

# Development of a machine learning software system for customer churn predictions

Student: Moskalev D.I.

Supervisor: Trakimus Y.V., Ph.D., Associate Professor, Department of PM

# OBJECTIVES AND WORK TASKS

## Target:

- Creating models for predicting customer churn in the banking sector using classification methods
- Comparative analysis of the accuracy of results using visualization of quality metrics

## Tasks:

- Searching and processing large number of data (Big Data)
- Determining possible customer loyalty in the analysis of input data
- Algorithm adaptation and modeling
- Comparative analysis of results
- Interpretation of results using visual metrics representation of output data

# WHY DO YOU NEED PREDICTIONS?

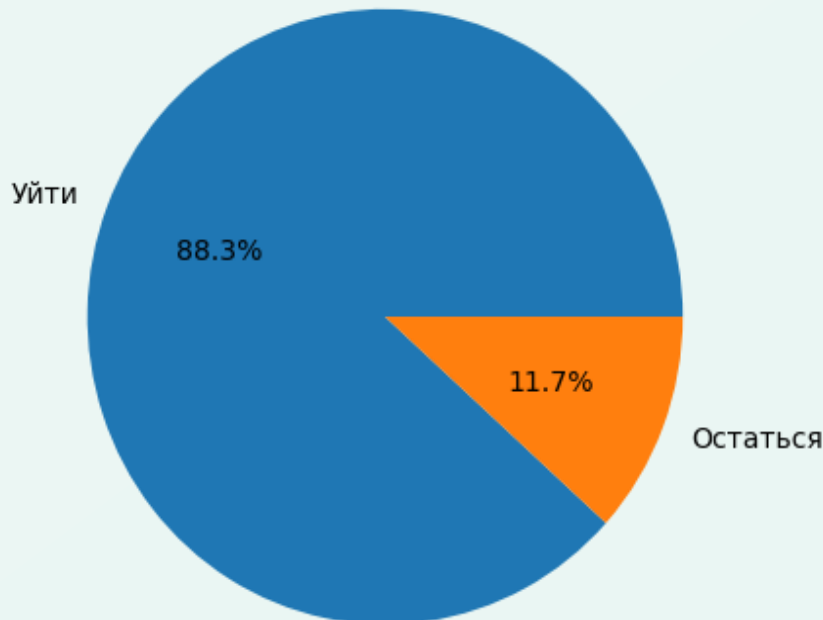
The customer churn prediction allows:

- Analyze the current state of the organization
- Receive recommendations regarding customer segmentation, banking products

## The ratio of clients in banks

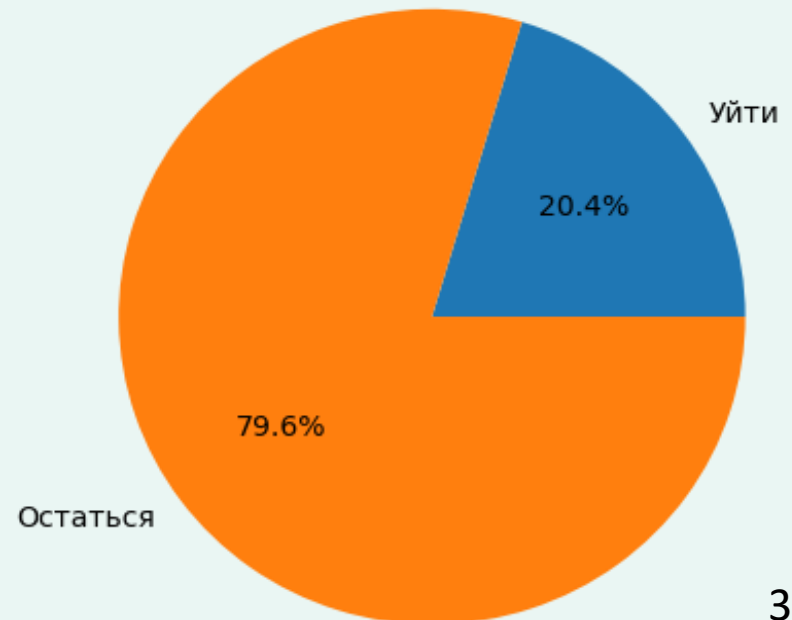
Bank A

Сколько клиентов хотят уйти?



Bank B

Сколько клиентов хотят уйти?



# TASK SETTINGS

- The formal setting of the classification task represents an unknown target dependency

$$y^*: X \rightarrow Y, \quad (1)$$

where  $X$  – many object descriptions,  $Y$  – finite number of class numbers.

The display values (1) are only known in the teaching sample objects:

$$X^n = \{(x_1, y_1), \dots, (x_n, y_n)\},$$

where  $n$  – number of rows of objects.

In a binary classification task, a number of class numbers  $Y = \{f_1, f_2\}$ .

Usually  $f_1 = 0, f_2 = 1$ .

# SELECTED METHODS OF SOLUTION

All selected methods are based on decision trees. Trees are a set of nodes that can be divided into two types:

- Decision nodes - signs on which the tree is built.
- Probabilistic (closing) nodes are leaves of trees in which subtotals or final values of characters are calculated.

**1. Boosting** – technology consistent building composition of machine learning algorithms, where each subsequent algorithm tries to compensate for the shortcomings of the composition of all previous algorithms.

Methods: XGBoost (Extreme Gradient Boosting), CatBoost (Categorical Boosting).

**2. Bagging** – classification technology, where all elementary classifiers are calculated in parallel before building decision trees.

Method: Random Forest.

# INPUT DATA

The input data is given by a matrix  $N \times M$ , where  $N$  – bank clients,  $M$  – their indicative description.

Bank A: 45211 clients.

age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no

Bank B: 10000 clients.

CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Characters are numeric and categorical data. The target mark for bank A is column "y", for bank B - "Exited".

# CHARACTER LAYOUT

The character layout shows the proportion of each unique hint value in the datasets and is a visual indicator of balance.

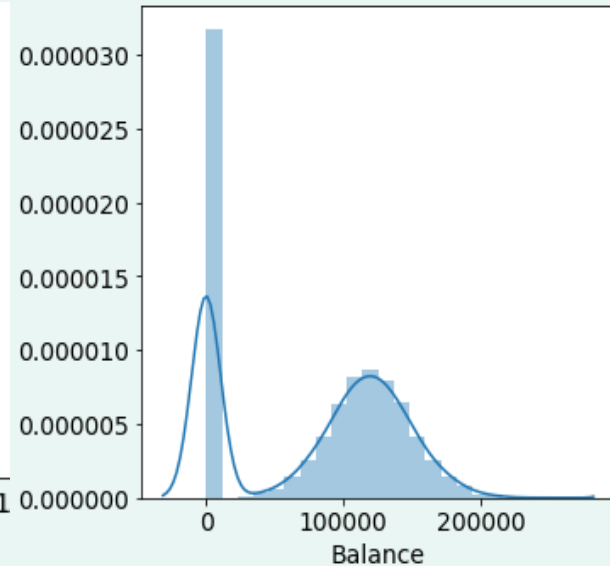
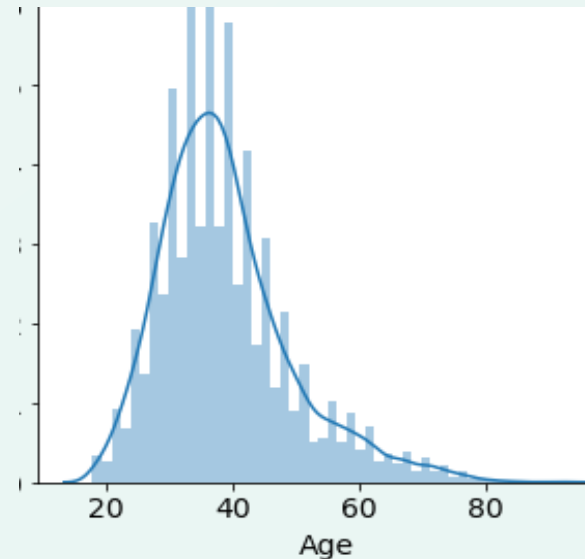
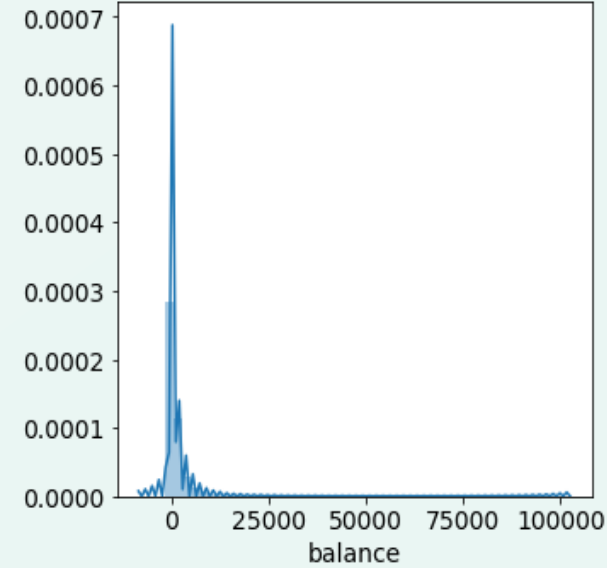
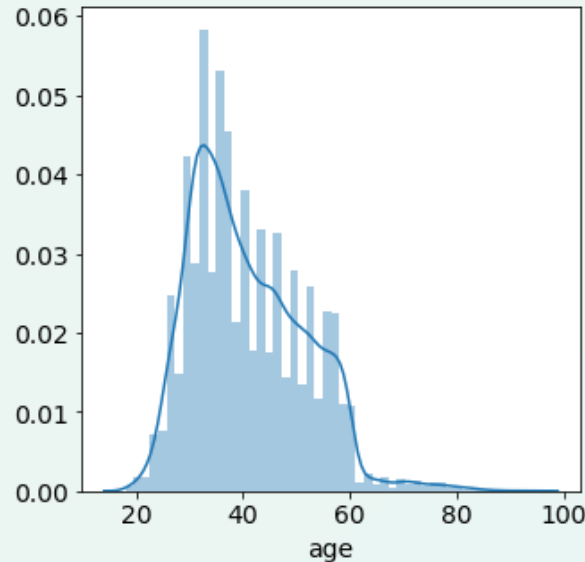
In axis **OX** the character is deferred, in the axis **OY** is the distribution of characters in the sample.

$$\sum_{i=1}^N Feature_p = 1, p \in \overline{1, M}.$$

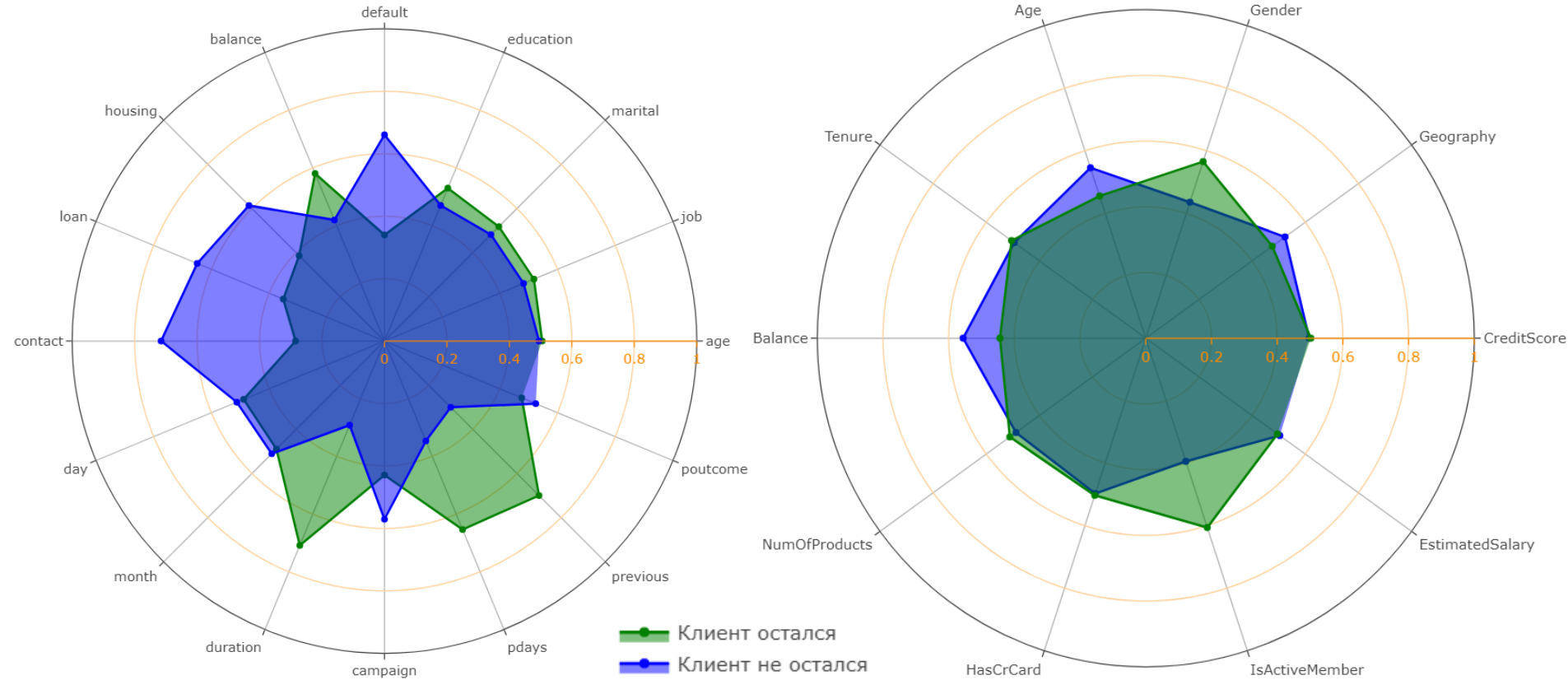
Features:

**Age** – client's age,

**Balance** – client's balance.



# AVERAGE FEATURE VALUES



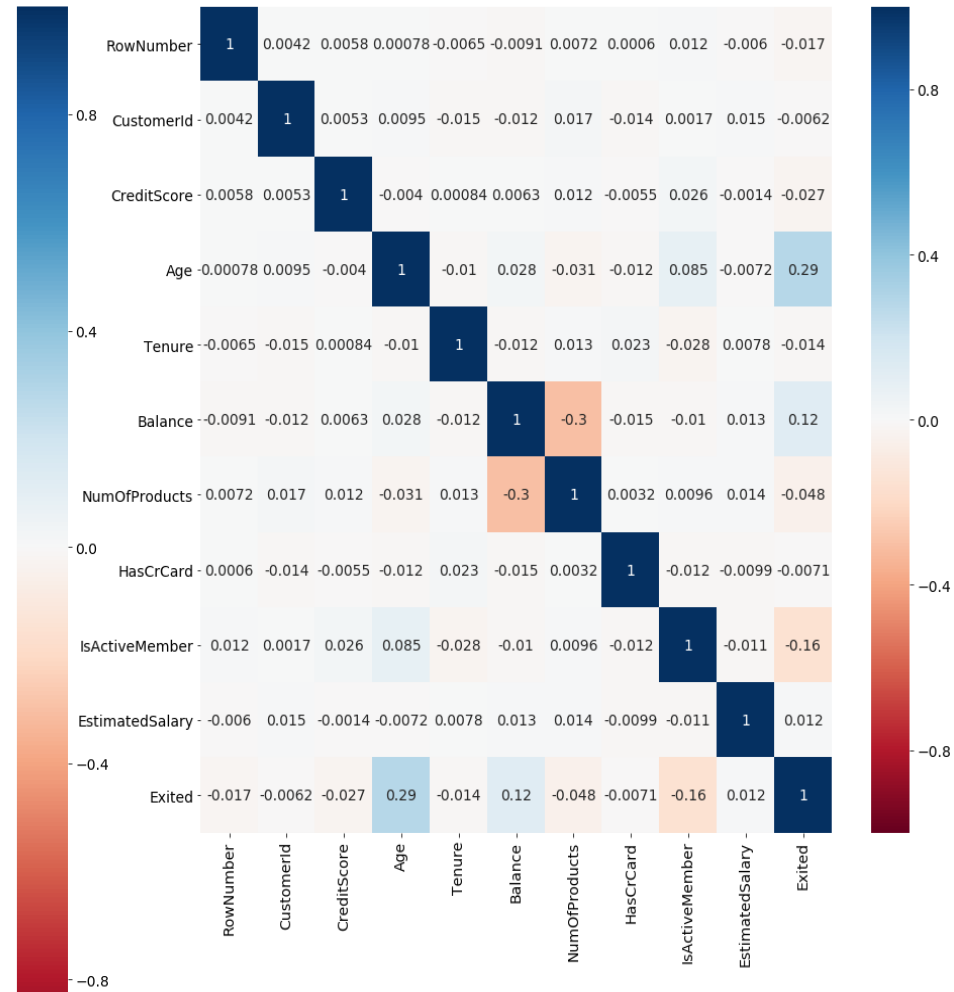
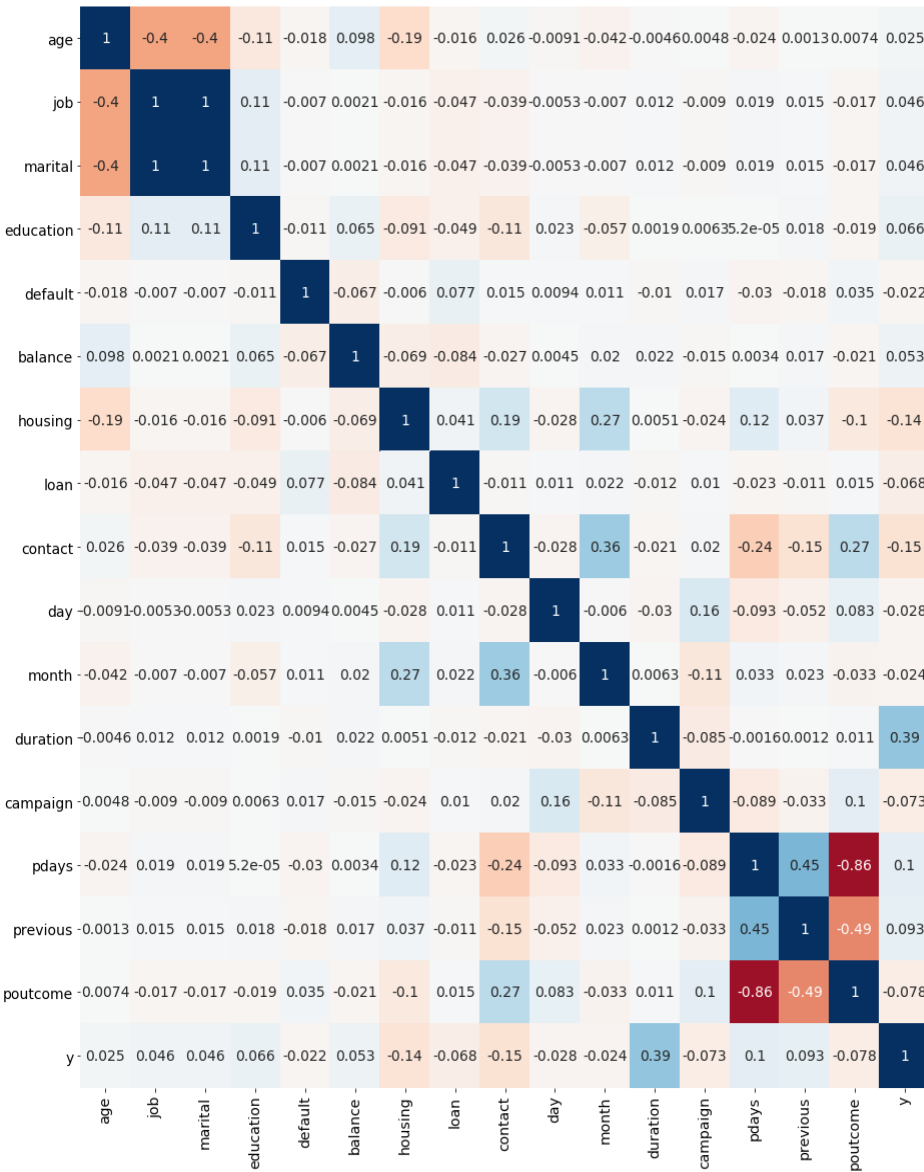
Feature values can be calculated according to the formula:

$$Feature_{mean_p} = \frac{Feature_{mean_i}}{Feature_{mean_0} + Feature_{mean_1}}, i = \{0,1\}, p = \overline{1, M}.$$

The graphs show that it differs more in bank A and it is easier to choose it's threshold.

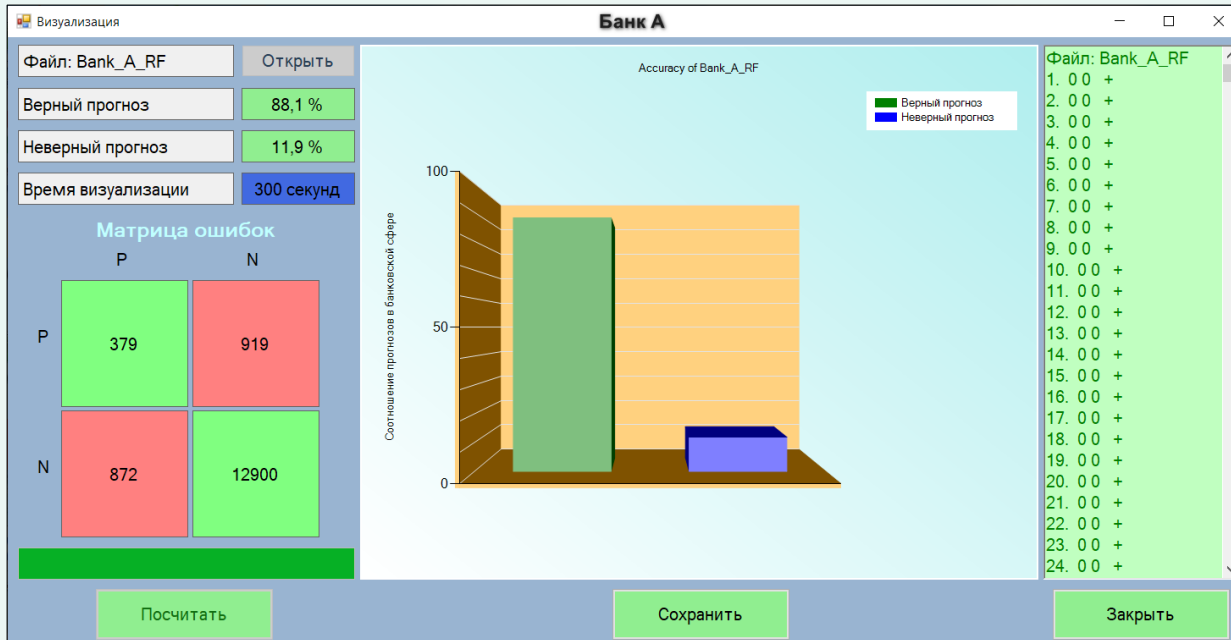


# CORRELATION MATRICES

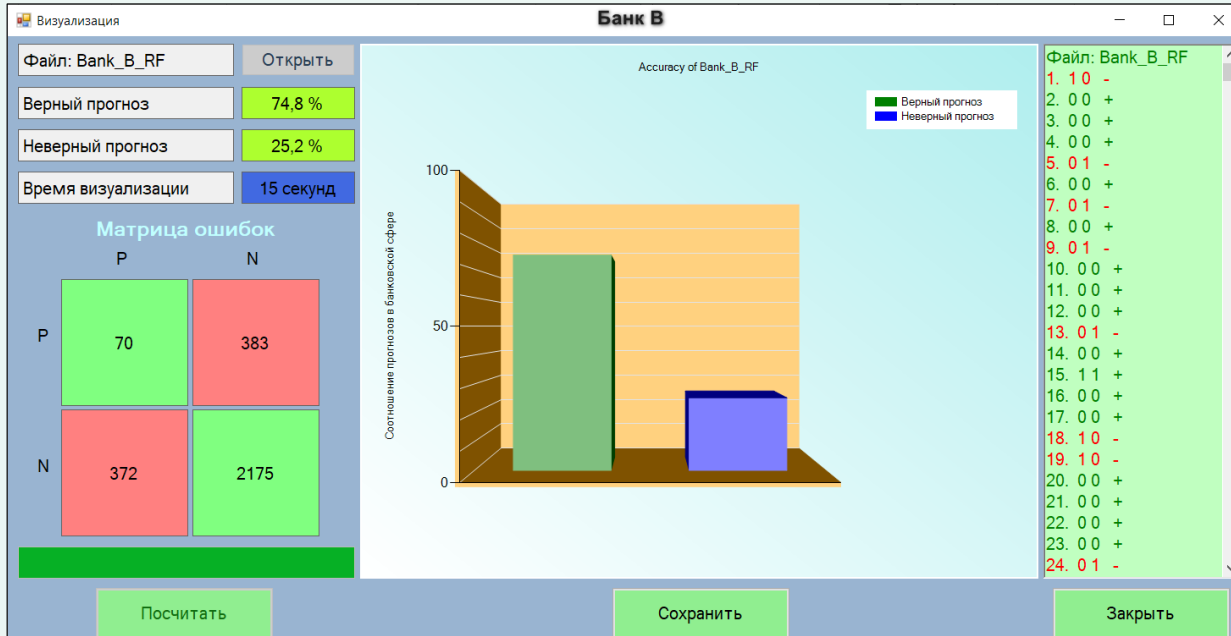


Blue color means strict direct relation, white - no relation, red - strict reverse relation.

# VISUALIZATION

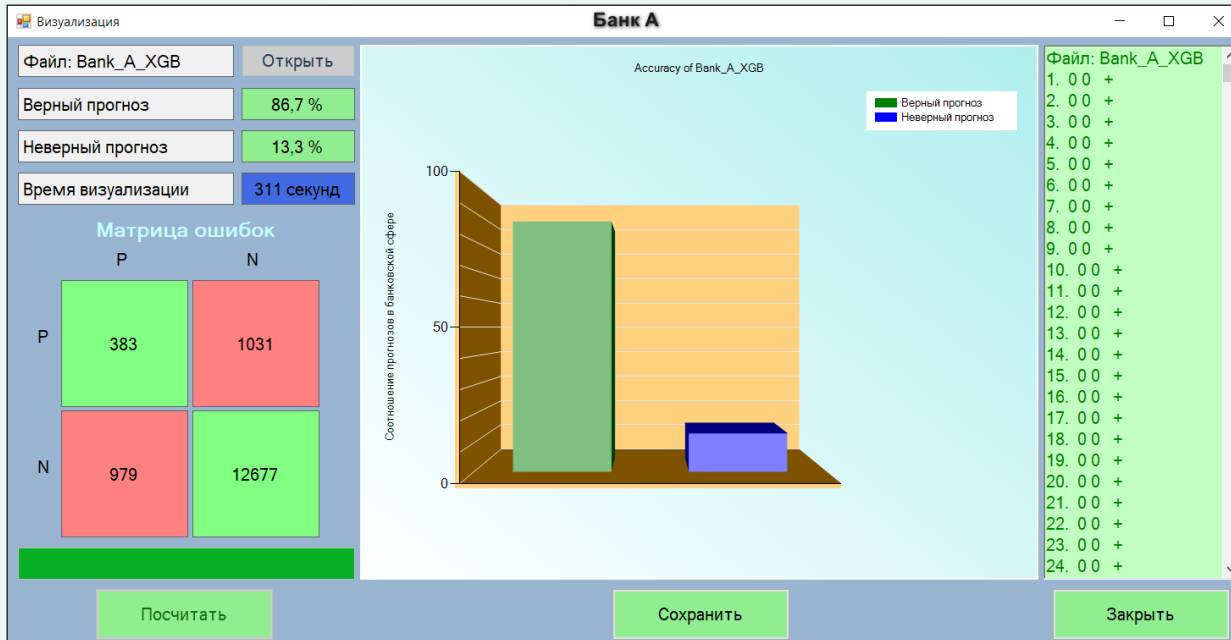


Visualization results of models trained by the Random Forest method. Accuracy metric value (correct prediction): Bank A: 88,1%, Bank B: 74,8%.

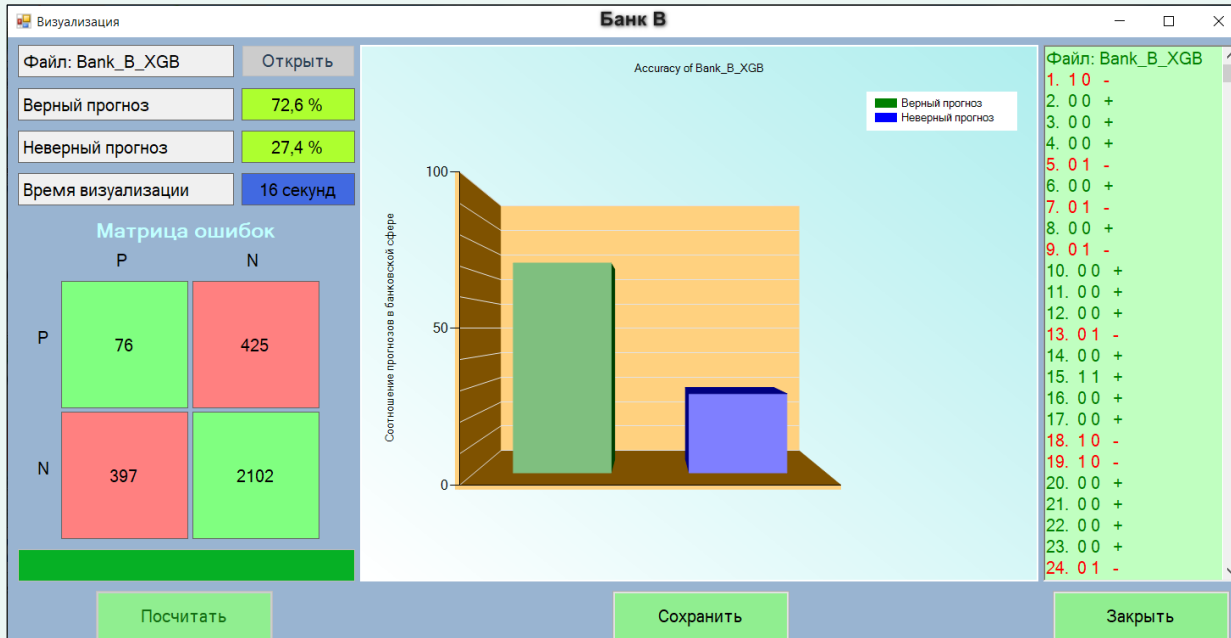


$$\text{Accuracy} = \frac{TP+TN}{P+N} \cdot 100\% .$$

# VISUALIZATION

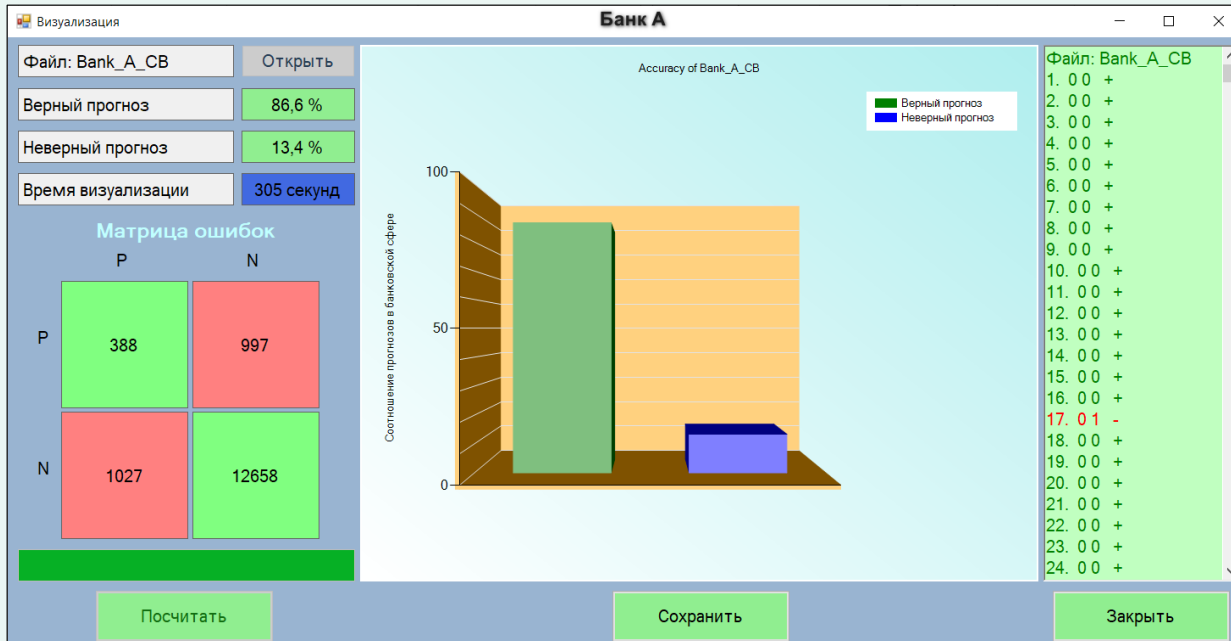


Visualization results of models trained by the XGBoost method. Accuracy metric value (correct prediction): Bank A: 86,7%, Bank B: 72,6%.

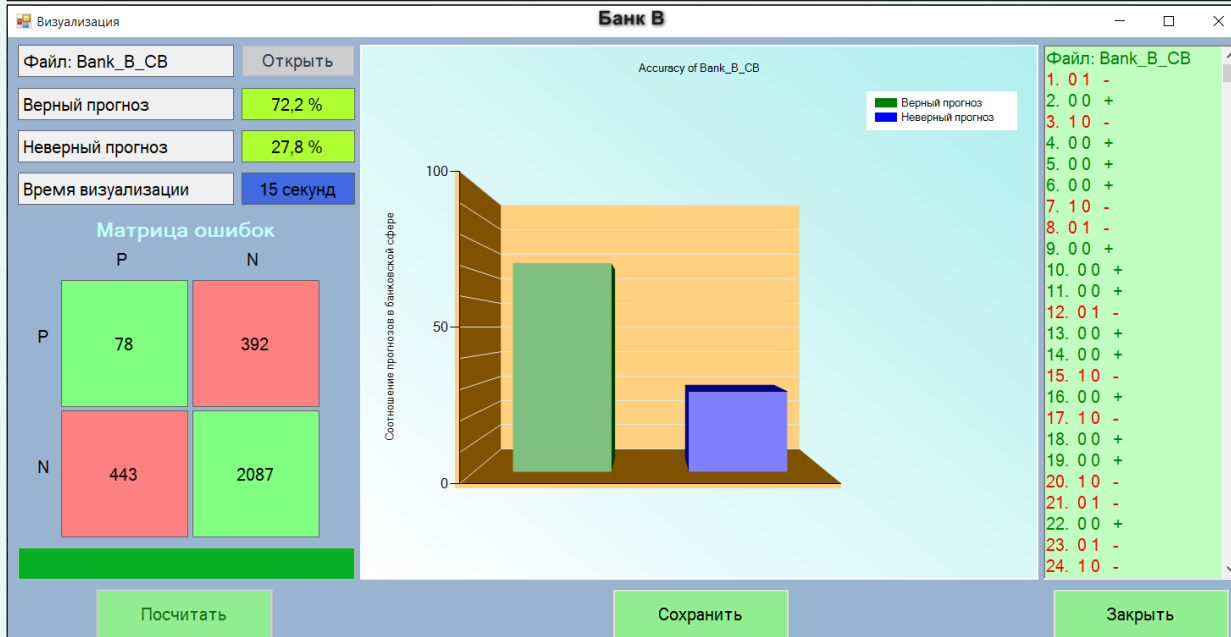


$$\text{Accuracy} = \frac{TP+TN}{P+N} \cdot 100\% .$$

# VISUALIZATION



Visualization results of models trained by the CatBoost method. Accuracy metric value (correct prediction): Bank A: 86,6%, Bank B: 72,2%.



$$\text{Accuracy} = \frac{TP+TN}{P+N} \cdot 100\%.$$

# CONCLUSION

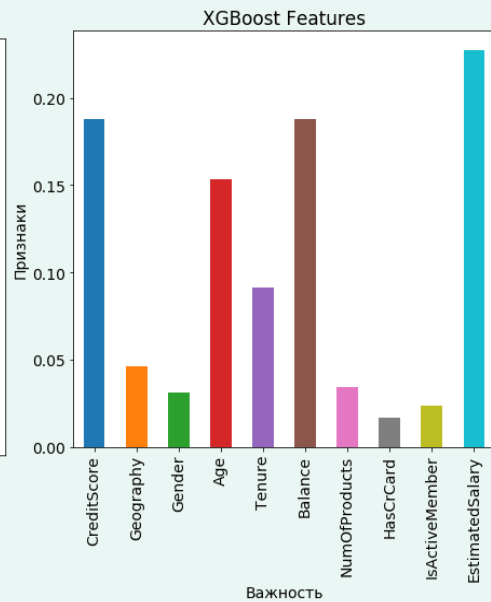
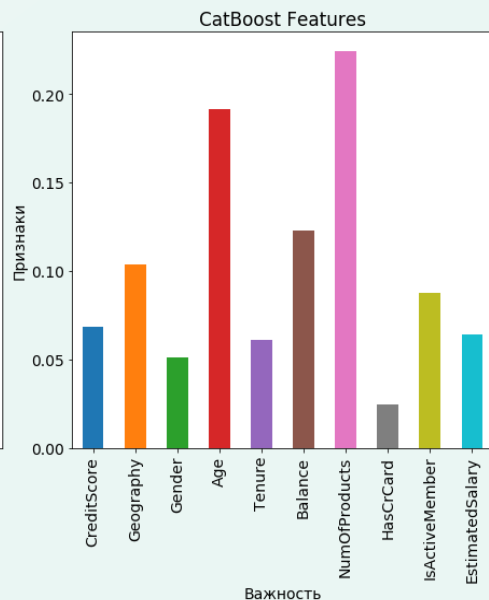
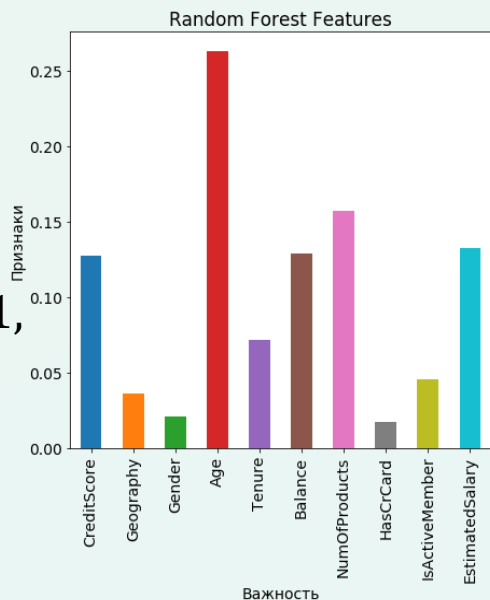
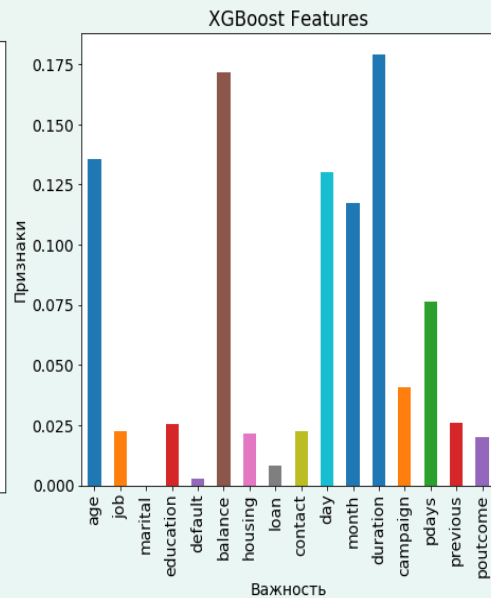
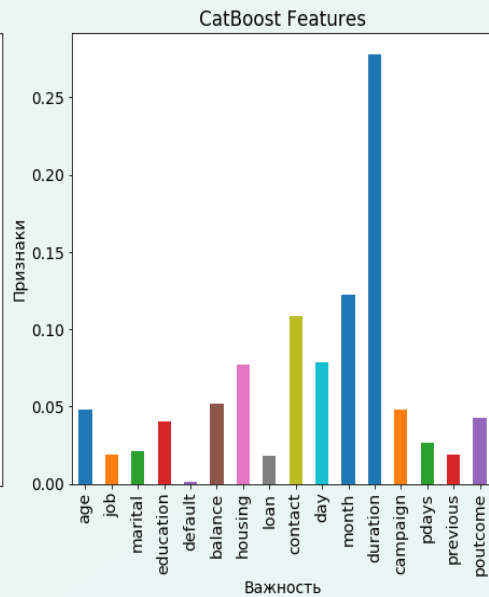
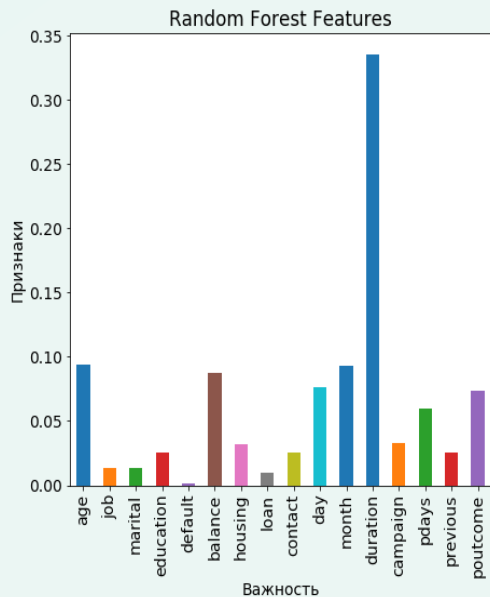
- A comparative analysis of the methods show that all classifiers have good accuracy in the range of 72.2% to 88.1%, which proves their effectiveness in solving the problem of machine learning classification. The best results on the test sample were shown by the Random Forest method (88.1%), in second place - XGBoost and in third place - the CatBoost classifier
- The running time of the Random Forest method proved to be the shortest due to the possibility of building solution trees in parallel
- The Random Forest method can be used in all areas of customer outflow forecasting

# CHARACTER MEANING GRAPHS

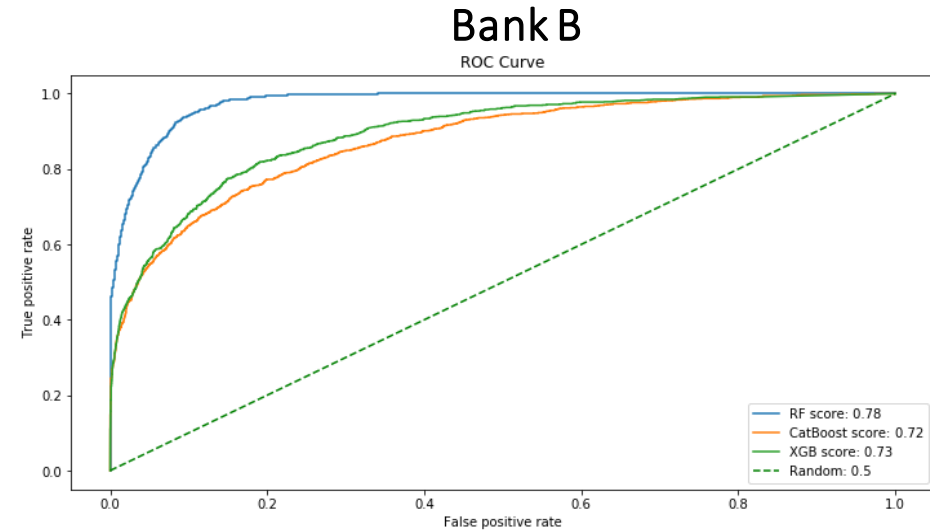
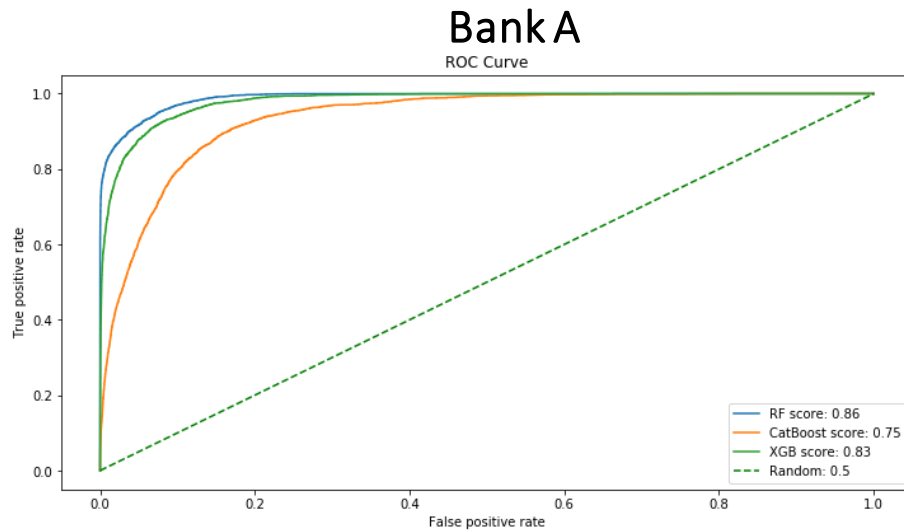
The graph of the meaning of the characters shows their influence on the construction of the model.

In axis **OX** features appear, along axis **OY** – the probability value of the feature.

$$\sum_{i=1}^N feature_p = 1, \text{ where } p \in \overline{1, M}.$$



# METRICS



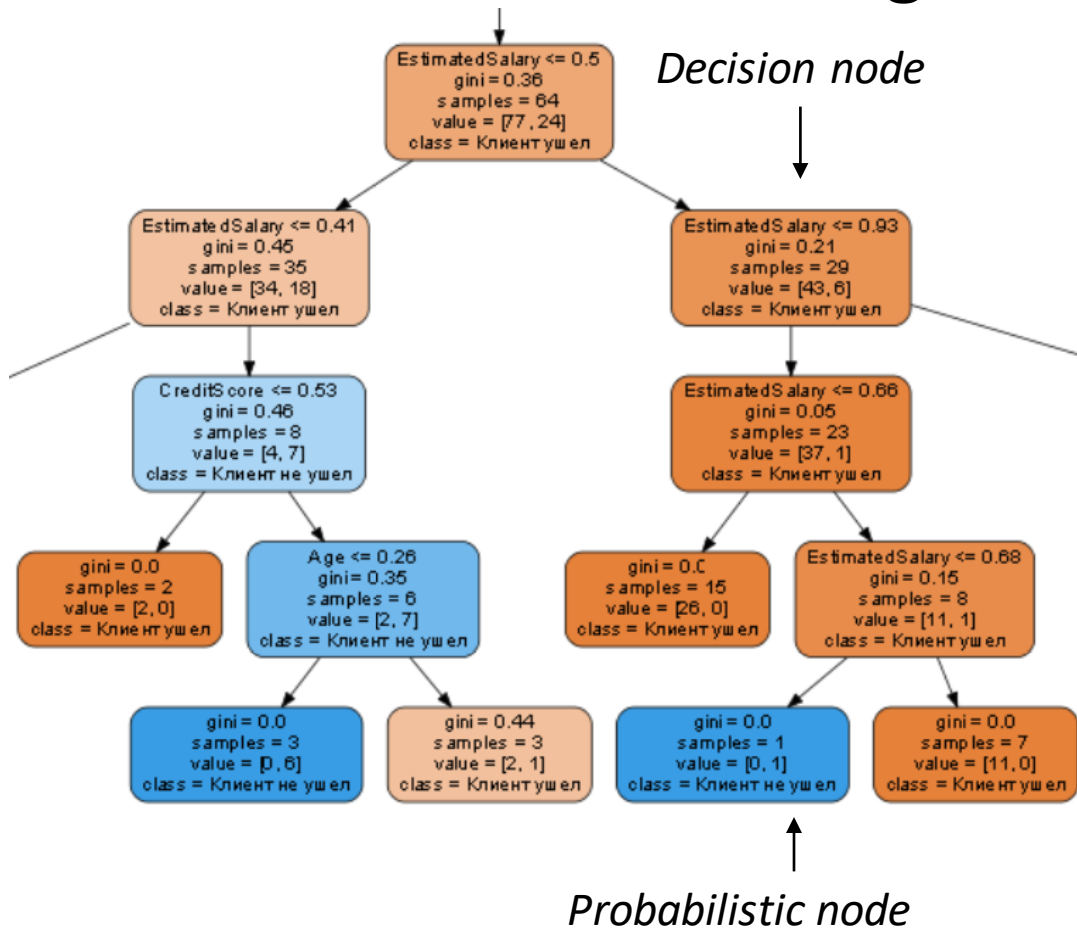
$$FPR = \frac{FP}{N}, TPR = \frac{TP}{P}$$

The ROC-curve construction algorithm is built on the grid points  $m \times n$ , where  $m$  – number of “1”,  $n$  – number of “0” on the following conditions. If the values of the target attributes are matched (true prediction), then there will be a shift up one division of the test sample, if the values of the target attributes are mismatched (false prediction), then there will be a shift to the right of one division. The overall execution of  $m$  steps up and  $n$  steps on the right will allow you to come to point (1,1).

Target:  $S_{max} \rightarrow 1$ .

Smoother lines on the bank chart are caused by the test sample having a large size matrix for constructing AUC ROC metrics.

# Solution tree fragment visualization

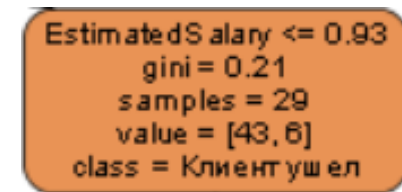


The purpose of building trees is to minimize the criterion of probability of incorrect classification:

$$Gini = 1 - \sum_{j=1}^j k_j^2,$$

where  $k_j$  – probability of  $j$  class.

There are two classes in binary classification =>  $j = 2$ .



Example of Gini calculation for solution node "EstimatedSalary <= 0.93":

$$Gini = 1 - (k_1^2 + k_2^2) = 1 - \left( \left( \frac{43}{49} \right)^2 + \left( \frac{6}{49} \right)^2 \right) = 1 - \frac{1849 + 36}{2401} \sim 1 - 0,785 \sim 0,21.$$



# Total building and visualization time

Total working time

