A Review of Liver Patient Analysis Methods Using Machine Learning

Team ID : N M 2 0 2 3 T M I D 3 1 9 8 6

Team Lead Name : A B I R A M I J

Team Members : B H U V A N E S H W A R I P

S N E G A M

S N E K A S

U S H A M

1.1 Overview

Liver diseases averts the normal function of the liver. This disease is caused by an assortment of elements that harm the liver. Diagnosis of liver infection at the preliminary stage is important for better treatment. In today's scenario devices like sensors are used for detection of infections. Accurate classification techniques are required for automatic identification of disease samples. This disease diagnosis is very costly and complicated. Therefore, the goal of this work is to evaluate the performance of different Machine Learning algorithms in order to reduce the high cost of liver disease diagnosis.

About the Project:

This disease diagnosis is very costly and complicated. Therefore, the goal of this work is to evaluate the performance of different Machine Learning algorithms in order to reduce the high cost of liver disease diagnosis. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time.

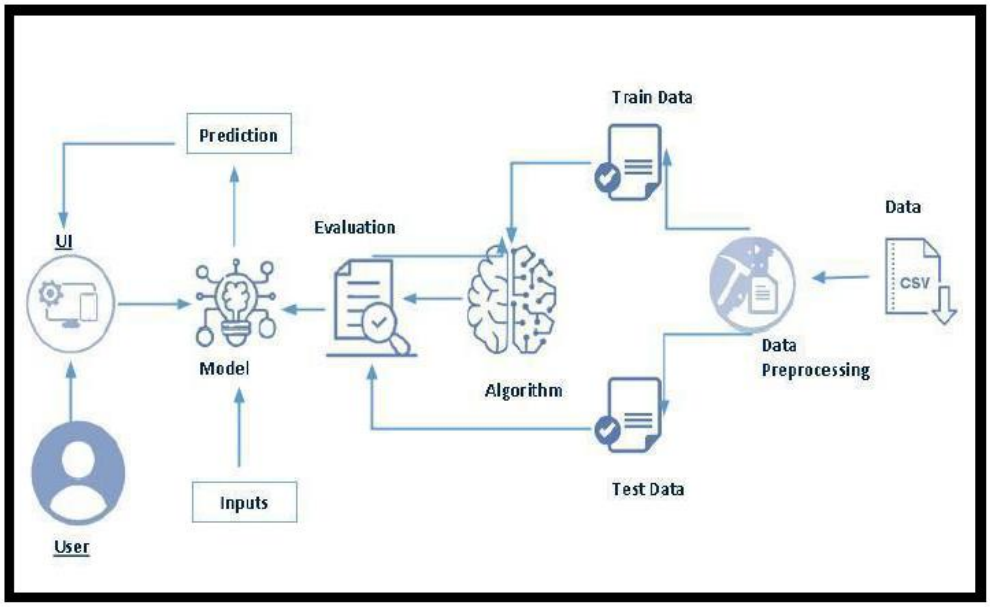
A Project Description:

In India, delays in diagnosing diseases are a major problem due to a lack of medical professionals. The typical scenario, which is mainly in rural and slightly urban areas:

1. A patient who sees a doctor with certain symptoms.
2. The doctor will perform some tests, such as blood and urine tests, depending on the symptoms.
3. The patient undergoes the above tests in the analytical laboratory.
4. The patient takes the reports back to the hospital, where they are examined and diagnosed.

The goal of this project is to reduce some of the delays caused by unnecessary detours between the hospital and the pathology laboratory. Historically, work has been done to detect the onset of heart disease, such as Parkinson's, and machine learning algorithms have been developed to predict liver disease.

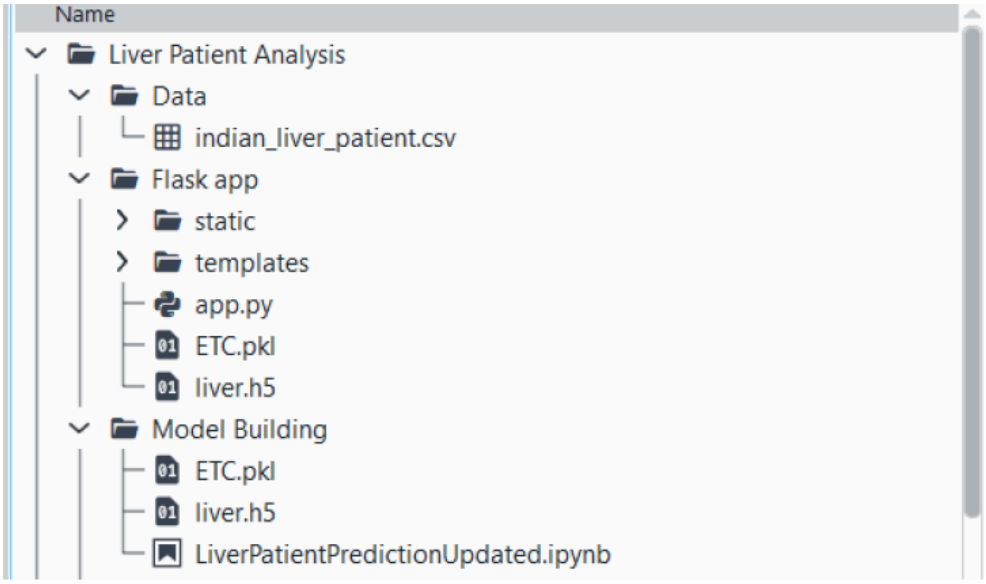
Technical Architecture:



Project Flow:
- User interacts with the UI to enter the input.
- Entered input is analysed by the model which is integrated.
- Once model analyses the input the prediction is showcased on the UI

Project Structure:

Create the Project folder which contains files as shown below



We are building a flask application which needs HTML pages stored in the templates folder and python script app.py for scripting.

- ETC.pkl is our saved model. Further we will use this model for flask integration.
- Training folder contains a model training file.

1.2 Purpose

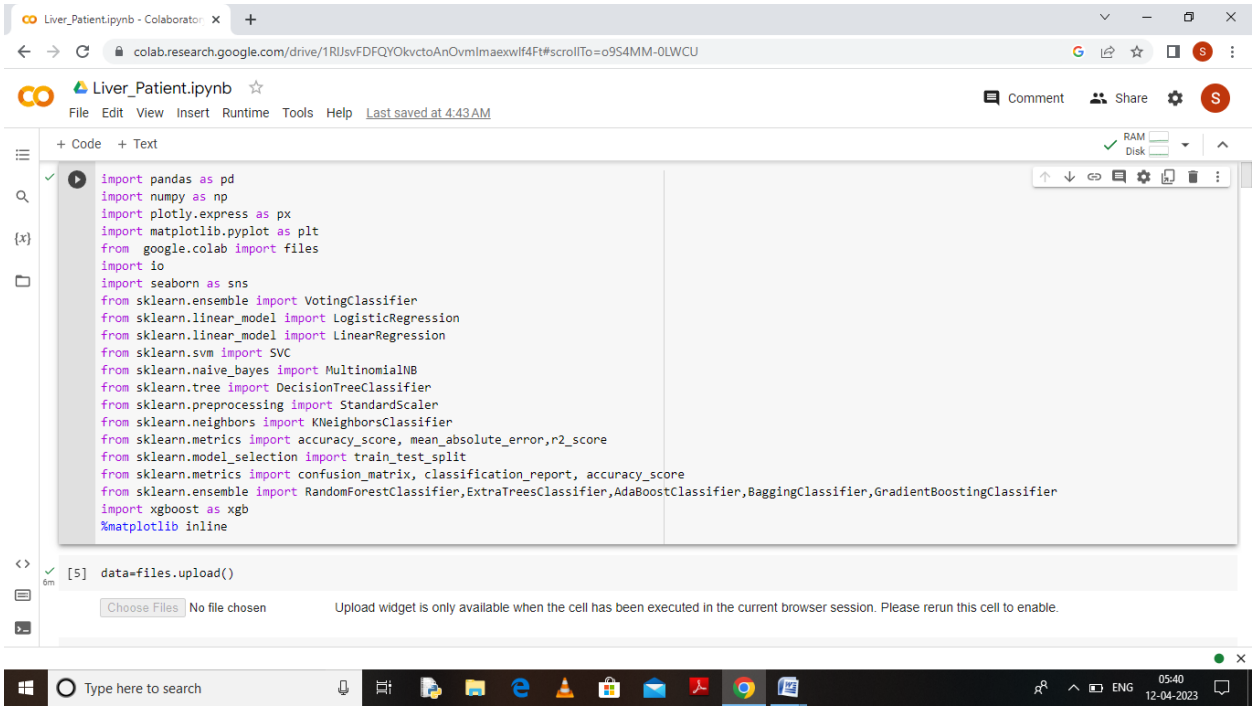
The goal of this project is to reduce some of the delays caused by unnecessary detours between the hospital and the pathology laboratory.

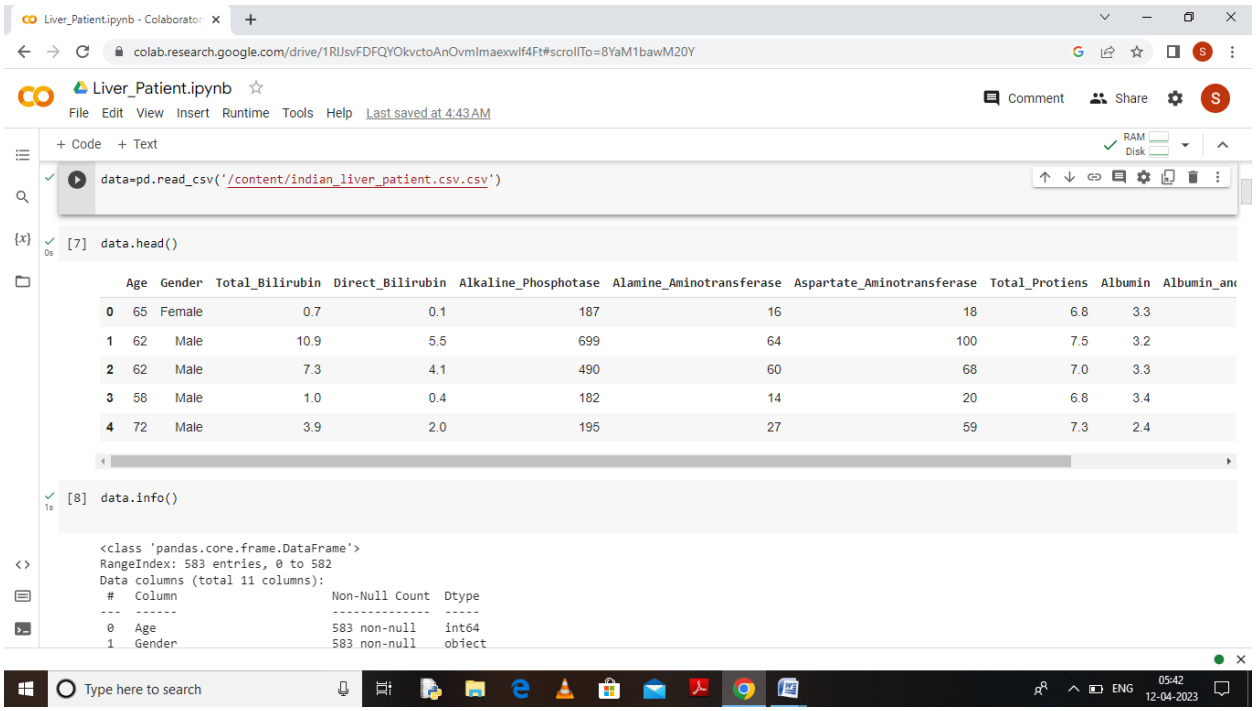
Problem Definition & Design Thinking

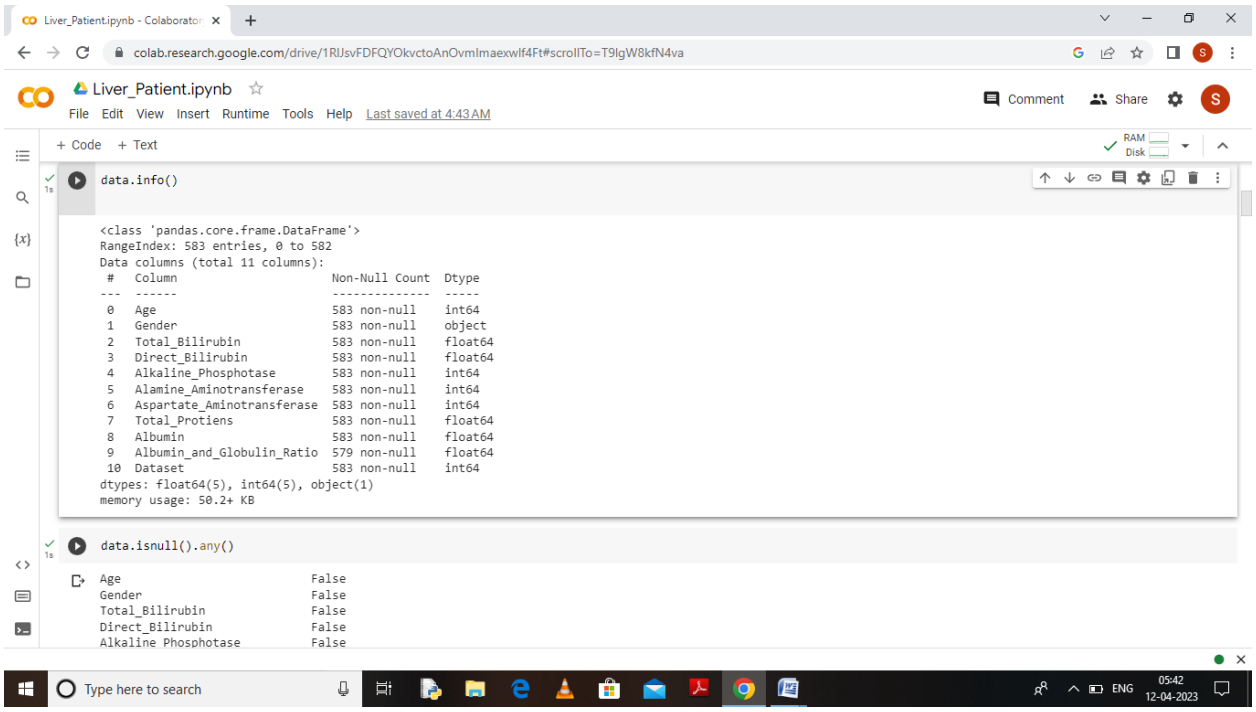
2.1 Empathy Map

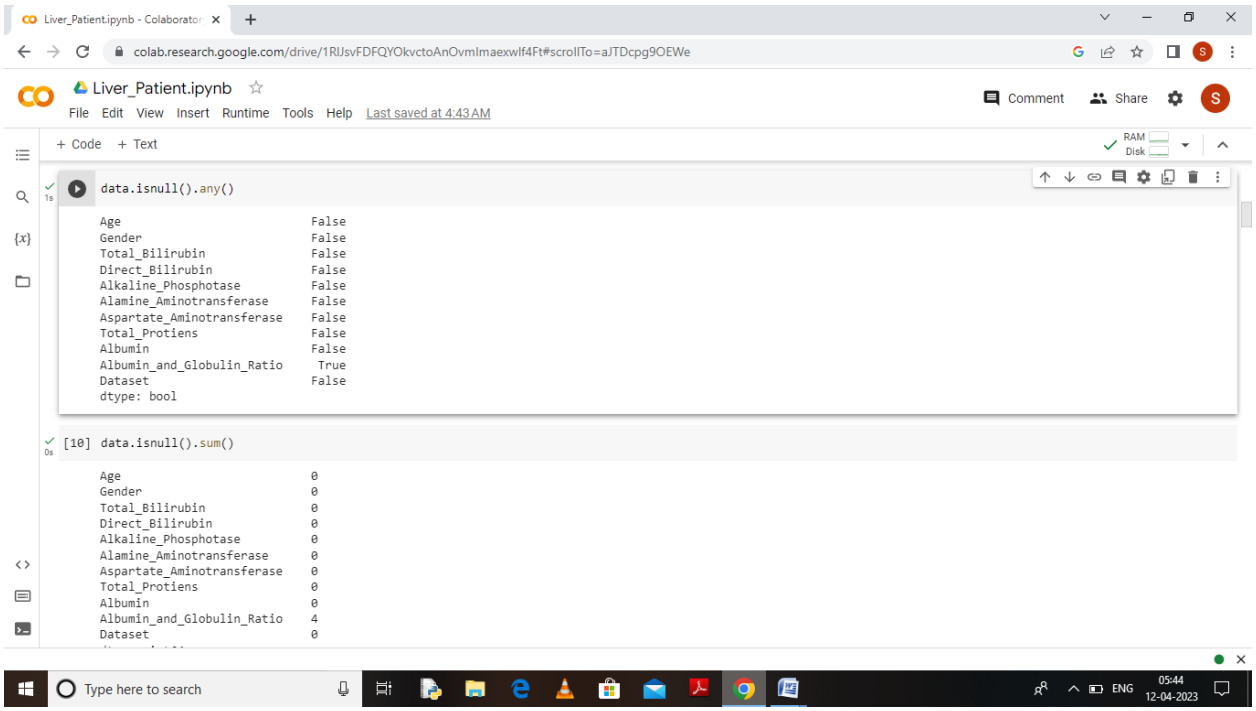
2.2 Ideation & Brainstorming Map

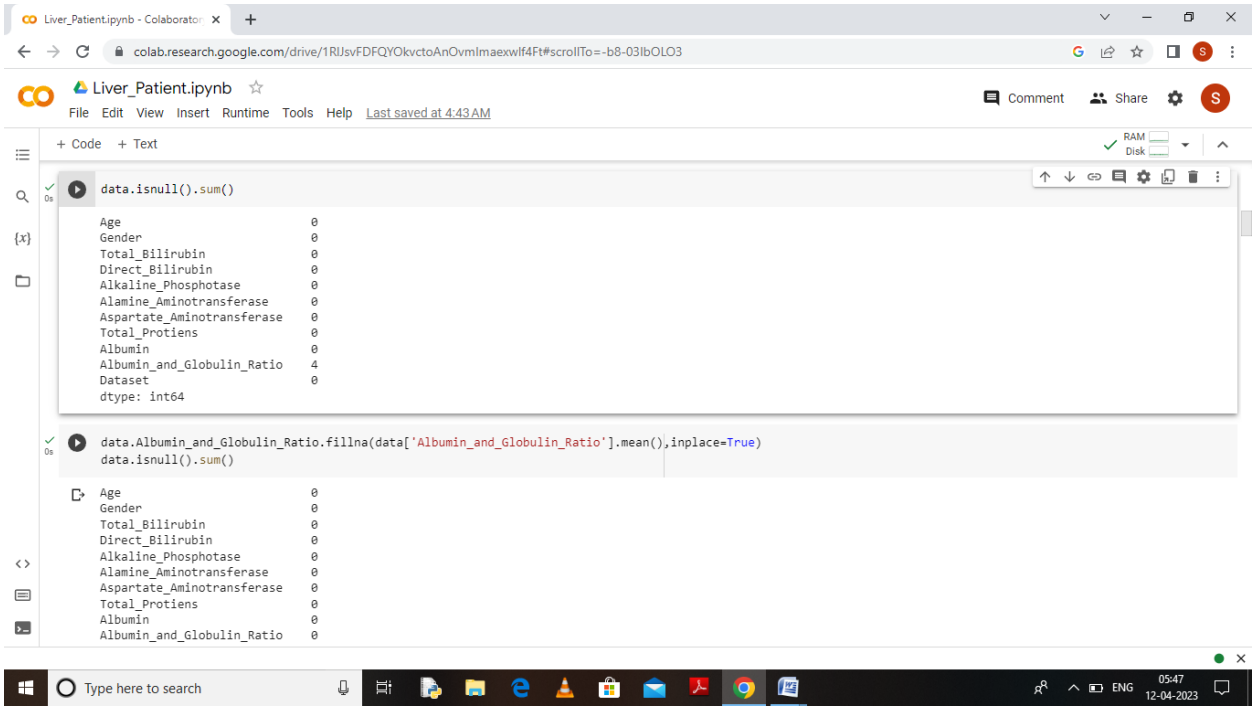
RESULT

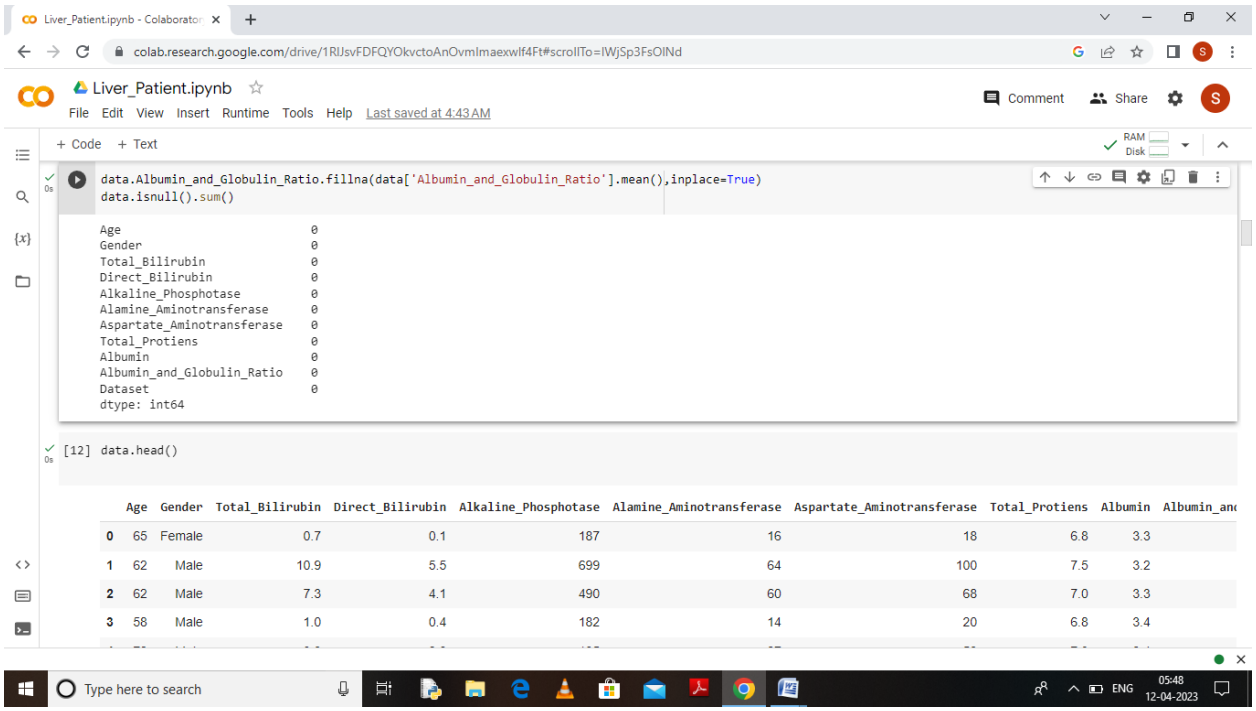


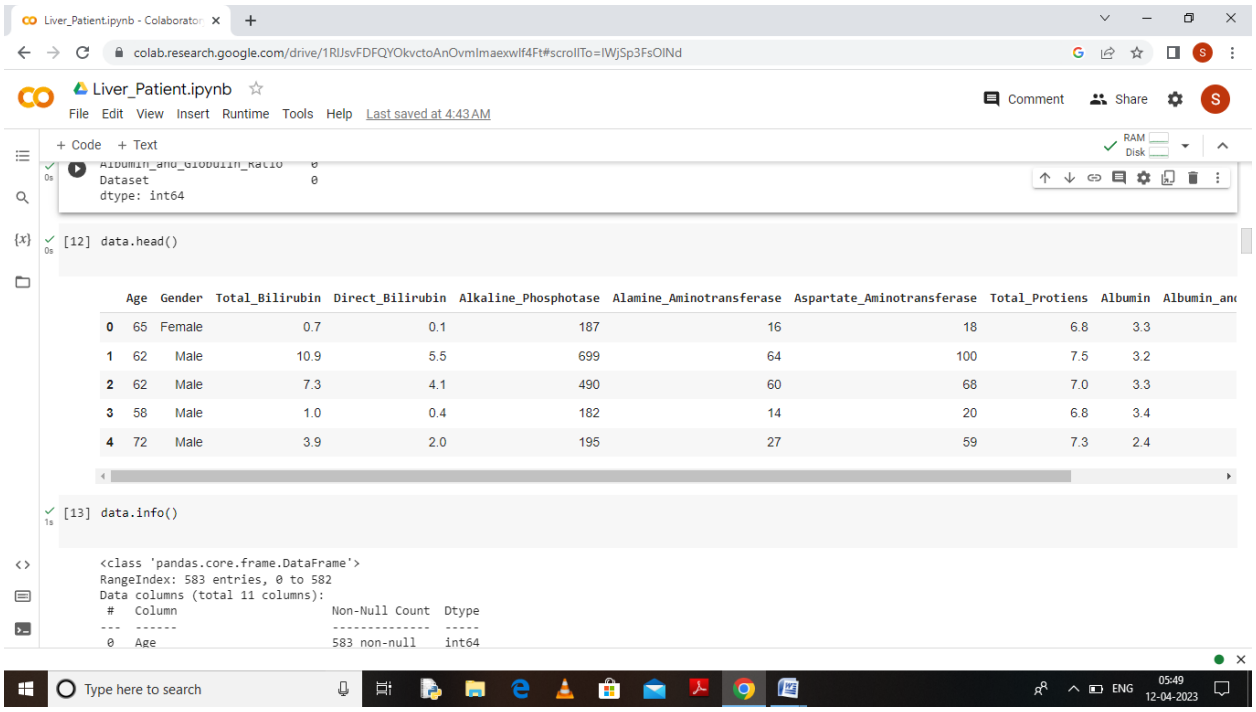

Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=8YaM1bawM20Y
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
data=pd.read_csv('/content/indian_liver_patient.csv.csv')
[7] data.head()
Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin Albumin_and
0 65 Female 0.7 0.1 187 16 18 6.8 3.3
1 62 Male 10.9 5.5 699 64 100 7.5 3.2
2 62 Male 7.3 4.1 490 60 68 7.0 3.3
3 58 Male 1.0 0.4 182 14 20 6.8 3.4
4 72 Male 3.9 2.0 195 27 59 7.3 2.4
[8] 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
Type here to search
ENG
05:42
12-04-2023

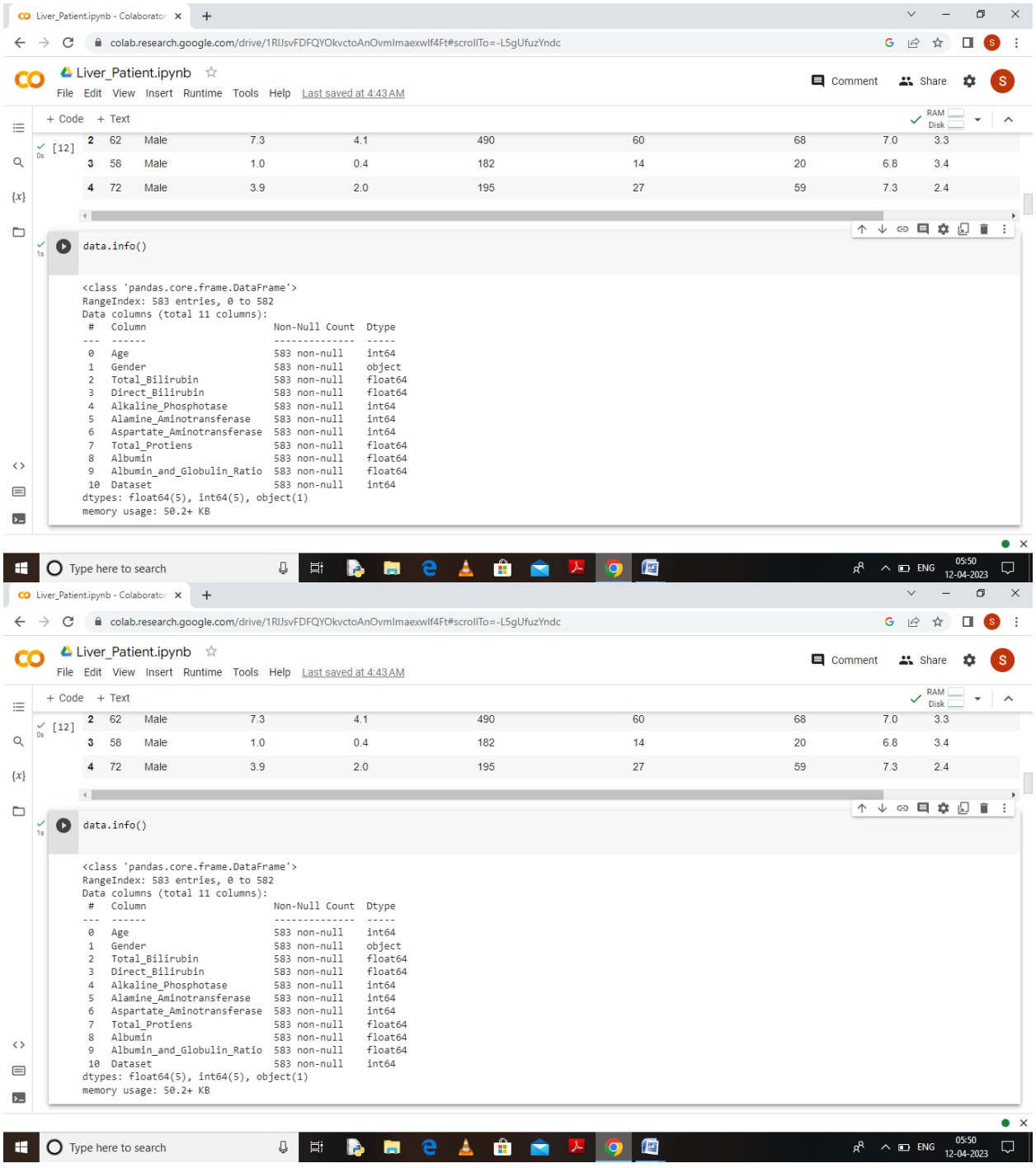

Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=T9IgW8kfN4va
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 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
Type here to search
ENG
05:42
12-04-2023


Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwlf4Ft#scrollTo=aJTDcpg9OEWe
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
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
[10] 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
Type here to search
ENG
05:44
12-04-2023


Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwlf4Ft#scrollTo=-b8-03IbOLO3
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
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.fillna(data['Albumin_and_Globulin_Ratio'].mean(),inplace=True)
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 0
Type here to search
ENG
05:47
12-04-2023


Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=IWjSp3FsOINd
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
data.Albumin_and_Globulin_Ratio.fillna(data['Albumin_and_Globulin_Ratio'].mean(),inplace=True)
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 0
Dataset 0
dtype: int64
[12] data.head()
Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin Albumin_and
0 65 Female 0.7 0.1 187 16 18 6.8 3.3
1 62 Male 10.9 5.5 699 64 100 7.5 3.2
2 62 Male 7.3 4.1 490 60 68 7.0 3.3
3 58 Male 1.0 0.4 182 14 20 6.8 3.4
Type here to search
ENG
05:48
12-04-2023


Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=IWjSp3FsOINd
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
Dataset 0
dtype: int64
[12] data.head()
Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin Albumin_and
0 65 Female 0.7 0.1 187 16 18 6.8 3.3
1 62 Male 10.9 5.5 699 64 100 7.5 3.2
2 62 Male 7.3 4.1 490 60 68 7.0 3.3
3 58 Male 1.0 0.4 182 14 20 6.8 3.4
4 72 Male 3.9 2.0 195 27 59 7.3 2.4
[13] 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
Type here to search
ENG
05:49
12-04-2023



Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=-L5gUfuzYndc
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
[12]
2 62 Male 7.3 4.1 490 60 68 7.0 3.3
3 58 Male 1.0 0.4 182 14 20 6.8 3.4
4 72 Male 3.9 2.0 195 27 59 7.3 2.4
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 583 non-null float64
10 Dataset 583 non-null int64
dtypes: float64(5), int64(5), object(1)
memory usage: 50.2+ KB
Type here to search
ENG
05:50
12-04-2023

Liver_Patient.ipynb - Colaborator

colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=skY1iR-0Y3Ya

Liver_Patient.ipynb

File Edit View Insert Runtime Tools Help Last saved at 4:43 AM

Comment Share

+ Code + Text

RAM Disk

```
 7   Total_Protiens              583 non-null    float64
 8   Albumin                     583 non-null    float64
 9   Albumin_and_Globulin_Ratio  583 non-null    float64
 10  Dataset                     583 non-null    int64
dtypes: float64(5), int64(5), object(1)
memory usage: 50.2+ KB
```

```
data.describe()
```

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Album
count	583.000000	583.000000	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	6.483190	3.141852	
std	16.189833	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.795519	
min	4.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.900000	
25%	33.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.600000	
50%	45.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.100000	
75%	58.000000	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.800000	
max	90.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.500000	

Type here to search

ENG 05:50 12-04-2023

Liver_Patient.ipynb - Colaborator

colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=5i9Q8rUYrrg2

Liver_Patient.ipynb

File Edit View Insert Runtime Tools Help Last saved at 4:43 AM

Comment Share

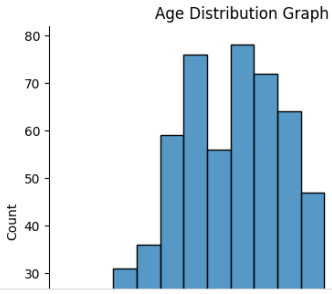
+ Code + Text

RAM Disk

```
from sklearn.preprocessing import LabelEncoder
lc = LabelEncoder()
data['Gender']= lc.fit_transform(data['Gender'])
```

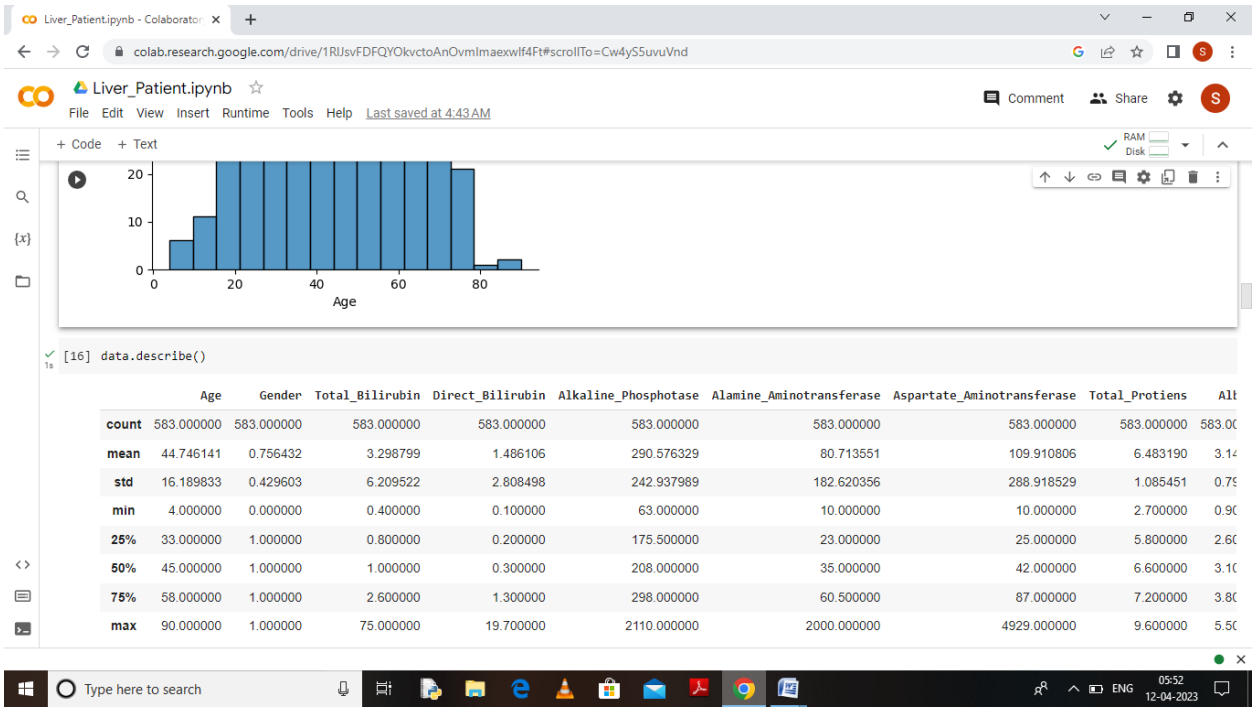
[17]

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

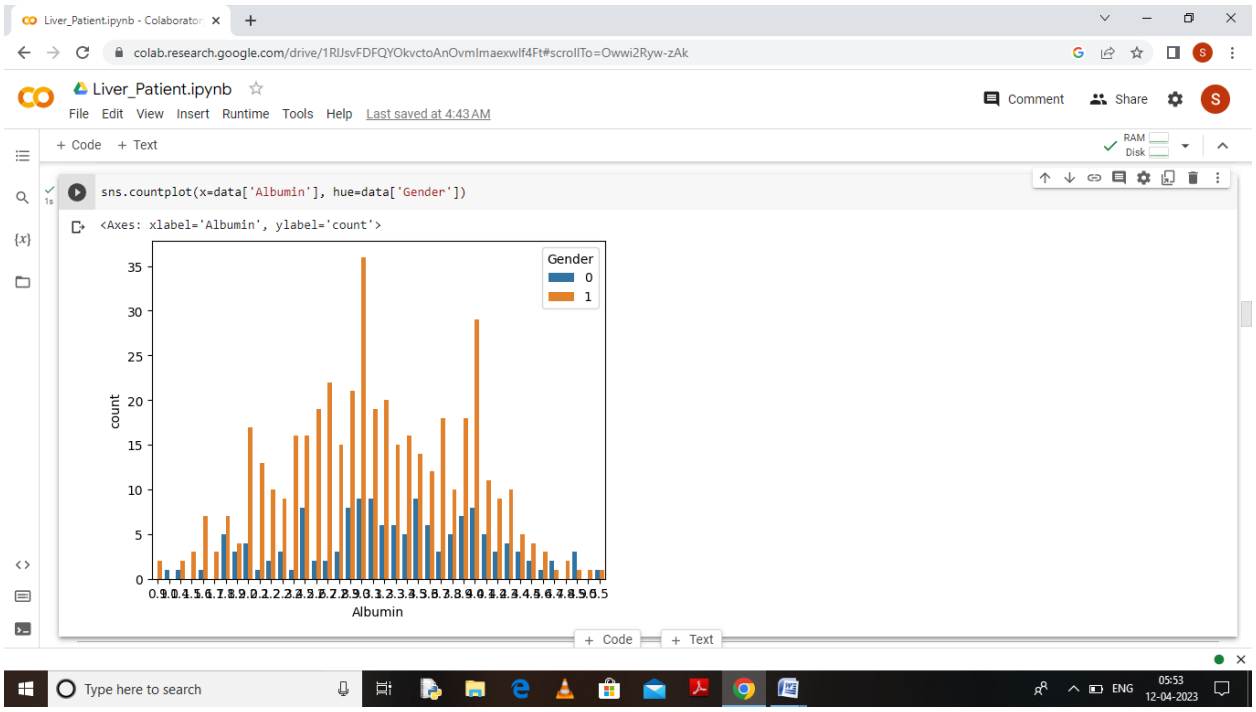


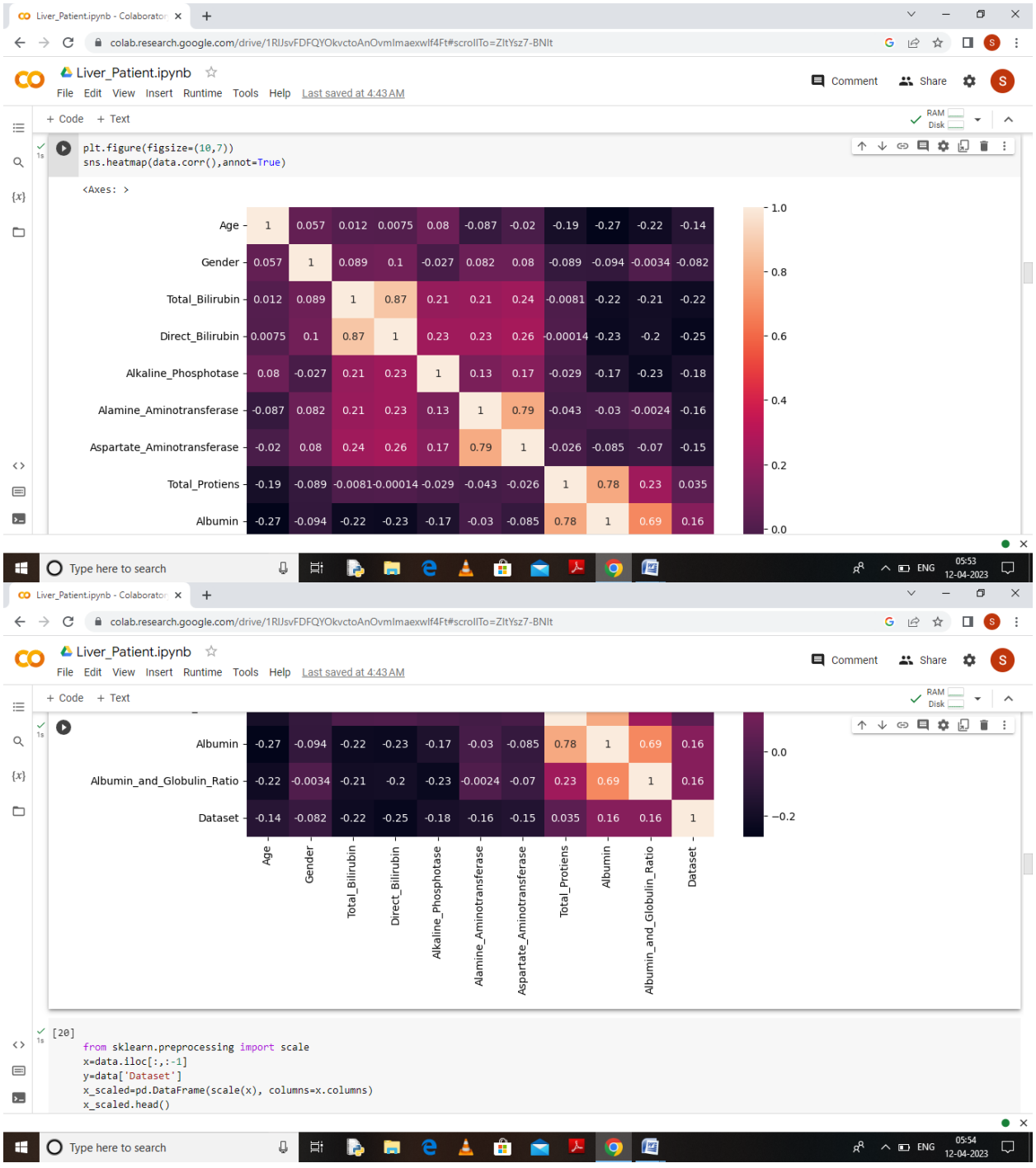
Type here to search

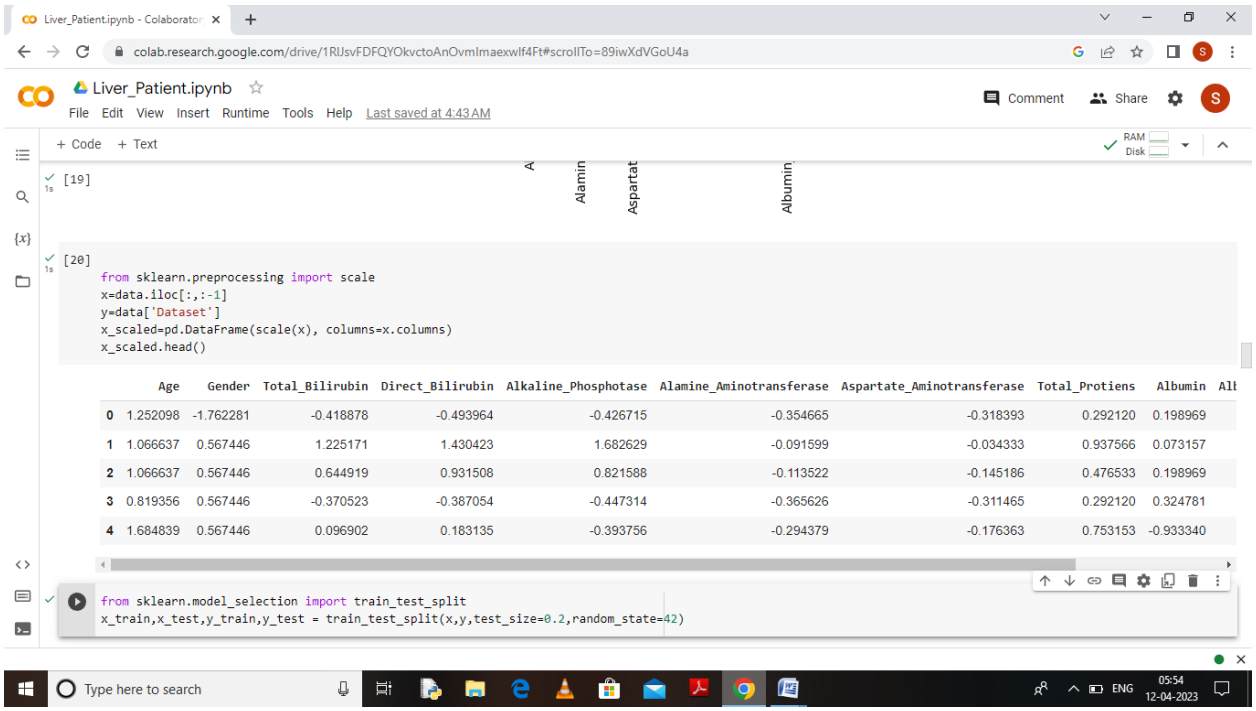
ENG 05:51 12-04-2023

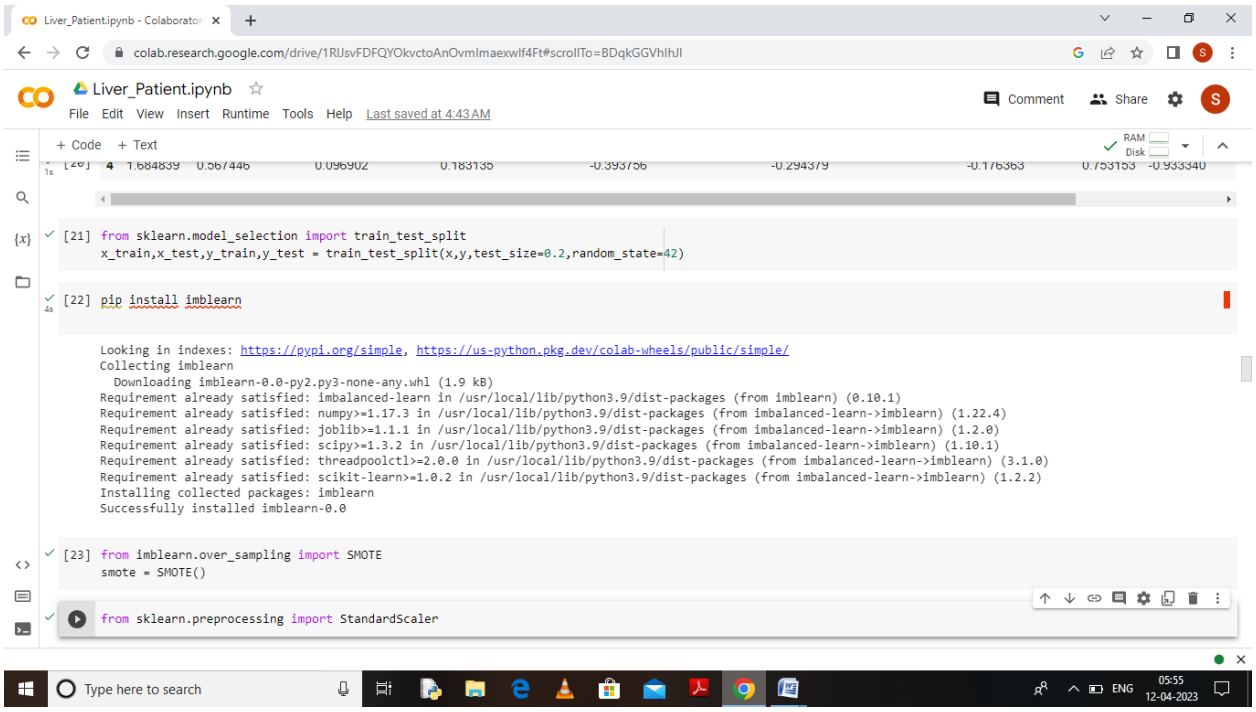


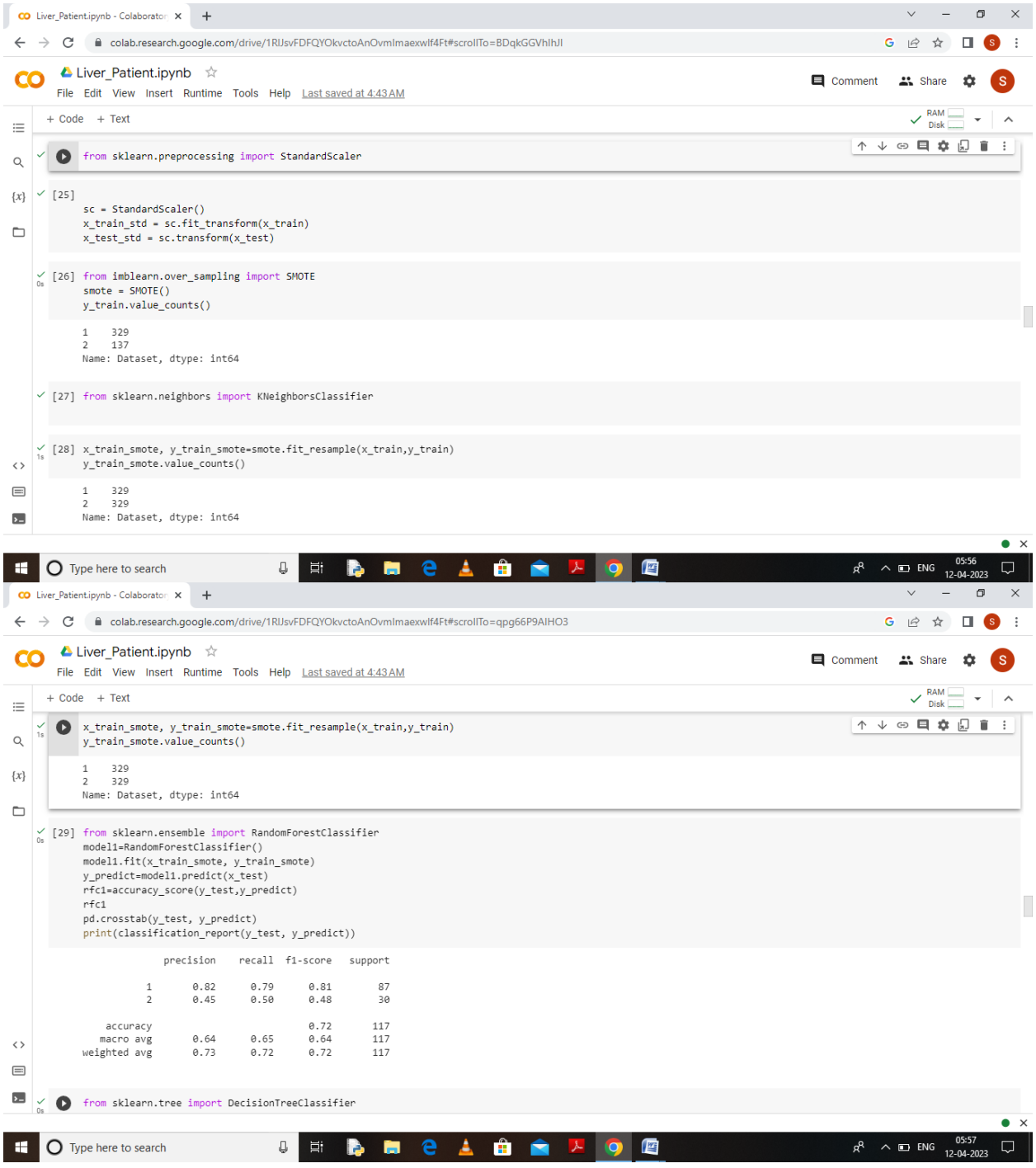
	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Alb
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.0
mean	44.746141	0.756432	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	3.14
std	16.189833	0.429603	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.79
min	4.000000	0.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.90
25%	33.000000	1.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.60
50%	45.000000	1.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.10
75%	58.000000	1.000000	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.80
max	90.000000	1.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.50

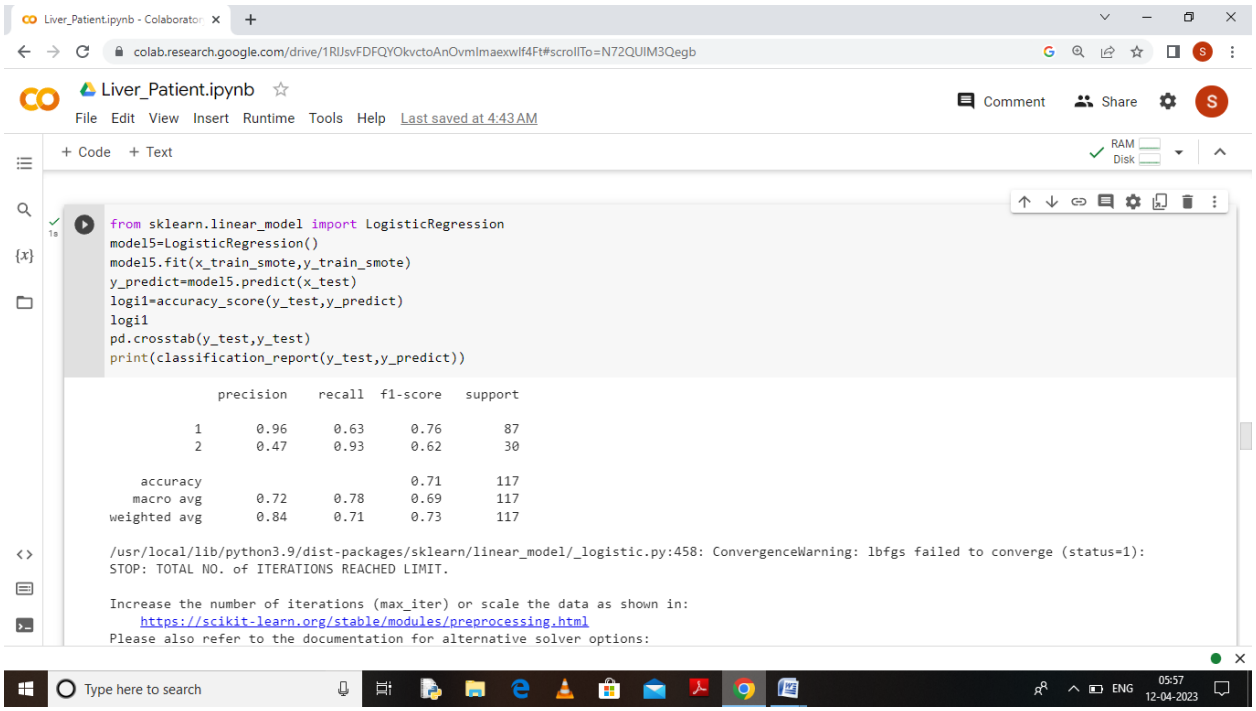


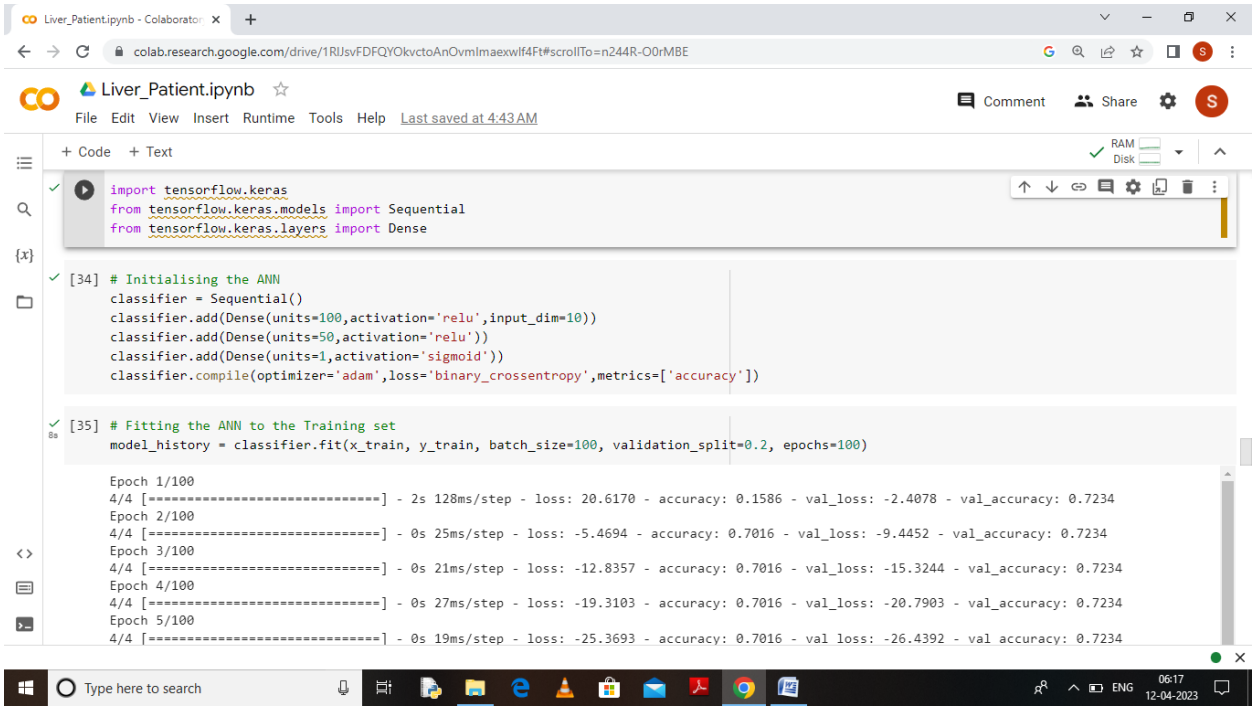

Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=ZItYsz7-BNIt
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
plt.figure(figsize=(10,7))
sns.heatmap(data.corr(),annot=True)
<Axes: >
Age 1 0.057 0.012 0.0075 0.08 -0.087 -0.02 -0.19 -0.27 -0.22 -0.14
Gender 0.057 1 0.089 0.1 -0.027 0.082 0.08 -0.089 -0.094 -0.0034 -0.082
Total_Bilirubin 0.012 0.089 1 0.87 0.21 0.21 0.24 -0.0081 -0.22 -0.21 -0.22
Direct_Bilirubin 0.0075 0.1 0.87 1 0.23 0.23 0.26 -0.00014 -0.23 -0.2 -0.25
Alkaline_Phosphotase 0.08 -0.027 0.21 0.23 1 0.13 0.17 -0.029 -0.17 -0.23 -0.18
Alamine_Aminotransferase -0.087 0.082 0.21 0.23 0.13 1 0.79 -0.043 -0.03 -0.0024 -0.16
Aspartate_Aminotransferase -0.02 0.08 0.24 0.26 0.17 0.79 1 -0.026 -0.085 -0.07 -0.15
Total_Protiens -0.19 -0.089 -0.0081 -0.00014 -0.029 -0.043 -0.026 1 0.78 0.23 0.035
Albumin -0.27 -0.094 -0.22 -0.23 -0.17 -0.03 -0.085 0.78 1 0.69 0.16
1.0
0.8
0.6
0.4
0.2
0.0
Type here to search
ENG
05:53
12-04-2023
Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=ZItYsz7-BNIt
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
Albumin -0.27 -0.094 -0.22 -0.23 -0.17 -0.03 -0.085 0.78 1 0.69 0.16
Albumin_and_Globulin_Ratio -0.22 -0.0034 -0.21 -0.2 -0.23 -0.0024 -0.07 0.23 0.69 1 0.16
Dataset -0.14 -0.082 -0.22 -0.25 -0.18 -0.16 -0.15 0.035 0.16 0.16 1
0.0
−0.2
Age
Gender
Total_Bilirubin
Direct_Bilirubin
Alkaline_Phosphotase
Alamine_Aminotransferase
Aspartate_Aminotransferase
Total_Protiens
Albumin
Albumin_and_Globulin_Ratio
Dataset
[20]
from sklearn.preprocessing import scale
x=data.iloc[:,:-1]
y=data['Dataset']
x_scaled=pd.DataFrame(scale(x), columns=x.columns)
x_scaled.head()
Type here to search
ENG
05:54
12-04-2023

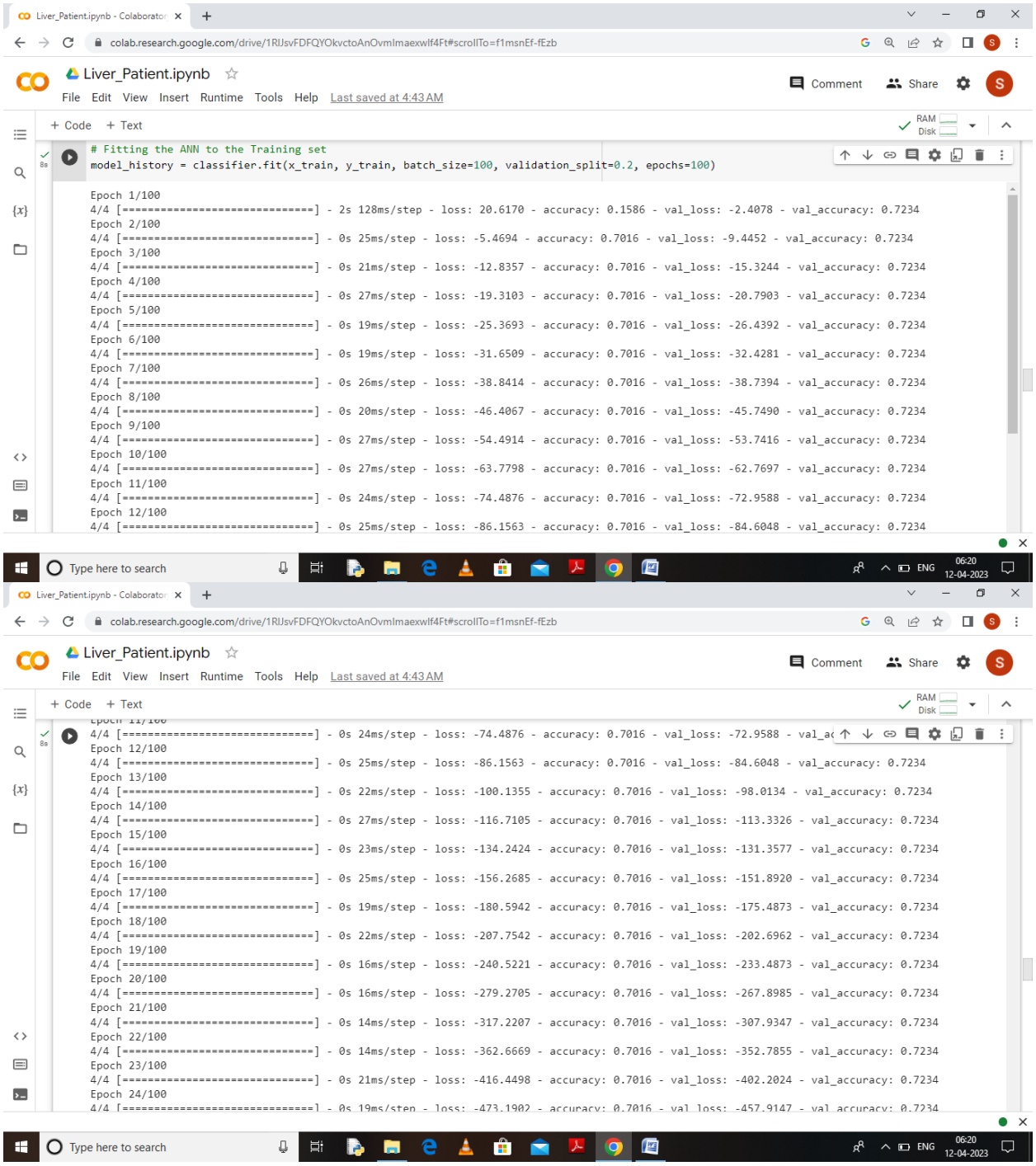

Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=89iwXdVGoU4a
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
[19]
Alamin
Aspartat
Albumin
[20]
from sklearn.preprocessing import scale
x=data.iloc[:,:-1]
y=data['Dataset']
x_scaled=pd.DataFrame(scale(x), columns=x.columns)
x_scaled.head()
Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin Alb
0 1.252098 -1.762281 -0.418878 -0.493964 -0.426715 -0.354665 -0.318393 0.292120 0.198969
1 1.066637 0.567446 1.225171 1.430423 1.682629 -0.091599 -0.034333 0.937566 0.073157
2 1.066637 0.567446 0.644919 0.931508 0.821588 -0.113522 -0.145186 0.476533 0.198969
3 0.819356 0.567446 -0.370523 -0.387054 -0.447314 -0.365626 -0.311465 0.292120 0.324781
4 1.684839 0.567446 0.096902 0.183135 -0.393756 -0.294379 -0.176363 0.753153 -0.933340
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=42)
Type here to search
ENG
05:54
12-04-2023

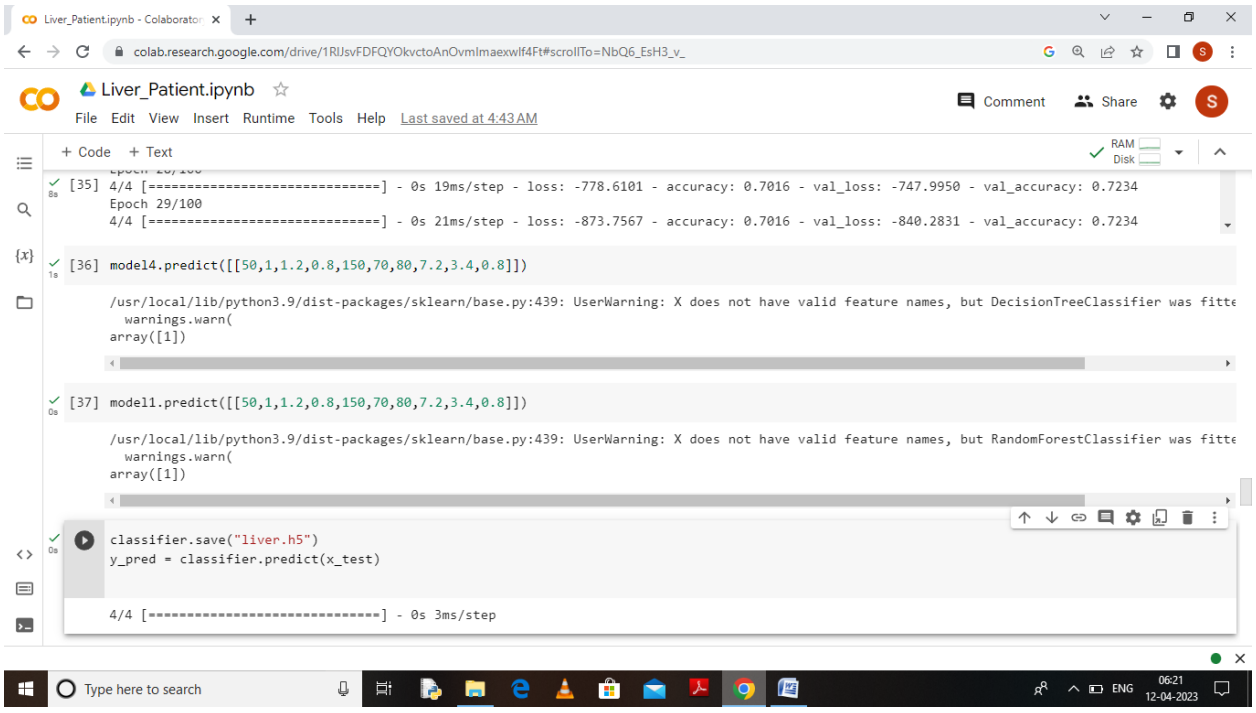

Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=BDqkGGVhIhJI
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
4 1.684839 0.567446 0.096902 0.183135 -0.393756 -0.294379 -0.176363 0.753153 -0.933340
[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=42)
[22] pip install imblearn
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/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: numpy>=1.17.3 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.22.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.2.0)
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: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (3.1.0)
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.2.2)
Installing collected packages: imblearn
Successfully installed imblearn-0.0
[23] from imblearn.over_sampling import SMOTE
smote = SMOTE()
from sklearn.preprocessing import StandardScaler
Type here to search
ENG
05:55
12-04-2023

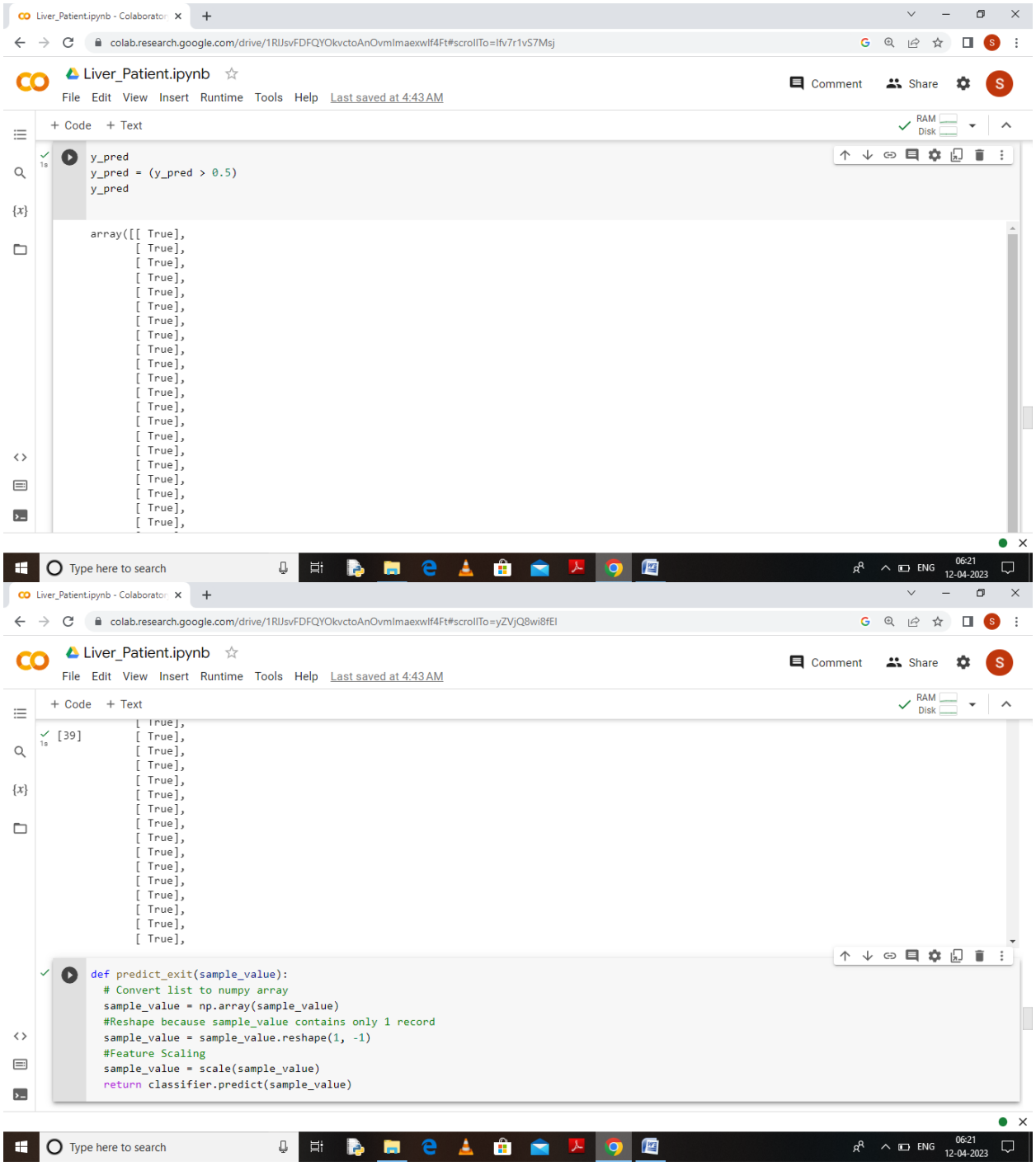

Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=BDqkGGVhIhJI
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
from sklearn.preprocessing import StandardScaler
[25]
sc = StandardScaler()
x_train_std = sc.fit_transform(x_train)
x_test_std = sc.transform(x_test)
[26] from imblearn.over_sampling import SMOTE
smote = SMOTE()
y_train.value_counts()
1 329
2 137
Name: Dataset, dtype: int64
[27] from sklearn.neighbors import KNeighborsClassifier
[28] x_train_smote, y_train_smote=smote.fit_resample(x_train,y_train)
y_train_smote.value_counts()
1 329
2 329
Name: Dataset, dtype: int64
Type here to search
ENG
05:56
12-04-2023
Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=qpg66P9AIHO3
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
x_train_smote, y_train_smote=smote.fit_resample(x_train,y_train)
y_train_smote.value_counts()
1 329
2 329
Name: Dataset, dtype: int64
[29] from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier()
model1.fit(x_train_smote, y_train_smote)
y_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))
precision recall f1-score support
1 0.82 0.79 0.81 87
2 0.45 0.50 0.48 30
accuracy 0.72 117
macro avg 0.64 0.65 0.64 117
weighted avg 0.73 0.72 0.72 117
from sklearn.tree import DecisionTreeClassifier
Type here to search
ENG
05:57
12-04-2023

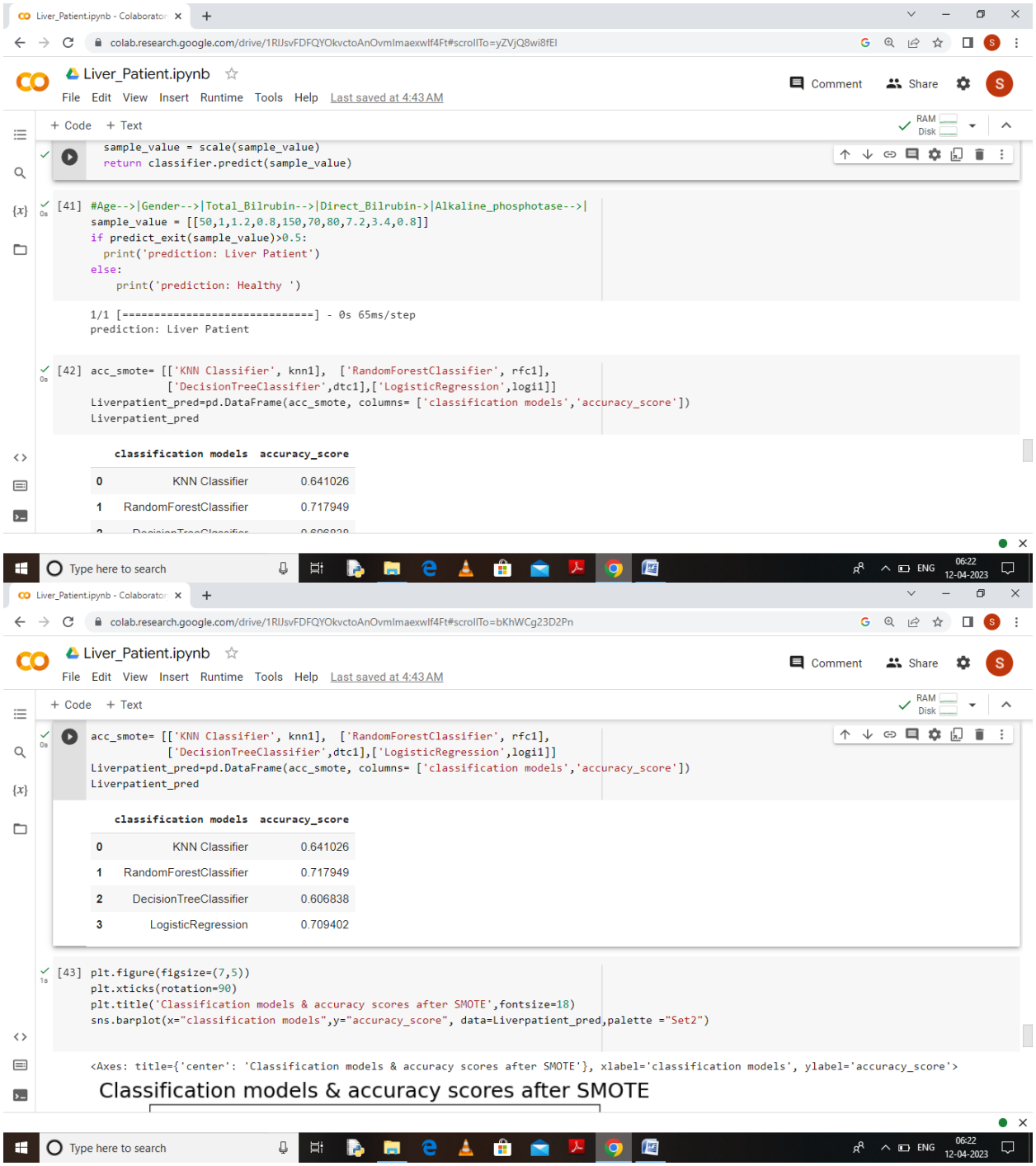

Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=N72QUIM3Qegb
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
from sklearn.linear_model import LogisticRegression
model5=LogisticRegression()
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_test)
print(classification_report(y_test,y_predict))
precision recall f1-score support
1 0.96 0.63 0.76 87
2 0.47 0.93 0.62 30
accuracy 0.71 117
macro avg 0.72 0.78 0.69 117
weighted avg 0.84 0.71 0.73 117
/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
Type here to search
ENG
05:57
12-04-2023

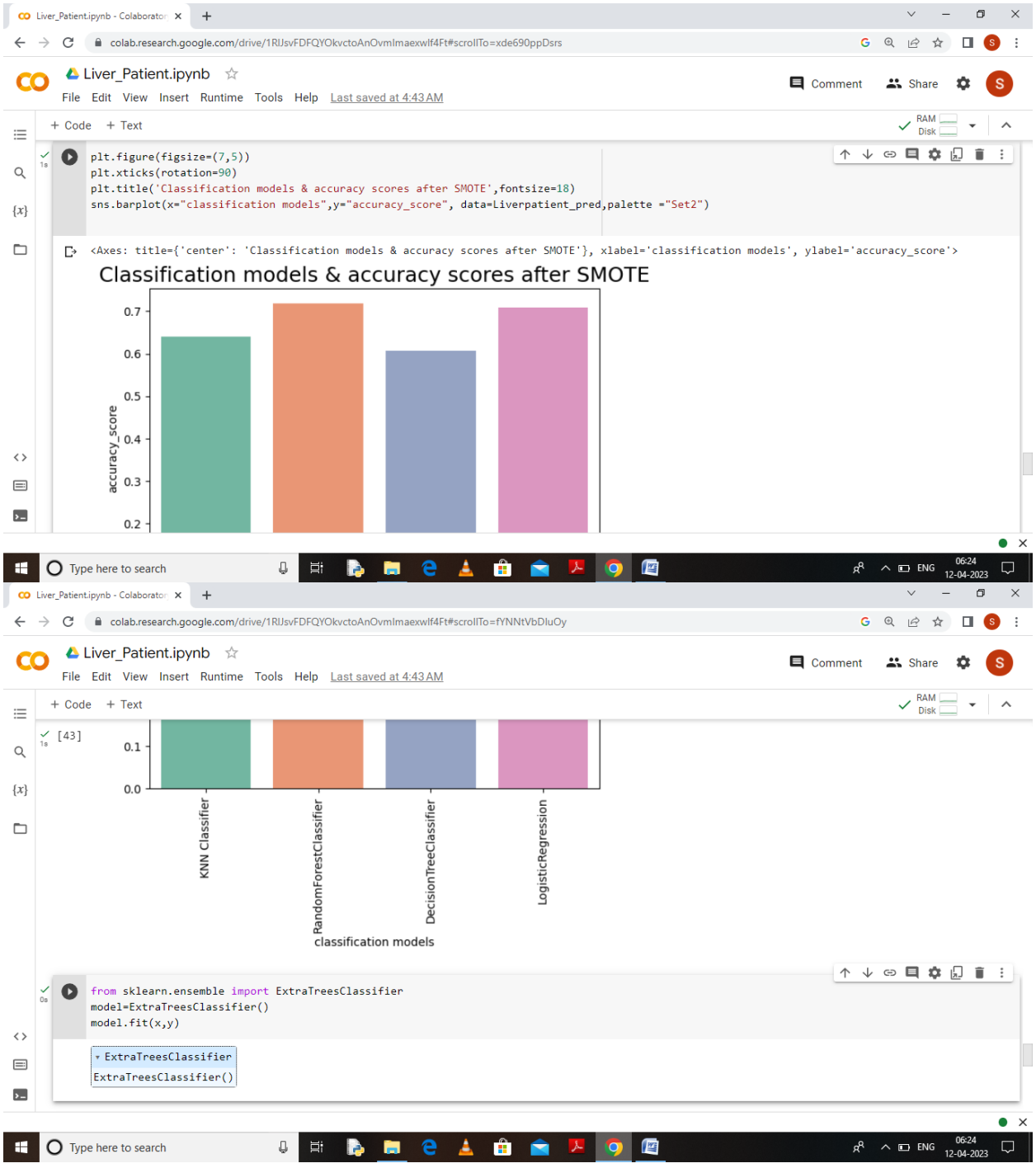

Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=n244R-O0rMBE
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
import tensorflow.keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
[34] # Initialising the ANN
classifier = Sequential()
classifier.add(Dense(units=100,activation='relu',input_dim=10))
classifier.add(Dense(units=50,activation='relu'))
classifier.add(Dense(units=1,activation='sigmoid'))
classifier.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
[35] # Fitting the ANN to the Training set
model_history = classifier.fit(x_train, y_train, batch_size=100, validation_split=0.2, epochs=100)
Epoch 1/100
4/4 [==========] - 2s 128ms/step - loss: 20.6170 - accuracy: 0.1586 - val_loss: -2.4078 - val_accuracy: 0.7234
Epoch 2/100
4/4 [==========] - 0s 25ms/step - loss: -5.4694 - accuracy: 0.7016 - val_loss: -9.4452 - val_accuracy: 0.7234
Epoch 3/100
4/4 [==========] - 0s 21ms/step - loss: -12.8357 - accuracy: 0.7016 - val_loss: -15.3244 - val_accuracy: 0.7234
Epoch 4/100
4/4 [==========] - 0s 27ms/step - loss: -19.3103 - accuracy: 0.7016 - val_loss: -20.7903 - val_accuracy: 0.7234
Epoch 5/100
4/4 [==========] - 0s 19ms/step - loss: -25.3693 - accuracy: 0.7016 - val_loss: -26.4392 - val_accuracy: 0.7234
Type here to search
ENG
06:17
12-04-2023

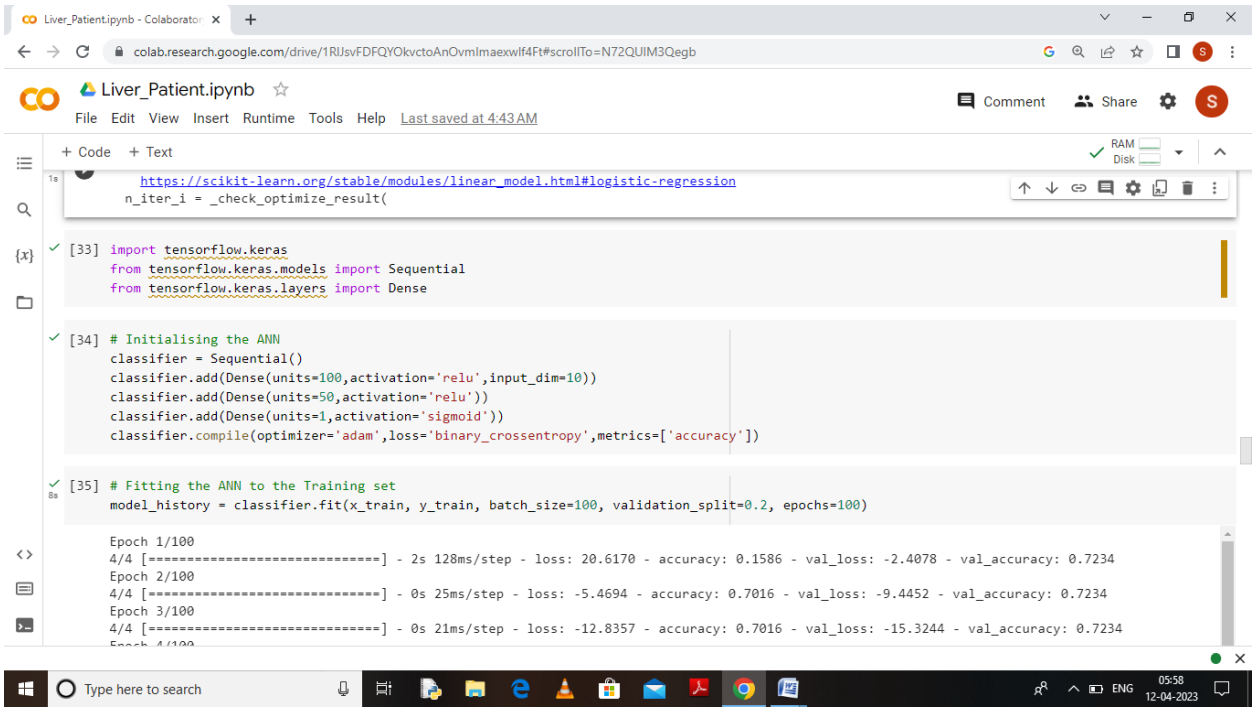

Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=f1msnEf-fEzb
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
# Fitting the ANN to the Training set
model_history = classifier.fit(x_train, y_train, batch_size=100, validation_split=0.2, epochs=100)
Epoch 1/100
4/4 [==============================] - 2s 128ms/step - loss: 20.6170 - accuracy: 0.1586 - val_loss: -2.4078 - val_accuracy: 0.7234
Epoch 2/100
4/4 [==============================] - 0s 25ms/step - loss: -5.4694 - accuracy: 0.7016 - val_loss: -9.4452 - val_accuracy: 0.7234
Epoch 3/100
4/4 [==============================] - 0s 21ms/step - loss: -12.8357 - accuracy: 0.7016 - val_loss: -15.3244 - val_accuracy: 0.7234
Epoch 4/100
4/4 [==============================] - 0s 27ms/step - loss: -19.3103 - accuracy: 0.7016 - val_loss: -20.7903 - val_accuracy: 0.7234
Epoch 5/100
4/4 [==============================] - 0s 19ms/step - loss: -25.3693 - accuracy: 0.7016 - val_loss: -26.4392 - val_accuracy: 0.7234
Epoch 6/100
4/4 [==============================] - 0s 19ms/step - loss: -31.6509 - accuracy: 0.7016 - val_loss: -32.4281 - val_accuracy: 0.7234
Epoch 7/100
4/4 [==============================] - 0s 26ms/step - loss: -38.8414 - accuracy: 0.7016 - val_loss: -38.7394 - val_accuracy: 0.7234
Epoch 8/100
4/4 [==============================] - 0s 20ms/step - loss: -46.4067 - accuracy: 0.7016 - val_loss: -45.7490 - val_accuracy: 0.7234
Epoch 9/100
4/4 [==============================] - 0s 27ms/step - loss: -54.4914 - accuracy: 0.7016 - val_loss: -53.7416 - val_accuracy: 0.7234
Epoch 10/100
4/4 [==============================] - 0s 27ms/step - loss: -63.7798 - accuracy: 0.7016 - val_loss: -62.7697 - val_accuracy: 0.7234
Epoch 11/100
4/4 [==============================] - 0s 24ms/step - loss: -74.4876 - accuracy: 0.7016 - val_loss: -72.9588 - val_accuracy: 0.7234
Epoch 12/100
4/4 [==============================] - 0s 25ms/step - loss: -86.1563 - accuracy: 0.7016 - val_loss: -84.6048 - val_accuracy: 0.7234
Type here to search
ENG
06:20
12-04-2023
Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=f1msnEf-fEzb
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
4/4 [==============================] - 0s 24ms/step - loss: -74.4876 - accuracy: 0.7016 - val_loss: -72.9588 - val_a
Epoch 12/100
4/4 [==============================] - 0s 25ms/step - loss: -86.1563 - accuracy: 0.7016 - val_loss: -84.6048 - val_accuracy: 0.7234
Epoch 13/100
4/4 [==============================] - 0s 22ms/step - loss: -100.1355 - accuracy: 0.7016 - val_loss: -98.0134 - val_accuracy: 0.7234
Epoch 14/100
4/4 [==============================] - 0s 27ms/step - loss: -116.7105 - accuracy: 0.7016 - val_loss: -113.3326 - val_accuracy: 0.7234
Epoch 15/100
4/4 [==============================] - 0s 23ms/step - loss: -134.2424 - accuracy: 0.7016 - val_loss: -131.3577 - val_accuracy: 0.7234
Epoch 16/100
4/4 [==============================] - 0s 25ms/step - loss: -156.2685 - accuracy: 0.7016 - val_loss: -151.8920 - val_accuracy: 0.7234
Epoch 17/100
4/4 [==============================] - 0s 19ms/step - loss: -180.5942 - accuracy: 0.7016 - val_loss: -175.4873 - val_accuracy: 0.7234
Epoch 18/100
4/4 [==============================] - 0s 22ms/step - loss: -207.7542 - accuracy: 0.7016 - val_loss: -202.6962 - val_accuracy: 0.7234
Epoch 19/100
4/4 [==============================] - 0s 16ms/step - loss: -240.5221 - accuracy: 0.7016 - val_loss: -233.4873 - val_accuracy: 0.7234
Epoch 20/100
4/4 [==============================] - 0s 16ms/step - loss: -279.2705 - accuracy: 0.7016 - val_loss: -267.8985 - val_accuracy: 0.7234
Epoch 21/100
4/4 [==============================] - 0s 14ms/step - loss: -317.2207 - accuracy: 0.7016 - val_loss: -307.9347 - val_accuracy: 0.7234
Epoch 22/100
4/4 [==============================] - 0s 14ms/step - loss: -362.6669 - accuracy: 0.7016 - val_loss: -352.7855 - val_accuracy: 0.7234
Epoch 23/100
4/4 [==============================] - 0s 21ms/step - loss: -416.4498 - accuracy: 0.7016 - val_loss: -402.2024 - val_accuracy: 0.7234
Epoch 24/100
4/4 [==============================] - 0s 19ms/step - loss: -473.1902 - accuracy: 0.7016 - val_loss: -457.9147 - val_accuracy: 0.7234
Type here to search
ENG
06:20
12-04-2023


Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=NbQ6_EsH3_v_
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
[35] 4/4 [==============================] - 0s 19ms/step - loss: -778.6101 - accuracy: 0.7016 - val_loss: -747.9950 - val_accuracy: 0.7234
Epoch 29/100
4/4 [==============================] - 0s 21ms/step - loss: -873.7567 - accuracy: 0.7016 - val_loss: -840.2831 - val_accuracy: 0.7234
[36] model4.predict([[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]])
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitte
warnings.warn(
array([1])
[37] model1.predict([[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]])
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitte
warnings.warn(
array([1])
classifier.save("liver.h5")
y_pred = classifier.predict(x_test)
4/4 [==============================] - 0s 3ms/step
Type here to search
ENG
06:21
12-04-2023


Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=Ifv7r1vS7Msj
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
y_pred
y_pred = (y_pred > 0.5)
y_pred
array([[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
Type here to search
ENG
06:21
12-04-2023
Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=yZVjQ8wi8fEl
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
[39]
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
[True],
def predict_exit(sample_value):
# Convert list to numpy array
sample_value = np.array(sample_value)
#Reshape because sample_value contains only 1 record
sample_value = sample_value.reshape(1, -1)
#Feature Scaling
sample_value = scale(sample_value)
return classifier.predict(sample_value)
Type here to search
ENG
06:21
12-04-2023


Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=yZVjQ8wi8fEI
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
sample_value = scale(sample_value)
return classifier.predict(sample_value)
[41] #Age-->|Gender-->|Total_Bilrubin-->|Direct_Bilrubin->|Alkaline_phosphotase-->|
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 ')
1/1 [==============================] - 0s 65ms/step
prediction: Liver Patient
[42] acc_smote= [['KNN Classifier', knn1], ['RandomForestClassifier', rfc1],
['DecisionTreeClassifier',dtc1],['LogisticRegression',logi1]]
Liverpatient_pred=pd.DataFrame(acc_smote, columns= ['classification models','accuracy_score'])
Liverpatient_pred
classification models accuracy_score
0 KNN Classifier 0.641026
1 RandomForestClassifier 0.717949
Type here to search
ENG
06:22
12-04-2023
Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=bKhWCg23D2Pn
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
acc_smote= [['KNN Classifier', knn1], ['RandomForestClassifier', rfc1],
['DecisionTreeClassifier',dtc1],['LogisticRegression',logi1]]
Liverpatient_pred=pd.DataFrame(acc_smote, columns= ['classification models','accuracy_score'])
Liverpatient_pred
classification models accuracy_score
0 KNN Classifier 0.641026
1 RandomForestClassifier 0.717949
2 DecisionTreeClassifier 0.606838
3 LogisticRegression 0.709402
[43] 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,palette ="Set2")
<Axes: title={'center': 'Classification models & accuracy scores after SMOTE'}, xlabel='classification models', ylabel='accuracy_score'>
Classification models & accuracy scores after SMOTE
Type here to search
ENG
06:22
12-04-2023


Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=xde690ppDsrs
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
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,palette ="Set2")
<Axes: title={'center': 'Classification models & accuracy scores after SMOTE'}, xlabel='classification models', ylabel='accuracy_score'>
Classification models & accuracy scores after SMOTE
accuracy_score
0.7
0.6
0.5
0.4
0.3
0.2
Type here to search
ENG
06:24
12-04-2023
Liver_Patient.ipynb - Colaborator
colab.research.google.com/drive/1RlJsvFDFQYOkvctoAnOvmImaexwIf4Ft#scrollTo=fYNNtVbDluOy
Liver_Patient.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:43 AM
Comment
Share
+ Code + Text
RAM
Disk
[43]
0.1
0.0
KNN Classifier
RandomForestClassifier
DecisionTreeClassifier
LogisticRegression
classification models
from sklearn.ensemble import ExtraTreesClassifier
model=ExtraTreesClassifier()
model.fit(x,y)
ExtraTreesClassifier
ExtraTreesClassifier()
Type here to search
ENG
06:24
12-04-2023



4 ADVANTAGES & DISADVANTAGES

List of advantages and disadvantage

Advantage

A liver biopsy may be done to: **Diagnose a liver problem that can't be otherwise identified with a health care provider's exam, blood tests or imaging studies**. Obtain a sample of tissue from an irregularity found by an imaging study. Determine the severity of liver disease, a process called staging.

Disadvantage

It's possible, and dangerous, to get too much vitamin A. Eating large amounts of liver can lead to symptoms of vitamin A toxicity, which happens when your own liver can't process the excess vitamin A quickly enough. Most doctors recommend that people without vitamin deficiencies eat just one serving of liver per week.

5 APPLICATIONS

Testing the model Define Problem / Problem Understanding

Specify the business problem

Business requirements

Literature Survey

Social or Business Impact.

Data Collection & Preparation

Collect the dataset

Data Preparation

Exploratory Data Analysis

Descriptive statistical

Visual Analysis

Model Building

Training the model in multiple algorithms

6 CONCLUSION

Explore and understand your data

Visualize the data at hand to gain a better intuition — Word cloud, N-gram Bar Chart

Text Cleaning — Word Stemmer and Word Lemmatization

Feature Extraction — Count Vectorizer, Tfidf Vectorizer, Word Embedding

Algorithm — Naive Bayes

Scoring & Metrics — Accuracy, Precision, Recall

7 FUTURE SCOPE

Generally, a **90% spam catch rate** (90 out of 100 spam messages are correctly identified as spam) and a false positive rate of less than 1% (less than 1 legitimate message out of a hundred incorrectly identified as spam) is considered good.

Help prevent similar attacks and respond to changing behaviour.

Time Saving

Cyber Security analyse Pattern

Fraud Detection.

8 APPENDIX

Source Code

```
import pandas as pd
import numpy as np
import plotly.express as px
import matplotlib.pyplot as plt
from  google.colab import files
import io
import seaborn as sns
from sklearn.ensemble import VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, mean_absolute_error,r2_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.ensemble import
RandomForestClassifier,ExtraTreesClassifier,AdaBoostClassifier,BaggingClassifier,GradientBoostingC
lassifier
import xgboost as xgb
# %matplotlib inline

data=files.upload()

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

data.head()

data.info()

data.isnull().any()

data.isnull().sum()

data.Albumin_and_Globulin_Ratio.fillna(data['Albumin_and_Globulin_Ratio'].mean(),inplace=True)
data.isnull().sum()

data.head()
```

```
data.info()

data.describe()

from sklearn.preprocessing import LabelEncoder
lc = LabelEncoder()
data['Gender']= lc.fit_transform(data['Gender'])

data.describe()

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

sns.countplot(x=data['Albumin'], hue=data['Gender'])

plt.figure(figsize=(10,7))
sns.heatmap(data.corr(),annot=True)

from sklearn.preprocessing import scale
x=data.iloc[:,:-1]
y=data['Dataset']
x_scaled=pd.DataFrame(scale(x), columns=x.columns)
x_scaled.head()

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=42)

pip install imblearn

from imblearn.over_sampling import SMOTE
smote = SMOTE()

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
x_train_std = sc.fit_transform(x_train)
x_test_std = sc.transform(x_test)

from imblearn.over_sampling import SMOTE
smote = SMOTE()
y_train.value_counts()

from sklearn.neighbors import KNeighborsClassifier

x_train_smote, y_train_smote=smote.fit_resample(x_train,y_train)
```

```
y_train_smote.value_counts()

from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier()
model1.fit(x_train_smote, y_train_smote)
y_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.tree import DecisionTreeClassifier
model4=DecisionTreeClassifier()
model4.fit(x_train_smote,y_train_smote)
y_predict=model4.predict(x_test)
dtc1=accuracy_score(y_test,y_predict)
dtc1
pd.crosstab(y_test,y_predict)
print(classification_report(y_test,y_predict))

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.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_test)
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()
classifier.add(Dense(units=100,activation='relu',input_dim=10))
classifier.add(Dense(units=50,activation='relu'))
```

```
classifier.add(Dense(units=1,activation='sigmoid'))
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
y_pred = (y_pred > 0.5)
y_pred

def predict_exit(sample_value):
  # Convert list to numpy array
  sample_value = np.array(sample_value)
  #Reshape because sample_value contains only 1 record
  sample_value = sample_value.reshape(1, -1)
  #Feature Scaling
  sample_value = scale(sample_value)
  return classifier.predict(sample_value)

#Age-->|Gender-->|Total_Bilrubin-->|Direct_Bilrubin->|Alkaline_phosphotase-->|
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',dtc1],['LogisticRegression',logi1]]
Liverpatient_pred=pd.DataFrame(acc_smote, columns= ['classification models','accuracy_score'])
Liverpatient_pred

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,palette ="Set2")

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.pkl')
```