

-

23 2022

JI 8W8YV8 A8 K

8 8

4

8

2,

72

-

16

2022

8

4

8

8 8

8

-1.	- - W -
-2.	-

	<p>W</p>
<p>-3.</p>	<p>- ; -</p>

	W
--	---

8

1.

8

2. W

8

8

3. W

8

4.

8

Grid-

5.

8

8

Microsoft. Yandex Cloud.

6.

8

Web-

7.

8

48

1		6	2			4
2		8	4			4
3		14	2		8	4
4		6	2			4
5	Cloud. Yandex	18	4		10	4
6		14	2		8	4
7		6	2			4
		72	18		26	28

58W

8

8

(2 )  
r K

- 1.
- 2.
- 3.
- 4.

8W

8

8

8

2.

(2 )  
r K

- 1.
- 2.
- 3.

3.  
(2 )  
r k

- 1.
- 2.
- 3.

8W

8

4.

(2 )  
r k

- 1.
- 2.

- 3. (Software as a Service, SaaS).
- 4. (Platform as a Service, PaaS).
- 5. (Infrastructure as a Service, IaaS).
- 6. XaaS).

8

8

5.

(2 )  
r k

- 1.

8

8

8

6 Yandex Cloud.  
(2 )  
r k

- 1. Yandex Cloud.
- 2.

Yandex Cloud.

7.

Amazon Apache.

(2 )  
r k

- 1.
- 2.

Amazon Elastic Computing Cloud.

8

8

8.

(2 )  
r k

- 1.

- 2. Web-

- 3.
- 4.

-

**A8**

**9.**

(2 )  
r k

**8**

**8**

- 1.

-

- 3.

Yandex Cloud.

**-2.**

*вгшн л ш*

*н л н й й к л в л в л м ф к в в й в н в г ш*  
Prezi.com, Google Office.

Microsoft Windows

*б к в*

- 1.

1.

13.



20.

-

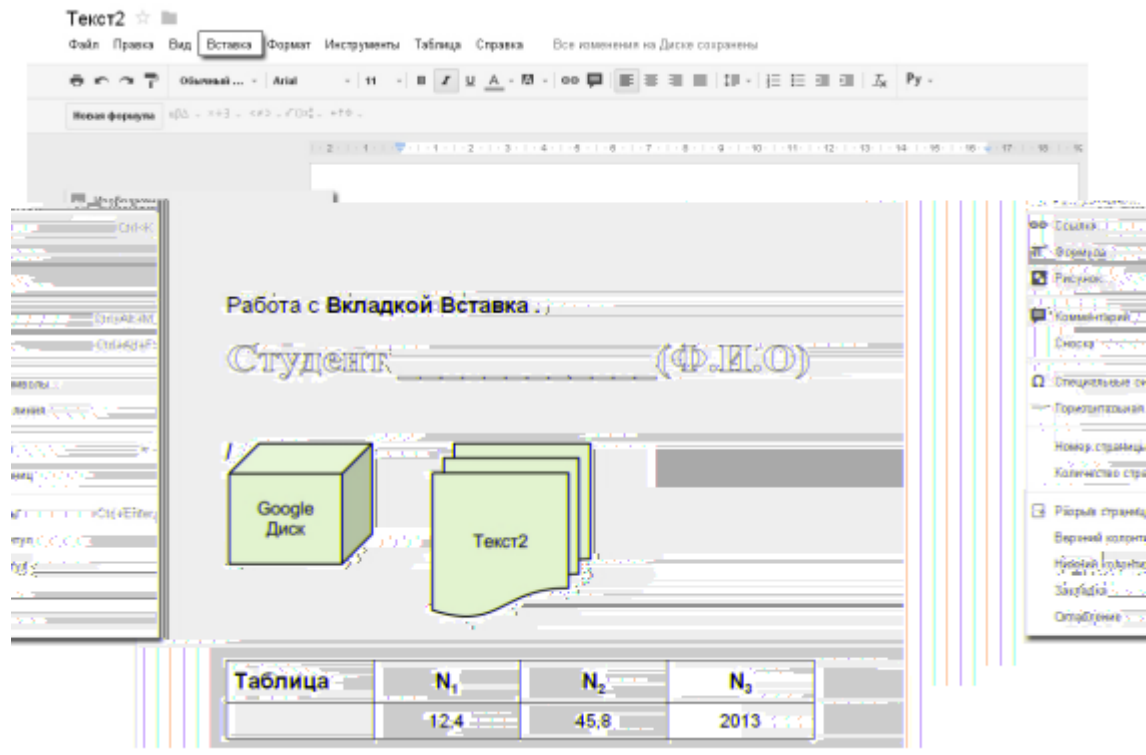
).

C

-

.

-



3-4.

к злї л л Yandex Cloud.

вгш л ш

Yandex Cloud

Yandex Cloud.

нлн ййквл л влмфккв в й влн гш

Microsoft Windows,

Yandex Cloud.

1.

2.

1.

2.

3.

4.

Л

Compute Cloud.

W .

:

5.

Windows. My n Wk uo z vkno:

6. Y

7.

Y

8. W :

RAM.

9.

K  
Yandex.Cloud.

IP-  
J

N

10.

Administrator

Administrator

11.

W .

IP-

Network Level

RUNNING

- 1.
- 2.
- 3.
- 4.

Administrator.

- 5.

> > > > **W** > >  
 > **W** > >

**W** 9

- 
- 

8l kxn idth

8SY

Y

Y

W

M

vCPU.  
Y

J

IP-

•

•

•

Y

У

5-6.

лтфкбв бл тмз лѳв кл зкллрб

вгцн л и:

нлн ййклев л влмфкбв б й ѳв рш

Microsoft Windows, Yandex

Cloud.

б кбъ

**Windows**

**Linux**

1.

2.

```
ssh -t -p 9600 -o IdentitiesOnly=yes -
```

3.

**-user:**

```
ssh -t -p 9600 -o IdentitiesOnly=yes -i ~/.ssh/id_rsa fhm0b28lgfp4tkoa3jl6.yc-  
user@serialssh.cloud.yandex.net
```

4.

Enter

Ctrl + D.

-

sudo passwd

<username>

-

7.

лб кбв l s M b bl mгил кбв й лкб лн бк

вгит л и

н л н йй клв л вл мфкбв бй внв ги

Microsoft Windows, Yandex

Cloud.

б кбв

1.

W

Compute Cloud

2.

W

3.

4.

5.

6.

7.

Stopped.

8.

...

**8.**

*лдж квб зв л Yandex Object Storage*

*вгцн л ш*

*Yandex Object Storage.*

*н л н йй клв л вл мф квб бй вв ги*

Microsoft Windows, Yandex

Cloud.

Л .

0

Л



Serverless:

- 1.
- 2.
- 3.
- 4.
- 
- 
- 
- 5.
- 6.

**Yandex Database.**

**Serverless.**

7.

Provisioning

Running.

YDB-

**Yandex**

**Database.**

YDB-

:

**Uniform**

**Explicit**

**Add split point.**

**By size**

**By load**

**Autopartition by size MBs**

**Min partitions**

**Max partitions**

**Key bloom filter**

**Yandex Database.**

**SQL-**

**SQL-**

**W**

**11-12.**

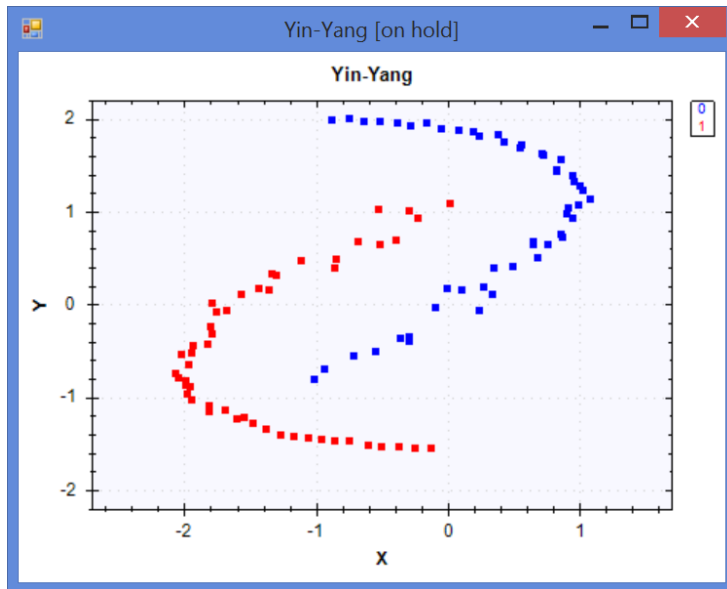
- вруч л и* *ви вкбв д б фзr llb бз vbb κ бн m ruκujÿ u бκ j Yandex Cloud* *Yandex Cloud.*
1. *Yandex Cloud.*
  2. MS Visual Studio.
  - 3.

Windows Forms.

```
// Read the Excel worksheet into a DataTable
DataTable table = new ExcelReader("examples.xls").GetWorksheet("Classification - Yin Yang");

// Convert the DataTable to input and output vectors
double[][] inputs = table.ToJagged<double>("X", "Y");
int[] outputs = table.Columns["G"].ToArray<int>();

// Plot the data
ScatterplotBox.Show("Yin-Yang", inputs, outputs).Hold();
```



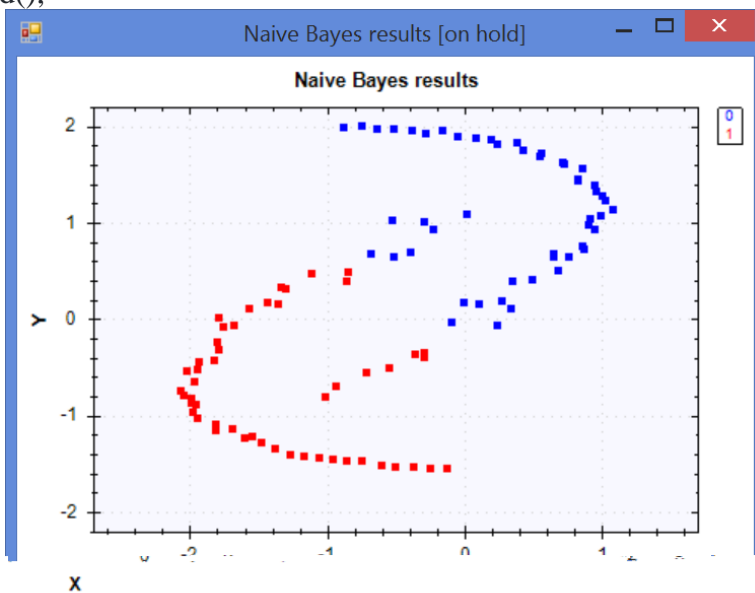
```
// Create a Naive Bayes learning algorithm
var teacher = new NaiveBayesLearning<NormalDistribution>();

// Use the learning algorithm to learn
var nb = teacher.Learn(inputs, outputs);

// At this point, the learning algorithm should have
// figured important details about the problem itself:
int numberOfClasses = nb.NumberOfClasses; // should be 2 (positive or negative)
int numberOfInputs = nb.NumberOfInputs; // should be 2 (x and y coordinates)

// Classify the samples using the model
int[] answers = nb.Decide(inputs);

// Plot the results
ScatterplotBox.Show("Expected results", inputs, outputs);
ScatterplotBox.Show("Naive Bayes results", inputs, answers)
.Hold();
```



```

var teacher = new LinearCoordinateDescent();

// Teach the vector machine
var svm = teacher.Learn(inputs, outputs);

// Classify the samples using the model
bool[] answers = svm.Decide(inputs);

// Convert to Int32 so we can plot:
int[] zeroOneAnswers = answers.ToZeroOne();

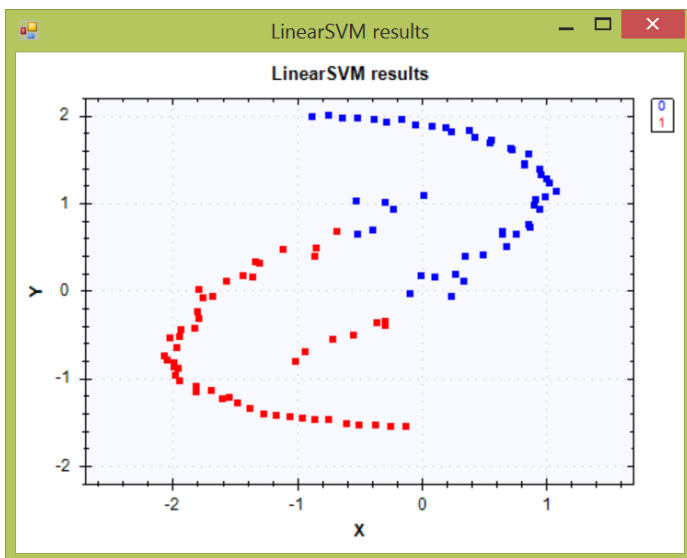
// Plot the results
ScatterplotBox.Show("Expected results", inputs, outputs);
ScatterplotBox.Show("LinearSVM results", inputs, zeroOneAnswers);

// Grab the index of multipliers higher than 0
int[] idx = teacher.Lagrange.Find(x => x > 0);

// Select the input vectors for those
double[][] sv = inputs.Get(idx);

// Plot the support vectors selected by the machine
ScatterplotBox.Show("Support vectors", sv).Hold();

```



13.

*ви вкбв д б фн в н в л л в л к к л л к г б д к бн т г и ж и й и б к j Yandex Cloud*  
*в и и л и*  
Yandex Cloud.  
1.

Windows Forms.

```

// Declare some sample test data.
double[] inputs = { 80, 60, 10, 20, 30 };

```

```

double[] outputs = { 20, 40, 30, 50, 60 };

// Use Ordinary Least Squares to learn the regression
OrdinaryLeastSquares ols = new OrdinaryLeastSquares();

// Use OLS to learn the simple linear regression
SimpleLinearRegression regression = ols.Learn(inputs, outputs);

// Compute the output for a given input:
double y = regression.Transform(85); // The answer will be 28.088

// We can also extract the slope and the intercept term
// for the line. Those will be -0.26 and 50.5, respectively.
double s = regression.Slope; // -0.264706
double c = regression.Intercept; // 50.588235

```

```

double[][] inputs =
{
    // variables: x1 x2 x3
    new double[] { 1, 1, 1 }, // input sample 1
    new double[] { 2, 1, 1 }, // input sample 2
    new double[] { 3, 1, 1 }, // input sample 3
};

double[][] outputs =
{
    // variables: y1 y2
    new double[] { 2, 3 }, // corresponding output to sample 1
    new double[] { 4, 6 }, // corresponding output to sample 2
    new double[] { 6, 9 }, // corresponding output to sample 3
};

// Use Ordinary Least Squares to create the regression
OrdinaryLeastSquares ols = new OrdinaryLeastSquares();

// Now, compute the multivariate linear regression:
MultivariateLinearRegression regression = ols.Learn(inputs, outputs);

// We can obtain predictions using
double[][] predictions = regression.Transform(inputs);

// The prediction error is
double error = new SquareLoss(outputs).Loss(predictions); // 0

// We will use Ordinary Least Squares to create a
// linear regression model with an intercept term
var ols = new OrdinaryLeastSquares()
{
    UseIntercept = true
};

```

```

// Now suppose you have some points
double[][] inputs =
{
    new double[] { 1, 1 },
    new double[] { 0, 1 },
    new double[] { 1, 0 },
    new double[] { 0, 0 },
};

// located in the same Z (z = 1)
double[] outputs = { 1, 1, 1, 1 };

// Use Ordinary Least Squares to estimate a regression model
MultipleLinearRegression regression = ols.Learn(inputs, outputs);

// As result, we will be given the following:
double a = regression.Coefficients[0]; // a = 0
double b = regression.Coefficients[1]; // b = 0
double c = regression.Intercept; // c = 1

// This is the plane described by the equation
//  $ax + by + c = z \Rightarrow 0x + 0y + 1 = z \Rightarrow 1 = z$ .

// We can compute the predicted points using
double[] predicted = regression.Transform(inputs);

// And the squared error loss using
double error = new SquareLoss(outputs).Loss(predicted);

double[][] input =
{
    // age, smokes?, had cancer?
    new double[] { 55, 0 }, // false - no cancer
    new double[] { 28, 0 }, // false
    new double[] { 65, 1 }, // false
    new double[] { 46, 0 }, // true - had cancer
    new double[] { 86, 1 }, // true
    new double[] { 56, 1 }, // true
    new double[] { 85, 0 }, // false
    new double[] { 33, 0 }, // false
    new double[] { 21, 1 }, // false
    new double[] { 42, 1 }, // true
};

bool[] output = // Whether each patient had lung cancer or not
{
    false, false, false, true, true, true, false, false, false, true
};

// Create a new Iterative Reweighted Least Squares algorithm
var learner = new IterativeReweightedLeastSquares<LogisticRegression>()
{
    Tolerance = 1e-4, // Let's set some convergence parameters
};

```

```

Iterations = 100, // maximum number of iterations to perform
Regularization = 0
};

// Now, we can use the learner to finally estimate our model:
LogisticRegression regression = learner.Learn(input, output);
// For the age variable, we have that individuals with
// higher age have 1.021 greater odds of getting lung
// cancer controlling for cigarette smoking.
double ageOdds = regression.GetOddsRatio(1); // 1.0208597028836701

double smokeOdds = regression.GetOddsRatio(2); // 5.8584748789881331

// If we would like to use the model to predict a probability for
// each patient regarding whether they are at risk of cancer or not,
// we can use the Probability function:

double[] scores = regression.Probability(input);

// Finally, if we would like to arrive at a conclusion regarding
// each patient, we can use the Decide method, which will transform
// the probabilities (from 0 to 1) into actual true/false values:

bool[] actual = regression.Decide(input);

```

- 
- 
- 

);

8 8      68

2      3

1.                      IT
- 2.
3.                      Cloud Service.
- 4.
- 5.
- 6.

- 7.
- 8.
- 9.
10. Infrastructure-as-a-Service (IaaS).
11. Software-as-a-Service (SaaS).
- 12.
- 13.
14. Platform-as-a-Service (PaaS).
15. Yandex Cloud.
16. Yandex Cloud.
17. Yandex Cloud.
18. Yandex Cloud Object Storage.
19. Yandex Cloud Database.
20. Yandex Cloud

$$R = 2 + \frac{3}{5} \sum_{i=1}^5 Q_i,$$

$Q_i$

([www.moodle.smolgu.ru](http://www.moodle.smolgu.ru)).

	2

**88**

- 1.
- 2.
- 3.

-

- 1.



		*)
1		
2		

(\*)

2.

1		4,75-5
2		3,75-4,5
3		3-3,5
4		

- 
- 
- 

**78**

**7.1.**

1.

507

ISBN 978-5-534-11380-8.

**A8 8Y**

1.

90

ISBN 978-5-9916-9975-4.

URL:

**7.3.**

1.

moodle.smolgu.ru).

2.

intuit.ru).

3.

opened.ru)

4. Cloud.Yandex.ru

**8.**

224 12

9.

- 1.
- 2.
3. Yandex Cloud
- 4.
- 5.

9

ДОКУМЕНТ ПОДПИСАН  
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 03B6A3C600B7ADA9B742A1E041DE7D81B0  
Владелец: Артеменков Михаил Николаевич  
Действителен: с 04.10.2021 до 07.10.2022