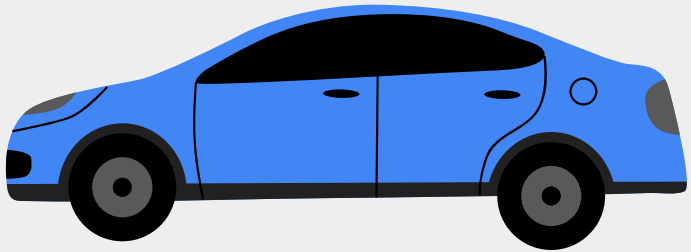
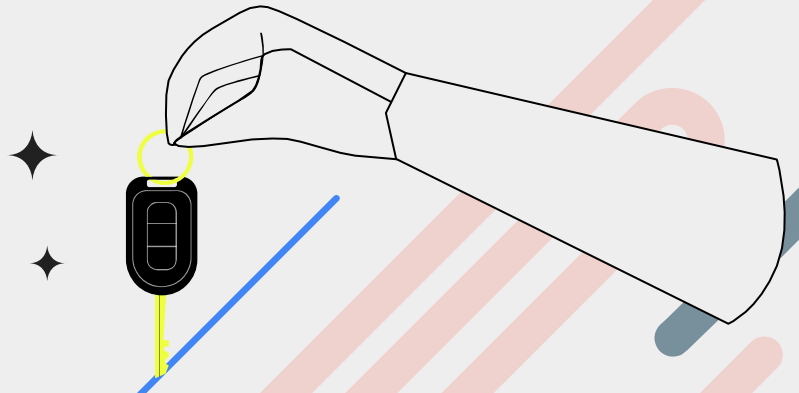
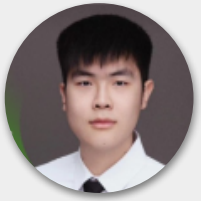
Price Modeling to Outdrive the Competition

StratEdge Solutions





Team Members



Huaiyuan Fan

Caleb Dimenstein

Hourui Guo

Chen Sun

Agenda

01
Problem Overview

02
EDA

03
Feature Engineering

04
Modeling

05
Evaluation

06
Deployment & Timeline

01

Problem Overview

Used Car Price Prediction Model

Problem:

California is one of the most competitive used car market in the US. Our project aims to provide advice to a dealership planning to enter the California used car market with an effective pricing strategy without investing a lot of human resources and brute force calculations.

Research Topic:

How can we build an effective price prediction system for the California used car market based on the available listing used car information?

Approach:

Design a machine learning model that can predict the used car price in the California market according to various features.

US Used Car Dataset

From Cargurus.com

Used Cars in CA

Listing used car information in 1 year

Data Available

Data Cleaning, remove duplicates, deal with outliers

Missing Data

Drop these records

Others

Other Region

02

EDA

Exploratory Data Analysis

Step 1: Check the data types

Step2: Deal with missing data

Step 3: Deal with Inconsistent Text & Typos

Step 4: Remove duplicate and Outliers

Step 5: Visualize data before modeling

Data Quality Access

- Some features have high null value rate
- Boolean values need to be converted
- Data needs to be standardized

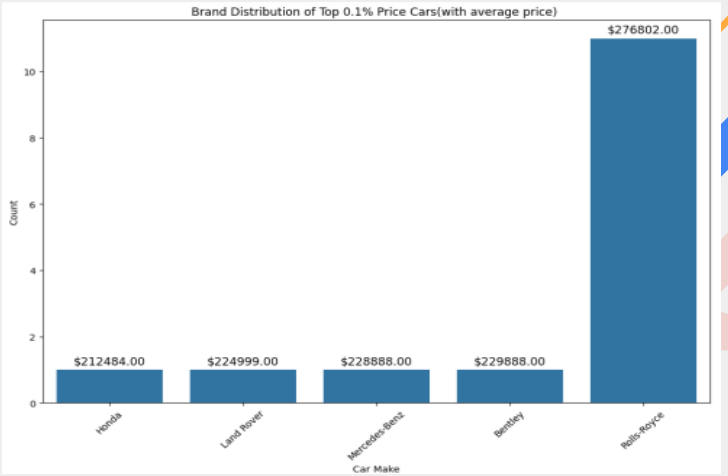
	Name	Type	% of missing values	Noisiness
vin	vin	object	0.0	Limited, depends on data quality and entry
sp_name	sp_name	object	0.0	Limited, depends on data quality and entry
savings_amount	savings_amount	int64	0.0	Outliers possible
price	price	float64	0.0	Outliers possible
model_name	model_name	object	0.0	Limited, depends on data quality and entry
...	...	...	...	...
cabin	cabin	object	99.036836	Limited, depends on data quality and entry
bed	bed	object	99.91691	Limited, depends on data quality and entry
combine_fuel_economy	combine_fuel_economy	float64	100.0	Outliers possible
is_certified	is_certified	float64	100.0	Outliers possible
vehicle_damage_category	vehicle_damage_category	float64	100.0	Outliers possible

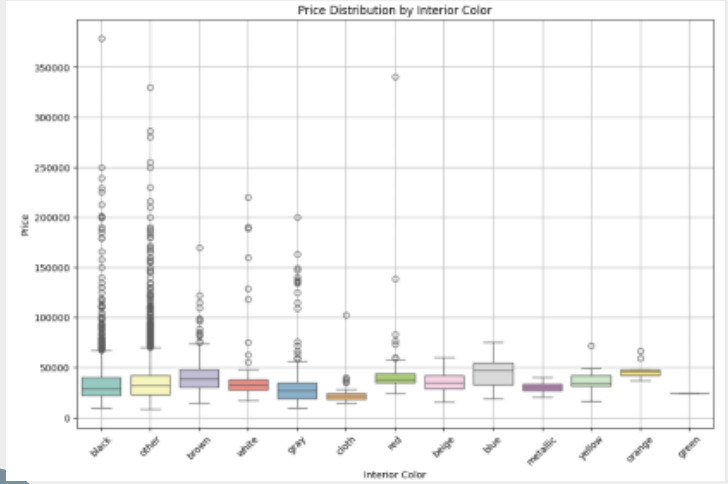
frame_damaged	franchise_dealer	franchise_make	front_legroom	fuel_tank_volume	fuel_type	has_accidents	height	hi
NaN	True	FIAT	37.7 in	21.1 gal	Gasoline	NaN	57.5 in	
NaN	True	FIAT	37.7 in	21.1 gal	Gasoline	NaN	57.5 in	
NaN	True	FIAT	37.7 in	21.1 gal	Gasoline	NaN	57.5 in	
NaN	True	FIAT	37.7 in	21.1 gal	Gasoline	NaN	57.5 in	
NaN	True	FIAT	37.7 in	21.1 gal	Gasoline	NaN	57.5 in	

Category Variables Access

For category variables:

- The outliers are concentrated in Luxury bands
- Interior/exterior colors can affect price

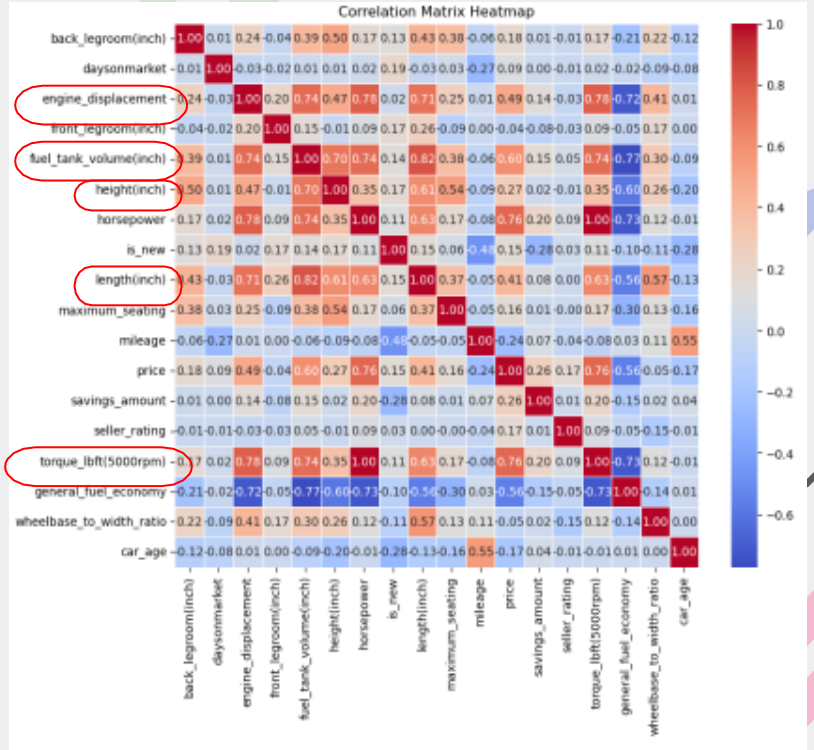




03
Feature Engineering

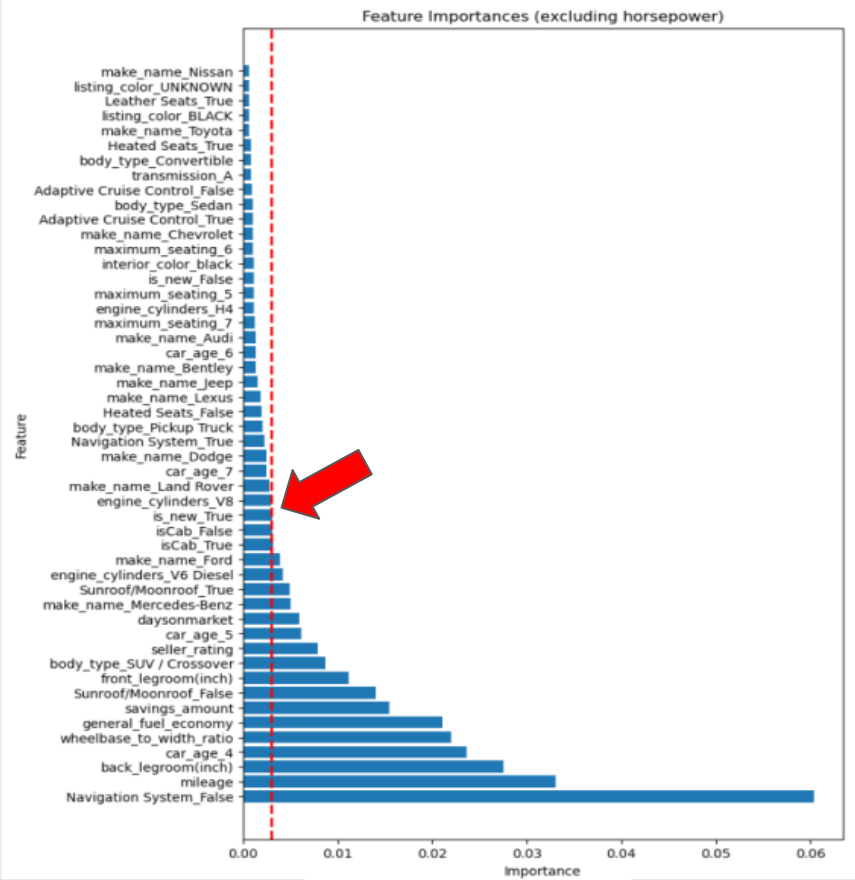
Pearson Correlation and Cramer's V used to reduce features

- Drop **high correlation numerical** features using **Pearson Correlation**
  - Threshold of **0.6**
  - **Five features** dropped
    - Engine Displacement
    - Fuel Tank Volume
    - Height
    - Length
    - Touque
- Drop **high correlation categorical** features using **Cramer's V**
  - Threshold of **0.5**
  - **Four features** dropped
    - Fuel type
    - Fleet
    - Wheel system
    - Third Row Seating



Feature Selection based on feature importance score

- Decision Tree Regressor for **Feature Importance Score**
- Threshold of **0.003**
- **17 features** were selected



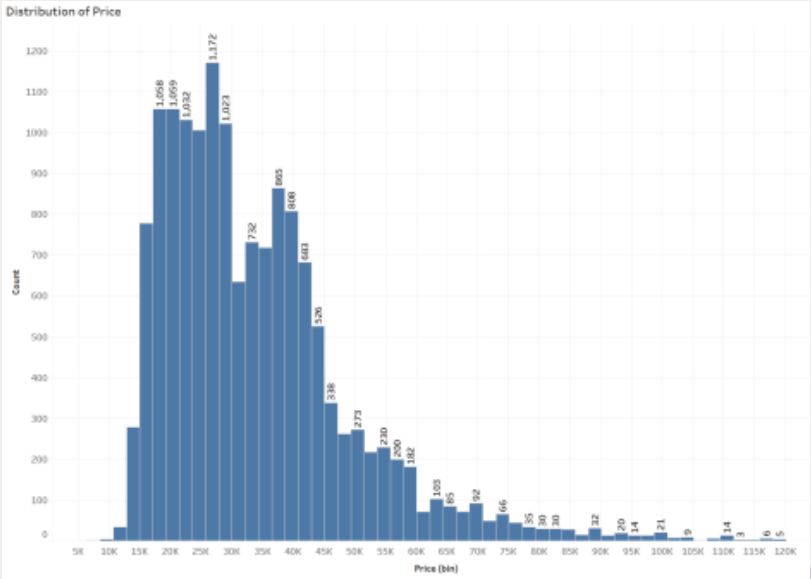
1. See Appendix for details

Transform data so Linear Regression model would produce accurate results

Right Skew

Log Transformation

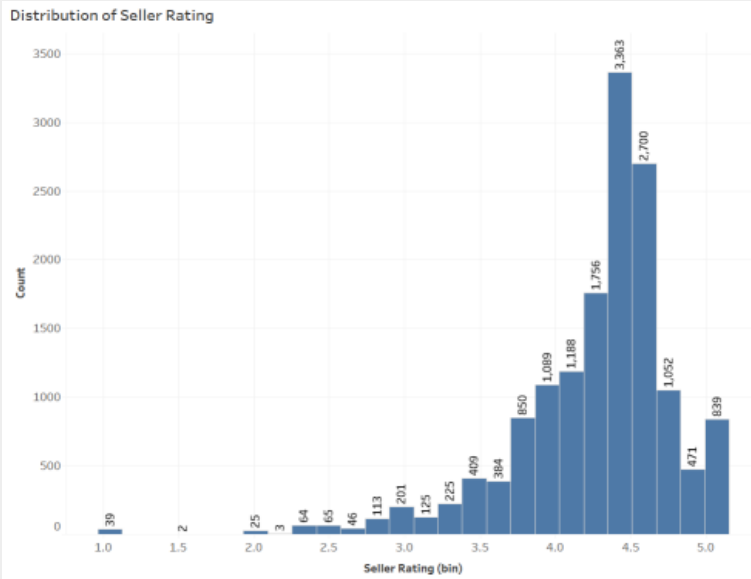
Features: Horsepower, Front legroom, Wheelbase to width ratio, general fuel economy, savings amount, days on market, price



Left Skew

Exponential Transformation

Feature: Seller Rating



04

Modeling

- **Four regression model** were selected based on the dataset
  - Linear Regression 
  - Random Forest 
  - Gradient Boosting 
  - Voting Average 
- **Random Search** were used in Random Forest and Gradient Boosting to find **optimal hyperparameter**

Linear Regression

Description
Find an linear Equation between Feature and Target

Benefits
Easier to interpret

Performance
Adjusted R2:0.894
RMSE:0.129

Random Forest

Description
Create multiple Decision Tree using **subset of data and feature**

Benefits
Can **handle non linear data**
No Normalization Needed

Performance
Adjusted R2:0.939
RMSE:0.097

Gradient Boosting

Description
Another tree model that **make predictions based on previous error**

Benefits
Can **handle non linear data**
Higher accuracy than Random Forest

Performance
Adjusted R2:0.949
RMSE:0.088

Voting Average

Description
Average prediction of previous model by **taking average**

Benefits
Robust Model
Able to achieve **high accuracy**

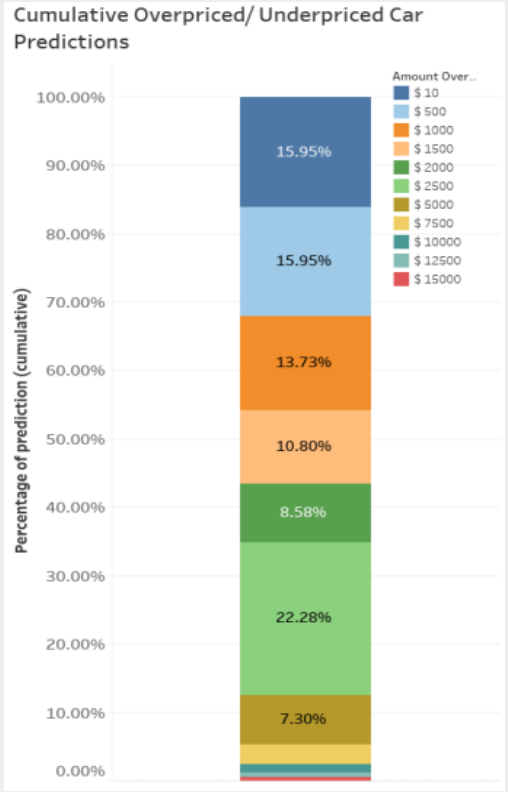
Performance
Adjusted R2: 0.951
RMSE: 0.088

05

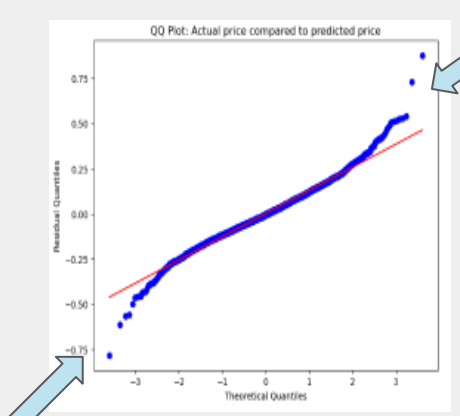
Evaluation

While our model over prices cars by 6%, most of this variation happens at the extremes

- Any model will over price at some level the goal is to minimize this
- With the average car price of, $33,500, we on average overpriced cars by 6%



Slightly over prices expensive cars



Slightly under prices less expensive cars

- Comparison with Github Top Star models Posted
  - Car-Price-Prediction [1]
  - Predicting-the-Price-of-Used-Cars [2]
  - Car-Price-Prediction [3]

Model	Adjusted R Square	RMSE
Github Model 1	0.883	3.796
Github Model 2	0.801	2.945
Github Model 3	0.92	1.410
Our Model (Voting Average)	0.951	0.884 (Exponential)

06

Deployment and Timeline

Process of Pricing A Car

Car Details are entered into the system	Model Produces Results	Final Price

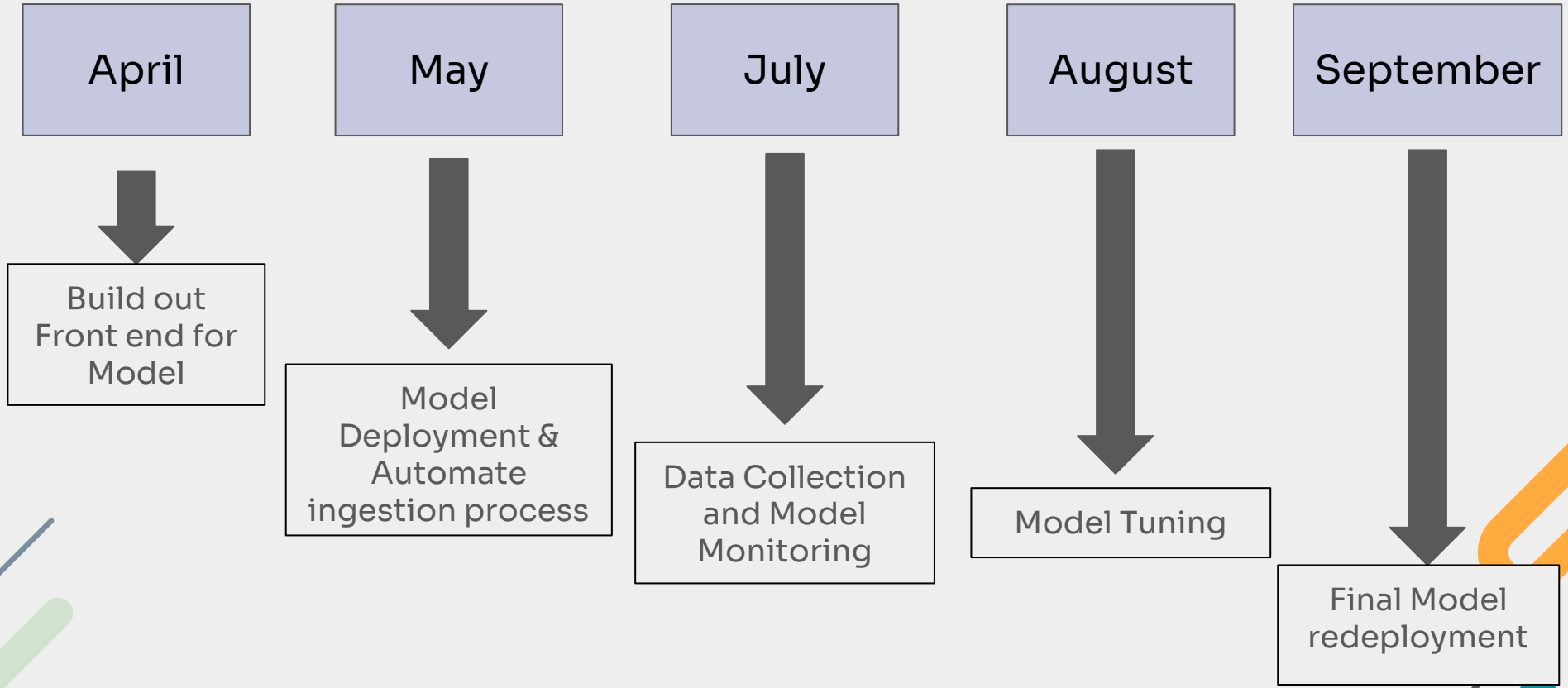
Horsepower: 5.3
Back Legroom: 3.87
Days on Market: 2.78
Front Legroom: 3.71
Mileage: 13,456
Savings Amount: 0
Seller Rating: 87.5
General Fuel Economy: 2.87
Wheelbase to Width Ratio: -0.9
Is Cab: False
Is New: False
Sunroof Moonroof: True
Navigation System: True
Body Type: SUV/Crossover
Engine Cylinders: V6
Make: Jaguar
Car Age: 5

Exponentiate

10.18

$26,534

Timeline for Deployment



Accurately predict used car prices with accuracy and more precise than the competition

- Our model accounts for **95%** of price variation
- Other prediction models account for **88%** to **93%** of price variation
- As more cars are brought into the dealership, our model will become more accurate

Future development

- Separate model for exotic cars
- Motorcycles and other vehicles

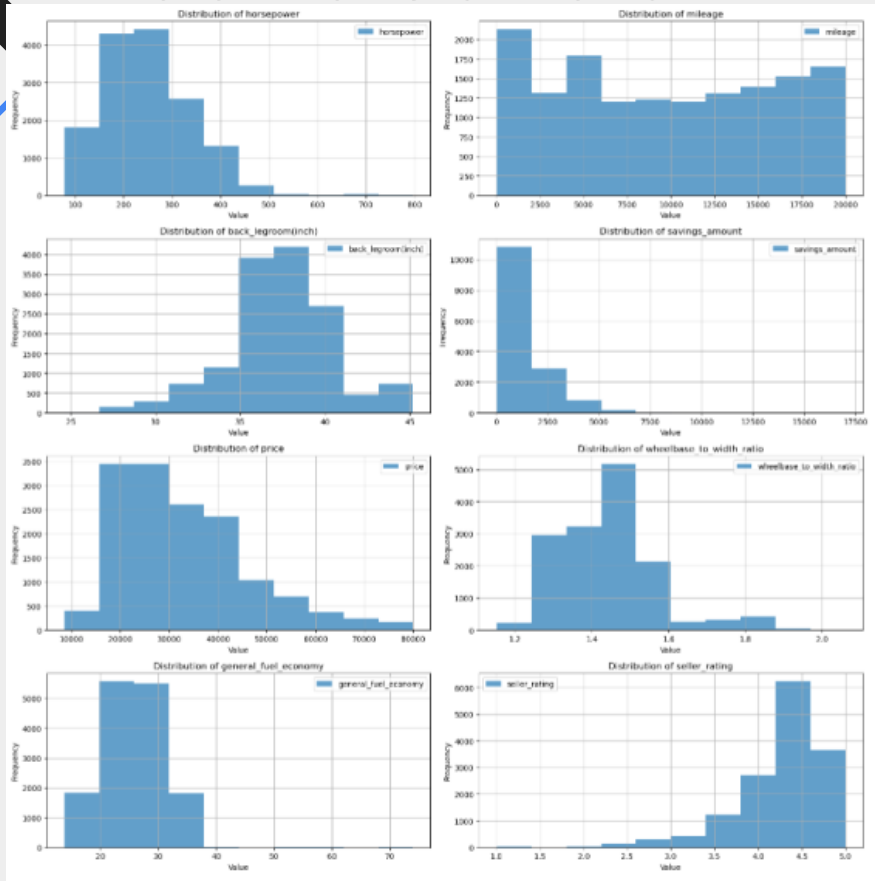
Thank you

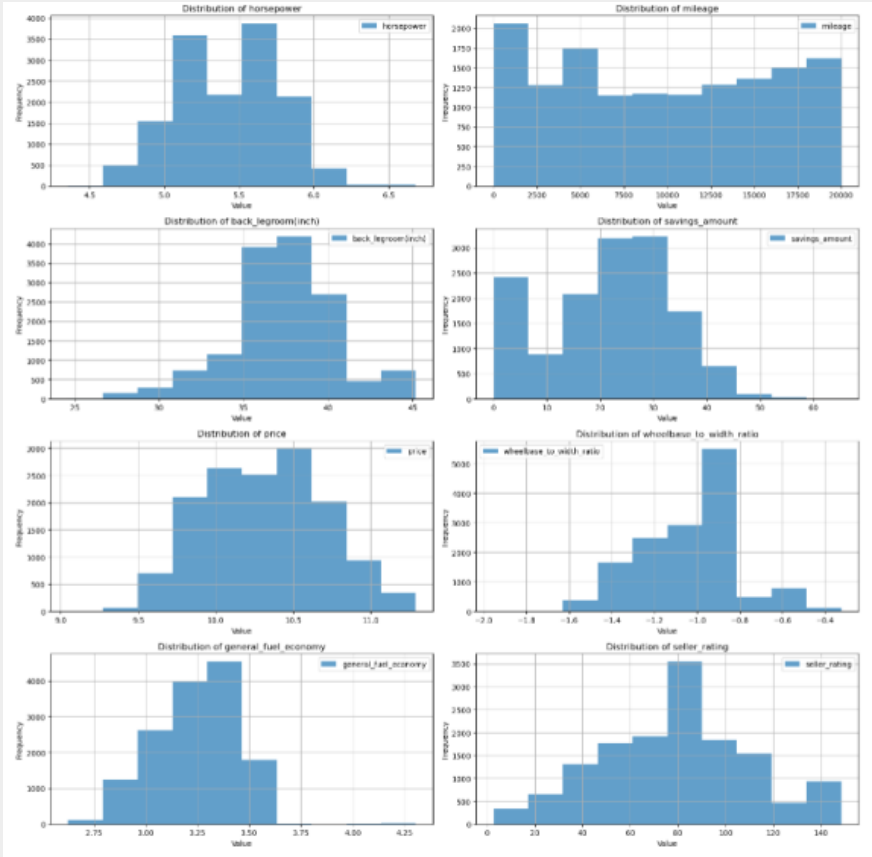
Appendix

Feature Importance

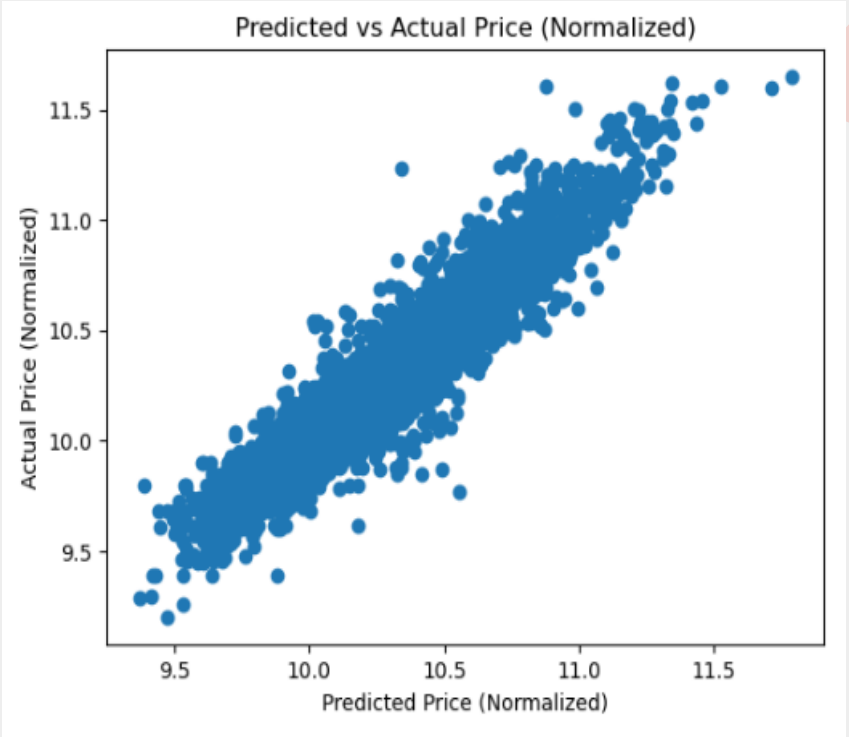
Feature Name	Importance Score - Contribution to Entropy
Horse Power	0.66
Navigation System False	0.06
Mileage	0.03
Back Legroom	0.02
Car Age 4	0.24
Wheel Base to Width	0.02
General Fuel Economy	0.02
Savings Amount	0.02
Sunroof	0.01

Data Transformation





Residual & Residual vs Actual



Model Hyperparameters

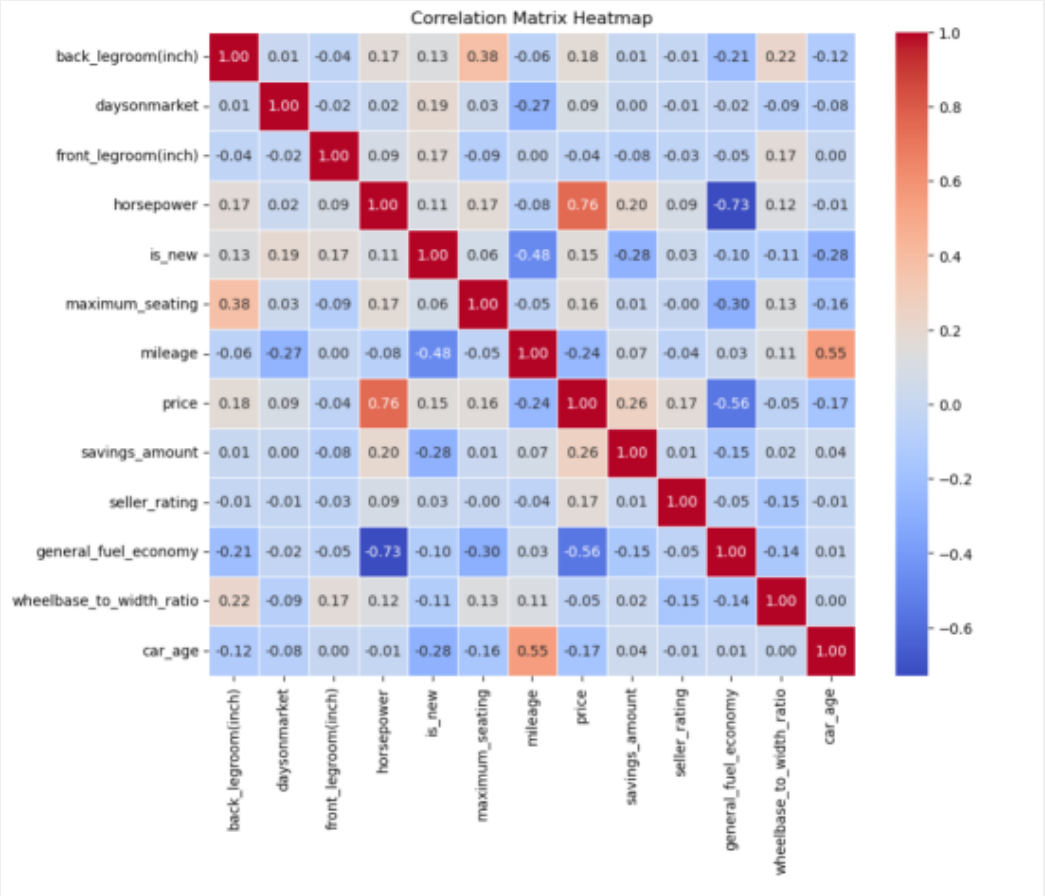
Random Forest Hyperparameters

- **Max Depth**: range from 5 to 15
- **Min Samples Split:** 20, 30, 40, 50, 60
- **N Estimators:** 100, 200, 300, 400, 500
- **Max Left Nodes:** range from 2 to 100

Gradient Boosting Hyperparameters

- **N Estimators**: 50, 100, 200
- **Learning Rate**: 0.01, 0.1, 0.2
- **Max Depth**: range from 1 to 100
- **Min Samples Split**: 20, 30, 40, 50, 60
- **Min Samples Leaf**: range from 1 to 100
- **Max Features**: None, sqrt, log2

Correlation Matrix After



High correlated categorical variables

```
[('body_type', 'wheel_system'), ('engine_cylinders', 'fuel_type'), ('fleet', 'isCab'), ('is_new', 'make_nam
e'), ('make_name', 'wheel_system'), ('make_name', 'Sunroof/Moonroof'), ('make_name', 'Navigation System'),
('make_name', 'Adaptive Cruise Control'), ('maximum_seating', 'Third Row Seating'), ('Android Auto', 'CarPla
y')]
```

Linear Regression

Linear Regression	Metric
Mean Absolute Error	0.0990
Mean Squared Error	0.0168
Root Mean Squared Error	0.1294
R-squared	0.8952
Adjusted R-squared	0.8944

Random Forest Regressor

Random Forest Regressor	Metric
Mean Absolute Error	0.0686
Mean Squared Error	0.0095
Root Mean Squared Error	0.0976
R-squared	0.9404
Adjusted R-squared	0.9399

Gradient Boosting Regressor

Gradient Boosting Regressor	Metric
Mean Absolute Error	0.0639
Mean Squared Error	0.0081
Root Mean Squared Error	0.0902
R-squared	0.9491
Adjusted R-squared	0.9488

Voting Average

Voting Average	Metric
Mean Absolute Error	0.0639
Mean Squared Error	0.0078
Root Mean Squared Error	0.0884
R-squared	0.9511
Adjusted R-squared	0.9508