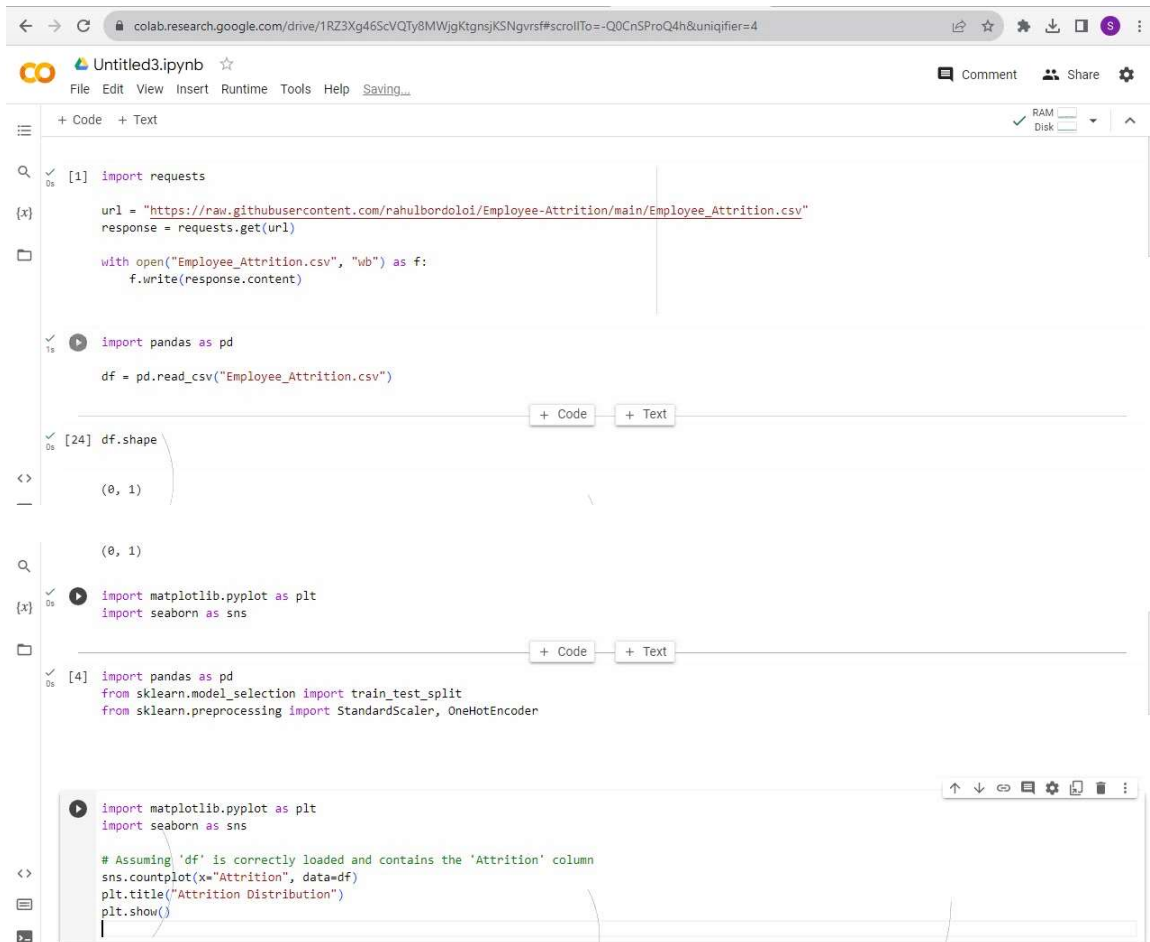


1.Download the Employee Attrition Dataset



```
[1] import requests

url = "https://raw.githubusercontent.com/rahulbordoloi/Employee-Attrition/main/Employee_Attrition.csv"
response = requests.get(url)

with open("Employee_Attrition.csv", "wb") as f:
    f.write(response.content)

import pandas as pd

df = pd.read_csv("Employee_Attrition.csv")

[24] df.shape

(0, 1)

import matplotlib.pyplot as plt
import seaborn as sns

[4] import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder

import matplotlib.pyplot as plt
import seaborn as sns

# Assuming 'df' is correctly loaded and contains the 'Attrition' column
sns.countplot(x="Attrition", data=df)
plt.title("Attrition Distribution")
plt.show()
```

2.Perform Data Preprocessing



```
[4] import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
```

3. Model Building using Logistic Regression and Decision Tree and Random Forest

```
[7] import pandas as pd
    from sklearn.linear_model import LogisticRegression
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.ensemble import RandomForestClassifier

# Instantiate your RandomForestClassifier and train it with data
# Assuming you have already trained your model
employee_features = [1, 2, 4, 5, 7]

try:
    prediction = rf.predict([employee_features])

    if prediction[0] == 1:
        print("The employee is likely to leave the company.")
    else:
        print("The employee is not likely to leave the company.")
except Exception as e:
    print(f"An error occurred: {e}")
```

4. Calculate Performance metrics

```
AN ERROR OCCURRED: NAME 'rf' IS NOT DEFINED
```

```
import pandas as pd
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```