CPSC 490 - Project Proposal

**An Automotive Price Prediction Engine**

Sean Haufler

February 5, 2014

**Advisor:** Lecturer Brad Rosen  
*Dept. of Computer Science, Yale University*

**Abstract**

I will build an automotive price prediction model that takes in a car’s make, model, year, geolocation, mileage, and economic trends (and potentially more info like trim and CARFAX vehicle history) and estimates its local market value. Currently, people still use Kelly Blue Book to estimate their auto’s price, which is notoriously inaccurate. Kelly Blue Book only uses a car’s make, model, year, and milage, but doesn’t take other factors into account. We will design a model that takes in data from nationwide and local car dealers to generate the price. This can help to solve a real-world problem: finally, people will be able to use an open-source price prediction model to find out a car’s price instead of being surprised (or duped) by a car dealer.

I’ll make a UI to browse through the database of cars. I’ll experiment with different methods to implement the models, first using a dynamic regression analysis, and potentially applying machine learning techniques. The exact implementation to be used is to be determined - I’ll select the method that provides the most accurate results given the data. For the user interface, I plan to use a modern client-side software engineering stack with open-source tools and frameworks including Backbone, React, Gulp, and others.

**Introduction**

For many people, buying or selling a car is often a highly stressful experience. When negotiating prices with a dealer, consumers face large informational disadvantages. Dealers have internal tools that provide them live data about their local automotive markets, which gives them leverage in negotiations with consumers. If a consumer wants to determine the value of a car, he may use a service like Kelley Blue Book; however, these consumer services largely don’t take real-time pricing or geolocation data into account. Consequently, automotive prediction models like Kelley Blue Book do not account for constantly changing local market conditions.

Kelley Blue Book pricing models can be improved. Automotive search engines like AutoList.com aggregate millions of car dealer listings and update their search results in real time. This body of data, when analyzed, can provide geolocation-specific and hour-by-hour insight on the nature of the automotive market. By integrating the prices of local car listings into a price prediction engine, one can generate a more accurate price prediction of a certain car.
The proliferation of cloud services has made this real-time analysis feasible. These services allow for the deployment of scheduled web crawlers to collect and aggregate this automotive data. We will build a low latency, robust set of crawlers to continually collect a corpus of data for use in our price prediction algorithm.

Motivation

A recent Gallup poll ranked car dealers as the ‘least trusted’ vocational field.\textsuperscript{1} They ranked lower than Congress. Economists posit that car dealers are financially incentivized to sell their cars for inflated prices through high-pressure sales techniques, even at the expense of future customer satisfaction. Since their customers buy new cars only once every several years on average, dealers have less reason to optimize for customer happiness. People move, on average, once every ten years. So when a car dealer sells a vehicle to a customer, it is more likely than not that the customer will live in a different neighborhood the next time they want to buy a car. In game theoretic terms, a dealer/customer interaction is a non-repeated game where the Nash Equilibrium is for the dealer to defect.

Since this economic anomaly will not likely be solved in the near future through policy changes, the best short-to-medium-term solution for the customer is to make sure he is not vulnerable to high-pressure sales tactics. My decreasing the informational asymmetry between himself and the car dealer, the customer can gain leverage in the price negotiation. The intent of this project is to make car price negotiations more fair - to give the power back to the consumer.

Dataset

The dataset will compose of dealer and car listing information provided by AutoList.com. AutoList aggregates and lists millions of used cars and provides an interface through which to view and browse through them. Each listing will have a car’s make, model, year, listing price, and location. Through the analysis of this dataset, I will build the price prediction model.

Price Prediction Model

The price prediction model will be determined through dynamic regression, with variables for mileage, location, car condition, and make/model/year.

The model may also use machine learning techniques to find patterns in pricing trends and predict future pricing for the vehicle. The model could give recommendations for users. For example, the model could suggest that the user trades in their car to a dealer towards the end of the month if the model discovers that trade-in prices increase sharply towards the end of each month.
User Interface and Software Engineering Stack

Overview

I will build a user interface to browse, sort and filter the corpus of live dealer listings. To facilitate code reuse and maintainability, the UI will be architected so that it can provide similar functionality to an for data objects with arbitrary field names and values. In order to build an interface following current software engineering best practices, I will employ several open source frameworks and libraries, including Backbone.js, React.js, and a built tool called Gulp.

Backbone

Backbone is a javascript framework that facilitates Model-View-Controller (MVC) frontend web architectures. It provides models with key-value binding and custom events, and provides collections with an API of list functions. In this application, a car data object would be represented by a Backbone model, and the set of cars would be a collection. This framework will tie into Facebook’s open source library, React.js, which will handle presentation logic and view rendering.

React

React is Facebook’s javascript library for building user interfaces. It acts as the V in the MVC architecture, providing smart view rendering functionality. It creates a virtual Document Object Model (DOM) and updates and re-renders this DOM when view models are updated.

Deliverables

• A working price prediction engine and user interface that allows the user to enter a car’s make, model, year, geolocation, milage and outputs an accurate valuation based on live pricing data
• A written report covering market insights found by the price prediction, including seasonal and location-based pricing trends.
• An additional user interface that allows the user to browse, filter, and sort through the entire corpus of car listing data, with all view logic performed in the web browser.