

9.APPENDIX

9.1 SAMPLE CODING

```
import pandas as pd
import numpy as np

import seaborn as sns

import sklearn

import matplotlib.pyplot as plt

from matplotlib import rcParams

from scipy import stats

from pandas import DataFrame

import pip

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

data.head()

data.info()

data.isnull().any()

data.isnull().sum()

#checking for the missing data after cleansing data(

data['Albumin_and_Globulin_Ratio'].fillna(data['Albumin_and_Globulin_Ratio'].median(),inplace=True)

data.isnull().sum()

from sklearn.preprocessing import LabelEncoder

lc = LabelEncoder()
```

```

data['Gender']=lc.fit_transform(data['Gender'])

data.describe()

sns.distplot(data['Age'])

plt.title('Age distribution Graph')

plt.show()

sns.countplot(data ['Dataset'],

hue=data['Gender'])

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

sns.heatmap(data.corr(),annot=True)

from sklearn.preprocessing import scale

X_scaled=pd.DataFrame(scale(X),columns=X.columns)

X_scaled.head()

X=data.iloc[:, :-1]

y=data.Dataset

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test= train_test_split(X_scaled,y,test_size=0.2, random_state=42)

pip install imblearn

from imblearn.over_sampling import SMOTE

smote = SMOTE()

y_train.value_counts()

X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)

```

```

y_train_smote.value_counts()

from sklearn.metrics import classification_report, confusion_matrix

from sklearn.metrics import accuracy_score

from sklearn.ensemble import RandomForestClassifier

model1=RandomForestClassifier()

model1.fit(X_train_smote, y_train_smote)

rf_predict=model1.predict(X_test)

rfc1=accuracy_score(y_test,y_predict)

rfc1

pd.crosstab(y_test,y_predict)

print(classification_report(y_test,y_predict))

from sklearn.metrics import classification_report, confusion_matrix

from sklearn.metrics import accuracy_score

from sklearn.tree import DecisionTreeClassifier

model4=DecisionTreeClassifier()

model4.fit(X_train_smote, y_train_smote)

y_predict=model4.predict(X_test)

dct1=accuracy_score(y_test,y_predict)

dct1

pd.crosstab(y_test,y_predict)

print(classification_report(y_test, y_predict))

```

```

from sklearn.metrics import classification_report, confusion_matrix

from sklearn.metrics import accuracy_score

from sklearn.neighbors import KNeighborsClassifier

model2=KNeighborsClassifier()

model2.fit(X_train_smote, y_train_smote)

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.metrics import classification_report, confusion_matrix

from sklearn.metrics import accuracy_score

from sklearn.linear_model import LogisticRegression

model5=LogisticRegression()

model5.fit(X_train_smote, y_train_smote)

y_predict=model5.predict(X_test)

logi1=accuracy_score(y_test, y_predict)

logi1

pd.crosstab(y_test, y_predict)

print (classification_report (y_test, y_predict))

import tensorflow.keras

```

```

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

#initialising the ANN

classifier = Sequential()

#adding the input layer and the first hidden layer

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

#adding the second hidden layer

classifier.add(Dense(units=50, activation='relu'))

#adding the output layer

classifier.add(Dense(units=1, activation='sigmoid'))

#compiling the ANN

classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

#fitting the ANN to the training set

model_history = classifier.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]])

model1.predict([50,1,1.2,0.8,150,70,80,7.2,3.4,0.8])

classifier.save("liver.h5")

y_pred = classifier.predict(X_test)

y_pred

def predict_exit(sample_value):

    sample_value = np.array(sample_value)

```

```

sample_value = sample_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')

acc_smote= [['KNN classifier',knn1],['RandomForestClassifier',rfc1],

['DecisionTreeClassifier',dtt1],['LogisticRegression',logi1]]

Liverpatient_pred= pd.DataFrame(acc_smote, columns=['classification models','accuracy_score'])

Liverpatient_pred

from sklearn.metrics import classification_report,confusion_matrix

from sklearn.metrics import accuracy_score

plt.figure(figsize=(7,5))

plt.xticks(rotation=90)

plt.title('classification models & accuracy scores after SMOTE',fontsize=18)

sns.barplot(x="classification models", y="accuracy_score", data=Liverpatient_pred)

from sklearn.ensemble import ExtraTreesClassifier

model=ExtraTreesClassifier()

model.fit(X,y)

```

```
model.feature_importances_  
  
dd=pd.DataFrame(model.feature_importances_,index=X.columns).sort_values(0,ascending=False)  
  
dd  
  
dd.plot(kind='barh',figsize=(7,6))  
  
plt.title("FEATURE IMPORTANCE",fontsize=14)  
  
import joblib  
  
joblib.dump(model1, 'ETC.pk1')
```

9.2 SAMPLE CODING

1.INDEX.HTML

<!DOCTYPE html>

<html>

<head>

<title>LIVER PREDICTION</title>

<link rel="stylesheet" type="text/css" href="css/style.css">

```
<link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
```

</head>

<body>

<header>

<div>


```
<li class="active"><a href="index.html"><i class="fa fa-home"></i>HOME</a></li>
```

- [GO TO PREDICT](home.html)

<div class="title">

INTRODUCTION

Liver is the largest internal organ in the human body, it is essential for digesting food and releasing the

toxic element of the body and plays a major role in metabolism and serving several vital functions. The

liver is the largest glandular organ of the body. It weighs about 3 lb (1.36 kg) .The liver's main job is to

strain the blood coming from the digestive tract, before passing it to the rest of the body. The liver also

detoxifies chemicals and metabolizes drugs.
As it does so, the liver hides bile that ends up back in the

intestines. The liver also makes proteins important for blood clotting and other functions.The liver supports

almost every organ in the body and is vital for our survival. Liver diseases are diagnosed based on the liver functional test.

Some of the diseases are Wilson's disease,liver cancer, and cirrhosis.</div>

</header>

</body>

</html>

[illegible]

<label>TOTAL _BILIRUBIN:</label>

<input type="text" name="total_bilirubin" id="total_bilirubin" >

<label>DIRECT_BILIRUBIN:</label>

<input type="text" name="direct_bilirubin" id="direct_bilirubin" >

<label>ALKALINE_PHOSPOTASE:</label>

<input type="text" name="alkaline_phospotase" id="alkaline_phospotase" >

<label>ALAMINE_AMINOTRANSFERASE:</label>

<input type="text" name="alamine_aminotransferase" id="alamine_aminotransferase" >

<label>ASPARTATE_AMINOTRANSFERASE:</label>

<input type="text" name="aspartate_aminotransferase" id="aspartate_aminotransferase" >

<label>TOTAL _PROTEINS:</label>

<input type="text" name="total_proteins" id="total_proteins" >

<label>ALBUMIN:</label>


```

<input type="text" name="albumin" id="albumin" >
<br><br>
<label>ALBUMIN_AND_GLOBULIN_RATIO:</label>
<br>
<input type="text" name="albumin_and_globulin_ratio" id="albumin_and_globulin_ratio" >
<br><br>
</form>
<header>
<div class="button">
<a href="Predict.html" class="btn">PREDICT</a><center></center></a><br><br>
</div>
</header>
</body>
</html>

```

3. PREDICT.HTML

```

<!DOCTYPE html>

<html>

<head>

<title>LIVER PATIENT ANALYSIS</title>

<link rel="stylesheet" href="style2.css" type="text/css">

</head>

<div class="col-md-3"></div>

<div class="col-md-6">

```

```

<body>

<div class="row" style="margin-bottom: 477px;">

<div class="card card-body alert alert-danger"><br><br><br><br><br><center>You have a
liver desease problem,You must and should consult a docter.Take care.</center></div><br>

<div class="card card-body alert alert-success"><br><br><br><br><center>You dont have a
liver desease problem.</center></div><br><br>

<div class="row">

<div class="col-md-4"></div>

<a href="index.html"><center>Back to Index</center></a>

<div class="col-md-4"></div>

</div>

</div>

<div class="col-md-3"></div>

</div>

</body>

</html>

```

STYLE 1

```
{  
margin:0;  
padding:0;  
}  
  
body{  
background:url("liver.jpg");  
background-size:cover;  
background-position:center;  
background-repeat:no-repeat;  
}  
  
div.main{  
width:400px;  
margin:100px auto 0px auto;  
}  
  
h1{  
text-align:center;  
color:yellow;  
padding:20px;  
font-family:sans-serif;  
}  
  
div.liver{  
background-color:rgba(0,0,0,0.5);  
width:100%;  
font-size:18px;  
border-radius:35px;
```

```

border: 1px solid rgba(255,255,255,0.3);
box-shadow: 2px, 2px, 15px
  rgba(0,0,0,0.3);
color: #fff;
}
form#liver{
margin: 40px;
}
label{
font-family: sans-serif;
font-size: 18px;
font-style: italic;
}
input#name{
width: 300px;
border: 1px solid #ddd;
border-radius: 3px;
outline: 0;
padding: 7px;
background-color: #fff;
box-shadow: inset 1px 1px 5px;
}
.btn{
border: 5px solid #fff;
padding: 5px 20px;
color: yellow;
transition: 0.6s ease;

```

```
font-size: 18px;
text-decoration:center;
text align: center;
}
label,span,h1{
text-shadow:1px 1px 5px
rgba(0,0,0,0.3);
}
```

STYLE 2

```
{
margin:0;
padding:0;
}
body{
background:url("liver2.jpg");
background-size:cover;
background-position:center;
background-repeat:no-repeat;
}
div.main{
width:400px;
margin:100px auto 0px auto;
}
h1{
text align: center;
color:yellow;
padding: 20px;
```



```

font-family: sans-serif;
}
div.row{
background-color:rgba(0,0,0,0.5);
width:100%;
font-size:30px;
font_family:Times New Roman;
border-radius:35px;
border:1px solid rgba(255,255,255,0.3);
box-shadow:2px,2px,15px
rgba(0,0,0,0.3);
color:#fff;
}
form#liver{
margin:40px;
}
label{
font-family:sans-serif;
font-size:18px;
font-style:italic;
}
input#name{
width:300px;
border:1px solid #ddd;
border-radius:3px;
outline:0;
padding:7px;

```

```
background-color:#fff;
box-shadow:inset 1px 1px 5px;
}
.btn{
border: 5px solid #fff;
padding:5px 20px;
color:yellow;
transition: 0.6s ease;
font-size: 18px;
text-decoration:center;
text align: center;
}
label,span,h1{
text-shadow:1px 1px 5px
rgba(0,0,0,0.3);
}
```

9.3 SAMPLE CODING

```
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']
```

```

#coverting data into float format

data = [[Float(age), float(gender), float(tb), float(db), float(ap), float(aa1), float
(aa2), Float(tp),float(tp),float(a),float(agr)]]

# Loading model which we saved

model = pickle.load(open('liver_analysis.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()

```