FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Elena Myazina

Machine Learning for Credit Scoring

Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the master thesis: Mgr. Martin Pilat, Ph.D.

Study programme: Software systems

Specialization: Informatics

Prague 2017

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague 19.07.2017 signature

I would like to thank my supervisor, Mgr. Martin Pilat, Ph.D. for his patience and for his valuable expert advices. Moreover, I would like to thank my parents and my friends for supporting me during my studies.

Title: Machine Learning for Credit Scoring

Author: Elena Myazina

Department / Institute: Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the master thesis: Mgr. Martin Pilát, Ph.D, Department of Theoretical Computer Science and Mathematical Logic

Abstract: Credit scoring is a technique used by banks to evaluate their clients who ask for different types of loan. Its goal is to predict, whether a given client will pay their loan or not. Traditionally, mathematical models based on logistic regression are used for this task. In this thesis, we approach the problem of credit scoring from a machine learning point of view. We investigate several machine learning methods (including neural networks, random forests, support vector machines and other), and evaluate their performance for the credit scoring task on three publicly available datasets..

Keywords: machine learning, credit scoring,logistic regression, neural networks, random forest

Contents

Introduction

Credit scoring [13] is a client's creditworthiness assessment (credit risks), based on numerical statistical methods - a well-known method of applying mathematical methods in a banking field.
For increasing the profitability of credit operations, a bank has to qualitatively assess credit risks. Basing on clients' classification into risk groups, a bank decides whether to issue a loan to a client or not, and what lending limit and interest should be set. The task of classifying clients into risk groups is solved by a scoring system. The main task of scoring is not only in finding out whether the client is able to pay the loan or not, but also in finding out a degree of reliability and compulsion of the client. In other words, scoring estimates how much the customer is "worthy" of the loan.
The scoring model uses a set of certain characteristics. The result is real valued score; the higher it is, the higher the client's reliability, and the bank can streamline its customers by the degree of credit enhancement. The indicator of each customer is compared with a certain numerical threshold, which can be called the break-even line. The complexity of constructing a model resides in the determination of which characteristics should be included in a model.
Characteristics of loan application (factors) can be either discrete (for example, gender of a borrower, level of education) or continuous (for example, borrower's age, work experience, income, expenses, loan amount).
Credit scoring systems play an important role in banks, since such systems allow for the reducing of costs and minimizing of operational risk by automated decision-making, reducing the processing time for loan applications, enabling banks to pursue their credit policy centrally and providing additional protection for financial organizations from fraud.

There are different types of credit scoring [14]:

1. Application-scoring: determines the level of credit risk of a potential borrower based on the data available at the time of filing application.

2. Fraud-scoring: estimates the possibility of fraud from the potential borrower. Quite often swindlers try to create an image of the ideal borrower. Counting fraud is designed to figh them.

3. Behavioral-scoring: a type of scoring which allows banks to estimate the financial behavior of the borrower and, thereby, to predict his/her behavior (existence / lack of delays, future profitability, probability of acquisition of other banking products) in the course of service of the credit.

4. Collection-scoring: this type of scoring is carried out when the client has delays on the credit. Collection-scoring establishes what should be made in relation to a debtor: to be limited to a reminder by phone or at once to submit the case to collectors.

In this work application scoring is considered.

Machine Learning [1] is a field of artificial intelligence that studies the methods of building models that can be trained, as well as algorithms for their construction and learning.
Machine learning can be of different types. The most common case is supervised learning. Both attributes and target are given for each instance of data. In unsupervised learning, only the attributes are given and the goal is often to cluster similar instances together.
Machine learning methods [3] are used for constructing a scoring model and for solving classification problems.
For example, neural networks, nearest-neighbor methods, multiple linear and logistic regressions, the support vector machine, trees and forests.
The problem of classification is a problem of discriminant analysis, in which there are many objects, separated by a certain feature into some classes. Herewith, there is a set of indicators of typical class representatives, called a training sample. On the basis of this sample, it is necessary to build a discrimination rule that allows us to recognize to which class a particular new object not included in the training sample belongs.
Such methods are relevant for economic researches, for researches in the field of medicine, psychology and political science, but recently such classification tasks are most actively solved in the banking field when solving the problem of granting loans, that is, in a credit scoring.
The purpose of this thesis is to build a mathematical model for assessing the solvency of the borrower by means of machine learning methods.
Tasks:

1. Overview of the topic
2. Overview of machine learning methods
3. Creation of scoring models
4. Evaluation of the quality of the constructed models

In our work we have used three data sources. As the main dataset we used the data of Tinkoff Bank[19]. And for checking the results of construction scoring models we used the data of the Kaggle portal [17] and LendingClub (Lending Club Statistics)[18].

Chapter 1

Related work

This chapter shows the basic concepts of the systems that estimate the risks of the bank, on the basis of which it makes a conclusion about the expediency of the bank to give credits to individuals.
Modern banks use two approaches to credit risks estimation:
• based on the opinion of the experts;
• with the help of a credit scoring system.
To estimate the credit solvency of individuals, the credit scoring approach is mainly used. Scoring estimates not only the probability of credit redemption, but also a client's reliability.

In the first paper [13], the techniques used for credit scoring are summarized and classified and the new method ensemble learning model is introduced.
It credit scores methods into statics models, AI models, hybrid methods and ensemble methods. Ensembled learning has been widely applied to personal credit evolution and has better classification ability and prediction accuracy.

Statistical models includes LDA (Linear Discriminant Analysis), MARS, and Decision Tree. AI methods include ANN SVM, and K-Nearest method. This paper also discusses about behavioral scoring methods. Behavioral scoring makes a decision about the management of credit based on the repayment performance of existing customers during a certain predefined period of time. It also includes repayment behavior and the payment history of the client. According to this paper, ensemble learning has better prediction accuracy and classification ability and is thus widely applied to personal credit evolution.

In the second paper [15] deals with the design aspects related to financial fraud detection. The aim of feature selection is to improve both the actual and computational performance of the solution, as well as providing a better understanding of the problem.
The dataset used in this research is a synthesized credit card dataset used for the 2009 UCSD-FICO data mining contest. It consists of entirely numerical data with 334 input attributes and 10,000 records. The dataset was first preprocessed with a Bayesian discretizer to convert each attribute to discrete values before conducting the main experiment.
In this example the authors have compared- the performances of genetic programming (GP), genetic algorithms (GA), ant colony optimizations (ACO), neural networks (NN), support vector machines (SVM), fuzzy logic (FL), decision trees (DT), functions (Fn), lazy evaluators (Lazy), and rule-based classifiers (Rule).

Additionally, we have investigated the affects that attribute selection had on fraud detection using seven feature selection methods (FS1-7). To improve reliability, the authors made use of 10-fold cross validation to reduce the chance of statistical errors.
Feature ranking algorithms assign ratings to individual features based on certain attributes such as accuracy, content and consistency and choose a suitable subset on the basis of ranking. In

classification method accuracy, sensitivity, specificity, precision, and false positive rate are the performance measures.
It has been found that, if misclassification costs are high, techniques with higher sensitivity, such as GP1 [15] or neural networks, may be suitable choices. If receptiveness to minor chances in dataset is desired, then the ant colony optimization or neural networks could be appropriate. Overall the support vector machine could be considered to have the best performance with the highest accuracy.

In the third paper [16] the authors have considered a methodology called a smart ubiquitous data mining (UDM) that consolidates homogeneous models in a smart ubiquitous computing environment. It tests the suggested model with financial datasets, then basically induces rules from the dataset using diverse rule extraction algorithms and combines the rules to build a metamodel. This paper builds several personal credit rating prediction models based on the UDM and benchmarks their performance against other models which employ logistic regression (LR), a Bayesian style frequency matrix (BFM), multilayer perceptron (MLP), classification tree methods (C5.0,) and neural network rule extraction (NR) algorithms [16].
The raw data used in this experiment comprises personal credit data, personal loan applications data, and personal loan data acquired from FHC, Korea's leading financial holding institution, with a 30-million-strong customer base and business network exceeding 1,200 branches, the largest in Korea.
To prove the efficiency of the suggested model using a 5-fold cross validation process. Multiple tests with 5 different training and validating sets confirm the efficiency of the model from UDM.
The result shows that the performance of UDM is superior to that of other models such as LR, NN, BFM, C5.0, and NR.

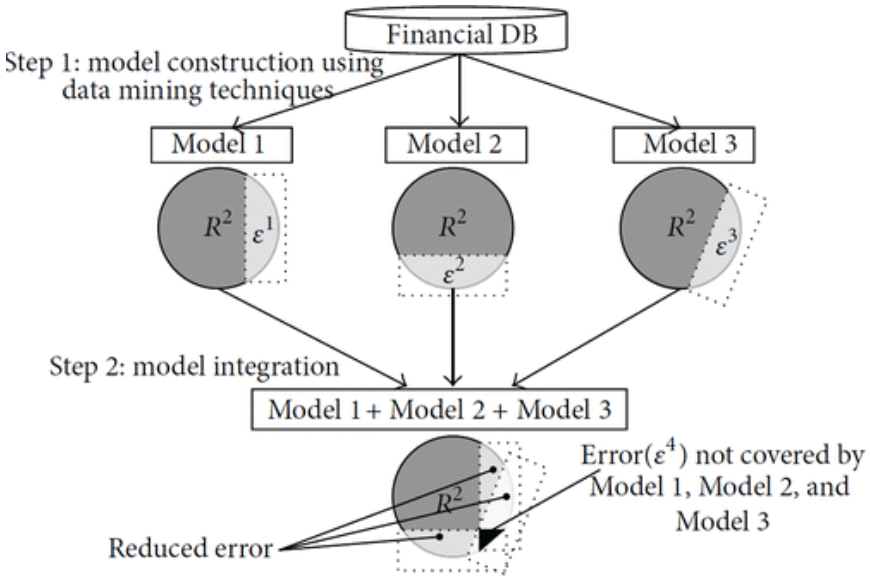


Figure: 1.1 The pictorial model of UDM [16].

Chapter 2

Data analysis

2.1 Data description and problem definition

In our project we used data provided by one of the largest online banks "Tinkoff Bank". Tinkoff Bank , formerly Tinkoff Credit Systems, is a Russian commercial bank based in Moscow and founded in 2006. The bank does not have branches or ATMs (automated teller machines). As of 2016, Tinkoff Bank has a credit rating of B+ on the Fitch Ratings and B2 on the Moody's Rating, and is the second largest provider of credit cards in Russia.
The bank specializes in the provision of unsecured loans and has issued more than 3,5 million cards. Net percentage margins following the results of 2014 are at 34, 8% - one of the highest rates in the banking system.
At the same time the bank gradually departs from the initial model of the monoline credit, extending the range of services, thereby increasing the share of non-interest incomes.
The bank widely uses systems of automation for improving business processes, including voice identification for protection against swindlers and acceleration of work of call center, and also the technology of handling of big data, including data from social networks, for the forecasting of risks with return of debt.

The bank requests an applicant's credit history from the three largest Russian credit bureaus.

The sample of bank clients is provided in the file SAMPLE_CUSTOMERS.CSV [19], which is divided into the following parts: "train" and "test". According to the sample "train" the goal variable value "bad", i.e. existence of "default" (allowing a client the delay of 90 and more days within the first year of using the credit), is known.
The data format **SAMPLE_CUSTOMERS** is information on the default possibility of a certain person. The data with reference to the response of credit bureaus to all queries on the corresponding clients are provided in the file **SAMPLE_ACCOUNTS** (CSV). The data format with reference to the response of bureaus is the information on a person's accounts transferred by other banks to this bureau. The data format is described in detail in file **SAMPLE_CUSTOMERS.CSV**.
The task is to develop the model defining a probability of "default" based on the sample "train", and to set its probabilities according to clients out of the sample "test". The characteristic **Area Under ROC Curve**[1] will be used for the model evaluation.

[1] The ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

Data description

Description of a format of a data set **SAMPLE_ACCOUNTS**:

Name	Description
TCS_CUSTOMER_ID	Customer ID
BUREAU_CD	Code of bureau from which the account is received
BKI_REQUEST_DATE	Date in which the request in bureau has been made
CURRENCY	Contract currency (ISO)
RELATIONSHIP	1 — Physical person 2 — The additional card / the Authorized user 4 — Joint 5 — Guarantor 9 — Legal entity
OPEN_DATE	Date of opening of the contract
FINAL_PMT_DATE	Date of final payment (planned)
TYPE	Code type of contract 1 – Credit for the car 4 – Leasing 6 – Mortgage 7 – Credit card 9 – Consumer credit 10 – The credit for business development 11 – Credit for working capital 12 – Credit for the purchase of equipment 13 – Credit for construction of real estate 14 – Credit for the purchase of shares 99 – Other
PMT_STRING_84M	Discipline (timeliness) of payments. The line is formed from codes of conditions of the account for the transfer moments a databank on the account in bureau, the first symbol — a state for date PMT_STRING_START, further consistently in decreasing order of dates. 0 – New, the assessment is impossible X – There is no information 1 – Payment without delays A – Delay from 1 to 29 days 2 – Delay from 30 to 59 days 3 – Delay from 60 to 89 days 4 – Delay from 90 to 119 days 5 – Delay more than 120 days 7 – The regular consolidated payments 8 – Repayment on the credit with pledge use 9 – The hopeless debt / is given to collecting / the missed payment

STATUS	Status of the contract 00 – Active 12 – It is paid due to providing 13 – The account is closed 14 – It is transferred to service to other bank 21 – Dispute 52 – Overdue 61 – Problems with return
OUTSTANDING	The remained outstanding debt
NEXT_PMT	Amount of the following payment
INF_CONFIRM_DATE	Date of confirmation of information on the account
FACT_CLOSE_DATE	Date of closing of the account (actual)
TTL_DELQ_5	The amount of overdue till 5 days
TTL_DELQ_5_29	The amount of overdue 5 to 29 days
TTL_DELQ_30_59	The amount of overdue 30 to 59 days
TTL_DELQ_60_89	The amount of overdue 30 to 59 days
TTL_DELQ_30	The amount of overdue up to 30 days
TTL_DELQ_90_PLUS	Number of delays of 90+ days
PMT_FREQ	Code of frequency of payments 1 – Weekly 2 – Every two weeks 3 – Monthly A — Every 2 months 4 – Quarterly B — Every 4 months 5 – Semiannually 6 — Annually 7 – Another
CREDIT_LIMIT	Credit limit
DELQ_BALANCE	Current arrears
MAX_DELQ_BALANCE	Maximum volume of arrears
CURRENT_DELQ	Current number of days of delay
PMT_STRING_START	Start date of a line PMT_STRING_84M
INTEREST_RATE	Loan interest rate
CURR_BALANCE_AMT	The total paid amount, including the sum of a principal debt, percent, penalty fee and penalties.

Table 1 Description of the data set

2.2 Preliminary analysis

We will use the feature “STATUS“, to look at the statistics of contracts.

We will use the following parameters:

TCS_CUSTOMER_ID	Customer ID
TYPE	Code type of contract
STATUS	Status of the contract 00 – Active 12 – It is paid due to providing 13 – The account is closed 14 – It is transferred to service to other bank 21 – Dispute 52 – Overdue 61 – Problems with return

Table 2 Statistics of contracts. Realization in R using "boxplot"

Status	Active (0)	The account is closed (13)	It is transferred to service to other bank (14)	Overdue (52)	Other (12,21,61)
Client (%)	46	52	0.8	7	0,3

Table 3 Statistics of STATUS value. Realization in R

We see that the database consists of the active and closed contracts. Quantity of "Overdue" is not big, on it we can assume that at us there isn't a lot of debtors (7 %).
We will try to analyze TYPE values. We will look at what credits are used most often:

Code type of contract	**Count**
1(Credit for the car)	4945
4 (Leasing)	2
6(Mortgage)	1691
7(Credit card)	49407
9(Consumer credit)	196707
10(The credit for business development)	900
11(Credit for working capital)	120
12(– Credit for the purchase of equipment)	78
13(Credit for construction of real estate	65
14(Credit for construction of real estate)	4
99(Other)	27023

We see that 7,9,1, 99 is most often used. We will unite in one column value those types of the credits which are not used often (4, 12,13,14). We will unite these values in one column: **type_463**.

Analysis of clients with a good credit history
For this purpose we will find those clients who have a good credit history. We will determine parameters which will be necessary - for creation of the table of clients with good credit history.

We will use the following parameters:

TCS_CUSTOMER_ID	Customer ID
TYPE	Code type of contract
STATUS	Status of the contract 00 – Active 12 – It is paid due to providing 13 – The account is closed 14 – It is transferred to service to other bank 21 – Dispute 52 – Overdue 61 – Problems with return
OUTSTANDING	The remained outstanding debt
DELQ_BALANCE	Current arrears
CURRENT_DELQ	Current number of days of delay

With the parameters we have decided. Now we need to install several filters, for obtaining more exact information. First of all it is:

- *STATUS = [0,13] (*Contracts with a good credit history*);*
- *OUTSTANDING = 0 (*There is no debt*);*
- *DELQ_BALANCE = 0 (*There is no debt*);*
- *CURRENT_DELQ = 0 (*There is no debt*);*

After all necessary transformations, we have received clients with the credits type who pay the credit in time.

Code type of contract	**Count**
1(Credit for the car)	177 (1,3%)
6(Mortgage)	47 (0,5 %)
7(Credit card)	8405(64%)
9(Consumer credit)	4002(30%)
10(The credit for business development)	2
11(Credit for working capital)	1
99(Other)	494 (4%)

Table 4 Clients with a good credit history

Analysis of clients with a bad credit history

Now we have the opposite problem. We need to find those clients who have a bad credit history. For this purpose we will determine parameters which will be necessary for us for creation of the table of clients with bad credit history.

We will use the following parameters:

TCS_CUSTOMER_ID	Customer ID
TYPE	Code type of contract
STATUS	Status of the contract 00 – Active 12 – It is paid due to providing

	13 – The account is closed 14 – It is transferred to service to other bank 21 – Dispute 52 – Overdue 61 – Problems with return
OUTSTANDING	The remained outstanding debt
DELQ_BALANCE	Current arrears
CURRENT_DELQ	Current number of days of delay

With the parameters we have decided. Now we need to install several filters, for obtaining more exact information. First of all it is:

- *STATUS = [21,61] (*Contracts with a bad credit history*);*
- *OUTSTANDING > 0 ;*
- *DELQ_BALANCE > 0;*
- *CURRENT_DELQ > 0;*

After all necessary transformations, we received statistics on clients who have overdue credit agreements, or clients who have one or several not repaid credits. We see that such clients much less than clients with good credit history.

Code type of contract	**Count**
1(Credit for the car)	212(1,7%)
6(Mortgage)	64(0,5%)
7(Credit card)	3013(24%)
9(Consumer credit)	8078(65%)
10(The credit for business development)	22 (0,2%)
11(Credit for working capital)	6
12(– Credit for the purchase of equipment)	2
13(Credit for construction of real estate	1
99(Other)	1042 (8%)

Table 5 Clients who have one or several not repaid credits

Analysis of not reliable clients

We will try to define those clients who get into a risk zone. These are people who had one or several credits,- but they cannot pay the loans on time.

We will use the following parameters:

TCS_CUSTOMER_ID	Customer ID
TYPE	Code type of contract
STATUS	Status of the contract 00 – Active 12 – Paid 13 – The account is closed 14 – Transferred to other bank

	21 – Dispute 52 – Overdue 61 – Problems with return
TTL_DELQ_5	The amount of overdue till 5 days
TTL_DELQ_5_29	The amount of overdue 5 to 29 days
TTL_DELQ_30_59	The amount of overdue 30 to 59 days
TTL_DELQ_60_89	The amount of overdue 30 to 59 days
TTL_DELQ_30	The amount of overdue up to 30 days
CURRENT_DELQ	Current number of days of delay

With the parameters we have decided. Now we need to install several filters, for obtaining more exact information. First of all it is:

- *STATUS = [0,21] (*Contracts with a bad credit history*);*
- *TTL_DELQ_5 > 0 OR TTL_DELQ_5_29 >0 OR TTL_DELQ_30_59 >0 OR TTL_DELQ_60_89 >0 OR TTL_DELQ_30 > 0 ;*
- *CURRENT_DELQ > 0 (*There is no debt*);*

As a result we receive statistics of clients who can be in a risk zone:

Code type of contract	**Count**
1(Credit for the car)	19(2,7%)
6(Mortgage)	4(0,6%)
7(Credit card)	95(14%)
9(Consumer credit)	568(83%)
10(The credit for business development)	2(0,3%)

Table 6 Clients who can be in a risk zone

	Count		
Code type of contract	**Risk zone**	**One or several not repaid credits**	**Good credit history**
1(Credit for the car)	19	212	177
6(Mortgage)	4	64	47
7(Credit card)	95	3013	8405
9(Consumer credit)	562	8078	4002
10(The credit for business development)	2	22	2
11(Credit for working capital)	-	6	1
12(– Credit for the purchase of equipment)	-	2	-
13(Credit for construction of real estate	-	1	-
99(Other)	-	1042	494

Table 7 Analysis of clients with different credit history

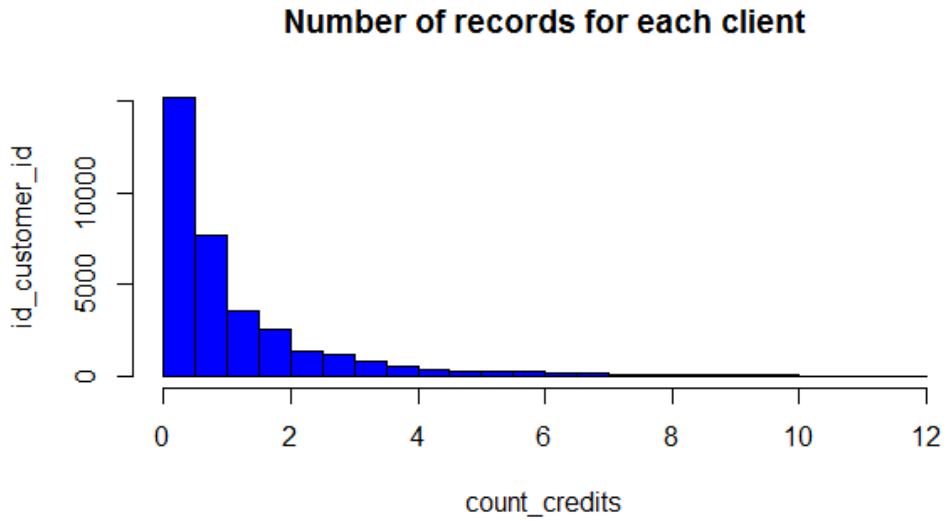


Figure: 2.1 Analysis number of records for each client

This histogram shows how many credits each client has. This greatly complicates our work. To obtain more accurate and reliable results, we need to conduct a more detailed analysis of our data.
In the next chapter, we will try to convert our data in such a way as to obtain more accurate information for the each customer.

2.3 Preprocessing

For a start, we need to transform data for the further analysis.
It is possible to assume that the set of SampleAccounts contains several records on one borrower, let‘s check them.
There are 50,000 unique borrowers out of 280,942 records. This is connected with the fact that one borrower can have several credits and can have different information on each of them. We need to cope with this problem, for the correct analysis of data.
To solve a problem, we need to concentrate full information on each client in one row. Therefore, it is necessary to modify SampleAccounts so what for one borrower corresponds to only one row.
Thus, we will do all our transformations of data in order that each client has only one index and one line, which will contain full information on the client. Having received this index of the client, we will be able to begin the training and testing of models of machine training.

In our database, some lines do not contain values. We can fill gaps with zero (for example "credit limit.
Now, when we have got rid of blank values, let‘s receive the most recent accessed date to any bureau on each credit. It will be useful to us when defining its attributes, such as the contract status, type, etc.

Let's process columns: pmt_string_84m (promptitude in payment), pmt_freq (the code of payment frequency), type (the contract type code), status (the contract status), relationship (the relation type to the contract), bureau_cd (the bureau code by which the account is received). Let's count the number of each unique value for each client and let's enter these values into new columns.
It means that we must calculate the amount of unique values in each row and write these values in the new columns. For example:

	New columns			
pmt_string_84m	**pmt_string_84m_0**	**pmt_string_84m_1**	**pmt_string_84m_2**	**pmt_string_84m_X**
000112XXX	3	2	1	3
XXX222	0	0	3	3
111100002	4	4	1	0

We will try to analyze columns values. We will look at what values are used most often:

Description of columns	**Count**
bureau_cd (Code of bureau from which the account is received)	
1	110345
2	68161
3	102436
Status (Status of the contract)	
00 – Active	127054
12 – It is paid due to providing	1
13 – The account is closed	137136
14 – It is transferred to service to other bank	1245
21 – Dispute	60
52 – Overdue	15362
61 – Problems with return	84
pmt_freq (Code of frequency of payments)	
0 - No data	31837
1 – Weekly	610
4 – Quarterly	23
5 – Semiannually	13
6 — Annually	57
7 – Another	25833
A — Every 2 months	6
B — Every 4 months	3
3 – Monthly	216715
relationship	
1 — Physical person	279672
2 —The additional card / the Authorized user	438
4 — Joint	603

5 — Guarantor	24
9 — Legal entity	205
Currency	
EUR	71
CHF	8
RUB	279885
USD	978

Main Database	**New Database**
pmt_string_84m	pmt_string_84m_0 pmt_string_84m_1 pmt_string_84m_2 pmt_string_84m_3 pmt_string_84m_4 pmt_string_84m_6 pmt_string_84m_7 pmt_string_84m_8 pmt_string_84m_9 pmt_string_84m_X pmt_string_84m_E
pmt_freq	pmt_freq_0 pmt_freq_1 pmt_freq_4 pmt_freq_5 pmt_freq_6 pmt_freq_7 pmt_freq_3
type	type_1 type_7 type_9 type_99 type_463
status	status_0 status_12 status_13 status_14 status_21 status_52 status_61
relationship	relationship_1 relationship_2 relationship_4 relationship_5 relationship_9

currency	currency_RUB currency_EUR currency_USA currency_CHF
bureau_cd	bureau_cd_1 bureau_cd_2 bureau_cd_3
We transfer other columns without changes	
credit_limit	credit_limit
delq_balanc	delq_balanc
next_pmt	next_pmt
curr_balance_amt	curr_balance_amt
current_delq	current_delq
interest_rate	interest_rate
outstanding	outstanding

Table 8 Converting of a data set

The next step is to transform the field **fact_close_date** which contains the date of the last actual payment for it to contain only 2 values: 0 – the last payment is not affected, 1 – the last payment is affected.

After the actions described above our sample is significantly reduced, but now we should generalize the full information on the credit that the borrower received earlier. For this purpose, let's group our dataset.

Our data is almost ready for the beginning of the analysis. It is necessary to perform some more operations:

1. To delete unnecessary columns
2. To convert all credit limits into rubles
3. To count what number of credits each borrower has according to the bureau information

Let's begin with deleting unnecessary columns in the table. These columns of value either are absent, or are so small that they won't influence neither the results of the analysis of data nor the creation of models.

```
bki_request_date,
inf_confirm_date,
pmt_string_start,
open_date,
final_pmt_date,
inf_confirm_date_max
```

In our dataset, the sums of the credits are written down in several currencies. The main currency which is used in this dataset is rubles. Therefore, we will convert all credit limits- into rubles.

Currency	Count	Exchange
CHF	8	57.89

EUR	71	61.96
RUB	279885	1
USD	978	57.02

To make it simpler. I take into account exchange rates at the moment. Though it would be more correct to take into account the exchange rate at the moment of opening the account. One more nuance is that we should delete the text field "currency" for the analysis, therefore, after converting currencies into rubles, we will carry out some actions with this field which we have carried out with the fields above.

So, before final grouping we will add the field completed with units to our set. Thus, when we execute last grouping, the sum will contain the number of borrower's credits.
Now, when we have all quantitative data in the dataset, it is possible to fill gaps in data 0 or write down average value (depending on features) and to execute final grouping on the client.
Then let's study how features correlate among themselves, for this purpose let's develop a matrix with the correlation coefficients of features.

After the action above, CorrKoef will contain the matrix with sizes 38 x 38. The corresponding names of fields will be its rows and columns, and the correlation coefficient value will be at their intersection.
There is a chance of lacking the correlation coefficient. It means that these fields are most likely completed only with one identical value and they can be omitted upon analyzing.
At the output, we got a list of fields that can be deleted. The values of these columns will be identical, and, therefore, will not affect the further analysis and the result.

```
pmt_string_84m_6
pmt_string_84m_8
pmt_freq_5
pmt_freq_A
pmt_freq_B
status_12
fact_close_date
ttl_delq_30
max_delq_balance
relationship_1
currency_RUB
currency_USD
currency_CHF
currency_EUR
count_credit
```

After processing of primary data we have **14968** rows and **38** columns.

For reduction of the dimension of our data and also for definition of the most important features there are several methods. We will consider some of them.

2.4 Principal components analysis (PCA)

Extract or print loadings in factor analysis (or principal components analysis).
PCA is one of the main methods to reduce the dimensionality of data by losing the least amount of information. It is used for data compression. Now let's reduce the dimension of our sample to take only significant parameters. Analysis, or PCA. PCA produces linear combinations of the original variables to generate the axes, also known as principal components.

Furthermore, we have to look at loadings of principal components:

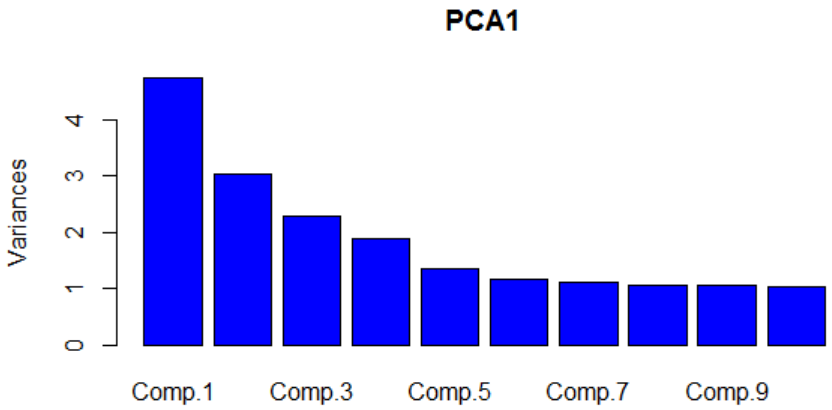


Figure: 2.2 Loadings of principal components in R

Now let's try to test different feature sets. In different datasets we have tried to use the most important features. Using the principal component analysis, we tested three feature sets: testing PCA for 9 features, for 18 features and for 38 features (all features).
9 the most important features using the principal component analysis:

Importance of 9 components	**Standard deviations**
Comp.1	1.9253806
Comp.2	1.4074913
Comp.3	1.2051288
Comp.4	1.0383948
Comp.5	1.0349746
Comp.6	1.0003642
Comp.7	0.9888289
Comp.8	0.9765621
Comp.9	0.9632105

18 the most important features using the principal component analysis :

Importance of 18 components	**Standard deviations**

Comp.1	1.9253806
Comp.2	1.4074913
Comp.3	1.2051288
Comp.4	1.0383948
Comp.5	1.0349746
Comp.6	1.0003642
Comp.7	0.9888289
Comp.8	0.9765621
Comp.9	0.9632105
Comp.10	0.9339218
Comp.11	0.8826064
Comp.12	0.8505497
Comp.13	0.7897450
Comp.14	0.7675450
Comp.15	0.7495107
Comp.16	0.6211384
Comp.17	0.5139296
Comp.18	0.2257936

We will use these data sets in the classification in the following chapters of our work.

2.5 Random Forest

There is another method, ideologically close to the classification trees, called "Random Forest"[1], because the basis of the method is the production of a large number of classification "trees".
Random Forest makes it possible to find out the importance of each feature, as well as the distance between all the objects of the training sampling (proximity), which can then be used for clustering or multidimensional scaling.
At last, this method allows "pure visualization" of data, which means it can work as a classificati on method without training.
We use the `importance` function. This is the extractor function for variable importance measures as produced by Random Forest:

Inc. Node Purity:	
outstanding	22168028.3
next_pmt	1146291928.1
curr_balance_amt	811643090.2
current_delq	20410718.6
interest_rate	262298770.0
delq_balance	995616802.5
credit_limit	219253806.1
pmt_string_84m_0	61787440.3
pmt_string_84m_1	851357124.9
pmt_string_84m_2	54687243.3
pmt_string_84m_3	25156564.8
pmt_string_84m_6	2302435.4
pmt_string_84m_7	9528237.1
pmt_string_84m_9	17752423.6
pmt_freq_0	2855561.6

pmt_freq_1	0.0
pmt_freq_4	0.0
pmt_freq_6	57843812.2
pmt_freq_7	144382417.4
type_1	342933.1
type_463	0.0
type_11	2262321.7
type_12	0.0
type_13	0.0
type_14	0.0
type_4	579144109.3
type_9	780779494.2
status_0	177008028.2
status_13	0.0
status_14	0.0
status_21	15640546.9
status_52	0.0
status_61	0.0
relationship_2	9515963.8
relationship_4	0.0
relationship_5	0.0
relationship_9	7495121.5
crs	15855139.7

Table 9 Preprocessing, using RandomForest

Chapter 3

Analysis and selection of models

The next step (after the completion of the preliminary data analysis) is to build the credit scoring models**.** To build a model, we first need to make a selection of credit history from the bank's customers who have fully paid or have a delay in paying their credit. The sample is divided into two groups: customers who were given credit and customers who have delayed the credit for a certain term (in Russia this term is two weeks)**.** Currently, there are a large number of methods to build the credit scoring models, which include:

- The statistical methods based on the discriminant analysis (linear regression, logistic regression);
- various variants of linear programming;
- classification tree or recursion-partitive algorithm (RPA);
- neural networks;
- genetic algorithms;
- The nearest-neighbor method.

Comparison of credit scoring models can be carried out by many criteria, but in practice in most cases, a comparison of probability estimates for each method is used, by what the most suitable method is chosen.To calculate the accuracy of a forecast the initial sample is divided into a piece of training and a testing set (for example 80% and 20% of the original sample). The model is built on the training sample, and then is checked on the testing sample.
For modeling, the following algorithms will be considered:

• SVM (SupportVectorMachine)
• Logistic regression(LR)
• Boosting
• Nearest Neighbor (kNN)
• Random Forest(RF)
• Neural Network(NN)

3.1 Selection of a training method

The task of model selection is as follows. Based on a comparison of results we need to choose we need to choose best and most accurate of all forecasting models. For this, we will build the whole further process.
There are T various training methods: $\mu_1, \dots, \mu_T$
All T methods are used in the training set X^l resulting in T algorithms of $a_t = \mu_T(X^l)$**.**

Here is another case – there is a fixed model, and its method of setting–**μ**, but this approach depends on some parameters that cannot be optimized on the training set, and should be assigned

a priori. Therefore, in addition to the selected model, we need to select the parameters of the model. For the choice of parameters and the choice of model, we use cross-validation.
A cross-validated model selection is the most widely spread:

$$CV(t, X^l) = \frac{1}{N} \sum_{n=1}^{N} Q(\mu_t(X^l), X_n^k), t = 1, \dots, T;$$
$$t^* = \arg \min_{t=1,\dots,T} CV(t, X^l);$$

where X^l is a full sample, divided by N number of ways on a training part X_n^l and a control part X_n^k.

3.2 ROC-analysis

Accuracy is an important feature of any regression model. The ability to distinguish "good" borrowers from "bad" ones defines an accuracy of a classifier. A discriminating ability of the model can be estimated by analyzing the classification table. It is very important to create a model, which is equally useful for defining both "good" and "bad" borrowers. ROC-curve (Receiver Operator Characteristic) is plotted to estimate the quality of the classification model, and this curve shows the dependence of the number of correctly classified positive outcomes on the number of incorrectly classified negative outcomes.

Some errors may occur when using the binary classification model – the inconsistencies of model output values and the actual sample values. The model can estimate a prediction corresponding to a positive outcome as a negative one, and vice versa.
Thus, the purpose of ROC-analysis [5] is to select such a cut-off point value, allowing the model to identify positive or negative outcomes with the maximum accuracy and provide the least amount of false positive or false negative errors.

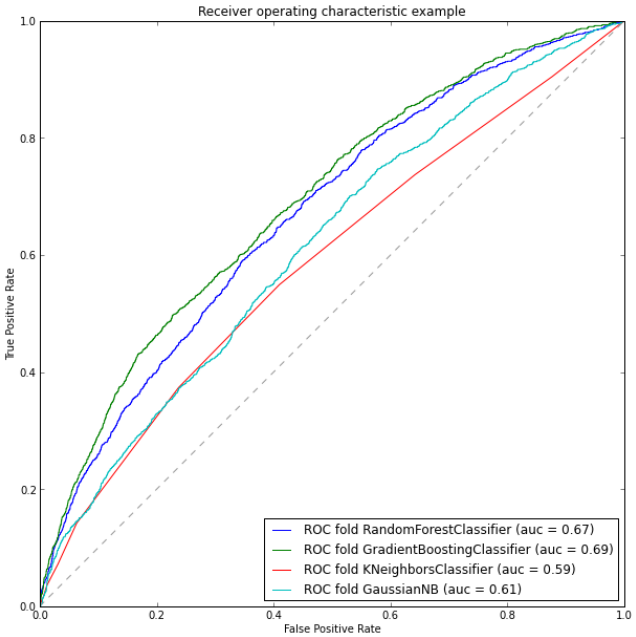


Figure: 3.1 Constructing an ROC curve.

The diagnosed value with a zero degree of prediction is plotted as a diagonal. The curve of an ideal model passes through the upper left corner, where the percentage of true positive cases is 100%. Therefore, the more curved is the ROC-curve, the more accurate is the prediction of the model results. The indicator of this feature is the area under the ROC-curve. To compare two or more models against each other, the area under the ROC-curves is compared - this parameter is called the AUC (Area Under Curve) and is within the range of (0.5…1). Accepting larger assumptions we can consider that the larger is the AUC ratio, the better is the predictive power of the model.
Table 1 shows how good is the predictive power of the models. It contains an excellent expert quality assessment of scale models, depending on the area under the ROC-curve[2].

Interval AUC	**Quality of model**
0,9-1	Excellent
0,8-0,9	Very good
0,7-0,8	Good
0,6-0,7	Average
0,5-0,6	Unsatisfactory

Table 10 Quality of the model according to the area under the curve [2]

An ideal model has a 100% sensitivity and specificity. But such result is impossible to achieve in practice since you cannot increase the sensitivity and specificity of the model at the same time. 100% of sensitivity and specificity means all the examples, both positive and negative are detected correctly. But in reality, a compromise must be found with the cut-off threshold, and this threshold affects both the sensitivity and specificity. In our case, the models presented provides not accurate results.

3.3 Cross-validation

Cross-validation[7] is most widely used method for estimating prediction error. This method directly estimates the expected extra-sample error $Err = E\left[L(Y), \hat{f}(x)\right]$, the average generalization error when the method $\hat{f}(x)$ is applied to an independent test sample from the joint distribution of X and Y [0].
The test error is the average error that results from using a statistical learning method to predict the effect on a new observation. Using a data set, the use of a particular statistical learning method is right if it results in a low test error. The training error can be calculated by applying the statistical learning method to the observations used in training. But the training error can be different from the test error rate [1].

If we had enough data, we could set aside a validation set and use it to assess the performance of the model. But data are often scarce, so this is impossible. To solve the problem, we can use K-fold cross – validation. For example, when K = 5, the result looks like this.

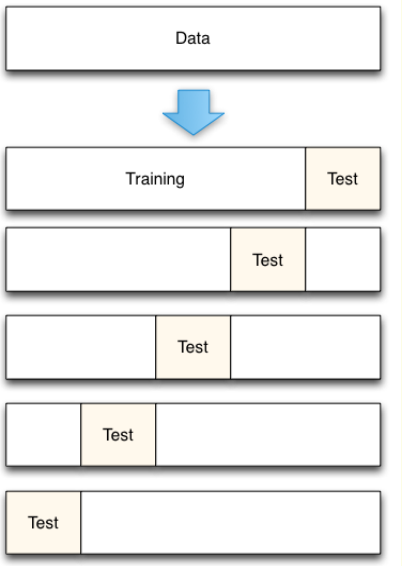


Figure: 3.2 K-fold cross-validation [3]

The initial data set is divided into *K* folds of equal size. One of *K* folds is left to test the model, and the rest *K-1* folds are used as a training set. The process is repeated for *K* times, each of the folds is used once as a test set. *K* results are obtained, one for each fold, they are averaged or combined in some other way, and provide a single estimation. The advantage of this method over the random subsampling is this: all the observations are used for model training and model testing, each observation is used for testing just once.
We can perform a quantitative comparison of these methods concerning their relative coefficients of the classification errors. Error coefficient may be used as the matching estimation.
What value is optimal for K? With $K = N$, the cross-validation estimator is approximately unbiased for the true (expected) prediction error, but can have high variance because the *N* "training sets" one similar to one another [7].
The computational burden is also considerable, requiring *N* applications of the learning method.

Here is the way to carry out cross-validation:
1. Divide the samples into K cross-validation folds (groups) at random.
2. For each fold k = 1, 2, . . . ,K
   - Find a subset of "good" predictors that show fairly strong (univariate) correlation with the class labels, using all of the samples except those in fold k.
   - Using this subset, build a multivariate classifier, using all of the samples except those in fold k.
   - Use the classifier to predict the class labels for the samples in fold k.

With a multi-step modeling procedure, cross-validation must be applied to the entire sequence of modeling steps [1].

3.4 The logistic regression model

Logistic regression is the most widely spread statistical model for making scorecards with a binary dependent variable.
Logistic regression is a kind of multiple regression, and its main purpose is to analyze the interaction between multiple independent variables (also called predictors) and the dependent variables. We can evaluate the probability of the event occurring for a particular subject with the help of binary logistic regression (loan repayment/default, etc.).
Consider the data set, where the response default falls into one of two categories, Yes or No. Logistic regression models the probability that Y belongs to a particular category [1].

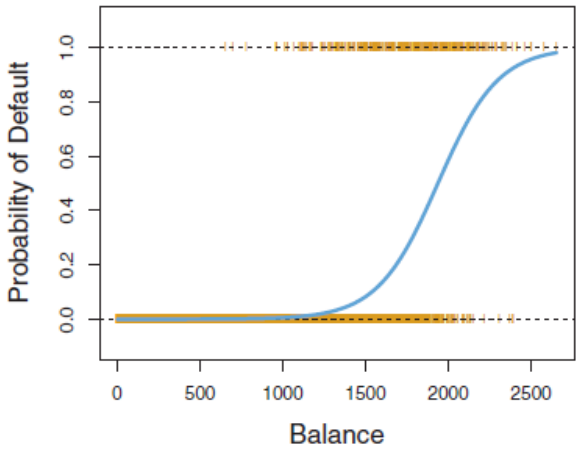


Figure: 3.3 Predicted probabilities of default using logistic regression [2]

The regression problem can be formulated in another way: we predict a continuous variable with the values within the interval of instead of predicting a binary variable, for all values of the independent variables. This can be achieved by using the following regression equation (logit-transform):

$$\ln \frac{p_i}{1-p_i} = b_0 + b_1 x_i^{(1)} + b_2 x_i^{(2)} + \cdots + b_k x_i^{(k)} + \varepsilon_i \quad (1)$$

where
p_i– is the probability of a default of *i-th* borrower on the loan;
x_i– is the value of the *j-th* independent variable;
b_0 – is the independent model constant;
b_j – are the parameters of the model;
ε_i – is a component of random error;

Equation (1) shows the linear dependence of the probability of default on loan, depending on the values of the independent variables. The constant in the model is the natural level of risk of simulated event occurrence in a case independent variables are equal to zero. The coefficient

values of the independent variables reflecting the degree of their influence on the probability of default on a logarithmic scale are used to construct the scorecard.

A constant value in a logistic regression model depends on the data distribution by the categories of the dependent variable.

3.5 Boosting model

Boosting [1] is one of the most powerful learning ideas which are used in the last several years. It was originally designed for classification problems, but it can be extended to regression as well. The motivation for boosting was a procedure that combines the outputs of many "weak" classifiers to produce a powerful "ensemble."
Boosting [8] – is the process of composition of serial machine learning algorithms, and each following algorithm seeks to compensate the shortcomings of the previous algorithms. Boosting is a quite greedy algorithm for constructing algorithms composition.
Let's look at a formal statement of the problem. Typically we have a training set, i.e. a pair of object-response and we are going to build a linear composition of the basic algorithms.
Consider the task of object recognition from the X multidimensional space with space marks $Y\in\{-1,+1\}$. Assume we are given a training set of $\{x_i\}_{i=1}^{N}$, where $x_i \in X$. And we know the true values of labels for each objec $\{y_i\}_{i=1}^{N}$, where $y_i \in Y$. So we should develop the recognizing operator, predicting labels for each new object x ∈ Xwith the maximum possible accuracy.
Suppose we are given a family of H, basic algorithms, each element $h(x,a) \in H : X \longrightarrow R$ of which is defined by a parameter vector $a \in A$.

The final classification algorithm is as follows:

$$F_m(x) = F_{m-1}(x) + b_m h(x, a_m), \text{ where } b_m \in R,\ a_m \in A.$$

So now our task is to find the optimal pair of parameters $\{a_m, b_m\}$, for a length classifier m.
Next, we introduce some function of losses $L(y_i, F_m(x_i))$, $i = 1..N$ it shows "how much" the pr edicted response $F_m(x_i)$, differs from the correct answer y_i then the error function is minimized:

$$Q = \sum_{i=1}^{N} L(y_i, F_m(x_i)) \longrightarrow min.$$

An example of boosting, AdaBoost, assumes the use of an exponential loss function

$$L(y, F) = \exp(-yF).$$

p_m stands for the error rate, resulting from the current basic algorithm.

Algorithm 1: Discrete AdaBoost [3]
Input: $\{x_i\}_{i=1}^{N}$, $\{y_i\}_{i=1}^{N}$, M;
Exit: $F_M(x)$
$w_i = \frac{1}{N}$, $i = \overline{1,N}$
for $m = \overline{1,M}$

a_m= **train** $(\{x_i, y_i, w_i\}_{i=1}^N)$;
$p_m = \frac{1}{N}\sum_{i=1}^N I[y_i \neq h(x_i, a_m)]$;
$b_m = \frac{1}{2}\log\frac{p_m}{1-p_m}$;
$w_i = \frac{w_i}{\sum_{j=1}^N w_i}, \overline{1,N}$;
$F_M(x) = \sum_{m=1}^M b_m h(x, a_m)$;

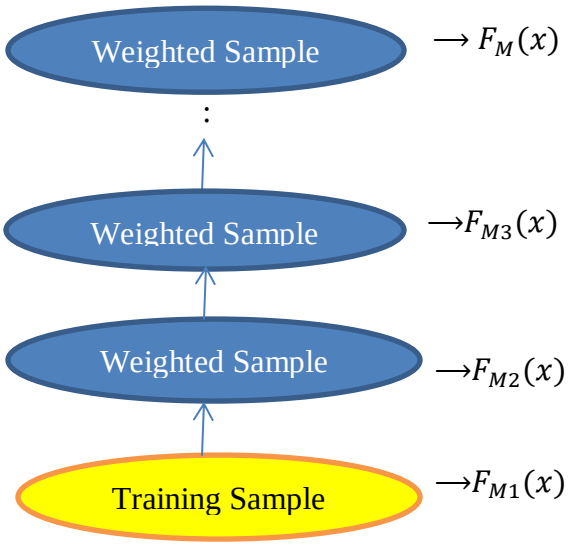


Figure: 3.4 Schematic of AdaBoost. Classifiers are trained on weighted versions of the dataset, and then combined to produce a final prediction.

3.6 Nearest-neighbor method

3.6.1 Nearest neighbor algorithm

The nearest neighbor algorithm [2] is the simplest metric classifier which is based on the estimation of the objects similarity.
The classified object belongs to the class which has training sample objects closest to it. The simplest nearest neighbor method is:

$$w(i,u) = [i = 1]; a(u, X^l) = y_u^{(1)}, \ where\ u \in X^l \text{ classified object.}$$

kNN training is reduced to memorization X^l of the sample. The only advantage of this algorithm is the easy implementation. But there are lots of limitations instead:

- Sensitivity to errors. If there is an overshoot in the training set – the object surrounded by those of another class, not only it will be classified incorrectly, but also the objects around it to which it will be the closest.

- No parameters one can set for a particular sample. The algorithm entirely depends on how well metric ρ was chosen.
- The result of this is a low quality of classification.

3.6.2 Algorithm of k nearest neighbors (kNN)

To reduce the effect of outliers we assign an object u to the class which has more representatives among the nearest neighbors $x_u^{(i)}, i = 1, \dots, k$:

$$w(i,u) = [i =\leq k]; a(u, X^l, k) = \arg\max_{y<Y} \sum_{i=1}^{k} [y_u^{(i)} = y].$$

The extreme values of k are undesirable. In practice, the optimum value of the k parameter(for the classified object (u)) is defined by the criterion of a cross-validation with a leave-one-out (LOO). Each object $x_i \in X^l$ is checked on the proper classification by its k nearest neighbors.
There is also an alternative *kNN* method: k nearest to u objects are selected in each class, and u object belongs to the class with the minimal average distance to the k nearest neighbors.

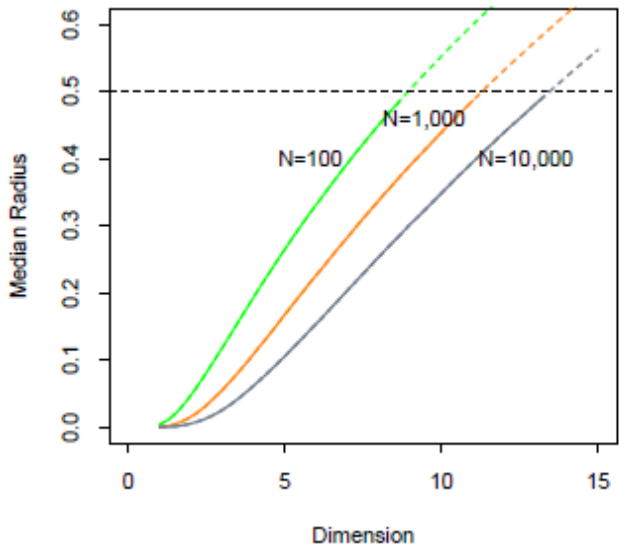


Figure: 3.5 Median radius of a 1-nearest-neighborhood,
for uniform data with N observations in p dimensions [1]

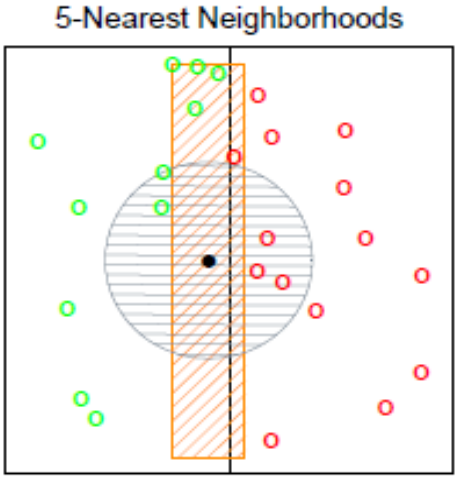


Figure: 3.6 The sphere shows the 5-nearest-neighbor region using both coordinates, and we see in this case it has extended into the class-red region [1]

3.6.3 The algorithm of k weighted nearest neighbors

There is one disadvantage of kNN: the maximum can be achieved in several classes at the same time. This can be avoided in tasks with the two classes if you take the odd *k*, or with a more general method in case of numerous classes – introduce a strictly decreasing sequence of real weights w_i , specifying the contribution of the *i*-th neighbor to the classification:

$$w(i,u) = [i =\leq k]w_i;\;\; a\big(u, X^l, k\big) = arg\max_{y<Y} \sum_{i=1}^{k} \big[y_u^{(i)} = y\big]\, w_i.$$

3.7 Naive Bayes classifiers

Bayesian classifiers are statistical classifiers. The Bayesian classifier is based on „Bayes“ theorem. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes [10].
Naive Bayes classifiers [10] are a particular special case of a Bayesian classifier, based on the additional assumption: objects *X* are described by *n* statistically independent features.
Let us assume, the objects $x \in X$ is represented by n numerical attributes $f_j: X \longrightarrow R, j = 1, \dots, n$.
Let $x = (\xi_1, \dots, \xi_n)$, an arbitrary element of the object space $X \in R^n$, where $\xi_j = f_j(x)$.

Hypothesis 3.1
Characteristics $(f_1(x), \dots, f_n(x))$ are the independent random quantities. So, the classes likelihood function can be represented in the form of $p_y(x) = p_{y_1}(\xi_1), \dots, p_{y_n}(\xi_n), y \in Y$,
where $p_{y_j}(\xi_j)$, the density of distribution of values of the j-th characteristic for the ***y*** class.

The independence assumption greatly simplifies the task, as it is much easier to estimate n one-dimensional densities than the one n-dimensional density. However, it is seldom performed in practice, therefore, the classification algorithms are called naive Bayesian algorithms.
Naive Bayes classifier assumes that the characteristics describing the sampling elements are conditionally independent at given classification class.
Its main advantages – easy implementation and low computational cost of training and classification. In those rare cases when characteristics are (almost) independent, Naive Bayes classifier is (almost) optimal.
Its main drawback is the low classification quality. It is used either as a standard for an experimental comparison of algorithms or as an "Elementary part" in some complex algorithm.

3.8 Support Vector Machines (SVM)

The support vector machine is a set of the similar supervised learning algorithm, used for problems of classification and regression analysis. It belongs to the family of linear classifiers can also by considered as a special case of Tikhonov regularization [6]. A special property of a method of support vector machine is a continuous decrease of an empirical classification error and an increase in the margin, so the method is also known as a method for classifier with the maximum margin [6].
The main idea of this method is as follows – to transform the original vectors in space with more dimensions and a search of separating hyperplane with the maximum margin in this space. Two parallel hyperplanes are constructed on both sides of the hyperplane, they separate our classes. The separating hyperplane is the hyperplane maximizing the distance to the two parallel hyperplanes. The work of algorithm is based on the assumption that the higher is the difference or distance between two parallel hyperplanes, the smaller is the average error of the classifier.

Statement of the problem

This is a typical case of linear separability. There may be lots of such hyperplanes. Therefore it is natural to assume that maximization of the gap between the classes results in more confident classification. That is, are we able to find a hyperplane where the distance from it to the nearest point would be maximum. This would mean that the distance between two closest points lying on opposite sides of the hyperplane is maximal. If such a hyperplane exists, it is the most interesting for us; it is called the *optimal separating hyperplane*, and the corresponding linear classifier is called the *optimal separating classifier*.
Support vectors are the points for which the distance to the hyperplane. They are the active elements in the training set.

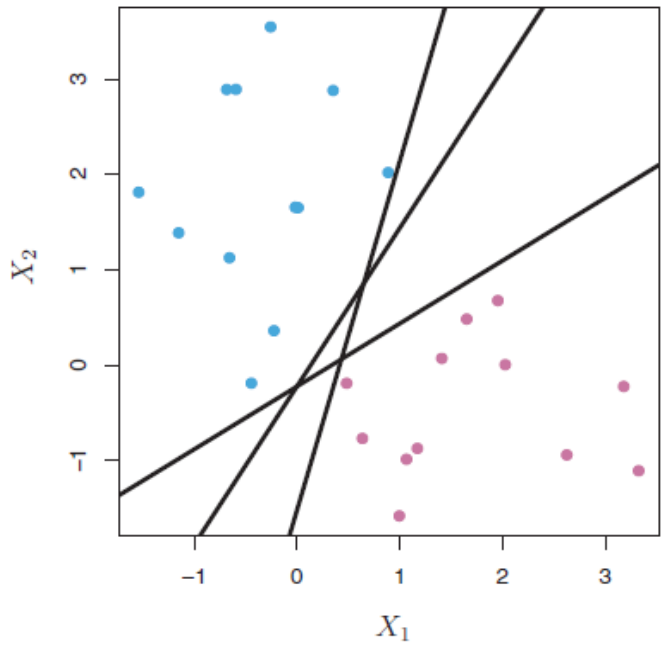


Figure: 3.7 There are two classes of observations, each of which has measurements on two variables. Three separating hyperplanes, are shown in black [2]

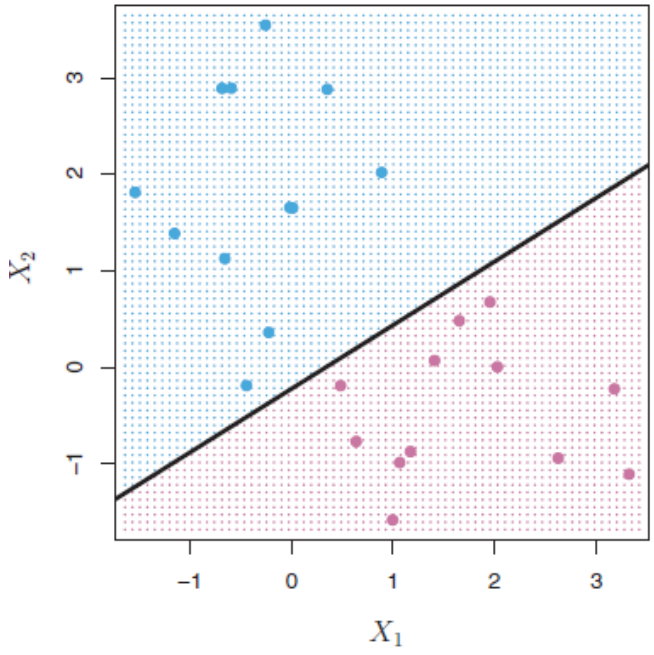


Figure: 3.8 A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane [2]

Property of support vectors method

If all points except the support vectors are removed, the support vectors model remains the same. This feature makes SVM a unique, differing from all other methods like kNN, NNet and NB,

where all the points of the training set are used to optimize the function. The deological difference leads to a significant difference between SVM and other methods in practice.

3.8.1 Linear Support Vectors Method

The observations *D* are given for training, the set consists of *n* objects:

$$D = \left\{ (x_i,\ y_i) \mid x_i \in R^p,\ y_i \in \{-1, 1\} \right\}_{i=1}^{n},$$

Where *y* takes the values «-1» or «1», determining which class each point x_i.belongs to.
Each point x_i. Dimension vector *p*. We want to find the hyperplane of maximum difference separating the observations having y_i=1 from objects with $y_i = -1$. Any hyperplane can be written as a set of points x, satisfying this expression:
$w * x - b = 0,$ where
* - where * is a scalar product of the normal to the hyperplane on the x vector. The parameter $\frac{b}{\|w\|}$ defines the shift of the hyperplane from the origin of coordinates along the normal w.

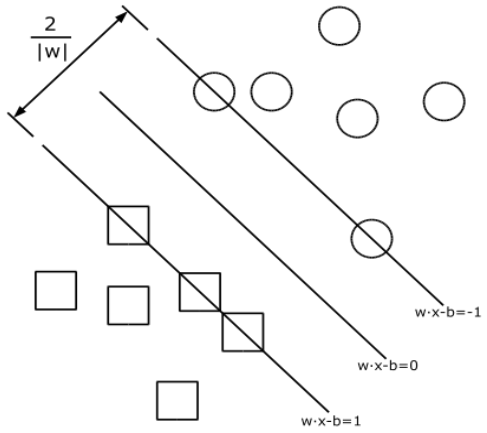


Figure: 3.9 Optimal separating hyperplane for support vector constructed on the points of the two classes [4]

In case the training data is linearly separable, we can select two parallel hyper planes in such a way, they separate a set of points on 2 classes, and there will be no points between them. Then we try to maximize the distance between them while in parallel making a turn and shift of parallel lines. The area within two hyperplanes is called "the margin." This hyperplane can be described by these equations:

$$w * x - b = 1$$
$$w * x - b = -1$$

Using the geometry, we define the distance between the hyperplanes $\frac{2}{\|w\|}$.
To maximize distance, we minimize $\|w\|$.
To exclude all the points from the band, we have to make sure that this condition applies to all observations:

$w * x_i - b \geq 1$, or all of the first class.
$w*x_i-b \leq -1$, for all of the second class.

Further the optimization problem is solved:

$$\|w\| \longrightarrow min$$
$$y_i(w * x_i - b \geq 1) for \ 0 \leq i \ \leq n.$$

3.8.2 Non-linear classifier

We use an arbitrary kernel function to create a non-linear classifier. Each scalar product is replaced by a nonlinear kernel function. This allows finding a hyperplane of a maximum difference in the transformed space of functions. The change may be non-linear and can be transformed into the higher dimension space. Despite the fact that the classifier is a hyperplane in multidimensional space of functions, it can be non-linear in the original space of the training set.
.

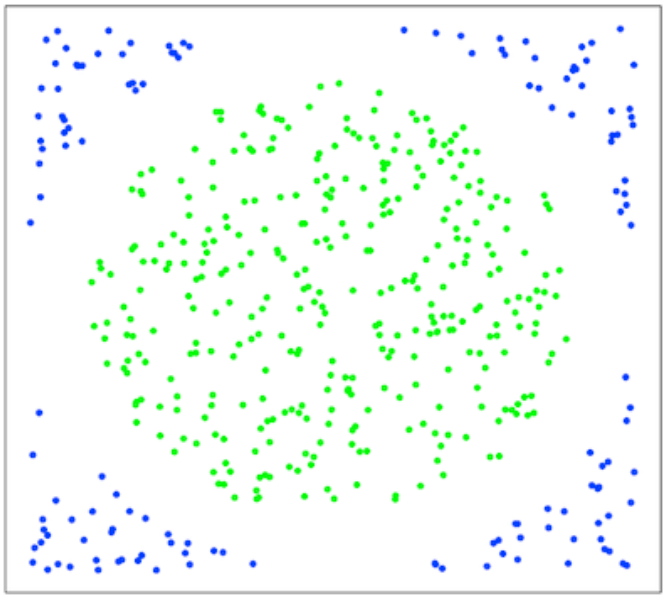


Figure: 3.10 Example of linear inseparability .
The method with the use of kernel functions

Some kernels include:

- A homogeneous polynomial: $k(x_i, x_j) = (x_i, y_j)^d$
- A non-homogeneous polynomial: $k(x_i, x_j) = (x_i * y_j + 1)^d$

- A Gaussian radial basis function: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|), \gamma > 0$
- Sigmoid: $k(x_i, x_j) = \tanh(kx_i * x_j + c)$, for almost all $k>0$, $c<0$.

Compared to other methods SVM works well at small samples and does not depend on the distribution of the input data. Besides SVM is based on the structural risk minimization principle, when the prediction error and the structural complexity of the system are optimized and there are theoretical results proving SVM provides stability results.

Advantages and disadvantages of SVM:

- The method is reduced to solving of quadratic programming task in a convex domain, which always has a unique solution;
- The method defines the separating band of the maximum width, allowing to perform a more confident classification;
- The method is sensitive to noises and data standardization;
- There is no standard approach for the automatic kernel selection (and constructing of a rectifying subspace as a whole) in the case of the linear inseparability of classes.

3.9 Artificial neural network

Artificial neural networks[20] are one of the main technologies for solving problems of processing and data analysis, image recognition, classification, and forecasting. Neural networks are based on the principle of connectionism - they connect a large number of relatively simple elements, and training is reduced to the construction of the optimum structure of communications and control of parameters communications. The basis of the neural network is a neuron an element which imitates work of neurons of a brain. Inputs and produces an output which is connected to inputs of other neurons. Also, there is an exit (axon) a signal from which arrives on synapses of other neurons.
Activation of the neuron is defined as a weighted sum of its inputs. Formally, the artificial neuron - is a single-layer perceptron (perceptron), i.e. model, in which input elements are directly connected with the output weights using the system, and performs a linear classification function.

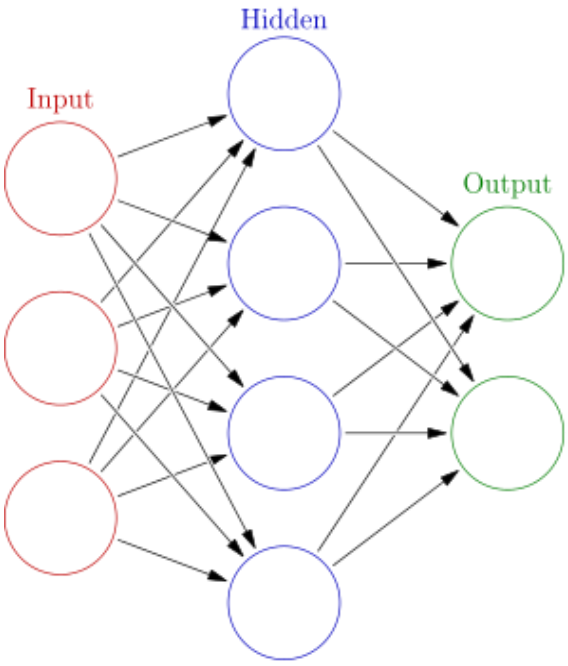


Figure: 3.11 Each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another [12].

The popularity of neural networks is partly caused by an opportunity to model difficult situations without special costs from the user. By their nature, neural networks automatically detect any situation in a non-linear data and adjust to it. Also multilayer neural networks are universal approximator, that can approximate any function as much as precisely.

Neural networks[2] are made up of layers which in turn are composed of nodes. There are three types of layers in the network: input, hidden, output. The input layer is formed customer attributes such as gender, age, etc.
The output y_k for the *k*-th node with *m* inputs is represented so:

$$y_k = \varphi(v_k) = \varphi\left(\sum_{j=0}^{m} \omega_j x_j\right) = \varphi(w^T x),$$

where φ - activation function, x- vector of input data, ω- the weight vector which designates communication force between knots.
Compared with the linear statistical methods (linear regression, autoregression, a linear discriminant), neural networks allow to build effectively the nonlinear dependences more precisely describing data sets.
The main disadvantage is that in spite of the possibility to achieve high accuracy of the forecast, it is impossible to understand the reasons for which this or that decision has been made.

Chapter 4

Training and selection of models

In this chapter we have tried to realize, in practice, the models that have been considered in chapter 3.
At once we want to note that we didn't manage to test all models.
A number of comparative researches were conducted -for scoring methods. As criteria for ranging served percent of mistakes in case of classification and a ROC curve. We studied 3 databases of 3 different banks.
For the problem of credit scoring we researched the following models:

1. Logistic Regression model
2. Nearest Neighbor model
3. SVM (Linear)
4. SVM (Radial)
5. Boosting model
6. Random Forest
7. Neural Network

4.1 Testing of logistic regression model

We use the training set derived from the credit history of borrowers. As a result of performance and of settlements, all borrowers are divided into two classes: reliable and unreliable.
In this work, we tested various sets of features. The following two sets of features, we chose using the Random Forest method.
The best result for 9 features from our data set is the following set:

```
credit_limit
pmt_string_84m_1
pmt_string_84m_9
pmt_freq_1
type_9
status_0
status_52
relationship_2
crs
```

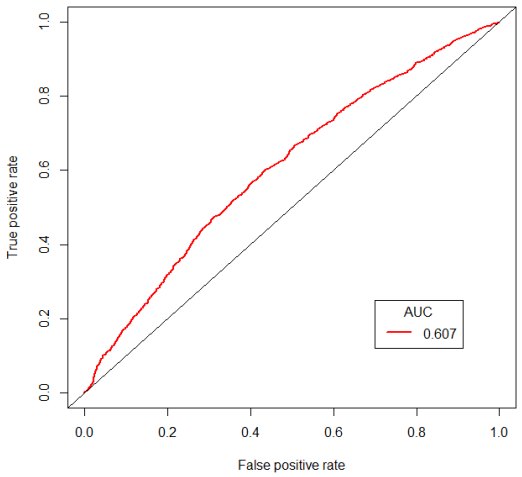
The best result for 18 features from our data set is the following set:

```
 pmt_string_84m_0
pmt_string_84m_1
pmt_string_84m_6
pmt_string_84m_7
pmt_string_84m_9
pmt_freq_1
pmt_freq_7
type_1
type_463
type_4
type_11
type_9
status_0
status_13
status_52
relationship_2
relationship_9
crs
```

For testing the model of logistic regression, we used the built-in GLM function and tried to take into account the maximum number of parameters, such as:

- data splitting
- pre-processing
- feature selection
- model tuning using resampling
- variable importance estimation

The calculated coefficients of logistic regression are shown in the figure 4.1.
We can see in Figure 4.1 that the AUC value is 0.6, and this result is not good at all.
Next, we see that the ROC-curve is close enough to the diagonal, so the forecast of the model is not highly accurate.



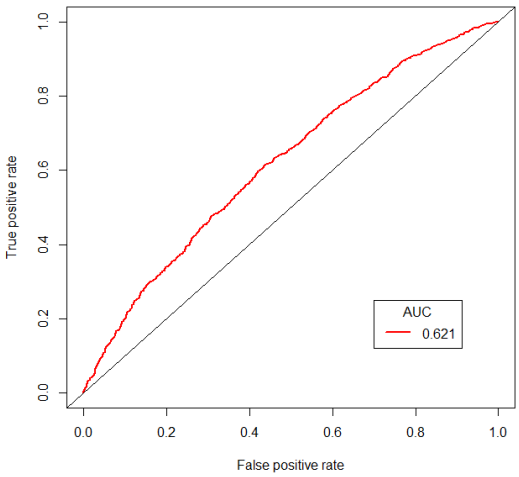


Figure**:** 4.1 Roc curve of logistic regression (9) Figure**:** 4.2 Roc curve of logistic regression (18)

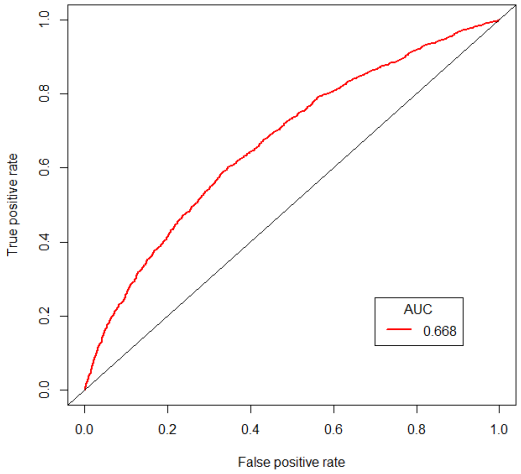


Figure**:** 4.3 Roc curve of logistic regression (38)

Sensitivity is the part of the true positive cases.
Specificity is the part of the true negatives cases, which were correctly identified by the model.

4.2 Testing of Boosting model

In working for Boosting model we used an algorithm: AdaBoost.
In our study we have researched the Boosting model, and here are the results obtained:
Figure 4.5 shows that the rate of AUC is 0.59, and this result is even worse than that of the previous example. Also, we see that the ROC-curve is located close to the diagonal, so the forecast of this model is not accurate enough.

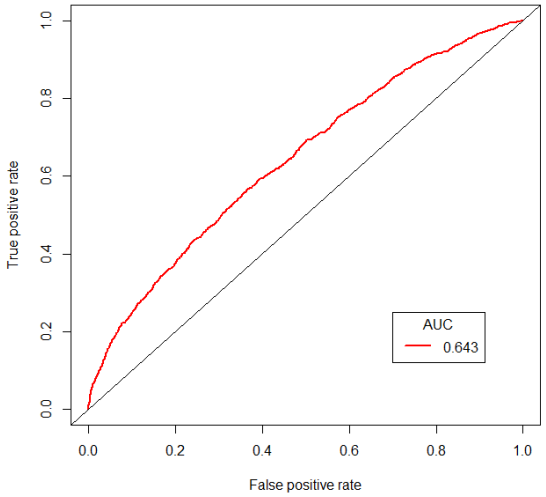


Figure: 4.4 Roc curve of AdaBoosting (9)

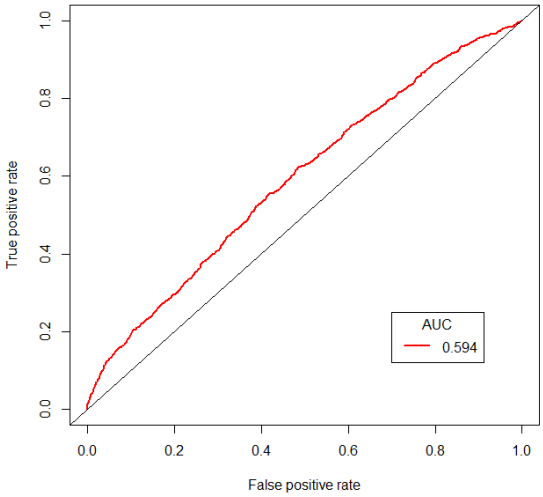


Figure: 4.5 Roc curve of AdaBoosting (18)

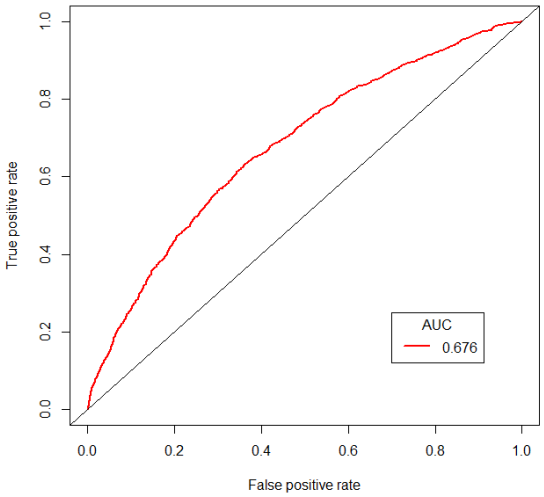


Figure: 4.6 Roc curve of AdaBoosting (38)

4.3 Testing of Nearest Neighbor model

When using a method of nearest neighbor is chosen the unit of measure for determination of distance between clients. The neighbors are taken from a training set. In our work, k = 10.
Now let us look on how this method proved itself in our work:

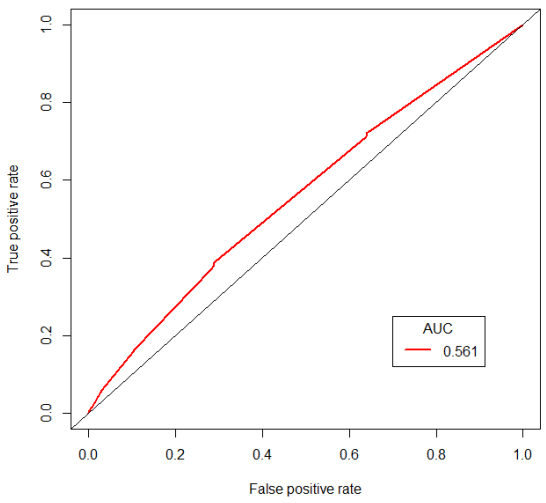


Figure: 4.7 Roc curve of kNN (9)

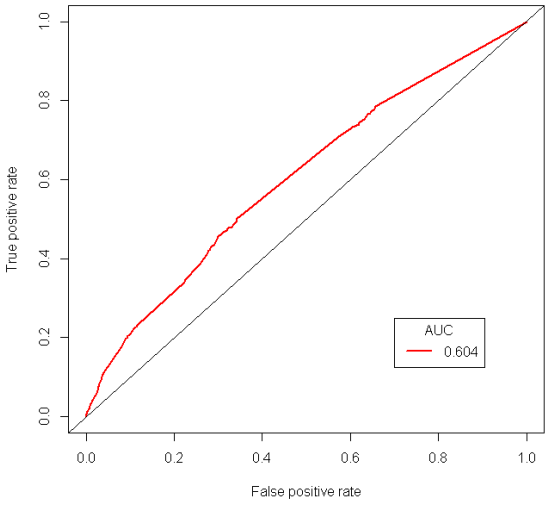


Figure: 4.8 Roc curve of k NN (18)

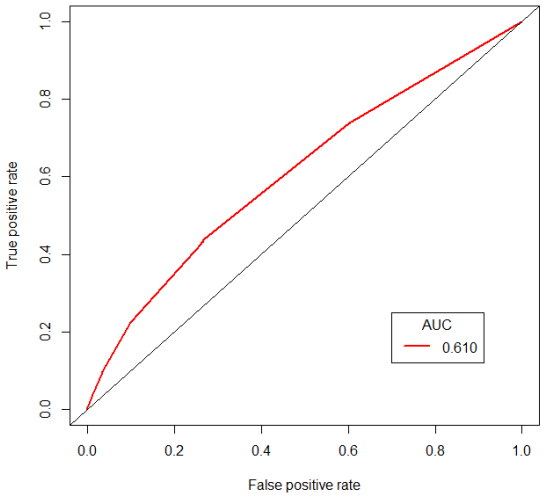


Figure: 4.9 Roc curve of kNN (38)

4.4 Testing of SVM model

The SVM method for creating a non-linear classifier uses an arbitrary kernel function.
Each scalar product is replaced by a non-linear kernel function. It allows us to find the hyperplane of the maximum difference in the transformed space of functions.
Here are the examples of the support vectors method with the use of different kernel functions:

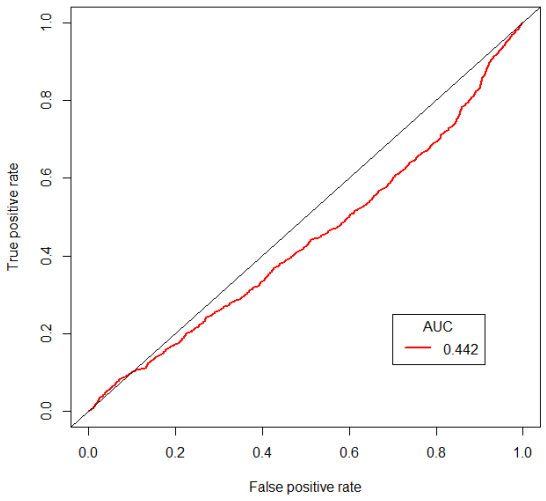


Figure: 4.10 Roc curve of SVM:Linear (9)

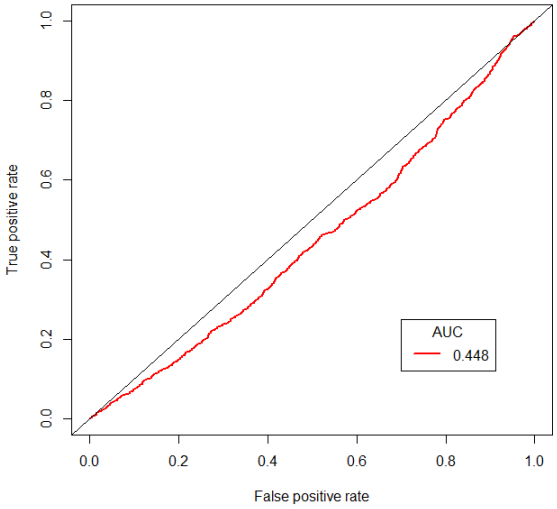


Figure: 4.11 Roc curve of SVM:Linear (18)

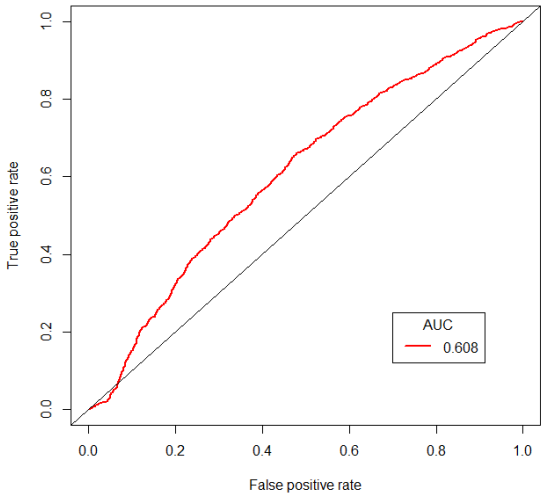


Figure: 4.12 Roc curve of SVM:Linear (38)

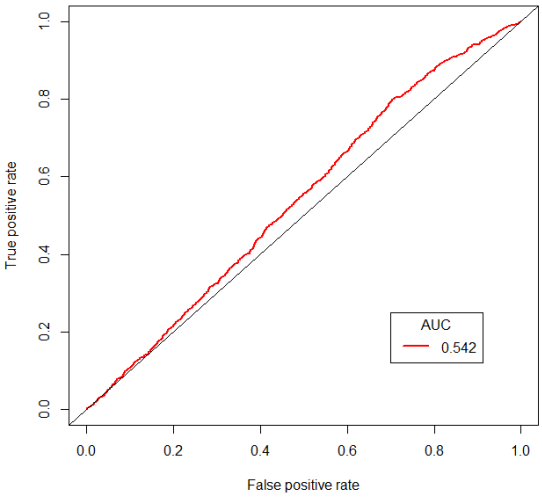


Figure: 4.13 Roc curve of SVM : Radial Basis Function Kernel (9)

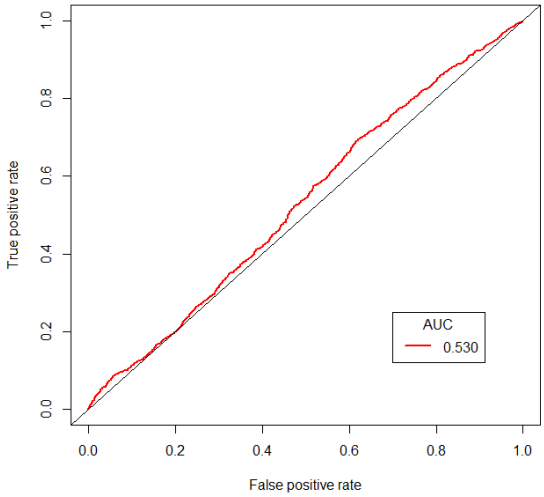


Figure: 4.14 Roc curve of SVM : Radial Basis Function Kernel (18)

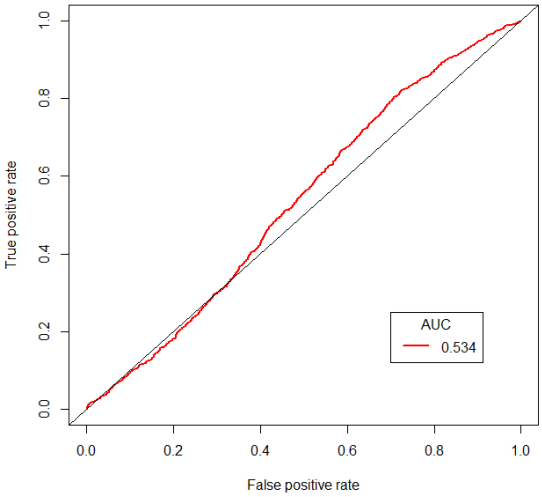


Figure: 4.15 Roc curve of SVM : Radial Basis Function Kernel (38)

4.5 Testing of Random Forest model

A random forest- implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.

A random forest algorithm and cross-validation involve randomness. Thus, a typical 5-fold CV in one pass leads to certain variable significance values, and another run (with different data partitioning) leads to different values of the variable.

In the parameter, we transfer the number of components which we want to save (I have chosen 38, 18, 9 - as taking them into account results in models that practically do not differ from the results of the basic data).
It is time to define models of the classification. Let's take several various algorithms and compare the results of their work by means of the characteristic Area Under ROC Curve (AUC). The ROC curve compares the rank of prediction and answer.

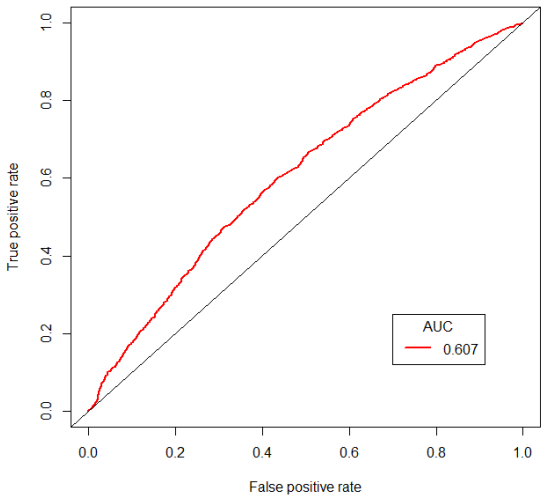


Figure: 4.16 Roc curve of RF (9)

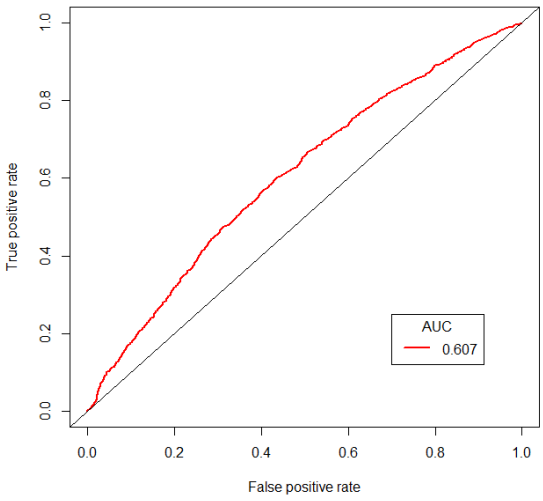


Figure: 4.17 Roc curve of RF (18)

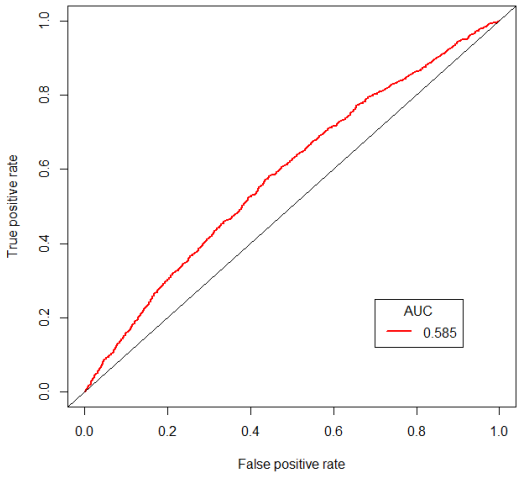
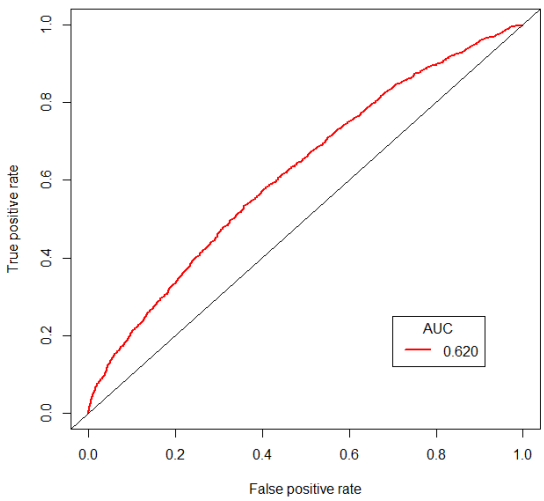


Figure: 4.18 Roc curve of RF (for 38 features)

4.6 Testing of artificial neural network



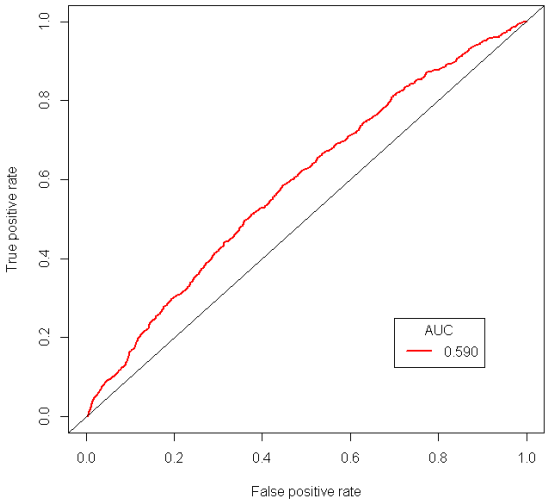


Figure: 4.19 Roc curve of NN (for 9 features) Figure: 4.20 Roc curve of NN (for 18 features)

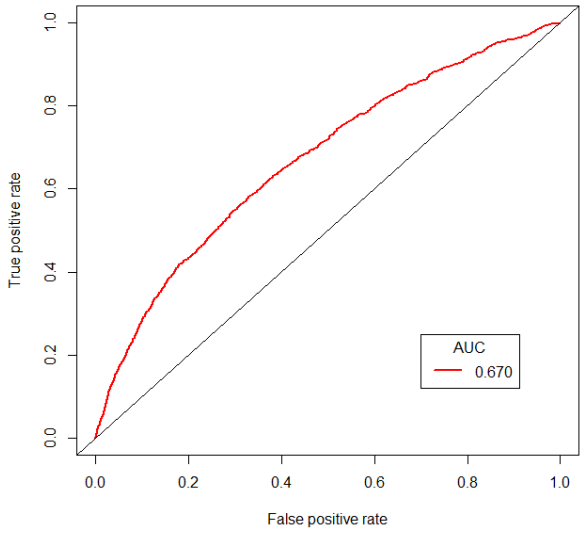


Figure: 4.21 Roc curve of NN (38)

4.7 Results of testing models

Having considered all the major models, we can make a conclusion and see which of the models has shown the best result. The following picture shows the four models with best results.

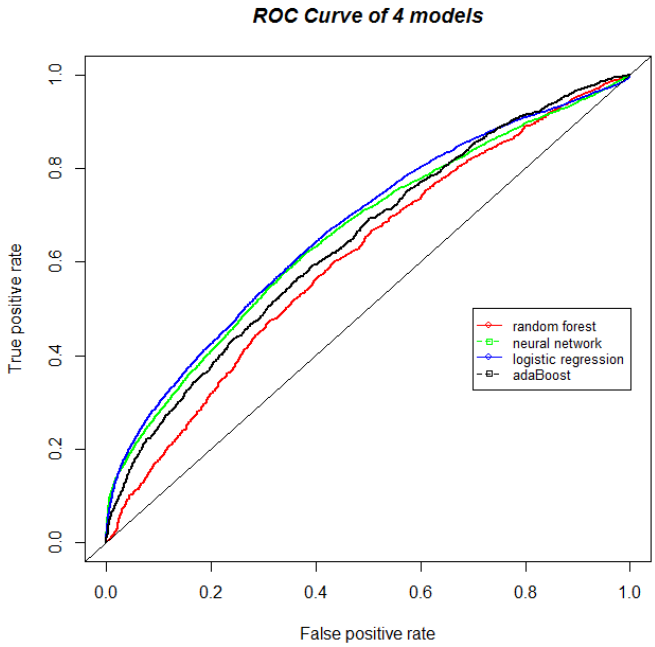


Figure: 4.22 Roc curve of 4 models using PCA (for 9 features)

Learning **Roc curve of model**	**Error.cv (9)**	**Error.cv (18)**	**Error.cv (38)**	**Roc curve of model (9)** (Using PCA)	**Roc curve of model (18)** (Using PCA)	**Roc curve of model (38)** (Using PCA)
Logistic Regression	0.120	0.110	0.121	0.607(+-0.070)	0.620(+-0.023)	0.669(+-0.003)
AdaBoosting	0.111	0.110	0.113	0.644(+-0.010)	0.594(+-0.003)	0.675(+-0.003)
Nearest Neighbors	0.052	0.122	0.123	0.561(+-0.012)	0.604(+-0.003)	0.610(+-0.003)
Random Forest	0.109	0.109	0.126	0.607(+-0.009)	0.607(+-0.002)	0.585(+-0.005)
Neural Network	0.103	0.109	0.101	0.620(+-0.013)	0.590(+-0.070)	0.670(+-0.003)
SVM(Linear Kernel)	0.116	0.116	0.116	0.442(+-0.020)	0.448(+-0.023)	0.608(+-0.013)
SVM(Radial Basis Function Kernel)	0.116	0.116	0.116	0.542(+-0.010)	0.530(+-0.013)	0.534(+-0.013)

Table 11 Analysis of models (Using PCA)

In the first case, when we use PCA for preprocessing, we obtained the following results. One can see in the final table that a Logistic Regression, AdaBoosting and Neural Network (NN) method proved itself to be the best.

In the second case, when we use RF for preprocessing, we obtained the following results.

Learning Roc curve of model	Error.cv (9)	Error.cv (18)	Error.cv (38)	Roc curve of model (9) (Using RF)	Roc curve of model (18) (Using RF)	Roc curve of model (38) (Using RF)
Logistic Regression	0.105	0.104	0.107	0.623(+-.0001)	0.638(+-0.001)	0.672(+-.0004)
AdaBoosting	0.104	0.103	0.104	0.636(+-0.001)	0.643(+-.0003)	0.688 (+-0.001)
Nearest Neighbors	0.110	0.106	0.108	0.571(+-0.013)	0.607(+-0.004)	0.604(+-0.006)
Random Forest	0.117	0.104	0.114	0.599(+-0.008)	0.632(+-0.012)	0.630 (+-0.008)
Neural Network	0.105	0.110	0.111	0.624(+-0.002)	0.622(+-0.001)	0.635 (+-0.002)
SVM(Linear Kernel)	0.115	0.113	0.120	0.416(+-0.001)	0.551(+-0.001)	0.507(+-0.001)
SVM(Radial Basis Function Kernel)	0.115	0.113	0.119	0.540(+0-.001)	0.452(+-0.002)	0.560(+-0.006)

Table 12 Analysis of models (Using Random Forest)

One can see in the final table that a Logistic Regression, AdaBoosting, Neural Network (NN) and Random Forest (RF) method proved itself to be the best.

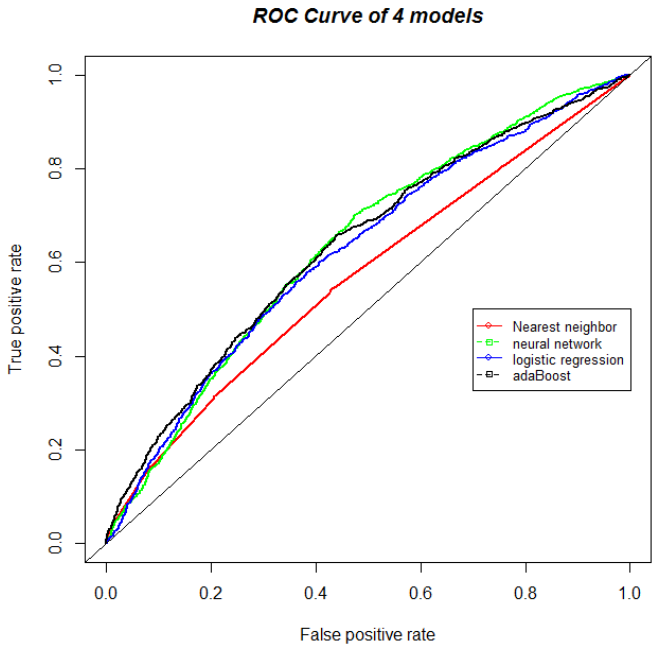


Figure: 4.23 Roc curve of 4 models using RF (for 9 features)

Now, when we have tested all models in two different ways, we can improve the result. We need to choose the models with the best results and test them with the optimal number of parameters.

We tested all the models in two ways and chose the best results. After testing, the best were the following models:

- Logistic Regression using PCA, (38)
- Random Forest using RF, (18)
- Neural Network using RF, (9)

Furthermore, we will try to test these models with the better set of parameters to get the better res ults.
For a logistic regression model we tried to add a set of parameters:

Parameters	Description
method	a string specifying which classification or re gression model to use. Possible values are f ound using names(getModelInfo()). A list of functions can also be passed for a custom m odel function
preProcess	a string vector that defines a pre-processing of the predictor data. Current possibilities ar e "BoxCox", "YeoJohnson", "expoTrans", " center", "scale", "range", "knnImpute", "bag Impute", "medianImpute", "pca", "ica" and "spatialSign". The default is no pre-processi ng. See preProcess and trainControl on the p rocedures and how to adjust them. Pre-proc essing code is only designed to work when x is a simple matrix or data frame
trControl	a list of values that define how this function acts
maximize	a logical: should the metric be maximized or minimized. A logical recycled from the function arguments.
tuneLength	an integer denoting the amount of granularit y in the tuning parameter grid. By default, t his argument is the number of levels for eac h tuning parameters that should be generate d by train.

Table 13 Advanced options for glm

Parameters "method", "preProcess", "trControl", we cannot change, since values of these parameters are standard:

method =”glm” (logistic regression model)
preprocess=”pca” (PCA)
trControl= trainControl(preProcOptions = list(pcaComp = 38))

But we can change the values of the two parameters that as to obtain a higher accuracy. We tried several values for the each parameter:

Parameter	Values
tuneLength	1, 5, 10, 20,30, 50, 100

Having compared the results, we have got an AUC of 0,68. with values of the parameters:

- maximize=TRUE;
- tuneLength=10.

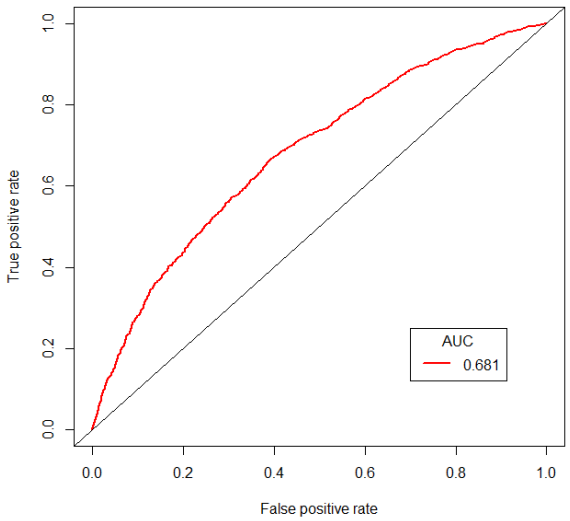


Figure: 4.24 Roc curve of GLM (for 38 features)

For the neural network model we tried to add a set of parameters:

Parameters	**Description**
linout	switch for linear output units. Default logistic output units.
maxit	maximum number of iterations. Default 100.
tuneGrid	an integer denoting the amount of granularity in the tuning parameter grid. By default, this argument is the number of levels for each tuning parameters that should be generated by train
preProcess	a string vector that defines a pre-processing of the predictor data. Current possibilities are "BoxCox", "YeoJohnson", "expoTrans", "center", "scale", "range", "knnImpute", "bagImpute", "medianImpute", "pca", "ica"
size	number of units in the hidden layer. Can be

	zero if there are skip-layer units.
decay	parameter for weight decay. Default 0.

Table 14 Advanced options for NN

For the neural networks model we tried to manipulate with the following parameters:

- maxit (1,10,20,30,50,100);
- size (1,2,5,10,20, c(1,5,10), c(1,2,3), c(1,2,5));
- decay(0.5, 1), c(0.001, 0.01, 0.1), c(0.01, 0.1)).

The highest accuracy was obtained using the following set of parameters:

- maxit=30;
- size=c(1,5,10);
- decay=c(0.5, 0.1);

Having compared the results, we have got the best values of 0,64.

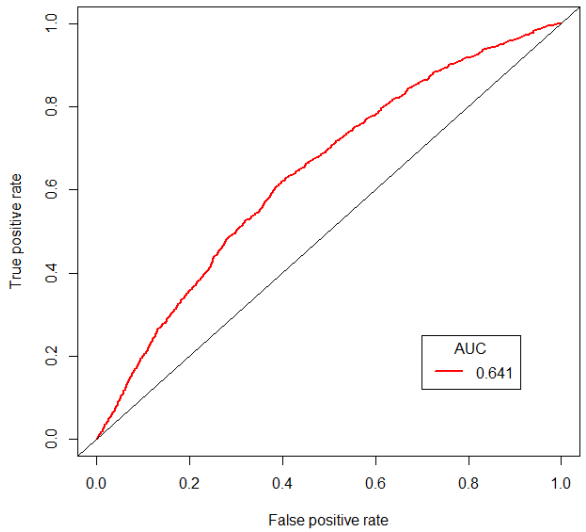


Figure: 4.25 Roc curve of nnet (for 9 features)

For the random forest model we tried to add a set of parameters:

Parameters	Description
method	a string specifying which classification or re gression model to use. Possible values are f ound using names(getModelInfo()). A list of functions can also be passed for a custom m odel function
preProcess	a string vector that defines a pre-processing of the predictor data. Current possibilities ar e "BoxCox", "YeoJohnson", "expoTrans", " center", "scale", "range", "knnImpute", "bag Impute", "medianImpute", "pca", "ica" and

	"spatialSign". The default is no pre-processi ng. See preProcess and trainControl on the p rocedures and how to adjust them. Pre-proc essing code is only designed to work when x is a simple matrix or data frame
trControl	a list of values that define how this function acts
tuneLength	an integer denoting the amount of granularit y in the tuning parameter grid. By default, t his argument is the number of levels for eac h tuning parameters that should be generate d by train.
tuneGrid	an integer denoting the amount of granularity in the tuning parameter grid. By default, this argu ment is the number of levels for each tuning par ameters that should be generated by train
ntree	Number of trees to grow

Table 15 Advanced options for rf

For the random forest model we tried to change the value of a parameter "ntree":

- ntree (5, 10, 50, 100, 200, 500, 700)

Having compared the results, we have got the best values of 0.66, for the parameter value ntree number of decision trees 500.

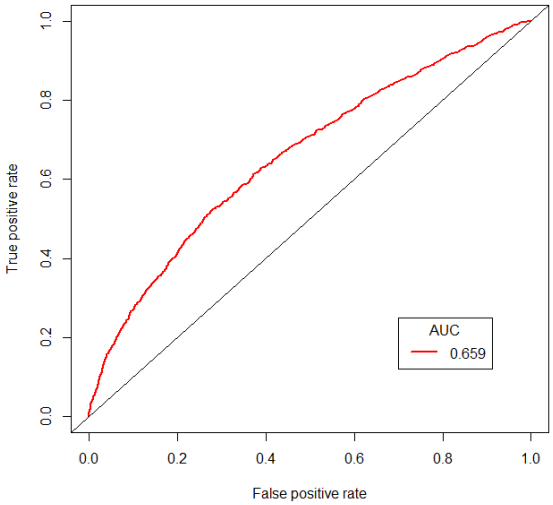


Figure: 4.25 Roc curve of RF (for 18 features)

Chapter 5

Construction of Alternative Scoring Model

In this chapter we will use two independent data bases for comparison of accuracy of the results. In the previous chapter, we tested different parameter values for each from the three selected models. In this chapter we can test two data bases with the optimal set of parameter values and compare the accuracy results of models constructed of two databases with a primary database.

5.1 Testing data from Kaggle portal

Input Data

In this chapter we used another independent data base. We took this data from Kaggle open portal [17]. Kaggle is a platform for competition in data science. Historic data are presented on 250,000 sets of data. We tried to analyze this data in the same ways as described in previous chapters of this paper.

The primary source of data is the client's questionnaire data at the moment of filing the credit application – social and demographic indices, data on the requested credit, financial indices, current balance, and preceding bank operations.

Description of the format of the data set for the second database:

Name and description
ID
Serious Dlqin 2 yrs
Revolving Utilization Of Unsecured Lines
age
Number Of Time 30-59 Days Past Due Not Worse
Debt Ratio
MonthlyIncome
Number Of Open Credit Lines And Loans
Number Of Times 90 Days Late
Number Real Estate Loans Or Lines
Number Of Time 60-89 Days Past Due Not Worse
Number Of Dependents

Data Preparation

As a rule, data includes blank spaces. This is due to various reasons: an error at data input made by the bank clerk, or the client refused to answer the question while filling in the application.
At this stage, the analyst decides what action should be taken in regards to the blank spaces. In practice, if blank spaces take up over 5%, data should be analyzed thoroughly – a blank space may indicate absence (for example, absence of dependants, apartment or cell phone), the client may consciously not indicate some data. In these cases, the blank space should be replaced with a value that was not spotted in the data and include it in the analysis.
In this database, in initial data strings with omitted values were spotted for almost every variable. Strings with blanks were removed as their number was insignificant.

Model Selection and Train/Validation/Test Sets

Normally, train selection is 70 - 80% of the entire selection. The rest is test selection. Test selection is used to check the accuracy of the constructed model. It should be noted that selections are formed randomly.

Scoring Model Construction

The resulting scoring map should have enough attributes for uninterrupted performance. To reveal the most meaningful features, two basic methods have been used:

- PCA;
- RF;

After having tested both methods, we can apply those models of machine tutoring, that showed the best results in work with the first database, namely:

- Logistic Regression: maximize= TRUE; tuneLength=100;
- Random Forest: ntree=100;
- Neural Network: maxit=10; size=c(1,5,10); decay=c(0.5, 1);

Now, we can compare the results of construction of alternative database models and primary databases that was used in previous chapters of the tesis.

Model Quality Evaluation

Based on the obtained results, models, which were built for alternative databases also have good prognostic accuracy.

Learning Roc curve of model	**Roc curve of model (4) (Using PCA)**	**Roc curve of model (10) (Using PCA)**	**Roc curve of model (4) (Using RF)**	**Roc curve of model (10) (Using RF)**	**Error.cv (4)**	**Error.cv (10)**

Logistic Regression	0.616(+-0.001)	0.667(+-0.002)	0.616(+-.009)	0.697(+-0.07)	0.064	0.060
Neural Network	0.703(+-0.01)	0.818(+-0.01)	0.670(0.001)	0.807(+-0.003)	0.055	0.055
Random Forest	0.784(+-0.1)	0.748 (+-0.12)	0.696(0.007)	0.806(0.14)	0.058	0.076

Table 16 Analysis of models (Alternative database)

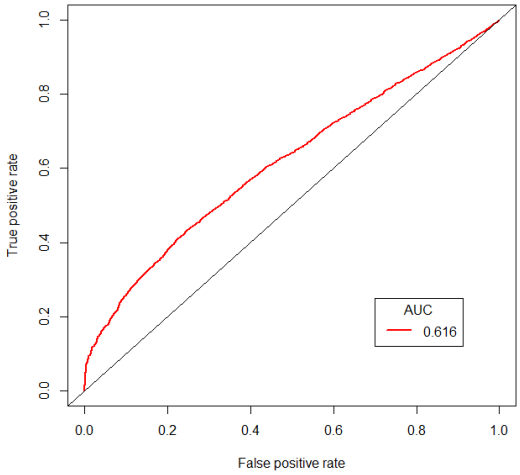


Figure: 5.1 PCA: Roc curve of glm (4 f.)

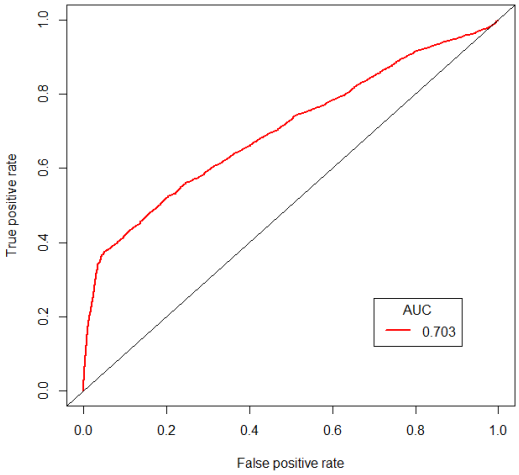


Figure: 5.2 PCA: Roc curve of nnet (4 f.)

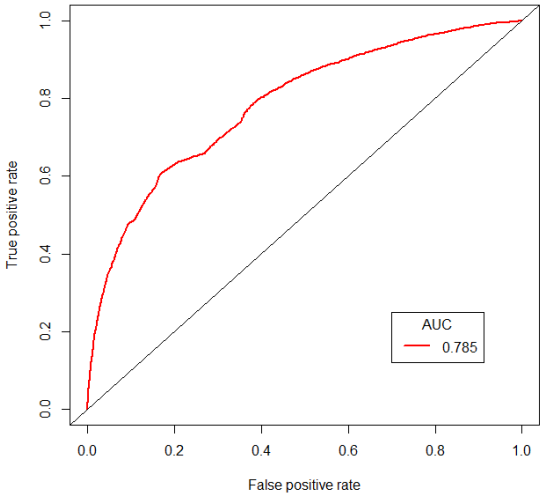


Figure: 5.3 PCA: Roc curve of RF (for 4 f.)

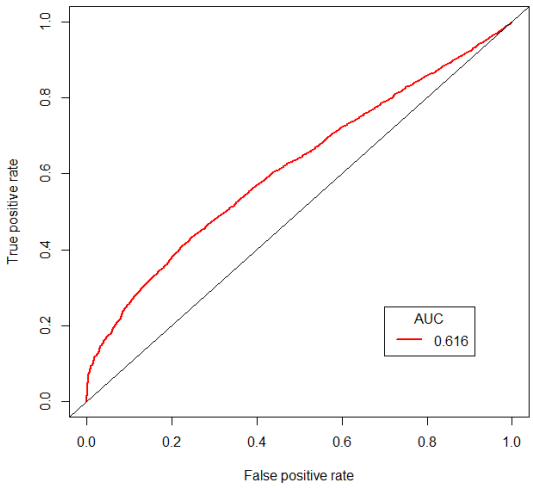


Figure: 5.4 RF: Roc curve of glm (4 f)

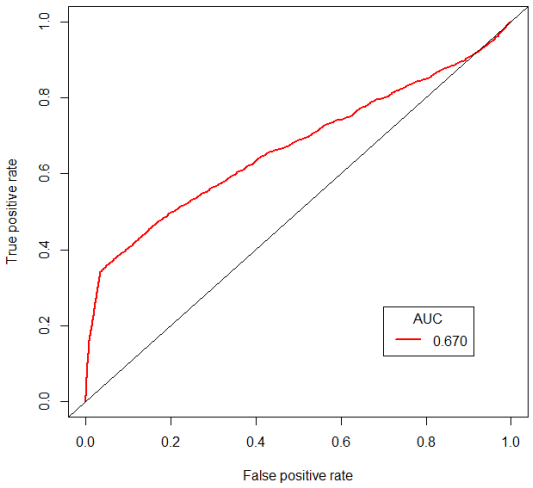


Figure: 5.5 RF: Roc curve of nnet (4 f)

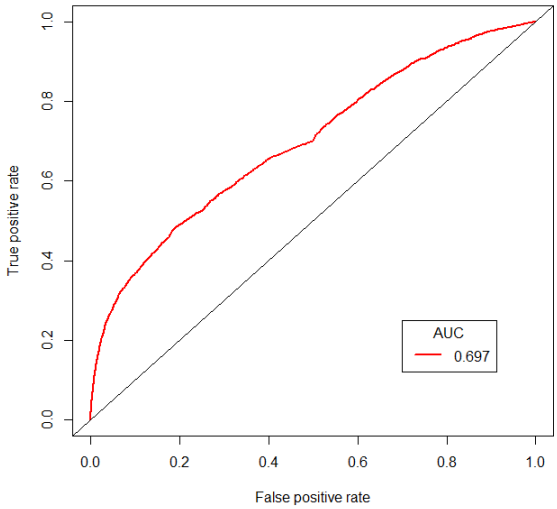
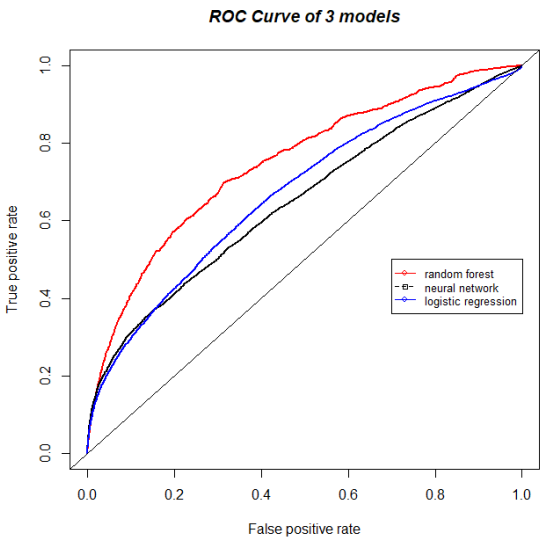


Figure: 5.6 RF: Roc curve of RF (4 f.)



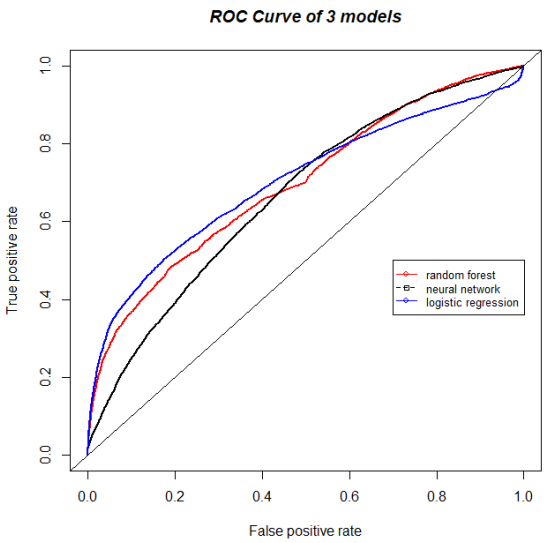


Figure: 5.7 PCA: Roc curve of 3 models (10 f.) Figure: 5.8 RF: Roc curve of 3 models (10 f.)

We tested all the models in two ways and chose the best results. After testing, the best were the following models:

- Random Forest using
- Neural Network using

5.2 Testing data from Lending Club Company

Lending Club [18] is a US peer-to-peer lending company. Lending Club operates an online lending platform that enables borrowers to obtain a loan, and investors to purchase notes backed by payments made on loans. It is the world's largest peer-to-peer lending platform.
Lending Club enables borrowers to create unsecured personal loans between $1,000 and $40,000.
Investors can search and browse the loan listings on the Lending Club website and select loans that they want to invest in based on the information supplied about the borrower, the amount of loan, loan grade, and loan purpose. Investors make money from interest.
Lending Club also makes traditional direct to consumer loans. The loans are not funded by investors but are assigned to other financial institutions.

Data Preparation

Before we start building scoring models, we need to transform and edit the data so that we can use it for analysis. Data are presented on 188,000 clients are available. A database is an assortment of data consisting of 129 features for each client. Most of the features we do not need, since they do not affect the accuracy of the result. The blank space should be replaced with a value and included in the analysis. After preliminary analysis of the data, we removed unnecessary features.

Description of the set of the dataset for the third database (after editing):

Name	Description
loan_status	Current status of the loan
funded_amnt	The total amount committed to that loan at that point in time.
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
installment	The monthly payment owed by the borrower if the loan originates.
home_ownership	The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER
annual_inc	The self-reported annual income provided by the borrower during registration.
verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified
mths_since_last_delinq	The number of months since the borrower's last delinquency.
revol_bal	Total credit revolving balance
total_pymnt	Payments received to date for total amount funded
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
total_rev_hi_lim	Total revolving high credit/credit limit
avg_cur_bal	Average current balance of all accounts
bc_util	Ratio of total current balance to high credit/credit limit for all bankcard accounts.
delinq_amnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
mort_acc	Number of mortgage accounts.
num_sats	Number of satisfactory accounts
num_tl_90g_dpd_24m	Number of accounts 90 or more days past due in last 24 months
tot_hi_cred_lim	Total high credit/credit limit
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years

Scoring Model Construction

The resulting scoring map should have enough attributes for uninterrupted performance. To reveal the most meaningful features, two basic methods have been used:

- PCA;
- RF;

After having tested both methods, we can apply those models of machine tutoring that showed the best results in work with the first database, namely:

- Logistic Regression: maximize= FALSE; tuneLength=10;
- Random Forest: ntree=50;
- Neural Network: maxit=30; size=c(1,5,10); decay=c(0.5, 1);

Now, we can compare the results of construction of alternative database models and primary databases that was used in previous chapters of the tesis.

Model Quality Evaluation

Based on the obtained results, models, which were built for three database also have good prognostic accuracy.

Learning Roc curve of model	**Roc curve of model (7) (Using PCA)**	**Roc curve of model (20) (Using PCA)**	**Roc curve of model (7) (Using RF)**	**Roc curve of model (20) (Using RF)**	**Error.cv (7)**	**Error.cv (20)**
Logistic Regression	0.722(+-0.001)	0.726(+-0.002)	0.727(+-.009)	0.697(+-0.07)	0.145	0.069
Neural Network	0.735(+-0.04)	0.760(+-0.03)	0.723(0.02)	0.752(+-0.03)	0.143	0.073
Random Forest	0.707(+-0.002)	0.718 (+-0.12)	0.718(0.002)	0.746(0.14)	0.147	0.083

Table 17 Analysis of models (data from LendingClub)

The highest results were shown by the Neural Network models.

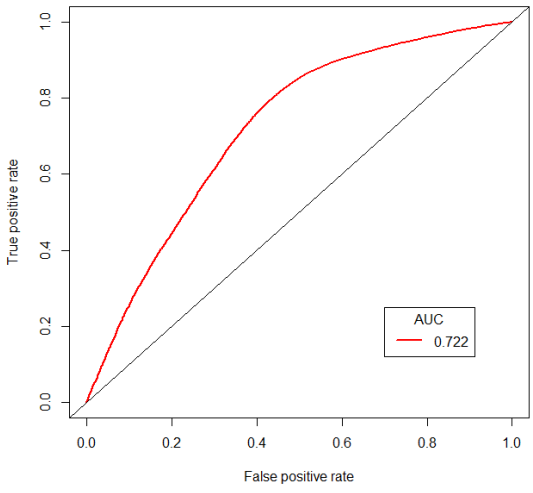


Figure: 5.9 PCA: Roc curve of glm (7 f)

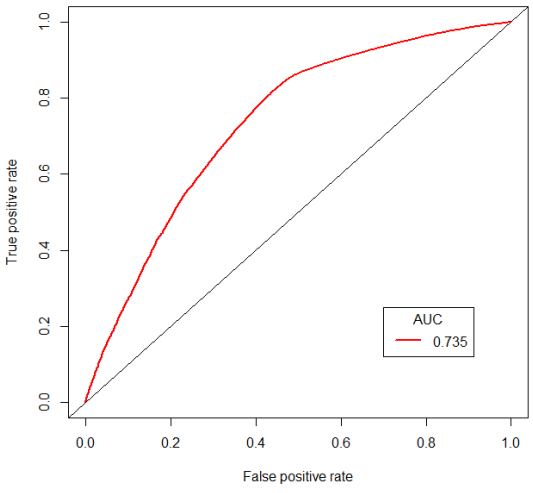


Figure: 5.10 PCA: Roc curve of nnet (7 f)

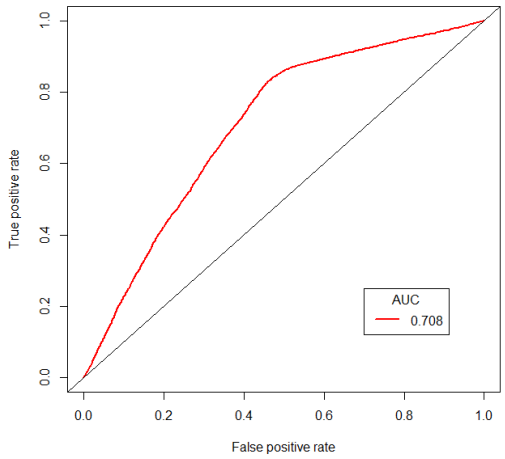


Figure: 5.11 PCA: Roc curve of rf (7 f)

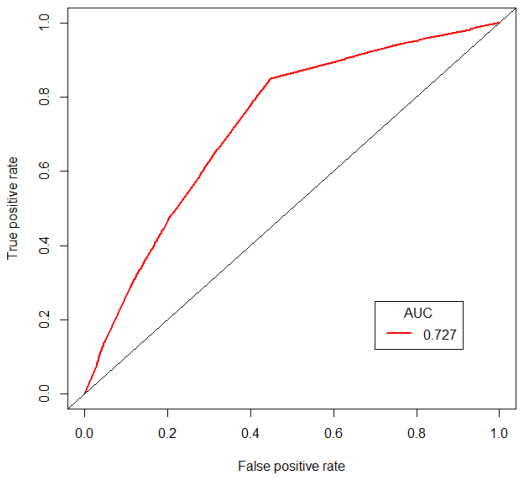


Figure: 5.12 RF: Roc curve of glm (7 f)

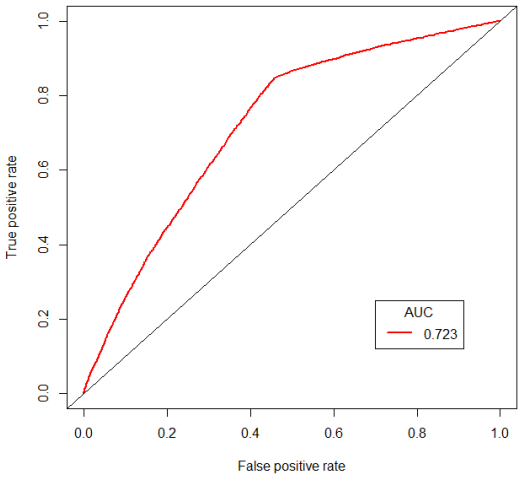


Figure: 5.13 RF: Roc curve of glm (7 f)

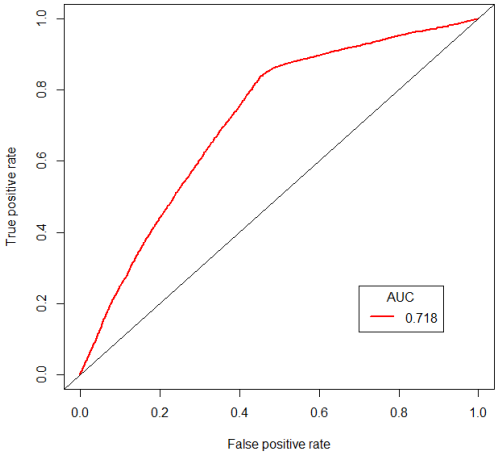
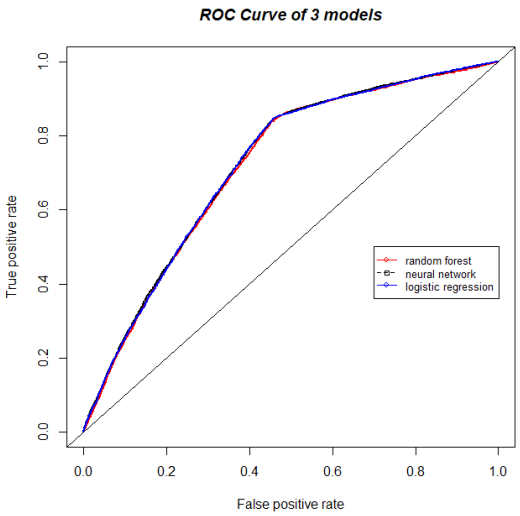


Figure: 5.14 RF: Roc curve of rf (7 f)



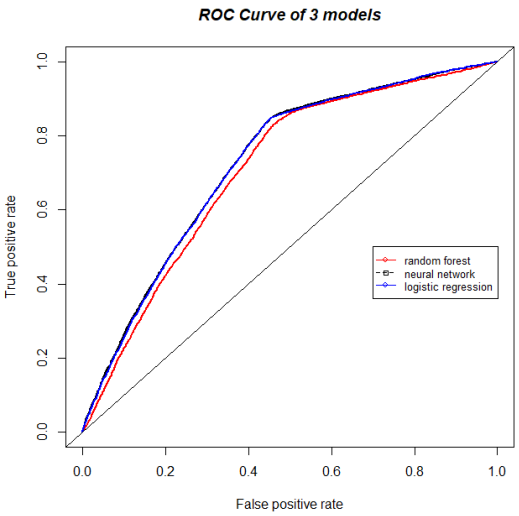


Figure: 5.15 RF: Roc curve of 3 models (7 f.) Figure: 5.16 PCA: Roc curve of 3 models (7 f.)

We tested all the models in two ways and chose the best results. After testing, the best were the following models:

- Logistic Regression using
- Neural Network using

Chapter 6

User Guide

The Work Project offers tools for the data visualization of CSV documents, a generation of analyzed data, building data models and a selection of models with the best results.
The following sections describe in detail “what” and “how” you can achieve.
The project consists of two parts. The first part is the analysis of data (ReadCsvFile), the second part is construction and the analysis of models machine learning (CreditScoring).

6.1 ReadCsvFile project

In this project we analyze data which was provided by a bank. We need to prepare data, for the second project.

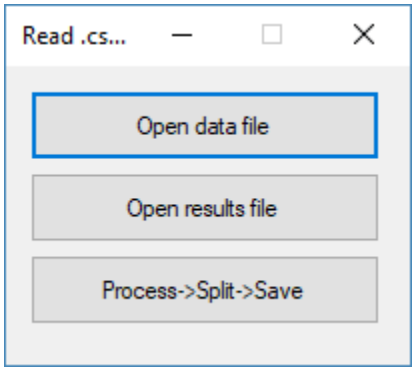


Figure 6.1: Main Window (ReadCsvFile)

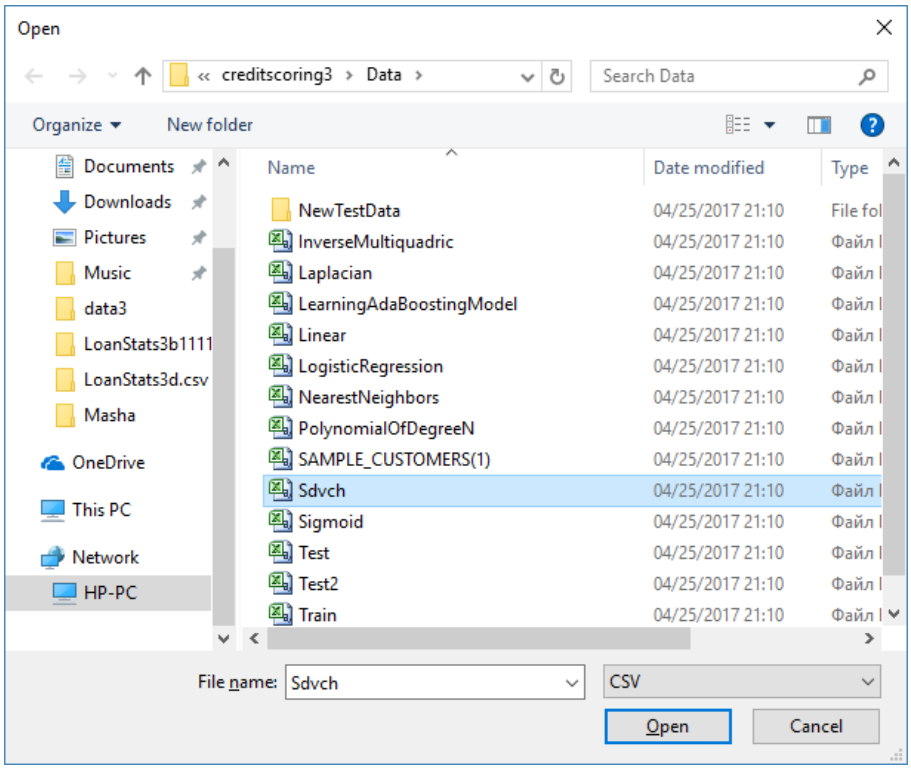


Figure 6.2: Open data file (ReadCsvFile)

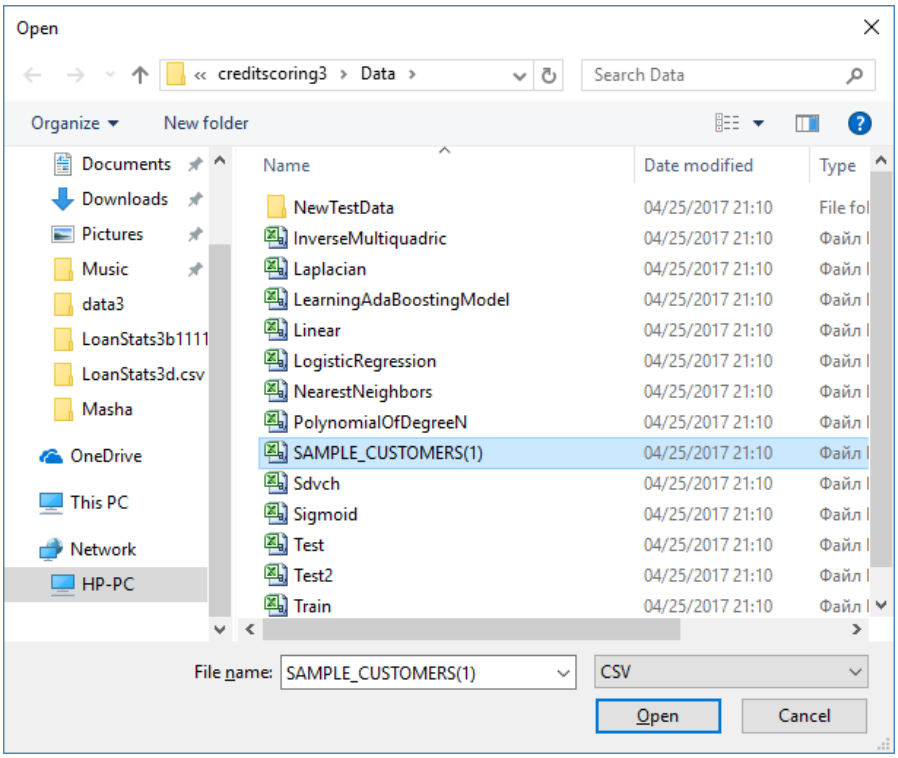


Figure 6.3: Open result file (ReadCsvFile)

It contains the following parts:

- ***Open data file*** - here you can upload your first document. It is the database offered by the bank (Sdvch.csv). This document, contains all data on each client.
- ***Open result file*** - here you can upload your second document. It is the database offered by the bank (Sdvch.csv). This document contains information on test and resulting samples for each client.
- ***Process->Split->Save*** - this is the main part of the program. In this part of the program there is an analysis of data, connection with the document Sdvch.csv and saving of new documents: Train.csv, Test.csv.

6.2 CreditScoring project

In this project we will work with the data that was prepared by the *ReadCsvFile* project. In the *CreditScoring* project we build models of data. For each model we calculate and draw the *ROC* - curve, and also we consider cross-validation.
Figure 6.4 shows the main window of the program (*CreditScoring*). It contains the following parts:

- *Load File* - here we need to choose the training selection which was prepared by the *ReadCsvFile* project.
- *Run Learning* - runs learning Train.csv.
- *Run ROC* - after the program has finished learning, we need to construct models of data and to draw ROC- curve.
- *Run Cross validation* - after the program finished learning, we need to count cross-validation. By default there is the minimum parameter 2, i.e. comparison between two models. We can write any parameter from 2 to 8.
- *Run test* - runs learning Test.csv. We compare results to test selection.

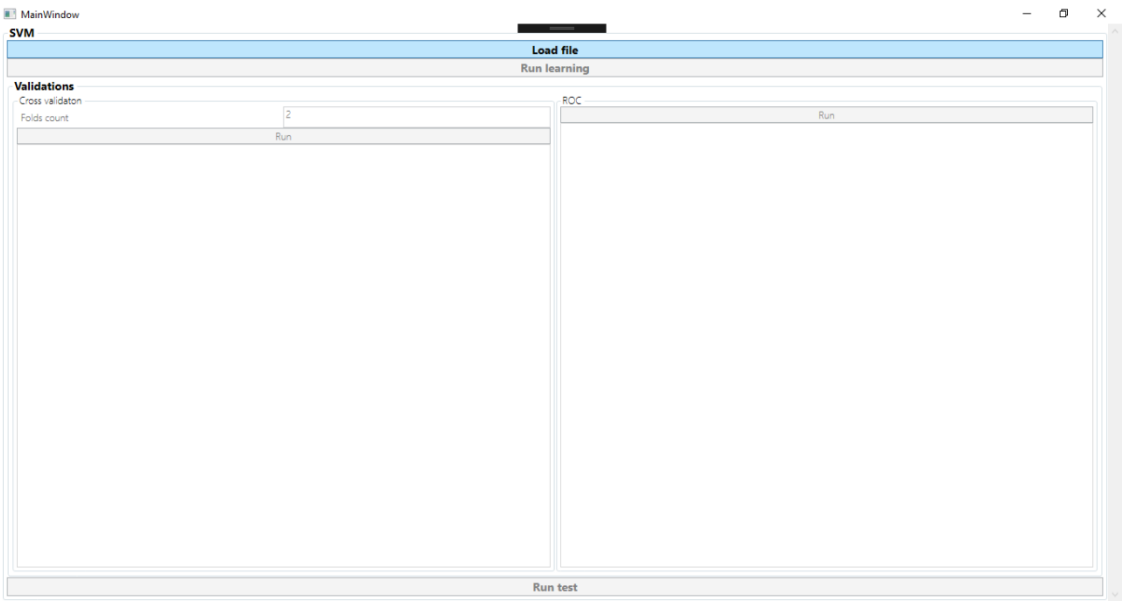


Figure 6.4: Main Window (*CreditScoring*)

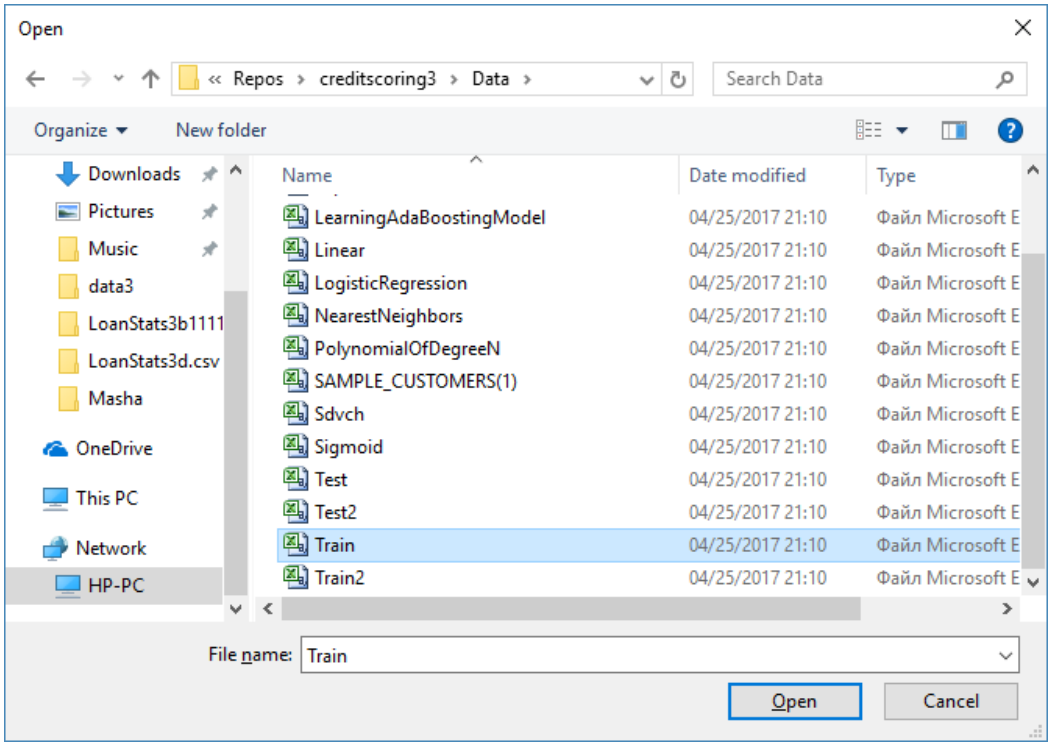


Figure 6.5: Load File (*CreditScoring*)

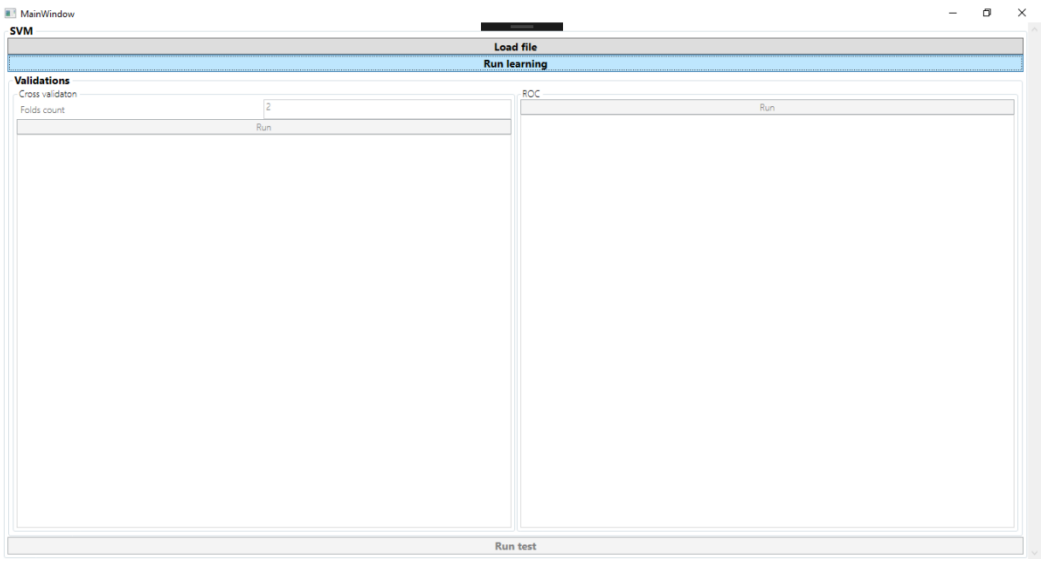


Figure 6.6: Run ROC (*CreditScoring*)

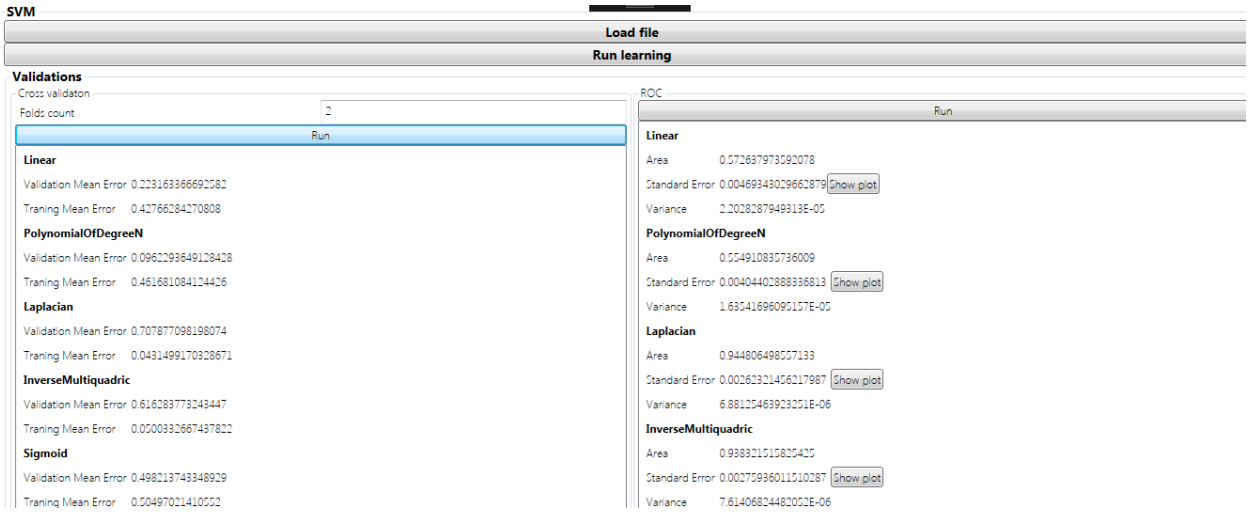


Figure 6.7: Run Cross Validation (*CreditScoring*)

6.3 RIntegration project

In this project we will work with the data that was prepared, trained and tested in R. This is an alternative version of the project CreditScoring. For each model we calculate the answer.
Figure 6.7 shows the main window of the program RIntegration. It contains the following parts:

- Load File - here we need to choose the model which was prepared by R.
- Teach - here we need to choose the client ID.
- Find - after the program has finished learning. We will see the results.

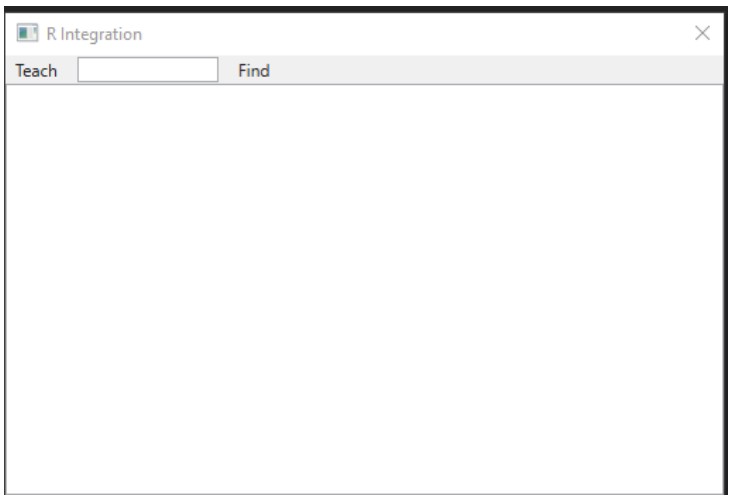


Figure 6.7: Main Window (*RIntegration*)

Conclusion

In our work, we tried to realize a methodology of building a banking scoring model for assessing the creditworthiness of individuals.

Determination of data

A scoring system for assessing a client's solvency is a statistical model that estimates the likelihood that a borrower will not pay their obligations on time. And, as is known, it's necessary to have a sufficient and high-quality database for constructing any statistical model. It is the lack of sufficient information on retail clients that is the main obstacle for financial institutions in building internal scoring models.
An optimal solution in this situation is the consolidation of databases of retail customers of several similar services and activities of banks in a single document, as well as construction of their scoring systems based on the common data of individuals of several banks.
In our work, we considered such data. Namely, as a basis, we took a database, which includes information from different banks regarding each client.
Database for building the scoring model should contain all possible information about clients for the last 2 - 5 years, including the client number, banking product, loan application decision, account opening date, debt status, account balance, etc.
Further, data should be broken down into categories: a "good" (solvent), "bad" (insolvent) client or "refusal" in repayment of a loan.

Scoring variables analysis

The next main stage in construction of a model is the selection and analysis of independent variables. The main source of data is the client's personal data at the time of filing a loan application.
Demographic characteristics: age, sex, nationality, place of residence, duration of residence in an actual place of residence, education, occupation, employment duration, availability of property, family situation, parental status, etc.
Requested loan data: loan purpose, total amount of a loan, financing schedule, initial payment, dimension ratio of loan, amount to amount of credit support, etc.
The next main information source is an internal credit banking history and information that was obtained from a credit history bureau at the time of application. Used scoring variables can be: number of current customer's accounts, number and availability of credit cards, total amount of all loans, time of last loan receipt, client's availability for other products of this financial institution, current account status, utilization of existing limits, credit bureau ratings, etc.
Analysis of scoring indicators began with checking their consistency and searching for possible errors.
Also, variable values were checked for extreme values, and, if they presented, they were removed from further analysis, or - variables were assigned average results of the group.
Correlation analysis is an important step in assessing scoring characteristics. All variables that were used in the model were checked for correlation between them.
The next step in the analysis of independent variables is checking their statistical significance (PCA). This analysis resides in verifying the presence and strength of the relationship between

one dependent and independent variable, which allows us to determine which variables are the most significant for further analysis and model building.

Scoring model building

Analysis of scoring variables allowed us to identify a number of the strongest and most qualitative characteristics (approximately 8 - 20 pieces), which the construction of a statistical model is based on.
For constructing a scoring model and solving classification problems, we applied various models of machine learning. The highest results were shown by the following models:

Logistic Regression model.
Random Forest.
Neural Network.

At the end of our work, we checked the results of the second database. We carried out the analysis of variables, calculated -the most significant variables and used those machine learning models that showed the best results while working with the first database.

During the course of our research, the work of a bank analyst in the sphere of scoring models construction has been studied. Several independent databases and methods of machine learning used by banks for scoring models construction have been analyzed. At the same time, all stages of scoring model elaboration process, starting from the preliminary data analysis and finishing with constructed model evaluation, have been carried out.

After all accomplished work, we concluded that PCA is the easiest and the fastest way to perform data analysis and identify the most important features. The Random Forest method takes more time.
When constructing and testing models, the most simple and stable model is the logistic regression model. This model avoids the need of a large number of parameters, has good running speed and has shown good results while testing all three databases. This model is more convenient to be used when working with large databases. In our work, the best results were shown by the model of neural networks. This model has more parameters and takes more time to test the data than the logistic regression model. The model of neural networks may be not very convenient for large databases.

Bibliography

1. T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning*. Springer, Berlin. 2009. ISBN: 978-0387848570.

2. G. James, D. Witten, T. Hastie and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer, Berlin. 2014. ISBN: 978-1-4614-7137-0

3. K. V. Vorontsov. *Combinatorial Substantiation of Learning Algorithms*. Dorodnitsyn Computing Center, Russian Academy of Sciences, Received January 30, 2004.

4. Hofmann, T., B. Schölkopf, and A. J. Smola. *Kernel methods in machine learning*. Institute of Mathematical Statistics, 2008, Vol. 36, No. 3, 1171–1220 DOI: 10.1214/009053607000000677.

5. Davis J., Goadrich M. *The Relationship Between Precision-Recall and ROC Curves*. ACM New York, NY, USA 2006. ISBN:1-59593-383-2

6. Ben-Hur, Asa, Horn, David, Siegelmann, Hava, and Vapnik, Vladimir. *Support vector clustering*. The Journal of Machine Learning Research.Volume 2, 3/1/2002 Pages 125-137. ISSN: 1532-4435.

7. Arlot, Sylvain, and Alain Celisse. *A survey of cross-validation procedures for model selection.* eprint arXiv:0907.4728. DOI:10.1214/09-SS054

8. Wei Gao, Zhi-Hua Zhou. *On the doubt about margin explanation of boosting*. Journal Artificial Intelligence. Elsevier Science Publishers Ltd. Essex, UK. doi>10.1016/j.artint.2013.07.002.

9. David A. Freedman. *Statistical Models: Theory and Practice.* Cambridge University Press. 2009. ISBN-10: 0521743850.

10. Russell S., Norvig, P. *Artificial Intelligence: A Modern Approach, 2nd ed.* Prentice Hall Series in Artificial Intelligence.2003. ISBN 978-0137903955.

11. Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining: Concepts and Techniques, 3ED.* The Morgan Kaufmann Series in Data Management. USA.2013. ISBN-10: 9380931913.

12. D. C. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber. *Deep, Big, Simple Neural Nets for Handwritten Digit Recognition*. Neural Computation, Volume 22, Number 12, December 2010. ISSN 0899-7667.

13. Li, Xiao-Lin, and Yu Zhong. *An overview of personal credit scoring: techniques and future work*. Journal: International Journal of Intelligence Science ISSN 2163-0283. 2012.

14. Lyn C. Thomas. *A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers*. International Journal of Forecasting, 16, (2), pp. 149-172. Netherlands. 2000. DOI: 10.1016/S0169-2070(00)00034-0.

15. West, Jarrod, Maumita Bhattacharya. *Some Experimental Issues in Financial Fraud Detection: An Investigation.* IEEE 2015. ISBN: 978-1-5090-1893-2.

16. Jae Kwon Bae, Jinhwa Kim. *A Personal Credit Rating Prediction Model Using Data Mining in Smart Ubiquitous Environments*. International Journal of Distributed Sensor Networks. DOI: 10.1155/2015/179060.

17. Montoya, Anna. *Kaggle Kernels: A New Name for "Scripts*".2016. http://blog.kaggle.com/author/annamontoya/. Last visited the page 15.07.2017

18. *Interest Rates and Fees on Lending Club & Prosper Loans*. LendingMemo. 2014-04-30. Retrieved 2017-03-28. http://blog.kaggle.com/author/annamontoya/. Last visited the page 18.07.2017.

19. Case study: *Tinkoff Credit Systems Bank – One of a kind*. IBS Intelligence. 8 March 2013. Retrieved 22 July 2016. https://ibsintelligence.com/ibs-journal/ibs-news/c381-ibsj-archive/c483-ibs-journal-archive-2013/case-study-tinkoff-credit-systems-bank-one-of-a-kindl/. Last visited the page 18.07.2017.

20. Abraham, A. *Meta-Learning Evolutionary Artificial NeuralNetworks*. Neurocomputing Journal, Vol. 56c, Elsevier Science, Netherlands, (1–38). 2004. DOI: 10.1016/S0925-2312(03)00369-2.

List of Tables

Attachments

Content of the attached CD

/SourceCode/	Contains the complete source code in the form of solution in Visual Studio 2015
/Documentation/	Contains HTML documentation generated from comments in the source code
/Release/	Contains the installer of project with all required dependencies
/Thesis/	Text of this thesis in PDF format