Avinash Navlani
December 16th, 2019

MUST READ PYTHON

Understanding Logistic Regression in Python

Learn about Logistic Regression, its basic properties, and build a machine learning model on a real-world application in Python.

Classification techniques are an essential part of machine learning and data mining applications. Approximately 70% of problems in Data Science are classification problems. There are lots of classification problems that are available, but the logistics regression is common and is a useful regression method for solving the binary classification problem. Another category of classification is Multinomial classification, which handles the issues where multiple classes are present in the target variable. For example, IRIS dataset a very famous example of multi-class classification. Other examples are classifying article/blog/document category.

Logistic Regression can be used for various classification problems such as spam detection. Diabetes prediction, if a given customer will purchase a particular product or will they churn another competitor, whether the user will click

fundamental concepts are also constructive in deep learning. Logistic regression describes and estimates the relationship between one dependent binary variable and independent variables.

In this tutorial, you will learn the following things in Logistic Regression:

- Introduction to Logistic Regression
- Linear Regression Vs. Logistic Regression
- Maximum Likelihood Estimation Vs. Ordinary Least Square Method
- How do Logistic Regression works?
- Model building in Scikit-learn
- Model Evaluation using Confusion Matrix.
- Advantages and Disadvantages of Logistic Regression

Logistic Regression

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.

$$y = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots\ldots + \beta nXn$$

Where, y is dependent variable and x1, x2 ... and Xn are explanatory variables.

Sigmoid Function:

$$p = 1/1 + e^{-y}$$

Apply Sigmoid function on linear regression:

$$p = 1/1 + e^{-(\beta 0 + \beta 1X1 + \beta 2X2 \ldots\ldots \beta nXn)}$$

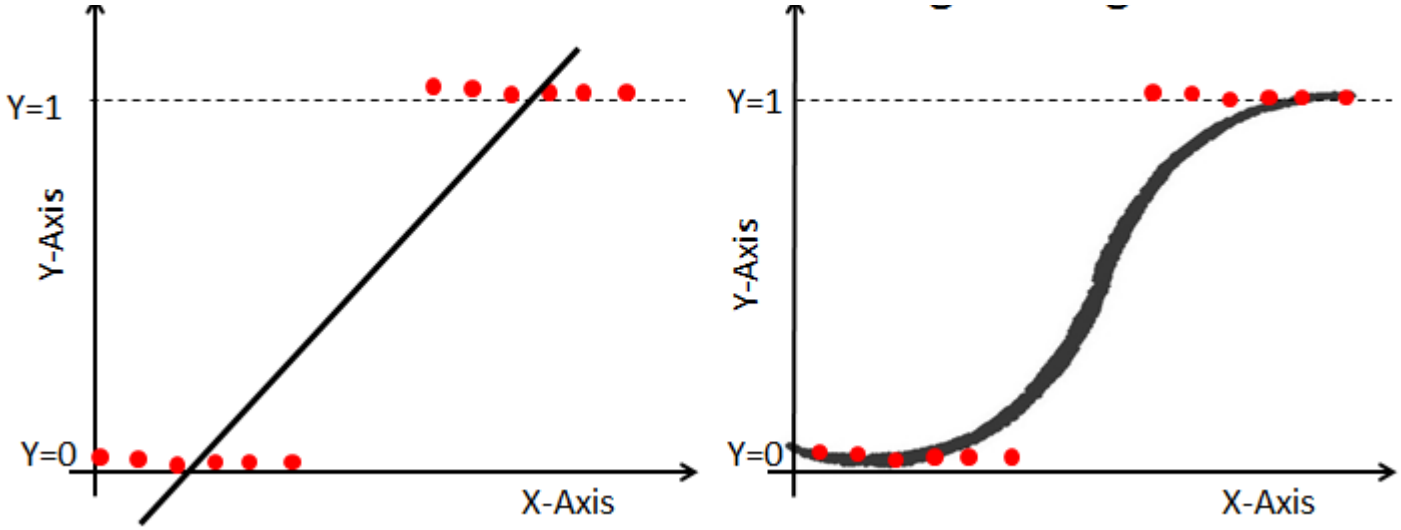
Properties of Logistic Regression:

- The dependent variable in logistic regression follows Bernoulli Distribution.
- Estimation is done through maximum likelihood.
- No R Square, Model fitness is calculated through Concordance, KS-Statistics.

Linear Regression Vs. Logistic Regression

Linear regression gives you a continuous output, but logistic regression provides a constant output. An example of the continuous output is house price and stock price. Example's of the discrete output is predicting whether a patient

Want to leave a comment?



Maximum Likelihood Estimation Vs. Least Square Method

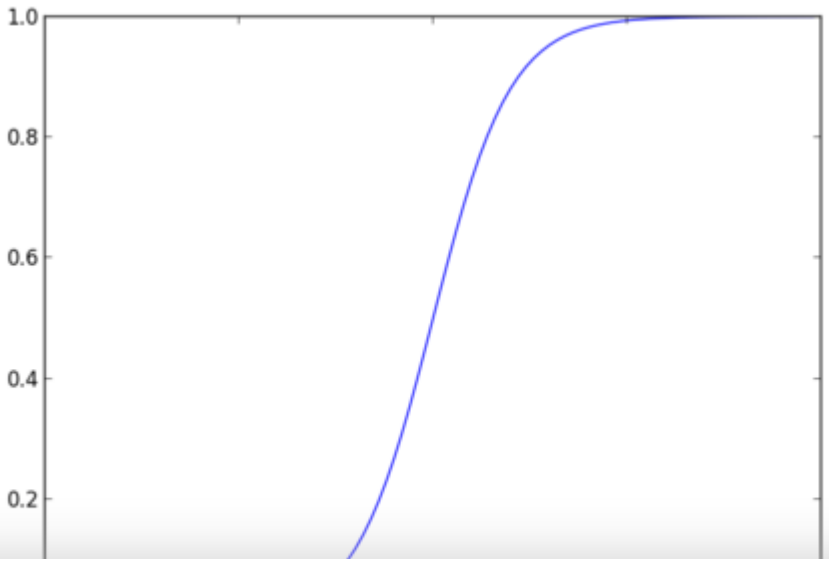
The MLE is a "likelihood" maximization method, while OLS is a distance-minimizing approximation method. Maximizing the likelihood function determines the parameters that are most likely to produce the observed data. From a statistical point of view, MLE sets the mean and variance as parameters in determining the specific parametric values for a given model. This set of parameters can be used for predicting the data needed in a normal distribution.

Ordinary Least squares estimates are computed by fitting a regression line on given data points that has the minimum sum of the squared deviations (least square error). Both are used to estimate the parameters of a linear regression model. MLE assumes a joint probability mass function, while OLS doesn't require any stochastic assumptions for minimizing distance.

Want to leave a comment?

goes to negative infinity, y predicted will become 0. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it as 0 or NO. The outputcannotFor example: If the output is 0.75, we can say in terms of probability as: There is a 75 percent chance that patient will suffer from cancer.

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



Want to leave a comment?

Types of Logistic Regression:

- Binary Logistic Regression: The target variable has only two possible outcomes such as Spam or Not Spam, Cancer or No Cancer.
- Multinomial Logistic Regression: The target variable has three or more nominal categories such as predicting the type of Wine.
- Ordinal Logistic Regression: the target variable has three or more ordinal categories such as restaurant or product rating from 1 to 5.

Model building in Scikit-learn

Let's build the diabetes prediction model.

Here, you are going to predict diabetes using Logistic Regression Classifier.

Let's first load the required Pima Indian Diabetes dataset using the pandas' read CSV function. You can download data from the following link: https://www.kaggle.com/uciml/pima-indians-diabetes-database

Loading Data

```
#import pandas
import pandas as pd
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
```

Want to leave a comment?

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
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

Selecting Feature

Here, you need to divide the given columns into two types of variables dependent(or target variable) and independent variable(or feature variables).

```
#split dataset in features and target variable
feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']
X = pima[feature_cols] # Features
y = pima.label # Target variable
```

Splitting Data

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

Let's split dataset by using function train_test_split(). You need to pass 3 parameters features, target, and test_set size. Additionally, you can use random_state to select records randomly.

Want to leave a comment?

```
from sklearn.cross_validation import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)
```

```
/home/admin/.local/lib/python3.5/site-packages/sklearn/cross_validation.py:41: DeprecationWarning: This module was de
  "This module will be removed in 0.20.", DeprecationWarning)
```

Here, the Dataset is broken into two parts in a ratio of 75:25. It means 75% data will be used for model training and 25% for model testing.

Model Development and Prediction

First, import the Logistic Regression module and create a Logistic Regression classifier object using LogisticRegression() function.

Then, fit your model on the train set using fit() and perform prediction on the test set using predict().

```
# import the class
from sklearn.linear_model import LogisticRegression

# instantiate the model (using the default parameters)
logreg = LogisticRegression()

# fit the model with data
```

Want to leave a comment?

Model Evaluation using Confusion Matrix

A confusion matrix is a table that is used to evaluate the performance of a classification model. You can also visualize the performance of an algorithm. The fundamental of a confusion matrix is the number of correct and incorrect predictions are summed up class-wise.

```
# import the metrics class
from sklearn import metrics
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
cnf_matrix
```

```
array([[119,  11],
       [ 26,  36]])
```

Here, you can see the confusion matrix in the form of the array object. The dimension of this matrix is 2*2 because this model is binary classification. You have two classes 0 and 1. Diagonal values represent accurate predictions, while non-diagonal elements are inaccurate predictions. In the output, 119 and 36 are actual predictions, and 26 and 11 are incorrect predictions.

Visualizing Confusion Matrix using Heatmap

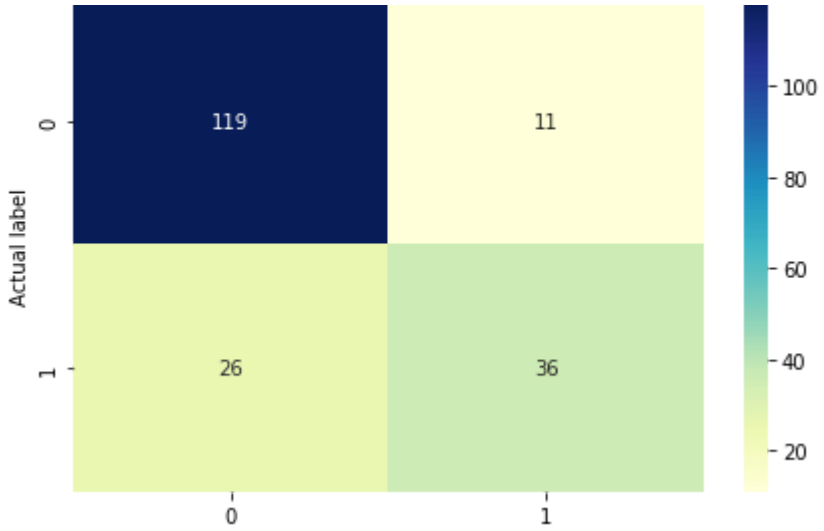
Let's visualize the results of the model in the form of a confusion matrix using matplotlib and seaborn.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline


class_names=[0,1] # name  of classes
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

```
Text(0.5,257.44,'Predicted label')
```

Want to leave a comment?



Confusion Matrix Evaluation Metrics

Let's evaluate the model using model evaluation metrics such as accuracy, precision, and recall.

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
```

```
Accuracy: 0.8072916666666666
Precision: 0.7659574468085106
Recall: 0.5806451612903226
```

Want to leave a comment?

predicted patients are going to suffer from diabetes, that patients have 76% of the time.

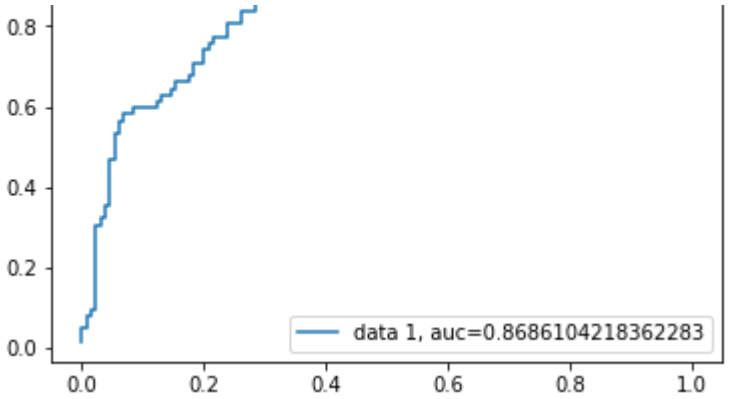
Recall: If there are patients who have diabetes in the test set and your Logistic Regression model can identify it 58% of the time.

ROC Curve

Receiver Operating Characteristic(ROC) curve is a plot of the true positive rate against the false positive rate. It shows the tradeoff between sensitivity and specificity.

```
y_pred_proba = logreg.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
```

Want to leave a comment?



AUC score for the case is 0.86. AUC score 1 represents perfect classifier, and 0.5 represents a worthless classifier.

Advantages

Because of its efficient and straightforward nature, doesn't require high computation power, easy to implement, easily interpretable, used widely by data analyst and scientist. Also, it doesn't require scaling of features. Logistic regression provides a probability score for observations.

Disadvantages

Logistic regression is not able to handle a large number of categorical features/variables. It is vulnerable to overfitting. Also, can't solve the non-linear problem with the logistic regression that is why it requires a transformation of non-linear features. Logistic regression will not perform well with independent variables that are not correlated to the target variable and are very similar or correlated to each other.

Conclusion

Want to leave a comment?

you covered some basic concepts such as the sigmoid function, maximum likelihood, confusion matrix, ROC curve.

Hopefully, you can now utilize the Logistic Regression technique to analyze your own datasets. Thanks for reading this tutorial!

If you would like to learn more about Logistic Regression, take DataCamp's Foundations of Predictive Analytics in Python (Part 1) course.

COMMENTS

Coldknight Coder
07/09/2018 10:16 AM
Explanation of differences between MLE and OLS is not up to the mark. Also, some points like ' logistic regression follows Bernoulli Distribution' isn't explained thoroughly. Consider doing some editing as the grammar is also not good.

▲ 3 REPLY

Avinash Navlani
10/09/2018 07:41 AM
Thanks for your critical feedback.

This tutorial is for absolute beginners. That is the reason that MLE and OLS were not explained in detailed. I want to keep it simple without using the statistics-heavy terminology. Bernoulli distribution is out of the scope of this tutorial.

Thanks a lot

▲ 2 REPLY

Yogesh Soni
13/07/2019 02:35 AM
Hi Avinash,

great article, would need your help in letting me know how are the coefficients interpreted in logistic regression?

Waiting for your reply

▲ 1

Avinash Navlani
10/09/2018 07:41 AM

Want to leave a comment?

▲ 1 REPLY

funny jokes
24/10/2018 11:49 PM
Oh! This article has suggested to me many new ideas. I will embark on doing it. Hope you can continue to contribute your talents in this area. Thank you.

light novel

▲ 1 REPLY

Kind Line
20/12/2019 12:12 PM
This article has suggested to me many new ideas. I will embark on doing it. Hope you can continue to contribute your talents in this area

▲ 1 REPLY

Khachatur Karapetyan
07/09/2018 02:06 PM
You well done!
It's very useful and interesting tutorial.
Thank you a lot.

▲ 2 REPLY

Want to leave a comment?

I have a question though:

Logistic Regression assumes **normality**, and when the features are continuous and on **different scales** you have to standardize them. how doesn't this algorithm require **scaling**?

▲ 2 REPLY

Avinash Navlani
10/09/2018 07:34 AM
I'm glad you like the article.
You don't need to scale data for logistic regression because logistic regression coefficients represent the effect of one unit change in the independent variable on the dependent variable(log odd). If we scale the data between range 0-1 than unit change will shift the value from low to high but there is no change in log odd values.If you are using logistic regression with regularization than it is recommended normalize.

▲ 3 REPLY

May Anne Laciste
26/11/2018 11:33 PM
I encountered some errors:

(1) *from sklearn.cross_validation import train_test_split ==> the cross_validation module is missing but I already installed sklearn.*

(2) In the model development and prediction:

ValueError: could not convert string to float: DiabetesPedigreeFunction

Want to leave a comment?

▲ 1 REPLY

Apurva Verma
19/12/2018 09:55 AM
Remove the first row from the dataset. It contains the column name.

▲ 1 REPLY

Avinash Navlani
19/12/2018 10:26 PM
Here are the solution for your errors:

1) for first one replace *from sklearn.cross_validation import train_test_split* with *from sklearn.model_selection import train_test_split*

2) Because of any of the row will contain missing values so it is unable to convert it into float.

▲ 3 REPLY

Nishanth Raja
08/01/2019 10:16 PM
Can you please elaborate on the second point and how to fix it?

▲ 3

Avinash Navlani
14/01/2019 03:15 AM

Want to leave a comment?

05/07/2019 07:36 AM

You can just change the names in dataset according to program or vice versa and use

pima = pd.read_csv("diabetes.csv")

it will work

▲ 1

Sanket Parate
22/04/2019 10:14 AM

Hi Laciste,

You can use this:

from sklearn.model_selection import train_test_split

▲ 2 REPLY

Pratik Gondhiya
31/01/2019 06:37 AM

Very good article, please keep contributing

▲ 2 REPLY

Lucy Nowacki

Want to leave a comment?

Anjelou Cautivo
01/05/2019 08:22 AM
The ROC curve is not appearing. How do I make that last chunk of codes work?

▲ 2 REPLY

Tracy Xing
27/05/2019 04:50 PM
For some reason, I'm not able to get results for precision and recall. I'm using jupyter lab by the way.

print("Precision:",metrics.precision_score(y_test, y_pred))

print("Recall:",metrics.recall_score(y_test, y_pred))

The error message I got is ValueError: pos_label=1 is not a valid label: array(['0', '1'], dtype = '<u1'). Can anyone help me with this?

▲ 1 REPLY

Tracy Xing
28/05/2019 02:02 PM
ok, it's fixed now with the following codes.

Want to leave a comment?

▲ 2 REPLY

Freeze Ltf
06/08/2019 08:32 PM
where did you add this line of code?

▲ 2

Anthony Doo
04/06/2019 07:34 AM
" from sklearn.cross_validation import train_test_split " is wrong

it should be : "from sklearn.model_selection import train_test_split"

▲ 2 REPLY

Avinash Navlani
06/09/2019 10:07 AM
It is not wrong. it is changed in latest version.

▲ 1 REPLY

---

suwarna choudhary

Want to leave a comment?

ValueError: could not convert string to float: pedigree"

I have used dropna () for pedigree column. There are no null values. Could you please help?

▲ 1 REPLY

Henna Sammy
22/06/2019 01:05 AM
The above post is very useful, but, follow this quick fixes from this guide Fix asus tablet stuck at loading screen and get rid of the problem within a short span of time.

▲ 1 REPLY

Aditya S K
05/07/2019 07:36 AM

You can just change the names in dataset according to program or vice versa and use

pima = pd.read_csv("diabetes.csv")

it will work

▲ 1 REPLY

Want to leave a comment?

Jose Guerra
01/07/2019 01:12 PM
Ok

▲ 1 REPLY

Anissa Amziani
15/08/2019 10:14 AM
Great article!How many categorical variables can a logistic regression can handle? (in the case of the dependent and independent variable). Thanks

▲ 2 REPLY

Avinash Navlani
22/08/2019 07:20 PM
Generally. it uses 2 classes. For multiple classes multi-nomial logistic regression can be used. In scikit learn you can pass following parameters to model object: solver='liblinear' and multi_class : 'multinomial'.

▲ 1 REPLY

David Severson
21/08/2019 06:33 AM
I think the sigmoid function is incorrect. Can someone smarter than me verify? Consider p=1/(1+e^-y)

Want to leave a comment?

▲ 1 REPLY

Lachlan Richardson
03/09/2019 05:40 PM
Encountered a problem. When fitting the data with logreg.fit() it doesn't accept just an x and y variable and needs a variable for self and sample_weight. Is there a way around this? Thanks

▲ 3 REPLY

Avinash Navlani
06/09/2019 10:08 AM
I am not getting your point. Can you please share your code or error?

▲ 1 REPLY

Veera Varma Rudraraju
01/12/2019 04:46 PM
how to view the actual and predicted class variables to showcase the results?

▲ 1 REPLY

jexeson569 jexeson569
19/12/2019 11:30 PM
Hotmail Se Connecter

Want to leave a comment?

22/12/2019 09:55 AM

This is such complex topic you have explained in very simple way. Can you please take more examples on logistic regression.

Thanks and regards,

www.payslipview.com

▲ 1 REPLY

James Hardy
27/12/2019 09:58 AM

Yes right Abhishek, very complex topic but Avinash has nailed it simply. Thanks for such easy explanation Avinash.

James Hardy

Native Notes

▲ 1 REPLY

akshaya vengala
25/12/2019 11:55 PM

videoder

videoder apk

thanks for this nice information.great work!

Want to leave a comment?

It is an exceptional instance of straight relapse where the objective variable is clear cut in nature. It utilizes a log of chances as the needy variable. Calculated Regression predicts the likelihood of event of a double occasion using a logit work read more

▲ 1 REPLY

Joe Morelli
02/01/2020 12:12 PM
Thank you for putting together such a clear and concise intro to logistic regression with Python

▲ 1 REPLY

Florah Melda
05/01/2020 10:55 PM
Our Business Research Paper Services are accessible online for those clients seeking Business Research Paper Writing Services and Affordable Business Research Paper Services Online.

▲ 1 REPLY

Jun Yin
06/01/2020 07:14 AM
Great introduction!

Just confused about this sentence " The outputcannotFor example: If the output is 0.75, we can say in terms of

Want to leave a comment?

shinichi okada
06/01/2020 01:52 PM
It is nice to see a tutorial for beginners. Thank you. I found a couple of errors while running codes. I fixed it and you can see it at https://gist.github.com/e9160f96f62757a40b5f6f319a7b9bcc

▲ 1 REPLY