StarrCars: Buy and Sell Cars with AI

CSC 491-492
Dr. Mugizi Robert Rwebangira
By: Adrick Malekian
21 March 2025

Table of Contents

Background

As a lifelong car enthusiast, I've always been fascinated by the intricate dance between automotive engineering and market dynamics. The process of buying or selling a car, however, often feels unnecessarily complex and stressful. Getting the right price point to ensure that both parties are satisfied is quite the challenge and requires a complex analysis of the vehicle. Traditional valuation methods, such as those offered by Kelley Blue Book, while valuable, can sometimes lag behind real-time market fluctuations and fail to capture the nuances of individual vehicle conditions and regional variations. This gap between static valuations and the dynamic reality of the used car market sparked a desire to develop a more intuitive and real-world data-driven platform for automotive transactions.

This project aims to bridge that gap by leveraging the power of artificial intelligence to create a combination of a car valuation and marketplace platform that provides users with more accurate price estimates based on real-time selling values of cars. By analyzing market data, including factors like mileage, vehicle condition, and even subtle trends in buyer behavior, the system aims to offer valuations that more closely reflect the actual market value of a vehicle.

The ultimate goal is to empower both buyers and sellers with the information they need to make informed decisions, fostering a more transparent and efficient automotive marketplace tailored to those with less experience in the car market. This project ultimately represents a fusion of my passion for automobiles and my belief in the transformative potential of AI to revolutionize the way we interact with the automotive market.

Data Collection

The initial phase of this project encountered challenges in identifying a dataset possessing sufficient features and data points to facilitate robust clustering. A comprehensive dataset, sourced from Kaggle, was selected, comprising 66 attributes detailing approximately 3 million used vehicles within the United States. This data, acquired in September 2020 through web crawling of the CarGurus automotive platform, demonstrated high topical relevance. However, the dataset's substantial file size necessitated reduction for practical processing within the Python environment.

Preliminary data preparation involved the elimination of non-essential features, such as exterior color, and attributes with prevalent null values, such as pickup truck cabin size. The Pandas library was employed to read the raw data in 10,000-row increments, retaining the ten most pertinent features and subsequently writing the refined data to a new file. While column reduction significantly decreased file size, subsequent clustering attempts on the full 3 million data points resulted in memory allocation errors. Consequently, a random sample of 10,000 data points was extracted and utilized as the operational dataset.

Development

Project Setup

The project was initialized using a monorepo structure, separating frontend and backend concerns. The backend was configured with Node.js and Express, with TypeScript ensuring type safety.

Database Design & API Development

- **Schema Design**: Utilized MongoDB Atlas to design collections for users, vehicles, transactions, and listings.
- **API Routes**: RESTful API endpoints were developed to handle CRUD operations for car listings and user management.
- **Authentication**: JWT tokens stored in HTTP-only cookies were implemented for secure user sessions.

Frontend Development

- **React with TypeScript** was used to build reusable components for listing, searching, and filtering vehicles.
- **Custom CSS Templates** were applied for styling, ensuring a clean and modern UI.
- **State Management** was handled via Context API/Redux for efficient data handling.
- **AI Assistant Integration**: A chatbot feature was embedded to provide users with AI-driven price estimations.

AI-Powered Price Estimation

- A pre-trained machine learning model was incorporated to analyze car attributes and suggest market prices.
- Backend endpoints were created to process user inputs and return estimated prices dynamically.

Cloud-Based Image Storage

- Images uploaded by users were stored using **Cloudinary**.
- API endpoints were developed to handle secure image uploads and retrievals.

Testing & Deployment

- **Unit and Integration Testing**: Jest and Postman were used for backend testing.
- **Frontend Testing**: Manually tested frontend components

Technology Stack

Frontend:

- **Framework**: React.js with TypeScript
- **Styling**: Custom CSS Styles with UI Theme
- **State Management**: Context API / Redux
- **UI Theme**: Daytona Blue and Light Grey

Backend:

- **Language**: TypeScript (Node.js)
- **Framework**: Express.js
- **Database**: MongoDB Atlas
- **Authentication**: JWT-based authentication with secure cookies
- **Middleware**: CORS, Cookie-Parser, Dotenv
- **Cloud Storage**: Cloudinary / AWS S3 for image uploads
- **AI Integration**: Machine Learning model for price estimation

Implementation and Testing

The AI implementation was quite interesting, as there was a lot of trial and error to achieve the best results. First and foremost, I decided to test two different price prediction models: one using K-Means and one using KNN. After some preliminary research, I determined that the KNN model would most likely produce the best results because object classification is more likely to use categorical variables. Since my goal was to generate an estimated price range, a KNN model based on feature vectors and distance metrics would provide a more precise estimate. Regardless, I tried both models to see what the results would be like.

K-Means Clustering Model

First, I trained the KMeans Clustering algorithm with the following parameters: make, model, year, price, mileage, city fuel economy, highway fuel economy, and horsepower. Unfortunately, certain critical parameters such as condition and engine were not included in the dataset. To simulate the engine size feature, I used the city and highway fuel economy and horsepower features, which are strong indicators of what type of engine is in the vehicle given the make and model. The next step was hyperparameter tuning, such as determining the optimal k. *Figure 1* shows the optimal k based on the elbow method and silhouette scores, which was three clusters.

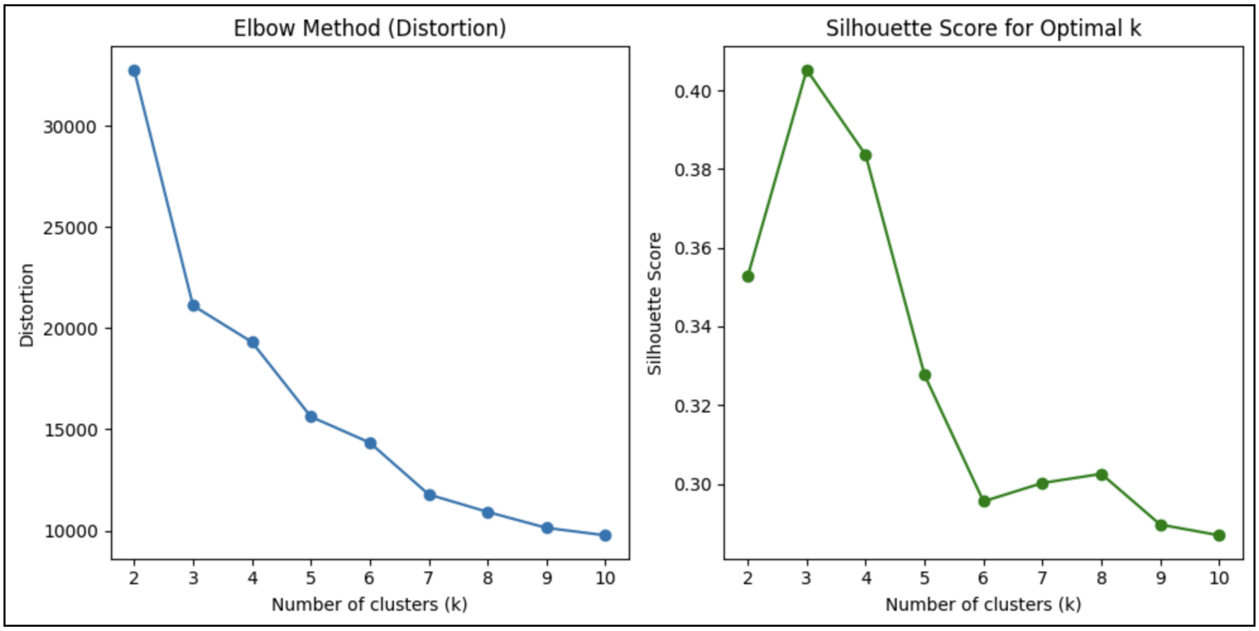


Figure 1: Finding the best KMeans k number of clusters

The K-Means clustering model achieved a silhouette score of 0.1910, which provides insight into the overall cohesion and separation of the identified clusters. The silhouette score ranges from -1 to 1, where higher values indicate that the data points are well-clustered and distinguishable from one another. A score of 0.1910 in my K-Means model suggested that while the clustering process is moderately effective, there was some overlap between clusters, indicating that the vehicle features used in the model have similarities across multiple categories. This suggests that while the model is useful for segmenting vehicles into similar groups based on their characteristics, there may be opportunities to refine the clustering process by incorporating additional distinguishing features or exploring alternative clustering techniques to improve separation between clusters.

K-Nearest Neighbors Model

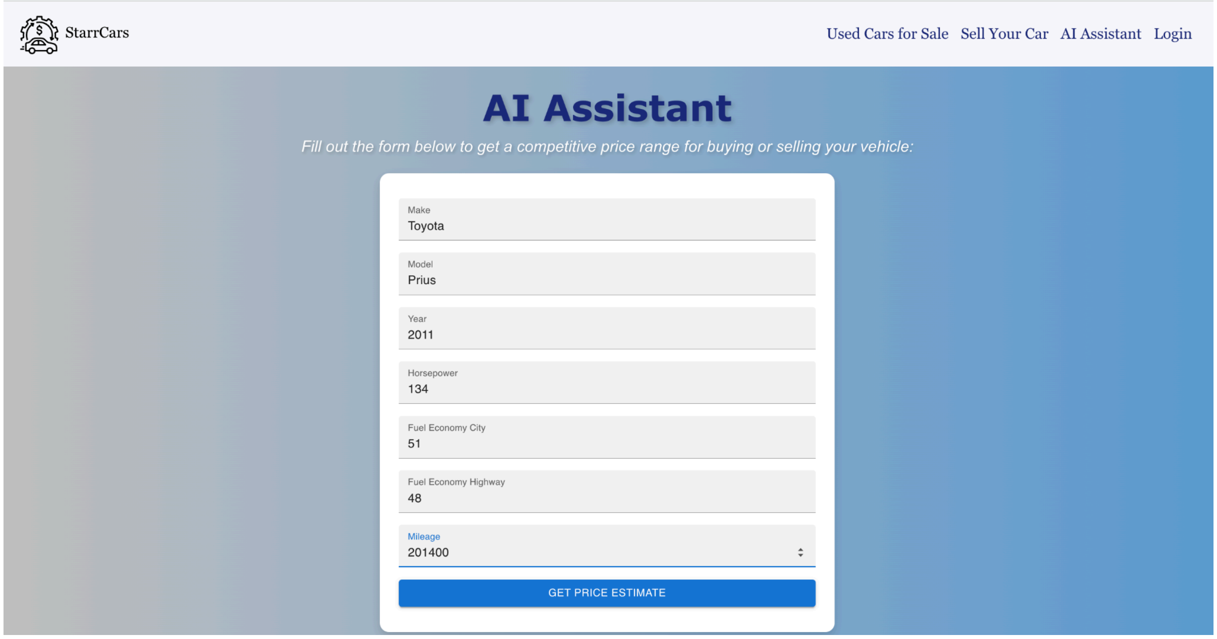
Next, I trained the KNN model with the same features, however, the categorical variables had to be encoded using the Pandas library get_dumies() function. After fitting the model and one hot encoding the categorical variables, some testing needed to be done to verify what k value would produce the highest accuracy for the model. To determine the best number of neighbors (k) for the K-Nearest Neighbors (KNN) regression model, a cross-validation approach was utilized. The function **find_best_k_knn(X_transformed, y_sample)** systematically evaluated multiple values of k, ranging from 1 to 20, using 5-fold cross-validation. The selection criterion was based on minimizing the negative mean squared error (MSE), ensuring that the chosen k value provided the most accurate predictions.

The dataset was preprocessed using one-hot encoding for categorical variables and standardization for numerical features. This ensured that all features contributed meaningfully to the distance-based calculations in KNN. Then, for each value of k, a KNN model was trained and validated using 5-fold cross-validation. The MSE scores were averaged across the folds to obtain a reliable estimate of the model's performance. The value of k that resulted in the highest cross-validation score was selected as the optimal number of neighbors. This was k = 4 and yielded the best performance, indicating that using the four closest neighbors provided the most accurate price predictions.

Considering the scale of vehicle prices, an obtained mean squared error (MSE) of 23,171,396.30 from the KNN model demonstrates a reasonably low level of error. While there are some variations between predicted and actual prices, the model is still effective in providing close estimations. Additionally, the R^2 score of 0.9197 signifies that approximately 91.97% of the variance in vehicle prices can be explained by the model’s features which is very reliable. This high R^2 value suggests a strong correlation between the predicted and actual prices, reinforcing the reliability of the model in estimating vehicle values. With all of this testing completed, I was confident in ensuring that users receive accurate price estimates based on similar vehicles in the dataset on StarrCars.

Results

Here is an example of the AI running on the website:



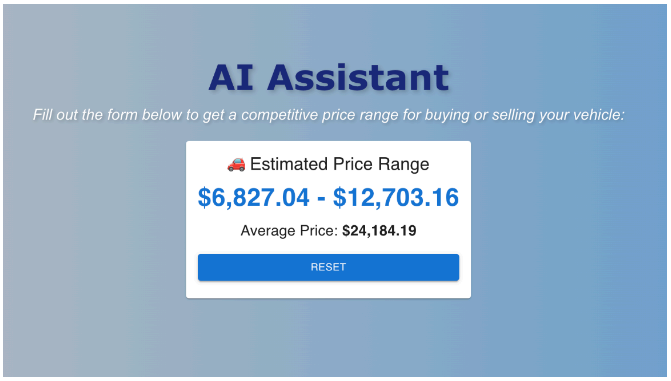
AI Assistant

Fill out the form below to get a competitive price range for buying or selling your vehicle:

Make: Toyota
Model: Prius
Year: 2011
Horsepower: 134
Fuel Economy City: 51
Fuel Economy Highway: 48
Mileage: 201400

Figure 2: AI Assistant Form to Submit a Vehicle for Price Estimation

As shown in *Figure 1*, a car is about to be sent to the backend for a price estimation. This vehicle is a 2011 Toyota Prius with high mileage and thus should spit out a price estimation range in the 6000-13000 range based on similar vehicles on Autotrader. One nice feature about my algorithm is that it understands its limitations in knowing the condition of the vehicle, so thus my price range will evaluate the knn closest price and then create a range based on the std dev of other nearest neighbors. Therefore, the user can use this price range as a starting point, gauge where their vehicle lies on the condition spectrum, and choose a competitive price in this range.



AI Assistant

Fill out the form below to get a competitive price range for buying or selling your vehicle:

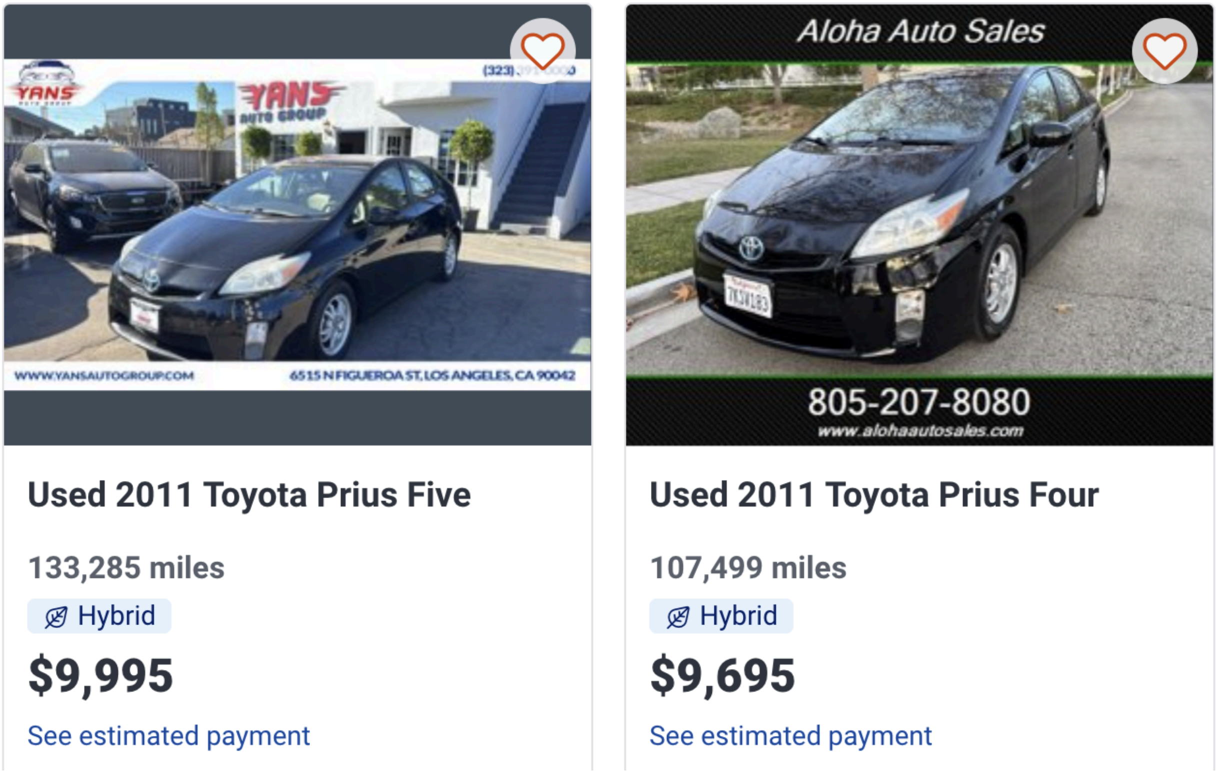
🚗 Estimated Price Range

$6,827.04 - $12,703.16

Average Price: **$24,184.19**

Figure 3: Resulting Price Range Estimate based on vehicle in Figure 1

Based on the results given by the AI model, we can see two different price estimates. I had the KNN model make the price range since I knew from the start that it would be the more accurate model in the first place. Sure enough, the price range was very accurate for vehicles that are similar to it in today's market.



Used 2011 Toyota Prius Five

133,285 miles

Hybrid

$9,995

See estimated payment

Used 2011 Toyota Prius Four

107,499 miles

Hybrid

$9,695

See estimated payment

Figure 4: Prices of similar vehicles to that of the one tested in Figure 2 [1]

Through extensive testing across multiple vehicles and rigorous validation against real-market pricing, this AI model has demonstrated its reliability and effectiveness. It serves as a powerful tool for inexperienced car buyers and sellers, enabling them to navigate the market with confidence. Pricing a vehicle competitively is often a challenge for individuals who are not car enthusiasts, and this model helps bridge that gap by providing accurate, data-driven price estimates.

Future Enhancements

To further enhance the car valuation system and provide an even more accurate and engaging user experience, I've planned the following enhancements for future work:

1. Continuous Data Integration and Real-Time Updates:

- **Automated Sales Data Ingestion:** Implement a system to automatically capture and integrate sales data from the website into the valuation model. This will ensure the model continuously learns from real-world transactions, adapting to market fluctuations and providing up-to-the-minute accurate valuations.

- **Dynamic Model Retraining:** Implement automated retraining of the valuation model with the newly ingested data, allowing it to adapt to evolving market trends and maintain accuracy.

2. Enhanced Model Robustness and Feature Engineering:

- **Expanded Variable Set:** Increase the number of variables used in the valuation model to improve its accuracy and robustness. This will include:
    - **Detailed Vehicle Condition:** Incorporate subjective and objective measures of vehicle condition, such as:
        - Detailed inspection reports (e.g., mechanical, cosmetic).
        - Mileage consistency and maintenance records.
        - Photographic evidence of vehicle condition. Utilize computer vision to analyze images and produce a condition score for the vehicle
    - **Specific Engine Models:** Add detailed engine specifications as a key variable, recognizing the significant impact of engine type and performance on vehicle value.
    - **Trim Levels and Optional Features:** Include more granular data on trim levels and optional features to capture variations in vehicle value within the same make and model.
    - **Geographic Data:** Add more detailed geographic data to account for regional market variations.
- **Advanced Machine Learning Techniques:** Explore and implement advanced machine learning algorithms, such as ensemble methods or deep learning, to further improve the model's predictive accuracy.
- **Feature Importance Analysis:** Conduct regular feature importance analysis to identify the most influential variables and refine the model accordingly.

3. Enhanced User Interface and Visual Appeal:

- **Modern and Intuitive Design:** Redesign the website with a modern and visually appealing interface to enhance user engagement and navigation.
- **Interactive Data Visualizations:** Incorporate interactive data visualizations to provide users with a clear and intuitive understanding of the factors influencing vehicle valuations.
- **Personalized User Experience:** Implement personalized features, such as saved searches, vehicle watchlists, and customized valuation reports, to enhance the user experience.
- **High Quality Image and Video Integration:** Allow for the easy uploading and viewing of high-quality images and videos of the cars being sold.

-

Citations

[1] “2011 Toyota Prius for sale in Woodland Hills, CA,” Autotrader, https://www.autotrader.com/cars-for-sale/all-cars/2011/toyota/prius/woodland-hills-ca (accessed Mar. 11, 2025).