IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 7, July 2019

AI and Business Intelligence for the Automobile Industry

Pallavi V Rao[1], Aditya Krishna M Gnaneshwar[2], Sanjana R Balagar[3], Sathiya Sudan B[4]

BE, Department of ISE, BNMIT, Bangalore, India[1]

BE, Department of CSE, BNMIT, Bangalore, India[2]

BE, Department of ISE, Jyothy Institute of Technology, Bangalore, India[3]

B. Tech, Department of IT, Panimalar Institute of Technology, Chennai, India[4]

Abstract: Data Analytics and Machine Learning are emerging as the leading technologies of 21st century. Any acquired/ given data can be analysed and conclusions drawn accordingly. The acquired or given data usually exists in its crude or raw state. Data pre-processing helps to format the data into useful form by removing redundancy and noise, eliminating missing and non-numerical values, and also by normalization. Data analysis and visualization are carried out to improve the statistical analysis of given data. Logistic regression is carried out on the data since it contains lot of columns with categorical values. Accuracy, precision, and f1 score of the model have been measured. Various conclusions can be drawn from this interdependent data set and can be stored as historical data for future analysis. Linear Regression is also carried out on the data set and r-squared values noted. R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. In this paper a data set of different car manufacturers in the automobile industry is taken and analysed. A ML model is built by employing both logistic regression and linear regression for the automobile industry. This Business Intelligence model is a boon to the manufacturers and sales department in identifying their product in the 21st century market

Keywords: Data Analytics and Machine Learning, Data pre-processing, Logistic regression, accuracy, precision, and f1 score, linear regression, categorical values, data analysis and visualization, R-squared, Business Intelligence

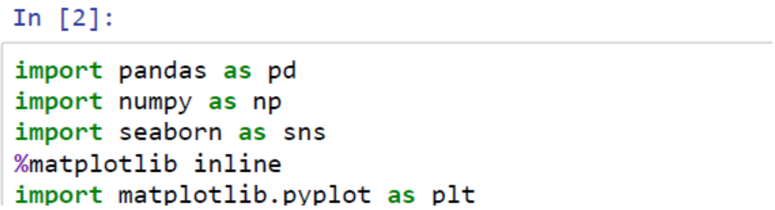
I. INTRODUCTION

We have created a dedicated marketing analytics unit with these key mandates:

1) Clearly show marketing's role in driving business outcomes such as revenue growth.
2) Do that through a "real world" lens reflecting automotive industry realities such as dealer relationships, dealer incentives and distribution constraints.
3) Deploy transparent analytics technology that challenges conventional wisdom and stands up to scientific scrutiny.
4) Develop predictive capabilities to help the company decide what to do next.

We have identified several fields in the data set

i. Engine size
ii. Engine power
iii. Sale amount
iv. Resale amount
v. Vehicle type
vi. Total price in units
vii. Manufacturer



```
In [2]:

import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
```

Figure 1 shows the Python code to import libraries.

II. METHODOLOGY

A. Importing Libraries [2]

Figure 1 shows the Python code to import libraries. We have used three libraries

- ‘numpy’ is a package for scientific computing with Python. This library is imported as ‘np’ and will be used throughout the project.
- ‘pandas’ is for data manipulation and analysis. panadas is an open source, BSD- licenced library providing easy-to-use data structures and data analysis tools. pandas is imported as pd.
- ‘matplotlib.pyplot’ is a collection of command style functions that make matplotlib work like MATLAB. It is imported as plt
- ‘seaborn’ is a Python data visualization library based on matplotlib for attractive and informative statistical graphics.

B. Importing data

Figure 2 shows the Python code to import data from respective directory/ file and assigning it to DataFrame df. The data stored in CSV format is being imported. [3] [4]

C. Checking for NaN

It is very essential in data pre-processing to check for NaN. In this attempt we could identify few NaN. Figure 3 shows the python code to check for NaN.

D. Manipulating NaN values

It is essential to remove the NaN values. This can be done by

- Removing the entire column containing many NaN values
- Forward fillna method
- Backward fillna method
- Mean method

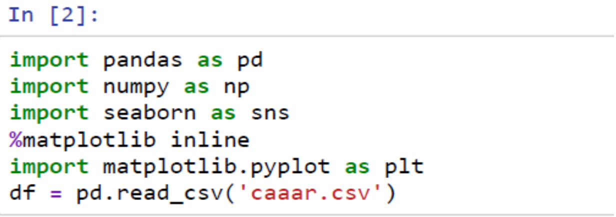
Figure 4 shows the technique of forward fillna method and figure 5 shows the method of dropping the column.

E. Plotting a Heatmap

Correlation between the fields of the recorded data is analysed by plotting a heatmap. The values may be negative or positive and the magnitude plays a key role in designing various predictive models in AI. Figure 6 shows a heatmap and correlation model.

F. Splitting the data into train and test sets. Figure 7 shows the python code to split the data set into train and test data.

G. Applying logistic regression on the split data. Figure 8 shows logistic regression on given data set.



```
In [2]:

import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
df = pd.read_csv('caaar.csv')
```

Figure 2 shows the Python code to import data and assigning it to DataFrame df

H.

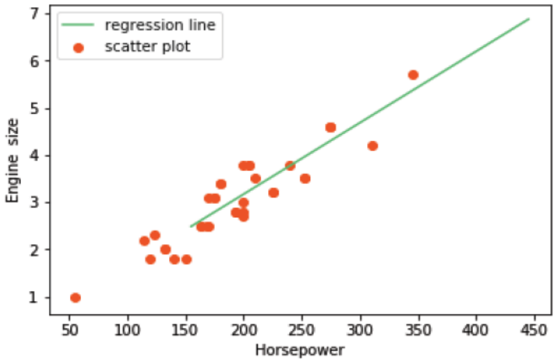
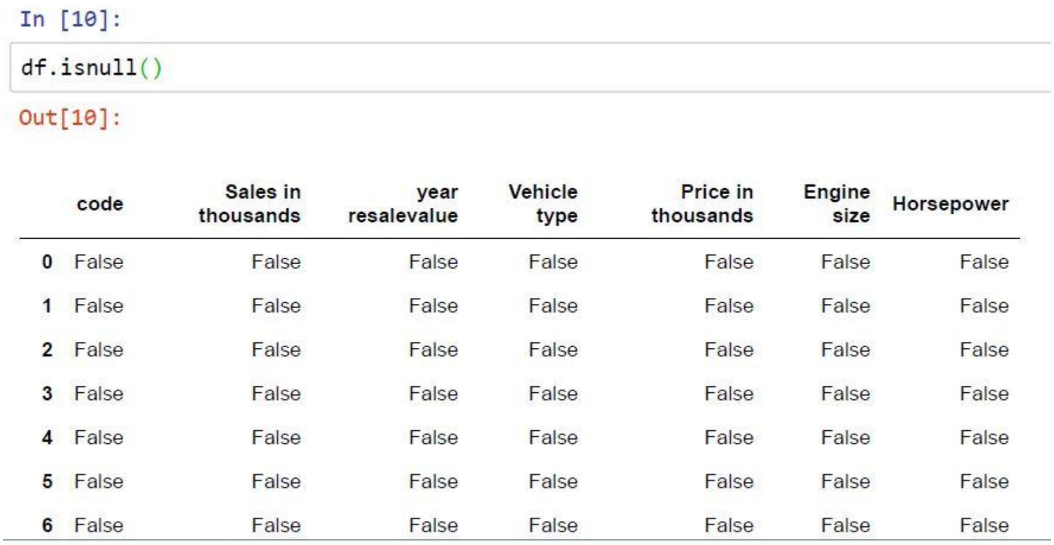


Figure 9 shows the linear regression plot

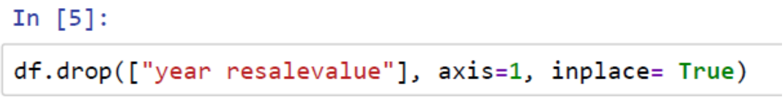
In statistics, linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables (or independent variables). [5] Linear regression is carried out on the data set. R ^{2} value or score is also measured. Figure 3 shows the linear regression plot.



```
In [10]:
df.isnull()
Out[10]:
```

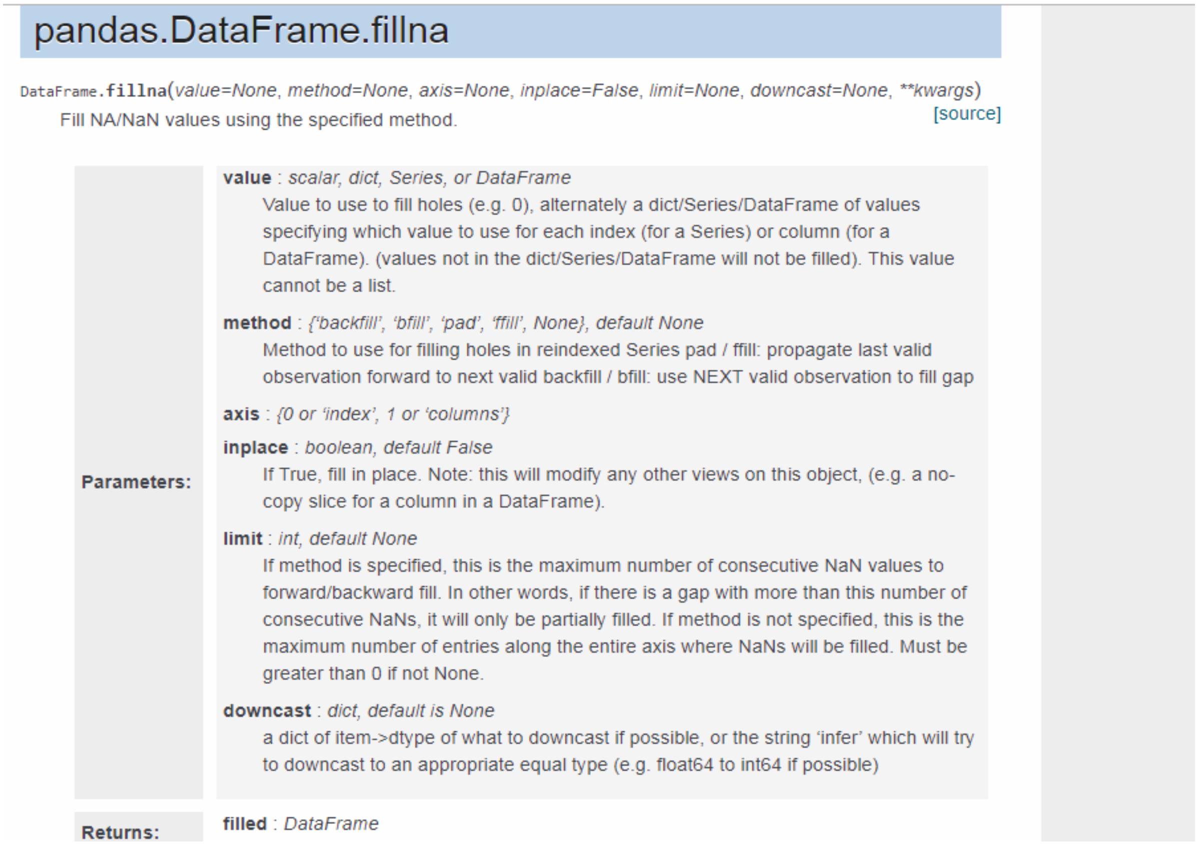
	code	Sales in thousands	year resalevalue	Vehicle type	Price in thousands	Engine size	Horsepower
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False

Figure 3 shows the Python code to check for NaN.



```
In [5]:
df.drop(["year resalevalue"], axis=1, inplace= True)
```

Figure 5 shows the method of dropping the column



pandas.DataFrame.fillna

`DataFrame.fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None, **kwargs)` [source]

Fill NA/NaN values using the specified method.

Parameters:

value : *scalar, dict, Series, or DataFrame*
Value to use to fill holes (e.g. 0), alternately a dict/Series/DataFrame of values specifying which value to use for each index (for a Series) or column (for a DataFrame). (values not in the dict/Series/DataFrame will not be filled). This value cannot be a list.

method : *{'backfill', 'bfill', 'pad', 'ffill', None}, default None*
Method to use for filling holes in reindexed Series pad / ffill: propagate last valid observation forward to next valid backfill / bfill: use NEXT valid observation to fill gap

axis : *{0 or 'index', 1 or 'columns'}*

inplace : *boolean, default False*
If True, fill in place. Note: this will modify any other views on this object, (e.g. a no-copy slice for a column in a DataFrame).

limit : *int, default None*
If method is specified, this is the maximum number of consecutive NaN values to forward/backward fill. In other words, if there is a gap with more than this number of consecutive NaNs, it will only be partially filled. If method is not specified, this is the maximum number of entries along the entire axis where NaNs will be filled. Must be greater than 0 if not None.

downcast : *dict, default is None*
a dict of item->dtype of what to downcast if possible, or the string 'infer' which will try to downcast to an appropriate equal type (e.g. float64 to int64 if possible)

Returns:

filled : *DataFrame*

Figure 4 shows the technique of forward fillna method

DOI 10.17148/IJARCCE.2019.8712

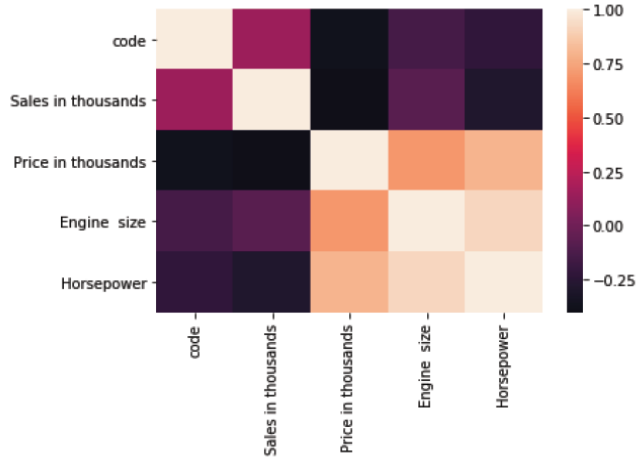
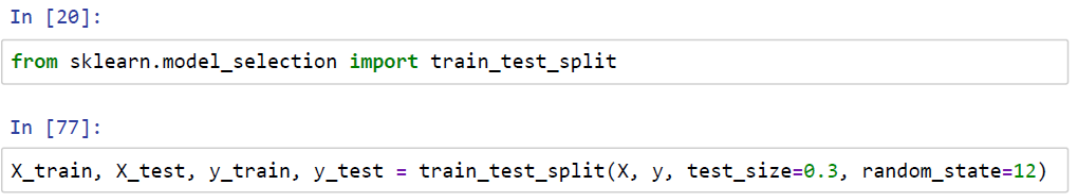


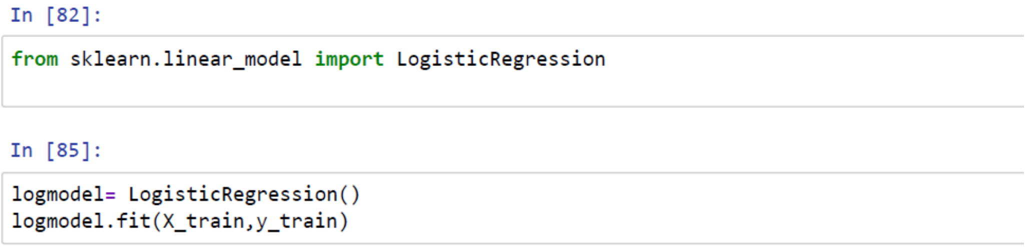
Figure 6 shows a heatmap and correlation of the model.



```
In [20]:
from sklearn.model_selection import train_test_split

In [77]:
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=12)
```

Figure 7 shows the python code to split the data set into train and test data.



```
In [82]:
from sklearn.linear_model import LogisticRegression

In [85]:
logmodel= LogisticRegression()
logmodel.fit(X_train,y_train)
```

Figure 8 shows logistic regression on given data set.

III. DATA VISUALIZATION

Data visualization is an integral part of data analytics and Machine Learning. When there is a huge data set, manual analytics becomes almost impossible. Data visualization plays a vital role in analysis in such situation. It involves use of various plots – bar graph, pie charts, box plots, line graphs and many more. Figure 10 includes a bar graph of horse power and figure 11 a plot of engine size.

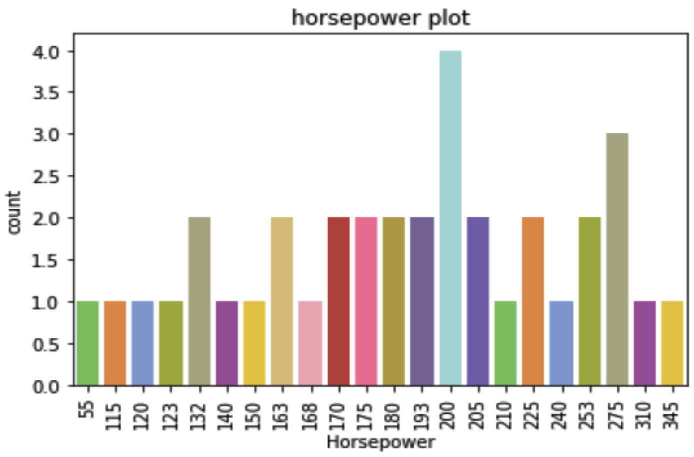


Figure 10 shows a bar graph of horsepower.

DOI 10.17148/IJARCCE.2019.8712

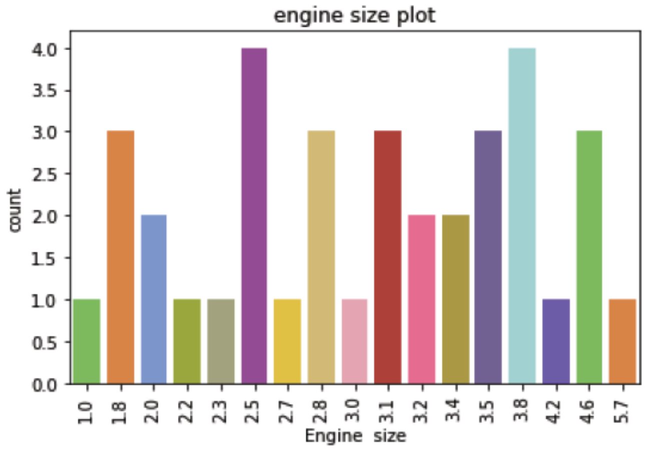
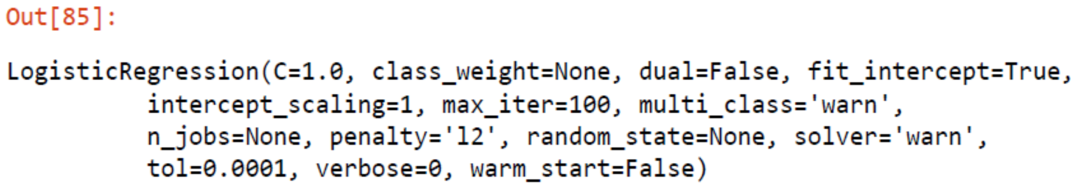


Figure 11 shows a bar graph of engine size.

IV. RESULTS

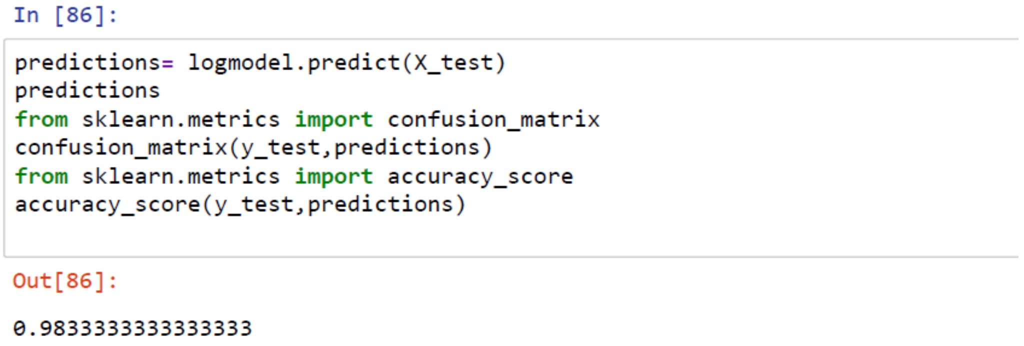
After analysing the heatmap and figuring out the correlation between different columns/ physiological parameters, Logistic regression needs to be carried out to create a prediction model. Figure 12 shows the results of logistic regression model. Figure 13 shows the Accuracy score of the designed model. From this data, precision, f1 score and reliability can be calculated. Figure 14 shows the R-squared calculation for the linear regression model. [6]



```
Out[85]:

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

Figure 12 shows the results of logistic regression model



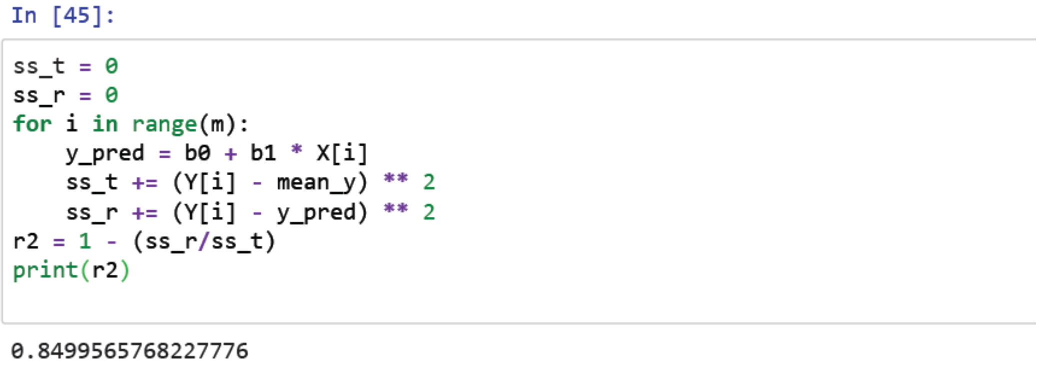
```
In [86]:

predictions= logmodel.predict(X_test)
predictions
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,predictions)
from sklearn.metrics import accuracy_score
accuracy_score(y_test,predictions)

Out[86]:

0.9833333333333333
```

Figure 13 shows the Accuracy score of the designed model.



```
In [45]:

ss_t = 0
ss_r = 0
for i in range(m):
    y_pred = b0 + b1 * X[i]
    ss_t += (Y[i] - mean_y) ** 2
    ss_r += (Y[i] - y_pred) ** 2
r2 = 1 - (ss_r/ss_t)
print(r2)

0.8499565768227776
```

Figure 14 shows the R-squared calculation.

DOI 10.17148/IJARCCE.2019.8712

V. CONCLUSION

Automobile companies proactive in business have recorded their sales data. Data analytics had to be carried out on the data –both historical and present trend to draw inference. The goal was to create or improve profit of the company and to create a visualization model using libraries like seaborn, matplotlib and data analysis using pandas. A python code was written and executed in the Jupyter platform to analyse and draw conclusions. Data pre-processing and data visualization has been carried out successfully and various conclusions drawn.). Linear regression is carried out on the data set. R ^{2} value or score is also measured. Logistic regression is carried out on the data since it contains lot of columns with categorical values. Accuracy, precision, and f1 score of the model have been measured.

REFERENCES

[1]. Interactions between kidney disease and diabetes- dangerous liaisons- Roberto Pecoits-Filho, Hugo Abensur, Carolina C.R. Betônico, Alisson Diego Machado, Erika B. Parente, Márcia Queiroz, João Eduardo Nunes Salles, Silvia Titan and Sergio Vencio- 2016- article 50.

[2]. The Python Standard Library — Python 3.7.1rc2 documentation https://docs.python.org/3/library/

[3]. Data Warehousing Architecture & Pre-Processing- Vishesh S, Manu Srinath, Akshatha C Kumar, Nandan A.S.- IJARCCE, vol6, iss5, May2017

[4]. Data Mining and Analytics: A Proactive Model - http://www.ijarcce.com/upload/2017/february-17/IJARCCE%20117.pdf

[5]. A comparative analysis on linear regression and support vector regression-DOI:10.1109/GET.2016.7916627- https://ieeexplore.ieee.org/abstract/document/7916627

[6]. Stock market predication using a linear regression- DOI: 10.1109/ICECA.2017.8212716, https://ieeexplore.ieee.org/abstract/document/8212716

DOI 10.17148/IJARCCE.2019.8712