DS

Effect of discretization, continuation, Normalization, Randomization on the data

September 22, 2021

Using Orange I learned about discretization, continuation, Normalization, Randomization

Discretization

Discretization replaces continuous features with the corresponding categorical features:

```
import Orange
iris = Orange.data.Table("titanic.tab")
disc = Orange.preprocess.Discretize()
disc.method = Orange.preprocess.discretize.EqualFreq(n=4)
d_iris = disc(iris)
print("Original dataset:")
for e in iris[:4]:
print(e)
print("Discretized dataset:")
for e in d_iris[:4]:
print(e)
```

```
>>>
Running script:
Original dataset:
[first, adult, male | yes]
[first, adult, male | yes]
[first, adult, male | yes]
[first, adult, male | yes]
Discretized dataset:
[first, adult, male | yes]
[first, adult, male | yes]
[first, adult, male | yes]
```

```
[first, adult, male | yes]
```

Continuation

Given a data table, return a new table in which the discretize attributes are replaced with continuous or removed.

- binary variables are transformed into 0.0/1.0 or -1.0/1.0 indicator variables, depending upon the argument `zero_based`.
- multinomial variables are treated according to the argument `multinomial_treatment`.
- discrete attributes with only one possible value are removed;

```
import Orange
titanic = Orange.data.Table("titanic.tab")
continuizer = Orange.preprocess.Continuize()
titanic1 = continuizer(titanic)
titanic.domain
titanic1.domain
print(titanic[15])
print(titanic1[15])
continuizer.multinomial_treatment = continuizer.FirstAsBase
continuizer(titanic).domain
continuizer.multinomial_treatment = continuizer.AsOrdinal
titanic1 = continuizer(titanic)
print(titanic[700])
print(titanic1[700])
continuizer.multinomial_treatment =
continuizer.AsNormalizedOrdinal
titanic1 = continuizer(titanic)
print(titanic1[700])
print(titanic1[15])
```

```
>>>
Running script:
[first, adult, male | yes]
```

```
[0, 1, 0, 0, 1, 0, 0, 1 | yes]
[third, adult, male | no]
[3, 0, 1 | no]
[1, 0, 1 | no]
[0.333333, 0, 1 | yes]
>>>
```

Normalization

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is **to change the values of numeric columns in the dataset to a common scale**, without distorting differences in the ranges of values. For machine learning, every dataset does not require normalization.

```
from Orange.data import Table
from Orange.preprocess import Normalize
data = Table("iris")
normalizer = Normalize(norm_type=Normalize.NormalizeBySpan)
normalized_data = normalizer(data)
print(normalized_data)
```

```
[0.3889, 0.3750, 0.5424, 0.50 | Iris-versicolor],
[0.5278, 0.3750, 0.5593, 0.50 | Iris-versicolor],
[0.2222, 0.2083, 0.3390, 0.4167 | Iris-versicolor],
[0.3889, 0.3333, 0.5254, 0.50 | Iris-versicolor],
[0.5556, 0.5417, 0.8475, 1.00 | Iris-virginica],
[0.4167, 0.2917, 0.6949, 0.75 | Iris-virginica],
[0.7778, 0.4167, 0.8305, 0.8333 | Iris-virginica],
[0.5556, 0.3750, 0.7797, 0.7083 | Iris-virginica],
[0.6111, 0.4167, 0.8136, 0.8750 | Iris-virginica],
[0.9167, 0.4167, 0.9492, 0.8333 | Iris-virginica],
[0.1667, 0.2083, 0.5932, 0.6667 | Iris-virginica],
[0.8333, 0.3750, 0.8983, 0.7083 | Iris-virginica],
[0.6667, 0.2083, 0.8136, 0.7083 | Iris-virginica],
[0.8056, 0.6667, 0.8644, 1.00 | Iris-virginica],
[0.6111, 0.50, 0.6949, 0.7917 | Iris-virginica],
[0.5833, 0.2917, 0.7288, 0.75 | Iris-virginica],
[0.6944, 0.4167, 0.7627, 0.8333 | Iris-virginica],
```

Randomization

Randomization in an experiment is **where you choose your experimental participants randomly**. For example, you might use simple random sampling, where participants' names are drawn randomly from a pool where everyone has an even probability of being chosen.

```
from Orange.data import Table
from Orange.preprocess import Randomize
data = Table("iris")
randomizer = Randomize(Randomize.RandomizeClasses)
randomized_data = randomizer(data)
print(randomized_data)
```

```
[6.1, 2.6, 5.6, 1.4 | Iris-versicolor],
[7.7, 3.0, 6.1, 2.3 | Iris-virginica],
[6.3, 3.4, 5.6, 2.4 | Iris-setosa],
[6.4, 3.1, 5.5, 1.8 | Iris-versicolor],
[6.0, 3.0, 4.8, 1.8 | Iris-setosa],
[6.9, 3.1, 5.4, 2.1 | Iris-virginica],
[6.7, 3.1, 5.6, 2.4 | Iris-setosa],
[6.9, 3.1, 5.1, 2.3 | Iris-virginica],
[5.8, 2.7, 5.1, 1.9 | Iris-setosa],
[6.8, 3.2, 5.9, 2.3 | Iris-virginica],
[6.7, 3.3, 5.7, 2.5 | Iris-versicolor],
[6.7, 3.0, 5.2, 2.3 | Iris-virginica],
[6.3, 2.5, 5.0, 1.9 | Iris-setosa],
[6.5, 3.0, 5.2, 2.0 | Iris-setosa],
[6.2, 3.4, 5.4, 2.3 | Iris-versicolor],
[5.9, 3.0, 5.1, 1.8 | Iris-setosa]]
```

Thank You.

Enter your comment...

Popular posts from this blog

Web Scraping Using Python

July 26, 2021

What is Web Scraping? Web scraping is a method to extract data from websites. Using web scarping we can store unstructured website data to structured website data. There are different ways of scrape we ...

READ MORE

Data Pre-processing using Scikit-learn

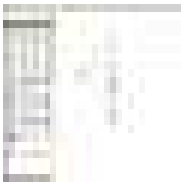
August 11, 2021

Data pre-processing is one technique of data mining using that you can convert your raw data into an understandable format. In his practical, we will take one dataset and performing the following task Standardiz ...

READ MORE

Getting Started with Orange Tool

September 06, 2021



This blog is about Orange tool,which is a tool used in the data science.In this blog I will discuss about basic of Orange Tool and its various features.So lets get started What is Orange Tool? Orange is an open-sourc ...

READ MORE

Powered by Blogger

Theme images by Michael Elkan

HARNISH CHAVDA

VISIT PROFILE

Archive

Report Abuse