

Author: Justin Bernstein

I pledge my honor that I have abided by the Stevens Honor System

Problem:

Buying a car is tough. There are so many options and so many different websites to try and help you find a car. These websites are good enough for the general consumer but fall flat for the gear-heads out there. Additionally, most car search sites are trying to push the user into a buying a car. General consumers mostly care about basics such as the make, model, year, price, and maybe MPG, so that's what current car search websites display front and center. They bury the technical stuff so deep that it's difficult to find and even search by. This leaves those with experience with cars at a loss and frustrated as they try to navigate consumer car search sites that prioritize the basics.

Solution:

The solution is to provide a highly detailed car search system that can be used by everyone. This system lets the user search by various metrics and criteria and then returns all cars that match the query. It goes beyond just the make, model, year, msrp, and mpg and shows the user information about such things as fuel type, horsepower, number of cylinders, transmission type, drive type, number of doors, size, and style. Also, it doesn't push the user towards buying a car because it's not a car sale application. Due to time constraints I was not able to add these features however I am looking to do so in the future: I wanted to add Machine Learning(probably Naive Bayes) to have the program predict information about a car that is not in the current dataset. Additionally, I would like to add a web interface with Django or Flask so that the user experience is better and easier to use.

Demo:

The current iteration of this search system is a Command Line Interface. Using the Pandas, Matplotlib, and Seaborn libraries, the program reads in a csv file containing all the information about the cars available. The rows and columns from the csv files are saved as a Pandas DataFrame. From there a menu is created in the command line using a while loop. The menu allows the user to control how they search for a car. The menu is designed to make every action explicit and easy to understand, there is no ambiguity. The user is asked for an input to navigate through the program. When it is necessary to retrieve data from the csv in the first three options of the menu, the DataFrame object is indexed through basic indexing, compound indexing, and conditional indexing which shows objects from the DataFrame which satisfy the conditions. In the fourth option of the menu, the Pandas and Matplotlib libraries are used to display various information about the dataset. The Pandas 'describe' method generates a table of statistics about the dataset while Matplotlib is used to generate graphs to give a visual representation of information about the cars in the dataset.

As something extra, I built this within a Docker cloud container so that dependencies are taken care of. Instructions to run this with Docker are in the Readme.txt. It is possible to run this program without Docker as well.