LaGuardia Community College

MAC 286 - Data Structures

Group Research Project

Forecasting sales using linear regression

(Walmart as an example)

June 4, 2019

Group Members:

Azim Waliyani

Gabriel Santos

Prof. Doyel Pal

Spring I, 2019

INTRODUCTION

Linear Regression is a machine learning algorithm based on be able to learn. It achieves a regression task. Regression copies a target prediction value based on independent variables and predicts the required output. It is mostly used for finding out the relationship between variables and predicting. Different regression models differ based on – the kind of connection between the dependent and independent variables, and the number of independent variables being used. The correlation coefficient r measures the strength of a connotation. Another name for r is the Pearson product moment correlation coefficient in honor of Karl Pearson, who developed it about 1900. There are at least three different formulas infrequently used to calculate this number, and these different formulas slightly represent different methods to the problem. However, the same value for r is obtained by any one of the various procedures. First, we give the raw score formula. N has the usual meaning of how many ordered pairs are in our sample. It is also essential to identify the variance between the sum of the squares and the squares of the quantities!

LITERATURE REVEW

According to the paper "Introduction to the statistical modelling " published in the Oxford University Press, in the year 2013 , "In many studies we wish to assess how a range of variables are associated with a particular outcome and also determine the strength of such relationships so that we can begin to understand how these factors relate to each other at a population level." In this research paper, the concepts of correlation and regression are reviewed and demonstrated. The review for measuring linear and nonlinear relationships between two continuous variables. In the case of measuring the linear relationship between a predictor and an outcome variable, simple linear regression analysis is conducted. These statistical concepts are illustrated by using a data set from published literature to assess a computed tomography-guided interventional technique. These statistical methods are important for exploring the relationships between variables and can be applied to many radiologic studies.

METHODOLOGY

1. We will begin by getting the data that we will be using on this research paper. We use an excel spreadsheet on csv format that includes the following data:
2. In the original excel spreadsheet will generate the original values.
3. We save our spreadsheet as .csv, which denotes “comma separated values” to simplify the read operations of our program.
4. We create an Algorithm coded using Java programming language (See Code bellow) to analyze the data.
5. Fill the arrays with the appropriate values from the .csv file using the comma “,” as a delimiter to store each line of values into an array of type String. Then we parse each element back to its original type and into its array, and we continue doing that for each line throughout the file.
6. Create arrays to holds the data and create variables to keep track of the successful and unsuccessful forecast information.
7. Display the results. The optimization of marketing processes supporting conclusions that are either made by knowledgeable managers or decision made by the support system that is dedicated to such responsibilities. Therefore, it is almost impossible to develop an algorithm of a decision-making system based on detailed knowledge with numerous open-endedly defined parameters.

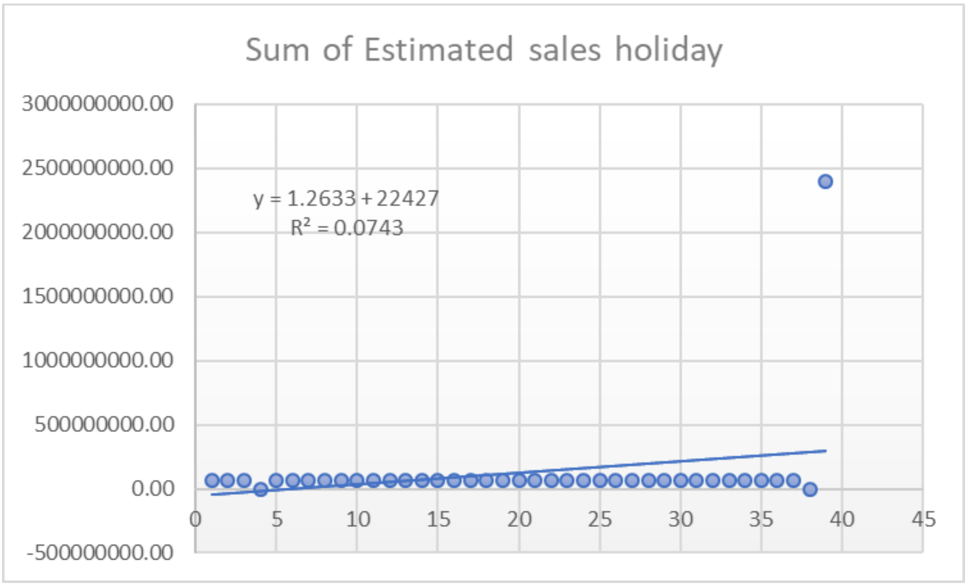
8. The idea of developing a decision-making support system as regards the control of the degree of marketing operation intensity adopts the possibility to construct a system that would take advantage of the experiential knowledge gained from learning by samples.

9. The prospect to apply a substantial amount of data regarding marketing and business.

10. Describe the behavior rules of the reality that we are going to control in the future through decisions that concern the power of online advertising.

<u>RESULTS:</u>

1. Before attempting to fit a linear model to observed data, a modeler should first determine whether there is a relationship between the variables of interest. This does not necessarily imply that one variable causes the other, but that there is some significant association between the two variables. A scatterplot can be a helpful tool in determining the strength of the relationship between two variables.



2. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model.
3. When we ran our data through the "train.csv" file, we got the slope and the intercept as follows:

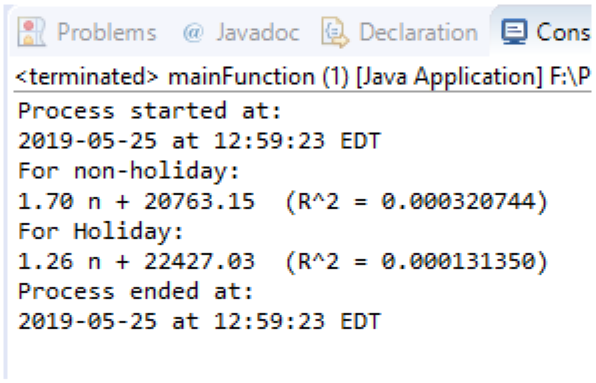
```
Process started at:
2019-05-25 at 13:58:09 EDT
For non-holiday:
1.70 n + 20763.15  (R^2 = 0.000320744)
For Holiday:
1.26 n + 22427.03  (R^2 = 0.000131350)
Process ended at:
2019-05-25 at 13:58:09 EDT
```

Using this result, we created another program to train our model:

x: input training data (univariate – one input variable(parameter))

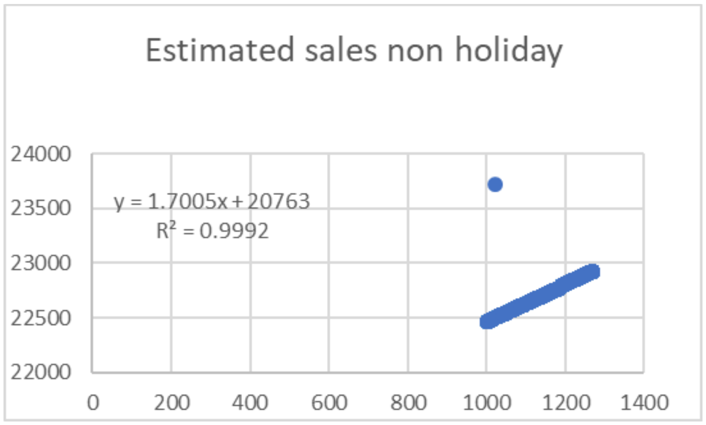
y: labels to data (supervised learning)

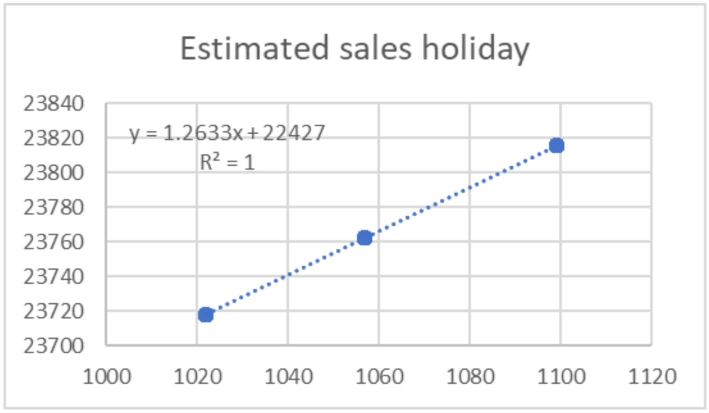
When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values.



```
<terminated> mainFunction (1) [Java Application] F:\P
Process started at:
2019-05-25 at 12:59:23 EDT
For non-holiday:
1.70 n + 20763.15  (R^2 = 0.000320744)
For Holiday:
1.26 n + 22427.03  (R^2 = 0.000131350)
Process ended at:
2019-05-25 at 12:59:23 EDT
```

Results of Trade for linear regression in Microsoft excel:





JAVA CODE:

class MainFunction

```
package predictionFinal;

import java.io.FileNotFoundException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class mainFunction {
    public static void main(String[] args) throws FileNotFoundException {
        SimpleDateFormat formatter= new SimpleDateFormat("yyyy-MM-dd 'at' HH:mm:ss
z");
        Date date = new Date(System.currentTimeMillis());

         System.out.println("Process started at:");
         System.out.println(formatter.format(date));
        Calendar.getInstance().toString();
       fileWrite f1 = new fileWrite();
       System.out.println("Process ended at:");
       System.out.println(formatter.format(date));

//          System.out.print(Calendar.getInstance().getTime().toString());


    }
}
```

class LinearRegression

```
package predictionFinal;

import java.util.ArrayList;

public class LinearRegression {
    private final double intercept, slope;
    private final double r2;
    private final double svar0, svar1;
    public LinearRegression(ArrayList<Integer> x, ArrayList<Double> y) {
        if (x.size() != y.size()) {
            throw new IllegalArgumentException("array lengths are not equal");
        }
        int n = x.size();

        // first pass
        double sumx = 0.0, sumy = 0.0, sumx2 = 0.0;
        for (int i = 0; i < n; i++) {
            sumx  += x.get(i);
            sumx2 += x.get(i)*x.get(i);
            sumy  += y.get(i);
        }
        double xbar = sumx / n;
        double ybar = sumy / n;

        // second pass: compute summary statistics
        double xxbar = 0.0, yybar = 0.0, xybar = 0.0;
        for (int i = 0; i < n; i++) {
            xxbar += (x.get(i) - xbar) * (x.get(i) - xbar);
            yybar += (y.get(i) - ybar) * (y.get(i) - ybar);
            xybar += (x.get(i) - xbar) * (y.get(i) - ybar);
        }
        slope  = xybar / xxbar;
        intercept = ybar - slope * xbar;

        // more statistical analysis
        double rss = 0.0;      // residual sum of squares
        double ssr = 0.0;      // regression sum of squares
        for (int i = 0; i < n; i++) {
            double fit = slope*x.get(i) + intercept;
            rss += (fit - y.get(i)) * (fit - y.get(i));
            ssr += (fit - ybar) * (fit - ybar);
        }

        int degreesOfFreedom = n-2;
        r2    = ssr / yybar;
        double svar  = rss / degreesOfFreedom;
        svar1 = svar / xxbar;
        svar0 = svar/n + xbar*xbar*svar1;
    }

    /**
     * Returns the <em>y</em>-intercept &alpha; of the best of the best-fit line
<em>y</em> = &alpha; + &beta; <em>x</em>.
     *
     * @return the <em>y</em>-intercept &alpha; of the best-fit line <em>y =
&alpha; + &beta; x</em>
     */
    public double intercept() {
        return intercept;
    }

    /**
     * Returns the slope &beta; of the best of the best-fit line <em>y</em> =
&alpha; + &beta; <em>x</em>.
     *
     * @return the slope &beta; of the best-fit line <em>y</em> = &alpha; + &beta;
<em>x</em>
     */
    public double slope() {
        return slope;
    }

    /**
     * Returns the coefficient of determination <em>R</em><sup>2</sup>.
     *
     * @return the coefficient of determination <em>R</em><sup>2</sup>,
     *         which is a real number between 0 and 1
     */
    public double R2() {
        return r2;
    }

    /**
     * Returns the standard error of the estimate for the intercept.
     *
     * @return the standard error of the estimate for the intercept
     */
    public double interceptStdErr() {
        return Math.sqrt(svar0);
    }
    /**
     * Returns the standard error of the estimate for the slope.
     *
     * @return the standard error of the estimate for the slope
     */
    public double slopeStdErr() {
        return Math.sqrt(svar1);
    }

    /**
     * Returns the expected response {@code y} given the value of the predictor
     * variable {@code x}.
     *
     * @param  x the value of the predictor variable
     * @return the expected response {@code y} given the value of the predictor
     *         variable {@code x}
     */
    public double predict(int x) {
        return slope*x + intercept;
    }

    /**
     * Returns a string representation of the simple linear regression model.
     *
     * @return a string representation of the simple linear regression model,
     *         including the best-fit line and the coefficient of determination
     *         <em>R</em><sup>2</sup>
     */
    public String toString() {
        StringBuilder s = new StringBuilder();
        s.append(String.format("%.2f n + %.2f", slope(), intercept()));
        s.append("  (R^2 = " + String.format("%.9f", R2()) + ")");
        return s.toString();
    }

}
```

class fileWrite

```
package predictionFinal;
import java.io.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.util.Calendar;
import java.util.Date;
import java.time.temporal.ChronoUnit;
public class fileWrite  {
	public fileWrite() {
		try {
			//SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
			String csvFile = "test1.csv";
```

```
                    String output = "output.csv";
               SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
               String token = ",";
               BufferedReader br = new BufferedReader(new FileReader(csvFile));
                    BufferedWriter bw = new BufferedWriter(new FileWriter(output));
               String line="";
               String [] item = new String[4];
               LocalDate startDate=LocalDate.parse("2010-02-05");
               line = br.readLine();//This will bring to the first line which is
not useful
               FileRead f1 = new FileRead();
            f1.read();
            LinearRegression l1 = new LinearRegression(f1.Weeklist,f1.salesList);
            LinearRegression l2 = new
LinearRegression(f1.holidayWeek,f1.holidaySales);
            System.out.println("For non-holiday:");
            System.out.println(l1.toString());
            //System.out.println(l1.R2());
            //System.out.println("Slope = " + l1.slope() + " Intercept = " +
l1.intercept());
            System.out.println("For Holiday:");
            System.out.println(l2.toString());
            bw.write("Day since " + startDate);
            bw.write(token);
            bw.write("Estimated sales");
            bw.write(token);
            bw.write("IS_HOLIDAY");
            bw.write("\n");
            while ((line =br.readLine())!= null) {

                // System.out.println(line.toString());
                item = line.split(",");
                String store = item[0];
                String department = item[1];
                String date = item[2];
               // double sales= Double.parseDouble(item[3]);
                String isTrue = item[3].toLowerCase();
                //System.out.println(isTrue);
                boolean isHoliday = Boolean.valueOf(isTrue);
                Calendar c1 = Calendar.getInstance();
                c1.setTime(sdf.parse(date));
                LocalDate currentDate = LocalDate.parse(date);
                long noOfDaysBetween = ChronoUnit.DAYS.between(startDate,
currentDate);
                String day = Long.toString(noOfDaysBetween);
                String prediction =
Double.toString(l1.predict((int)noOfDaysBetween));
                String holidayPrediction =
Double.toString(l2.predict((int)noOfDaysBetween));
                if (isHoliday==false) {
                    bw.write(day);
                    bw.write(token);
                    bw.write(prediction);
                    bw.write(token);
                    bw.write(isTrue);
                    bw.write("\n");
                } else if (isHoliday==true){
                    bw.write(day);
                    bw.write(token);
                    bw.write(holidayPrediction);
                    bw.write(token);
                    bw.write(isTrue);
                    bw.write("\n");
                }
            }
        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
```

class FileRead

```
package predictionFinal;

import java.io.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.util.Date;
import java.util.LinkedList;
import java.util.*;
import java.time.temporal.ChronoUnit;



public class FileRead {
    ArrayList<Integer> Weeklist = new ArrayList<>();
    ArrayList<Double> salesList = new ArrayList<>();
    ArrayList<Integer> holidayWeek = new ArrayList<>();
    ArrayList<Double> holidaySales = new ArrayList<>();
    private int firstYear;
    public void read() throws FileNotFoundException {
        String csvFile = "train.csv";
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        String token = ",";

        BufferedReader br= new BufferedReader(new FileReader(csvFile));;
        String line="";
        String [] item = new String[5];
        LocalDate startDate=LocalDate.parse("2010-02-05");

        {
            try {
                line = br.readLine();//This will bring to the first line which is
not useful
                while ((line =br.readLine())!= null) {

                    // System.out.println(line.toString());
                    item = line.split(",");
                    String store = item[0];
                    String department = item[1];
                    String date = item[2];
                    double sales= Double.parseDouble(item[3]);
                    String isTrue = item[4].toLowerCase();
                    //System.out.println(isTrue);
                    boolean isHoliday = Boolean.valueOf(isTrue);
                    //System.out.println(isHoliday);
                    int replacedDate = Integer.valueOf(date.replace("-", ""));
                    // System.out.println(replacedDate);
                    Calendar c1 = Calendar.getInstance();
                    c1.setTime(sdf.parse(date));
                    /*if(Weeklist.size()==1) {
                        startDate=LocalDate.parse(date);
                    }*/
                    LocalDate currentDate = LocalDate.parse(date);
                    long noOfDaysBetween = ChronoUnit.DAYS.between(startDate,
                    currentDate);
                    //System.out.println(noOfDaysBetween);
                    //int currentYear = c1.get(Calendar.YEAR);
                    //System.out.println(c1.getTime());
                    int weekNumber = c1.get(Calendar.WEEK_OF_YEAR);
                    if (isHoliday==false) {
                        Weeklist.add((int) noOfDaysBetween);
                        salesList.add(sales);
                    } else if (isHoliday==true){
                        holidayWeek.add((int)noOfDaysBetween);
                        holidaySales.add(sales);
                    }


                }




            } catch (IOException | ParseException e) {
                e.printStackTrace();
            }finally {
                //System.out.println("This is the end of train class.");
                }
            }
        }
public int getFirstYear(){
            return firstYear;
}
}
```

DISCUSSION OF RESULTS:

Once we find the best x and y values, we get the best fit line. So, when we are finally using our model for prediction, it will predict the value of y for the input value of x. Whenever a linear regression model is fit to a group of data, the range of the data should be carefully observed. Attempting to use a regression equation to predict values outside of this range is often inappropriate, and may yield incredible answers

CONCLUSION:

After researching and creating the program (see Code), we believe that our methods can help beginners better understand and identify the patterns to forecast data and make successful marketing decisions. Developing a tool that would support decision-making about the cost optimization of marketing operations in the basic purpose of the system, it is also to support the decision-making processes of any company concerning the optimization of the advertising and marketing operations. Based on the data collected, with the application of the regression equations that is determined based on all historical data. The new data acquired will constitute the basis for a possible modification of advertising activities and future enhancements in productivity.

BIBLIOGRAPHY

Linear Regression, www.stat.yale.edu/Courses/1997-98/101/linreg.htm.

4.1.4.1. Linear Least Squares Regression,

www.itl.nist.gov/div898/handbook/pmd/section1/pmd141.htm.

https://www.researchgate.net/publication/263036348_Properties_of_Weighted_Least_Squares_R
egression_for_Cutoff_Sampling_in_Establishment_Surveys

"Coding Games and Programming Challenges to Code Better." *CodinGame*, www.codingame.com/playgrounds/3771/machine-learning-with-java---part-1-linear-regression.

Princeton University, The Trustees of Princeton University,

algs4.cs.princeton.edu/14analysis/LinearRegression.java.

Hettmansperger, Thomas P. "Weighted Least-Squares Rank Estimators." *Encyclopedia of Statistical Sciences*, 2006, doi:10.1002/0471667196.ess2910.pub2.