Shiny App to Predict Agricultural Tire Dimensions

Ana Rita Antunes(✉) and Ana Cristina Braga

ALGORITMI Centre, University of Minho, 4800-058 Guimarães, Portugal
ana_antunes96@hotmail.com, acb@dps.uminho.pt

Abstract. The main objective of this project, carried out in an industrial context, was to apply a multivariate analysis to variables related to the specifications required for the production of an agricultural tire and the dimensional test results. With the exploratory data analysis, it was possible to identify strong correlations between predictor variables and with the response variables of each test. In this project, the principal component analysis (PCA) serves to eliminate the effects of multicollinearity. The use of regression analysis was intended to predict the behavior of the agricultural tire considering the selected variables of each test. In the case of Test 1, when applying the Stepwise methods to select the variables, the model with the lowest value of Akaike Information Criterion (AIC) was achieved with the technique "Both". However, the lowest value of AIC for Test 2 was achieved with "Backward". Regarding the validation of assumptions, both Test 1 and Test 2 were validated. Therefore, all the quantitative variables are important, both in Test 1 and Test 2, because they are a linear combination that determines the principal components. In order to make it easier to compute predictions for future agricultural tires, an application that was developed in Shiny allows the company to know the behavior of the tire before it was produced. Using the application, it is possible to reduce the industrialization time, materials and resources, thus increasing efficiency and profits.

Keywords: Multiple linear regression · Principal component analysis · Siny application · Agricultural tires

1 Introduction

In the industrial process of production of a new tire, it is necessary to consider some specifications. The agricultural tire is constituted with different components like the tread, belt, inner liner, sidewall, bead, among others. In this case, it is important to define the mold, the material and the quantity. After that, the tire has to pass some tests, for example, dimensional and endurance tests, among others. The tire passes the test if the results are in accordance with the legal norms, where the maximum and minimum of dimensional and endurance values are defined. So, when the test result is greater than the maximum defined,

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12251, pp. 247–260, 2020.
https://doi.org/10.1007/978-3-030-58808-3_19

the tire doesn't pass and the company has to make changes in the type and/or the quantity of materials.

In this study, the main goal was to apply multivariate analysis to variables related to the specifications required for agricultural tire production and dimensional test results, Test 1 and Test 2. The purpose of this was to understand which variables influence the test results and to predict their values. So, to develop a tire it is important to consider a lot of variables simultaneously and, if it is possible to predict the values for the two tests, it will make it easier for the producers. Multiple Linear Regression (MLR) will help to achieve the results that we want, because the predictors variables are quantitative. MLR has many assumptions to be considered and one of them is multicollinearity effects.

Multicollinearity effects are when two or more predictor variables have a strong correlation among themselves. This can cause problems in MLR. When we estimate regression coefficients and the predictor variables are highly correlated, the coefficients tend to vary widely. Another problem is when we want to make an interpretation of a regression coefficient, the signal can be misleading [5]. One possibility to correct this problem is using Principal Component Regression (PCR), which is a linear regression using principal components. Maxwell et al. [7] wrote an article to tackle with multicollinearity effects and here 5 methodologies were tested: Partial Least Square Regression (PLSR), Ridge Regression (RR), Ordinary Least Square Regression (OLS), Least Absolute Shrinkage and Selector Operator (LASSO) Regression, and the Principal Component Regression (PCR). To compare the 5 methodologies, they used a different number of observations and a number of predictor variables. Root Mean Square Error (RMSE) and AIC were used to compare the performance of each model. With this analysis the authors concluded that PCR has the lowest AMSE and AIC, which means that according to them, PCR is the most efficient in handling critical multicollinearity effects [7]. Lafi and Kaneene used Principal Component Analysis (PCA) to detect and correct multicollinearity effects in a veterinary epidemiological study. In this article were compare OLS and PCR to adjust regression coefficents. The PCR coefficients were more reliable than OLS [6].

After selecting the model for Test 1 and Test 2, a web application was developed to predict the test results before the tire was produced.

2 Methods

2.1 Principal Component Analysis

PCA is a statistic procedure for multivariate problems. It was introduced in 1901 by Pearson and in 1933 it was independently developed by Hotelling [8].

PCA is useful when there are many predictor variables regarding the number of observations in the dataset. It is also used when the predictor variables are highly correlated with each other because it eliminates the effects of multicollinearity. Normally, PCA is used to reduce the dimensionality of a problem and principal component represents most of the information contained in the dataset. This means that the first PC explains the greater proportion of the

original variables variation and the second explains the second greater proportion, but it is independent of the first, and so on.

As it is widely known, PCA transforms the variation of a set of variables highly correlated into a new set of variables that are uncorrelated and orthogonal. This new set of variables is a linear combination of the initial p variables. Linear combinations are described as follows (Eq. 1):

$$\begin{aligned} PC_1 &= a_{11}x_1 + a_{12}x_2 + ... + a_{1p}x_p \\ PC_2 &= a_{21}x_1 + a_{22}x_2 + ... + a_{2p}x_p \\ &\vdots \\ PC_p &= a_{p1}x_1 + a_{p2}x_2 + ... + a_{pp}x_p \end{aligned} \tag{1}$$

where a_{ij} are the loadings, x_1, x_2, ..., x_p are the initial variables and PC_1, PC_2, ..., PC_p are the p PCs [4].

After obtaining the linear combinations for each component, and when replacing them with the values of the initial variables, the scores are obtained [5].

2.2 Multiple Linear Regression

With linear regression it is possible to study the linear relationship between response variable (y_i, $i = 1, ..., n$) and one or more predictor variables (x_{ij}, $j = 1, ..., p$), where response variable is a quantitative variable and predictor variables can be quantitative or qualitative. When there is more than one predictor variables it is called Multiple Linear Regression (MLR), (Eq. 2), where β_0 is the constant term and β_p are the coefficients for each variable.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \varepsilon_i \tag{2}$$

To validate the model, it is necessary to verify some assumptions and this can be performed through an explanatory analysis of residuals. Thus, the assumptions to be validated are as follows [3]:

- $E[\varepsilon_i] = 0$, this means the average of the errors must be zero;
- $Var[\varepsilon_i] = \sigma^2$, so the errors variance must be constant;
- $\varepsilon_i \sim N(0, \sigma^2)$, with this, errors must follow a normal distribution;
- Errors are independent.

Another condition to be verified is the existence of multicollinearity and this can be identified by the correlations values and/or considering the Variance Inflation Factor (VIF). When VIF is greater than 10, it means that there are multicollinearity effects in the data. VIF is given by the expression:

$$VIF_j = \frac{1}{1 - R_j^2} \tag{3}$$

where R_j^2 is the coefficient of determination of x_j relative to the other predictor variables in the model [9].

Variable Selection Method. The stepwise method is used to obtain a model with predictor variables that better explain the variable response and it is possible to consider different criteria, for example, AIC. The "Backward" method builds the regression model using all the predictor variables and removes them considering the chosen criteria. The "Forward" method adds the predictor variables one by one until there are no more candidates that increase the value of the sum of squares in the regression model. However, it is possible to build a regression model with the entry and elimination of the predictor variables, considering the chosen criteria, called "Both" method. The iterative process ends when there are no more variables to be introduced or eliminated according to the criterion adopted [9].

One way to analyze the model that better explains the data in study is the value of AIC. This criterion compares the adequacy of the models when an attempt is made to balance the accuracy of the adjustment and the smallest number of explanatory variables [2]. The AIC value is calculated as follows:

$$AIC_c = -2log(L_p) + 2p \tag{4}$$

where L_p is the maximum value of likelihood function for the model and p is the number of predictor variables present in the model. The models with lowest AIC are the chosen ones [1].

3 Results and Discussion

For this analysis were used 146 experimental agricultural tires, 31 predictor variables, 4 of which are qualitative variables and 27 quantitative variables. We used 2 response variables, y_1 and y_2 for Test 1 and Test 2, respectively. The variables were coded due to a confidentiality agreement. All computations were made in R software using the appropriate packages available to perform the analysis.

Figure 1 represents the correlation coefficients (color intensity and the size of the circle are proportional to the correlation coefficients) and there are strong relationships with variable y_1, variable response for Test 1, as well as with y_2, variable response for Test 2.

Taking into account the values of the correlations of Fig. 1, multicollinearity effects are expected due to the values taken from r between the predictor variables once these variables are correlated with each other. It is also possible to see that x_6, x_7, x_8, x_9, x_{12}, x_{13}, x_{18}, x_{19}, x_{23}, x_{25}, x_{27}, x_{32} and y_1 are correlated (where $r > 0.90$), as well as between x_5, x_{10}, x_{11}, x_{15} and y_2 (where $r > 0.90$).

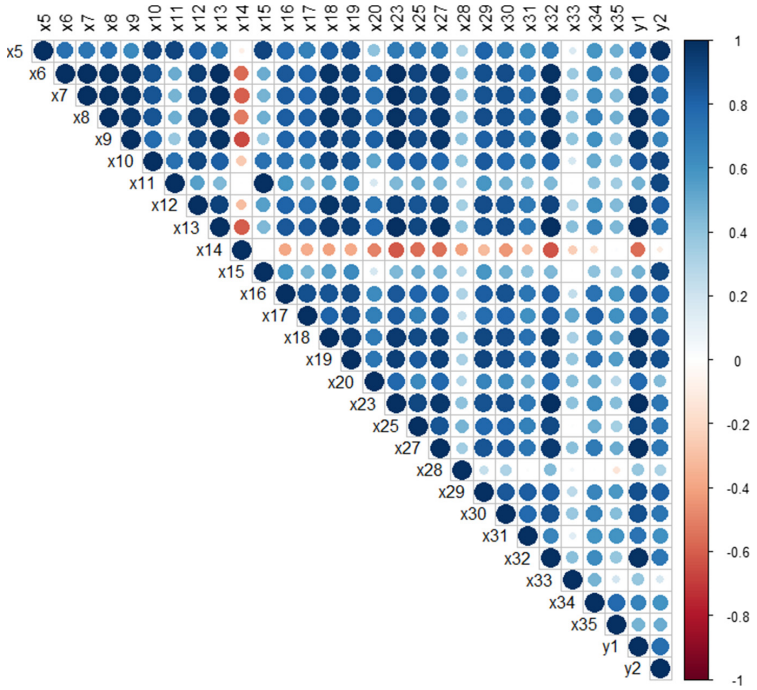

Fig. 1. Pearson's correlations. (Color figure online)

3.1 Principal Component Analysis Results

As referred in Sect. 2.1, PCA can be used to reduce the dimensionality of a problem or to eliminate multicollinearity effects. In this study, it was necessary to prove if multicollinearity effects exist. Regarding this problem, the data were normalized since the variables take different scales of measures.

In Table 1, the VIF values for 19 quantitative predictor variables are presented and the results are the same in Test 1 and Test 2, when an MLR was made for both tests. The VIF values for the other response variables are less than 20. Regarding the results obtained in Table 1, there are multicollinearity effects in the study, because most of the VIFs values are higher then 10.

Since the main objective is to build models that allow predictions for Test 1 and Test 2, the conditions of applicability of MLR models must be guaranteed. For this reason, we opted to use PCA to eliminate the effects of multicollinearity. For this reason, the 27 principal components were used in the models for Test 1 and Test 2 instead of the original variables.

Table 1. VIF value for quantitative predictor variables.

Variable	VIF
X_5	927.10
X_6	608.11
X_7	405.50
X_8	128.60
X_9	843.43
X_{10}	37.90
X_{11}	26452.1
X_{12}	360.31
X_{13}	2707.65
X_{14}	49.92
X_{15}	27120.15
X_{17}	25.91
X_{18}	510.04
X_{19}	77.13
X_{23}	927.66
X_{25}	51.99
X_{27}	41.99
X_{32}	193.62

The graph in Fig. 2 represents the biplot after the rotation varimax for the first two principal components, where the first explains 52.5% and the second 15.6% of the total data variation. It can be seen in Fig. 2 that variables x_5, x_{11} and x_{15} have the greatest positive contribution for the second principal component. Variable x_{14} has the greatest negative contribution for the first component. However, the other variables have the greatest positive contribution for the first component.

In this study, the 27 principal components were used because it was necessary to consider all the information and, for this reason, it was difficult to perform the interpretation of each principal component.

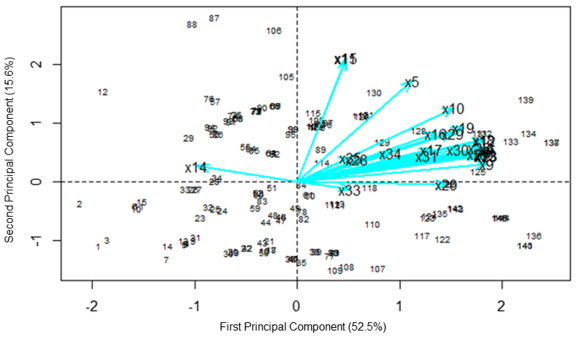

Fig. 2. Biplot for the first and second principal components.

3.2 Multiple Linear Regression Models

After the determination of each PC we proceeded to the construction of MLR models for each tire test. Two models were found using stepwise methods and considering AIC criteria to select the model for Test 1 and Test 2.

In Table 2 the model using "Both" technique has the lowest AIC value, for Test 1, for this reason it was the selected model. For Test 2, the lowest AIC value is using "Backward" and this was the chosen one.

Table 2. AIC values for Test 1 and Test 2.

	Test 1	Test 2
Backward	648.05	764.03
Forward	648.05	766.02
Both	647.45	778.42

Figure 3 shows the set of graphs produced in R using the plot (model) function to validate the assumptions. The first graph, Residuals vs Fitted, proves that the variance of residuals is constant and that residuals are independent because there isn't any pattern or tendency. The second graph shows that the errors follow a normal distribution, since the values are according to the diagonal, except on the extremes, which can indicate the presence of outliers. The Kolmogorov-Smirnov test was used to confirm if the errors follow a normal distribution, considering the following hypotheses: $H_0 : \varepsilon_i \sim N(\mu, \sigma^2)$ VS $H_1 : \varepsilon_i \nsim N(\mu, \sigma^2)$. For this test, the $p - value = 0.615$ reveals that the errors could follow the normal distribution for a significance level $\alpha = 0.05$. The last graph, Residual vs Leverage, shows there are no influence points.

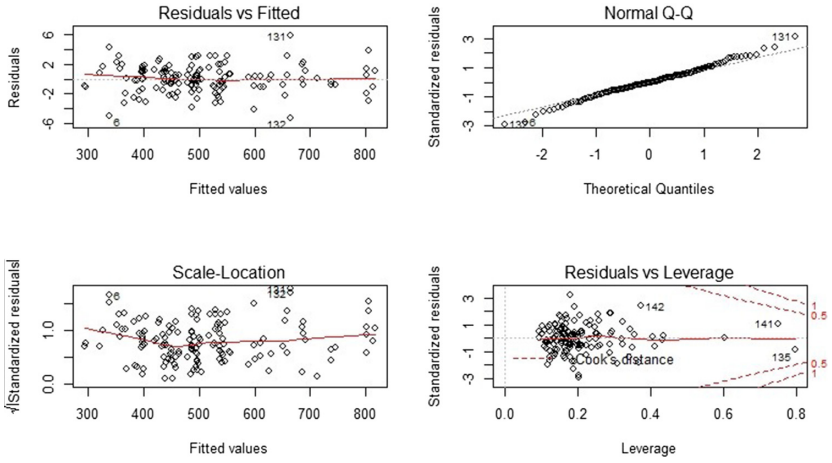

Fig. 3. Validated assumptions for Test 1.

Based on graphs in Fig. 4 it is possible to draw the same conclusions for Test 2. Looking at the Normal Q-Q plot, most of the values are according to the diagonal, except on the extremes, which means there isn't evidence to reject the null hypothesis. Regarding the Kolmogorov-Smirnov test, $p - value = 0.966$, the null hypothesis isn't rejected and the errors could follow the normal distribution for a significance level $\alpha = 0.05$.

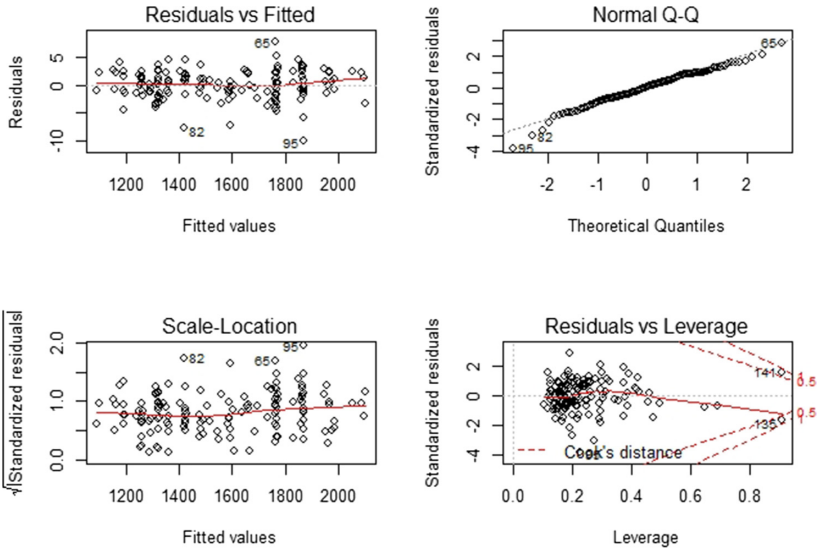

Fig. 4. Validated assumptions for Test 2.

When the extremes in Normal Q-Q plot are straight out it can mean there are outliers. The graphs in Fig. 5 reveal that there are five outliers for Test 1 and four for Test 2. All of them were individually analyzed to understand if it is a process problem or a human error since most of the values are not automatically introduced into company programs.

The entire analysis was repeated, for both models, after removing the outliers and it was found that by using the same criteria the results were not very different and outliers continued to exist. Since all the possible variables were not used in this study and the values of each observation, considered as an outlier, do not appear to be a human error, so it was decided to keep all the observations.

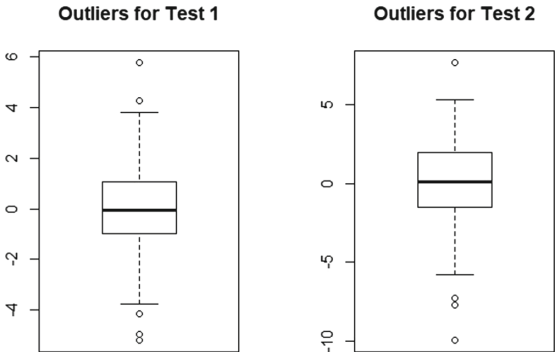

Fig. 5. Box-plots for the residuals for Test 1 and Test 2.

3.3 Shiny Application

The main objective of this study was to predict the results for Test 1 and Test 2 based on the constructed models. For this reason, it was developed a web application using Shiny. In the application it is possible to do two things: upload the dataset and make predictions based on the values of the variables.

Before creating the application it was important to define the necessary steps for its construction, which are represented in Fig. 6.

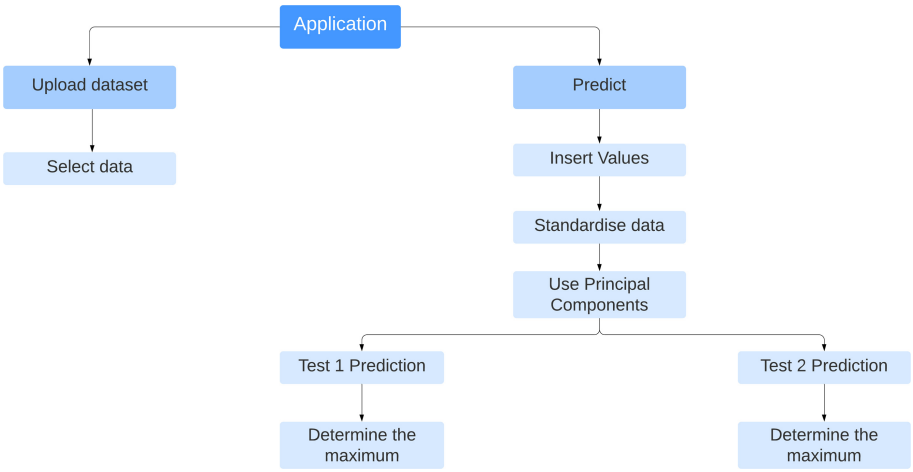

Fig. 6. Flowchart to create the application.

Programming Code. In order to obtain the application interface, a programming code was developed. Shiny is divided into two parts, "ui" and "server". "ui", known as the interface, is used to define how the web application is going

to look like. In contrast, "server" is used to define what the application is going to do and this is where the calculations for making predictions for Test 1 and Test 2, are made.

Before starting programming, four Excel documents were added to be used in a later stage. Figure 7 shows the information related to the dataset under study, the coefficients for Test 1 and Test 2, and the loadings for each principal component.

```
 9
10 library(shiny)
11 library(markdown)
12 library(datasets)
13 #  setwd("C:/Users/Ana Antunes/Desktop/DADOS")
14 setwd("C:/Users/Ana Antunes/Desktop/Universidade/Tese/Tese/DADOS")
15 dados=read.csv2(file="pesos.csv",sep=";",header=TRUE)
16 teste1=read.csv2(file="teste1.csv",sep=";",header=TRUE)
17 teste2=read.csv2(file="test2.csv",sep=";",header=TRUE)
18 exper=read.csv2(file="experimental2.csv",sep=";",header=TRUE)
19 attach(exper)
20
```

Fig. 7. Information to start the web application.

Firstly, in "ui" the menus were defined as "Upload Dataset" and "Prediction". In lines 24 and 25 is where the user can choose the file to load for the application. Regarding "Prediction", it specifies the quantitative variables, using "numericInput", and the qualitative variables, using "selectInput" (Fig. 8).

```
23 ui <- navbarPage("Agricultural Tire",
24                  tabPanel("Upload Dataset", fluid=TRUE,
25                           fileInput("FileInput", "Choose file"),
26                           DT::dataTableOutput("table")
27                  ),
28                   tabPanel("Prediciton", fluid=TRUE,
29                                          sidebarPanel("Variables",
30                                                       numericInput(inputId = "obs1",
31                                                                    label = "x5",
32                                                                    value = ""),
33                                                       numericInput(inputId = "obs2",
34                                                                    label=" x6",
35                                                                    value=""),
36                                                       numericInput(inputId = "obs3",
37                                                                    label="x7",
38                                                                    value=""),
39                                                       selectInput(inputId = "obs17",
40                                                                   label="x21",
41                                                                   choices = c("1","2")),
```

Fig. 8. Interface code in Shiny.

In order to show the prediction for Test 1 and Test 2, in line 124, was created a button "Go" and the next line is to show the table. On the following lines the colors of the application are defined (Fig. 9).

```
                actionButton("go2", "Go"),
                mainPanel(DT::dataTableOutput("tabela"))
    ),

    tags$style(type = 'text/css', '.navbar { background-color: #04E2FF;
                font-family: Calibri;
                font-size: 13px;
                color: #232426; }',

                '.navbar-dropdown { background-color: #04E2FF;
                font-family: Calibri;
                font-size: 13px;
                color: #232426; }',

                '.navbar-default .navbar-brand {
                color: #232426;
                }'
```

Fig. 9. Interface code in Shiny (continuation).

The next step was to define the necessary calculation to predict the value for Test 1 and Test 2. In the first place, the data have different scales and for this reason the data were standardized and the values introduced for each variable were saved (Fig. 10).

```
observeEvent(input$go2,{
  a <- ifelse(input$obs1>=1,round((input$obs1-mean(exper$x5))/sd(exper$x5),6),0)
  b <- ifelse(input$obs2>=1,round((input$obs2-mean(exper$x6))/sd(exper$x6),6),0)
  q1 <- ifelse(input$obs17=="1",1,0)
  q2 <- ifelse(input$obs17=="2",1,0)
  c <- ifelse(input$obs3>=1,round((input$obs3-mean(exper$x7))/sd(exper$x7),6),0)
```

Fig. 10. Server code for the values introduced.

Thereafter, it was important to define which variable is quantitative to determine the principal components for the 27 variables. After that, using the quantitative variables and the loadings obtained before, the principal components were calculated (Fig. 11).

```
  quant <- c(a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,s,u,w,x,y,z,aa,ab,ac,ad,ae)

  # componentes
  cp1 <- round(quant %*% dados[,1],5)
  cp2 <- round(quant %*% dados[,2],5)
  cp3 <- round(quant %*% dados[,3],5)
  cp4 <- round(quant %*% dados[,4],5)
```

Fig. 11. Server code for creating the principal components.

Finally, the models for Test 1 and Test 2 were calculated using the selected model coefficients for each test and the principal components obtained before. In Fig. 12, n1 and n4 represent the MLR for Test 1 and Test 2, respectively. The maximum is calculated using the expression in n2 and n5. After this, one condition was created to verify if a tire passes the test, represented by n3 and n6. With this information, lines 264, 265 and 266 were used to construct the table with the calculated results. The last line is used to run the application.

```
    n1 <- teste1[1,2]+teste1[2,2]*cp1+teste1[3,2]*cp2+teste1[4,2]*cp3+teste1[5,2]*cp4+teste1

    # maximum for test 1
    n2 <- (i1+0.4*(w1*25.4-i1*0.8))*1.05

    # see if the tire passes in test 1
    n3 <- ifelse(n2<n1,"Not Passed", "Passed")

    # prediction for test 2 using Backward
    n4 <- teste2[1,2]+teste2[2,2]*cp1+teste2[3,2]*cp2+teste2[4,2]*cp3+teste2[5,2]*cp4+teste2

    # maximum for test 2
    n5 <- 2*((i1*j1)/100)*1.04+k1*25.4

    # see if the tire passes in test 2
    n6 <- ifelse(n5<n4, "Not Passed", "Passed")

    output$sum1 <- renderPrint(n1)
    output$sum2 <- renderPrint(n2)
    output$sum3 <- renderPrint(n3)
    output$sum4 <- renderPrint(n4)
    output$sum5 <- renderPrint(n5)
    output$sum6 <- renderPrint(n6)

    tabela <- data.frame(round(n1,3),n3,round(n4,3),n6)
    colnames(tabela) <- c("Test 1"," Result", "Test 2", "Result")
    output$tabela <- DT::renderDataTable({DT::datatable(tabela, options = list(dom = 't'))})
  }
  )
}

# Run the application
shinyApp(ui = ui, server = server)
```

Fig. 12. Server code for predicting Test 1 and Test 2.

Application Interface. In Upload dataset it is possible to filter the data considering what is necessary to predict the value for Test 1 and Test 2. In Fig. 13 there is an example using a created dataset for an agricultural tire to explain only this functionality.

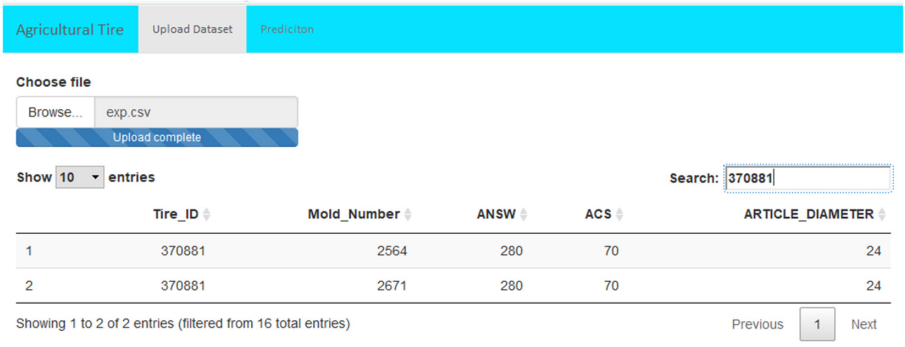

Fig. 13. Upload dataset.

In this case there are five variables and "Search" is an input for what we want to look for: for example, the tire identification number. The data have 15 different tires, where there are 2 tires that contain the number identification 370881 (lower left corner). Whoever wants to use the application for agricultural tires can filter for the tire number identification and its specification appears. This will be necessary for predicting Test 1 and Test 2.

Making predictions was one of the aims for this study and by using the developed application, the results for Test 1 and Test 2 can be predicted before tire production (Fig. 14).

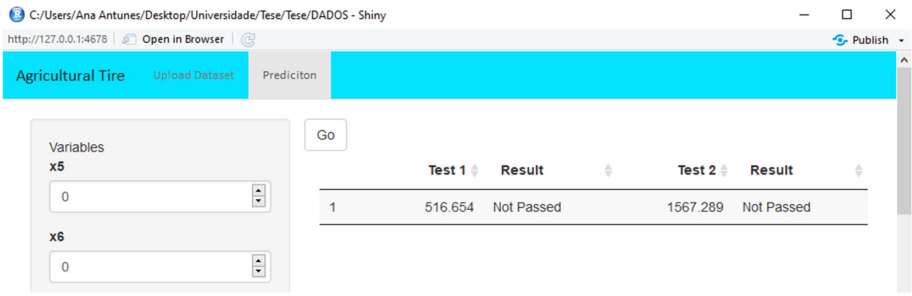

Fig. 14. Making predictions.

The chosen models for Test 1 and Test 2 use principal components that are a linear combination of the initial variables and for this reason it is fundamental to insert the 27 initial variables and the 4 qualitative variables. Therefore, when the variable is quantitative the user has to introduce the value, and when it is qualitative he has to select the pretended level. To make predictions, all the variables have to be filled and after that the results appear when the button "Go" is clicked. The application gives the results for Test 1, y_1, and Test 2, y_2. In addition, the "Result" (Fig. 14) indicates if the tire passed the test. For the production of agricultural tires it is necessary to consider legal norms and both tests have a maximum that cannot be exceeded. When the result is greater than the maximum, the tires do not pass the test, the specification has to be modified and in "Result" appears "Not passed". Otherwise, the agricultural tire passes the test and in "Result" appears "Passed".

4 Conclusion

The main goal was to apply multivariate analysis to variables related to tire production and identify the influences on the two tires tests. In the exploratory analysis it was possible to identify strong correlations between the quantitative variables, including the response variables for each test. With the variance inflation factor, it was possible to identify the existence of multicollinearity between quantity variables and this could be a problem when applying linear regression.

Principal component analysis was used to eliminate multicollinearity effects and to retain as much information as possible to apply to the models. For this reason, it was decided to use the 27 principal components and it was difficult to understand the meaning of each principal component considering the loadings' values.

Multiple linear regression was used to identify the significant variables to improve the agricultural tire production. This was also difficult to identify because we considered the 27 principal components and the qualitative variables. One of the objectives of this study was to find a multiple linear regression for the two tests. For the selection of variables we used Stepwise methods and the choice of the model to be considered was made taking into account the AIC value.

After obtaining the models for the two tests, an application was developed in Shiny in order to quickly and efficiently determine the test results for future agricultural tires. By using the application it is possible to reduce the quantity of materials and resources as there is an increase in efficiency and profits since this application can predict the performance of the tire before starting its production. In addition, reducing the industrialization time is also an advantage, because some specifications can be canceled before the production phase. It also helps to preserve the environment by reducing the destruction of tires with bad performances. Therefore, this application helps the users to select the best specification for the agricultural tire, thus generating more security in the specification to be used and enabling a reduction of errors by the research and development department.

Acknowledgments. This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

References

1. Burnham, K.P., Anderson, D.R.: Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, 2nd edn. Springer, New York (2002). https://doi.org/10.1007/b97636
2. Chattefuee, S., Hadi, A.S.: Regression Analysis by Example. 4. edn. (2006)
3. Chatterjee, S., Simonoff, J.S.: Handbook of Regression Analysis (2013)
4. Johnson, R., Wichern, D.: Applied Multivariate Statistical Analysis, 5th edn. Prentice Hall, Upper Saddle River (2002)
5. Kutner, M.H., Nachtsheim, C.J., Neter, J., Li, W.: Applied Linear Statistical Models, 5th edn. McGraw-Hill Irwin, New York (2005)
6. Lafi, S.Q., Kaneene, J.B.: An explanation of the use of principal-components analysis to detect and correct for multicollinearity. Preventive Veterinary Medicine (1992)
7. Maxwell, O., Amaeze Osuji, G., Precious Onyedikachi, I., Obi-Okpala, C.I., Udoka Chinedu, I., Ikenna Frank, O.: Handling critical multicollinearity using parametric approach. Acad. J. Appl. Math. Sci. **5**(11), 150–163 (2019)
8. Mishra, S.P., et al.: Multivariate statistical data analysis- Principal Component Analysis (PCA). Int. J. Livestock Res. **7**(5), 60–78 (2017)
9. Rawlings, J.O., Pantula, S.G., Dickey, D.A.: Applied Regression Analysis: A Research Tool. Springer Texts in Statistics, 2nd edn. Springer, Heidelberg (1998). https://doi.org/10.1007/b98890