

In [1]:

```
import pandas as pd
pharma_data = pd.read_csv('https://raw.githubusercontent.com/dphi-official/Datasets/master/pharma_data/Training_set_begs.csv')
```

In [2]:

```
pharma_data["Treated_with_drugs"].fillna(method='ffill',inplace=True)
pharma_data.drop(["A","B","C","D","E","F","Z","Number_of_prev_cond"],axis=1,inplace=True)
```

In []:

In [3]:

```
pharma_data.isnull().sum()
pharma_data1=pharma_data.drop(['Patient_ID'],axis=1)
pharma_data1.isnull().sum()
```

Out[3]:

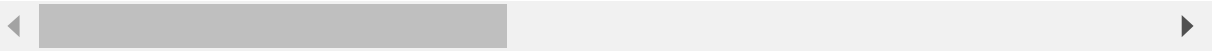
```
ID_Patient_Care_Situation      0
Diagnosed_Condition            0
Treated_with_drugs             0
Patient_Age                   0
Patient_Body_Mass_Index        0
Patient_Smoker                 0
Patient_Rural_Urban            0
Patient_mental_condition       0
Survived_1_year                0
dtype: int64
```

In [4]:

```
d1_treated_with_drugs=pd.get_dummies(pharma_data["Treated_with_drugs"],prefix='Drug')
display(d1_treated_with_drugs.head())
```

	Drug_DX1	Drug_DX1 DX2	Drug_DX1 DX2 DX3	Drug_DX1 DX2 DX3 DX4	Drug_DX1 DX2 DX3 DX4 DX5	Drug_DX1 DX2 DX3 DX5	Drug_DX1 DX2 DX4	Drug_DX1 DX2 DX4 DX5	Di
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	1	0	0	0	0	0	0	0	
4	1	0	0	0	0	0	0	0	

5 rows × 32 columns



In [5]:

```
d2_patient_smoker=pd.get_dummies(pharma_data['Patient_Smoker'])  
d2_patient_smoker=d2_patient_smoker.drop(['Cannot say'],axis=1)  
d2_patient_smoker.head()
```

Out[5]:

	NO	YES
0	0	1
1	0	1
2	0	1
3	1	0
4	0	0

In [6]:

```
d3_patient_rural_urban=pd.get_dummies(pharma_data["Patient_Rural_Urban"])  
d3_patient_rural_urban.head()
```

Out[6]:

	RURAL	URBAN
0	0	1
1	1	0
2	1	0
3	0	1
4	1	0

In [7]:

```
d4_mental=pd.get_dummies(pharma_data["Patient_mental_condition"])  
d4_mental.sum()
```

Out[7]:

```
Stable      23097  
dtype: int64
```

In [8]:

```
d4=pd.concat([d2_patient_smoker,d3_patient_rural_urban],axis=1)
d5=pd.concat([d4,d1_treated_with_drugs],axis=1)
d5.columns
```

Out[8]:

```
Index(['NO', 'YES', 'RURAL', 'URBAN', 'Drug_DX1 ', 'Drug_DX1 DX2 ',
      'Drug_DX1 DX2 DX3 ', 'Drug_DX1 DX2 DX3 DX4 ',
      'Drug_DX1 DX2 DX3 DX4 DX5 ', 'Drug_DX1 DX2 DX3 DX5 ',
      'Drug_DX1 DX2 DX4 ', 'Drug_DX1 DX2 DX4 DX5 ', 'Drug_DX1 DX2 DX5 ',
      'Drug_DX1 DX3 ', 'Drug_DX1 DX3 DX4 ', 'Drug_DX1 DX3 DX4 DX5 ',
      'Drug_DX1 DX3 DX5 ', 'Drug_DX1 DX4 ', 'Drug_DX1 DX4 DX5 ',
      'Drug_DX1 DX5 ', 'Drug_DX2 ', 'Drug_DX2 DX3 ', 'Drug_DX2 DX3 DX4 ',
      'Drug_DX2 DX3 DX4 DX5 ', 'Drug_DX2 DX3 DX5 ', 'Drug_DX2 DX4 ',
      'Drug_DX2 DX4 DX5 ', 'Drug_DX2 DX5 ', 'Drug_DX3 ', 'Drug_DX3 DX4 ',
      'Drug_DX3 DX4 DX5 ', 'Drug_DX3 DX5 ', 'Drug_DX4 ', 'Drug_DX4 DX5 ',
      'Drug_DX5 ', 'Drug_DX6'],
      dtype='object')
```

In [9]:

```
new_data_Frame=pd.concat([pharma_data1,d5],axis=1)
new_data_Frame.shape
```

Out[9]:

(23097, 45)

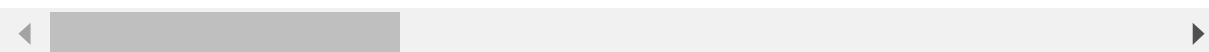
In [10]:

```
new_data_Frame.head()
```

Out[10]:

	ID_Patient_Care_Situation	Diagnosed_Condition	Treated_with_drugs	Patient_Age	Patient_Body
0	22374	8	DX6	56	
1	18164	5	DX2	36	
2	6283	23	DX6	48	
3	5339	51	DX1	5	
4	33012	0	DX1	128	

5 rows × 45 columns



In [11]:

```
new_data_Frame.drop(["Patient_Smoker"],axis=1,inplace=True)
```

In [12]:

```
new_data_Frame.shape
```

Out[12]:

```
(23097, 44)
```

In [13]:

```
new_data_Frame.drop(["Survived_1_year"],axis=1,inplace=True)
```

In [14]:

```
new_data_Frame.drop(["Patient_Rural_Urban","Patient_mental_condition","Treated_with_drugs"],axis=1,inplace=True)
```

In [15]:

```
X=new_data_Frame  
y=pharma_data["Survived_1_year"]
```

In [16]:

```
new_data_Frame.shape
```

Out[16]:

```
(23097, 40)
```

In [17]:

```
X.isnull().shape
```

Out[17]:

```
(23097, 40)
```

In [18]:

```
from sklearn.model_selection import train_test_split  
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=4)
```

In [19]:

```
from sklearn.linear_model import LogisticRegression  
log_model = LogisticRegression()
```

In [20]:

```
log_model.fit(X_train,y_train)
```

```
C:\Users\rupes\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:
940: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

Out[20]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1, l1_ratio=None, max_iter=100,
                    multi_class='auto', n_jobs=None, penalty='l2',
                    random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                    warm_start=False)
```

In [21]:

```
y_pred=log_model.predict(X_test)
```

In [22]:

```
y_pred
```

Out[22]:

```
array([0, 0, 1, ..., 1, 1, 1], dtype=int64)
```

In [23]:

```
from sklearn.metrics import accuracy_score
```

In [24]:

```
accuracy_score(y_test,y_pred)
```

Out[24]:

```
0.7088023088023088
```

In [25]:

```
test_new = pd.read_csv('https://raw.githubusercontent.com/dphi-official/Datasets/master/ph
arma_data/Testing_set_begs.csv')
```

In [26]:

```
test_new.head()
```

Out[26]:

	ID_Patient_Care_Situation	Diagnosed_Condition	Patient_ID	Treated_with_drugs	Patient_Age
0	19150	40	3709	DX3	16
1	23216	52	986	DX6	24
2	11890	50	11821	DX4 DX5	63
3	7149	32	3292	DX6	42
4	22845	20	9959	DX3	50

In [27]:

```
test_new.drop(['A', 'B', 'C', 'D', 'E', 'F', 'Z', 'Number_of_prev_cond', 'Patient_ID'],axis=1,inplace=True)
```

In [28]:

```
test_new.head()
```

Out[28]:

	ID_Patient_Care_Situation	Diagnosed_Condition	Treated_with_drugs	Patient_Age	Patient_Body
0	19150	40	DX3	16	
1	23216	52	DX6	24	
2	11890	50	DX4 DX5	63	
3	7149	32	DX6	42	
4	22845	20	DX3	50	

In [29]:

```
n1=pd.get_dummies(test_new['Treated_with_drugs'])
n2=pd.get_dummies(test_new['Patient_Smoker'])
n3=pd.get_dummies(test_new['Patient_Rural_Urban'])
n4=pd.get_dummies(test_new['Patient_mental_condition'])
```

In [30]:

```
n5=pd.concat([n1,n2,n3,n4],axis=1)
n5.head()
```

Out[30]:

	DX1	DX1 DX2	DX1 DX2 DX3	DX1 DX2 DX3 DX4	DX1 DX2 DX3 DX4 DX5	DX1 DX2 DX3 DX5	DX1 DX2 DX4	DX1 DX2 DX4 DX5	DX1 DX2 DX5	DX1 DX3	...	DX3 DX5	DX4	DX4 DX5	DX5	DX6	NO
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	1
2	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	1
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1

5 rows × 37 columns



In [31]:

```
new1=test_new.drop(['Treated_with_drugs','Patient_Smoker','Patient_Rural_Urban','Patient_m  
ental_condition'],axis=1,inplace=True)
```

In [32]:

```
q=pd.concat([test_new,n5],axis=1)
```

In [33]:

```
q.drop("Stable",axis=1,inplace=True)
```

In [34]:

```
q.shape
```

Out[34]:

(9303, 40)

In [36]:

```
result=log_model.predict(q)
```

In [38]:

```
res=pd.DataFrame(result)
res.index=q.index
res.columns=["Survived"]
res.to_csv("P.csv")
```

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []: