```
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.decomposition import PCA
from tensorflow.keras import Sequential
from tensorflow.keras import layers
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
```

```
import warnings
warnings.filterwarnings("ignore")
```

```
df = pd.read_csv("transfusion.csv")
```

```
df
```

	Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time (months)	whether he/she donated blood in March 2007
0	2	50	12500	98	1
1	0	13	3250	28	1
2	1	16	4000	35	1
3	2	20	5000	45	1
4	1	24	6000	77	0
...	...	...	...	...	...
743	23	2	500	38	0
744	21	2	500	52	0
745	23	3	750	62	0
746	39	1	250	39	0
747	72	1	250	72	0

748 rows × 5 columns

```
df.isna().sum()
```

```
Recency (months)                              0
Frequency (times)                             0
Monetary (c.c. blood)                         0
Time (months)                                 0
whether he/she donated blood in March 2007    0
dtype: int64
```

```
x  = df.iloc[:,:-1]
```

```
y = df.iloc[:,4:]
y
```

	whether he/she donated blood in March 2007
0	1
1	1
2	1
3	1
4	0
...	...
743	0
744	0
745	0
746	0
747	0

748 rows × 1 columns

Normalize the data

```
scale = MinMaxScaler()
x_scaled = scale.fit_transform(x)
x_scaled
```

```
array([[0.02702703, 1.        , 1.        , 1.        ],
       [0.        , 0.24489796, 0.24489796, 0.27083333],
       [0.01351351, 0.30612245, 0.30612245, 0.34375   ],
       ...,
       [0.31081081, 0.04081633, 0.04081633, 0.625     ],
       [0.52702703, 0.        , 0.        , 0.38541667],
       [0.97297297, 0.        , 0.        , 0.72916667]])
```

PCA

```
pca = PCA()
x_p = pca.fit_transform(x_scaled)
x_p.shape
```

```
(748, 4)
```

```
pca.explained_variance_ratio_
```

```
    array([7.54765775e-01, 1.71897415e-01, 7.33368109e-02, 9.08020820e-33])
```

Ann Classifier

```
x_train,x_test,y_train,y_test=train_test_split(x_p,y,test_size=0.2)
```

```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)

    (598, 4)
    (598, 1)
    (150, 4)
    (150, 1)
```

```
from sklearn.neural_network import MLPClassifier
```

```
mlp = MLPClassifier()
```

```
mlp.fit(x_train,y_train)
predictions = mlp.predict(x_test)
```

```
print(accuracy_score(predictions,y_test))

    0.78
```