**Old Cars Data**

A Case Study in Data Cleaning and Provenance

Rahul Varma
Dept. of Computer Science
University of Illinois-UC
Champaign, IL, USA
Rvarma4@illinois.edu

Hemendra Choudhary
Dept. of Computer Science
University of Illinois-UC
Champaign, IL, USA
hsc4@illinois.edu

Final Project Report

**Abstract**

**Data Preparation** - Data cleansing is the process of spotting and rectifying inaccurate or corrupt data from a database. The process is mainly used in databases where incorrect, incomplete, inaccurate or irrelevant part of the data are identified and then modified, replaced or deleted. Business enterprises largely rely on data whether it is the integrity of customers' addresses or ensuring accurate invoices are emailed or posted to the recipients. To ensure that the customer data is used in the most productive and meaningful manner that can increase the intrinsic value of the brand, business enterprises must give importance to data quality.
.

## 1 INTRODUCTION

This report summarizes our experience with an end- to-end data preparation work-flow; in practice of data Cleaning and Provenance establishment. We use tools and techniques introduced in CS598 Theory and Practice of Data Cleaning with a real-world dataset and document the whole work-flow along with findings. Tools used include OpenRefine, SQLite and YesWorkFlow.

## 2 DATASET OVERVIEW AND INITIAL ASSESSMENT

### 2.1 The Dataset

The data was scraped from several websites in Czech Republic and Germany over a period of more than a year. Originally, I wanted to build a model for estimating whether a car is a good buy or a bad buy based on the posting. But I was unable to create a model I could be satisfied with and now have no use for this data. I'm a great believer in open data, so here goes.

### 2.2 Content

The scrapers were tuned slowly over the course of the year and some of the sources were completely unstructured, so as a result the data is dirty, there are missing values and some values are very obviously wrong (e.g. phone numbers scraped as mileage etc.)

There are roughly 3,5 Million rows and the following columns:

1. maker - normalized all lowercase
2. model - normalized all lowercase
3. mileage - in KM
4. manufacture_year

5. engine_displacement - in ccm
6. engine_power - in kW
7. body_type - almost never present, but I scraped only personal cars, no motorcycles or utility vehicles
8. color_slug - also almost never present
9. stk_year - year of the last emission control
10. transmission - automatic or manual
11. door_count
12. seat_count
13. fuel_type - gasoline, diesel, cng, lpg, electric
14. date_created - when the ad was scraped
15. date_last_seen - when the ad was last seen. Our policy was to remove all ads older than 60 days
16. price_eur - list price converted to EUR

**2.3 Potential use-cases of cleaned Data**

a) Which factors determine the price of a car?
b) With what accuracy can the price be predicted?
c) Can a model trained on all cars be used to accurately predict prices of models with only a few samples?

**3 DATA CLEANING WITH OPENREFINE**

1. TRIM and COLLAPSE WHITE SPACES. It is very common to see unnecessary white spaces in datasets. A lot of times white spaces are hidden at the beginning or the end of a string, and sometimes they are hidden as two consecutive white spaces in a phrase. Here's what you can do to help clean up white spaces.
- Trim all the leading and trailing white spaces in ALL columns that are texts (strings). This includes the maker, model, body_type, transmission columns.
- Collapse consecutive white spaces in ALL columns that are texts (strings).

2. NUMBER. Incorrect data types are almost always the second thing you inspect in a dataset. Usually numeric data will be seen (or converted to) as text data in a lot of platforms. To correct these, you can do the following:
- Transform all columns that should be in numeric form to number
- Note that whatever you have converted to number will be shown in green

3. Fix uppercase/lowercase Upper case on columns (e.g. maker id.)

4. Convert the date related columns into datetime instead of string.

5. Check for clustering and see if better results can be achieved.

6. DELETE IRRELEVANT COLUMN. we noticed that there are almost no values on the color_slug column, and we decided that this is an irrelevant column for further analysis.

7. Using power of facets- find crazy mileage (e.g.> 200k) and create a column which points if the mileage is crazy high

8. Similarly create a column for crazy_new if the mileage is less that a threshold says 100 km.

9. Rectify the manufacturing date of outliers (e.g. made in 1300) and use averaging techniques to predict mfg. date.

10. Backfill missing data on engine transmission type (automatic / manual)

11. Remove all rows where maker was blank.


**3 SQL**

Using SQLite Browser, we first imported CSV file into Database and then perform integrity constrains check.

**3.1 The schema**

After cleaning data on open refine, we create database in SQLite db. and generate table using following command.

```
CREATE TABLE "CarSells" (
   "maker" TEXT,
   "model" TEXT,
   "mileage"   INTEGER,
   "crazy_new" TEXT,
   "crazy_run" TEXT,
   "manufacture_year"  INTEGER,
   "engine_displacement"   INTEGER,
   "engine_power"  INTEGER,
   "body_type" TEXT,
   "stk_year"  NUMERIC,
   "transmission"  INTEGER,
   "door_count"    INTEGER,
   "seat_count"    INTEGER,
   "fuel_type" TEXT,
   "date_created"  TEXT,
   "date_last_seen"    TEXT,
   "price_eur" REAL
)
```

**3.2 Integrity constrains check**

Once data is loaded to SQLite, we perform integrity check on following items -

- Check where maker is not defined (Check null value)
  /* Check if marker is NULL */
  SELECT * from CarSells where maker is NULL;

- Maker and Model both columns value is null.

  /* CHECK where both model and maker is null */
  SELECT * from CarSells where maker is NULL and model is NULL;

- Mileage and Engine power is not negative.
  /* Check mileage is not negative */
  SELECT * FROM CarSells where mileage < 0;

- Door count value should be in between 0, 6.
  /* Check if door_count is not in between 0, 6 */
  SELECT * from CarSells where NOT ( door_count > 0 AND door_count < 6);

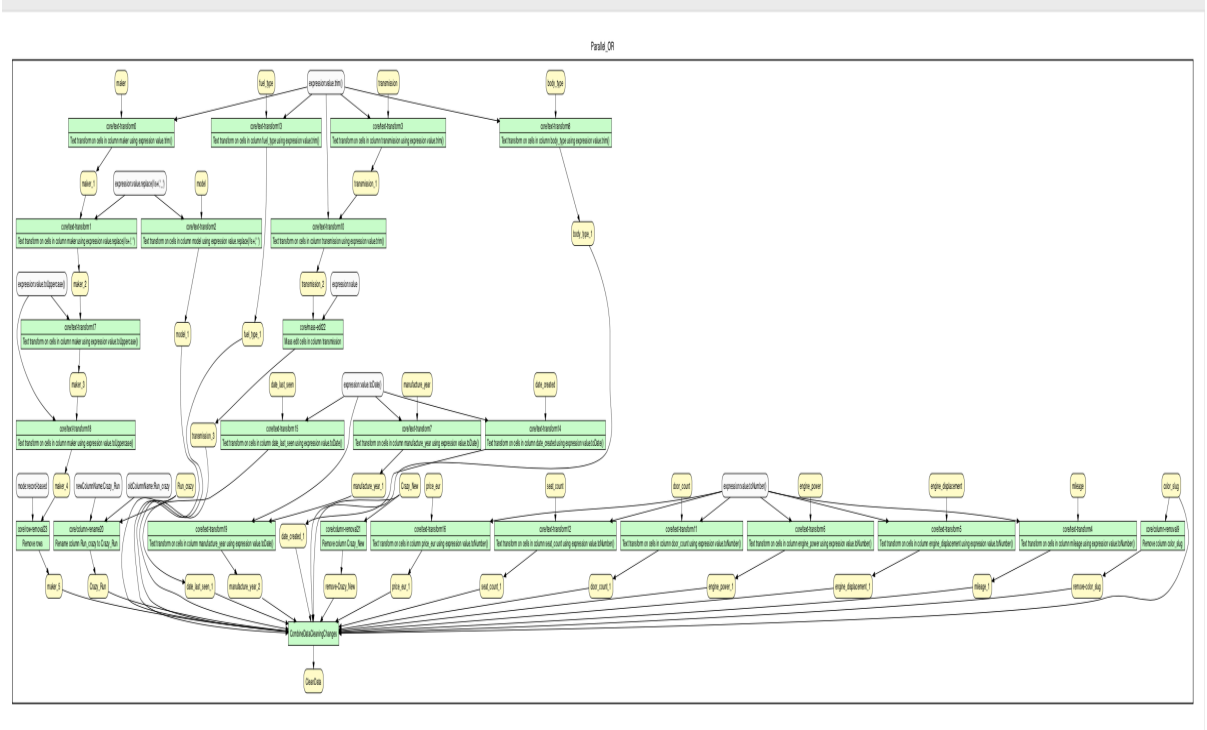
- Seat Count value should be in between 0, 10.
  /* Check if seat_count is not in between 0, 10 */
  SELECT * from CarSells where NOT (seat_count > 0 AND seat_count < 10);

- Transmission should not any other value apart from man (manual) and auto (automatic).

  /* Check if transmission value is not other than auto, man*/
  SELECT * from CarSells where transmission NOT IN ("auto", "man");

**4 YesWorkflow**

Represents the workflow of the data and how it changed during the different steps of the process. Firstly, the data root is the car_sells.csv and perform different cleaning steps using openrefine. The below diagram shows process of cleaning.



## 5. CHALLENGES

In our dataset- the pricing variance and pattern was not very obvious for second hand cars along with missing values in car maker and model. There were also cases of missing manufacture date and we had to derive the dates based on mapping engine size and correlation with similar rows.
The most challenging problem within data cleaning remains the correction of values to remove duplicates and invalid entries. In many cases, the available information on such anomalies is limited and insufficient to determine the necessary transformations or corrections, leaving the deletion of such entries as a primary solution. The deletion of data, though, leads to loss of information; this loss can be particularly costly if there is a large amount of deleted data. When data is this huge; there is no way to completely verify the integrity of our steps.

## 6. CONCLUSION

Data Cleaning or Data Wrangling is not only an effective tool for removing the unwanted data ”dirty” data, but also the medium to make data in our system selective, concise and appropriate in order to perform the better analysis. Based on our experience, we can say Data Cleaning is one of the most time-consuming steps in any Data Analysis. We tried to clean the data using OpenRefine and SQLite. It took us significant time to clean the data using Open- Refine, specially because of the way it enables its usage. Tools like OpenRefine have their limitations in terms of how much data can they handle; and that’s why more people are inclined towards using scripting methods with Python, R etc. but they do not provide the visibility that OpenRefine does.

## 7. ACKNOWLEDGMENTS

We would like to thank Prof. Bertram Ludaescher, for this opportunity. We would also like to our TA and our fellow MCD-DS students, their community at Slack/Piazza has proved to be an exceptional support structure. Special mention to the Kaggle website to provide an online community of data scientists and machine learners