

23 2022

4
8
4, 144
8

16 2022

10

2022

<p>-1.</p>	<p>-</p> <p>-</p> <p>:</p> <p>-</p>
<p>-2.</p>	<p>-</p>

--	--

1.

2.

, , ,

3.

4.

- - -

5.

-

6.

7.

.

8.

EM-

(k-means).
Fuzzy c-means

9.

10.

11.

12.

13.

REPAST. : Java Agent Development Framework (Jade), ABLE,

14.

4

1		8	2		2	4
2		8	2		2	4
3		10	2		4	4
4	curve).	8	2		2	4
5		8	2		2	4
6		8	2		2	4
7		12	2		4	6
8		12	2	-	4	6
9		12	2	-	4	6
10		12	4	-	2	6

11		12	4	-	2	6
12		12	4	-	2	6
13		12	4	-	2	6
14		10	2	-	2	6
		144	36		36	45+27

5

1.

.

2.

, , ,

3.

4.

(ROC curve).

OC-

-

5.

-

6.

.

7.

.

.

8.

EM-

(k-means).

-

9.

10-11.

12-13.

14-15.

16-17.

18.

-3

4-5

6

-

9-10

-12

-

-17

-

ROC

- dataset **Titanic:** test.csv train.csv
 (https://russianblogs.com/article/49401398867/).

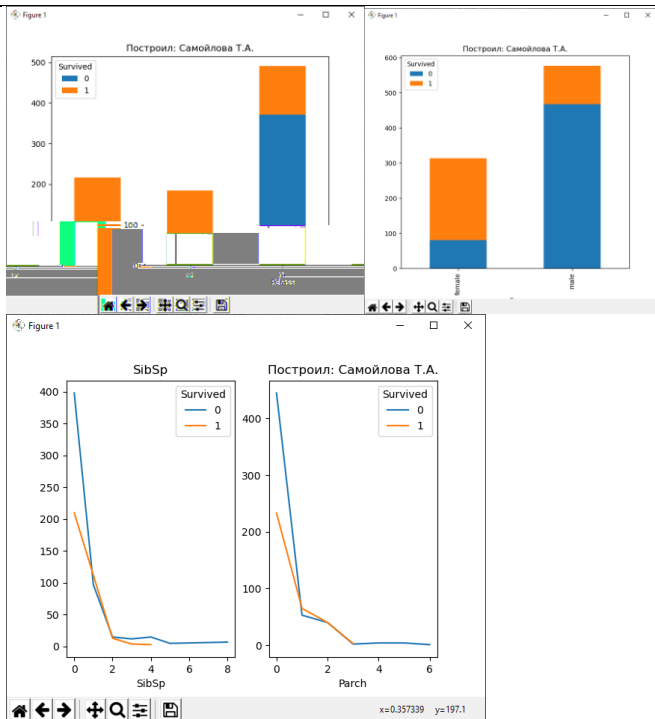
test.csv train.csv

-
-
-
-

:

	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked
0	0	3	male	1	0	7.2500	S
1	1	1	female	1	0	71.2833	C
2	1	3	female	0	0	7.9250	S
3	1	1	female	1	0	53.1000	S
4	0	3	male	0	0	8.0500	S

	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked
0	0	3	1	1	0	7.2500	2
1	1	1	0	1	0	71.2833	0
2	1	3	0	0	0	7.9250	2
3	1	1	0	1	0	53.1000	2
4	0	3	1	0	0	8.0500	2



python-

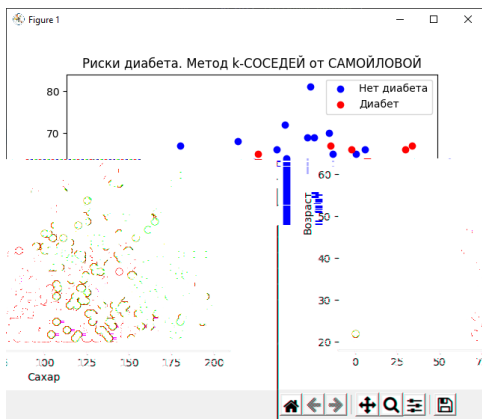
age,salary,house,class

18,25,1,1
22,100,1,1
30,10,0,0
32,120,0,1
24,15,1,0
25,22,1,1
32,20,0,0
19,15,1,0
52,135,0,1
.....

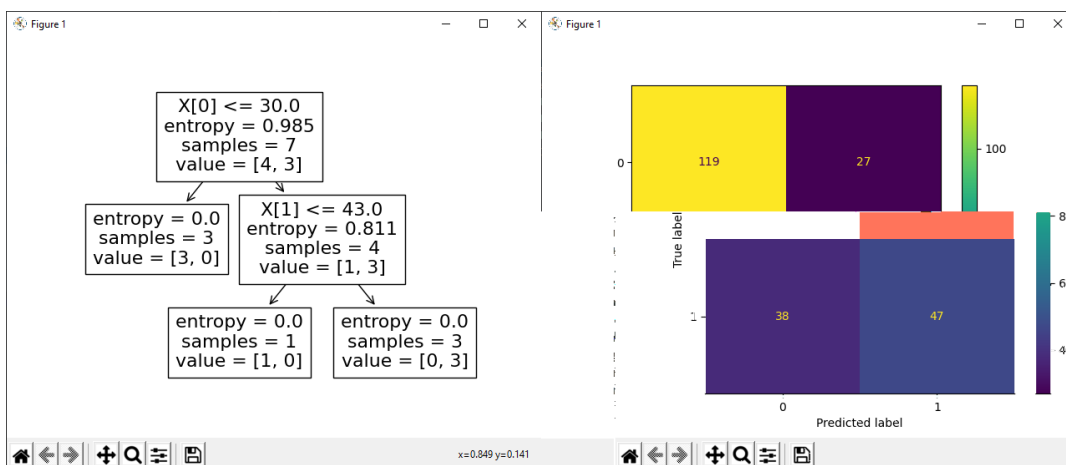
2 .

F-

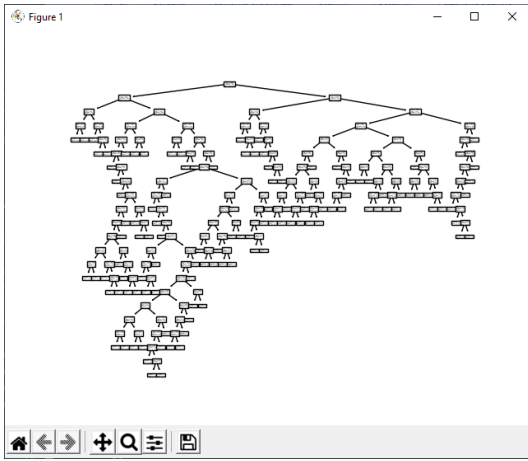
: UCI Machine Learning Repository <http://archive.ics.uci.edu/ml/>



ID

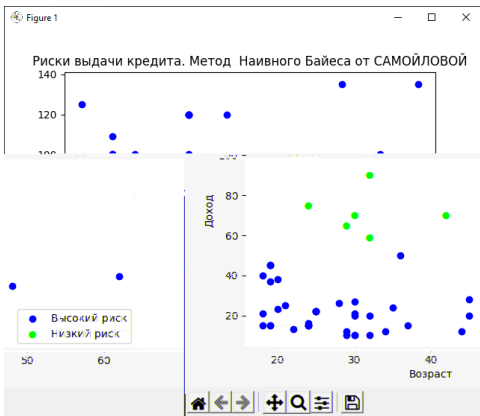


C CART.

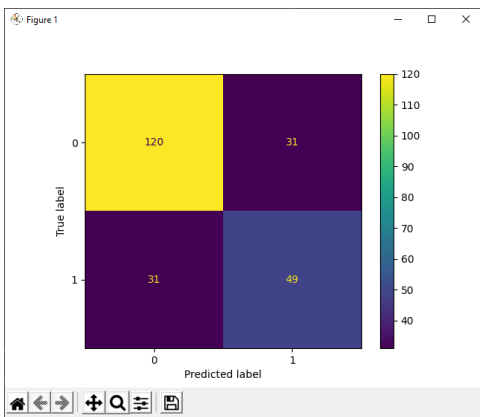


-

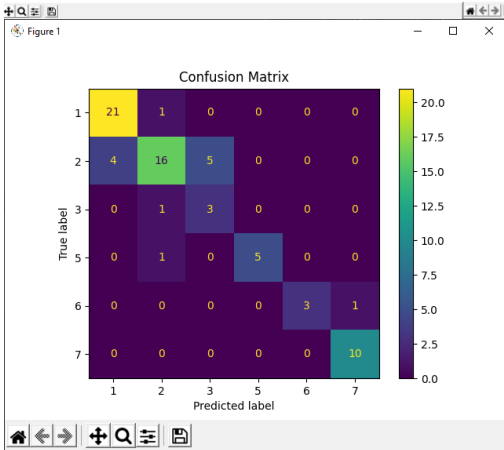
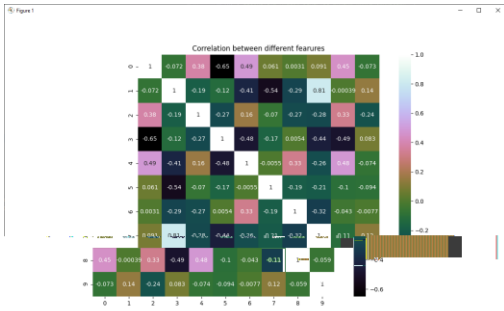
f-



-



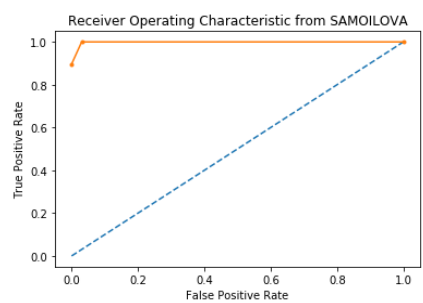
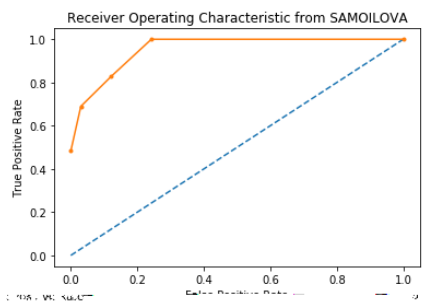
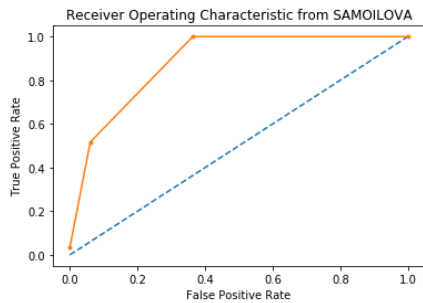
-



ROC

f-

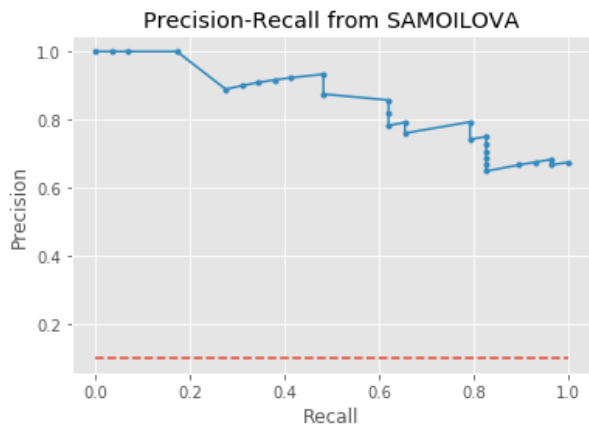
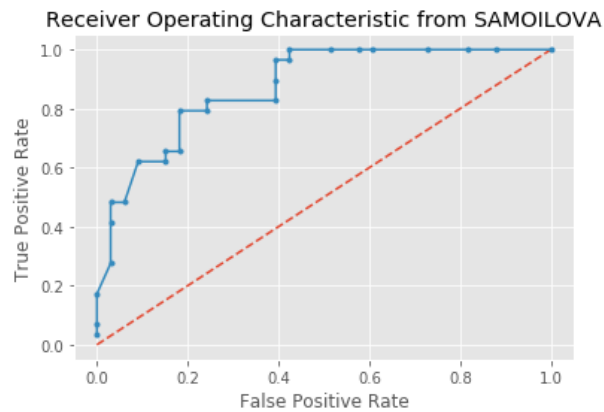
ROC-

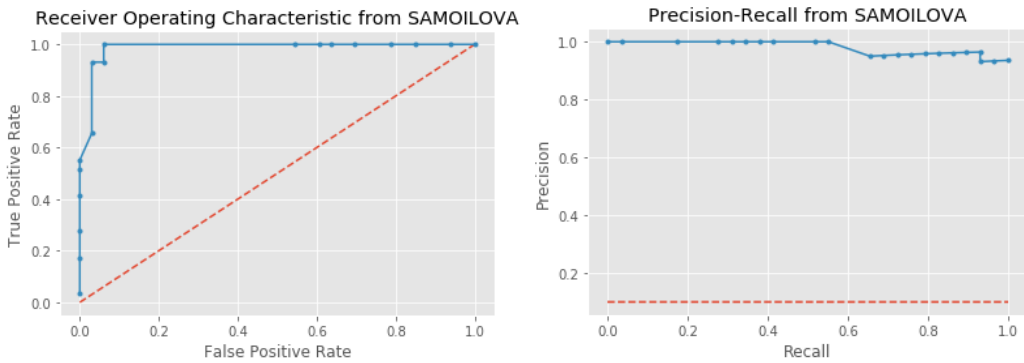
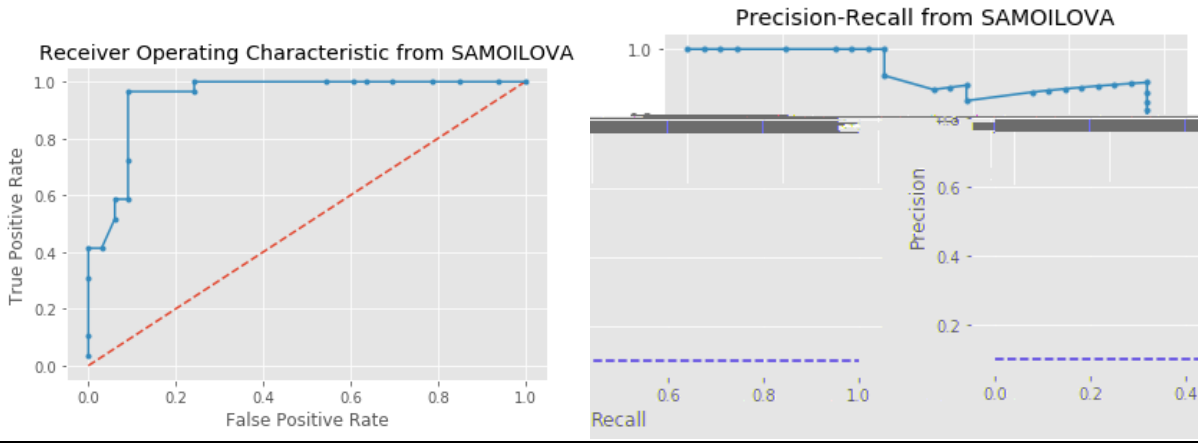


ROC -

PR-

:



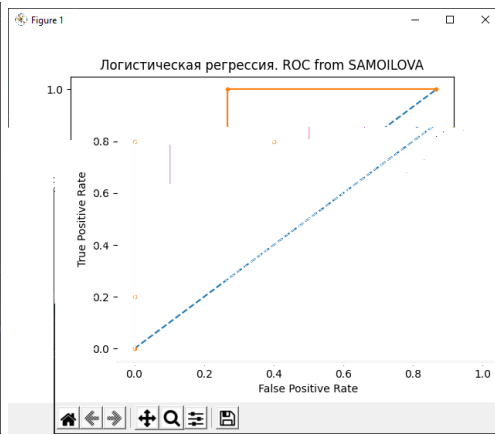
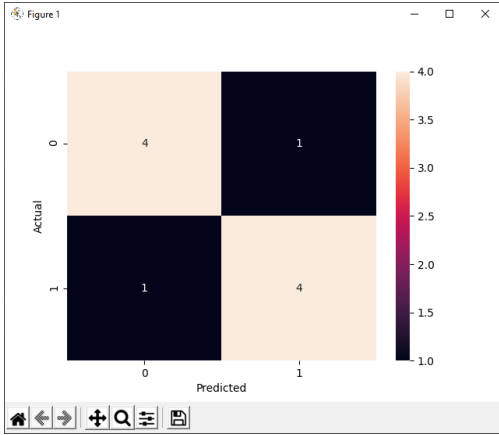


AdaBoostClassifier
CSV

f- ROC AUC.

The image shows two screenshots of the Spyder Python IDE. The left screenshot shows the code for training and evaluating the SAMOILOVA classifier. The code includes data loading, splitting into train and test sets, training an AdaBoostClassifier, and calculating the ROC curve and Precision-Recall curve. The right screenshot shows the code for training and evaluating the AdaBoostClassifier. The code includes data loading, splitting into train and test sets, training an AdaBoostClassifier, and calculating the ROC curve and Precision-Recall curve. Both screenshots show the resulting ROC and Precision-Recall plots for each classifier.

f- ROC AUC



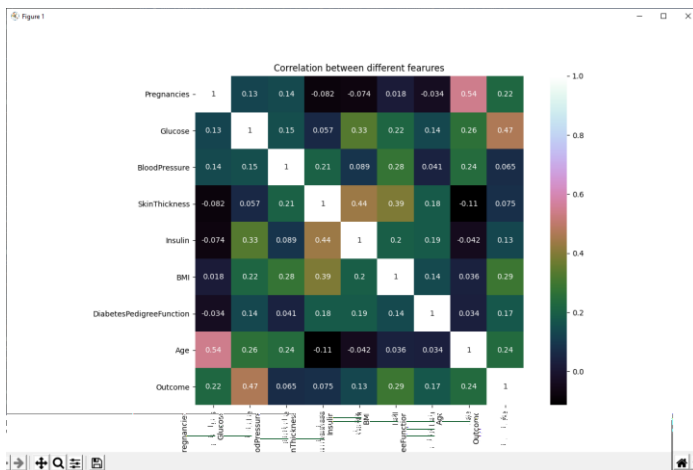
```
[[0.3018346 0.6981654 ]
 [0.69176147 0.30823853]
 [0.05322031 0.94677969]
 [0.20417163 0.79582837]
 [0.58169393 0.41830607]
 [0.88906961 0.11093039]
 [0.02920068 0.97079932]
 [0.11033348 0.88966652]
 [0.67029328 0.32970672]
 [0.12638422 0.87361578]]
```

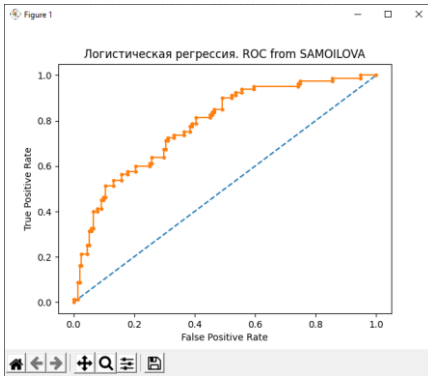
Accuracy: 0.8

	precision	recall	f1-score	support
0	0.80	0.80	0.80	5
1	0.80	0.80	0.80	5
accuracy			0.80	10
macro avg	0.80	0.80	0.80	10
weighted avg	0.80	0.80	0.80	10

AUC: 0.920

2





```
Name: Outcome, Length: 537, dtype: int64
Pregnancies  Glucose  Insulin  BMI  Age
334          1     95     58 23.9  22
139          5    105    325 36.9  28
485          0    135    250 42.3  24
547          4    131    166 33.1  28
18           1    103     83 43.3  33
..          ...    ...    ...    ...
71           5    139    140 28.6  26
106          1     96     0 22.4  27
270         10    101     0 45.6  38
435          0    141     0 42.4  29
102          0    125     0 22.5  21
[537 rows x 5 columns]

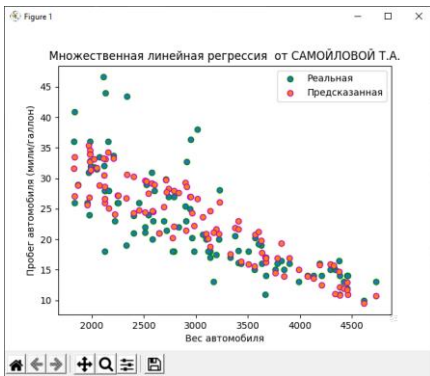
      precision  recall  f1-score  support
0           0.76   0.87   0.81     151
1           0.66   0.49   0.56     80
accuracy                0.74     231
macro avg       0.71   0.68   0.69     231
weighted avg    0.73   0.74   0.72     231
```

LinearRegression.

<https://archive.ics.uci.edu/ml/datasets/Auto+MPG>

('cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'model_year', 'origin', 'car_name'):

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.



G:\PyCharm_tatsamoilova\sklearn_simple\REGR\My_Auto_Lin_Regr.py
 RMSE= 4.069215579662049

Process finished with exit code 0

LinearRegression,

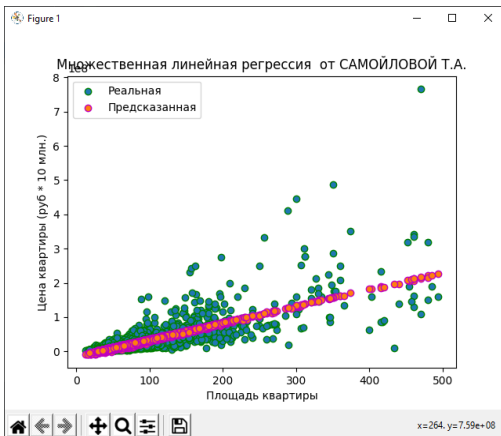
DataFrame: wallsMaterial, floorNumber, floorsTotal, totalArea, kitchenArea, latitude, longitude, price

DataFrame: floorNumber, floorsTotal, totalArea, kitchenArea.

price

moscow_dataset_2020.csv

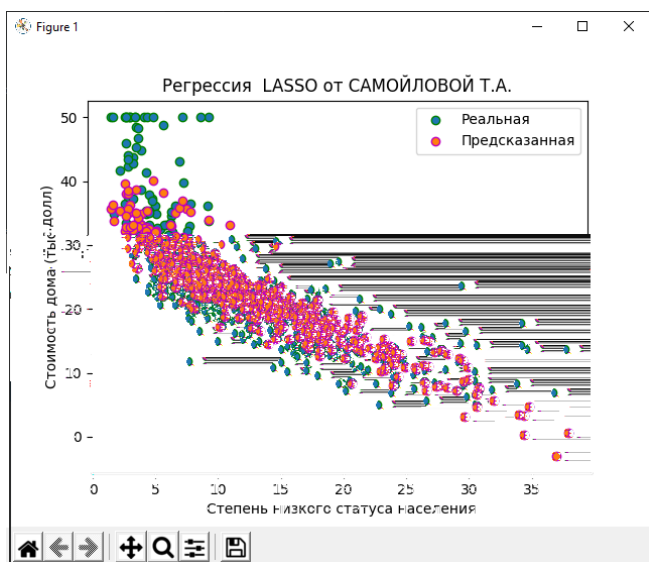
-2019.



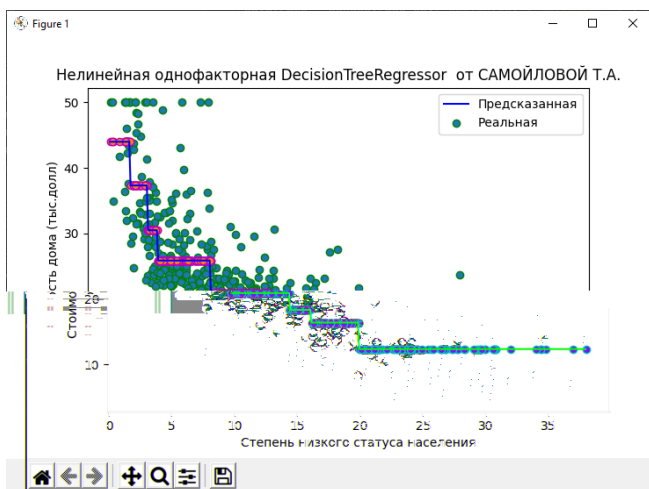
```
wallsMaterial floorNumber floorsTotal ... latitude longitude price
0 brick 1 5.0 ... 55.723379 37.628577 5600000
1 brick 1 5.0 ... 55.725980 37.671031 4650000
2 brick 1 5.0 ... 55.735976 37.657817 2990000
3 brick 1 7.0 ... 55.786698 37.595321 4390000
4 brick 2 5.0 ... 55.767894 37.665920 4890000
[5 rows x 8 columns]
RMSE= 16059592.911492676
```

Задание 3.

LassoCV

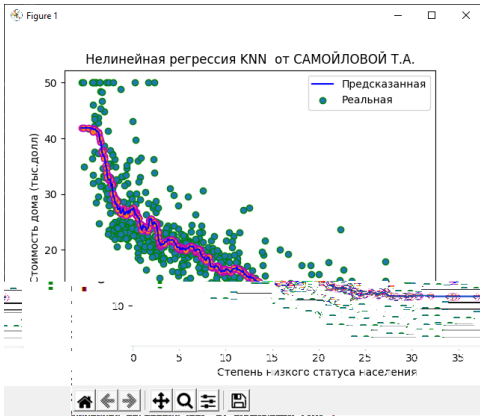


4



5

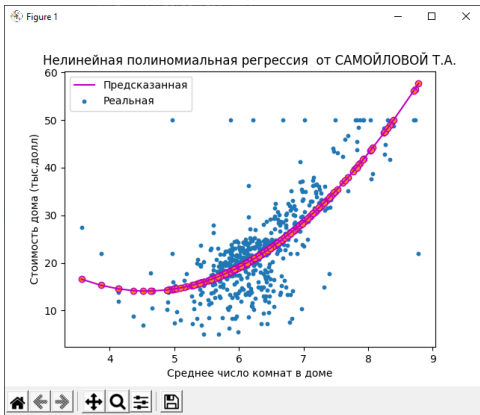
KNeighborsRegressor



rmse= 5.089057644349284
 - r2 0.6932172724449991

6

LinearRegression



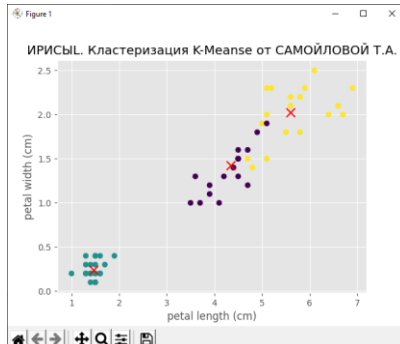
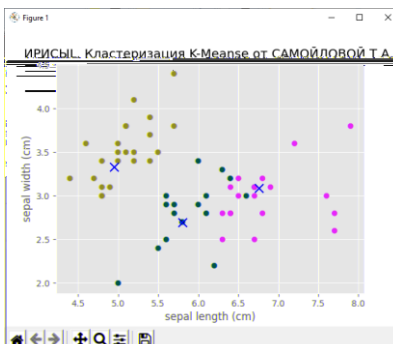
Repository - <http://archive.ics.uci.edu/ml/>

- mean_squared_error (MSE),

r2_score

1

K-Means




```
1 1 1 0 0 2 1 1 0 2 2 0 2 0 2 0 1 2 0 1 1 1 0]
1 1 1 2 0 0 1 1 0 2 2 0 2 0 2 0 1 2 0 1 1 1 0]
```

0 1 1 2 0

- accuracy_score = 0.9333333333333333

- F-

	precision	recall	f1-score	support
0	0.89	0.89	0.89	19
1	1.00	1.00	1.00	23
2	0.89	0.89	0.89	18
accuracy			0.93	60
macro avg	0.93	0.93	0.93	60
weighted avg	0.93	0.93	0.93	60

Spending Score (1-

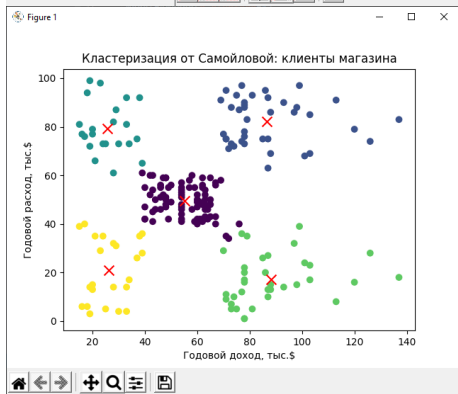
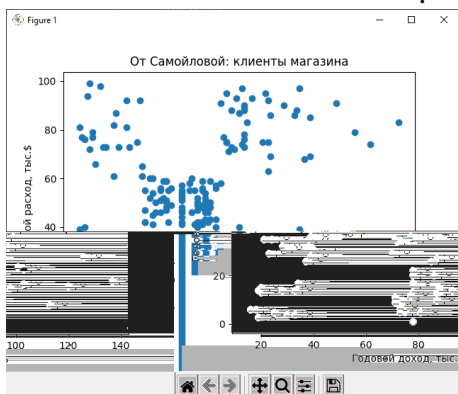
CustomerID,Genre,Age,Annual Income (k\$),Spending Score (1-100)

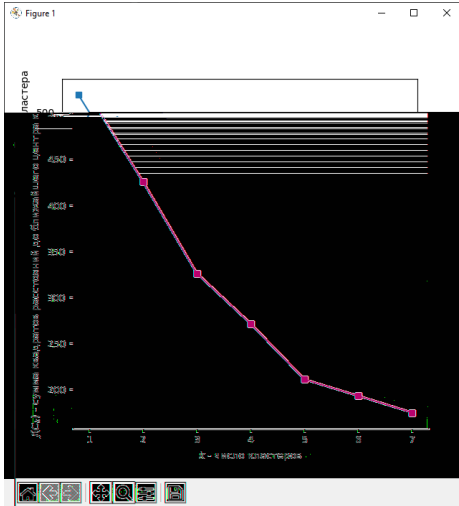
0001, Male, 19, 15, 39

0002, Male, 21, 15, 81

0003, Female, 20, 16, 6

0004, Female, 23, 16, 77

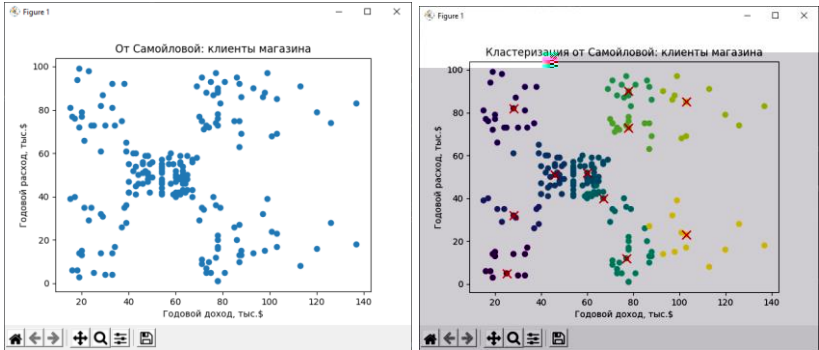




```
[4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4
2 4 2 4 2 4 0 4 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 3 1 0 1 3 1 3 1 0 1 3 1 3 1 3 1 3 1 0 1 3 1 3 1
3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1
1 3 1 3 1 3 1 3 1 3 1 3 1]
```

3 Affinity Propagation (<https://habr.com/ru/post/321216/>)

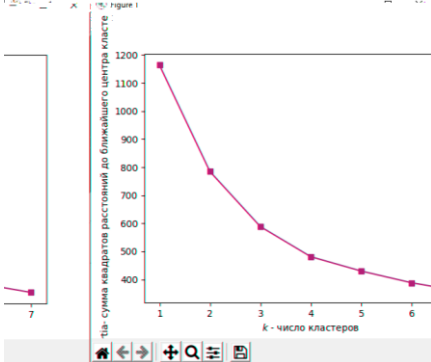
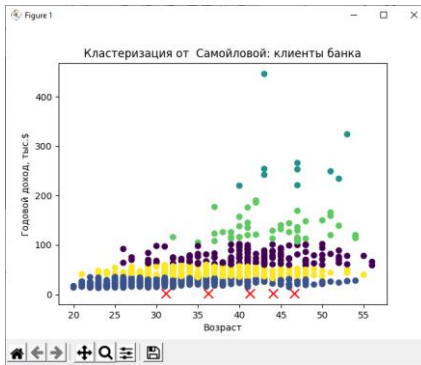
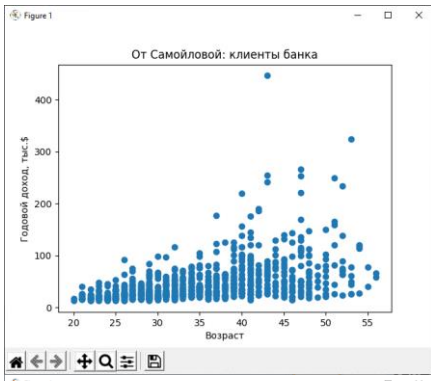
Genre, Age, Annual Income (k\$), Spending Score (1-100)].



```
[2 1 0 1 2 1 0 1 0 1 0 1 0 1 0 1 2 1 2 1 2 1 0 1
0 1 2 3 2 1 0 1 0 1 0 1 0 1 2 1 2 1 2 3 2 3 3 3
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 4 5 5 4
4 4 5 4 5 4 4 4 4 5 4 4 5 4 4 4 5 5 4 4 5 4 5 4
4 5 4 7 5 8 5 7 6 8 6 8 5 8 6 7 6 8 6 8 6 7 5 7
6 7 5 8 6 7 6 7 6 8 6 7 6 8 6 8 5 7 6 7 6 8 6 7
10 8 6 8 6 7 6 7 6 8 10 9 10 9 10 9 10 9 10 9 10 9
10 9 10 9 10 9 10 9]
```

4.

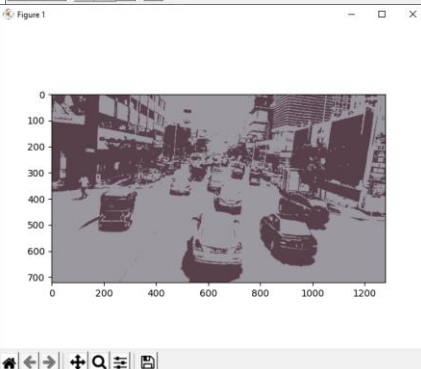
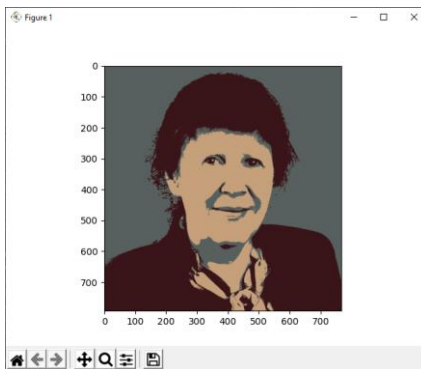
Num, Customer Id, Age, Edu, Years Employed, Income, Card Debt, Other Debt, Defaulted, DebtIncomeRatio
 0,1,41,2,6,19,0.124,1.073,0.0,6.3
 1,2,47,1,26,100,4.582,8.218,0.0,12.8
 2,3,33,2,10,57,6.1110000000000001,5.8020000000000005,1.0,20.9
 3,4,29,2,4,19,0.6809999999999999,0.516,0.0,6.3
 4,5,47,1,31,253,9.308,8.908,0.0,7.2

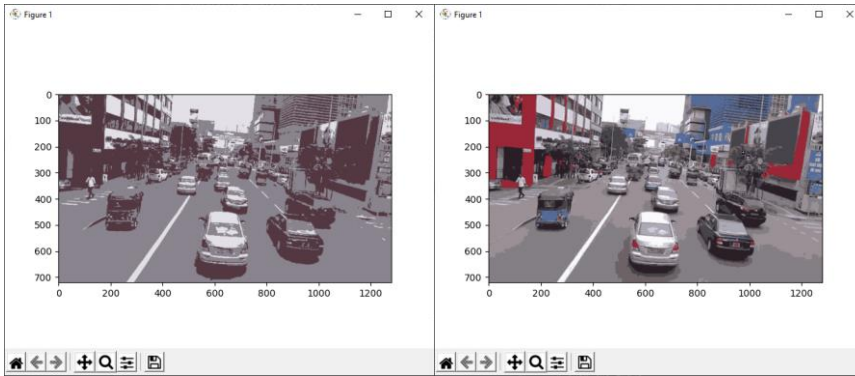


5

K-Means

CV

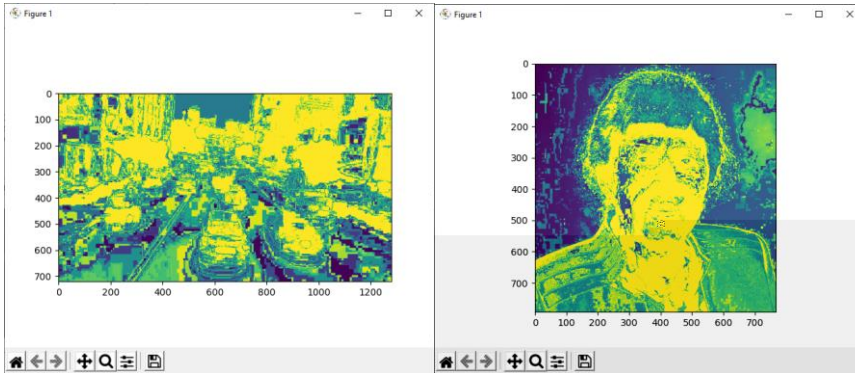




6

DBSCAN

sklearn.cluster



7.

8

python-keras tensorflow.

<https://archive.ics.uci.edu/ml/datasets/diabetas>

5. 2-

<https://archive.ics.uci.edu/ml/datasets/Auto+MPG>

('cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'model_year', 'origin', 'car_name'):

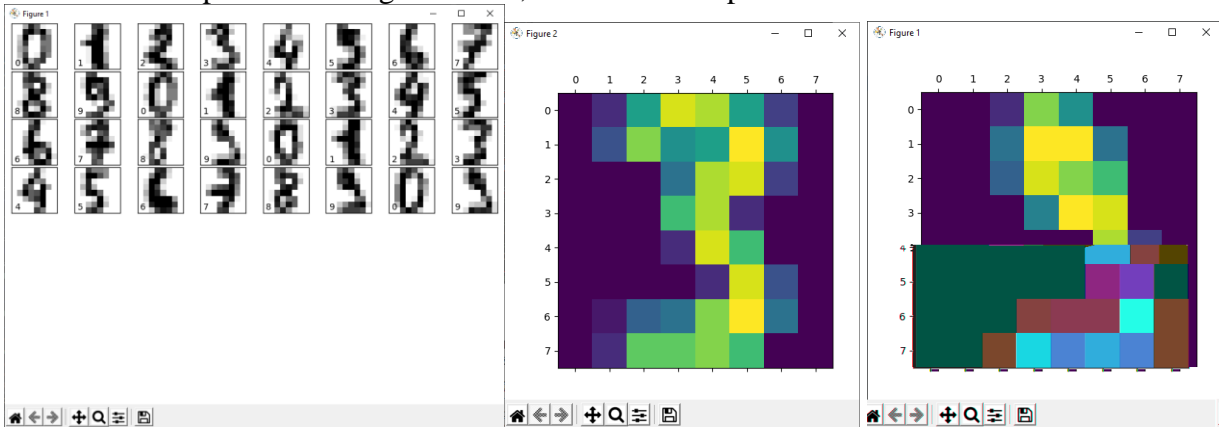
- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.

MLPClassifier

MNIST

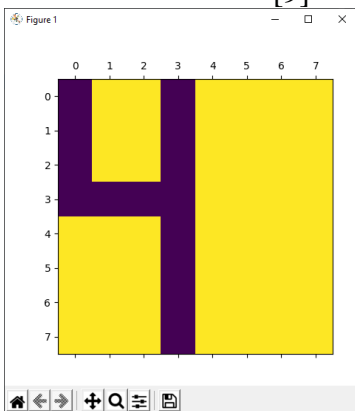
We have 1797 samples

Number of samples in training set: 1437, number of samples in test set: 360



- [3]

- [9]



- [4]

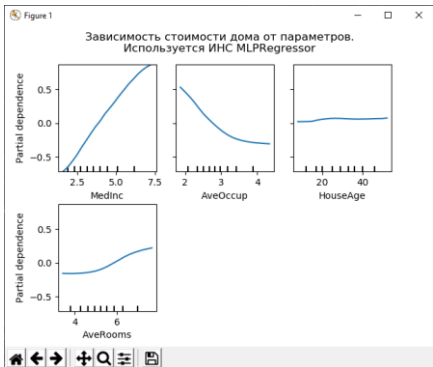
4.

MLPClassifier

https://scikit-learn.org/stable/auto_examples/inspection/plot_partial_dependence.html#california-housing-data-preprocessing

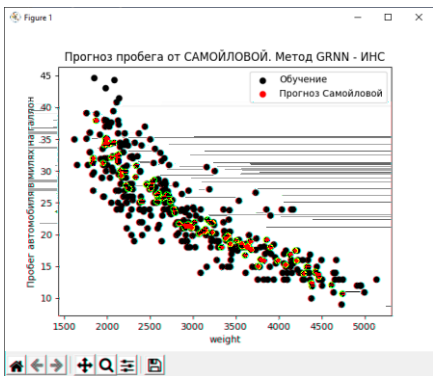
MedInc, HouseAge, AveRooms, AveBedrms, Population, AveOccup, Latitude Longitude.

Target -



(Neural Networks with Radial Basis

-



[[11.86674329]
[15.98118317]

[21.34850449]
[12.45185296]]

OK!!

WindowsForm -

AForge .NET

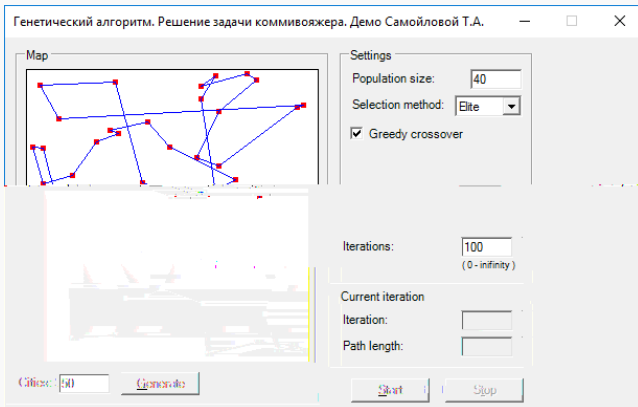
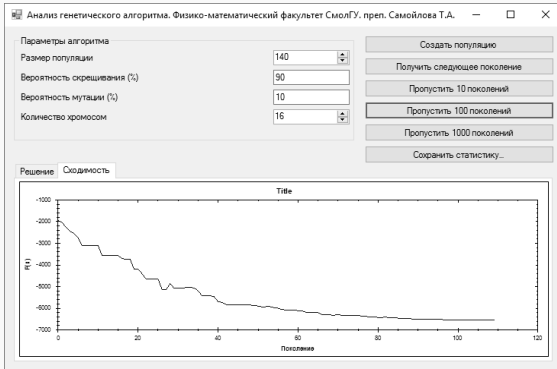
AForge

Chart

Accord .NET

Accord.Genetic

Excel



scikit-fuzzy

Fuzzy C-Means

Fuzzy partition coefficient (FPC).

random -

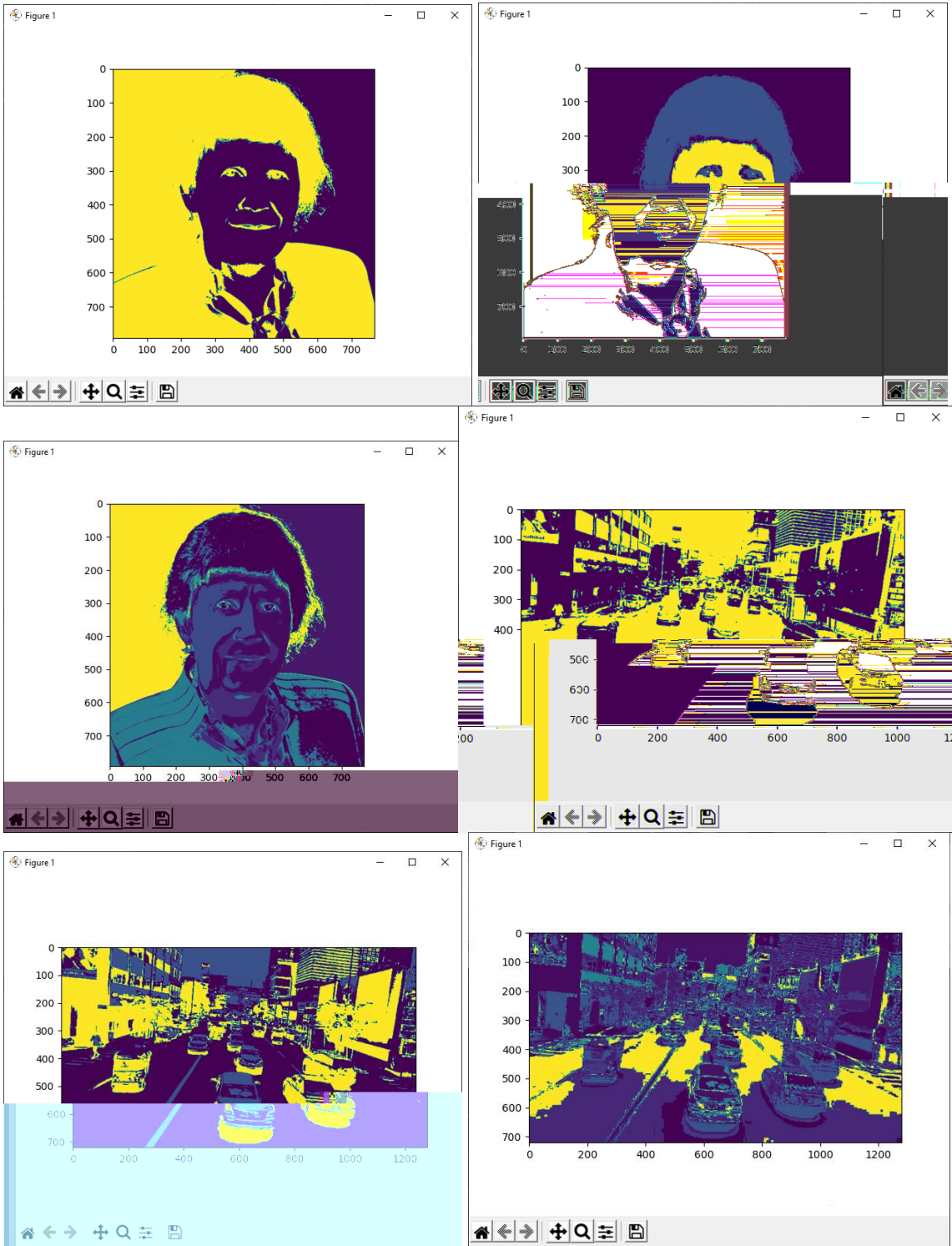
x,y)

FPC)

FPC
2

K-Means

scikit-fuzzy

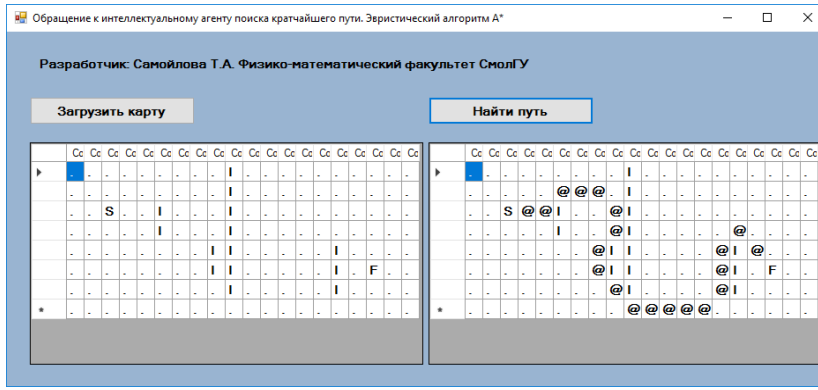


7

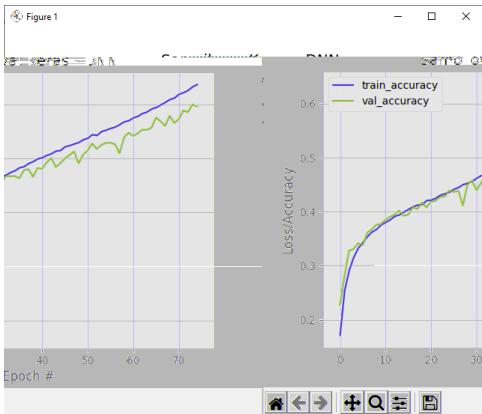
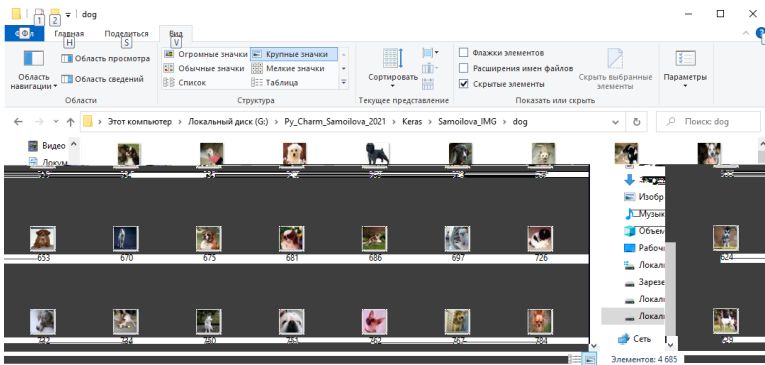
c#-

WEBAPI-

Windows Form -



8
1.



```
[Loading images...
Загрузила рисунки
Epoch 1/75
1172/1172 [=====] - 17s 14ms/step - loss: 2.2387 - accuracy: 0.1713 -
val_loss: 2.1468 - val_accuracy: 0.2271
Epoch 2/75
1172/1172 [=====] - 18s 15ms/step - loss: 2.0790 - accuracy: 0.2540 -
val_loss: 2.0104 - val_accuracy: 0.2804
Epoch 3/75
1172/1172 [=====] - 17s 14ms/step - loss: 1.9818 - accuracy: 0.2912 -
val_loss: 1.9385 - val_accuracy: 0.3279
Epoch 4/75
1172/1172 [=====] - 17s 14ms/step - loss: 1.9275 - accuracy: 0.3157 -
val_loss: 1.9023 - val_accuracy: 0.3309
Epoch 5/75
```

```

Epoch 72/75
1172/1172 [=====] - 17s 14ms/step - loss: 1.0821 - accuracy: 0.6217 -
val_loss: 1.1715 - val_accuracy: 0.5888
Epoch 73/75
1172/1172 [=====] - 17s 14ms/step - loss: 1.0719 - accuracy: 0.6258 -
val_loss: 1.1702 - val_accuracy: 0.5856
Epoch 74/75
1172/1172 [=====] - 17s 14ms/step - loss: 1.0566 - accuracy: 0.6325 -
val_loss: 1.1427 - val_accuracy: 0.6000
Epoch 75/75
1172/1172 [=====] - 17s 14ms/step - loss: 1.0427 - accuracy: 0.6367 -
val_loss: 1.1614 - val_accuracy: 0.5962
[Evaluating network...]

```

	precision	recall	f1-score	support
Samoilova_IMG/airplane	0.69	0.57	0.62	1250
Samoilova_IMG/bird	0.44	0.71	0.54	1317
Samoilova_IMG/car	0.83	0.57	0.68	1274
Samoilova_IMG/cat	0.56	0.39	0.46	1240
Samoilova_IMG/deer	0.51	0.52	0.51	1254
Samoilova_IMG/dog	0.59	0.45	0.51	1146
Samoilova_IMG/frog	0.61	0.65	0.63	1277
Samoilova_IMG/horse	0.79	0.57	0.66	1235
Samoilova_IMG/ship	0.56	0.82	0.66	1281
Samoilova_IMG/truck	0.65	0.69	0.67	1226
accuracy			0.60	12500
macro avg	0.62	0.59	0.59	12500
weighted avg	0.62	0.60	0.60	12500

```

Serializing network and label binarizer...
Process finished with exit code 0

```



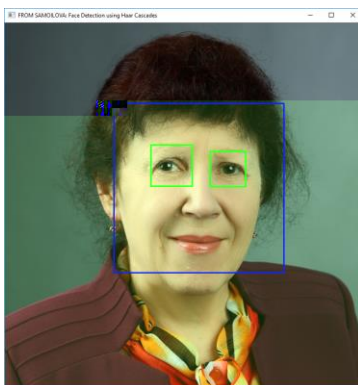
```

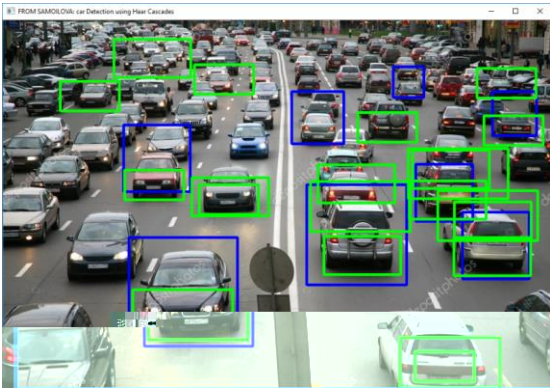
[INFO] loading network and label binarizer...
Samoilova_IMG/dog

```

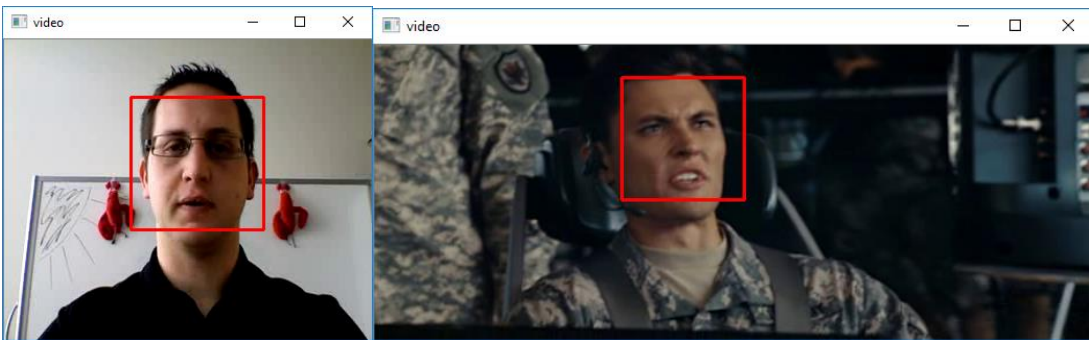
Python-

OpenCV
XML -



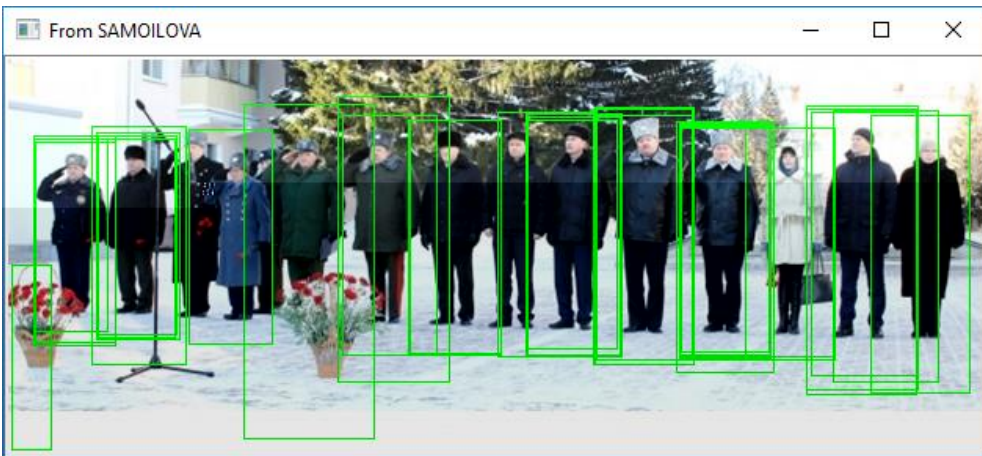


Python- XML -



5. python- opencv

HOG,

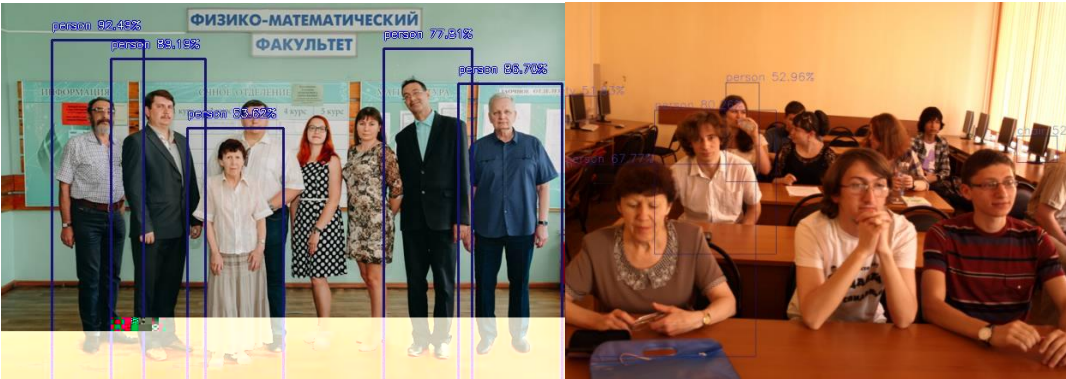
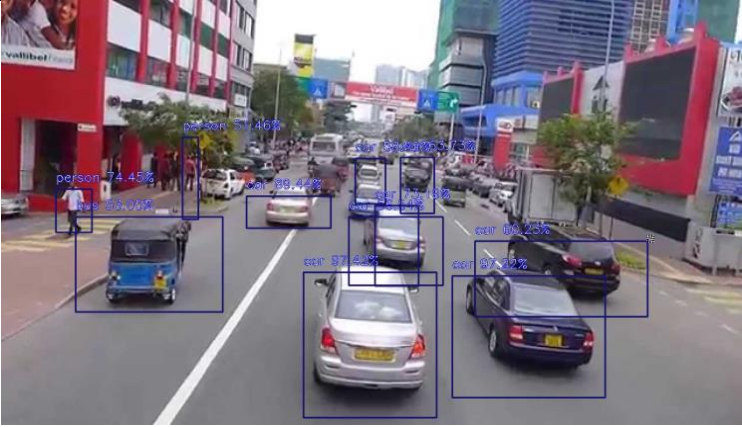


Python- ImageAI

YOLO - yolo-tiny.h5. 1:

car : 56.54352307319641

```
bus : 53.0504584312439
car : 60.22621989250183
car : 97.22371101379395
car : 97.42370843887329
person : 51.46095156669617
car : 58.49268436431885
car : 63.7304425239563
person : 74.44759607315063
car : 89.44361805915833
car : 73.18670153617859
```



-
-

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

2.

1.

		*)
1		
2		

(*)

2.

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

1.

		*)
1		
2		

2.

1.

2.

3.

4.

5.

ROC-

6.

7.

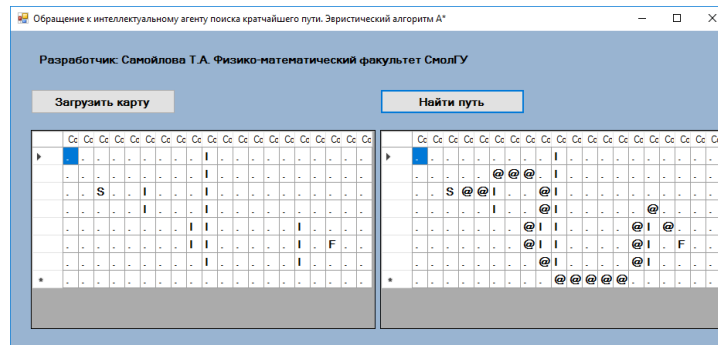
8.

9.

- 10.
- 11.
- 12.
- 13.
- 14.
- 15.
- 16.
- 17.
- 18.
- 19.

(k-

- 1.
- 2.



1.

1		
2		
3		

(*)

2.

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

1.

/

2.

- 3. ISBN 978-5-534-07818-3. URL: <https://urait.ru/bcode/494506>
- 4. ISBN 978-5-534-11361-7. URL: <https://urait.ru/bcode/494434>
- 5. ISBN 978-5-534-07467-3. URL : <https://urait.ru/bcode/490657>
- 6. ISBN 978-5-9916-8250-3. URL: <https://urait.ru/bcode/490259>
- 7. ISBN 978-5-9916-8251-0. URL: <https://urait.ru/bcode/471000>
- 8. ISBN 978-5-534-14916-6. URL: <https://urait.ru/bcode/485440>
- 9. ISBN 978-5-534-07819-0. URL : <https://urait.ru/bcode/494505>
- 10. ISBN 978-5-534-00551-6. URL : <https://urait.ru/bcode/492094>
- 11. ISBN 978-5-534-00734-3. URL: <https://urait.ru/bcode/490386>
- 12. ISBN 978-5-534-15561-7. URL: <https://urait.ru/bcode/508804>
- 13. ISBN 978-5-534-13619-7. URL: <https://urait.ru/bcode/497448>
- ISBN 978-5-534-11659-5. URL: <https://urait.ru/bcode/495988>

- 1. -Hill, 1997. 414 p.
- 2. URL: <http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf>
- 3.
- 4.

- 7.
- 8.

7.3.

1. moodle.smolgu.ru).
2. intuit.ru).
3. opened.ru).
4.).
5. (http://window.edu.ru).
6. (http:// http://msdn.microsoft.com).

8

224 12

9.

- 1.
- 2.
3. Python 3.9
4. PyCharm Pro
- 5.

9

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

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