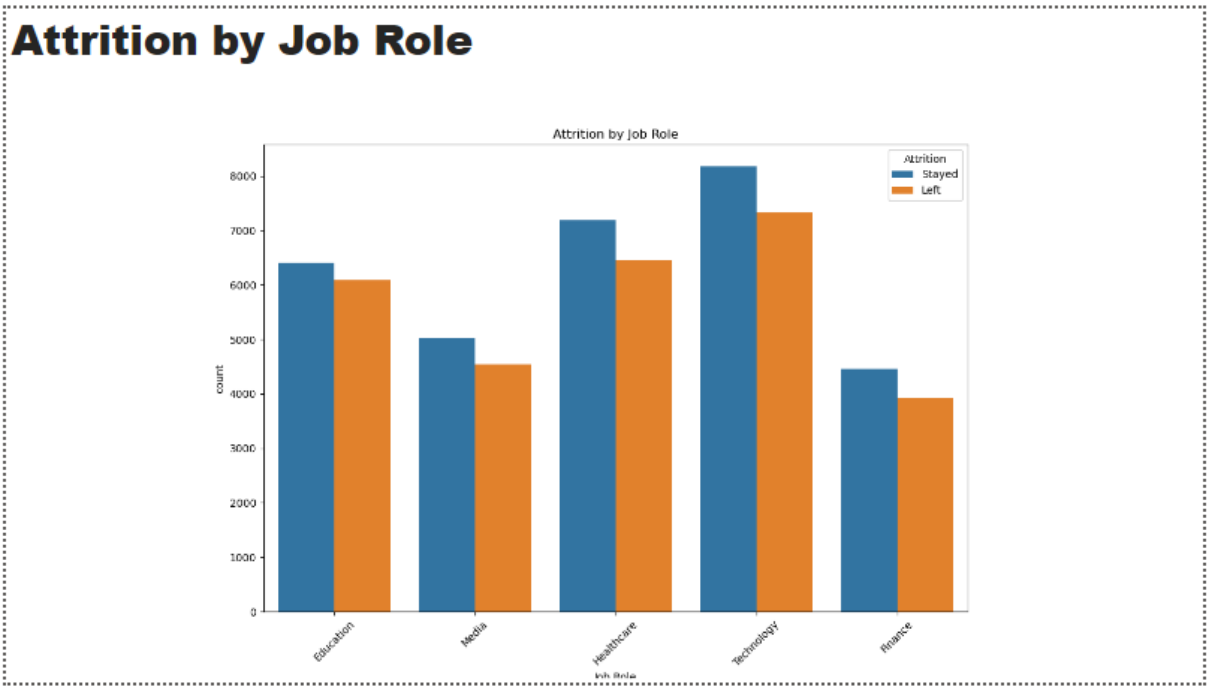
train

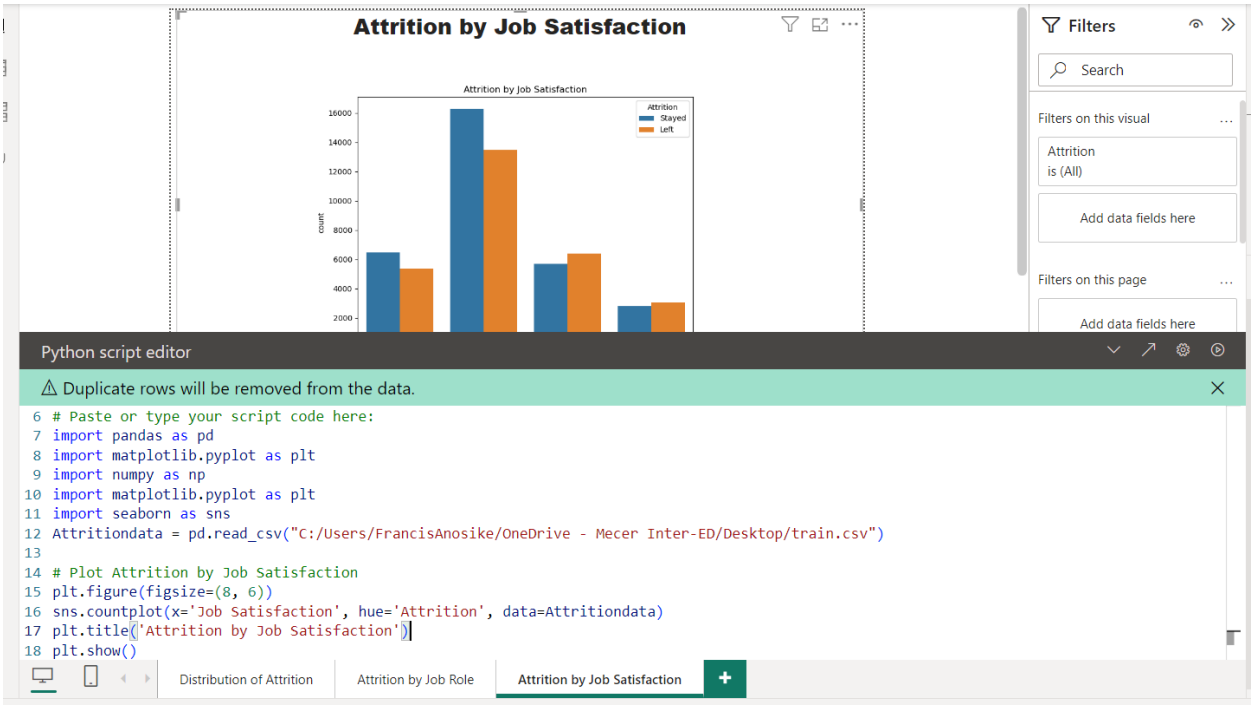
	Remote Work	Leadership Opportunities	Innovation Opportunities	Company Reputation	Employee Recognition	Attrition
1		No	No	Excellent	Medium	0
2		No	No	Fair	Low	0
3		No	No	Poor	Low	0
4	s	No	No	Good	Medium	0
5		No	No	Fair	Medium	0
6		No	Yes	Fair	High	1
7		No	No	Good	Medium	1
8		No	No	Excellent	Low	0
9		No	No	Fair	Medium	0
10		No	Yes	Good	Low	1
11		No	No	Good	Medium	1
12		No	No	Poor	Medium	1
13		No	No	Good	Low	1
14		No	No	Good	Medium	0
15		No	No	Good	High	0
16	s	No	No	Good	High	1
17		No	No	Fair	High	1
18		No	No	Good	Low	0
19		No	No	Good	High	0
20		No	No	Good	High	0
21		No	No	Good	High	0
22		No	No	Fair	Medium	0
23		No	No	Poor	Low	1
24		No	No	Good	High	1
25		No	No	Good	Low	1
26		No	Yes	Good	Medium	1
27		No	Yes	Good	Low	0
28		No				

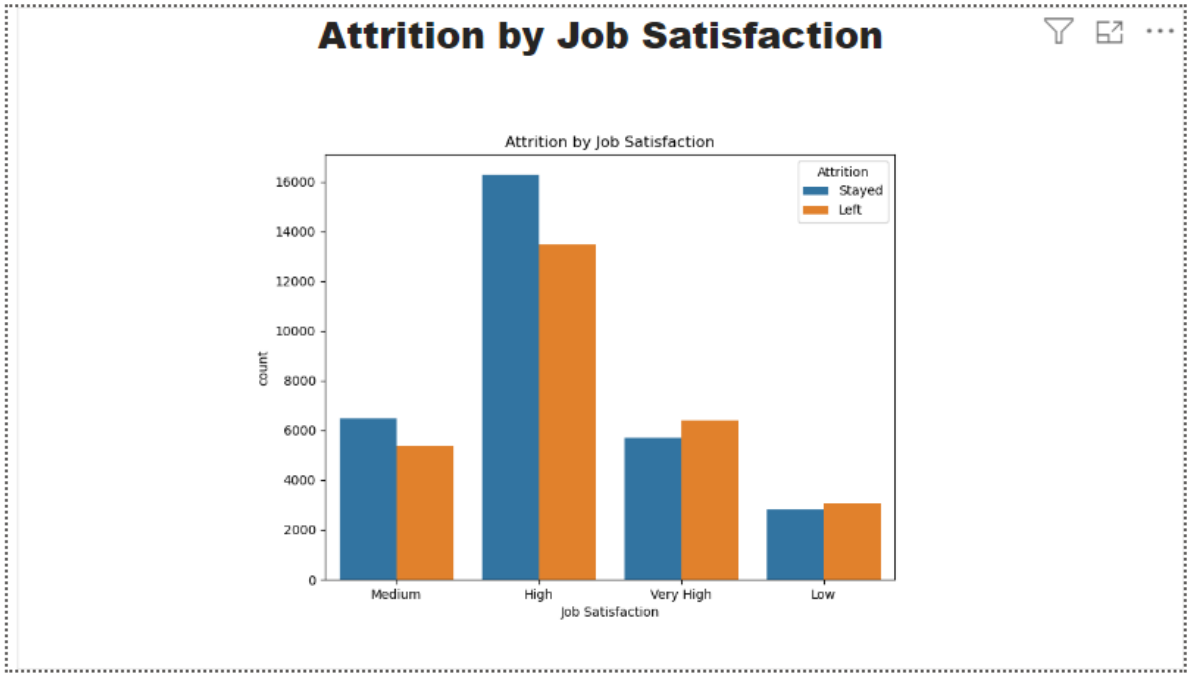

Distribution of Attrition
Filters
Search
Filters on this visual
Attribution
is (All)
Add data fields here
Filters on this page
Add data fields here
Python script editor
⚠ Duplicate rows will be removed from the data.
# dataset = dataset.drop_duplicates()
# Paste or type your script code here:
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
Attritiondata = pd.read_csv("C:/Users/FrancisAnosike/OneDrive - Mecer Inter-ED/Desktop/train.csv")
plt.figure(figsize=(8, 6))
sns.countplot(x='Attrition', data=Attritiondata)
plt.title('Distribution of Attrition')
plt.show()
Distribution of Attrition

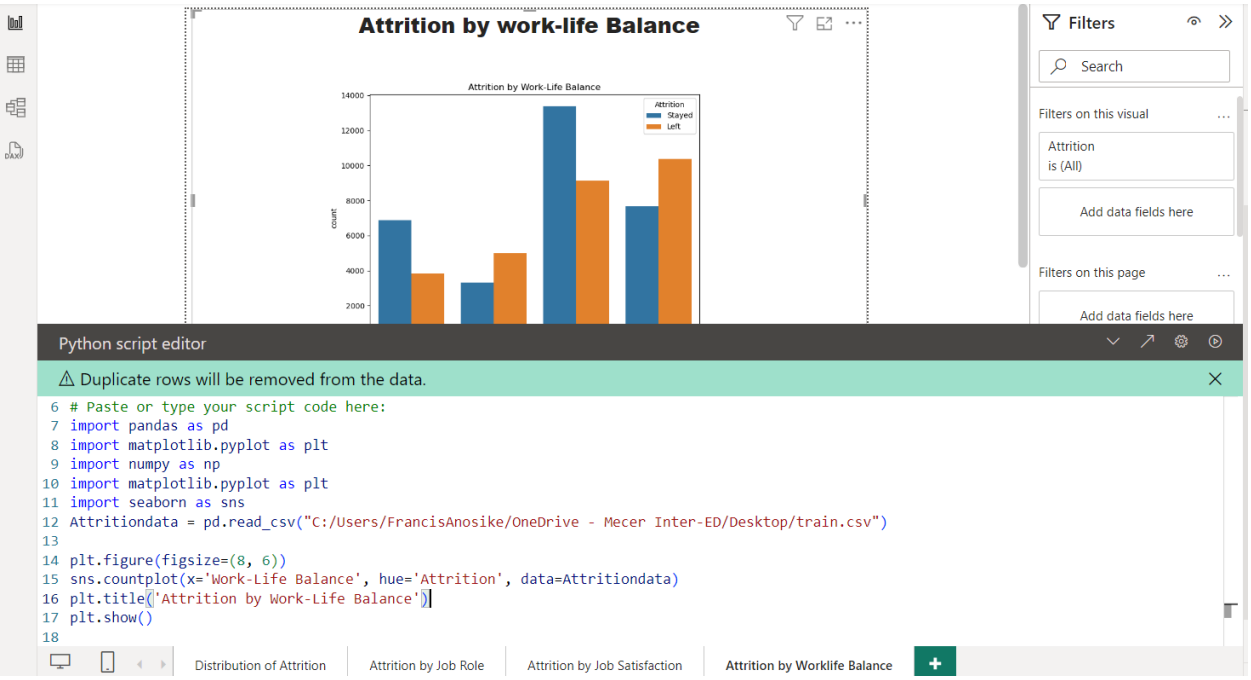

Distribution of Attrition
Distribution of Attrition
30000
25000
20000
15000
10000
5000
0
count
Stayed
Left
Attrition

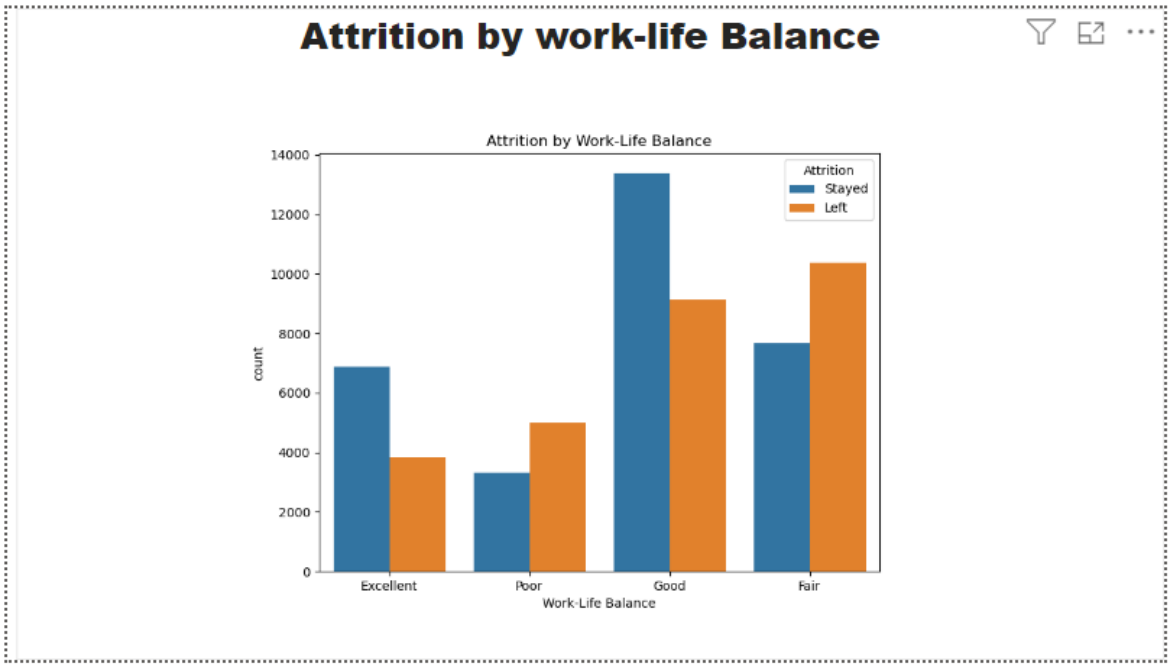

Attrition by Job Role
Attrition by Job Role
Filters
Search
Filters on this visual
Attrition
is (All)
Add data fields here
Filters on this page
Add data fields here
Python script editor
⚠ Duplicate rows will be removed from the data.
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
Attritiondata = pd.read_csv("C:/Users/FrancisAnosike/OneDrive - Mecer Inter-ED/Desktop/train.csv")
# Plot Attrition by Job Role
plt.figure(figsize=(12, 8))
sns.countplot(x='Job Role', hue='Attrition', data=Attritiondata)
plt.title('Attrition by Job Role')
plt.xticks(rotation=45)
plt.show()
Distribution of Attrition
Attrition by Job Role

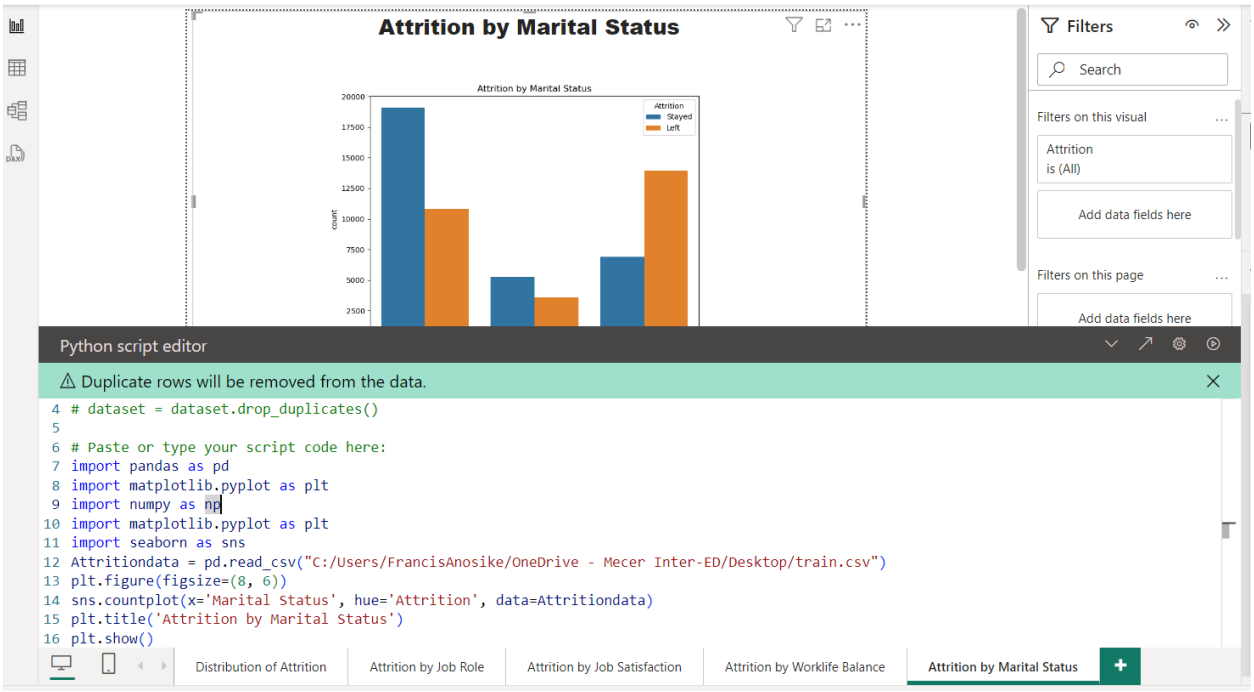

Attrition by Job Role
Attrition by Job Role
Attrition
Stayed
Left
count
0
1000
2000
3000
4000
5000
6000
7000
8000
Education
Media
Healthcare
Technology
Finance
Job Role

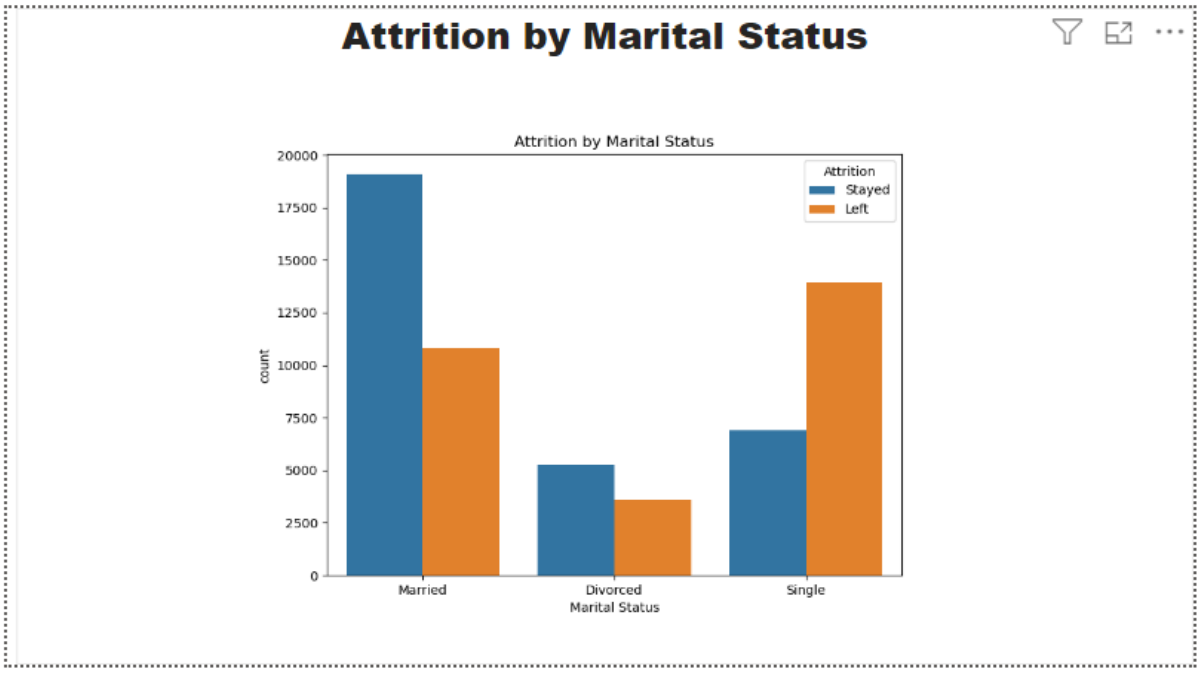

Attrition by Job Satisfaction
Attrition by Job Satisfaction
Attrition
Stayed
Left
count
Filters
Search
Filters on this visual
Attrition
is (All)
Add data fields here
Filters on this page
Add data fields here
Python script editor
⚠ Duplicate rows will be removed from the data.
6 # Paste or type your script code here:
7 import pandas as pd
8 import matplotlib.pyplot as plt
9 import numpy as np
10 import matplotlib.pyplot as plt
11 import seaborn as sns
12 Attritiondata = pd.read_csv("C:/Users/FrancisAnosike/OneDrive - Mecer Inter-ED/Desktop/train.csv")
13
14 # Plot Attrition by Job Satisfaction
15 plt.figure(figsize=(8, 6))
16 sns.countplot(x='Job Satisfaction', hue='Attrition', data=Attritiondata)
17 plt.title('Attrition by Job Satisfaction')
18 plt.show()
Distribution of Attrition
Attrition by Job Role
Attrition by Job Satisfaction


Attrition by Job Satisfaction
Attrition by Job Satisfaction
Attrition
Stayed
Left
count
Medium
High
Very High
Low
Job Satisfaction


Attrition by work-life Balance
Attrition by Work-Life Balance
Attrition
Stayed
Left
count
Filters
Search
Filters on this visual
Attrition
is (All)
Add data fields here
Filters on this page
Add data fields here
Python script editor
⚠ Duplicate rows will be removed from the data.
6 # Paste or type your script code here:
7 import pandas as pd
8 import matplotlib.pyplot as plt
9 import numpy as np
10 import matplotlib.pyplot as plt
11 import seaborn as sns
12 Attritiondata = pd.read_csv("C:/Users/FrancisAnosike/OneDrive - Mecer Inter-ED/Desktop/train.csv")
13
14 plt.figure(figsize=(8, 6))
15 sns.countplot(x='Work-Life Balance', hue='Attrition', data=Attritiondata)
16 plt.title('Attrition by Work-Life Balance')
17 plt.show()
18
Distribution of Attrition
Attrition by Job Role
Attrition by Job Satisfaction
Attrition by Worklife Balance


Attrition by work-life Balance
Attrition by Work-Life Balance
Attrition
Stayed
Left
count
Excellent
Poor
Good
Fair
Work-Life Balance


Attrition by Marital Status
Attrition by Marital Status
Attrition
Stayed
Left
count
Filters
Search
Filters on this visual
Attrition
is (All)
Add data fields here
Filters on this page
Add data fields here
Python script editor
⚠ Duplicate rows will be removed from the data.
# dataset = dataset.drop_duplicates()
# Paste or type your script code here:
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
Attritiondata = pd.read_csv("C:/Users/FrancisAnosike/OneDrive - Mecer Inter-ED/Desktop/train.csv")
plt.figure(figsize=(8, 6))
sns.countplot(x='Marital Status', hue='Attrition', data=Attritiondata)
plt.title('Attrition by Marital Status')
plt.show()
Distribution of Attrition
Attrition by Job Role
Attrition by Job Satisfaction
Attrition by Worklife Balance
Attrition by Marital Status


Attrition by Marital Status
Attrition by Marital Status
Attrition
Stayed
Left
count
Married
Divorced
Single
Marital Status

Task 4

3. Deployment
   1. Deploy the model in a controlled environment, continuously monitor its performance, and update it as necessary based on new data and changing business needs.

   2. Evidence of the above needs to form part of the Documentation in point 2.a above.

Deployment of Trained Model

```
import pandas as pd
from azureml.core import Dataset,Workspace, Datastore
from azureml.train.automl import AutoMLConfig
from azureml.core.experiment import Experiment
from azureml.widgets import RunDetails

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error
from math import sqrt

import logging
```

```
ws = Workspace.from_config()
ws
```

```
Workspace.create(name='attritionworkspaces', subscription_id='85d7de82-6489-45e9-b561-2db060c6008b', resource_group='trainrsg')
```

```
df = pd.read_csv("train.csv",index_col=0)
df
```

Job ction	Performance Rating	Number of Promotions	Overtime	...	Number of Dependents	Job Level	Company Size	Company Tenure	Remote Work	Leadership Opportunities	Innovation Opportunities	Company Reputation	Employee Recognition	Attrition
edium	Average	2	No	...	0	Mid	Medium	89	No	No	No	Excellent	Medium	Stayed
High	Low	3	No	...	3	Mid	Medium	21	No	No	No	Fair	Low	Stayed
High	Low	0	No	...	3	Mid	Medium	74	No	No	No	Poor	Low	Stayed
High	High	1	No	...	2	Mid	Small	50	Yes	No	No	Good	Medium	Stayed
High	Average	0	Yes	...	0	Senior	Medium	68	No	No	No	Fair	Medium	Stayed
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
High	Average	1	Yes	...	2	Senior	Small	35	No	No	Yes	Poor	Very High	Left
High	High	3	No	...	0	Senior	Medium	73	No	No	No	Fair	Medium	Left
High	High	0	Yes	...	2	Entry	Small	29	No	Yes	No	Good	Medium	Stayed
High	Average	0	No	...	0	Entry	Large	9	No	No	No	Good	Low	Left
High	Average	0	Yes	...	3	Mid	Medium	81	No	No	No	Good	Low	Stayed

```
ds = Datastore.get_default(ws)
ds
```

```
{
  "name": "workspaceblobstore",
  "container_name": "azureml-blobstore-b3e168bb-5f52-473a-9968-57d9dd91aa53",
  "account_name": "attritionworks1751666779",
  "protocol": "https",
  "endpoint": "core.windows.net"
}
```

```
x_train, x_test = train_test_split(df, test_size=0.2, random_state=129)
```

```
print(len(x_train),",",len(x_test))
```

```
47678 , 11920
```

```
dataset = Dataset.Tabular.register_pandas_dataframe(x_train,ds,"train.csv",show_progress=True)
test_dataset = Dataset.Tabular.register_pandas_dataframe(x_test,ds,"train.csv",show_progress=True)
```

```
Validating arguments.
Arguments validated.
Validating arguments.
Arguments validated.
'overwrite' is set to True. Any file already present in the target will be overwritten.
Successfully uploaded file to datastore.
Creating and registering a new dataset.
Files copied=1, skipped=0, failed=0
Successfully created and registered a new dataset.
```

```
automl_settings={
    "iteration_timeout_minutes":10,
    "experiment_timeout_minutes":15,
    "enable_early_stopping":True,
    "primary_metric":'AUC_weighted',
    "featurization":'auto',
    "verbosity":logging.INFO,
    "n_cross_validations":5

}
```

```
automl_config = AutoMLConfig(task='classification',
                             debug_log='auto_errors.log',
                             training_data=dataset,
                             label_column_name="Attrition",
                             compute_target="francisa1",
                             **automl_settings)
```

```
experiment = Experiment(ws,"train-experiment")
run = experiment.submit(automl_config, show_output=True)
```

```
Submitting remote run.
No run_configuration provided, running on francisa1 with default configuration
Running on remote compute: francisa1
```

Experiment	Id	Type	Status	Details Page	Docs Page
train-experiment	AutoML_d6b931bc-4851-4405-a5cb-9b40068f9284	automl	NotStarted	Link to Azure Machine Learning studio	Link to Documentation

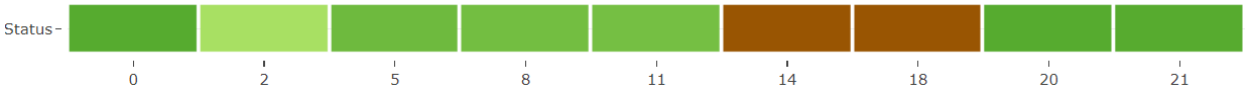
```
Current status: FeaturesGeneration. Generating features for the dataset.
Current status: DatasetFeaturization. Beginning to fit featurizers and featurize the dataset.
Current status: DatasetCrossValidationSplit. Generating individually featurized CV splits.
Current status: ModelSelection. Beginning model selection.

********************************************************************************************
DATA GUARDRAILS:

TYPE:         Class balancing detection
STATUS:       PASSED
DESCRIPTION:  Your inputs were analyzed, and all classes are balanced in your training data.
              Learn more about imbalanced data: https://aka.ms/AutomatedMLImbalancedData

********************************************************************************************
```

ITER	PIPELINE	DURATION	METRIC	BEST
0	MaxAbsScaler LightGBM	0:00:34	0.8469	0.8469
1	MaxAbsScaler XGBoostClassifier	0:01:05	0.8372	0.8469
2	MaxAbsScaler ExtremeRandomTrees	0:00:40	0.7904	0.8469
3	MaxAbsScaler RandomForest	0:00:30	0.8015	0.8469
4	StandardScalerWrapper LightGBM	0:00:30	0.8100	0.8469
5	SparseNormalizer XGBoostClassifier	0:00:42	0.8304	0.8469
6	SparseNormalizer RandomForest	0:01:11	0.8174	0.8469
7	StandardScalerWrapper XGBoostClassifier	0:00:34	0.8250	0.8469
8	SparseNormalizer XGBoostClassifier	0:00:54	0.8269	0.8469
9	MaxAbsScaler RandomForest	0:00:32	0.8009	0.8469
10	SparseNormalizer LightGBM	0:00:30	0.8045	0.8469
11	MaxAbsScaler ExtremeRandomTrees	0:01:44	0.8252	0.8469
12	StandardScalerWrapper XGBoostClassifier	0:00:38	0.8361	0.8469
13	StandardScalerWrapper ExtremeRandomTrees	0:00:32	0.8152	0.8469
14	StandardScalerWrapper RandomForest	0:05:48	nan	0.8469
15		0:00:00	nan	0.8469
16		0:00:00	nan	0.8469
17		0:00:00	nan	0.8469
18		0:00:00	nan	0.8469
19		0:00:00	nan	0.8469
20	VotingEnsemble	0:01:05	0.8468	0.8469
21	StackEnsemble	0:01:19	0.8465	0.8469

```
best_run, model = run.get_output()
```

```
RunDetails(run).show()
```

AutoML_d6b931bc-4851-4405-a5cb-9b40068f9284:
Status: Completed



Iteration	Pipeline	Iteration metric	Best metric	Status	Duration	Started	R
0	MaxAbsScaler, LightGBM	0.84688531	0.84688531	Completed	0:01:01	Jul 28, 2024 10:20 PM	
20	VotingEnsemble	0.84675047	0.84688531	Completed	0:01:14	Jul 28, 2024 10:37 PM	
21	StackEnsemble	0.84649208	0.84688531	Completed	0:01:27	Jul 28, 2024 10:39 PM	
5	SparseNormalizer, XGBoostClassifier	0.83041092	0.84688531	Completed	0:04:30	Jul 28, 2024 10:24 PM	
8	SparseNormalizer, XGBoostClassifier	0.82688323	0.84688531	Completed	0:07:11	Jul 28, 2024 10:26 PM	

```
model_name = best_run.properties["model_name"]
registered_name = run.register_model(model_name=model_name, description="Attrition Employee", tags=None)
```

```
from azureml.core.model import InferenceConfig
from azureml.core.webservice import AciWebservice, Webservice
from azureml.core.model import Model
from azureml.core.environment import Environment
```

```
best_run.download_file("outputs/scoring_file_v_1_0_0.py","inference/score.py")
```

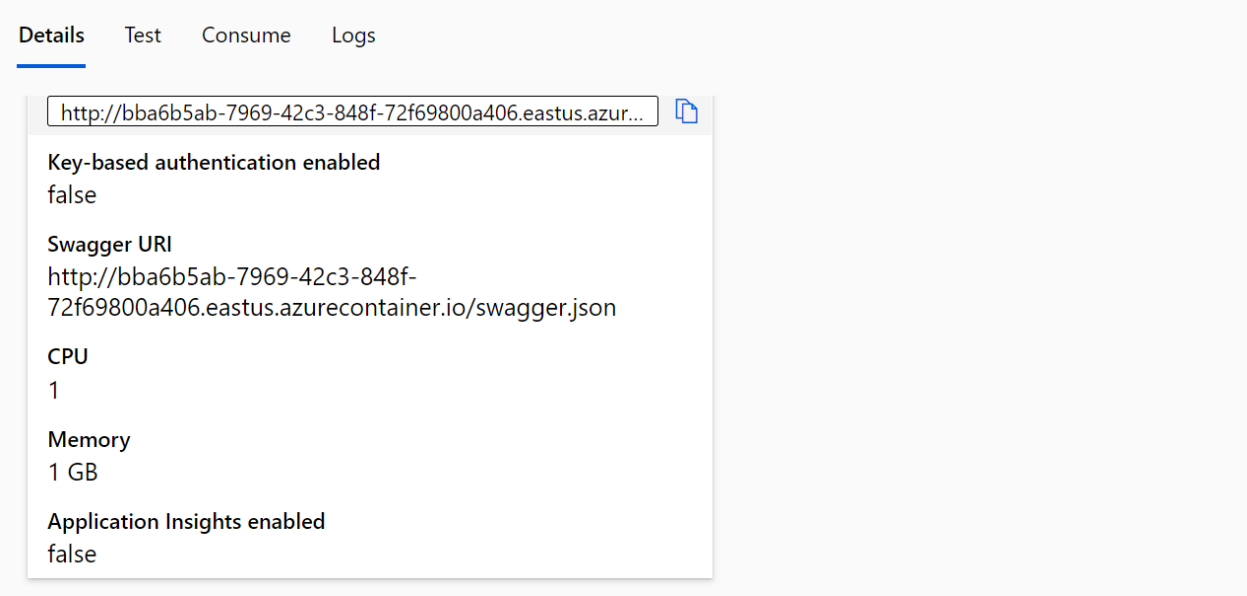
```
from azureml.automl.core.shared import constants

best_run.download_file(constants.CONDA_ENV_FILE_PATH, "myenv.yml")
env = Environment.from_conda_specification(name="myenv",file_path="myenv.yml")

inference_config = InferenceConfig(entry_script="inference/score.py",environment=env)
aciconfig = AciWebservice.deploy_configuration(cpu_cores=1, memory_gb=1, description="Attrition classification")
service = Model.deploy(ws, "automltrition",[registered_name],inference_config,aciconfig)

service.wait_for_deployment(True)
```

```
Tips: You can try get_logs(): https://aka.ms/debugimage#dockerlog or local deployment: https://aka.ms/debugimage#debug-locally
to debug if deployment takes longer than 10 minutes.
```

```
from azureml.automl.core.shared import constants

best_run.download_file(constants.CONDA_ENV_FILE_PATH, "myenv.yml")
env = Environment.from_conda_specification(name="myenv",file_path="myenv.yml")

inference_config = InferenceConfig(entry_script="inference/score.py",environment=env)
aciconfig = AciWebservice.deploy_configuration(cpu_cores=1, memory_gb=1, description="Attrition classification")
service = Model.deploy(ws, "automltrition",[registered_name],inference_config,aciconfig)

service.wait_for_deployment(True)
```

```
Tips: You can try get_logs(): https://aka.ms/debugimage#dockerlog or local deployment: https://aka.ms/debugimage#debug-locally
to debug if deployment takes longer than 10 minutes.
Running
2024-07-29 08:58:31+00:00 Creating Container Registry if not exists..
2024-07-29 09:08:32+00:00 Registering the environment.
2024-07-29 09:08:34+00:00 Building image..
2024-07-29 09:22:07+00:00 Generating deployment configuration.
2024-07-29 09:22:08+00:00 Submitting deployment to compute..
2024-07-29 09:22:13+00:00 Checking the status of deployment automltrition..
2024-07-29 09:24:08+00:00 Checking the status of inference endpoint automltrition.
Succeeded
ACI service creation operation finished, operation "Succeeded"
```

```
2024-07-29 09:31:58,002 W [130] azmlinfsrv - x-ms-request-id header has been deprecated and will be removed from
future versions of the server. Please use x-ms-client-request-id.
2024-07-29 09:31:58,003 I [130] gunicorn.access - 127.0.0.1 - - [29/Jul/2024:09:31:58 +0000] "GET / HTTP/1.0" 200 7
"-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.29 Safari/537.36 OPR/15.0.
1147.24 (Edition Next)"
2024-07-29 09:33:16,253 W [130] azmlinfsrv - x-ms-request-id header has been deprecated and will be removed from
future versions of the server. Please use x-ms-client-request-id.
2024-07-29 09:33:16,254 I [130] azmlinfsrv - GET /score 400 0.484ms 64
2024-07-29 09:33:16,254 I [130] gunicorn.access - 127.0.0.1 - - [29/Jul/2024:09:33:16 +0000] "GET /score HTTP/1.0"
400 64 "-" "PostmanRuntime/7.40.0"
2024-07-29 09:33:38,786 W [130] azmlinfsrv - x-ms-request-id header has been deprecated and will be removed from
future versions of the server. Please use x-ms-client-request-id.
2024-07-29 09:33:38,786 I [130] azmlinfsrv - POST /score 415 0.321ms 58
2024-07-29 09:33:38,787 I [130] gunicorn.access - 127.0.0.1 - - [29/Jul/2024:09:33:38 +0000] "POST /score HTTP/1.0"
415 58 "-" "PostmanRuntime/7.40.0"
```



http://bba6b5ab-7969-42c3-848f-72f69800a406.eastus.azurecontainer.io/score