

Gradient_Descent_Logistic X | sklearn.linear_model.LogisticRegression X | Impartus X | Virtual Classroom X | (1) WhatsApp X

← → C ⌂ webwhatsapp.com

Apps e-Filing Home Page... IBM Quantum Experience Qiskit textbook Sankar's NMIMS de... Defining Quantum... informs tutorial de... Other bookmarks Reading list

Machine Learning at BITS
Chaitanya, Divya, Rupam, +91 90722 01448, +91 81486 79345, +91 97933 23705, +91 99123 47176, +91 99001 06367, you

Get notified of new messages. Turn on desktop notifications?

Search or start New chat

Machine Learning at BITS

+91 99123 47176 7:31 am

Close your whatsapp numm

+91 97132 44888 7:31 am

Ok in 1 am. no problem

Akash 7:31 am

This is What Sri Sri Prabhupada has given ..

+91 99306 58087 7:31 am

Nothing don't worry

Technical Project 7:31 am

Machine Learning at BITS

+91 99123 47176 7:31 am

go to aimpartus.com and use the above login and password.

It's still not open to log in

+91 99123 47176 7:31 am

Yes it will start at 7:30 PM sharp.

earlier it used to open before 5 min

For sharing

+91 99123 47176 7:31 am

These days they allow a login only at the appointed time.

I UNREAD MESSAGE

+91 99123 47176 7:31 am

You can start now ..

aimpartus.com is sharing your screen.

Stop sharing Hide

Gradient_Descent_Logistic.ipynb

File Edit View Insert Runtime Tools Help

+ 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

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

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

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

importus.com is sharing your screen.

Stop sharing

Hide

Gradient_Descent x skleanlinear x Inventory x Virtual Disk x WhatsApp x

colab.research.google.com/drive/1Dap6j.../vVvMkDpG1wmWMiPXWcwam1vkscrollToomerHCjog... New window
by Start

Apps e-Filing Home Page IBM Quantum Espe... Qiskit textbook Sivam's NMIMS dee... Other documents Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

+ Code + Text

RANGE 1 DAY 300 Filtering

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

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

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

$$\frac{p}{1-p}$$

airpartus.com is sharing your screen. Stop sharing Hide

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Impartus X Virtual Classroom X

colab.research.google.com/drive/1Dsp6j7jVVVvMkDpG1wnWmJiFXwicwam1y#scrollTo=mwhCjogPjy

Apps e-Filing Home Page IBM Quantum Espe... Qiskit textbook Svm's NMMIS de... Defining Quantum... informs tutorial de... 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

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

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

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

Impartus.com is sharing your screen. Stop sharing more

Gradient_Descent_Logistic_Reg... x skitstutorial_modelling.ipynb x Importus x Virtual Classroom x

colab.research.google.com/drive/1DspIj7gVVVtMkDpG1wnWmfxWowam1v#scrollTo=mwhCjogkPjv

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Svm's NMIMS dee... Defining Quantum... informs tutorial dee... Other bookmarks Reading list

Gradient_Descent_Logistic.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

+ Code + Text

RAM 1 GB

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

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

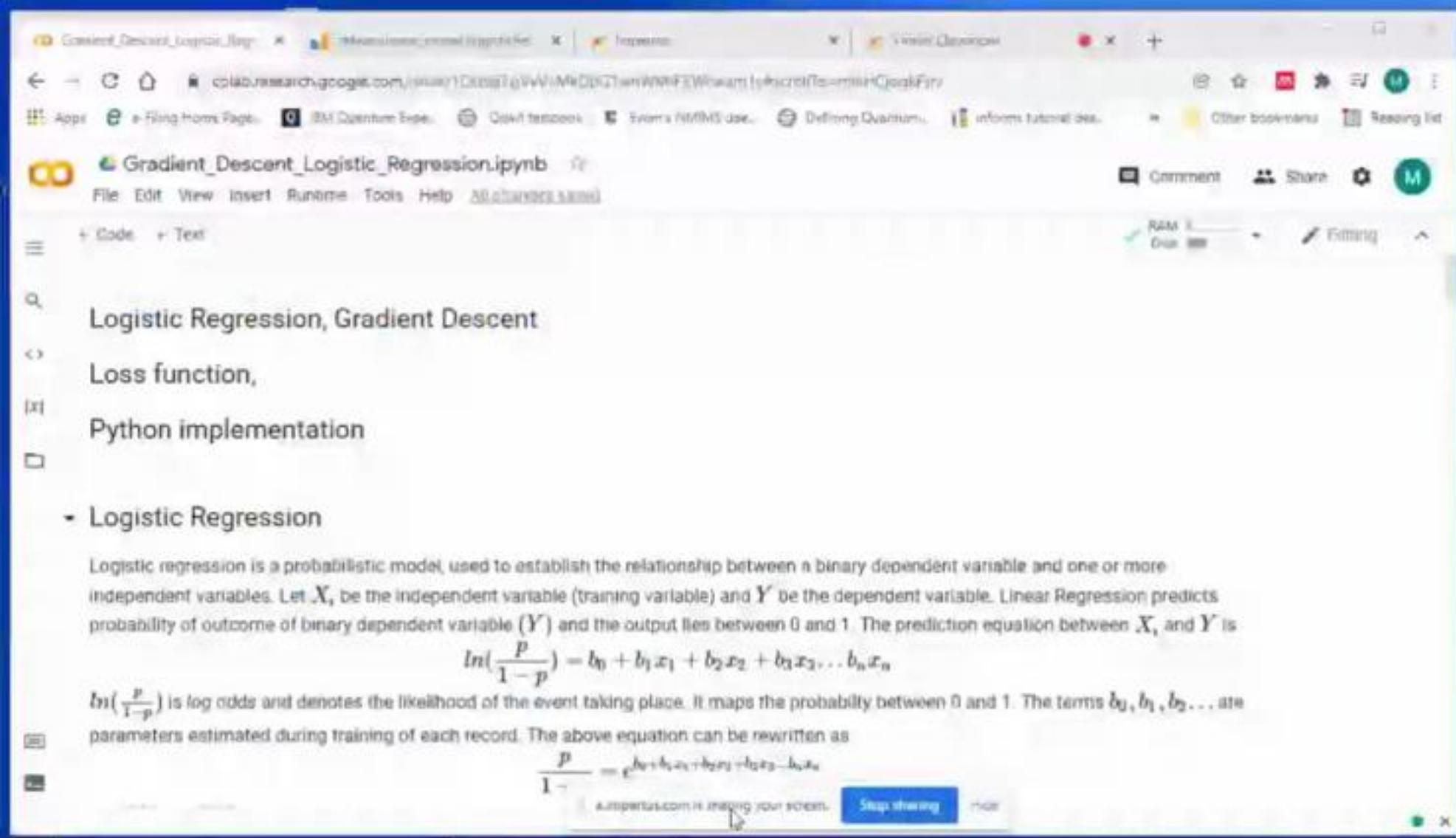
$\ln\left(\frac{p}{1-p}\right)$ is log odds and denotes the likelihood of the event taking place. It maps the probability between 0 and 1. The terms $b_0, b_1, b_2 \dots$ are parameters estimated during training of each record. The above equation can be rewritten as

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

a-impartus.com is sharing your screen.

Stop sharing

Hide



Gradient_Descent_Logistic_Reg... x colab.research.google.com/drive/1Dpus... x Impartus x

← → C O colab.research.google.com/drive/1Dpus.../.../VhMzDpiG1emWlfFjWosamI/#scrollTo=mmtCjogDwv

Apps e-Filing Home Page IBM Quantum Espl... Qubit notebook Srim's RIMMS des... Defining Quantum... inform tutorial des... Other bookmarks Reading list

Gradient_Descent_Logistic.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

a. 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

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

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

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

a-impartus.com is sharing your screen.

Stop sharing

Hide

Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

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

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

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

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

simplifying it further

$$P = \frac{e^{b_0+b_1x_1+b_2x_2+b_3x_3\dots+b_nx_n}}{1 + e^{b_0+b_1x_1+b_2x_2+b_3x_3\dots+b_nx_n}}$$

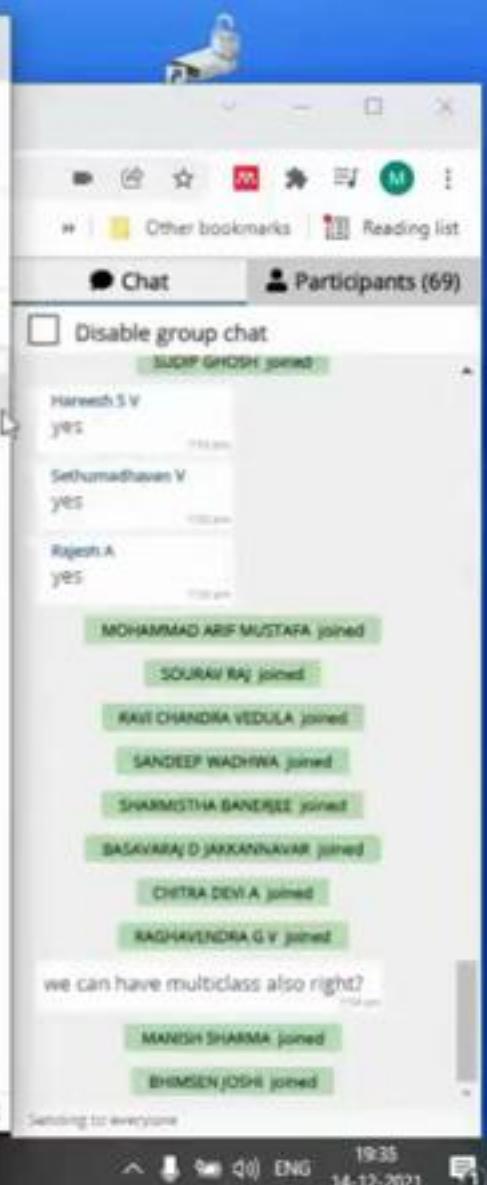
or

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

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

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

airpatus.com is sharing your screen.





Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

$$\frac{1}{1+e^{-x}}$$

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

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

Deriving the sigmoid function,

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

Using reciprocal rule, then rule of linearity,

$$\begin{aligned}&= -(1+e^{-X})^{-2} \cdot \frac{d}{dX} (1+e^{-X}) \\ &= -(1+e^{-X})^{-2} \cdot \left(\frac{d}{dX}[1] + \frac{d}{dX}[e^{-X}]\right)\end{aligned}$$

Derivative of constant is 0,

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

aiimpartus.com is sharing your screen.

Stop sharing

Hide

Comment Share M

RAM Disk Editing

Chat Participants (75)

Disable group chat

RAVICHANDRA VEDULA joined

SANDEEP WADHWALA joined

SHARMISTHA BANERJEE joined

DASAVAIYA DJAGGANATH joined

CHITRA DANTA joined

RASHAVENDRA G V joined

We can have multiclass also right?

MANISH SHARMA joined

BHIMSEN JOSHI joined

NEERAJ SABHANI joined

ALOK joined

BONKALAMAN DINISHI joined

GIRI TANJALI RAI joined

ONE CLASS IS USED FOR MULTICLASS IN LOGISTIC REGRESSION

SAHOO SUJIT KUMAR joined

GEETA RAMANI SALK PRASADA joined

Sending to everyone

Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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 $b_0, b_1, b_2 \dots$ 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 gradient descent only converge convex function into a global minimum.

$$J = -y \log(h(x)) - (1-y) \log(1-h(x))$$

here, y is the real target value

$$h(x) = \text{sigmoid}(w \cdot x + b)$$

For $y = 0$,

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


Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help all changes saved

+ Code + Text

$$\begin{aligned}
 &= -(1 + e^{-X})^{-2} \cdot \frac{d}{dX}(1 + e^{-X}) \\
 &= -(1 + e^{-X})^{-2} \cdot \left(\frac{d}{dX}[1] + \frac{d}{dX}[e^{-X}] \right) \\
 \text{Derivative of constant is 0,} \\
 &= -(1 + e^{-X})^{-2} \cdot (0 + e^{-X} \frac{d}{dX}[-X]) \\
 &= (1 + e^{-X})^{-2} \cdot e^{-X} \\
 &= \frac{e^{-X}}{(1 + e^{-X})^2} \\
 &= \frac{1}{(1 + e^{-X})} \cdot \frac{e^{-X} + 1 - 1}{(1 + e^{-X})} \\
 &= \frac{1}{(1 + e^{-X})} \cdot \left(\frac{1 + e^{-X}}{1 + e^{-X}} - \frac{1}{1 + e^{-X}} \right) \\
 &= \frac{1}{(1 + e^{-X})} \cdot \left(1 - \frac{1}{1 + e^{-X}} \right) \\
 &= \sigma(X) \cdot (1 - \sigma(X))
 \end{aligned}$$

aimportas.com is sharing your screen. Stop sharing Hide



Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Derivative of constant is 0,

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


aImpartus.com is sharing your screen.

Stop sharing Hide

Chat Participants (78)

Disable group chat

BASAVARAJA KANNAN joined
CHITRA DEVI A joined
RAGHAVENDRA G V joined
we can have multiclass also right?
MANISH SHARMA joined
BHIMSEN JOSHI joined
NEERAJ SASHIKANT joined
ALOK joined
BOHRE AAMAN DINISH joined
GESTANGALI RAJ joined
one vs rest is used for multiclass in logistic regression
SAHOO SUJIT KUMAR joined
GEETA RAMANI BALA PRASADA joined
ASHU joined
DEENATH BIBI joined
JYATHERTHA BABACHANDRA KATI joined

19:39 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb

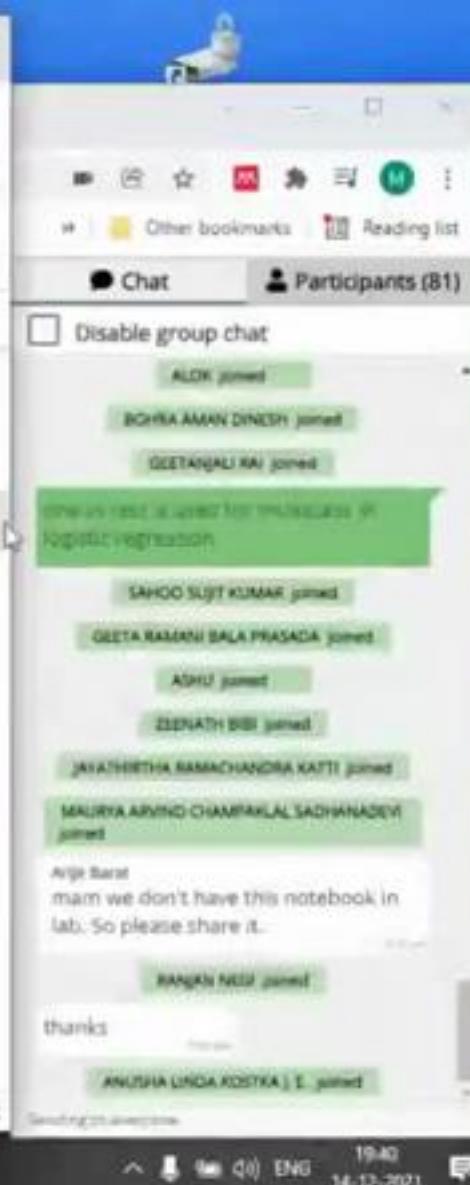
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

$$\begin{aligned}
 & \frac{(1+e^{-x}) - (1+e^{-X})}{(1+e^{-X})} \\
 &= \frac{1}{(1+e^{-X})} \cdot \left(\frac{1+e^{-X}}{1+e^{-X}} - \frac{1}{1+e^{-X}} \right) \\
 &= \frac{1}{(1+e^{-X})} \cdot \left(1 - \frac{1}{1+e^{-X}} \right) \\
 &= \sigma(X) \cdot (1 - \sigma(X))
 \end{aligned}$$


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 $b_0, b_1, b_2 \dots$ that result in least error. This

aimportus.com is sharing your screen. Stop sharing Hide



Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X importus X

colab.research.google.com/drive/1Dsp6j7jVvVvMkDpG1wmWMiFXwcam1v#scrollTo=zmwH... Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Sixt's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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

(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 $b_0, b_1, b_2 \dots$ 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. They

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

Type here to search

Comment Share M

RAM Disk

Editing

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

DEEPAK BISHI joined

JAYATHGEETHA RAMACHANDRA KATTI joined

MALIBYA ARVIND CHAMPAKAL SADHANADEV joined

Arijit Banerjee

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

KANAN NEGI joined

thanks

ANUSHA LINDA KOSTKA joined

AMOGH SINGHAL joined

SARVESHWARAN K joined

Sharing to everyone

19:41 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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

here, $a = \text{sigmoid}(z)$ 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)

# Importing the dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.arff"
df = pd.read_arff(url)
print(df.shape)
print(df.head())
print(df.describe())
print(df.groupby('species').size())

# Separating X and y
X = df.drop(['species'], axis=1)
y = df['species']

# Standardizing the features
scaler = preprocessing.StandardScaler()
X = scaler.fit_transform(X)

# Splitting the data into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

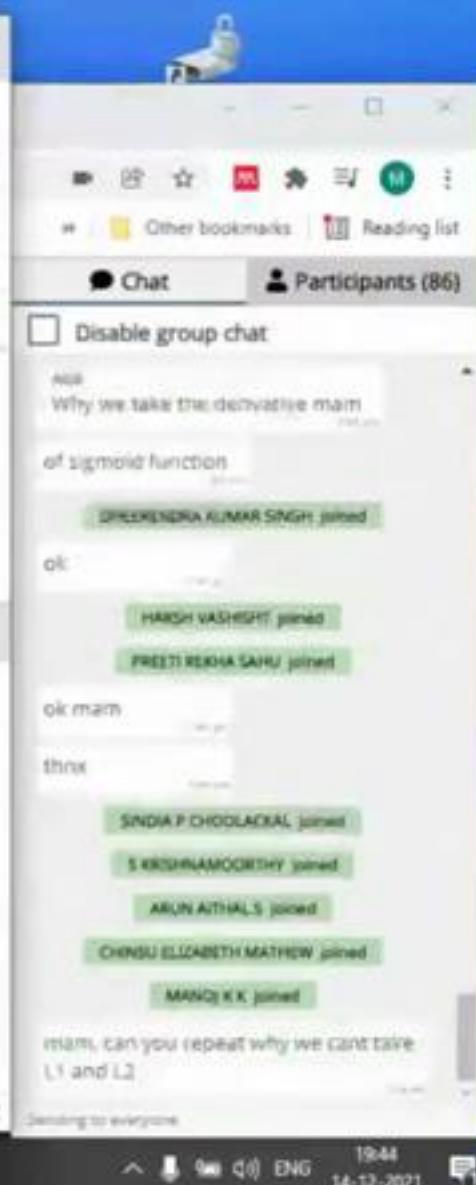
# Fitting the logistic regression model
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)

# Predicting the classes for the test set
y_pred = logreg.predict(X_test)

# Evaluating the model's performance
from sklearn.metrics import classification_report, confusion_matrix
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

# Visualizing the results
plt.figure(figsize=(10, 6))
sns.heatmap(logreg.coef_, annot=True, cmap='viridis')
plt.title('Coefficient matrix for Logistic Regression')
plt.show()
```

Stop sharing Hide



Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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, $a = \text{sigmoid}(z)$ 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()
```

Stop sharing Hide



Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X impartus X

colab.research.google.com/drive/1Dsp6j7jVvVvMkDpG1wmWMiFXWcwam1v#scrollTo=-5hZdb3o...

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Svm's NMIMS dee...

Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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

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

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



Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X importus X

colab.research.google.com/drive/1Dsp6j7sjVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=-5hZdb3o... Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Sivam's NMIMS dee... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

[x]

[7]: # Logistic Regression
class LogisticRegression():
 def __init__(self, learning_rate=0.001, iteration=1000):

Stop sharing Hide



Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticReg... X impartus X

colab.research.google.com/drive/1Dsp67zjVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=-5hZdb3o... Apps e-Filing Home Page IBM Quantum Espe... Qiskit textbook Svkrm's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM: Disk: Editing

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

(x)

count:

Outcome

```
[?] # Logistic Regression
class LogisticRegression():
    def __init__(self, learning_rate, iterations):
        self.learning_rate = learning_rate
        self.iterations = iterations
        self.theta = None
        self.m = None
        self.n = None
        self.J_history = None
```

Os comp

Stop sharing Hide

Impartus is sharing your screen.

Chat Participants (86)

Disable group chat

PREETI REKHA SAHJ joined

ok mam

thnx

SINDHU P CHODLACKAL joined

S KRISHNAMOORTHY joined

ARUN ATHAL 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

SAHUL SAHA joined

ABHAY RAMESH JUNGSHARE joined

K GURISH GOPINATHAN joined

Sending to everyone

Type here to search

19:49 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRe X Importus X

colab.research.google.com/drive/1Dsp67sjVvVvMkDpG1wnWMiFXwam1v#scrollTo=dtCilyLyM... 89

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Sivam's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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

The chart is a bar plot titled "sns.countplot(x='Outcome', data=data, palette='magma')". The x-axis is labeled "Outcome" and has two categories: 0 and 1. The y-axis is labeled "count" and ranges from 0 to 500. Category 0 is represented by a red bar reaching approximately 500 on the y-axis. Category 1 is represented by a teal bar reaching approximately 250 on the y-axis.

Importus

Other bookmarks Reading list

Chat Participants (87)

Disable group chat

using MSE

Arijit Basar
lease square error

BAHUL SANA joined

ABHAY RAMESH JUNGHARE joined

K GRISS GOPINATHAN joined

Arik
how they are different

model 1 and Model 2

Rakhi Hande
Yes Ma'am

Sekhar Reddy
yes mam,

Ganesakaran R
Mam, are you balancing the data points?

S Bhagath
what will be model(2)?

Sending to everyone

19:51 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRe... X Impartus X

colab.research.google.com/drive/1Disp67sjVvVvMkDpG1wnWMiFXYwcam1v#scrollTo=dtCityLyM... Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Swami's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[6] In [0] Outcome

```
# Logistic Regression
class LogisticRegression():
    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
```

RAM Disk Editing

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

Other bookmarks Reading list

Chat Participants (86)

Disable group chat

ABHAY RAMESH JUNGRAH joined

A LOKESH GOENKAL joined

Ask how they are different

model 1 and Model 2

RukhiNandy Yes Mam

Sekhar Reddy Mam... yes mam,

GuruSekhar R. Mam, are you balancing the data points?

SilpaRith what will be model2?

Ask Both are scikit learn Model ?

SL3 N V V S Subbarao do we need to maintain the balanced data for regression case?

Sending to everyone

19:51 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Impartus X

colab.research.google.com/drive/1Dspfj7sVVvVvMkDpG1wmWMiFXWcwam1v#scrollTo=dtCillyyM... 89

Apps e-Filing Home Page... IBM Quantum Expe... Qiskit testbook Svkrm's NMIMS de...

Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[6] 0 0 1

Outcome

```
# Logistic Regression
class LogisticRegression():
    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

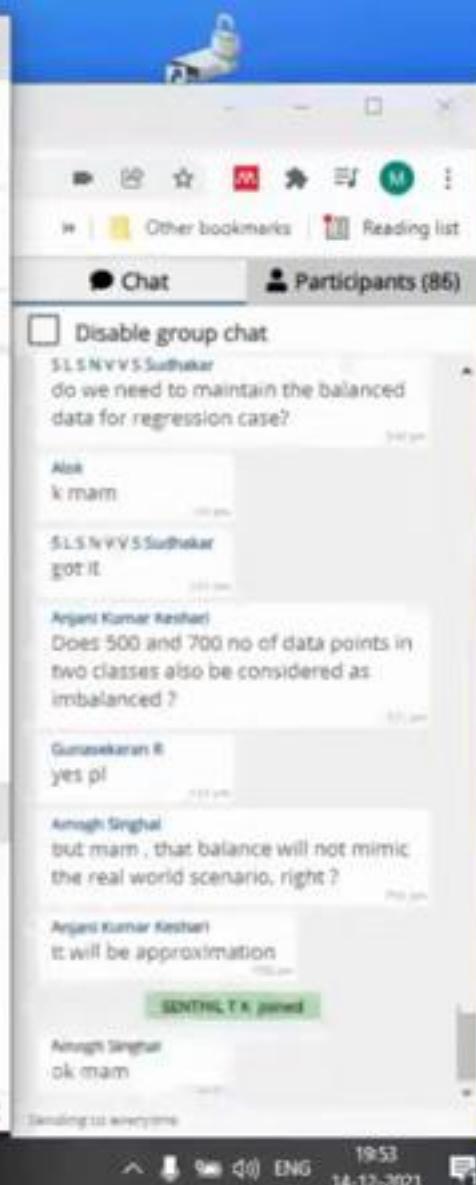
        # gradient descent learning

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

    def update_weights(self):
        # calculate hypothesis
        hypothesis = self.X.dot(self.W) + self.b
        error = hypothesis - self.Y
```

RAM: ✓ Disk: Editing

aaimpartus.com is sharing your screen. Stop sharing Hide



Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRegr X Impartus X

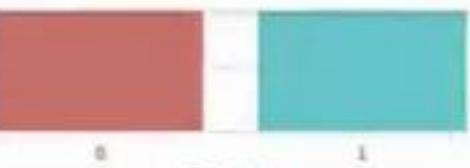
colab.research.google.com/drive/1Dsp6j7jVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=dtCBylyM... 29

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Sivam's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[6] 

```
[7] # Logistic Regression
class LogisticRegression():
    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

Other bookmarks Reading list

Chat Participants (85)

Disable group chat

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

Alok
ok mam

SLS N V V S Sudhakar
got it

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

Gunesekaran R
yes pl

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

Anjali Kumar Kashish
it will be approximation

SENTHIL T K joined

Anmol Singhvi
ok mam

Sending to everyone

19:54 14-12-2021

Gradient_Descent_Logistic_Reg X colab.research.google.com/drive/1DspUj7zjVvvMkDpG1wnWMiEXWowam1v#scrollTo=IzyaaQM... 29 ★ M

Apps e-Filing Home Page IBM Quantum Espe... Qiskit textbook Sivam's NMIMS dee...

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
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.iaptus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Comment Share M

RAM Disk Editing

Chat Participants (85)

Disable group chat

Alok
k mam

SLSNVVSSubhakar
got it

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

Gunesekaran R
yes pl

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

Arjan Kumar Kashuri
it will be approximation

SANTHILK joined

Amogh Singhvi
ok mam

Sanjeev Kumar
why model1 is required using sklearn?

Sending to everyone

19:55 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X importus X

colab.research.google.com/drive/1Dsp6j7zjVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=jzyaaQ... 89 ★ 20 M :

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Svm's NMIMS dee... 9 Other bookmarks 10 Reading list

Gradient_Descent_Logistic_Regression.ipynb

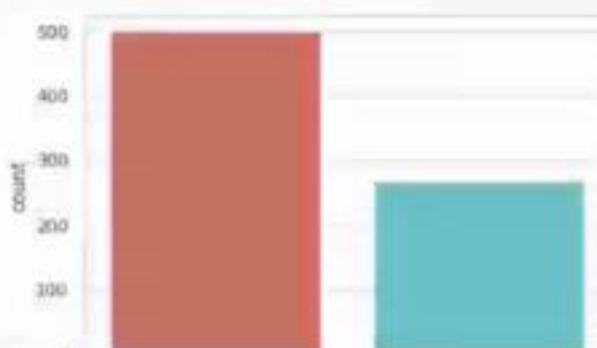
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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

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



aiimportus.com is sharing your screen. Stop sharing Hide

Chat Participants (85)

Disable group chat

SLSNVYVSSuhalak Got it 10:51 pm

Anupam Kumar Kashuri Does 500 and 700 no of data points in two classes also be considered as imbalanced ? 10:51 pm

Gunasekaran R yes pl 10:51 pm

Amogh Singhvi but marn , that balance will not mimic the real world scenario, right ? 10:51 pm

Anupam Kumar Kashuri It will be approximation 10:51 pm

SENTHIL TK joined

Amogh Singhvi ok marn 10:51 pm

Sergeev Kumar Why model1 is required using sklearn? 10:51 pm

1st line is related to sigmoid function? 10:51 pm

Sending to everyone

Gradient_Descent_Logistic_Reg X sklean.linear_model.LogisticReg X | importus X

colab.research.google.com/drive/1Dsp6j7jVvvMkDpG1wnWMiFXWcwam1v#scrollTo=IzyaaQM... Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Svm's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# 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 * dw
    self.b = self.b - self.learning_rate * db
```

RAM: Disk: Editing

Stop sharing Hide

importus.com is sharing your screen.

Chat Participants (86)

Disable group chat Got it

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

Gunasekaran R yes pl

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

Anjan Kumar Kashar it will be approximation

SENTHIL T K joined

Amogh Singh ok mam

Sergeev Kumar Why model1 is required using sklearn?

1st line is related to sigmoid function?

ARAVIND P joined

Sending to everyone



Type here to search



1957

14-12-2021

Gradient_Descent_Logistic_Reg.ipynb

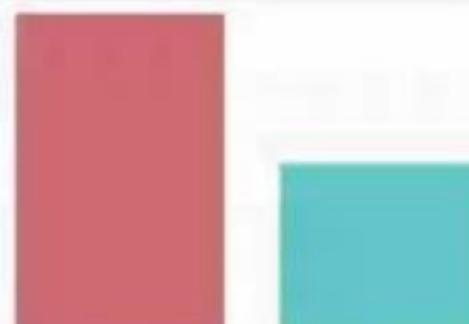
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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

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



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

Comment Share M

RAM Disk Editing

Other bookmarks Reading list

Chat Participants (86)

Disable group chat

Dot It

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

Guneswaran R yes pl

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

Anjali Kumar Kashuri it will be approximation

SENTHIL Y K joined

Amogh Singhvi ok mam

Sergeev Kumar why model1 is required using sklearn?

1st line is related to sigmoid function?

AAVINO P joined

Sending to everyone

Stop sharing Hide

1957 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
self.b = self.b - self.learning_rate * db
return self

# hypothetical function h(z)

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 = LogisticRegression( learning_rate = 0.01, iterations = 50 )
```

Stop sharing Hide

alimpartus.com is sharing your screen.

Comment Share M

RAM Disk Editing

Other bookmarks Reading list

Chat Participants (86)

Disable group chat

Dot It

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

Gunesekaran R yes pl

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

Anjan Kumar Keshari It will be approximation

SENTHIL T K joined

Amogh Singh ok mam

Sergeev Kumar why model1 is required using sklearn?

1st line is related to sigmoid function?

AMARIND P joined

Sending to everyone

19:58 14-12-2021

Gradient_Descent_Logistic_Reg... x colab.research.google.com/drive/1Dispj7jVvvVtVkpG1wmWkfFXYowam1v#scrollTo=izyaaQM... +

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
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
```

Stop sharing Hide

Comment Share M

RAM Disk Editing

Other bookmarks Reading list

Chat Participants (88)

Disable group chat

Suresh Kumar R
yes pl

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

Argya Kumar Dasgupta
it will be approximation

SANTHIL K joined

Amogh Singhal
ok mam

Suresh Kumar
why model1 is required using sklearn?

1st line is related to sigmoid function?

ARAVIND P joined

Alok
where we are calculating error in our function mam ..

ANSHUJU PUM joined

Sharing to everyone



Type here to search

19:58
14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Importus X

colab.research.google.com/drive/1Dsp6j7sjVvVbMkDpG1wmWMiFXWowam1v#scrollTo=izyaaQM... Apps e-Filing Home Page... IBM Quantum Exp... Qiskit textbook Svkrm's NMIMS de...

Comment Share M

Gradient_Descent_Logistic_Regression.ipynb

All changes saved

+ Code + Text

[6] count

RAM Disk

Editing

```
# Logistic Regression
class LogisticRegression():
    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 importus.com is sharing your screen.

Stop sharing Hide

Other bookmarks Reading list

Chat Participants (88)

Disable group chat

Gunesekaran R
yes pl

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

Arijani Kumar Basu
it will be approximation

SENTHIL TK joined

Amogh Singhul
Ok mam

Sangeeta Kumar
why model1 is required using sklearn?

first line is related to sigmoid function?

ARAVIND P joined

Ask
where we are calculating error in our function mam ..

ANCHALI PURI joined

Sending to everyone

19:59 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help

```
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 = LogisticRegression(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
```

Open Bookmarks Reading list

Chat Participants (87)

Disable group chat

Why model1 is required using sklearn?

1st line is related to sigmoid function?

MAHENDRA BHAKTA

Where we are calculating error in our function main -

ANISHA PURI joined

but we are not comparing with real output

MANOJ SINGH MAHENDRA BHAKTA joined

yes main

RAQUELLAH KHAN joined

Sessions Session History (You can't hear me)

Bonita Ahmed Ghosh No Audio

Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Run Cell 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 = 1/3, random_state = 0)

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

    model.fit(X_train, Y_train)
    model1 = LogisticRegression(solver='liblinear', max_iter=100)
    model1.fit(X_train, Y_train)

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

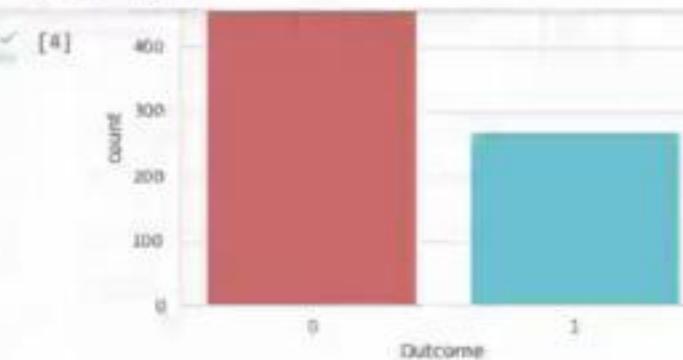
    # measure performance
    correctly_classified = 0
```



Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

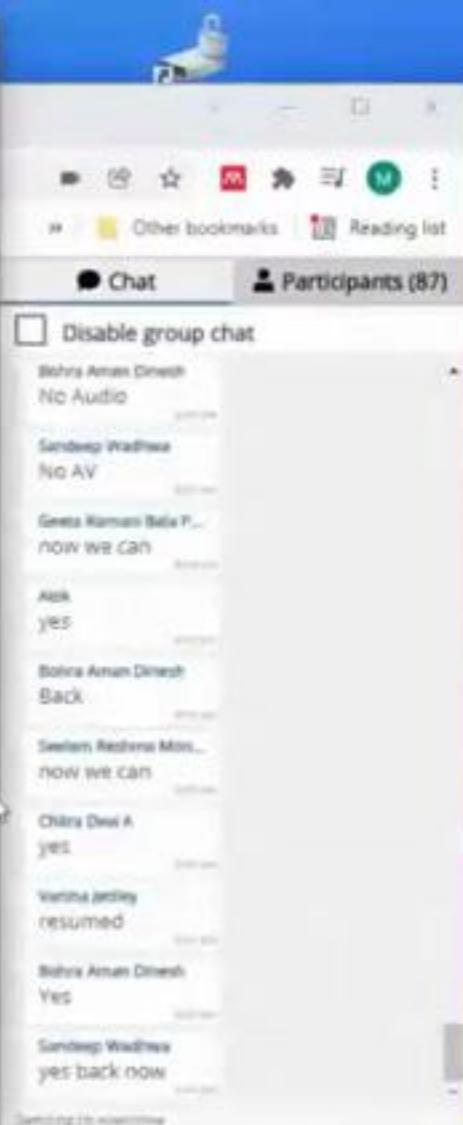
+ Code + Text

[4] 

[?] # logistic Regression

```
class LogisticRegression():
    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
```



Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRe... X Importus X

colab.research.google.com/drive/1DspIj7jVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=dtCiBylyM... Apps e-Filing Home Page IBM Quantum Expe... Qiskit textbook Sivam's NMIMS dee... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
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      : 63.54166666666666
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,
```

RAM Disk Editing

Bohra Arman Dinesh No Audio

Sandeep Wadhwa No AV

Geeta Ramani Balaji P... now we can

Ask yes

Bohra Arman Dinesh Back

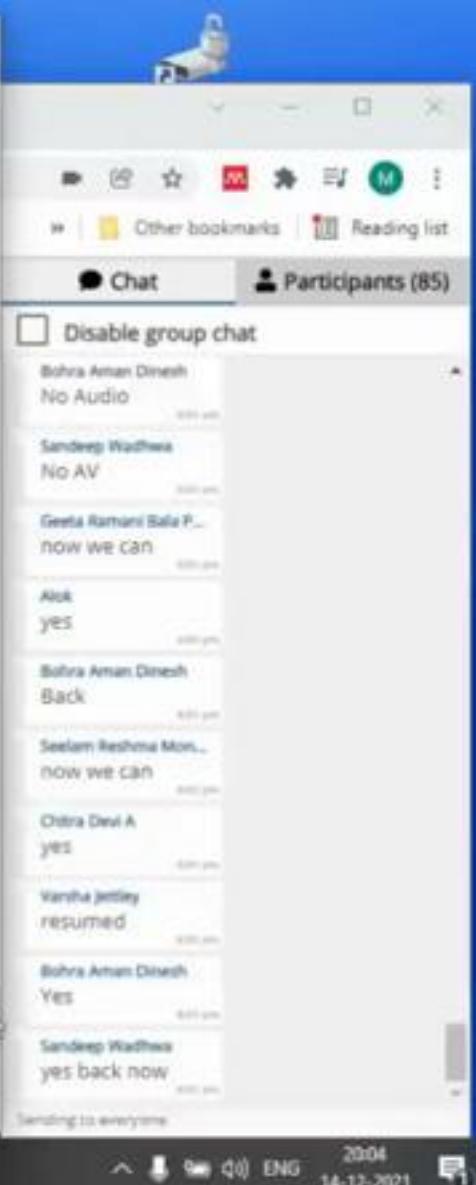
Seelam Reshma Mon... now we can

Chitra Devi A yes

Vanisha Jettley resumed

Bohra Arman Dinesh Yes

Sandeep Wadhwa yes back now



Gradient_Descent_Logistic_Reg X sklearn.linear_model.logisticRe... X Importus X

colab.research.google.com/drive/1DspI67jVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=dtCiBylyM... Apps e-Filing Home Page IBM Quantum Expe... Qiskit textbook Sivam's NMIMS dee... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

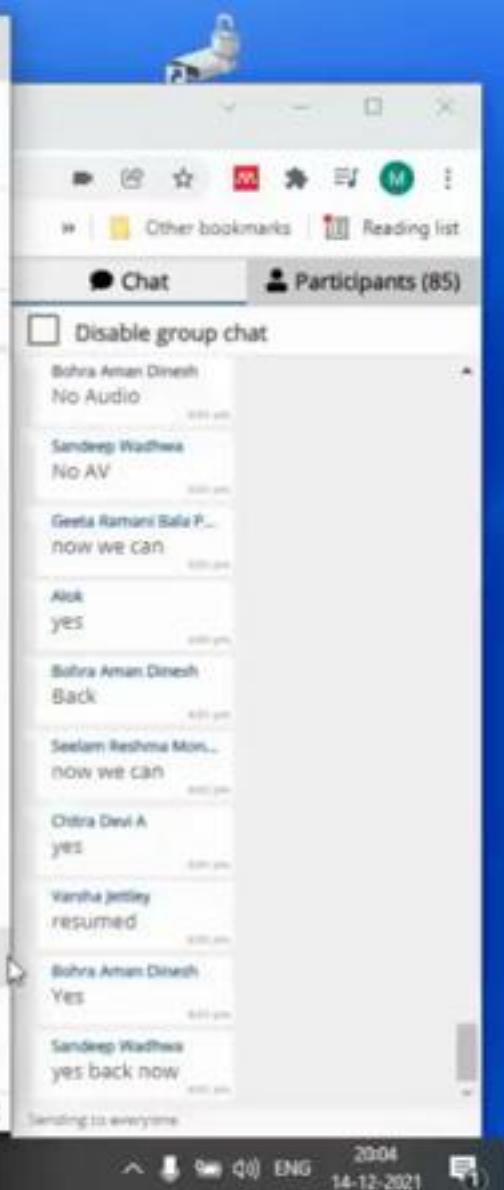
+ Code + Text

```
model = LogisticRegression( 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
```

RAM Disk Editing

Bohra Arman Dinesh No Audio 0:01 pm
Sandeep Wadhwa No AV 0:01 pm
Geeta Ramani Balaji P... now we can 0:01 pm
Ask yes 0:01 pm
Bohra Arman Dinesh Back 0:01 pm
Seelam Reshma Mon... now we can 0:01 pm
Chitra Devi A yes 0:01 pm
Vanisha Jettley resumed 0:01 pm
Bohra Arman Dinesh Yes 0:01 pm
Sandeep Wadhwa yes back now 0:01 pm

Stop sharing Hide



Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRe... X importus X

← → C ⌂ colab.research.google.com/drive/1Dsp6j7zjVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=dtCillyM... ⌂ ⌂ M ⌂

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Sivam's NMIMS dee... ⌂ Other bookmarks ⌂ Reading list

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_classified / count ) * 100 )

if __name__ == "__main__":
    main()

Accuracy on test set by our model : 63.54166666666666
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,
```

RAM: Disk: Editing

aimportus.com is sharing your screen. Stop sharing Hide

Chat Participants (86)

Disable group chat

yes

Gaurav Jaiswal resumed

Santosh Anand Direct Yes

Sandeep Mehta yes back now

AAKANSHA JAIN joined

Rahul Saha check the indentation

Indent

AGASTYA RAMANI KRISHNA KUMAR joined

Alok 11.30?

Sorry

Wrong ping

Ravi Chandra Venkata 11.45 25% or 20%

Sending to everyone

2007 ENG 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
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,
```

Impartus

Comment Share M

RAM: Disk: Editing

Other bookmarks Reading list

Chat Participants (85)

Disable group chat

SLSNVVS Sudhakar
change LR

Anup Singh
regularization

Ram Chandra Verma
How to decide on no of iterations for
good fit

Siddhagya
learning rate

Ram Saha
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

Debuger Chatbox:
on what params do we select the
right solver

Sending to everyone

Type here to search

Stop sharing Hide

OS comp

20:19 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Impartus X

colab.research.google.com/drive/1DspIg7sjVvvMxDpG1wnWMiPXwam1v#scrollTo=dtCillyM... ☆ M

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Svm's NMIMS dee... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
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,
```

Impartus is sharing your screen. Stop sharing Hide

RAM Disk Editing

Comment Share M

Chat Participants (84)

Disable group chat

change LR

Amogh Singhvi regularization

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

S.Bhagat learning rate

Rahul Saha yes

Aakar 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 solve!

Sending to everyone

20:10 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[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
```

Comment Share M

RAM Disk Editing

Disable group chat

Anush Singh regularization

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

S Bhagat 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

Debjyoti Chakraborty on what params do we select the right solver

sklearn solver

ROUDRAV CHAKRABORTY joined

Sending to everyone

Gradient_Descent_Logistic_Reg... X | sklearn.linear_model.LogisticReg... X | Impartus X

colab.research.google.com/drive/1Dsp6j7gVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=dtCibLyM... M

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Svkms NMIMS dee... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[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 = LogisticRegression( learning_rate = 0.001, 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 )
```

Impartus

Comment Share M

RAM Disk Editing

Disable group chat

Gauta Ramani Rata Pramoda increase the iterations

SLSN N V VSSubhakar increase Epoch

Alok may be weights are not updated in 100

Rahul Saha It didn't reached the minima

S. Bhagath stopped very early...

Gauta Ramani Rata P... to 1000

Argya Kumar Senthil 1000

Alok 1000.7

S. Bhagath 1000

Sending to everyone

Stop sharing Hide

Os comp

2013 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticReg... X Impartus X

colab.research.google.com/drive/1Dsp6j7gVvVvMkDpG1wnWMiFXwam1v#scrollTo=dtCillyM... ☆ M

Apps e-Filing Home Page IBM Quantum Espe... Qiskit testbook Sivam's NMIMS dee... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

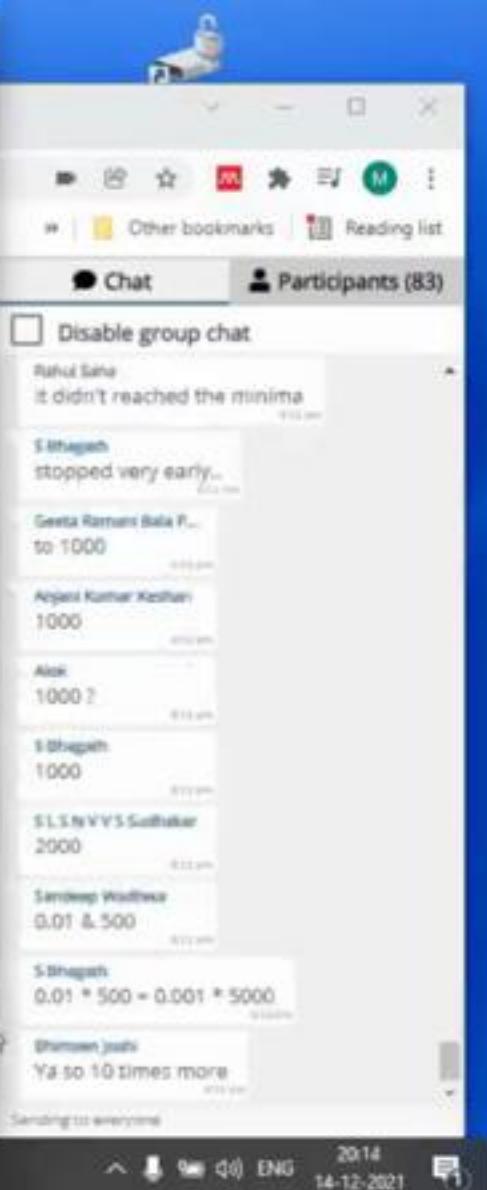
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

Accuracy on test set by our model      57.5
Accuracy on test set by sklearn model  79.16066066666666
/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,
```



Gradient_Descent_Logistic_Reg... x colab.research.google.com/drive/1Dsp67jzVvVvMxDpG1wnWMiFXwcam1v#scrollTo=atCityLyM... 8 | +

← → C ⌂ colab.research.google.com/drive/1Dsp67jzVvVvMxDpG1wnWMiFXwcam1v#scrollTo=atCityLyM... 8 | +

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Sivam's NMIMS dee... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help

+ 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 $b_0, b_1, b_2 \dots$ 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.

$J = -y \log(h(x)) - (1-y) \log(1-h(x))$

here, y is the real target value

$h(x) = \text{sigmoid}(w_0 + w_1 x)$

For $y > 0$.

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



Gradient_Descent_Logistic_Reg x sklearn.linear_model.LogisticReg x Impartus x

← → C ⌂ ⌂ solkit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps e e-Filing Home Page... IBM Quantum Exp... Qiskit testbook Svm's NMIMS de... Other bookmarks Reading list

learn

Home Up Home

scikit-learn 1.0.1

Do Not Disturb

Please cite us if you use this software

sklearn.linear_model.LogisticRegression

Example using

sklearn.linear_model.LogisticRegr

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:

Support Vector Classification

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

Other bookmarks Reading list

Chat Participants (82)

Disable group chat

Arijit Kumar Keshari lr = 0.005

Alok Loop thru

Gaurav Anand I remember we used Tou for learning rate

Arijit Kumar Keshari hypertuning

M R Vijay Krishnan Gridsearch

Random search ok

Alok 0.01 + 500

SLSN VVS Sudhakar can't we normalize the data?

ok

charmin Panigrahi for assignment can we use sklearn ?

Type here to search

2018 ENG 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRegr... X Impartus X

← → C ⌂ ⌂ scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps e e-Filing Home Page... IBM Quantum Expe... Qiskit testbook Svm's NMIMS de... Other bookmarks Reading list

learn Install User Guide API Examples More

Home Up News

sklearn 1.0.1

Other versions

Please cite us if you use the software.

sklearn.linear_model.LogisticRegression

Examples using sklearn.linear_model.LogisticRegression

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 solver.

Impartus.com is sharing your screen. Stop sharing Hide id by the saga solver

Chat Participants (82)

Disable group chat

Arijit Kumar Kachari
lr = 0.005

Alok
Loop thru

Gaurav Anand
I remember we used Tou for learning rate

Arijit Kumar Kachari
hypertuning

M R Vijay Krishnan
Gridsearch

Random search ok

Alok
0.01 + 500

SLSN VVSS Sudhakar
can't we normalize the data?

ok

Charmin Parangry
for assignment can we use sklearn ?

Sending to everyone

2018 ENG 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticReg X Impartus X

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

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Svm's NMIMS dee...

Other bookmarks Reading list

learn Install User Guide API Examples More

Prev Up Next

sklearn 1.0.1 Other versions

Please cite us if you use the software.

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

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 solver.

Impartus.com is sharing your screen. Stop sharing Hide

Type here to search

Chat Participants (82)

Disable group chat

Anjani Kumar Keshari
lr = 0.005

Alek
Loop thru

Gaurav Anand
I remember we used Tou for learning rate

Anjani Kumar Keshari
hypertuning

M R Vijay Krishnan
Gridsearch

Random search ok

Alek
0.01 + 500

SLSN VVSS Sudhakar
can't we normalize the data?

ok

Charan Panigrahi
for assignment can we use sklearn ?

Sending to everyone

2018 ENG 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticReg... X impartus X

← → C ⌂ scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps e e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Svkrm's NMIMS de... Other bookmarks Reading list

learn Install User Guide API Examples More

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.LogisticRegression

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 solver.

a-impartus.com is sharing your screen.

Stop sharing

Hide

id by the 'saga'

Toggle Menu

Type here to search



20:18 14-12-2021

Chat Participants (82)

Disable group chat

I remember we used Tou for learning rate

Arijit Kumar Sarker hypertuning

M R Vijay Krishnan Gridsearch

Random searchok

Abis 0.01 + 500

SLSN VVSSubhakar Can't we normalize the data?

OK

Chaitin Panigrahy for assignment can we use sklearn 7

Sharmeen Javed is 72 max we can get?

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

Sharing to everyone

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticReg X impartus X

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

Apps e-Filing Home Page IBM Quantum Expe... Qiskit textbook Svm's NMIMS de...

Other bookmarks Reading list

learn Install User Guide API Examples More

Prev Up Next

sklearn 1.0.1 Other versions

Please cite us if you use the software.

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

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 solver.

Impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (82)

Disable group chat

Random searchok

Nik 0.01 + 500

SLSN VV33dhsaur can't we normalize the data?

ok

Qurant Parapry for assignment can we use sklearn ?

Sharmeen Jadoh 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

Mulase Lakshmi Shw... Lasso and Ridge

Sending to everyone

2018 ENG 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticReg... X Impartus X

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

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Sivam's NMIMS de...

Other bookmarks Reading list

learn Install User Guide API Examples More

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.LogisticRegression

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 solver.

Impartus.com is sharing your screen. Stop sharing Hide

Chat Participants (83)

Disable group chat

Alex L1 and L2 are not covered

M R Vijay Krishna Ok

Mulisa Lakshmi Sow... Lasso and Ridge

SLS N V V S Sudhakar Those are

Arijit Sen not covered

Seelam Reshma Mon... no

Karthik Ramach... Not yet mam

Mulisa Lakshmi Sow... nope

Rahul Saha |lambda|/2

Alex No mam

Impartus.com is sharing your screen. Stop sharing Hide

2019 ENG 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRegr... X importus X

← → C O 🔍 colab-learn.org/file/cdrw/gouspmtm/vkMunJlinear_model.LogisticRegression.html

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Svm's NMIMS de... Other bookmarks Reading list

learn Install User Guide API Examples More

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',
                                             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 case.

importus.com is sharing your screen. Stop sharing Hide

Chat Participants (84)

Disable group chat

no

Jayashriha Ramach... Not yet mom

Mulisa Lakshmi Sow... nope

Rahul Saha |lambda|/2

Alok No mom

SLS N V V S Sudhakar LASSO

Arijit Berat mom why default solver is lbfgs?

KADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA DOWRI V G joined

Alok reached minima

7

Sending to everyone

20:22 ENG 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... Impartus X

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

Apps e-Filing Home Page... IBM Quantum Espe... Qisitit testbook Sivm's NMIMS dee...

Other bookmarks Reading list

learn Install User Guide API Examples More

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.LogisticReg

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 solver.

aimpartus.com is sharing your screen.

Stop sharing

Hide

Toggle Menu

Type here to search



20:23 14-12-2021

Chat Participants (83)

Disable group chat

Nehru Saha (lambda)/2

Alok No man

SLS NIVYAS Sudhakar Lasto

AKR Sankar man why default solver is lbfgs?

KADAVALU NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V B joined

Alok reached minima

?

Arjeet Rawat in assignment we can use scikit learn or we have to write our own functions?

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

Sending to everyone

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticReg... X Impartus X

← → C ⌂ ⌂ scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Svm's NMIMS de... Other bookmarks Reading list

learn Install User Guide API Examples More

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.LogisticRegression

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 solver.

Impartus.com is sharing your screen. Stop sharing Hide

Type here to search

Chat Participants (83)

Disable group chat

Alok No man 10:00 AM

E.S.N.V.V.S.S. Subbarao Lasso 10:00 AM

Arijit Banerjee man why default solver is lbfgs? 10:00 AM

YADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok reached minima 10:00 AM

Arijit Banerjee in assignment we can use scikit learn or we have to write our own function? 10:00 AM

Alok Does this mean... Penalty L2 is being used on the top of Gradient Descent 10:00 AM

Shreya Joshi Seems there is no effect of over fit 10:00 AM

Sending to everyone

20:23 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticReg X impartus X

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

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Svm's NMIMS des...

Other bookmarks Reading list

learn Install User Guide API Examples More

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.LogisticRegression

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 solver.

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

Chat Participants (83)

Disable group chat

Arijit Barat
mam why default solver is lbfgs?

TADIVALLI NAGA VENKAT RAMA SIRIKANTH joined

SHYAMALA GOWRI V G joined

Arijit Barat
reached minima

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

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

Bhimesan Joshi
Seems there is no effect of over fitting

Rise
ok mam

Arijit Kumar Kasturi
what tolerance limit we are using?

Sending to everyone

20:23 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticReg X impartus X

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

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Svm's NMIMS des... Other bookmarks Reading list

learn Install User Guide API Examples More

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.LogisticRegression

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 solver.

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

Type here to search

Chat Participants (83)

Disable group chat

YADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok reached minima

Arijit Basu

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

Shivam Joshi

Seems there is no effect of over fitting

Alok

Ok mom

Anjan Kumar Keshari

what tolerance limit we are using ?

Shivam Joshi

Ok

Sending to everyone

20:23 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRe... X Impartus X

← → C ⌂ ⌂ [skikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

Apps e-Filing Home Page IBM Quantum Expe... Qiskit textbook Skim's NMIMS de... Other bookmarks Reading list

learn

Prev Up Next

sklearn 1.0.1 Other versions

Please cite us if you use the software.

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

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:

- `'l1'` || `'elasticnet'` || `'none'`
- `'l2'` || `'lbfgs'`

Stop sharing Hide

Impartus.com is sharing your screen.

Type here to search

Other bookmarks Reading list

Chat Participants (83)

Disable group chat

TADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok reached minima

?

Rajat Bera 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

Shivam joshi Seems there is no effect of over fitting

Alok Ok mom

Anjali Kumar Keshan what tolerance limit we are using?

Shivam joshi Ok

Sending to everyone

20:23 14-12-2021

Gradient_Descent_Logistic_Reg x sklearn.linear_model.LogisticRe... x Impartus x

← → C ⌂ ⌂ scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps e-Filing Home Page... IBM Quantum Expe... Qiskit testbook Skim's NMIMS dee... Other bookmarks Reading list

learn

Home 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.LogisticReg

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 not. If True, the objective function is dual. Impartus.com is sharing your screen.

Stop sharing Hide

penalty

intended for l2

$\alpha > n_features$

Chat Participants (83)

Disable group chat

TADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Amit reached minima

7

Arpit Bera 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.

Shivam joshi Seems there is no effect of over fitting

Alok ok mam

Anjali Kumar Keshari what tolerance limit we are using?

Shivam joshi Ok

Sending to everyone

20:24 14-12-2021

Gradient_Descent_Logistic_Reg X X X

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

Impartus

← → C ⌂ scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

IBM Quantum Espe... Qiskit testbook Svm's NMIMS de...

Apps e-Filing Home Page 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.LogisticRegression

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

is > n_features.

Chat Participants (83)

Disable group chat

KADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok reached minima

Arijit Basu 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

Shivam Joshi Seems there is no effect of over fitting

Alok ok mam

Anjani Kumar Kashuri What tolerance limit we are using?

Shivam Joshi Ok

Sending to everyone

20:24 ENG 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Impartus X

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

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Svm's NMIMS de...

Other bookmarks Reading list

learn Install User Guide API Examples More

Pyv Python Host

scikit-learn 1.0.1 Other versions

Please cite us if you use the software.

sklearn.linear_model.LogisticRegression Examples using sklearn.Linear_model.LogisticReg

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 pnorm formulation or no regularization. The 'liblinear' solver supports both L1 and L2 regularization, with a dual formulation only for the L2.

Impartus.com is sharing your screen. Stop sharing Hide id by the saga solver

Chat Participants (83)

Disable group chat

YADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok reached minima

?

Ankit Barai
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

Shimseen joshi
Seems there is no effect of over fitting

Alok
Ok mam

Anjani Kumar Kashuri
what tolerance limit we are using?

Shimseen joshi
Ok

Sending to everyone

20:24 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Impartus X

← → C ⌂ ↻ <https://scikit-learn.org/stable/modules/classes.html#sklearn-linear-model-LogisticRegression>

Apps e-Filing Home Page... Bt Quantum Expo... Qua-Interboox Sivam's NMIMS de... Other bookmarks Reading list

learn Install User Guide API Examples More

Home Up Home

scikit-learn 1.0.1

Other releases

Python site user documentation

GitHub

scikit-learn - Home - scikit-learn.org

repository

Documentation

scikit-learn - Home - scikit-learn.org

Toggle Menu

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=-1, ratio='auto')
```

(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 pmin=1 formulation (no regularization). The `liblinear` solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty.

Impartus.com is sharing your screen. Stop sharing Hide

Stop sharing Hide

Impartus

Chat Participants (83)

Disable group chat

KADAVALLI 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

Shreyaan joshi

Seems there is no effect of over fitting

Alok

ok mam

Anjali Kumar Reddy

what tolerance limit we are using?

Shreyaan joshi

Ok

Sending to everyone

20:24 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRe... X impartus X

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

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Svm's NMIMS de... 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.LogisticRegression

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 0.13: 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

Chat Participants (83)

Disable group chat

YADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok reached minima

Arijit Basu

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

Shivam joshi

Seems there is no effect of over fitting

Alok

ok mam

Anjan Kumar Rathar

what tolerance limit we are using?

Shivam joshi

Ok

Sending to everyone

20:25 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRe... X impartus X

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

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Svm's NMIMS de... 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.LogisticReg...

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

Chat Participants (83)

Disable group chat

YADAVALLI NAGA VENKAT RAMA SRIKANTH joined

SHYAMALA GOWRI V G joined

Alok reached minima

Arijit Basu 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

Shivam joshi Seems there is no effect of over fitting

Alok ok mam

Anjan Kumar Rathore what tolerance limit we are using?

Shivam joshi Ok

Sending to everyone

20:25 14-12-2021

Gradient_Descent_logistic_Reg x sklearn.linear_model.LogisticRe... x Impartus x

← → C ⌂ ⌂ scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

App e-Filing Home Page... IBM Quantum Expe... Qiskit textbook Siem's NMMMS doc... 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.LogisticReg

- 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. ← → most: E501 →

New in ampartus.com is sharing your screen. Stop sharing Hide et.

Toggle Menu

Chat Participants (84)

Disable group chat

Anuj Bansal
in assignment we can use scikit learn or we have to write our own function?

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

Shivam Joshi
Seems there is no effect of over fitting

Amit
Ok mam

Arijit Kumar Ranchari
what tolerance limit we are using?

Shivam Joshi
Ok

Mohamed Naseef
Rakutma & them

Since we haven't learnt about penalty or different solvers yet, are supposed to learn and pick the correct one for the assignment?

Sharing with everyone

20:25 14-12-2021

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

← → C ⌂ 🔍 Type here to search

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

Apps e-Filing Home Page IBM Quantum Espe... Qiskit testbook Svm'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.LogisticRegression

- 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 Impartus.com is sharing your screen. Stop sharing Hide er.

Chat Participants (84)

Disable group chat

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

Bhimsen.joshi Seems there is no effect of over fitting

Alok ok marr

Anjan Kumar Kasturi What tolerance limit we are using?

Bhimsen.joshi Ok

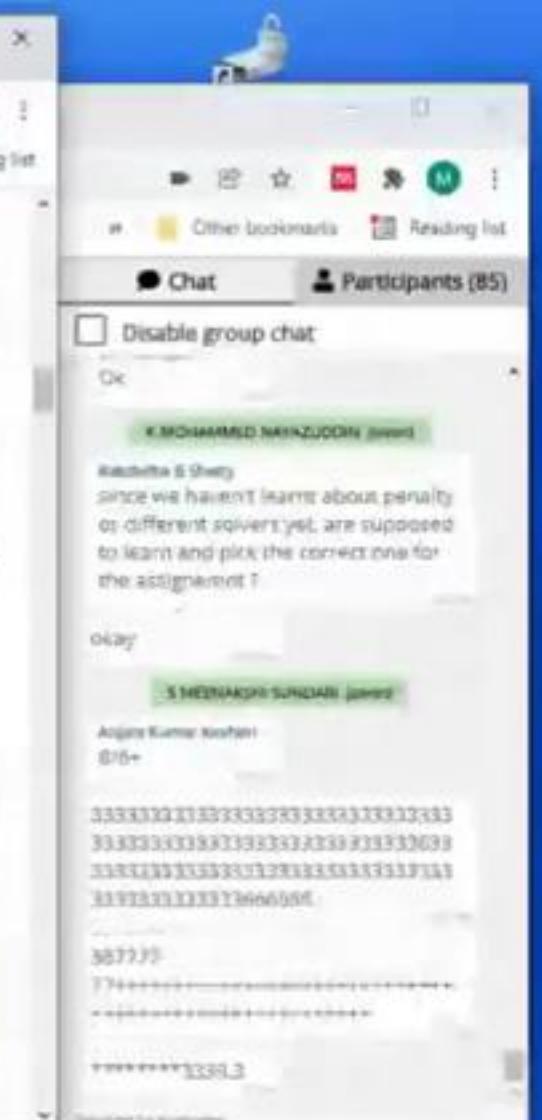
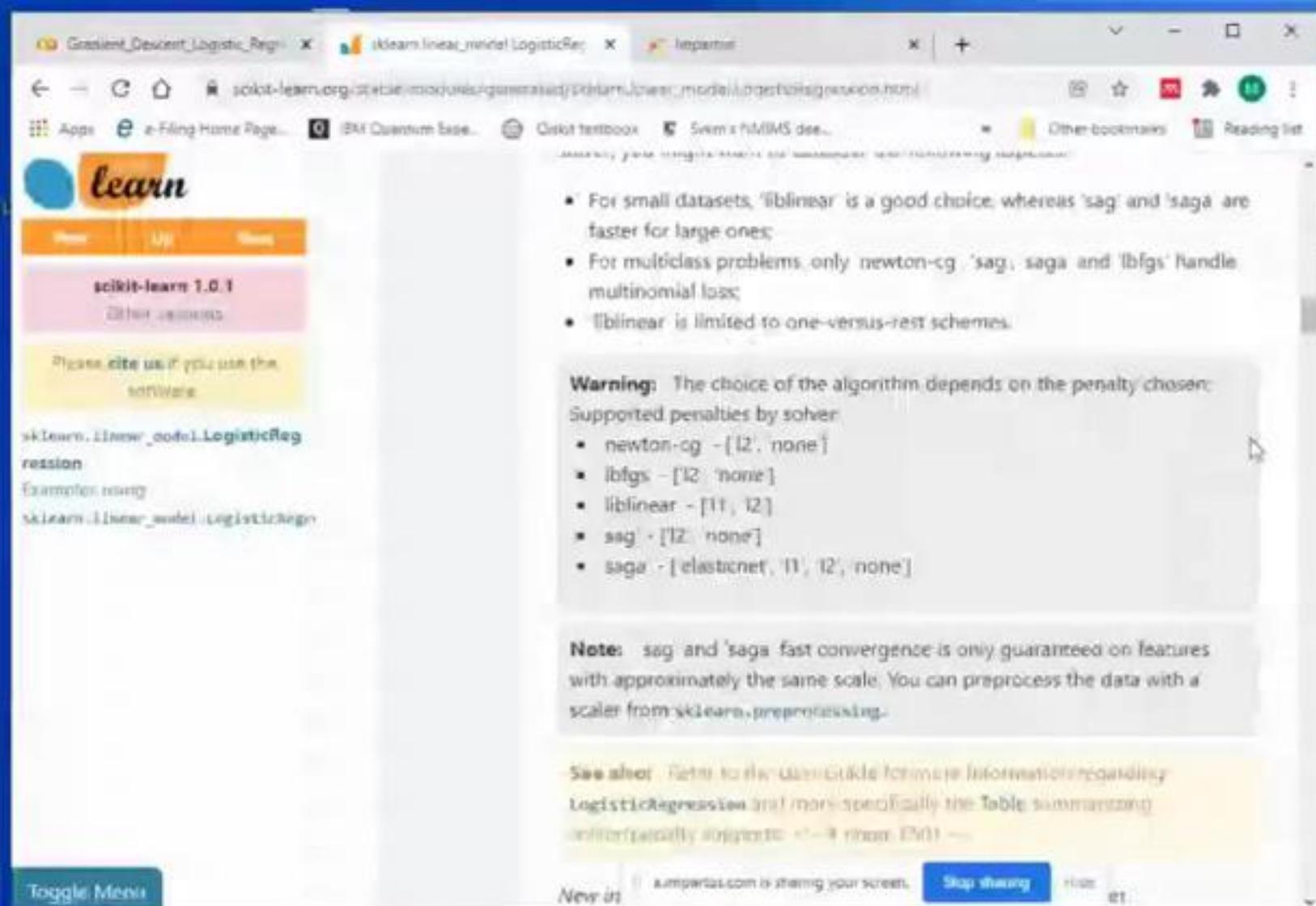
K MOHAMMED NAYAZUDDIN joined

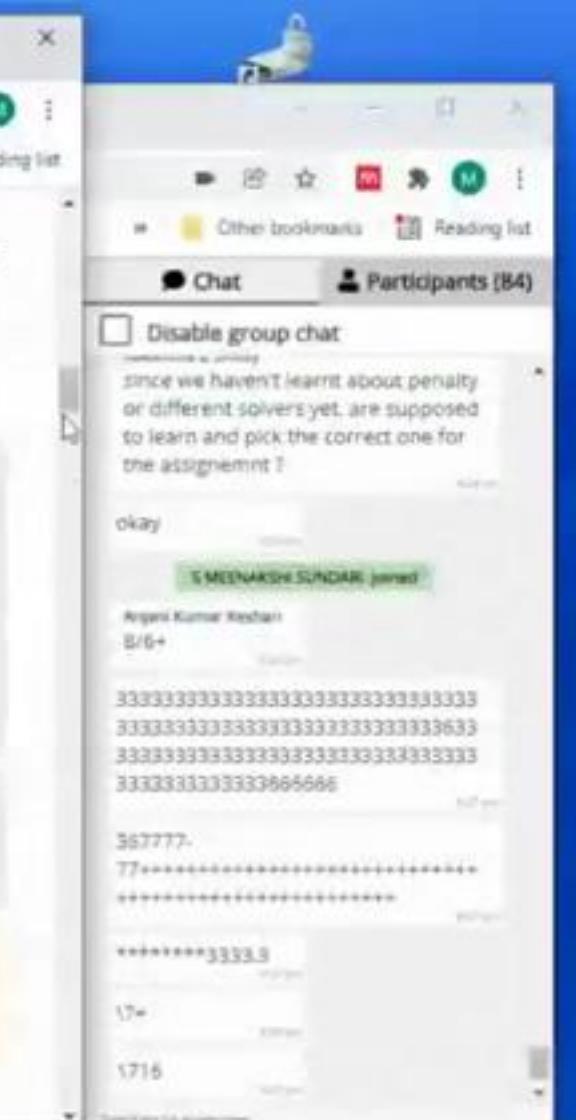
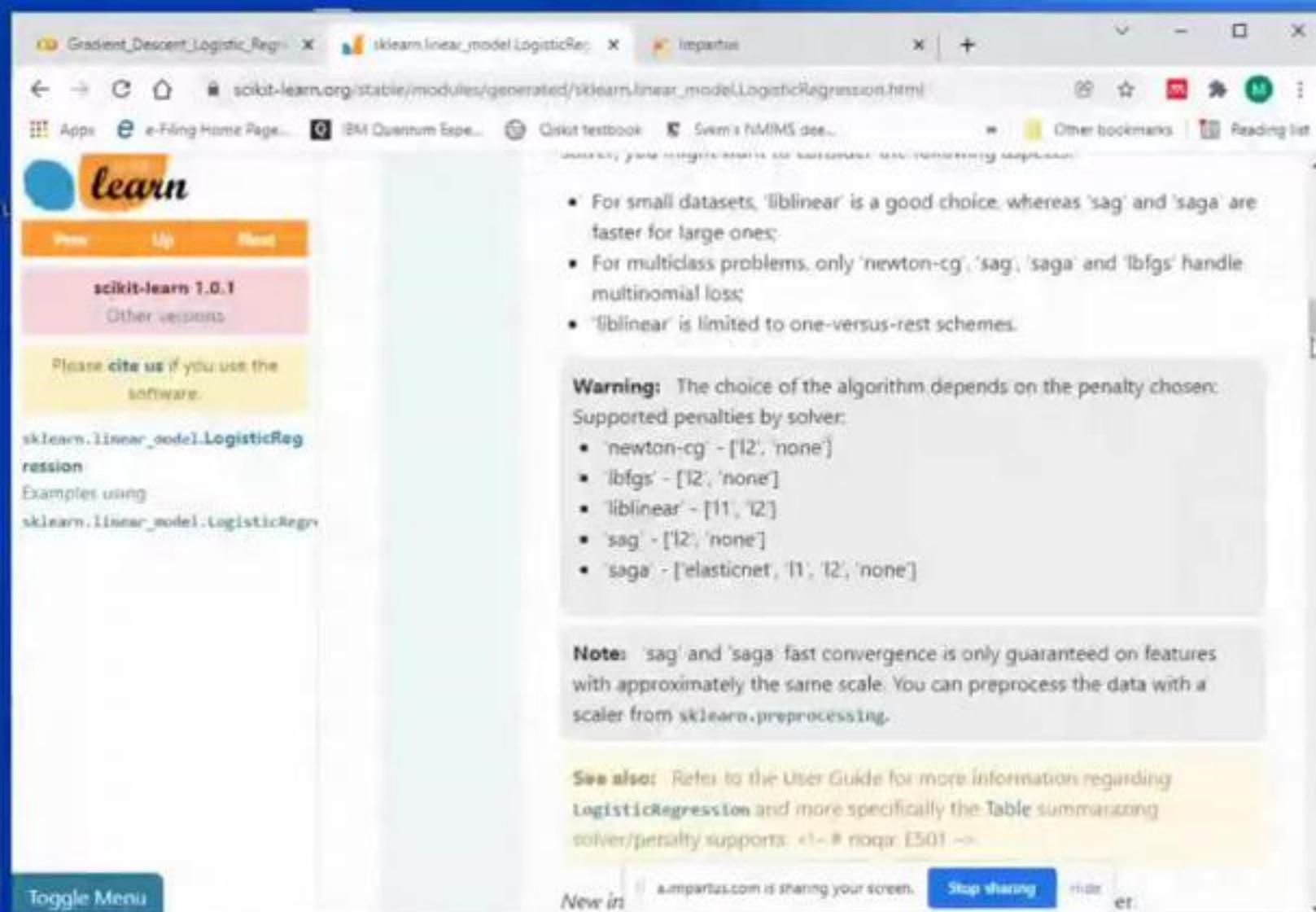
Rakshita B Shetty since we haven't learnt about penalty or different solvers yet, are supposed to learn and pick the correct one for the assignment?

okay

Sending to everyone

20:25 14-12-2021





Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticReg X Impartus X

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

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Sivam's NMIMS de... 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.LogisticRegression

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

When set to True, reuse the solution of the previous call to fit as initialization, otherwise ignore it. This option is only available if solver='lbfgs' or 'sag'. See the notes for an illustration ofwarm_start.

Stop sharing Hide

Other bookmarks Reading list

Chat Participants (84)

Disable group chat

30/12/21 77*****

*****3333.3

17-

1715

+

Arijit Barat

mam, I got assignment on linear regression(housing price). May I request you to spend some time on multi linear regression?

Anjali Kumar Keshari

sorry

Arijit Barat

ok

Sending to everyone

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

← → C O ⌂ scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Apps e-Filing Home Page... IBM Quantum Expe... Qiskit textbook Svm's NMIMS de...

Other bookmarks Reading list

learn Install User Guide API Examples More

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.LogisticReg

Attributes:

l1_ratio : float, default=None

The Elastic-Net mixing parameter, with $0 \leq l1_ratio \leq 1$. 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.

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).

importus.com is sharing your screen. Stop sharing Hide

n feature

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?

Anjali Kumar Kashuri
sorry

Arijit Barat
ok

Rahul Saha
if we increase the data volume, does the iteration impact

Alok
Mam, can't we give 3 classes in the output?

Sending to everyone

20:32 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticReg... X Impartus X

colab.research.google.com/drive/1Dsp6j7sVvVvMkDpG1wmWmJiFXWowam1v#scrollTo=dtCillyMzQf

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Sivam's NMIMS de...

Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# 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 = LogisticRegression( 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

Other bookmarks Reading list

Chat Participants (84)

Disable group chat

sklearn's logisticregression?

Alok
Can we look at the function you created

SUBHODEEP MITRA joined

S.Bhagath
Ok ma am..

SAYANTAN GUPTA joined

Anur Adithi S
Any tips on assignment?, I mean how and where should I start?

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

M R Vijay Krishna
Gradient decent for linear regression

Arpit Barat
more than one feature

Karishma Jethley
outcome is continuous

Sharing to everyone

20:35 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticReg... X Impartus... X

colab.research.google.com/drive/1Dsp6j7sjVvVmkDpG1wmWmFXwcam1v#scrollTo=dtCillyMzQf

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Sivam's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

All changes saved

File Edit View Insert Runtime Tools Help

+ 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

$$\ln\left(\frac{P}{1-P}\right) = b_0 + b_1x_1 + b_2x_2 + b_3x_3 \dots b_nx_n$$

$\ln\left(\frac{P}{1-P}\right)$ 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, \dots are parameters estimated during training of each record. The above equation can be rewritten as

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

simplifying it further

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

Other bookmarks Reading list

Chat Participants (84)

Disable group chat

SUBHODEEP MITRA joined

S@Bhagath
ok ma'am.,

SAYANTANI GUPTA joined

Anur Alitali S
Any tips on assignment2, I mean how and where should I start?

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

M R Vijay Krishnan
Gradient decent for linear regression

Arijit Barua
more than one feature

varsha patney
outcome is continuous

Rajesh R
linear is continuous

S@Bhagath
continuous

sharing to everyone

20:35 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRegr... X Importus X

colab.research.google.com/drive/1Dsp6j7sVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=dtCillyMzQf

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Sivam's NMIMS dee... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[1]: 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	85	66	29	0	26.6		0.351	31
2	8	133	64	0	0	23.3		0.672	32
3	1	89	66	23	94	28.1		0.167	21
4	0	137	40	35	168	43.1		2.288	33

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

(768, 9)

['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

05 comp aimpertus.com is sharing your screen. Stop sharing Hide

Importus

Comment Share M

RAM Disk Editing

Chat Participants (84)

Disable group chat

SUBHODEEP MITRA joined

S Bhagath
ok maram..

SAYANTAN GUPTA joined

Anur Alitali 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

Vansha Jemley
outcome is continuous

Rajesh A
linear is continuous

S Bhagath
Continuous

Sending to everyone

20:35 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Importut X

colab.research.google.com/drive/1Dkpg7jVvVvMcDpG1emWliiFXwcam1v#scrollTo=ctCityLyMzOf

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Sivam's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[1]: 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', 'Outcome']

0s comp aimpetus.com is sharing your screen. Stop shaking Hide

Other bookmarks Reading list

Chat Participants (85)

Disable group chat

Unlink We previously discussed various values

Shyamala Gopal V/S

linear is for continuous and logistic for categorical

Anup Patel

logistic we scale in 0 & 1

Sreyashika Kanchanapara

Logistic vs categorical

Anjali Kumar Kanheri

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

Ama

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 increase or decrease weight values. But in our function we are not comparing the predicted vs actual at each iteration.. 7

Sharing in progress

20:37 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb x | sklearn.linear_model.LogisticRe... x | Imports x

colab.research.google.com/notebook/D.../v.../McD.../tw.../M.../K.../c.../l...#scrollTo=adC.../yL.../M.../

Apps e-Filing Home Page... IBM Quantum Expe... Qiskit textbook Srinivas NMIMS de...

Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

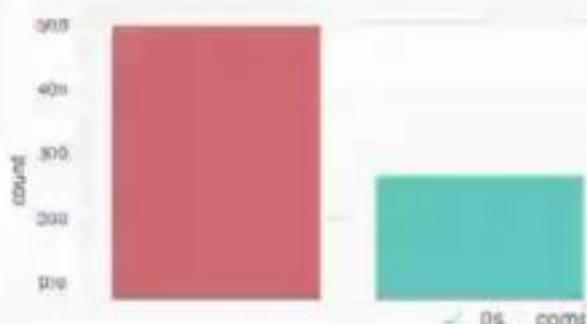
+ Code + Text

RAM Disk 2.2GB 30 1 Editing

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

[6]: (768, 9)
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

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



aumpertus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Chat Participants (85)

Disable group chat

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

Amit
Still not clear on weight update... 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 increase or decrease weight values. But in our function we are not comparing the predicted vs actual at each iteration..?

M. Arvind Kumar
Gradient descent for linear regression

Arijit Basu
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.

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Impartus X

colab.research.google.com/drive/1Dhp6j7gVvVvMkDpG1wnWMiFXNcwam1v#scrollTo=izyaaQMi.com

Apps e-Filing Home Page... IBM Quantum Exp... Qiskit testbook Svm's NMIMS dee...

Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# 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

[31] def main() :
```

aiimpartus.com is sharing your screen. Stop sharing Hide

RAM Disk Editing

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 increase or decrease weight values. But in our function we are not comparing the predicted vs actual at each iteration.. ?

M # Vijay Krishna
Gradient decent for linear regression

Ankit Bawa
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

Starting to execute

20:39 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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 $b_0, b_1, b_2 \dots$ 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

Comment Share M

RAM: 8 Disc: 100 Editing

Chat Participants (82)

Disable group chat

each iteration..?

M R Vijay Krishnan Gradient decent for linear regression

Arpit Srivastava 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.

Arpit Srivastava But where we are comparing result in function to update weight each iteration

ok mam

got it

Arpit Srivastava pls spend some time on linear regression using gradient descent

M R Vijay Krishnan Gradient decent for linear regression

14:49 14-12-2021

Gradient_Descent_Logistic_Reg x colab.research.google.com/drive/1Dsp67jVvyytikDpG1wmWMiFXWcwam1v#scrollTo=jbyaaQMIurom

Impartus x

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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

simplifying it further

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

or

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

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

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

Deriving the sigmoid function,

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

Using reciprocal rule, then rule of linearity,

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

a.impartus.com is sharing your screen.

Stop sharing Hide

Other bookmarks Reading list

Chat Participants (78)

Disable group chat

M R Vijay Krishnan Do you have some links

Arijit Basu ok, only st line eq.

Sekarum Rathnam Monica and derivative will change

M R Vijay Krishnan Or GitHub link it will help

Arijit Basu Ok

M R Vijay Krishnan Thank you

Arijit Basu in assignment , we need to run different combinations to show how results are changing ?

Rishabh Dahiya We can use the package and its functions

Landing to everyone

2042 ENG 14-12-2021

Gradient_Descent_Logistic_Reg x sklearn.linear_model.LogisticRe x | Importus x

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

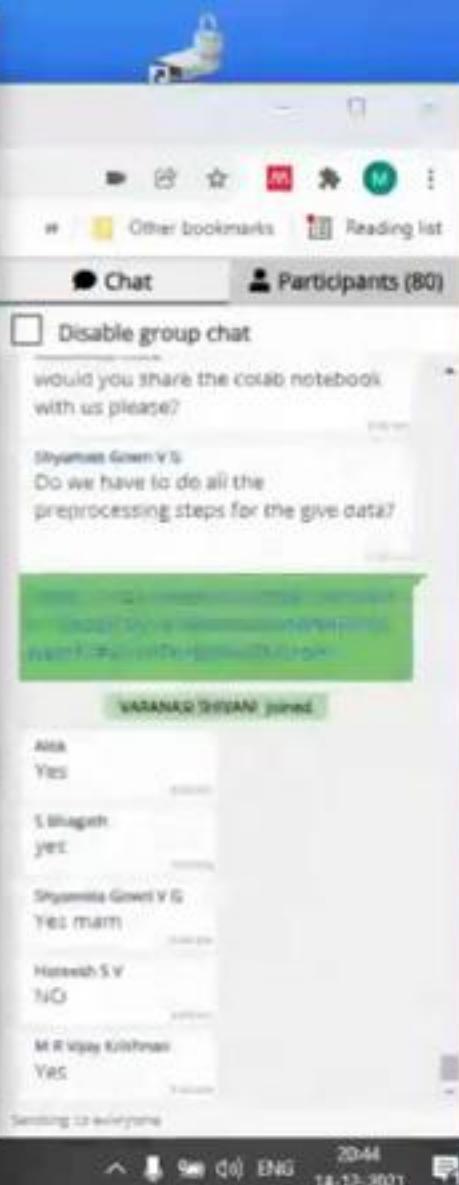
```
[5]: print(list(data.columns))
```

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

```
[x]: data['Outcome'].value_counts()
```

```
sns.countplot(x='Outcome', data=data, palette='Ryl')
```

aimportus.com is sharing your screen. Stop sharing Hide



Gradient_Descent_Logistic_Reg.ipynb x sklearn.linear_model.LogisticRe... x Impartus x

colab.research.google.com/drive/1Dap6j7jVvVvMkDpG1wnWmIFXwcam1v#scrollTo=-5hZdb3ob9Hi

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Svm's NMIMS de... 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

[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	163	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		2.268	33	1

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

(768, 9)

['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

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

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

Other bookmarks Reading list

Chat Participants (80)

Disable group chat

Shyamala Gowri V G
Yes mam

Narmesh S V
NO

M R Vijay Krishnam
Yes

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

Shyamala Gowri V G
Outliers mam?

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

thanks

Siddhanth Rithika Mon... balancing

Sending to everyone

2045 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRe... X importus X

colab.research.google.com/drive/1Dsp67s/vVvMkDpG1wnWMiFXVicwam1v#scrollTo=-5hZdb3ob9Hi

Apps e-Filing Home Page IBM Quantum Expe... Qiskit textbook Sivam's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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', 'Outcome']

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

a-importus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Chat Participants (81)

Disable group chat

Outliers mam?

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

thanks

Senthil Ramaswami...
balancing

Alok
Normalization

Silagach
Can we drop some column, due to high correlation?

Arpit Barat
check null values

DEVARAPALLIN V & SANJAYA joined

Iyashtha Ramchandran...
Missing or null data

Sending to everyone

Stop sharing Hide

Type here to search

05 comp

2046 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb x sklearn.linear_model.LogisticRegression x Impetus x

colab.research.google.com/drive/1Oapi67jyVvVvMkDpG1vnWMiFXXcwam1v#scrollTo=-5hZdb3ob9hi

Apps e-Filing Home Page IBM Quantum Es... Qiskit textbook Stem's NMIMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM: 8 Disk: 80% Editing

✓ Saving 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	133	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', 'Outcome']

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

aumparlus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Chat Participants (81)

Disable group chat

Mulay Lakshmi Sankarragere
balancing—
oversampling/undersampling

Alok
Outlier removals

Chitra Devi K
duplicates

Katireddi Naga Venkatesh
Duplicates if any

Jayanthi Ramachandra Katti
feature reduction, if any

Hemanjali Sachinrao
for balancing do we expect 50% of each class

Shyamala Gowri V.G
But Naive bayes if we are using correlation will be handled right mam
—do we have to handle that as well

Hemanjali Sachinrao
ok, thanks

Sending to everyone

Stop sharing Hide

Type here to search

05.com

20:46 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Importat... X

← → C ⌂ colab.research.google.com/drive/1Dspq7zjVvVvMxDpG1wmWMIxFXWowamIv#scrollTo=5h2db3ob3Hr ⌂ ⌂ ⌂ M ⌂

App e-Filing Home Page... IBM Quantum Expe... Qiskit textbook Salem's NIMMS de... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ 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

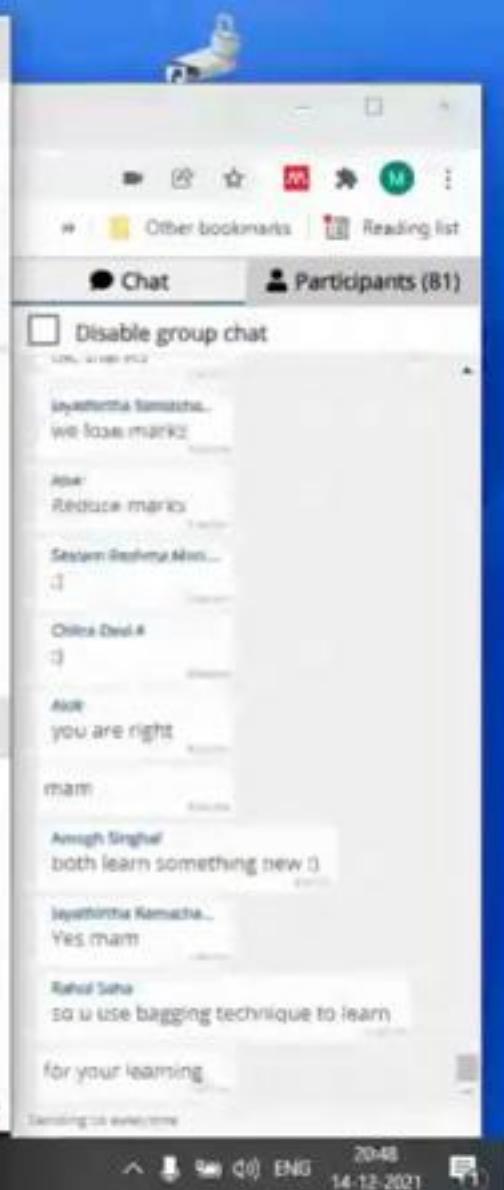
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome	
0	6	148	72	35	0	33.6		0.627	50	1
1	1	85	66	29	0	26.6		0.351	31	0
2	8	163	64	0	0	23.3		0.672	32	1
3	1	89	66	23	94	28.1		0.167	21	0
4	0	137	40	35	168	43.1		2.288	33	1

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

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

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

0s comp aimportus.com is sharing your screen. Stop sharing Hide



Gradient_Descent_Logistic_Reg.ipynb sklearn.linear_model.LogisticReg... importus

colab.research.google.com/drive/1Dsp6j7jVvVvMkOpG1wnWMiFXWowam1v#scrollTo=-5hZdb3ob9Hi

Apps e-Filing Home Page IBM Quantum Exp... Qiskit textbook Sivam's NMIMS dee... Other bookmarks Reading list

Gradient_Desastic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

+ Code + Text

RAM Disk Editing

Choose File 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	163	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		2.268	33	1

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

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

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

aimportus.com is sharing your screen. Stop sharing

Other bookmarks Reading list

Chat Participants (80)

Disable group chat

mam:

Anil Singh both learn something new

sayanthra Ramach...

Yes mam:

Natali Saha So u use bagging technique to learn

for your learning

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 Igoonast

Sending to everyone

20:49 14-12-2021

Gradient_Descent_Logistic_Reg X colab.research.google.com/drive/1Dsp6j7jVvVvMkDpG1wmWMiFXicwam1v#scrollTo=-5hZdb3ob9Hi

Impartus X

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[4] Choose File: 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	133	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', 'Outcome']

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

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
main 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 main

ok

Sending to everyone

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X importus X

colab.research.google.com/drive/1DspIj7sjVVvMkDpG1wmWMiFXwcam1v#scrollTo=-5hZdb3ob9Hi

Apps e-Filing Home Page... IBM Quantum Expe... Qiskit testbook Svkrm'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

[4] Choose File: 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	163	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))
```

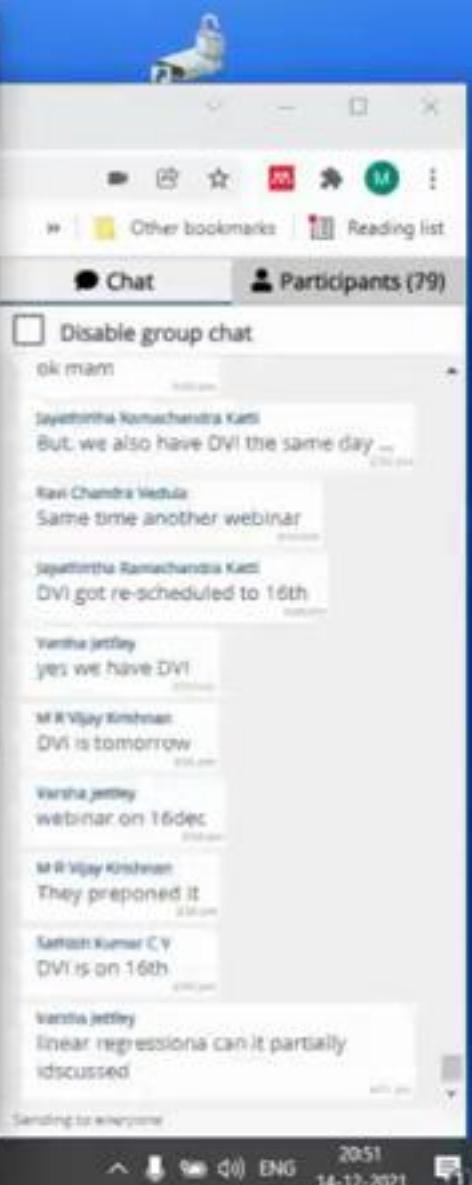
(768, 9)

['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

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

0 500
1 268

importus.com is sharing your screen. Stop sharing Hide



Gradient_Descent_Logistic_Reg... X sklearn.linear_model.LogisticRe... X Impartus X

colab.research.google.com/drive/1Dsp6j7sjVvVvMkDpG1wnWMiFXWcwam1v#scrollTo=-5hZdb3ob9Ht

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit textbook Svitl's NMIMS dee... Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[4] diabetes.csv

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

Saving diabetes.csv to diabetes.csv

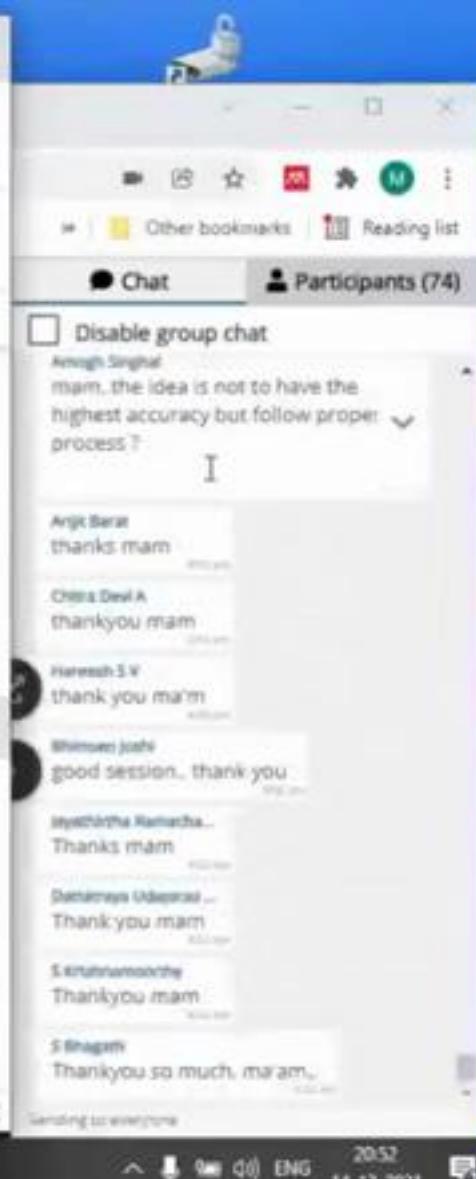
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome	
0	6	148	72	35	0	33.6		0.627	50	1
1	1	85	66	29	0	26.6		0.351	31	0
2	8	163	64	0	0	23.3		0.672	32	1
3	1	89	66	23	94	28.1		0.167	21	0
4	0	137	40	35	168	43.1		2.288	33	1

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

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

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

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



Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticReg X lmipartus

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

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Sivam's NMMNS de... Other bookmarks Reading list

learn

Home 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.LogisticReg

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 feature

lmipartus.com is sharing your screen. Stop sharing Hide

ation as all other feature weight

Toggle Menu

Type here to search

Chat Participants (73)

Disable group chat

TG@iitk.ac.in: Thankyou mam

pratik@: Thankyou so much, mam.

S-Krishna: Thanks mam

Ritika Arora Dinesh: Thank You :)

Hemant kumar: Please upload notebook link in canvas.

Sachin@MATH: will start recording of the session available soon - i missed the beginning

Anup Singh: all mam

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

Sharing is everything

20:59 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb x sklearn.linear_model.LogisticRe... x Impartus x

colab.research.google.com/drive/1Dsp6j7jVvVvMcDpG1wnWMiPXWowam1v#scrollTo=-5hZdb3ob9Hi

File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

+ Code + Text

RAM Disk Editing

independent variables. Let X_i be the independent variable (training variable) and Y be the dependent variable. It 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

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

$\ln\left(\frac{p}{1-p}\right)$ 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, \dots are parameters estimated during training of each record. The above equation can be rewritten as

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

simplifying it further

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

or

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

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

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

Deriving the sigmoid function

Impartus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Chat Participants (72)

Disable group chat

S.Kishoremorthy
Thankyou mam

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

S.Kavitha
thanks mam

Brijesh Arman Dinesh
Thank You..!!

Hansh Vasilchi
Please upload notebook link in canvas.

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

Anmol Singh
ok mam

Vishwanath Shiva
mam i've joined late. Can you pls tell if there notebook links are included in the ppt?

20:53 14-12-2021

Gradient_Descent_Logistic_Reg... X sklearn.linear_model.Logistic... X importus... X

colab.research.google.com/drive/1DspGjzjVvVvNkDpG1vnWMiFXWovam1v#scrollTo=-5hZdb3ab9Hs

Apps e-Filing Home Page... IBM Quantum Espe... Qiskit testbook Sivam's NMIMS de...

Comment Share M

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

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

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 $b_0, b_1, b_2 \dots$ that result in least error. This is where the error or loss function comes in.

Loss Function

importus.com is sharing your screen. Stop sharing Hide

Other bookmarks Reading list

Comment Share M

RAM: 8 Disk: 100% Editing

Chat Participants (72)

Disable group chat

S.Krishnamoorthy
Thankyou mam

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

S.Kavitha
thanks mam

Sohna Arora Dinesh
Thank You..!!

Harsh Vaidya
Please upload notebook link in canvas.

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

Anmol Singh
ok mam

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

20:53 14-12-2021

Gradient_Descent_Logistic_Reg X sklearn.linear_model.LogisticRe... X Impartus X

colab.research.google.com/drive/1Dnplg7yVvVvMkDpG1wmWMIrKwcam1v#scrollTo=-5hZdb3ob9Ht

Apps e-Filing Home Page... IBM Quantum Exp... Qiskit textbook Sankar's NMIMS dee...

Comment Share M

Gradient_Descent_Logistic_Regression.ipynb

All changes saved

+ Code + Text

```
[3]: import matplotlib.pyplot as plt
plt.rcParams["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()

Uploading 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

Stop sharing Hide

Impartus.com is sharing your screen.

Other bookmarks Reading list

Chat Participants (70)

Disable group chat

Angela Parikh
mam we are not sure on 16 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.

Mayanthika Ramachandran Katti
is linear SVM same as linear regression ?

Vijay Krishna
Diabetes.csv file where can we find

The python file is in your google colab

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

Mayanthika Ramachandran Katti
Ahh ... kinda clustering

Vijay Krishna
Features could change from different sources

Sharing to everyone

20:55 14-12-2021

Gradient_Descent_Logistic_Reg.ipynb x silvan.bisognin@ibm.com [object Object] x Inspect x

← → C ⌂ colab.research.google.com/notebooks/1C9qg7JxvVtMxDpc7amWf4P7Q/warm_start/0-512d30c3h

App e IBM Quantum Es... Qiskit textbook E IBM's NIMIS data Other bookmarks Reading list

Gradient_Descent_Logistic_Regression.ipynb

File Edit View Insert Runtime Tools Help All codecell-based

+ Code + Text

[6]:

300
200
100
0

Outcome

```
- [7]: # Logistic Regression
class LogisticRegression():
    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
```

Stop sharing

Other bookmarks Reading list

Chat Participants (70)

Disable group chat

man 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.

jayanthika Ramachandra katt is linear SVM same as linear regression?

M A very common Classification.csv file where can we find

The python file is in your google colab

Amogh Nagrani Can we also attach the requirements file to avoid any dependency conflict

Insomniac Samyakumar R... Ann ... kinda clustering

M A very common Features could change from different sources

Stop sharing

Gradient_Descent_Logistic_Reg.ipynb

sklearn.linear_model.LogisticRegression

Impartus

colab.research.google.com/drive/1D�jg7jVvVtMkDpG1wmWMIKWcwan1v#scrollTo=-5h2db3ok9Hi

Apps e-Filing Home Page... IBM Quantum Exp... Gitkit textbook Svm's NMIMS de...

Comment Share M

File Edit View Insert Runtime Tools Help All changes saved

RAM: 0.351 Disk: 31 Editing

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	1	85	65	29	0	26.5	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	25	94	28.1	0.167	21	0
4	0	157	40	35	168	43.1	2.288	33	1

```
+ Code + Text
```

[4] 1 1 85 65 29 0 26.5 0.351 31 0

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

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

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



Impartus.com is sharing your screen. Stop sharing Hide

