Sanket Gore

Internship ID : UMIP27591

In [86]:
```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

In [87]:
```
#Load the dataset
df = pd.read_csv(r"dataset_med.csv")
```

EDA

In [88]:
```
df.head(5)
```

Out[88]:

	id	age	gender	country	diagnosis_date	cancer_stage	family_history	smoking_status
0	1	64.0	Male	Sweden	2016-04-05	Stage I	Yes	Passive Smoker
1	2	50.0	Female	Netherlands	2023-04-20	Stage III	Yes	Passive Smoker
2	3	65.0	Female	Hungary	2023-04-05	Stage III	Yes	Former Smoker
3	4	51.0	Female	Belgium	2016-02-05	Stage I	No	Passive Smoker
4	5	37.0	Male	Luxembourg	2023-11-29	Stage I	No	Passive Smoker

In [89]:
```
df.shape
```

Out[89]:
```
(890000, 17)
```

In [90]:
```
df["age"].unique()
```

Out[90]:
```
array([ 64.,  50.,  65.,  51.,  37.,  49.,  56.,  48.,  47.,  67.,  45.,
        46.,  21.,  62.,  60.,  57.,  36.,  61.,  71.,  74.,  35.,  54.,
        44.,  68.,  59.,  58.,  63.,  69.,  70.,  52.,  40.,  78.,  75.,
        72.,  42.,  53.,  39.,  66.,  41.,  43.,  55.,  38.,  30.,  34.,
        76.,  73.,  80.,  31.,  85.,  28.,  79.,  87.,  77.,  33.,  32.,
        25.,  90.,  84.,  81.,  27.,  82.,  83.,  22.,  86.,  26.,  93.,
        29.,  23.,  19.,  24.,  89.,  18.,  91.,  95.,  88.,  20.,  94.,
       101.,  15.,  92.,  16.,  17.,  10.,  14.,  99.,  13.,  97.,   9.,
        12.,  98.,   7.,  96.,   4., 104.,   8.])
```

In [91]:
```
df["country"].unique()
```

Out[91]:
```
array(['Sweden', 'Netherlands', 'Hungary', 'Belgium', 'Luxembourg',
       'Italy', 'Croatia', 'Denmark', 'Malta', 'Germany', 'Poland',
       'Ireland', 'Romania', 'Spain', 'Greece', 'Estonia', 'Cyprus',
       'France', 'Slovenia', 'Latvia', 'Portugal', 'Austria',
       'Czech Republic', 'Finland', 'Lithuania', 'Slovakia', 'Bulgaria'],
      dtype=object)
```

In [92]:
```
df["smoking_status"].unique()
```

Out[92]:
```
array(['Passive Smoker', 'Former Smoker', 'Never Smoked',
       'Current Smoker'], dtype=object)
```

In [93]:
```
df.isnull().sum()
```

Out[93]:
```
id                    0
age                   0
gender                0
country               0
diagnosis_date        0
cancer_stage          0
family_history        0
smoking_status        0
bmi                   0
cholesterol_level     0
hypertension          0
asthma                0
cirrhosis             0
other_cancer          0
treatment_type        0
end_treatment_date    0
survived              0
dtype: int64
```

In [94]:
```
df["cancer_stage"].unique()
```

Out[94]:
```
array(['Stage I', 'Stage III', 'Stage IV', 'Stage II'], dtype=object)
```

In [95]:
```
df1 = df.drop(columns = ["id", "country"], axis = 1)
```

In [96]:
```
df1.shape
```

Out[96]:
```
(890000, 15)
```

In [97]:
```
df1.columns
```

Out[97]:
```
Index(['age', 'gender', 'diagnosis_date', 'cancer_stage', 'family_history',
       'smoking_status', 'bmi', 'cholesterol_level', 'hypertension', 'asthma',
       'cirrhosis', 'other_cancer', 'treatment_type', 'end_treatment_date',
       'survived'],
      dtype='object')
```

In [98]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 890000 entries, 0 to 889999
Data columns (total 15 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   age                 890000 non-null  float64
 1   gender              890000 non-null  object 
 2   diagnosis_date      890000 non-null  object 
 3   cancer_stage        890000 non-null  object 
 4   family_history      890000 non-null  object 
 5   smoking_status      890000 non-null  object 
 6   bmi                 890000 non-null  float64
 7   cholesterol_level   890000 non-null  int64  
 8   hypertension        890000 non-null  int64  
 9   asthma              890000 non-null  int64  
 10  cirrhosis           890000 non-null  int64  
 11  other_cancer        890000 non-null  int64  
 12  treatment_type      890000 non-null  object 
 13  end_treatment_date  890000 non-null  object 
 14  survived            890000 non-null  int64  
dtypes: float64(2), int64(6), object(7)
memory usage: 101.9+ MB
```

In [99]:
```
scaler = StandardScaler()
```

In [100]:
```
l_enc = LabelEncoder()
```

In [101]:
```
for col in df1.columns:
    if df1[col].dtype == object and col not in ["diagnosis_date", "end_treatme
nt_date"]:
        df1[col] = l_enc.fit_transform(df1[col])
    # if col == ["treatment_type", "cancer_stage", "gender", "smoking_status"]
```

In [102]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 890000 entries, 0 to 889999
Data columns (total 15 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   age                 890000 non-null  float64
 1   gender              890000 non-null  int32  
 2   diagnosis_date      890000 non-null  object 
 3   cancer_stage        890000 non-null  int32  
 4   family_history      890000 non-null  int32  
 5   smoking_status      890000 non-null  int32  
 6   bmi                 890000 non-null  float64
 7   cholesterol_level   890000 non-null  int64  
 8   hypertension        890000 non-null  int64  
 9   asthma              890000 non-null  int64  
 10  cirrhosis           890000 non-null  int64  
 11  other_cancer        890000 non-null  int64  
 12  treatment_type      890000 non-null  int32  
 13  end_treatment_date  890000 non-null  object 
 14  survived            890000 non-null  int64  
dtypes: float64(2), int32(5), int64(6), object(2)
memory usage: 84.9+ MB
```

In [103]:
```
df1.head(5)
```

Out[103]:

	age	gender	diagnosis_date	cancer_stage	family_history	smoking_status	bmi	cholesterc
0	64.0	1	2016-04-05	0	1	3	29.4	
1	50.0	0	2023-04-20	2	1	3	41.2	
2	65.0	0	2023-04-05	2	1	1	44.0	
3	51.0	0	2016-02-05	0	0	3	43.0	
4	37.0	1	2023-11-29	0	0	3	19.7	

In [104]:
```
df1["days_of_treatment"] = (pd.to_datetime(df1["end_treatment_date"]) - pd.to_
datetime(df1["diagnosis_date"])).dt.days
```

In [105]:
```
df1.head(5)
```

Out[105]:

	age	gender	diagnosis_date	cancer_stage	family_history	smoking_status	bmi	cholesterc
0	64.0	1	2016-04-05	0	1	3	29.4	
1	50.0	0	2023-04-20	2	1	3	41.2	
2	65.0	0	2023-04-05	2	1	1	44.0	
3	51.0	0	2016-02-05	0	0	3	43.0	
4	37.0	1	2023-11-29	0	0	3	19.7	

In [106]:
```
df1.shape
```

Out[106]: (890000, 16)

In [107]:
```
X = df1.drop(columns = ["survived", "diagnosis_date", "end_treatment_date"], a
xis =1)
```

In [108]:
```
X.shape
```

Out[108]: (890000, 13)

In [109]:
```
Y = df1["survived"]
```

Train-test split

In [110]:
```
x_train, x_test, y_train, y_test = train_test_split(X,Y, random_state = 42, te
st_size = 0.2)
```

In [111]:
```
x_train.shape
```

Out[111]: (712000, 13)

In [112]:
```
print(x_train)
```

```
         age  gender  cancer_stage  family_history  smoking_status   bmi  \
518386  45.0       0             3               1               0  21.1
79332   46.0       0             0               1               1  29.3
615180  35.0       1             3               1               0  21.0
529637  51.0       1             3               1               1  35.0
609009  48.0       1             2               0               2  25.0
...      ...     ...           ...             ...             ...   ...
259178  52.0       1             2               0               2  24.9
365838  66.0       1             2               1               1  29.5
131932  50.0       0             1               0               0  28.0
671155  64.0       0             3               1               2  38.1
121958  54.0       0             1               0               1  35.9

        cholesterol_level  hypertension  asthma  cirrhosis  other_cancer  \
518386                200             0       0          0             0
79332                 225             1       1          0             0
615180                184             1       1          0             0
529637                253             1       0          1             0
609009                151             1       1          0             0
...                   ...           ...     ...        ...           ...
259178                218             1       1          1             0
365838                158             1       0          1             0
131932                202             1       1          0             0
671155                269             1       0          0             0
121958                262             0       0          0             0

        treatment_type  days_of_treatment
518386               3                460
79332                2                370
615180               0                461
529637               2                204
609009               0                382
...                ...                ...
259178               3                254
365838               2                454
131932               2                683
671155               2                450
121958               3                611

[712000 rows x 13 columns]
```

In [113]:
```
x_train_std = scaler.fit_transform(x_train)
```

In [114]:
```
print(x_train_std)
```

```
[[-1.0025276  -1.00084024  1.34116312 ... -0.31154005  1.34019772
   0.01396398]
 [-0.90241398 -1.00084024 -1.34134396 ... -0.31154005  0.44728688
  -0.63184937]
 [-2.00366388  0.99916046  1.34116312 ... -0.31154005 -1.3385348
   0.02113968]
 ...
 [-0.50195946 -1.00084024 -0.44717494 ... -0.31154005  0.44728688
   1.61414594]
 [ 0.89963132 -1.00084024  1.34116312 ... -0.31154005  0.44728688
  -0.05779306]
 [-0.10150495 -1.00084024 -0.44717494 ... -0.31154005  1.34019772
   1.09749526]]
```

In [115]:
```
x_test_std = scaler.fit(x_test)
```

In [116]:
```
from sklearn.linear_model import LogisticRegression
```

In [117]:
```
log_reg = LogisticRegression()
```

In [118]:
```
log_reg.fit(x_train_std, y_train)
```

Out[118]:
```
LogisticRegression
LogisticRegression()
```

In [119]:
```
y_pred = log_reg.predict(x_test)
```

```
c:\Users\sanke\anaconda3\envs\tensorflow_env\lib\site-packages\sklearn\base.p
y:432: UserWarning: X has feature names, but LogisticRegression was fitted wit
hout feature names
  warnings.warn(
```

In [120]:
```
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_s
core, accuracy_score
```

In [121]:
```
accuracy = accuracy_score(y_test, y_pred)
```

In [122]:
```
print(f"Accuracy of the model: {accuracy * 100:.2f}%")
```

```
Accuracy of the model: 77.89%
```

In [123]:
```
y_prob = log_reg.predict_proba(x_test)[:, 1]
```

```
c:\Users\sanke\anaconda3\envs\tensorflow_env\lib\site-packages\sklearn\base.p
y:432: UserWarning: X has feature names, but LogisticRegression was fitted wit
hout feature names
  warnings.warn(
```

In [124]:
```
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
Confusion Matrix:
[[138639      0]
 [ 39361      0]]
```

In [125]:
```
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Classification Report:

c:\Users\sanke\anaconda3\envs\tensorflow_env\lib\site-packages\sklearn\metrics
\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are il
l-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\Users\sanke\anaconda3\envs\tensorflow_env\lib\site-packages\sklearn\metrics
\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are il
l-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

              precision    recall  f1-score   support

           0       0.78      1.00      0.88    138639
           1       0.00      0.00      0.00     39361

    accuracy                           0.78    178000
   macro avg       0.39      0.50      0.44    178000
weighted avg       0.61      0.78      0.68    178000


c:\Users\sanke\anaconda3\envs\tensorflow_env\lib\site-packages\sklearn\metrics
\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are il
l-defined and being set to 0.0 in labels with no predicted samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [126]:
```
print("\nAUC-ROC Score:", roc_auc_score(y_test, y_prob))
```

```
AUC-ROC Score: 0.4987857754596844
```

In [ ]: