```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
```

```
from google.colab import files
uploaded =files.upload()
```

Choose files indian_liver_patient.csv

- **indian_liver_patient.csv**(text/csv) - 23930 bytes, last modified: 21/09/2019 - 100% done

```
Saving indian_liver_patient.csv to indian_liver_patient (1).csv
```

```
data = pd.read_csv('indian_liver_patient.csv')
```

```
data.head()
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotran
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.tail()
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotr
578	60	Male	0.5	0.1	500	
579	40	Male	0.6	0.1	98	
580	52	Male	0.8	0.2	245	
581	31	Male	1.3	0.5	184	
582	38	Male	1.0	0.3	216	

```
data.describe()
```

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminot
count	583.000000	583.000000	583.000000	583.000000	
mean	44.746141	3.298799	1.486106	290.576329	
std	16.189833	6.209522	2.808498	242.937989	
min	4.000000	0.400000	0.100000	63.000000	
25%	33.000000	0.800000	0.200000	175.500000	
50%	45.000000	1.000000	0.300000	208.000000	
75%	58.000000	2.600000	1.300000	298.000000	
max	90.000000	75.000000	19.700000	2110.000000	2

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 583 entries, 0 to 582
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 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[data['Dataset']==1]
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotr
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	
...	...	...	...	...	...	
576	32	Male	15.0	8.2	289	
577	32	Male	12.7	8.4	190	
579	40	Male	0.6	0.1	98	
580	52	Male	0.8	0.2	245	
581	31	Male	1.3	0.5	184	

416 rows × 11 columns

```
data['Dataset'].unique()
```

```
array([1, 2])
```

```
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_1 = data.dropna()
```

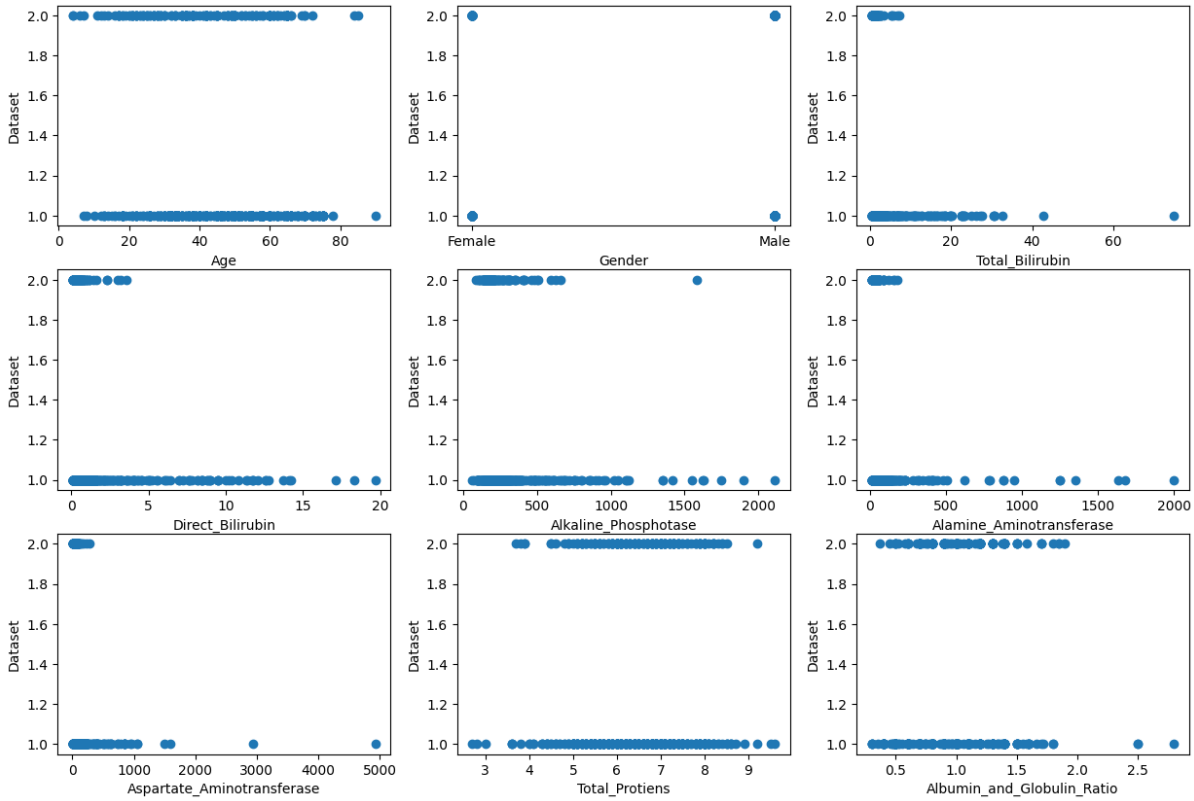
```
data_1.isnull().sum()
```

```
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    0
Dataset                       0
dtype: int64
```

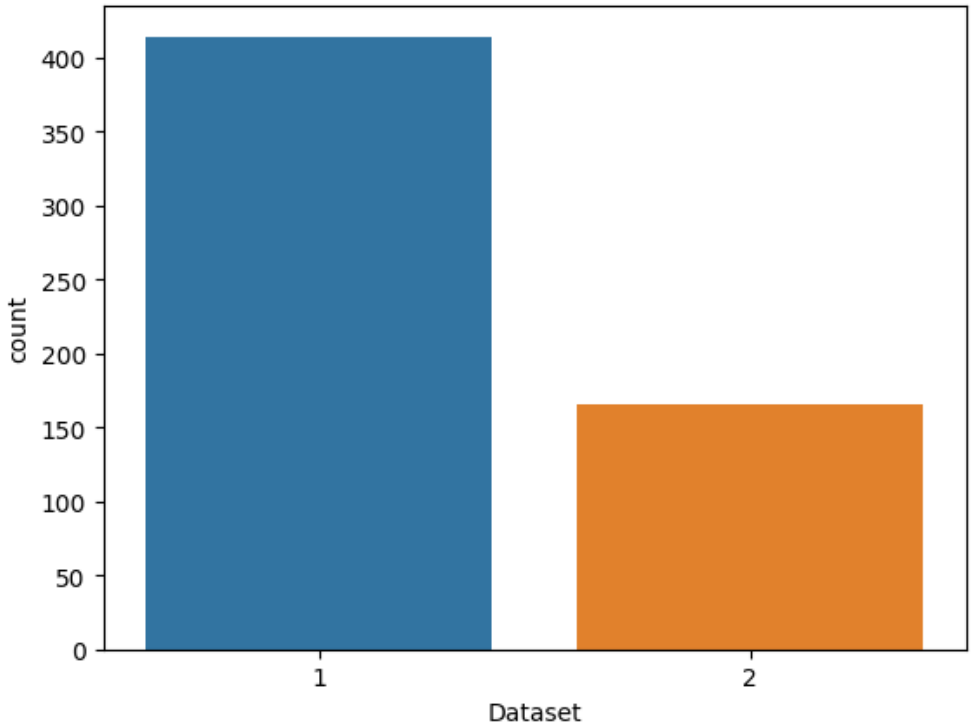
```
plt.figure(figsize=(15,10))
plt.subplot(3,3,1)
plt.scatter(data_1['Age'], data_1['Dataset'])
plt.ylabel('Dataset')
plt.xlabel('Age')


plt.subplot(3,3,2)
plt.scatter(data_1['Gender'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Gender')

plt.subplot(3,3,3)
plt.scatter(data_1['Total_Bilirubin'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Total_Bilirubin')


plt.subplot(3,3,4)
plt.scatter(data_1['Direct_Bilirubin'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Direct_Bilirubin')

plt.subplot(3,3,5)
plt.scatter(data_1['Alkaline_Phosphotase'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Alkaline_Phosphotase')

plt.subplot(3,3,6)
plt.scatter(data_1['Alamine_Aminotransferase'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Alamine_Aminotransferase')

plt.subplot(3,3,7)
plt.scatter(data_1['Aspartate_Aminotransferase'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Aspartate_Aminotransferase')

plt.subplot(3,3,8)
plt.scatter(data_1['Total_Protiens'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Total_Protiens')

plt.subplot(3,3,9)
plt.scatter(data_1['Albumin_and_Globulin_Ratio'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Albumin_and_Globulin_Ratio')
```

```
Text(0.5, 0, 'Albumin_and_Globulin_Ratio')
```



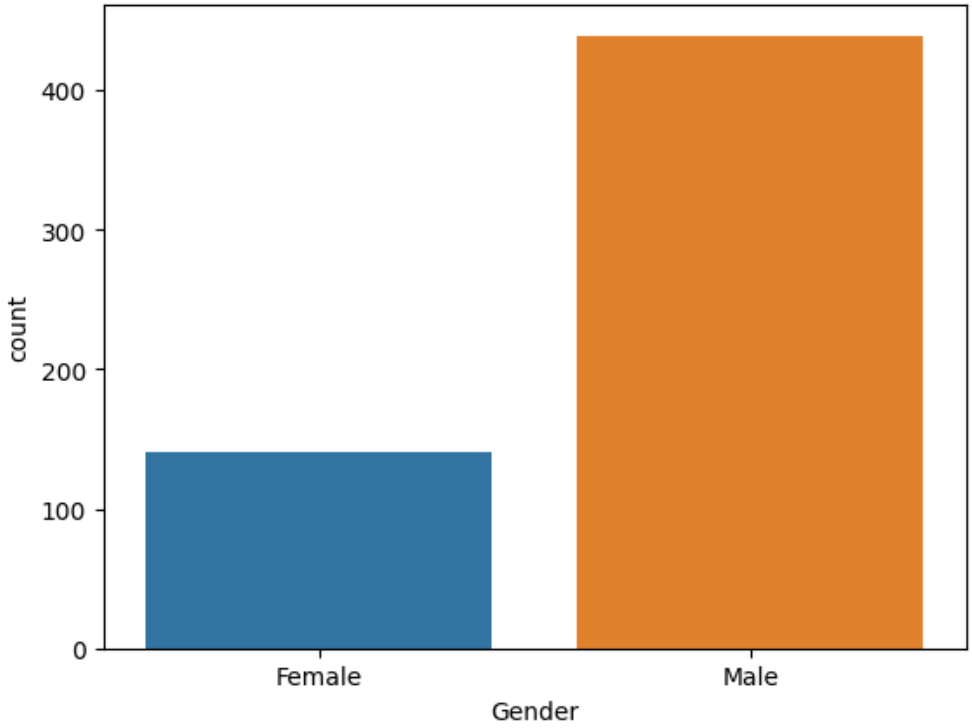

```
sns.countplot(data=data_1, x = 'Dataset')
LD,NLD=data_1['Dataset'].value_counts()
print("liver disease patients:",LD)
print("Non-liver disease patients:",NLD)
```

```
liver disease patients: 414
Non-liver disease patients: 165
```



```
sns.countplot(data=data_1, x = 'Gender', label='Count')
m,f=data_1['Gender'].value_counts()
print("No of Males:",m)
print("No of Females:",f)
```

```
No of Males: 439
No of Females: 140
```




```
# Importing the LabelEncoder library from scikit.learn
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
# Converting Textual data into numeric data
data_1['Gender'] = le.fit_transform(data_1['Gender'])
data_1.head()
```

```
<ipython-input-28-fb2e0f8fb876>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
  data_1['Gender'] = le.fit_transform(data_1['Gender'])
```

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

```
data['Gender'] = le.fit_transform(data['Gender'])
```

```
data_1.head()
```

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

```
x=data_1.iloc[:,0:-1]
y=data_1.iloc[:,-1]
```

```
x=data_1.iloc[:,0:1]
y=data_1.iloc[:,-1]
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3)
```

```
xtrain.shape
```

```
(405, 1)
```

```
xtest.shape
```

```
(174, 1)
```

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

```
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

```
svc=SVC()
RFmodel=RandomForestClassifier()
KNNmodel=KNeighborsClassifier()
```

```
from sklearn.svm import SVC
svm=SVC()
```

```
svc.fit(xtrain, ytrain)
```

```
SVC
```

```
SVCpred=svc.predict(xtest)
```

```
from sklearn.metrics import accuracy_score,confusion_matrix
# checking for accuracy score from actual data and predicted data
SVCaccuracy=accuracy_score(SVCpred, ytest)
SVCaccuracy
```

```
0.6494252873563219
```

```
SVCcm=confusion_matrix(SVCpred, ytest)
SVCcm
```

```
array([[113,  61],
       [  0,   0]])
```

```
from sklearn.ensemble import RandomForestClassifier
RFmodel=RandomForestClassifier()
```

```
RFmodel.fit(xtrain, ytrain)
```

```
RandomForestClassifier
RandomForestClassifier()
```

```
RFpred=RFmodel.predict(xtest)
```

```
RFaccuracy=accuracy_score(RFpred, ytest)
RFaccuracy
```

```
0.6666666666666666
```

```
RFcm=confusion_matrix(RFpred, ytest)
RFcm
```

```
array([[104,  49],
       [  9,  12]])
```

```
from sklearn.neighbors import KNeighborsClassifier
KNN = KNeighborsClassifier()
```

```
KNN.fit(xtrain, ytrain)
```

```
KNeighborsClassifier
KNeighborsClassifier()
```

```
KNNpred=KNN.predict(xtest)
```

```
KNNaccuracy=accuracy_score(KNNpred, ytest)
KNNaccuracy
```

```
0.6609195402298851
```

```
KNNcm=confusion_matrix(KNNpred, ytest)
KNNcm
```

```
array([[101,  47],
       [ 12,  14]])
```

```
print("Support Vector Machine Algorithm accuracy score : {value:.2f} %".format(value=SVCaccuracy*100))
print("Random Forest Algorithm accuracy score : {value:.2f} %".format(value=RFaccuracy*100))
print("K=Nearest Neighbors Algorithm accuracy score : {value:.2f} %".format(value=KNNaccuracy*100))
```

```
Support Vector Machine Algorithm accuracy score : 64.94 %
Random Forest Algorithm accuracy score : 66.67 %
K=Nearest Neighbors Algorithm accuracy score : 66.09 %
```

```
import pickle
pickle.dump(svm, open('liver_analysis_1.pkl', 'wb'))
```

```
!pip install nbconvert
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: nbconvert in /usr/local/lib/python3.9/dist-packages (6.5.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.9/dist-packages (from nbco
Requirement already satisfied: defusedxml in /usr/local/lib/python3.9/dist-packages (from nbconver
Requirement already satisfied: traitlets>=5.0 in /usr/local/lib/python3.9/dist-packages (from nbco
Requirement already satisfied: jupyter-core>=4.7 in /usr/local/lib/python3.9/dist-packages (from n
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.9/dist-packages (from
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-packages (from nbconvert
Requirement already satisfied: pygments>=2.4.1 in /usr/local/lib/python3.9/dist-packages (from nbc
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from nbc
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.9/dist-packages (from nbcon
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.9/dist-packages (from
Requirement already satisfied: bleach in /usr/local/lib/python3.9/dist-packages (from nbconvert) (
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.9/dist-packages (from nbc
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.9/dist-packages (fro
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.9/dist-packages (from nbconvert)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.9/dist-packages (from n
Requirement already satisfied: jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages (from nbconve
Requirement already satisfied: lxml in /usr/local/lib/python3.9/dist-packages (from nbconvert) (4.
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.9/dist-packages (from j
Requirement already satisfied: jupyter-client>=6.1.12 in /usr/local/lib/python3.9/dist-packages (f
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.9/dist-packages (from nbfo
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.9/dist-packages (from nbf
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.9/dist-packages (from beaut
Requirement already satisfied: webencodings in /usr/local/lib/python3.9/dist-packages (from bleach
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.9/dist-packages (from bleach->
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.9/dist-packages (from jsons
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /usr/local/lib/pyt
Requirement already satisfied: tornado>=4.1 in /usr/local/lib/python3.9/dist-packages (from jupyte
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.9/dist-packages (fro
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.9/dist-packages (from jupyter-c
```

```
!jupyter nbconvert --to html Review_of_liver_patient_analysis.ipynb
```

```
                If a relative path is given, it must be a subdirectory of the
                current directory (from which the server is run).
                See the usage documentation
                (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html-slideshow)
                for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb --to html

            Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf

            > jupyter nbconvert --to latex mynotebook.ipynb

            Both HTML and LaTeX support multiple output templates. LaTeX includes
            'base', 'article' and 'report'.  HTML includes 'basic', 'lab' and
            'classic'. You can specify the flavor of the format used.

            > jupyter nbconvert --to html --template lab mynotebook.ipynb

            You can also pipe the output to stdout, rather than a file

            > jupyter nbconvert mynotebook.ipynb --stdout

            PDF is generated via latex

            > jupyter nbconvert mynotebook.ipynb --to pdf

            You can get (and serve) a Reveal.js-powered slideshow

            > jupyter nbconvert myslides.ipynb --to slides --post serve

            Multiple notebooks can be given at the command line in a couple of
            different ways:

            > jupyter nbconvert notebook*.ipynb
            > jupyter nbconvert notebook1.ipynb notebook2.ipynb

            or you can specify the notebooks list in a config file, containing::

                c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

            > jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.
```

```
!pip install flask-ngrok
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting flask-ngrok
  Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Requirement already satisfied: Flask>=0.8 in /usr/local/lib/python3.9/dist-packages (from flask-ng
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from flask-ngro
```

```
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.9/dist-packages (from F
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0
Requirement already satisfied: Werkzeug>=2.2.2 in /usr/local/lib/python3.9/dist-packages (from Fla
Requirement already satisfied: importlib-metadata>=3.6.0 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (fr
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from reques
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (from importlib
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from Jin
Installing collected packages: flask-ngrok
Successfully installed flask-ngrok-0.0.25
```

```
from flask import Flask,render_template, request
import numpy as np
import pickle


app=Flask(__name__)

@app.route('/')
def home():
  return render_template('home.html')
  @app.route('/predict')
  def index() :
    return render_template("index.html")


@app.route('/data_predict', methods=['post'])
def predict():
  age = request.form['age']
  gender = request.form['gender']
  tb = request.form['tb']
  db = request.form['db']
  ap = request.form['ap']
  aa1 = request.form['aa1']
  aa2 = request.form['aa2']
  tp = request.form['tp']
  a = request.form['a']
  agr = request.form['agr']


  model = pickle.load(open('liver_analysis_1.pkl', 'rb'))

  prediction= model.predict(data)[0]
  if (prediction == 1):
      return render_template('noChance.html', prediction='you have a liver desease problem, you must and
  else:
      return render_template('chance.html', prediction='you dont have a liver desease problem')


if __name__ == '__main__':
  app.run()
```

```
 * Serving Flask app '__main__'
 * Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
```

Executing (1m 19s) <cell line: 1> > run() > run_simple() > serve_forever() > serve_forever() > select()