Project 4 - Group 1
Samrawit Basazinew
Alex Moore
Jessica Price
Andrew Vick

Used Car Price Predictions

Introduction

The dataset used in this project contains detailed information about used cars, including features such as make, model, year, mileage, fuel type, engine type, and transmission. The target variable in this dataset is price, with each record representing a unique car. The goal of this project is to develop a tool that can help both consumers and car dealerships predict used car prices based on these features, providing data-driven insights to assist in making informed decisions when buying or selling vehicles. By applying machine learning models, the tool offers price estimates, helping users navigate the used car market with greater transparency and understanding. Ensuring that the tool is both accessible and user-friendly is a key objective of this project. Originally, the dataset included two CSV files: one with information about used cars in Canada and the other focused on the U.S. market. We decided to work with the U.S. dataset as we believed it would be easier to clean and analyze due to our familiarity with state and city data in the U.S.

Data Cleaning

The original dataset resulted from eight years of daily data collection from over 65,000 dealership websites across the U.S. It contained 21 columns and over seven million rows. The data underwent two phases of cleaning: initial cleaning to fit the dataset to the project's needs and data engineering to prepare it for the machine learning model.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7104304 entries, 0 to 7104303
Data columns (total 21 columns):
 #   Column        Dtype
---  ------        -----
 0   id            object
 1   vin           object
 2   price         float64
 3   miles         float64
 4   stock_no      object
 5   year          float64
 6   make          object
 7   model         object
 8   trim          object
 9   body_type     object
 10  vehicle_type  object
 11  drivetrain    object
 12  transmission  object
 13  fuel_type     object
 14  engine_size   float64
 15  engine_block  object
 16  seller_name   object
 17  street        object
 18  city          object
 19  state         object
 20  zip           object
dtypes: float64(4), object(17)
memory usage: 1.1+ GB
```

Initial Cleaning

The first step in the initial cleaning process was to remove all duplicate values by keeping only the first instance of each Vehicle Identification Number (VIN). This reduced the dataset from 7,104,303 rows to 2,387,394 rows. Next, rows containing null values were dropped, further reducing the dataset by 484,788 rows.

The second step involved filtering the dataset to include only five years of data (2016–2021), bringing the dataset down to 1,253,412 rows. Some outliers were identified in the mileage column, so vehicles with fewer than 500 miles and more than 300,000 miles were filtered out to avoid skewing the data. Additionally, vehicles priced under $7,000 were excluded. Data from the U.S. territories, accounting for approximately

20,000 rows, were also removed. Further filtering was applied by removing rows where the vehicle make appeared fewer than 10,489 times.

After completing these steps, we realized that removing rows with null values inadvertently excluded Tesla vehicles, as those rows lacked values in the “engine_size” and “engine_block” columns. To resolve this, Tesla data was separated into its own dataframe before null values were dropped. Once non-electric vehicles were reprocessed, the Tesla dataframe was re-merged with the full dataset.

Columns that did not directly contribute to the machine learning model or the statistical information shown in Tableau were dropped, including “id,” “vin,” “stock_no,” “seller_name,” and “street.” Finally, the data was randomly sampled down to 150,000 rows and saved into a new .csv file.

```
<class 'pandas.core.frame.DataFrame'>
Index: 150000 entries, 580231 to 434767
Data columns (total 16 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   price         150000 non-null  float64
 1   miles         150000 non-null  int64
 2   year          150000 non-null  int64
 3   make          150000 non-null  object
 4   model         150000 non-null  object
 5   trim          150000 non-null  object
 6   body_type     150000 non-null  object
 7   vehicle_type  150000 non-null  object
 8   drivetrain    150000 non-null  object
 9   transmission  150000 non-null  object
 10  fuel_type     150000 non-null  object
 11  engine_size   150000 non-null  object
 12  engine_block  150000 non-null  object
 13  city          150000 non-null  object
 14  state         150000 non-null  object
 15  zip           150000 non-null  object
dtypes: float64(1), int64(2), object(13)
memory usage: 19.5+ MB
```

Data Engineering for Machine Learning Model

To build an effective machine learning model for predicting used car prices, robust data engineering processes were essential. We began by dropping two additional columns (model and trim) and further binning data to reduce the post-categorical encoding data size. We processed the data using Pandas’ `get_dummies` function to numerically encode categorical columns, such as vehicle make, model, and fuel type. The numeric price column was processed through a log transformation method to normalize the distribution. Additionally, a standard scaler was applied to the other numeric columns (miles, year, and engine size) to reduce the distance between values.

Once the data was fully prepped, it was split into training and test datasets using Scikit-learn's `train_test_split` function. The training and test sets were processed through several machine learning models, including Linear Regression, Ridge, Lasso, Elastic Net, Decision Tree Regressor, Random Forest Regressor, AdaBoost Regressor, Extra Trees Regressor, and Gradient Boosting Regressor. The Random Forest Regressor had the highest performance, with an R^2 score of 92.4%, MSE of 1.3%, RMSE of 11.7%, and MAE of 8.4% on the test set.

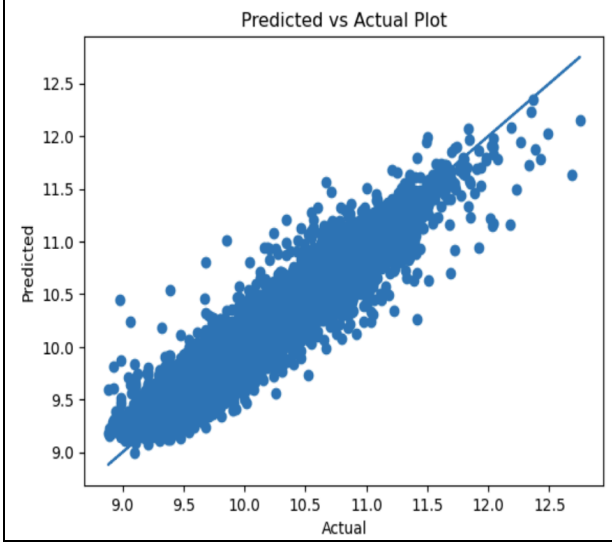
Despite being outperformed by the Random Forest Regressor, we decided to use the Linear Regression model for this web application. The Linear Regression model isn’t as robust as the Random Forest Regressor model, but it takes less time and computing power to process the data. The Linear Regression model had an R^2 score of 82.3%, MSE of 3.2%, RMSE of 18%, and MAE of 13.7%.

```
# Step 2: Init the Model
rf = RandomForestRegressor(random_state=42)

# Do Machine Learning
doRegression(rf, X_train, X_test, y_train, y_test)
```

```
TRAIN METRICS
    R2: 0.9893547064800674
    MSE: 0.0019459776874071947
    RMSE: 0.04411323709961892
    MAE: 0.03142479225860241

TEST METRICS
    R2: 0.9244069296268753
    MSE: 0.013878468461845 05
    RMSE: 0.11780691177450095
    MAE: 0.08437418719231171
```

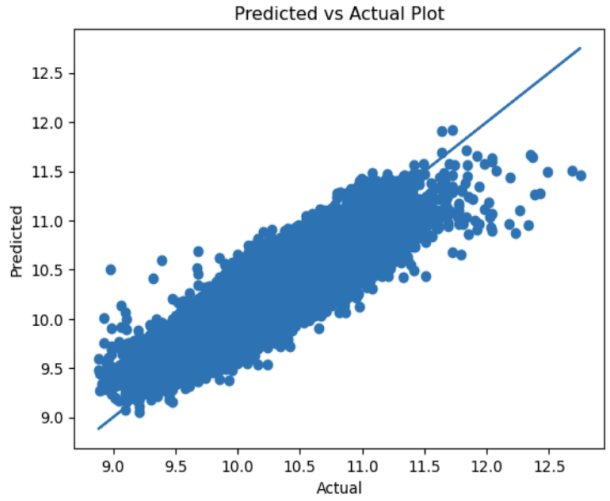



```
# Step 2: Init the Model
lr = Pipeline([
    ('preprocessor', preprocessor),
    ('classifier', LinearRegression())
])

# Do Machine Learning
doRegression(lr, X_train, X_test, y_train, y_test)
```

```
TRAIN METRICS
    R2: 0.8234827461072715
    MSE: 0.03226765301252265
    RMSE: 0.1796319932877288
    MAE: 0.13754974082840868

TEST METRICS
    R2: 0.8233264273388194
    MSE: 0.03243628806340194
    RMSE: 0.18010077196781232
    MAE: 0.13780426024695136
```



Color Design Considerations

For the color design, we chose a simple and visually pleasing palette that's easy on the eyes. Since we expect consumers of all ages to use this tool, we aimed for a design that is easy to read, intuitive to navigate, and uses colors that stand out without being overwhelming. We also considered accessibility in our design choices by ensuring sufficient contrast between colors, making the visualizations easier to interpret for users with visual impairments, such as color blindness. In addition, we used a consistent color scheme across all dashboards to maintain a cohesive user experience and to help users quickly recognize key patterns and outliers in the data. Overall, our goal was to balance aesthetics with functionality, ensuring the tool is not only informative but also engaging for users.

Dashboard Design Concepts

For the dashboards, we aimed to use colors that highlight key information. For example, in visuals displaying the average price of used cars by make or model, we wanted high-priced makes or models to stand out clearly. Similarly, for the map showing average car prices by

state, we followed a similar approach: the higher the car prices in a state, the more intense the red color, making it easy to identify states with higher prices at a glance.

Additionally, we focused on including as much relevant information as possible in the dashboards. Consumers can compare the average price of used cars based on fuel type, with our visualization showing that electric vehicles have the highest average price, followed by diesel vehicles. We also compared the top car makes and models, where Ford emerged as the top make by price, and the F-150 stood out as the top model.

How Does the Dashboard and Machine Learning Model Answer Research Questions?

How can a used car's price be predicted from its features?

The web application allows users to enter their vehicle's information (make, model, trim, year, etc.) and run it through a predictive model. The model analyzes the data and provides an estimated vehicle price with an accuracy of 82%.

Which states have the highest used car prices by make, model, and body type?

The top three states with the highest average used car prices are Wyoming at $37,446, Alaska at $35,996, and Montana at $35,378.

Which makes and models are the most commonly used?

Ford is the most commonly used make across all body types. However, when filtering by body type, such as SUVs, Chevrolet becomes the most common, followed by Ford and Toyota. For wagon body types, Subaru is the most commonly used.

Bias/Limitations

This dataset presented several challenges. With over 7 million rows of nonuniform data, cleaning and preprocessing were cumbersome. Even after the initial cleaning and sampling, the dataset still contained columns with a large number of unique values, primarily due to dealerships manually entering vehicle information without any standardization.

```
UNIQUE VALUES PER COLUMN:
**********
price: 29181
miles: 70223
year: 6
make: 31
model: 511
trim: 801
body_type: 18
vehicle_type: 2
drivetrain: 3
transmission: 2
fuel_type: 19
engine_size: 47
engine_block: 4
city: 4228
state: 50
zip: 11810
```

Time constraints did not allow for thorough preprocessing. Ideally, trim, make, model, and engine size should have been scrubbed for duplicates (e.g., "Yaris" and "yaris"), typos, and data errors. Initially, the columns were quickly cleaned, and obvious errors were mitigated by combining different variations of the same vehicle type into a single value. Columns with values under a certain threshold were binned into an "Other" category. Despite these efforts, the dataset remained too large. Processing it through a categorical encoding function increased the size significantly. The time required for the machine learning model to process the data was not practical for this project or for user experience. To reduce data size

and processing time—at the expense of model accuracy—the model and trim columns were dropped, and we chose to use Scikit-learn's Linear Regression model instead of the more robust Random Forest Regressor model.

Another limitation was the absence of vehicle accident history and the number of previous owners in the dataset. Both of these factors contribute to a vehicle's value.

Conclusions/Reflection

This project successfully demonstrated the development of a machine learning model and accompanying dashboard to predict the price of used cars based on various features such as make, model, year, mileage, and fuel type. By cleaning and preprocessing a large dataset, applying robust data engineering techniques, and using intuitive visual design, we created a tool that provides valuable insights for consumers and car dealerships. The final dashboard allows users to easily compare car prices across different factors, empowering them to make informed decisions when buying or selling vehicles.

Despite its successes, the project faced challenges such as the lack of accident history and standardization issues within the dataset. These limitations, along with time constraints, affected the model's overall accuracy and usability. Future improvements could include incorporating additional vehicle history data, enhancing preprocessing efforts, and optimizing the model for larger datasets.

Overall, this tool provides a solid foundation for further refinement and serves as a useful resource for navigating the used car market. The combination of predictive modeling and user-friendly dashboards ensures that the project not only offers data-driven insights but also enhances transparency and ease of use in a complex market.