Министерство образования и науки Российской Федерации

ФГБОУ ВО Рыбинский государственный авиационный технический университет имени П.А. Соловьева

Факультет радиоэлектроники и информатики

Кафедра математического и программного обеспечения электронных вычислительных средств

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

по дисциплине

Математические методы анализа данных

по теме

Разработка регрессионной модели объекта по результатам экспериментов

Вариант 8

Студент группы ИПБ-13 Иванов Р.А.

Преподаватель, доцент Воробьев К. А.

Рыбинск 2017

Содержание

1. Линейная регрессия

Ниже представлена выборка в исходном и стандартизированном видах (таблица 1).

Таблица 1 – Выборка для линейной регрессии

X Исходное	Y Исходное	X Стандартизированное	Y Стандартизированное
0.32	13.24	-0.697175	-0.668287
0.08	12.38	-1.103296	-1.094642
1.02	15.65	0.487346	0.526499
0.19	12.78	-0.917158	-0.896337
0.0	12.04	-1.23867	-1.263201
1.06	15.66	0.555033	0.531457
0.26	12.96	-0.798705	-0.8071
0.16	12.59	-0.967923	-0.990532
0.45	13.58	-0.477193	-0.499728
1.58	17.56	1.434962	1.473404
1.32	16.63	0.994997	1.012346
1.47	17.04	1.248823	1.215608
0.55	13.99	-0.307975	-0.296466
0.63	14.24	-0.172602	-0.172525
1.89	18.48	1.959536	1.929505

Ниже представлено математическое ожидание и среднеквадратичное отклонение исходной выборки.

Таблица 2 – Характеристики исходной выборки

Выборка	Мат. ожидание	Среднеквадратичное отклонение
X	0.7320000000000001	0.5909562871594931
Y	14.588	2.0170975848150396

Применив метод наименьших квадратов получим такую систему линейных уравнений для определени параметров уравнения регрессии.

$$y = ax + b$$

$$F(a,b) = \sum_{i=1}^{n}(ax_i + b - y_i)^2$$

$$\begin{cases} \frac{\partial F(a,b)}{\partial a} = 2\sum_{i=1}^{n}(ax_i + b - y_i)x_i = a\ 2\sum_{i=1}^{n}x_i^2 + b\ 2\sum_{i=1}^{n}x_i - 2\sum_{i=1}^{n}x_i y_i \\ \frac{\partial F(a,b)}{\partial b} = 2\sum_{i=1}^{n}(ax_i + b - y_i) = a\ 2\sum_{i=1}^{n}x_i + b\ 2n - 2\sum_{i=1}^{n}y_i \end{cases}$$

Решив систему было получено следующее уравнении линейной регрессии:

$$y = 0.9996986902215902 * x + 0.00000000000000032566$$

На графике представлена выборка в стандартизированном виде(красные точки) и уравнение регрессии (рис. 1). Видно, что уравнение достаточно хорошо описывает взаимосвязь рассматриваемых величин, но все же присутсвуют точки, которые немного выбиваются, что говорит о не особо идиальной взаимосвязи.

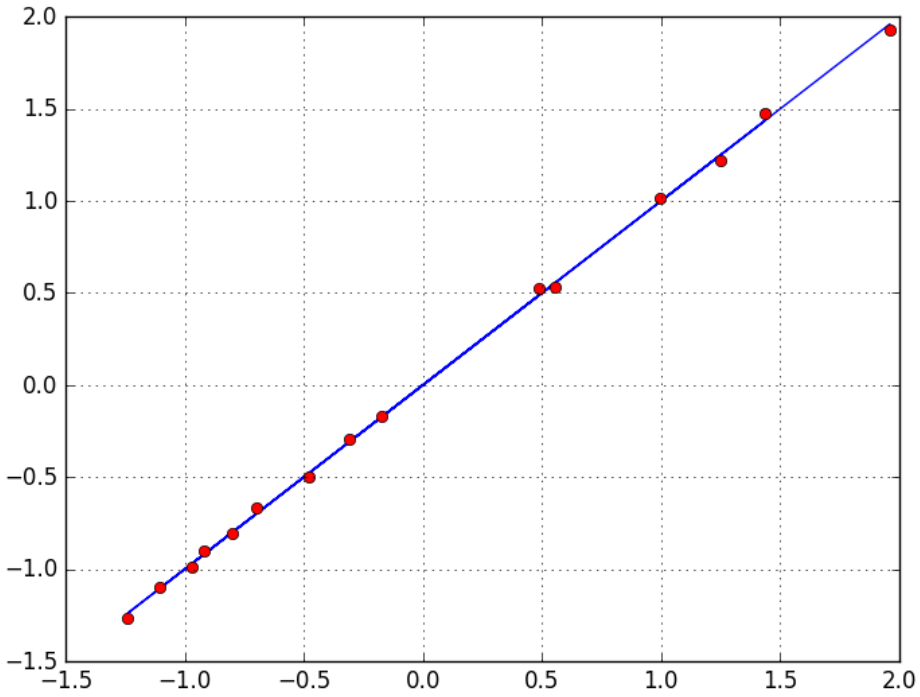

Рис. 1 – Линейная регрессия

Расчётное значение критерия Фишера (F-критерий) равно 21562.733249157056. Табличное значение критерия для уровня доверия 95% равно 4.67. Так как расчётный критерий много больше табличного, мы можем утверждать, что полученное уравнение регрессии статистически значимо.

Расчетное значение критерия Стьюдента для коэффициентов a и b равно 3.7357497490332565 и $1.2169545309210383e-15$ соответственно. Критическому значению t-критерия соответствует 2.16037, таким образом можно утверждать, что параметр b статистически незначим и им можно пренебречь, в отличии от параметра a.

Вычисленный коэффициент детерминации $R^2 = 0.9996986902215901$ означает, что изменение X в $99,99\%$ случаев влечет за собой изменение Y.

2. Параболическая регрессия

Ниже представлена выборка в исходном и стандартизированном видах (таблица 3), а также параметры первой (таблица 4)

Таблица 3 – Выборка для параболической регресии

X Исходное	Y Исходное	X Стандартизированное	Y Стандартизированное
0.83	14.65	-0.658295	-0.57669
4.7	152.1	1.034838	0.739438
3.46	92.27	0.492336	0.166547
0.07	2.82	-0.990797	-0.689966
0.8	14.0	-0.67142	-0.582914
8.58	416.15	2.732347	3.267802
0.4	7.49	-0.846421	-0.645249
2.05	43.76	-0.124542	-0.297952
4.8	155.84	1.078589	0.77525
1.69	34.17	-0.282043	-0.389779
0.54	9.71	-0.785171	-0.623992
3.82	110.14	0.649837	0.337658
2.29	50.53	-0.019542	-0.233127
0.84	15.1	-0.65392	-0.572381
0.15	4.42	-0.955796	-0.674645

Таблица 4 – Характеристики исходной выборки

Выборка	Мат. ожидание	Среднеквадратичное отклонение
X	2.3346666666666667	2.285703003940412
Y	74.87666666666668	104.43514585085276

Применив метод наименьших квадратов получим такую систему линейных уравнений для определени параметров уравнения регрессии.

$$y = ax^2 + bx + c$$

$$F(a,b,c) = \sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)^2$$

$$\begin{cases} \frac{\partial F(a,b,c)}{\partial a} = 2\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)x_i^2 = a\ 2\sum_{i=1}^{n}x_i^4 + b\ 2\sum_{i=1}^{n}x_i^3 + c\ 2\sum_{i=1}^{n}x_i^2 - 2\sum_{i=1}^{n}x_i^2y_i \\ \frac{\partial F(a,b,c)}{\partial b} = 2\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)x_i = a\ 2\sum_{i=1}^{n}x_i^3 + b\ 2\sum_{i=1}^{n}x_i^2 + c\ 2\sum_{i=1}^{n}x_i - 2\sum_{i=1}^{n}x_iy_i \\ \frac{\partial F(a,b,c)}{\partial c} = 2\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i) = a\ 2\sum_{i=1}^{n}x_i^2 + b\ 2\sum_{i=1}^{n}x_i + c\ 2n - 2\sum_{i=1}^{n}y_i \end{cases}$$

Итоговое уравнение:

$$y = 0.2135817829018743 * x^2 + 0.6909999979747826 * x - 0.21358178290187405$$

На графике представлена выборка в стандартизированном виде(красные точки) и уравнение регрессии (рис. 2).

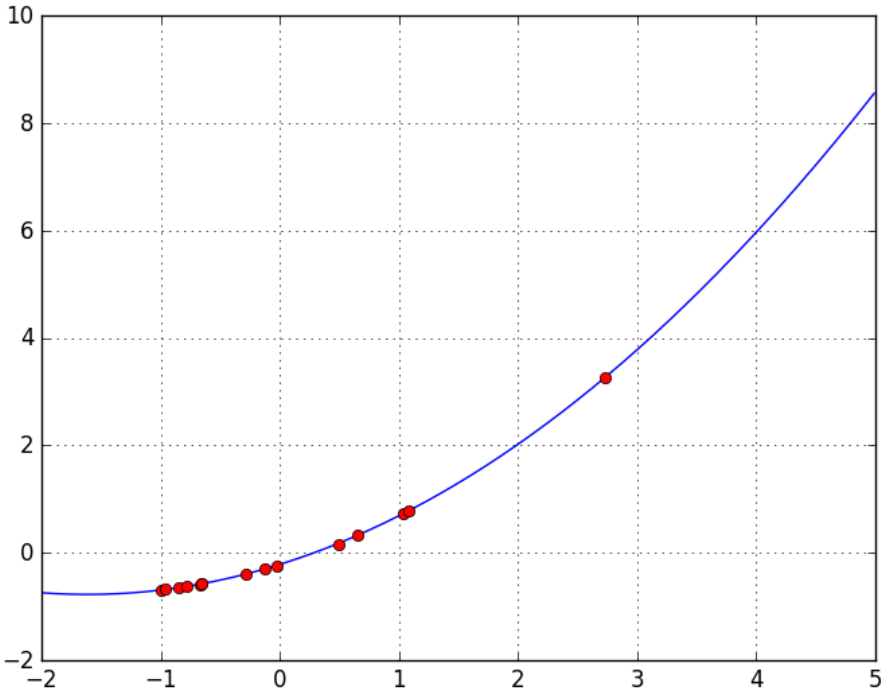

Рис. 2 – Параболическая регрессия

3. Множественная регрессия

Ниже представлена выборка в исходном и стандартизированном видах (таблица 5), а также параметры первой (таблица 6)

Таблица 5 – Выборка для множественной регресии

X Исходное	Y Исходное	Z Исходное	X Станд.	Y Станд.	Z Станд.
2.3	6.33	59.06	-0.078401	1.285818	1.158237
3.33	0.03	15.79	1.170353	-0.950696	-0.558056
3.16	0.23	15.7	0.964248	-0.879696	-0.561626
2.43	0.15	8.81	0.079209	-0.908096	-0.834916
1.34	0.28	2.36	-1.242288	-0.861946	-1.090754
1.68	1.17	11.51	-0.830078	-0.545994	-0.727822
1.91	8.58	74.74	-0.55123	2.084574	1.78018
3.54	5.29	61.22	1.424954	0.916616	1.243913
2.64	2.53	31.03	0.333809	-0.06319	0.046434
3.42	2.59	35.02	1.279468	-0.04189	0.204696
1.12	0.22	0.16	-1.509012	-0.883246	-1.178016
1.94	1.89	20.75	-0.514858	-0.290392	-0.361319
3.3	7.9	76.99	1.133982	1.843172	1.869426
1.05	1.08	6.81	-1.593879	-0.577944	-0.914246
2.31	2.35	27.94	-0.066277	-0.127091	-0.07613

Таблица 6 – Характеристики исходной выборки

Выборка	Мат. ожидание	Среднеквадратичное отклонение
X	2.3646666666666665	0.8248221357089998
Y	2.708	2.816882910831285
Z	29.859333333333336	25.21130949042953

Применив метод наименьших квадратов получим такую систему линейных уравнений для определени параметров уравнения регрессии.

$$z = ax + by + c$$

$$F(a,b,c) = \sum_{i=1}^{n}(ax_i + by_i + c - z_i)^2$$

$$\begin{cases} \frac{\partial F(a,b,c)}{\partial a} = 2\sum_{i=1}^{n}(ax_i + by_i + c - z_i)x_i = a\ 2\sum_{i=1}^{n}x_i^2 + b\ 2\sum_{i=1}^{n}x_iy_i + c\ 2\sum_{i=1}^{n}x_i - 2\sum_{i=1}^{n}x_iz_i \\ \frac{\partial F(a,b,c)}{\partial b} = 2\sum_{i=1}^{n}(ax_i + by_i + c - z_i)y_i = a\ 2\sum_{i=1}^{n}x_iy_i + b\ 2\sum_{i=1}^{n}y_i^2 + c\ 2\sum_{i=1}^{n}y_i - 2\sum_{i=1}^{n}y_iz_i \\ \frac{\partial F(a,b,c)}{\partial c} = 2\sum_{i=1}^{n}(ax_i + by_i + c - z_i) = a\ 2\sum_{i=1}^{n}x_i + b\ 2\sum_{i=1}^{n}y_i + c\ 2n - 2\sum_{i=1}^{n}z_i \end{cases}$$

Итоговое уравнение:

$$z = 0.24084103100742218000 * x + 0.90807086877838860000 * y - 0.00000000000000022177$$

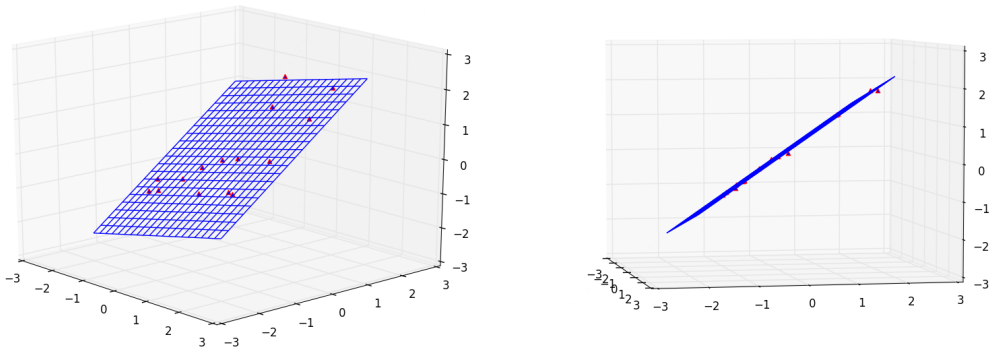

Рис. 3 – Множественная регрессия

4. Выводы

Подводя общий итог можно сказать, что все модели достаточно точно описывают взяимосвязь между рассмотренными СВ и могут применяться для оценки данных за пределами эксперементальной выборки.

5. Приложения

Класс для вспомогательных вычислений

```java
package helpers;

import java.util.stream.DoubleStream;

/**
 * Класс для вспомогательных вычислений
 *
 * @author Roman
 */
public class CalculationHelper {

    /**
     * Метод для нормализации массива значений
     *
     * @param array массив значений
     * @return нормализованный массив
     */
    public static double[] normalizedArray(double[] array) {
        double expValue = calculateExpectedValue(array);
        double standartDeviation = calculateStandardDeviation(array);
        return DoubleStream.of(array).map(xi ->
                (xi - expValue) / standartDeviation
        ).toArray();
    }

    /**
     * Метод для подсчета стандартного отклонения для массива значений
     *
     * @param array массив значений
     * @return стандартное отклонение
     */
    private static double calculateStandardDeviation(double[] array) {
        return Math.sqrt(calculateDispersion(array));
    }

    /**
     * Метод для подсчета диссперсии для массива значений
     *
     * @param array массив значений
     * @return диссперсия
     */
    public static double calculateDispersion(double[] array) {
        double expValue = calculateExpectedValue(array);
        return calculateExpectedValue(
                DoubleStream.of(array)
                        .map(s -> Math.pow(s - expValue, 2))
                        .toArray()
        );
    }

    /**
```

```
52      * Метод для подсчет математического ожидания для массива значений
53      *
54      * @param array массив значений
55      * @return математическое ожидание
56      */
57     public static double calculateExpectedValue(double[] array) {
58         return DoubleStream.of(array).sum() / array.length;
59     }
60 
61     /**
62      * Метод для подсчета коэффициента Фишера
63      * нужен для анализа реграссии
64      *
65      * @param y    первоначальный массив нормализованных значений
66      * @param newY новые значения Y, высчитаные по полученной формуле
67      * @return коэффициент Фишера
68      */
69     public static double calculateCoefficientFisher(double[] y,
70     double[] newY) {
71         double expValueY = CalculationHelper.calculateExpectedValue(y);
72         double numerator = 0;
73         double denominator = 0;
74         for (int i = 0; i < y.length; i++) {
75             numerator += Math.pow(newY[i] - expValueY, 2);
76             denominator += Math.pow(y[i] - newY[i], 2) / (y.length - 2);
77         }
78         return numerator / denominator;
79     }
80 
81     /**
82      * Метод для высчитывания коэффициентов Стьюдента
83      * @param a значение стоящее у переменной X в уравнении функции
84      * @param b значение b в уравнениее функции
85      * @param y массив значений Y
86      * @param x массив значений X
87      * @return коэффициенты Стьюдента
88      */
89     public static double[] calculateCoefficientStudent(double a,
90     double b, double[] y, double[] x) {
91         double[] result = new double[2];
92         double expValueY = calculateExpectedValue(y);
93         double expValueX = calculateExpectedValue(x);
94         double residualDispersionY = DoubleStream.of(y)
95                 .map(valuyY -> Math.pow(valuyY - expValueY, 2))
96                 .sum();
97         residualDispersionY = Math.sqrt(residualDispersionY / (y.length - 2));
98         double numeratorA = DoubleStream.of(x)
99                 .map(value -> Math.pow(value, 2))
100                .sum();
101        double denominatorA = DoubleStream.of(x)
102                .map(value -> Math.pow(value - expValueX, 2))
103                .sum();
104        numeratorA *= residualDispersionY;
105        denominatorA *= y.length;
106        result[0] = a / Math.sqrt(numeratorA / denominatorA);
```

```
107
108         double denominatorB = DoubleStream.of(x)
109                 .map(value -> Math.pow(value - expValueX, 2))
110                 .sum();
111         result[1] = b / Math.sqrt(residualDispersionY / denominatorB);
112         return result;
113     }
114 }
```

Вспомогательный класс для вычисления различных видов регрессий

```
1  package helpers;
2
3  import org.apache.commons.math.linear.RealVector;
4  import org.apache.commons.math.stat.regression.OLSMultipleLinearRegression;
5  import org.apache.commons.math.stat.regression.SimpleRegression;
6  import org.apache.commons.math3.fitting.PolynomialCurveFitter;
7  import org.apache.commons.math3.fitting.WeightedObservedPoints;
8
9  public class RegressionHelper extends OLSMultipleLinearRegression {
10     /**
11      * Метод для высчитывания коэффициентов параболической регрессии
12      * @param arrayX массив значений X
13      * @param arrayY массив значений Y
14      * @return коэффициенты уравнения
15      */
16     public static double[] calculateCoefficientPolynomialRegression(
17     double[] arrayX, double[] arrayY) {
18         WeightedObservedPoints obs = new WeightedObservedPoints();
19         for (int j = 0; j < arrayX.length; j++) {
20             obs.add(arrayX[j], arrayY[j]);
21         }
22         PolynomialCurveFitter fitter = PolynomialCurveFitter.create(2);
23         return fitter.fit(obs.toList());
24     }
25
26     /**
27      * Метод для высчитывания коэффициентов множественной регрессии
28      * @param z массив значений Z
29      * @param data масив со значениям X и Y
30      * @return коэффициенты уравнения
31      */
32     public static double[] calculateCoefficientMultipleRegression(
33     double[] z, double[][] data) {
34         RegressionHelper regressionHelper = new RegressionHelper();
35         regressionHelper.newSampleData(z, data);
36         RealVector vector = regressionHelper.calculateBeta();
37         return vector.getData();
38     }
39
40     /**
41      * Метод для получения линейной зависимости по vfccbdfv значений
42      * @param x массив значений X
43      * @param y массив значений Y
44      * @return линейная регрессия
45      */
```

```
46      public static SimpleRegression getSimpleRegression(double[] x,
47
48      double[] y) {
49          SimpleRegression regression = new SimpleRegression();
50          for (int i = 0; i < x.length; i++) {
51              regression.addData(x[i], y[i]);
52          }
53          return regression;
54      }
55  }
```