

Used Car Price Estimator Application Proposal

C964 – Computer Science Capstone
Western Governors University

Letter of Transmittal	3
Problem Summary	4
Application Benefits	4
Application Description.....	5
Data Description	5
Objectives and Hypothesis.....	5
Methodology.....	6
Funding Requirements.....	6
Data Precautions.....	7
Developer's Expertise	7
Project Proposal	9
Problem Statement.....	9
Customer Summary	9
Existing System Analysis.....	10
Data.....	10
Project Methodology	11
Project Outcomes	11
Implementation Plan	12
Evaluation Plan.....	12
Resources and Costs	12
Timeline and Milestones.....	14
Post-implementation Report	14
A Business Vision.....	14
Datasets	20
Data Product Code	24
Objective Verification	27
Effective Visualization and Reporting	28
Accuracy Analysis.....	29
Application Testing	30
Application Files	31
User Guide	32
Summation of Learning Experience	34
References	35

Letter of Transmittal

March 23rd, 2023

Mike Smith
Director of Sales
Acme Automotive Group

Subject: Proposal for a Used Car Price Estimation Application

Dear Mike Smith,

I am writing you today to present an innovative machine-learning solution that addresses a considerable problem faced by our organization as of late within the used car market. As you may be aware, accurately estimating used car prices can be considerably complex, with various factors including make, model, year, odometer readings, and more. The inability to effectively or efficiently estimate these prices leads to lost revenue due to low-profit margins and potential customer dissatisfaction via inflated prices or perceived lowball offers regarding our asset acquisitions.

To address the increasingly concerning issue, I propose developing a data-driven application that implements an advanced machine learning algorithm to estimate used car prices based on recent prices offered by private sellers and using our transaction data. This application, which I call the "Used Car Price Estimator (UCPE)," will enable our sales team to provide customers, both those who buy and sell with us, with more accurate and consistent pricing information, based upon up-to-date news and market trends—leading ultimately to increased customer satisfaction and higher revenue.

The benefits of implementing the Used Car Price Estimator include the following:


1. Improved pricing accuracy, reducing the risk of undervaluing or overpricing used cars.
2. Enhanced customer satisfaction due to more consistent and accurate pricing information.
3. Streamlined pricing processes for our sales team, allowing them to focus on other aspects of customer service.

As an experienced Data Engineer with a background in data science and machine learning, I am confident in my ability to lead the development and implementation of this solution. I have completed similar projects in the past, which have resulted in significant improvements in operational efficiency and revenue generation for the organizations involved. The estimated cost for developing and implementing this solution is \$60,334/month. The total includes the costs associated with cloud hosting, labor, software development, testing, and integration with our existing systems.

The Used Car Price Estimator application has the potential to significantly enhance our organization's ability to accurately price used cars, leading to increased customer satisfaction and revenue growth. I am eager to discuss this proposal further and address any questions or concerns you may have.

Thank you for considering this proposal, and I look forward to the opportunity to work together on this exciting project.

Sincerely,



Daniel LaForce
Data Engineer, Acme Automotive Group

Problem Summary

The Used Car Price Estimator project aims to develop an innovative application that will enable our organization to estimate the market value of used cars accurately and efficiently, considering various factors such as age, mileage, make, model, and condition. This application is necessary due to the increasing demand for used cars and the need to stay competitive by providing fair and accurate pricing for our customers. Due to the integrated circuit (chip) shortage worldwide, car prices have become volatile and unpredictable, causing increased asset acquisition issues and disgruntled customers.

By implementing this solution into our Sales and Acquisition team, our organization will be better equipped to make informed pricing decisions when purchasing and selling our inventory. This will lead to increased customer satisfaction and revenue growth.

Application Benefits

The proposed solution will address several critical business needs for our organization, including:

1. **Improved pricing accuracy:** By leveraging advanced data analysis techniques, the application will enable us to accurately estimate the market value of used cars, resulting in more competitive pricing and increased sales.
2. **Continually updated:** Using data scraping techniques, we can produce periodically updated datasets that will evolve with the marketplace. Allowing us to adjust our purchasing and selling prices as the integrated chip shortage increases and decreases and consumer trends change.
3. **Enhanced customer satisfaction:** Providing accurate and fair pricing for used cars will strengthen our reputation in the market and improve customer satisfaction.
4. **Streamlined operations:** The application will automate used car pricing, reducing manual effort and increasing operational efficiency through excessive input prompts or laggy third-party car valuation services.

Application Description

The Used Car Price Estimator application will utilize machine learning algorithms to analyze recent sales data and determine the most relevant factors affecting the market value of used cars. The application will generate accurate price estimates by processing this data, considering variables such as make, model, age, mileage, and in the next iteration, the condition and add-on packages of the vehicle. The application will also incorporate an intuitive user interface, enabling our team members to quickly input relevant information and receive instant price estimates.

Data Description

We will source the raw data for this project from various channels, including our organization's historical sales records, public datasets from websites like Kaggle.com, and reputable third-party sources, and scrape our data from online marketplaces such as eBay and Craigslist. The data will be structured, containing nominal (e.g., make, model, add-ons) and quantitative (e.g., mileage, age, condition) variables. The dependent variable in this project will be the market value of the used cars, while the independent variables will include factors such as age, mileage, make, model, condition, and add-ons.

We will carefully analyze the data for any anomalies, such as outliers or missing values and invalid or obvious odometer errors, and address these issues accordingly to ensure the accuracy of our predictions. While we will hardcode much of the pre-processing steps needed to provide the machine-learning algorithms with good clean data, we will visually inspect the data while this process becomes automated.

Objectives and Hypothesis

The primary objectives of this project are to:

1. Develop a reliable used car price estimation application.
2. Improve our organization's pricing accuracy and competitiveness within the used car market.
3. Increase customer satisfaction by offering fair and transparent pricing.

4. Identify market trends as they start to happen.

The application will significantly improve our pricing accuracy and trend knowledge and, in turn, lead to increased sales and customer satisfaction.

Methodology

We will adopt an agile development methodology for this project, as it allows for flexibility, iterative improvements, and effective collaboration between team members. We will divide the project into several phases, including:

1. Design: Defining project requirements, identifying data sources, and selecting the most appropriate machine learning algorithms.
2. Implementation: Developing the application, incorporating the selected algorithms, and creating an intuitive user interface.
3. Testing: Evaluating the application's performance and accuracy by comparing its predictions against market values.
4. Deployment: Integrating the application into our organization's existing processes and systems.
5. Continuous Improvement: Regularly updating the application based on user feedback and the latest market trends.

Funding Requirements

The initial funding we require for this project is an estimated \$181,002 (\$60,334/month for the next three months), which will cover the costs of one calendar quarter, including the following:

1. Development environment setup, including necessary cloud/hardware and software.
2. Personnel expenses for DevOps, software/data engineering, testing, and deployment.
3. Licensing fees for any proprietary tools, frameworks, or datasets.
4. Overhead expenses for additional office space and utilities.

Data Precautions

Though not a massive concern for the type of datasets we are collecting, we will be cognizant of the potential of sensitive or protected data in which we will hard code safeguards into our data pre-processing. As part of this goal to protect the public and our customers, we will adhere to the following guidelines:

1. Ensuring compliance with all applicable data protection regulations.
2. Anonymizing any personally identifiable information (PII) before analysis.
3. Implementing strict access controls and encryption measures to protect the integrity and confidentiality of the data.

Developer's Expertise

As a data engineer, I bring a strong background in computer science, data analysis, and machine learning. My academic training includes a degree in Computer Science from Western Governor's University, where I studied data analytics and artificial intelligence. My professional experience includes working on similar projects within my higher education courses and similar innovative proposals for other organizations, giving me valuable insights into the challenges and best practices of developing data-driven solutions.

The combination of my academic and professional background aligns well with the needs of this project, positioning me to deliver a high-quality, effective solution that will benefit our organization. My expertise in data analysis, machine learning, and software development will ensure the successful development and implementation of the Used Car Price Estimator application. Furthermore, my experience working with cross-functional teams and managing projects from inception to completion will contribute to the smooth execution of the project.

The Used Car Price Estimator project is a valuable investment for our organization, as it addresses critical business needs and promises to deliver significant benefits in terms of pricing accuracy, customer satisfaction, trend identification, and operational efficiency. My computer science and data

analytics expertise and relevant professional experience will ensure this innovative solution's successful development and implementation.

Project Proposal

Problem Statement

The automotive industry is highly competitive, with numerous manufacturers offering various car models to cater to multiple customer preferences. Car dealerships must understand which features contribute to the success of a car model in terms of sales. The successful dealership should also be able to identify trends as they happen and before the competition can capitalize upon them. With the uncertainty of the integrated chip market, the automotive industry has been shaken up by the speed at which circumstances of changed. Many needed to adjust to the new trends before it was too late and, without inventory, shut their doors.

Customer Summary

Our clientele includes the public, both car buyers and car sellers, and even other dealerships. We need a technology solution that will give us a leading edge to know what price we should buy a car to sell for a price that will produce the biggest bang for the buck while helping us retain a steady inventory. As prices soar due to chip shortages, yesterday's excellent profit margin is today's empty car lot. A strategic price with tomorrow in mind is needed. But we need an inside look into the future via data trends. If we can offer potential customers a buy offer that they accept and appreciate, they are more likely to walk away feeling as if they were treated fairly and referring their family and friends. We have lost many potential clients due to making lowball offers that, in hindsight, would have paid quite a handsome profit margin with no more than a month of their car idling in our parking lot.

Existing System Analysis

Acme Automotive Group uses manual research. Expert opinions and standard industry tools to determine used car prices. This approach is time-consuming, prone to errors, and subject to biases. More importantly, our competition relies upon this same information, which means we have no unique data at our disposal. Our solution will automate this process and provide a more objective and data-driven approach to price estimation. We thoughtfully consider the problem of using only the same tools our competition uses. We know what price they want to buy at and the price they want to sell at as we use the same automotive industry tools to determine this. But there is no leading edge in doing what others do without creating our technological sales niche.

Data

Raw Data Set: The raw data set will consist of recent used car sales data, including features such as make, model, year, mileage, condition, and add-ons, as well as the final sale price. We can consider incorporating many additional features within the dataset in the future.

Data Collection, Processing, and Management: We will collect data from various sources, including customer records, public sales data, and industry databases. Data pre-processing steps will include cleaning, normalization, and feature engineering. The data will be stored and managed in a secure, scalable cloud-based database.

Data Anomalies: We will handle anomalies by identifying and removing outliers manually and through code functions, imputing missing values, and performing robust statistical analyses to ensure accurate predictions.

Project Methodology

We will use the Agile development methodology to develop and deploy our application. Agile allows for iterative improvements and flexibility in responding to changing requirements, making it an appropriate choice for this project.

Phases of the Agile Methodology:

1. Requirements Gathering: We will work closely with our sales representatives to define the application's features, goals, and constraints.
2. Design: We will create wireframes, mockups, and system architecture diagrams to illustrate the final application's user interface and backend components.
3. Development: We will build a standalone application using Python and frameworks, adhering to best practices for code quality, security, and performance.
4. Testing: We will perform unit, integration, and system tests to ensure the application functions correctly and meets our users' requirements.
5. Deployment: We will deploy the application to a cloud-based server and integrate it with our branches' existing systems.
6. Maintenance: We will monitor the application's performance, address any issues, and make necessary improvements.

Project Outcomes

Deliverables will include:

- The finished Used Car Price Estimator application will be securely deployed to the cloud by our web team.
- Company-wide deployment to all individual branches.
- Employee training on its strategic implementation and how to use it daily.

- Comprehensive documentation, including a user guide and technical specifications.

Implementation Plan

The implementation plan will consist of the following:

1. General Strategy: We will work closely with our sales representatives to ensure a smooth transition from their current system to our application.
2. Phases of the Rollout: We will divide the rollout into stages, including a pilot phase with limited users and a full-scale deployment to all relevant personnel.
3. Dependencies: We will identify and manage dependencies, such as integration with our employee's existing systems, data migration, and user training.
4. Testing and Distribution: The application will undergo rigorous testing to ensure its accuracy and reliability. We will deploy company-wide using the Amazon Web Services platform as the host and add necessary shortcuts to all appropriate company devices.

Evaluation Plan

Verification Methods: We will employ various verification methods at each stage of development, including code reviews, unit tests, and automated testing tools.

Validation Method: Upon project completion, we will use real-world data and performance metrics to validate the accuracy and effectiveness of the application.

Resources and Costs

Cloud and Hardware Costs: Server and infrastructure costs for hosting the application.

AWS Lambda: \$0.20 per million requests (\$200 for 1 million requests per month)

AWS API Gateway: \$3.50 per million requests (\$350 for 100 million requests per month)

AWS RDS (for the database): \$0.025 per GB-month of storage (\$25 for 1 TB-month of storage)

AWS S3 (for storage): \$0.023 per GB-month of storage (\$23 for 1 TB-month of storage)

Total cloud and hardware costs: \$598 per month

Software Costs: Licensing fees for programming tools, frameworks, and cloud-based services.

Programming tools (e.g., PyCharm, VS Code): \$0 - most IDEs have a free version

Frameworks (e.g., Flask, Pandas): \$0 - most Python frameworks are open source

Cloud-based services (e.g., AWS Lambda, API Gateway, RDS, S3): as calculated above

Total software costs: \$0 per month (Cloud already included above)

Labor Time and Costs: Development, testing, and deployment team salaries and overhead expenses.

Development team: 2 developers, each making \$100,000 per year (\$16,667 per month)

Testing team: 1 tester making \$75,000 per year (\$12,500 per month)

Deployment team: 1 DevOps engineer making \$120,000 per year (\$20,000 per month)

Overhead expenses (e.g., office space, utilities): \$10,000 per month

Total labor and overhead costs: \$59,167 per month

Environment Costs: Expenses associated with deploying, hosting, and maintaining the application.

Monitoring tools (e.g., New Relic): \$149 per month

Security tools (e.g., AWS Shield): \$300 per month

Domain registration and SSL certificate: \$20 per month

Other miscellaneous expenses (e.g., backups, disaster recovery): \$100 per month

Total environment costs: \$569 per month

Grand Total: \$60,334 per month

Timeline and Milestones

Sprint	Start	End
1: Requirements Gathering	May 1, 2023	March 15, 2023
2: Design	March 15, 2023	June 12, 2023
3: Development	June 12, 2023	August 21, 2023
4: Testing	August 21, 2023	September 18, 2023
5: Deployment	September 18, 2023	October 2, 2023
6: Maintenance	October 2, 2023	Ongoing

Please note that these dates are approximate and may be subject to change based on project progress, our corporate requirements, and any unforeseen challenges. We will maintain progress updates and communication with leadership throughout the project to ensure alignment with expectations and address any issues.

Post-implementation Report

A Business Vision

Problem:

The used car market is vast and complex, with many factors influencing a vehicle's pricing.

Purchasing a used car can be challenging for customers and dealers due to the difficulty in accurately estimating a fair price for a specific vehicle, especially in today's volatile market. Determining what price to sell can be challenging to avoid holding onto inventory for many months, often worse, not

having inventory to sell. The difficulty resides in used car prices that depend on its make, model, year, mileage, condition, add-ons, market, and often demand. The car market is very dynamic, with prices constantly changing due to fluctuations in supply and demand. As a result, potential buyers often need help finding a suitable car at a fair price, while sellers may have difficulty determining an appropriate asking price for their vehicle, so they sell at a controlled yet steady pace.

Application Solution:

Our application addresses this problem by giving users an estimated price for a used car's valuation based on the input factors (i.e., make, model, year, and odometer reading). It utilizes a machine learning model trained on a comprehensive dataset of used car transactions, enabling users to obtain a data-driven price estimate for their vehicles. Graphical analysis of tens of thousands of recent vehicle ads assists the algorithm and, in turn, the user in determining the demand for a particular make, model, and vehicle year.

I aimed to develop a machine-learning application that could predict the market value of a used car based on its features, such as the make, model, year, and mileage. By providing a user-friendly interface for users to input these features, our application helps them estimate a fair price for a specific used car. This tool benefits both buyers and sellers in the used car market by offering a data-driven approach to pricing, which can improve the overall efficiency and transparency of transactions.

The tool also provides a glimpse inside the window of the future in predicting trends. Our competition has only begun to consider ways to take advantage of the increased development of machine-learning algorithms and data scraping of datasets.

User Experience:

Users can interact with the application through a user-friendly interface that prompts them to input the relevant information (make, model, year, and odometer reading). Upon submitting the data, the application processes the input and returns an estimated price for the used vehicle. Users can utilize this estimate to make informed decisions when buying or selling used cars.

Several different graphs (e.g., scatterplot, bar graph, histogram, heatmap) show you how features relate to price and help identify other trends by viewing the data in different ways.

One of the most exciting is the Feature Importance graphs to see what features matter most for any make available on the market. In the future, we will also implement graphical representations of how prices fluctuate weekly, month to month, and year to year.

Example

Price Prediction:

A user wants to determine the estimated price for a 2018 Honda Civic with 45,000 miles on the odometer. They input the required information into the application, which processes the data and returns an estimated price of \$14,973.48.

Figure 5

Example of Histogram Price Distribution for All Makes

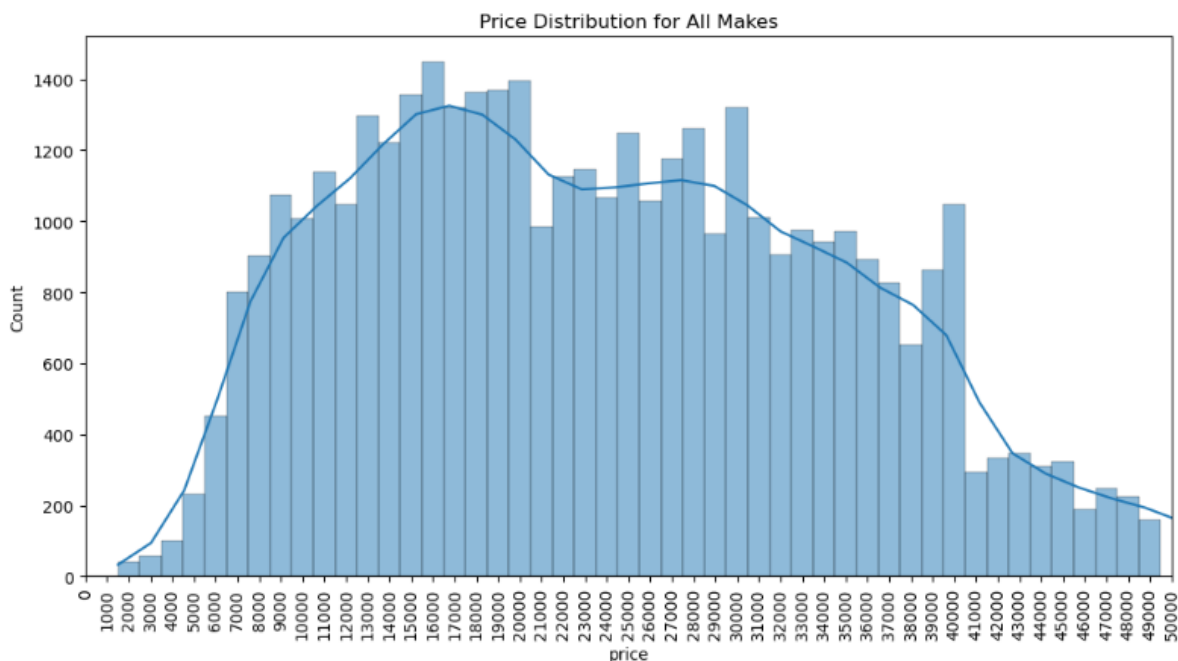


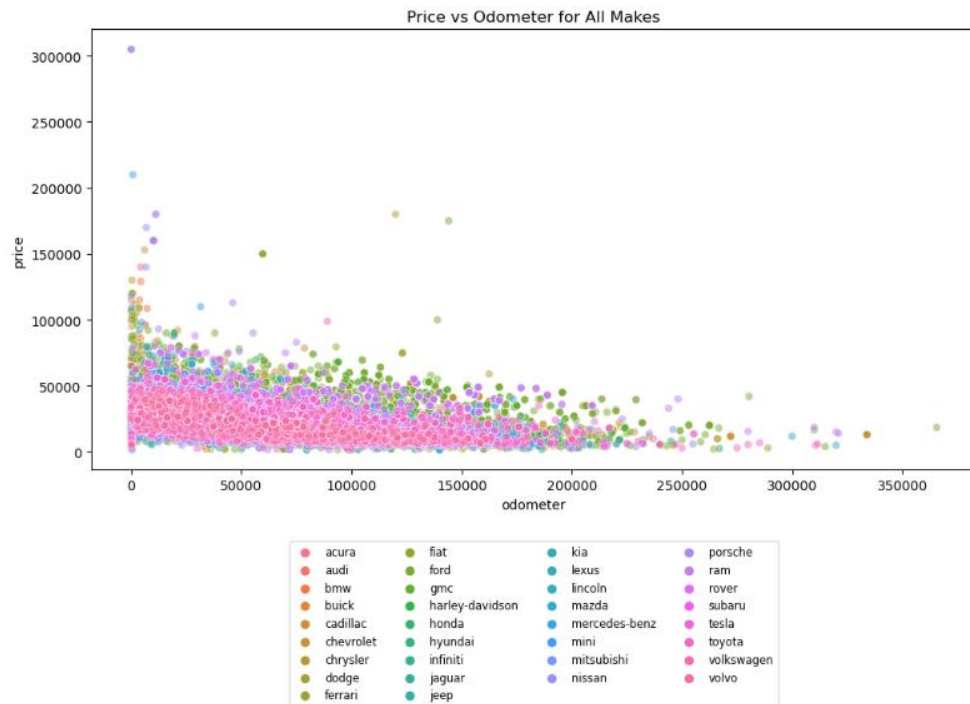
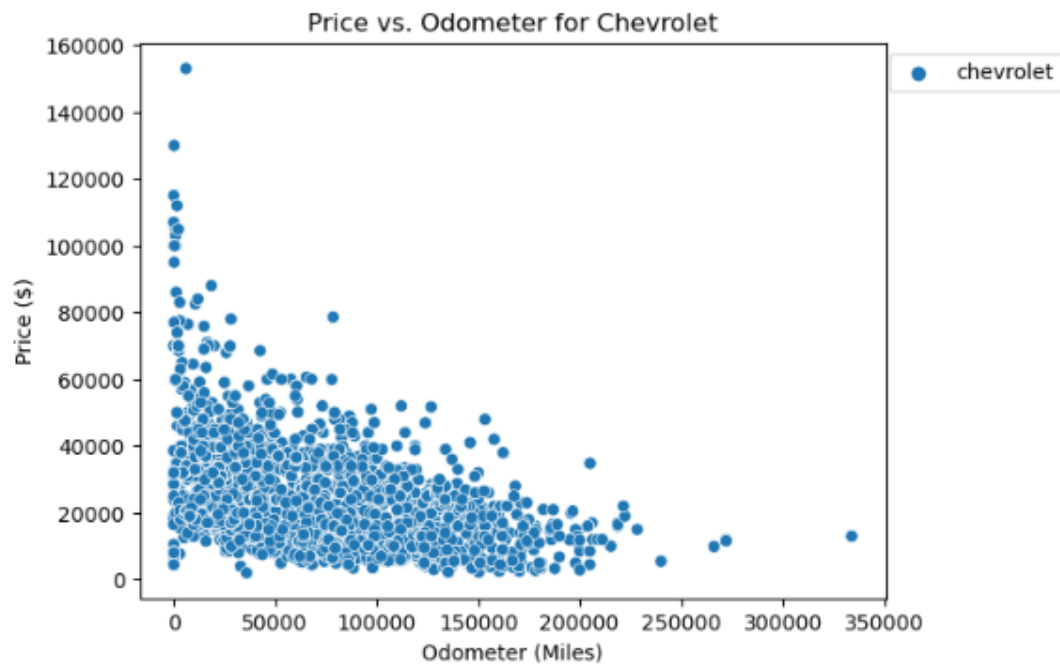
Figure 2**Example of Scatterplot Price vs. Odometer for All Makes****Figure 3****Example of Scatterplot Price vs. Odometer for Specific Make**

Figure 4

Example of Heatmap by Year and Make

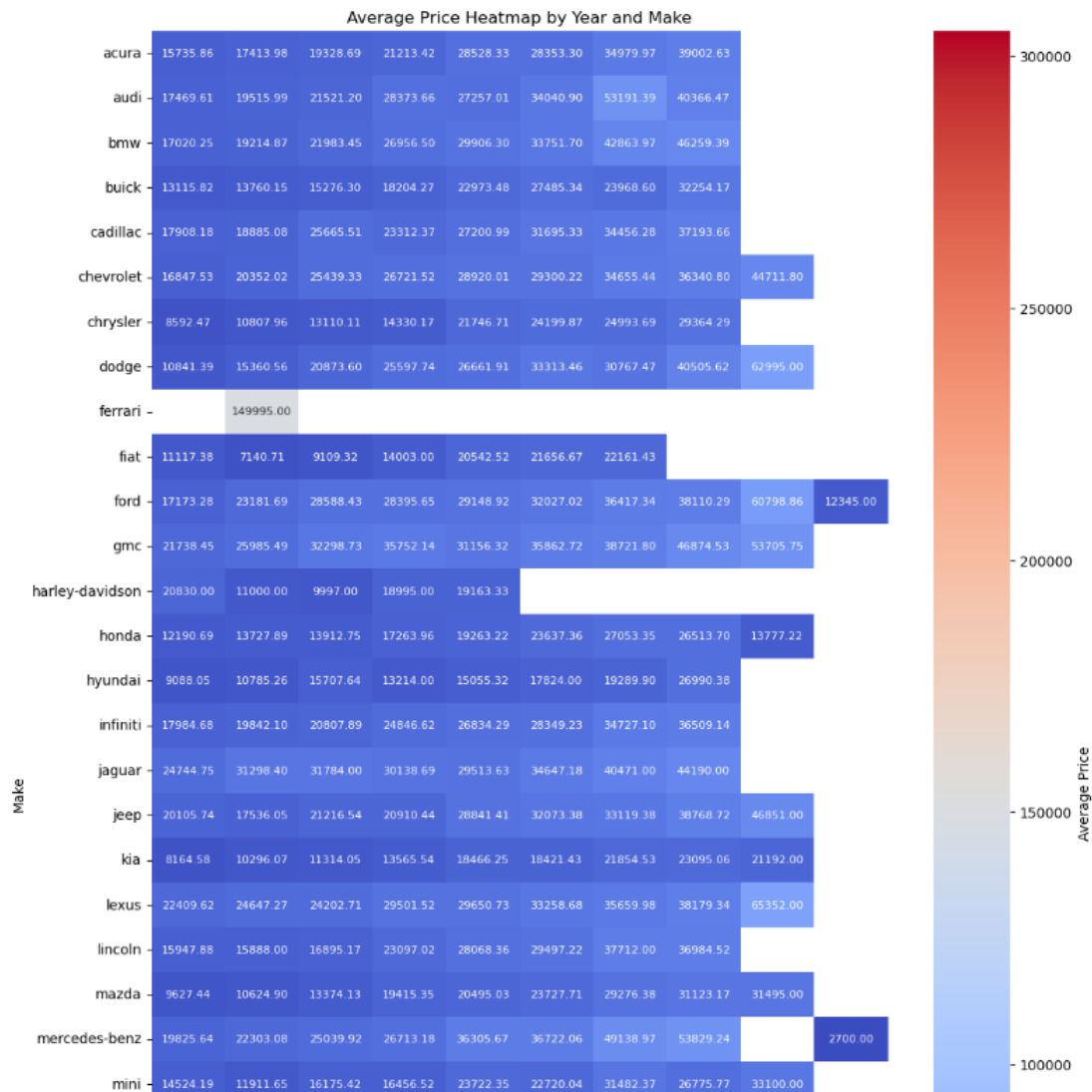


Figure 5

Example of Bar Graph Average Price per Make

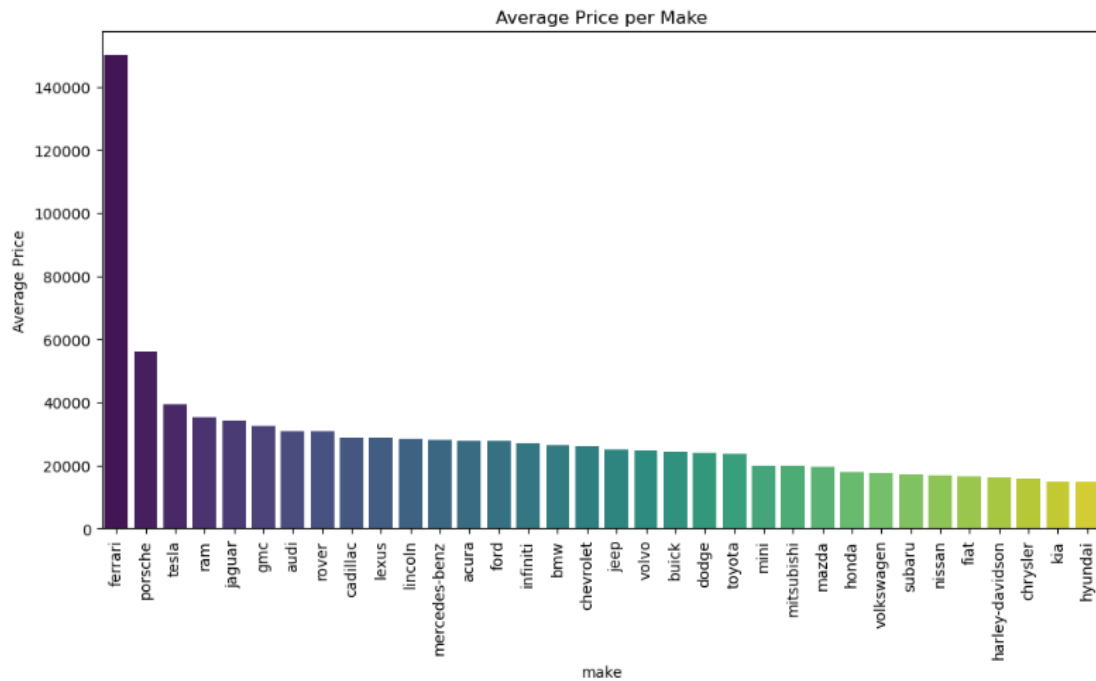
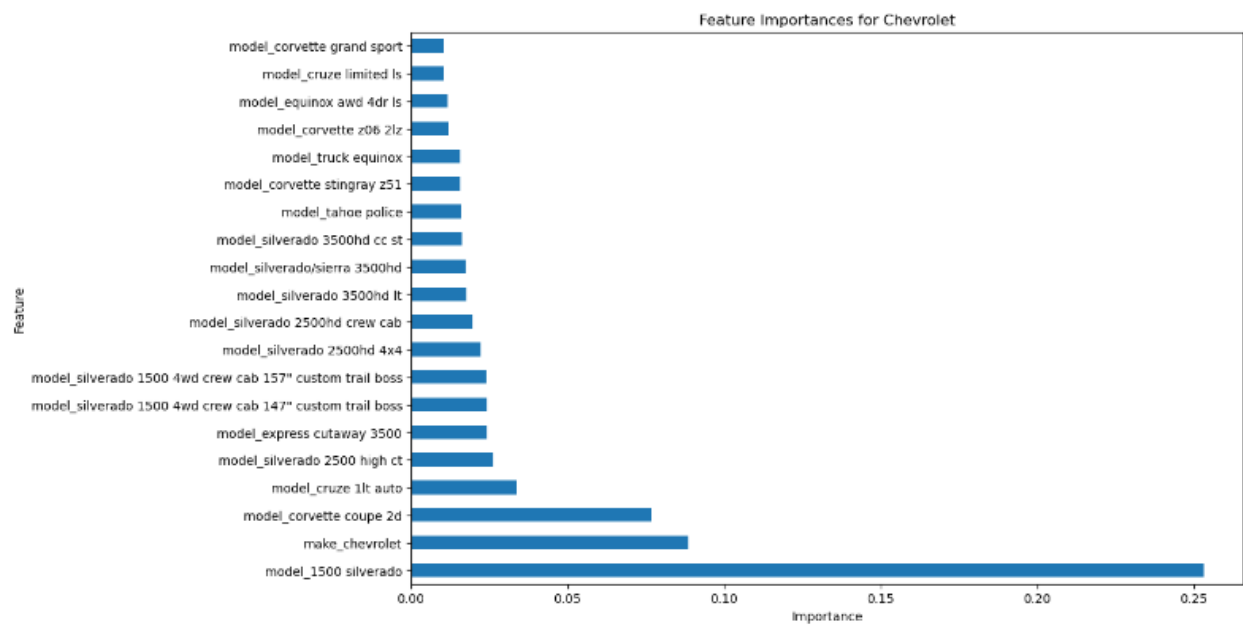


Figure 6

Example of Feature Importances for Specific Make



Datasets

Raw and Processed Data:

I sourced the raw data for this project from a Kaggle dataset of used car transactions sourced from Craigslist, which included but was not limited to information on the make, model, year, odometer reading, and selling price of each vehicle. The processed data is a cleaned and transformed version of the raw data, ensuring I can address any inconsistencies or missing values and that the data is suitable for input into the machine learning model.

Data Processing:

The raw data has been pre-processed by performing the following steps:

- Removing any duplicate or irrelevant entries.
- Identifying and eliminating outliers to prevent them from damaging the models' effectiveness.
- I am handling missing values by imputing them using an Iterative Imputer.
- One-hot encoding categorical features (make and model) and normalizing numerical features (year and odometer) using MinMaxScaler.
- I trimmed off cities west of Ohio for the initial prototype to reduce training complexity, but I will expand to allow the data to be viewed and processed in different markets.
- I then deleted the "description" column contents to save space.
- I then normalized model names.
- I deleted entries without model or suspected false odometer readings.
- I filtered out years older than 2013 as we are only interested in ten years or newer vehicles.

Example of Raw Data:

id,URL,region,region_url,price,year,manufacturer,model,condition,cylinders,fuel,odometer,title_status,transmission,VIN,drive,size,type,paint_color,image_url,description,county,state,lat,long,posting_date

7316814884,https://auburn.craigslist.org/ctd/d/auburn-university-2014-gmc-sierra-

1500/7316814884.html,auburn,https://auburn.craigslist.org,33590,2014,gmc,sierra 1500 crew cab
slt,good,8

cylinders,gas,57923,clean,other,3GTP1VEC4EG551563,,,pickup,white,https://images.craigslist.org/0
0R0R_lwWjXSEWNa7z_0x20oM_600x450.jpg,"Carvana is the safer way to buy a car During these
uncertain times, Carvana is dedicated to ensuring safety for all of our customers. In addition to our
100% online shopping and selling experience that allows all customers to buy and trade their cars
without ever leaving the safety of their house, we're providing touchless delivery that makes all
aspects of our process even safer. Now, you can get the car you want, and trade in your old one, while
avoiding person-to-person contact with our friendly advocates. There are some things that can't be put
off. And if buying a car is one of them, know that we're doing everything we can to keep you keep
moving while continuing to put your health safety, and happiness first. Vehicle Stock# 2000909557 📱

Want to instantly check this car's availability? Call us at 334-758-9176 Just text that stock number to
855-976-4304 or head to <http://www.carvanaauto.com/7171237-74502> and plug it into the search
bar! Get PRE-QUALIFIED for your auto loan in 2 minutes - no hit to your

credit: <http://finance.carvanaauto.com/7171237-74502> Looking for more cars like this one? We have
63 GMC Sierra 1500 Crew Cab in stock for as low as \$23990! Why buy with Carvana? We have one
standard: the highest. Take a look at just some of the qualifications all of our cars must meet before
we list them. 150-POINT INSPECTION: We put each vehicle through a 150-point inspection so that
you can be 100% confident in its quality and safety. See everything that goes into our inspections
at: <http://www.carvanaauto.com/7171237-74502> NO REPORTED ACCIDENTS: We do not sell cars
that have been in a reported accident or have a frame or structural damage. 7 DAY TEST OWN
MONEY BACK GUARANTEE: Every Carvana car comes with a 7-day money-back guarantee.

Why? It takes more than 15-minutes to make a decision on your next car. Learn more about test owning at <http://about.carvanaauto.com>

FLEXIBLE FINANCING, TRADE INS WELCOME: We're all about real-time financing without the middle man. Need financing? Pick a combination of down and monthly payments that work for you. Have a trade-in? We'll give you a value in 2 minutes. Check out everything about our financing at: <http://finance.carvanaauto.com/7171237-74502>

COST SAVINGS: Carvana's business model has fewer expenses and no bloated fees compared to your local dealership. See how much we can save you at <http://about.carvanaauto.com>

PREMIUM DETAIL: We go the extra mile so that your car is looking as good as new. There are a lot of specifics that we won't list here (we wash, clean, buff, paint, polish, wax, seal), but trust us that when your car arrives, it's going to look sweet.

Vehicle Info for Stock# 2000909557 Trim: SLT Pickup 4D 5 3/4 ft pickup
 Mileage: 57k miles Exterior Color: White Interior Color: Lt. Brown Engine: EcoTec3 5.3L Flex Fuel V8 355hp 383ft. lbs. Drive: Two Wheel Drive Transmission: VIN: 3GTP1VEC4EG551563

Dealer Disclosure: Price excludes tax, title, and registration (which we handle for you).

Disclaimer: You agree that by providing your phone number, Carvana, or Carvana's authorized representatives*, may call and/or send text messages (including by using equipment to automatically dial telephone numbers) about your interest in a purchase, for marketing/sales purposes, or for any other servicing or informational purpose related to your account. You do not have to consent to receiving calls or texts to purchase from Carvana. While every reasonable effort is made to ensure the accuracy of the information for this GMC Sierra 1500 Crew Cab, we are not responsible for any errors or omissions contained in this ad. Please verify any information in question with Carvana at 334-758-9176

*Including, but not limited to, Bridgecrest Credit Company, GO Financial and SilverRock Automotive.

GMC *Sierra* *1500* *Crew* *Cab* *Base* *GMC* *Sierra* *1500* *Crew* *Cab* *SLE* *GMC* *Sierra* *1500* *Crew* *Cab* *SLT* *GMC* *Sierra* *1500* *Crew* *Cab* *Denali* *GMC* *Sierra* *1500* *Crew* *Cab* *Work* *Truck* *4x2* *GMC* *Sierra* *1500* *Crew* *Cab* *4x4* *Crew* *Cab* *GMC* *Sierra* *1500* *Crew* *Cab* *Regular* *Cab* *Extended* *Cab* *Truck* 2022 2021 2020 2019 2018 2017 2016 2015 2014 2013 2012 2011 2010 2009 2008 2007 2006 2005 2004 2003 2002 2001 2000

22 21 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00",,al,32.59,-
85.48,2021-05-04T12:31:18-0500

Example of Processed Data:

id,URL,region,region_url,price,year,make,model,condition,cylinders,fuel,odometer,title_status,trans
mission,VIN,drive,size,type,paint_color,image_url,description,county,state,lat,long,posting_date
7313788270,https://hartford.craigslist.org/ctd/d/dont-miss-out-on-our-2013-fiat-500-
trim/7313788270.html,hartford,https://hartford.craigslist.org,11999,2013,fiat,500,good,4
cylinders,gas,29861,clean>manual,3C3CFFJH9DT665883,fwd,,other,red,https://images.craigslist.org/
00Z0Z_l8tOKNyAWwnz_0eu0aS_600x450.jpg,NA,,ct,,,2021-04-28T09:58:03-0400

Dataset access: [https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-
dataImplementation](https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-dataImplementation)

Data Product Code

Processing Raw Data:

The code processing begins by importing and processing raw data vehicle listings from Craigslist posts and performing the previously mentioned pre-processing steps, including cleaning, handling missing values, removing outliers and dreck data, and aggregating data to create a comprehensive dataset for analysis.

Descriptive Methods and Visualizations:

The code incorporates and utilizes various descriptive methods and visualizations to present and analyze the data in a user-friendly and valuable way. These techniques aim to provide insights into the data and assess the effectiveness of the machine learning model.

1. **Machine-learning Statistics:** The code calculates summary statistics, such as training and testing scores, to evaluate the model's performance. These scores help users understand how well the model generalizes to new data and whether it is underfitting or overfitting. Grid search statistics show what the machine-learning algorithm does behind the scenes to create models using hyperparameter tuning. This technique is used to fine-tune the hyperparameters of any given model. After the best hyperparameters, the UI displays:

max_depth – The tree depth determined to avoid overfitting or underfitting.

min_samples_leaf – The minimum number of samples required within a leaf node to prevent overfitting.

min_samples_split – The minimum number of pieces needed to split an internal node. This also helps reduce overfitting.

n_estimators – The number of trees in the random forest model. A higher number can be better but at the expense of computation time and memory requirements.

Training Score - Measures accuracy the model has learned from the training data and is calculated using the model's score method. It inputs the training features (`X_train_scaled`) and the corresponding target values (`y_train`). In the case of `RandomForestRegressor`, the default scoring metric is the coefficient of determination (R^2). This ranges from 0 to 1. The higher the R^2 value is, the better the model is perceived to be performing. A value of one equates to the interpretation of the model explaining 100% of the variation in the target variable. Using this information, we can determine if the models we create from the data should be considered helpful in predicting future samples.

Test Score - Measures how well the model generalizes to new and unseen information. The test score is calculated similarly to the training score but with the testing features (`X_test_scaled`) and the corresponding target values (`y_test`) as inputs. Like the training score, the test score is also an R^2 value, and a higher value indicates better performance. This score should be very close to the training score. This would suggest that the model has learned to generalize the patterns within the data while not overfitting the data.

By comparing the training and testing, we can interpret the effectiveness of the model and whether the model is underfitting, overfitting, or how well it is generalizing to newly input data from properly formatted datasets. This provides valuable insights into refining and improving the model as we advance. Incorporating these non-descriptive methods is exceptionally beneficial for predicting used car prices accurately and efficiently, making it an essential part of our proposal.

2. **Histogram:** The included histogram displays the distribution of selling prices in the dataset.

This chart allows users to identify typical selling prices quickly. Any features do not separate this, and its purpose is purely to understand what most vehicles are being sold for.

3. **Bar Chart:** A bar chart is employed to showcase the average selling price for each vehicle "make" within the entire data set, unlike some of the plot charts. This graphical representation of the entire dataset makes it easy to compare the prices across different makes and to identify which ones have higher or lower average pricing trends.

4. **Heatmap:** The generated heatmap illustrates the average price for each vehicle “make” over the years, starting from 2013. The heatmap uses color gradients to represent the values, making it simple to spot trends or changes in average prices over time. Great to visualize the depreciation of valuation on a year-by-year basis.
5. **Scatterplots:** There are two different types of scatterplots within the interface. Both of which visualizes the relationships between mileage and price. One graph for all makes, and the other displays when you select a particular "Make" dataset. These plots allow users to investigate potential correlations or trends between odometer and price, which could be important for understanding how mileage influences vehicle prices even on a make-by-make basis.
6. **Features Importance Graph:** This is available for each "make" dataset displaying the distribution of vehicle prices, essential features, and significant factors. It will also help us refine our model in the future and helps to optimize the user experience by giving the sales representative essential elements when estimating the price of a used car.

Non-Descriptive Methods:

The Used Car Price Estimator utilizes machine learning techniques, specifically Random Forest Regression models, to help predict used car prices based on various features. It also incorporates a grid search algorithm to find the best hyperparameters, fine-tuning the machine learning model to predict prices better. These two algorithms allow for a more accurate estimation of vehicle prices based on current market trends.

This price prediction prototype utilizes Python libraries such as Pandas, NumPy, and Scikit-Learn to analyze a Craigslist dataset containing used car data and create a machine-learning model to predict used car prices. The code loads the dataset, pre-processes the data, trains a random forest regressor, and makes interactive visualizations to explore and understand the data. The code performs exploratory data analysis and displays visualizations of the used car data, then defines a function to predict the car price based on user inputs (year, make, model, and odometer).

Regression models are a highly effective machine learning technique for this application as they can accurately estimate vehicle prices based on various features, considering current market trends. In addition, incorporating a grid search to find the best hyperparameters is highly beneficial for fine-tuning the machine learning model, resulting in even more accurate price estimates.

Objective Verification

The Used Car Price Proposal's objective was to predict used car prices accurately using machine learning techniques. The goal was to utilize regression models by fine-tuning them with hyperparameters using a grid search and to develop a highly accurate machine-learning model that could provide reliable estimates of used car prices. It would use features such as make, model, year, and odometer reading to look into future trends by visualizing the data differently.

Our model has analyzed a large Craigslist used car sales dataset utilizing Python libraries such as Pandas, NumPy, and Scikit-Learn. It uses these to create a prediction model and User Interface that includes exploratory data analysis, interactive visualizations, and statistics that help us understand the data, ensuring we use the best techniques to continue refining and developing the model's accuracy.

As noted, the test and training scores confirm that the project's objective was achieved. While the score was almost too good in some cases with some "Makes," we have a great foundation that we can further use to build upon and train with even more enormous datasets. The model achieved very high scores in both the training and testing phases, indicating that it learned the underlying patterns in the data. However, without overfitting, it is expected to accurately predict the prices of used cars based on various features.

Therefore, the objective we set out to achieve in developing machine learning techniques using a regression model and hyperparameter fine-tuning using a grid to predict used car prices has been completed.

Effective Visualization and Reporting

Data exploration

The datasets have been examined and analyzed using descriptive methods such as summary statistics, distribution plots, and scatter plots to understand the relationships between different features and vehicle prices. This process helped identify potential correlations, outliers, and trends in the data, which informed the selection of appropriate regression models and pre-processing techniques.

Data analysis

Data analysis involved visualizations to compare different regression models' performance and identify improvement areas. Visualizations such as residual plots, actual vs. predicted value plots, and importance features provided insights into model performance, enabling us to refine the models and optimize their predictions. The descriptive methods also helped reinforce the predicted price accuracy by using the visual representations of the dataset.

Data summary

Data summary methods, including summary statistics and correlation matrix/heatmap, were employed to condense the information from the dataset and provide a comprehensive view of the data. These summaries help identify critical features and relationships within the data, which informed our feature selection and engineering processes and improved the efficiency and effectiveness of our non-descriptive methods. We have a good idea of what we want to implement to continue enhancing our model's prediction abilities.

Accuracy Analysis

Scatter plot: A Scatterplot to visualize the "Price" vs. "Odometer" for all makes.

Histogram: A histogram to visualize the price distribution for all "makes" allows us to see the most common prices for vehicles consumers and other dealers are advertising their cars at.

Bar plot: A bar plot represents the average price per make, allowing us to visualize the most expensive vehicle makes.

Heatmap: I included a heatmap to visualize the correlation matrix, allowing us to identify the most relevant features for predicting vehicle prices. Specifically, we wanted to see how the "Make" and "Year" correlated to "Price."

Feature Importances graph: The Feature Importances graph enables users to identify the key factors that contribute to the success of a car model in terms of sales. It highlights which car attributes have the most decisive influence on consumer preferences, allowing us to focus on optimizing our purchases for better conversion. Additionally, by understanding the relative importance of each feature, users can make more informed decisions regarding feature selection, model tuning, or even when interpreting the model results. In turn, this can lead to more accurate and reliable predictions, enhancing the overall performance and effectiveness of the machine learning model.

Using these descriptive methods and visualizations, we could iteratively refine our non-descriptive methods, ultimately developing a highly accurate and reliable vehicle price prediction model.

Accuracy Analysis

The metric to assess my model is R-Squared or R^2 , which ranges from zero to one, with one being a perfect fit. The higher the score, the better the model is at differences within the data.

The code first prepares the data by splitting it into two sets: training and test set. It gives 80% of the samples to exercise and 20% for testing the model. Using the `train_test_split` function from the `sklearn` library, the data is pre-processed, encoded, and scaled. This library's score function uses the R^2 scoring algorithm to determine the model's training and test scores.

The method for determining the accuracy of the non-descriptive way I utilized within this application is as follows:

- 1) Load dataset

- 2) Split dataset into training and testing sets
- 3) Standardize the features using the StandardScaler library and function and transform and fit the data
- 4) Train the Random Forest Regressor model
- 5) Predict the target values for both training sets
- 6) Calculate the R^2 scores for both sets

The dataset determines the accuracy of the method. Some datasets are ideal for this Random Forest Regression, while other machine-learning algorithms would better suit others. In this case, it is rare for any of the datasets this tool uses to have scored under .8, and most are in the mid .90s making this an excellent fit for the data.

Application Testing

This application created a Pandas DataFrame with the feature data. Then the dataset was split into training and testing sets. Eighty percent of the data went to the training set and twenty percent to the testing set. Twenty percent was reserved for testing the performance of the model. These sets were then standardized using 'StandardScaler' from 'sklearn. Pre-processing' module. This helped ensure that the features were all on a similar scale, which helped the performance of the algorithm. Then the model was trained using RandomForestRegressor with the standardized data with one hundred estimators or decision trees. I then defined a set of hyperparameters for the model to be used in the grid search. I then instantiated the 'GridSearchCV' class with the RandomForestRegressor, the hyperparameters, and a 5-fold cross-validation. I then fit the grid search objects on the training dataset using 'grid_search.fit()'. The best hyperparameters were found using 'grid_search.best_params_'. Using the 'grid_search.best_estimator_' to calculate the R^2 score was performed on both training and test models utilizing the 'score()' function.

Application Files

To execute this tool on a Windows 10 machine, the following files and libraries are required:

Jupyter Notebook: The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It is required to run our Python code and create interactive visualizations.

Python: Python is a high-level programming language that is used for a wide range of applications, including machine learning. Our code was written in Python and required a Python environment to execute.

Python libraries: The following Python libraries are required to execute our code:

Pandas: Pandas is a fast, robust, flexible, easy-to-use open-source data analysis and manipulation library.

NumPy: NumPy is a fundamental package for scientific computing with Python, supporting arrays and matrices.

Scikit-Learn: Scikit-Learn is a machine-learning library for Python that provides simple and efficient data mining and analysis tools.

Dataset: Our code requires either a dataset containing used car data to train the machine-learning model and provide accurate price estimates or the appropriately trained model files. Both of which are included.

The files in my submission are organized hierarchically as follows:

Main folder:

This folder contains the “Used Car Price Estimator Prototype as a Jupyter Notebook file (in .ipynb format).

Subfolders:

data – contains the datasets broken down from the entire dataset by "make." These are broken down to make the training process quicker as you only train the "make" you request an estimate from.

Models– Similar to the data directory, the model has been broken down by "make. The instant results make this tool pleasant for the user experience. With these files, retraining the data from the data directory is unnecessary, which can be very time consuming.

User Guide

1. Open “Used Car Price Estimator Prototype.ipynb” inside of Jupyter Notebook
2. Install any missing libraries
 - a. Create a new cell by left-clicking "In [1]" and type the letter “A”
 - b. Inside the cell, type `!pip install <library_name>`. Replace "library_name" with the name of the library you want to install.
 - c. Run the cell by clicking the "Run" button or pressing "Shift + Enter" on your keyboard.
 - d. Repeat for any other missing libraries
3. Navigate to the "Cell" menu and select "Run All."
4. Scroll through the graphics to visualize the data.
5. The last cell, "In [7]" has widgets. Notice the default "make" is listed as Acura in bold under the cell stating the "Training" and "Test" scores. These scores reflect the machine-learning score based on the "Acura" dataset in the data directory. Under the "Estimate Price" button, you will also notice a "make" specific "Price vs. Odometer for Acura" scatterplot as you are currently on the Acura dataset, free to pick a "Year," "Model," and "Odometer" mileage and click "Estimate Price" to test this current dataset. The "History of Estimates:" section populates with your estimate.
6. Pick any other "Make" from the drop-down, and you will instantly move to that dataset with the previously trained model to create a gratifying, quick, pleasurable user experience.

7. Pick a "Year," "Model," and "Odometer," and click "Estimate Price." You will notice that the scatterplot and feature importances graph will update automatically based upon the "Make" you selected. Likewise, you will see this estimate in the "History of Estimates:" section.
8. To test the tool's ability to retrain its models, navigate to the main folder, go under the model folder, and delete one or many of these "random_forest_<Make>.pkl" files, making a note of at least one in which you deleted.
9. Return to the Jupyter Notebook and select "Make" from the drop-down.
10. You will now notice the text **"Loading <Make> data, please wait..."** you may have missed previously due to the incredible speed of not being required to train these "Make" models. This might take a few minutes to train, depending on the "Make" you chose. When this message disappears, the dataset has been prepared, and the Training Score and Test Score will be reflected. If you scroll down the bottom of the page, you will now notice two lines that discuss the grid search:
 1. The first, during the "fitting" process, shows the number of "folds" as the model is being trained five times with a five-fold cross-validation to each candidate for a total of 900 fits.
 2. The second line appears after it finds the optimal hyperparameters for the machine-learning model. "Best hyperparameters" refers to the set of hyperparameters that resulted in the best performance of the machine learning model on the validation data during the grid search. The message means that the grid search determined the best combination of hyperparameters for this dataset's machine-learning model.

Summation of Learning Experience

As a Computer Science student, I have extensive development, mathematics, and data science education. In past positions, I used Python to create scripts to automate my job duties for myself and others. Both my education and experience were an asset in getting this project completed. I enjoyed this project immensely as it incorporates what I considered my "future" hobby. Though I was planning a TensorFlow dashcam project, I'm glad I did something else beforehand. I still plan on doing that project just for fun and the experience. However, this project widens my knowledge of artificial intelligence and helps me continue my lifelong education in technology, as I'm sure it is very much future-proof and, no doubt, immensely fascinating.

I utilized many resources to compile the project, including many random YouTube. My favorite resource was Andrew Ng from Stanford's "Machine Learning Specialization" course on Coursera.

Some additional resources I utilized are on the following References page.

References

- DataCamp. (n.d.). Retrieved March 27, 2023, from <https://www.datacamp.com/courses/introduction-to-machine-learning-with-python>
- Fast.ai. (n.d.). Retrieved March 27, 2023, from <https://www.fast.ai/>
- IPython. (n.d.). Retrieved March 27, 2023, from <https://ipython.readthedocs.io/en/stable/api/generated/IPython.display.html> and <https://ipywidgets.readthedocs.io/en/stable/examples/Widget%20Basics.html>
- Jupyter Widgets. (n.d.). Retrieved March 27, 2023, from <https://ipywidgets.readthedocs.io/en/latest/>
- Matplotlib: Visualization with Python. (n.d.). Retrieved March 27, 2023, from <https://matplotlib.org>
- NumPy. (n.d.). Retrieved March 27, 2023, from <https://numpy.org/>
- pandas: Python Data Analysis Library. (n.d.). Retrieved March 27, 2023, from <https://pandas.pydata.org/>
- Python 3.10. (n.d.). pickle: Python object serialization. Retrieved March 27, 2023, from <https://docs.python.org/3/library/pickle.html>
- Python Software Foundation. (2020). The Python Language Reference, version 3.9.1. Retrieved March 27, 2023, from <https://docs.python.org/3/reference/>
- scikit-learn: Machine Learning in Python — scikit-learn 1.1 documentation. (n.d.). Retrieved March 27, 2023, from <https://scikit-learn.org/stable/>
- Scikit-learn. (n.d.). Pre-processing, RandomForestRegressor, IterativeImputer, and GridSearchCV. Retrieved March 27, 2023, from <https://scikit-learn.org/stable/modules/preprocessing.html>, <https://scikit-learn.org/stable/modules/preprocessing.html>

[learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html), <https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html>, and https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

seaborn: statistical data visualization. (n.d.). Retrieved March 27, 2023, from <https://seaborn.pydata.org/>

Seaborn. (n.d.). histplot and scatterplot. Retrieved March 27, 2023, from <https://seaborn.pydata.org/generated/seaborn.histplot.html> and <https://seaborn.pydata.org/generated/seaborn.scatterplot.html>

Towards Data Science. (n.d.). Retrieved March 27, 2023, from <https://towardsdatascience.com/>

Machine Learning Mastery. (n.d.). Retrieved March 27, 2023, from <https://machinelearningmastery.com/>