IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 4, April 2019

Data Analytics: Car Sales in a Calendar Year

Vishesh S[1], Pavan Kumar C K[2]

BE, Department of TCE, VTU, Belgaum, India[1]

BE, Department of CSE, Adichunchanagiri Institute of Technology, India[2]

Abstract: The manipulation of raw or crude data and drawing conclusions out of it is called data analytics. In this paper we will be analysing car sales data of a particular dealer in a calendar year. Various car models have been sold to customers in the market by this dealer. We are interested in analysing this recorded data and various aspects profit making. The data is stored in .csv format and includes various fields which may or may not be dependent on each other. Real world data are generally noisy, inconsistent, contains many errors and incomplete. Proper manipulation of these factors improves the quality of analysis and prediction. The focus of data analytics lies in inference, the process of deriving conclusions with the help of graphs, statistics and many other non-statistical tools. In this paper we have carried out data analysis in steps yielding the best result exploiting various data science libraries and the code is written in Python.

Keywords: Manipulation of Raw or Crude Data and Drawing Conclusions - Data Analytics, Car Sales, Graphs, Data Pre-Processing, Statistics and Many Other Non-Statistical Tools

I. INTRODUCTION

21st century is the era of information, big data and AI. Large volumes of data are exchanged and conditioned. The volume of data that one has to deal with has exploded to unimaginable levels. Most of the data exists in its crude form and needs to be converted to useful format before analysis. This process of converting raw data into useful format is called data pre-processing. Real world data is [1]

- Incomplete: consists of missing attribute values or consists of only aggregate data.
- Noisy: containing errors or outliers.
- Inconsistent: containing discrepancies in code.
- Redundant

II. PROBLEM STATEMENT

In this paper we consider a car dealer doing business in this 21st century world. This company has sold many cars in a calendar year. The sales data has been recorded and needs to be analysed in all aspects to increase the profit of the company. The company is existent in business from a very long period of time and undergoing stiff competition from other companies. The company's historical data is well tabulated and records maintained. The company wants to understand the present trend in the market in terms of various aspects of the product they sell like- resale price, horsepower, engine size, type of vehicle preferred and many others. These data may be used to predict the sales of the next calendar year and to understand the taste of the consumer.

```
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.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

III. 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. Figure 3 shows the Python code to check for NaN. In this attempt we could identify few 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.

```
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

```
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.

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 4, April 2019

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)`

Fill NA/NaN values using the specified method. [source]

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

Out[22]:

	code	Sales in thousands	year resalevalue	Vehicle type	Price in thousands	Engine size	Horsepower
code	1.000000	0.147072	-0.488276	0.122035	-0.378106	-0.144061	-0.222043
Sales in thousands	0.147072	1.000000	-0.400124	0.113588	-0.400925	-0.080030	-0.286395
year resalevalue	-0.488276	-0.400124	1.000000	-0.360114	0.864542	0.537485	0.648351
Vehicle type	0.122035	0.113588	-0.360114	1.000000	-0.344820	-0.230505	-0.242171
Price in thousands	-0.378106	-0.400925	0.864542	-0.344820	1.000000	0.702371	0.797861
Engine size	-0.144061	-0.080030	0.537485	-0.230505	0.702371	1.000000	0.921931
Horsepower	-0.222043	-0.286395	0.648351	-0.242171	0.797861	0.921931	1.000000

DOI 10.17148/IJARCCE.2019.8439

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

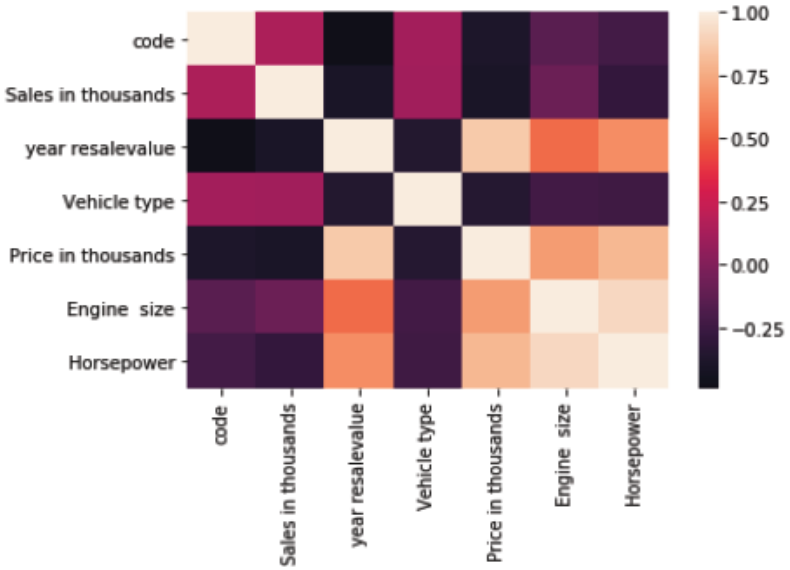

Figure 6 shows a heatmap and correlation model.

IV. RESULTS

Figure 7 shows the plot of horsepower using the seaborn library. Figure 8 represents a graph of engine size. Figure 9 represents a subplot of horsepowe vs count and hue = Vehicle type. The graphical results obtained arecrucial in analysing the sales and the taste of the consumer. Later these statistics and figures can be used to develop a predictive model.

```
In [14]:
color_types = ['#78C850','#F08030','#6890F0','#A8B820','#A8A878','#A040A0','#F8D030']
# Count Plot (a.k.a. Bar Plot)
sns.countplot(x='Horsepower', data=df, palette=color_types).set_title('horsepower plot')
plt.xticks(rotation=-90)
```

```
Out[14]:

(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21]), <a list of 22 Text xticklabel objects>)
```

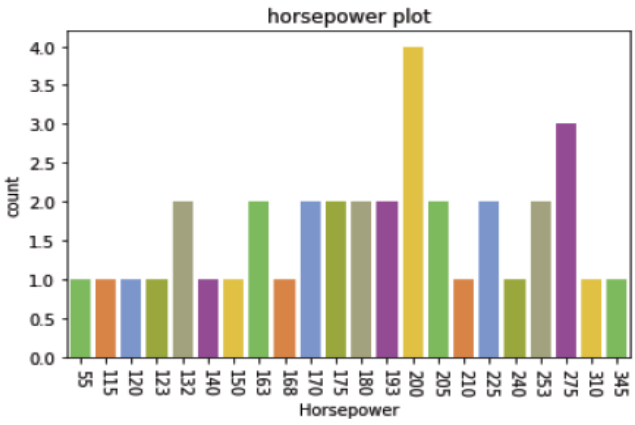

Figure 7 shows the plot of horsepower using the seaborn library

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 4, April 2019

In [15]:

```
color_types = ['#78C850','#F08030','#6890F0','#A8B820','#A8A878','#A040A0','#F8D030']
# Count Plot (a.k.a. Bar Plot)
sns.countplot(y='Engine  size', data=df, palette=color_types).set_title('  Engine Size  ')
plt.xticks(rotation=-90)
```

Out[15]:

```
(array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5]),
 <a list of 10 Text xticklabel objects>)
```

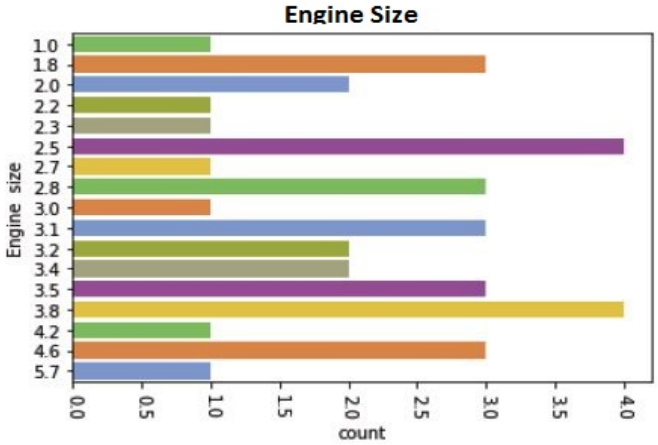

Figure 8 represents a graph of engine size

In [21]:

```
f= plt.subplots(figsize=(12, 10))
sns.countplot(x="Horsepower", hue='Vehicle type', data=df).set_title('');
```

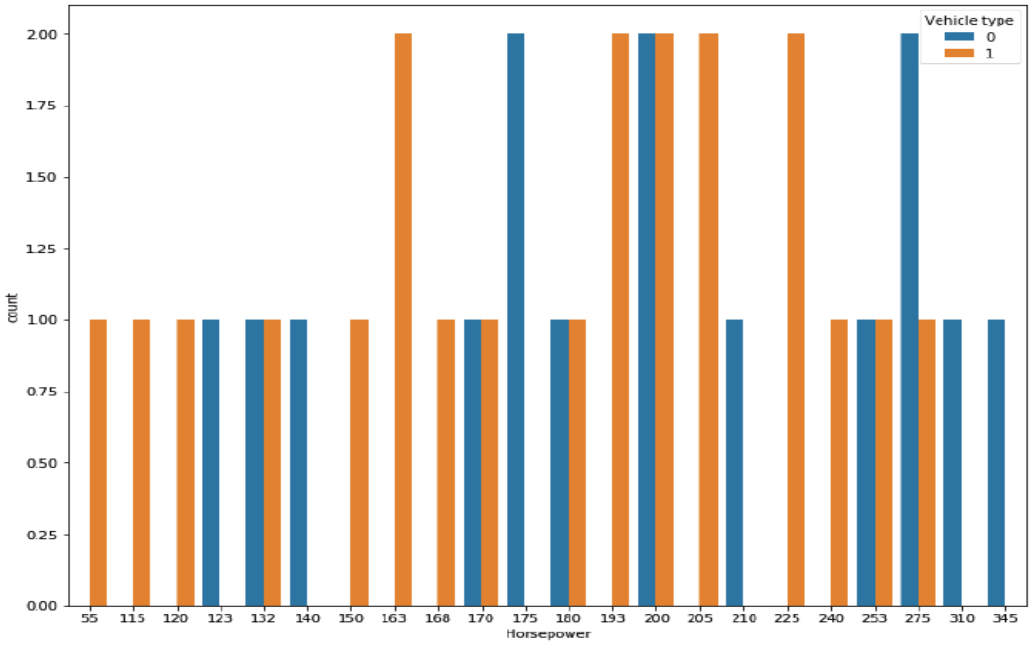

Figure 9 represents a subplot of horsepowe vs count and hue = Vehicle type.

DOI 10.17148/IJARCCE.2019.8439

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 4, April 2019

V. CONCLUSION

A car selling company proactive in business in this 21st century world had recorded its 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.

REFERENCES

[1]. Principles of data mining DJ Hand - Drug safety, 2007 - Springer
[2]. The Python Standard Library — Python 3.7.1rc2 documentation https://docs.python.org/3/library/
[3]. Data Warehousing Architecture and Pre-Processing- Vishesh S, Manu Srinath, Akshatha C Kumar, Nandan A.S.- IJARCCE, vol 6, issue 5, May 2017.
[4]. Data Mining and Analytics: A Proactive Model - http://www.ijarcce.com/upload/2017/february- 17/IJARCCE%2017.pdf

BIOGRAPHY

VISHESH S born on 13th June 1992, hails from Bangalore (Karnataka) and has completed B.E in Telecommunication Engineering from VTU, Belgaum, Karnataka in 2015. He also worked as an intern under Dr. Shivananju BN, former Research Scholar, Department of Instrumentation, IISc, Bangalore. His research interests include Embedded Systems, Wireless Communication, BAN and Medical Electronics. He is also the Founder and Managing Director of the corporate company Konigtronics Private Limited. He has guided over a hundred students/interns/professionals in their research work and projects. He is also the co-author of many International Research Papers. He is currently pursuing his MBA in e-Business and PG Diploma in International Business. Presently Konigtronics Private Limited has extended its services in the field of Software Engineering and Webpage Designing. Konigtronics also conducts technical and non-technical workshops on various topics.