

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import rcParams
from scipy import stats

data=pd.read_csv('/content/indian_liver_patient.csv')
data.head()
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_A
0	65	Female	0.7	0.1		187
1	62	Male	10.9	5.5		699
2	62	Male	7.3	4.1		490
3	58	Male	1.0	0.4		182
4	72	Male	3.9	2.0		195

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 583 entries, 0 to 582
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```
0    Age                583 non-null    int64
1    Gender              583 non-null    object
2    Total_Bilirubin     583 non-null    float64
3    Direct_Bilirubin    583 non-null    float64
4    Alkaline_Phosphotase 583 non-null    int64
5    Alamine_Aminotransferase 583 non-null int64
6    Aspartate_Aminotransferase 583 non-null int64
7    Total_Protiens      583 non-null    float64
8    Albumin             583 non-null    float64
9    Albumin_and_Globulin_Ratio 579 non-null float64
10   Dataset            583 non-null    int64
dtypes: float64(5), int64(5), object(1)
memory usage: 50.2+ KB
```

```
data.isnull().any()
```

Age	False
Gender	False
Total_Bilirubin	False
Direct_Bilirubin	False
Alkaline_Phosphotase	False
Alamine_Aminotransferase	False
Aspartate_Aminotransferase	False
Total_Protiens	False
Albumin	False
Albumin_and_Globulin_Ratio	True
Dataset	False

```
dtype: bool
```

```
data.isnull().sum()
```

Age	0
Gender	0
Total_Bilirubin	0
Direct_Bilirubin	0
Alkaline_Phosphotase	0
Alamine_Aminotransferase	0
Aspartate_Aminotransferase	0
Total_Protiens	0
Albumin	0
Albumin_and_Globulin_Ratio	4
Dataset	0

```
dtype: int64
```

```
data['Albumin_and_Globulin_Ratio']=data.fillna(data['Albumin_and_Globulin_Ratio'].mode()[0])
data.isnull().sum
```

ValueError

Traceback (most recent call last)

<ipython-input-8-0dc72afe1add> in <cell line: 1>()
----> 1 data['Albumin_and_Globulin_Ratio']=data.fillna(data['Albumin_and_Globulin_Ratio'].mode()[0])
2 data.isnull().sum

1 frames

/usr/local/lib/python3.9/dist-packages/pandas/core/frame.py in _set_item_frame_value(self, key, value)
3773 len_cols = 1 if is_scalar(cols) else len(cols)
3774 if len_cols != len(value.columns):
-> 3775 raise ValueError("Columns must be same length as key")
3776
3777 # align right-hand-side columns if self.columns

task-3

Double-click (or enter) to edit

task-3

data.describe()

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Tc
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	
mean	44.746141	3.298799	1.486106	290.576329	80.713551	109.910806	
				989	182.620356	288.918529	
				000	10.000000	10.000000	
25%	33.000000	0.800000	0.200000	175.500000	23.000000	25.000000	
50%	45.000000	1.000000	0.300000	208.000000	35.000000	42.000000	
75%	58.000000	2.600000	1.300000	298.000000	60.500000	87.000000	
max	90.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	

Automatic saving failed. This file was updated remotely or in another tab.

Show diff

sns.distplot(data['Age'])
plt.title('Age Distribution Graph')
plt.show()

<ipython-input-10-a9533a3b6a8d>:1: UserWarning:

'distplot' is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

sns.distplot(data['Age'])

Age Distribution Graph

https://colab.research.google.com/drive/1eL6d99dYqh2X3A3P2Xci0FEdDiaoiwLO#scrollTo=y6MI_D4lxHg4&printMode=true

2/5

```
sns.countplot(data['outcome'],hue=data['gender'])
```

```
File "<ipython-input-14-216339e436ff>", line 1
sns.countplot(data['outcome'],hue=data['gender'])
^
```

SyntaxError: cannot assign to function call

SEARCH STACK OVERFLOW

```
plt.figure(figsize=(10,7))
```

```
<Figure size 1000x700 with 0 Axes>
<Figure size 1000x700 with 0 Axes>
```

```
from sklearn.preprocessing import scale
```

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

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

```
!pip install imblearn
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

colab-wheels/public/simple/

```
Collecting imblearn
  Downloading imblearn-0.0-py2.py3-none-any.whl (1.9 kB)
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.9/dist-packages (from imblearn) (0.10.1)
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.1.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (3.
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.22.4)
Installing collected packages: imblearn
Successfully installed imblearn-0.0
```

```
from imblearn.over_sampling import SMOTE
smote=SMOTE()
```

```
x_train,y_train=fit_resample(x_train,y_train)
```

task-4

```
import sklearn
from RandomForestClassifier()
model1=RandomForestClassifier()
y_predict=model1.predict(x_test)
rfc1
rfc1=accuracy(y_test,y_predict)
print(classification_report(y_test,y_predict))
```

```
from sklearn.tree import DecisionTreeClassifier
model1=DecisionTreeClassifier()
model1.fit(x_train_test,y_train_test)
y_predict=model4.predict(x_test)
dfc1=accuracy_score(y_test,y_predict)
dfc1
pd.crosstab(y_test,y_predict)
print(classification_report(y_test,y_predict))
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
model2.fit(x_train_smote,y_train_smote)
model2=KNeighborsClassifier()
y_predict=model2.predict(x_test)
knn1=(accuracy_score(y_test,y_predict))
knn1
pd.crosstab(y_test,y_predict)
print(classification_report(y_test,y_predict))
```

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

```
model5=LogisticReggredssion()
model.fit(x_train_smote,y_train_smote)
y_predict=model5.predict(x_test)
logil.accuracy_score(y_test,y_predict)
logil
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-29-131b1281d9c5> in <cell line: 1>()
----> 1 model5=LogisticReggredssion()
      2 model.fit(x_train_smote,y_train_smote)
      3 y_predict=model5.predict(x_test)
      4 logil.accuracy_score(y_test,y_predict)
      5 logil
```

NameError: name 'LogisticReggredssion' is not defined

SEARCH STACK OVERFLOW

```
import tensorflow.keras
```

```
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Dense
```

```
classifier=Sequential()
```

```
classiefier.add(Dense(units=100,activation='relu',input_dim=10))
```

```
model_history=classiefier.fit(x_train,y_train,batch_size=100,validation_split=0.2,epochs=100)
```

```
model4.predict([[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]])
```

```
def predict_exit(sample_value):
    sample_value=np.array(sample_value)
    sample_value=np_value.reshape(1,-1)
    sample_value=scale(sample_value)
    return classifier.predict(sample_value)
```

```
sample_value=[[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]
if predict_exit(sample_value)>0.5:
    print('Prediction:Liver patient')
else:
    print('Prediction:Healthy')
```

```
print('Prediction:Liver patient')
```

Prediction:Liver patient

✓ 0s completed at 7:44 PM

● ×

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)