ASSIGNMENT 02

NAME: BHOSALE SRUSHTI AMAR

CLASS: TY AIDS-A

ROLL NO.: 23107008

In [5]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
df=pd.read_csv("Downloads/WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

In [4]:
```
df
```

Out[4]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Edu
0	41	Yes	Travel_Rarely	1102	Sales	1	
1	49	No	Travel_Frequently	279	Research & Development	8	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	
4	27	No	Travel_Rarely	591	Research & Development	2	
...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	
1466	39	No	Travel_Rarely	613	Research & Development	6	
1467	27	No	Travel_Rarely	155	Research & Development	4	
1468	49	No	Travel_Frequently	1023	Sales	2	
1469	34	No	Travel_Rarely	628	Research & Development	8	

1470 rows × 35 columns

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Age                       1470 non-null   int64 
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64 
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64 
 6   Education                 1470 non-null   int64 
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64 
 9   EmployeeNumber            1470 non-null   int64 
 10  EnvironmentSatisfaction   1470 non-null   int64 
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64 
 13  JobInvolvement            1470 non-null   int64 
 14  JobLevel                  1470 non-null   int64 
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64 
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64 
 19  MonthlyRate               1470 non-null   int64 
 20  NumCompaniesWorked        1470 non-null   int64 
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64 
 24  PerformanceRating         1470 non-null   int64 
 25  RelationshipSatisfaction  1470 non-null   int64 
 26  StandardHours             1470 non-null   int64 
 27  StockOptionLevel          1470 non-null   int64 
 28  TotalWorkingYears         1470 non-null   int64 
 29  TrainingTimesLastYear     1470 non-null   int64 
 30  WorkLifeBalance           1470 non-null   int64 
 31  YearsAtCompany            1470 non-null   int64 
 32  YearsInCurrentRole        1470 non-null   int64 
 33  YearsSinceLastPromotion   1470 non-null   int64 
 34  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

In [10]:

```
df.tail()
```

Out[10]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Edu
1465	36	No	Travel_Frequently	884	Research & Development	23	
1466	39	No	Travel_Rarely	613	Research & Development	6	
1467	27	No	Travel_Rarely	155	Research & Development	4	
1468	49	No	Travel_Frequently	1023	Sales	2	
1469	34	No	Travel_Rarely	628	Research & Development	8	

5 rows × 35 columns

In [11]:

```
df.isnull().sum()
```

```
Out[11]: Age                         0
         Attrition                   0
         BusinessTravel              0
         DailyRate                   0
         Department                  0
         DistanceFromHome            0
         Education                   0
         EducationField              0
         EmployeeCount               0
         EmployeeNumber              0
         EnvironmentSatisfaction     0
         Gender                      0
         HourlyRate                  0
         JobInvolvement              0
         JobLevel                    0
         JobRole                     0
         JobSatisfaction             0
         MaritalStatus               0
         MonthlyIncome               0
         MonthlyRate                 0
         NumCompaniesWorked          0
         Over18                      0
         OverTime                    0
         PercentSalaryHike           0
         PerformanceRating           0
         RelationshipSatisfaction    0
         StandardHours               0
         StockOptionLevel            0
         TotalWorkingYears           0
         TrainingTimesLastYear       0
         WorkLifeBalance             0
         YearsAtCompany              0
         YearsInCurrentRole          0
         YearsSinceLastPromotion     0
         YearsWithCurrManager        0
         dtype: int64
```

```
In [12]: df = df.dropna()
```

```
In [19]: print(df.columns.tolist())
```

```
['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department', 'DistanceFromHome',
'Education', 'EducationField', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'Job
Involvement', 'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus', 'MonthlyIncom
e', 'MonthlyRate', 'NumCompaniesWorked', 'OverTime', 'PercentSalaryHike', 'Performance
Rating', 'RelationshipSatisfaction', 'StockOptionLevel', 'TotalWorkingYears', 'Trainin
gTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinc
eLastPromotion', 'YearsWithCurrManager']
```

```
In [20]: num_cols = df.select_dtypes(include=['int64', 'float64']).columns
```

```
In [22]: Q1 = df[num_cols].quantile(0.25)
         Q3 = df[num_cols].quantile(0.75)
         IQR = Q3 - Q1
         df1 = df[~((df[num_cols] < (Q1 - 1.5 * IQR)) |
                    (df[num_cols] > (Q3 + 1.5 * IQR))).any(axis=1)]
```

```
In [23]: print("Before outlier removal:", df.shape)
         print("After outlier removal (df1):", df1.shape)
```

```
Before outlier removal: (1470, 31)
After outlier removal (df1): (779, 31)
```

```
In [24]: from sklearn.preprocessing import LabelEncoder
```

In [25]:

```
le = LabelEncoder()
```

In [26]:

```
for col in df1.select_dtypes(include='object').columns:
    df1[col] = le.fit_transform(df1[col])
```

```
/tmp/ipykernel_5149/2731492867.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/use
r_guide/indexing.html#returning-a-view-versus-a-copy
  df1[col] = le.fit_transform(df1[col])
/tmp/ipykernel_5149/2731492867.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/use
r_guide/indexing.html#returning-a-view-versus-a-copy
  df1[col] = le.fit_transform(df1[col])
/tmp/ipykernel_5149/2731492867.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/use
r_guide/indexing.html#returning-a-view-versus-a-copy
  df1[col] = le.fit_transform(df1[col])
/tmp/ipykernel_5149/2731492867.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/use
r_guide/indexing.html#returning-a-view-versus-a-copy
  df1[col] = le.fit_transform(df1[col])
/tmp/ipykernel_5149/2731492867.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/use
r_guide/indexing.html#returning-a-view-versus-a-copy
  df1[col] = le.fit_transform(df1[col])
/tmp/ipykernel_5149/2731492867.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/use
r_guide/indexing.html#returning-a-view-versus-a-copy
  df1[col] = le.fit_transform(df1[col])
/tmp/ipykernel_5149/2731492867.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/use
r_guide/indexing.html#returning-a-view-versus-a-copy
  df1[col] = le.fit_transform(df1[col])
/tmp/ipykernel_5149/2731492867.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/use
r_guide/indexing.html#returning-a-view-versus-a-copy
  df1[col] = le.fit_transform(df1[col])
```

In [27]:

```
X = df1.drop('Attrition', axis=1)
y = df1['Attrition']
```

In [28]:

```
from sklearn.model_selection import train_test_split
```

In [31]:

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42)
```

In [32]:

```
from sklearn.preprocessing import StandardScaler
```

In [33]:

```
scaler = StandardScaler()
```

In [34]:

```
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [35]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
2026-02-05 10:27:35.031758: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:
485] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT
when one has already been registered
2026-02-05 10:27:35.218667: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:
8454] Unable to register cuDNN factory: Attempting to register factory for plugin cuDN
N when one has already been registered
2026-02-05 10:27:35.339587: E external/local_xla/xla/stream_executor/cuda/cuda_blas.c
c:1452] Unable to register cuBLAS factory: Attempting to register factory for plugin c
uBLAS when one has already been registered
2026-02-05 10:27:35.928529: I tensorflow/core/platform/cpu_feature_guard.cc:210] This
TensorFlow binary is optimized to use available CPU instructions in performance-critic
al operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlo
w with the appropriate compiler flags.
2026-02-05 10:27:37.883446: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF
-TRT Warning: Could not find TensorRT
```

In [36]:

```
model = Sequential()
```

In [37]:

```
model.add(Dense(32, activation='relu', input_shape=(X_train.shape[1],)))
model.add(Dense(16, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

```
/home/admin1/anaconda3/lib/python3.12/site-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using
Sequential models, prefer using an `Input(shape)` object as the first layer in the mod
el instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [38]:

```
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)
```

In [39]:

```
model.fit(
    X_train,
    y_train,
    epochs=20,
    batch_size=16,
    verbose=1
)
```

```
Epoch 1/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.6369 - loss: 0.6463
Epoch 2/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8145 - loss: 0.5069
Epoch 3/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8345 - loss: 0.4293
Epoch 4/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8338 - loss: 0.4041
Epoch 5/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8576 - loss: 0.3587
Epoch 6/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 973us/step - accuracy: 0.8523 - loss: 0.3550
Epoch 7/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8692 - loss: 0.3342
Epoch 8/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8553 - loss: 0.3717
Epoch 9/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8770 - loss: 0.3027
Epoch 10/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8668 - loss: 0.3282
Epoch 11/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8957 - loss: 0.2755
Epoch 12/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8956 - loss: 0.2751
Epoch 13/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9047 - loss: 0.2580
Epoch 14/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8938 - loss: 0.2737
Epoch 15/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8902 - loss: 0.2829
Epoch 16/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8942 - loss: 0.2633
Epoch 17/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9035 - loss: 0.2554
Epoch 18/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9065 - loss: 0.2313
Epoch 19/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8824 - loss: 0.2524
Epoch 20/20
39/39 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9070 - loss: 0.2305
```

```
Out[39]: <keras.src.callbacks.history.History at 0x7f35b4105ac0>
```

```
In [43]: from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
         y_pred = (model.predict(X_test) > 0.5).astype(int)
         print("Accuracy:", accuracy_score(y_test, y_pred))
         print(classification_report(y_test, y_pred))
         print("Confusion Matrix:", confusion_matrix(y_test, y_pred))
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
Accuracy: 0.8461538461538461
              precision    recall  f1-score   support

           0       0.88      0.95      0.91       131
           1       0.53      0.32      0.40        25

    accuracy                           0.85       156
   macro avg       0.71      0.63      0.66       156
weighted avg       0.82      0.85      0.83       156

Confusion Matrix: [[124   7]
 [ 17   8]]
```

```
In [ ]:
```