

# KE5106-DATA WAREHOUSING FOR BUSINESS ANALYTICS



## SINGAPORE USED CARS MARKET

AN END TO END PIPELINE TO SOURCE, SCRAPE, CLEAN  
AND ANALYZE LISTINGS FROM THE 2 MOST POPULAR USED  
CAR LISTING SITES IN SINGAPORE

## TEAM MEMBERS

ANURAG CHATTERJEE (A0178373U)  
BHUJBAL VAIBHAV SHIVAJI (A0178321H)  
LIM PIER (A0178254X)  
LIU THEODORUS DAVID LEONARDI (A0178263X)  
TSAN YEE SOON (A0178316Y)

## Table of Contents

1. Executive summary .....	1
2. Background and Project Objective .....	2
3. Data sources and NoSQL data modeling .....	2
3.1 Listings Collection .....	2
3.2 Manufacturers Collection .....	3
3.3 Models Collection .....	3
3.4 Uninserted Collection .....	4
4. Data Crawling Strategy .....	4
4.1 Initial data collection .....	4
4.1.1 <i>Gathering URLs to crawl</i> .....	4
4.1.2 <i>Handling different layouts:</i> .....	5
4.1.3 <i>Handling of data formats and data types:</i> .....	5
4.1.4 <i>Batch insertion of crawled records</i> .....	5
4.1.5 <i>Handling errors during crawling</i> .....	5
5. Data Storage and Aggregate computation Strategies .....	5
5.1 MongoDB Operations API .....	5
5.1.1 <i>Data cleaning</i> .....	7
5.1.2 <i>Mandatory fields verification</i> .....	7
5.1.3 <i>Data types verification</i> .....	7
5.1.4 <i>Computations of aggregates during data insertion</i> .....	7
5.1.5 <i>Aggregates creation using map reduce</i> .....	7
5.1.6 <i>Handling errors</i> .....	7
6. Proposed strategy for incremental load .....	7
6.1 New car listed in source website .....	8
6.2 Details of existing listing is updated in one of the source websites .....	8
7. Analytics on collected data .....	8
7.1 Descriptive Analytics .....	8
7.2 Predictive Analytics .....	9
8. Conclusion .....	10
9. References .....	10

## 1. EXECUTIVE SUMMARY

---

In Singapore, buying a new car is a costly affair. In order to reduce the initial purchasing price, more and more people are opting for a used car. This is why Singapore has a relatively large used car market. Nevertheless, buying a used car still requires a substantial amount of money, and this project aims to provide buyers and sellers with additional insights from data mining two of the leading used car websites – sgCarMart and OneShift.

We start by describing our database structure, which is based on a MongoDB NoSQL database. In designing this NoSQL database, one of the considerations we did was to put specific fields like Price, COE and Road Tax, which are used for computations, at the root level of the database for efficiency. Details like upfront payment and seller are only required for additional display purposes and thus are nested.

We also described our breadth-first data mining methodology. For greater convenience, we developed a Python-based API to parse, check for errors and insert new mining data in a structured way into the MongoDB collections. Erroneous data was inserted into a separate collection for checking afterwards. We also considered that we would be running these mining scripts on a periodical basis and have proposed ways to most effectively implement the database updating methods.

Lastly, we also developed visual analytics based on the data collected. This analytics were able to give good insights to buyers and sellers and are in line with what we have physically observed in the news. Sellers can also use our predictive model to predict a suitable selling price for their car based on the market conditions. Our analytics are also beneficial to new cars buyers if they want to check what could be estimated selling price of their new cars in the near future if they want to sell their car.

## 2. BACKGROUND AND PROJECT OBJECTIVE

---

Singapore used car market is relatively large with up to 9,000 used cars sold monthly. The used car market frequently exceeded the new cars market due to the sky-high prices of the cars ownership and many motorists are opting to purchase used cars to reduce the initial purchasing cost [1].

Singapore used car market is significant enough that Singapore Press Holdings is willing to shell out S\$60 million to purchase online car portal sgCarMart, a record amount in the country's digital media sector [2]. In addition to sgCarMart, other online marketplaces for used cars have also sprung up such as OneShift, giving sellers and buyers more choices to list or source for vehicles online.

Although used cars can cost significantly lower than a new car, it is still a costly purchase running over tens of thousands. From our initial exploration, there is additional information, which will be useful to both sellers and buyers of the used car market, that we can data-mine from the online car portals.

Our project objective is to create an end to end pipeline that can efficiently crawl used car listings information from sgCarMart and OneShift online used car market. We will then use this data to create analytics that is useful for car sellers and buyers.

For example, a motorist wants to sell his car and wants to know what price he should price his car. He can use our predictive model, created from the data crawled, to suggest the price. Likewise, a 2<sup>nd</sup> hand buyer can also use the same model to know what price he is expected to pay if the car is of certain requirements. Other analytics, like postings per day in the week, can help a seller to decide the best day to post a car listing, such that he/she will not get bumped down the list of postings so quickly.

---

## 3. DATA SOURCES AND NOSQL DATA MODELING

---

The websites considered for the crawling are [sgCarMart](#) and [OneShift](#). A total of 38,950 records were crawled from these two websites and inserted into MongoDB. Based on the objective of our analysis, we came up with the following data model in MongoDB.

### 3.1 LISTINGS COLLECTION

This is the primary collection where the cars listings crawled from the cars websites is stored. Based on the snapshot of a document in the collection below, the root fields like transmission, category, engine\_cap indicate the various specifications of the cars. Some of the root fields like price, coe, omv, road\_tax are used for computing aggregates. Since these fields are used for computations, keeping them at the root level is more efficient. Nested fields like upfront\_payment and seller tell us additional details regarding the car and are only for additional details display purpose.

Key	Value	Type
price	68800.0	Double
availability	true	Boolean
reg_date	2013-04-19	String
coe	67010.0	Double
omv	20178.0	Double
depreciation	12583.0	Double
road_tax	1663.0	Double
transmission	automatic	String
engine_cap	2384.0	Double
mileage	59533.0	Double
description	pristine condition with low mileage, clocked at 59533km only. viewing by appoi...	String
features	2.4l dohc 4 cylinder engine producing 164bhp, 6 speed auto transmission, srs air...	String
fuel_type	petrol	String
no_of_owners	3	Int32
type_of_veh	suv	String
category	parf car	String
colour	black	String
posted_on	2018-08-04	String
upfront_payment	{ 4 fields }	Object
transfer_fee	25.0	Double
down_payment	20640.0	Double
1st_installment	839.65	Double
total_upfront_payment	21504.65	Double
seller	{ 4 fields }	Object
contact_persons	bobby boh	String
dealer_name	cartrade sg	String
dealer_address	18 boon lay way #06-127 tradehub 21, singapore 609966	String
type	dealer	String

Figure 1. Part of a document in the listings collection

### 3.2 MANUFACTURERS COLLECTION

This collection consists of car manufacturers and their corresponding models. This collection acts as a reference collection and is used for comparing models and manufacturers while inserting data in the “listings” collection. This collection also does the aggregations based on the manufacturers. The fields `sum_of_prices`, `total_days_posted` and `quantity` contain the aggregate values. The below is the snapshot for the “Bentley” manufacturer.

_id	ObjectId("5b7a5b005b822825b5ae1b87")	ObjectId
name	Bentley	String
models	[ 5 elements ]	Array
[0]	Mulsanne	String
[1]	Arnage	String
[2]	Bentayga	String
[3]	Flying	String
[4]	Continental	String
sum_of_prices	54161088.0	Double
quantity	132	Int32
total_days_posted	4566	Int32

Figure 2. A document in the manufacturers collection

### 3.3 MODELS COLLECTION

This collection consists of model level aggregations for the car model. The fields `sum_of_prices`, `quantity` and `total_days_posted` contain the aggregate values. The following is the snapshot for “Audi A5” model in the collection.







 _id	ObjectId("5b7a5b005b822825b5ae1be6")	ObjectId
 name	A5	String
 manufacturer	Audi	String
 sum_of_prices	10082428.0	Double
 quantity	139	Int32
 total_days_posted	2889	Int32

Figure 3. A document in the model collection

### 3.4 UNINSERTED COLLECTION

This collection is used to trace the reasons for crawling or insertion failures. While scraping data from the source websites, we may encounter some unusual data formats or layouts. This erroneous data goes inside this collection with the error message. Also, any exceptions in the application layer while inserting records to MongoDB are stored here. The following is the snapshot from MongoDB for this collection.






Key	Value	Type
 (1) ObjectId("5b7ae33b5b82283d831ff1de")	{ 4 fields }	Object
 _id	ObjectId("5b7ae33b5b82283d831ff1de")	ObjectId
 url	http://www.sgcar mart.com/used_cars/info.php?ID=713928&DL=1359	String
 data	{ 0 fields }	Object
 error_details	Crawling error: could not convert string to float: 'n.a.'	String

Figure 4. A document in the uninserted collection

## 4. DATA CRAWLING STRATEGY

### 4.1 INITIAL DATA COLLECTION

#### 4.1.1 Gathering URLs to crawl

We incorporated a Breadth-First strategy for crawling the websites. The algorithm can be illustrated by using the following figure.

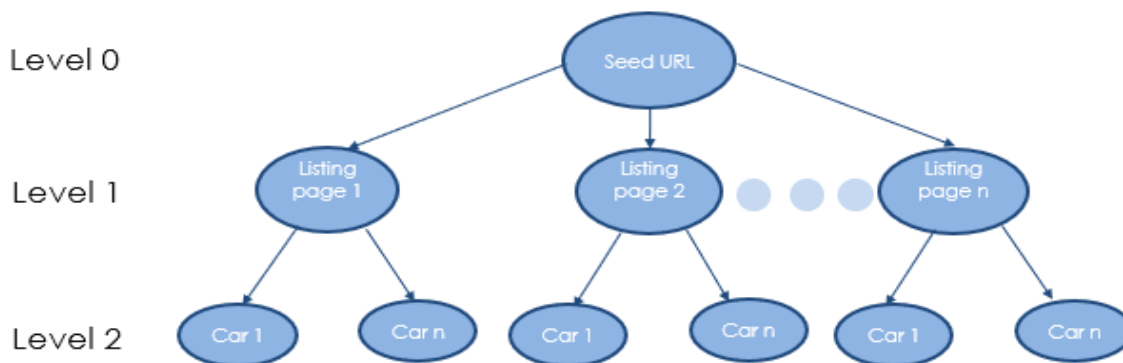


Figure 5. Breadth-First strategy for crawling

As can be seen from the above figure, we fetched out the seed URL for the website (Level 0). Then all the main listing pages URL's have been taken out (Level 1). Then we took out each of the individual cars from the cars listing pages (Level 2). We used Breadth-First strategy because our crawling was not intended to be deep. We restricted the crawling till the Level 2.

#### 4.1.2 Handling different layouts:

We considered 50 different webpages of the individual cars from the different time intervals to cover different layouts of the webpages to encounter inconsistency in layout while scrapping the data from those individual pages.

#### 4.1.3 Handling of data formats and data types:

Only numeric data was taken out from the alphanumeric fields to do aggregations and other operations on top of them. E.g. if power is given as "110 bhp" then only "110" has been scraped for that field. Then these numerical fields are converted into an integer or real data types. Date fields like 'Posted on' are converted into MongoDB friendly ISO date format. The fields are having values like '-' or 'n.a.' then for these fields are inserted with "null" in MongoDB.

#### 4.1.4 Batch insertion of crawled records

To ensure that the database insertions are scalable and proper error handling, the crawled records are inserted into the database in batches.

#### 4.1.5 Handling errors during crawling

Errors during crawling, due to bad data in the source website which lead to parsing errors are directly inserted into the Uninserted collection in MongoDB with details of the reason for the failure.

---

## **5. DATA STORAGE AND AGGREGATE COMPUTATION STRATEGIES**

---

### **5.1 MONGODB OPERATIONS API**

The Crawler inserts the data into MongoDB via the API layer. The advantage of having the API layer rather than directly inserting the data into MongoDB from the crawler is that the data can be cleaned by validating the presence of mandatory fields, data types, etc. before the insertion. Also, computation of aggregates can be done for some online aggregate computations at this stage. The crawled records that fail the validations are inserted into a separate "Uninserted" collection with a string containing the error details for later examination of the reason for the failure. The design of the API methods allows batch calls and so multiple records are inserted into MongoDB in a batch.

We initially populate the manufacturers and the models into the database. Since both the sources clearly mention the manufacturers and the models in their user interface, these details are crawled and exported to CSVs due to their small numbers. A Python script then imports them into MongoDB after performing the necessary transformations to align the data to the data

model. This portion is done one-time, and only re-run when we know new models / manufacturers of cars exist.

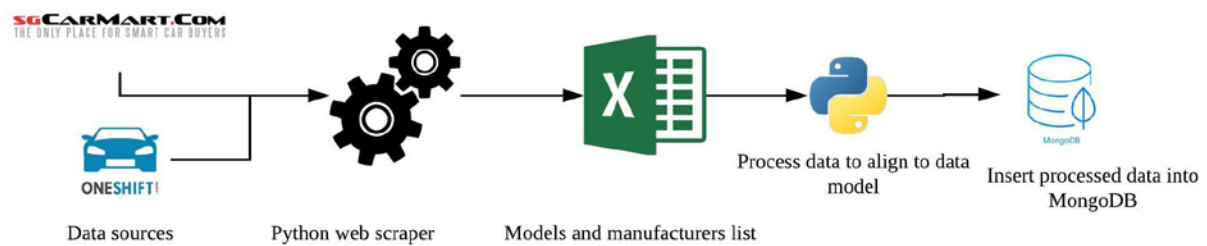


Figure 6: Part 1 - Manufacturers and models data insertion

The second part of the data insertion consists of the web crawler inserting the details of the car listings into MongoDB via the Python interface in batches. This portion will be run periodically in order to keep the database up-to-date.

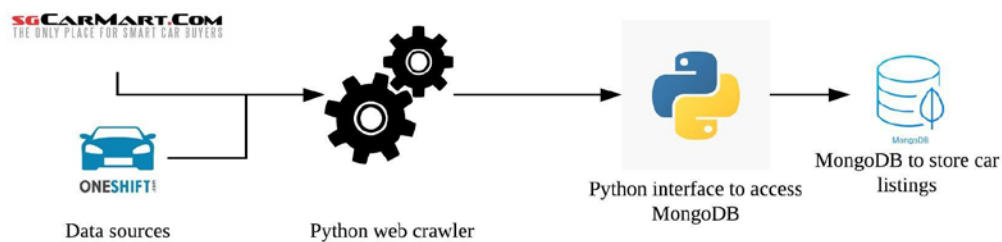


Figure 7: Part 2 - Inserting car listings via Python API

This periodical program consists of fine-grained data manipulation where certain checks are performed on the incoming listings to ensure that the data adheres to the data model to be suitable for the analysis later.

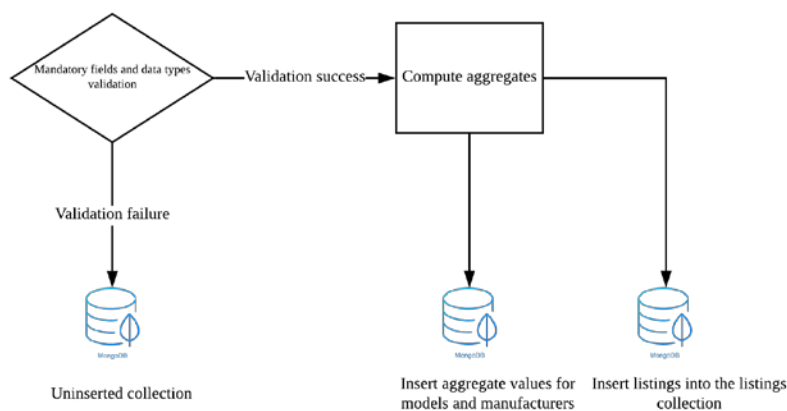


Figure 8: Car listings insertion process flow in Part 2



The below sections describe the various functionalities of the operations API and how certain use cases are handled both at the API level and at the MongoDB level.

#### 5.1.1 Data cleaning

The manufacturer and the model for the vehicle are extracted from the title of the listing to ensure that the values that go to the database are from the existing values of manufacturer and model that have been crawled earlier.

#### 5.1.2 Mandatory fields verification

Certain fields like URL, source of the listing, manufacturer and model of the vehicle, when it is posted on are required for further processing, and so any records without these fields are not inserted into the main listings collection.

#### 5.1.3 Data types verification

The data types of fields like price and availability are checked to ensure that they are double and Boolean respectively when they are present. A validation failure would lead to the listing being inserted into the uninserted collection.

#### 5.1.4 Computations of aggregates during data insertion

To keep some of the fields holding the aggregates like the number of cars sold for each manufacturer, sum of price for the models, etc. up to date for online queries, these are computed during the insertion of each of the listing record.

#### 5.1.5 Aggregates creation using map reduce

Some aggregates that were required for analysis and were not computed online during the data insertion process were computed using map reduce operations in Mongo DB written using JavaScript. The results of the map-reduce operation were inserted in a new collection that could be later processed.

#### 5.1.6 Handling errors

Exception handling is put in place to ensure that any exception that happens during the whole flow is logged into the Uninserted collection with the appropriate reason for the failure so that these records can be reviewed later.

---

## **6. PROPOSED STRATEGY FOR INCREMENTAL LOAD**

---

This section will focus on the design for the incremental load, as new listings are added, or existing listings are updated in the source websites. To efficiently achieve the same, we propose some changes to the existing data model. First, we create a new field that will hold the MD5 checksum of the fields that are most likely to change in the listing record, like the car price, description, etc. Then we create indexes on the URL and the newly created MD5 checksum fields. Finally, we have a field to capture the date when the record was crawled.

Following scenarios are considered while collecting the data incrementally.

## 6.1 NEW CAR LISTED IN SOURCE WEBSITE

To insert the cars that are newly listed on the source website the below algorithm has been designed:

- a. Check the frequency at which listings are added in the source website, for our case both the source websites are updated daily with new listings
- b. Go on scanning the car listing URLs till we find an URL that already exists in the database. This is efficiently performed due to the index on the URL field. Also, since the car listings are sorted in descending order by the date of posting, we do not scan any further once a listing has been found to exist.
- c. Insert all the absent car details by scraping the details from the URLs absent in the database.

## 6.2 DETAILS OF EXISTING LISTING IS UPDATED IN ONE OF THE SOURCE WEBSITES

When the detail of an existing listing is updated in the source website the below algorithm will be used to ensure the latest data is present in our database:

- a. Check the date range during which the listings are updated in the source websites – we observed that within three months on average a listed car is sold and so we propose to keep three months as the date range.
- b. Query records in the listings collection that were crawled within the last three months of the current date.
- c. For each of the records returned by the above query:
  - i. Match the MD5 checksum of the current record with that of the current record in the source website by requesting the record using the URL.
  - ii. If the MD5 checksums are different, then we scrape the record from the source website and insert the record in the database with the latest crawled date field to differentiate between current and historical record.
  - iii.

The index on the URL field will be created as a non-unique index to ensure that records with duplicate URLs can be inserted based on our proposed design [3].

---

## 7. ANALYTICS ON COLLECTED DATA

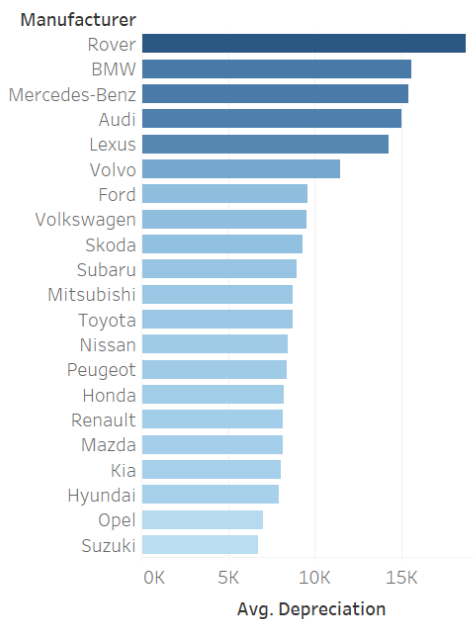
---

### 7.1 DESCRIPTIVE ANALYTICS

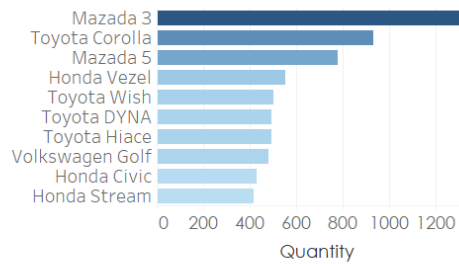
For our analytics on collected data, we can create meaningful descriptive analytics that will be helpful to both used cars buyers and sellers. For example, the selected dashboard view shown in figure 9 allows the potential buyer to know at a glance the average depreciation of each manufacturer, and from there he can also zoom into the average depreciation of the car model he wanted to search on.

## Selected Dashboard View

## Average Depreciation by Popular Manufacturers



## Top Models Listed for Sale



## Avg Days Listed before Sold- Selected Manufacturers

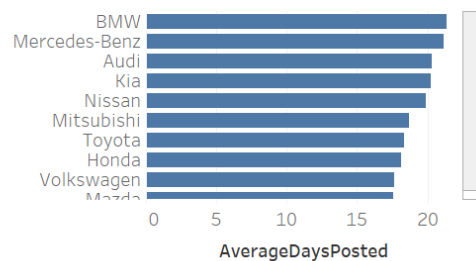


Figure 9. Sample descriptive analytics dashboard

For the seller, he can know what is the average days of the manufacturers/models that his car will take to sell on the website. If the seller is unable to sell after the certain number of days over the average, he can review his listing and perhaps revise his selling price. Other descriptive analytics are also done which are not shown here can help car buyers and sellers understand better the online used car marketplace. For example, a majority of online listings on the two portals are listed not by direct owners but by agents or car companies and even the most popular color of the listed cars etc.

## 7.2 PREDICTIVE ANALYTICS

In addition to useful descriptive analytics, we also built a predictive model from the data we crawled. The model can provide a pricing guide to the seller to price his car in the listing. This model is also built using assistance from experts in the used car market.

These analytics also may help new car buyers as he can also see how much his new car could be sold if he were to trade in his car in the future.

Hyundai Elantra 1.6A Elite

Overview

Financial

Accessories

Similar

Research

Photos

Add to Shortlist

Add to Compare

Add a Note

Report Error

Car Details

Price	<b>\$43,800</b>
Depreciation	\$9,640 /yr <a href="#">View models with similar depreciation</a>
Reg Date	13-Jun-2012 (3yrs 9mths 17days COE left)
Manufactured	2012
Mileage	101,000 km (16.3k /yr)
Transmission	Auto
Engine Cap	1,591 cc
Road Tax	\$738 /yr
Power	95.6 kW (128 bhp) <a href="#">View specs of the Hyundai Elantra (2011-2014)</a>
Curb Weight	1,267 kg
Features	Powerful And Responsive 16L DOHC, CVT Engine, 6 Speed. Producing 128 BHP. Multi Function Steering.

How much is your car worth?

Age left [COE] (years)

3.80

Mileage (km)

101000

Engine Capacity (cc)

1591

Road Tax (\$)

738

COE (\$)

64201

OMV (\$)

14417

No. of Owners

1

Vehicle Type

sedan

Manufacturer \* Future \*

Hyundai

Model \* Future \*

Elantra

Current COE Price \* Future \*

40000

Predict!

Predicted Price: \$44,800.00

Figure 10: Predicting the selling price of your car

## 8. CONCLUSION

As part of the project, we have successfully developed an end to end pipeline to the source, scrape, clean and analyze listings from the two most popular used car listing sites in Singapore. We have leveraged a popular NoSQL document database, MongoDB to store the crawled data. We have designed the application as a tiered application where the inserts into the MongoDB happen via the application layer which ensures that the inserted data is clean and adheres to our data model. The inserts are performed in batches, and all errors are logged in a separate collection in the database. To perform descriptive analytics efficiently, we have captured the fields to perform aggregates on at the root level of the records. To compute the aggregates, both online and offline methods using map-reduce have been evaluated. A predictive model has also been developed to help predict the price of the car for the seller to list the car with the correct price.

## 9. REFERENCES

- [1] LTA, "Land Transport data mall," [Online]. Available: <https://www.mytransport.sg/content/mytransport/home/dataMall.html>.
- [2] W. Wee, "Tech in Asia," 2013. [Online]. Available: <https://www.techinasia.com/sph-acquires-singapore-sgcarmart>.
- [3] C. Heald, "Advantage of a unique index in MongoDB," 10 9 2012. [Online]. Available: <https://stackoverflow.com/questions/12350879/advantage-of-a-unique-index-in-mongodb>.