Liver patient analysis

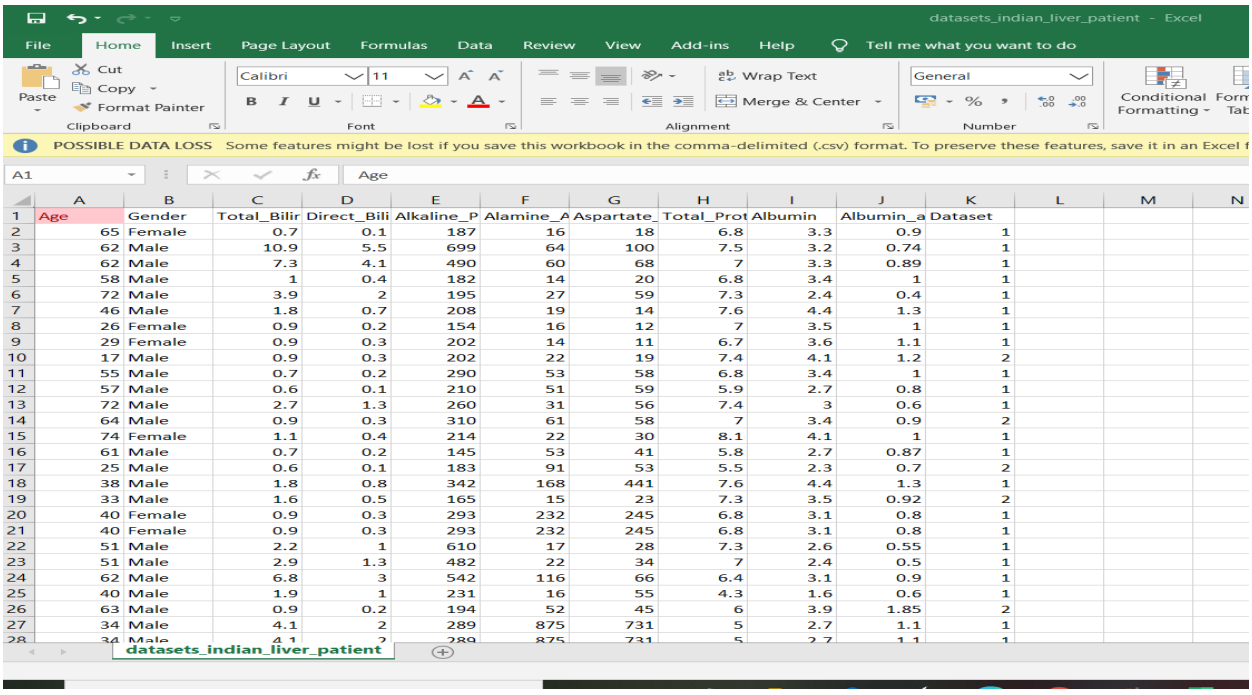
This Project examines data from liver patients concentrating on relationships between a key list of liver enzymes, proteins, age and gender using them to try and predict the likeliness of liver disease. Here we are building a model by applying various machine learning algorithms find the best accurate model. And integrate to flask based web application. User can predict the disease by entering parameters in the web application.

There are different Steps to Build a Machine Learning Mode:l

1. Gathering data
2. Data pre-processing
3. Researching the model that will be best for the type of data
4. Training and testing the model
5. Evaluation

1) Gathering of data

Here we have created data according to our project. Which contain key lists of liver enzymes, proteins, age and gender. we have stored all the data in the form of the excel sheet



	A	B	C	D	E	F	G	H	I	J	K
1	Age	Gender	Total_Bilir	Direct_Bili	Alkaline_P	Alamine_A	Aspartate_	Total_Prot	Albumin	Albumin_a	Dataset
2	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.9	1
3	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
4	62	Male	7.3	4.1	490	60	68	7	3.3	0.89	1
5	58	Male	1	0.4	182	14	20	6.8	3.4	1	1
6	72	Male	3.9	2	195	27	59	7.3	2.4	0.4	1
7	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.3	1
8	26	Female	0.9	0.2	154	16	12	7	3.5	1	1
9	29	Female	0.9	0.3	202	14	11	6.7	3.6	1.1	1
10	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.2	2
11	55	Male	0.7	0.2	290	53	58	6.8	3.4	1	1
12	57	Male	0.6	0.1	210	51	59	5.9	2.7	0.8	1
13	72	Male	2.7	1.3	260	31	56	7.4	3	0.6	1
14	64	Male	0.9	0.3	310	61	58	7	3.4	0.9	2
15	74	Female	1.1	0.4	214	22	30	8.1	4.1	1	1
16	61	Male	0.7	0.2	145	53	41	5.8	2.7	0.87	1
17	25	Male	0.6	0.1	183	91	53	5.5	2.3	0.7	2
18	38	Male	1.8	0.8	342	168	441	7.6	4.4	1.3	1
19	33	Male	1.6	0.5	165	15	23	7.3	3.5	0.92	2
20	40	Female	0.9	0.3	293	232	245	6.8	3.1	0.8	1
21	40	Female	0.9	0.3	293	232	245	6.8	3.1	0.8	1
22	51	Male	2.2	1	610	17	28	7.3	2.6	0.55	1
23	51	Male	2.9	1.3	482	22	34	7	2.4	0.5	1
24	62	Male	6.8	3	542	116	66	6.4	3.1	0.9	1
25	40	Male	1.9	1	231	16	55	4.3	1.6	0.6	1
26	63	Male	0.9	0.2	194	52	45	6	3.9	1.85	2
27	34	Male	4.1	2	289	875	731	5	2.7	1.1	1
28	34	Male	4.1	2	289	875	731	5	2.7	1.1	1

2)Dataprocessing

whenever the data is gathered from different sources it is collected in a raw format and this data isn't feasible for the analysis. Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing .

There are different steps to process your Data .

1.Import the Libraries

1) Importing files ¶

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.feature_selection import SelectFromModel
```

2 . Importing the dataset

Importing dataset

```
In [2]: dataset=pd.read_csv("datasets_indian_liver_patient.csv")
        dataset
```

3.Taking care of Missing Data

Cheaking and filling Null values

```
In [3]: dataset.isnull().any()
```

```
Out[3]: 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
```

```
In [4]: dataset['Albumin_and_Globulin_Ratio'].fillna(dataset['Albumin_and_Globulin_Ratio'].mean(),inplace=True)
        dataset
```

4.Label encoding

Lable encoding

```
In [19]: from sklearn.preprocessing import LabelEncoder
         lb=LabelEncoder()

In [20]: x[:,1]=lb.fit_transform(x[:,1])
         x

Out[20]: array([[65, 0, 0.7, ..., 6.8, 3.3, 0.9],
                [62, 1, 10.9, ..., 7.5, 3.2, 0.74],
                [62, 1, 7.3, ..., 7.0, 3.3, 0.89],
                ...,
                [52, 1, 0.8, ..., 6.4, 3.2, 1.0],
                [31, 1, 1.3, ..., 6.8, 3.4, 1.0],
                [38, 1, 1.0, ..., 7.3, 4.4, 1.5]], dtype=object)
```

5.Feature Scaling

Feature scaling

```
In [23]: from sklearn.preprocessing import StandardScaler
         sc1=StandardScaler()
         x_train=sc1.fit_transform(x_train)
         x_test=sc1.transform(x_test)
```

6.Splitting Data into Train and Test

Spliting dependent and independant data

```
In [17]: x= dataset.iloc[:,0:10].values
         x

Out[17]: array([[65, 'Female', 0.7, ..., 6.8, 3.3, 0.9],
                [62, 'Male', 10.9, ..., 7.5, 3.2, 0.74],
                [62, 'Male', 7.3, ..., 7.0, 3.3, 0.89],
                ...,
                [52, 'Male', 0.8, ..., 6.4, 3.2, 1.0],
                [31, 'Male', 1.3, ..., 6.8, 3.4, 1.0],
                [38, 'Male', 1.0, ..., 7.3, 4.4, 1.5]], dtype=object)

In [18]: y= dataset.iloc[:,10:].values
         y

Out[18]: array([[1]
```

3. Researching the model that will be best for the type of data

There are different machine learning models. We have to choose best model according our project .Best fit means model which give maximum accuracy about prediction.

In our project we have chosen Logistic regression algorithm model.

```
In [27]: from sklearn.linear_model import LogisticRegression
         regressor = LogisticRegression()
```

4. Training and testing the model

The model uses any one of the models that we had chosen in step 3. Once the model is trained we can use the same trained model to predict using the testing data i.e. the unseen data.

Splitting The dataset in to Train set and Testing set

```
In [21]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

In [22]: x_train

Out[22]: array([[55, 1, 1.1, ..., 6.2, 2.9, 0.8],
                [26, 1, 1.0, ..., 7.1, 3.7, 1.0],
                [66, 0, 0.7, ..., 6.4, 3.2, 1.0],
                ...,
                [69, 0, 0.8, ..., 8.4, 4.9, 1.4],
                [60, 1, 2.3, ..., 6.6, 3.5, 1.1],
                [26, 1, 42.8, ..., 7.5, 2.6, 0.5]], dtype=object)
```

Training and testing the model

```
In [27]: from sklearn.linear_model import LogisticRegression
         regressor = LogisticRegression()

In [28]: regressor.fit(x_train, y_train)

         C:\Users\sayali\anaconda3\lib\site-packages\sklearn\utils\validation.py:760: DataConversionWarning: A column-vector y was passe
         d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
           y = column_or_1d(y, warn=True)

Out[28]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,

In [30]: y_predict=regressor.predict(x_test)
         y_predict

Out[30]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2

In [31]: y_test
```

5.Evaluation of model

Once this is done we can calculate the performance of the logistic regression model by calculating accuracy

Evaluation of model

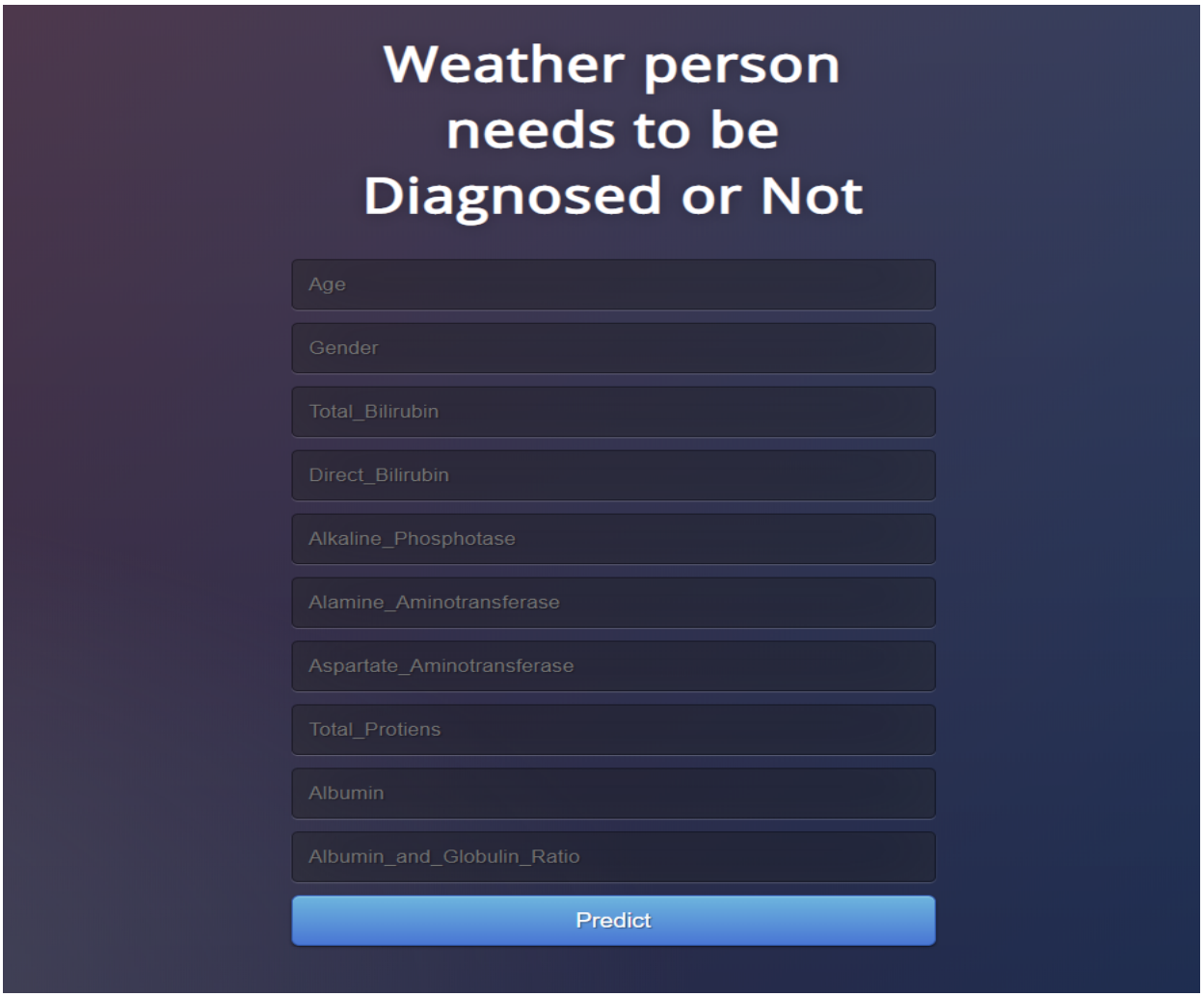
In [33]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_predict)
```

Out[33]:
```
0.7008547008547008
```

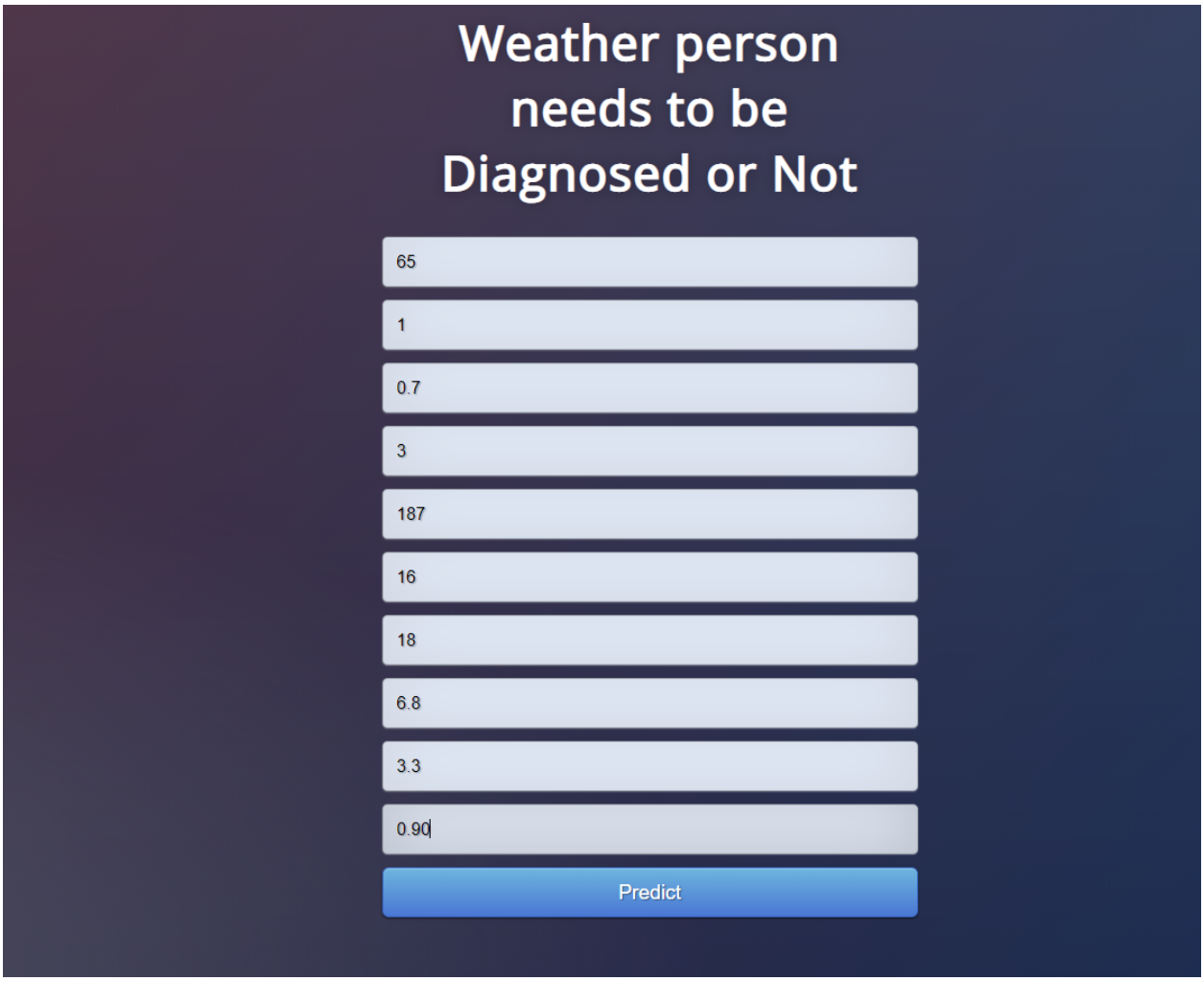
6.Application Building

Here we have built the web page for our model.

i)web page before entering data.



ii) web page after entering data



iii)Result :


Weather person needs to be Diagnosed or Not
Age
Gender
Total_Bilirubin
Direct_Bilirubin
Alkaline_Phosphotase
Alamine_Aminotransferase
Aspartate_Aminotransferase
Total_Protiens
Albumin
Albumin_and_Globulin_Ratio
Predict
personneeds to be diagnosed