Data Visualization on Volkswagen's car scandal in 2015

Declaration on Plagiarism

This form must be filled in and completed by the student submitting an assignment

Name:	Bibek Prasad Gupta
Student Number:	20210617
Programme:	MSc in Computing
Module Code:	CA682
Assignment Title:	Data Visualisation
Submission Date:	18th Dec 2020
Module Coordinator:	Dr Suzanne Little

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the referencing guidelines found at http://www.dcu.ie/info/regulations/plagiarism.shtml, https://www4.dcu.ie/students/az/plagiarism and/or recommended in the assignment guidelines

Name: Bibek Prasad Gupta Date: 18th December 2020

Abstract

This data visualization is based on the 2015 Volkswagen “Defeat device” scandal. A software system was installed in their cars to deceive the pollutant emission testing system but the cars were emitting more than 40% pollutants as compared to the allowed value. We will see the effect of this scandal through data visualization and also compare the results with other car manufactures. I have written a case study on this scandal for the CA640 module assignment. It was an interesting case study so got motivated to create a visualization of it.

In my view, a significant decrease in usage of Volkswagen’s cars can be seen in the year 2015 and the growth of other car manufacturers in the USA car market.

Data Collection

The biggest challenge was to find a dataset that is similar to my problem and also has some aspects of Big data but after a lot of searches, I found it in Kaggle. The size of the data is 9.29 GB and it has approx 3 million records. The number columns in the dataset are 66 and it is in the CSV format. The dataset has the volume aspect of Big data, considering my system’s processing power having configuration i7 5th generation processor and 8 GB of RAM. The dataset is structured so it does not contain the variety aspect and it neither has the velocity aspect as data is downloaded and processed within the system.

Data Exploration

I used Kaggle's inbuilt interface to explore the data. There were 66 columns in my dataset, therefore, I filtered them as per my problem statement requirement and narrowed done it to 3 columns i.e. vin, make_name and year. I got an overview of the data from the Kaggle interface for these 3 columns. vin column stored vehicle identification number so it is a nominal variable. make_name is also nominal variable as it contains car manufacturing company names. Ford and Chevrolet occupy 16% and 12% of the data respectively. Lastly, year column stores year-wise data starting from 1915 to 2021 therefore it is an ordinal variable. The Kaggle interface figure used for data exploration is given below for reference.

vin	make_name		year
3000000 unique values	Ford	16%	
	Chevrolet	13%	
	Other (2146812)	72%	
ZACNJABB5KPJ92081	Jeep		2019
SALCJ2FX1LH858117	Land Rover		2020
JF1VA2M67G9829723	Subaru		2016
SALRR2RV0L2433391	Land Rover		2020
SALCJ2FXXLH862327	Land Rover		2020
SALYK2EX1LA261711	Land Rover		2020
3MZBPABL6KM107908	Mazda		2019
SALYK2EX5LA275434	Land Rover		2020

Data Cleaning and Preprocessing

I started the cleaning part with Open refine but due to high volume Open refine crashed so I divided the data into small chunks using pandas and generated CSV files. I picked up the first generated CSV file and cleaned it using Open refine but I felt that cleaning each file manually is not a good approach and started looking for alternatives. I came across “Dask” library and tried to process my dataset and it worked for me. The operations performed during cleaning the data is stated below –

- Remove duplicate records based vin column.
- Check for null values but no null values were there.
- Parsed the year column from text to integer.

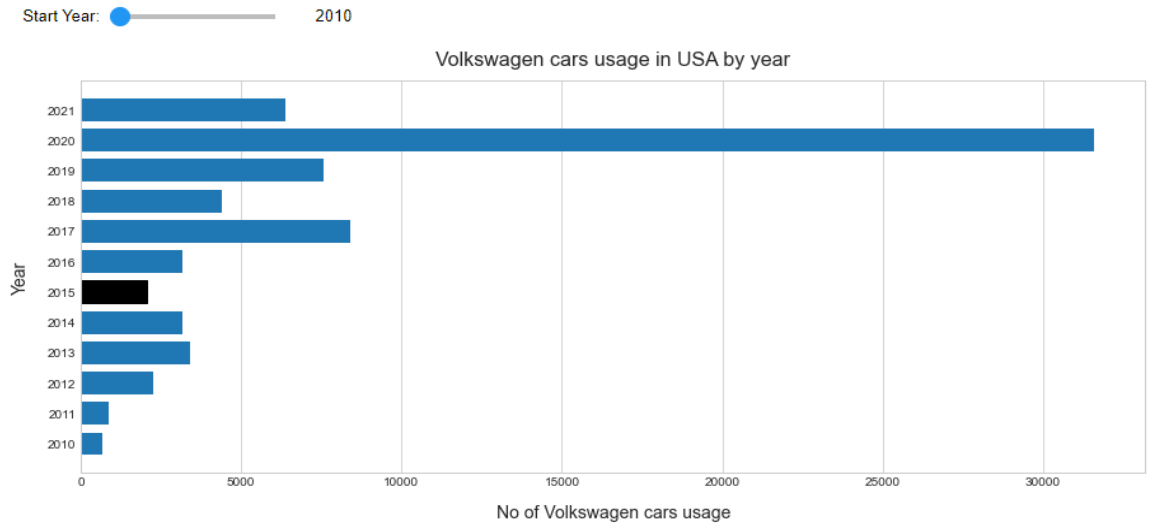
Dataset is filtered out to contain only the necessary data. The operations performed to prepare the data for visualization are listed below –

- Drop the columns other than vin, make_name and year
- The data is present from 1915 to 2021 but we will visualize the data from 2010 onwards so data before 2021 is filtered out.

I have taken the data from 2010 as the scandal happened in 2015, therefore, our visualization will cover information about Volkswagen’s car usage before the scandal and also the effect of its car usages post the scandal. Only year and manufacturer name are required to create the visualization so selected the make_name and year columns only.

Visualization

The first chart is a horizontal bar chart that describes Volkswagen’s car usage by year.



Choice of chart type

My goal was to compare Volkswagen's car usage each year starting from 2010 so the only column required was the year. It is an ordinal variable so a bar chart is a good candidate to visualize this data.

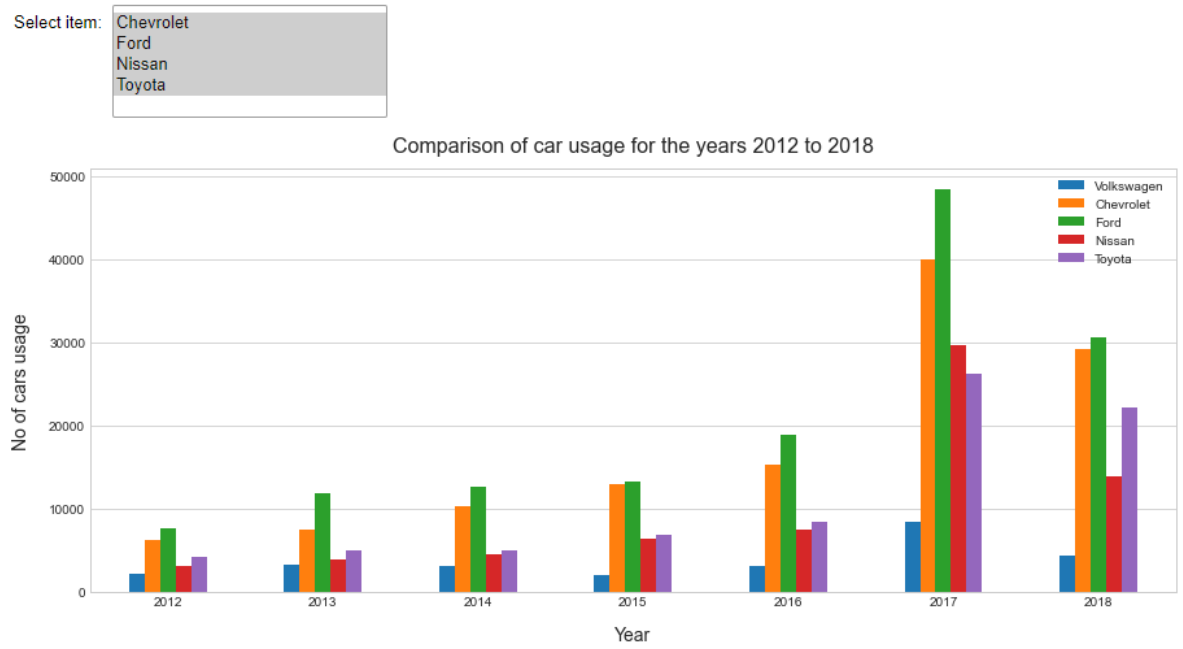
Design choices

I started with a bar chart but I found the graph difficult to read as x-axis was displaying the year values and they looked congested. Therefore, I changed the orientation and made it as a horizontal bar chart and it looked better as compared to the earlier bar chart. Followed the 'less is more' principle and tried to use fewer items as possible. Used only two colors in the chart to make the comparison easy. The bar representing the 2015 year is marked with blue color to show the decline of Volkswagen's car usage.

Interactivity

A slider is given to interact with the chart and it can be used for comparing as it changes the start year.

The second chart is a grouped bar chart that compares the cars usage of Volkswagen against the top 4 car manufacturers for the years 2012 to 2018.



Choice of chart type

My target on this graph was top make the comparisons with top performers in the USA car market so I queried the top 4 performers and it is a nominal variable. The other input required was the comparison period and it was 2012 to 2018, therefore, is an ordinal variable.
The line chart and Grouped Bar are suitable to deal with this type of variable.

Design choices

I started with Line chart but I was not satisfied with graph as to read the chart lot of effort was required. Therefore, I plotted the data as Grouped Bar chart and it was easier to read as compared to the line chart.

Interactivity

Multiple Selector widget is used to interact with the graph. Comparison is done based on selected items by the user.

Tools and libraries

Python, pandas, seaborn and matplotlib is used to clean and create the visualization.

Ipywidgets library is used for adding interactivity. I have added two widgets to interacts with the graph. They are IntSlider and SelectMultiple. IntSlider allows us to change the year via a slider whereas SelectMultiple is used to select multiple manufacturers for comparison.

Conclusion

The most challenging part for me was to find a dataset which is similar to my problem. Handling the data was also difficult but “Dask” library made it simpler. Data Cleaning and Preprocessing was easy as I had to deal with just 3 columns. I spent some time in learning Ipywidgets. In my view, the first graph explains Volkswagen’s fall down in 2015 and great growth in 2020. On the other hand, the second graph shows the growth of other car manufacturers as compared to Volkswagen. I have used “seaborn-whitegrid” style rather than explicitly selected colors and font style so it is one of the required improvements. I will discuss the other possible improvements below.

In the first chart the things that can be improved are stated below -

- I tried to remove the x-axis to display the value for each year on top of the bar but could not managed to do that. It would have been better to see the count directly on the bars rather than looking at the axis again for it.
- Widget to select the end year to interact with the graph.

In the scond chart the things that can be improved are stated below –

- It is hard to memorize the colors to compare the performance. So, I tried to add the manufacturer names on the bar itself and removing the legends but could not managed to do so. The multiple select widget reduces the pain of remembering colors to some extent.
- Year range is fixed so the user can only see the data for 2012 to 2018 therefore a date picker widget will be a great enhancement.

References

Dataset was taken from here.

Referred this article for creating the bar chart.

Referred this tutorial series to learn matplotlib library.

Referred this article for learning ipywidgets. Also looked into the documentation of ipywidgets to get the widget syntaxes.