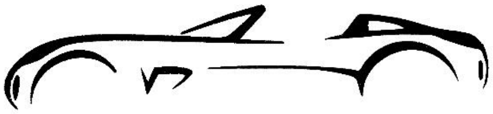


Car-Affordability

Project Report

Ebin Lazer Paiva

X13112635

ebin.paiva@gmail.com

National College of Ireland

BSc (Hons) in Computing

2016/2017

Declaration Cover Sheet for Project Submission

Name: Ebin Lazer Paiva
Student ID: X13112635
Supervisor: Keith Maycock & Paul Stynes

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: ____

Date: ____________

Abstract

The automotive industry has been booming since the last decade. We are focussing on the car sales industry with the development of this application. It is estimated that every household has at least one car within the boundaries of Ireland. There we found a market gap where a potential car buyer could find help in making the right decision.

The application is developed to cater needs of a potential car buyer. The buyer or the user of this application is able to get useful information regarding the car they intend to buy.

The users of this application can gather valuable information based on the analysis of the cars and the manufacturers, and the car sales over the years to make an inference and thereby able to make the right decision while buying a car.

It also focuses on the predictive analysis of the car sales with respect to the analysis of the sales over the past years in Ireland. This aspect will cater the needs of a businessman who intends to become a car dealer in making the right choice with the respect to the area and the make of cars that he intends to take the dealership.

Contents

Introduction

Car-Affordability, the web application, is developed with the car buyers as the primary audience. The user can gain access to the functionalities of the application and get the useful information that would help to make the apt decision while buying car.

There appears to have considerable amount of people who sell their car in not less than a year's gap. A few find it difficult maintain as the insurance cost, taxes, fuel cost, and service cost are considerably beyond their limits. This application is aimed at those people who falls in the middle group of income scale.

People would buy a car for travel to work, business, or for leisure drives. Hence by using this application, users can gain access to sufficient information to make the right decision.

Background

An article published in a government website (www.igees.gov.ie) states that 75% of all journeys in the year 2013 in Ireland has been made using cars. Compared to other modes of transport, the above figure is considerably huge. The usage of cars has been on the rise over the years, 2011-2013, by 1.2%.

Statistical analysis shows a 26.3% increase in sales of private cars in the first half of 2015. The increased sales show that there is a huge space for a web application like ours with all the potential car buyers as the target market. As per Irish Journal, 40,000 cars were registered in the month of January 2015. Compared to the previous month, there has been a 33% increase. Every possible car buyer can be a potential user for our web application.

Our target market includes both new and used cars; hence the audience will be considerably huge.

The application is developed foreseeing the market gap that exists in the sales sector of cars. There is no web application currently in the market that can aid a potential car buyer to help while they are in a dilemma whether to go for it or not.

When you can have accurate information regarding a specific car the buyer seems to have enough knowledge to confirm that he is making the right choice, which in turn can avoid regrets at a later stage.

Opportunities for this application is vast with the sales sector of cars in Ireland is immensely huge.

At the start, I felt that it was a great idea to build an app to check the expenses on owning their desired car. For the final year project, I decided to work from where we left during the internship. I am planning to have additional features for comparing cars so that a distinctive comparison can be made. Users can view the price for their desired model, a fresh one from showroom. Reviews on various cars will be available. Additional links for insurance brokers and helpful tips will be available. Data analysis on car sales over the years will be available and the users can view the most sold cars as well as the least. Some predictive analysis on that will also be made available These are the ideas that I explained during the pitch.

I will try to integrate twitter into the web application. I think I can do much more by channeling live data and analyzing the trends out in the public. It will aid in analyzing the real-time sentiment analysis

Sales data of all the cars in Ireland has been collected. Each file of sales data has been collected by the respective years. Integration of the data for all the years into one single database was achieved.

Front end modification, the home page was made a bit more clutter free by removing few unwanted texts and items and creating a review section where car-model reviews are available. And thats how the project started.

Objectives

The main aims of the development of this application are

- Check the affordability of their desired car.
- Compare the cost of maintenance of cars.
- Analyse the trends in car sales.
- Check the most selling car model.
- Check the car sales with respect to counties.
- Review of car models by car experts
- Check the price for new cars.

The main objective of this project is to build a web application to help potential car buyers in making the right decision. The web application will allow users to check on the expenses that might be incurred if they buy the car of their choice. Thus the users will be able to do an affordability check before they buy the car.

When a car is bought, whether used or new, there are various expenses that needs to be met to keep the car roadworthy. Tax, insurance, fuel cost, NCT, and maintenance cost, etc. are some of them.

Users will be able to compare to cars, which will aid in evaluating cars in detail so that proper decision making could be possible. Price of the desired car will be shown alongside if they are looking for a new car and details of the dealers will be also made available.

Detailed analysis of car sales by years will be done and graphical representation will be shown for users to review and analyse for any particular trends.

Independent reviews on individual cars will be available for users.

Technologies

The technologies that are used in the development of this application are listed in detail below.

Bootstrap

Bootstrap is one of the most popular front-end framework that is widely used by most of the developers in web development. Bootstrap is open source and easy to use. Responsive and fluidic front end can be developed using the Bootstrap templates.

PHP

PHP is a scripting language which is used in the back end in web development. It is a server side technology. In this project, PHP files are used to connect to the MySQL database. PHP files are also used for Twitter API connection to provide the real-time sentiment analysis on car manufacturers.

MySQL

MySQL is database framework that allows us to store data in a relational architecture. MySQL database is used to store data of car sales as well as car information. The information required for the users are fetched by querying the database.

R Studio

R Studio is an integrated development environment (IDE) for R programming language. R programming language is a powerful tool for statistical computing. It also provides graphical representations. R studio is free and open source.

Twitter

Twitter is the most popular microblogging website that is most common means of social networking. Twitter data can be retrieved and analysed to make useful interpretations which can be beneficial for the business.

Google Charts

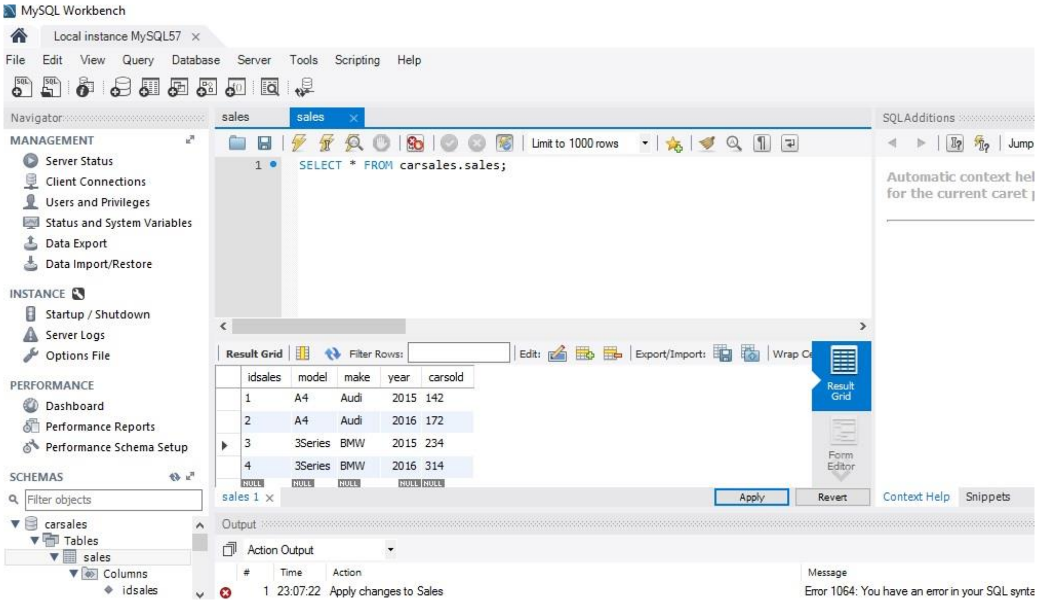
Google Charts are used to populate charts by the results of the PHP files after querying the database. The codes are open sourced and is available for use for developers.

Methodology

Bootstrap has been easy to use and provided a fluidic and responsive design. Bootstrap is chosen as the skeleton for our front end.

A sample dataset on cars was selected. Work started on designing the database with appropriate tables. Linking tables across is under way as well as normalization of the database.

Database on cars will be having 4 tables, make, model, engine size, and fuel efficiency. Normalizing the data might reduce one table by combining the engine size and fuel efficiency table.



```
SELECT * FROM carsales.sales;
```

idsales	model	make	year	carsold
1	A4	Audi	2015	142
2	A4	Audi	2016	172
3	3Series	BMW	2015	234
4	3Series	BMW	2016	314
NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message
1	23:07:22	Apply changes to Sales	Error 1064: You have an error in your SQL synta

The sales data will be used for analyzing the trends of sales and to populate it.

- A car manufacturer’s sales performance over the years.

- Sales based on individual counties.
- Sales based on a specific make.

I believe this information are relevant to any potential car buyer in understanding the trends over the past years. Even a businessman who wants to be car dealer can open the showroom with respect to the analysis of the sales, where a specific manufacturer has loyal customers in a specific place or county.

Technical Approach

The basis of the website will be the database of cars. The data of cars that has been collected as a CSV file will be written to MySQL database using PHP codes. Implementing the database with appropriate tables for cars including makes, models, engine size, etc. will be one of the primary tasks.

The user will be able to input the car details and query the database for necessary information. An algorithm which will process the information and provide an estimated total expense as output. Twitter sentiments for the particular make and model will analysed and graphical representation will be made available.

Database on previous years' car sales will be used for more data analysis to provide information on any particular trends that has been observed.

Technical Details

The web application will be build using HTML5, PHP, JavaScript, CSS, MYSQL, and Bootstrap.

Initial Requirements

Functional requirements

Navigating through the website

- User friendly interface.
- Ease of navigation through the web pages.

- Form submission for input of car details of the user should be flawless.
- Each form input should be processed appropriately to provide the options for the next input.
- The output should be displayed in appropriate manner for users of all ages.

Data requirements

Database

- Database should have consistency, availability, and integrity.
- Querying should be possible to read the required data.
- Database should not be altered by the user.
- Database should provide only read-only functionality for the users.
- Administrator should have the authorization to do CRUD operations.

User requirements

The users are required to use a desktop computer or a mobile device with internet connection. A web browser should be installed in the device to browse the website. The users are not required to sign up and log in. All the services and functionalities of this website are open to all users straight away.

Environmental requirements

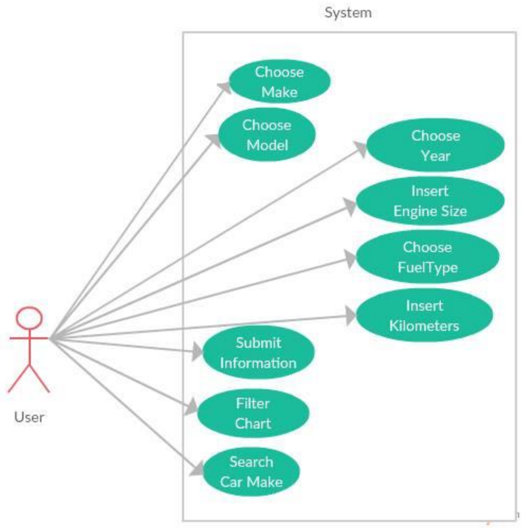
User interface should be developed with respect to the environmental constraints of the users. The users can be using the mobile devices and may outside in the bright light. The use of colours in designing the web application will be made keeping all these aspects in mind.

Usability requirements

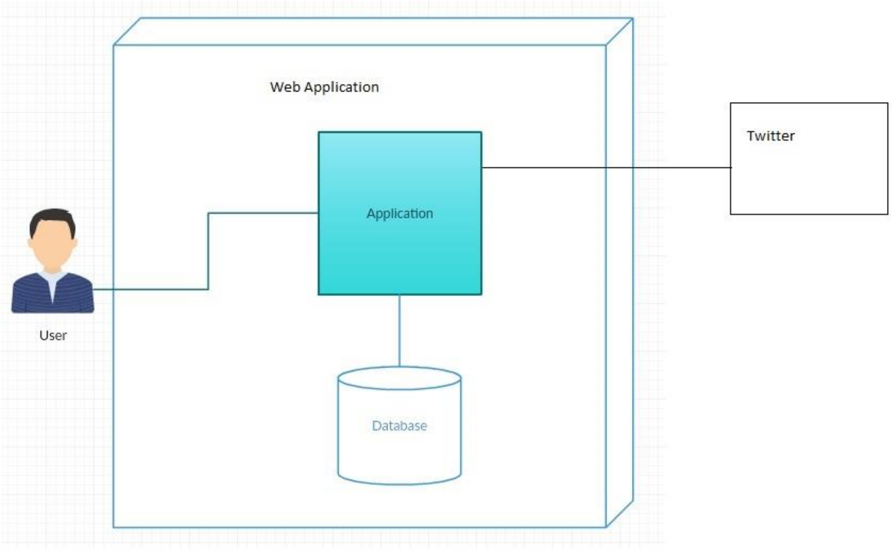
The usability requirements include ease of navigation through the webpages and a user-friendly interface. Contents should be vivid and clear so that it is comprehensible for users of all ages. Consistency should be maintained.

Design and Architecture

The use case model for the car maintenance cost check function has been displayed below.



The architectural representation of the system is shown below. The MySQL database serves at the back end with the data that is required to process the user's input. Twitter data is used to analyse the sentiments.



Implementation of functionalities

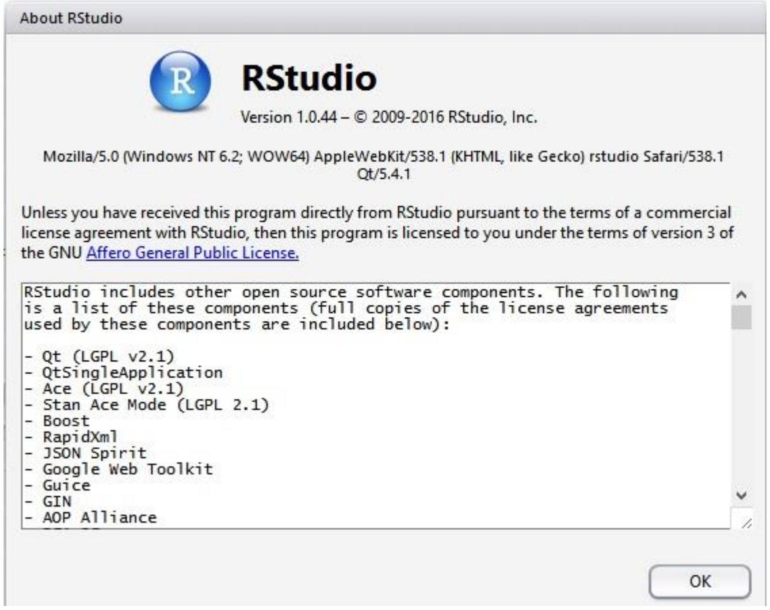
Car Sales Analysis

Analysing the car sales data with respective to each county has been retrieved from the SIMI (Society for Irish Motor Industry) website. The car sales data has been categorically accessed by year as well as by each county.

Method

We are using the multiple linear regression model to analyse the car sales and to derive some coefficients that could aid in predictive analysis. We are using R studio for this purpose, which is a very powerful and simple IDE for working with statistics in R language.

The version of R studio that was used for this project is:



Version 1.0.44 – © 2009-2016 RStudio, Inc.

Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/538.1 (KHTML, like Gecko) rstudio Safari/538.1 Qt/5.4.1

Unless you have received this program directly from RStudio pursuant to the terms of a commercial license agreement with RStudio, then this program is licensed to you under the terms of version 3 of the GNU Affero General Public License.

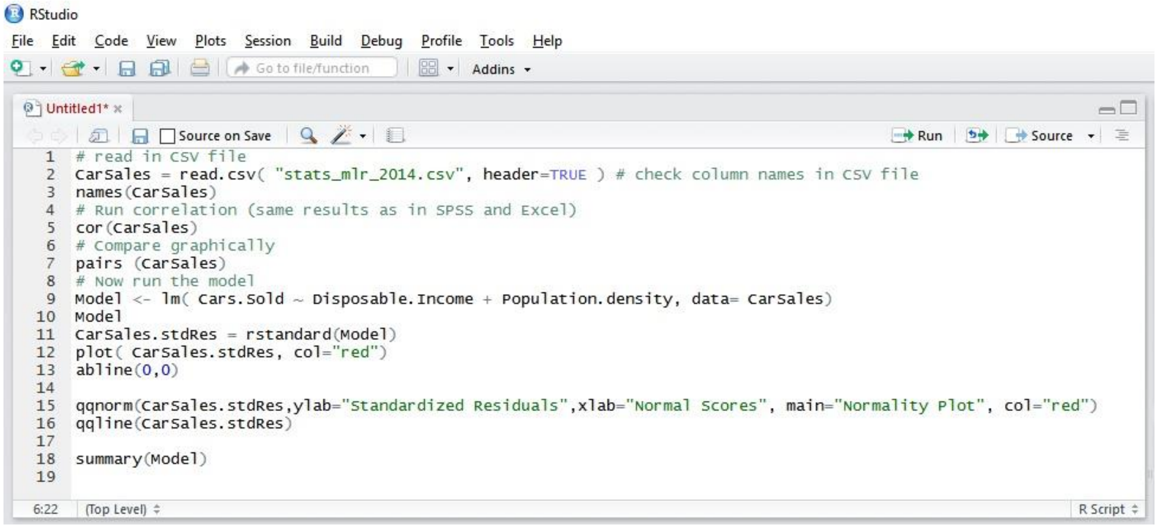
```
RStudio includes other open source software components. The following
is a list of these components (full copies of the license agreements
used by these components are included below):

- Qt (LGPL v2.1)
- QtSingleApplication
- Ace (LGPL v2.1)
- Stan Ace Mode (LGPL 2.1)
- Boost
- RapidXml
- JSON Spirit
- Google Web Toolkit
- Guice
- GIN
- AOP Alliance
```

Data collection

The variables for the dataset are Car Sold, Disposable income, and Population Density. The data for Cars Sold were gathered from the SIMI website and the data for the Disposable income and population density were collected from the CSO (Central Statistical Office) website.

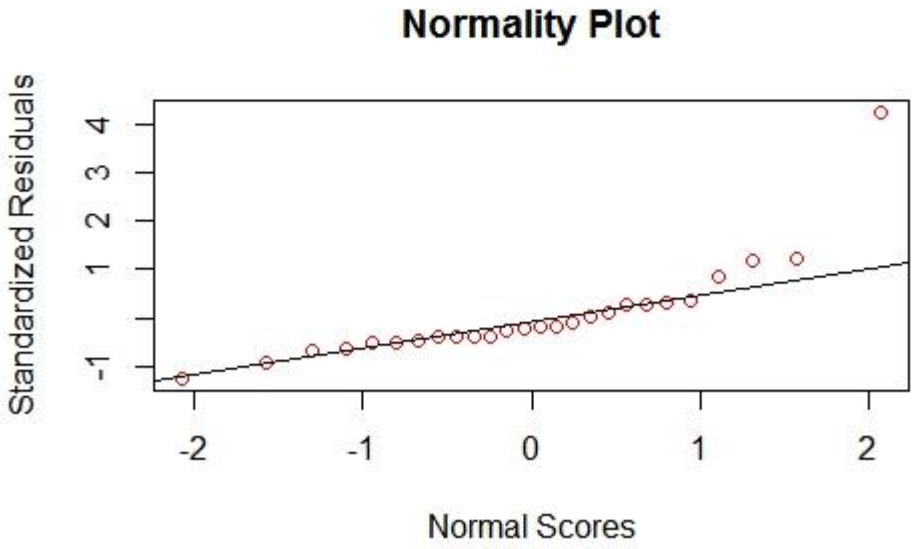
R Codes

The R script for the multiple linear regression model is displayed below.

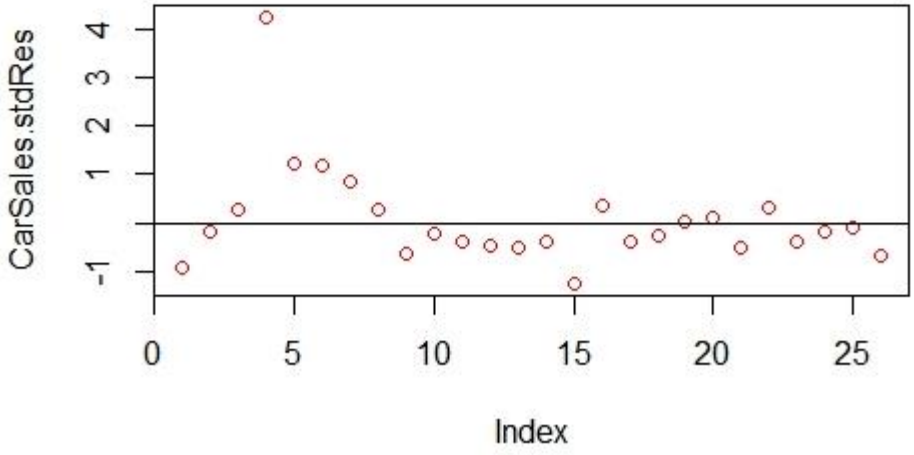


```
# read in csv file
CarSales = read.csv( "stats_mlr_2014.csv", header=TRUE ) # check column names in csv file
names(CarSales)
# Run correlation (same results as in SPSS and Excel)
cor(CarSales)
# Compare graphically
pairs (CarSales)
# Now run the model
Model <- lm( Cars.Sold ~ Disposable.Income + Population.density, data= CarSales)
Model
CarSales.stdRes = rstandard(Model)
plot( CarSales.stdRes, col="red")
abline(0,0)

qqnorm(CarSales.stdRes,ylab="Standardized Residuals",xlab="Normal Scores", main="Normality Plot", col="red")
qqline(CarSales.stdRes)

summary(Model)

```

The normality test for the distribution of the data is checked. The data points seem to fall on the line. Hence the data is believed to be normally distributed.

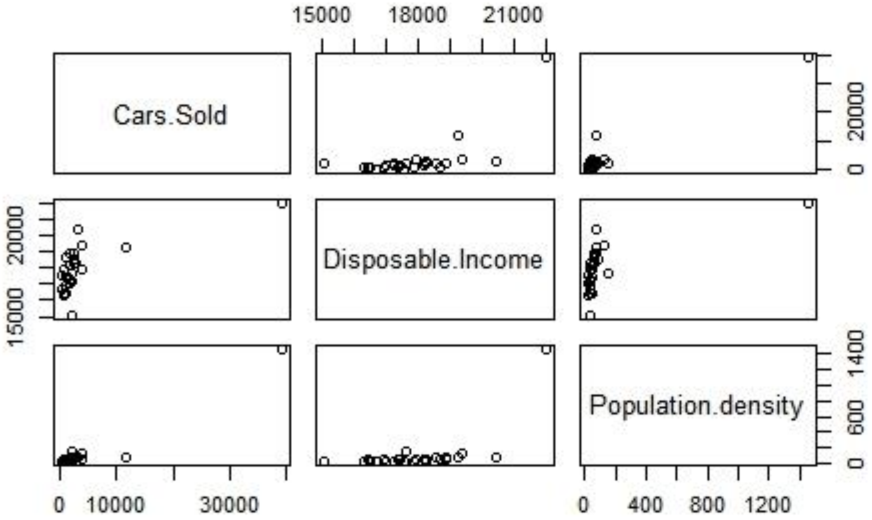


QQ Plot



The QQ plot shows that most of the standardised residuals fall in between 2 standard deviations, i.e. between the values -2 and +2.

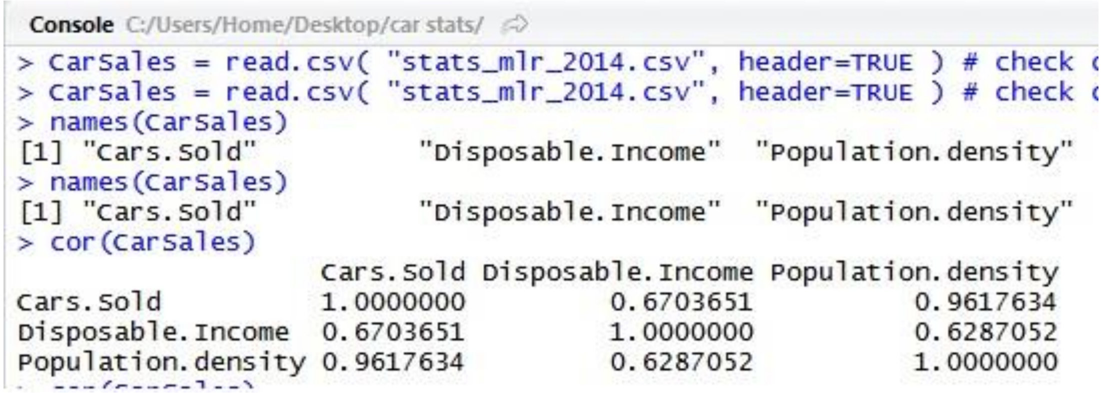
Pairs Plot



Here we compare each variable graphically in a single plot.

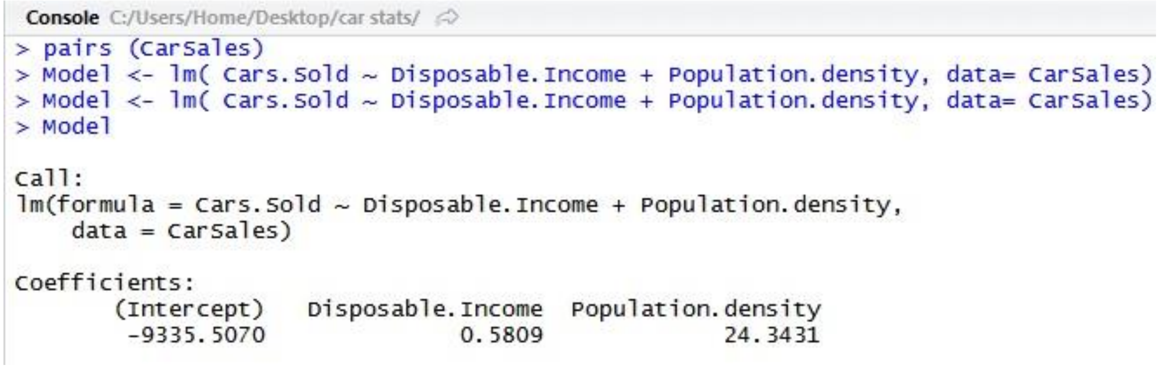
Correlation Matrix

The correlation matrix for the car sales data shows how each variable are correlated. The disposable income seems to have moderate correlation (0.67) with number of cars sold. The population density shows a very high correlation (0.96) with the number of cars sold.



```
> CarSales = read.csv( "stats_mlr_2014.csv", header=TRUE ) # check c
> CarSales = read.csv( "stats_mlr_2014.csv", header=TRUE ) # check c
> names(CarSales)
[1] "Cars.Sold"          "Disposable.Income"   "Population.density"
> names(CarSales)
[1] "Cars.Sold"          "Disposable.Income"   "Population.density"
> cor(CarSales)
                   Cars.Sold Disposable.Income Population.density
Cars.Sold          1.0000000         0.6703651          0.9617634
Disposable.Income  0.6703651         1.0000000          0.6287052
Population.density 0.9617634         0.6287052          1.0000000
> cor(CarSales)
```

The population density shows a very high correlation and the disposable income a moderate correlation. The correlation among the independent variables, i.e. the disposable income and the population density is moderate at 0.62. Hence, we cannot dispose the absence of multicollinearity. In the event of the multicollinearity it is advised that one of the independent variable be dropped if two of the independent variables shows high collinearity with each other.



```
> pairs (CarSales)
> Model <- lm( Cars.Sold ~ Disposable.Income + Population.density, data= CarSales)
> Model <- lm( Cars.Sold ~ Disposable.Income + Population.density, data= CarSales)
> Model

Call:
lm(formula = Cars.Sold ~ Disposable.Income + Population.density,
    data = CarSales)

Coefficients:
       (Intercept)   Disposable.Income  Population.density
        -9335.5070              0.5809             24.3431
```

Coefficients of the model

Y = dependent variable

a = Y intercept

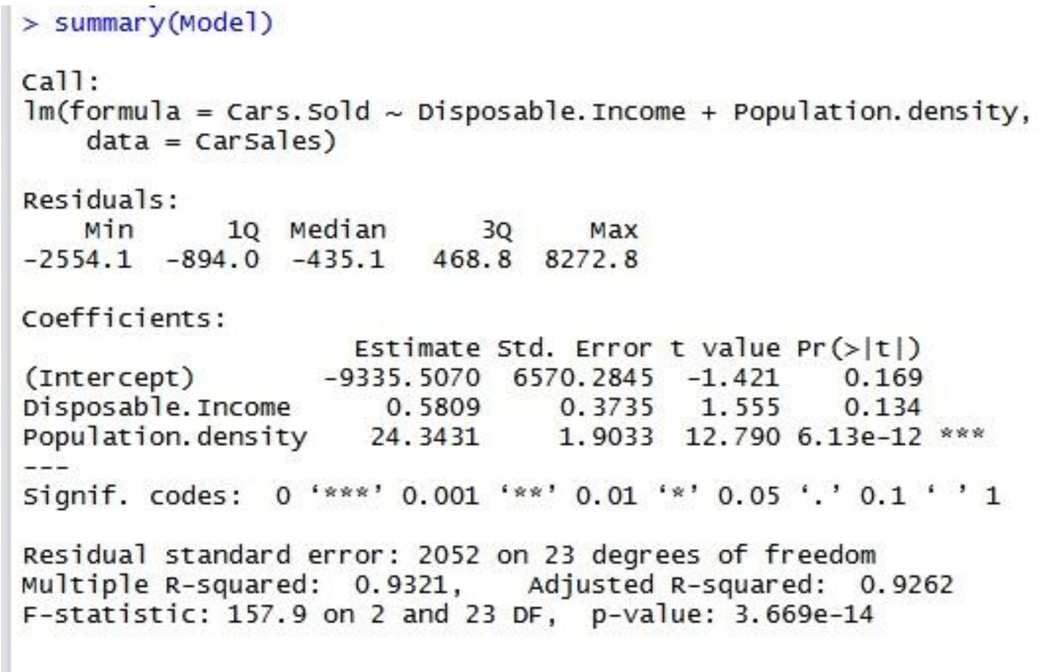
b = slope of the regression line

x1 = independent variable 1

x2 = independent variable 2

y = a + bx1 + bx2

= -**9335.507 + 0.580 x1 + 24.343 x2**



```
> summary(Model)

Call:
lm(formula = Cars.Sold ~ Disposable.Income + Population.density,
    data = CarSales)

Residuals:
    Min      1Q  Median      3Q     Max 
-2554.1  -894.0  -435.1   468.8  8272.8 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -9335.5070  6570.2845  -1.421    0.169    
Disposable.Income     0.5809     0.3735   1.555    0.134    
Population.density   24.3431     1.9033  12.790 6.13e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2052 on 23 degrees of freedom
Multiple R-squared:  0.9321,	Adjusted R-squared:  0.9262 
F-statistic: 157.9 on 2 and 23 DF,  p-value: 3.669e-14
```

The Multiple R-squared value, which is known as the coefficient of determination stand as a measure to describe the effectiveness of our model as a whole in explaining the values of the dependent variable.

The adjusted R square which is a modified and corrected value of the multiple R squared value is 0.926. The model seems to be of good strength and it describes 92% of the variations in the dependent variable in our model.

Reporting the result

F(2,23) = 157.9, p < .005 at α = .05

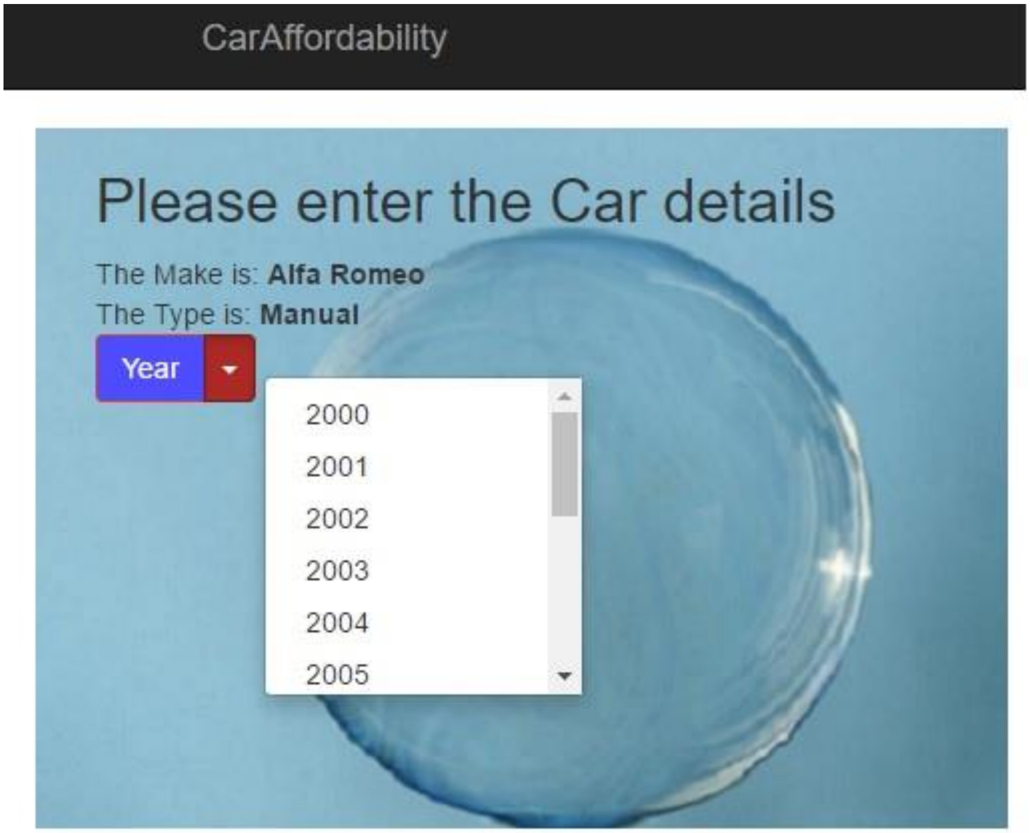
Findings

The results show that the population density show a very high correlation with the number cars sold with respect to each county in Ireland. The disposable income per person in each county does have only a moderate correlation with the number of cars sold.

These findings shed light into the trends in the car sales industry where Dublin seems to have the highest recorded number of car sales in the country. It is the population density that does have an effect on the number of car sales.

Calculate maintenance cost

The maintenance cost is calculated using the information that is provided by the user while using the application. JavaScript is used alongside PHP for this functionality.



Please enter the Car details

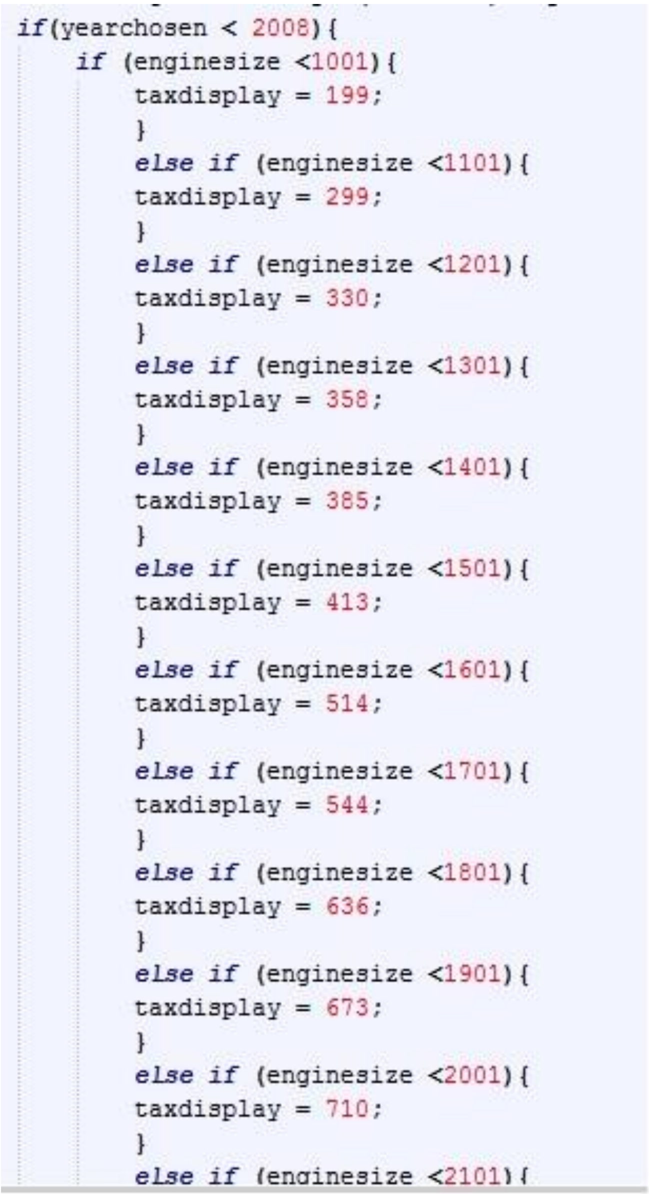
The Make is: **Alfa Romeo**

The Type is: **Manual**

The steps in calculating the maintenance cost is as follows. The user is allowed the select the desired car make. The PHP codes are activated with the aid of JQuery which in turn query the database with the selected make and further lists out the transmission types for the particular make.

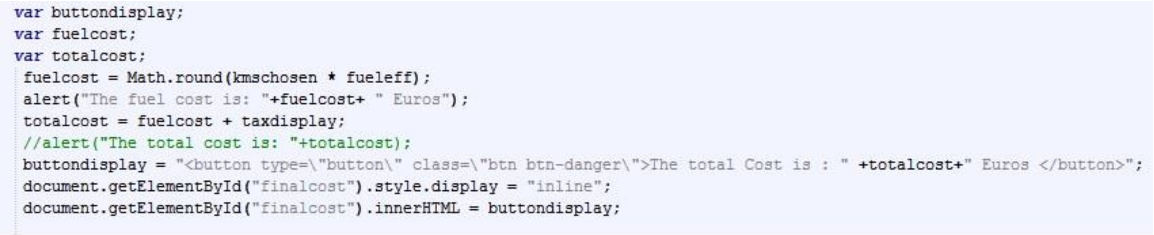
On selecting the transmission type, year, and model, the system will calculate the maintenace cost for the selected car. The javascript function that does the work is displayed below.

Based on the year, the tax is calculated based on the engine size of the car or the CO2 emission of the car.



```
if(yearchosen < 2008){
    if (enginesize <1001){
        taxdisplay = 199;
        }
        else if (enginesize <1101){
        taxdisplay = 299;
        }
        else if (enginesize <1201){
        taxdisplay = 330;
        }
        else if (enginesize <1301){
        taxdisplay = 358;
        }
        else if (enginesize <1401){
        taxdisplay = 385;
        }
        else if (enginesize <1501){
        taxdisplay = 413;
        }
        else if (enginesize <1601){
        taxdisplay = 514;
        }
        else if (enginesize <1701){
        taxdisplay = 544;
        }
        else if (enginesize <1801){
        taxdisplay = 636;
        }
        else if (enginesize <1901){
        taxdisplay = 673;
        }
        else if (enginesize <2001){
        taxdisplay = 710;
        }
        else if (enginesize <2101){
```

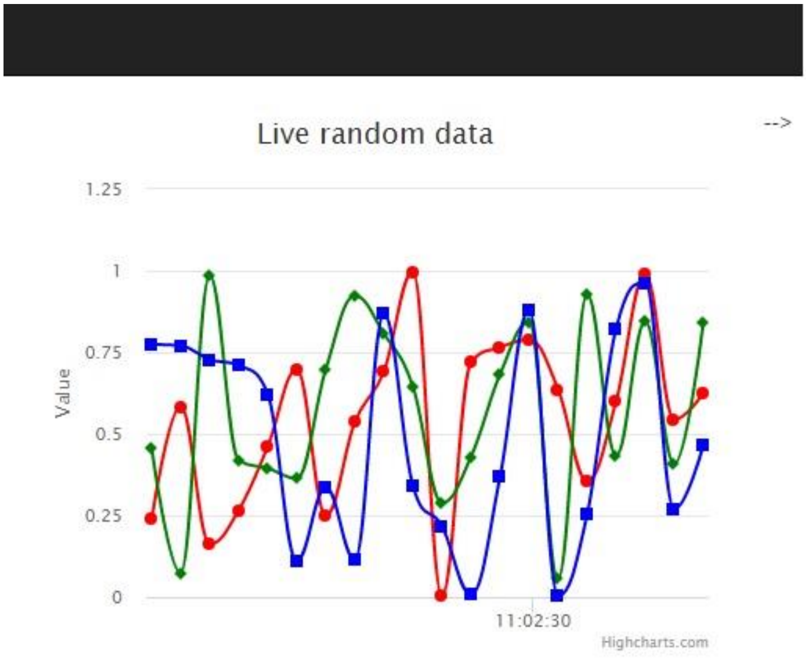
The fuel cost is calculated using the fuel efficiency of the selected model and the average mileage of the user. All the costs are added up to provide the approximate total mainteance cost of a particular model.



```
var buttondisplay;
var fuelcost;
var totalcost;
 fuelcost = Math.round(kmschosen * fueleff);
 alert("The fuel cost is: "+fuelcost+ " Euros");
 totalcost = fuelcost + taxdisplay;
 //alert("The total cost is: "+totalcost);
 buttondisplay = "<button type=\"button\" class=\"btn btn-danger\">The total Cost is : " +totalcost+" Euros </button>";
 document.getElementById("finalcost").style.display = "inline";
 document.getElementById("finalcost").innerHTML = buttondisplay;
```

This is one of the useful functionalities of this application that gives the users a clear picture of what is about to come.

Real Time Sentiment Analysis on Twitter

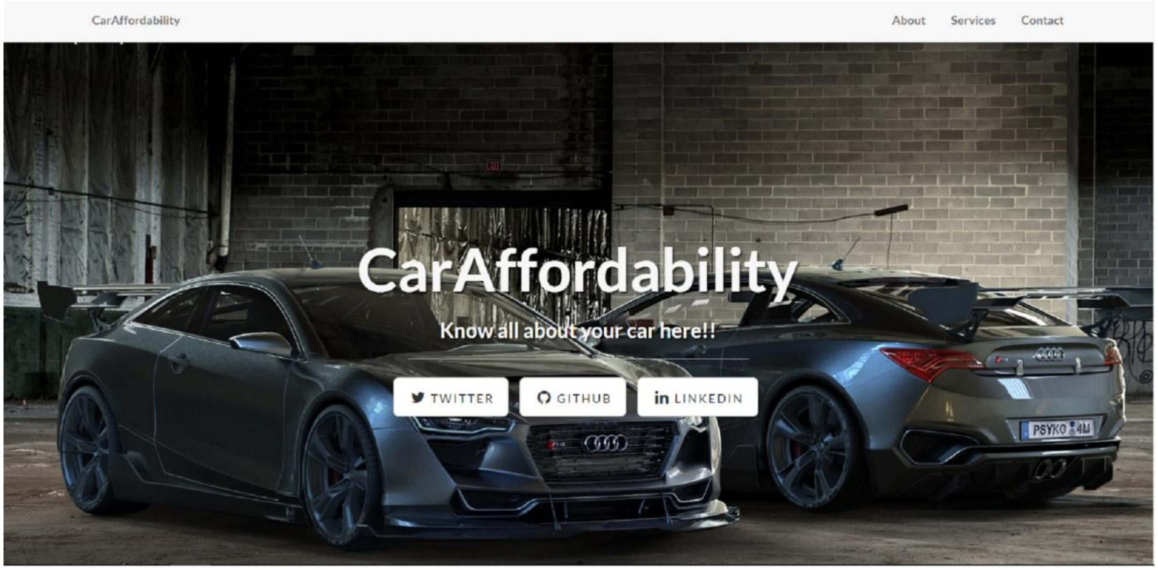


Real time sentiment analysis on twitter is one of the key functionalities of this application. The chart will display the sentiment analysis for the desired make of the car. It will compare graphically the positive and negative sentiments. The list of positive words and negative words have been created. The list can be updated at any point of time, which will reflect the same appropriately in the real time chart.

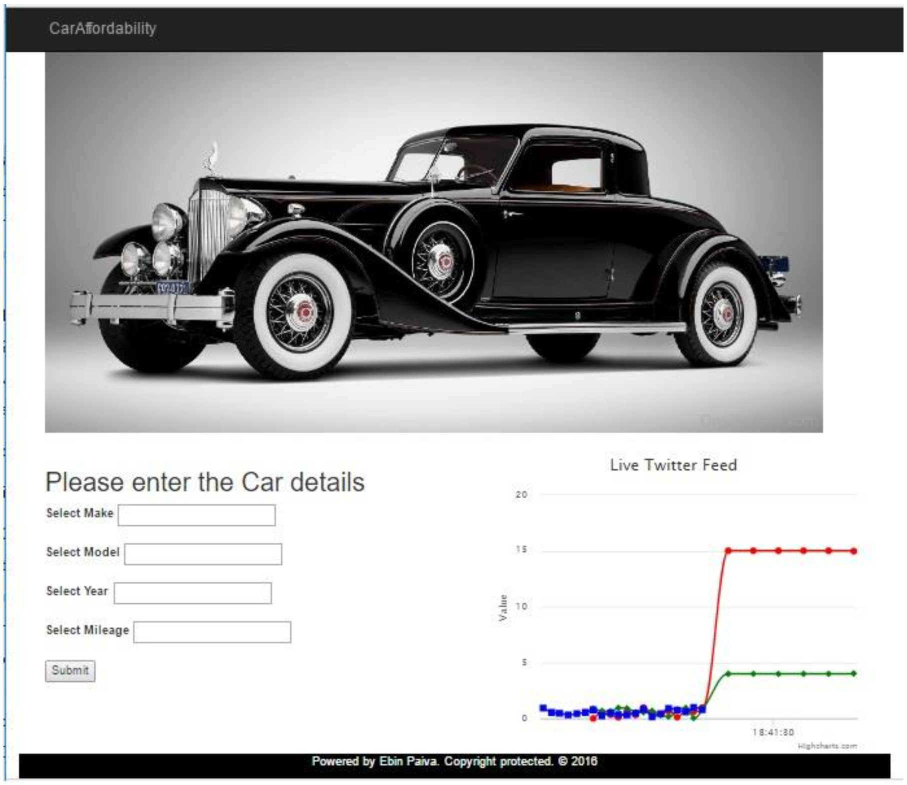
This chart will give a vivid and clear picture of the sentiments of the public in real time towards a particular make by querying the tweets on twitter and analysing it to provide information in graphical format.

Graphical User Interface (GUI) Layout

The design of the application was provided by Bootstrap. The navigation is smooth and proper. Design is responsive and works fine in all the browsers.



The landing page is displayed above.



Please enter the Car details

Select Make

Select Model

Select Year

Select Mileage

Powered by Ebin Paiva. Copyright protected. © 2016

Testing

Evaluation testing will be done to ensure proper integration of front end and back end as well as the proper functionality of this web application. Necessary amendments will be made depending on the test results.

Functionality Testing

Tests are done to check all the links are functional. Database is tested for proper connectivity, consistency, and integrity. Tests will be carried to ensure querying the database functions as expected. Input field validation checks will be done. HTML/CSS validation is essential to ensure crawlability for search engines. It can also avoid syntax errors in HTML codes.

Compatibility Testing

Tests will be done to check cross-browser compatibility. The application should be accessible in all desktop browsers as well as the mobile browsers. Tests to check the performance on various operating systems will also be conducted.

Usability Testing

Tests to check that the users can navigate through the webpages with ease. Interface is checked for user friendliness. The content should be properly visible and comprehensible for users of all ages. Link to the home page should be available in all the pages.

Performance Testing

It includes web load testing as well as web stress testing. The load testing is used to test whether system will take the load during the peak periods and at varying internet connection speeds. Stress testing is done to check how far the system can take the stress. The input fields are used to generate stress to system.

Testing Methods

The testing methods that were used in this project include black box testing, white box testing, and usability testing. The reports of the tests and their results are displayed respectively.

White box testing: In this testing method, we test the source codes. The functioning of the codes is tested for efficiency. Unlike the black box testing, the tester is interesting in know how the code functions.

Black box testing: In this testing method, we inspect the functionality of the application with no knowledge of codes that does the job behind the scenes. We only know about the intended result of the functionality but unaware of how it functions.

Usability testing: In this method tests are done to check the ease of using the application and navigating through the pages. How well an ordinary user can use the application without any aid from a second person.

1.1.1 White Box Testing

	W B Test 1		(*first page*)
AUT Name	Dropdown list for makes	**Version**	1.0
Iteration ID	1.0	**Test Date**	17/01/17

Test ID	EP 005
Aim of the test	Make sure that the dropdown list is displayed with the list returned from the querying the database.
Testing Environment	Environment consists of Hardware: Dell Inspiron laptop with latest windows 10. Software: MySQL community edition 5.7, XAMPP, Google Chrome PHP codes
Steps	Select the link for the maintenance cost check in the main page, which then will navigate you to the concerned page.
Expected Result	On page load, the dropdown menu should be visible with the list of make returned from querying the database for the makes.
Actual result	The dropdown list is populated with the list of makes.
Suggested action	None
Resolution	None

W B Test 2			(*first page*)
AUT Name	Dropdown list for transmission	**Version**	1.0
Iteration ID	1.0	**Test Date**	27/01/17

Test ID	EP 006
Aim of the test	Make sure that the dropdown list for the transmission is displayed with the list returned from the querying the database.
Testing Environment	Environment consists of Hardware: Dell Inspiron laptop with latest windows 10. Software: MySQL community edition 5.7, XAMPP, Google Chrome PHP codes
Steps	While in the page for maintenance cost check, select the make of your desired car from the dropdown list provided.
Expected Result	On selecting the make from the dropdown menu, a button should be visible with the list of transmission options returned from querying the database.
Actual result	The dropdown list is populated with the list of transmission options.
Suggested action	None.
Resolution	None.

1.1.2 Black Box Testing

	Black Box Test 1		(*first page*)
AUT Name	Read dataset	**Version**	1.0
Iteration ID	1.0	**Date of Test**	14/10/2016

Test ID	EP1
Purpose of Test	Check that the system reads data in
Test Environment	Hardware: Dell Inspiron laptop with latest windows 10. R Studio version 1.0.44 2009-2016 edition
Test Steps	Ensure that the dataset to be read in is in the working directory. In R Studio, run the R script, which allows to read in the dataset.
Expected Result	A data frame in the global environment is displayed with the data frame name and the number of the variables and observations.
Actual result	Dataset was read in successfully as expected.
Suggested action	None
Resolution	None

B B Test 2			
AUT Name	Insert data into database	**Version**	1.0
Iteration ID	1.0	**Test Date**	11/11/16

Test ID	EP2
Aim of the test	Make sure that the data can be read from a csv file and inserted into the database
Testing Environment	Environment consists of Hardware: Dell Inspiron laptop with latest windows 10. Software: MySQL community edition 5.7, XAMPP, Google Chrome A dataset in CSV format. PHP codes.
Steps	Run the PHP file in the chrome with the open connection to database and local server running.
Expected Result	Tables in the database should be updated with the data from the CSV file
Actual result	Tables updated as expected.
Suggested action	None
Resolution	None

B B Test 3			
AUT Name	Populate the chart with data from the database	**Version**	1.0
Iteration ID	1.0	**Test Date**	02/03/17

Test ID	EP3
Aim of the test	Make sure that the charts are displayed by reading the data from the database
Testing Environment	Environment consists of Hardware: Dell Inspiron laptop with latest windows 10. Software: MySQL community edition 5.7, XAMPP, Google Chrome. PHP codes.
Steps	Run the PHP file in the chrome with the open connection to database and local server running.
Expected Result	The charts should pop up displaying the data with proper labels
Actual result	The charts are displayed as expected.
Suggested action	None
Resolution	None

1.1.3 Usability testing

Usability acceptance Test 1			(*first page*)
AUT Name	Maintenance cost displayed	**Version**	1.0
Iteration ID	1.0	**Test Date**	03/03/17

Test ID	EP 101
Aim of the test	Make sure that the maintenance cost is displayed for the desired model of car selected by the user with respect to the user’s driving requirements
Testing Environment	Environment consists of Hardware: Dell Inspiron laptop with latest windows 10. Software: MySQL community edition 5.7, XAMPP, Google Chrome.
Steps	While using the application, select the desired model using the dropdown menu and input the details. Submit these information for the display of the maintenance cost.
Expected Result	The cost should be displayed appropriately and in Euros.
Actual result	The cost is displayed as expected.
Suggested action	None
Resolution	None

Usability acceptance Test 2			(*first page*)
AUT Name	Data displayed graphically for sales	**Version**	1.0
Iteration ID	1.0	**Test Date**	14/4/17

Test ID	EP 102
Aim of the test	Make sure that the data is displayed for the car sales graphically with appropriate labels.
Testing Environment	Environment consists of Hardware: Dell Inspiron laptop with latest windows 10. Software: MySQL community edition 5.7, XAMPP, Google Chrome.
Steps	While using the application, select the proper link from the dropdown list in the car sales statistics section to display the charts.
Expected Result	The chart should be displayed appropriately with labels.
Actual result	The chart is displayed as expected.
Suggested action	None
Resolution	None

Customer testing and feedback

The web application was tested on a probable user looking to buy a car, who do not have computer programming skills.

The user was able to navigate through the pages with ease. The functionalities seemed to work fine. The maintenance cost for the car was displayed appropriately. The sales charts of cars were also displayed appropriately where a normal person could use the functionalities of the application and gather information, which could become beneficial while choosing the right car.

Customer's feedback was promising. The customer was able to point out the area of insurance sector, which is an area of interest for any potential car buyer, which we do not have at the moment.

We will analyse the user's experience and their feedback. Effective measures will be made to make the interface to become more user friendly and accurate information to be displayed to the customer.

Evaluation and modification

The performance of the web application will be closely monitored. The traffic to the website will be monitored with respect gather information regarding the trends and peaks and drops in traffic.

The performance will be monitored to avoid any lag in fetching the information that has sought by the user.

With respect to user evaluation, navigation through the website and user interface will be modified to create better user experience.

Findings

The web application, Car-Affordability, is an application developed to help the potential car buyers who would like to know more about their desired car as well as to reconfirm their decision by using the application.

Car-Affordability allows the users to input their details and based on the information provided, the system calculates the cost of maintaining their car. All the expenses except the insurance cost are added to provide the total maintenance cost, which gives an approximate cost for a year.

Users can gather enough information to foresee the costs that can incur if they buy the car of their choice. This has been the main advantage of this web application. There are websites such ‘done deal’ and ‘car zone’, which do not provide such information to the users. The sales of the car manufacturers are analysed to provide useful performance indications. The market gap is profound and there seem to have a huge audience with respect to the figures of car sales each year in this country.

Car sales analysis done by the application do provide beneficial insights into the trends and patterns shown. The population density seems to be a major factor contributing to the increase in the number of car sales rather than the average disposable income with respect to each county.

Reviews of the cars by the experts can help the users in making the right choice as well as boost car sales in the country.

The web application’s limitation to provide insurance costs is worth noting. The varying insurance quotes by various insurance providers for each driver was unable to be calculated. In further development stages, steps for accessing the insurance providers’ website using their API to collect this information will be taken.

Further development or research

Further developments include the car insurance sector, which is one of the key area of interest for almost every car buyer. The insurance cost is on the rise every year. The various criteria for calculating the insurance makes the younger drivers to have a higher insurance. Hence, knowing the possible insurance quotes will surely be a beneficial information for any prospective car buyer.

A dedicated section with details of the emergency services in case of any events will be provided. The links to all the insurance providers will also be made available, which will help getting the best quote in the market.

Listing out the garages across the island will be a wonderful idea, which will enable drivers to plan their journey ahead without any hassle, since the users can find the nearest garage at any point on the map. Listing out the details of the mechanics and repair shops will prove beneficial in the event of any emergency call outs. Car affordability does have enough areas for futuristic expansion. Modifying this web application with respect to users' feedback and requirements of the current time will enable this application to stand alone providing beneficial information for the users.

Bibliography

"Web Testing: A Complete Guide About Testing Web Applications". *Software Testing Help*. N.p., 2016. Web. 1 Dec. 2016.

"Population Of Each Province, County And City, 2011 - CSO - Central Statistics Office". Cso.ie. N.p., 2017. Web. 12 April 2017.

"Statbank Database - County Incomes And Regional Accounts Listing Generated At 11:11 On 11APR17". Cso.ie. N.p., 2017. Web. 9 Apr. 2017.