Exercise - Train a simple linear regression model

8 minutes

This module requires a sandbox to complete. You have used 1 of 10 sandboxes for today. More sandboxes will be available tomorrow.

Activate sandbox

Runtime File Edit View

Run all Kernel Compute not connected

Exercise: Train a simple linear regression model

In this exercise, we'll train a simple linear regression model to predict body temperature based on dogs' ages and interpret the result.

Loading data

Let's begin by having a look at our data.

```
import pandas
!pip install statsmodels
!wget https://raw.githubusercontent.com/MicrosoftDocs/mslearn-introduction-to-machine-learning
!wget https://raw.githubusercontent.com/MicrosoftDocs/mslearn-introduction-to-machine-learning
```

```
!wget https://raw.githubusercontent.com/MicrosoftDocs/mslearn-introduction-to-machine-learning/

# Convert it into a table using pandas
dataset = pandas.read_csv("doggy-illness.csv", delimiter="\t")

# Print the data
print(dataset)
```

We have a variety of information, including what the dogs did the night before, their age, whether they're overweight, and their clinical signs. In this exercise, our y values, or labels, are represented by the `core_temperature` column, while our feature will be the `age` in years.

Data visualization

Let's have a look at how the features and labels are distributed.

```
import graphing

graphing.histogram(dataset, label_x='age', nbins=10, title="Feature", show=True)
graphing.histogram(dataset, label_x='core_temperature', nbins=10, title="Label")
```

Looking at our feature (`age`), we can see dogs were at or less than 9 years of age, and ages are evenly distributed. In other words, no particular age is substantially more common than any other.

Looking at our label (`core_temperature`), most dogs seem to have a slightly elevated core temperature (we would normally expect ~37.5 degrees celcius), which indicates they're unwell. A small number of dogs have a temperature above 40 degrees, which indicates

they're quite unwell.

Simply because the shape of these distributions is different, we can guess that the feature won't be able to predict the label extremely well. For example, if old age perfectly predicted who would have a high temperature, then the number of old dogs would exactly match the number of dogs with a high temperature.

The model might still end up being useful, though, so lets continue.

```
graphing.scatter_2D(dataset, label_x="age", label_y="core_temperature", title='core temperatur
```


It does seem that older dogs tended to have higher temperatures than younger dogs. The relationship is quite "noisy," though; many dogs of the same age have quite different temperatures.

Simple linear regression

Let's formally examine the relationship between our labels and features by fitting a line (simple linear-regression model) to the dataset.

```
import statsmodels.formula.api as smf
import graphing # custom graphing code. See our GitHub repo for details

# First, we define our formula using a special syntax
# This says that core temperature is explained by age
formula = "core_temperature ~ age"

# Perform linear regression. This method takes care of
# the entire fitting procedure for us.
model = smf.ols(formula = formula, data = dataset).fit()
```

```
# Show a graph of the result
graphing.scatter_2D(dataset,    label_x="age",
                                label_y="core_temperature",
                                trendline=lambda x: model.params[1] * x + model.params[0]
                                )
```

The line seems to fit the data quite well, validating our hypothesis that there's a positive correlation between a dog's age and their core temperature.

Interpreting our model

Visually, simple linear regression is easy to understand. Let's recap on what the parameters mean, though.

```
print("Intercept:", model.params[0], "Slope:", model.params[1])
```

Remember that simple linear regression models are explained by the line intercept and the line slope.

Here, our intercept is 38 degrees celsius. This means that when `age` is `0`, the model will predict 38 degrees.

Our slope is 0.15 degrees celsius, meaning that for every year of age, the model will predict temperatures 0.15 degrees higher.

In the following box, try to change the age to a few different values to see different predictions, and compare these with the line in the preceding graph.

```
def estimate_temperature(age):
    # Model param[0] is the intercepts and param[1] is the slope
    return age * model.params[1] + model.params[0]
```

```
print("Estimate temperature from age")
print(estimate_temperature(age=0))
```

Summary

We covered the following concepts in this exercise:

- Quickly visualizing a dataset
- Qualitatively assessing a linear relationship
- Building a simple linear-regression model
- Understanding parameters of a simple linear-regression model

Next unit: Multiple linear regression and R-squared

Continue >