INFORMATICS PRACTICES (NEW)

PROJECT ON

ANALYSIS OF CAR SALES

Software Development

BOARD ROLL CODE: 21666728

AISSCE 2022

ST FRANCIS SCHOOL, INDIRAPURAM

CERTIFICATE

This is to certify that HARSH SRIVASTAVA a student of Class XII A has successfully completed his Informatics Practices New(065) software project in the two defined terms as per the curriculum 2021-22 on Analysis Of Car Sales Software Development with my team members –

1. JAPINDER SINGH

2. KARTIKEY TIWARI

3. SHREYANSH BHADANI

under the guidance of Ma'am Vineeta Robert Simon.

Internal External

Signature School Stamp

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my IP Teacher (Vineeta Robert Simon Ma'am) who gave me and my team the instructions and guidance to complete this software development project.

I would also like to extend my gratitude to my parents for providing me with all the facilities that were required amidst this corona period.

HARSH SRIVASTAVA

Table of Contents

TOPICS	Page No.
Certificate	1
Acknowledgement	2
Index	3
Term – 1	
Project Synopsis	5-13
1.1 Project Objective	5-7
1.2 Project Method	8-10
1.3 Project Requirement Analysis	11-12
1.4 Project System Design	13

TOPICS Page No.

Term - 2

Project Implementation 14 -46

2.1 Project Coding 19 -31

2.2 Project Testing 32-34

2.3 Project Data Source Screenshots 35-38

2.4 Project Output Screenshots 39-45

2.4 Project Visualisation Screenshots 46

Bibliography 47

TERM – 1

Project Synopsis

Project Title Analysis Of Car Sales Software Development

Analysis Of Car Sales Software Development Introduction

This project ' The Analysis of Car Sales ' is a report based project for which the data is of the cars with respect to its sales for the year 2011-12 of the region USA is sourced from Github website.

The source data is ported to a data frame using pandas module and the user interactive interface is designed using streamlit module.

Math based calculations through the math module and data representation in graphs using matplotlib module.

The project is designed to get input from users as the choices being displayed and accordingly the analysis of the data will be shown to them further to which visualisation also has been framed to give a better visual understanding to the data pattern and behaviour.

Analysis Of Car Sales Software Development Objectives

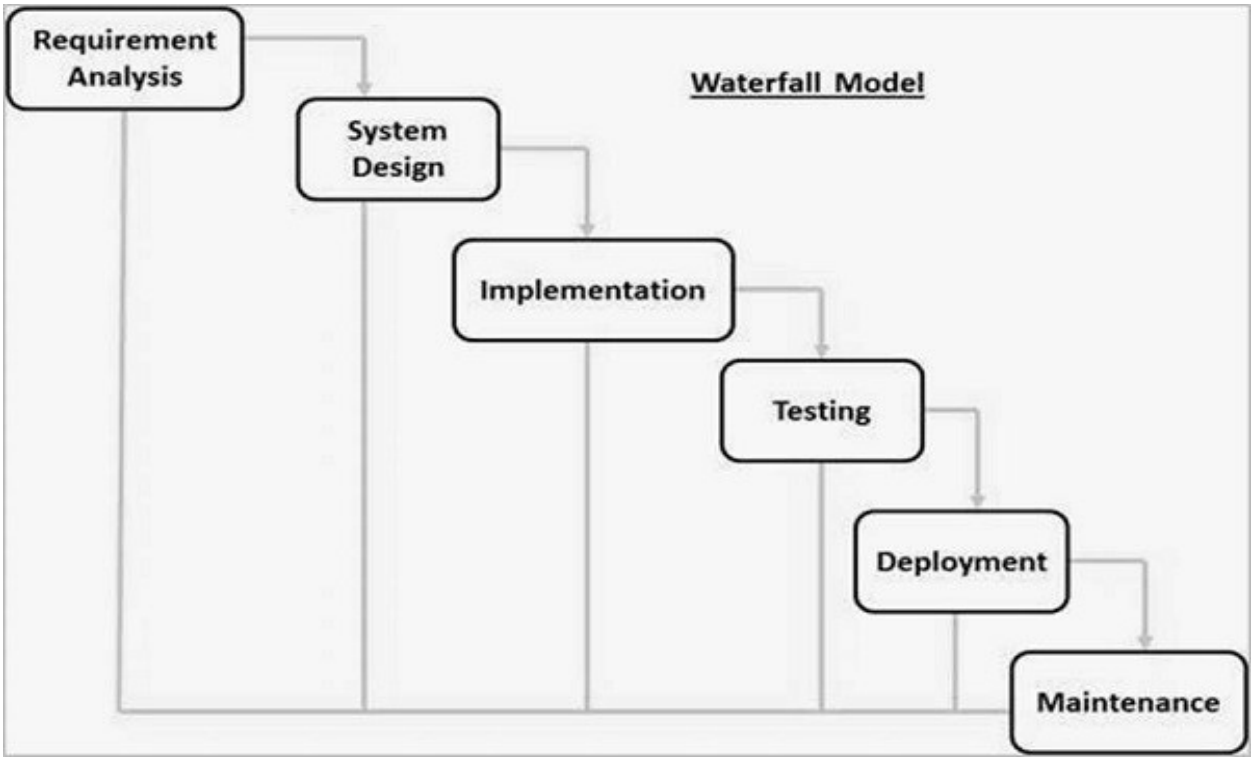
1. To gather the data from reliable sources
2. To save and store data in an electronic format
3. To secure data in terms of consistency
4. To understand the need of the users and what actually are they looking in this sector
5. To make a perfect analysis of the data using the Python coding
6. To design the code to best suit the need of the user and the optimum utilization of the data.
7. To execute the code at various stages of programming
8. To ensure the software if error free by testing and trial
9. To get a review among a sample user of this sector
10. To document the entire stages of this software development

Project Method

The Waterfall Model is also referred to as a linear-sequential life cycle model.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are –

- Requirement Gathering and analysis – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- System Design – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- Implementation – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- Integration and Testing – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- Deployment of system – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- Maintenance – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are –

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance.

Some of the major advantages of the Waterfall Model are as follows –

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows –

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

Project Requirement Analysis

User's Needs

1. To understand the data pattern

2. To predict the nature and behaviour of the current data design

3. To get ideas and suggestions to reroute the data in favourable behaviour

4. To get solutions for the current pattern of data for the desired result

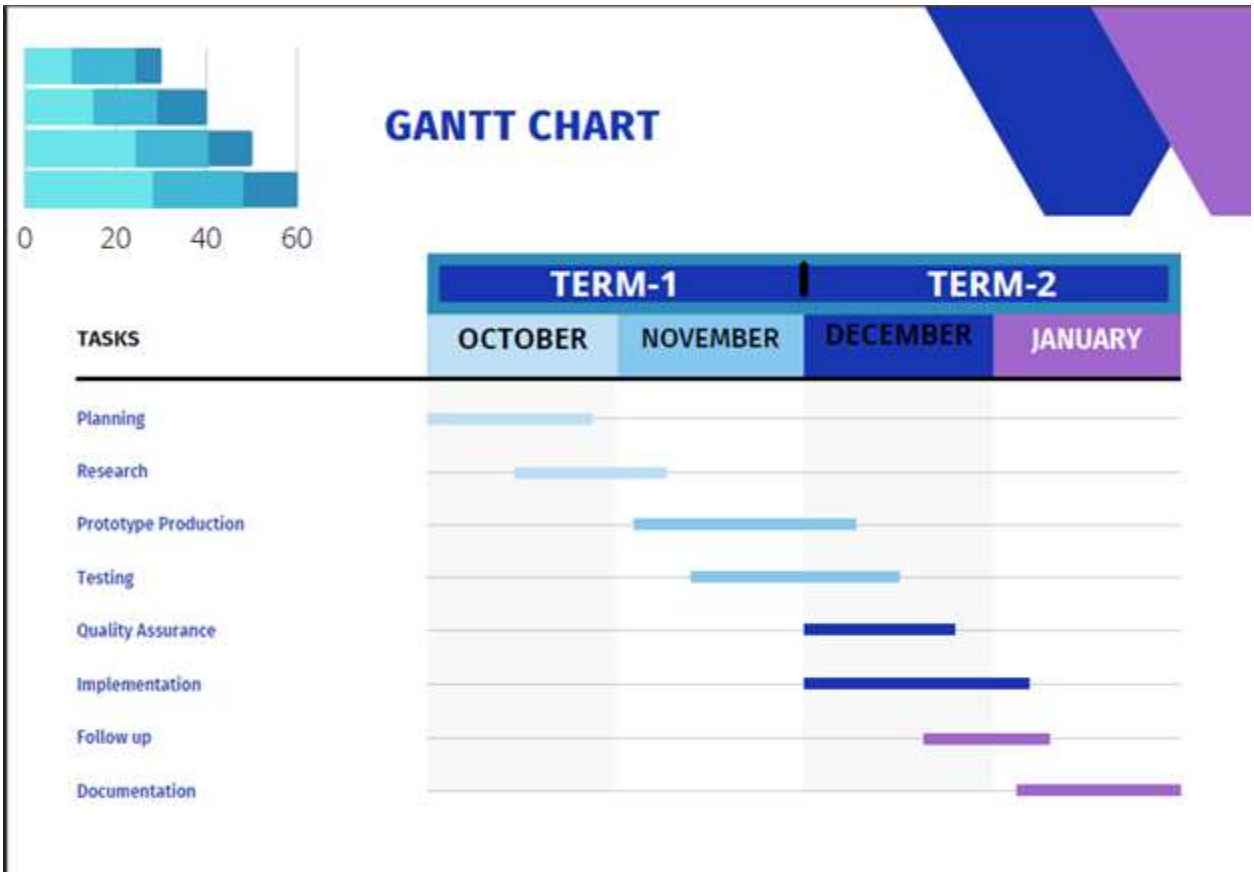
Software Requirement

Software requirement is done by a number of methods available. Which helps to understand the exact need of the kind of software required to run the software and to achieve the desired solution.

Requirement Analysis Techniques

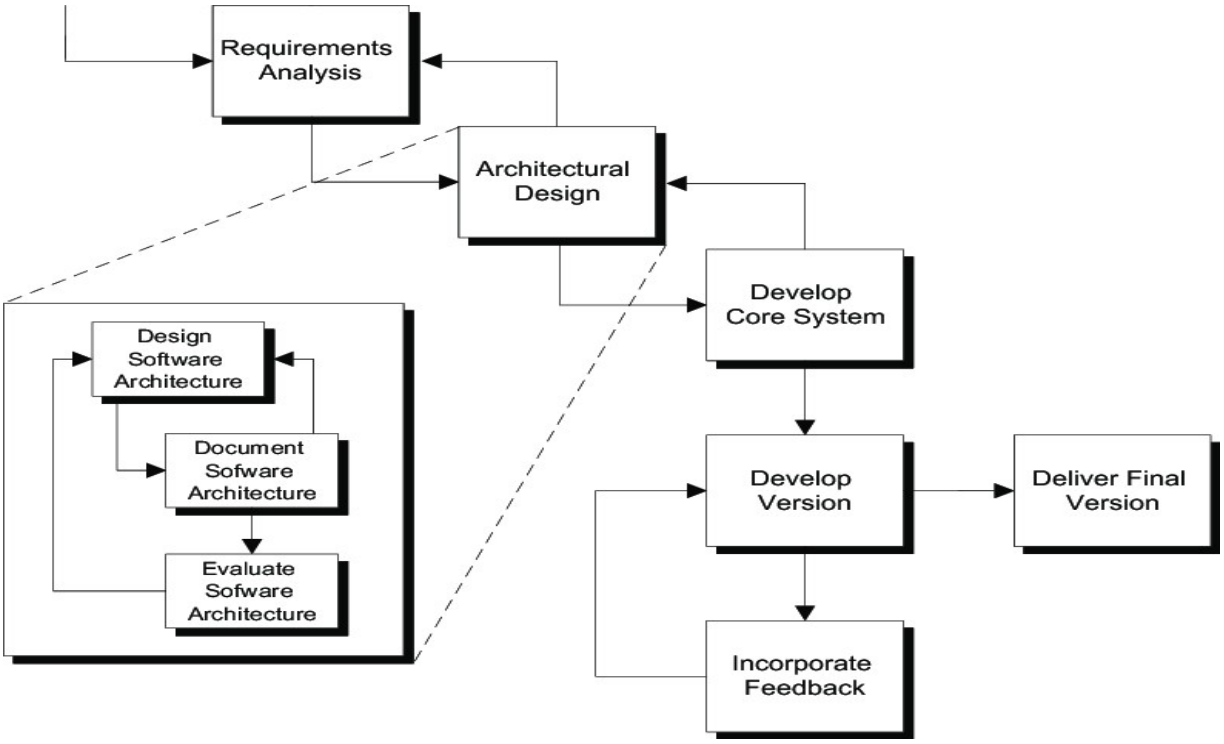
Gantt Charts

Gantt Charts provide a visual representation of tasks along with their scheduled timelines. They help business analysts visualize the start and end dates of all the tasks in a project.



Project System Design

System Software Architecture



TERM – 2

Project Implementation

Operating Environment	**Windows**
System S/w Specs	**Python**

User Interface requirements

UI is an important part of any software or hardware or hybrid system. A software is widely accepted if it is –

- easy to operate
- quick in response
- effectively handling operational errors
- providing simple yet consistent user interface

Data Collection Procedure

Data collection is defined as the procedure of collecting, measuring and analysing accurate insights for research using standard validated techniques. A researcher can evaluate their hypothesis on the basis of collected data. In most cases, data collection is the primary and most important step for research, irrespective of the field of research. The approach of data collection is different for different fields of study, depending on the required information.

The most critical objective of data collection is ensuring that information-rich and reliable data is collected for statistical analysis so that data-driven decisions can be made for research.

Web/Online

Pros: Cheap, can self-administer, very low probability of data errors
Cons: Not all your customers might have an email address/be on the internet, customers may be wary of divulging information online.

Data Analysis Procedure

Once we had set out to collect data for analysis, we were overwhelmed by the amount of information to make a clear, concise decision. With so much data to handle, we need to identify relevant data for your analysis to derive an accurate conclusion and make informed decisions.

The following simple steps helped us identifying and sorting the data for analysis.

.

- Organizing the data and making sure adding side notes, if any.
- Cross-checking data with reliable sources.
- Converting the data as per the scale of measurement as defined earlier.
- Excluding irrelevant data.

Data Analysis

- Once collected the data, perform sorting, plotting, and identifying correlations.
- Manipulating and organizing the data.
- To be prepared if need be then to traverse the steps again from the beginning.
- If need be then to redefine parameters, and reorganize data.
- Making use of the different tools available for data analysis.

Infer and Interpret Results

- Review whether all parameters have been considered and taken into coding application for making the decision
- Review if there is any hindering factor for implementing the decision.
- Choose data visualization techniques to communicate the message better.
- Applying added visualization techniques colour coding, elements, graphs detailed and more.

Since, it is only a hypothesis. Real-life scenarios may always interfere with our results. In Data Analysis, there are a few related terminologies that identity with different phases of the process.

1. Data Mining

This process involves methods in finding patterns in the data sample.

2. Data Modelling

This refers to how an organization organizes and manages its data.

Data Analysis Techniques

There are different techniques for Data Analysis depending upon the question at hand, the type of data, and the amount of data gathered. Each focuses on taking onto the new data, mining insights, and drilling down into the information to transform facts and figures into decision-making parameters. Accordingly, the different techniques of data analysis can be categorized as follows:

1. Techniques based on Mathematics and Statistics

- **Descriptive Analysis**: Descriptive Analysis considers the historical data, Key Performance Indicators and describes the performance based on a chosen benchmark. It takes into account past trends and how they might influence future performance.
- **Dispersion Analysis**: Dispersion in the area onto which a data set is spread. This technique allows data analysts to determine the variability of the factors under study.
- **Regression Analysis**: This technique works by modelling the relationship between a dependent variable and one or more independent variables. A regression model can be linear, multiple, logistic, ridge, non-linear, life data, and more.
- **Factor Analysis**: This technique helps to determine if there exists any relationship between a set of variables. This process reveals other factors or variables that describe the patterns in the relationship among the original variables. Factor Analysis leaps forward into useful clustering and classification procedures.
- **Discriminant Analysis**: It is a classification technique in data mining. It identifies the different points on different groups based on variable measurements. In simple terms, it identifies what makes two groups different from one another; this helps to identify new items.
- **Time Series Analysis**: In this kind of analysis, measurements are spanned across time, which gives us a collection of organized data known as time series.

2. Techniques based on Visualization and Graphs

- **Column Chart, Bar Chart**: Both these charts are used to present numerical differences between categories. The column chart takes to the height of the columns to reflect the differences. Axes interchange in the case of the bar chart.
- **Line Chart**: This chart represents the change of data over a continuous interval of time.
- **Area Chart**: This concept is based on the line chart. It also fills the area between the polyline and the axis with colour, representing better trend information.
- **Pie Chart**: It is used to represent the proportion of different classifications. It is only suitable for only one series of data. However, it can be made multi-layered to represent the proportion of data in different categories.

Project Coding

```
# importing the modules
from os import path
import pandas as pd
import streamlit as st
import plotly.express as px
import pandas as pd
import numpy as np
import matplotlib.pyplot as py
import datetime

# giving a suitable heading to the streamlit module
st.title('CARS SURVEY 2011-2012')
st.write('THE CARSALES ARE AS FOLLOWS ')
st.balloons()
# importing the excel file and displaying the table on the screen
df = pd.read_excel('1CARSALE.xlsx',
                   usecols='A,c,D:F,I,J,K,L,M,N,P',
                   header=0)
st.write('the survey is as follows')
st.dataframe(df)

# displaying a pie chart on the basis of total cars sold by different manufacturers in year 2011 and
2012

pie_chart = px.pie(df,
                   title='TOTAL NUMBER OF MANUFACTURERS',
                   values='Sales_in_thousands',
                   names='Manufacturer')

# plotting the pie chart
st.plotly_chart(pie_chart,figsize=(20,20))

# to find the total manufactures in the year 2011 and 2012
Manufacturer = df['Manufacturer'].unique().tolist()
sales = df['Fuel_efficiency'].unique().tolist()
#horsepower = df['Horspower'].unique().tolist()

# adding a slider in our streamlit project on fuel_efficiency and different manufactures
sales_selection = st.slider('Fuel_efficiency:',
                        min_value= min(sales),
                        max_value= max(sales),
                        value=(min(sales),max(sales)))

# adding a multi select bar with the slider
# so that the user can find therir suitable car by selecting the fuel efficiency and the manufacturer
department_selection = st.multiselect('Manufacturer:',
                                  options=df["Manufacturer"].unique(),
                                  default=df["Manufacturer"].unique())
```

```
# executing the slider and multiselect
mask = (df['Fuel_efficiency'].between(*sales_selection)) &
(df['Manufacturer'].isin(department_selection))
number_of_result = df[mask].shape[0]
st.markdown(f'*Available Results: {number_of_result}*')

df_grouped = df[mask].groupby(by=['Manufacturer']).count()[['Fuel_efficiency']]
df_grouped = df_grouped.reset_index()

df[mask]

import pandas as pd
import numpy as np
import matplotlib.pyplot as py
import datetime

df=pd.read_excel('1CARSALE.xlsx')
df['Latest_Launch']=pd.to_datetime(df['Latest_Launch'])

#STARTING OF THE PROGRAMME

#Adding a header and sub header to the if code
st.header("If you wish to pick a car for yourself, enter Y")
st.subheader("If you do not wish to pick a car for yourself, enter N")

#starting of the code
a=(st.text_input("Enter your choice from Y/N  :  "))

if a=='Y':

    b=(st.selectbox("CAR SURVEY FOR THE YEAR YOU WANTED TO SEE: ",
                    ['A) Do you wish to see the performane of cars of year 2011...',
                     'B) Do you wish to see the performane of cars of year 2012...',
                     'Do you wish to see the performane of cars of years 2011 and 2012 combined...']))



    if b=='A) Do you wish to see the performane of cars of year 2011...':

        C=(st.selectbox("Enter your choice to see the performance of cars in 2011 in various aspects ",
                    [ 'Enter 1 if you wish to see the 10 leading fuel efficient cars :', 'Enter 2 if you wish to
see the Leading power Performance Stats : ',
                     'Enter 3 if you wish to see the Best HorsePower Engine :','Enter 4 if you wish to see
the price list of cars (from highest to lowest) :',
                      'Enter 5 if you wish to see the information regarding the most sold car :']))


        if C=='Enter 1 if you wish to see the 10 leading fuel efficient cars :':
            filter1=df["Latest_Launch"]< datetime.datetime(2011,12,31)
            kk=df.where(filter1,inplace=True)
            aa=df.sort_values(by='Fuel_efficiency',ascending=False)
            ab=aa.head(10)
            ac=ab[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Fuel_effici
ency']]
            pd.set_option('expand_frame_repr', False)
```

```
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("10 leading fuel efficient cars are : \n ",ac)

    if C=='Enter 2 if you wish to see the Leading power Performance Stats:':
        filter1=df["Latest_Launch"]< datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        ad=df.sort_values(by='Power_perf_factor',ascending=False)
        ae=ad.head(1)
        af=ae[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Power_per
f_factor']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("Cars for leading Power performance stats are : \n ",af)

    if C=='Enter 3 if you wish to see the Best HorsePower Engine :':
        filter1=df["Latest_Launch"]< datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        ag=df.sort_values(by='Horsepower',ascending=False)
        ah=ag.head(1)
        ai=ah[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepow
er']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("Cars with best horsepower engine : \n ",ai)

    if C=='Enter 4 if you wish to see the price list of cars (from highest to lowest) :':
        filter1=df["Latest_Launch"]< datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        aj=df.sort_values(by='Price_in_thousands',ascending=False)
        ak=aj[['Manufacturer','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",ak)

    if C=='Enter 5 if you wish to see the information regarding the most sold car :':
        filter1=df["Latest_Launch"]<datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        al=df.sort_values(by='Sales_in_thousands',ascending=False)
        aw=al.head(1)
        am=aw[['Manufacturer','Model','Latest_Launch','Vehicle_type','Sales_in_thousands','Price_in
_thousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
```

```
            pd.set_option('display.width', None)
            pd.set_option('display.max_colwidth', None)
            st.write("price list of the cars : \n ",am)

    elif b=='B) Do you wish to see the performane of cars of year 2012...':

        d=(st.selectbox("Enter your choice to see the performance of cars in 2012 in various aspects ",
                    ['Enter 1 if you wish to see the 10 leading fuel efficient cars :','Enter 2 if you wish to see
the Leading power Performance Stats :',
                     'Enter 3 if you wish to see the Best HorsePower Engine :','Enter 4 if you wish to see
the price list of cars (from highest to lowest) :',
                     'Enter 5 if you wish to see the information regarding the most sold car :']))

        if(d=='Enter 1 if you wish to see the 10 leading fuel efficient cars :'):
            filter1=df["Latest_Launch"]>datetime.datetime(2011,12,31)
            kk=df.where(filter1,inplace=True)
            aa=df.sort_values(by='Fuel_efficiency',ascending=False)
            ab=aa.head(10)
            ac=ab[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Fuel_efficie
ncy']]
            pd.set_option('display.max_rows', None)
            pd.set_option('display.max_columns', None)
            pd.set_option('display.width', None)
            pd.set_option('display.max_colwidth', None)
            st.write("10 leading fuel efficient cars are : \n ",ac)

        if(d=='Enter 2 if you wish to see the Leading power Performance Stats :'):

            filter1=df["Latest_Launch"]> datetime.datetime(2011,12,31)
            kk=df.where(filter1,inplace=True)
            ad=df.sort_values(by='Power_perf_factor',ascending=False)
            ae=ad.head(1)
            af=ae[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Power_perf_f
actor']]
            pd.set_option('display.max_rows', None)
            pd.set_option('display.max_columns', None)
            pd.set_option('display.width', None)
            pd.set_option('display.max_colwidth', None)
            st.write("Cars for leading Power performance stats are : \n ",af)

        if(d=='Enter 3 if you wish to see the Best HorsePower Engine :'):
            filter1=df["Latest_Launch"]> datetime.datetime(2011,12,31)
            kk=df.where(filter1,inplace=True)
            ag=df.sort_values(by='Horsepower',ascending=False)
            ah=ag.head(1)
            ai=ah[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
            pd.set_option('display.max_rows', None)
            pd.set_option('display.max_columns', None)
            pd.set_option('display.width', None)
            pd.set_option('display.max_colwidth', None)
            st.write("Cars with best horsepower engine : \n ",ai)

        if(d=='Enter 4 if you wish to see the price list of cars (from highest to lowest) :'):
            filter1=df["Latest_Launch"]>datetime.datetime(2011,12,31)
            kk=df.where(filter1,inplace=True)
```

```
        aj=df.sort_values(by='Price_in_thousands',ascending=False)
        ak=aj[['Manufacturer','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",ak)
        st.bar_chart(data=ak,use_container_width=True)

       if(d=='Enter 5 if you wish to see the information regarding the most sold car :'):
        filter1=df["Latest_Launch"]>datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        al=df.sort_values(by='Sales_in_thousands',ascending=False)
        ay=al.head(1)
        am=ay[['Manufacturer','Model','Latest_Launch','Vehicle_type','Sales_in_thousands','Price_in_th
ousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",am)

    elif b=="Do you wish to see the performane of cars of years 2011 and 2012 combined...":


       t=(st.selectbox("Enter your choice to see the performance of cars in 2011 AND 2012 in various
aspects ",
                    ['Enter 1 if you wish to see the 10 leading fuel efficient cars :','Enter 2 if you wish to see
the Leading power Performance Stats :',
                     'Enter 3 if you wish to see the Best HorsePower Engine :','Enter 4 if you wish to see
the price list of cars (from highest to lowest) :',
                     'Enter 5 if you wish to see the information regarding the most sold car :']))


       if(t=='Enter 1 if you wish to see the 10 leading fuel efficient cars :'):
        aa=df.sort_values(by='Fuel_efficiency',ascending=False)
        ab=aa.head(10)
        ac=ab[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Fuel_efficien
cy']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("10 leading fuel efficient cars are : \n ",ac)


       if(t=='Enter 2 if you wish to see the Leading power Performance Stats :'):
        ad=df.sort_values(by='Power_perf_factor',ascending=False)
        ae=ad.head(1)
        af=ae[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Power_perf_f
actor']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
```

```
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("Cars for leading Power performance stats are : \n ",af)

       if(t=='Enter 3 if you wish to see the Best HorsePower Engine :'):
        ag=df.sort_values(by='Horsepower',ascending=False)
        ah=ag.head(1)
        ai=ah[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("Cars with best horsepower engine : \n ",ai)

       if(t==' Enter 4 if you wish to see the price list of cars (from highest to lowest) :'):
        aj=df.sort_values(by='Price_in_thousands',ascending=True)
        AX=aj.head(10)
        CC=AX[['Manufacturer','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",CC)




       if(t=='Enter 5 if you wish to see the information regarding the most sold car :'):
        al=df.sort_values(by='Sales_in_thousands',ascending=False)
        az=al.head(1)
        am=az[['Manufacturer','Model','Vehicle_type','Sales_in_thousands','Price_in_thousands','Horse
power']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",am)

if a=='N':

   st.write('IF YOU WANT TO SEE DATA OF A PARTICULAR MANUFACTURE')


   s=(st.selectbox("SELECT THE MANUFACTURER:",
                   (df['Manufacturer'].unique().tolist())))

   df_selection=df.query('Manufacturer == @s')
   st.write("the cars are :\n",df_selection )
```

```
st.sidebar.header('THE DATA FOR THE SALES OF VECHILES FROM 2011 TO 2012 ARE GIVE IN THE WEB PAGE. YOU CAN ALSO CHOOSE THE MOST SUITABLE VECHILE AS PER YOUR CHOICE BY SELECTINGTHE SUITABLE OPTION FROM THE DATAFRAME HOPE YOU LIKED IT THANK YOU !')

#hello = df.
Horsepower = df["Latest_Launch"]>datetime.datetime(2011,12,31)
st.bar_chart(data=Horsepower,use_container_width=True)


# importing the modules
from os import path
import pandas as pd
import streamlit as st
import plotly.express as px
import pandas as pd
import numpy as np
import matplotlib.pyplot as py
import datetime

# giving a suitable heading to the streamlit module
st.title('CARS SURVEY 2011-2012')
st.write('THE CARSALES ARE AS FOLLOWS ')
st.balloons()
# importing the excel file and displaying the table on the screen
df = pd.read_excel('1CARSALE.xlsx',
                   usecols='A,c,D:F,I,J,K,L,M,N,P',
                   header=0)
st.write('the survey is as follows')
st.dataframe(df)

# displaying a pie chart on the basis of total cars sold by different manufacturers in year 2011 and 2012

pie_chart = px.pie(df,
                   title='TOTAL NUMBER OF MANUFACTURERS',
                   values='Sales_in_thousands',
                   names='Manufacturer')

# plotting the pie chart
st.plotly_chart(pie_chart,figsize=(20,20))

# to find the total manufactures in the year 2011 and 2012
Manufacturer = df['Manufacturer'].unique().tolist()
sales = df['Fuel_efficiency'].unique().tolist()
#horsepower = df['Horspower'].unique().tolist()

# adding a slider in our streamlit project on fuel_efficiency and different manufactures
sales_selection = st.slider('Fuel_efficiency:',
                            min_value= min(sales),
                            max_value= max(sales),
                            value=(min(sales),max(sales)))

# adding a multi select bar with the slider
# so that the user can find therir suitable car by selecting the fuel efficiency and the manufacturer
```

```
department_selection = st.multiselect('Manufacturer:',
                                      options=df["Manufacturer"].unique(),
                                      default=df["Manufacturer"].unique())

# executing the slider and multiselect
mask = (df['Fuel_efficiency'].between(*sales_selection)) &
(df['Manufacturer'].isin(department_selection))
number_of_result = df[mask].shape[0]
st.markdown(f'*Available Results: {number_of_result}*')

df_grouped = df[mask].groupby(by=['Manufacturer']).count()[['Fuel_efficiency']]
df_grouped = df_grouped.reset_index()

df[mask]

import pandas as pd
import numpy as np
import matplotlib.pyplot as py
import datetime

df=pd.read_excel('1CARSALE.xlsx')
df['Latest_Launch']=pd.to_datetime(df['Latest_Launch'])

#STARTING OF THE PROGRAMME

#Adding a header and sub header to the if code
st.header("If you wish to pick a car for yourself, enter Y")
st.subheader("If you do not wish to pick a car for yourself, enter N")

#starting of the code
a=(st.text_input("Enter your choice from Y/N  :  "))

if a=='Y':

    b=(st.selectbox("CAR SURVEY FOR THE YEAR YOU WANTED TO SEE: ",
                    ['A) Do you wish to see the performane of cars of year 2011...',
                     'B) Do you wish to see the performane of cars of year 2012...',
                     'Do you wish to see the performane of cars of years 2011 and 2012 combined...']))



    if b=='A) Do you wish to see the performane of cars of year 2011...':

        C=(st.selectbox("Enter your choice to see the performance of cars in 2011 in various aspects ",
                    [ 'Enter 1 if you wish to see the 10 leading fuel efficient cars :', 'Enter 2 if you wish to
see the Leading power Performance Stats : ',
                     'Enter 3 if you wish to see the Best HorsePower Engine :','Enter 4 if you wish to see
the price list of cars (from highest to lowest) :',
                      'Enter 5 if you wish to see the information regarding the most sold car :']))


        if C=='Enter 1 if you wish to see the 10 leading fuel efficient cars :':
            filter1=df["Latest_Launch"]< datetime.datetime(2011,12,31)
            kk=df.where(filter1,inplace=True)
            aa=df.sort_values(by='Fuel_efficiency',ascending=False)
```

```
        ab=aa.head(10)
        ac=ab[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Fuel_effici
ency']]
        pd.set_option('expand_frame_repr', False)

        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("10 leading fuel efficient cars are : \n ",ac)

    if C=='Enter 2 if you wish to see the Leading power Performance Stats:':
        filter1=df["Latest_Launch"]< datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        ad=df.sort_values(by='Power_perf_factor',ascending=False)
        ae=ad.head(1)
        af=ae[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Power_per
f_factor']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("Cars for leading Power performance stats are : \n ",af)

    if C=='Enter 3 if you wish to see the Best HorsePower Engine :':
        filter1=df["Latest_Launch"]< datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        ag=df.sort_values(by='Horsepower',ascending=False)
        ah=ag.head(1)
        ai=ah[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepow
er']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("Cars with best horsepower engine : \n ",ai)

    if C=='Enter 4 if you wish to see the price list of cars (from highest to lowest) :':
        filter1=df["Latest_Launch"]< datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        aj=df.sort_values(by='Price_in_thousands',ascending=False)
        ak=aj[['Manufacturer','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",ak)

    if C=='Enter 5 if you wish to see the information regarding the most sold car :':
        filter1=df["Latest_Launch"]<datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        al=df.sort_values(by='Sales_in_thousands',ascending=False)
        aw=al.head(1)
```

```
        am=aw[['Manufacturer','Model','Latest_Launch','Vehicle_type','Sales_in_thousands','Price_in
_thousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",am)

    elif b=='B) Do you wish to see the performane of cars of year 2012...':

      d=(st.selectbox("Enter your choice to see the performance of cars in 2012 in various aspects ",
                ['Enter 1 if you wish to see the 10 leading fuel efficient cars :','Enter 2 if you wish to see
the Leading power Performance Stats :',
                 'Enter 3 if you wish to see the Best HorsePower Engine :','Enter 4 if you wish to see
the price list of cars (from highest to lowest) :',
                 'Enter 5 if you wish to see the information regarding the most sold car :']))

      if(d=='Enter 1 if you wish to see the 10 leading fuel efficient cars :'):
        filter1=df["Latest_Launch"]>datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        aa=df.sort_values(by='Fuel_efficiency',ascending=False)
        ab=aa.head(10)
        ac=ab[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Fuel_efficie
ncy']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("10 leading fuel efficient cars are : \n ",ac)

      if(d=='Enter 2 if you wish to see the Leading power Performance Stats :'):

       filter1=df["Latest_Launch"]> datetime.datetime(2011,12,31)
       kk=df.where(filter1,inplace=True)
       ad=df.sort_values(by='Power_perf_factor',ascending=False)
       ae=ad.head(1)
       af=ae[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Power_perf_f
actor']]
       pd.set_option('display.max_rows', None)
       pd.set_option('display.max_columns', None)
       pd.set_option('display.width', None)
       pd.set_option('display.max_colwidth', None)
       st.write("Cars for leading Power performance stats are : \n ",af)

      if(d=='Enter 3 if you wish to see the Best HorsePower Engine :'):
       filter1=df["Latest_Launch"]> datetime.datetime(2011,12,31)
       kk=df.where(filter1,inplace=True)
       ag=df.sort_values(by='Horsepower',ascending=False)
       ah=ag.head(1)
       ai=ah[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
       pd.set_option('display.max_rows', None)
       pd.set_option('display.max_columns', None)
       pd.set_option('display.width', None)
       pd.set_option('display.max_colwidth', None)
```

```
        st.write("Cars with best horsepower engine : \n ",ai)

       if(d=='Enter 4 if you wish to see the price list of cars (from highest to lowest) :'):
        filter1=df["Latest_Launch"]>datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        aj=df.sort_values(by='Price_in_thousands',ascending=False)
        ak=aj[['Manufacturer','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",ak)
        st.bar_chart(data=ak,use_container_width=True)

       if(d=='Enter 5 if you wish to see the information regarding the most sold car :'):
        filter1=df["Latest_Launch"]>datetime.datetime(2011,12,31)
        kk=df.where(filter1,inplace=True)
        al=df.sort_values(by='Sales_in_thousands',ascending=False)
        ay=al.head(1)
        am=ay[['Manufacturer','Model','Latest_Launch','Vehicle_type','Sales_in_thousands','Price_in_th
ousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",am)

    elif b=="Do you wish to see the performane of cars of years 2011 and 2012 combined...":


       t=(st.selectbox("Enter your choice to see the performance of cars in 2011 AND 2012 in various
aspects ",
                    ['Enter 1 if you wish to see the 10 leading fuel efficient cars :','Enter 2 if you wish to see
the Leading power Performance Stats :',
                     'Enter 3 if you wish to see the Best HorsePower Engine :','Enter 4 if you wish to see
the price list of cars (from highest to lowest) :',
                     'Enter 5 if you wish to see the information regarding the most sold car :']))


       if(t=='Enter 1 if you wish to see the 10 leading fuel efficient cars :'):
        aa=df.sort_values(by='Fuel_efficiency',ascending=False)
        ab=aa.head(10)
        ac=ab[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Fuel_efficien
cy']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("10 leading fuel efficient cars are : \n ",ac)


       if(t=='Enter 2 if you wish to see the Leading power Performance Stats :'):
        ad=df.sort_values(by='Power_perf_factor',ascending=False)
```

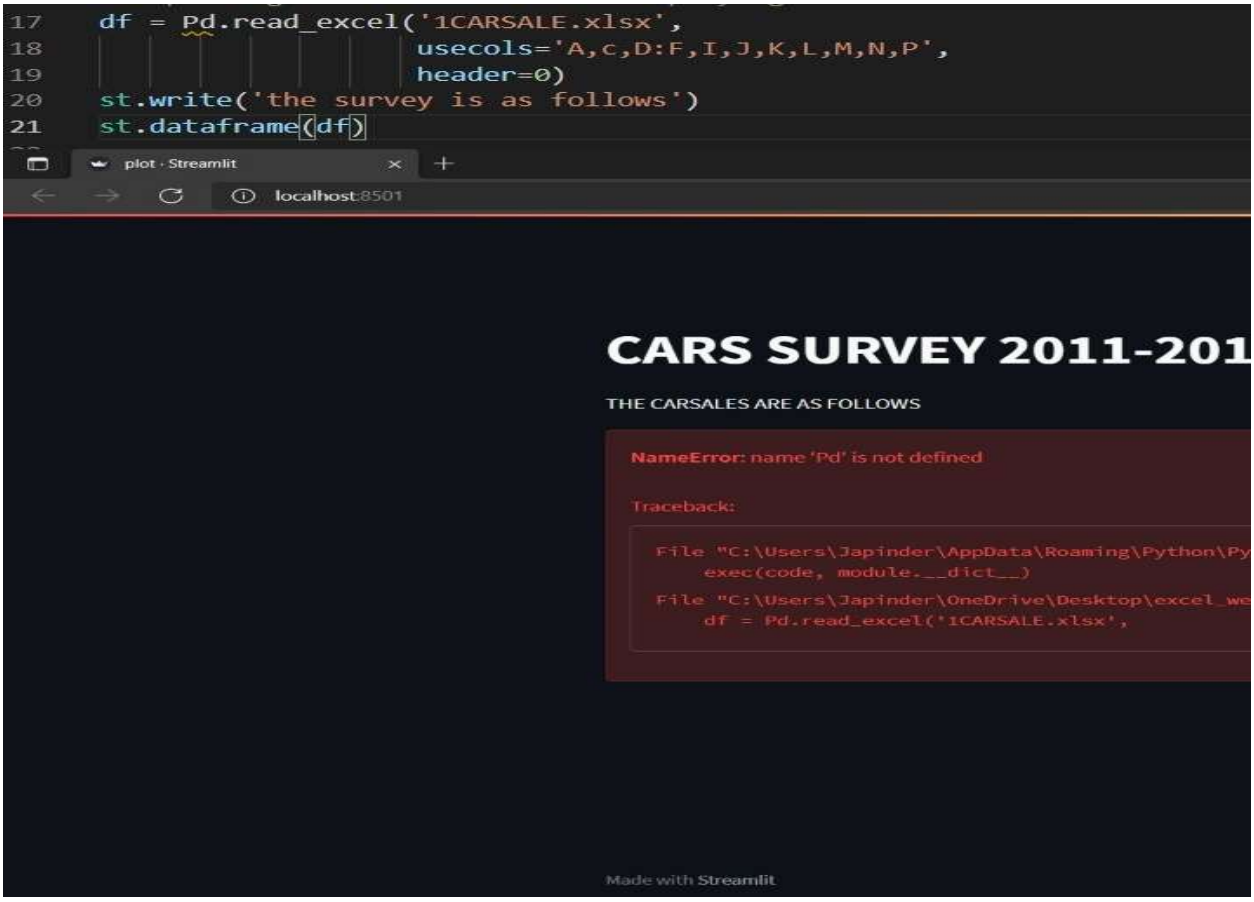
```
        ae=ad.head(1)
        af=ae[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Power_perf_f
actor']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("Cars for leading Power performance stats are : \n ",af)

       if(t=='Enter 3 if you wish to see the Best HorsePower Engine :'):
        ag=df.sort_values(by='Horsepower',ascending=False)
        ah=ag.head(1)
        ai=ah[['Manufacturer','Model','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("Cars with best horsepower engine : \n ",ai)

       if(t==' Enter 4 if you wish to see the price list of cars (from highest to lowest) :'):
        aj=df.sort_values(by='Price_in_thousands',ascending=True)
        AX=aj.head(10)
        CC=AX[['Manufacturer','Latest_Launch','Vehicle_type','Price_in_thousands','Horsepower']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",CC)




       if(t=='Enter 5 if you wish to see the information regarding the most sold car :'):
        al=df.sort_values(by='Sales_in_thousands',ascending=False)
        az=al.head(1)
        am=az[['Manufacturer','Model','Vehicle_type','Sales_in_thousands','Price_in_thousands','Horse
power']]
        pd.set_option('expand_frame_repr', False)
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        pd.set_option('display.width', None)
        pd.set_option('display.max_colwidth', None)
        st.write("price list of the cars : \n ",am)

if a=='N':

   st.write('IF YOU WANT TO SEE DATA OF A PARTICULAR MANUFACTURE')


   s=(st.selectbox("SELECT THE MANUFACTURER:",
                   (df['Manufacturer'].unique().tolist())))

   df_selection=df.query('Manufacturer == @s')
```

```
   st.write("the cars are :\n",df_selection )


st.sidebar.header('THE DATA FOR THE SALES OF VECHILES FROM 2011 TO 2012 ARE GIVE IN THE WEB PAGE. YOU CAN ALSO CHOOSE THE MOST SUITABLE VECHILE AS PER YOUR CHOICE BY SELECTINGTHE SUITABLE OPTION FROM THE DATAFRAME HOPE YOU LIKED IT THANK YOU !')

#hello = df.
Horsepower = df["Latest_Launch"]>datetime.datetime(2011,12,31)
st.bar_chart(data=Horsepower,use_container_width=True)
```

Project Testing

Screenshots of error and corrected code

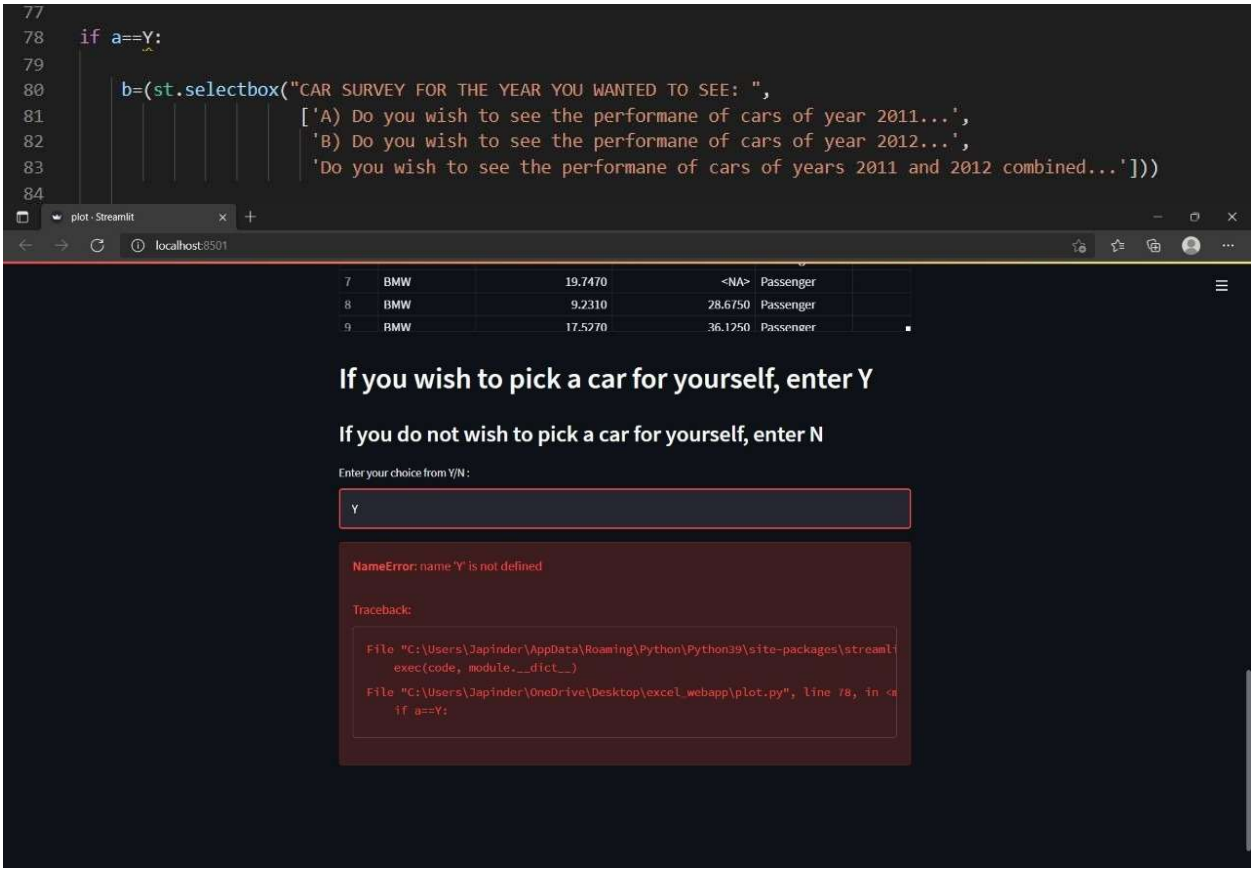
incorrect



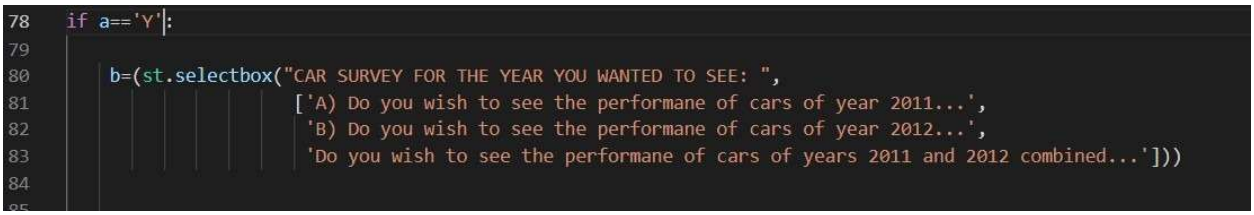
correct

```
17  df = pd.read_excel('1CARSALE.xlsx',
18                     usecols='A,c,D:F,I,J,K,L,M,N,P',
19                     header=0)
20  st.write('the survey is as follows')
21  st.dataframe(df)
```

incorrect

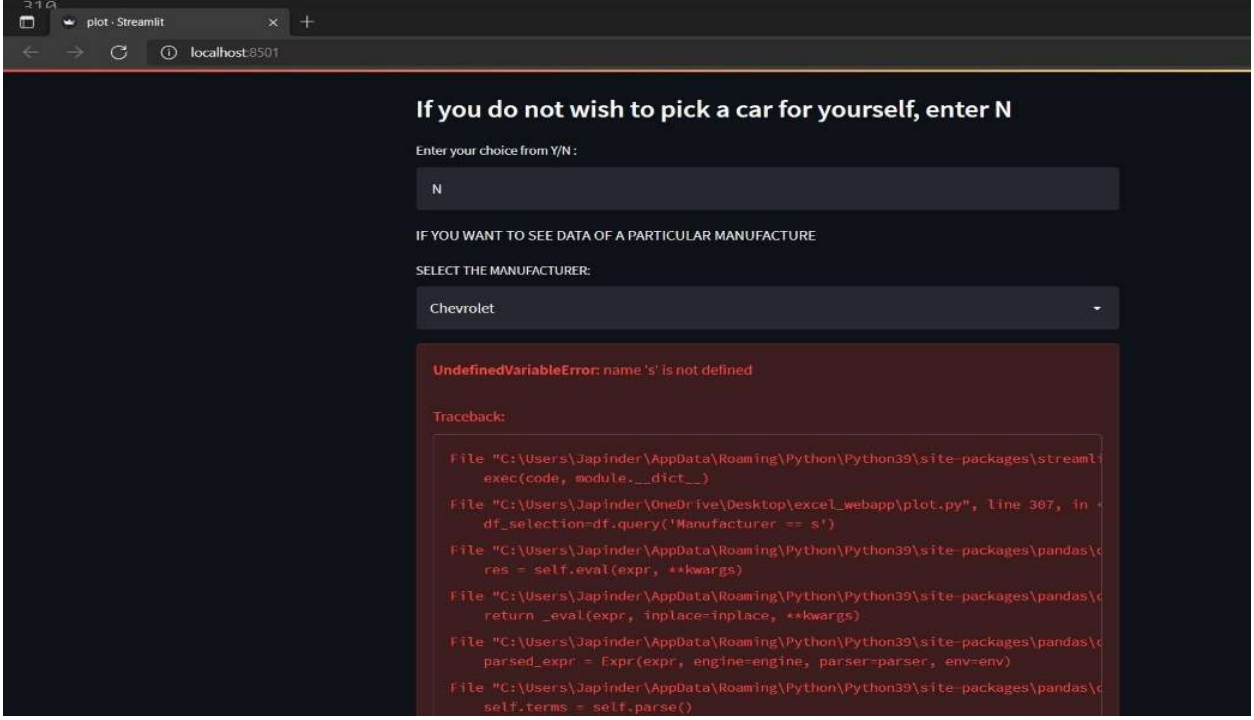


correct



incorrect

```
299 if a=='N':
300
301     st.write('IF YOU WANT TO SEE DATA OF A PARTICULAR MANUFACTURE')
302
303
304     s=(st.selectbox("SELECT THE MANUFACTURER:",
305                     (df['Manufacturer'].unique().tolist())))
306
307     df_selection=df.query('Manufacturer == s')
308     st.write("the cars are :\n",df_selection )
309
```



correct

```
299 if a=='N':
300
301     st.write('IF YOU WANT TO SEE DATA OF A PARTICULAR MANUFACTURE')
302
303
304     s=(st.selectbox("SELECT THE MANUFACTURER:",
305                     (df['Manufacturer'].unique().tolist())))
306
307     df_selection=df.query('Manufacturer == @s')
308     st.write("the cars are :\n",df_selection )
309
```

Project Deployment

SCREENSHOTS OF DATA SOURCE

Manufacturer	Model	Sales in thousands	4-year resale value	Vehicle type	Price in thousands	Engine size	Horsepower	Wheelbase	Width	Length	Curb weight	Fuel capacity	Fuel efficiency
Acura	Integra	16.919	16.36	Passenger	21.5	1.8	140	101.2	67.3	172.4	2.639	13.2	28
Acura	TL	39.384	19.875	Passenger	28.4	3.2	225	108.1	70.3	192.9	3.517	17.2	25
Acura	CL	14.114	18.225	Passenger	.	3.2	225	106.9	70.6	192	3.47	17.2	26
Acura	RL	8.588	29.725	Passenger	42	3.5	210	114.6	71.4	196.6	3.85	18	22
Audi	A4	20.397	22.255	Passenger	23.99	1.8	150	102.6	68.2	178	2.998	16.4	27
Audi	A6	18.78	23.555	Passenger	33.95	2.8	200	108.7	76.1	192	3.561	18.5	22
Audi	A8	1.38	39	Passenger	62	4.2	310	113	74	198.2	3.902	23.7	21
BMW	323i	19.747	.	Passenger	26.99	2.5	170	107.3	68.4	176	3.179	16.6	26
BMW	328i	9.231	28.675	Passenger	33.4	2.8	193	107.3	68.5	176	3.197	16.6	24
BMW	528i	17.527	36.125	Passenger	38.9	2.8	193	111.4	70.9	188	3.472	18.5	25
Buick	Century	91.561	12.475	Passenger	21.975	3.1	175	109	72.7	194.6	3.368	17.5	25
Buick	Regal	39.35	13.74	Passenger	25.3	3.8	240	109	72.7	196.2	3.543	17.5	23
Buick	Park Avenue	27.851	20.19	Passenger	31.965	3.8	205	113.8	74.7	206.8	3.778	18.5	24
Buick	LeSabre	83.257	13.36	Passenger	27.885	3.8	205	112.2	73.5	200	3.591	17.5	25
Cadillac	DeVille	63.729	22.525	Passenger	39.895	4.6	275	115.3	74.5	207.2	3.978	18.5	22
Cadillac	Seville	15.943	27.1	Passenger	44.475	4.6	275	112.2	75	201	.	18.5	22
Cadillac	Eldorado	6.536	25.725	Passenger	39.665	4.6	275	108	75.5	200.6	3.843	19	22
Cadillac	Catera	11.185	18.225	Passenger	31.01	3	200	107.4	70.3	194.8	3.77	18	22
Cadillac	Escalade	14.785	.	Car	46.225	5.7	255	117.5	77	201.2	5.572	30	15
Chevrolet	Cavalier	145.519	9.25	Passenger	13.26	2.2	115	104.1	67.9	180.9	2.676	14.3	27
Chevrolet	Malibu	135.126	11.225	Passenger	16.535	3.1	170	107	69.4	190.4	3.051	15	25
Chevrolet	Lumina	24.629	10.31	Passenger	18.89	3.1	175	107.5	72.5	200.9	3.33	16.6	25
Chevrolet	Monte Carlo	42.593	11.525	Passenger	19.39	3.4	180	110.5	72.7	197.9	3.34	17	27
Chevrolet	Camaro	26.402	13.025	Passenger	24.34	3.8	200	101.1	74.1	193.2	3.5	16.8	25
Chevrolet	Corvette	17.947	36.225	Passenger	45.705	5.7	345	104.5	73.6	179.7	3.21	19.1	22
Chevrolet	Prizm	32.299	9.125	Passenger	13.96	1.8	120	97.1	66.7	174.3	2.398	13.2	33
Chevrolet	Metro	21.855	5.16	Passenger	9.235	1	55	93.1	62.6	149.4	1.895	10.3	45
Chevrolet	Impala	107.995	.	Passenger	18.89	3.4	180	110.5	73	200	3.389	17	27
Chrysler	Sebring Coupe	7.854	12.36	Passenger	19.84	2.5	163	103.7	69.7	190.9	2.967	15.9	24
Chrysler	Sebring Conv.	32.775	14.18	Passenger	24.495	2.5	168	106	69.2	193	3.332	16	24
Chrysler	Concorde	31.148	13.725	Passenger	22.245	2.7	200	113	74.4	209.1	3.452	17	26
Chrysler	Cirrus	32.306	12.64	Passenger	16.48	2	132	108	71	186	2.911	16	27
Chrysler	LHS	13.462	17.325	Passenger	28.34	3.5	253	113	74.4	207.7	3.564	17	23
Chrysler	Town & Country	53.48	19.54	Car	.	.	.	.	.	.	.	.	.
Chrysler	300M	30.696	.	Passenger	29.185	3.5	253	113	74.4	197.8	3.567	17	23
Dodge	Neon	76.034	7.75	Passenger	12.64	2	132	105	74.4	174.4	2.567	12.5	29
Dodge	Avenger	4.734	12.545	Passenger	19.045	2.5	163	103.7	69.1	190.2	2.879	15.9	24
Dodge	Stratus	71.186	10.185	Passenger	20.23	2.5	168	108	71	186	3.058	16	24
Dodge	Intrepid	88.028	12.275	Passenger	22.505	2.7	202	113	74.7	203.7	3.489	17	.
Dodge	Viper	0.916	58.47	Passenger	69.725	8	450	96.2	75.7	176.7	3.375	19	16
Dodge	Ram Pickup	227.061	15.06	Car	19.46	5.2	230	138.7	79.3	224.2	4.47	26	17
Dodge	Ram Wagon	16.767	15.51	Car	21.315	3.9	175	109.6	78.8	192.6	4.245	32	15
Dodge	Ram Van	31.038	13.425	Car	18.575	3.9	175	127.2	78.8	208.5	4.298	32	16
Dodge	Dakota	111.313	11.26	Car	16.98	2.5	120	131	71.5	215	3.557	22	19
Dodge	Durango	101.323	.	Car	26.31	5.2	230	115.7	71.7	193.5	4.394	25	17
Dodge	Caravan	181.749	12.025	Car	19.565	2.4	150	113.3	76.8	186.3	3.533	20	24
Ford	Escort	70.227	7.425	Passenger	12.07	2	110	98.4	67	174.7	2.468	12.7	30

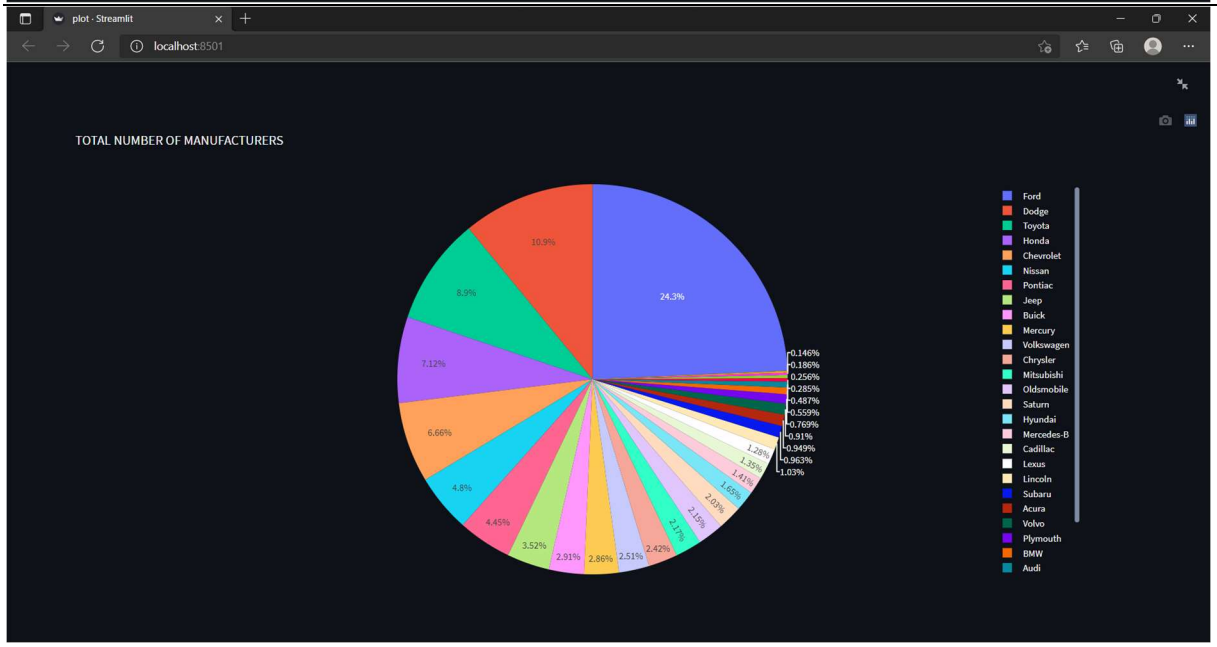
49	Ford	Mustang	113.369	12.76	Passenger	21.56	3.8	190	101.3	73.1	183.2	3.203	15.7	24
50	Ford	Contour	35.068	8.835	Passenger	17.035	2.5	170	106.5	69.1	184.6	2.769	15	25
51	Ford	Taurus	245.815	10.055	Passenger	17.885	3	155	108.5	73	197.6	3.368	16	24
52	Ford	Focus	175.67	.	Passenger	12.315	2	107	103	66.9	174.8	2.564	13.2	30
53	Ford	Crown Victoria	63.403	14.21	Passenger	22.195	4.6	200	114.7	78.2	212	3.908	19	21
54	Ford	Explorer	276.747	16.64	Car	31.93	4	210	111.6	70.2	190.7	3.876	21	19
55	Ford	Windstar	155.787	13.175	Car	21.41	3	150	120.7	76.6	200.9	3.761	26	21
56	Ford	Expedition	125.338	23.575	Car	36.135	4.6	240	119	78.7	204.6	4.808	26	16
57	Ford	Ranger	220.65	7.85	Car	12.05	2.5	119	117.5	69.4	200.7	3.086	20	23
58	Ford	F-Series	540.561	15.075	Car	26.935	4.6	220	138.5	79.1	224.5	4.241	25.1	18
59	Honda	Civic	199.685	9.85	Passenger	12.885	1.6	106	103.2	67.1	175.1	2.339	11.9	32
60	Honda	Accord	230.902	13.21	Passenger	15.35	2.3	135	106.9	70.3	188.8	2.932	17.1	27
61	Honda	CR-V	73.203	17.71	Car	20.55	2	146	103.2	68.9	177.6	3.219	15.3	24
62	Honda	Passport	12.855	17.525	Car	26.6	3.2	205	106.4	70.4	178.2	3.857	21.1	19
63	Honda	Odyssey	76.029	19.49	Car	26	3.5	210	118.1	75.6	201.2	4.288	20	23
64	Hyundai	Accent	41.184	5.86	Passenger	9.699	1.5	92	96.1	65.7	166.7	2.24	11.9	31
65	Hyundai	Elantra	66.692	7.825	Passenger	11.799	2	140	100.4	66.9	174	2.626	14.5	27
66	Hyundai	Sonata	29.45	8.91	Passenger	14.999	2.4	148	106.3	71.6	185.4	3.072	17.2	25
67	Infiniti	I30	23.713	19.69	Passenger	29.465	3	227	108.3	70.2	193.7	3.342	18.5	25
68	Jaguar	S-Type	15.467	.	Passenger	42.8	3	240	114.5	71.6	191.3	3.65	18.4	21
69	Jeep	Wrangler	55.557	13.475	Car	14.46	2.5	120	93.4	66.7	152	3.045	19	17
70	Jeep	Cherokee	80.556	13.775	Car	21.62	4	190	101.4	69.4	167.5	3.194	20	20
71	Jeep	Grand Cherokee	157.04	18.81	Car	26.895	4	195	105.9	72.3	181.5	3.88	20.5	19
72	Lexus	ES300	24.072	26.975	Passenger	31.505	3	210	105.1	70.5	190.2	3.373	18.5	23
73	Lexus	GS300	12.698	32.075	Passenger	37.805	3	225	110.2	70.9	189.2	3.638	19.8	23
74	Lexus	GS400	3.334	.	Passenger	46.305	4	300	110.2	70.9	189.2	3.693	19.8	21
75	Lexus	LS400	6.375	40.375	Passenger	54.005	4	290	112.2	72	196.7	3.89	22.5	22
76	Lexus	LX470	9.126	.	Car	60.105	4.7	230	112.2	76.4	192.5	5.401	25.4	15
77	Lexus	RX300	51.238	.	Car	34.605	3	220	103	71.5	180.1	3.9	17.2	21
78	Lincoln	Continental	13.798	20.525	Passenger	39.08	4.6	275	109	73.6	208.5	3.868	20	22
79	Lincoln	Town car	48.911	21.725	Passenger	43.33	4.6	215	117.7	78.2	215.3	4.121	19	21
80	Lincoln	Navigator	22.925	.	Car	42.66	5.4	300	119	79.9	204.8	5.393	30	15
81	Mitsubishi	Mirage	26.232	8.325	Passenger	13.987	1.8	113	98.4	66.5	173.6	2.25	13.2	30
82	Mitsubishi	Eclipse	42.541	10.395	Passenger	19.047	2.4	154	100.8	68.9	175.4	2.91	15.9	24
83	Mitsubishi	Galant	55.616	10.595	Passenger	17.357	2.4	145	103.7	68.5	187.8	2.945	16.3	25
84	Mitsubishi	Diamante	5.711	16.575	Passenger	24.997	3.5	210	107.1	70.3	194.1	3.443	19	22
85	Mitsubishi	3000GT	0.11	20.94	Passenger	25.45	3	161	97.2	72.4	180.3	3.131	19.8	21
86	Mitsubishi	Montero	11.337	19.125	Car	31.807	3.5	200	107.3	69.9	186.6	4.52	24.3	18
87	Mitsubishi	Montero Sport	39.348	13.88	Car	22.527	3	173	107.3	66.7	178.3	3.51	19.5	20
88	Mercury	Mystique	14.351	8.8	Passenger	16.24	2	125	106.5	69.1	184.8	2.769	15	28
89	Mercury	Cougar	26.529	13.89	Passenger	16.54	2	125	106.4	69.6	185	2.892	16	30
90	Mercury	Sable	67.956	11.03	Passenger	19.035	3	153	108.5	73	199.7	3.379	16	24
91	Mercury	Grand Marquis	81.174	14.875	Passenger	22.605	4.6	200	114.7	78.2	212	3.958	19	21
92	Mercury	Mountaineer	27.609	20.43	Car	27.56	4	210	111.6	70.2	190.1	3.876	21	18
93	Mercury	Villager	20.38	14.795	Car	22.51	3.3	170	112.2	74.9	194.7	3.944	20	21
94	Mercedes-Benz	C-Class	18.392	26.05	Passenger	31.75	2.3	185	105.9	67.7	177.4	3.25	16.4	26
95	Mercedes-Benz	E-Class	27.602	41.45	Passenger	49.9	3.2	221	111.5	70.8	189.4	3.823	21.1	25
96	Mercedes-Benz	S-Class	16.774	50.375	Passenger	69.7	4.3	275	121.5	73.1	203.1	4.133	23.2	21
97	Mercedes-Benz	SL-Class	3.311	58.6	Passenger	82.6	5	302	99	71.3	177.1	4.125	21.1	20
98	Mercedes-Benz	SLK	7.998		Passenger	38.9	2.3	190	94.5	67.5	157.9	3.055	15.9	26

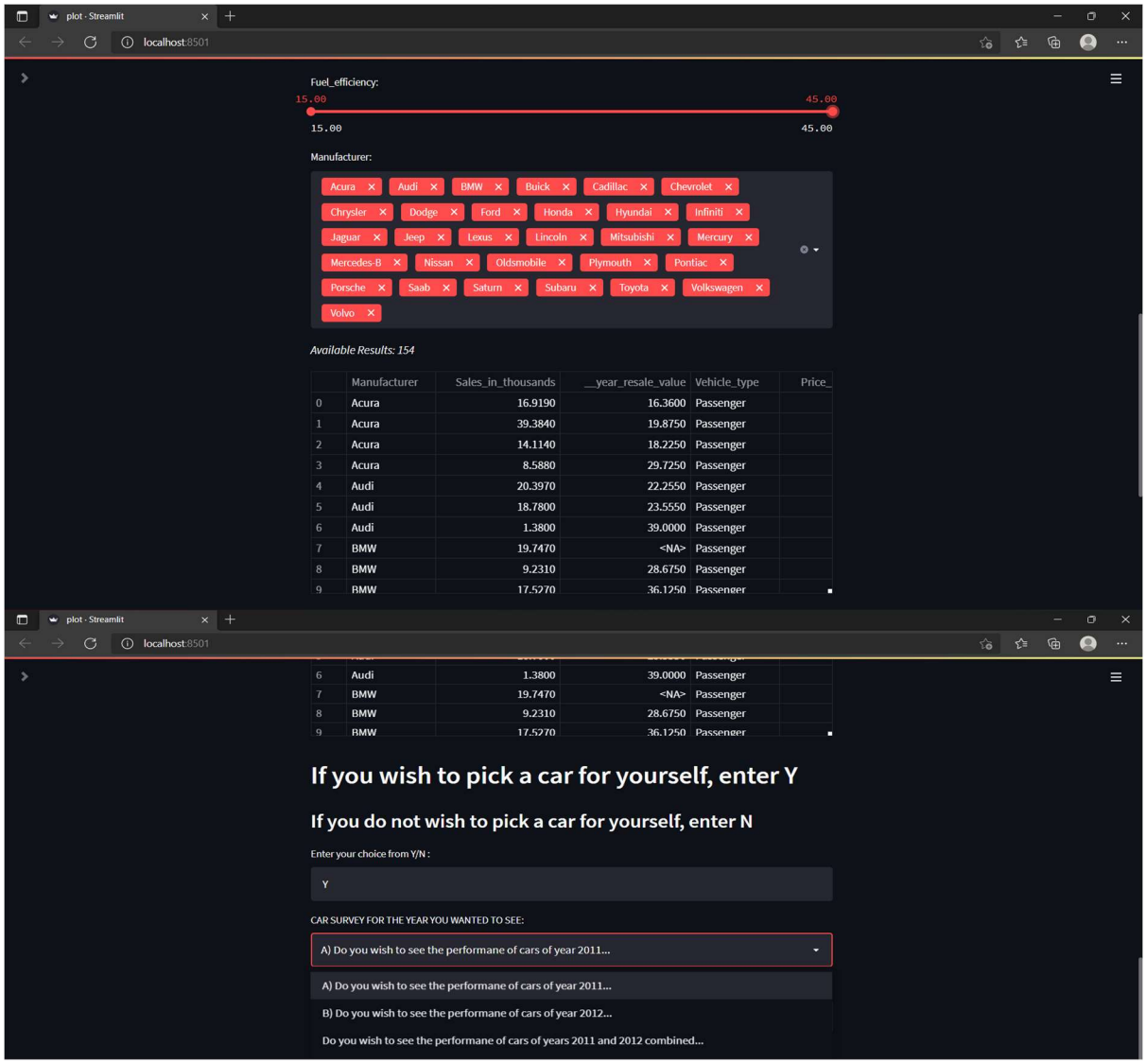
99	Mercedes-Benz	SLK230	1.526	.	Passenger	41	2.3	185	94.5	67.5	157.3	2.975	14	27
100	Mercedes-Benz	CLK Coupe	11.592	.	Passenger	41.6	3.2	215	105.9	67.8	180.3	3.213	16.4	26
101	Mercedes-Benz	CL500	0.954	.	Passenger	85.5	5	302	113.6	73.1	196.6	4.115	23.2	20
102	Mercedes-Benz	M-Class	28.976	.	Car	35.3	3.2	215	111	72.2	180.6	4.387	19	20
103	Nissan	Sentra	42.643	8.45	Passenger	13.499	1.8	126	99.8	67.3	177.5	2.593	13.2	30
104	Nissan	Altima	88.094	11.295	Passenger	20.39	2.4	155	103.1	69.1	183.5	3.012	15.9	25
105	Nissan	Maxima	79.853	15.125	Passenger	26.249	3	222	108.3	70.3	190.5	3.294	18.5	25
106	Nissan	Quest	27.308	15.38	Car	26.399	3.3	170	112.2	74.9	194.8	3.991	20	21
107	Nissan	Pathfinder	42.574	17.81	Car	29.299	3.3	170	106.3	71.7	182.6	3.947	21	19
108	Nissan	Xterra	54.158	.	Car	22.799	3.3	170	104.3	70.4	178	3.821	19.4	18
109	Nissan	Frontier	65.005	.	Car	17.89	3.3	170	116.1	66.5	196.1	3.217	19.4	18
110	Oldsmobile	Cutlass	1.112	11.24	Passenger	18.145	3.1	150	107	69.4	192	3.102	15.2	25
111	Oldsmobile	Intrigue	38.554	.	Passenger	24.15	3.5	215	109	73.6	195.9	3.455	18	.
112	Oldsmobile	Alero	80.255	.	Passenger	18.27	2.4	150	107	70.1	186.7	2.958	15	27
113	Oldsmobile	Aurora	14.69	19.89	Passenger	36.229	4	250	113.8	74.4	205.4	3.967	18.5	22
114	Oldsmobile	Bravada	20.017	19.925	Car	31.598	4.3	190	107	67.8	181.2	4.068	17.5	19
115	Oldsmobile	Silhouette	24.361	15.24	Car	25.345	3.4	185	120	72.2	201.4	3.948	25	22
116	Plymouth	Neon	32.734	7.75	Passenger	12.64	2	132	105	74.4	174.4	2.559	12.5	29
117	Plymouth	Breeze	5.24	9.8	Passenger	16.08	2	132	108	71	186.3	2.942	16	27
118	Plymouth	Voyager	24.155	12.025	Car	18.85	2.4	150	113.3	76.8	186.3	3.528	20	24
119	Plymouth	Prowler	1.872	.	Passenger	43	3.5	253	113.3	76.3	165.4	2.85	12	21
120	Pontiac	Sunfire	51.645	13.79	Passenger	21.61	2.4	150	104.1	68.4	181.9	2.906	15	27
121	Pontiac	Grand Am	131.097	10.29	Passenger	19.72	3.4	175	107	70.4	186.3	3.091	15.2	25
122	Pontiac	Firebird	19.911	17.805	Passenger	25.31	3.8	200	101.1	74.5	193.4	3.492	16.8	25
123	Pontiac	Grand Prix	92.364	14.01	Passenger	21.665	3.8	195	110.5	72.7	196.5	3.396	18	25
124	Pontiac	Bonneville	35.945	13.225	Passenger	23.755	3.8	205	112.2	72.6	202.5	3.59	17.5	24
125	Pontiac	Montana	39.572	.	Car	25.635	3.4	185	120	72.7	201.3	3.942	25	23
126	Porsche	Boxter	8.982	41.25	Passenger	41.43	2.7	217	95.2	70.1	171	2.778	17	22
127	Porsche	Carrera Coupe	1.28	60.625	Passenger	71.02	3.4	300	92.6	69.5	174.5	3.032	17	21
128	Porsche	Carrera Cabriolet	1.866	67.55	Passenger	74.97	3.4	300	92.6	69.5	174.5	3.075	17	23
129	Saab	5-Sep	9.191	.	Passenger	33.12	2.3	170	106.4	70.6	189.2	3.28	18.5	23
130	Saab	3-Sep	12.115	.	Passenger	26.1	2	185	102.6	67.4	182.2	2.99	16.9	23
131	Saturn	SL	80.62	9.2	Passenger	10.685	1.9	100	102.4	66.4	176.9	2.332	12.1	33
132	Saturn	SC	24.546	10.59	Passenger	12.535	1.9	100	102.4	66.4	180	2.367	12.1	33
133	Saturn	SW	5.223	10.79	Passenger	14.29	1.9	124	102.4	66.4	176.9	2.452	12.1	31
134	Saturn	LW	8.472	.	Passenger	18.835	2.2	137	106.5	69	190.4	3.075	13.1	27
135	Saturn	LS	49.989	.	Passenger	15.01	2.2	137	106.5	69	190.4	2.91	13.1	28
136	Subaru	Outback	47.107	.	Passenger	22.695	2.5	165	103.5	67.5	185.8	3.415	16.9	25
137	Subaru	Forester	33.028	.	Car	20.095	2.5	165	99.4	68.3	175.2	3.125	15.9	24
138	Toyota	Corolla	142.535	10.025	Passenger	13.108	1.8	120	97	66.7	174	2.42	13.2	33
139	Toyota	Camry	247.994	13.245	Passenger	17.518	2.2	133	105.2	70.1	188.5	2.998	18.5	27
140	Toyota	Avalon	63.849	18.14	Passenger	25.545	3	210	107.1	71.7	191.9	3.417	18.5	26
141	Toyota	Celica	33.269	15.445	Passenger	16.875	1.8	140	102.4	68.3	170.5	2.425	14.5	31
142	Toyota	Tacoma	84.087	9.575	Car	11.528	2.4	142	103.3	66.5	178.7	2.58	15.1	23
143	Toyota	Sienna	65.119	.	Car	22.368	3	194	114.2	73.4	193.5	3.759	20.9	22
144	Toyota	RAV4	25.106	13.325	Car	16.888	2	127	94.9	66.7	163.8	2.668	15.3	27
145	Toyota	4Runner	68.411	19.425	Car	22.288	2.7	150	105.3	66.5	183.3	3.44	18.5	23
146	Toyota	Land Cruiser	9.835	34.08	Car	51.728	4.7	230	112.2	76.4	192.5	5.115	25.4	15
147	Volkswagen	Golf	9.761	11.425	Passenger	14.9	2	115	98.9	68.3	163.3	2.767	14.5	26

148	Volkswagen	Jetta	83.721	13.24	Passenger	16.7	2	115	98.9	68.3	172.3	2.853	14.5	26
149	Volkswagen	Passat	51.102	16.725	Passenger	21.2	1.8	150	106.4	68.5	184.1	3.043	16.4	27
150	Volkswagen	Cabrio	9.569	16.575	Passenger	19.99	2	115	97.4	66.7	160.4	3.079	13.7	26
151	Volkswagen	GTI	5.596	13.76	Passenger	17.5	2	115	98.9	68.3	163.3	2.762	14.6	26
152	Volkswagen	Beetle	49.463	.	Passenger	15.9	2	115	98.9	67.9	161.1	2.769	14.5	26
153	Volvo	S40	16.957	.	Passenger	23.4	1.9	160	100.5	67.6	176.6	2.998	15.8	25
154	Volvo	V40	3.545	.	Passenger	24.4	1.9	160	100.5	67.6	176.6	3.042	15.8	25
155	Volvo	S70	15.245	.	Passenger	27.5	2.4	168	104.9	69.3	185.9	3.208	17.9	25
156	Volvo	V70	17.531	.	Passenger	28.8	2.4	168	104.9	69.3	186.2	3.259	17.9	25
157	Volvo	C70	3.493	.	Passenger	45.5	2.3	236	104.9	71.5	185.7	3.601	18.5	23
158	Volvo	S80	18.969	.	Passenger	36	2.9	201	109.9	72.1	189.8	3.6	21.1	24

SCREENSHOTS OF OUTPUT OF DATA

	Manufacturer	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_
0	Acura	16.9190	16.3600	Passenger	
1	Acura	39.3840	19.8750	Passenger	
2	Acura	14.1140	18.2250	Passenger	
3	Acura	8.5880	29.7250	Passenger	
4	Audi	20.3970	22.2550	Passenger	
5	Audi	18.7800	23.5550	Passenger	
6	Audi	1.3800	39.0000	Passenger	
7	BMW	19.7470	<NA>	Passenger	
8	BMW	9.2310	28.6750	Passenger	
9	BMW	17.5270	36.1250	Passenger	




plot - Streamlit
localhost:8501
Fuel_efficiency:
15.00
45.00
15.00
45.00
Manufacturer:
Acura Audi BMW Buick Cadillac Chevrolet
Chrysler Dodge Ford Honda Hyundai Infiniti
Jaguar Jeep Lexus Lincoln Mitsubishi Mercury
Mercedes-B Nissan Oldsmobile Plymouth Pontiac
Porsche Saab Saturn Subaru Toyota Volkswagen
Volvo
Available Results: 154
Manufacturer Sales_in_thousands __year_resale_value Vehicle_type Price_
0 Acura 16.9190 16.3600 Passenger
1 Acura 39.3840 19.8750 Passenger
2 Acura 14.1140 18.2250 Passenger
3 Acura 8.5880 29.7250 Passenger
4 Audi 20.3970 22.2550 Passenger
5 Audi 18.7800 23.5550 Passenger
6 Audi 1.3800 39.0000 Passenger
7 BMW 19.7470 <NA> Passenger
8 BMW 9.2310 28.6750 Passenger
9 BMW 17.5270 36.1250 Passenger
plot - Streamlit
localhost:8501
6 Audi 1.3800 39.0000 Passenger
7 BMW 19.7470 <NA> Passenger
8 BMW 9.2310 28.6750 Passenger
9 BMW 17.5270 36.1250 Passenger
If you wish to pick a car for yourself, enter Y
If you do not wish to pick a car for yourself, enter N
Enter your choice from Y/N :
Y
CAR SURVEY FOR THE YEAR YOU WANTED TO SEE:
A) Do you wish to see the performane of cars of year 2011...
A) Do you wish to see the performane of cars of year 2011...
B) Do you wish to see the performane of cars of year 2012...
Do you wish to see the performane of cars of years 2011 and 2012 combined...

Enter your choice to see the performance of cars in 2011 in various aspects

Enter 1 if you wish to see the 10 leading fuel efficient cars :

10 leading fuel efficient cars are :

	Manufacturer	Model	Latest_Launch	Vehicle_type	Price_in_thousands
25	Chevrolet	Prizm	2011-11-09T00:00:00	Passenger	13.9600
130	Saturn	SC	2011-03-16T00:00:00	Passenger	12.5350
136	Toyota	Corolla	2011-11-04T00:00:00	Passenger	13.1080
57	Honda	Civic	2011-10-21T00:00:00	Passenger	12.8850
131	Saturn	SW	2011-01-15T00:00:00	Passenger	14.2900
101	Nissan	Sentra	2011-08-31T00:00:00	Passenger	13.4990
114	Plymouth	Neon	2011-04-26T00:00:00	Passenger	12.6400
35	Dodge	Neon	2011-12-12T00:00:00	Passenger	12.6400
142	Toyota	RAV4	2011-06-05T00:00:00	Car	16.8880
115	Plymouth	Breeze	2011-11-14T00:00:00	Passenger	16.0806

Enter your choice to see the performance of cars in 2011 in various aspects

Enter 4 if you wish to see the price list of cars (from highest to lowest) :

price list of the cars :

	Manufacturer	Latest_Launch	Vehicle_type	Price_in_thousands	Horsepo
99	Mercedes-B	2011-11-04T00:00:00	Passenger	85.5000	302.0
95	Mercedes-B	2011-03-17T00:00:00	Passenger	82.6000	302.0
126	Porsche	2011-11-07T00:00:00	Passenger	74.9700	300.0
39	Dodge	2011-07-08T00:00:00	Passenger	69.7250	450.0
94	Mercedes-B	2011-06-13T00:00:00	Passenger	69.7000	275.0
144	Toyota	2011-09-25T00:00:00	Car	51.7280	230.0
93	Mercedes-B	2011-12-07T00:00:00	Passenger	49.9000	221.0
155	Volvo	2011-04-26T00:00:00	Passenger	45.5000	236.0
15	Cadillac	2011-04-29T00:00:00	Passenger	44.4750	275.0
3	Acura	2011-10-03T00:00:00	Passenger	42.0000	210.0

Enter your choice to see the performance of cars in 2011 in various aspects

Enter 5 if you wish to see the information regarding the most sold car :

price list of the cars :

	Manufacturer	Model	Latest_Launch	Vehicle_type	Sales_in_thousands
137	Toyota	Camry	2011-10-02T00:00:00	Passenger	247.9940

CAR SURVEY FOR THE YEAR YOU WANTED TO SEE:

B) Do you wish to see the performane of cars of year 2012...

Enter your choice to see the performance of cars in 2012 in various aspects

Enter 1 if you wish to see the 10 leading fuel efficient cars :

10 leading fuel efficient cars are :

	Manufacturer	Model	Latest_Launch	Vehicle_type	Price_in_thousands
26	Chevrolet	Metro	2012-04-13T00:00:00	Passenger	9.2350
129	Saturn	SL	2012-08-16T00:00:00	Passenger	10.6850
62	Hyundai	Accent	2012-10-09T00:00:00	Passenger	9.6990
139	Toyota	Celica	2012-12-29T00:00:00	Passenger	16.8750
50	Ford	Focus	2012-07-22T00:00:00	Passenger	12.3150
46	Ford	Escort	2012-03-31T00:00:00	Passenger	12.0700
79	Mitsubishi	Mirage	2012-04-23T00:00:00	Passenger	13.9870
87	Mercury	Cougar	2012-02-23T00:00:00	Passenger	16.5400
0	Acura	Integra	2012-02-02T00:00:00	Passenger	21.5000
133	Saturn	LS	2012-04-12T00:00:00	Passenger	15.0106

Enter your choice to see the performance of cars in 2012 in various aspects

Enter 2 if you wish to see the Leading power Performance Stats :

Cars for leading Power performance stats are :

	Manufacturer	Model	Latest_Launch	Vehicle_type	Price_in_thousands
24	Chevrolet	Corvette	2012-12-05T00:00:00	Passenger	45.7050

Enter your choice to see the performance of cars in 2012 in various aspects

Enter 3 if you wish to see the Best HorsePower Engine :

Cars with best horsepower engine :

	Manufacturer	Model	Latest_Launch	Vehicle_type	Price_in_thousands
24	Chevrolet	Corvette	2012-12-05T00:00:00	Passenger	45.7050

Enter your choice to see the performance of cars in 2012 in various aspects

Enter 4 if you wish to see the price list of cars (from highest to lowest) :

price list of the cars :

	Manufacturer	Latest_Launch	Vehicle_type	Price_in_thousands	Horsepo
125	Porsche	2012-12-21T00:00:00	Passenger	71.0200	300.0
6	Audi	2012-02-27T00:00:00	Passenger	62.0000	310.0
74	Lexus	2012-10-30T00:00:00	Car	60.1050	230.0
73	Lexus	2012-03-29T00:00:00	Passenger	54.0050	290.0
72	Lexus	2012-11-28T00:00:00	Passenger	46.3050	300.0
18	Cadillac	2012-04-17T00:00:00	Car	46.2250	255.0
24	Chevrolet	2012-12-05T00:00:00	Passenger	45.7050	345.0
77	Lincoln	2012-04-06T00:00:00	Passenger	43.3300	215.0
117	Plymouth	2012-06-27T00:00:00	Passenger	43.0000	253.0
66	Jaguar	2012-03-11T00:00:00	Passenger	42.8000	240.4

Enter your choice to see the performance of cars in 2012 in various aspects

Enter 5 if you wish to see the information regarding the most sold car :

price list of the cars :

	Manufacturer	Model	Latest_Launch	Vehicle_type	Sales_in_thousands
56	Ford	F-Series	2012-08-16T00:00:00	Car	540.5610

CAR SURVEY FOR THE YEAR YOU WANTED TO SEE:

Do you wish to see the performane of cars of years 2011 and 2012 combined...

Enter your choice to see the performance of cars in 2012 in various aspects

Enter 1 if you wish to see the 10 leading fuel efficient cars :

10 leading fuel efficient cars are :

	Manufacturer	Model	Latest_Launch	Vehicle_type	Price_in_thousands
26	Chevrolet	Metro	2012-04-13T00:00:00	Passenger	9.2350
136	Toyota	Corolla	2011-11-04T00:00:00	Passenger	13.1080
130	Saturn	SC	2011-03-16T00:00:00	Passenger	12.5350
129	Saturn	SL	2012-08-16T00:00:00	Passenger	10.6850
25	Chevrolet	Prizm	2011-11-09T00:00:00	Passenger	13.9600
57	Honda	Civic	2011-10-21T00:00:00	Passenger	12.8850
62	Hyundai	Accent	2012-10-09T00:00:00	Passenger	9.6990
139	Toyota	Celica	2012-12-29T00:00:00	Passenger	16.8750
131	Saturn	SW	2011-01-15T00:00:00	Passenger	14.2900
46	Ford	Escort	2012-03-31T00:00:00	Passenger	12.0706

Enter your choice to see the performance of cars in 2011 AND 2012 in various aspects

Enter 2 if you wish to see the Leading power Performance Stats :

Cars for leading Power performance stats are :

	Manufacturer	Model	Latest_Launch	Vehicle_type	Price_in_thousands
39	Dodge	Viper	2011-07-08T00:00:00	Passenger	69.7250

Enter your choice to see the performance of cars in 2011 AND 2012 in various aspects

Enter 3 if you wish to see the Best HorsePower Engine :

Cars with best horsepower engine :

	Manufacturer	Model	Latest_Launch	Vehicle_type	Price_in_thousands
39	Dodge	Viper	2011-07-08T00:00:00	Passenger	69.7250

Enter your choice to see the performance of cars in 2011 AND 2012 in various aspects

Enter 5 if you wish to see the information regarding the most sold car :

price list of the cars :

	Manufacturer	Model	Vehicle_type	Sales_in_thousands	Price_in_thousands
56	Ford	F-Series	Car	540.5610	26.9350

Enter your choice from Y/N :

N

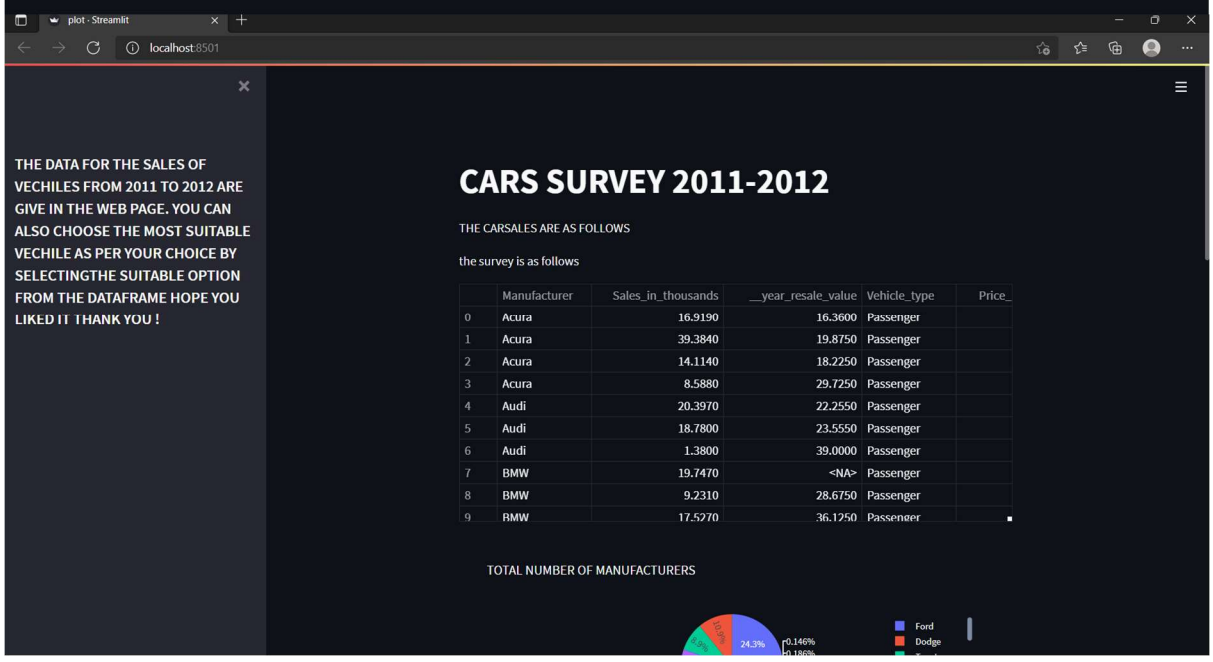
IF YOU WANT TO SEE DATA OF A PARTICULAR MANUFACTURE

SELECT THE MANUFACTURER:

Audi

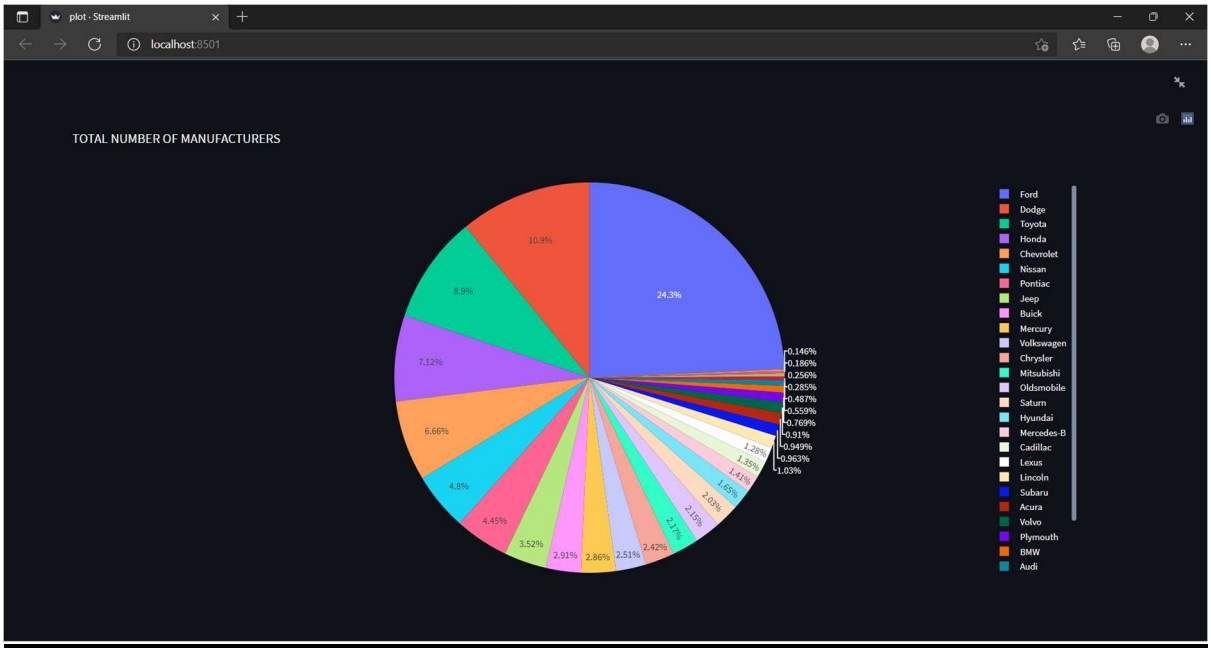
the cars are :

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type
4	Audi	A4	20.3970	22.2550	Passenger
5	Audi	A6	18.7800	23.5550	Passenger
6	Audi	A8	1.3800	39.0000	Passenger



	Manufacturer	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_
0	Acura	16.9190	16.3600	Passenger	
1	Acura	39.3840	19.8750	Passenger	
2	Acura	14.1140	18.2250	Passenger	
3	Acura	8.5880	29.7250	Passenger	
4	Audi	20.3970	22.2550	Passenger	
5	Audi	18.7800	23.5550	Passenger	
6	Audi	1.3800	39.0000	Passenger	
7	BMW	19.7470	<NA>	Passenger	
8	BMW	9.2310	28.6750	Passenger	
9	BMW	17.5270	36.1250	Passenger	

SCREENSHOTS OF GRAPHS



BIBLIOGRAPHY

For making this project I have taken help from the following websites and e-books:

i) https://www.geeksforgeeks.org/create-interactive-dashboard-in-python-using-streamlit/#:~:text=%20%20%201%20Step%201%3A%20First%2C%20We,to%20the%20select%20box%2C%20we%20have...%20More%20

ii)https://github.com/chandanverma07/DataSets/blob/master/Car_sales.csv