In [1]:

```python
#Importing required libraries
import pandas as pd
import numpy as np
```

In [2]:

```python
#Importing the dataset
data_preprocessed = pd.read_csv("Absenteeism_preprocessed.csv")
data_preprocessed.head()
```

Out[2]:

	Reason Type 1	Reason Type 2	Reason Type 3	Reason Type 4	Month	Weekdays	Transportation Expense	Distance to Work	Age	Daily Work Load Average	Body Mass Index	Education	Children	Pets
0	0	0	0	1	7	1	289	36	33	239.554	30	0	2	1
1	0	0	0	0	7	1	118	13	50	239.554	31	0	1	0
2	0	0	0	1	7	2	179	51	38	239.554	31	0	0	0
3	1	0	0	0	7	3	279	5	39	239.554	24	0	2	0
4	0	0	0	1	7	3	289	36	33	239.554	30	0	2	1

In [3]:

```python
#Creating Targets
data_preprocessed['Absenteeism Time in Hours'].median()
```

Out[3]:

```
3.0
```

In [4]:

```python
#creating targets
targets = np.where(data_preprocessed['Absenteeism Time in Hours']>3,1,0)
data_preprocessed['Excessive Absenteeism'] = targets
data_with_targets = data_preprocessed.drop(['Absenteeism Time in Hours','Distance to Work','Daily
Work Load Average','Weekdays'],axis=1)
data_with_targets.head()
```

Out[4]:

	Reason Type 1	Reason Type 2	Reason Type 3	Reason Type 4	Month	Transportation Expense	Age	Body Mass Index	Education	Children	Pets	Excessive Absenteeism
0	0	0	0	1	7	289	33	30	0	2	1	1
1	0	0	0	0	7	118	50	31	0	1	0	0
2	0	0	0	1	7	179	38	31	0	0	0	0
3	1	0	0	0	7	279	39	24	0	2	0	1
4	0	0	0	1	7	289	33	30	0	2	1	0

targets.sum()/targets.shape()

In [5]:

```python
targets
```

Out[5]:

```
array([1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0,
       1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0,
       0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0,
       1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1,
       1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0,
       1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0])
```

In [6]:

```
targets.sum()
```

Out[6]:

```
319
```

In [7]:

```
targets.shape[0]
```

Out[7]:

```
700
```

In [8]:

```
targets.sum()/targets.shape[0]
```

Out[8]:

```
0.45571428571428574
```

In [9]:

```
unscaled_inputs = data_with_targets.iloc[:,:-1]
```

In [10]:

```
#Standardizing the data
from sklearn.preprocessing import StandardScaler
absenteeism_scaler = StandardScaler()
absenteeism_scaler.fit(unscaled_inputs)
scaled_inputs = absenteeism_scaler.transform(unscaled_inputs)
scaled_inputs
```

Out[10]:

```
array([[-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
         0.88046927,  0.26848661],
       [-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
        -0.01928035, -0.58968976],
       [-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
        -0.91902997, -0.58968976],
       ...,
       [ 1.73205081, -0.09298136, -0.31448545, ...,  2.23224237,
        -0.91902997, -0.58968976],
       [-0.57735027, -0.09298136, -0.31448545, ...,  2.23224237,
        -0.91902997, -0.58968976],
       [-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
        -0.01928035,  0.26848661]])
```

In [11]:

```
from sklearn.model_selection import train_test_split
train_test_split(scaled_inputs,targets)
```

Out[11]:

```
[array([[ 1.73205081, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.91902997, -0.58968976],
        [-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.91902997, -0.58968976],
        [ 1.73205081, -0.09298136, -0.31448545, ...,  2.23224237,
         -0.91902997, -0.58968976],
        ...,
        [ 1.73205081, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.01928035,  1.12666297],
        [-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.91902997, -0.58968976],
        [-0.57735027, -0.09298136, -0.31448545, ...,  2.23224237,
         -0.91902997, -0.58968976]]),
 array([[-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.91902997, -0.58968976],
        [-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.01928035,  1.12666297],
        [ 1.73205081, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.91902997, -0.58968976],
        ...,
        [-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.91902997, -0.58968976],
        [ 1.73205081, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.91902997, -0.58968976],
        [-0.57735027, -0.09298136, -0.31448545, ..., -0.44798003,
         -0.01928035,  1.12666297]]),
 array([1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
        1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,
        1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1,
        0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0,
        0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
        0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0,
        1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0,
        0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0,
        1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1,
        1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0,
        0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0,
        1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1,
        1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
        0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1,
        1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1,
        0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1,
        0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0,
        0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0,
        1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1,
        0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1,
        1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
        0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1,
        0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0,
        1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0]),
 array([0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0,
        0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1,
        0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0,
        0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1,
        1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1,
        1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0,
        0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1,
        0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0])]
```

In [12]:

```
#splitting the data into train and test
x_train,x_test,y_train,y_test = train_test_split(scaled_inputs,targets,train_size=0.8,random_state=
20)
```

In [13]:

```
print(x_train.shape,y_train.shape)
```

```
(560, 11) (560,)
```

In [22]:

```
#performing logistic regression
from sklearn.linear_model import LogisticRegression
reg = LogisticRegression()
reg.fit(x_train,y_train)
reg.score(x_train,y_train)
```

Out[22]:

```
0.7839285714285714
```

In [15]:

```
#finding the intercept
reg.intercept_
```

Out[15]:

```
array([-0.22375632])
```

In [16]:

```
#finding the coefficient
reg.coef_
```

Out[16]:

```
array([[ 2.0695214 ,  0.33581057,  1.56203942,  1.31277993,  0.18469755,
         0.69126151, -0.19828303,  0.32435951, -0.12665599,  0.37017705,
        -0.32535864]])
```

In [17]:

```
feature_names = unscaled_inputs.columns.values
```

In [18]:

```
summary_table = pd.DataFrame(columns=['Features'],data=feature_names)
summary_table
```

Out[18]:

	Features
0	Reason Type 1
1	Reason Type 2
2	Reason Type 3
3	Reason Type 4
4	Month
5	Transportation Expense

	Features
6	Age
7	Body Mass Index
8	Education
9	Children
10	Pets

In [19]:

```
summary_table['Cofficients'] = np.transpose(reg.coef_)
summary_table
```

Out[19]:

	Features	Cofficients
0	Reason Type 1	2.069521
1	Reason Type 2	0.335811
2	Reason Type 3	1.562039
3	Reason Type 4	1.312780
4	Month	0.184698
5	Transportation Expense	0.691262
6	Age	-0.198283
7	Body Mass Index	0.324360
8	Education	-0.126656
9	Children	0.370177
10	Pets	-0.325359

In [20]:

```
summary_table.index = summary_table.index+1
summary_table
```

Out[20]:

	Features	Cofficients
1	Reason Type 1	2.069521
2	Reason Type 2	0.335811
3	Reason Type 3	1.562039
4	Reason Type 4	1.312780
5	Month	0.184698
6	Transportation Expense	0.691262
7	Age	-0.198283
8	Body Mass Index	0.324360
9	Education	-0.126656
10	Children	0.370177
11	Pets	-0.325359

In [21]:

```
summary_table.loc[0] = ['Intercept',reg.intercept_[0]]
summary_table = summary_table.sort_index()
summary_table
```

Out[21]:

	Features	Cofficients

	Features	Cofficients
0	Intercept	-0.423756
1	Reason Type 1	2.069521
2	Reason Type 2	0.335811
3	Reason Type 3	1.562039
4	Reason Type 4	1.312780
5	Month	0.184698
6	Transportation Expense	0.691262
7	Age	-0.198283
8	Body Mass Index	0.324360
9	Education	-0.126656
10	Children	0.370177
11	Pets	-0.325359

In [23]:

```
reg.score(x_test,y_test)
```

Out[23]:

```
0.7357142857142858
```

In [30]:

```
predicted_proba = reg.predict_proba(x_test)[:,:-1]
predicted_proba
```

Out[30]:

```
array([[0.70804629],
       [0.57178351],
       [0.39895299],
       [0.78714111],
       [0.06691516],
       [0.31109284],
       [0.28582378],
       [0.08093986],
       [0.79926793],
       [0.7496687 ],
       [0.46772305],
       [0.18449682],
       [0.04086632],
       [0.75596678],
       [0.23709315],
       [0.54140798],
       [0.53354387],
       [0.52039975],
       [0.40507469],
       [0.02759587],
       [0.70148836],
       [0.78714111],
       [0.40545618],
       [0.40545618],
       [0.17391667],
       [0.75362875],
       [0.48817233],
       [0.88053255],
       [0.13059235],
       [0.78714111],
       [0.62247425],
       [0.30437845],
       [0.28895164],
       [0.44157966],
       [0.78714111],
       [0.43705325],
       [0.79067389],
       [0.18986981],
       [0.58710261],
       [0.36998588],
       [0.79584274],
       [0.57848411],
       [0.77268351],
       [0.89513347],
       [0.14276767],
       [0.34529847],
       [0.29370238],
       [0.70804629],
       [0.78792994],
       [0.80427711],
       [0.43450126],
       [0.92225898],
       [0.31109284],
       [0.73029003],
       [0.14019617],
       [0.54578776],
       [0.06949094],
       [0.78211875],
       [0.91468256],
       [0.91047636],
       [0.29440565],
       [0.3225053 ],
       [0.71278024],
       [0.13921991],
       [0.81010673],
       [0.72936044],
       [0.98595531],
       [0.79067389],
       [0.18707135],
       [0.71499343],
       [0.79067389],
       [0.93124347],
       [0.07140162],
       [0.52206954],
       [0.36134577],
       [0.78714111],
       [0.14962662],
       [0.26408704],
       [0.22931828],
       [0.32006017],
       [0.76328838],
       [0.98845557],
       [0.78211875],
       [0.22297761],
       [0.55457206],
       [0.90100263],
       [0.26777377],
       [0.40545618],
       [0.94848341],
       [0.30437845],
       [0.85602571],
       [0.86122209],
       [0.77817196],
       [0.72936044],
       [0.7496687 ],
       [0.12573789],
       [0.69702716],
       [0.21427718],
       [0.76029427],
       [0.81248944],
       [0.33737954],
       [0.30437845],
       [0.27918665],
       [0.24273428],
       [0.53520877],
       [0.4957175 ],
       [0.71882631],
       [0.12573789],
       [0.20662001],
       [0.86740544],
       [0.98959909],
       [0.07487408],
       [0.2494824 ],
       [0.65886996],
       [0.45006774],
       [0.42034788],
       [0.14588413],
       [0.15471885],
       [0.45730036],
       [0.69032997],
       [0.73964286],
       [0.862864  ],
       [0.14806763],
       [0.57848411],
       [0.78714111],
       [0.65838291],
       [0.79584274],
       [0.90442096],
       [0.22848616],
       [0.70148836],
       [0.35778052],
       [0.79584274],
       [0.69702716],
       [0.65838291],
       [0.74404968],
       [0.45006774],
       [0.52039975],
       [0.70412694],
       [0.76029427],
       [0.52206954]])
```

In [31]:

```
import pickle
with open('model','wb') as file:
    pickle.dump(reg,file)
with open('scaler','wb') as file:
    pickle.dump(absenteeism_scaler,file)
```

In [ ]: