

**Shri Vaishnav Vidyapeeth Vishwavidyalaya**

**Indore**

**DATA SCIENCE**

**using Python**

“**CAR EVALUATION**”

**TRAINING PROJECT REPORT**

**Submitted by**:

**AYUSH JAIN** 17100BTCSE01217

**AYUSH SAHU** 17100BTCSE01218

**DEEPAK BALODIYA** 17100BTCSE01221

**JAYANT GAWALI** 17100BTCSE01231

**KANHA GUPTA** 17100BTCSE01233

**COMPUTER SCIENCE & ENGINEERING**

**IN BACHELOR OF TECHNOLOGY**

**JULY-DECEMBER 2020**

**A REPORT OF TWO WEEKS INDUSTRIAL TRAINING AT**

**WebTek Labs Pvt. Ltd.**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF**

**BACHELOR OF TECHNOLOGY**

**COMPUTER SCIENCE & ENGINEERING**



**JULY-DECEMBER 2020**

## Submitted by:

**AYUSH JAIN** 17100BTCSE01217

**AYUSH SAHU** 17100BTCSE01218

**DEEPAK BALODIYA** 17100BTCSE01221

**JAYANT GAWALI** 17100BTCSE01231

**KANHA GUPTA** 17100BTCSE01233

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY, INDORE**

**SVVV, INDORE**

**CANDIDATE‘S DECLARATION**

We hereby declare that we have undertaken industrial training at “**WEBTEK LABS PVT. LTD.”** during a period from 27th July to 16th August in partial fulfilment of requirements for the award of degree of B.Tech (COMPUTER SCIENCE & ENGINEERING) at **“Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore”**. The work which is being presented in the training report submitted to Department of **Computer Science & Engineering** at “Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore” is an authentic record of training work.

**AYUSH JAIN \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**AYUSH SAHU \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DEEPAK BALODIYA \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**JAYANT GAWALI \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**KANHA GUPTA \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Student name Signature of the Student

**ACKNOWLEDGEMENT**

It gives us great pleasure to acknowledge the guidance, assistance and support of **Ms. Mousita Dhar** in making the Project and this Project report successful, which has been structured under her valued suggestion.

She has helped us to accomplish the challenging task in a very short period of time.

Finally, we express the constant support of our friends, family and professors for inspiring us throughout and encouraging us.

**AYUSH JAIN** 17100BTCSE01217

**AYUSH SAHU** 17100BTCSE01218

**DEEPAK BALODIYA** 17100BTCSE01221

**JAYANT GAWALI** 17100BTCSE01231

**KANHA GUPTA** 17100BTCSE01233

SEMESTER: **VII**

Computer Science and Engineering

**CERTIFICATE OF APPROVAL**

The project **“CAR EVALUATION”** made by the efforts of **Ayush Jain, Ayush Sahu, Deepak Balodiya, Jayant Gawali, Kanha Gupta** is hereby approved as a creditable study for the Bachelor of Technology in **Computer Science & Engineering** and presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned this project only for the purpose for which it is submitted.

\_\_\_\_\_\_\_\_\_\_\_\_\_

**Ms. Mousita**

(Project In charge)

**1. INTRODUCTION**

**1.1 PYTHON**

**About Python**

* Python is a high-level, general-purpose, open source, strictly typed programming language. The language provides constructs intended to enable clear programs on both a small and large scale.
* Python was created By Guido van Rossum.
* The Python Software Foundation (PSF) is the organization behind Python.

**Python Versions**

* First released in 1991.
* Python 2.0 was released on 16 October 2000
* Python 3.0 was released on 3 December 2008

**Current Version**

* 3.8.0

**Python features**

**•** Easy to understand

**•** Dynamic

**•** Object oriented

**•** Multipurpose

**•** Strongly typed

**•** Open Sourced

**Python Uses**

**•** Web Development

**•** Data Analysis

**•** Machine Learning

**•** Internet Of Things

**•** GUI Development

**•** Image processing

**•** Data visualization

**•** Game Development

**IDLE:**

IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language.

**1.2 ANACONDA**

**1.2.1 About Anaconda**

Anaconda is a open source Distribution for data science and machine learning using python. It includes hundreds of popular data science packages and the conda package and virtual environment manager for Windows, Linux, and MacOS. Conda makes it quick and easy to install, run, and upgrade complex data science and machine learning environments like scikit-learn, TensorFlow, and SciPy. Anaconda Distribution is the foundation of millions of data science projects as well as Amazon Web Service Machine Learning AMIs and Anaconda for Microsoft on Azure and Windows.

**1.3 PACKAGES**

**1.3.1 Numpy**

NumPy is the fundamental package for scientific computing with Python.

It contains among other things:

• a powerful N-dimensional array object

• sophisticated (broadcasting) functions

• tools for integrating C/C++ and Fortran code

• useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**1.3.2 Pandas**

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Pandas library is well suited for data manipulation and analysis using python. In particular, it offers data structures and operations for manipulating numerical tables

and time series.

**1.3.3 Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web applicationservers, and four graphical user interface toolkits. Matplotlib was originally written by John D. Hunter, since then it has an active development community, and is distributed under a BSD- style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012, and further joined by Thomas Caswell.

**1.3.4 Seaborn**

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

* A dataset-oriented API for examining relationships between multiple variables
* Specialized support for using categorical variables to show observations or aggregate statistics
* Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data.

Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

**1.3.5 Scikit-Learn**

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn is largely written in Python, and uses numpy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible.

**2. TRAINING WORK UNDERTAKEN**

**2.1 COLLECTING DATA FROM KAGGLE**

Kaggle, a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

Kaggle got its start in 2010 by offering machine learning competitions and now also offers a public data platform, a cloud-based workbench for data science, and Artificial Intelligence education. Its key personnel were Anthony Goldbloom and Jeremy Howard. Nicholas Gruen was founding chair succeeded by Max Levchin. Equity was raised in 2011 valuing the company at $25 million. On 8 March 2017, Google announced that they were acquiring Kaggle.

**2.2 DATA SCIENCE**

Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. Data science is related to data mining, machine learning and big data.

Data science is a "concept to unify statistics, data analysis, machine learning, domain knowledge and their related methods" in order to "understand and analyze actual phenomena" with data. It uses techniques and theories drawn from many fields within the context of mathematics, statistics, computer science, domain knowledge and information science. Turing award winner Jim Gray imagined data science as a "fourth paradigm" of science (empirical, theoretical, computational and now data-driven) and asserted that "everything about science is changing because of the impact of information technology" and the data deluge.

Specializations and associated careers

* **Machine Learning Scientist**: Machine learning scientists research new methods of data analysis and create algorithms.
* **Data Engineer**: Data Engineers prepare the “big data” infrastructure to be analyzed by Data Scientists. They are software engineers who design, build, integrate data from various resources, and manage big data.
* **Data Analyst**: Data analysts utilize large data sets to gather information that meets their company’s needs.
* **Data Consultant**: Data consultants work with businesses to determine the best usage of the information yielded from data analysis.
* **Data Architect**: Data architects build data solutions that are optimized for performance and design applications.
* **Applications Architect**: Applications architects track how applications are used throughout a business and how they interact with users and other applications.

**2. 3 SOURCE CODE & OUTPUT**

1. **Import Packages**

import numpy as np

import pandas as pd

import seaborn as sb

import matplotlib.pyplot as plt

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report,accuracy\_score,confusion\_matrix

from sklearn import preprocessing

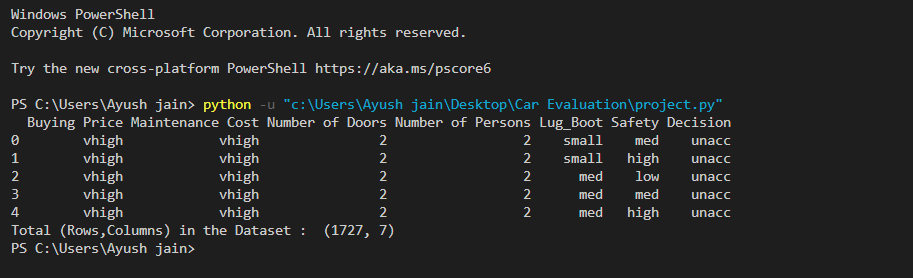
1. **Reading CSV File**

url = r"C:\Users\Ayush jain\Desktop\car\_evaluation.csv"

car = pd.read\_csv(url)

print("The First Five Datas are : ",car.head())

print("Total (Rows,Columns) in the Dataset : ",car.shape)



1. **Feature Engineering**

#1 CONVERT STRING TO NUMBER

# a) Using Map function

car['Lug\_Boot'] = car['Lug\_Boot'].map({'small':0,'med':1,'big':2})

car['Safety'] = car['Safety'].map({'low':0,'med':1,'high':2})

car['Maintenance Cost'] =car['Maintenance Cost'].map({'low':0,'med':1,'high':2,'vhigh':3})

car['Buying Price'] = car['Buying Price'].map({'low':0,'med':1,'high':2,'vhigh':3})

# b) Using Label-Encoder

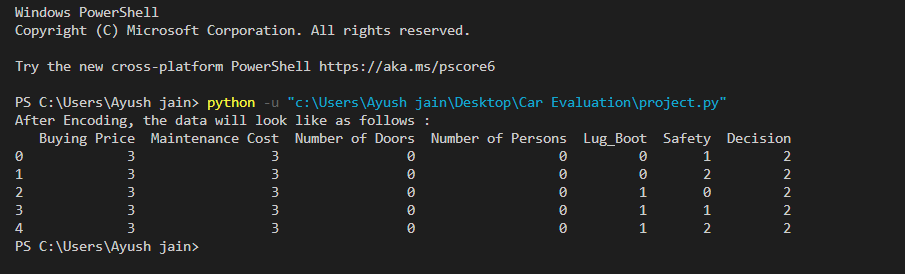
label\_encoder = preprocessing.LabelEncoder()

car['Number of Doors'] = label\_encoder.fit\_transform(car['Number of Doors']) # 2=0, 3=1, 4=2, 5more=3

car['Number of Persons'] = label\_encoder.fit\_transform(car['Number of Persons']) # 2=0, 4=1, more=2

car['Decision'] = label\_encoder.fit\_transform(car['Decision'])

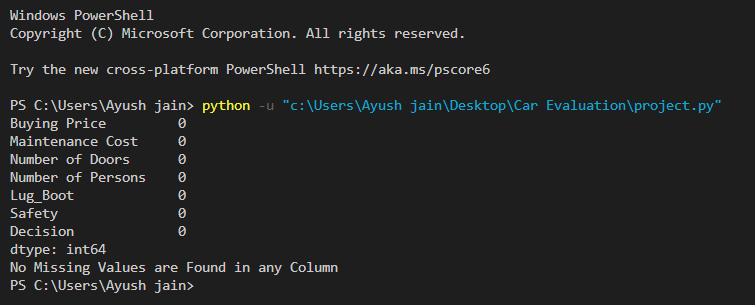
print("After Encoding, the data will look like as follows :", car.head())



#2 CHECKING FOR MISSING VALUES

print(car.isnull().sum())

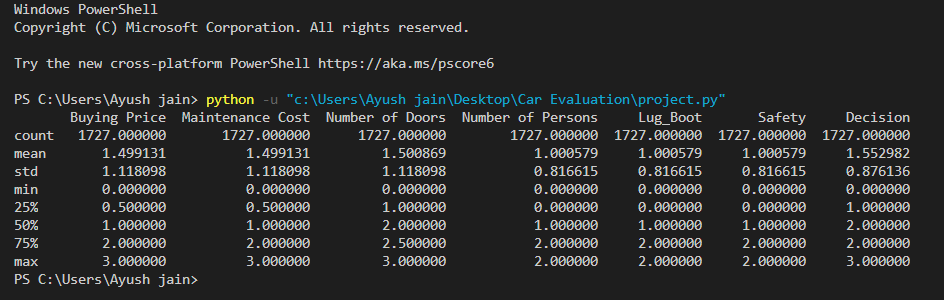
print("No Missing Values are Found in any Column")



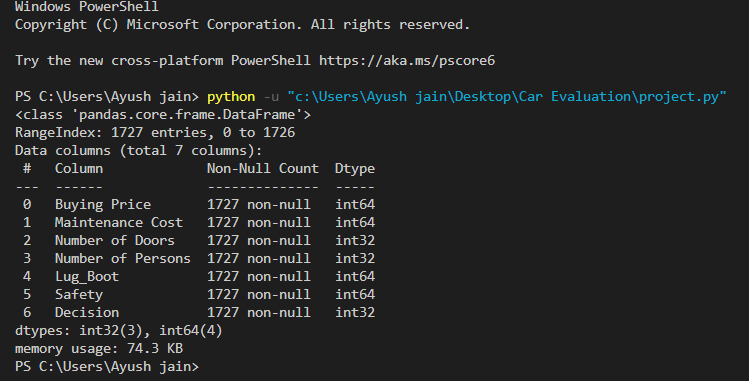
1. **Analyzing (Pre-Processing)**

# For Detail Information like Quartile Range, Mean etc...

car.describe()



car.info()



# Types of Graph Plotting

#1

print("The Histogram for Each Column are :")

for column in car:

plt.hist(car[column], color="red")

plt.xlabel(column)

plt.ylabel("Count")

plt.title(f"Histogram for {column}")

plt.show()

#2

print("The BoxPlot for Each Column are :")

fig, axs = plt.subplots(ncols=7,nrows=1,figsize=(20,10))

index=0

axs = axs.flatten()

for k,v in car.items():

sb.boxplot(y=v, data=car, ax=axs[index])

index = index + 1

plt.tight\_layout(pad=0.4, w\_pad=0.1, h\_pad=5.0)

plt.show()

#3

print("The DistPlot for Each Column are :")

fig, axs = plt.subplots(ncols=7,nrows=1,figsize=(20,10))

index=0

axs = axs.flatten()

for k,v in car.items():

sb.distplot(v,ax=axs[index])

index = index + 1

plt.tight\_layout(pad=0.4, w\_pad=0.1, h\_pad=5.0)

plt.show()

#4

print("The RegPlot for Each Column are :")

for column in car:

sb.regplot(x=column, y='Decision', data=car)

plt.show()

#5

print("The CountPlot for Each Column are :")

for column in car:

sb.countplot(car[column])

plt.show()

#6

print("The BarPlot for Each Column are :")

for column in car:

sb.barplot(car[column])

plt.show()

# Value Count for Each Column

car['Buying Price'].value\_counts(),

car['Maintenance Cost'].value\_counts(),

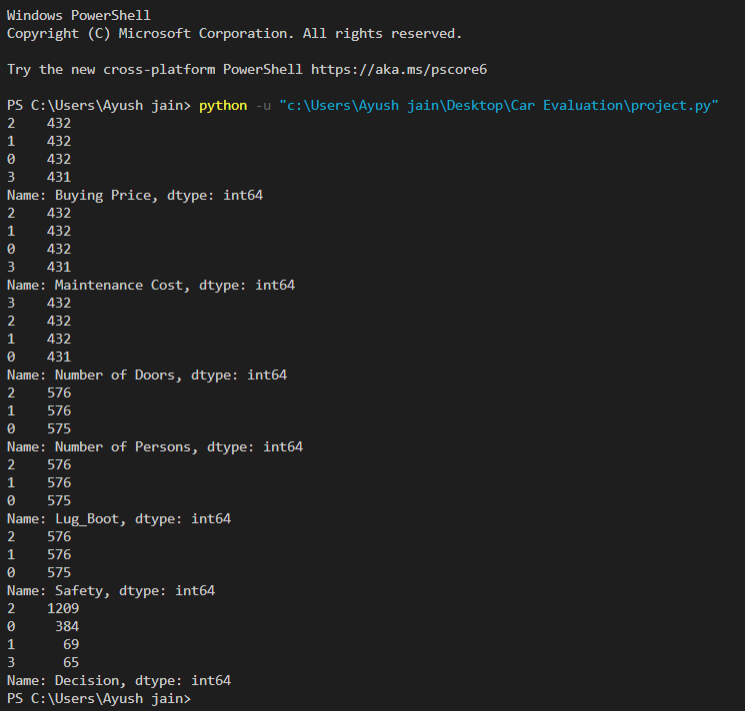
car['Number of Doors'].value\_counts(),

car['Number of Persons'].value\_counts(),

car['Lug\_Boot'].value\_counts(),

car['Safety'].value\_counts(),

car['Decision'].value\_counts()



# Relationship of one column with target column

car[['Buying Price', 'Decision']].groupby(['Buying Price'], as\_index=False).mean()

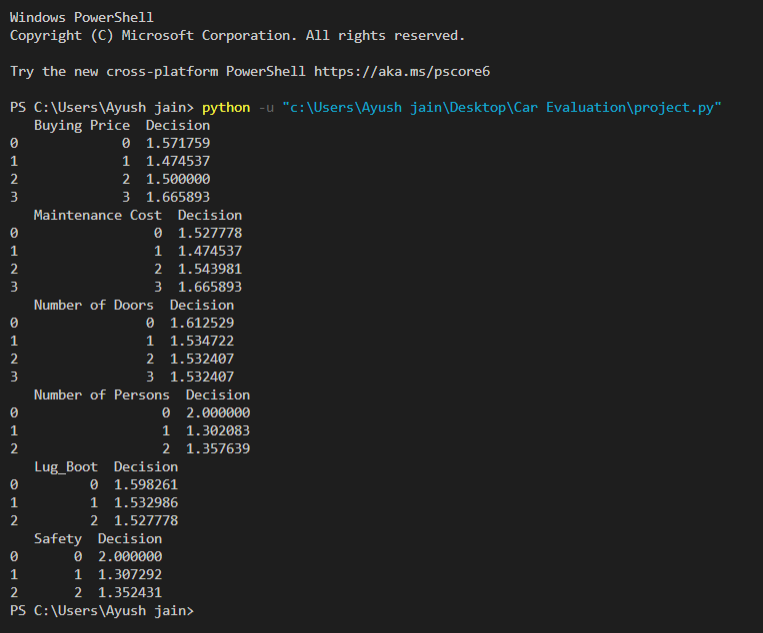
car[['Maintenance Cost', 'Decision']].groupby(['Maintenance Cost'], as\_index=False).mean()

car[['Number of Doors', 'Decision']].groupby(['Number of Doors'], as\_index=False).mean()

car[['Number of Persons', 'Decision']].groupby(['Number of Persons'], as\_index=False).mean()

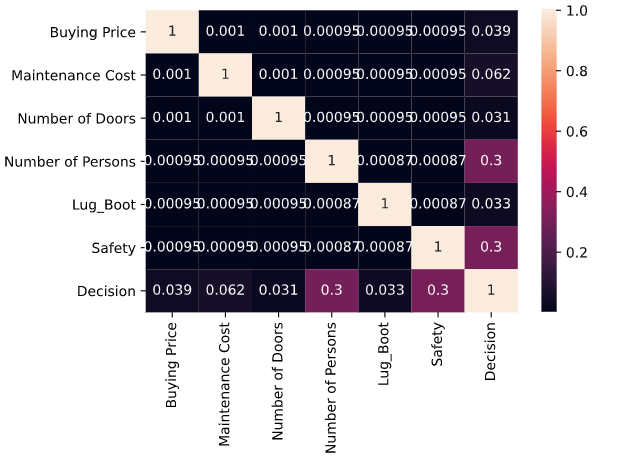
car[['Lug\_Boot', 'Decision']].groupby(['Lug\_Boot'], as\_index=False).mean()

car[['Safety', 'Decision']].groupby(['Safety'], as\_index=False).mean()



# Correlation Matrix

sb.heatmap(car.corr().abs(), annot=True)



1. **Classification Algorithms**

X = car.iloc[:, 0:6]

Y = car.iloc[:, 6]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,Y, test\_size=0.2, random\_state=0)

#1 **KNeighborsClassifier Model**

kn = KNeighborsClassifier(n\_neighbors=7)

kn.fit(X\_train,y\_train)

y\_pred = kn.predict(X\_test)

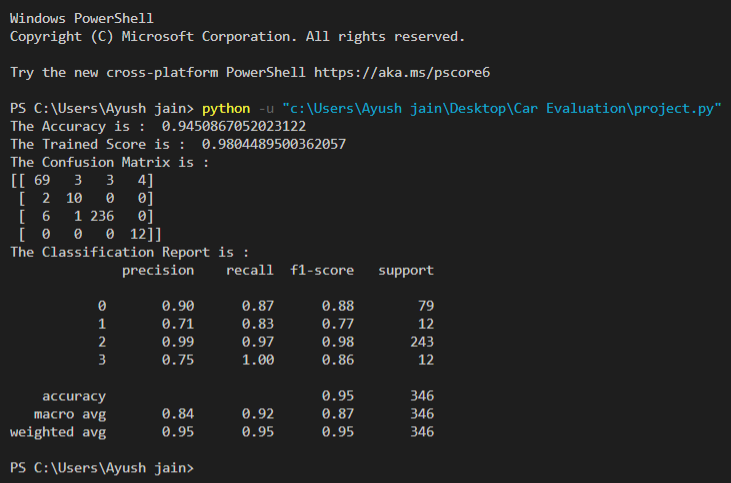
print("The Accuracy is : ",accuracy\_score(y\_pred,y\_test))

print("The Trained Score is : ",kn.score(X\_train,y\_train))

print("The Confusion Matrix is : ",confusion\_matrix(y\_pred, y\_test))

print("The Classification Report is :")

print(classification\_report(y\_pred, y\_test))



#2 **RandomForestClassifier Model**

rf = RandomForestClassifier(n\_estimators=100)

rf.fit(X\_train,y\_train)

y\_pred = rf.predict(X\_test)

print("The Accuracy is : ",accuracy\_score(y\_pred,y\_test))

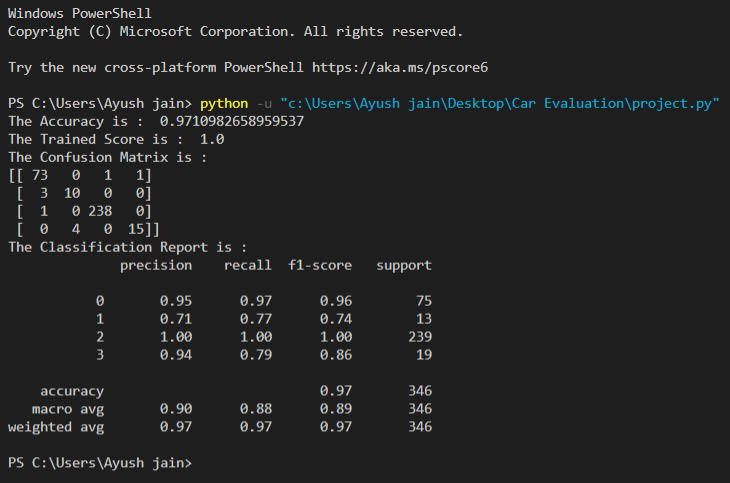
print("The Trained Score is : ",rf.score(X\_train,y\_train))

print("The Confusion Matrix is : ")

print(confusion\_matrix(y\_pred, y\_test))

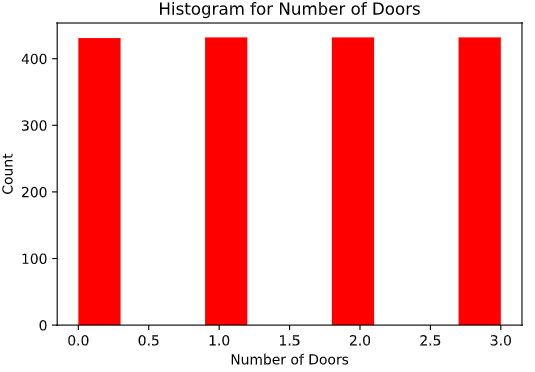
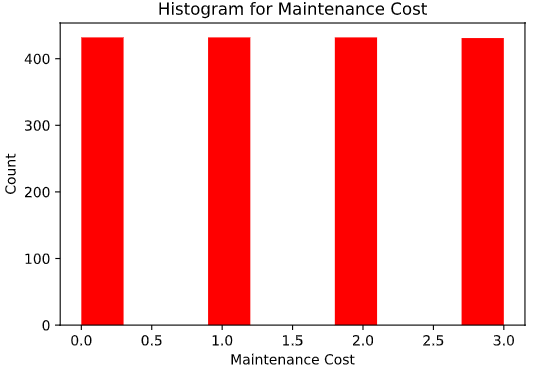
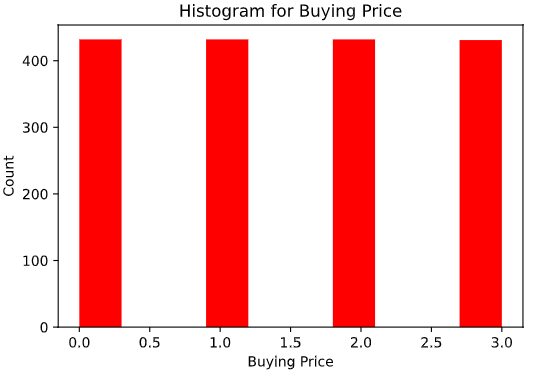
print("The Classification Report is :")

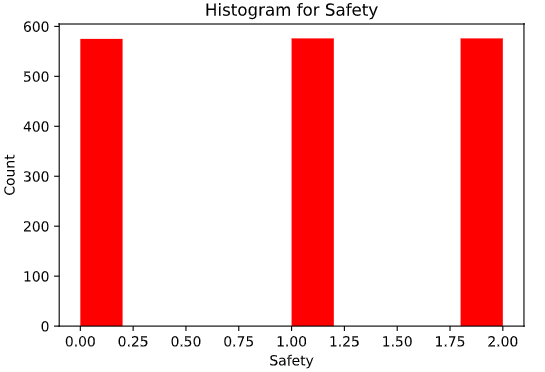
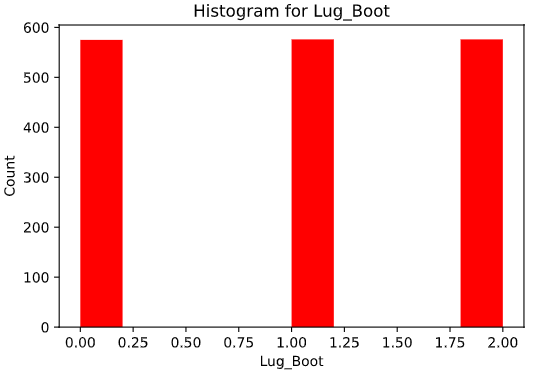
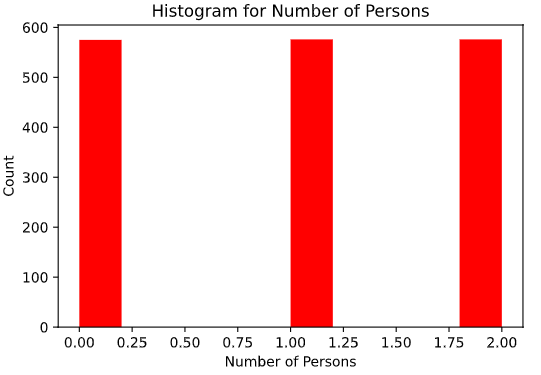
print(classification\_report(y\_pred, y\_test))



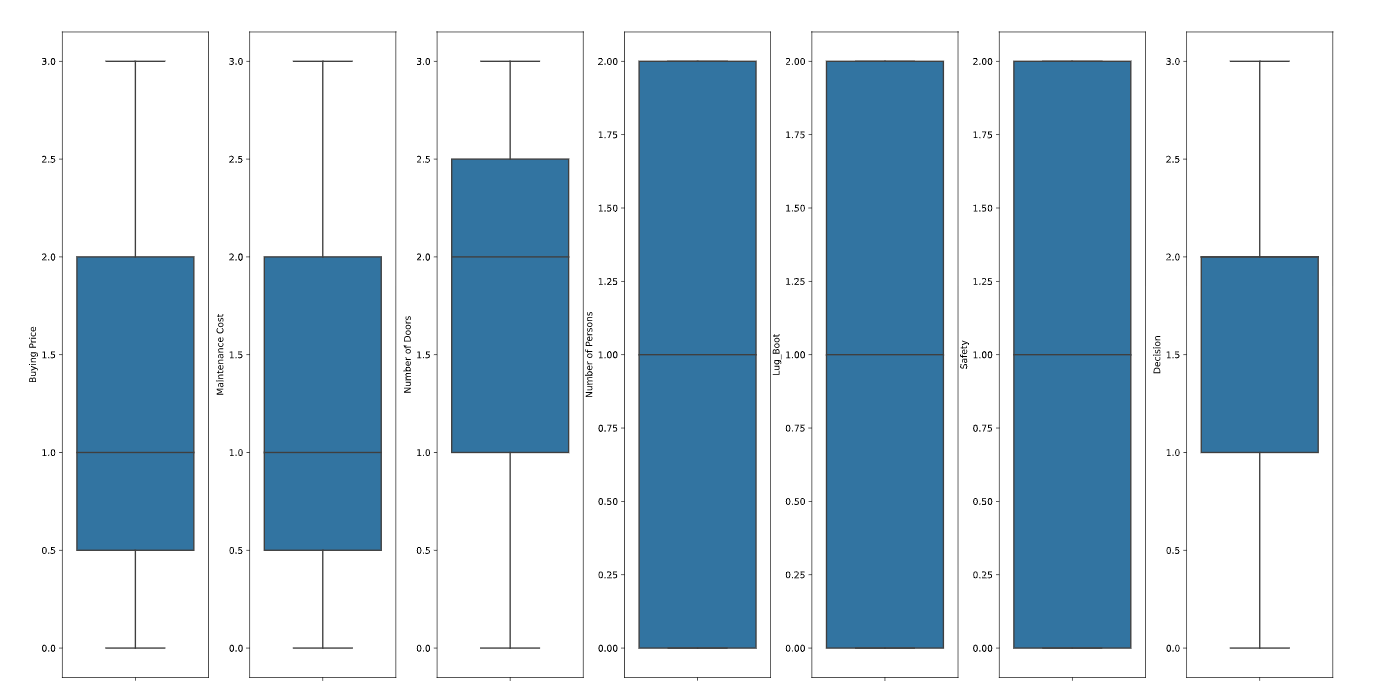
**3. DATA VISUALIZATION**

1. **Histogram Plot**

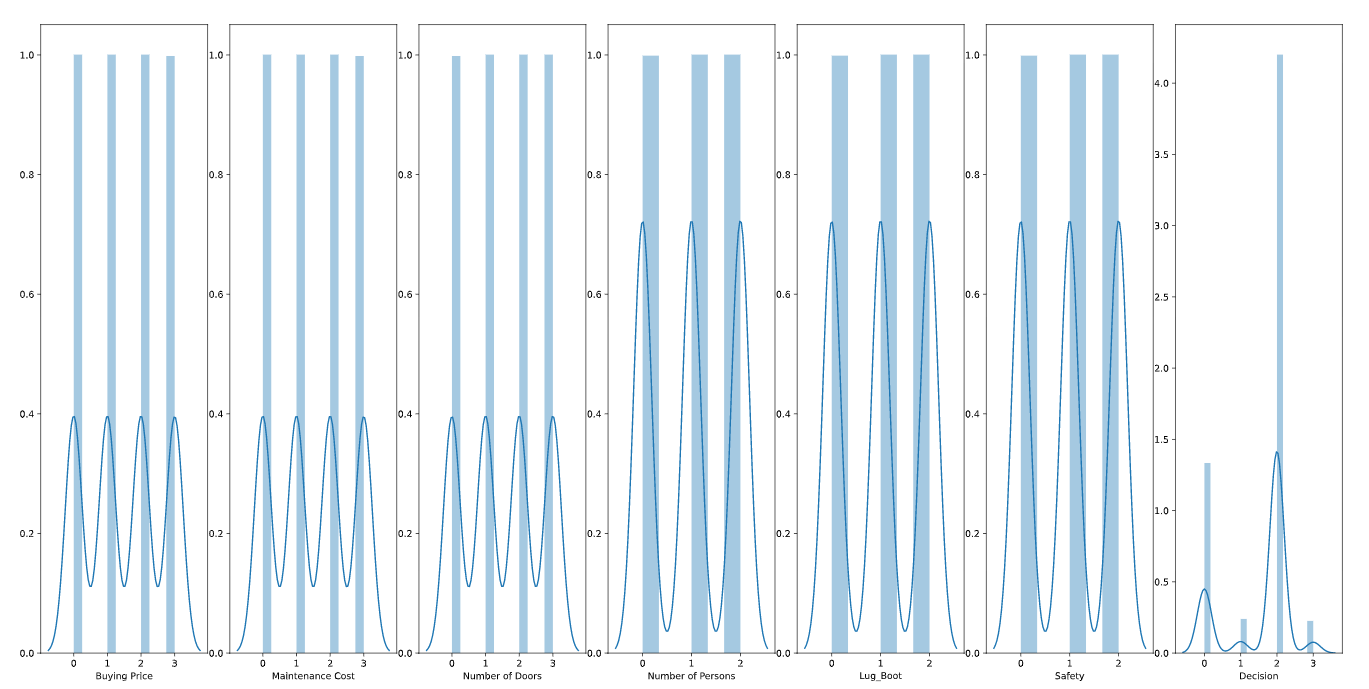




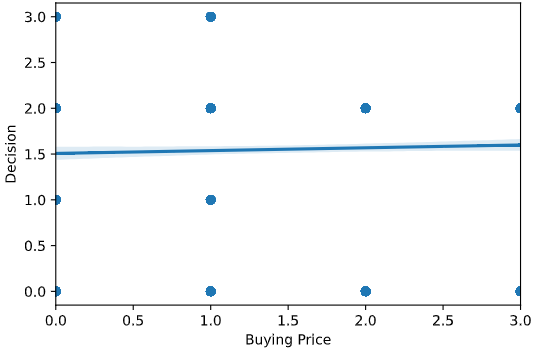
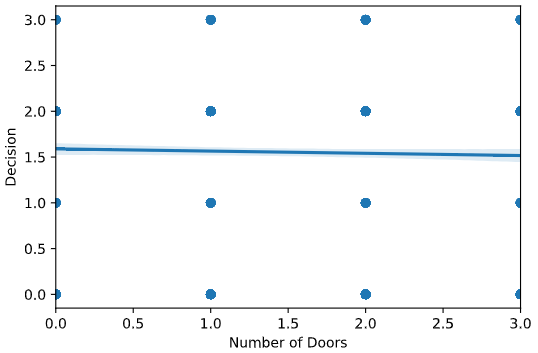
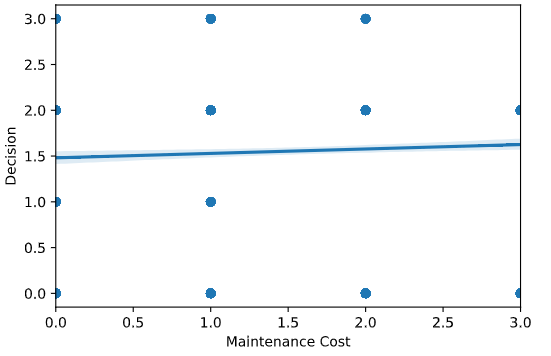
1. **Box Plot**

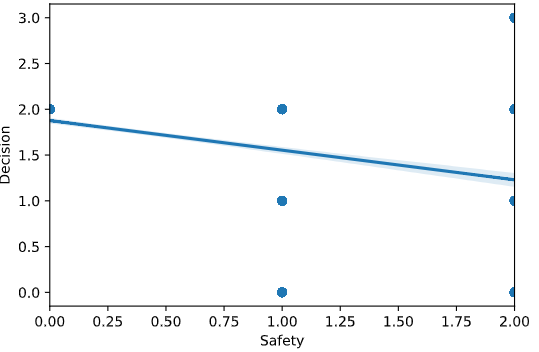
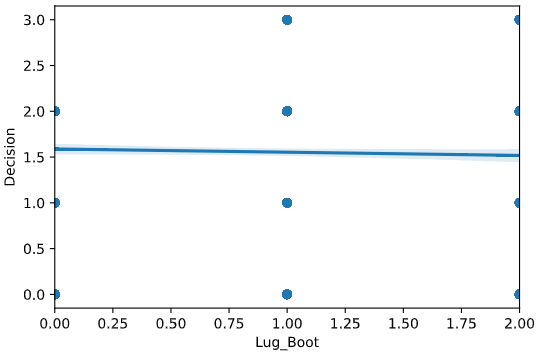
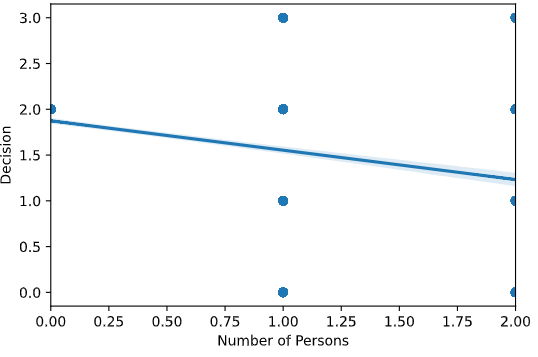
****

1. **Distributiom Plot**

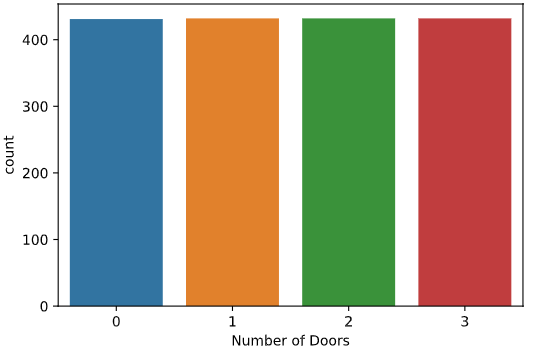
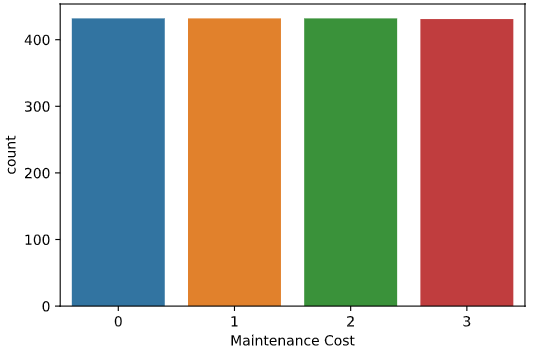
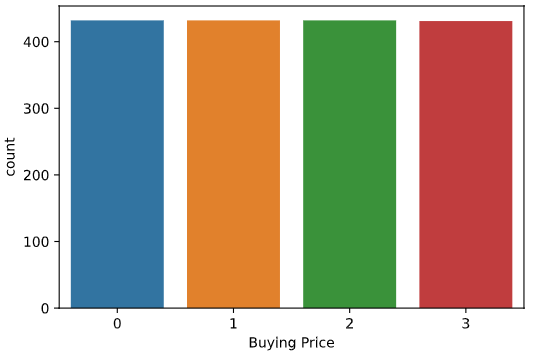
****

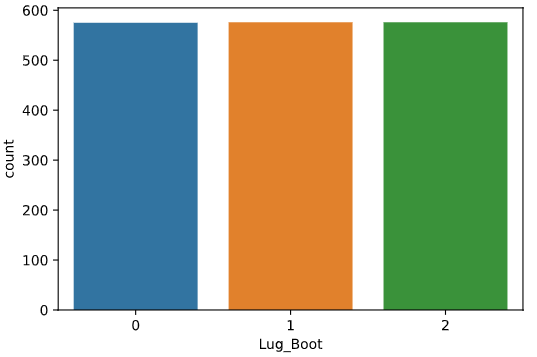
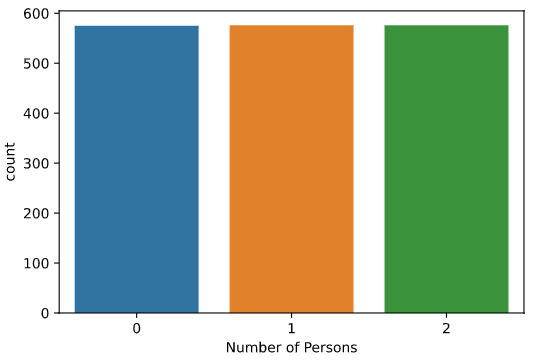
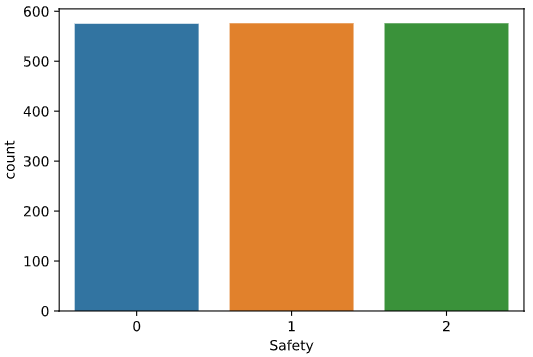
1. **Regression Plot**

** **

****

1. **Count Plot**

****

****

**4. CONCLUSION**

* The main goal of this project is to classify or to evaluate the **Car i.e. Unaccept, Accept, Good, Very Good** by which you feel safe and enjoy its good features.
* Another goal was to highlight the importance of safety on their overall quality.
* We do that by knowing different factor such as Buying price, No. of Doors, Maintenance cost, Safety etc.
* In this paper, we evaluated the different classifiers for car evaluation dataset. Based on the customer feedback about the cars used, the model is very appropriate to judge the best car segment as per the requirement of the customer.
* This project involved Two supervised learning algorithms i.e. KNN Classifier and Random Forest Classifier. Both of them have different accuracy score.

I got more accuracy using RandomForest Classifier Algorithm but it may vary depending upon the datasets.

**5. REFERENCES**

https://www.kaggle.com/

https://www.python.org/

https://anaconda.org/anaconda/python/

http://www.numpy.org/

https://matplotlib.org/

http://scikit-learn.org/

https://pandas.pydata.org/