How to Remove Outliers for Machine Learning

by **Jason Brownlee** on April 25, 2018 in **Data Preparation**

Last Updated on August 18, 2020

When modeling, it is important to clean the data sample to ensure that the observations best represent the problem.

Sometimes a dataset can contain extreme values that are outside the range of what is expected and unlike the other data. These are called outliers and often machine learning modeling and model skill in general can be improved by understanding and even removing these outlier values.

In this tutorial, you will discover outliers and how to identify and remove them from your machine learning dataset.

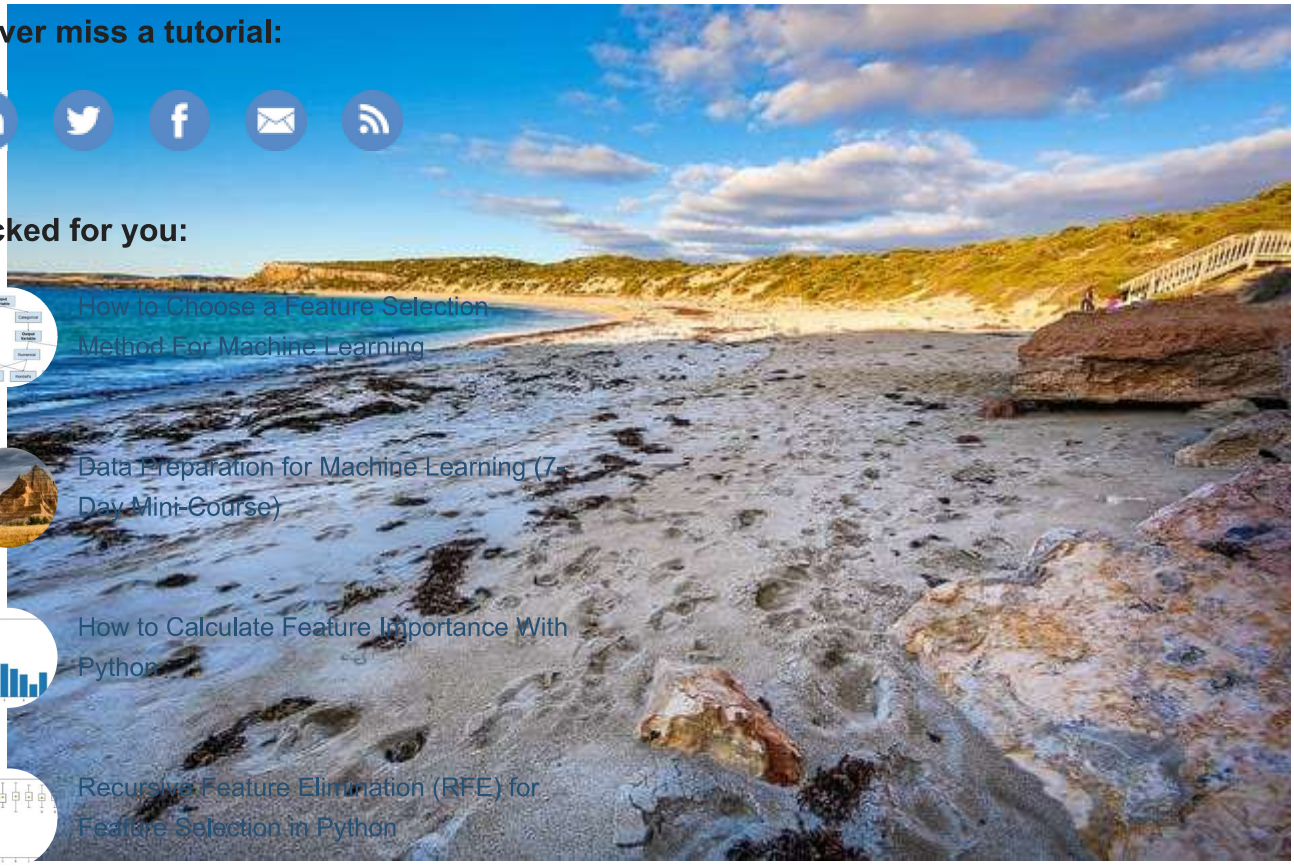
After completing this tutorial, you will know:

- That ... observation in a dataset and may have one of many causes.
- How to use simple univariate statistics like standard deviation and interquartile range to identify and remove outliers from a data sample.
- How to use an outlier detection model to identify and remove rows from a training dataset in order to lift predictive modeling performance.

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

Let's get started.

- **Update May/2018**: Fixed bug when filtering samples via outlier limits.
- **Update May/2020**: Updated to demonstrate on a real dataset.



How to Use Statistics to Identify Outliers in Data
Photo by Jeff Richardson, some rights reserved.

Tutorial Overview

This tutorial is divided into five parts; they are:

1. What are Outliers?
2. Test Dataset
3. Standard Deviation Method
4. Interc
5. Automatic Outlier Detection

What are Outliers?

An outlier is an observation that is unlike the other observations.

It is rare, or distinct, or does not fit in some way.

> *We will generally define outliers as samples that are exceptionally far from the mainstream of the data.*

— Page 33, Applied Predictive Modeling, 2013.

Outliers can have many causes, such as:

- Measurement or input error.

- Data corruption.
- True outlier observation (e.g. Michael Jordan in basketball).

There is no precise way to define and identify outliers in general because of the specifics of each dataset. Instead, you, or a domain expert, must interpret the raw observations and decide whether a value is an outlier or not.

Even with a thorough understanding of the data, outliers can be hard to define. […] Great care should be taken not to hastily remove or change values, especially if the sample size is small.

— Page 33, Applied Predictive Modeling, 2013.

Nevertheless, we can use statistical methods to identify observations that appear to be rare or unlikely given the available data.

Identifying outliers and bad data in your dataset is probably one of the most difficult parts of data cleanup, and it takes time to get right. Even if you have a deep understanding of statistics and how outliers might affect your data, it's always a topic to explore cautiously.

— Page 167, Data Wrangling with Python, 2016.

This does not mean that the values identified are outliers and should be removed. But, the tools described in this tutorial can be helpful in shedding light on rare events that may require a second look.

A good tip is to consider plotting the identified outlier values, perhaps in the context of non-outlier values to see if there are any systematic relationship or pattern to the outliers. If there is, perhaps they are not outliers and can be explained, or perhaps the outliers themselves can be identified more systematically.

Test Dataset

Before we look at outlier identification methods, let's define a dataset we can use to test the methods.

We will generate a population 10,000 random numbers drawn from a Gaussian distribution with a mean of 50 and a standard deviation of 5.

Numbers drawn from a Gaussian distribution will have outliers. That is, by virtue of the distribution itself, there will be a few values that will be a long way from the mean, rare values that we can identify as outliers.

We will use the *randn()* function to generate random Gaussian values with a mean of 0 and a standard deviation of 1, then multiply the results by our own standard deviation and add the mean to shift the values into the preferred range.

The pseudorandom number generator is seeded to ensure that we get the same sample of numbers each time the code is run.

```
# generate gaussian data
from numpy.random import seed
from numpy.random import randn
from numpy import mean
from numpy import std
# seed the random number generator
seed(1)
# generate univariate observations
data = 5 * randn(10000) + 50
# summarize
print('mean=%.3f stdv=%.3f' % (mean(data), std(data)))
```


Running the example generates the sample and then prints the mean and standard deviation. As expected, the values are very close to the expected values.

```
mean=50.049 stdv=4.994
```


Standard Deviation Method

If we know that the distribution of values in the sample is Gaussian or Gaussian-like, we can use the standard deviation of the sample as a cut-off for identifying outliers.

The Gaussian distribution has the property that the standard deviation from the mean can be used to reliably summarize the percentage of values in the sample.

For example, within one standard deviation of the mean will cover 68% of the data.

So, if the mean is 50 and the standard deviation is 5, as in the test dataset above, then all data in the sample between 45 and 55 will account for about 68% of the data sample. We can cover more of the data sample if we expand the range as follows:

- 1 Standard Deviation from the Mean: 68%
- 2 Standard Deviations from the Mean: 95%
- 3 Standard Deviations from the Mean: 99.7%

A value that falls outside of 3 standard deviations is part of the distribution, but it is an unlikely or rare event at approximately 1 in 370 samples.

Three standard deviations from the mean is a common cut-off in practice for identifying outliers in a Gaussian or Gaussian-like distribution. For smaller samples of data, perhaps a value of 2 standard deviations (95%) can be used, and for larger samples, perhaps a value of 4 standard deviations (99.9%) can be used.

e 19, Data Cleaning, 2019.

Let's make this concrete with a worked example.

nes, the data is standardized first (e.g. to a Z-score with zero mean and unit variance) so that the outlier detection can be performed using standard Z-score cut-off values. This is a convenience and is not required in general, and we will perform the calculations in the original scale of the data here to make things clear.

We can calculate the mean and standard deviation of a given sample, then calculate the cut-off for identifying outliers as more than 3 standard deviations from the mean.


```
# calculate summary statistics
data_mean, data_std = mean(data), std(data)
# identify outliers
cut_off = data_std * 3
lower, upper = data_mean - cut_off, data_mean + cut_off
```

We can then identify outliers as those examples that fall outside of the defined lower and upper limits.


```
...
# identify outliers
outliers = [x for x in data if x < lower or x > upper]
```

Alternately, we can filter out those values from the sample that are not within the defined limits.

```
...
# remove outliers
outliers_removed = [x for x in data if x > lower and x < upper]
```

We can put this all together with our sample dataset prepared in the previous section.

The complete example is listed below.

```
# identify outliers with standard deviation
from numpy.random import seed
from numpy.random import randn
from numpy import mean
from numpy import std
# seed the random number generator
seed(1)
# generate univariate observations
data = 5 * randn(10000) + 50
# calculate summary statistics
data_mean, data_std = mean(data), std(data)
```

```
# identify outliers
cut_off = data_std * 3
lower, upper = data_mean - cut_off, data_mean + cut_off
# identify outliers
outliers = [x for x in data if x < lower or x > upper]
print('Identified outliers: %d' % len(outliers))
# remove outliers
outliers_removed = [x for x in data if x >= lower and x <= upper]
print('Non-outlier observations: %d' % len(outliers_removed))
```


Running the example will first print the number of identified outliers and then the number of observations that are not outliers, demonstrating how to identify and filter out outliers respectively.

```
Identified outliers: 29
Non-outlier observations: 9971
```


So far we have only talked about univariate data with a Gaussian distribution, e.g. a single variable. You can use the same approach if you have multivariate data, e.g. data with multiple variables, each with a different Gaussian distribution.

You can imagine bounds in two dimensions that would define an ellipse if you have two variables. Observations that fall outside of the ellipse would be considered outliers. In three dimensions, this would be an ellipsoid, and so on into higher dimensions.

Alternately, if you knew more about the domain, perhaps an outlier may be identified by exceeding the limits on one or a subset of the data dimensions.

Interquartile Range Method

Not all data is normal or normal enough to treat it as being drawn from a Gaussian distribution.

A good statistic for summarizing a non-Gaussian distribution sample of data is the Interquartile Range, or IQR for short.

The IQR is calculated as the difference between the 75th and the 25th percentiles of the data and defines the box plot.

Remember that percentiles can be calculated by sorting the observations and selecting values at specific indices. The 50th percentile is the middle value, or the average of the two middle values for an even number of examples. If we had 10,000 samples, then the 50th percentile would be the average of the 5000th and 5001st values.

We refer to the percentiles as quartiles ("*quart*" meaning 4) because the data is divided into four groups via the 25th, 50th and 75th values.

The IQR defines the middle 50% of the data, or the body of the data.

> *Statistics-based outlier detection techniques assume that the normal data points would appear in high probability regions of a stochastic model, while outliers would occur in the low probability regions of a stochastic model.*

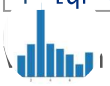
— Page 12, Data Cleaning, 2019.

The IQR can be used to identify outliers by defining limits on the sample values that are a factor *k* of the IQR below the 25th percentile or above the 75th percentile. The common value for the factor *k* is the value 1.5. A factor *k* of 3 or more can be used to identify values that are extreme outliers or "*far outs*" when described in the context of box and whisker plots.

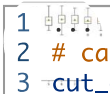
In a box and whisker plot, these limits are drawn as fences on the whiskers (or the lines) that are drawn from the box. Values that fall outside of these values are drawn as dots.

We can calculate the percentiles of a dataset using the *percentile()* NumPy function that takes the dataset and specification of the desired percentile. The IQR can then be calculated as the difference between the 75th and 25th percentiles.


```
...
# calculate interquartile range
q25, q75 = percentile(data, 25), percentile(data, 75)
iqr = q75 - q25
```



We can then calculate the cutoff for outliers as 1.5 times the IQR and subtract this cut-off from the 25th percentile and add it to the 75th percentile to give the actual limits on the data.




```
...
# calculate the outlier cutoff
cut_off = iqr * 1.5
lower, upper = q25 - cut_off, q75 + cut_off
```

We can then use these limits to identify the outlier values.

```
...
# identify outliers
outliers = [x for x in data if x < lower or x > upper]
```


We can also use the limits to filter out the outliers from the dataset.

```
...
# remove outliers
outliers_removed = [x for x in data if x > lower and x < upper]
```

We can tie all of this together and demonstrate the procedure on the test dataset.

The complete example is listed below.

```
# identify outliers with interquartile range
from numpy.random import seed
from numpy.random import randn
from numpy import percentile
# seed the random number generator
seed(1)
# generate univariate observations
data = 5 * randn(10000) + 50
# calculate interquartile range
q25, q75 = percentile(data, 25), percentile(data, 75)
iqr = q75 - q25
print('Percentiles: 25th=%.3f, 75th=%.3f, IQR=%.3f' % (q25, q75, iqr))
# calculate the outlier cutoff
cut_off = iqr * 1.5
lower, upper = q25 - cut_off, q75 + cut_off
# identify outliers
outliers = [x for x in data if x < lower or x > upper]
print('Identified outliers: %d' % len(outliers))
# remove outliers
```

```
20 outliers_removed = [x for x in data if x >= lower and x <= upper]
21 print('Non-outlier observations: %d' % len(outliers_removed))
```


Running the example first prints the identified 25th and 75th percentiles and the calculated IQR. The number of outliers identified is printed followed by the number of non-outlier observations.

```
1 Percentiles: 25th=46.685, 75th=53.359, IQR=6.674
2 Identified outliers: 81
3 Non-outlier observations: 9919
```


This can be used for univariate data by calculating the limits on each variable in the dataset in turn, and taking outliers as observations that fall outside of the rectangle or hyper-rectangle.

omatic Outlier Detection

In machine learning, an approach to tackling the problem of outlier detection is one-class classification.

ass Classification, or OCC for short, involves fitting a model on the "*normal*" data and predicting w[illegible]er new data is normal or an outlier/anomaly.

A one-class classifier aims at capturing characteristics of training instances, in order to be able to distinguish between them and potential outliers to appear.

— page 139, Learning from Imbalanced Data Sets, 2018.

A one-class classifier is fit on a training dataset that only has examples from the normal class. Once prepared, the model is used to classify new examples as either normal or not-normal, i.e. outliers or anomalies.

A simple approach to identifying outliers is to locate those examples that are far from the other examples in the feature space.

This can work well for feature spaces with low dimensionality (few features), although it can become less reliable as the number of features is increased, referred to as the curse of dimensionality.

The local outlier factor, or LOF for short, is a technique that attempts to harness the idea of nearest neighbors for outlier detection. Each example is assigned a scoring of how isolated or how likely it is to be outliers based on the size of its local neighborhood. Those examples with the largest score are more likely to be outliers.

> *We introduce a local outlier (LOF) for each object in the dataset, indicating its degree of outlier-ness.*

— LOF: Identifying Density-based Local Outliers, 2000.

The scikit-learn library provides an implementation of this approach in the LocalOutlierFactor class.

We can demonstrate the LocalOutlierFactor method on a predictive modelling dataset.

We will use the Boston housing regression problem that has 13 inputs and one numerical target and requires learning the relationship between suburb characteristics and house prices.

The dataset can be downloaded from here:

- Boston Housing Dataset (housing.csv)
- Boston Housing Dataset Details (housing.names)

Looking in the dataset, you should see that all variables are numeric.

```
0.00632,18.00,2.310,0,0.5380,6.5750,65.20,4.0900,1,296.0,15.30,396.90,4.98,24.00
0.02731,0.00,7.070,0,0.4690,6.4210,78.90,4.9671,2,242.0,17.80,396.90,9.14,21.60
0.02729,0.00,7.070,0,0.4690,7.1850,61.10,4.9671,2,242.0,17.80,392.83,4.03,34.70
0.03237,0.00,2.180,0,0.4580,6.9980,45.80,6.0622,3,222.0,18.70,394.63,2.94,33.40
0.06905,0.00,2.180,0,0.4580,7.1470,54.20,6.0622,3,222.0,18.70,396.90,5.33,36.20
...
```


Need to download the dataset, we will download it automatically.

First, we can load the dataset as a NumPy array, separate it into input and output variables and then split it into train and test datasets.

The complete example is listed below.

```
# load and summarize the dataset
from pandas import read_csv
from sklearn.model_selection import train_test_split
# load the dataset
url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/housing.csv'
df = read_csv(url, header=None)
# retrieve the array
data = df.values
# split into input and output elements
X, y = data[:, :-1], data[:, -1]
# summarize the shape of the dataset
print(X.shape, y.shape)
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1)
# summarize the shape of the train and test sets
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```


Running the example loads the dataset and first reports the total number of rows and columns in the dataset, then the data number of examples allocated to the train and test datasets.

```
(506, 13) (506,)
(339, 13) (167, 13) (339,) (167,)
```

It is a regression predictive modeling problem, meaning that we will be predicting a numeric value. All input variables are also numeric.

In this case, we will fit a linear regression algorithm and evaluate model performance by training the model on the test dataset and making a prediction on the test data and evaluate the predictions using the mean absolute error (MAE).

The complete example of evaluating a linear regression model on the dataset is listed below.

```
# evaluate model on the raw dataset
from pandas import read_csv
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
# load the dataset
url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/housing.csv'
df = read_csv(url, header=None)
# retrieve the array
data = df.values
# split into inpiut and output elements
X, y = data[:, :-1], data[:, -1]
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1)
# fit the model
model = LinearRegression()
model.fit(X_train, y_train)
# evaluate the model
yhat = model.predict(X_test)
# evaluate predictions
mae = mean_absolute_error(y_test, yhat)
print('MAE: %.3f' % mae)
```

g the example fits and evaluates the model then reports the MAE.

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.

In this case, we can see that the model achieved a MAE of about 3.417.

```
MAE: 3.417
```

Next, we can try removing outliers from the training dataset.

The expectation is that the outliers are causing the linear regression model to learn a bias or skewed understanding of the problem, and that removing these outliers from the training set will allow a more effective model to be learned.

We can achieve this by defining the LocalOutlierFactor model and using it to make a prediction on the training dataset to identify rows in the training dataset as normal (1) or an outlier (-1). We will use the default hyperparameters for the outlier detection model, although it is a good idea to tune the configuration to the specifics of your dataset.

```
...
# identify outliers in the training dataset
lof = LocalOutlierFactor()
yhat = lof.fit_predict(X_train)
```

We can then use these predictions to remove all outliers from the training dataset.

```
...
# select all rows that are not outliers
mask = yhat != -1
X_train, y_train = X_train[mask, :], y_train[mask]
```

We can then fit and evaluate the model as per normal.

The updated example of evaluating a linear regression model with outliers deleted from the training dataset is listed below.

```
# evaluate model on training dataset with outliers removed
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import LocalOutlierFactor
from sklearn.metrics import mean_absolute_error
# load the dataset
url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/housing.csv'
df = read_csv(url, header=None)
# retrieve the array
data = df.values
# split into inpiut and output elements
X, y = data[:, :-1], data[:, -1]
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1)
# summarize the shape of the training dataset
print(X_train.shape, y_train.shape)
# identify outliers in the training dataset
lof = LocalOutlierFactor()
yhat = lof.fit_predict(X_train)
# select all rows that are not outliers
mask = yhat != -1
X_train, y_train = X_train[mask, :], y_train[mask]
# summarize the shape of the updated training dataset
print(X_train.shape, y_train.shape)
# fit the model
model = LinearRegression()
model.fit(X_train, y_train)
# evaluate the model
yhat = model.predict(X_test)
# evaluate predictions
mae = mean_absolute_error(y_test, yhat)
print('MAE: %.3f' % mae)
```

Running the example fits and evaluates the linear regression model with outliers deleted from the training dataset.

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.

Firstly, we can see that the number of examples in the training dataset has been reduced from 339 to 305, meaning 34 rows containing outliers were identified and deleted.

We can also see a reduction in MAE from about 3.417 by a model fit on the entire training dataset, to about 3.356 on a model fit on the dataset with outliers removed.

```
(339, 13) (339,)
(305, 13) (305,)
MAE: 3.356
```

The Scikit-Learn library provides other outlier detection algorithms that can be used in the same way such as the IsolationForest algorithm. For more examples of automatic outlier detection, see the tutorial:

- 4 Automatic Outlier Detection Algorithms in Python

Extensions

This section lists some ideas for extending the tutorial that you may wish to explore.

- Develop your own Gaussian test dataset and plot the outliers and non-outlier values on a histogram.
- Test out the IQR based method on a univariate dataset generated with a non-Gaussian distribution.
- Choose one method and create a function that will filter out outliers for a given dataset with an arbitrary number of dimensions.

If you explore any of these extensions, I'd love to know.

Further Reading

This section provides more resources on the topic if you are looking to go deeper.

Tutorials

- How to Identify Outliers in your Data
- Automatic Outlier Detection Algorithms in Python

Books

- Applied Predictive Modeling, 2013.
- Data Cleaning, 2019.
- Data Wrangling with Python, 2016.

- seed() NumPy API
- randn() NumPy API
- mean() NumPy API
- std() NumPy API
- percentile() NumPy API

Article

- Outlier on Wikipedia
- Anomaly detection on Wikipedia
- 68–95–99.7 rule on Wikipedia
- Interquartile range
- Box plot on Wikipedia

Summary

In this tutorial, you discovered outliers and how to identify and remove them from your machine learning dataset.

Specifically, you learned:

- That an outlier is an unlikely observation in a dataset and may have one of many causes.
- How to use simple univariate statistics like standard deviation and interquartile range to identify and remove outliers from a data sample.

- How to use an outlier detection model to identify and remove rows from a training dataset in order to lift predictive modeling performance.

Never miss a tutorial:

Do you have any questions?

Ask your questions in the comments below and I will do my best to answer.

Picked for you:

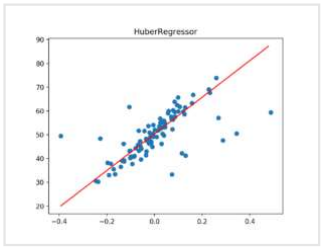
More On This Topic

4 Automatic Outlier Detection Algorithms in Python

How to Create Custom Data Transforms for Scikit-Learn

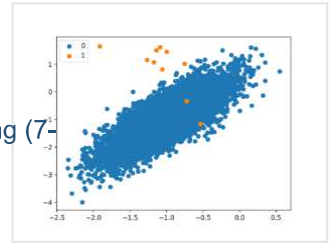
Never miss a tutorial:

Picked for you:



Robust Regression for Machine Learning in Python

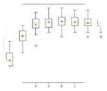
How to Choose a Feature Selection Method For Machine Learning



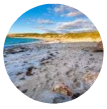
Data Preparation for Machine Learning (7-Day Mini-Course)

One-Class Classification Algorithms for Imbalanced Datasets

How to Calculate Feature Importance With Python

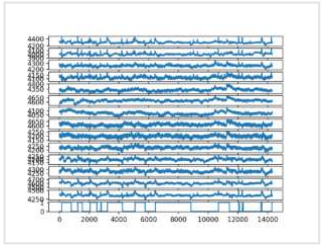


Recursive Feature Elimination (RFE) for Feature Selection in Python



How to Remove Outliers for Machine Learning

How to Identify Outliers in your Data



Loving the Tutorials?

The Data Preparation EBook is where you'll find the ***Really Good*** stuff.

Predict Whether a Persons Eyes are Open or Closed…

>> SEE WHAT'S INSIDE

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 →

‹ Introduction to Random Number Generators for Machine Learning in Python

How to Calculate Correlation Between Variables in Python ›

116 Responses to *How to Remove Outliers for Machine Learning*

Nitin Panwar April 25, 2018 at 5:05 pm #

REPLY

Nice explained. well done.

Jason Brownlee April 26, 2018 at 6:21 am #

REPLY

Thanks.

Haneesh June 27, 2019 at 2:32 am #

REPLY

Hello, can you explain me in R, how to find out how many outliers exists in one variable using Q1-1.5*IQR & Q3+1.5*IQR. Please help me on this only in R as I'm new to analysis.

Jason Brownlee June 27, 2019 at 7:58 am #

REPLY

Sorry, I don't have an example of this in R.

a.k December 4, 2020 at 4:02 am #

Hi, I am considering buying ur book Data Preparation for ML that u mention at the end of the article.

I want to ask if this tutorial about ouliers is included in the book in more detail.

Do your books include all the blog posts generally in more detail perhaps?

Jason Brownlee December 4, 2020 at 6:44 am #

The book does include a version of this tutorial.

More on the difference between books and posts can be found here: https://machinelearningmastery.com/faq/single-faq/how-are-your-books-different-from-the-blog

Rissy January 12, 2021 at 4:40 am #

REPLY

usa el metodo summary(variable)

Marish April 26, 2018 at 11:06 am # REPLY

Jason Brownlee April 26, 2018 at 3:02 pm # REPLY

I'm glad to hear that.

talha anwar April 27, 2018 at 3:48 am # REPLY

Once i remove the outlier, how can i fill the space left by that outlier. Becuase in other features length is more than in outlier removed features

Jason Brownlee April 27, 2018 at 6:09 am # REPLY

The entire record could be removed.

Alternately, the value in the record could be removed, and then imputed:
https://machinelearningmastery.com/handle-missing-data-python/

Sourav Mahavana April 27, 2018 at 5:16 am # REPLY

Jason's Brownlee articles and content are amazing as always

Jason Brownlee April 27, 2018 at 6:15 am # REPLY

Thanks!

Vishesh sharma April 27, 2018 at 5:35 am # REPLY

If suppose we have 50 features and we run this for each of the features then then the won't the number of rows I would have to delete be a lot because of missing data?

Also would dbscan be preferable to this?

Jason Brownlee April 27, 2018 at 6:16 am # REPLY

It may be. It is a very simple/rough method, perhaps not suitable for large numbers of features.

Alternately, obs could be deleted and the missing values imputed.

REPLY

jimmy April 27, 2018 at 10:20 am #

I liked your post, I think would be better with plotting.

REPLY

Jason Brownlee April 27, 2018 at 2:27 pm #

Thanks for the suggestion.

REPLY

Yishai E April 29, 2018 at 12:26 am #

Your code has a flaw – especially for the quantile example, which define the outlier borders based on data points from the dataset. If your outliers are >< from the border and your non-outliers are , then your borders are missing from both sets.

REPLY

Jason Brownlee April 29, 2018 at 6:28 am #

What do you mean exactly, can you give a concrete example?

REPLY

peter May 7, 2018 at 6:00 am #

I assume Yishai means that we need to add a '>=` and '<=' in the code to include samples that are equal to upper/lower.

REPLY

Mukund April 30, 2018 at 11:38 pm #

Hi Dr.Jason.

Thanks for all the tips and I have been following your posts for a long time.

I don't know, if this is the right forum to ask my following question. I am trying to evaluate various classifier algorithims, like decision tree , ADtree etc for a particular problem of detecting whether a candidate is Autistic or not, using very well known interview questionnaire ADI-R. Various literature claim to use A algorithim or B algorithim to show how they could use reduce the question sets (original 99 questions) and yet achieve great accuracy. Many literature state Adtree is best for this purposes. Yet, Adtree has its own limitation. I am confused. Could you kindly, explain what is the best way to proceed, given the complexity of this problem.

Jason Brownlee May 1, 2018 at 5:34 am #

REPLY

at is pro exactly?

Brad Smith May 16, 2018 at 5:43 am #

REPLY

I've been thinking about the Standard Deviation method, and how some people have suggested that a very large outlier could skew the mean and standard deviation enough to interfere with outlier moval.

ever, couldn't this problem be mitigated by comparing each value to bounds that come from the mean and standard deviation of all the *other* values (leaving out the one value that you're currently on in the list)? If the one value that you're currently looking at is an outlier, then it will be left out of the mean standard deviation calculations, making it much more likely to be deemed an outlier, even if it is a large value.

This method may have a cost when it comes to efficiency, but the cost may be worth it depending on the lication. Thanks!

Jason Brownlee May 16, 2018 at 6:10 am #

REPLY

How to Remove Outliers for Machine Learning

It might be easier to visually inspect plots of the data prior to calculating limits to ensure they make sense.

Kevin Arvai May 25, 2018 at 5:05 am #

REPLY

ison. It inspired me to write a Kaggle kernel exploring the topic in more d tandard deviation and IQR methods 🙂
https://www.kaggle.com/kevinarvai/outlier-detection-practice-uni-multivariate

Jason Brownlee May 25, 2018 at 9:32 am #

REPLY

Well done! That is a very impressive kernel Kevin.

Tobi Adeyemi May 31, 2018 at 1:45 am #

REPLY

Hi Jason, are these methods covered in your new text; Statistical Methods for Machine Learning?

Jason Brownlee May 31, 2018 at 6:21 am #

REPLY

y a e m ds I think you need to know how to use when working through an applied machine learning project.

Bhukya Neeharika June 5, 2018 at 8:02 pm #

REPLY

Respected sir,

ave issue with drawing boxplot in python using iqr method,where i know an minimum maximum,q1,q3.could you please him sir.

Jason Brownlee June 6, 2018 at 6:40 am #

REPLY

Perhaps the API will help:

https://matplotlib.org/api/_as_gen/matplotlib.pyplot.boxplot.html

Bhukya Neeharika June 6, 2018 at 6:06 pm #

REPLY

Respected sir,

i couldn't understand that.could you please explain me in detail

Aman August 7, 2018 at 3:13 am #

REPLY

I have data where the standard deviation is very close to the mean. So when I do the :

lower = mean – cutoff

it gives me a negative number. Is this alright? My data does not contain values less than 0.

Jason Brownlee August 7, 2018 at 6:33 am #

REPLY

Perhaps this methods is not suitable for your data?

Matheus September 21, 2018 at 3:02 am #

REPLY

Hi Jason,

When you say that the data needs to be standardized first, are you referring to data transformation (Normalization, StandartScaler, Box-cox)?

Never miss a tutorial:

Jason Brownlee September 21, 2018 at 6:31 am #

REPLY

Standardization explicitly, zero mean and unit standard deviation.

Picked for you:

Felix October 10, 2018 at 1:16 am #

REPLY

Hi Jason,

nk your for your expertise!

the following TypeError using your IRM code:

```
TypeError Traceback (most recent call last)
ver, upper = q25 – cutoff, q75 + cutoff
 # identify outliers
—> 11 outliers = [x for x in dfg if x upper]
rint(‘Identified outliers: %d’ % len(outliers))
remove outliers

in (.0)
lower, upper = q25 – cutoff, q75 + cutoff
identify outliers
> 11 outliers = [x for x in dfg if x upper]
12 print(‘Identified outliers: %d’ % len(outliers))
13 #remove outliers
```


TypeError: ‘>’ not supported between instances of ‘numpy.ndarray’ and ‘str’

:ober 10, 2018 at 6:12 am #

REPLY

Sorry to hear that, I have some suggestions for you here: https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me

Ravinder Ahuja November 12, 2018 at 6:55 am #

REPLY

Can you please put a post for replacing outlier with median using python..

Thanks

Jason Brownlee November 12, 2018 at 2:06 pm #

REPLY

Thanks for the suggestion.

Sa No er 25 18 at 4:25 pm #

REPLY

Thanks a lot, it is helpful.

Jason Brownlee November 26, 2018 at 6:15 am #

REPLY

I'm happy to hear that.

Srinivasa Rao Raghupatruni December 18, 2018 at 10:20 pm #

REPLY

Hi Jason,

Thank you for the wonderful article. I have implemented the above for my dataset. But when doing train_test split, I'm getting the below error:

Error: Input variables with inconsistent numbers of samples: [459, 489].

Please suggest how to resolve the unequal shapes

Jason Brownlee December 19, 2018 at 6:34 am #

REPLY

I'm sorry to hear that. I have some suggestions here:
https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me

Bijoy January 25, 2019 at 5:30 pm #

REPLY

Hi Jason,

Thanks for the wonderful work that you have been doing. I have just started working on ML to solve some problems we have.

Recently I have been trying to use ML to detect problems in machines(like motors) based on the vibration data collected from them. This will be a time series data. When the machine starts wearing out, the vibration data starts spiking. So ideally, the data would be all healthy and as the machine runs over a period of time, the vibration data would slowly start changing. The ML algo should find these deviations as they happen. What would you recommend to solve this kind of scenario? Statistical methods, ARIMA, NN ….Thanks in advance.

Jason Brownlee January 26, 2019 at 6:08 am #

REPLY

I recommend looking into "change detection" algorithms.

Never miss a tutorial:

REPLY

don jiang February [illegible], 2019 at 9:58 am #

Very clear introduction to outliers and practical codes. Thanks~

Picked for you:

How to Choose a Feature Selection Method For Machine Learning

REPLY

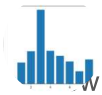
Jason Brownlee February 17, 2019 at 6:29 am #

Thanks, I'm glad it helped.

Data Preparation for Machine Learning (7-Day Mini-Course)

REPLY

Harriet April 8, 2019 at 8:01 pm #

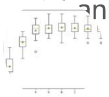


How to Calculate Feature Importance With Python

Hi,

[illegible]w would you justify using the interquartile range method over other methods to identify outliers? I.e, does it have any particular strengths, and in what circumstances would we use it over others?

[illegible]anks



Recursive Feature Elimination (RFE) for Feature Selection in Python

REPLY

Jason Brownlee April 9, 2019 at 6:23 am #

How to Remove Outliers for Machine Learning

It is simple and well understood.

Other methods may be complex and poorly understood.

Loving the Tutorials?

The Data Preparation EBook is where you'll find the ***Really Good*** stuff.

REPLY

Disha April 8, 2019 at 10:40 pm #

>> SEE WHAT'S INSIDE

Can you please tell which method to choose – Z score or IQR for removing outliers from a dataset. Dataset is a likert 5 scale data with around 30 features and 800 samples and I am trying to cluster the data in groups.
If I calculate Z score then around 30 rows come out having outliers whereas 60 outlier rows with IQR. I am confused as which one to prefer.
Thanks.

REPLY

Jason Brownlee April 9, 2019 at 2:43 pm #

Perhaps try a suite of values, evaluate their effect on the data and choose a value that result in the desired effect.

You might want to plot the results, e.g outliers vs non-outliers.

Anna May 20, 2019 at 2:07 am #

REPLY

Hi,

Is it usefull to use this method when I only have 6 datapoints?
Or what is the minimum I need?

Thank you!

Jason Brownlee May 20, 2019 at 6:33 am #

REPLY

Probably not. At least 30 points.

Srinivasa V June 15, 2019 at 4:43 pm #

REPLY

Well Explained!

is case, we have removed the outliers
suppose we want to replace outliers with NAN how to do this

ld you explain the same

nks in Advance

Jason Brownlee June 16, 2019 at 7:11 am #

REPLY

You can use the index to index the outlier values and set the values at those indexes to anything you wish, such as NaN.

You can use the replace() function, I give an example here:
https://machinelearningmastery.com/handle-missing-data-python/

Artur October 20, 2019 at 5:12 am #

REPLY

Hello Jason,

as I understand, with
“outliers = [x for x in data if x upper]”
we get a list of the outlier-values (NOT the index).
Suppose that we have a multivariable DataFrame, how do we get the position of the outliers?
Meaning: Can we get a list with the indices of the outliers, so that we just drop them?

Many thanks for your interesting article!

Artur

Jason Brownlee October 20, 2019 at 6:25 am # REPLY

...hap...e th...ere() numpy function?

Artur October 23, 2019 at 2:41 am # REPLY

Thx Jason,

thouht so too, but so far the np.where() func only gives me the position of outliers in my first column of interest. Maybe there is a problem with my loop..

But I figured an alternative 🙂

Jason Brownlee October 23, 2019 at 6:54 am # REPLY

Happy to hear that.

Kreecha November 12, 2019 at 11:40 am # REPLY

Do you have a reference for this of your statement: “ A good statistic for summarizing a non-Gaussian distribution sample of data is the Interquartile Range, or IQR for short.”. Personally, I am not sure IQR is suitable for all non Gaussian, but I would like to learn more if you can provide a reference.

Anyway thank for writing good articles.

...vember 12, 2019 at 2:06 pm # REPLY

It’s a heuristic more than a rule, e.g. not in all cases.

Any good book on stats will describe this method.

Also see this:
https://en.wikipedia.org/wiki/Interquartile_range#Outliers

Karthikeyan April 28, 2020 at 9:12 pm # REPLY

How to find an outlier in a multivariate data as each feature has its own values.

Jason Brownlee April 29, 2020 at 6:25 am # REPLY

Perhaps start with a univariate approach for each feature?

Loving the Tutorials?

REPLY

Karthikeyan April 30, 2020 at 5:38 am #

Hi, Jason thanks for your quick response. I can able to extract outliers in univariate data as per the instruction mentioned above, but how to apply this procedure in Multivariate data. Is there any way to detect outliers in multivariate data by considering all the instances in a pattern?

REPLY

Jason Brownlee April 30, 2020 at 6:54 am #

You may need to check the literature for multivariate methods once you exhaust univariate methods.

REPLY

Shreya February 12, 2020 at 2:18 am #

Can box plot or histogram be applied to find ouliers on whole dataset i.e. every data columns present in the dataset as a whole or we will have to apply it on every single column ?

REPLY

Jason Brownlee February 12, 2020 at 5:50 am #

Yes, but it is applied one column at a time.

REPLY

Shreya February 18, 2020 at 2:24 am #

Are there any ways in which instead of removing the outliers, we could replace them with some values so that the shape of our dataset will not be changed ?

REPLY

Jason Brownlee February 18, 2020 at 6:22 am #

Perhaps, but why?

Shreya February 18, 2020 at 3:48 pm #

Because if we remove the outliers then the number of data in all the columns of the dataset would be different which could create difficulty in training the model.

Jason Brownlee February 19, 2020 at 7:56 am #

Tes d confirm.

REPLY

Vedashree March 24, 2020 at 6:55 pm #

How to Choose a Feature Selection Method For Machine Learning

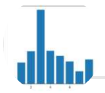
Hi Jason,

This article is very helpful. Thanks.

Data Preparation for Machine Learning (7-Day Mini-Course)

ve one doubt. While considering IQR logic, how do we decide the value of constant? Why is it rally considered as 1.5?

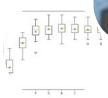
Also, Can we define some kind of relationship between this constant and the IQR value in order to get data-dependent value for the constant?



How to Calculate Feature Importance With Python

REPLY

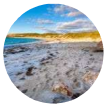
Jason Brownlee March 25, 2020 at 6:29 am #



Recursive Feature Elimination (RFE) for Feature Selection in Python

Thanks.

The constant is 1.5 as a standard definition. No need to change it – it is already data independent.



How to Remove Outliers for Machine Learning

REPLY

Ankit April 16, 2020 at 4:50 pm #

Can we detect outliers with django/regular expression/histogram? Or with all above three functions?

il 17, 2020 at 6:14 am #

Perhaps test them out and see?

REPLY

MF May 28, 2020 at 7:25 pm #

Hi Jason,

Why outliers are only removed from training dataset ? And not test dataset too ?
What happen during prediction, the model faces outliers in the test dataset ?

Please give your advice. Thank you.

REPLY

Jason Brownlee May 29, 2020 at 6:29 am #

If you remove outliers from the test data you will not give any prediction for them.

This may or may not be desirable depending on the goals of your project.

Edivaldo June 1, 2020 at 7:48 am #

REPLY

Hi,

article is excellent. Jason always explain much fine.

Jason Brownlee June 1, 2020 at 1:38 pm #

REPLY

Thanks!

Jeremy July 22, 2020 at 6:32 pm #

REPLY

Hi Jason,

Appreciate that this is a ‘how-to’ article but I think you glossed over the potential problems associated with outlier removal a bit, and it would be useful to give some more detail.

Outlier removal can be an easy way to make your data look nice and tidy but it should be emphasised that, in many cases, you’re removing useful information from the data set. This is especially true in small (n<100) data sets. Instead of discarding them and moving on to the fun stuff, I use outliers as a hint that I need to dig into the data and understand my problem space better.

Jason Brownlee July 23, 2020 at 6:04 am #

REPLY

Like other transforms, test and confirm that it lifts skill of your modeling pipeline on your test harness.

vishnu December 1, 2020 at 4:36 pm #

REPLY

As we removed the outliers from the training data, why shouldn’t we also detect and remove outliers from testing data too, to get better results?

Jason Brownlee December 2, 2020 at 7:37 am #

REPLY

You can, if you think that is a valid way to evaluate your model. E.g. the system would report “cannot predict” or similar.

Ashfaque Salman T K December 2, 2020 at 12:39 am # REPLY

Hi Jason, Excellent as always!!! 🙂

I have a doubt regarding the standard deviation test, it is generally applied for normal distributions.

But is it a right way to make a skewed data normal and then find the outliers using the same method?

Jason Brownlee December 2, 2020 at 7:48 am # REPLY

Yes, you can make data normal first then use that method.

Ashfaque Salman T K December 4, 2020 at 3:45 am # REPLY

then when we will use inter quartile range method?

Jason Brownlee December 4, 2020 at 6:43 am # REPLY

When the data distribution is not gaussian.

Ashfaque Salman T K December 4, 2020 at 3:44 pm #

ok, then , for a skewed data, what method should we normally use? iqr or std method?
or the choice really depend on the problem at hand?

Jason Brownlee December 5, 2020 at 8:02 am #

It always depends.

One approach would be to use a power transform then gaussian method. Another approach would be to use the IQR method directly.

Use whatever gives the best resulting predictive modeling.

Elvin Aghammadzada December 3, 2020 at 5:55 am # REPLY

well explained!

Malek December 5, 2020 at 9:33 am #

REPLY

Your blog is a must for anyone new to machine learning, thanks a lot.

Jason Brownlee December 6, 2020 at 6:59 am #

REPLY

Thanks!

Satish Kumar Dubey January 16, 2021 at 7:52 am #

REPLY

Dear Brownlee, Seems like this demo is based on dataset which either series or numpy array (basically single column) and that is fine you can remove outlier for single column.

But what happens when we have pandas dataframe and each column has different number of outliers and how you deal with removal of outliers? In this case we remove outliers on single column (for example), and it will impact entire records on row level. Meaning if we consider outliers from all columns and remove outliers each column , we end up with very few records left in dataset. Meaning removing outliers for one column impact other columns.

What I am trying to say is the outlier is detected on column level but removal are on row level. which destroy the dataset.

Do you still think removing outlier is practical in machine learning? What is your thoughts on this?

Br,
Satish

Jason Brownlee January 16, 2021 at 8:03 am #

REPLY

You can apply the same method to each variable.

Alternately, you can try more sophisticated methods:
https://machinelearningmastery.com/model-based-outlier-detection-and-removal-in-python/

Whether it is effective to remove outliers depends on the dataset and the model being used. Perhaps try it and compare results to working with the raw dataset.

cidakada February 4, 2021 at 9:32 pm #

REPLY

hi jason , I has run your code, but with my datafram
the question is how do I export the machine learning result todataframe format to excel?

REPLY

Jason Brownlee February 5, 2021 at 5:38 am #

You can save your results to CSV to load later in excel, this will show you how in python:
https://machinelearningmastery.com/how-to-save-a-numpy-array-to-file-for-machine-learning/

REPLY

cidakada March 8, 2021 at 9:51 pm #

first of all, thank you for your reply, sir, but what I'm trying to say is, when I try to print (y_test, yhat) the output result is an array, but the initial input is dataframe , my question is how to produce a dataframe with a desirable outlier without output?

REPLY

Jason Brownlee March 9, 2021 at 5:19 am #

You can create a dataframe from an array directly.

Perhaps this will help:
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

REPLY

Ethan February 23, 2021 at 7:01 am #

Hello, is there a way to increase the lower bound and upper bound to detect fewer outliers, depending on the nature of the data?

REPLY

Ethan February 23, 2021 at 7:14 am #

I suppose choosing k =3 for Q1-3xIQR Q3 + 3xIQR can be an option?

REPLY

Jason Brownlee February 23, 2021 at 7:38 am #

Sure, you can specify anything you want. Test and see if it lifts model skill.

REPLY

Jason Brownlee February 23, 2021 at 7:37 am #

Yes, you can set the bounds to be anything you like based on stats or domain knowledge or whim.

Nit arch 021 2 pm #

REPLY

Great tutorial , thanks for the amazing work sir . just want to share one quick tip ..with sklearn we can split and return x-train, x_test with these 3 lines of code. For eg in Boston housing data case

```
m sklearn.datasets import load_boston
= load_boston(return_X_y=True)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=23)
```

Jason Brownlee March 20, 2021 at 5:19 am #

REPLY

You're welcome.

Thanks for sharing.

Kevin May 13, 2021 at 8:20 am #

REPLY

Does necessary to remove outliers in deep learning?

Jason Brownlee May 14, 2021 at 6:15 am #

REPLY

It depends on the data and the model.

Experiment with and without outlier removal for your model and data and discover what works best for you.

Jasbir Singh June 9, 2021 at 7:13 pm #

REPLY

Hi,

Can we apply IQR rules multiple times? is it recommended?

Jason Brownlee June 10, 2021 at 5:24 am #

REPLY

You can use it on each variable.

Melika July 19, 2021 at 9:26 am #

REPLY

Hi Jason,

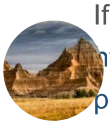
I tested two scaling methods (MinMaxScaler and RobustScaler) on the same MLP model. With MinMaxScaler model predicts very well, but with RobustScaler it doesn't perform well. What could be the reason?

How to Choose a Feature Selection Method For Machine Learning

Jason Brownlee October 20, 2021 at 5:31 am # REPLY

The cause will be the difference in the scaling method used on the input data.

If you're asking why are neural nets impacted by the scale of input, then perhaps see this:



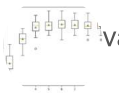
Data Preparation for Machine Learning (7-Day Mini-Course)

https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/

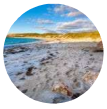
How to Calculate Feature Importance With Python

Mona October 19, 2021 at 12:09 am # REPLY

Thanks for the great content. I am wondering however, which methods works best in variable dataset, considering all features together.



Recursive Feature Elimination (RFE) for Feature Selection in Python



How to Remove Outliers for Machine Learning

Adrian Tam October 20, 2021 at 9:51 am # REPLY

A lot of models work. What did you tried? If your data is not very complicated, try a decision tree or SVM first.

PAdmin November 20, 2021 at 2:34 am # REPLY

. I really enjoy reading. very clear explanation. Thank you

john November 27, 2021 at 9:06 pm # REPLY

Hi Dr Jason, I have a dataframe with 14 numerical features . I was able to detect the outliers . But i don 't know when is it safe to delete a row containing outliers. Seeing that i hava 14 features i deleted every row containing 7 or more outlier values . Is my reasoning correct ? Any advice would help. Thanks for your hard work.

Adrian Tam November 29, 2021 at 8:48 am # REPLY

Correct – but try also count the number of rows you deleted. By definition of an outlier, I would not expect to have 20% (for example) of the entire dataset as outliers.

Leave a Reply

Never miss a tutorial:

Picked for you:

How to Choose a Feature Selection Method For Machine Learning

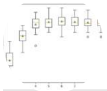
Data Preparation for Machine Learning (7-Day Mini-Course)

Name (required)

Email (will not be published) (required)

How to Calculate Feature Importance With Python

SUBMIT COMMENT



Recursive Feature Elimination (RFE) for Feature Selection in Python

How to Remove Outliers for Machine Learning

Welcome!

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

Read more

Loving the Tutorials?

The Data Preparation EBook is where you'll find the ***Really Good*** stuff.

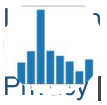
>> SEE WHAT'S INSIDE

Never miss a tutorial:

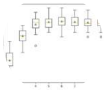
Picked for you:

How to Choose a Feature Selection Method For Machine Learning

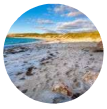
Data Preparation for Machine Learning (7-Day Mini-Course)



How to Calculate Feature Importance With Python



Recursive Feature Elimination (RFE) for Feature Selection in Python



How to Remove Outliers for Machine Learning