Statistical Methods for Machine Learning

7-Day Crash-Course

Jason Brownlee

MACHINE LEARNING MASTERY

Statistical Methods for Machine Learning Crash Course

© Copyright 2020 Jason Brownlee. All Rights Reserved.

Edition: v1.7

Find the latest version of this guide online at: http://MachineLearningMastery.com

Contents

Before We Get Started...

Statistics is a field of mathematics that is universally agreed to be a prerequisite for a deeper understanding of machine learning. Although statistics is a large field with many esoteric theories and findings, the nuts and bolts tools and notations taken from the field are required for machine learning practitioners. With a solid foundation of what statistics is, it is possible to focus on just the good or relevant parts. In this crash course, you will discover how you can get started and confidently read and implement statistical methods used in machine learning with Python in 7 days. Let's get started.

Who Is This Crash-Course For?

Before we get started, let's make sure you are in the right place. This course is for developers that may know some applied machine learning. Maybe you know how to work through a predictive modeling problem end-to-end, or at least most of the main steps, with popular tools. The lessons in this course do assume a few things about you, such as:

You need to know:

- You know your way around basic Python for programming.
- You may know some basic NumPy for array manipulation.
- You want to learn statistics to deepen your understanding and application of machine learning.

You do NOT need to know:

- You do not need to be a math wiz!
- You do not need to be a machine learning expert!

This crash course will take you from a developer that knows a little machine learning to a developer who can navigate the basics of statistics. This crash course assumes you have a working Python3 SciPy environment with at least NumPy installed. If you need help with your environment, you can follow the step-by-step tutorial here:

- How to Setup a Python Environment for Machine Learning and Deep Learning

Crash-Course Overview

This crash course is broken down into 7 lessons. You could complete one lesson per day (recommended) or complete all of the lessons in one day (hardcore). It really depends on the time you have available and your level of enthusiasm. Below is a list of the 7 lessons that will get you started and productive with statistics for machine learning in Python:

- **Lesson 01**: Statistics and Machine Learning.
- **Lesson 02**: Introduction to Statistics.
- **Lesson 03**: Gaussian Distribution and Descriptive Stats.
- **Lesson 04**: Correlation Between Variables.
- **Lesson 05**: Statistical Hypothesis Tests.
- **Lesson 06**: Estimation Statistics.
- **Lesson 07**: Nonparametric Statistics.

Each lesson could take you 60 seconds or up to 30 minutes. Take your time and complete the lessons at your own pace. Ask questions and even post results online. The lessons expect you to go off and find out how to do things. I will give you hints, but part of the point of each lesson is to force you to learn where to go to look for help on and about the statistics and the NumPy API and the best-of-breed tools in Python (hint, I have all of the answers directly on this blog, use the search box). Post your results online, I'll cheer you on!

Hang in there, don't give up!

Lesson 01: Statistics and Machine Learning

In this lesson, you will discover the 5 reasons why a machine learning practitioner should deepen their understanding of statistics.

Statistics in Data Preparation

Statistical methods are required in the preparation of train and test data for your machine learning model. This includes techniques for:

- Outlier detection.
- Missing value imputation.
- Data sampling.
- Data scaling.
- Variable encoding.
- And much more.

A basic understanding of data distributions, descriptive statistics, and data visualization is required to help you identify the methods to choose when performing these tasks.

Statistics in Model Evaluation

Statistical methods are required when evaluating the skill of a machine learning model on data not seen during training. This includes techniques for:

- Data sampling.
- Data resampling.
- Experimental design.

Resampling techniques such as k-fold cross-validation are often well understood by machine learning practitioners, but the rationale for why this method is required is not.

Statistics in Model Selection

Statistical methods are required when selecting a final model or model configuration to use for a predictive modeling problem. These include techniques for:

- Checking for a significant difference between results.
- Quantifying the size of the difference between results.

This might include the use of statistical hypothesis tests.

Statistics in Model Presentation

Statistical methods are required when presenting the skill of a final model to stakeholders. This includes techniques for:

- Summarizing the expected skill of the model on average.
- Quantifying the expected variability of the skill of the model in practice.

This might include estimation statistics such as confidence intervals.

Statistics in Prediction

Statistical methods are required when making a prediction with a finalized model on new data. This includes techniques for:

- Quantifying the expected variability for the prediction.

This might include estimation statistics such as prediction intervals.

Your Task

For this lesson, you must list three reasons why you personally want to learn statistics.

Next

In the next lesson, you will discover a concise definition of statistics.

Lesson 02: Introduction to Statistics

In this lesson, you will discover a concise definition of statistics. Statistics is a required prerequisite for most books and courses on applied machine learning. But what exactly is statistics? Statistics is a subfield of mathematics. It refers to a collection of methods for working with data and using data to answer questions.

It is because the field is comprised of a grab bag of methods for working with data that it can seem large and amorphous to beginners. It can be hard to see the line between methods that belong to statistics and methods that belong to other fields of study. When it comes to the statistical tools that we use in practice, it can be helpful to divide the field of statistics into two large groups of methods: descriptive statistics for summarizing data, and inferential statistics for drawing conclusions from samples of data.

- **Descriptive Statistics**: Descriptive statistics refer to methods for summarizing raw observations into information that we can understand and share.
- **Inferential Statistics**: Inferential statistics is a fancy name for methods that aid in quantifying properties of the domain or population from a smaller set of obtained observations called a sample.

Your Task

For this lesson, you must list three methods that can be used for each descriptive and inferential statistics.

Next

In the next lesson, you will discover the Gaussian distribution and how to calculate summary statistics.

Lesson 03: Gaussian Distribution and Descriptive Stats

In this lesson, you will discover the Gaussian distribution for data and how to calculate simple descriptive statistics. A sample of data is a snapshot from a broader population of all possible observations that could be taken from a domain or generated by a process. Interestingly, many observations fit a common pattern or distribution called the normal distribution, or more formally, the Gaussian distribution. It is the bell-shaped distribution that you may be familiar with.

A lot is known about the Gaussian distribution, and as such, there are whole sub-fields of statistics and statistical methods that can be used with Gaussian data. Any Gaussian distribution, and in turn any data sample drawn from a Gaussian distribution, can be summarized with just two parameters:

- **Mean**. The central tendency or most likely value in the distribution (the top of the bell).
- **Variance**. The average difference that observations have from the mean value in the distribution (the spread).

The units of the mean are the same as the units of the distribution, although the units of the variance are squared, and therefore harder to interpret. A popular alternative to the variance parameter is the standard deviation, which is simply the square root of the variance, returning the units to be the same as those of the distribution. The mean, variance, and standard deviation can be calculated directly on data samples in NumPy. The example below generates a sample of 100 random numbers drawn from a Gaussian distribution with a known mean of 50 and a standard deviation of 5 and calculates the summary statistics.

```
# calculate summary stats
from numpy.random import seed
from numpy.random import randn
from numpy import mean
from numpy import var
from numpy import std
# seed the random number generator
seed(1)
# generate univariate observations
data = 5 * randn(10000) + 50
# calculate statistics
print('Mean: %.3f' % mean(data))
print('Variance: %.3f' % var(data))
print('Standard Deviation: %.3f' % std(data))
```

Listing 1: Example of calculating summary statistics.

Run the example and compare the estimated mean and standard deviation from the expected values.

Your Task

For this lesson, you must implement the calculation of one descriptive statistic from scratch in Python, such as the calculation of a sample mean.

Next

In the next lesson, you will discover how to quantify the relationship between two variables.

Lesson 04: Correlation Between Variables

In this lesson, you will discover how to calculate a correlation coefficient to quantify the relationship between two variables. Variables in a dataset may be related for lots of reasons. It can be useful in data analysis and modeling to better understand the relationships between variables. The statistical relationship between two variables is referred to as their correlation. A correlation could be positive, meaning both variables move in the same direction, or negative, meaning that when one variable's value increases, the other variables' values decrease.

- **Positive Correlation**: Both variables change in the same direction.
- **Neutral Correlation**: No relationship in the change of the variables.
- **Negative Correlation**: Variables change in opposite directions.

The performance of some algorithms can deteriorate if two or more variables are tightly related, called multicollinearity. An example is linear regression, where one of the offending correlated variables should be removed in order to improve the skill of the model. We can quantify the relationship between samples of two variables using a statistical method called Pearson's correlation coefficient, named for the developer of the method, Karl Pearson. The `pearsonr()` NumPy function can be used to calculate the Pearson's correlation coefficient for samples of two variables. The complete example is listed below showing the calculation where one variable is dependent upon the second.

```python
# calculate correlation coefficient
from numpy.random import seed
from numpy.random import randn
from scipy.stats import pearsonr
# seed random number generator
seed(1)
# prepare data
data1 = 20 * randn(1000) + 100
data2 = data1 + (10 * randn(1000) + 50)
# calculate Pearson's correlation
corr, p = pearsonr(data1, data2)
# display the correlation
print('Pearsons correlation: %.3f' % corr)
```

Listing 2: Example of calculating a correlation coefficient.

Run the example and review the calculated correlation coefficient.

Your Task

For this lesson, you must load a standard machine learning dataset and calculate the correlation between each pair of numerical variables.

Next

In the next lesson, you will discover statistical hypothesis tests.

Lesson 05: Statistical Hypothesis Tests

In this lesson, you will discover statistical hypothesis tests and how to compare two samples. Data must be interpreted in order to add meaning. We can interpret data by assuming a specific structure our outcome and use statistical methods to confirm or reject the assumption. The assumption is called a hypothesis and the statistical tests used for this purpose are called statistical hypothesis tests. The assumption of a statistical test is called the null hypothesis, or hypothesis zero (H0 for short). It is often called the default assumption, or the assumption that nothing has changed. A violation of the test's assumption is often called the first hypothesis, hypothesis one, or H1 for short.

- **Hypothesis 0 (H0)**: Assumption of the test holds and is failed to be rejected.
- **Hypothesis 1 (H1)**: Assumption of the test does not hold and is rejected at some level of significance.

We can interpret the result of a statistical hypothesis test using a p-value. The p-value is the probability of observing the data, given the null hypothesis is true. A large probability means that the H0 or default assumption is likely. A small value, such as below 5% (0.05) suggests that it is not likely and that we can reject H0 in favor of H1, or that something is likely to be different (e.g. a significant result).

A widely used statistical hypothesis test is the Student's t-test for comparing the mean values from two independent samples. The default assumption is that there is no difference between the samples, whereas a rejection of this assumption suggests some significant difference. The tests assumes that both samples were drawn from a Gaussian distribution and have the same variance. The Student's t-test can be implemented in Python via the `ttest_ind()` SciPy function. Below is an example of calculating and interpreting the Student's t-test for two data samples that are known to be different.

```
# student's t-test
from numpy.random import seed
from numpy.random import randn
from scipy.stats import ttest_ind
# seed the random number generator
seed(1)
# generate two independent samples
data1 = 5 * randn(100) + 50
data2 = 5 * randn(100) + 51
# compare samples
stat, p = ttest_ind(data1, data2)
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
```

```
if p > alpha:
  print('Same distributions (fail to reject H0)')
else:
  print('Different distributions (reject H0)')
```

Listing 3: Example of calculating the Student's t-test.

Run the code and review the calculated statistic and interpretation of the p-value.

Your Task

For this lesson, you must list three other statistical hypothesis tests that can be used to check for differences between samples.

Next

In the next lesson, you will discover estimation statistics as an alternative to statistical hypothesis testing.

Lesson 06: Estimation Statistics

In this lesson, you will discover estimation statistics that may be used as an alternative to statistical hypothesis tests. Statistical hypothesis tests can be used to indicate whether the difference between two samples is due to random chance, but cannot comment on the size of the difference. A group of methods referred to as *new statistics* are seeing increased use instead of or in addition to p-values in order to quantify the magnitude of effects and the amount of uncertainty for estimated values. This group of statistical methods is referred to as estimation statistics. Estimation statistics is a term to describe three main classes of methods. The three main classes of methods include:

- **Effect Size**. Methods for quantifying the size of an effect given a treatment or intervention.
- **Interval Estimation**. Methods for quantifying the amount of uncertainty in a value.
- **Meta-Analysis**. Methods for quantifying the findings across multiple similar studies.

Of the three, perhaps the most useful methods in applied machine learning are interval estimation. There are three main types of intervals. They are:

- **Tolerance Interval**: The bounds or coverage of a proportion of a distribution with a specific level of confidence.
- **Confidence Interval**: The bounds on the estimate of a population parameter.
- **Prediction Interval**: The bounds on a single observation.

A simple way to calculate a confidence interval for a classification algorithm is to calculate the binomial proportion confidence interval, which can provide an interval around a model's estimated accuracy or error. This can be implemented in Python using the `confint()` Statsmodels function. The function takes the count of successes (or failures), the total number of trials, and the significance level as arguments and returns the lower and upper bound of the confidence interval. The example below demonstrates this function in a hypothetical case where a model made 88 correct predictions out of a dataset with 100 instances and we are interested in the 95% confidence interval (provided to the function as a significance of 0.05).

```python
# calculate the confidence interval
from statsmodels.stats.proportion import proportion_confint
# calculate the interval
lower, upper = proportion_confint(88, 100, 0.05)
print('lower=%.3f, upper=%.3f' % (lower, upper))
```

Listing 4: Example of calculating a confidence interval.

Run the example and review the confidence interval on the estimated accuracy.

Your Task

For this lesson, you must list two methods for calculating the effect size in applied machine learning and when they might be useful. As a hint, consider one for the relationship between variables and one for the difference between samples.

Next

In the next lesson, you will discover nonparametric statistical methods.

Lesson 07: Nonparametric Statistics

In this lesson, you will discover statistical methods that may be used when your data does not come from a Gaussian distribution. A large portion of the field of statistics and statistical methods is dedicated to data where the distribution is known. Data in which the distribution is unknown or cannot be easily identified is called nonparametric. In the case where you are working with nonparametric data, specialized nonparametric statistical methods can be used that discard all information about the distribution. As such, these methods are often referred to as distribution-free methods.

Before a nonparametric statistical method can be applied, the data must be converted into a rank format. As such, statistical methods that expect data in rank format are sometimes called rank statistics, such as rank correlation and rank statistical hypothesis tests. Ranking data is exactly as its name suggests. The procedure is as follows:

1. Sort all data in the sample in ascending order.
2. Assign an integer rank from 1 to N for each unique value in the data sample.

A widely used nonparametric statistical hypothesis test for checking for a difference between two independent samples is the Mann-Whitney U test, named for Henry Mann and Donald Whitney. It is the nonparametric equivalent of the Student's t-test but does not assume that the data is drawn from a Gaussian distribution. The test can be implemented in Python via the `mannwhitneyu()` SciPy function. The example below demonstrates the test on two data samples drawn from a uniform distribution known to be different.

```
# example of the mann-whitney u test
from numpy.random import seed
from numpy.random import rand
from scipy.stats import mannwhitneyu
# seed the random number generator
seed(1)
# generate two independent samples
data1 = 50 + (rand(100) * 10)
data2 = 51 + (rand(100) * 10)
# compare samples
stat, p = mannwhitneyu(data1, data2)
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
  print('Same distribution (fail to reject H0)')
else:
  print('Different distribution (reject H0)')
```

Listing 5: Example of calculating the Mann-Whitney U test.

Run the example and review the calculated statistics and interpretation of the p-value.

Your Task

For this lesson, you must list three additional nonparametric statistical methods. This was the final lesson in the mini-course.

Final Word Before You Go...

You made it. Well done! Take a moment and look back at how far you have come. You discovered:

- The importance of statistics in applied machine learning.
- A concise definition of statistics and a division of methods into two main types.
- The Gaussian distribution and how to describe data with this distribution using statistics.
- How to quantify the relationship between the samples of two variables.
- How to check for the difference between two samples using statistical hypothesis tests.
- An alternative to statistical hypothesis tests called estimation statistics.
- Nonparametric methods that can be used when data is not drawn from the Gaussian distribution.

This is just the beginning of your journey with statistics for machine learning. Keep practicing and developing your skills. Take the next step and check out my book on *Statistical Methods for Machine Learning*.

How Did You Go With The Crash-Course?

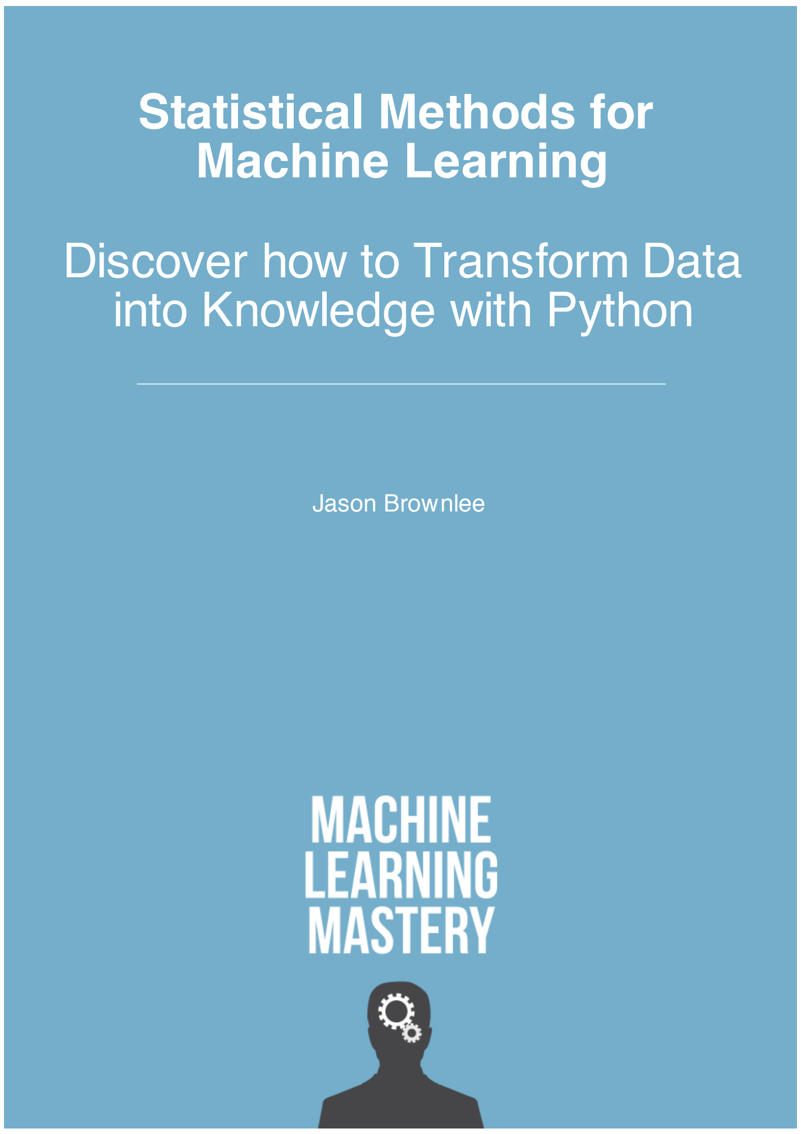
Did you enjoy this crash-course?
Do you have any questions or sticking points?

Let me know, send me an email at: **jason@MachineLearningMastery.com**

Take the Next Step

Looking for more help with Statistics for Machine Learning?

Grab my new book:
Statistical Methods for Machine Learning
https://machinelearningmastery.com/statistics_for_machine_learning/


Statistical Methods for Machine Learning
Discover how to Transform Data into Knowledge with Python
Jason Brownlee
MACHINE LEARNING MASTERY