

Vroom Data Engineering Challenge Summary

David Hey

There were a few issues I ran into along the way.

1. It does not seem that “condition” and “zip” parameters needed for the certified used price estimate were available from the VIN Decoding API response. I therefore ended up using static, arbitrary values for these two parameters (condition = clean, and zip = 90019), but provided the rest of the parameters from the actual data found via the API or provided when the script was ran. To get accurate measurements I would want to get actual condition and zip code data for these VIN’s (perhaps via another source)
2. Initially, I had condition = average and zip = 27514, however I found this was giving me an estimate value of 0.0 for the certified used price. This issue was resolved when I changed the zip and condition values to clean and 90019 respectively.
3. When pulling data from the API with the CSV file provided, it appears that there was no data found for 5 of the VINs provided. I added error checking based on the HTTP response codes to ensure that the script would handle these responses gracefully, and proceed to the next line. Provided more time, I would do further investigation into why these VINs were not found, perhaps there was a small typo when the data was entered.
4. It is not so much an issue as it is a limitation, but at this point the script assumes that the CSV file is in the same directory as the python script. If we were to run the job with CRON, I would want to clean up the way that the CSV is read.
5. I was not 100% sure which field body type was, so I decided to use the sub model body field (i.e. “Sedan”) as it seemed the most descriptive (better than “car”, which was the primary body type field within categories). Generally, I would work with a stakeholder to ensure the data I am pulling is appropriate.

All that being said, the script can be run in one of two ways:

Option 1 - VIN and Mileage:

```
python vroom_sql.py --vim 4T1BK1EB6DU056165 --mileage 120000
```

Option 2 - a CSV File:

```
python vroom_sql.py --csv vroom-inventory.csv
```

Help is available by using -h or --help:

```
python vroom_sql.py -h
```

The database itself is very straightforward, with only one table (“edmunds”), and can be found in “vroom.db”.