

In [47]:

```
import pandas as pd
```

In [48]:

```
df = pd.read_csv('exp4.csv')
df.head()
```

Out[48]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

In [49]:

```
df.drop(['PassengerId', 'Name', 'Age', 'SibSp', 'Parch', 'Ticket', 'Cabin', 'Embarked'], axis=1).head()
```

Out[49]:

	Survived	Pclass	Sex	Fare
0	0	3	male	7.2500
1	1	1	female	71.2833
2	1	3	female	7.9250
3	1	1	female	53.1000
4	0	3	male	8.0500

In [50]:

```
inputs = df.drop('Survived', axis='columns')
inputs.head()
```

Out[50]:

	Pclass	Sex	Fare
0	3	male	7.2500
1	1	female	71.2833
2	3	female	7.9250
3	1	female	53.1000
4	3	male	8.0500

In [51]:

```
inputs.Sex = inputs.Sex.map({'male':1, 'female':2})
inputs.head()
```

Out[51]:

	Pclass	Sex	Fare
0	3	1	7.2500
1	1	2	71.2833
2	3	2	7.9250
3	1	2	53.1000
4	3	1	8.0500

In [52]:

```
target = df.Survived
target.head()
```

Out[52]:

```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [53]:

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

In [54]:

```
scaler = StandardScaler()
scaler.fit(inputs)
final_inputs = scaler.transform(inputs)
print(final_inputs)

[[ 0.82737724 -0.73769513 -0.50244517]
 [-1.56610693  1.35557354  0.78684529]
 [ 0.82737724  1.35557354 -0.48885426]
 ...
 [ 0.82737724  1.35557354 -0.17626324]
 [-1.56610693 -0.73769513 -0.04438104]
 [ 0.82737724 -0.73769513 -0.49237783]]
```

In [55]:

```
from sklearn.decomposition import PCA
```

In [56]:

```
pca = PCA(n_components = 2)
pca.fit(final_inputs)
x = pca.transform(final_inputs)
```

In [57]:

```
print('Eigen Vectors = \n', pca.components_)
print('Eigen Values = \n', pca.explained_variance_)

# Transformed 2D dataset
print('Transformed 2D dataset = \n', x)
```

```
Eigen Vectors =
[[-0.65932625  0.333941  0.67362624]
 [-0.28610981 -0.93998137  0.18594676]]
Eigen Values =
[1.63005311 0.92482132]
Transformed 2D dataset =
[[-1.13031844  0.36327089]
 [ 2.01529662 -0.67982399]
 [-0.422135  -1.60183548]
 ...
 [-0.21156549 -1.5437102 ]
 [ 0.75633252  1.13324572]
 [-1.12353681  0.36514288]]
```

In [58]:

```
pca = PCA(n_components = 1)
pca.fit(final_inputs)
y = pca.transform(final_inputs)
```

In [59]:

```
print('Eigen Vectors = \n', pca.components_)
print('Eigen Values = \n', pca.explained_variance_)

# Transformed 1D dataset
print('Transformed 1D dataset = \n', y)
```

```
Eigen Vectors =
[[-0.65932625  0.333941    0.67362624]]
Eigen Values =
[1.63005311]
Transformed 1D dataset =
[[-1.13031844e+00]
 [ 2.01529662e+00]
 [-4.22135000e-01]
 [ 1.76867192e+00]
 [-1.11946783e+00]
 [-1.11392996e+00]
 [ 1.05285915e+00]
 [-9.42806456e-01]
 [-3.78620014e-01]
 [ 6.67277569e-01]
 [-3.03117450e-01]
 [ 1.40856754e+00]
 [-1.11946783e+00]
 [-8.04461270e-01]
 [ 1.33005311e+00]]
```

In [60]:

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

In [61]:

```
def split_df(inputs, target):
    x_train, x_test, y_train, y_test = train_test_split(inputs, target, test_size = 0.3)
    return x_train, x_test, y_train, y_test
```

In [62]:

```
from sklearn.linear_model import LogisticRegression
```

In [63]:

```
def accuracy(x_train, x_test, y_train, y_test):
    model = LogisticRegression()
    model.fit(x_train, y_train)
    return model.score(x_test, y_test)
```

In [64]:

```
import numpy as np
```

In [65]:

```
df_x = pd.DataFrame(x)
df_x.head()
```

Out[65]:

	0	1
0	-1.130318	0.363271
1	2.015297	-0.679824
2	-0.422135	-1.601835
3	1.768672	-0.747902
4	-1.119468	0.366266

In [66]:

```
df_y = pd.DataFrame(y)
df_y.head()
```

Out[66]:

	0
0	-1.130318
1	2.015297
2	-0.422135
3	1.768672
4	-1.119468

In [67]:

```
x_train, x_test, y_train, y_test = split_df(inputs,target)
acc_1 = accuracy(x_train, x_test, y_train, y_test)
print("Original Dataset Accuracy:", acc_1)
```

Original Dataset Accuracy: 0.7350746268656716

In [68]:

```
x_train, x_test, y_train, y_test = split_df(df_x,target)
acc_2 = accuracy(x_train, x_test, y_train, y_test)
print("2D Dataset Accuracy:",acc_2)
```

2D Dataset Accuracy: 0.7798507462686567

In [69]:

```
x_train, x_test, y_train, y_test = split_df(df_y,target)
acc_3 = accuracy(x_train, x_test, y_train, y_test)
print("1D Dataset Accuracy:",acc_3)
```

1D Dataset Accuracy: 0.746268656716418

