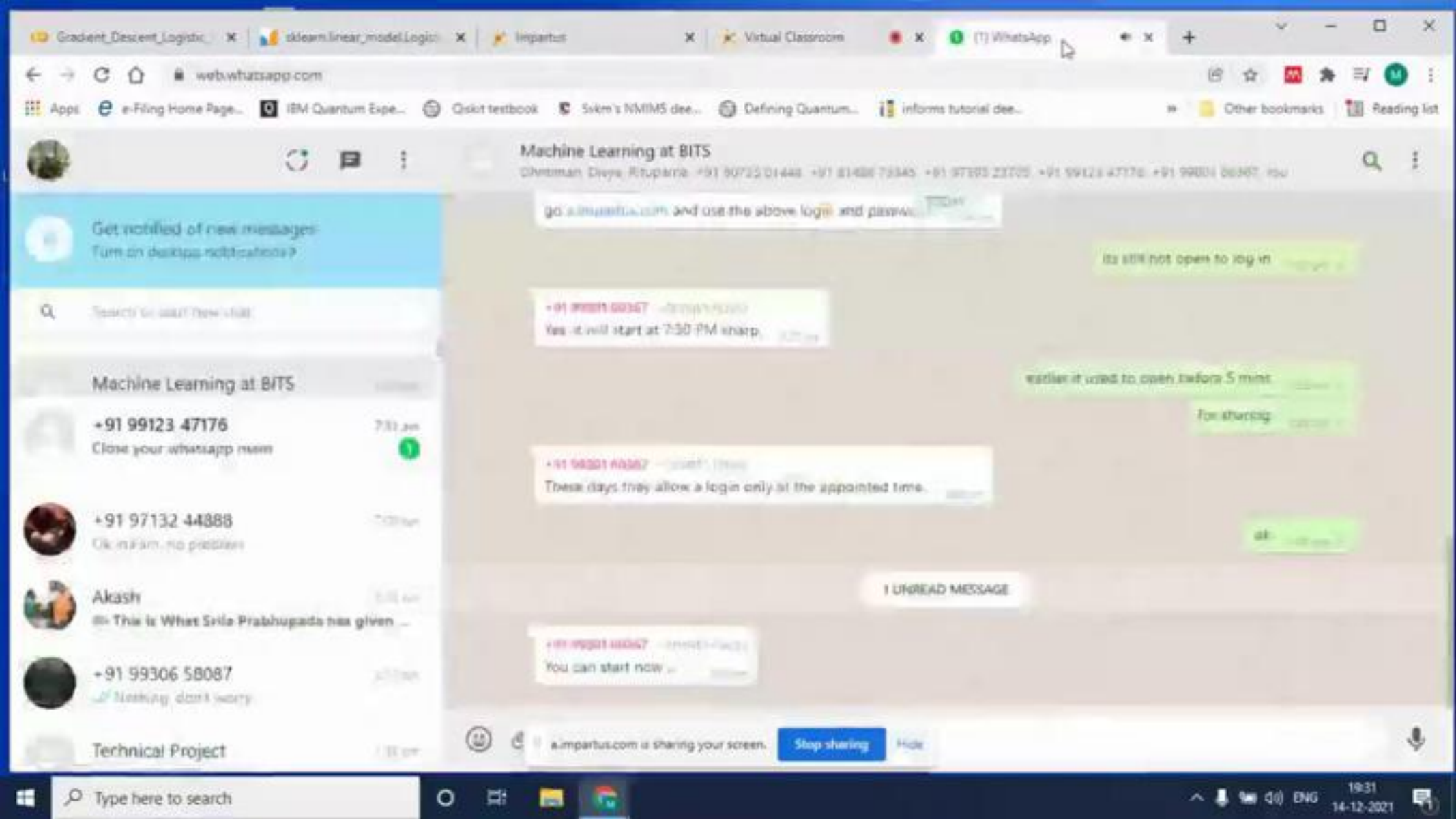

Machine Learning at BITS
Get notified of new messages
Turn on desktop notifications
Machine Learning at BITS
+91 99123 47176
Close your whatsapp mmm
+91 97132 44888
Akash
This is What Srila Prabhupada has given ...
+91 99306 58087
Technical Project
go .impartus.com and use the above login and passw
its still not open to log in
Yes it will start at 7:30 PM sharp.
earlier it used to open before 5 mins
For sharing
These days they allow a login only at the appointed time.
ok
1 UNREAD MESSAGE
You can start now ..
.impartus.com is sharing your screen.
Stop sharing
Hide
Type here to search
ENG
19:31
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Comment Share

Logistic Regression, Gradient Descent

Loss function,

Python implementation

Logistic Regression

Logistic regression is a probabilistic model, used to establish the relationship between a binary dependent variable and one or more independent variables. Let X_i be the independent variable (training variable) and Y be the dependent variable. Linear Regression predicts probability of outcome of binary dependent variable (Y) and the output lies between 0 and 1. The prediction equation between X_i and Y is

$$ln(\frac{p}{1-p}) = b_0 + b_1x_1 + b_2x_2 + b_3x_3 \ldots b_nx_n$$

$ln(\frac{p}{1-p})$ is *log odds* and denotes the likelihood of the event taking place. It maps the probability between 0 and 1. The terms $b_0, b_1, b_2 \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{1-} = e^{b_0+b_1x_1+b_2x_2+b_3x_3 \ldots b_nx_n}$$

a.impartus.com is sharing your screen. Stop sharing Hide

Type here to search

ENG 19:33 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

+ Code + Text

Logistic Regression, Gradient Descent

Loss function,

Python implementation

Logistic Regression

Logistic regression is a probabilistic model, used to establish the relationship between a binary dependent variable and one or more independent variables. Let X_i be the independent variable (training variable) and Y be the dependent variable. Linear Regression predicts probability of outcome of binary dependent variable (Y) and the output lies between 0 and 1. The prediction equation between X_i and Y is

$$ln(\frac{p}{1-p}) = b_0 + b_1x_1 + b_2x_2 + b_3x_3 \ldots b_nx_n$$

$ln(\frac{p}{1-p})$ is *log odds* and denotes the likelihood of the event taking place. It maps the probability between 0 and 1. The terms $b_0, b_1, b_2 \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{}$$

0s com

aumpartus.com is sharing your screen. Stop sharing Hide

Type here to search

ENG 19:33 14-12-2021

Gradient_Descent_Logistic_Regr | sklearn.linear_model.LogisticRe | Impartus | Virtual Classroom

colab.research.google.com/drive/1Dsp6j7qjVVVvMkDpG1wnWMiFXWcwam1y#scrollTo=mwHCjogkPjry

Apps | e-Filing Home Page... | IBM Quantum Expe... | Qiskit textbook | Svkm's NMIMS dee... | Defining Quantum... | informs tutorial dee... | Other bookmarks | Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment | Share | M

+ Code + Text | RAM | Disk | Editing

Logistic Regression, Gradient Descent

Loss function,

Python implementation

Logistic Regression

Logistic regression is a probabilistic model, used to establish the relationship between a binary dependent variable and one or more independent variables. Let X_i be the independent variable (training variable) and Y be the dependent variable. Linear Regression predicts probability of outcome of binary dependent variable (Y) and the output lies between 0 and 1. The prediction equation between X_i and Y is

$$ln(\frac{p}{1-p}) = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n$$

$ln(\frac{p}{1-p})$ is *log odds* and denotes the likelihood of the event taking place. It maps the probabilty between 0 and 1. The terms $b_0, b_1, b_2 \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{1-} = e^{b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n}$$

a.impartus.com is sharing your screen. Stop sharing | Hide

Type here to search | ENG | 19:33 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

Editing

Logistic Regression, Gradient Descent

Loss function,

Python implementation

Logistic Regression

Logistic regression is a probabilistic model, used to establish the relationship between a binary dependent variable and one or more independent variables. Let X_i be the independent variable (training variable) and Y be the dependent variable. Linear Regression predicts probability of outcome of binary dependent variable (Y) and the output lies between 0 and 1. The prediction equation between X_i and Y is

$$ln(\frac{p}{1-p}) = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n$$

$ln(\frac{p}{1-p})$ is *log odds* and denotes the likelihood of the event taking place. It maps the probabilty between 0 and 1. The terms $b_0, b_1, b_2 \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{1-} = e^{b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n}$$

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

Logistic Regression, Gradient Descent

Loss function,

Python implementation

Logistic Regression

Logistic regression is a probabilistic model, used to establish the relationship between a binary dependent variable and one or more independent variables. Let X_i be the independent variable (training variable) and Y be the dependent variable. Linear Regression predicts probability of outcome of binary dependent variable (Y) and the output lies between 0 and 1. The prediction equation between X_i and Y is

$$ln(\frac{p}{1-p}) = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n$$

$ln(\frac{p}{1-p})$ is *log odds* and denotes the likelihood of the event taking place. It maps the probability between 0 and 1. The terms $b_0, b_1, b_2 \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{1-} = e^{b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n}$$

...s.uspertus.com is viewing your screen. Stop sharing Hide

Type here to search

ENG 19:33 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Comment Share

Logistic Regression, Gradient Descent

Loss function,

Python implementation

Logistic Regression

Logistic regression is a probabilistic model, used to establish the relationship between a binary dependent variable and one or more independent variables. Let X_i be the independent variable (training variable) and Y be the dependent variable. Linear Regression predicts probability of outcome of binary dependent variable (Y) and the output lies between 0 and 1. The prediction equation between X_i and Y is

$$ln(\frac{p}{1-p}) = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n$$

$ln(\frac{p}{1-p})$ is *log odds* and denotes the likelihood of the event taking place. It maps the probability between 0 and 1. The terms $b_0, b_1, b_2 \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{1-} = e^{b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n}$$

e.impartus.com is sharing your screen. Stop sharing Hide

Type here to search

ENG 19:33 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Logistic Regression

Logistic regression is a probabilistic model, used to establish the relationship between a binary dependent variable and one or more independent variables. Let X_i be the independent variable (training variable) and Y be the dependent variable. Linear Regression predicts probability of outcome of binary dependent variable (Y) and the output lies between 0 and 1. The prediction equation between X_i and Y is

$$ln(\frac{p}{1-p}) = b_0 + b_1x_1 + b_2x_2 + b_3x_3 \ldots b_nx_n$$

$ln(\frac{p}{1-p})$ is log odds and denotes the likelihood of the event taking place. It maps the probability between 0 and 1. The terms $b_0, b_1, b_2, \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{1-p} = e^{b_0 + b_1x_1 + b_2x_2 + b_3x_3 \ldots b_nx_n}$$

simplifying it further

$$p = \frac{e^{b_0 + b_1x_1 + b_2x_2 + b_3x_3 \ldots b_nx_n}}{1 + e^{b_0 + b_1x_1 + b_2x_2 + b_3x_3 \ldots b_nx_n}}$$

or

$$p = \frac{1}{1 + e^{-(b_0 + b_1x_1 + b_2x_2 + b_3x_3 \ldots b_nx_n)}}$$

This is the equation of the "Sigmoid Function" given by:

$$\sigma(x) = \frac{1}{}$$

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (69)

Disable group chat

Harvesh S V: yes

Sethumadhavan V: yes

Rajesh A: yes

MOHAMMAD ARIF MUSTAFA joined

SOURAV RAJ joined

RAVI CHANDRA VEDULA joined

SANDEEP WADHWA joined

SHARMISTHA BANERJEE joined

BASAVARAJ D JAKKANNAVAR joined

CHITRA DEVI A joined

RAGHAVENDRA G V joined

we can have multiclass also right?

MANISH SHARMA joined

BHIMSEN JOSHI joined

Sending to everyone

Type here to search

ENG 19:35 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

$$1+e^{-(\theta_0+\theta_1x_1+\theta_2x_2)}$$

This is the equation of the "Sigmoid Function" given by:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Deriving the sigmoid function,

$$\sigma'(x) = \frac{d}{dX}\sigma(x) = \frac{d}{dX}\frac{1}{(1+e^{-X})}$$

$$= \frac{d}{dX}(1+e^{-X})^{-1}$$

Using reciprocal rule, then rule of linearity,

$$= -(1+e^{-X})^{-2}.\frac{d}{dX}(1+e^{-X})$$

$$= -(1+e^{-X})^{-2}.(\frac{d}{dX}[1]+\frac{d}{dX}[e^{-X}])$$

Derivative of constant is 0,

$$= -(1+e^{-X})^{-2}.(0+e^{-X}\frac{d}{dX}[-X])$$

$$= (1+e^{-X})^{-2}.e^{-X}$$

$$= \frac{e^{-X}}{(1+e^{-X})}$$

0s completed at 19:37

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (75)

Disable group chat

BAVI CHANDRA VEDULA joined

SANDEEP WADHWA joined

SHARMISTHA BANERJEE joined

BASAVARAJ D JAKKANNAVAR joined

CHITRA DEVI A joined

RAGHAVENDRA G V joined

we can have multiclass also right?

MANISH SHARMA joined

BHIMSEN JOSHI joined

NEERAJ SABHNANI joined

ALOK joined

BOHRA AMAN DINESH joined

GEETANJALI RAI joined

SAHOO SUJIT KUMAR joined

GEETA KAMANI BALA PRASADA joined

Type here to search

ENG 19:37 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Now we will be using the above derived equation to make our predictions. Before that we will train our model to obtain the values of our parameters $b_0, b_1, b_2 \ldots$ that result in least error. This is where the error or loss function comes in.

Loss Function

Loss functions are used to calculate error in a model. There are two types of Loss functions. L1 and L2.

L1 loss function is used to minimize the error which is the sum of the all the absolute differences between the true value and the predicted value.

L2 Loss Function L2 Loss Function is used to minimize the error which is the sum of the all the squared differences between the true value and the predicted value.

The cost function of linear regression L1 and L2 can't be used in logistic regression because it is a non-convex function of weights. Optimizing algorithms like i.e gradient descent only converge convex function into a global minimum.

J = - ylog(h(x)) - (1 - y)log(1 - h(x))

here, y is the real target value

h(x) = sigmoid(wx + b)

For y = 0,

J = - log(1 - h(x))

and y = 1

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (77)

Disable group chat

SHARMISTHA BANERJEE joined

BASAVARAJ D JARKANDIAVAR joined

CHITRA DEVI A joined

RAGHAVENDRA G V joined

we can have multiclass also right?

MANISH SHARMA joined

BHIMSEN JOSHI joined

NEERAJ SABHNANI joined

ALOK joined

BOHRA AMAN DINESH joined

GEETHANJALI BAI joined

SAHOO SUJIT KUMAR joined

GEETA KAMANI BALA PRASADA joined

ASHU joined

ZEENATH BIBI joined

Type here to search

ENG 19:39 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

$$= -(1+e^{-X})^{-2}.\frac{d}{dX}(1+e^{-X})$$

$$= -(1+e^{-X})^{-2}.(\frac{d}{dX}[1]+\frac{d}{dX}[e^{-X}])$$

Derivative of constant is 0,

$$= -(1+e^{-X})^{-2}.(0+e^{-X}\frac{d}{dX}[-X])$$

$$= (1+e^{-X})^{-2}.e^{-X}$$

$$= \frac{e^{-X}}{(1+e^{-X})^2}$$

$$= \frac{1}{(1+e^{-X})}.\frac{e^{-X}+1-1}{(1+e^{-X})}$$

$$= \frac{1}{(1+e^{-X})}.(\frac{1+e^{-X}}{1+e^{-X}}-\frac{1}{1+e^{-X}})$$

$$= \frac{1}{(1+e^{-X})}.(1-\frac{1}{1+e^{-X}})$$

$$= \sigma(X).(1-\sigma(X))$$

a.impartus.com is sharing your screen. Stop sharing Hide

we can have multiclass also right?

Type here to search

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Derivative of constant is 0,

$$
\begin{aligned}
&= -(1+e^{-X})^{-2}.(0+e^{-X}\frac{d}{dX}[-X]) \\
&= (1+e^{-X})^{-2}.e^{-X} \\
&= \frac{e^{-X}}{(1+e^{-X})^2} \\
&= \frac{1}{(1+e^{-X})}.\frac{e^{-X}+1-1}{(1+e^{-X})} \\
&= \frac{1}{(1+e^{-X})}.(\frac{1+e^{-X}}{1+e^{-X}}-\frac{1}{1+e^{-X}}) \\
&= \frac{1}{(1+e^{-X})}.(1-\frac{1}{1+e^{-X}}) \\
&= \sigma(X).(1-\sigma(X))
\end{aligned}
$$

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (78)

Disable group chat

BASAVARAJ D JAKKANNAVAR joined

CHITRA DEVI A joined

RAGHAVENDRA G V joined

we can have multiclass also right?

MANISH SHARMA joined

BHIMSEN JOSHI joined

NEERAJ SABHNANI joined

ALOK joined

BOHRA AMAN DINESH joined

GEETANJALI RAI joined

one vs rest is used for multiclass in logistic regression

SAHOO SUJIT KUMAR joined

GEETA RAMANI BALA PRASADA joined

ASHU joined

ZEENATH BIBI joined

JAYATIRTHA BABACHANDRA KATTI joined

Type here to search

ENG 19:39 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

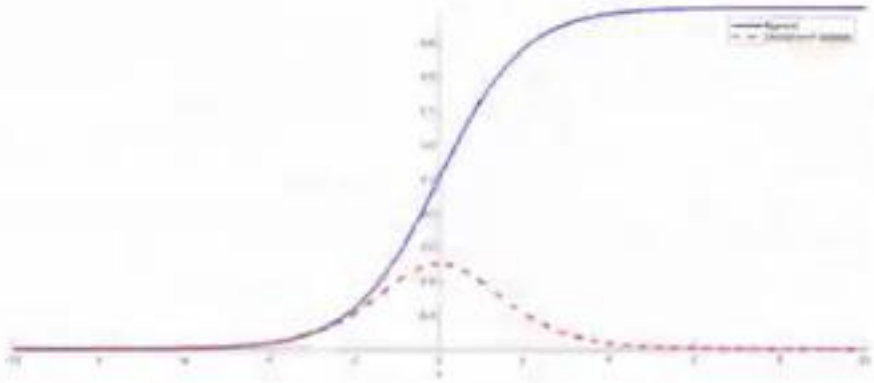
+ Code + Text

Editing

$$= \frac{1}{(1+e^{-X})} \cdot \left(\frac{1+e^{-X}}{1+e^{-X}} - \frac{1}{1+e^{-X}}\right)$$

$$= \frac{1}{(1+e^{-X})} \cdot \left(1 - \frac{1}{1+e^{-X}}\right)$$

$$= \sigma(X).(1-\sigma(X))$$



Now we will be using the above derived equation to make our predictions. Before that we will train our model to obtain the values of our parameters $b_0, b_1, b_2 \ldots$ that result in least error. This

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (81)

Disable group chat

ALOK joined

BOHRA AMAN DINESH joined

GEETANJALI RAI joined

SAHOO SUJIT KUMAR joined

GEETA RAMANI BALA PRASADA joined

ASHU joined

ZEENATH BIBI joined

JAYATHIRTHA RAMACHANDRA KATTI joined

MAURYA ARVIND CHAMPAKLAL SADHANADEVI joined

Arijit Barat
mam we don't have this notebook in lab. So please share it.

RANJAN NEGI joined

thanks

ANUSHA LINDA KOSTKA J. S. joined

Type here to search

ENG 19:40 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

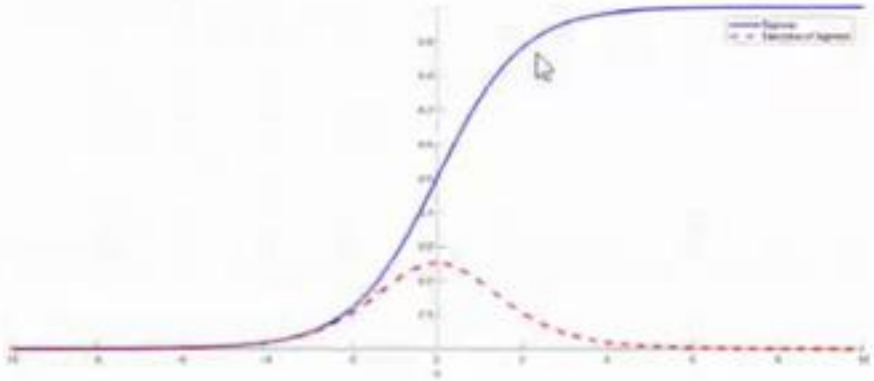
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

$$= \frac{1}{(1+e^{-X})} . (1 - \frac{1}{1+e^{-X}})$$
$$= \sigma(X) . (1 - \sigma(X))$$



Now we will be using the above derived equation to make our predictions. Before that we will train our model to obtain the values of our parameters $b_0, b_1, b_2 \ldots$ that result in least error. This is where the error or loss function comes in.

Loss Function

Loss functions are used to calculate error in a model. Thes

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Chat Participants (81)

Disable group chat

GEETANJALI RAI joined

one is next is used for multiclass in logistic regression

SAHOO SUJIT KUMAR joined

GEETA RAMANI BALA PRASADA joined

ASHU joined

ZEENATH BIBI joined

JAYATHIRTHA RAMACHANDRA KATTI joined

MAURYA ARVIND CHAMPAKLAL SADHANADEVI joined

Arijit Barat
mam we don't have this notebook in lab. So please share it.

RANJAN NEGI joined

thanks

ANUSHA LINDA KOSTKA J. E. joined

AMOGH SINGHAL joined

SARVESHWARAN K joined

Sending to everyone

Type here to search

ENG 19:41 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

and y = 1,

J = - log(h(x))

Gradient Calculation :

repeat until convergence {

tmpi = wi - alpha * dwi

wi = tmpi }

where alpha is the learning rate.

Using the chain rule

$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial a} \bullet \frac{\partial a}{\partial z} \bullet \frac{\partial z}{\partial w}$$

here, a = sigmoid(z) and z = wx + b.

```
# Importing libraries
import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
plt.rc("font", size=14)
```

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (86)

Disable group chat

Why we take the derivative main of sigmoid function

DHEERENDRA KUMAR SINGH joined

ok

HARSH VASHISHT joined

PREETI REKHA SAHU joined

ok mam

thnx

SINDIA P CHOOLACKAL joined

S KRISHNAMOORTHY joined

ARUN AITHAL S joined

CHINSU ELIZABETH MATHEW joined

MANOJ K K joined

mam, can you repeat why we cant take L1 and L2

Sending to everyone

Type here to search

ENG 19:44 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Using the chain rule

$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w}$$

here, a = sigmoid(z) and z = wx + b

```
# Importing libraries
import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
```

[4]
```
from google.colab import files
uploaded = files.upload()
import io
data = pd.read_csv(io.BytesIO(uploaded['diabetes.csv']))
data.head()
```

0s comp

Chat Participants (86)

Disable group chat

ok

HARSH VASHISHT joined

PREETI REKHA SAHU joined

ok mam

thnx

SINDIA P CHOOLAKKAL joined

S KRISHNAMOORTHY joined

ARUN AITHAL S joined

CHINSU ELIZABETH MATHEW joined

MANOJ K K joined

mam, can you repeat why we cant take L1 and L2

using MSE

Arijit Barat
least square error

RAHUL SAHA joined

e.umpartus.com is sharing your screen. Stop sharing Hide

Type here to search

ENG 19:46 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[5] data = data.dropna()
    print(data.shape)
    print(list(data.columns))

    (768, 9)
    ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

```
[6] data['Outcome'].value_counts()
    sns.countplot(x='Outcome',data=data, palette='hls')
    plt.show()
```

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Chat Participants (87)

Disable group chat

HARSH VASHISHT joined

PREETI REKHA SAHU joined

ok mam

thnx

SINDIA P CHOOLACKAL joined

S KRISHNAMOORTHY joined

ARUN AITHAL S joined

CHINGU ELIZABETH MATHEW joined

MANOJ K K joined

mam, can you repeat why we cant take L1 and L2

using MSE

Arijit Barat
lease square error

RAHUL SAHA joined

ABHAY RAMESH JUNGHARE joined

Type here to search

ENG 19:47 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

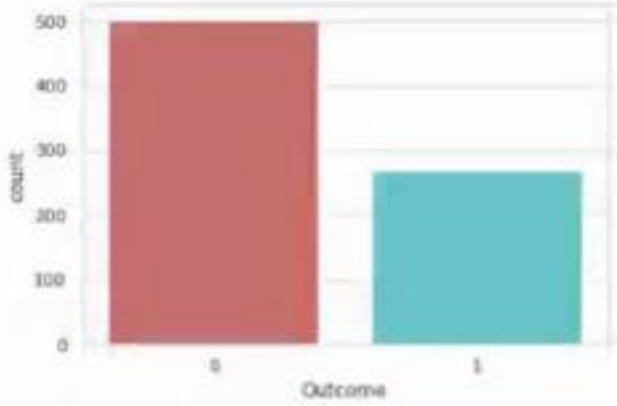
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[6] data['Outcome'].value_counts()
    sns.countplot(x='Outcome',data=data, palette='hls')
    plt.show()
```




```
[7] # Logistic Regression
    class LogitRegression() :
        def __init__( self, learning_rate, iterations):
```

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (86)

Disable group chat

HARSH VASHISHT joined

PREETI REKHA SAHU joined

ok mam

thnx

SINDIA P CHOOLACKAL joined

S KRISHNAMOORTHY joined

ARUN AITHAL S joined

CHINSU ELIZABETH MATHEW joined

MANOJ K K joined

mam, can you repeat why we cant take L1 and L2

using MSE

Arijit Barat
lease square error

RAHUL SAHA joined

ABHAY RAMESH JUNGHARE joined

Type here to search

ENG 19:48 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

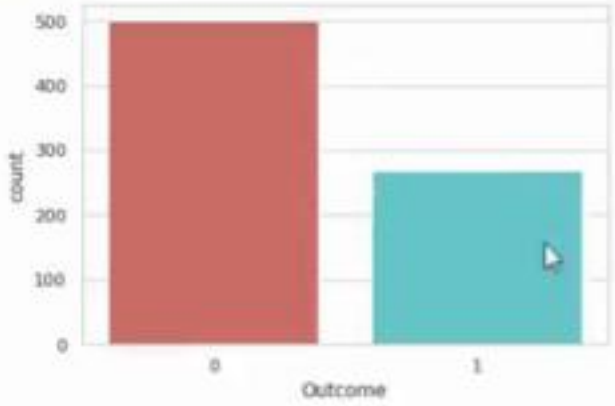
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
data['Outcome'].value_counts()
sns.countplot(x='Outcome',data=data, palette='hls')
plt.show()
```




```
[7] # Logistic Regression
    class LogitRegression() :
        def __init__( self, learning_rate, iterations
```

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (86)

Disable group chat

PREETI REKHA SAHU joined

ok mam

thnx

SINDIA P CHOOLACKAL joined

S KRISHNAMOORTHY joined

ARUN AITHAL S joined

CHINSU ELIZABETH MATHEW joined

MANOJ K K joined

mam, can you repeat why we cant take L1 and L2

using MSE

Arijit Barat
lease square error

RAHUL SAHA joined

ABHAY RAMESH JUNGHARE joined

K GIRISH GOPINATHAN joined

Type here to search

ENG 19:49 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

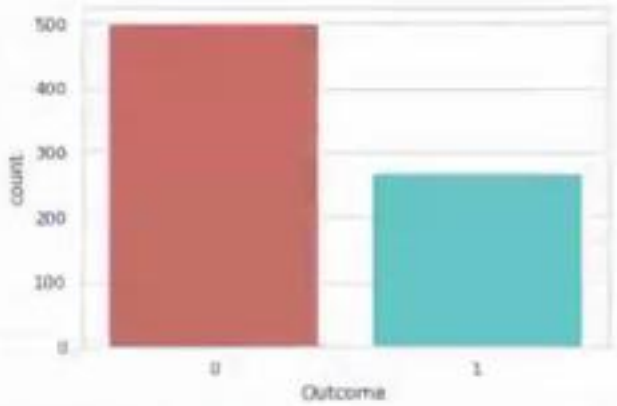
+ Code + Text

RAM Disk

Editing

```
[5] (768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

```
data['Outcome'].value_counts()
sns.countplot(x='Outcome',data=data, palette='hls')
plt.show()
```



a.impartus.com is sharing your screen. Stop sharing Hide

0s comp

Chat Participants (87)

Disable group chat

using MSE

Arijit Barat
lease square error

RAHUL SAHA joined

ABHAY RAMESH JUNGHARE joined

K GIRISH GOPINATHAN joined

Alok
how they are different

model 1 and Model 2

Rakhi Nandy
Yes Ma'am

Seelam Reshma Mon...
yes mam.

Gunasekaran R
Mam, are you balancing the data points?

S Bhagath
what will be model2?

Sending to everyone

Type here to search

ENG 19:51 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[6]

[7]

```
# Logistic Regression
class LogitRegression() :
    def __init__( self, learning_rate, iterations ) :
        self.learning_rate = learning_rate
        self.iterations = iterations

    # Function for model training
    def fit( self, X, Y ) :
        # no_of_training_examples, no_of_features
        self.m, self.n = X.shape
        # weight initialization
        self.W = np.zeros( self.n )
        self.b = 0
        self.X = X
        self.Y = Y

        # gradient descent learning
```

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (86)

Disable group chat

how they are different

model 1 and Model 2

Yes Ma'am

yes mam.

Mam, are you balancing the data points?

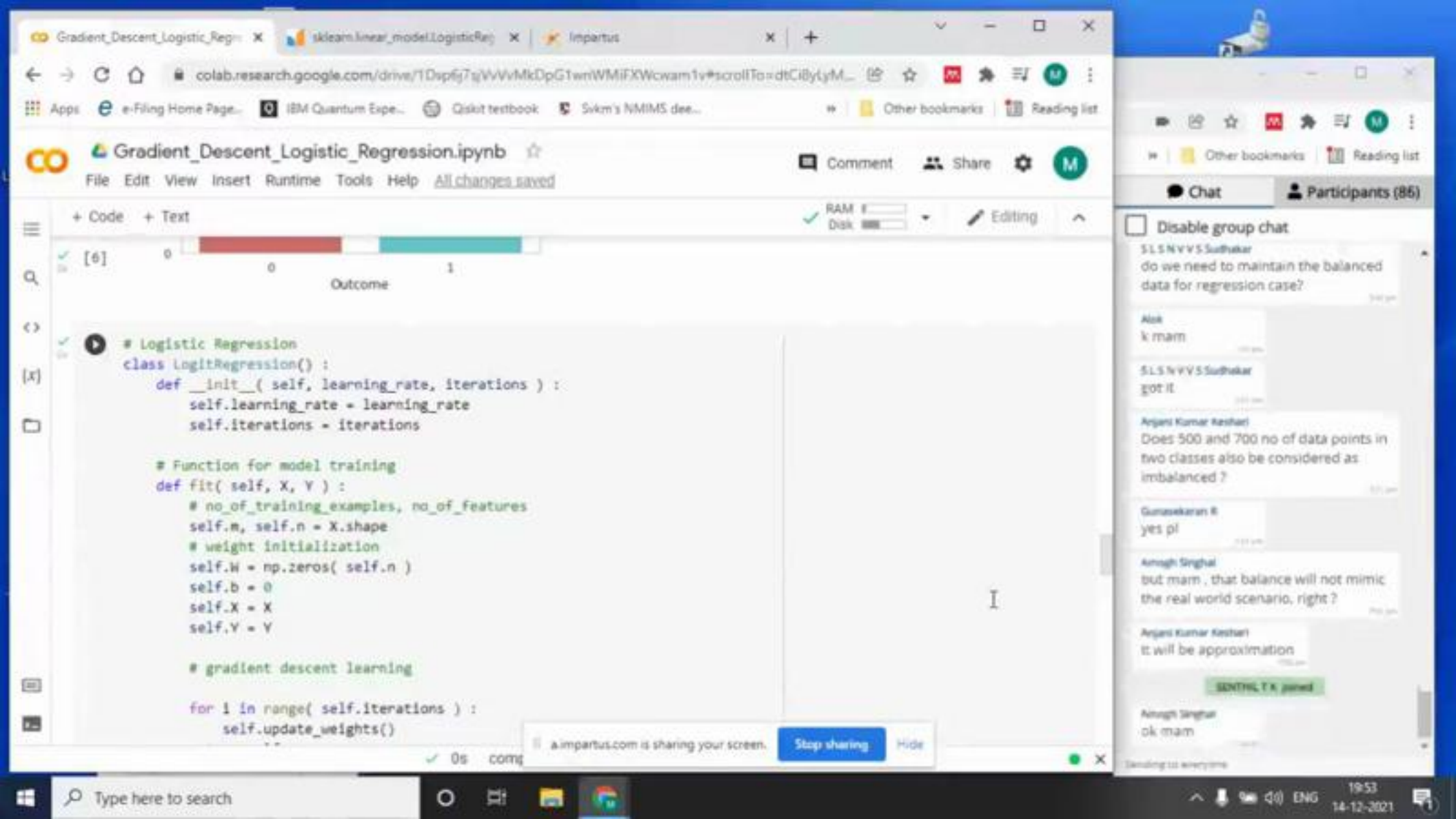
what will be model2?

Both are scikit learn Model ?

do we need to maintain the balanced data for regression case?

Type here to search

ENG 19:51 14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[6]
0
1
Outcome
Logistic Regression
class LogitRegression() :
def __init__(self, learning_rate, iterations) :
self.learning_rate = learning_rate
self.iterations = iterations
# Function for model training
def fit(self, X, Y) :
no_of_training_examples, no_of_features
self.m, self.n = X.shape
# weight initialization
self.W = np.zeros(self.n)
self.b = 0
self.X = X
self.Y = Y
# gradient descent learning
for i in range(self.iterations) :
self.update_weights()
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (86)
Disable group chat
do we need to maintain the balanced data for regression case?
k mam
got it
Does 500 and 700 no of data points in two classes also be considered as imbalanced ?
yes pl
but mam , that balance will not mimic the real world scenario, right ?
It will be approximation
ok mam
Type here to search
ENG
19:53
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

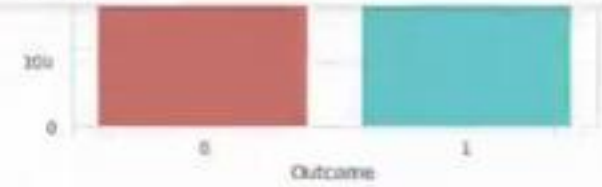
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[6]



[7]

```
# Logistic Regression
class LogitRegression() :
    def __init__( self, learning_rate, iterations ) :
        self.learning_rate = learning_rate
        self.iterations = iterations

    # Function for model training
    def fit( self, X, Y ) :
        # no_of_training_examples, no_of_features
        self.m, self.n = X.shape
        # weight initialization
        self.W = np.zeros( self.n )
        self.b = 0
        self.X = X
        self.Y = Y

        # gradient descent learning
```

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (85)

Disable group chat

S L S N V V S Sudhakar
do we need to maintain the balanced data for regression case?

Alok
k mam

S L S N V V S Sudhakar
got it

Anjani Kumar Kashari
Does 500 and 700 no of data points in two classes also be considered as imbalanced ?

Gunasekaran R
yes pl

Amogh Singhal
but mam , that balance will not mimic the real world scenario. right ?

Anjani Kumar Kashari
it will be approximation

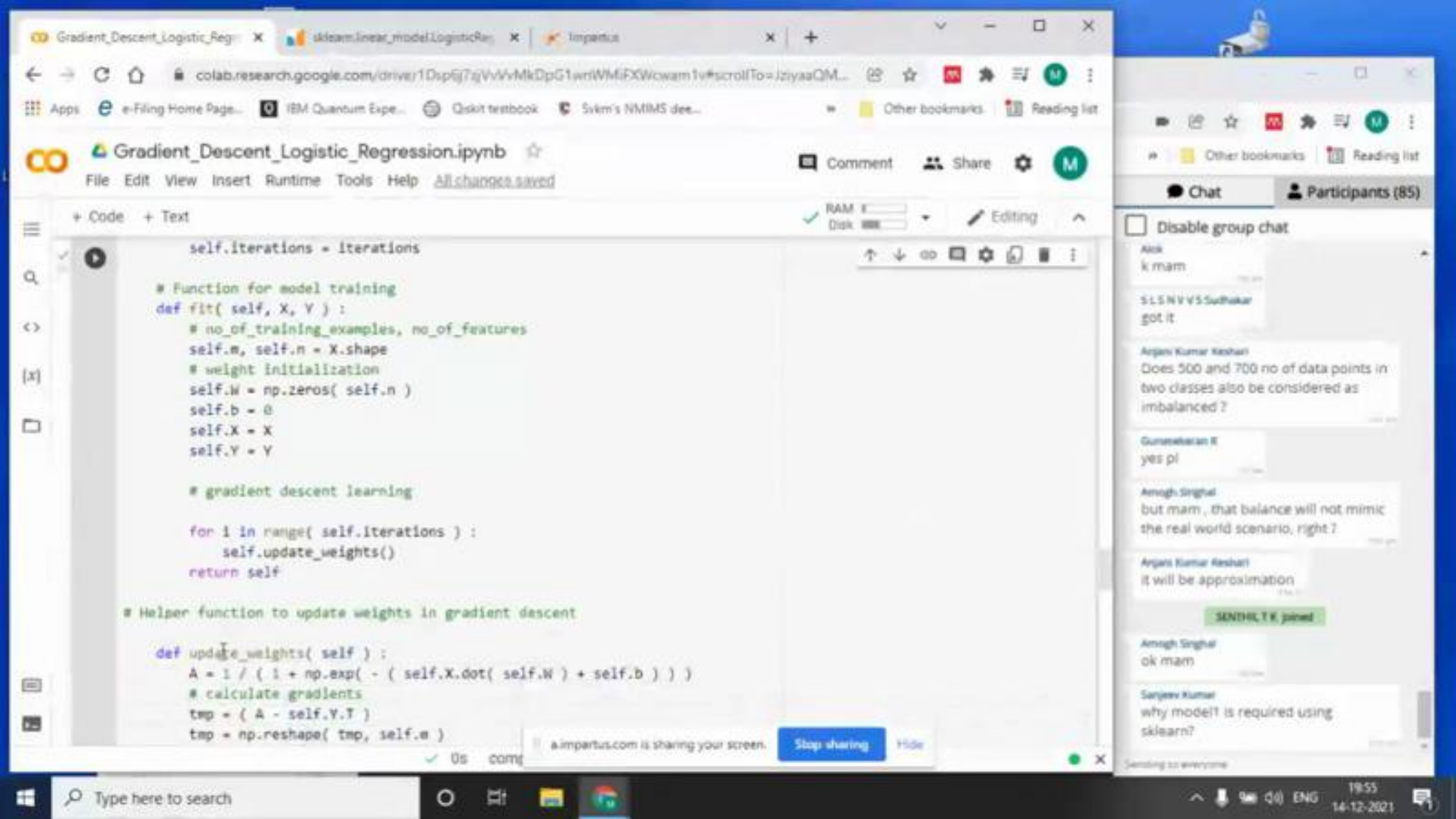
SENTHIL T K joined

Amogh Singhal
ok mam

Sending to everyone

Type here to search

ENG 19:54 14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
self.iterations = iterations
# Function for model training
def fit(self, X, Y) :
no_of_training_examples, no_of_features
self.m, self.n = X.shape
# weight initialization
self.W = np.zeros(self.n)
self.b = 0
self.X = X
self.Y = Y
# gradient descent learning
for i in range(self.iterations) :
self.update_weights()
return self
Helper function to update weights in gradient descent
def update_weights(self) :
A = 1 / (1 + np.exp(- (self.X.dot(self.W) + self.b)))
# calculate gradients
tmp = (A - self.Y.T)
tmp = np.reshape(tmp, self.m)
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (85)
Disable group chat
Alok
k mam
S L S N V V S Sudhakar
got it
Anjani Kumar Keshari
Does 500 and 700 no of data points in two classes also be considered as imbalanced ?
Gunaseelaran R
yes pl
Amogh Singhal
but mam , that balance will not mimic the real world scenario, right ?
Anjani Kumar Keshari
it will be approximation
SENTHIL T K joined
Amogh Singhal
ok mam
Sanjeev Kumar
why model1 is required using sklearn?
Type here to search
ENG
19:55
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[5] data = data.dropna()
    print(data.shape)
    print(list(data.columns))

    (768, 9)
    ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

```
[6] data['Outcome'].value_counts()
    sns.countplot(x='Outcome',data=data, palette='hls')
    plt.show()
```

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (85)

Disable group chat

S L S N V V S Sudhakar
got it

Anjani Kumar Keshari
Does 500 and 700 no of data points in two classes also be considered as imbalanced ?

Gunasekaran R
yes pl

Amogh Singhal
but mam , that balance will not mimic the real world scenario, right ?

Anjani Kumar Keshari
it will be approximation

SENTHIL T K joined

Amogh Singhal
ok mam

Sanjeev Kumar
why model1 is required using sklearn?

1st line is related to sigmoid function?

Sending to everyone

Type here to search

ENG 19:56 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
        # no_of_training_examples, no_of_features
        self.m, self.n = X.shape
        # weight initialization
        self.W = np.zeros( self.n )
        self.b = 0
        self.X = X
        self.Y = Y

        # gradient descent learning

        for i in range( self.iterations ) :
            self.update_weights()
        return self

# Helper function to update weights in gradient descent

    def update_weights( self ) :
        A = 1 / ( 1 + np.exp( - ( self.X.dot( self.W ) + self.b ) ) )
        # calculate gradients
        tmp = ( A - self.Y.T )
        tmp = np.reshape( tmp, self.m )
        dW = np.dot( self.X.T, tmp ) / self.m
        db = np.sum( tmp ) / self.m
        # update weights
        self.W = self.W - self.learning_rate *
```

a.impartus.com is sharing your screen. Stop sharing Hide

0s comp

Chat Participants (86)

Disable group chat

got it

Anjani Kumar Keshari
Does 500 and 700 no of data points in two classes also be considered as imbalanced ?

Gunasekaran R
yes pl

Amogh Singhal
but mam , that balance will not mimic the real world scenario, right ?

Anjani Kumar Keshari
it will be approximation

SENTHIL T K joined

Amogh Singhal
ok mam

Sanjeev Kumar
why model1 is required using sklearn?

1st line is related to sigmoid function?

ARAVIND P joined

Type here to search

ENG 19:57 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

```
[5] data = data.dropna()
    print(data.shape)
    print(list(data.columns))

    (768, 9)
    ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

```
[6] data['Outcome'].value_counts()
    sns.countplot(x='Outcome',data=data, palette='bwr')
    plt.show()
```

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (86)

Disable group chat

got it

Anjani Kumar Keshari
Does 500 and 700 no of data points in two classes also be considered as imbalanced ?

Gunasekaran R
yes pl

Anugh Singhal
but mam , that balance will not mimic the real world scenario, right ?

Anjani Kumar Keshari
it will be approximation

SENTHIL Y K joined

Anugh Singhal
ok mam

Sanjeev Kumar
why model1 is required using sklearn?

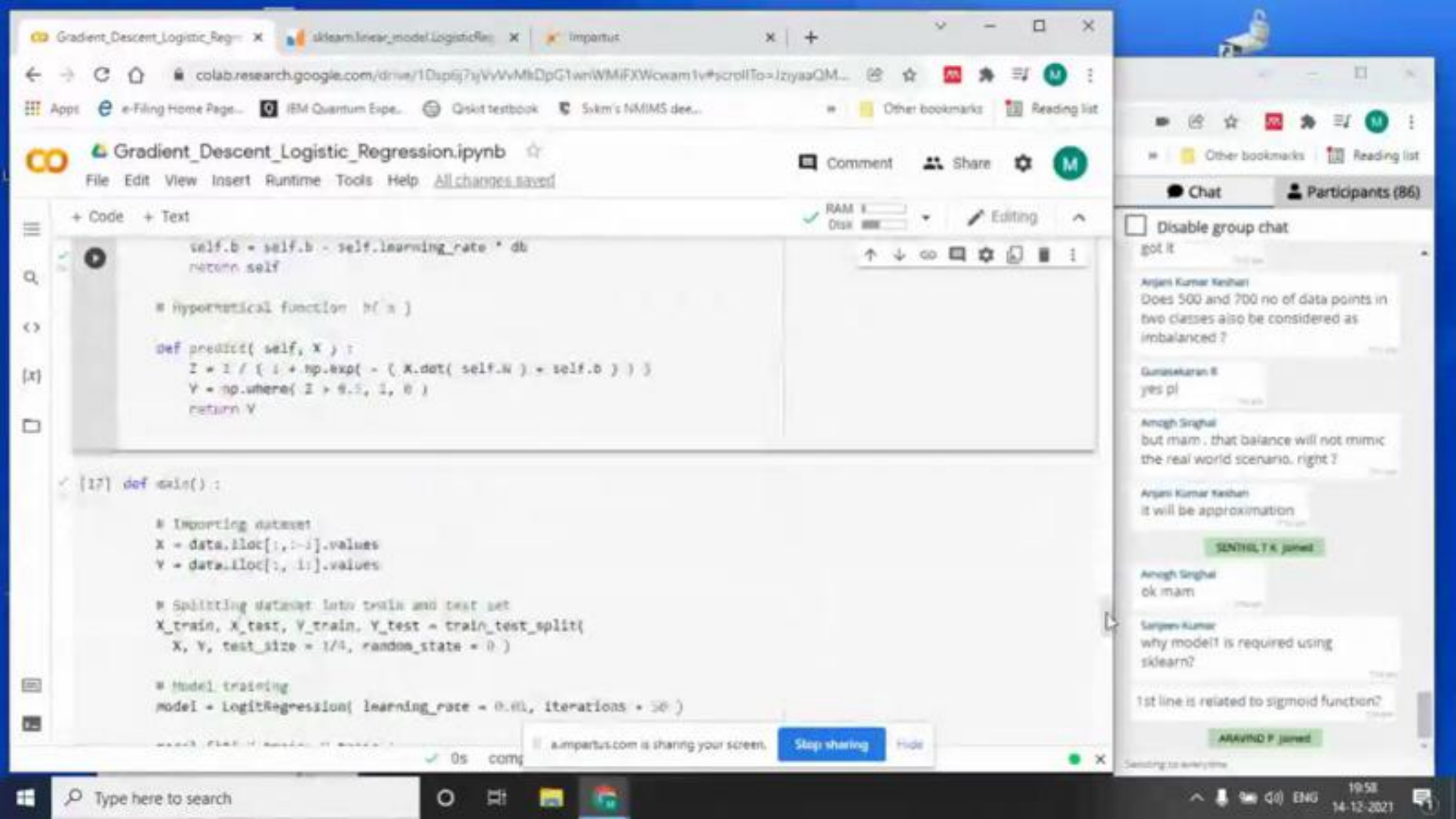
1st line is related to sigmoid function?

ARAVIND P joined

Sending to everyone

Type here to search

ENG 19:57 14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
self.b = self.b - self.learning_rate * db
return self
Hypothetical function h(x)
def predict(self, X) :
Z = 1 / (1 + np.exp(- (X.dot(self.W) + self.b)))
Y = np.where(Z > 0.5, 1, 0)
return Y
[17] def main() :
# Importing dataset
X = data.iloc[:,:-1].values
Y = data.iloc[:, 1:].values
# Splitting dataset into train and test set
X_train, X_test, Y_train, Y_test = train_test_split(
X, Y, test_size = 1/4, random_state = 0)
# Model training
model = LogitRegression(learning_rate = 0.01, iterations = 50)
0s
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (86)
Disable group chat
got it
Anjani Kumar Keshari
Does 500 and 700 no of data points in two classes also be considered as imbalanced ?
Gunasekaran R
yes pl
Anugh Singhal
but mam . that balance will not mimic the real world scenario. right ?
Anjani Kumar Keshari
it will be approximation
SENTHIL T K joined
Anugh Singhal
ok mam
Sanjeev Kumar
why model1 is required using sklearn?
1st line is related to sigmoid function?
ARAVIND P joined
Type here to search
ENG
19:58
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
        for i in range( self.iterations ) :
            self.update_weights()
        return self

# Helper function to update weights in gradient descent

    def update_weights( self ) :
        A = 1 / ( 1 + np.exp( - ( self.X.dot( self.W ) + self.b ) ) )
        # calculate gradients
        tmp = ( A - self.Y.T )
        tmp = np.reshape( tmp, self.m )
        dW = np.dot( self.X.T, tmp ) / self.m
        db = np.sum( tmp ) / self.m
        # update weights
        self.W = self.W - self.learning_rate * dW
        self.b = self.b - self.learning_rate * db
        return self

    # Hypothetical function  h( x )

    def predict( self, X ) :
        Z = 1 / ( 1 + np.exp( - ( X.dot( self.W ) + self.b ) ) )
        Y = np.where( Z > 0.5, 1, 0 )
        return Y
```

a.impartus.com is sharing your screen. Stop sharing Hide

0s comp

Chat Participants (88)

Disable group chat

yes p!

but mam , that balance will not mimic the real world scenario. right?

it will be approximation

SENTHIL T K joined

ok mam

why model1 is required using sklearn?

1st line is related to sigmoid function?

ARAVIND P joined

where we are calculating error in our function mam ..

ANSHUU PUN joined

Type here to search

ENG 19:58 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

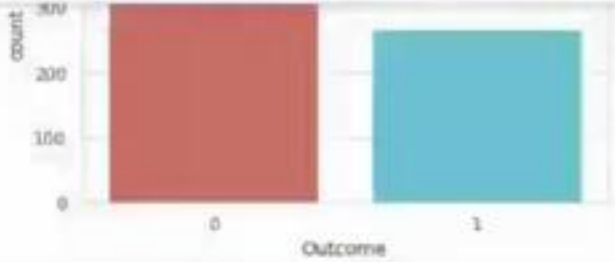
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[6]




```
# Logistic Regression
class LogitRegression() :
    def __init__( self, learning_rate, iterations ) :
        self.learning_rate = learning_rate
        self.iterations = iterations

    # Function for model training
    def fit( self, X, Y ) :
        # no_of_training_examples, no_of_features
        self.m, self.n = X.shape
        # weight initialization
        self.W = np.zeros( self.n )
        self.b = 0
        self.X = X
```

a.impartus.com is sharing your screen. Stop sharing Hide

0s comp

Chat Participants (88)

Disable group chat

yes pl

but mam , that balance will not mimic the real world scenario. right ?

it will be approximation

ok mam

why model1 is required using sklearn?

1st line is related to sigmoid function?

where we are calculating error in our function mam ..

Type here to search

ENG 19:59 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
def main() :

    # Importing dataset
    X = data.iloc[:,:-1].values
    Y = data.iloc[:, -1:].values

    # Splitting dataset into train and test set
    X_train, X_test, Y_train, Y_test = train_test_split(
      X, Y, test_size = 1/4, random_state = 0 )

    # Model training
    model = LogitRegression( learning_rate = 0.01, iterations = 1000 )

    model.fit( X_train, Y_train )
    model1 = LogisticRegression(solver='lbfgs', max_iter=10)
    model1.fit( X_train, Y_train)

    # Prediction on test set
    Y_pred = model.predict( X_test )
    Y_pred1 = model1.predict( X_test )

    # measure performance
    correctly_classified = 0
```

Chat | Participants (87)

Disable group chat

why model1 is required using sklearn?

1st line is related to sigmoid function?

Where we are calculating error in our function mam ..

but we are not comparing with real outputs

yes mam

mam cant hear you

No Audio

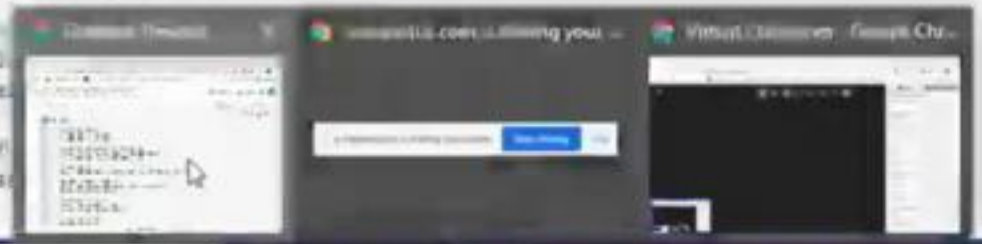
Type here to search

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
def main() :

    # Importing dataset
    X = data.iloc[:,:-1].values
    Y = data.iloc[:, 4:].values

    # splitting dataset into train and test set
    X_train, X_test, Y_train, Y_test = train_test_split(
      X, Y, test_size = 1/4, random_state = 0 )

    # Model training
    model = LogitRegression( learning_rate = 0.01, iterations = 1000 )

    model.fit( X_train, Y_train )
    model1 = LogisticRegression(solver='lbfgs', max_iter=10)
    model1.fit( X_train, Y_train)

    # Prediction
    Y_pred = model
    Y_pred1 = model

    # accuracy
    correctly_clas
```



Type here to search

ENG 20:01 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

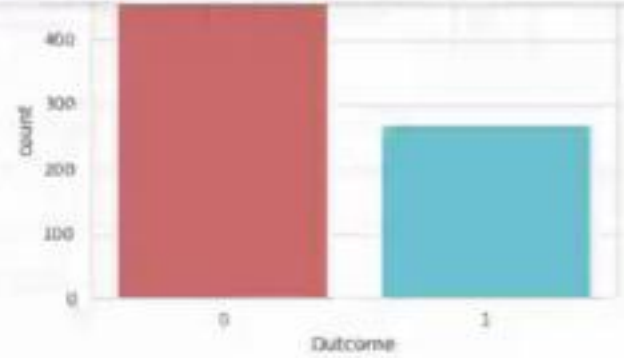
Comment Share

+ Code + Text

RAM Disk

Editing

[4]



[7]

```
# Logistic Regression
class LogitRegression() :
    def __init__( self, learning_rate, iterations ) :
        self.learning_rate = learning_rate
        self.iterations = iterations

    # function for model training
    def fit( self, X, Y ) :
        # no_of_training_examples, no_of_features
        self.m, self.n = X.shape
        # weight initialization
```

0s comp

aumpartus.com is sharing your screen. Stop sharing Hide

Chat Participants (87)

Disable group chat

Bohra Aman Dinesh: No Audio

Sandeep Wadhwa: No AV

Geeta Ramani Bala P...: now we can

Alok: yes

Bohra Aman Dinesh: Back

Seelam Reshma Mon...: now we can

Chitra Devi A: yes

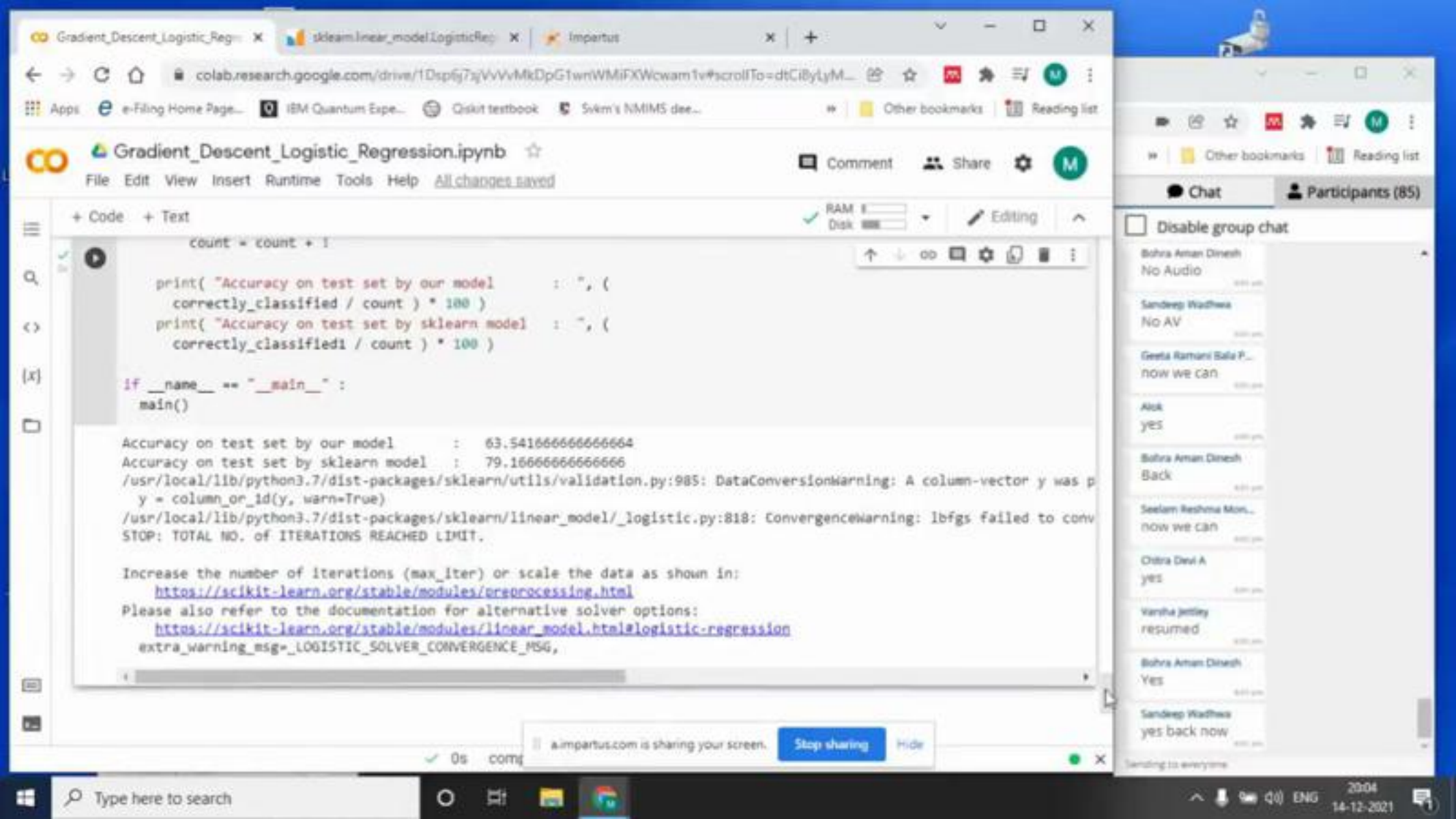
Vanitha Jestley: resumed

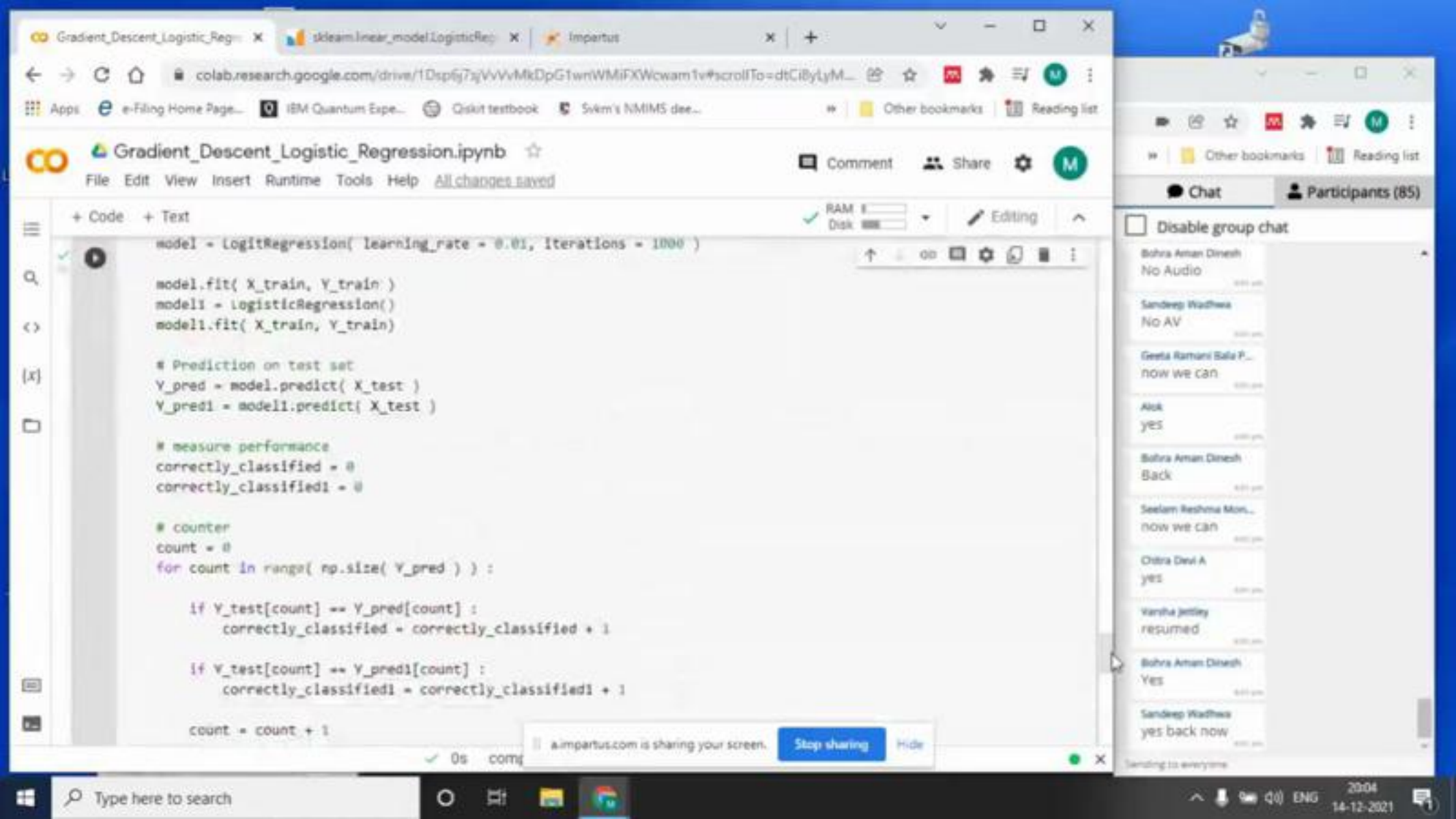
Bohra Aman Dinesh: Yes

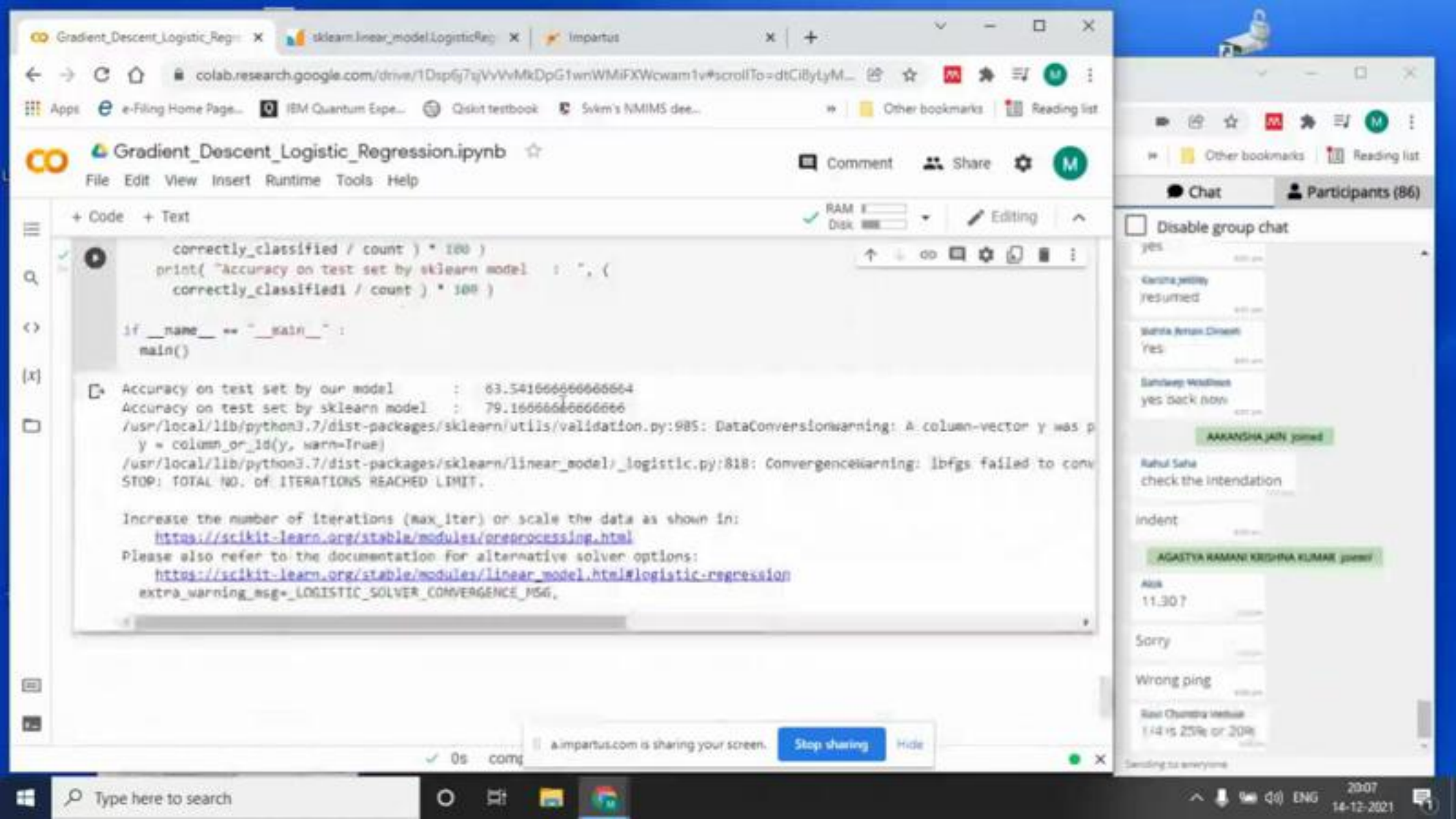
Sandeep Wadhwa: yes back now

Type here to search

ENG 20:02 14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
count = count + 1
print("Accuracy on test set by our model : ", (
correctly_classified / count) * 100)
print("Accuracy on test set by sklearn model : ", (
correctly_classified1 / count) * 100)
if __name__ == "__main__" :
main()
Accuracy on test set by our model : 63.541666666666664
Accuracy on test set by sklearn model : 79.16666666666666
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:985: DataConversionWarning: A column-vector y was p
y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to conv
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
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (85)
Disable group chat
Bohra Aman Dinesh
No Audio
Sandeep Wadhwa
No AV
Geeta Ramani Balla P...
now we can
Alok
yes
Bohra Aman Dinesh
Back
Seelam Reshma Mon...
now we can
Chitra Devi A
yes
Varsha Jetley
resumed
Bohra Aman Dinesh
Yes
Sandeep Wadhwa
yes back now
Type here to search
ENG
20:04
14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
model = LogitRegression(learning_rate = 0.01, iterations = 1000)
model.fit(X_train, Y_train)
model1 = LogisticRegression()
model1.fit(X_train, Y_train)
# Prediction on test set
Y_pred = model.predict(X_test)
Y_pred1 = model1.predict(X_test)
# measure performance
correctly_classified = 0
correctly_classified1 = 0
# counter
count = 0
for count in range(np.size(Y_pred)) :
if Y_test[count] == Y_pred[count] :
correctly_classified = correctly_classified + 1
if Y_test[count] == Y_pred1[count] :
correctly_classified1 = correctly_classified1 + 1
count = count + 1
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (85)
Disable group chat
No Audio
No AV
now we can
yes
Back
now we can
yes
resumed
Yes
yes back now
Type here to search
ENG
20:04
14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
correctly_classified / count) * 100)
print("Accuracy on test set by sklearn model : ", (
correctly_classified1 / count) * 100)
if __name__ == "__main__" :
main()
Accuracy on test set by our model : 63.541666666666664
Accuracy on test set by sklearn model : 79.16666666666666
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:985: DataConversionWarning: A column-vector y was p
y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to conv
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
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
Chat
Participants (86)
Disable group chat
yes
resumed
Yes
yes back now
AAKANSHA JAIN joined
check the intendation
indent
AGASTYA RAMANI KRISHNA KUMAR joined
11.30 ?
Sorry
Wrong ping
a.impartus.com is sharing your screen.
Stop sharing
Hide
Type here to search
ENG
20:07
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
    print( "Accuracy on test set by our model       : ", (
      correctly_classified / count ) * 100 )
    print( "Accuracy on test set by sklearn model   : ", (
      correctly_classified1 / count ) * 100 )

if __name__ == "__main__" :
  main()
```

```
Accuracy on test set by our model       :  72.91666666666666
Accuracy on test set by sklearn model   :  79.16666666666666
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:985: DataConversionWarning: A column-vector y was p
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to conv
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
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

a.impartus.com is sharing your screen. Stop sharing Hide

0s

Chat Participants (85)

Disable group chat

change LR

regularization

How to decide on no of iterations for good fit

learning rate

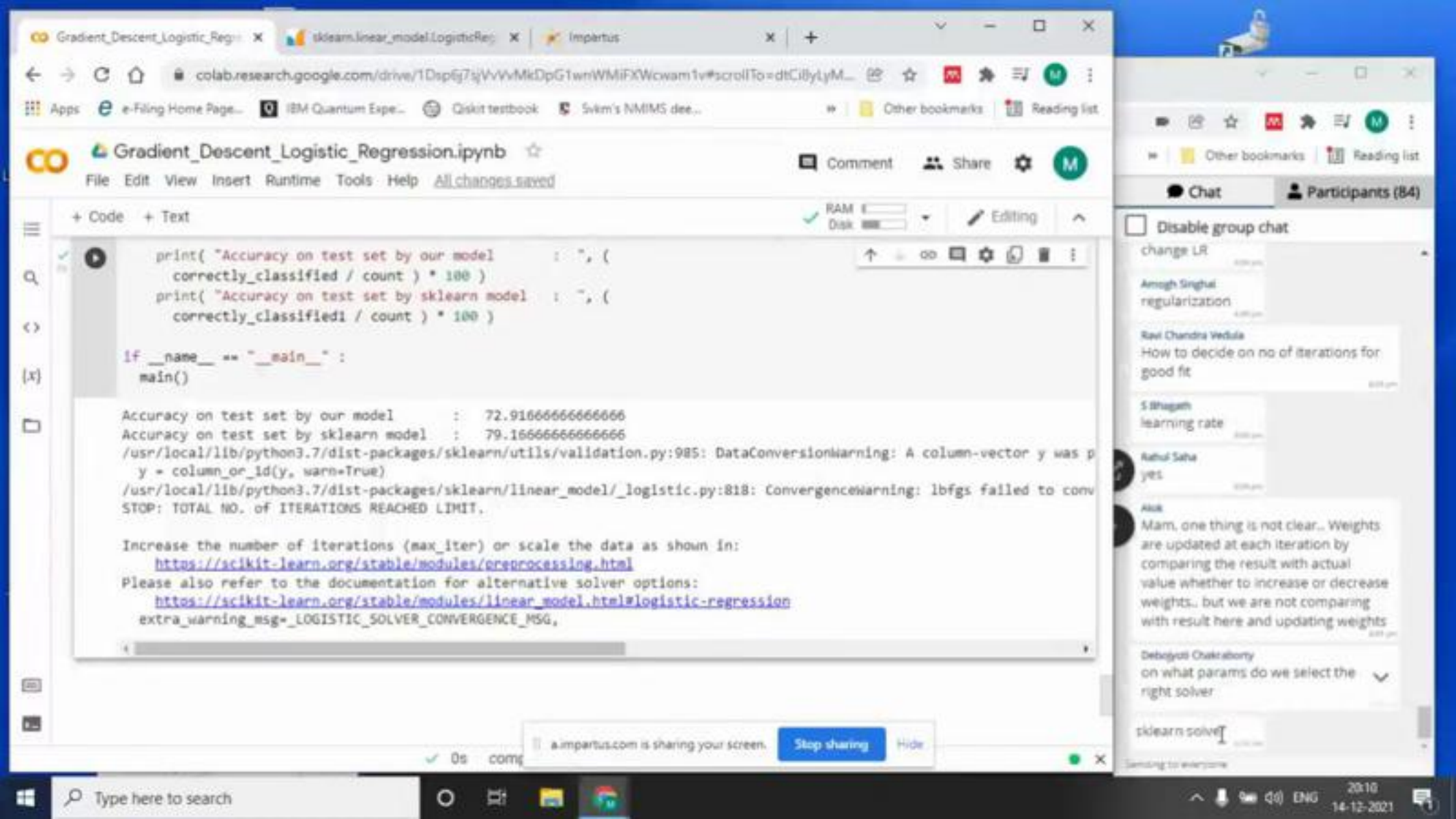
yes

Mam, one thing is not clear.. Weights are updated at each iteration by comparing the result with actual value whether to increase or decrease weights.. but we are not comparing with result here and updating weights

on what params do we select the right solver

Type here to search

ENG 20:10 14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
print("Accuracy on test set by our model : ", (
correctly_classified / count) * 100)
print("Accuracy on test set by sklearn model : ", (
correctly_classified1 / count) * 100)
if __name__ == "__main__" :
main()
Accuracy on test set by our model : 72.91666666666666
Accuracy on test set by sklearn model : 79.16666666666666
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:985: DataConversionWarning: A column-vector y was p
y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to conv
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
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (84)
Disable group chat
change LR
Amogh Singhal
regularization
Ravi Chandra Vedula
How to decide on no of iterations for good fit
S Bhagath
learning rate
Rahul Saha
yes
Alok
Mam, one thing is not clear... Weights are updated at each iteration by comparing the result with actual value whether to increase or decrease weights.. but we are not comparing with result here and updating weights
Debojyoti Chakraborty
on what params do we select the right solver
sklearn solver
Type here to search
ENG
20:10
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

```
[7]         # weight initialization
            self.W = np.zeros( self.n )
            self.b = 0
            self.X = X
            self.Y = Y

            # gradient descent learning

            for i in range( self.iterations ) :
                self.update_weights()
            return self

    # Helper function to update weights in gradient descent

        def update_weights( self ) :
            A = 1 / ( 1 + np.exp( - ( self.X.dot( self.W ) + self.b ) ) )
            # calculate gradients
            tmp = ( A - self.Y.T )
            tmp = np.reshape( tmp, self.m )
            dW = np.dot( self.X.T, tmp ) / self.m
            db = np.sum( tmp ) / self.m
            # update weights
            self.W = self.W - self.learning_rate * dW
            self.b = self.b - self.learning_rate * db
            return self
```

0s comp

a.umpartus.com is sharing your screen. Stop sharing Hide

Chat Participants (86)

Disable group chat

Amogh Singhal
regularization

Ravi Chandra Vedula
How to decide on no of iterations for good fit

S Bhagath
learning rate

Rahul Saha
yes

Alok
Mam, one thing is not clear.. Weights are updated at each iteration by comparing the result with actual value whether to increase or decrease weights.. but we are not comparing with result here and updating weights

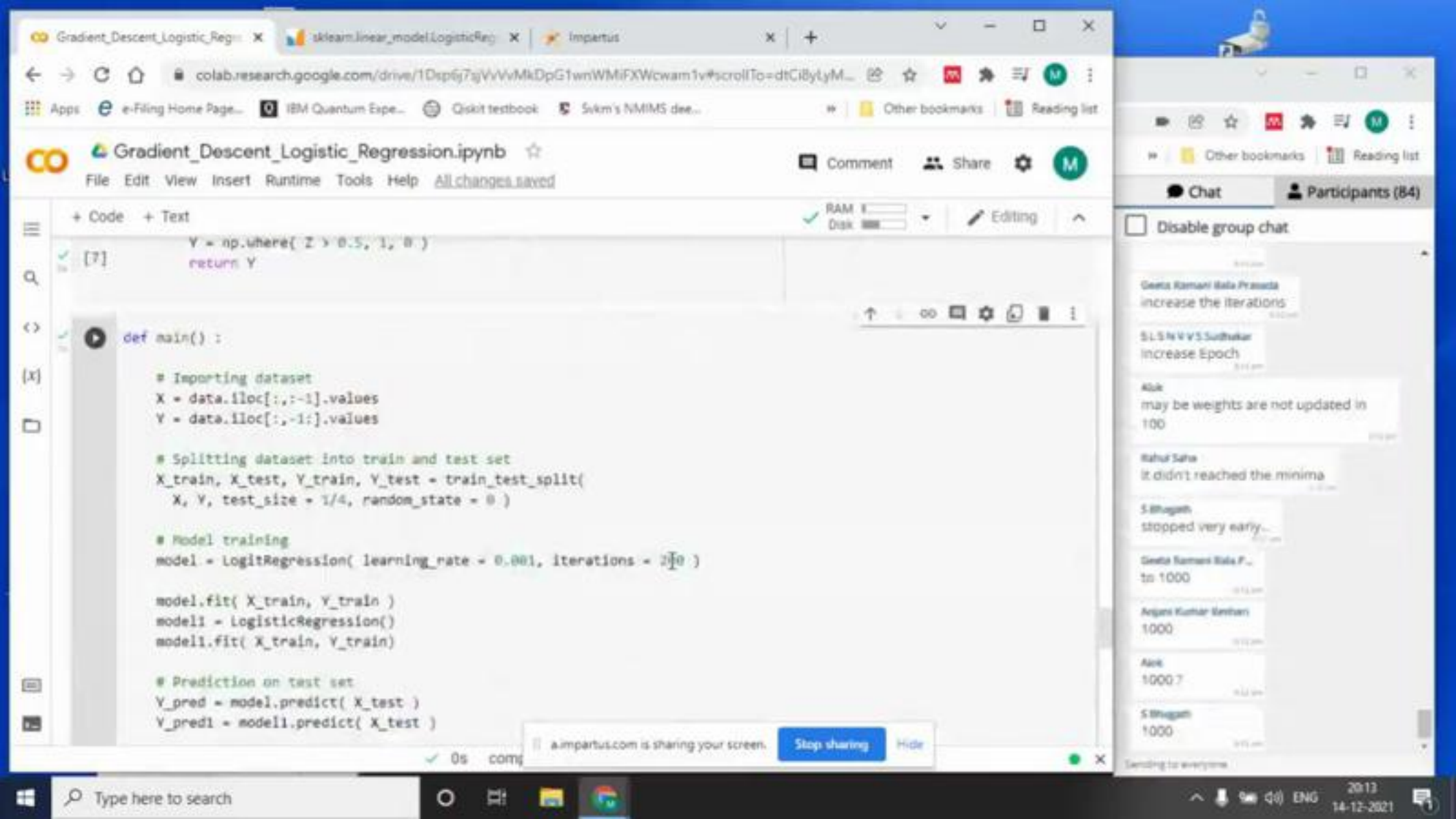
Debojyoti Chakraborty
on what params do we select the right solver

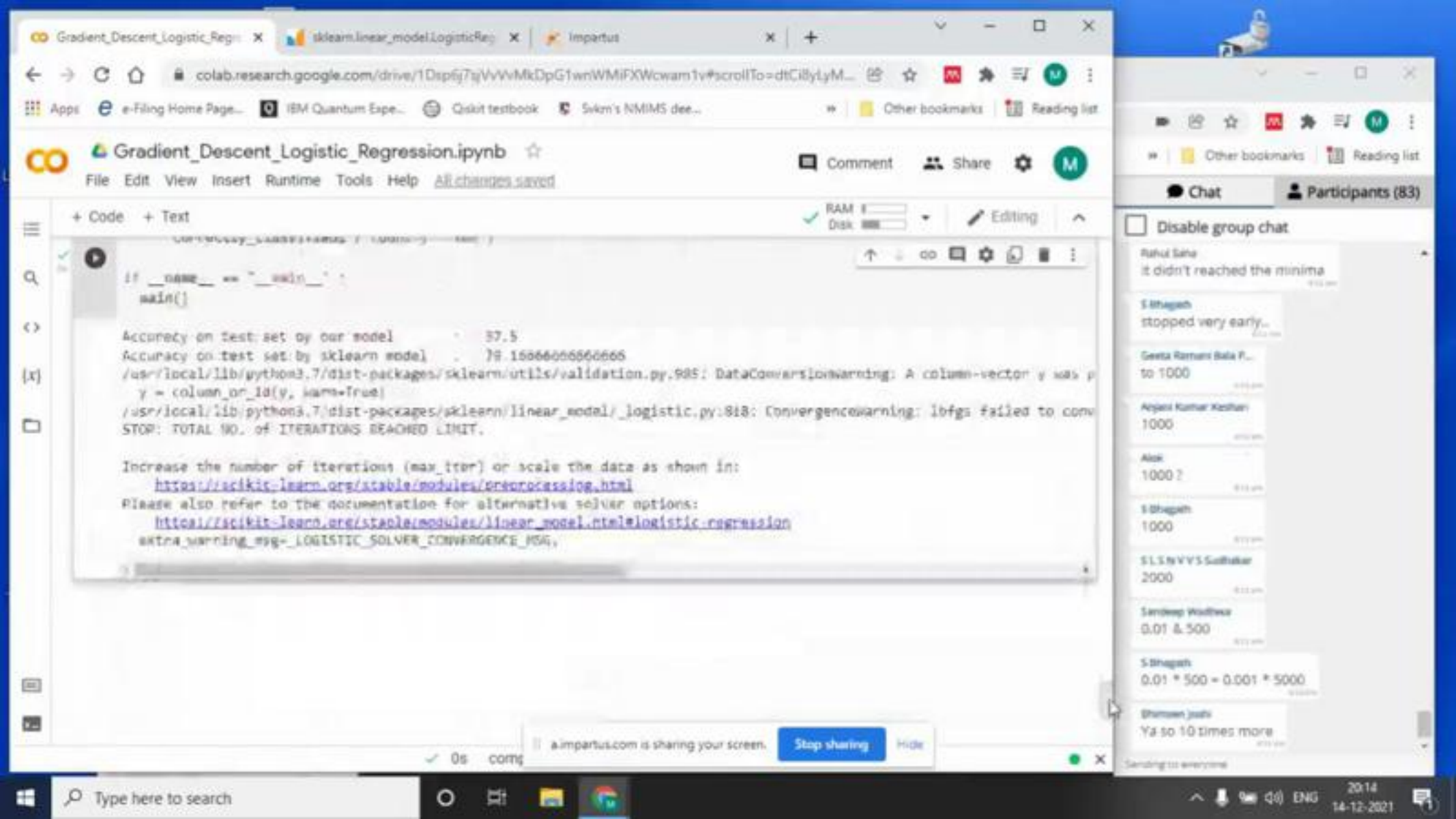
sklearn solver

BOUDHAYAN CHAKRABORTY joined

Type here to search

ENG 20:11 14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[7] Y = np.where(Z > 0.5, 1, 0)
return Y
def main() :
# Importing dataset
X = data.iloc[:,:-1].values
Y = data.iloc[:,-1:].values
# Splitting dataset into train and test set
X_train, X_test, Y_train, Y_test = train_test_split(
X, Y, test_size = 1/4, random_state = 0)
# Model training
model = LogitRegression(learning_rate = 0.001, iterations = 200)
model.fit(X_train, Y_train)
model1 = LogisticRegression()
model1.fit(X_train, Y_train)
# Prediction on test set
Y_pred = model.predict(X_test)
Y_pred1 = model1.predict(X_test)
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (84)
Disable group chat
increase the iterations
increase Epoch
may be weights are not updated in 100
It didn't reached the minima
stopped very early...
to 1000
1000
1000 ?
1000
Type here to search
ENG
20:13
14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
if __name__ == "__main__" :
main()
Accuracy on test set by our model : 97.5
Accuracy on test set by sklearn model : 79.16666666666666
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:985: DataConversionWarning: A column-vector y was p
y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to conv
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
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (83)
Disable group chat
Rahul Saha
it didn't reached the minima
S Bhagath
stopped very early..
Geeta Ramani Bala P...
to 1000
Anjani Kumar Keshari
1000
Alok
1000 ?
S Bhagath
1000
S L S N V V S Sudhakar
2000
Sandeep Wadhwa
0.01 & 500
S Bhagath
0.01 * 500 = 0.001 * 5000
Bhimsen Joshi
Ya so 10 times more
Type here to search
ENG
20:14
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

Editing

Now we will be using the above derived equation to make our predictions. Before that we will train our model to obtain the values of our parameters $b_0, b_1, b_2 \ldots$ that result in least error. This is where the error or loss function comes in.

Loss Function

Loss functions are used to calculate error in a model. There are two types of Loss functions. L1 and L2.
L1 loss function is used to minimize the error which is the sum of the all the absolute differences between the true value and the predicted value.
L2 Loss Function L2 Loss Function is used to minimize the error which is the sum of the all the squared differences between the true value and the predicted value

The cost function of linear regression L1 and L2 can't be used in logistic regression because it is a non-convex function of weights. Optimizing algorithms like i e gradient descent only converge convex function into a global minimum.

J = - ylog(h(x)) - (1 - y)log(1 - h(x))

here, y is the real target value

h(x) = sigmoid(wx + b)

For y = 0,

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (82)

Disable group chat

Bhimsen Joshi
Ya so 10 times more

Seelam Reshma Mon...
5000 ?

Anjani Kumar Keshari
lr = 0.005

Alok
Loop thru

Gaurav Anand
I remember we used Tou for learning rate

Anjani Kumar Keshari
hypertuning

M R Vijay Krishnan
Gridsearch

Random searchok

Alok
0.01 + 500

Type here to search

ENG 20:16 14-12-2021

The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`.

Note that these weights will be multiplied with sample_weight (passed through the fit method) if sample_weight is specified.

New in version 0.17: class_weight='balanced'

random_state : *int, RandomState instance, default=None*

Used when `solver == 'sag'` 'saga' or 'liblinear' to shuffle the data. See Glossary for details.

solver : *{'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}, default='lbfgs'*

Algorithm to use in the optimization problem. Default is 'lbfgs'. To choose a solver, you might want to consider the following aspects:

- For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones;
- For multiclass problems, only 'newton-cg', 'sag', 'saga' and 'lbfgs' handle multinomial loss;
- 'liblinear' is limited to one-versus-rest schemes.

Warning: The choice of the algorithm depends on the penalty chosen:

Toggle Menu

a.impartus.com is sharing your screen. Stop sharing Hide

Chat | Participants (82)

Disable group chat

Anjani Kumar Kashari: lr = 0.005

Alok: Loop thru

Gaurav Anand: I remember we used Tou for learning rate

Anjani Kumar Kashari: hypertuning

M B Vijay Krishnan: Gridsearch

Random searchok

Alok: 0.01 * 500

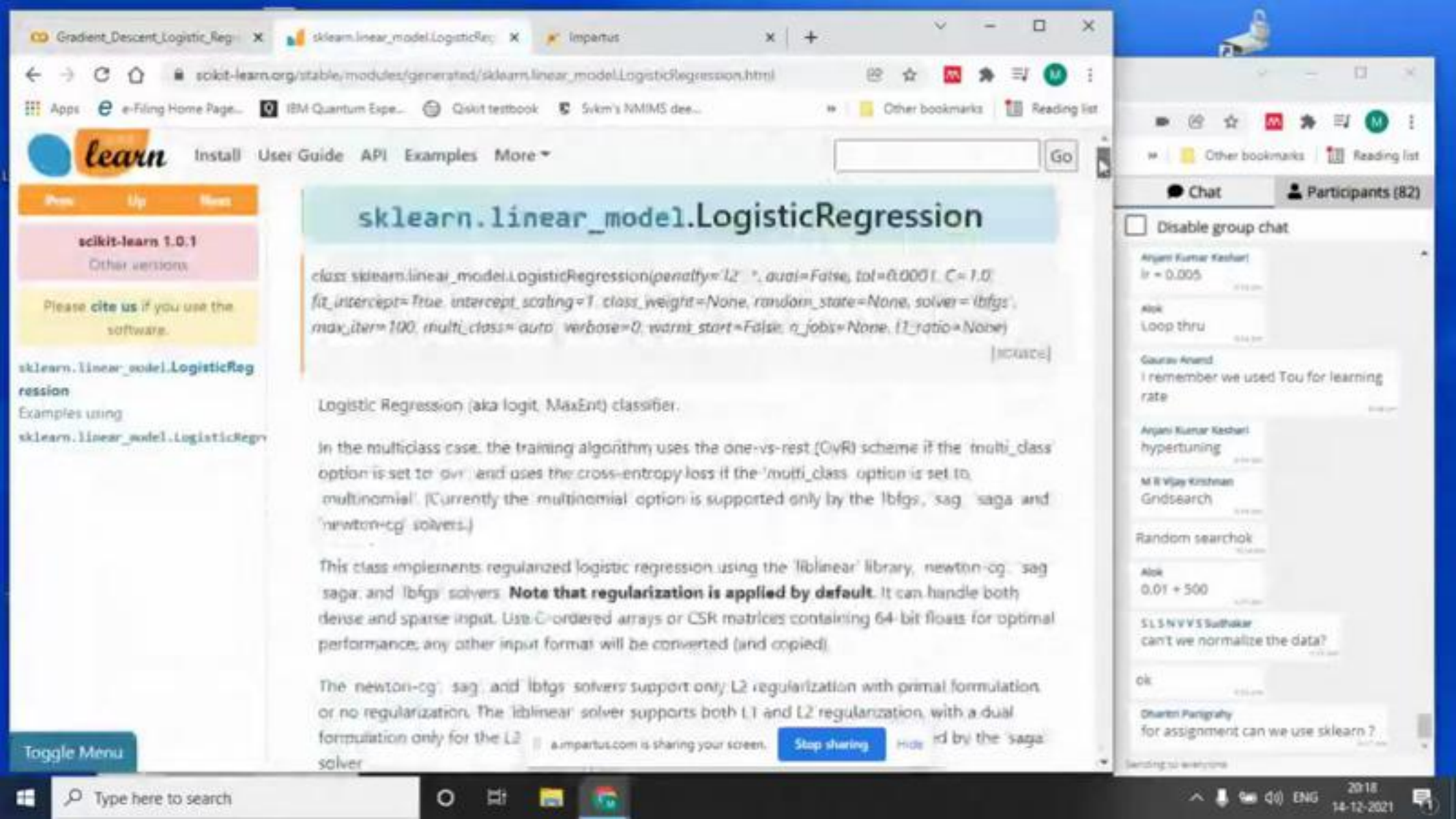
S L S N V V S Sudhakar: can't we normalize the data?

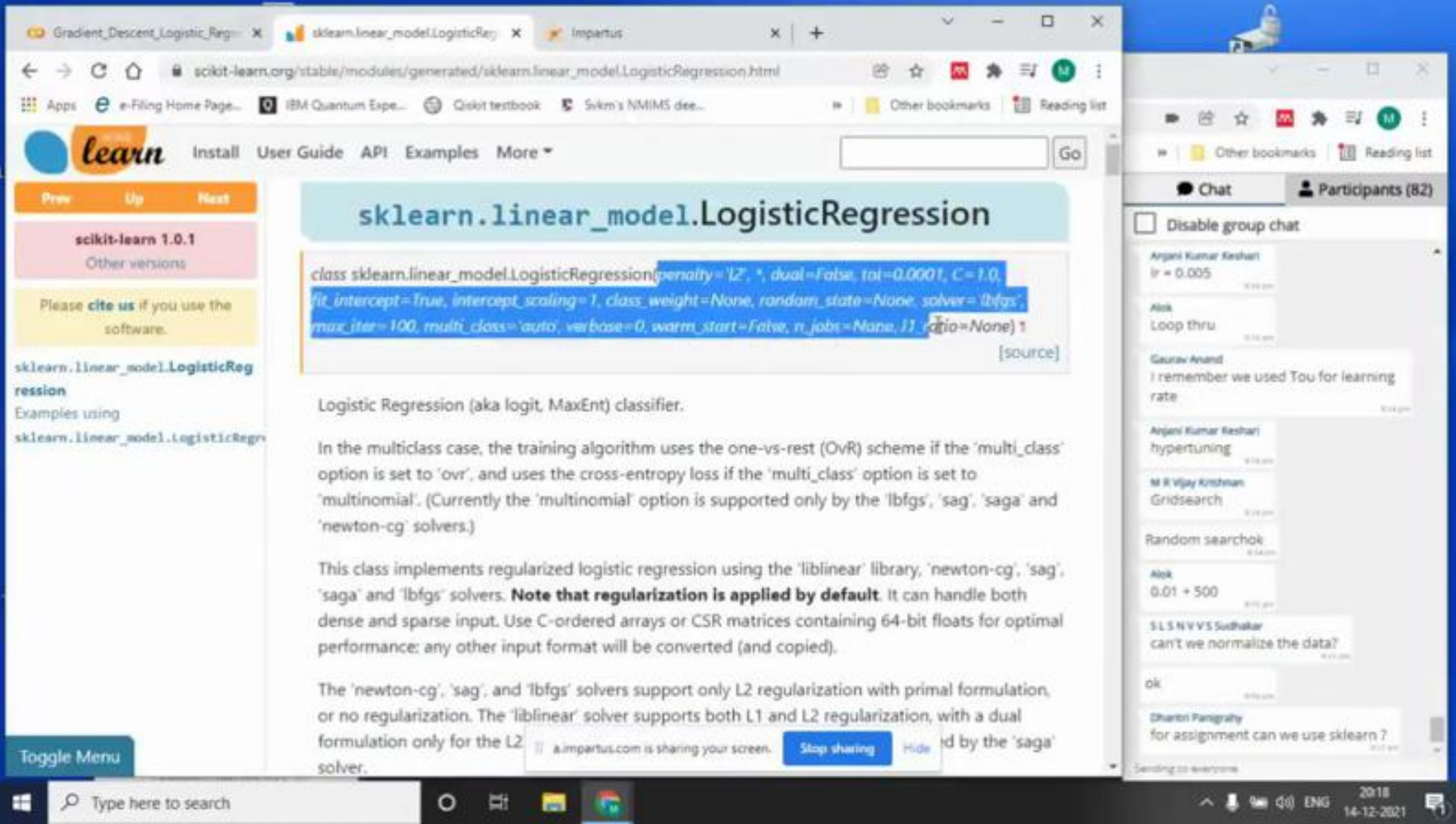
ok

Dharitri Panigrahy: for assignment can we use sklearn ?

Type here to search

ENG 20:18 14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticRe
Impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Install
User Guide
API
Examples
More
Go
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
sklearn.linear_model.LogisticRegression
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
[source]
Logistic Regression (aka logit, MaxEnt) classifier.
In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)
This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. Note that regularization is applied by default. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).
The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2
d by the 'saga'
solver.
Toggle Menu
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (82)
Disable group chat
Anjani Kumar Kashari
lr = 0.005
Alok
Loop thru
Gaurav Anand
I remember we used Tou for learning rate
Anjani Kumar Kashari
hypertuning
M R Vijay Krishnan
Gridsearch
Random searchok
Alok
0.01 * 500
S L S N V V S Sudhakar
can't we normalize the data?
ok
Dharitri Panigrahy
for assignment can we use sklearn ?
Sending to everyone
Type here to search
ENG
20:18
14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticRe
Impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Install
User Guide
API
Examples
More
Go
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
sklearn.linear_model.LogisticRegression
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
[source]
Logistic Regression (aka logit, MaxEnt) classifier.
In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)
This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. Note that regularization is applied by default. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).
The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2
d by the 'saga' solver.
a.impartus.com is sharing your screen.
Stop sharing
Hide
Toggle Menu
Other bookmarks
Reading list
Chat
Participants (82)
Disable group chat
Anjani Kumar Keshari
lr = 0.005
Alok
Loop thru
Gaurav Anand
I remember we used Tou for learning rate
Anjani Kumar Keshari
hypertuning
M R Vijay Krishnan
Gridsearch
Random searchok
Alok
0.01 * 500
S L S N V V S Sudhakar
can't we normalize the data?
ok
Dharitri Panigrahy
for assignment can we use sklearn ?
Type here to search
ENG
20:18
14-12-2021

Gradient_Descent_Logistic_Regr... | sklearn.linear_model.LogisticReg... | Impartus

scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps | e-Filing Home Page... | IBM Quantum Expe... | Qiskit textbook | Svkm's NMIMS dee... | Other bookmarks | Reading list

learn Install User Guide API Examples More Go

Prev Up Next

scikit-learn 1.0.1
Other versions

Please cite us if you use the software.

sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr

sklearn.linear_model.LogisticRegression

class sklearn.linear_model.LogisticRegression(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

Logistic Regression (aka logit, MaxEnt) classifier.

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)

This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. **Note that regularization is applied by default**. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2 ...d by the 'saga' solver.

Toggle Menu

a.impartus.com is sharing your screen. Stop sharing Hide

Chat | Participants (82)

Disable group chat

I remember we used Tou for learning rate

Anjani Kumar Kashari: hypertuning

M R Vijay Krishnan: Gridsearch

Random searchok

Alok: 0.01 + 500

S L S N V V S Sudhakar: can't we normalize the data?

ok

Dharani Panigrahy: for assignment can we use sklearn ?

Bhanwari Joshi: is 72 max we can get?

M R Vijay Krishnan: Can we use gradient decent for multiple linear regression as well?

Sending to everyone

Type here to search

ENG 20:18 14-12-2021

Gradient_Descent_Logistic_Regr | sklearn.linear_model.LogisticReg | Impartus

scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps | e-Filing Home Page... | IBM Quantum Expe... | Qiskit textbook | Svkm's NMIMS dee... | Other bookmarks | Reading list

learn | Install | User Guide | API | Examples | More | Go

Prev | Up | Next

scikit-learn 1.0.1
Other versions

Please cite us if you use the software.

sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr

sklearn.linear_model.LogisticRegression

class sklearn.linear_model.LogisticRegression(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

Logistic Regression (aka logit, MaxEnt) classifier.

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)

This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. **Note that regularization is applied by default**. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. The Elastic-Net regularization is only supported by the 'saga' solver.

Toggle Menu

a.impartus.com is sharing your screen. Stop sharing | Hide

Chat | Participants (82)

Disable group chat

Random searchok

Alok
0.01 * 500

S L S N V V S Sudhakar
can't we normalize the data?

Ok

Dharini Pangrahy
for assignment can we use sklearn ?

Bhavsen Joshi
is 72 max we can get?

M R Vijay Krishnan
Can we use gradient decent for multiple linear regression as well?

Alok
L1 and L2 are not covered

M R Vijay Krishnan
Ok

Mulasa Lakshmi Sow...
Lasso and Ridge

Sending to everyone

Type here to search

ENG 20:18 14-12-2021

Gradient_Descent_Logistic_Regr | sklearn.linear_model.LogisticReg | Impartus

scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps | e-Filing Home Page... | IBM Quantum Expe... | Qiskit textbook | Svkm's NMIMS dee... | Other bookmarks | Reading list

learn | Install | User Guide | API | Examples | More | Go

Prev | Up | Next

scikit-learn 1.0.1
Other versions

Please **cite us** if you use the software.

sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr

sklearn.linear_model.LogisticRegression

class sklearn.linear_model.LogisticRegression(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

Logistic Regression (aka logit, MaxEnt) classifier.

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)

This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. **Note that regularization is applied by default**. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. The Elastic-Net regularization is only supported by the 'saga' solver.

Toggle Menu

a.impartus.com is sharing your screen. Stop sharing Hide

Chat | Participants (83)

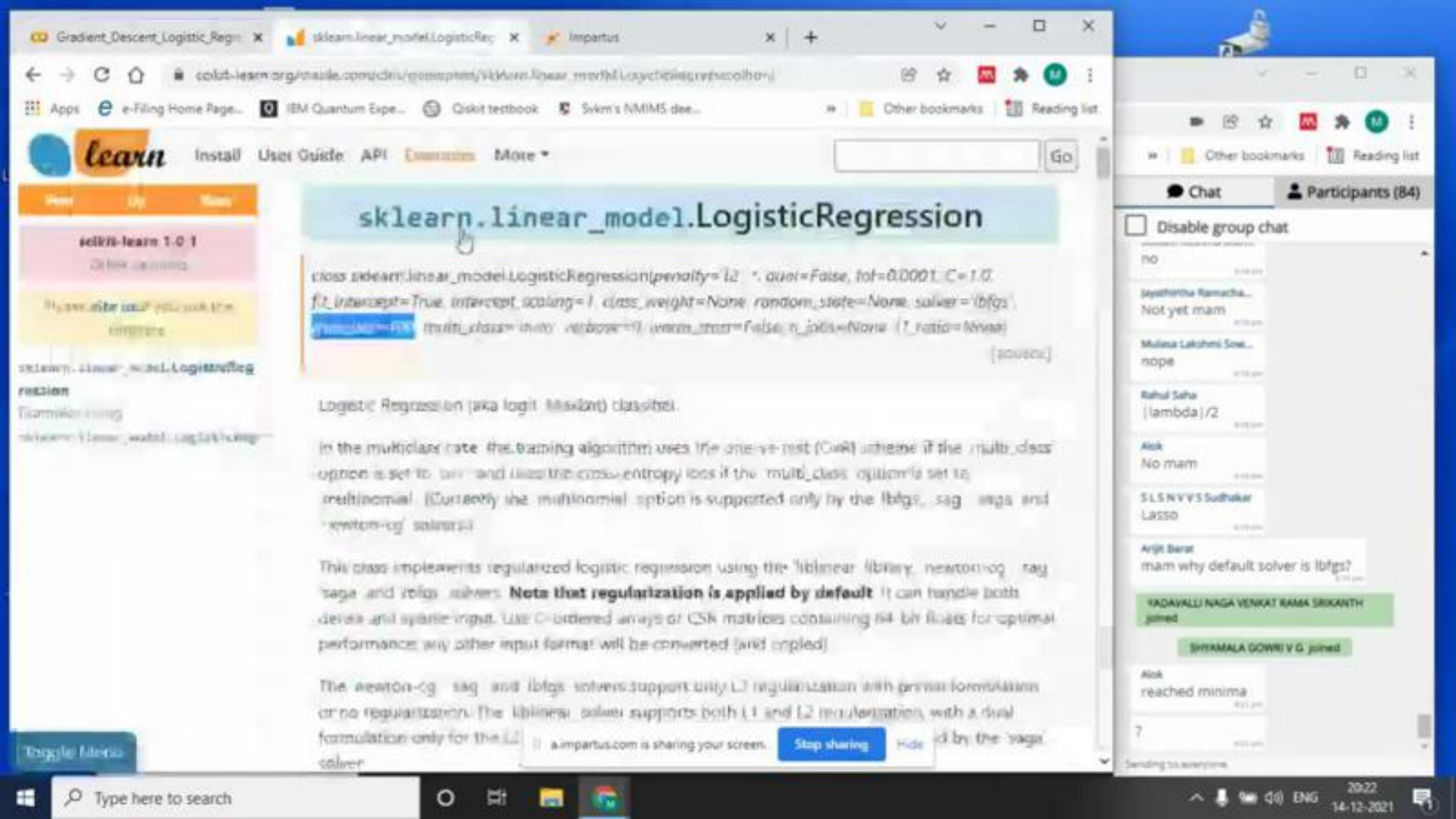
Disable group chat

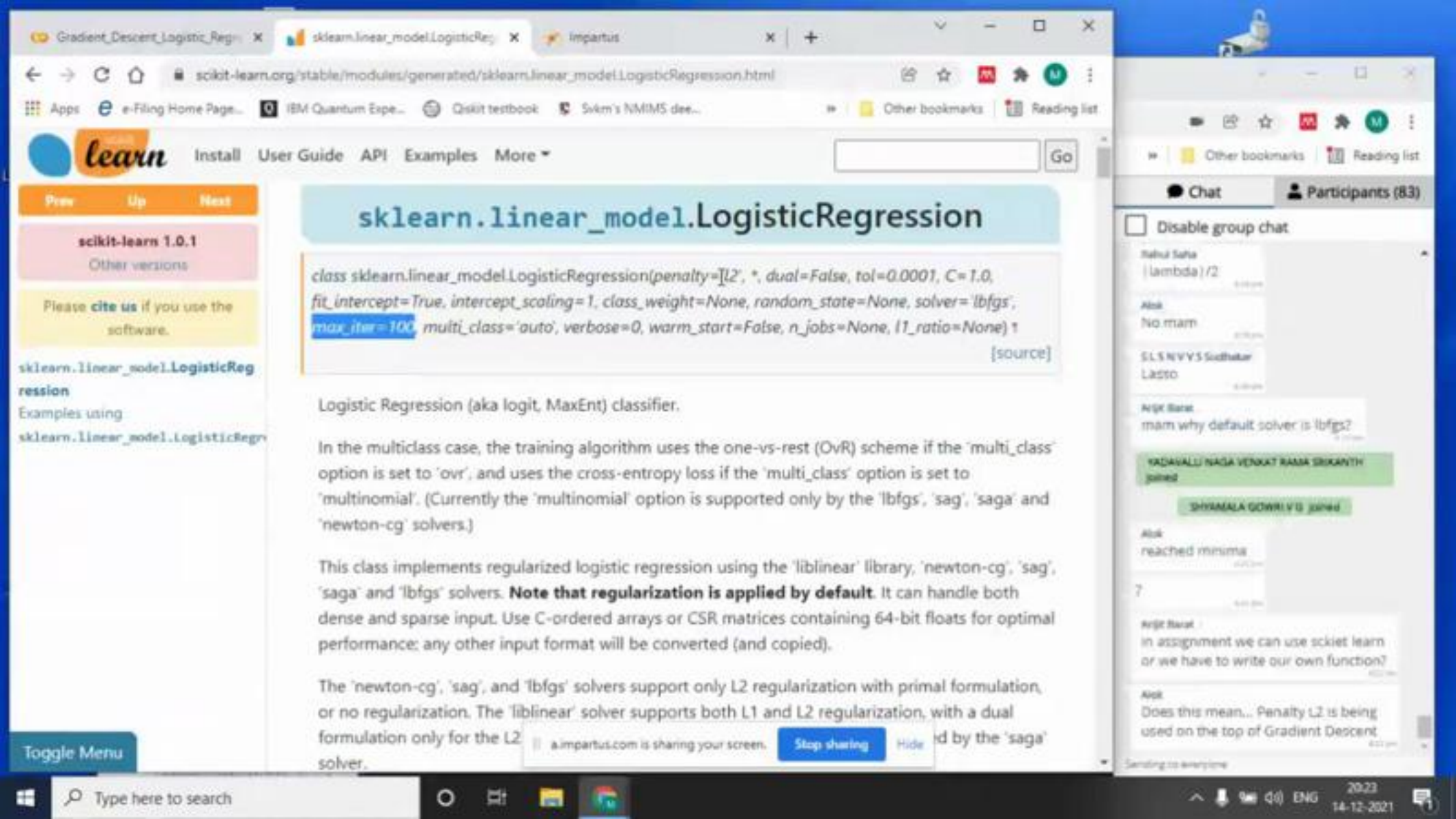
Alok: L1 and L2 are not covered
M R Vijay Krishnan: Ok
Mulasa Lakshmi Sow...: Lasso and Ridge
S L S N V V S Sudhakar: Those are
Arijit Barat: not covered
Seelam Reshma Mon...: no
Jayathirtha Ramacha...: Not yet mam
Mulasa Lakshmi Sow...: nope
Rahul Saha: |lambda|/2
Alok: No mam

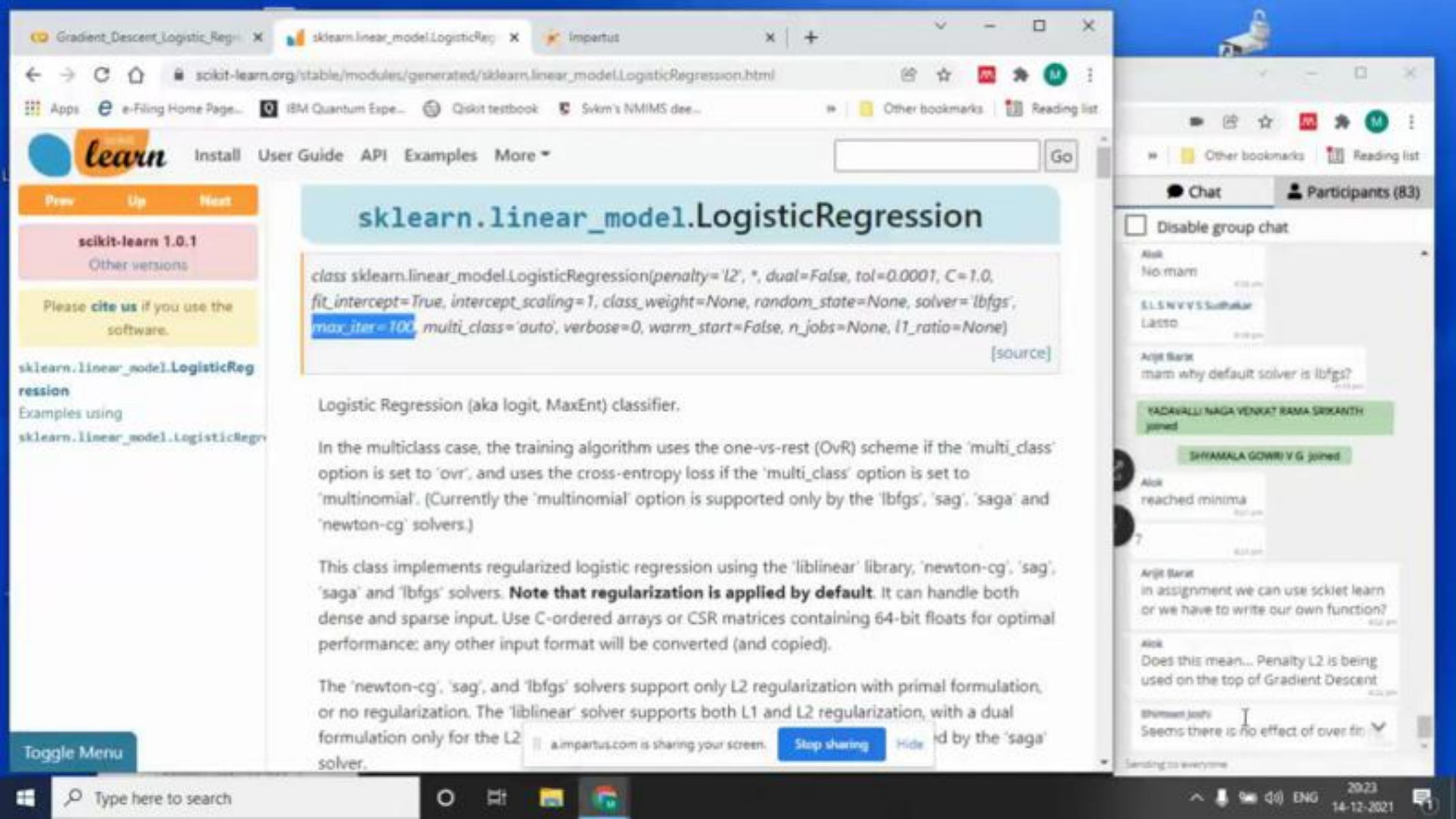
Sending to everyone

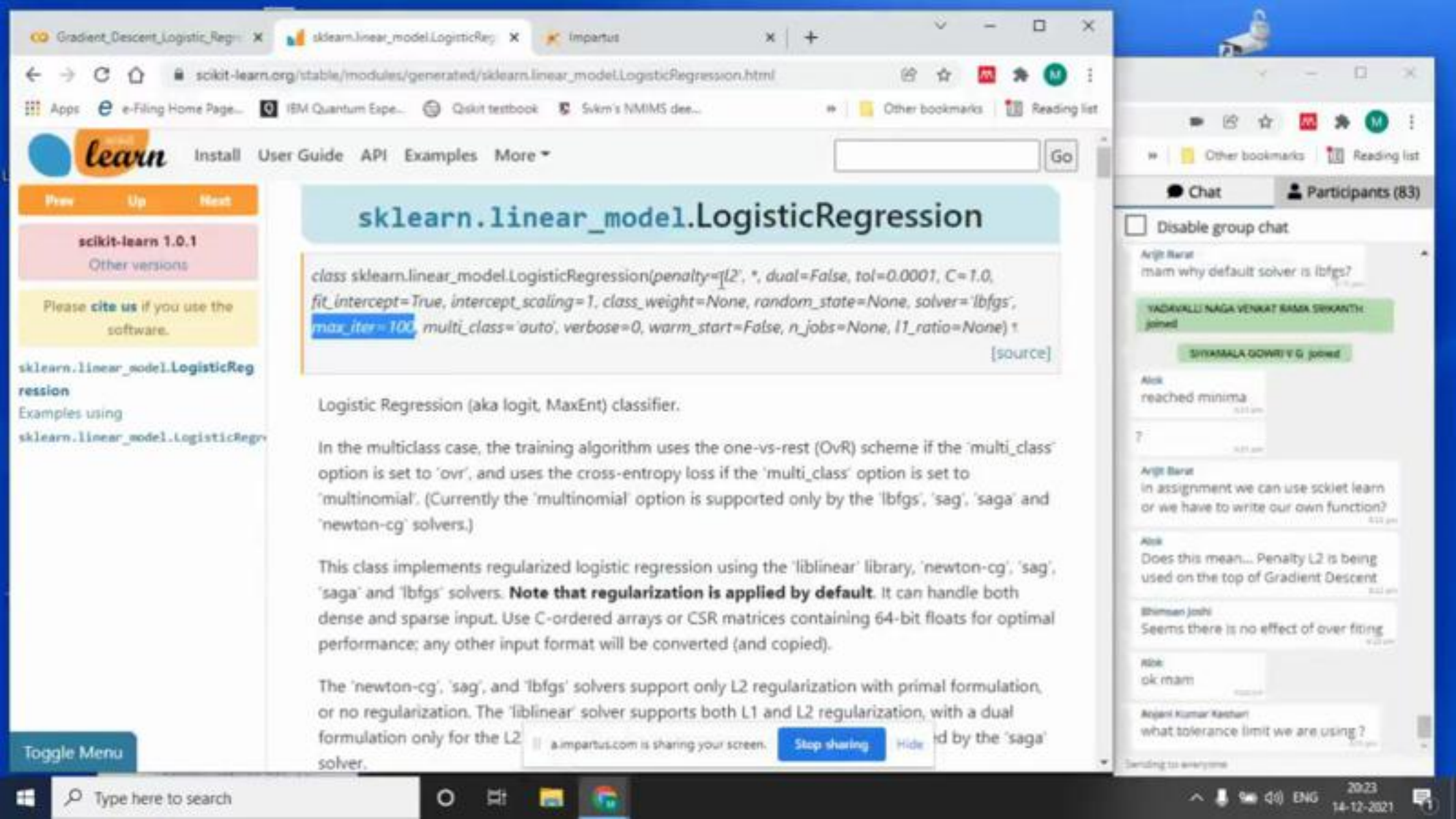
Type here to search

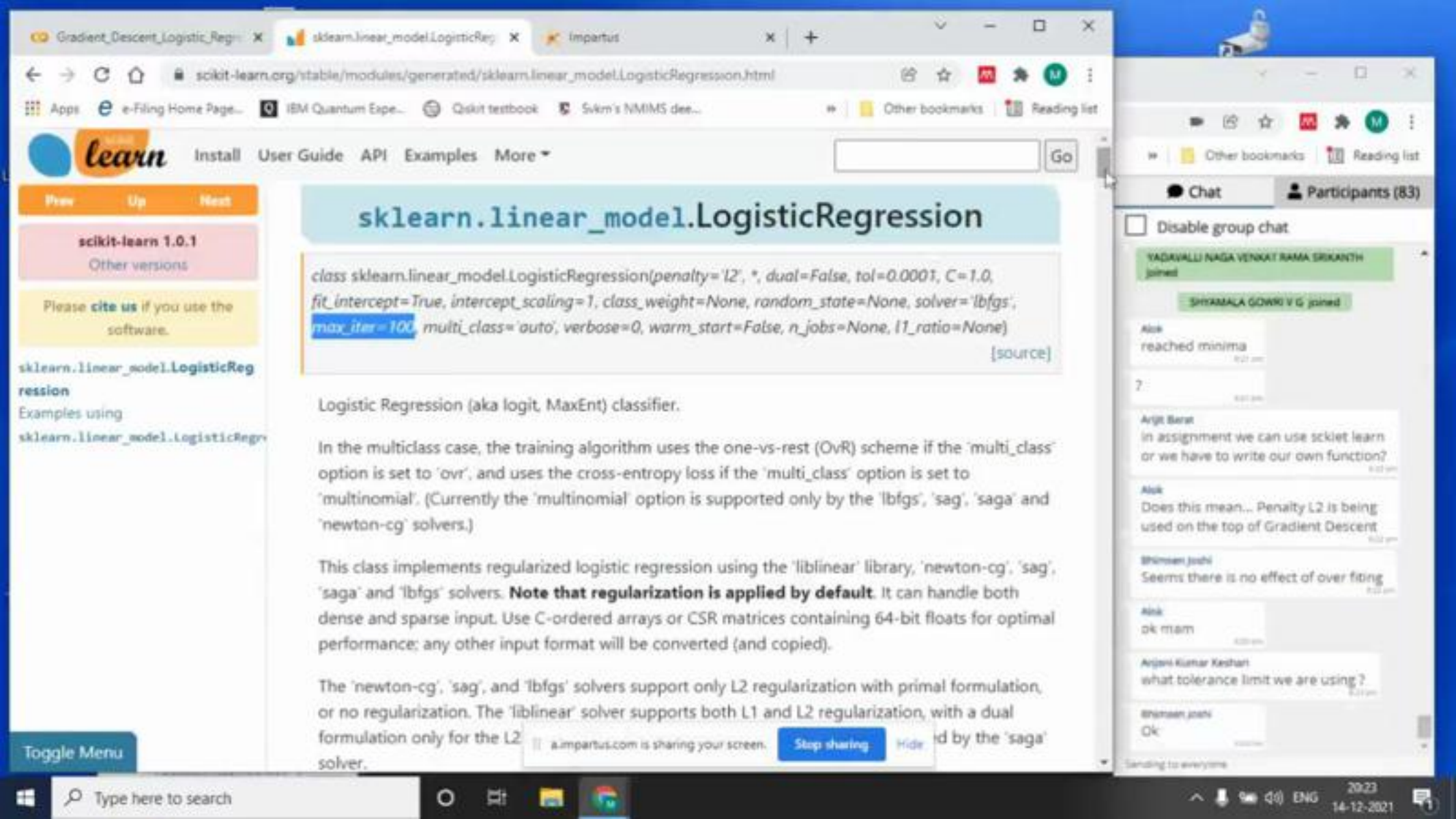
ENG 20:19 14-12-2021


sklearn.linear_model.LogisticRegression
Install
User Guide
API
More
Go
Chat
Participants (84)
Disable group chat
Not yet mam
nope
(lambda)/2
No mam
Lasso
mam why default solver is lbfgs?
reached minima
a.impartus.com is sharing your screen.
Stop sharing
Hide
Type here to search
ENG
20:22
14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticRe
Impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Install
User Guide
API
Examples
More
Go
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
sklearn.linear_model.LogisticRegression
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
[source]
Logistic Regression (aka logit, MaxEnt) classifier.
In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)
This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. Note that regularization is applied by default. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).
The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2
d by the 'saga'
solver.
a.impartus.com is sharing your screen.
Stop sharing
Hide
Toggle Menu
Other bookmarks
Reading list
Chat
Participants (83)
Disable group chat
Rahul Saha
(lambda)/2
Alok
No mam
S L S N V Y S Siddhakar
Lasso
Arijit Barat
mam why default solver is lbfgs?
YADAVALLI NAGA VENKAT RAMA SRIKANTH joined
SHYAMALA GOWRI V G joined
Alok
reached minima
?
Arijit Barat
in assignment we can use scikit learn or we have to write our own function?
Alok
Does this mean... Penalty L2 is being used on the top of Gradient Descent
Sending to everyone
Type here to search
ENG
20:23
14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticReg
Impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Install
User Guide
API
Examples
More
Go
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
sklearn.linear_model.LogisticRegression
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
[source]
Logistic Regression (aka logit, MaxEnt) classifier.
In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)
This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. Note that regularization is applied by default. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).
The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2
d by the 'saga'
solver.
a.impartus.com is sharing your screen.
Stop sharing
Hide
Toggle Menu
Other bookmarks
Reading list
Chat
Participants (83)
Disable group chat
Alok
No mam
S L S N V V S Sudhakar
Lasso
Arijit Barat
mam why default solver is lbfgs?
YADAVALLI NAGA VENKAT RAMA SRIKANTH joined
SHYAMALA GOWRI V G joined
Alok
reached minima
?
Arijit Barat
in assignment we can use sciket learn or we have to write our own function?
Alok
Does this mean... Penalty L2 is being used on the top of Gradient Descent
Bhimsen Joshi
Seems there is no effect of over fit
Sending to everyone
Type here to search
ENG
20:23
14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticRe
Impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Install
User Guide
API
Examples
More
Go
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
sklearn.linear_model.LogisticRegression
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
[source]
Logistic Regression (aka logit, MaxEnt) classifier.
In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)
This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. Note that regularization is applied by default. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).
The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2
d by the 'saga'
solver.
a.impartus.com is sharing your screen.
Stop sharing
Hide
Toggle Menu
Other bookmarks
Reading list
Chat
Participants (83)
Disable group chat
Arijit Barat
mam why default solver is lbfgs?
YADAVALLI NAGA VENKAT RAMA SRIKANTH joined
SIVAMALA GOWRI V G joined
Alok
reached minima
?
Arijit Barat
in assignment we can use sckiet learn or we have to write our own function?
Alok
Does this mean... Penalty L2 is being used on the top of Gradient Descent
Bhimsen Joshi
Seems there is no effect of over fitting
Alok
ok mam
Anjani Kumar Keshari
what tolerance limit we are using ?
Sending to everyone
Type here to search
ENG
20:23
14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticRe
Impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Install
User Guide
API
Examples
More
Go
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
sklearn.linear_model.LogisticRegression
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
[source]
Logistic Regression (aka logit, MaxEnt) classifier.
In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)
This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. Note that regularization is applied by default. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).
The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2
d by the 'saga' solver.
a.impartus.com is sharing your screen.
Stop sharing
Hide
Toggle Menu
Other bookmarks
Reading list
Chat
Participants (83)
Disable group chat
YADAVALLI NAGA VENKAT RAMA SRIKANTH joined
SHYAMALA GOWRI V G joined
Alok
reached minima
?
Arijit Barat
in assignment we can use scklet learn or we have to write our own function?
Alok
Does this mean... Penalty L2 is being used on the top of Gradient Descent
Bhimsen Joshi
Seems there is no effect of over fitting
Alok
ok mam
Anjani Kumar Keshari
what tolerance limit we are using?
Bhimsen Joshi
Ok
Sending to everyone
Type here to search
ENG
20:23
14-12-2021

Gradient_Descent_Logistic_Regr | sklearn.linear_model.LogisticRe | Impartus

scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps | e-Filing Home Page... | IBM Quantum Expe... | Qiskit textbook | Svkm's NMIMS dee... | Other bookmarks | Reading list

learn

Prev | Up | Next

scikit-learn 1.0.1
Other versions

Please cite us if you use the software

sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr

[source]

Logistic Regression (aka logit, MaxEnt) classifier.

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)

This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. **Note that regularization is applied by default**. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. The Elastic-Net regularization is only supported by the 'saga' solver.

Read more in the User Guide.

Parameters: **penalty : {'l1', 'l2', 'elasticnet', 'none'}, default='l2'**

Specify the norm of the penalty:

- 'none': no penalty is added
- 'l2': add a L2 penalty term and it is the default choice;

a.impartus.com is sharing your screen. Stop sharing | Hide

Toggle Menu

Other bookmarks | Reading list

Chat | Participants (83)

Disable group chat

YADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok: reached minima

?

Arijit Barat: in assignment we can use scklet learn or we have to write our own function?

Alok: Does this mean... Penalty L2 is being used on the top of Gradient Descent

Bhimsen Joshi: Seems there is no effect of over fitting

Alok: ok mam

Anjani Kumar Keshari: what tolerance limit we are using ?

Bhimsen Joshi: Ok

Sending to everyone

Type here to search

ENG 20:23 14-12-2021

dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. The Elastic-Net regularization is only supported by the 'saga' solver.

Read more in the User Guide.

Parameters:

penalty : *{'l1', 'l2', 'elasticnet', 'none'}, default='l2'*

Specify the norm of the penalty:

- `'none'`: no penalty is added;
- `'l2'`: add a L2 penalty term and it is the default choice;
- `'l1'`: add a L1 penalty term;
- `'elasticnet'`: both L1 and L2 penalty terms are added.

Warning: Some penalties may not work with some solvers. See the parameter `solver` below, to know the compatibility between the penalty and solver.

New in version 0.19: l1 penalty with SAGA solver (allowing 'multinomial' + L1)

dual : *bool, default=False*

Dual or primal formulation. Dual formulation is only implemented for l2 penalty with liblinear solver. Preferred when n_samples > n_features.

performance: any other input format will be converted (and copied).

The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. The Elastic-Net regularization is only supported by the 'saga' solver.

Read more in the User Guide.

Parameters:

penalty : *{'l1', 'l2', 'elasticnet', 'none'}, default='l2'*

Specify the norm of the penalty:

- `'none'`: no penalty is added;
- `'l2'`: add a L2 penalty term and it is the default choice;
- `'l1'`: add a L1 penalty term;
- `'elasticnet'`: both L1 and L2 penalty terms are added.

> **Warning:** Some penalties may not work with some solvers. See the parameter `solver` below, to know the compatibility between the penalty and solver.

New in version 0.19: l1 penalty with SAGA solver (allowing 'multinomial' + L1)

dual : *bool, default=False*

Dual or primal formulation. Dual formulation is only implemented for l2 penalty

a.impartus.com is sharing your screen. Stop sharing Hide

s > n_features.

Toggle Menu

Chat Participants (83)

Disable group chat

YADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok: reached minima

?

Arijit Barai: in assignment we can use scklet learn or we have to write our own function?

Alok: Does this mean... Penalty L2 is being used on the top of Gradient Descent

Bhimsen Joshi: Seems there is no effect of over fitting

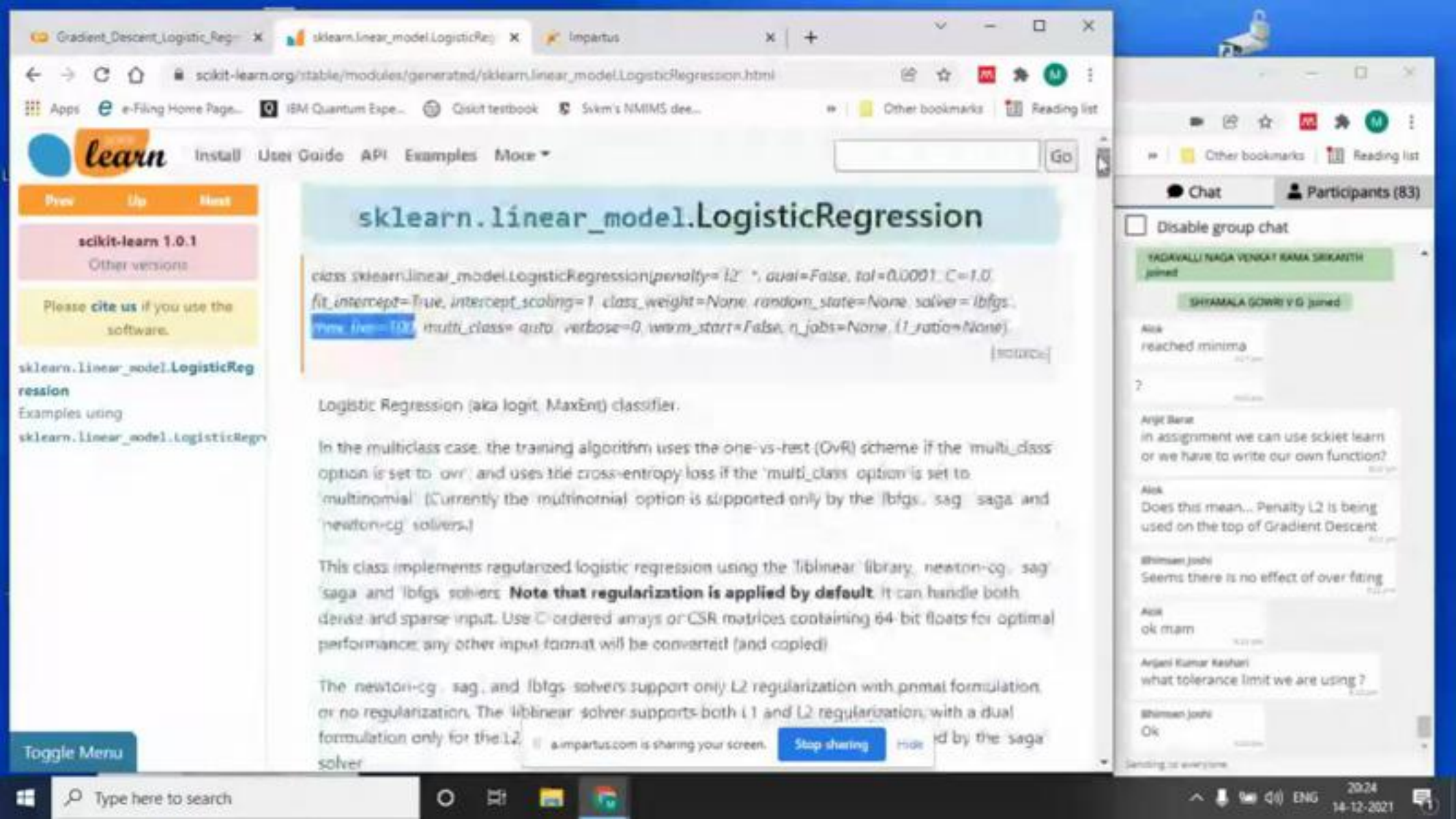
Alok: ok mam

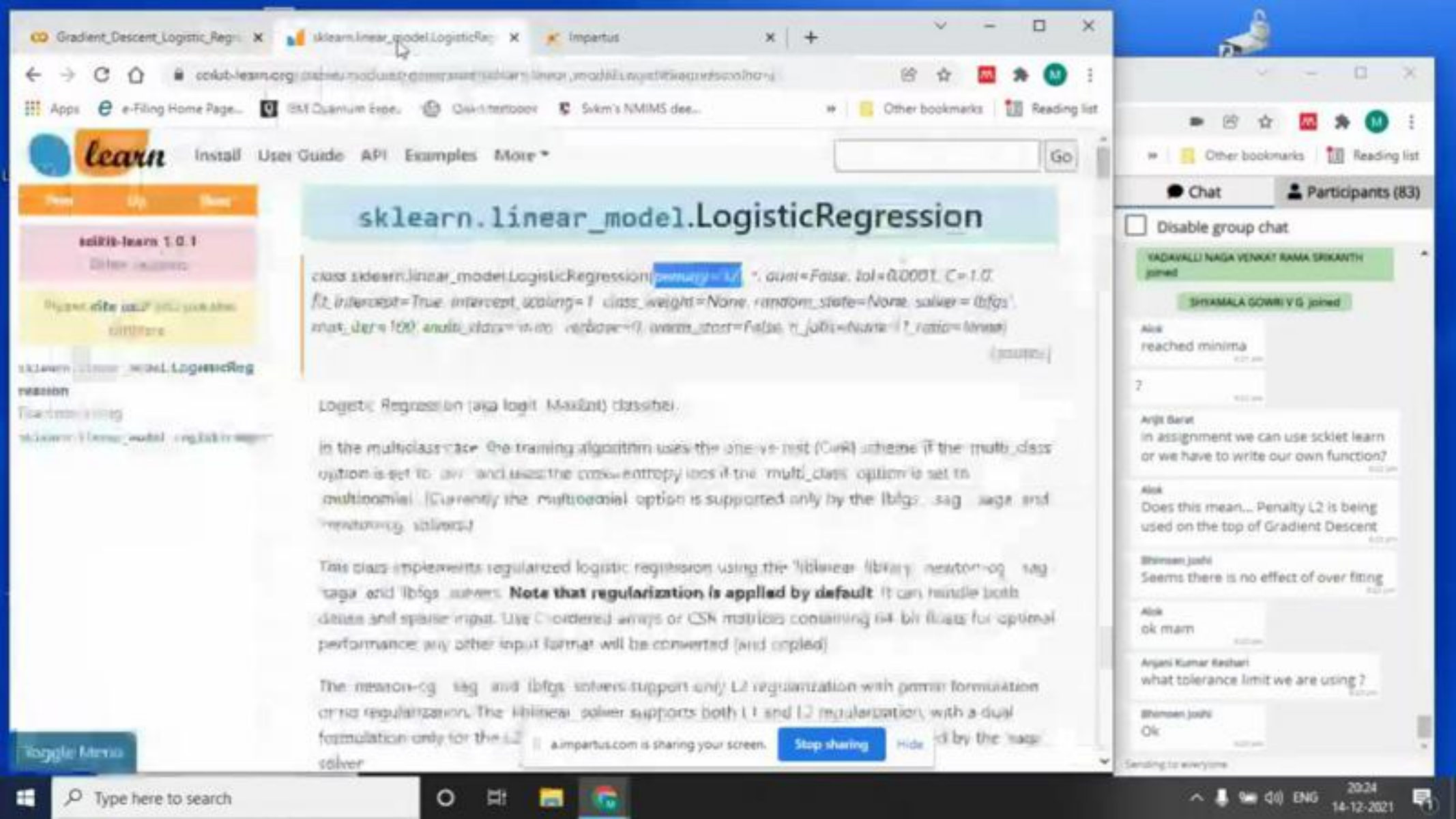
Anjani Kumar Keshari: what tolerance limit we are using ?

Bhimsen Joshi: Ok

Type here to search

ENG 20:24 14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticRe
Impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Install
User Guide
API
Examples
More
Go
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
sklearn.linear_model.LogisticRegression
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
[source]
Logistic Regression (aka logit, MaxEnt) classifier.
In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)
This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. Note that regularization is applied by default. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).
The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2
d by the 'saga' solver.
Toggle Menu
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (83)
Disable group chat
YADAVALLI NAGA VENKAT RAMA SRIKANTH joined
SHYAMALA GOWRI V G joined
Alok
reached minima
?
Arijit Barai
in assignment we can use sckiet learn or we have to write our own function?
Alok
Does this mean... Penalty L2 is being used on the top of Gradient Descent
Bhimsen Joshi
Seems there is no effect of over fiting
Alok
ok mam
Arijani Kumar Keshari
what tolerance limit we are using ?
Bhimsen Joshi
Ok
Type here to search
ENG
20:24
14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticRe
Impartus
Apps
e-Filing Home Page...
Other bookmarks
Reading list
learn
Install
User Guide
API
Examples
More
Go
sklearn.linear_model.LogisticRegression
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
Logistic Regression (aka logit, MaxEnt) classifier.
In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the multi_class option is set to 'ovr', and uses the cross-entropy loss if the multi_class option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs', 'sag', 'saga' and 'newton-cg' solvers.)
This class implements regularized logistic regression using the 'liblinear' library, 'newton-cg', 'sag', 'saga' and 'lbfgs' solvers. Note that regularization is applied by default. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).
The 'newton-cg', 'sag', and 'lbfgs' solvers support only L2 regularization with primal formulation, or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. The Elastic-Net regularization is only supported by the 'saga' solver.
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (83)
Disable group chat
YADAVALLI NAGA VENKAT RAMA SRIKANTH joined
SHYAMALA GOWRI V G joined
Alok
reached minima
?
Arijit Barat
in assignment we can use scklet learn or we have to write our own function?
Alok
Does this mean... Penalty L2 is being used on the top of Gradient Descent
Bhimsen Joshi
Seems there is no effect of over fitting
Alok
ok mam
Anjani Kumar Keshari
what tolerance limit we are using ?
Bhimsen Joshi
Ok
Type here to search
ENG
20:24
14-12-2021

Gradient_Descent_Logistic_Regr | sklearn.linear_model.LogisticReg | Impartus

scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps | e-Filing Home Page... | IBM Quantum Expe... | Qiskit textbook | Svkm's NMIMS dee... | Other bookmarks | Reading list

learn

Prev Up Next

scikit-learn 1.0.1
Other versions

Please cite us if you use the software.

sklearn.linear_model.LogisticRegression

Examples using sklearn.linear_model.LogisticRegr

solver

Read more in the User Guide.

Parameters:

penalty : {'l1', 'l2', 'elasticnet', 'none'}, default='l2'

Specify the norm of the penalty:

- 'none': no penalty is added;
- 'l2': add a L2 penalty term and it is the default choice;
- 'l1': add a L1 penalty term;
- 'elasticnet': both L1 and L2 penalty terms are added.

Warning: Some penalties may not work with some solvers. See the parameter solver below, to know the compatibility between the penalty and solver.

New in version 0.19: l1 penalty with SAGA solver (allowing 'multinomial' + L1)

dual : bool, default=False

Dual or primal formulation. Dual formulation is only implemented for l2 penalty with liblinear solver. Prefer dual=False when n_samples > n_features.

tol : float, default=1e-4

Tolerance for stopping criteria.

C : float,

a.impartus.com is sharing your screen. Stop sharing Hide

Toggle Menu

Chat | Participants (83)

Disable group chat

YADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok: reached minima

?

Arijit Barai: in assignment we can use sciket learn or we have to write our own function?

Alok: Does this mean... Penalty L2 is being used on the top of Gradient Descent

Bhimsen Joshi: Seems there is no effect of over fiting

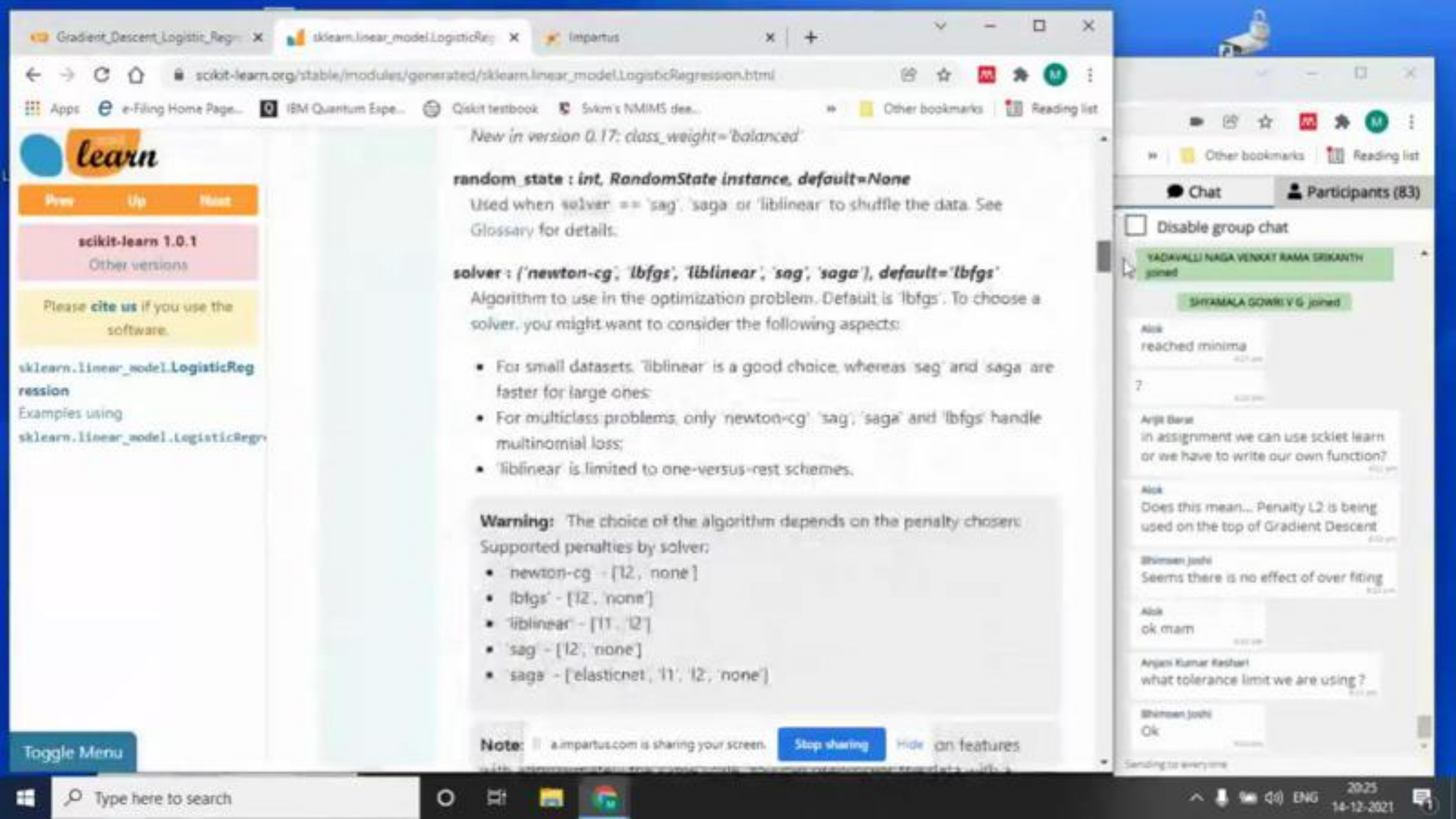
Alok: ok mam

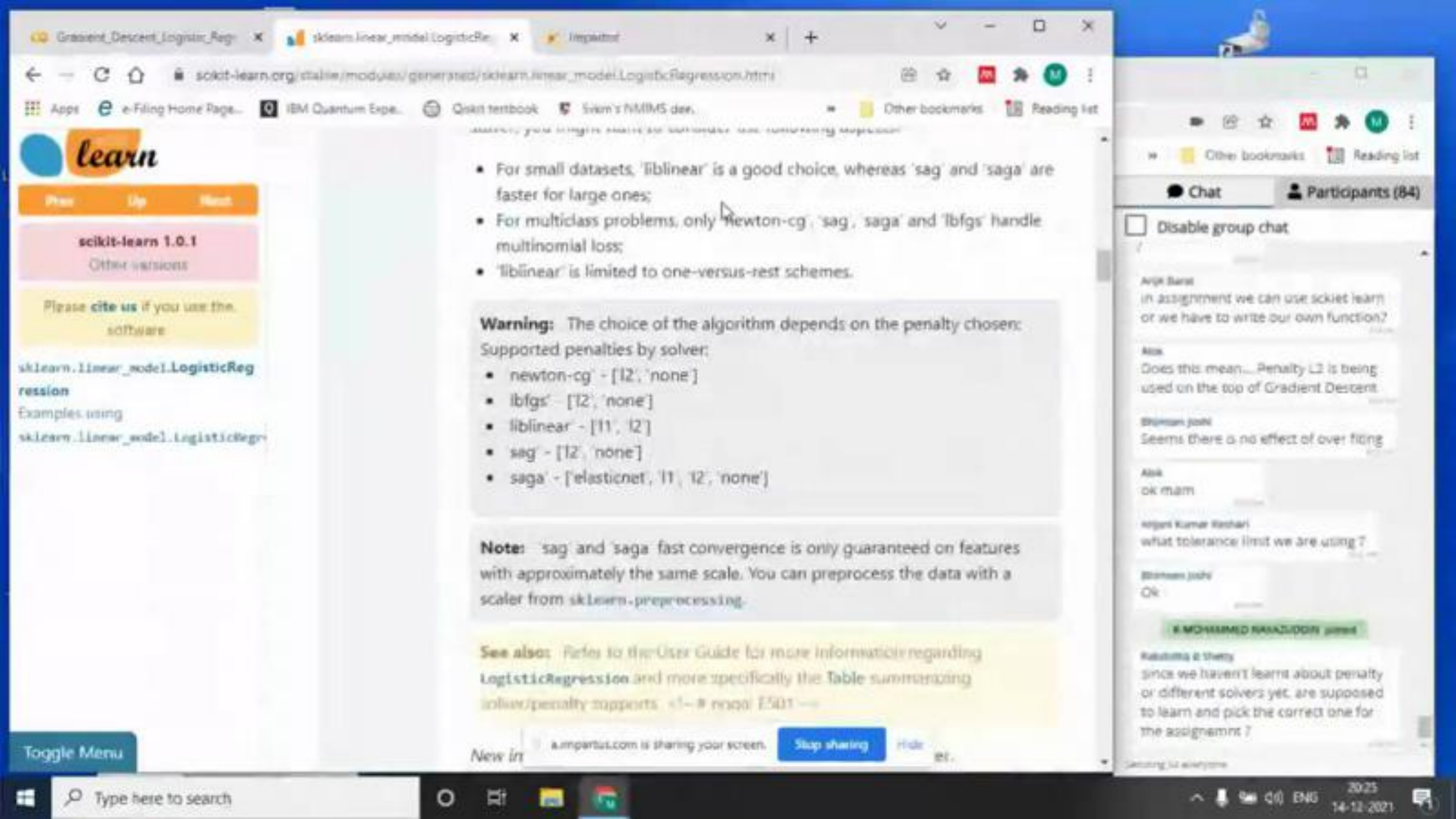
Anjani Kumar Keshari: what tolerance limit we are using ?

Bhimsen Joshi: Ok

Type here to search

ENG 20:25 14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticReg
impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
New in version 0.17: class_weight='balanced'
random_state : int, RandomState instance, default=None
Used when solver == 'sag', 'saga' or 'liblinear' to shuffle the data. See Glossary for details.
solver : {'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}, default='lbfgs'
Algorithm to use in the optimization problem. Default is 'lbfgs'. To choose a solver, you might want to consider the following aspects:
For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones;
For multiclass problems, only 'newton-cg', 'sag', 'saga' and 'lbfgs' handle multinomial loss;
'liblinear' is limited to one-versus-rest schemes.
Warning: The choice of the algorithm depends on the penalty chosen: Supported penalties by solver:
newton-cg - ['l2', 'none']
'lbfgs' - ['l2', 'none']
'liblinear' - ['l1', 'l2']
'sag' - ['l2', 'none']
'saga' - ['elasticnet', 'l1', 'l2', 'none']
Note:
a.impartus.com is sharing your screen.
Stop sharing
Hide
on features
Toggle Menu
Other bookmarks
Reading list
Chat
Participants (83)
Disable group chat
YADAVALLI NAGA VENKAT RAMA SRIKANTH joined
SHYAMALA GOWRI V G joined
Alok
reached minima
?
Arijit Barai
in assignment we can use sciket learn or we have to write our own function?
Alok
Does this mean... Penalty L2 is being used on the top of Gradient Descent
Bhimsen Joshi
Seems there is no effect of over fiting
Alok
ok mam
Anjani Kumar Keshari
what tolerance limit we are using ?
Bhimsen Joshi
Ok
Type here to search
ENG
20:25
14-12-2021


scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Other bookmarks
Reading list
learn
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones;
For multiclass problems, only 'newton-cg', 'sag', 'saga' and 'lbfgs' handle multinomial loss;
'liblinear' is limited to one-versus-rest schemes.
Warning: The choice of the algorithm depends on the penalty chosen: Supported penalties by solver:
newton-cg - ['l2', 'none']
lbfgs - ['l2', 'none']
liblinear - ['l1', 'l2']
sag - ['l2', 'none']
saga - ['elasticnet', 'l1', 'l2', 'none']
Note: 'sag' and 'saga' fast convergence is only guaranteed on features with approximately the same scale. You can preprocess the data with a scaler from sklearn.preprocessing.
See also: Refer to the User Guide for more information regarding LogisticRegression and more specifically the Table summarizing solver/penalty supports
Toggle Menu
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (84)
Disable group chat
Anjit Barai
in assignment we can use sckiet learn or we have to write our own function?
Alok
Does this mean.....Penalty L2 is being used on the top of Gradient Descent
Bhimsen Joshi
Seems there is no effect of over fitting
Alok
ok mam
Anjani Kumar Keshari
what tolerance limit we are using ?
Bhimsen Joshi
Ok
MOHAMMED RAHAZUDDIN joined
Rakshitha K Shetty
since we haven't learnt about penalty or different solvers yet, are supposed to learn and pick the correct one for the assignemnt ?
Type here to search
ENG
20:25
14-12-2021

Gradient_Descent_Logistic_Regr | sklearn.linear_model.LogisticReg | Impartus

scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps | e-Filing Home Page... | IBM Quantum Expe... | Qiskit textbook | Sukm's NMIMS dee... | Other bookmarks | Reading list

learn

Prev | Up | Next

scikit-learn 1.0.1
Other versions

Please cite us if you use the software.

sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr

- For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones;
- For multiclass problems, only 'newton-cg', 'sag', 'saga' and 'lbfgs' handle multinomial loss;
- 'liblinear' is limited to one-versus-rest schemes.

Warning: The choice of the algorithm depends on the penalty chosen: Supported penalties by solver:

- 'newton-cg' - ['l2', 'none']
- 'lbfgs' - ['l2', 'none']
- 'liblinear' - ['l1', 'l2']
- 'sag' - ['l2', 'none']
- 'saga' - ['elasticnet', 'l1', 'l2', 'none']

Note: 'sag' and 'saga' fast convergence is only guaranteed on features with approximately the same scale. You can preprocess the data with a scaler from sklearn.preprocessing.

See also: Refer to the User Guide for more information regarding LogisticRegression and more specifically the Table summarizing solver/penalty supports. <!-- # noqa: E501 -->

New in

a.impartus.com is sharing your screen. Stop sharing | Hide

Toggle Menu

Other bookmarks | Reading list

Chat | Participants (84)

Disable group chat

in assignment we can use sckiet learn or we have to write our own function?

Alok
Does this mean... Penalty L2 is being used on the top of Gradient Descent

Bhimsen Joshi
Seems there is no effect of over fiting

Alok
ok mam

Anjani Kumar Keshari
what tolerance limit we are using ?

Bhimsen Joshi
Ok

K MOHAMMED NAYAZUDDIN joined

Rakshitha B Shetty
since we haven't learnt about penalty or different solvers yet, are supposed to learn and pick the correct one for the assignemnt ?

okay

Sending to everyone

Type here to search

ENG 20:25 14-12-2021

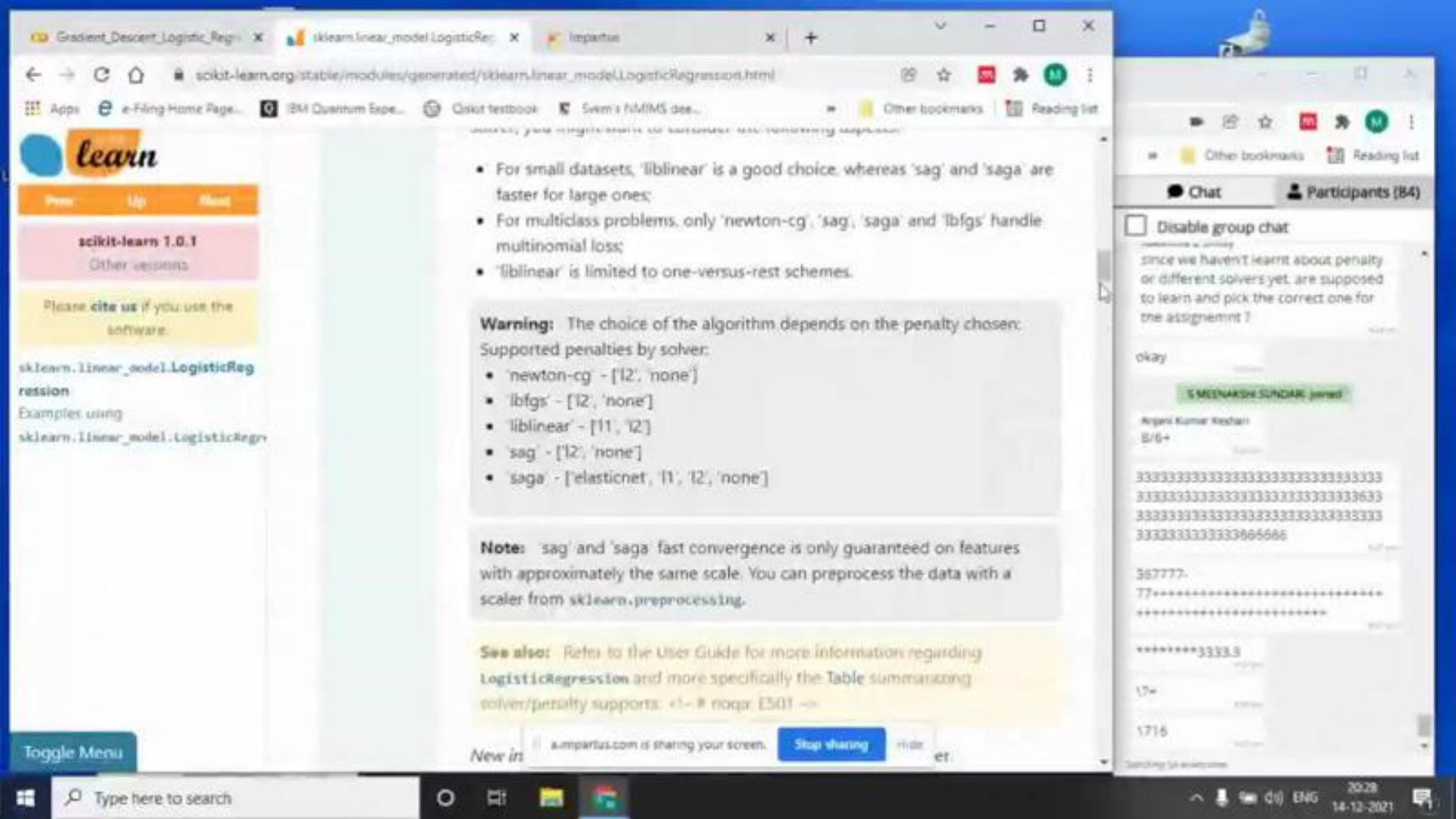
- For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones;
- For multiclass problems, only newton-cg, 'sag', saga and 'lbfgs' handle multinomial loss;
- 'liblinear' is limited to one-versus-rest schemes.

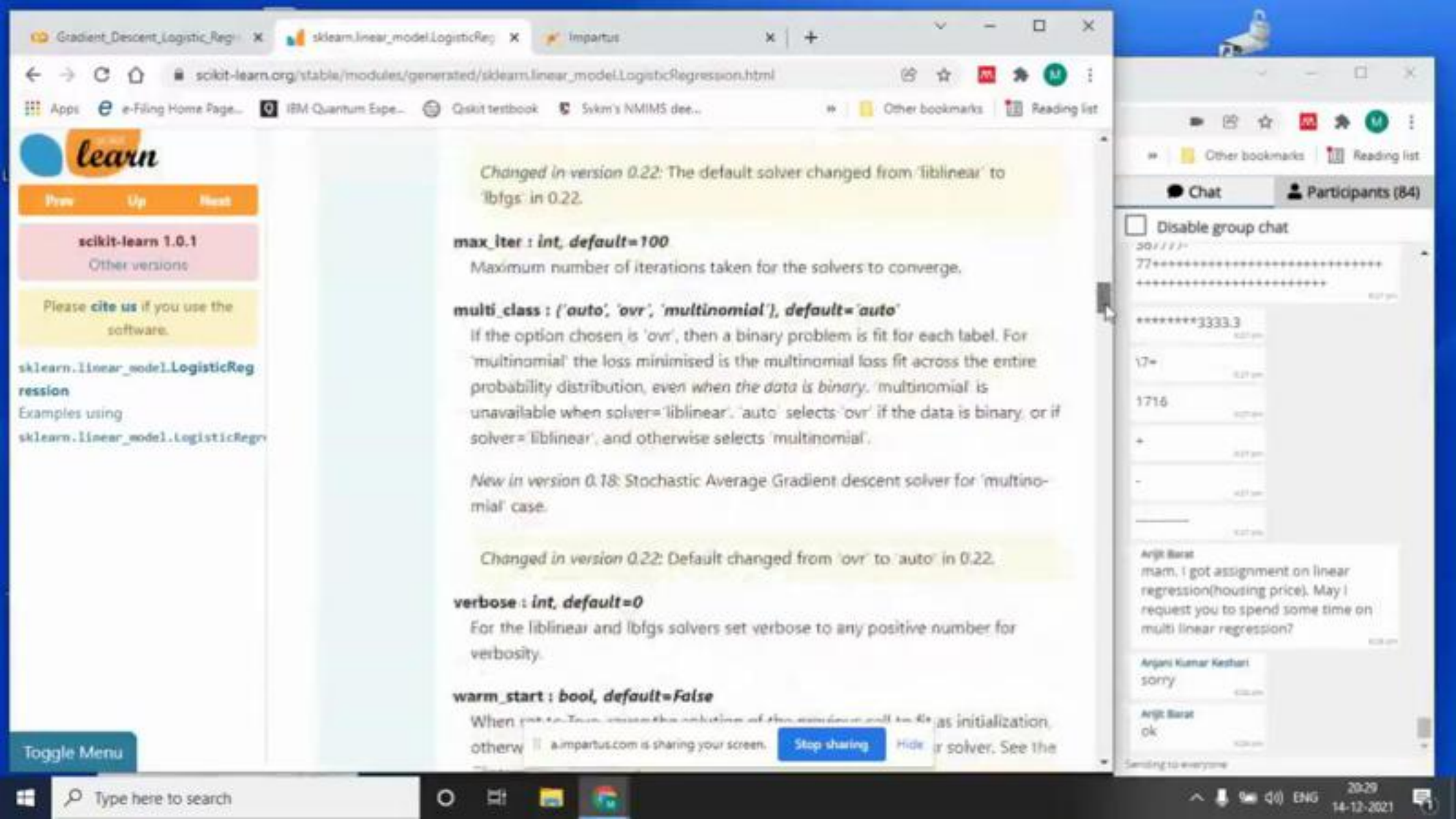
Warning: The choice of the algorithm depends on the penalty chosen: Supported penalties by solver:

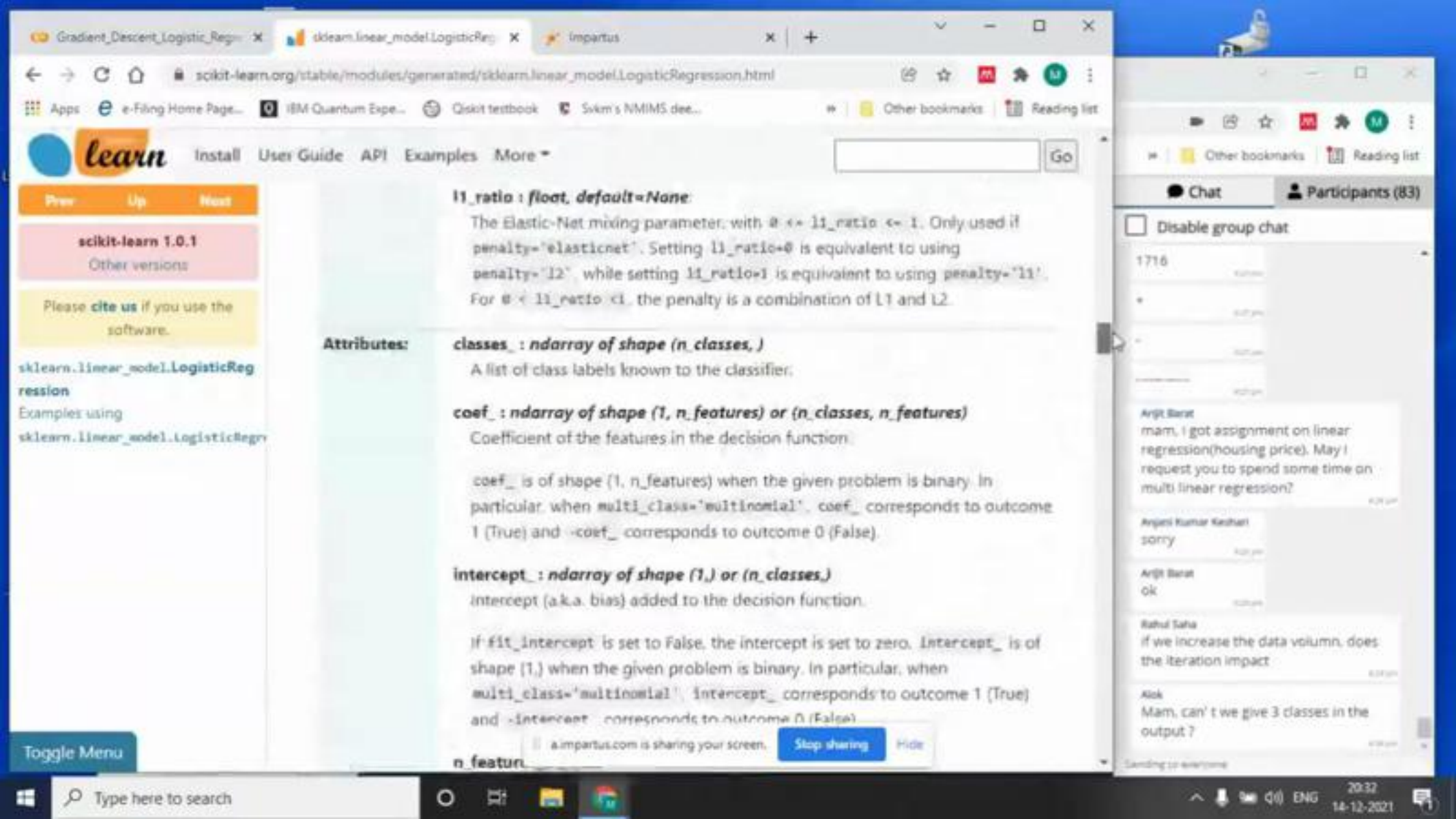
- newton-cg - ['l2', 'none']
- lbfgs - ['l2', 'none']
- liblinear - ['l1', 'l2']
- sag - ['l2', 'none']
- saga - ['elasticnet', 'l1', 'l2', 'none']

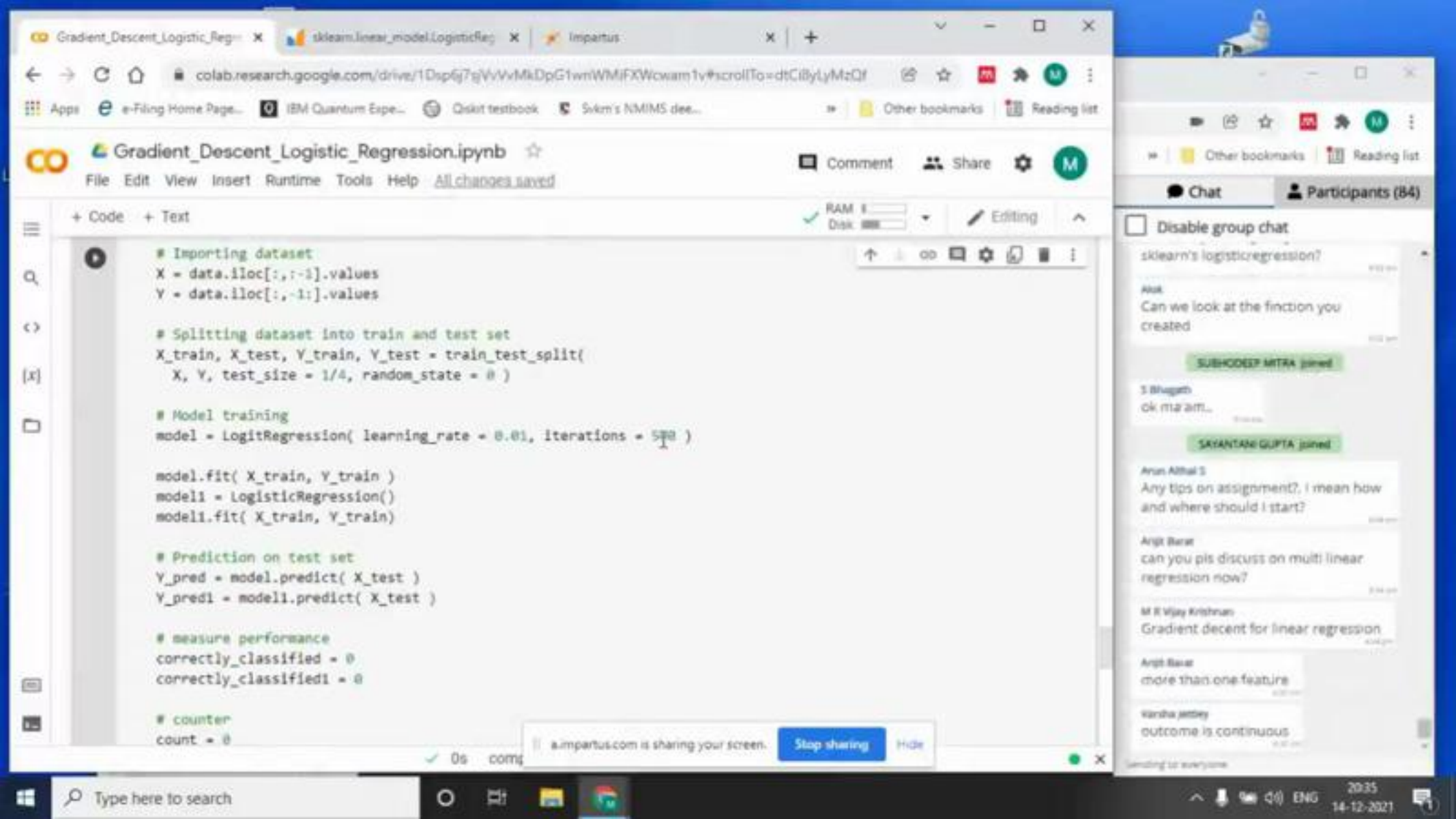
Note: sag and 'saga' fast convergence is only guaranteed on features with approximately the same scale. You can preprocess the data with a scaler from `sklearn.preprocessing`.

See also: Refer to the user guide for more information regarding `LogisticRegression` and more specifically the Table summarizing solver/penalty supports.


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticRe
tirpathai
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svem's IMIMS dee...
Other bookmarks
Reading list
learn
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones;
For multiclass problems, only 'newton-cg', 'sag', 'saga' and 'lbfgs' handle multinomial loss;
'liblinear' is limited to one-versus-rest schemes.
Warning: The choice of the algorithm depends on the penalty chosen: Supported penalties by solver:
'newton-cg' - ['l2', 'none']
'lbfgs' - ['l2', 'none']
'liblinear' - ['l1', 'l2']
'sag' - ['l2', 'none']
'saga' - ['elasticnet', 'l1', 'l2', 'none']
Note: 'sag' and 'saga' fast convergence is only guaranteed on features with approximately the same scale. You can preprocess the data with a scaler from sklearn.preprocessing.
See also: Refer to the User Guide for more information regarding LogisticRegression and more specifically the Table summarizing solver/penalty supports. <!-- # noqa: E501 -->
New in
et.
Toggle Menu
s.ampartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (84)
Disable group chat
since we haven't learnt about penalty or different solvers yet, are supposed to learn and pick the correct one for the assignemnt ?
okay
S MEENAKSHI SUNDARI joined
Anjani Kumar Keshari
8/6+
3333333333333333333333333333
3333333333333333333333333633
3333333333333333333333333333
333333333333666666
367777-
77**********
****
********3333.3
17+
1716
Type here to search
ENG
20:28
14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticReg
Impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using
sklearn.linear_model.LogisticRegr
Changed in version 0.22: The default solver changed from 'liblinear' to 'lbfgs' in 0.22.
max_iter : int, default=100
Maximum number of iterations taken for the solvers to converge.
multi_class : {'auto', 'ovr', 'multinomial'}, default='auto'
If the option chosen is 'ovr', then a binary problem is fit for each label. For 'multinomial' the loss minimised is the multinomial loss fit across the entire probability distribution, even when the data is binary. 'multinomial' is unavailable when solver='liblinear'. 'auto' selects 'ovr' if the data is binary, or if solver='liblinear', and otherwise selects 'multinomial'.
New in version 0.18: Stochastic Average Gradient descent solver for 'multinomial' case.
Changed in version 0.22: Default changed from 'ovr' to 'auto' in 0.22.
verbose : int, default=0
For the liblinear and lbfgs solvers set verbose to any positive number for verbosity.
warm_start : bool, default=False
as initialization,
solver. See the
a.impartus.com is sharing your screen.
Stop sharing
Hide
Toggle Menu
Other bookmarks
Reading list
Chat
Participants (84)
Disable group chat
********3333.3
\7=
1716
+
-
Arijit Barat
mam, I got assignment on linear regression(housing price). May I request you to spend some time on multi linear regression?
Anjani Kumar Keshari
sorry
Arijit Barat
ok
Sending to everyone
Type here to search
ENG
20:29
14-12-2021


Gradient_Descent_Logistic_Regr
sklearn.linear_model.LogisticReg
Impartus
scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Apps
e-Filing Home Page...
IBM Quantum Expe...
Qiskit textbook
Svkm's NMIMS dee...
Other bookmarks
Reading list
learn
Install
User Guide
API
Examples
More
Go
Prev
Up
Next
scikit-learn 1.0.1
Other versions
Please cite us if you use the software.
sklearn.linear_model.LogisticRegression
Examples using sklearn.linear_model.LogisticRegr
l1_ratio : float, default=None
The Elastic-Net mixing parameter, with 0 <= l1_ratio <= 1. Only used if penalty='elasticnet'. Setting l1_ratio=0 is equivalent to using penalty='l2', while setting l1_ratio=1 is equivalent to using penalty='l1'. For 0 < l1_ratio <1, the penalty is a combination of L1 and L2.
Attributes:
classes_ : ndarray of shape (n_classes,)
A list of class labels known to the classifier.
coef_ : ndarray of shape (1, n_features) or (n_classes, n_features)
Coefficient of the features in the decision function.
coef_ is of shape (1, n_features) when the given problem is binary. In particular, when multi_class='multinomial', coef_ corresponds to outcome 1 (True) and -coef_ corresponds to outcome 0 (False).
intercept_ : ndarray of shape (1,) or (n_classes,)
Intercept (a.k.a. bias) added to the decision function.
If fit_intercept is set to False, the intercept is set to zero. intercept_ is of shape (1,) when the given problem is binary. In particular, when multi_class='multinomial', intercept_ corresponds to outcome 1 (True) and -intercept_ corresponds to outcome 0 (False).
a.impartus.com is sharing your screen.
Stop sharing
Hide
Toggle Menu
Chat
Participants (83)
Disable group chat
1716
Arijit Barat
mam, I got assignment on linear regression(housing price). May I request you to spend some time on multi linear regression?
Anjani Kumar Keshari
sorry
Arijit Barat
ok
Rahul Saha
if we increase the data volumn, does the iteration impact
Alok
Mam, can' t we give 3 classes in the output ?
Type here to search
ENG
20:32
14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
# Importing dataset
X = data.iloc[:,:-1].values
Y = data.iloc[:,-1:].values
# Splitting dataset into train and test set
X_train, X_test, Y_train, Y_test = train_test_split(
X, Y, test_size = 1/4, random_state = 0)
# Model training
model = LogitRegression(learning_rate = 0.01, iterations = 500)
model.fit(X_train, Y_train)
model1 = LogisticRegression()
model1.fit(X_train, Y_train)
# Prediction on test set
Y_pred = model.predict(X_test)
Y_pred1 = model1.predict(X_test)
# measure performance
correctly_classified = 0
correctly_classified1 = 0
# counter
count = 0
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (84)
Disable group chat
sklearn's logisticregression?
Can we look at the finction you created
SUBHODEEP MITRA joined
ok ma am..
SAYANTANI GUPTA joined
Any tips on assignment?, I mean how and where should I start?
can you pls discuss on multi linear regression now?
Gradient decent for linear regression
more than one feature
outcome is continuous
Type here to search
ENG
20:35
14-12-2021

Gradient_Descent_Logistic_Regr | sklearn.linear_model.LogisticReg | Impartus

colab.research.google.com/drive/1Dsp6j7gjVvVvMkDpG1wnWMiFXWowarm1v#scrollTo=dtCiByLyMzQf

Apps | e-Filing Home Page... | IBM Quantum Expe... | Qiskit textbook | Svkm's NMIMS dee... | Other bookmarks | Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment | Share | M

+ Code + Text | RAM | Disk | Editing

Logistic Regression, Gradient Descent

Loss function,

Python implementation

Logistic Regression

Logistic regression is a probabilistic model, used to establish the relationship between a binary dependent variable and one or more independent variables. Let X_i be the independent variable (training variable) and Y be the dependent variable. Linear Regression predicts probability of outcome of binary dependent variable (Y) and the output lies between 0 and 1. The prediction equation between X_i and Y is

$$ln(\frac{p}{1-p}) = b_0 + b_1x_1 + b_2x_2 + b_3x_3 \ldots b_nx_n$$

$ln(\frac{p}{1-p})$ is log odds and denotes the likelihood of the event taking place. It maps the probability between 0 and 1. The terms $b_0, b_1, b_2 \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{1-p} = e^{b_0+b_1x_1+b_2x_2+b_3x_3 \ldots b_nx_n}$$

simplifying it further

a.impartus.com is sharing your screen. Stop sharing Hide

0s comp

Other bookmarks | Reading list

Chat | Participants (84)

Disable group chat

SUBHODEEP MITRA joined

S Bhagath
ok ma'am,,

SAYANTANI GUPTA joined

Arun Abhat S
Any tips on assignment?, I mean how and where should I start?

Arijit Barat
can you pls discuss on multi linear regression now?

M R Vijay Krishnan
Gradient decent for linear regression

Arijit Barat
more than one feature

Vanisha Jaidev
outcome is continuous

Rajesh R
Linear is continuous

S Bhagath
continous

Sending to everyone

Type here to search

ENG 20:35 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
data = pd.read_csv(io.BytesIO(uploaded['diabetes.csv']))
data.head()
```

diabetes.csv

- **diabetes.csv**(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done

Saving diabetes.csv to diabetes.csv

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
[5] data = data.dropna()
    print(data.shape)
    print(list(data.columns))

    (760, 9)
    ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (84)

Disable group chat

SUBHODEEP MITRA joined

S Bhagath: ok ma'am..

SAYANTANI GUPTA joined

Arun Aithal S: Any tips on assignment?, I mean how and where should I start?

Arijit Barat: can you pls discuss on multi linear regression now?

M R Vijay Krishnan: Gradient decent for linear regression

Arijit Barat: more than one feature

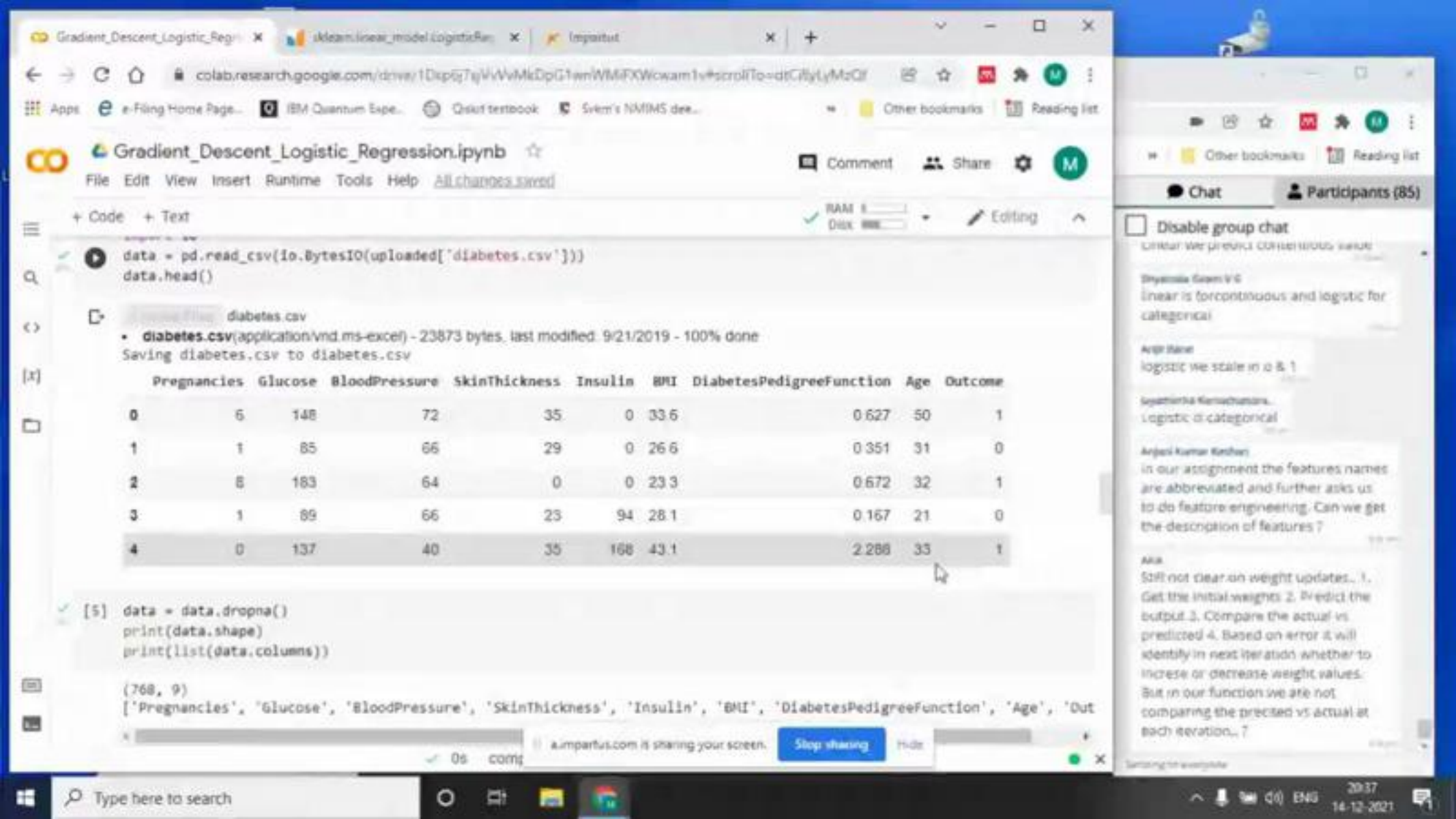
Varsha Jetley: outcome is continuous

Rajesh A: linear is continuous

S Bhagath: continous

Type here to search

ENG 20:35 14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
data = pd.read_csv(io.BytesIO(uploaded['diabetes.csv']))
data.head()
diabetes.csv
diabetes.csv(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done
Saving diabetes.csv to diabetes.csv
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome
0 6 148 72 35 0 33.6 0.627 50 1
1 1 85 66 29 0 26.6 0.351 31 0
2 8 183 64 0 0 23.3 0.672 32 1
3 1 89 66 23 94 28.1 0.167 21 0
4 0 137 40 35 168 43.1 2.288 33 1
[5] data = data.dropna()
print(data.shape)
print(list(data.columns))
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
0s
a.umpartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (85)
Disable group chat
Linear is forcontinuous and logistic for categorical
logistic we scale in 0 & 1
Logistic is categorical
in our assignment the features names are abbreviated and further asks us to do feature engineering. Can we get the description of features ?
Still not clear on weight updates.. 1. Get the initial weights 2. Predict the output 3. Compare the actual vs predicted 4. Based on error it will identify in next iteration whether to increse or decrease weight values. But in our function we are not comparing the precited vs actual at each iteration.. ?
Type here to search
ENG
20:37
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[5] data = data.dropna()
    print(data.shape)
    print(list(data.columns))
```

```
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

```
[6] data['Outcome'].value_counts()
    sns.countplot(x='Outcome',data=data, palette='hls')
    plt.show()
```

0s comp

a.umpartus.com is sharing your screen. Stop sharing Hide

Chat Participants (85)

Disable group chat

Logistic is categorical

In our assignment the features names are abbreviated and further asks us to do feature engineering. Can we get the description of features ?

Still not clear on weight updates.. 1. Get the initial weights 2. Predict the output 3. Compare the actual vs predicted 4. Based on error it will identify in next iteration whether to increse or decrease weight values. But in our function we are not comparing the precited vs actual at each iteration.. ?

Gradient decent for linear regression

I told multi linear regression means if we have more than one x column (feature), and simple linear regression means only one x feature column.

Type here to search

ENG 20:38 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

```
# Helper function to update weights in gradient descent

    def update_weights( self ) :
        A = 1 / ( 1 + np.exp( - ( self.X.dot( self.W ) + self.b ) ) )
        # calculate gradients
        tmp = ( A - self.Y.T )
        tmp = np.reshape( tmp, self.m )
        dW = np.dot( self.X.T, tmp ) / self.m
        db = np.sum( tmp ) / self.m
        # update weights
        self.W = self.W - self.learning_rate * dW
        self.b = self.b - self.learning_rate * db
        return self

    # Hypothetical function  h( x )

    def predict( self, X ) :
        Z = 1 / ( 1 + np.exp( - ( X.dot( self.W ) + self.b ) ) )
        Y = np.where( Z > 0.5, 1, 0 )
        return Y
```

[21] def main() :

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (84)

Disable group chat

Alok
Still not clear on weight updates.. 1. Get the initial weights 2. Predict the output 3. Compare the actual vs predicted 4. Based on error it will identify in next iteration whether to increse or decrease weight values. But in our function we are not comparing the precited vs actual at each iteration.. ?

M R Vijay Krishnan
Gradient decent for linear regression

Arijit Bishi
I told multi linear regression means if we have more than one x column (feature), and simple linear regression means only one x feature column.

Alok
But where we are comparing result in function to update weight

each iteration

ok mam

Type here to search

ENG 20:39 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

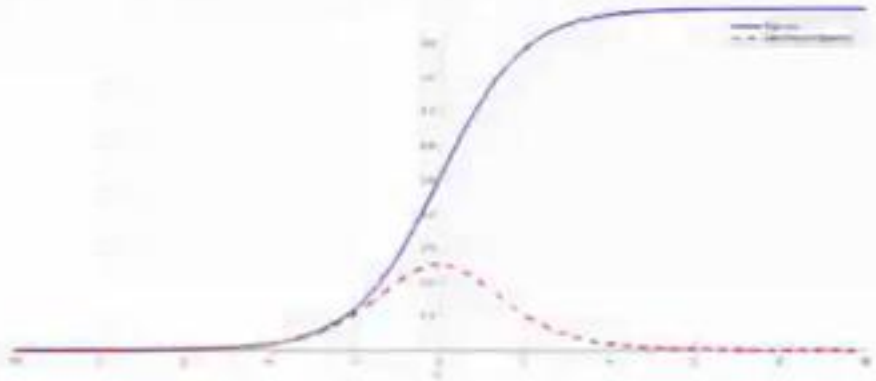
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

$$= \frac{1}{(1+e^{-X})} \cdot (1 - \frac{1}{1+e^{-X}})$$
$$= \sigma(X) \cdot (1 - \sigma(X))$$



Now we will be using the above derived equation to make our predictions. Before that we will train our model to obtain the values of our parameters $b_0, b_1, b_2 \ldots$ that result in least error. This is where the error or loss function comes in.

Loss Function

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (82)

Disable group chat

each iteration.. ?

M R Vijay Krishnan
Gradient decent for linear regression

Arijit Barat
I told multi linear regression means if we have more than one x column (feature), and simple linear regression means only one x feature column.

Alok
But where we are comparing result in function to update weight

each iteration

ok mam

got it

Arijit Barat
pls spend some time on linear regression using gradient descent

M R Vijay Krishnan
Gradient decent for linear regression

Type here to search

ENG 20:40 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

$b_0, b_1, b_2 \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{1-p} = e^{b_0+b_1x_1+b_2x_2+b_3x_3 \ldots b_nx_n}$$

simplifying it further

$$p = \frac{e^{b_0+b_1x_1+b_2x_2+b_3x_3 \ldots b_nx_n}}{1+e^{b_0+b_1x_1+b_2x_2+b_3x_3 \ldots b_nx_n}}$$

or

$$p = \frac{1}{1+e^{-(b_0+b_1x_1+b_2x_2+b_3x_3 \ldots b_nx_n)}}$$

This is the equation of the *Sigmoid Function* given by:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Deriving the sigmoid function,

$$\sigma'(x) = \frac{d}{dX}\sigma(x) = \frac{d}{dX}\frac{1}{(1+e^{-X})}$$

$$= \frac{d}{dX}(1+e^{-X})^{-1}$$

Using reciprocal rule, then rule of linearity,

$$= -(1+e^{-X}$$

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (78)

Disable group chat

Other bookmarks Reading list

Type here to search

ENG 20:42 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

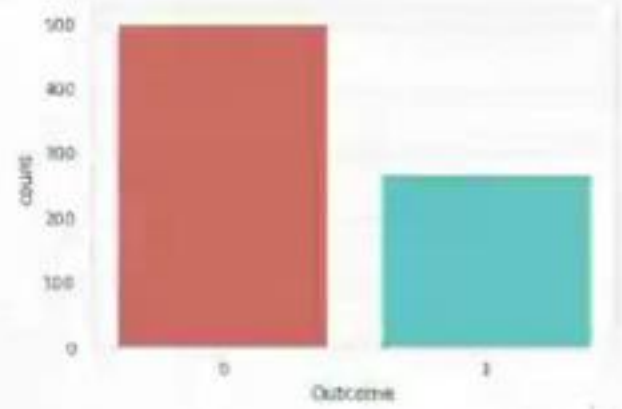
Comment Share

+ Code + Text

RAM Disk Editing

```
[5] print(list(data.columns))

    (768, 9)
    ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

```
data['Outcome'].value_counts()
sns.countplot(x='Outcome',data=data, palette='hls')
plt.show()
```



0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Chat Participants (80)

Disable group chat

would you share the colab notebook with us please?

Shyamala Gowri V G
Do we have to do all the preprocessing steps for the give data?

VARANASI SHIVANI joined

Akila
Yes

S Bhagath
yes

Shyamala Gowri V G
Yes mam

Hareesh S V
NO

M R Vijay Krishnan
Yes

Sending to everyone

Type here to search

ENG 20:44 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[4]

diabetes.csv

- **diabetes.csv**(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done

Saving diabetes.csv to diabetes.csv

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
data = data.dropna()
print(data.shape)
print(list(data.columns))
```

```
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

[6] data['Outcome'].value_counts()

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (80)

Disable group chat

Yes mam

NO

Yes

Can we use all the functions within sklearn which would be relevant to the assignment?

Outliers mam?

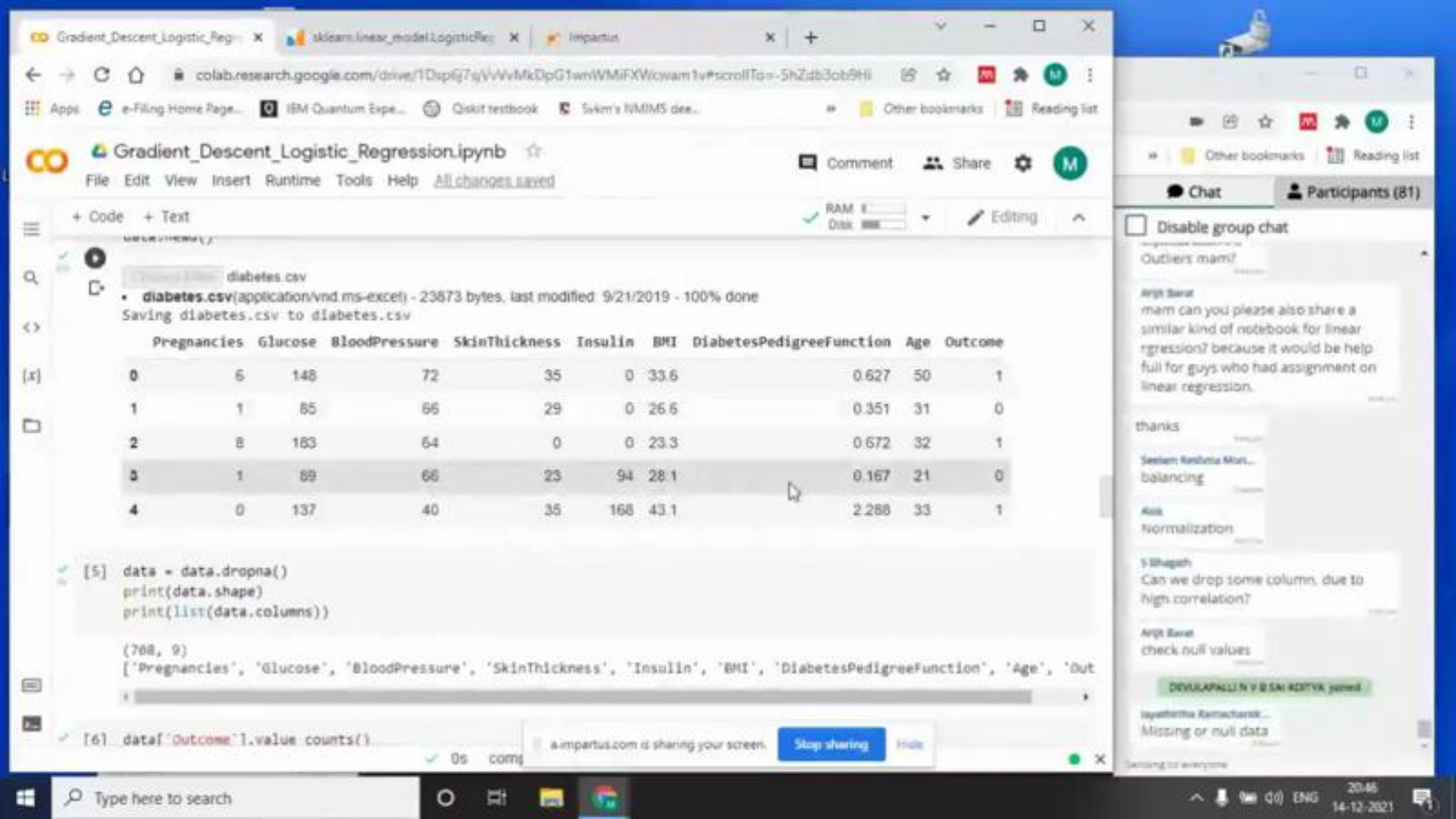
mam can you please also share a similar kind of notebook for linear rgression? because it would be help full for guys who had assignment on linear regression.

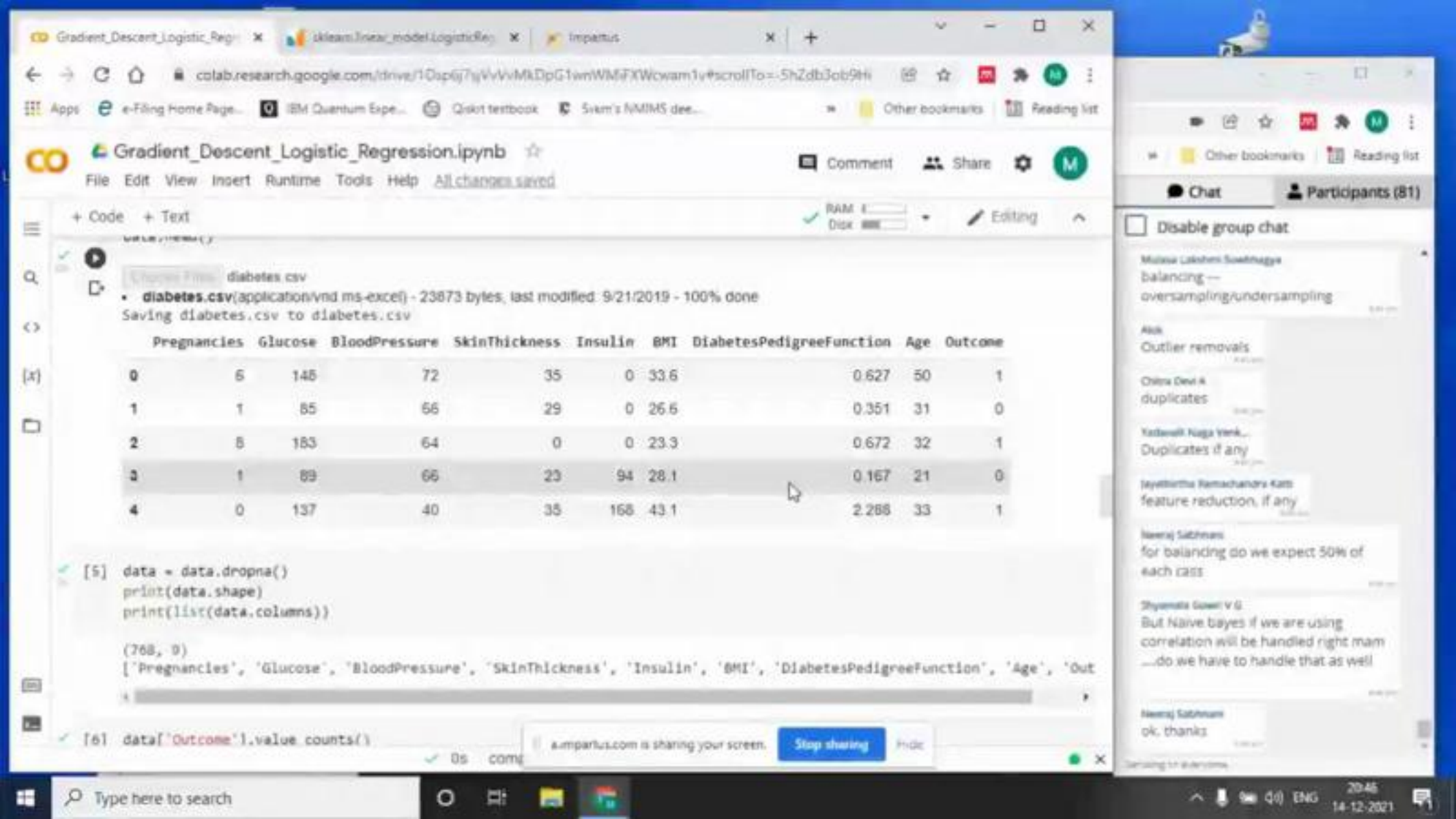
thanks

balancing

Type here to search

ENG 20:45 14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
diabetes.csv
diabetes.csv(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done
Saving diabetes.csv to diabetes.csv
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome
0 6 148 72 35 0 33.6 0.627 50 1
1 1 85 66 29 0 26.6 0.351 31 0
2 8 183 64 0 0 23.3 0.672 32 1
3 1 89 66 23 94 28.1 0.167 21 0
4 0 137 40 35 168 43.1 2.288 33 1
[5] data = data.dropna()
print(data.shape)
print(list(data.columns))
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
[6] data['Outcome'].value_counts()
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (81)
Disable group chat
Outliers mam?
Arijit Barat
mam can you please also share a similar kind of notebook for linear rgression? because it would be help full for guys who had assignment on linear regression.
thanks
balancing
Normalization
S Bhagath
Can we drop some column, due to high correlation?
Arijit Barat
check null values
DEVULAPALLI N V B SAI ADITYA joined
Missing or null data
Other bookmarks
Reading list
Type here to search
ENG
20:46
14-12-2021


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
diabetes.csv
diabetes.csv(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done
Saving diabetes.csv to diabetes.csv
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome
0 6 148 72 35 0 33.6 0.627 50 1
1 1 85 66 29 0 26.6 0.351 31 0
2 8 183 64 0 0 23.3 0.672 32 1
3 1 89 66 23 94 28.1 0.167 21 0
4 0 137 40 35 168 43.1 2.288 33 1
[5] data = data.dropna()
print(data.shape)
print(list(data.columns))
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
[6] data['Outcome'].value_counts()
a.umpartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (81)
Disable group chat
balancing --- oversampling/undersampling
Outlier removals
duplicates
Duplicates if any
feature reduction, if any
for balancing do we expect 50% of each class
But Naive bayes if we are using correlation will be handled right mam ....do we have to handle that as well
ok, thanks
Type here to search
ENG
20:46
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

diabetes.csv

- **diabetes.csv**(application/vnd.ms-excel) - 23873 bytes, last modified: 5/21/2019 - 100% done

```
Saving diabetes.csv to diabetes.csv
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

[5]
```
data = data.dropna()
print(data.shape)
print(list(data.columns))
```

```
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

[6]
```
data['Outcome'].value_counts()
```

Chat:

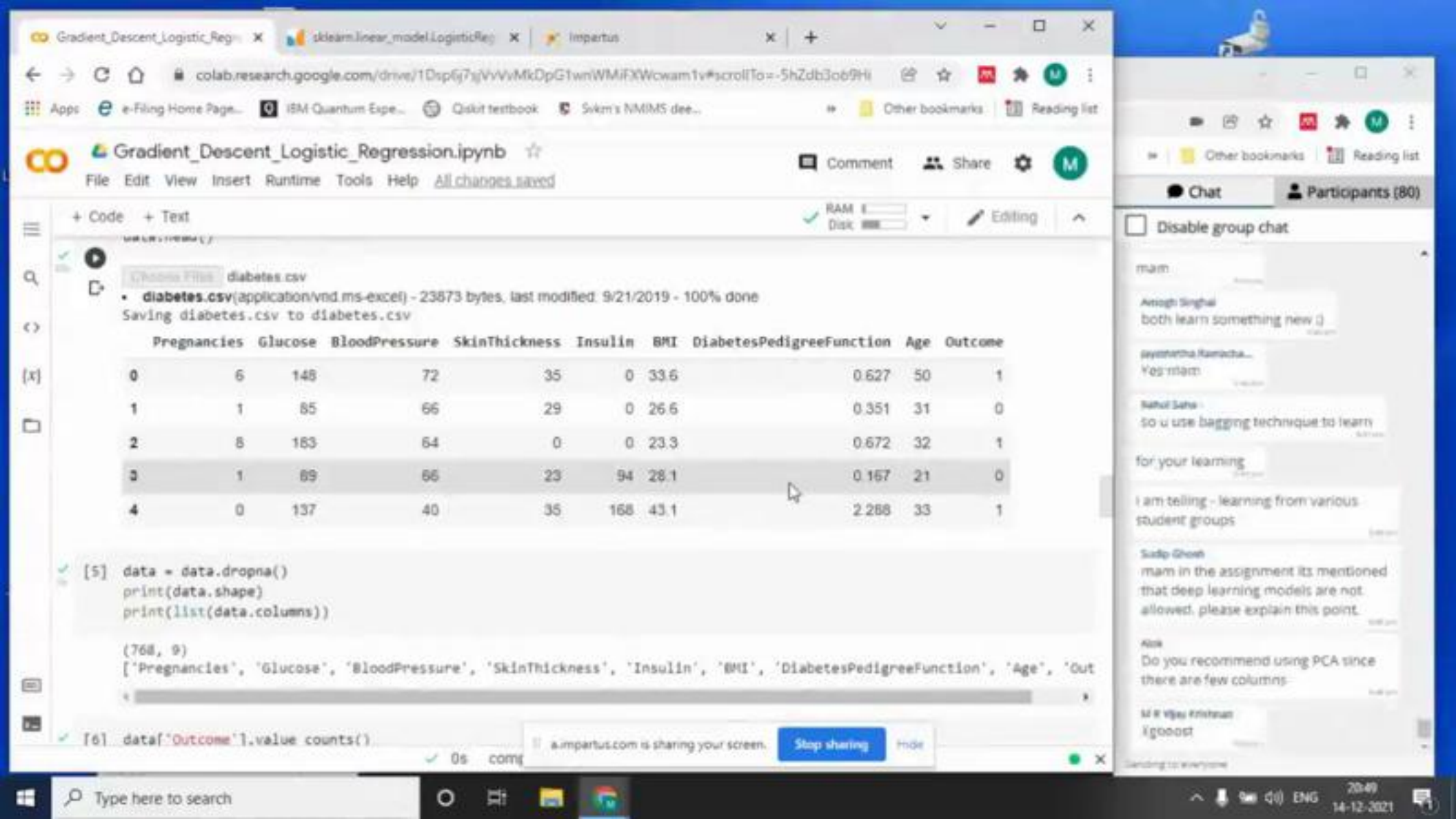
you are right

both learn something new :)

Yes mam

so u use bagging technique to learn

for your learning


Gradient_Descent_Logistic_Regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Choose Files diabetes.csv
diabetes.csv(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done
Saving diabetes.csv to diabetes.csv
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome
0 6 148 72 35 0 33.6 0.627 50 1
1 1 85 66 29 0 26.6 0.351 31 0
2 8 183 64 0 0 23.3 0.672 32 1
3 1 89 66 23 94 28.1 0.167 21 0
4 0 137 40 35 168 43.1 2.288 33 1
[5] data = data.dropna()
print(data.shape)
print(list(data.columns))
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
[6] data['Outcome'].value_counts()
a.impartus.com is sharing your screen.
Stop sharing
Hide
Chat
Participants (80)
Disable group chat
mam
both learn something new :)
Yes mam
so u use bagging technique to learn
for your learning
i am telling - learning from various student groups
mam in the assignment its mentioned that deep learning models are not allowed. please explain this point.
Do you recommend using PCA since there are few columns
Igoooot
Type here to search
ENG
20:49
14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[4]

Choose Files diabetes.csv

- **diabetes.csv**(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done

Saving diabetes.csv to diabetes.csv

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
data = data.dropna()
print(data.shape)
print(list(data.columns))
```

```
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

[6] data['Outcome'].value_counts()

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (80)

Disable group chat

I am telling - learning from various student groups

Sudip Ghosh
mam in the assignment its mentioned that deep learning models are not allowed. please explain this point.

Alok
Do you recommend using PCA since there are few columns

M R Vijay Krishnan
xgboost

Can we use xgboost

Alok
yes

M R Vijay Krishnan
Ok

Alok
what do we have in day after tomorrow webinar mam

ok

Type here to search

ENG 20:50 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[4]

Choose Files diabetes.csv

- **diabetes.csv**(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done

Saving diabetes.csv to diabetes.csv

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
data = data.dropna()
print(data.shape)
print(list(data.columns))
```

```
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

[6] data['Outcome'].value_counts()

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (79)

Disable group chat

ok mam

Jayathirtha Ramachandra Katti
But, we also have DVI the same day ...

Ravi Chandra Vedula
Same time another webinar

Jayathirtha Ramachandra Katti
DVI got re-scheduled to 16th

Varsha Jettley
yes we have DVI

M R Vijay Krishnan
DVI is tomorrow

Varsha Jettley
webinar on 16dec

M R Vijay Krishnan
They preponed it

Santosh Kumar C V
DVI is on 16th

Varsha Jettley
linear regressiona can it partially idscussed

Sending to everyone

Type here to search

ENG 20:51 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[4]

Choose Files diabetes.csv

- **diabetes.csv**(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done

Saving diabetes.csv to diabetes.csv

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

[5]

```
data = data.dropna()
print(data.shape)
print(list(data.columns))
```

```
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

[6] data['Outcome'].value_counts()

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (74)

Disable group chat

Amogh Singhal: mam, the idea is not to have the highest accuracy but follow proper process ?

Arijit Barat: thanks mam

Chitra Devi A: thankyou mam

Hareesh S V: thank you ma'm

Bhimsen Joshi: good session.. thank you

Jayathirtha Ramacha...: Thanks mam

Dakshinaya Udaipraki...: Thank you mam

S Krishnamoorthy: Thankyou mam

S Bhagath: Thankyou so much, ma'am..

Sending to everyone

Type here to search

ENG 20:52 14-12-2021

New in version 0.19: l1 penalty with SAGA solver (allowing 'multinomial' + L1)

dual : *bool, default=False*
Dual or primal formulation. Dual formulation is only implemented for l2 penalty with liblinear solver. Prefer dual=False when n_samples > n_features.

tol : *float, default=1e-4*
Tolerance for stopping criteria.

C : *float, default=1.0*
Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization.

fit_intercept : *bool, default=True*
Specifies if a constant (a.k.a. bias or intercept) should be added to the decision function.

intercept_scaling : *float, default=1*
Useful only when the solver 'liblinear' is used and self.fit_intercept is set to True. In this case, x becomes [x, self.intercept_scaling], i.e. a "synthetic" feature with constant value equal to intercept_scaling is appended to the instance vector. The intercept becomes `intercept_scaling * synthetic_feature_weight`.

Note! t ation as all other feature iture weight

scikit-learn 1.0.1
Other versions

Please **cite us** if you use the software.

sklearn.linear_model.LogisticRegression

Examples using sklearn.linear_model.LogisticRegr

Toggle Menu

aumpartus.com is sharing your screen. Stop sharing Hide

Chat | Participants (73)

Disable group chat

Thankyou mam

Thankyou so much, ma'am.

Thanks mam

Thank You :)

These upload notebooks link in cam as

will this recording of the session available soon ? I missed the beginning

ok mam

mam I've joined late. Can you pls tell if these notebook links are included in the ppt?

Type here to search

ENG 20:53 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

independent variables. Let X_i be the independent variable (training variable) and Y be the dependent predicts probability of outcome of binary dependent variable (Y) and the output lies between 0 and 1. The prediction equation between X_i and Y is

$$ln(\frac{p}{1-p}) = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n$$

$ln(\frac{p}{1-p})$ is log odds and denotes the likelihood of the event taking place. It maps the probability between 0 and 1. The terms $b_0, b_1, b_2 \ldots$ are parameters estimated during training of each record. The above equation can be rewritten as

$$\frac{p}{1-p} = e^{b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n}$$

simplifying it further

$$p = \frac{e^{b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n}}{1 + e^{b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n}}$$

or

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots b_n x_n)}}$$

This is the equation of the "Sigmoid Function" given by:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Deriving the sigmoid function

0s completed at 20:53

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (72)

Disable group chat

S Krishnamoorthy
Thankyou mam

S Bhagath
Thankyou so much, ma'am...

S Kavitha
thanks mam

Bishra Aman Dinesh
Thank You..!!

Harsh Vashisht
Please upload notebook link in canvas.

Subhodeep Mitra
will this recording of the session available soon , i missed the beginning

Anmgh Singhal
ok mam

Vaishnavi Shivani
mam i've joined late. Can you pls tell if these notebook links are included in the ppt?

Type here to search

ENG 20:53 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

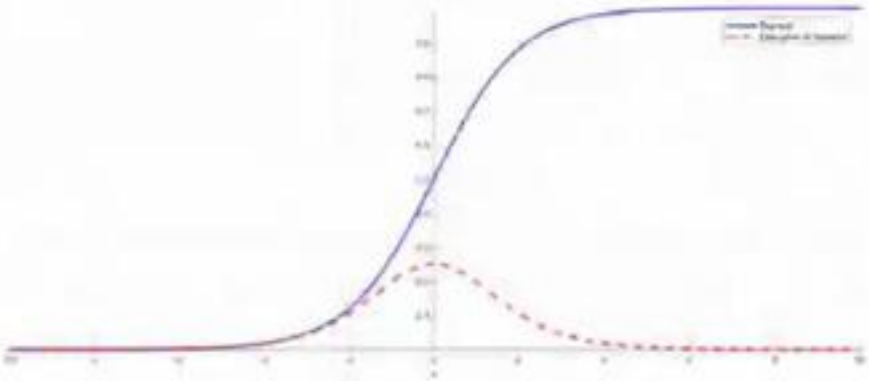
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

$$= \frac{1}{(1+e^{-X})} . (1 - \frac{1}{1+e^{-X}})$$
$$= \sigma(X) . (1 - \sigma(X))$$



Now we will be using the above derived equation to make our predictions. Before that we will train our model to obtain the values of our parameters $b_0, b_1, b_2 \ldots$ that result in least error. This is where the error or loss function comes in.

Loss Function

a.impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (72)

Disable group chat

S Krishnamoorthy
Thankyou mam

S Bhagath
Thankyou so much, ma'am..

S Kavitha
thanks mam

Bohra Aman Dinesh
Thank You..!!

Harsh Vadishi
Please upload notebook link in canvas.

Subhodeep Mitra
will this recording of the session available soon , I missed the beginning

Anvegh Singhal
ok mam

Varanasi Shivani
mam I've joined late. Can you pls tell if these notebook links are included in the ppt?

Type here to search

ENG 20:53 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[3] import matplotlib.pyplot as plt
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
```

```
from google.colab import files
uploaded = files.upload()
import io
data = pd.read_csv(io.BytesIO(uploaded['diabetes.csv']))
data.head()
```

Choose Files diabetes.csv

- **diabetes.csv**(application/vnd.ms-excel) - 23873 bytes, last modified: 9/21/2019 - 100% done

Saving diabetes.csv to diabetes.csv

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1

0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Chat Participants (70)

Disable group chat

mam we are not sure on 16 if linear will be discuss or not. also may postpobe. and assignment is on 1 st jan. so if possible kindly share notebook on linear regression so that we can get the idea and start assignment

is linear SVM same as linear regression ?

Diabetes.csv file where can we find

The python file is in your google colab

can we also attach the requirements file to avoid any dependency conflict

Ahh .... kinda clustering

Features could change from different sources

Type here to search

ENG 20:55 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

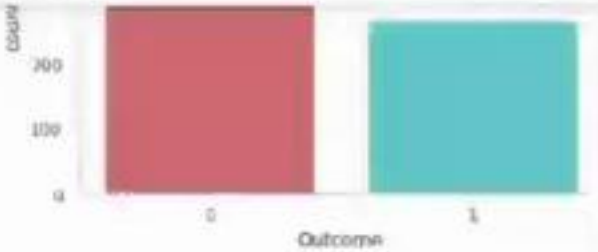
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

Editing

[6]



[7]

```
# Logistic Regression
class LogitRegression() :
    def __init__ ( self, learning_rate, iterations ) :
        self.learning_rate = learning_rate
        self.iterations = iterations

    # function for model training
    def fit( self, X, Y ) :
        # no_of_training_examples, no_of_features
        self.m, self.n = X.shape
        # weight initialization
        self.w = np.zeros( self.n )
        self.b = 0
        self.X = X
        self.Y = Y
```

0s comp

s.amperlas.com is sharing your screen. Stop sharing Hide

Chat Participants (70)

Disable group chat

mam we are not sure on 18 if linear will be discuss or not..also may postpone, and assignment is on 1 st jan. so if possible kindly share notebook on linear regression so that we can get the idea and start assignment.

is linear SVM same as linear regression ?

Diabetes.csv file where can we find

The python file is in your google colab

can we also attach the requirements file to avoid any dependency conflict

Ahh ... kinda clustering

Features could change from different sources

Type here to search

ENG 20:55 14-12-2021

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

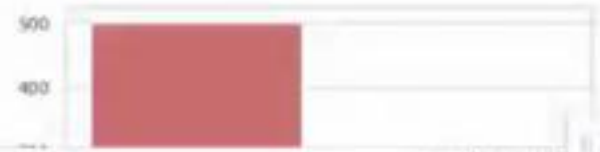
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

[4]

[5]
```
data = data.dropna()
print(data.shape)
print(list(data.columns))
```

```
(768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Out
```

[6]
```
data['Outcome'].value_counts()
sns.countplot(x='Outcome',data=data, palette='hls')
plt.show()
```



0s comp

a.impartus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Chat Participants (70)

Disable group chat

mam we are not sure on 16 if linear will be discuss or not. also may postpobe. and assignment is on 1 st jan. so if possible kindly share notebook on linear regression so that we can get the idea and start assignment.

is linear SVM same as linear regression ?

Diabaties.csv file where can we find

The python file is in your google colab

can we also attach the requirements file to avoid any dependency conflict

Ahh ... kinda clustering

Features could change from different sources

Type here to search

ENG 20:56 14-12-2021