Machine Learning - Percentiles

ML Study

What are Percentiles?

Percentiles are used in statistics to give you a number that describes the value that a given percent of the values are lower than.

Example: Let's say we have an array of the ages of all the people that live in a street.

ages = [5,31,43,48,50,41,7,11,15,39,80,82,32,2,8,6,25,36,27,61,31]

What is the 75. percentile? The answer is 43, meaning that 75% of the people are 43 or younger.

The NumPy module has a method for finding the specified percentile:

Use the NumPy percentile() method to find the percentiles:

```
import numpy

ages =[5,31,43,48,50,41,7,11,15,39,80,82,32,2,8,6,25,36,27,61,31]

x = numpy.percentile(ages, 75)

print(x)
```

What is the age that 90% of the people are younger than?

```
import numpy

ages = [5,31,43,48,50,41,7,11,15,39,80,82,32,2,8,6,25,36,27,61,31]

x = numpy.percentile(ages, 90)

print(x)
```

Machine Learning - Data Distribution

ML Study

Data Distribution

In the real world, the data sets are much bigger, but it can be difficult to gather real world data, at least at an early stage of a project.

How Can we Get Big Data Sets?

To create big data sets for testing, we use the Python module NumPy, which comes with a number of methods to create random data sets, of any size.

Create an array containing 250 random floats between 0 and 5:

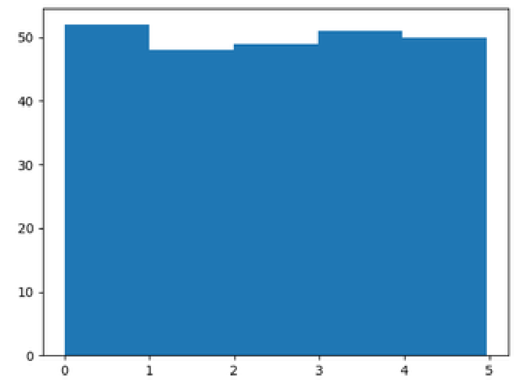
```
import numpy

x = numpy.random.uniform(0.0, 5.0, 250)

print(x)
```

Histogram

To visualize the data set we can draw a histogram with the data we collected.
We will use the Python module Matplotlib to draw a histogram.

Draw a histogram:

```
import numpy
import matplotlib.pyplot as plt
x = numpy.random.uniform(0.0, 5.0, 250)
plt.hist(x, 5)
plt.show()
```



ML Study

Machine Learning - Data Distribution

Histogram Explained

We use the array from the example above to draw a histogram with 5 bars.
The first bar represents how many values in the array are between 0 and 1.
The second bar represents how many values are between 1 and 2.

Which gives us this result:

- 52 values are between 0 and 1
- 48 values are between 1 and 2
- 49 values are between 2 and 3
- 51 values are between 3 and 4
- 50 values are between 4 and 5

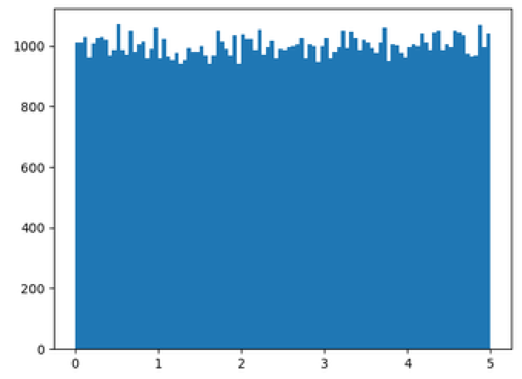
Q. Will the array values would be same in every computer?

Big Data Distributions

An array containing 250 values is not considered very big, but now you know how to create a random set of values, and by changing the parameters, you can create the data set as big as you want.

Create an array with 100000 random numbers, and display them using a histogram with 100 bars:

```
import numpy
import matplotlib.pyplot as plt
x = numpy.random.uniform(0.0, 5.0, 100000)
plt.hist(x, 100)
plt.show()
```



Machine Learning - Normal Data Distribution

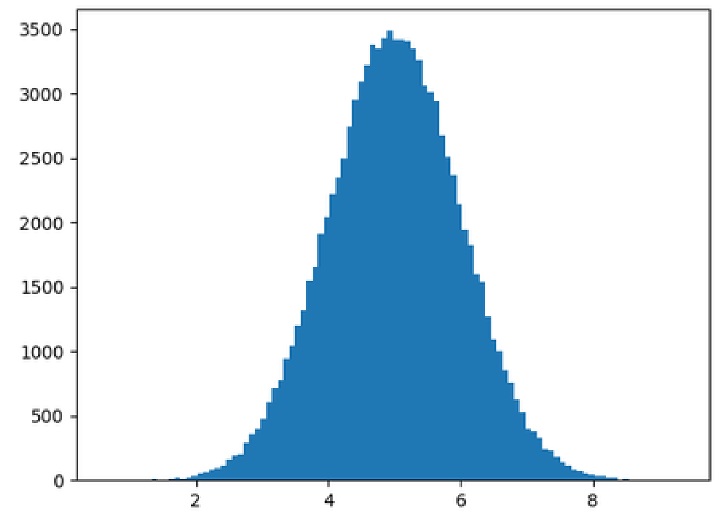
ML Study

Normal Data Distribution

How to create an array where the values are concentrated around a given value.
In probability theory this kind of data distribution is known as the normal data distribution, or the Gaussian data distribution, after the mathematician Carl Friedrich Gauss who came up with the formula of this data distribution.

A typical normal data distribution:

```
import numpy
import matplotlib.pyplot as plt
x = numpy.random.normal(5.0, 1.0, 100000)
plt.hist(x, 100)
plt.show()
```



Histogram Explained

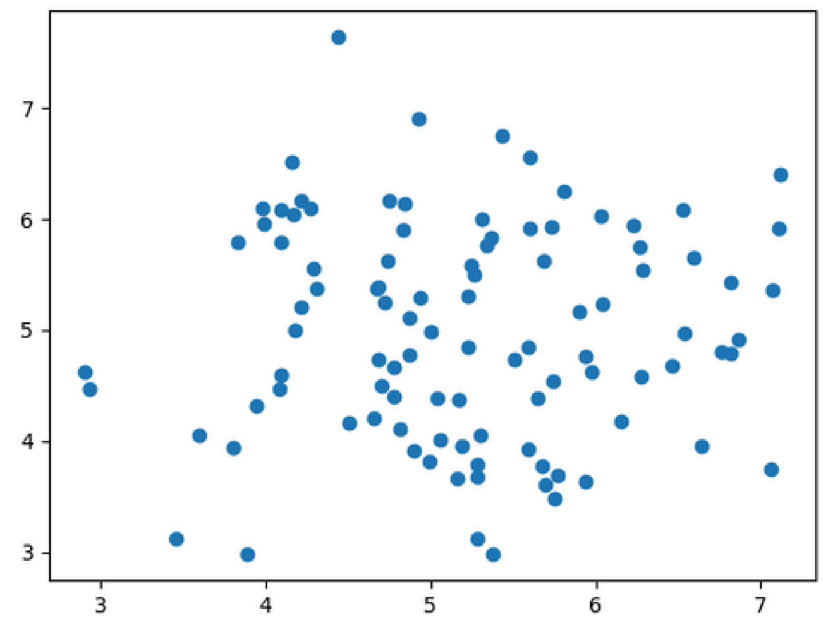
We use the array from the numpy.random.normal() method, with 100000 values, to draw a histogram with 100 bars.
We specify that the mean value is 5.0, and the standard deviation is 1.0.
Meaning that the values should be concentrated around 5.0, and rarely further away than 1.0 from the mean.
And as you can see from the histogram, most values are between 4.0 and 6.0, with a top at approximately 5.0.

ML Study

Machine Learning - Scatter Plot

Scatter Plot

A scatter plot is a diagram where each value in the data set is represented by a dot.



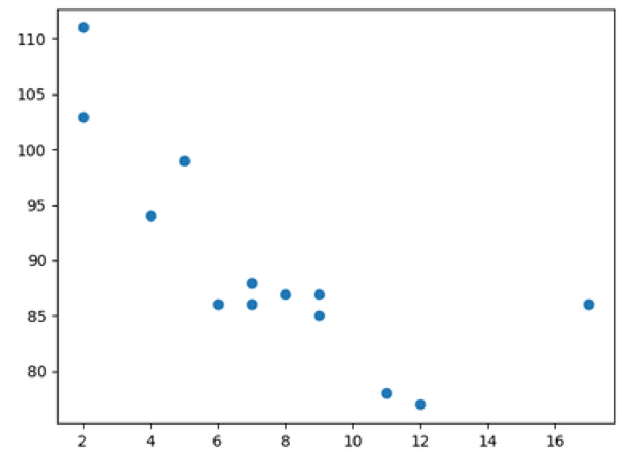
The Matplotlib module has a method for drawing scatter plots, it needs two arrays of the same length, one for the values of the x-axis, and one for the values of the y-axis:

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

The x array represents the age of each car.
The y array represents the speed of each car.

Use the scatter() method to draw a scatter plot diagram:

```
import matplotlib.pyplot as plt
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]
plt.scatter(x, y)
plt.show()
```



ML Study

Machine Learning - Scatter Plot

Scatter Plot Explained

The x-axis represents ages, and the y-axis represents speeds. What we can read from the diagram is that the two fastest cars were both 2 years old, and the slowest car was 12 years old.

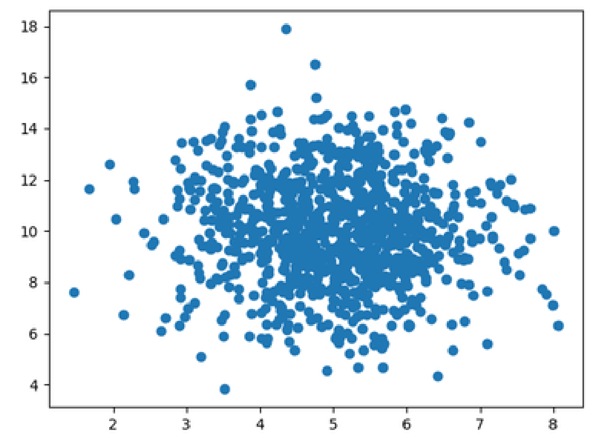
Random Data Distributions

In Machine Learning the data sets can contain thousands-, or even millions, of values.
You might not have real world data when you are testing an algorithm, you might have to use randomly generated values.

Let us create two arrays that are both filled with 1000 random numbers from a normal data distribution.
The first array will have the mean set to 5.0 with a standard deviation of 1.0.
The second array will have the mean set to 10.0 with a standard deviation of 2.0:

A scatter plot with 1000 dots:

```
import numpy
import matplotlib.pyplot as plt
x = numpy.random.normal(5.0, 1.0, 1000)
y = numpy.random.normal(10.0, 2.0, 1000)
plt.scatter(x, y)
plt.show()
```



We can see that the dots are concentrated around the value 5 on the x-axis, and 10 on the y-axis.
We can also see that the spread is wider on the y-axis than on the x-axis.

Machine Learning - Hypothesis Testing

ML Study JAM

Hypothesis Testing:

Hypothesis testing is a statistical method used to make inferences about a population based on sample data. It involves formulating two competing hypotheses, the null hypothesis (H0) and the alternative hypothesis (H1), and using statistical tests to determine which hypothesis is supported by the data.

1. Null Hypothesis (H0):
    - Represents the default assumption or the status quo.
    - Typically states that there is no significant difference or relationship between variables.
2. Alternative Hypothesis (H1):
    - Represents the opposite of the null hypothesis.
    - States that there is a significant difference or relationship between variables.

In machine learning, hypothesis testing is often used in tasks such as comparing models, evaluating the significance of features, or testing the effectiveness of a new algorithm.

```
import numpy as np
from scipy.stats import ttest_ind
# Generate two sets of sample data (e.g., test scores of two
groups)
group1_scores = np.random.normal(80, 10, 100)  # Mean=80,
Std=10, Sample size=100
group2_scores = np.random.normal(85, 12, 100)  # Mean=85,
Std=12, Sample size=100
# Perform independent t-test to compare means of two groups
t_stat, p_value = ttest_ind(group1_scores, group2_scores)
# Print the t-statistic and p-value
print(f"T-statistic: {t_stat}")
print(f"P-value: {p_value}")
# Interpret the results based on the p-value
if p_value < 0.05:
   print("Reject null hypothesis: There is a significant difference
between the groups.")
else:
   print("Fail to reject null hypothesis: There is no significant
difference between the groups.")
```

Machine Learning - Probability Distribution

ML Study

Probability Distribution:

Probability distributions describe the likelihood of different outcomes in a random experiment. In machine learning, probability distributions are fundamental for modeling uncertainty, generating random samples, and defining probabilistic models.

1. Normal Distribution (Gaussian Distribution):
   - Represents a symmetrical bell-shaped curve.
   - Defined by its mean and standard deviation .
   - Widely used in statistical analysis and modeling.
2. Binomial Distribution:
   - Represents the number of successes in a fixed number of independent trials.
   - Defined by parameters (number of trials) and (probability of success).
3. Poisson Distribution:
   - Represents the number of events occurring in a fixed interval of time or space.
   - Defined by the rate parameter λ (average number of events per interval).

```
import matplotlib.pyplot as plt
from scipy.stats import norm
# Define parameters for the normal distribution
mu = 0  # Mean
sigma = 1  # Standard deviation
# Generate random data from a normal distribution
data = np.random.normal(mu, sigma, 1000)  # Generate 1000 samples
# Plot the histogram of the generated data
plt.hist(data, bins=30, density=True, alpha=0.6, color='blue')
# Plot the probability density function (PDF) of the normal distribution
x = np.linspace(mu - 3*sigma, mu + 3*sigma, 100)
plt.plot(x, norm.pdf(x, mu, sigma), 'r-', linewidth=2)
# Add labels and title to the plot
plt.xlabel('Value')
plt.ylabel('Probability Density')
plt.title('Normal Distribution')
# Show the plot
plt.show()
```

TensorFlow
User Group Durg

ML Study Jam

Thank You!