## Foundations of Data Science - Capstone Project

Jonathon Thompson

**1. What is the problem you want to solve?**

Getaround's [current search function](https://www.getaround.com/search) is utilitarian in nature and requires users to have preexisting knowledge about the types of cars they want to rent. Results are presented to potential renters based on proximity, availability and cost, with simple vehicle filters limited to transmission type, parking, body type and make.

This project’s goal is to demonstrate a new approach to searching and presenting results to renters based on what they intend to do with the car during their trip. For example, instead of blindly sifting through listings, a renter could search for cars that are good for “running errands”, “economical” or “fun”.

**2. Who is your client and why do they care about this problem? In other words, what will your client DO or DECIDE based on your analysis that they wouldn’t have otherwise?**

“Getaround is an online car sharing or peer-to-peer car sharing service that allows drivers to rent cars from private car owners, and owners to rent out their cars for payment. Owners set their rental prices and earn a 60% commission from their rental revenue. The company currently operates rentals in the San Francisco Bay Area, San Diego, Austin, Portland, Chicago, and Washington, D.C.” <https://en.wikipedia.org/wiki/Getaround>

This project is intended to be a road map for product growth, designed to show Getaround how additional search features could enhance both the renters' and owners’ experience. While today’s customers may only come to Getaround to get from point A to point B, a more dynamic way of presenting listings could expand the variety of vehicles that repeat renters utilize and attract a broader customer base; think Discover Weekly for car sharing.

This concept could start with using 3rd-party data like Edmunds, but could shift towards Getaround becoming a content curator - using owners’ listings and descriptions to allow owners to have an impact into their listing’s rankings. The renter’s review process could be expanded to capture more feedback like rankings in specific categories and questions asking what the vehicle was used for.

Imagine a search that one day incorporates a renter’s own history, ratings from other renters, listing descriptions and features from owners and is able to present you with the perfect car for your trip.

**3. What data are you going to use for this? How will you acquire this data?**

I’m going to leverage data accessible via APIs from two primary sources - Getaround for the base listing of vehicles and Edmunds for car features and ratings data. Below are the sources I’ve identified along with some of the preliminary data considerations.

[Getaround](https://index.getaround.com/v1.0/search?product=web&uid=100005837281185&user_lat=37.7717185&user_lng=-122.44389289999998&viewport=37.514591%2C-122.644614%2C38.028441%2C-122.243172&properties=car_id,car_name,car_photo,carkit_enabled,distance,latitude,longitude,make,model,price_daily,price_hourly,price_weekly,total_price,timezone,year,dedicated_parking&sort=best&page_sort=magic&page_size=1000" \t "_blank)

year, make, model, car\_name, car\_id, car\_photo, dedicated\_parking, carkit\_enabled, price\_daily, price\_weekly, longitude, latitude, distance, timezone

[Edmunds](http://developer.edmunds.com/api-documentation/overview/" \t "_blank)

SPEC: VEHICLE STYLE

GET STYLES DETAILS BY VEHICLE MAKE/MODEL/YEAR

- This allows me to take the year/make/model from Getaround and return a list of trims and the style\_id (this is a primary identifier in the API).

SPEC: VEHICLE EQUIPMENT

GET EQUIPMENT DETAILS BY STYLE ID

- This is where the heart of the details are at.

- Since I don't know the specific trim, I plan on taking one style\_id and using that to pull back only "STANDARD" features that would be common across all trims for that year.

CONTENT: EDMUNDS CAR RATINGS

GET CAR RATINGS BY MAKE/MODEL/YEAR

- Rankings in various categories by Edmunds reviewers as a score of 1-10 or a grade of A-F.

**4. In brief, outline your approach to solving this problem (knowing that this might change later).**

* Import the data into R via the APIs and save a static data set as .csv for each of the calls.
* Cleanse the data
* Join Getaround listings with Edmunds SPEC: VEHICLE STYLE style\_id on year, make and model.
* Get full list of features from SPEC: VEHICLE EQUIPMENT for the universe of Getaround listings.
* Create buckets of features that map to usage definitions like “running errands”, “economical” or “fun”.
* Get full list of review categories from CONTENT: EDMUNDS CAR RATINGS for the universe of Getaround listings.
* Look at relationships between features, categories and ratings.
* Create algorithm or look for pre-existing recommender systems and test results.

**5. What are your deliverables? Typically, this would include code, along with a paper and/or a slide deck.**

I’ll deliver to the client working code featuring a recommender system to enhance search results along with a slide deck that shows how Getaround can continue to build out on this concept to drive future product growth.