Machine Learning

Machina learning гэдэг нь дата анализ хийж үр дүнг урьдчилан таамагладаг програм юм.

Агуулга

1. Mean median and mode
2. Standard Deviation
3. Percentile
4. Data Distribution
5. Normal data distribution
6. Scatter plot
7. Linear Regression
8. Polynormal regression
9. Multiple regression
10. Scall
11. Train/test
12. Decision tree
13. Confusion matrix
14. Hierarchical clustering
15. Logistic regression
16. Grid search
17. Categorical data
18. K-means
19. Bootstrap aggregation
20. Cross validation
21. Auc-roc curve
22. K-nearest neighbors (knn)

Mean median and mode

example

```
from scipy import stats
import numpy

arr = [77, 78, 85, 86, 86, 86, 87, 87, 88, 94, 99, 103, 111]
print(numpy.mean(arr))    # output: 89 is average
print(numpy.median(arr))  # output: 87 is middle
print(stats.mode(arr))    # output: 86 is common
```

Standard deviation

Formula

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

σ = population standard deviation
N = the size of the population
x_i = each value from the population
μ = the population mean

```
import numpy

speed = [86, 87, 88, 86, 87, 85, 86]

x = numpy.std(speed)

print(x)  # output: 0.9
```

Percentiles

75% нь 43 наснаас залуу байна.

```
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)  # out: 43
```

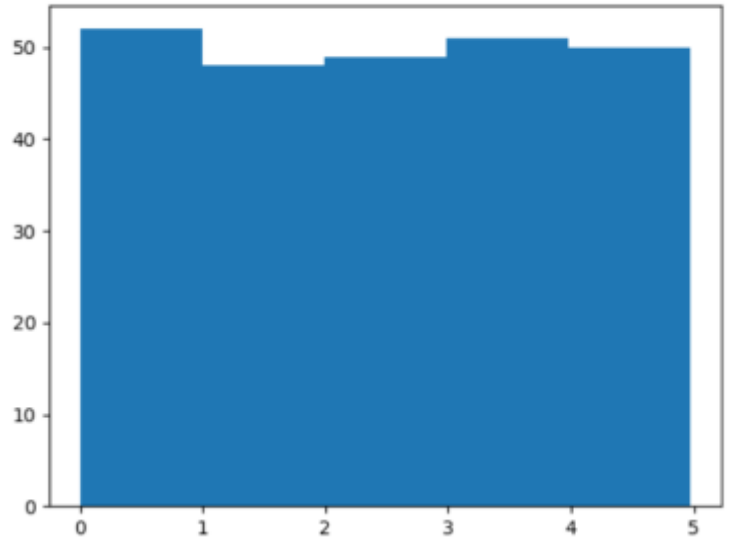
Data Distribution (histogram)

Example

```
import numpy
import matplotlib.pyplot as plt

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

plt.hist(x, 5)
plt.show()
```

```
# 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
```

result:

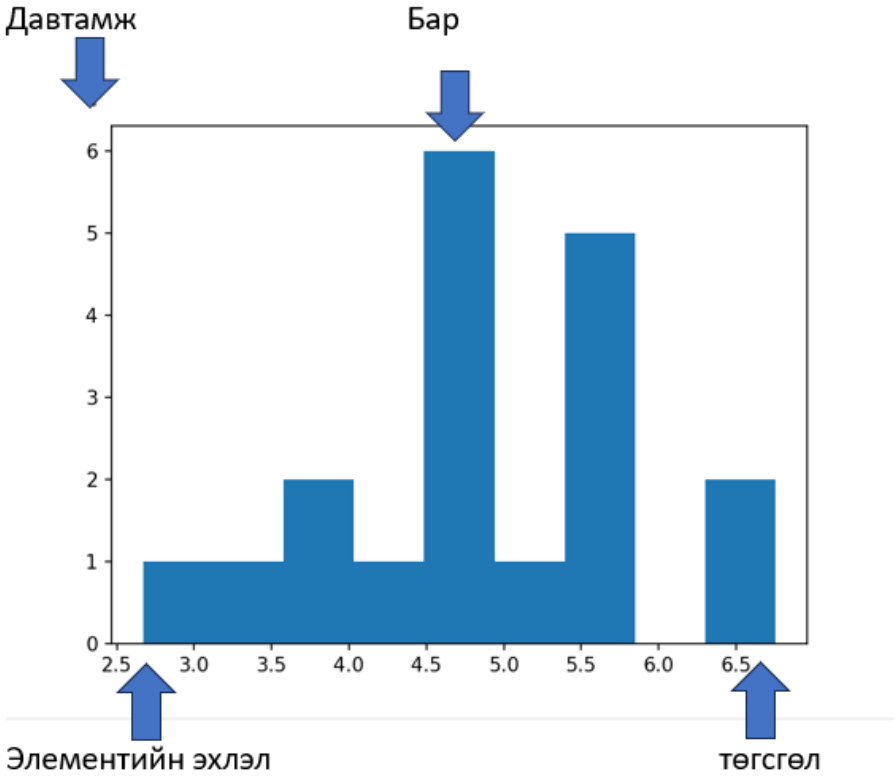


Normal Data Distribution

Example

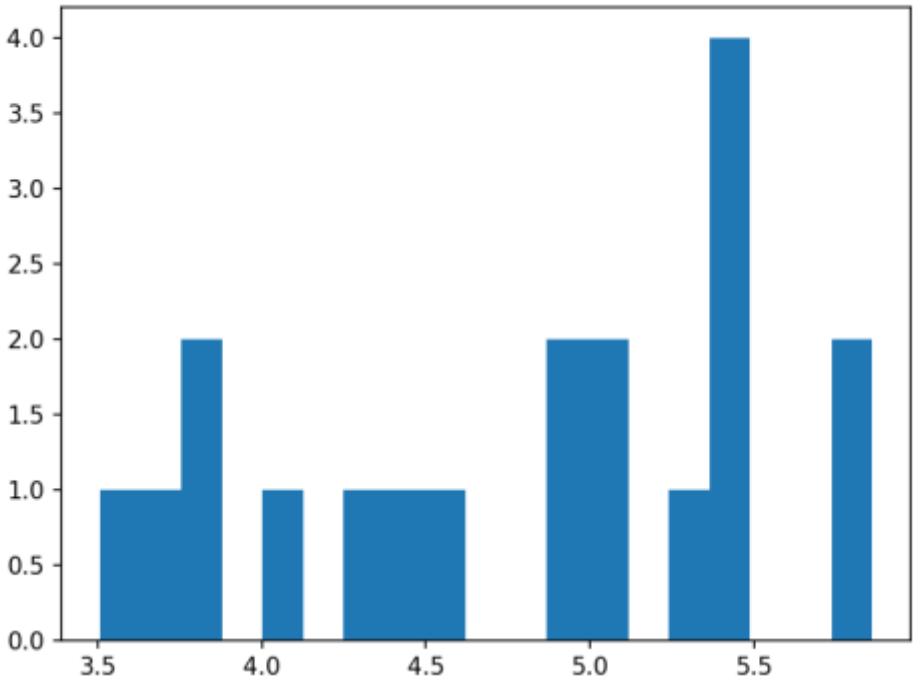
```python
import numpy
import matplotlib.pyplot as plt

# 4<5<6 хүртэл 19 тоо үүсгэхйг заасан бна
x = numpy.random.normal(5.0, 1.0, 19)

print(sorted(x))
plt.hist(x, 9)  # 9ширхэг бар зурна гэж заажээ
plt.show()
```

result:



Array to histogram

```
[3.5070618200937433, 3.6728809617483242, 3.779021661271438, 3.840123438490986, 4.114660317955027,
 4.326657753243537, 4.380181463867804, 4.559520786839379, 4.904500854688624, 4.951701618505617, 5
.057969875981125, 5.091870951523413, 5.265967941688235, 5.39689314930072, 5.411494019550138, 5.41
46268864841245, 5.421664151267173, 5.766663157042094, 5.859822450800329]
```

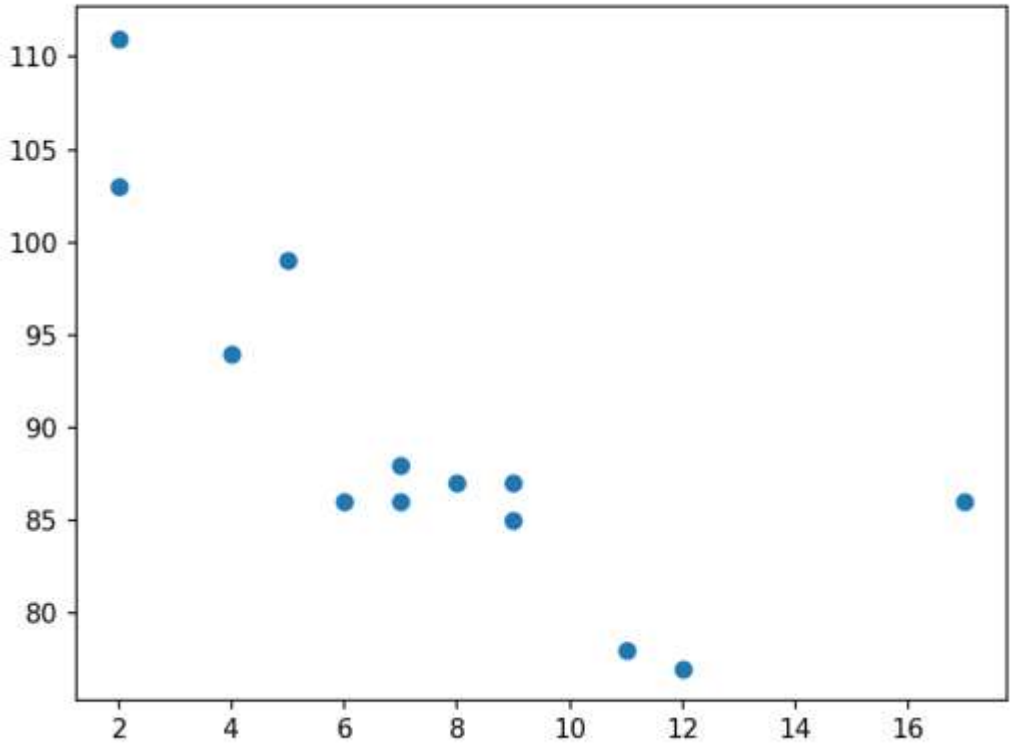


Scatter Plot (diagram)

Example

```
import matplotlib.pyplot as plt

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

plt.scatter(x, y)
plt.show()
```

result:

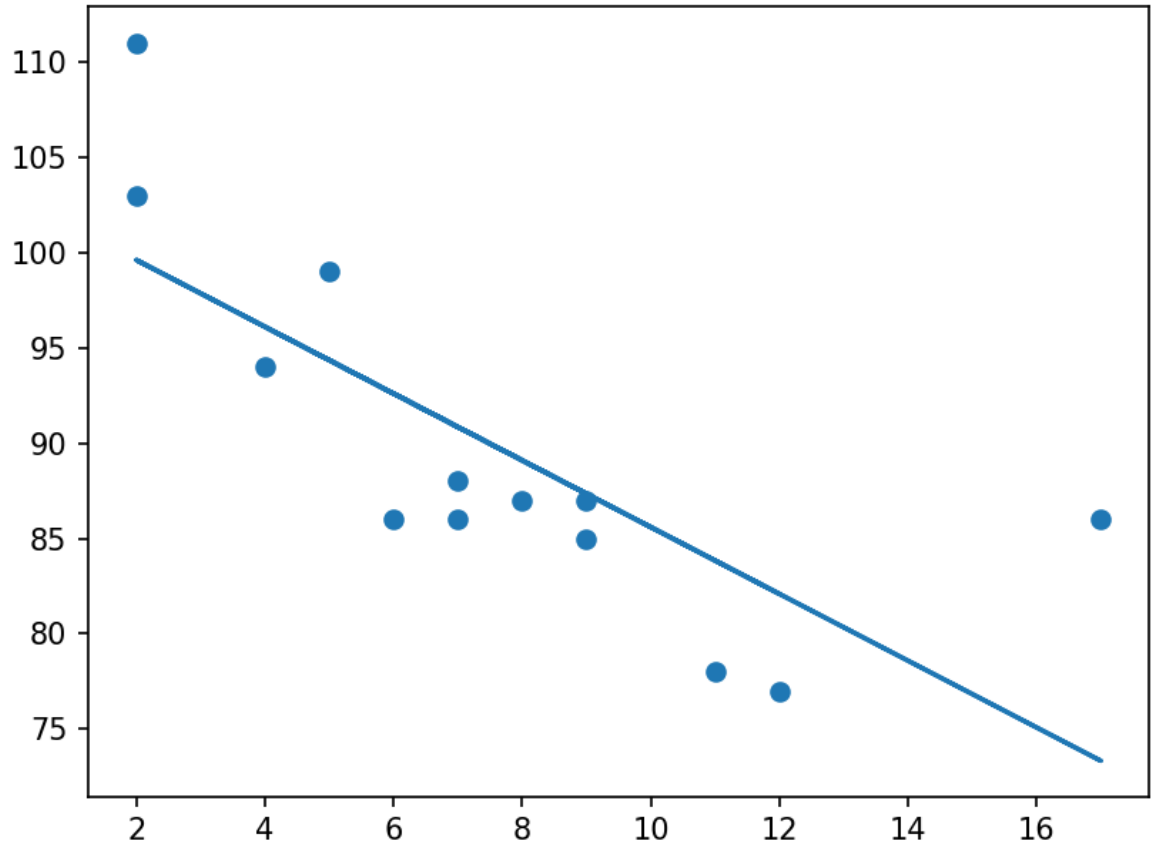


Linear Regression (to predict outcome)

Example

```
import matplotlib.pyplot as plt
from scipy import stats

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]
# linregress method n taamagllig gargdg
slope, intercept, r, p, std_err = stats.linregress(x, y)


def myfunc(x):
    return slope * x + intercept


mymodel = list(map(myfunc, x))

plt.scatter(x, y)
plt.plot(x, mymodel)  # draw line
plt.show()
```

result:

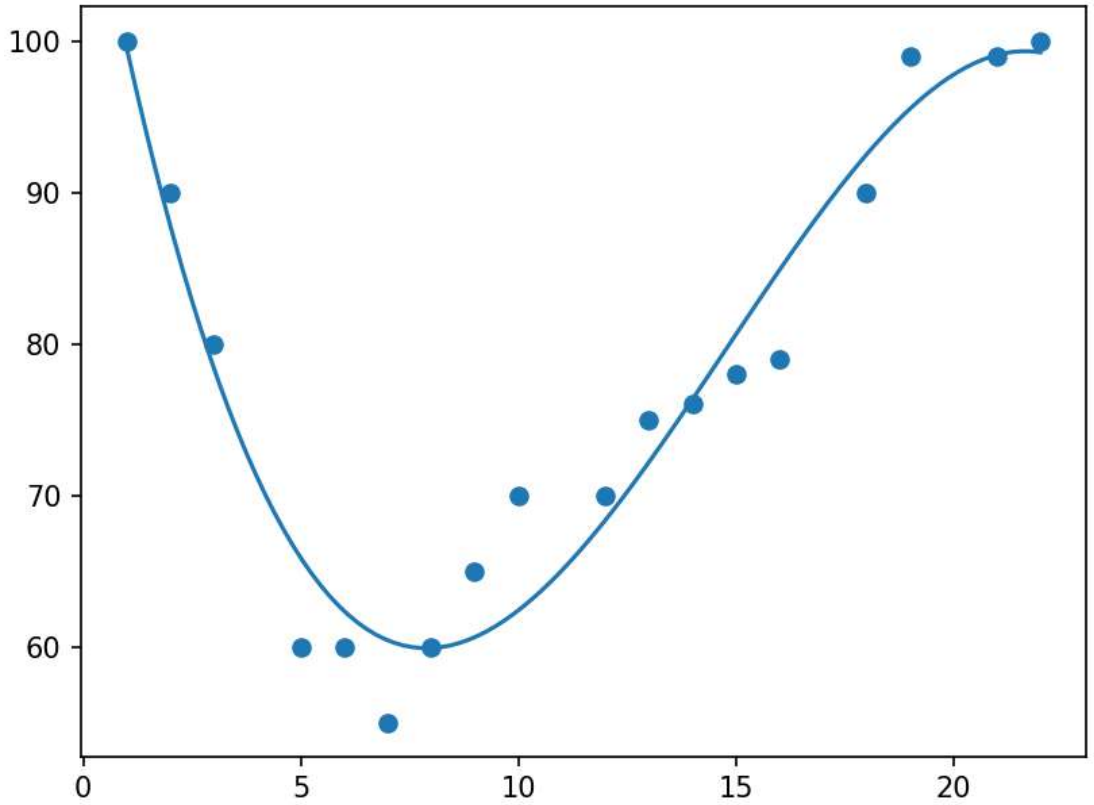


Polynomial Regression

Example

```
import numpy
import matplotlib.pyplot as plt

x = [1, 2, 3, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 18, 19, 21, 22]
y = [100, 90, 80, 60, 60, 55, 60, 65, 70, 70, 75, 76, 78, 79, 90, 99, 99, 100]


mymodel = numpy.poly1d(numpy.polyfit(x, y, 3))
myline = numpy.linspace(1, 22, 100)


plt.scatter(x, y)
plt.plot(myline, mymodel(myline))
plt.show()
```

result:



Multiple Regression

Example

```python
import pandas
from sklearn import linear_model

df = pandas.read_csv("data.csv")

# it is like linear regression But to predict a value based on two or more variables
X = df[['Weight', 'Volume']]
y = df['CO2']

regr = linear_model.LinearRegression()
regr.fit(X, y)

# predict the CO2 emission of a car where the weight is
# 2300kg, and the volume is 1300cm3:
predictedCO2 = regr.predict([[2300, 1300]])

print(predictedCO2)  # output: [107.2087328]
```

csv file

	Car	Model	Volume	Weight	CO2
0	Toyoty	Aygo	1000	790	99
1	Mitsubishi	Space Star	1200	1160	95
2	Skoda	Citigo	1000	929	95
3	Fiat	500	900	865	90
4	Mini	Cooper	1500	1140	105
5	VW	Up!	1000	929	105
6	Skoda	Fabia	1400	1109	90
7	Mercedes	A-Class	1500	1365	92
8	Ford	Fiesta	1500	1112	98
9	Audi	A1	1600	1150	99
10	Hyundai	I20	1100	980	99
11	Suzuki	Swift	1300	990	101
12	Ford	Fiesta	1000	1112	99
13	Honda	Civic	1600	1252	94
14	Hundai	I30	1600	1326	97
15	Opel	Astra	1600	1330	97
16	BMW	1	1600	1365	99
17	Mazda	3	2200	1280	104
18	Skoda	Rapid	1600	1119	104
19	Ford	Focus	2000	1328	105
20	Ford	Mondeo	1600	1584	94
21	Opel	Insignia	2000	1428	99
22	Mercedes	C-Class	2100	1365	99
23	Skoda	Octavia	1600	1415	99
24	Volvo	S60	2000	1415	99
25	Mercedes	CLA	1500	1465	102
26	Audi	A4	2000	1490	104
27	Audi	A6	2000	1725	114
28	Volvo	V70	1600	1523	109
29	BMW	5	2000	1705	114

Scale

Example

```
import pandas
from sklearn import linear_model
from sklearn.preprocessing import StandardScaler
scale = StandardScaler()

df = pandas.read_csv("data.csv")

X = df[['Weight', 'Volume']]

scaledX = scale.fit_transform(X)

print(scaledX)
```

result

```
[[-2.02676985 -1.53669964]
 [-0.66516323 -1.0976426 ]
 [ 0.97244472  0.87811408]
 [ 1.49132725  0.65858556]
 [-0.38916189 -0.21952852]
 [ 0.18124088 -0.21952852]
 [ 0.23644115 -0.21952852]
 [ 0.19964097  1.75622816]]
```

formula

`z = (x - u) / s`

Where `z` is the new value, `x` is the original value, `u` is the mean and `s` is the standard deviation.

If you take the **weight** column from the data set above, the first value is 790, and the scaled value will be:

`(790 - 1292.23) / 238.74 = -2.1`

DataFrame is data structure

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0
5	60	102	127	300.5
6	60	110	136	374.0
7	45	104	134	253.3
8	30	109	133	195.1
9	60	98	124	269.0

A dataset is a collection of data

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