### Исследование методов прогнозирования с помощью среды R. Линейная регрессия

##### Обучение модели.

Линейная регрессия (Linear regression model) является самой простой моделью, используемой в задачах классификации и регрессии. В R она входит в базовый пакет и не требует дополнительных загрузок.

Табл. 1. Структура данных

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Описание** | **Тип** |
| SeriousDlqin2yrs (целевая переменная) | Имеет ли клиент 90 или более дней просрочки по выплатам | Да/Нет |
| RevolvingUtilizationOfUnsecuredLines | Общий остаток по кредитным картам за исключением недвижимого имущества и автокредитования, делённый на сумму кредитных лимитов | percentage |
| Age | Возраст клиента в годах | integer |
| NumberOfTime30-59DaysPastDueNotWorse | Количество раз, когда клиент просрочивал на 30-59 дней в последние 2 года | integer |
| DebtRatio | Месячные выплаты долгов, алиментов, прожиточного минимума делённые на месячный валовый доход | percentage |
| MonthlyIncome | Месячный доход | integer |
| NumberOfOpenCreditLinesAndLoans | Число открытых кредитов (рассрочка на покупку автомобиля или ипотека) и кредитные линии (например, кредитные карты) | integer |
| NumberOfTimes90DaysLate | Количество раз, когда клиент просрочивал выплаты на 90 дней и более | integer |
| NumberRealEstateLoansOrLines | Количество ипотечных кредитов и кредитов на объекты недвижимости, в том числе собственный капитал кредитных линий | integer |
| NumberOfTime60-89DaysPastDueNotWorse | Количество раз, когда клиент просрочивал на 60-89 дней в последние 2 года | integer |
| NumberOfDependents | Количество иждивенцев в семье (супруг, дети и т.д.) | integer |

Все данные cсоставляют 145000 наблюдений. Этот набор необходимо разбить случайным образом на три части (триплет):

* Тренировочный набор – значения целевой переменной известны. Необходим для обучения модели.
* Валидационный набор – значения целевой переменной также известны. Необходим для предварительной проверки качества модели.
* Тестирующий набор – значения целевой переменной необходимо предсказать, используя полученную ранее модель.

Прежде всего, необходимо загрузить данные в среду R:

|  |
| --- |
| A <- read.table("cs-test.csv",header=TRUE,sep=",")  A\_test<- read.table("test1.csv",header=TRUE,sep=";") |

Далее сформируем 3 набора данных:

|  |
| --- |
| A\_train<- A[sample(1:nrow(A), 80000, replace=FALSE),]  A\_validate<- A[sample(1:nrow(A), 30000, replace=FALSE),] |

Построить модель можно следующим образом:

*object <- lm(tg~.,data=A)*

*tg~.* – формула, показывающая, что ищется зависимость целевой переменной (или метки) от всех имеющихся признаков. Отметим, что формулу можно модифицировать, дополнив конкретными переменными. Например – *tg ~ v1+v2.*

*data = A* – выбор набора данных для тренировки модели.

|  |
| --- |
| object <- lm(A\_train$SeriousDlqin2yrs~.,data=A\_train) |

После выполнения кода можно посмотреть подробную информацию о полученной модели: summary*(object).*

|  |
| --- |
| Call:  lm(formula = A\_train$SeriousDlqin2yrs ~ ., data = A\_train)  Residuals:  Min 1Q Median 3Q Max  -0.84530 -0.08201 -0.05930 -0.03445 1.85556  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 1.274e-01 4.344e-03 29.322 < 2e-16 \*\*\*  X 6.629e-08 2.340e-08 2.833 0.004612 \*\*  RevolvingUtilizationOfUnsecuredLines -2.545e-06 3.359e-06 -0.758 0.448748  age -1.413e-03 7.060e-05 -20.018 < 2e-16 \*\*\*  NumberOfTime30.59DaysPastDueNotWorse 4.870e-02 1.456e-03 33.450 < 2e-16 \*\*\*  DebtRatio -3.049e-06 2.143e-06 -1.422 0.154904  MonthlyIncome -3.917e-07 8.116e-08 -4.827 1.39e-06 \*\*\*  NumberOfOpenCreditLinesAndLoans -7.402e-04 2.128e-04 -3.478 0.000506 \*\*\*  NumberOfTimes90DaysLate 5.023e-02 1.980e-03 25.372 < 2e-16 \*\*\*  NumberRealEstateLoansOrLines 2.509e-03 9.586e-04 2.618 0.008856 \*\*  NumberOfTime60.89DaysPastDueNotWorse -9.267e-02 2.208e-03 -41.972 < 2e-16 \*\*\*  NumberOfDependents 5.385e-03 8.834e-04 6.096 1.10e-09 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.2481 on 64096 degrees of freedom  (15892 observations deleted due to missingness)  Multiple R-squared: 0.0505, Adjusted R-squared: 0.05034  F-statistic: 309.9 on 11 and 64096 DF, p-value: < 2.2e-16 |

Основным элементом *object* является вектор регрессионных коэффициентов. Используя эти коэффициенты, можно вычислить решающую функцию (1):

 (1)

Решение о положительности клиента принимается согласно следующему правилу:

 (2)

где Δ – параметр усечения.

##### Валидация модели.

Оценить качество модели можно при помощи AUC. Area under ROC (Receiver Operating Curve) - это кривая, отображающая качество разделения двух классов. Большее значение AUC соответствует более высокому качеству разделения.

Для вычисления построения ROC кривых в среде R существует пакет “*pROC*”. Загрузка и подключение производится стандартным образом.

|  |
| --- |
| install.packages("pROC")  library(pROC) |

Построим прогноз для первых двух наборов данных:

*sol\_trn <- predict(object,newdata=A)*

Тренировочное и валидационное решения вычисляются при помощи следующих команд:

|  |
| --- |
| sol\_trn <- predict(object,newdata=A\_train)  sol\_vld<-predict(object,newdata=A\_validate) |

В наборах данных *A* и *V* нам известны значения целевой переменной, значит можно применить функции *auc()* для нахождения площади под ROC кривой. Выполним следующие команды:

|  |
| --- |
| S <- auc(A\_train[,2],sol\_trn)  Z <- auc(A\_validate[,2],sol\_vld) |

В качестве аргументов функции *auc()* мы указываем вектор меток, а также решение, качество которого необходимо оценить. Значение *AUC* получаем при помощи команды *print().*

> print(S)

Area under the curve: 0.6951

> print(Z)

Area under the curve: 0.6946

Полученные результаты говорят о качестве построенной модели – 0.69 является удовлетворительным результатом для линейной модели.

Чтобы посмотреть сам график нужно выполнить команду *plot():*

> plot(roc(A\_validate[,2],sol\_vld))

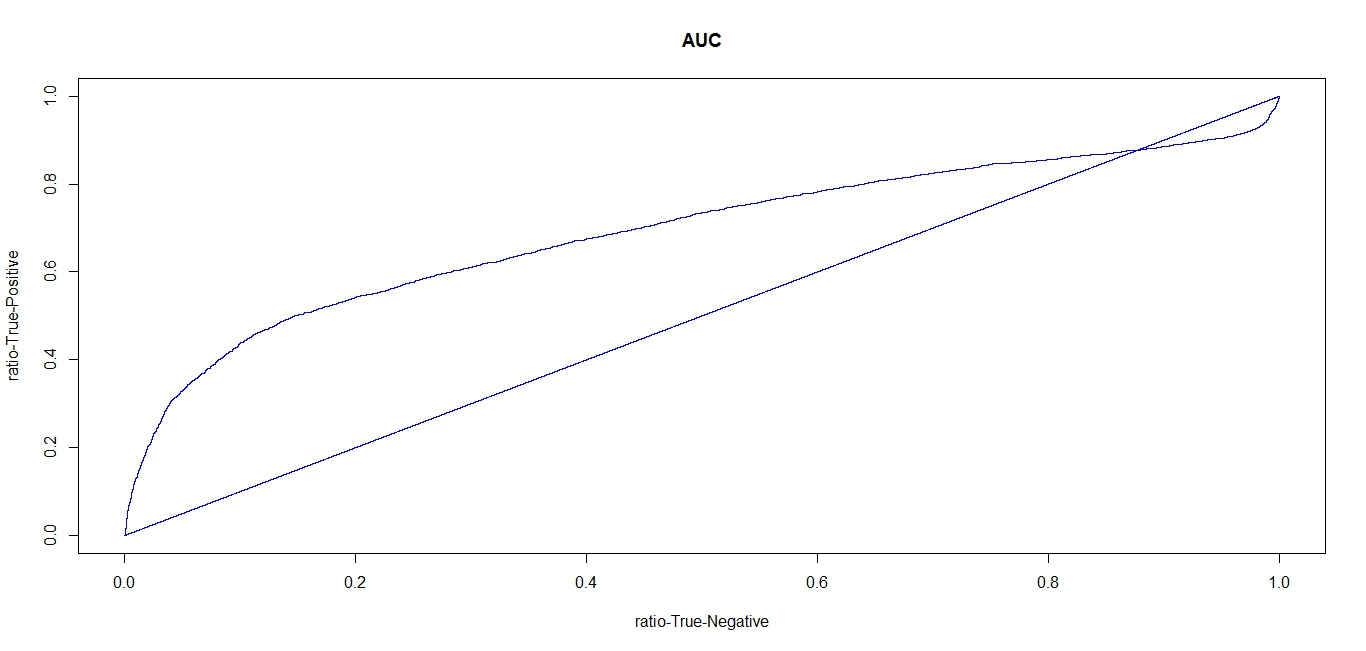


Рис. 2. ROC-кривая в случае модели линейная регрессия,

AUC = 0.69

##### Тестирование модели

Заключительный этап заключается в формировании прогноза для тестирующего набора данных, в котором значения целевой функции неизвестны.

|  |
| --- |
| sol\_tst<-predict(object,newdata=A\_test) |

Запись полученных решений в файл:

|  |
| --- |
| sol\_tst <- round(sol\_tst, digits=2)  write.table(sol\_tst, file = "E:/tst.txt", quote=FALSE, row.names=FALSE, col.names=FALSE, sep = ";") |

##### Аппроксимирующие и разделяющие функции потерь

По определению функция потерь характеризует потери при неправильном принятии решений на основе данных наблюдений. Существуют два основных класса функций потерь: сглаживающие (approximation) и разделяющие (separation).

В качестве примеров, рассмотрим две функции потерь – квадратичную (сглаживание) и экспоненциальную (разделение):

 (3)

Рассмотрим следующий иллюстрирующий пример, где *h* – решающая функция*:*

Табл. 2. Пример иллюстрирующих данных

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| y | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| h | -0.23 | -0.98 | -0.54 | -0.1 | 0.35 | 0.56 | 0.72 |
| 10\*h | -2.3 | -9.8 | -5.4 | -1.0 | 3.5 | 5.6 | 7.2 |

Значения функций потерь приведены в таблице:

Табл. 3. Значения функции потерь

|  |  |  |
| --- | --- | --- |
|  | Квадратичная | Экспоненциальная |
| h | 0,5743 | 0,6314 |
| 10\*h | 4,8453 | 0,0724 |

Видно, что при умножении решающей функции на 10, значение квадратичной функции потерь увеличивается, а экспоненциальной – уменьшается.

##### Задание

1) Построить несколько моделей линейной регрессии на основании различных наборов обучающих выборок (изменяя количество строк в обучающей выборке, например, для выборок размером 5000, 10000, 50000 и т.д.), проанализировать точность аппроксимации. Сделать выводы о влиянии величины обучающей выборки на точность модели.

При размере обучающей выборки 5000, качество построенной модели стало 0.73:

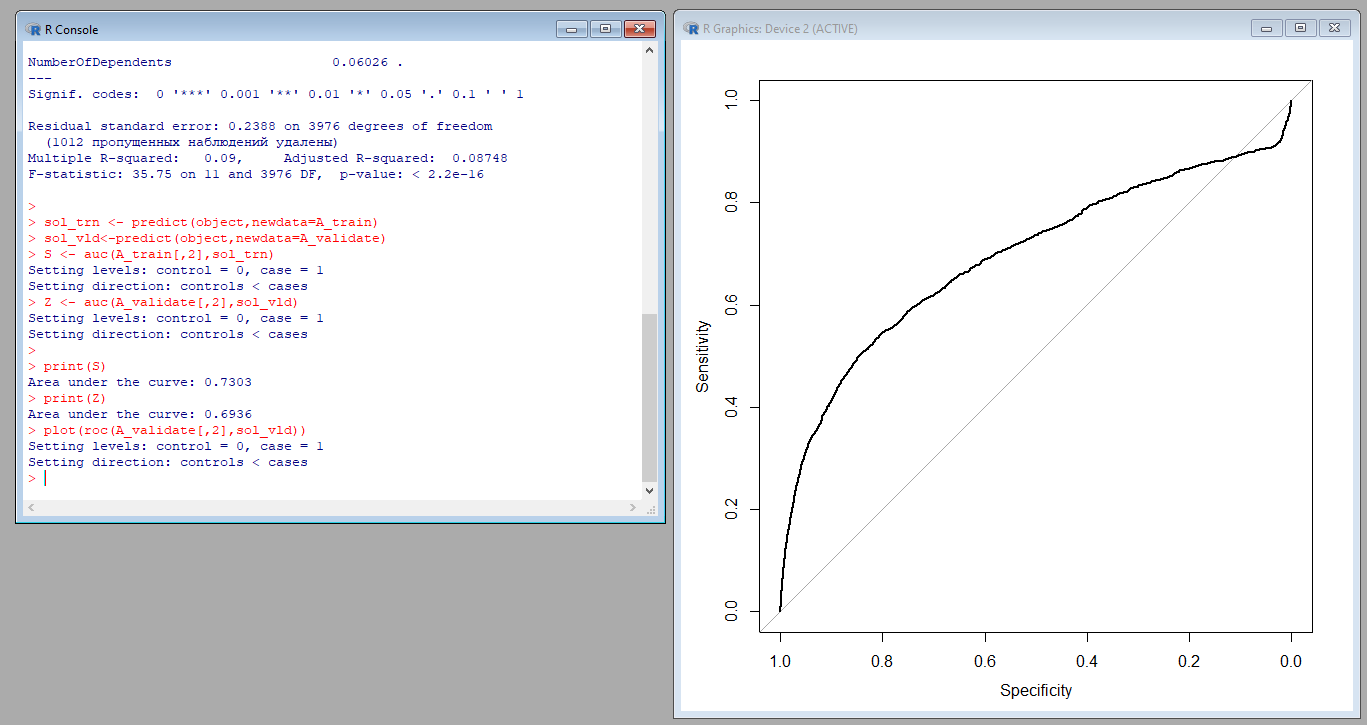


Рисунок 1 – Размер обучающей выборки 5000.

При размере обучающей выборки 10000, качество построенной модели стало 0.6744:

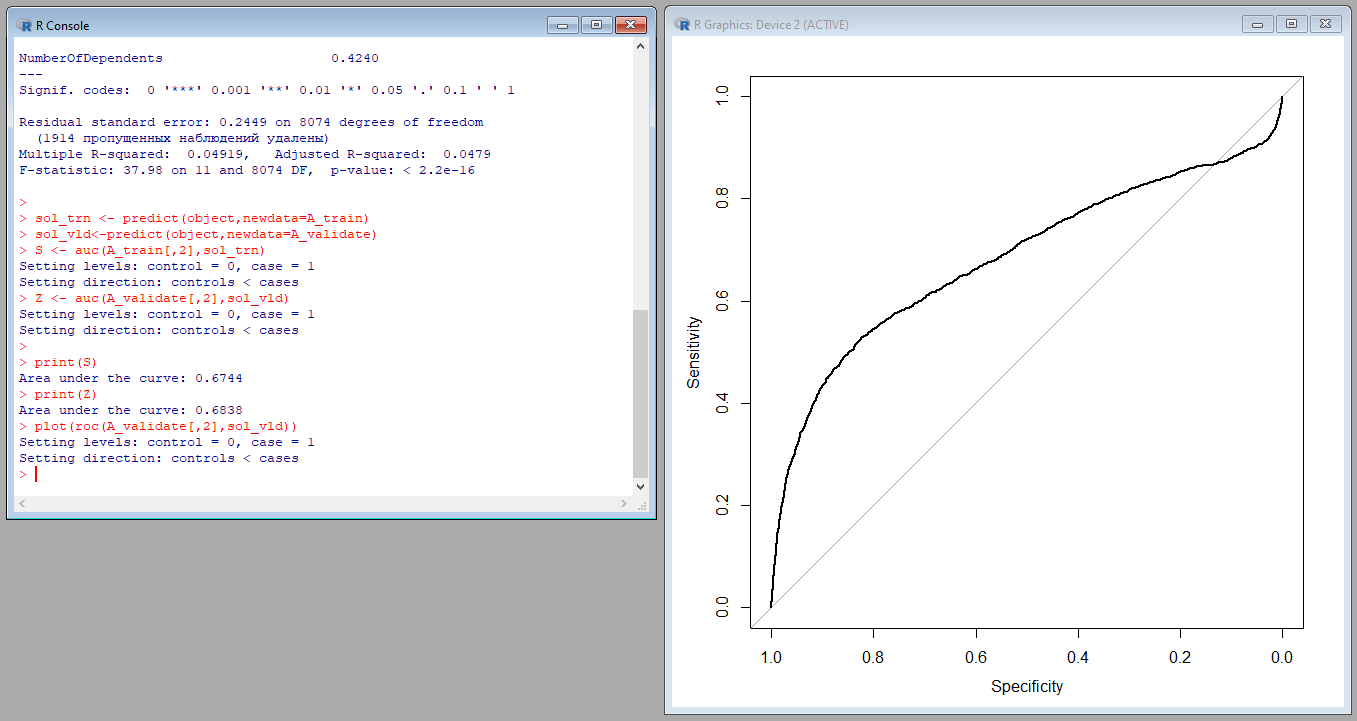


Рисунок 2 – Размер обучающей выборки 10000.

При размере обучающей выборки 50000, качество построенной модели стало 0.6876:

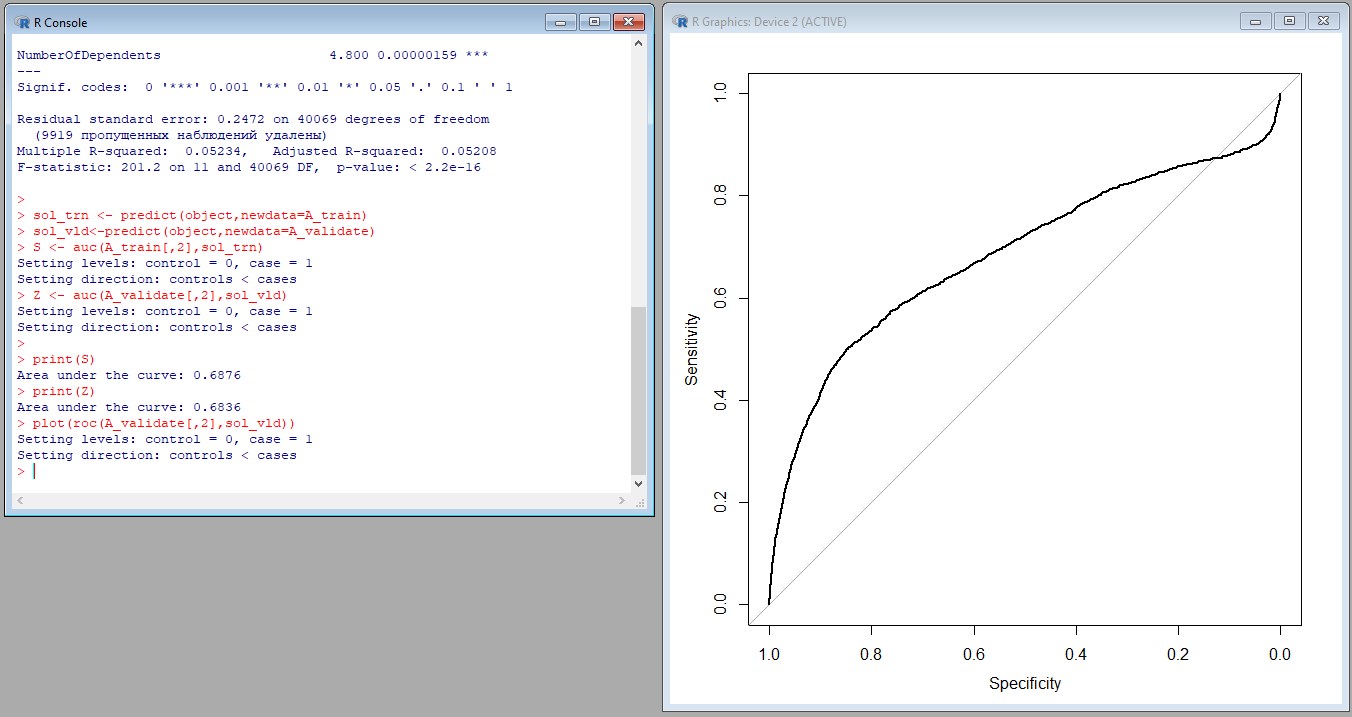


Рисунок 3 – Размер обучающей выборки 50000.

При размере обучающей выборки 100000, качество построенной модели стало 0.6855:

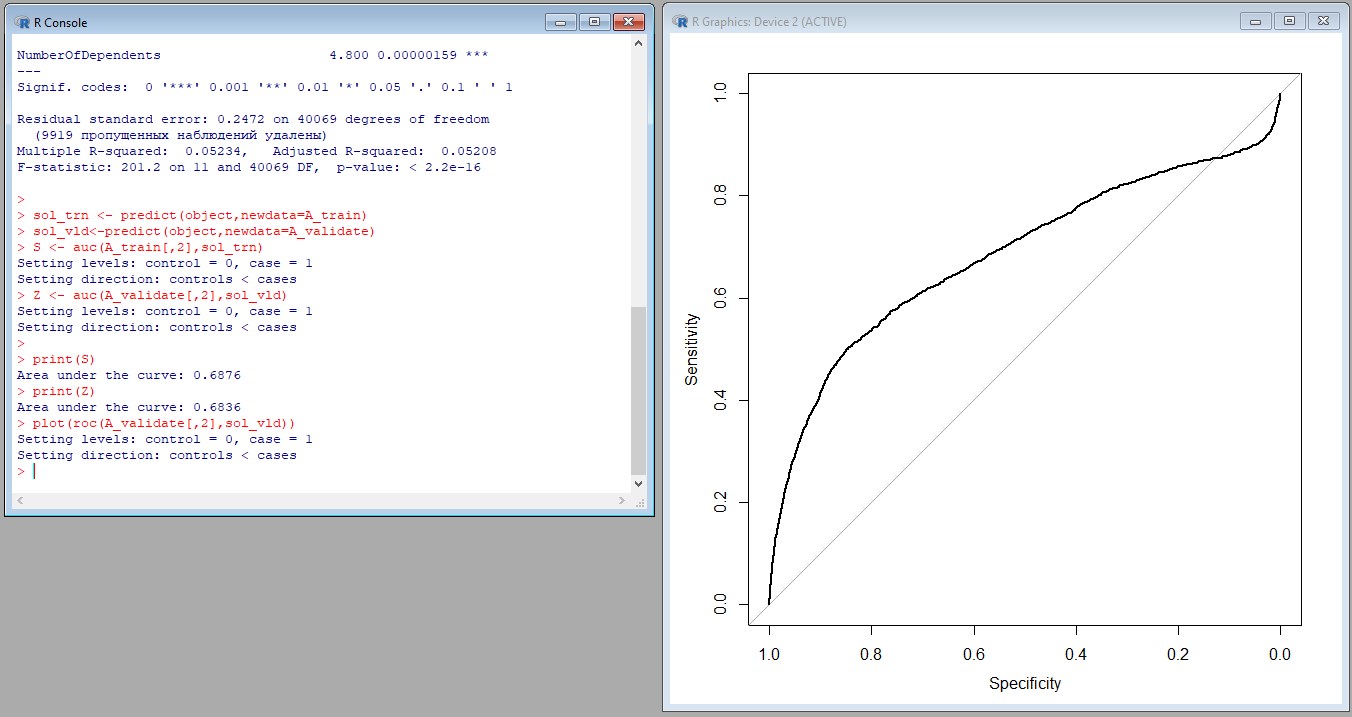


Рисунок 4 – Размер обучающей выборки 100000.

##### Содержание отчёта

1. Полученные результаты по каждому пункту (или «скриншоты»).

2. Выводы о проделанной работе.