|  |  |  |
| --- | --- | --- |
|  |  | PREDECTIVE ANALYSIS USING LOGISTIC REGRESSION |
|  |  | Objective: |
|  |  | The main goal of this project is to clean the extracted data and briefly analyse the given variables to understand their types and create an appropriate model using any of the statistical models like, logistic regression ,decision tree in this project we are using logistic regression and we perform exploratory analysis, cross tabulation , co-relation, step wise regression, confusion matrix, ROC curve(Receiver Operating characteristic) and then we create a model using most significant variables in train data to predict whether a person is having cancer or not and we test the model by using test data also we visualize the information using a web interface like r shiny in that we represent co- relation with every variables. |
|  |  | Summary: |
|  |  | Exploratory data analysis (EDA) is the first step of any data analysis. EDA is used to analyze data types, outliers, missing values, and distributions for both numerical and categorical data. Summary, structure, Boxplot, Histograms, Barplots and correlation plots were utilized for EDA for this project. The above functions were performed to understand the structure and distribution of the Cancerdata to analyze for patterns. Upon performing EDA, Cancerdata set was used train the model for predicting the model. Diagnosis was used as dependent variable and rest of the variables in the data were used as independent variable. Once the model was trained, confusion matrix was used to check for the summary of the model. The trained model was then used to predict the whether a person is having cancer or not of the test data set. The model predicted the test data with an accuracy of around 97 percent. R shiny was used to represent the data to give a better understanding of the data very easily and without having to write the code. The ui fuction was used to create a user interface or visual representation for the data and server function was used to perform the necessary action. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | CODE ANALYSIS |
|  |  | Data cleaning |
|  |  | Generally, data is taken from various sources and it can be present in many formats and also have outliers and empty blocks for that cleaning is required and converting the data into one format. |
|  |  | Importing data into R environment: We converted the excel file into CSV (comma separated value).and then used R’s built in functionality “read.csv” to read and manipulate the data. Here “file. choose ()”, which allows us the file selection to be limited to files of specific types. |
|  |  |  |
|  |  | In the above snippet we created object cancerdata and loaded the data into that object and then str( )is used to know number of rows, columns and importantly classes of each column. |
|  |  | preparing data |
|  |  |  |
|  |  | From the above snippet in dataset the variable is (id no) it not required for model creation so removing first variable (cancerdata[,-1]) and next attaching data to r environment using attach()function and then diagnosis variables are in the form of “M”,”B” so for our calculation creating a new column called diagnosis1 and converting diagnosis into binary digits (1,0) now delete the diagnosis variable and bind new column to dataset using cbind() then by using names() we can see the variable names |
|  |  | EXPLORATORY DATA ANALYSIS |
|  |  | BOX PLOT: It is a visual representation of five summary details and it detects the commonly plot outliers that go beyond the bulk of the data. |
|  |  | R makes it easy to combine multiple plots into one overall graph using par().With the par() function you can include the options mfrow=c(nrows,nclos) to create a matrix of nrows X ncols plots that are filled in by rows. |
|  |  |  |
|  |  | HISTOGRAM: These are useful to look at when we want to see more details on full distribution of data. Here we can see the high concentration of values in the data set. |
|  |  |  |
|  |  |  |
|  |  | BAR PLOT: These are useful for visualizing categorical data with the number of entries for each category being proportional to the height of bar. |
|  |  | In above snippet barplot() is used to count how many 1’s and 0’s are present in the given data. |
|  |  |  |
|  |  | CO-RRELATION PLOT: Correlation is used to describe linear relationship between two continuous variables. In general, it measures the strength and direction of linear relationship between two or more variables. |
|  |  | “ggplot2” this library to declaratively make beautiful plots or charts of our data it allows us to create both univariate and multivariate numerical and categorical data in a straight forward manner |
|  |  | “ggcorrplot” this package can be used to visualize easily a correlation matrix using ggplot2 .it provides a solution for reordering the correlation matrix and displays the significance level of plot. |
|  |  |  |
|  |  |  |
|  |  | SPLITTING DATA: |
|  |  | In this project we used “caTools” package to split the given data into two parts as training and testing and executed the model whether it’s working or not. |
|  |  | First splitting data into two parts for training and testing the model we create |
|  |  |  |
|  |  | In above snippet sample.split( ) function is used to split data randomly in ratio of 80% and 20% |
|  |  | STEP-WISE REGRESSION: |
|  |  | The stepwise regression (or stepwise selection) consists of iteratively adding and removing predictors, in the predictive model, in order to find the subset of variables in the data set resulting in the best performing model that is a model that lowers prediction error. |
|  |  | There are three strategies of stepwise regression |
|  |  | 1. Forward selection, which starts with no predictors in the model, iteratively adds the most contributive predictors, and stops when the improvement is no longer statistically significant. |
|  |  | 2. Backward selection (or backward elimination), which starts with all predictors in the model (full model), iteratively removes the least contributive predictors, and stops when you have a model where all predictors are statistically significant. |
|  |  | 3. Stepwise selection (or sequential replacement), which is a combination of forward and backward selections. You start with no predictors, then sequentially add the most contributive predictors (like forward selection). After adding each new variable, remove any variables that no longer provide an improvement in the model fit (like backward selection |
|  |  |  |
|  |  | In this project we use forward stepwise regression |
|  |  | For execution of this we used a package called “MASS” in R. |
|  |  |  |
|  |  | Using the above code, we get formula using stepwise forward regression that considers most significant variables in the data. |
|  |  |  |
|  |  | LOGISTIC REGRESSION: |
|  |  | glm is the built in function in R, used for creating a logistic model in r programming language |
|  |  | We create a model using the formula obtained by calculated using step-wise regression |
|  |  |  |
|  |  | Know after creating a model by using trainingdata we predict the model analysis using predict fuction in R to test the testdata. |
|  |  |  |
|  |  | CREATING TABLE: |
|  |  | This code creates a table using premodel value if premodelvalue lessthan 0.4 it is “B”, else it is “M” |
|  |  |  |
|  |  |  |
|  |  | CONFUSION MATRIX |
|  |  | A confusion matrix is a technique for summarizing the performance of a classification algorithm. Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making. |
|  |  | This gives us: |
|  |  | • “True positive” for correctly predicted event values. |
|  |  | • “False positive” for incorrectly predicted event values. |
|  |  | • “True negative” for correctly predicted no-event values. |
|  |  | • “False negative” for incorrectly predicted no-event values. |
|  |  |  |
|  |  |  |
|  |  | Events No-event |
|  |  | Event true positive false positive |
|  |  | No-event false negative true negative |
|  |  |  |
|  |  |  |
|  |  | In the above code we created a confusion matrix using premodel |
|  |  | table(actualvalue=training$diagnosis1,predictedvalue=premodel1>0.4) |
|  |  |  |
|  |  |  |
|  |  | ROC CURVE (receiver operating characteristic curve): |
|  |  | ROC curve is a complete sensitivity/specificity report. The ROC curve is a fundamental tool for diagnostic test evaluation. |
|  |  | In a ROC curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (Specificity) for different cut-off points of a parameter. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The area under the ROC curve (AUC) is a measure of how well a parameter can distinguish between two diagnostic groups (diseased/normal). |
|  |  | Sensitivity and specificity versus criterion value: |
|  |  | When you select a higher criterion value, the false positive fraction will decrease with increased specificity but on the other hand the true positive fraction and sensitivity will decrease: |
|  |  |  |
|  |  | When you select a lower threshold value, then the true positive fraction and sensitivity will increase. On the other hand the false positive fraction will also increase, and therefore the true negative fraction and specificity will decrease. |
|  |  | The ROC curve |
|  |  | In a Receiver Operating Characteristic (ROC) curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. A test with perfect discrimination (no overlap in the two distributions) has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity). Therefore the closer the ROC curve is to the upper left corner, the higher the overall accuracy of the test. |
|  |  |  |
|  |  |  |
|  |  | Code executed in r program for ROC Curve |
|  |  |  |
|  |  | The above code draws an ROC curve using true positive and false positive rate so that we can visually see where there is more deviation and fix threshold value. |
|  |  |  |
|  |  | FINDING ACCURACY OF MODEL: |
|  |  | Accuracy of model is created, dividing (true positive+ true negative) by (true positive+ false negative+ false positive+ true negative) |
|  |  | Below is the snippet executed in r console. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | CODE USED IN RSHINY FOR DATA EDA: |
|  |  | library(shiny) |
|  |  | # Define UI for application that draws a histogram |
|  |  | ui <- fluidPage( |
|  |  | # Application title |
|  |  | titlePanel("cancer data"), |
|  |  | # Sidebar with a slider input for number of bins |
|  |  | sidebarLayout( |
|  |  | sidebarPanel( |
|  |  | selectInput("x","select the fields to create histogram",choices = names(cancerdata1)), |
|  |  | radioButtons("s","select x-axis:", list("radius\_mean"="a1","texture\_mean"='b1',"perimeter\_mean"="c1","area\_mean"="d1","smoothness\_mean"="e1","compactness\_mean"="f1","concavity\_mean"="g1","concave.points\_mean"="h1","symmetry\_mean"="i1","fractal\_dimension\_mean"="j1","radius\_se"="k1","texture\_se"="l1","perimeter\_se"="m1","area\_se"="n1","smoothness\_se"="o1","compactness\_se"="p1","concavity\_se"="q1","concave.points\_se"="r1","symmetry\_se"="s1","fractal\_dimension\_se"="t1","radius\_worst"="u1","texture\_worst"="v1","perimeter\_worst"="w1","area\_worst"="x1","smoothness\_worst"="y1","compactness\_worst"="z1","concavity\_worst"="a11","concave.points\_worst"="b11","symmetry\_worst"="c11","fractal\_dimension\_worst"="d11","diagnosis"="e11")), |
|  |  | radioButtons("k","select y-axis:", list("radius\_mean"="a2","texture\_mean"='b2',"perimeter\_mean"="c2","area\_mean"="d2","smoothness\_mean"="e2","compactness\_mean"="f2","concavity\_mean"="g2","concave.points\_mean"="h2","symmetry\_mean"="i2","fractal\_dimension\_mean"="j2","radius\_se"="k2","texture\_se"="l2","perimeter\_se"="m2","area\_se"="n2","smoothness\_se"="o2","compactness\_se"="p2","concavity\_se"="q2","concave.points\_se"="r2","symmetry\_se"="s2","fractal\_dimension\_se"="t2","radius\_worst"="u2","texture\_worst"="v2","perimeter\_worst"="w2","area\_worst"="x2","smoothness\_worst"="y2","compactness\_worst"="z2","concavity\_worst"="a12","concave.points\_worst"="b12","symmetry\_worst"="c12","fractal\_dimension\_worst"="d12","diagnosis"="e12")) |
|  |  | ), |
|  |  | # Show a plot of the generated distribution |
|  |  | mainPanel( |
|  |  | plotOutput("CD"), |
|  |  | plotOutput("distPlot") |
|  |  | ) |
|  |  | ) |
|  |  | ) |
|  |  | # Define server logic required to draw a histogram |
|  |  | server <- function(input, output) { |
|  |  | output$CD<-renderPlot({ |
|  |  | hist(cancerdata1[,input$x],col = rainbow(10)) |
|  |  | }) |
|  |  | output$distPlot <- renderPlot( |
|  |  | { |
|  |  | if(input$s=='a1'){i<-1} |
|  |  | if(input$s=='b1'){i<-2} |
|  |  | if(input$s=='c1'){i<-3} |
|  |  | if(input$s=='d1'){i<-4} |
|  |  | if(input$s=='e1'){i<-5} |
|  |  | if(input$s=='f1'){i<-6} |
|  |  | if(input$s=='g1'){i<-7} |
|  |  | if(input$s=='h1'){i<-8} |
|  |  | if(input$s=='i1'){i<-9} |
|  |  | if(input$s=='j1'){i<-10} |
|  |  | if(input$s=='k1'){i<-11} |
|  |  | if(input$s=='l1'){i<-12} |
|  |  | if(input$s=='m1'){i<-13} |
|  |  | if(input$s=='n1'){i<-14} |
|  |  | if(input$s=='o1'){i<-15} |
|  |  | if(input$s=='p1'){i<-16} |
|  |  | if(input$s=='q1'){i<-17} |
|  |  | if(input$s=='r1'){i<-18} |
|  |  | if(input$s=='s1'){i<-19} |
|  |  | if(input$s=='t1'){i<-20} |
|  |  | if(input$s=='u1'){i<-21} |
|  |  | if(input$s=='v1'){i<-22} |
|  |  | if(input$s=='w1'){i<-23} |
|  |  | if(input$s=='x1'){i<-24} |
|  |  | if(input$s=='y1'){i<-25} |
|  |  | if(input$s=='z1'){i<-26} |
|  |  | if(input$s=='a11'){i<-27} |
|  |  | if(input$s=='b11'){i<-28} |
|  |  | if(input$s=='c11'){i<-29} |
|  |  | if(input$s=='d11'){i<-30} |
|  |  | if(input$s=='e11'){i<-31} |
|  |  | if(input$k=='a2'){j<-1} |
|  |  | if(input$k=='b2'){j<-2} |
|  |  | if(input$k=='c2'){j<-3} |
|  |  | if(input$k=='d2'){j<-4} |
|  |  | if(input$k=='e2'){j<-5} |
|  |  | if(input$k=='f2'){j<-6} |
|  |  | if(input$k=='g2'){j<-7} |
|  |  | if(input$k=='h2'){j<-8} |
|  |  | if(input$k=='i2'){j<-9} |
|  |  | if(input$k=='j2'){j<-10} |
|  |  | if(input$k=='k2'){j<-11} |
|  |  | if(input$k=='l2'){j<-12} |
|  |  | if(input$k=='m2'){j<-13} |
|  |  | if(input$k=='n2'){j<-14} |
|  |  | if(input$k=='o2'){j<-15} |
|  |  | if(input$k=='p2'){j<-16} |
|  |  | if(input$k=='q2'){j<-17} |
|  |  | if(input$k=='r2'){j<-18} |
|  |  | if(input$k=='s2'){j<-19} |
|  |  | if(input$k=='t2'){j<-20} |
|  |  | if(input$k=='u2'){j<-21} |
|  |  | if(input$k=='v2'){j<-22} |
|  |  | if(input$k=='w2'){j<-23} |
|  |  | if(input$k=='x2'){j<-24} |
|  |  | if(input$k=='y2'){j<-25} |
|  |  | if(input$k=='z2'){j<-26} |
|  |  | if(input$k=='a12'){j<-27} |
|  |  | if(input$k=='b12'){j<-28} |
|  |  | if(input$k=='c12'){j<-29} |
|  |  | if(input$k=='d12'){j<-30} |
|  |  | if(input$k=='e12'){j<-32} |
|  |  | cancerdata=read.csv("C:/Users/Admin/Documents/cancer123.csv") |
|  |  | x<-cancerdata[,i] |
|  |  | y<-cancerdata[,j] |
|  |  | plot(x,y) |
|  |  | }) |
|  |  | } |
|  |  | # Run the application |
|  |  | shinyApp(ui = ui, server = server) |
|  |  |  |
|  |  |  |
| . |  |  |
|  |  | 6.CONCLUSION  This was performed on the data set and logistic regression was used to create a model using train data set. The obtained model was tested on the test data and an accuracy of around 97% was achieved. R-Shiny Web App was used to represent the data graphically. |