

# **Project Report-Major Projects**

**Given by Rinex**

**Name – Ritam Sarkar**  
**College- Kalyani Government Engineering**  
**college**  
**Year - 3<sup>rd</sup>**

# Major Project 1:

## Logistic regression model for prediction of heart disease

```
major project1.ipynb - Colab
https://colab.research.google.com/drive/1LBbNOQ7H25LIFz_ZT1hBQI-Q4sNNQv5s#scrollTo=9lWmQxppgkdg

Getting Started | Inbox | ritam.formal... | happy new year - Go...
Firefox prevented this site from opening 2 pop-up windows. Preferences

major project1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text RAM Disk Editing

[37] #logistic regression to predict heart deseas

[38] import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

[39] #loading the csv file
df = pd.read_csv('/content/heart.csv')

[40] df.head()

  age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
0   63   1   3    145   233    1      0    150     0     2.3    0  0   1     1
1   37   1   2    130   250    0      1    187     0     3.5    0  0   2     1
2   41   0   1    130   204    0      0    172     0     1.4    2  0  2     1
3   56   1   1    120   236    0      1    178     0     0.8    2  0  2     1
4   57   0   0    120   354    0      1    163     1     0.6    2  0  2     1

[41] df.tail()

  age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
298  57   0   0    140   241    0      1    123     1     0.2    1  0  3     0
299  45   1   3    110   264    0      1    132     0     1.2    1  0  3     0
300  68   1   0    144   193    1      1    141     0     3.4    1  2  3     0
301  57   1   0    130   131    0      1    115     1     1.2    1  1  3     0
302  57   0   1    130   236    0      0    174     0     0.0    1  1  2     0

0s completed at 10:11 AM
```

```
major project1.ipynb - Colab
https://colab.research.google.com/drive/1LBbNOQ7H25LIFz_ZT1hBQI-Q4sNNQv5s#scrollTo=9lWmQxppgkdg

Getting Started | Inbox | ritam.formal... | happy new year - Go...
Firefox prevented this site from opening 2 pop-up windows. Preferences

major project1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text RAM Disk Editing

[42] df.shape #shape of the data set

(303, 14)

[43] df.info() #some information about the data

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype  
---  --
 0   age           303 non-null   int64  
 1   sex           303 non-null   int64  
 2   cp            303 non-null   int64  
 3   trestbps      303 non-null   int64  
 4   chol          303 non-null   int64  
 5   fbs           303 non-null   int64  
 6   restecg       303 non-null   int64  
 7   thalach       303 non-null   int64  
 8   exang         303 non-null   int64  
 9   oldpeak       303 non-null   float64 
10  slope         303 non-null   int64  
11  ca            303 non-null   int64  
12  thal          303 non-null   int64  
13  target        303 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 Kb

[44] df.isnull().sum() #checking for missing values

age      0
sex      0
cp       0
trestbps 0
chol     0
fbs      0
restecg  0
thalach  0
exang    0
oldpeak  0
slope    0
ca       0
target   0

0s completed at 10:11 AM
```

major project1.ipynb - Colab x

https://colab.research.google.com/drive/1LBbNQ7H25LFlz\_ZT1hBQI-Q4sNNQv5s4scrollTo=9lWmQxpgkgd

Getting Started | Inbox | ritam.formal... | happy new year - Go...

Firefox prevented this site from opening 2 pop-up windows. Preferences

major project1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

df.describe() #statistical information about the data

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

[46] df['target'].value\_counts()

```
1    165
0    138
Name: target, dtype: int64
```

[47] #splitting the features and target

```
x = df.drop(columns='target', axis=1)
y = df['target']
```

[48] print(x)

```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0    63   1   3    145    233   1      0    150      0     2.3
1    37   1   2    130    250   0      1    187      0     3.5
2    41   0   1    130    204   0      0    172      0     1.4
3    56   1   1    120    236   0      1    170      0     0.8
4    57   0   0    120    354   0      1    163      1     0.6
..   ..   ..   ..   ..   ..   ..   ..   ..   ..   ..
298  57   0   0    140    241   0      1    123      1     0.2
```

0s completed at 10:11 AM

major project1.ipynb - Colab x

https://colab.research.google.com/drive/1LBbNQ7H25LFlz\_ZT1hBQI-Q4sNNQv5s4scrollTo=9lWmQxpgkgd

Getting Started | Inbox | ritam.formal... | happy new year - Go...

Firefox prevented this site from opening 2 pop-up windows. Preferences

major project1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

print(y) # 0 means healthy and 1 is otherwise

```
0    1
1    1
2    1
3    1
4    1
..
298  0
299  0
300  0
301  0
302  0
Name: target, Length: 303, dtype: int64
```

[50] #splitting the data for training and testing purpose

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, stratify=y, random_state=2)
```

[51] print(x.shape, x\_train.shape, x\_test.shape)

```
print(y.shape, y_train.shape, y_test.shape)
```

```
(303, 13) (242, 13) (61, 13)
(303,) (242,) (61,)
```

[52] model = LogisticRegression()

[53] model.fit(x\_train.values, y\_train.values) #training with test data

```
/usr/local/lib/python3.8/dist-packages/sklearn/linear_model/_logistic.py:814: 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:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_1 = check_optimize_result(
LogisticRegression()
```

#evaluation

0s completed at 10:11 AM

```
LogisticRegression()

[54] #evaluation
#accuracy for training data
x_train_prediction = model.predict(x_train.values)
training_accuracy = accuracy_score(x_train_prediction, y_train.values)
print("the accuracy for the training dataset is ", training_accuracy)

the accuracy for the training dataset is  0.851239694214877

[55] #accuracy for test data
x_test_prediction = model.predict(x_test.values)
test_accuracy = accuracy_score(x_test_prediction, y_test.values)
print("the accuracy for test dataset is ", test_accuracy)

the accuracy for test dataset is  0.819672131147541

[56] #prediction
input_f = (88,0,0,150,280,1,0,188,1,2,6,0,0,1) #these numbers represent the age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
#these are the features based on which the heart is healthy or not will be classified

[57] input_f = np.asarray(input_f) #converting the data in to numpy array
input_f = input_f.reshape(1,-1) #reshape the numpy array as we are predicting for only on instance
prediction = model.predict(input_f)
print(prediction)

if (prediction[0] == 0):
    print('The heart is most probably healthy')
else:
    print('The heart is not healthy, visit a doctor')
```

```
[57] if (prediction[0] == 0):
    print('The heart is most probably healthy')
else:
    print('The heart is not healthy, visit a doctor')

[1]
The heart is not healthy, visit a doctor

[2]
input_f2 = (28,1,0,140,220,0,1,108,1,2,6,0,0,1)
input_f2 = np.asarray(input_f2) #converting the data in to numpy array
input_f2 = input_f2.reshape(1,-1) #reshape the numpy array as we are predicting for only on instance
prediction = model.predict(input_f2)
print(prediction)

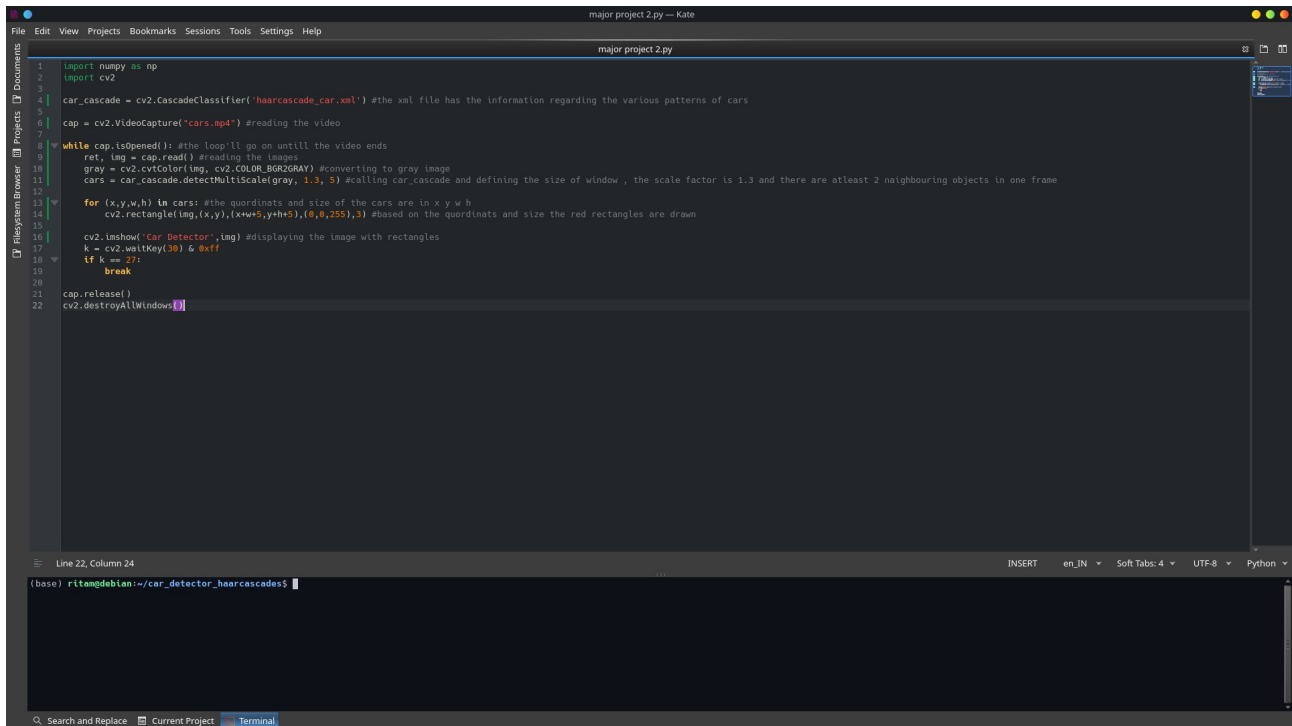
if (prediction[0] == 0):
    print('The heart is most probably healthy')
else:
    print('The heart is not healthy, visit a doctor')

[0]
The heart is most probably healthy
```

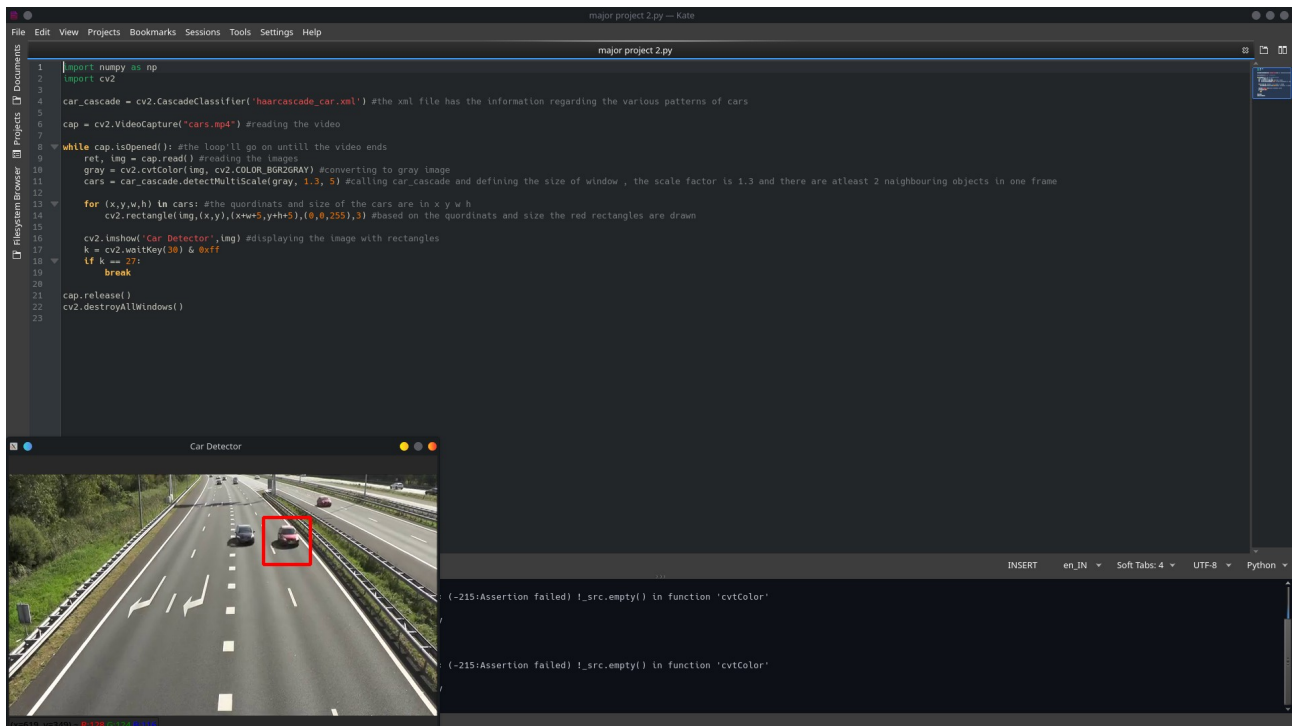
The Github link: [https://github.com/ritamsarkar2000/rinex-training-and-internship/tree/main/major\\_project\\_1](https://github.com/ritamsarkar2000/rinex-training-and-internship/tree/main/major_project_1)

# Major Project 2

## Image Processing- Car detection using Opencv



```
major project 2.py — Kate
major project 2.py
1 import numpy as np
2 import cv2
3 car_cascade = cv2.CascadeClassifier('haarcascade_car.xml') #the xml file has the information regarding the various patterns of cars
4
5 cap = cv2.VideoCapture("cars.mp4") #reading the video
6
7 while cap.isOpened(): #the loop'll go on until the video ends
8     ret, img = cap.read() #reading the images
9     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #converting to gray image
10    cars = car_cascade.detectMultiScale(gray, 1.3, 5) #calling car_cascade and defining the size of window , the scale factor is 1.3 and there are atleast 2 neighbouring objects in one frame
11
12    for (x,y,w,h) in cars: #the quordinats and size of the cars are in x y w h
13        cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),3) #based on the quordinats and size the red rectangles are drawn
14
15    cv2.imshow('Car Detector',img) #displaying the image with rectangles
16    k = cv2.waitKey(30) & 0xff
17    if k == 27:
18        break
19
20 cap.release()
21 cv2.destroyAllWindows()
```



The github link: <https://github.com/ritamsarkar2000/rinex-training-and-internship/tree/main/major%20project%202>

Github Link to the repository:

<https://github.com/ritamsarkar2000/rinex-training-and-internship.git>