How to Develop Your First XGBoost Model in Python

by **Jason Brownlee** on August 19, 2016 in **XGBoost**

Last Updated on January 19, 2021

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance that is dominative competitive machine learning.

In this post you will discover how you can install and create your first XGBoost model in Python.

After reading this post you will know:

- How to install XGBoost on your system for use in Python.
- How to prepare data and train your first XGBoost model.
- How to make predictions using your XGBoost model.

Kick-start your project with my new book XGBoost With Python, including *step-by-step tutorials* and the *Python source code* files for all examples.

Let's get started.

- **Update Jan/2017**: Updated to reflect changes in scikit-learn API version 0.18.1.
- **Update Mar/2017**: Adding missing import, made imports clearer.
- **Update Mar/2018**: Added alternate link to download the dataset.

How to Develop Your First XGBoost Model in Python with scikit-learn
Photo by Justin Henry, some rights reserved.

Tutorial Overview

This tutorial is broken down into the following 6 sections:

1. Install XGBoost for use with Python.
2. Problem definition and download dataset.
3. Load and prepare data.
4. Train XGBoost model.
5. Make predictions and evaluate model.
6. Tie it all together and run the example.

1. Install XGBoost for Use in Python

Assuming you have a working SciPy environment, XGBoost can be installed easily using pip.

For example:

```
sudo pip install xgboost
```

To update your installation of XGBoost you can type:

```
sudo pip install --upgrade xgboost
```

An alternate way to install XGBoost if you cannot use pip or you want to run the latest code from GitHub requires that you make a clone of the XGBoost project and perform a manual build and installation.

For example to build XGBoost without multithreading on Mac OS X (with GCC already installed via macports or homebrew), you can type:

```
git clone --recursive https://github.com/dmlc/xgboost
cd xgboost
cp make/minimum.mk ./config.mk
make -j4
cd python-package
sudo python setup.py install
```

You can learn more about how to install XGBoost for different platforms on the XGBoost Installation Guide. For up-to-date instructions for installing XGBoost for Python see the XGBoost Python Package.

For reference, you can review the XGBoost Python API reference.

2. Problem Description: Predict Onset of Diabetes

In this tutorial we are going to use the Pima Indians onset of diabetes dataset.

This dataset is comprised of 8 input variables that describe medical details of patients and one output variable to indicate whether the patient will have an onset of diabetes within 5 years.

You can learn more about this dataset on the UCI Machine Learning Repository website.

This is a good dataset for a first XGBoost model because all of the input variables are numeric and the problem is a simple binary classification problem. It is not necessarily a good problem for the XGBoost algorithm because it is a relatively small dataset and an easy problem to model.

Download this dataset and place it into your current working directory with the file name "**pima-indians-diabetes.csv**" (update: download from here).

3. Load and Prepare Data

In this section we will load the data from file and prepare it for use for training and evaluating an XGBoost model.

We will start off by importing the classes and functions we intend to use in this tutorial.

```
from numpy import loadtxt
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

Next, we can load the CSV file as a NumPy array using the NumPy function **loadtext()**.

```
# load data
dataset = loadtxt('pima-indians-diabetes.csv', delimiter=",")
```

We must separate the columns (attributes or features) of the dataset into input patterns (X) and output patterns (Y). We can do this easily by specifying the column indices in the NumPy array format.

```
# split data into X and y
X = dataset[:,0:8]
Y = dataset[:,8]
```

Finally, we must split the X and Y data into a training and test dataset. The training set will be used to prepare the XGBoost model and the test set will be used to make new predictions, from which we can evaluate the performance of the model.

For this we will use the **train_test_split()** function from the scikit-learn library. We also specify a seed for the random number generator so that we always get the same split of data each time this example is executed.

```
# split data into train and test sets
seed = 7
test_size = 0.33
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=test_size, random_state=seed
```

We are now ready to train our model.

4. Train the XGBoost Model

XGBoost provides a wrapper class to allow models to be treated like classifiers or regressors in the scikit-learn framework.

This means we can use the full scikit-learn library with XGBoost models.

The XGBoost model for classification is called **XGBClassifier**. We can create and and fit it to our training dataset. Models are fit using the scikit-learn API and the **model.fit()** function.

Parameters for training the model can be passed to the model in the constructor. Here, we use the sensible defaults.

```
# fit model no training data
model = XGBClassifier()
```

```
model.fit(X_train, y_train)
```

You can see the parameters used in a trained model by printing the model, for example:

```
print(model)
```

You can learn more about the defaults for the **XGBClassifier** and **XGBRegressor** classes in the XGBoost Python scikit-learn API.

You can learn more about the meaning of each parameter and how to configure them on the XGBoost parameters page.

We are now ready to use the trained model to make predictions.

5. Make Predictions with XGBoost Model

We can make predictions using the fit model on the test dataset.

To make predictions we use the scikit-learn function **model.predict()**.

By default, the predictions made by XGBoost are probabilities. Because this is a binary classification problem, each prediction is the probability of the input pattern belonging to the first class. We can easily convert them to binary class values by rounding them to 0 or 1.

```
# make predictions for test data
y_pred = model.predict(X_test)
predictions = [round(value) for value in y_pred]
```

Now that we have used the fit model to make predictions on new data, we can evaluate the performance of the predictions by comparing them to the expected values. For this we will use the built in **accuracy_score()** function in scikit-learn.

```
# evaluate predictions
accuracy = accuracy_score(y_test, predictions)
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

6. Tie it All Together

We can tie all of these pieces together, below is the full code listing.

```
# First XGBoost model for Pima Indians dataset
from numpy import loadtxt
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# load data
dataset = loadtxt('pima-indians-diabetes.csv', delimiter=",")
# split data into X and y
X = dataset[:,0:8]
Y = dataset[:,8]
# split data into train and test sets
seed = 7
test_size = 0.33
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=test_size, random_state=see
```

```
# fit model no training data
model = XGBClassifier()
model.fit(X_train, y_train)
# make predictions for test data
y_pred = model.predict(X_test)
predictions = [round(value) for value in y_pred]
# evaluate predictions
accuracy = accuracy_score(y_test, predictions)
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

Note: Your results may vary given the stochastic nature of the algorithm or evaluation procedure, or differences in numerical precision. Consider running the example a few times and compare the average outcome.

Running this example produces the following output.

```
Accuracy: 77.95%
```

This is a good accuracy score on this problem, which we would expect, given the capabilities of the model and the modest complexity of the problem.

Summary

In this post you discovered how to develop your first XGBoost model in Python.

Specifically, you learned:

- How to install XGBoost on your system ready for use with Python.
- How to prepare data and train your first XGBoost model on a standard machine learning dataset.
- How to make predictions and evaluate the performance of a trained XGBoost model using scikit-learn.

Do you have any questions about XGBoost or about this post? Ask your questions in the comments and I will do my best to answer.

More On This Topic

BigML Tutorial: Develop Your First Decision Tree and…

Your First Machine Learning Project in Python Step-By-Step

Your First Deep Learning Project in Python with…

How to Run Your First Classifier in Weka

Design and Run your First Experiment in Weka

Your First Machine Learning Project in R Step-By-Step

About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

View all posts by Jason Brownlee →

‹ A Gentle Introduction to XGBoost for Applied Machine Learning

Data Preparation for Gradient Boosting with XGBoost in Python ›

162 Responses to *How to Develop Your First XGBoost Model in Python*

Qichang Feng August 26, 2016 at 8:21 pm #

REPLY

Hi Jason,

First of all thanks for all your great posts. I have learned a lot from them.

I have a question regarding the code seperating input features X and response variable Y. It seems you include the last column in the features as well which should not be the case.

X = dataset[:,0:8]

The correct one should be X = dataset[:, 0:7] to match 8 input variables for the medical details of patients.

The error happened in your mini-course handbook as well.

Jason Brownlee August 27, 2016 at 11:32 am #

REPLY

You're welcome Qichang.

Perhaps you are getting different results based on the version of Python or Numpy you are using.

I can confirm that the code in the post is correct:

```
1 import numpy
2 dataset = numpy.loadtxt('pima-indians-diabetes.csv', delimiter=",")
3 X = dataset[:,0:8]
4 Y = dataset[:,8]
5 dataset.shape
6 X.shape
7 Y.shape
```

There are 9 columns, only the first 8 are stored in X with the 9th stored in Y. The above snippet produces:

```
1 (768, 9)
2 (768, 8)
3 (768,)
```

Does that help?

Tested on Python 2.7.11 and numpy 1.11.1.

Qichang August 28, 2016 at 10:27 am #

REPLY

Hi Jason,

Thanks a lot for your quick reply. It is my mistake as I am confused with 0:8 because I am also learning R recently. In R, the last number of 0:8 is included while it is excluded in Python. I should have checked the shape.

Thanks again.

Jason Brownlee August 29, 2016 at 8:08 am #

REPLY

No problem at all Qichang.

Joao Pires September 21, 2016 at 6:42 am #

REPLY

Hi
I run the code and I get this error:
model = xgboost.XGBClassifier()
AttributeError: 'module' object has no attribute 'XGBClassifier'

Do you know why?

Thks

Jason Brownlee September 21, 2016 at 8:30 am #

REPLY

You need to import xgboost.

Taro December 3, 2018 at 8:53 pm #

REPLY

Hi Jason. Thanks for this well elucidated tutorial. But I seem to encounter this same issue whereas I've already imported xgboost.

Jason Brownlee December 4, 2018 at 6:01 am #

REPLY

You may have a typo in your code, perhaps ensure that you have copied the code exactly.

Shubham October 16, 2020 at 8:09 am #

REPLY

what you are doing is this –

import xgboost

do this and the code should run fine –

from xgboost import XGBClassifer

SG Huang September 29, 2016 at 7:40 pm #

REPLY

Thanks Jason for the clear guide.

What is the normal ways to improve the accuracy in practice? Shall we do some featuring engineering, or change to a different model?

I have learned the basics of machine learning through online courses, but there is still a gap between what I learned in the courses and the practical problems such as the competitions on Kaggle. Can you share some insights?

Jason Brownlee September 30, 2016 at 7:51 am #

REPLY

I would recommend trying some feature engineering first.

Try some new framings of the problem.

Then later try algorithm tuning and ensemble methods.

I have a list of things to try in the following post, it talks about deep learning but the techniques are general enough for most methods:
http://machinelearningmastery.com/improve-deep-learning-performance/

I hope that helps as a start.

Jessica November 11, 2016 at 4:39 am #

REPLY

Thank you for this, it's extremely helpful.

I wrote a model for my data last night, and it performed very well.
I tried to re-run it today, and it gave me an error trying to import xgboost.

I typed in "import xgboost"
And I got: "ImportError: No module named xgboost"

Jason Brownlee November 11, 2016 at 10:06 am #

REPLY

Sorry to hear that Jessica.

I wonder if something changed with your environment.

Perhaps try running everything from the command line.
Confirm you're using the same user.
Confirm xgboost is still installed on the system (pip show or something…)

Trupti November 21, 2016 at 5:26 pm #

REPLY

hello, thanks for the fantastic explanation!!
I have a query. Can we get the list of significant variables that entered in the model? How do we read the "feature_importances_"?
Also, how to fin-tune the xgboost model?
Thanks again!

Jason Brownlee November 22, 2016 at 6:59 am #

REPLY

Great questions Trupti,

Here's a tutorial on feature importance with xgboost:
http://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/

Here's a tutorial on tuning xgboost:
http://machinelearningmastery.com/tune-number-size-decision-trees-xgboost-python/

And I have many more, try the search feature.

Trupti December 1, 2016 at 10:14 pm #

REPLY

Thanks a lot! Will try this.
For this we will have to install joblib right ?

Jason Brownlee December 2, 2016 at 8:16 am #

REPLY

You may.

Varma August 27, 2018 at 5:20 am #

REPLY

Hey Jason

Can you let me if there are any parameters for XG Boost

Jason Brownlee August 27, 2018 at 6:15 am #

REPLY

I have many posts on how to tune xgboost, you can get started here:
https://machinelearningmastery.com/start-here/#xgboost

Trupti November 21, 2016 at 7:55 pm #

REPLY

Hello. Thanks for the explanation!
Can you tell me if I can see the list of variables entering in the model. Also, how do we fine tune the model further??
Once we have the xgboost model..how do we productionise it? In logistic regression we get an equation which can be automated to run in real time production, what do we get in xgboost?

Jason Brownlee November 22, 2016 at 7:03 am #

REPLY

I would recommend saving the model to file for use in production. Here's an example:
http://machinelearningmastery.com/save-gradient-boosting-models-xgboost-python/

Peter Tan December 8, 2016 at 8:26 am #

REPLY

Hi Jason, I am running into the same issue as some of the readers here:

AttributeError: 'module' object has no attribute 'XGBClassifier'

To ensure I did not have any typo, I have created a complete copy of your sample code and I still get the same issue.

(I do have import xgboost in my code).

I am using xgboost 0.6a2 with anaconda2-4.2.0. Just wondering if you have run into similar issues.

Hector December 30, 2016 at 1:29 pm #

REPLY

Hello Jason, I ran the example code here and one error returned as:

File “./test.py”, line 21
model = xgboost.XGBClassifier()
^
SyntaxError: invalid syntax

Can you tell me what I did wrong? I can successfully import the packages.

I am using python 3.5 and xgboost 0.6.

Jason Brownlee December 31, 2016 at 7:02 am #

REPLY

Perhaps a copy paste error? Check for extra white space in your copy of the code.

Trupti January 7, 2017 at 5:31 pm #

REPLY

I am using predict_proba to create predicted probabilities by xgboost model. Can I save these probs in the same train data on which model is built so that I can further create reports to show management about validations of the scorecard.

Jason Brownlee January 8, 2017 at 5:20 am #

REPLY

Sorry, I don’t think I understand.

Predicted probabilities on the training dataset will be biased. You may want to report on the probabilities for a hold-out dataset.

Niranjan March 14, 2017 at 3:23 am #

REPLY

Hi, It was a very nice intro to xgboost. Please add a import for train_test_split function

---

Jason Brownlee March 21, 2017 at 8:51 am #

REPLY

Fixed, thanks for the note.

Keren March 27, 2017 at 12:15 am #

REPLY

Hi Jason,
I didn't manage to find a clear explanation for the way the probabilities given as output by predict_proba() are computed.

In random forest for example, I understand it reflects the mean of proportions of the samples belonging to the class among the relevant leaves of all the trees.

However in XGBoost I couldn't understand the computation from the documentation or the code. Shouldn't it give different weights for each tree?

REPLY

Jason Brownlee March 27, 2017 at 7:56 am #

Good question Keren, I'm not sure off hand.

You could check some of the original stochastic gradient boosting papers or even reach out to the xgboost authors.

REPLY

Niranjan April 20, 2017 at 8:31 pm #

Hi, Jason, Thank you for such a nice explaination, would you help me out regarding how to print the training accuracy while we call the fit function in xgboost?

REPLY

Jason Brownlee April 21, 2017 at 8:35 am #

I'm glad it helped.

You can evaluate the fit model on a new test. Is that what you mean?

See this post:
http://machinelearningmastery.com/evaluate-performance-machine-learning-algorithms-python-using-resampling/

REPLY

sumi May 25, 2017 at 3:52 pm #

Hi,

Thankyou for your post. It was really helpful.But can you tell me why do I get 'ImportError: cannot import name XGBClassifier' when I run this code?i have installed XG Boost successfully and I still have this error. Please help me.

REPLY

Jason Brownlee June 2, 2017 at 11:42 am #

Perhaps you do not have sklearn installed?

REPLY

vishwas May 25, 2017 at 10:20 pm #

how to combine Xgboost classifier and Deep learning and create ensemble(voting classifier)…can you please elaborate more on ensemble techniques

REPLY

Jason Brownlee June 2, 2017 at 11:46 am #

Perhaps voting or stacking.

REPLY

joao June 10, 2017 at 6:29 pm #

In your step by step explanation you have: “from xgboost import XGBClassifier” and then you use: “model = xgboost.XGBClassifier()”. This will give an error.
In the full code you have it right though.

REPLY

Jason Brownlee June 11, 2017 at 8:22 am #

Thanks joao. Fixed!

REPLY

Mahmoud July 18, 2017 at 6:56 pm #

Hello Dr Jason, thanks for the quick cool tutorial. It is fundamental and very beneficial.
one question, how do I use GPU for training and prediction purposes in XGBoost? I am working on large dataset. thanks a lot in advance.

REPLY

Jason Brownlee July 19, 2017 at 8:22 am #

I don’t know off hand, sorry.

REPLY

Bhupendra singh October 6, 2017 at 5:54 am #

hey ! this performed very well but how will I know which features are selected ?

REPLY

Bhupendra singh October 6, 2017 at 5:55 am #

sorry I asked a wrong question …

REPLY

xuyuewei October 25, 2017 at 7:39 pm #

Thanks a lot

REPLY

Jason Brownlee October 26, 2017 at 5:25 am #

You're welcome.

REPLY

Eric Wu November 11, 2017 at 4:56 am #

Gee, the 20 or so lines of code is the basic recipe for almost all supervised learning tasks and XGBoost is like the default algorithm. I wish there is a way I could "double" bookmark this page. Well done!

REPLY

Jason Brownlee November 11, 2017 at 9:24 am #

Thanks Eric!

REPLY

kono November 14, 2017 at 8:52 am #

Hi Jason,

XGBClassifier's default objective is binary:logisitc. For binary:logistic, is its objective function the summation of logloss? If so, why XGBoost use "error"(accuracy score) as the default evaluation metric instead of "logloss"?

https://github.com/dmlc/xgboost/blob/master/doc/parameter.md#learning-task-parameters

Kono

mit December 12, 2017 at 6:41 pm #

REPLY

Could you please give me an example how a model should be developed using training data and perform a test on the test data?

I mean, How I can do the following:

1. Use training data to develop model and use test data to predict;
2. Use the combined data set (Train and test dataset) and apply Cross-validation.

Jason Brownlee December 13, 2017 at 5:29 am #

REPLY

This post should you develop a final model:
https://machinelearningmastery.com/train-final-machine-learning-model/

Frankli December 13, 2017 at 2:01 pm #

REPLY

Hi, Jason

how to adjust the parameters in this model?

it seems that this blackbox can do everything, but we don't know the detail in it

---

Jason Brownlee December 13, 2017 at 4:14 pm #

REPLY

You can use the hyperparameters to change the way the model is trained.

---

frankli December 13, 2017 at 5:31 pm #

REPLY

thanks, but what is hyperparameters? a package in xgboost?
any sample codes?

Jason Brownlee December 14, 2017 at 5:33 am #

REPLY

Hyperparameters are ways to configure the algorithm, learn more here:
https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/

frankli December 14, 2017 at 6:15 pm #

REPLY

thanks!

Nasir January 13, 2018 at 8:25 am #

REPLY

Hi Jason

Thanks for very nice tutorial. I would appreciate, if you give me advice.
I have vibration data (structured format). I am using deep learning Keras using tensorflow. But I read that "Specifically, gradient boosting is used for problems where structured data is available, whereas deep learning is used for perceptual problems such as image classification.

Practitioners of the former almost always use the
excellent XGBoost library, which offers support for the two most popular languages of
data science: Python and R"

I am very confused and would like to know your expert opinion that I have to switch and use gradient boosting? I am interested to use for regression purpose.

Hope to hear from you.

Nasir

Jason Brownlee January 14, 2018 at 6:32 am #

REPLY

Yes, you have heard good advice!

Pratip January 30, 2018 at 7:51 pm #

REPLY

Hello sir ,

I am trying use this :
from xgboost import XGBClassifier

but it gives me an error as cannot import name 'XGBClassifier'

But when i import xgboost it works .
Can you tell me my error why its not working ?

Jason Brownlee January 31, 2018 at 9:40 am #

REPLY

Perhaps the API has changed?

Matthew March 10, 2018 at 12:01 am #

REPLY

The diabetes dataset link is returning a 404. Any idea where it has gone?

Jason Brownlee March 10, 2018 at 6:29 am #

REPLY

I will fix that up ASAP.

Gary March 19, 2018 at 8:31 am #

REPLY

I'm getting an error `XGBoostError: sklearn needs to be installed in order to use this module` however I _do_have sklearn installed in the active environment (and in all the other. I think)

Jason Brownlee March 20, 2018 at 6:09 am #

REPLY

Perhaps you are able to confirm that sklearn is installed by checking its version?

This post can help:
https://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/

Deep March 25, 2018 at 9:31 pm #

REPLY

I am new in ML concept & your examples are very helpful & simple to understand.

I have recreated the same example based on my data.

My code below:

model = XGBClassifier()
model.fit(X_test,Y_test)

Q = vectorizer.transform([“I want to play online game”]).toarray()
pred_data = model.predict(Q)

I am getting correct prediction but how can I get the score of the prediction correctly.
Even I used predict_proba of xgboost & getting all the scores but is this the way to get the score of my prediction or some other way is there?

Jason Brownlee March 26, 2018 at 10:01 am #

REPLY

Looks like you're trying to work with text data, perhaps start here:
https://machinelearningmastery.com/start-here/#nlp

File April 8, 2018 at 6:55 pm #

REPLY

Thank you Jason, this blog really helps a lot.
And I noticed that the dataset you referred is not available anymore. Could you recommend another bi-classification dataset please, thanks –

Jason Brownlee April 9, 2018 at 6:08 am #

REPLY

You can download it from here:
https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv

---

IrriAnalytics May 3, 2018 at 7:16 pm #

REPLY

How can I obtain the set of decision rules (cuts on the features), once I have built the model?

Jason Brownlee May 4, 2018 at 7:41 am #

REPLY

Good question, generally this is not feasible given that there many be hundreds or thousands of trees in the model.

---

charliew May 9, 2018 at 3:12 am #

REPLY

Thanks for the work. I ran into an error when trying to do:

model = XGBClassifier(objective='multi:softprob')
model.fit(X_train, Y_train)

the error is: b'value 0for Parameter num_class should be greater equal to 1′

It works fine if I don't specify objective='multi:softprob'. Just wondering if you have any experience with XGBClassifier(objective='multi:softprob')?

Thanks

Jason Brownlee May 9, 2018 at 6:26 am #

REPLY

Sorry, I have not seen this error.

Perhaps post to stackoverflow?

Kate May 20, 2018 at 9:52 pm #

REPLY

Hi!
I'm currently experimenting with XGBoost for an important project and have uploaded a question on StackOverflow. I just read this post and it is clearer to me now, but you do not use the xgboost.train method. It this included in the XGBRegressor wrapper? I did use xgboost.train, which gave me an error, while xgboost.fit does not produce this error. Could you maybe take a look at it?
https://stackoverflow.com/questions/50426680/xgboost-gives-keyerror-best-msg

Thanks in advance!
Kind regards

Jason Brownlee May 21, 2018 at 6:30 am #

REPLY

Perhaps you can summarize your problem for me in one or two lines?

Michael June 7, 2018 at 3:36 pm #

REPLY

Hi,

I am using XGBRegressor wrapper to predict the sales of a product, there are 50 products, I want to know the coefficient as in linear regression to see which product sales is affecting how much to the dependent sales variable. Let say Y = B1X1 + B2X2 + ….. BnXn + C , i want the values of B1,B2,….Bn from tree regressor(XGBRegressor).

Jason Brownlee June 8, 2018 at 6:05 am #

REPLY

An xgboost model is different from a linear regression. There are no list of coefficients, just a ton of trees.

Michael June 8, 2018 at 11:38 pm #

REPLY

Thanks, but is there a way where I can determine that what percentage of other product sales is affecting the sales of my dependent (sales)
variable

REPLY

Jason Brownlee June 9, 2018 at 6:54 am #

Yes, but this might be a question of statistical methods, not predictive modeling.

REPLY

Todd June 8, 2018 at 6:38 am #

Jason, thanks for the great article (and site)
I have a text classification problem that I normally use Logistic Regression to solve. So I'm used to transforming the features in order to fit a model, but I normally don't have to do anything to the text labels. The labels are text categories (e.g. labels = ['cancel', 'change', 'contact support', etc]. I am now receiving error

dtrain = xgb.DMatrix(X_train_dtm, label=y_train)

TypeError: must be real number, not str

y_train is text data. How would I start to solve for this? Any pointers? Do I need to do some sort of transformation to the labels?

REPLY

Jason Brownlee June 9, 2018 at 6:43 am #

You must encode the labels as integers. You can use a label encoder to do this.

I explain more here:
https://machinelearningmastery.com/faq/single-faq/how-to-handle-categorical-data-with-string-values

REPLY

Leote Cherradi September 18, 2018 at 12:13 am #

Hello,

Nice article
juste wanted to say that for classification better to use F1 score, precision and recall and a confusion Matrix.

Here is some python code to add at the end :

```
predictions = model.predict(X_test)
Y_Testshaped = y_test.values

cm = confusion_matrix(Y_Testshaped, predictions)
print('F1 : ' + str(f1_score(Y_Testshaped, predictions,average=None)) )
```

```
print('Precision : ' + str(precision_score(Y_Testshaped, predictions,average=None)) )
print('Recall : ' + str(recall_score(Y_Testshaped, predictions,average=None)) )

fig, ax = plot_confusion_matrix(conf_mat=cm)
plt.show()
```

Jason Brownlee September 18, 2018 at 6:17 am #

REPLY

It depends on the goals of your project.

Choose a measure that help you best demonstrate the performance of your model to your stakeholders.

Anshita October 3, 2018 at 9:48 pm #

REPLY

Hi Jason,

When I put test-size = 0.2, then the model accuracy increases. It shows the accuracy_score = 81.17% and when I take test-size = 0.15 then accuracy_score = 81.90% and if I take test-size = 0.1 then accuracy_score = 80.52%. So, is it good to take the test-size = 0.15 as it increases the accuracy_score? I normally see the test-size = 0.2 or 0.3 or in-between. So, for good model should I select that model which gives me higher model accuracy_score? If not, why?

Jason Brownlee October 4, 2018 at 6:16 am #

REPLY

More data is generally better.

The differences may not be real, e.g. statistical noise.

Hoang December 3, 2018 at 8:42 pm #

REPLY

model.predict(X_test) gives class predictions.
model.predict_proba(X_test) gives score predictions.

So I guess if we do model.predict(X_test), we don't need to round the results. Am I right?
Thank you!

Jason Brownlee December 4, 2018 at 6:00 am #

REPLY

Yes, the API has changed a lot in recent years.

Eric Ewald December 7, 2018 at 4:46 am #

REPLY

Jason,

I am new to machine learning, but have a familiarity w/ regression. So what i take from the output of this model is that these variables (X), are 77.95% accurate in predicting Y. My question is how would i apply this data? Can in create a function that i can input these variables (X), to predict the probability for someone to become stricken with diabetes Y?

Eric

Jason Brownlee December 7, 2018 at 5:27 am #

REPLY

Yes, you can use the model as part of a software application that accepts input and uses the output.

Chao January 27, 2019 at 3:28 am #

REPLY

Hi Jason

Thanks for the tutorial, I ran my train/test data with the default param on the xgboost and GradientBoostingClassifier from sklearn, they have same results but xgboost is slower than GB in terms of training and testing (around 30% difference).

It seems weird? is Xgboost supposed to be much faster than GBM from sklearn?

My laptop is a i7-5600u, it supposed to have 4 threads.

Thanks!

Jason Brownlee January 27, 2019 at 7:41 am #

REPLY

Perhaps it was not an apples to apples comparison, e.g. different model configuration?

Salman March 16, 2019 at 7:05 pm #

REPLY

Hi,
I am trying to convert my X and y into xgb,DMatix to make computation faster. My X has dimensions (1020, 421) and my y (1020,1).
I get an error and don’t know where my problem is.
I’d appreciate if you could help.

```
# Making xgDMatrix optimized dataset

dabsorb = xgb.DMatrix(absorb)
y = np.reshape(y,(-1, 1))
dy = xgb.DMatrix(y)

# Fitting XGBoost to the Training set
from xgboost import XGBClassifier
classifier = XGBClassifier ()
classifier.fit(dabsorb,dy)
```

I get this error:
File “C:\Users\AU529763\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\utils\validation.py”, line 797, in column_or_1d
raise ValueError(“bad input shape {0}”.format(shape))

ValueError: bad input shape ()

REPLY

Jason Brownlee March 17, 2019 at 6:17 am #

I’m not sure sorry, perhaps try posting to stackoverflow?

REPLY

Nirmine March 26, 2019 at 5:58 am #

hello Jason

I want to know what is the difference between the two codes? and which one do you advise me to use it?

```
# load data
# split data into (X_train, X_test, y_train, y_test)
from xgboost import XGBClassifier
model = XGBClassifier(learnin_rate=0.2, max_depth= 8,…)
eval_set = [(X_test, y_test)]
model.fit(X_train, y_train, eval_metric=”auc”, early_stopping_rounds=50, eval_set=eval_set, verbose=True)
y_pred = model.predict(X_test)
```

Code 2

```
# load data
# split data into (X_train, X_test, y_train, y_test)
import xgboost as xgb
dtrain = xgb.DMatrix(X_train,y_train)
dtest = xgb.DMatrix(X_test,y_test)
eval_set = [(X_test, y_test)]
param = {‘learnin_rate’:0.2,’max_depth’: 8, ‘eval_metric’:’auc’, ‘boost’:’gbtree’, ‘objective’: ‘binary:logistic’, …
}
```

```
num_round = 300
bst = xgb.train(param, dtrain, num_round)
```

Jason Brownlee March 26, 2019 at 8:13 am #

REPLY

Perhaps try both on your problem and use the one that results in the best performance on your dataset?

Eug May 23, 2019 at 7:24 am #

REPLY

How can I use Xgboost inside logistic regression.

Jason Brownlee May 23, 2019 at 2:30 pm #

REPLY

I don't know, sorry.

Eug May 23, 2019 at 11:16 pm #

REPLY

I heard we can use xgboost to extract the most important features and fit the logistic regression with those features. For example if we have a dataset of 1000 features and we can use xgboost to extract the top 10 important features to improve the accuracy of another model. such Logistic regression, SVM,… the way we use RFE.

Jason Brownlee May 24, 2019 at 7:53 am #

REPLY

You can use xgboost to give feature importance scores, then use the scores to select those most important features, then fit a model from those features.

Perhaps start here:
https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/

Luana Letícia May 28, 2019 at 12:29 pm #

REPLY

Thaaanks very much!!! So good explanation!!

Jason Brownlee May 28, 2019 at 2:43 pm #

REPLY

You're welcome, I'm glad it helped.

rajkamal May 31, 2019 at 3:35 pm #

REPLY

First of all, thank u so much of such great content. Actually, I've trying to implement a multi-class text classification, for that, I've tried to generate the word embeddings using the Word2Vec model, have u got any other suggestions to generate word embeddings ??

The other question I've got is, how am I supposed to handle the data which has both texts (which is not categorical) as well as numeric values? Have you got any worked out examples for this kind?

Thanks in advance.

Jason Brownlee June 1, 2019 at 6:09 am #

REPLY

My best advice on text classification is here:
https://machinelearningmastery.com/best-practices-document-classification-deep-learning/

For text and numeric data, you can use a multi-input model, this post will show you how:
https://machinelearningmastery.com/keras-functional-api-deep-learning/

Nada June 2, 2019 at 5:29 pm #

REPLY

Hi im working with a dataset with a shape of (7026,63) i tried to run xgboost, gradientboosting and adaboost classifiers on it however it returns a low accuracy rate i tried to tune the parameters a bit but stil ada gave me 60% and xgboost gave me 45% as for the gradient boosting it gave me 0.023 i would very much appreciate it if you coulx answer as to why its not working well.

Jason Brownlee June 3, 2019 at 6:37 am #

REPLY

I have suggestions on how to configure xgboost here that might help:
https://machinelearningmastery.com/start-here/#xgboost

Maryam July 24, 2019 at 4:44 am #

REPLY

Hi! Thanks for the useful post. I have a weird problem when it comes to rounding the y_pred in this line:

predictions = [round(value) for value in y_pred]

It apparently is a 2d array and python gives me an error saying:
Error “TypeError: type numpy.ndarray doesn’t define __round__ method”

Any chance you have encountered this error or know why that happens?

REPLY

Jason Brownlee July 24, 2019 at 8:15 am #

Sorry to hear that.

Perhaps try working with predictions directly without the rounding?

REPLY

Merik August 24, 2019 at 2:16 pm #

Hi Jason, thanks for the awesome post!

Is there a way to implement incremental/batched learning?

REPLY

Jason Brownlee August 25, 2019 at 6:31 am #

With Xgboost? Not sure off the cuff, sorry.

REPLY

Prem August 28, 2019 at 4:08 pm #

Hi Jason,
when I run prediction on xgboost model I get error as

ValueError: feature_names mismatch: [‘f0’, ‘f1’,….] [‘Application’, ‘Amount’….]

expected f20, f12,….. in input data

training data did not have the following fields: Application, Amount,……..

REPLY

Prem August 28, 2019 at 4:30 pm #

For one record, prediction happending, For the Test_Data, I am getting the above Error, (Train and Test data is not made from train_test_split, both are separate datasets)

Jason Brownlee August 29, 2019 at 6:00 am #

REPLY

Perhaps remove the heading from your CSV file? Or load the data without the column heading?

roger September 20, 2019 at 6:36 am #

REPLY

so, let's say that our researchers go back and acquire new data from this population, and now want you to feed that new data into your model to predict the risk of diabetes on the current population. Would you just split new_data in the same manner (z_train and z_test) and feed it into your refit your model?

“””
model.fit(X_train, y_train)
z_pred = model.predict(z_test)
accuracy = accuracy_score(z_test, predictions)
print(“Accuracy: %.2f%%” % (accuracy * 100.0))
“””

or would you just feed the entire dataset as is and judge it against y_test?

“””
z_pred = model.predict(new_data)
accuracy = accuracy_score(y_test, predictions)
print(“Accuracy: %.2f%%” % (accuracy * 100.0))

apologies for my lack of understanding, but a lot of tutorials stop at the point of an accuracy test and don't cover the 'what's next'.

thanks

Jason Brownlee September 20, 2019 at 1:33 pm #

REPLY

No, making predictions on new data involves fitting a model on all available labelled training data, then using that model to make predictions on new data where there is no label.

More details here:
https://machinelearningmastery.com/train-final-machine-learning-model/

Does that help?

Lucas September 24, 2019 at 5:24 am #

REPLY

Hi Jason, I am trying to build a simple XGBoost binary classifier using your model with my own dataset. The dataset I am working with has about 18000 inputs, 30 features, and 1 label for classes. By

making use of your code, when trying to compile predictions = [round(value) for value in y_pred], I get the error: type bytes doesn't define __round__ method.

Another issue is that when I run the model I always get the error: You appear to be using a legacy multi-label data representation. Sequence of sequences are no longer supported; use a binary array or sparse matrix instead – the MultiLabelBinarizer transformer can convert to this format.

Does this have to do with the way I am defining the features and targets for the training and testing samples? I am doing this by defining them as features = df.drop('class', axis=1) and targets = df['target_class'] and then I am defining the train and test sample size with X_train, X_test, y_train, y_test = train_test_split(features, targets, test_size=0.33, random_state=7).

Jason Brownlee September 24, 2019 at 7:53 am #

REPLY

No need to round any longer, I believe the API will correctly predict classes directly. e.g.

```
yhat = model.predict(newX)
```

I don't believe so, the example works fine. Running now on the latest version I get:

```
Accuracy: 77.95%
```

Perhaps double check you have all of the code and the latest version of the library:
https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me

Jack October 14, 2019 at 5:04 pm #

REPLY

I use XGBoost with one feature (attribute), and got this error:

IndexError Traceback (most recent call last)
in
1 # fit model on training data
2 model = XGBClassifier()
—-> 3 model.fit(X_train, y_train,sample_weight='None')
4 print(model)

~\Anaconda2\envs\mypython3\lib\site-packages\xgboost\sklearn.py in fit(self, X, y, sample_weight, eval_set, eval_metric, early_stopping_rounds, verbose, xgb_model, sample_weight_eval_set, callbacks)
717 evals = ()
718
–> 719 self._features_count = X.shape[1]
720
721 if sample_weight is not None:

IndexError: tuple index out of range

was it because I use only the only one attribute? How to fix it?

Thanks in advance

Jason Brownlee October 15, 2019 at 6:07 am #

REPLY

That is odd.

No, XGBoost can have one feature as input just fine.

Perhaps confirm your data is loaded correctly, and that you have 1 column with n rows.

Sophia Yue November 7, 2019 at 8:50 am #

REPLY

Hi Jason,
1). How to apply the model built in the article into production?
2)(.After we build the model, could you please point the direction or articles to Deploy Machine Learning Models?

Thanks,
Sophia

Jason Brownlee November 7, 2019 at 2:05 pm #

REPLY

A final model must be developed:
https://machinelearningmastery.com/train-final-machine-learning-model/

Then you can deploy your model, perhaps this will help:
https://machinelearningmastery.com/faq/single-faq/how-do-i-deploy-my-python-file-as-an-application

Pragya Sharma November 8, 2019 at 8:39 pm #

REPLY

Can you please give an example with XGBRegressor and it's parameters?

Jason Brownlee November 9, 2019 at 6:12 am #

REPLY

Yes, see thus tutorial:
https://machinelearningmastery.com/spot-check-machine-learning-algorithms-in-python/

Michał Bargiel November 12, 2019 at 3:53 am #

REPLY

Hey, thank you for the tutorial.

I played around with variables for learning and changing parameters of XGBClassifier did not improve accuracy, however, I decreased test_size to 0.14 (I was trying different values) and accuracy peaked at 84%. I used Python 3.6.8 with 0.9 XGBoost lib.

Do you think it varies because of improvements in the algorithm or was suggested size overfitting the results?

REPLY

Jason Brownlee November 12, 2019 at 6:44 am #

Test set might be too small.

Perhaps try k-fold cross-validation to estimate the model performance?

REPLY

Pieter Willaert December 15, 2019 at 4:11 am #

Hi,

I'm trying to run this snipit with my data, but my kernel keeps dying… I don't know why, i get no errors. Just a popup : Your kernel has died. Any suggestions on what to do?

REPLY

Jason Brownlee December 15, 2019 at 6:09 am #

Sorry to hear that.

Perhaps there is a problem with your development environment? This might help:
https://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/

REPLY

young.chan January 10, 2020 at 3:48 pm #

how to apply XGBoost in Time Series Prediction?

REPLY

Jason Brownlee January 11, 2020 at 7:20 am #

First transform lag observations into input features:
https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/

Reza January 20, 2020 at 2:32 am #

REPLY

Thanks for the tutorial
Btw, does the label must be in numeric?
Because my label is in str and always error

Jason Brownlee January 20, 2020 at 8:42 am #

REPLY

String labels must be label/integer encoded.

Reza January 24, 2020 at 5:49 pm #

REPLY

Thanks for the info ☺

Jason Brownlee January 25, 2020 at 8:31 am #

REPLY

You're welcome.

fede February 19, 2020 at 6:49 am #

REPLY

What if I want to label a single row with XGB ?
I've trained my XGB model on a dataset (cardiovascular disease from Kaggle) with 13 features +1 target (0/1).
I have an array with 13 values which I want to be predicted (1 row x 13 columns)

array_to_predict = [[…],[…]……]

print(model.predict(array_to_predict))

how must be initialized the array in order to be correctly predicted ?

Jason Brownlee February 19, 2020 at 8:08 am #

REPLY

Use argmax on the predicted probabilities.

Perhaps see this:
https://machinelearningmastery.com/make-predictions-scikit-learn/

Steven May 6, 2020 at 2:39 am #

REPLY

Hi Jason,

Is it possible to use support vector machines as base learners in the xgbclassifier? I tried out ‘gbtree’ and ‘gblinear’ and surprisingly ‘gblinear’ beats ‘gbtree’ in several metrics for my breast cancer classification dataset. Is that possible since ‘gblinear’ can only make linea relationships, while ‘gbtrees’ can also consider non-linear relationships?

Jason Brownlee May 6, 2020 at 6:28 am #

REPLY

No.

Ezgi June 20, 2020 at 2:53 am #

REPLY

Hi Jason,

I want to predict percentages, so I have target values in the range [0,1]. The problem is reg:linear gives output out of the range. I saw in stackoverflow, somebody suggested use reg:logistic with XGBRegressor() class. I tried reg:logistic and the results are really promising! But I don’t have a valid ground to do that. Do you think it is okay to apply reg:logistic or is it non-sense?
Thanks a lot!

Jason Brownlee June 20, 2020 at 6:17 am #

REPLY

Perhaps try it and also perhaps try calibrating the predicted probabilities.

Laís June 27, 2020 at 6:38 am #

REPLY

Hi Jason, I’m trying to use XGBClassifier but it won’t work.
I am working with a fraud detection dataset called Paysim (available on Kaggle)
This is part of my code:

class Classificacao:
def __init__(self, classif, model_name):
self.name = model_name
self.classifier = classif

def norm_under(self, normalizar, under):
if normalizar & under:
steps = [(‘Norma’, StandardScaler()), (‘over’, SMOTE(sampling_strategy=0.1)),

```
('under', RandomUnderSampler(sampling_strategy=0.5)), ('Class', self.classifier)]
elif normalizar:
steps = [('Norma', StandardScaler()), ('over', SMOTE(sampling_strategy=0.1)), ('Class', self.classifier)]
elif under:
steps = [('over', SMOTE(sampling_strategy=0.1)), ('under', RandomUnderSampler(sampling_strategy=0.5)),
('Class', self.classifier)]
else:
steps = [('over', SMOTE(sampling_strategy=0.1)), ('Class', self.classifier)]
return steps

def holdout(self, normalizar=False, under=False):
global X_train, y_train, X_test, y_test

steps = self.norm_under(normalizar, under)
pipeline = Pipeline(steps=steps)
pipeline.fit(X_train, y_train)
pred = pipeline.predict(X_test)
print('Acuracia do {}: {}' .format(self.name, accuracy_score(y_test, pred)))
print('Média da curva ROC_AUC do {}: {}' .format(self.name, mean(roc_auc_score(y_test, pred))))
print('F1 score do {}: {}' .format(self.name, f1_score(y_test, pred, average='macro')))
return pred

def crossvalidation(self, normalizar=False, under=False):
global X_train, y_train, X_test, y_test

steps = self.norm_under(normalizar, under)
pipeline = Pipeline(steps=steps)
kfold = StratifiedKFold(n_splits=10, random_state=42)
scorers = {'accuracy_score': make_scorer(accuracy_score),
'roc_auc_score': make_scorer(roc_auc_score),
'f1_score': make_scorer(f1_score, average='macro')
}
resultado = cross_validate(pipeline, X_train, y_train, scoring=scorers, cv=kfold)
for name in resultado.keys():
media_scorers = np.average(resultado[name])
print('{} do {}: {}' .format(name, self.name, media_scorers))
```

And when I do this: xxg =
Classificacao(xgb.XGBClassifier(objective='binary:logistic', n_estimator=10, seed=123), 'XGB')
xg.holdout(False, False)

or this: Classificacao(xgb.XGBClassifier(objective='binary:logistic', n_estimator=10, seed=123), 'XGB')
xg.crossvalidation(False, False)

I get this error message:

KeyError Traceback (most recent call last)
/usr/local/lib/python3.6/dist-packages/sklearn/metrics/_scorer.py in _cached_call(cache, estimator, method,
*args, **kwargs)
54 try:

—> 55 return cache[method]
56 except KeyError:

KeyError: ‘predict’

During handling of the above exception, another exception occurred:

ValueError Traceback (most recent call last)
19 frames
/usr/local/lib/python3.6/dist-packages/xgboost/core.py in _validate_features(self, data)
1688
1689 raise ValueError(msg.format(self.feature_names,
-> 1690 data.feature_names))
1691
1692 def get_split_value_histogram(self, feature, fmap=”, bins=None, as_pandas=True):

ValueError: feature_names mismatch: [‘f0’, ‘f1’, ‘f2’, ‘f3’, ‘f4’, ‘f5’, ‘f6’] [‘step’, ‘amount’, ‘oldbalanceOrg’, ‘newbalanceOrig’, ‘oldbalanceDest’, ‘newbalanceDest’, ‘TRANSFER’]
expected f1, f6, f3, f2, f0, f4, f5 in input data
training data did not have the following fields: oldbalanceDest, amount, oldbalanceOrg, step, TRANSFER, newbalanceOrig, newbalanceDest

---

Jason Brownlee June 27, 2020 at 2:07 pm #

REPLY

I’m sorry to hear that, perhaps some of these suggestions will help:
https://machinelearningmastery.com/faq/single-faq/can-you-read-review-or-debug-my-code

---

Sowmya July 10, 2020 at 9:56 pm #

REPLY

Thanks for the clear explaination. i am new to Machine learning.
I created a model with XGBRegressor and trained it. I would like to get the optimal bias and residual for each feature and use it in the front end of my app as linear regression. will that be possible? if so, How can I achieve it.
Thanks again for your help.

Jason Brownlee July 11, 2020 at 6:12 am #

REPLY

You’re welcome.

Sorry, I don’t understand what you mean by ” optimal bias and residual for each feature”, can you elaborate?

Ishita July 17, 2020 at 8:01 pm #

REPLY

Hi,
I really like the way you've explained everything but I'm unable to download the dataset. The link is opening the dataset but how do I download it?

Jason Brownlee July 18, 2020 at 6:01 am #

REPLY

Thanks.

Perhaps right click the link and choose save as.

Gokul October 4, 2020 at 3:37 pm #

REPLY

Can I get the equation of the line if I use XGBoost regressor?

Jason Brownlee October 5, 2020 at 6:49 am #

REPLY

No, an xgboost cannot be reduced (easily) to an equation. It is a large collection of weighted decision trees.

Ankit January 1, 2021 at 2:10 am #

REPLY

if I want to make prediction using xgboost and I have 6 feature as input then what will be user_input command to get on that prediction result?

model.predict( )

what can I put inside the paraenthesis?

I put the feature value in list [0,0,44,18,201,5430]

model.predict([0,0,44,18,201,5430])

but I get error?

plz give soluton

Jason Brownlee January 1, 2021 at 5:31 am #

REPLY

One row of data, e.g.:

```
row = [...]
yhat = model.predict(row)
```

You can learn more here:
https://machinelearningmastery.com/make-predictions-scikit-learn/

Priya February 1, 2021 at 8:10 pm #

REPLY

Hello sir,
I am facing problem in installing the XGBoost. I am getting an error ‘sudo is not recognized as an internal and external command’. Can you please help me to rectify this error.

Jason Brownlee February 2, 2021 at 5:43 am #

REPLY

Perhaps drop the “sudo” if you are on windows.

Jessy February 3, 2021 at 6:03 am #

REPLY

Hello Jason! Thank you for the *simple* explanation.

I’m using XGboost to train a multi-class dataset and I’m getting very poor overall accuracy (70%), However, when using SVM+TFIDF I got a better accuracy of 79%. Is it because of my high vector dimensions (using tri-grams) ? or maybe parameter tuning? Isn’t XGBoost supposed to perform better or even the same as SVM? but not worse

Jason Brownlee February 3, 2021 at 6:28 am #

REPLY

You’re welcome!

Perhaps xgboost is not well suited for your problem?
Perhaps some data preparation is required?
Perhaps some model tuning is required?

Jessy February 3, 2021 at 7:03 am #

REPLY

I’ve done extensive pre-processing but still my problem in overlapping words between my classes. Can you please recommend an algorithm that might help?

Jason Brownlee February 3, 2021 at 7:32 am #

REPLY

It is hard to know what algorithm will work best for a given dataset, instead, you must use systematic experiments to discover what works best.

Sriram February 28, 2021 at 3:48 am #

REPLY

hi Jason, Thank you for this useful article.

I have been trying to find suitable algorithm/library to implement solution for a learn-to-rank problem wherein the response variable has large values 1..200000 which needs to be ranked/trained.

I explored a lot on the web and came across options such as RankSVM, LamdaRank, XGBRanker, etc. but only to find that they don't actually work – either resulting in errors or are hard to implement(i.e., can't directly adapt to my problem).

As part of the DMLC implementation I came across the XGBRankerMixIn class. Can this be adapted to my solution ? Or could you suggest suitable references/implementations for my problem ?

Jason Brownlee February 28, 2021 at 4:36 am #

REPLY

You're welcome.

Sorry, I have not used it. I cannot give you good off the cuff advice. Hopefully I can write about the topic in the future.

Dinani April 22, 2021 at 5:11 pm #

REPLY

Have you found references/implementations for your problem?
Please let me know if u have some references, I have the same problem

Sriram February 28, 2021 at 5:05 am #

REPLY

Ok Jason. Thank you for your quick reply.

Jason Brownlee February 28, 2021 at 5:41 am #

REPLY

You're welcome.

Ritwic April 16, 2021 at 3:38 am #

REPLY

Hi! thanks for the article.

I have been doing exactly how you did it in the article. However, in google Colab, the code gets

```
from xgboost import XGBClassifier
xgb1 = XGBClassifier()
xgb1.fit(X_train,y_train)
```

Colab gets stuck on xgb1.fit(X_train,y_train). Will it take a lot of time to train or is there some error. I am getting no errors, it is just executing.

Jason Brownlee April 16, 2021 at 5:33 am #

REPLY

Perhaps try running the code on your own machine.

Akshar July 8, 2021 at 7:13 pm #

REPLY

Hi Jason,

I have this query regarding “subsample” parameter.

I ran the the classifier with the default values except “subsample”, which was taken as 0.9. And the accuracy came more than 79%.

However, upon re-running the same classifier multiple times, the accuracy were varying from 77% to 79% and that is because of the random selection of observations to build a boosting tree based on the subsample value. Correct me if I am wrong here.

Is there an option to control or giving seed values for XGBoost classifier when we keep subsample value less than 1? We do that in Train_Test_Split.

Thanks

Jason Brownlee July 9, 2021 at 5:07 am #

REPLY

Yes, that is correct, see this:
https://machinelearningmastery.com/different-results-each-time-in-machine-learning/

Akshar July 9, 2021 at 11:27 pm #

REPLY

Thanks Jason!

Your blogs are a big help for me. And reading those queries in the comment sections equally helps to get a deeper understanding.

Thanks once again ☺

Jason Brownlee July 10, 2021 at 6:11 am #

REPLY

You're very welcome!

Leave a Reply

Name (required)

Email (will not be published) (required)

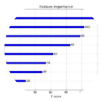
Website

SUBMIT COMMENT

Welcome!
I'm *Jason Brownlee* PhD
and I **help developers** get results with **machine learning**.
Read more

Never miss a tutorial:

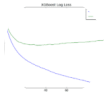
Picked for you:



Feature Importance and Feature Selection With XGBoost in Python

How to Develop Your First XGBoost Model in Python

Data Preparation for Gradient Boosting with XGBoost in Python



Avoid Overfitting By Early Stopping With XGBoost In Python

A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning

Loving the Tutorials?

The XGBoost With Python EBook is
where you'll find the ***Really Good*** stuff.

>> SEE WHAT'S INSIDE