craigslist

Used Car Price Prediction

A6 - Mathias Leys, Alejandro Calderón, Monica Jiang, Johannes Jung, Micael Cunha Alves

Business Problem

Business Problem

Creating an Online Platform For Used Cars

- Car dealer aiming to **digitize** its business
- Appropriate **price determination** for used cars

GOAL

- Create a **price prediction** model

USES

- **Benchmark** for car purchase and resale for users
- **Pricing service** to customers listing their cars for sale

→ **Competitive Advantage**

Machine Learning Model

Supervised ML Model

- Listings of **used cars** for sale, scraped from **craigslist**

Regression Problem

- Prediction of a **continuous variable** (price) based on car features

Interpretability

- Interesting but not critical
- Higher model effectiveness and persuasion with interpretability

Development of a **complex** model for **accurate** prediction and a **simple** model for **interpretation** purposes

Data Preparation

Data Preparation

Before Cleaning

- 458,213 instances
- 26 columns
- 91% of the instances contain missing values

After Cleaning

- 307,422 instances
- 13 columns
- All missing values handled

Identified Problems

- **Errors** And High **Diversity** In Car Models
- Sample Too **Limited** For Some Car Models
- Instances With **Unrealistic Prices**
- Instances **Not Relevant** For Our Business Case
- Large Dataset And Many **Useless Variables**
- High Number Of **Missing Values**

Data Preparation

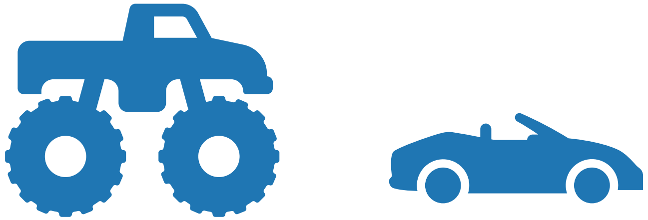
Identified Problems

Errors And High Diversity In Car Models

31K unique car model names

Model names entered **incorrectly** (free form field)

Same model name for different brands



Solutions

✓Selection of the first word in the model variable

✓New variable “car_model”
manufacturer brand + model (1st word)

Example: ‘Fiat – 500 clean condition’ and ‘500 good value’ both become ‘fiat 500’

Data Preparation

Identified Problems

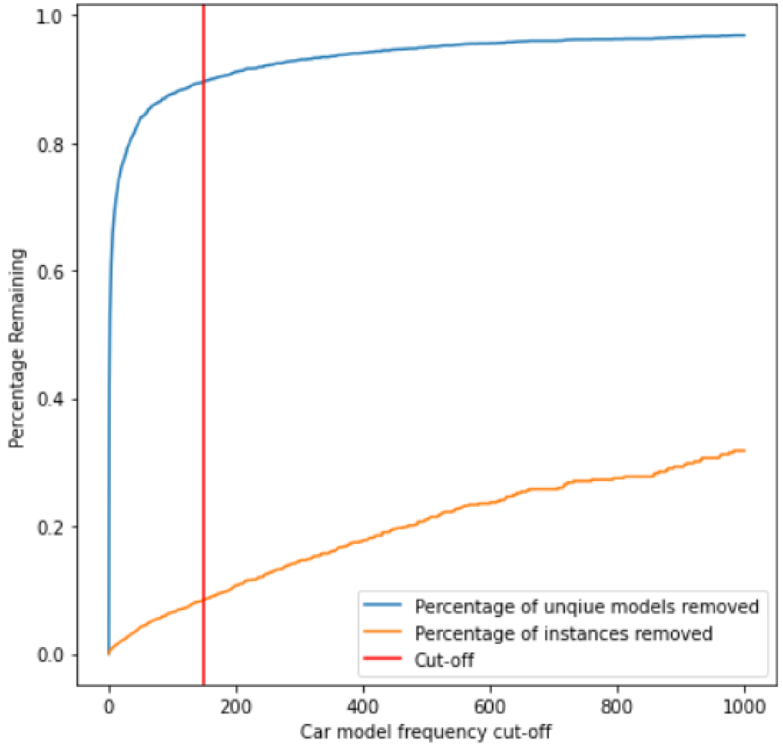
Sample Too Limited For Some Models

Still many unique car models

Some models don’t appear many times

Solutions

Removal of all instances with a car model that appears **less than 150 times**



Trade-off: reducing number of unique models vs not deleting too many instances

Data Preparation

Identified Problems

Instances Not Relevant For Our Business Case

- Presence of **car parts** and **not fully functioning** car listings
- Some cars **too old** or **too high mileage** for our intended business case (cf. Appendix 2)

Solutions

- ✓ Removal of all instances with “title_status” different from “clean” and “cylinders” equalling “Other”
- ✓ Removal of the cars older than 1960 (year)
- ✓ Removal of instances with a mileage over the 99% percentile (“odometer”)

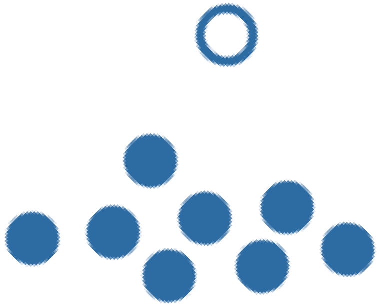
Data Preparation

Identified Problems

Instances With Unrealistic Prices

Prices **too low** or equalling zero

Prices **excessively high** (cf. Appendix 3)



Solutions

- ✓ Removal of instances with price below $ 100 and above $ 500,000
- ✓ Removal of instances with a price 3x higher than the mean of the car model

Data Preparation

Identified Problems

Large Dataset And Useless Information

1.2 GB dataset

Numerous variables **not useful** to the analysis (cf. Appendix 4)

Solutions

Feature selection, removal of :

- Unnamed
- ID
- URL
- Region URL
- Lat
- Long, VIN
- Image URL
- Description
- Posting date

Data Preparation

Identified Problems

High Number Of Missing Values

5 variables with **over 30% missing** values

(cf. Appendix 3)

Solutions

- ✓ Removal of the variable “size” and "title_status“
- ✓ Removal of the instances with missing values for “fuel”, “transmission”, "model", "manufacturer"
- ✓ Replacing the missing values with the median/mode of the car model for : "cylinders", "drive", "type"
- ✓ Replacing missing values with a term "unknown" for "condition" and "paint_color“

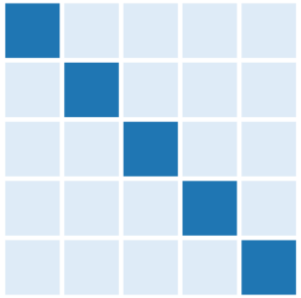
Model Creation & Evaluation

Train-Test Split

Chosen Method

Classic **80%-20% split** (industry standard) with fixed seed

Sufficient amount of data: **307,422 instances**



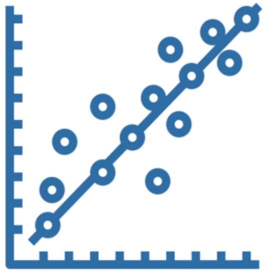
Utilized **5 fold cross validation** to detect and prevent overfitting

Model Analysis

Baseline Linear Regression

- Initial **linear regression** trained on **cleaned** dataset
- Model **performance** is reasonably **poor** but no sign of overfitting

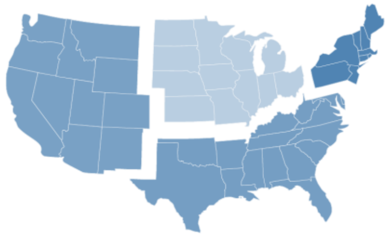
MAE	$ 4,127
MSE	41,984,970
R^2	73.68%



Feature Engineering

Creation of the Region Variable

- **Region –** Feature grouping states into:
  - North-East
  - Mid-West
  - South
  - West



- **Replacement** of “**State**” feature

MAE	$ 4,125
MSE	41,700,422
R^2	73.74%

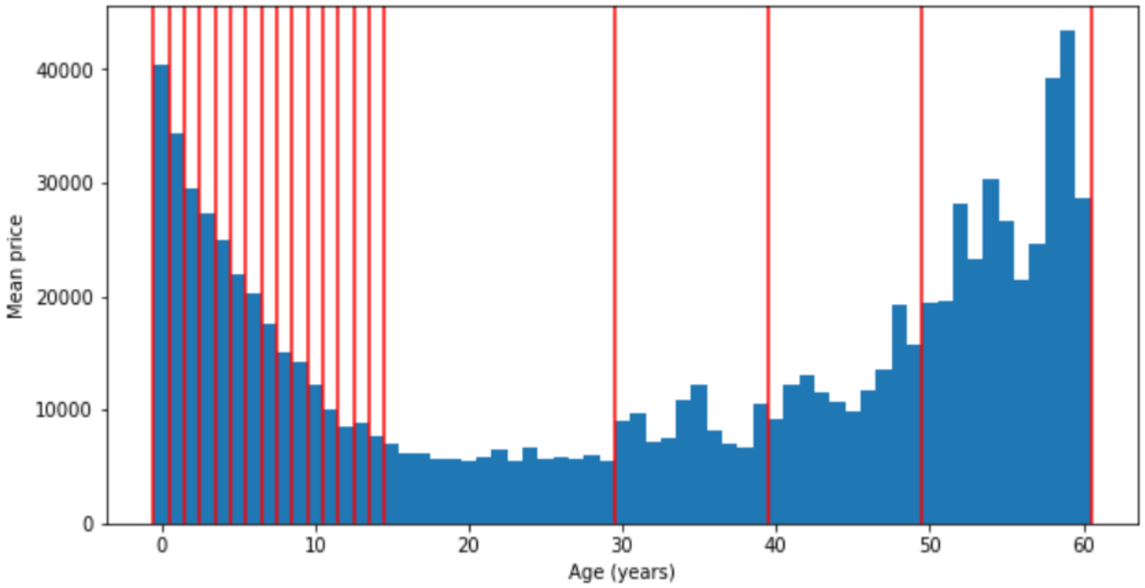
Results

- **No change** in performance (cf. Appendix 5)
- Simplification of the model, higher **interpretability**

Feature Engineering

Creation of the Age Variables

- **Age** – Feature based on the difference *(2021-year)*
- **Age Groups –** Feature based on the mean price per age **distribution**



Results

- Age
  - MAE decreased by 5.6%
- Age Groups
  - Improved performance
  - No sign of overfitting
  - Replaced the year variable

Other Attempted Changes

Feature Selection

- Dropping the "state" column
- Dropping the "car_model" column
- Dropping the ''cylinders'' column

(cf. Appendix 6)

Feature Engineering

- Binning ''year'' variable into ''vintage", "medium" and "recent"
- Change the variable "condition" into a numerical feature
- ''Color'' variable categorization based on the values popularity
- Create a new variable named "miles_per_year"

Deterioration or No Significant Improvement

Model Comparison – Linear Regression

Linear Regression Model

- Ran on the **feature engineered** dataset
- Feature engineering **improved performance** without sign of overfitting
- MAE improved by 5.47% compared to initial model

Metrics	Testing dataset	Training dataset
MAE	$ 3,893	$ 3,915
MSE	36,388,712	36,864,231
R^2	77.09%	76.89%

Model Comparison – Random Forest

Random Forest

- Two random forest
  - 20 decision trees
  - 100 decision trees
- **Cross-validation** on both models
- Significant **outperformance** compared to linear regression
- 5-fold Cross-validation **does not signal overfitting**

Metrics	Testing dataset of 20-tree random forest	Testing dataset of 100-tree random forest
MAE	$1,690.99	$1,647.87
MSE	14,200,740	13,731,284.05
R^2	91.06%	91.35%
Std. dev. of MAE	$188.17 (5-fold Cross Validation on 20-tree RF)	

Model Comparison – Other Models

Boosted Tree Regressor

- Boosted Tree performs **significantly worse** than linear regression

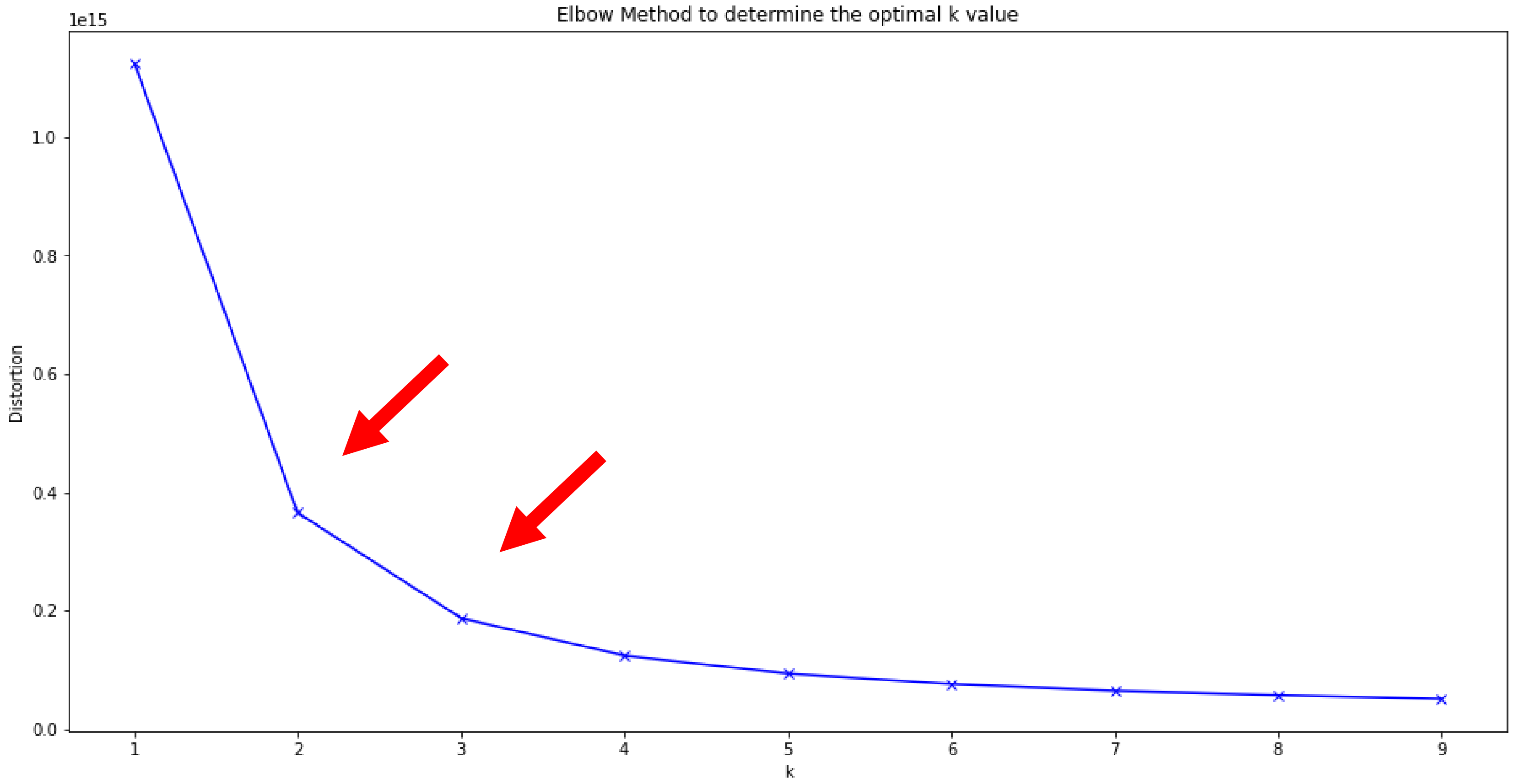
Neural Network

- Better performance than the Boosted Tree Regression but lower than the Random Forest

Metrics	Boosted Tree Regressor	Neural Network
MAE	$ 5,921.78	$ 2,721.09
MSE	72,108,195	22,009,531
R^2	54.6%	86.2%

Cluster Analysis

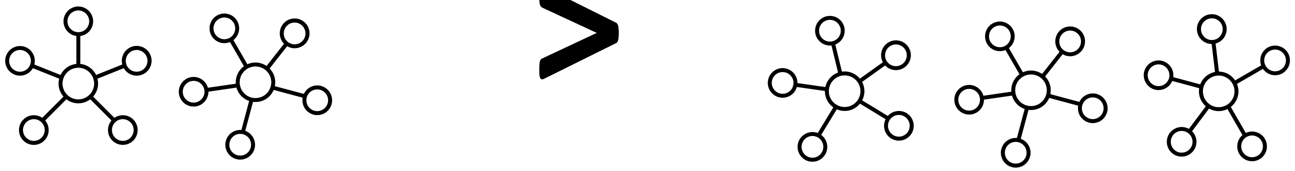
We implemented two **K-means clustering** models, trying **2 and 3** as values for **K** based on the elbow plot below (cf. Appendix 7)



Cluster Analysis

A **20-tree Random Forest** model was executed for each of the clusters. The models in the **2 means clustering** showed significantly better results and also have a **clearer business interpretation**.

K = 2 > **K = 3**



Worst performing cluster of the **2 means** algorithms still **outperforms the best performing** cluster of the **3 means** algorithm (MAE: $ 2,011 < $ 3,014)

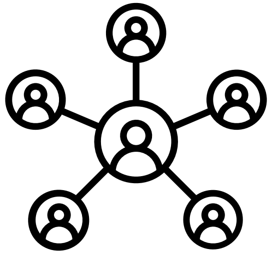
Cluster Analysis

	Middle range	Low range
Details	 Instances: **140 k** Average price: **$ 20 k** Mileage: **40 k miles** Age group: **1-5 years old**	 Instances: **170 k** Average price: **$ 7 k** Mileage: **140 k miles** Age group: **10-30 years old**
MAE (RF)	**$ 2,011 (+ $320)**	**$ 1,295 (- $396)**
R^2 (RF)	**87.92%**	**90.01%**

Final Model

Random Forest model using 2-means clustering

Clusters profiles aid model **interpretability**

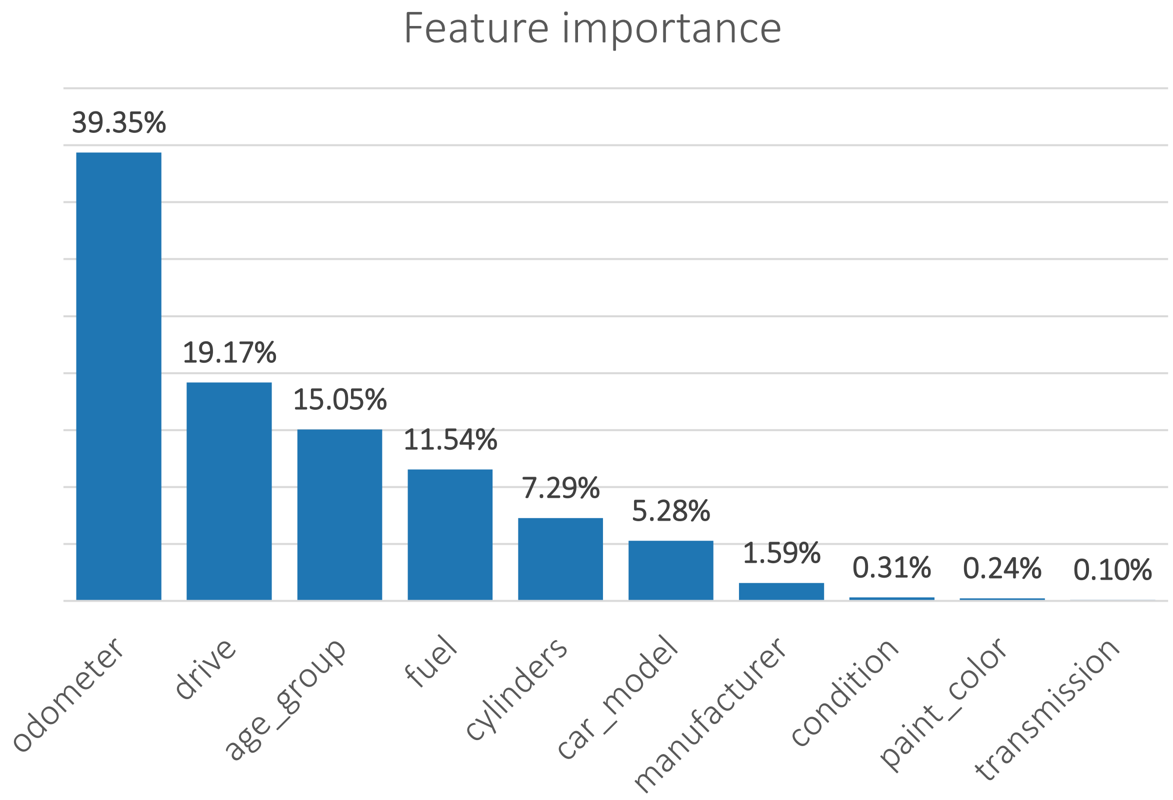


Net improvement of clustering RF compared to general RF (cf. Appendix 8)

Net Change in Absolute Error:
- $22.52 M

Feature Importance

Analyzing feature importance



- Cars in dataset mainly from **lower-middle class manufactures** → biggest impact through **mileage and age**
- Relatively **little variation** through **manufacturer and model**
- Significant role of **major car characteristics** (type of drive, number of cylinders)

Feature Interpretation

Analyzing impact of features with the highest importance level

- Odometer: **$ 4.55** price **decrease** per **100 miles**
- Age group: **New** cars **$ 5,500 more expensive** than **1 year old** cars. Cars lose between **$ 1k and $ 2k per year** over the next 10 years
- Drive: **Four-wheel drive** cars are **$ 1,500 - $ 2,500 more expensive** than front wheel drive/rear wheel drive
- Fuel: **Diesel** fuelled cars are **$ 7,000 more expensive** than cars with other fuel types

(cf. appendix 9)

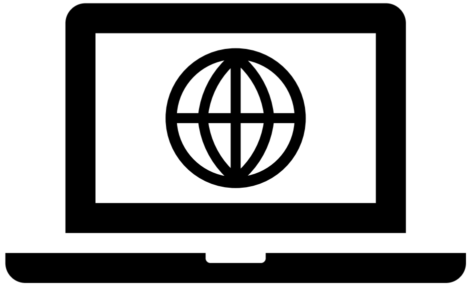
Implementation

Offer **free pricing service** when listing used cars on platform

Drawing users (and cars) to platform

Accelerate transactions due to **efficient prices**

Let users **sell cars directly to platform** – our pricing based on model



Resell cars to car dealers, car rental agencies, and users at slight **mark up**

Future Considerations

Data Quality Improvement

- **Data quality** main issue in this problem
- **Ease process of adding information** to the listing
  - **Direct information** retrieval via VIN
  - **Drop down menu** to select the car model per manufacturer (<-> free form)
  - **Autocompletion** proposal while filling the information

Future Considerations

Data Quality Improvement

- Model based on **USA-exclusive data** may not generalize well to other countries
- Collect similar car data from **different countries**

Continuous Improvement of the Model

- Retrain model on past data **corrected by** using the **VIN** number
- Limited to car models with **sufficient instances**
  - Increase the prediction capability of the model with new models when data sufficient

THANK YOU

APPENDIX

Appendix

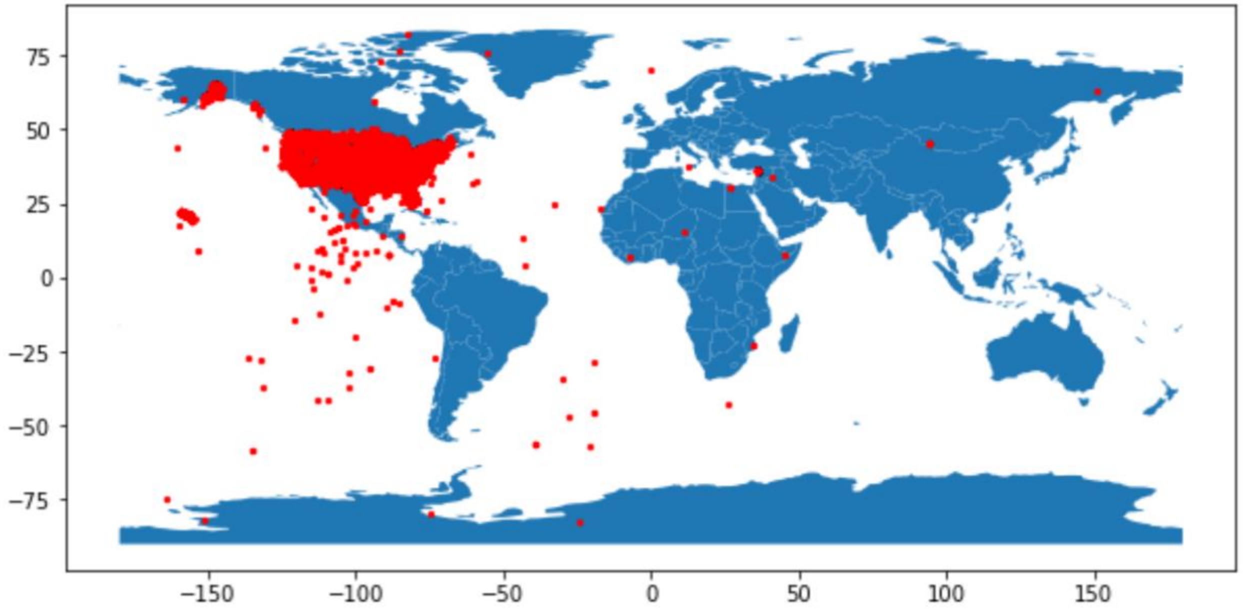
All **Python code** can be found on **GitHub**: https://github.com/Micael-Alves/Craiglist_Second_Hand_Cars_Price_Prediction

We made one **Jupyter notebook** where we perform all the **data processing** and **modeling** steps in a **step-by-step** manner with (sub)titles and comments that make the code and the steps **self-explanatory**

The used **Kaggle dataset** can be found in the following link: https://www.kaggle.com/austinreese/craigslist-carstrucks-data

Appendix 1

Corrupt location values



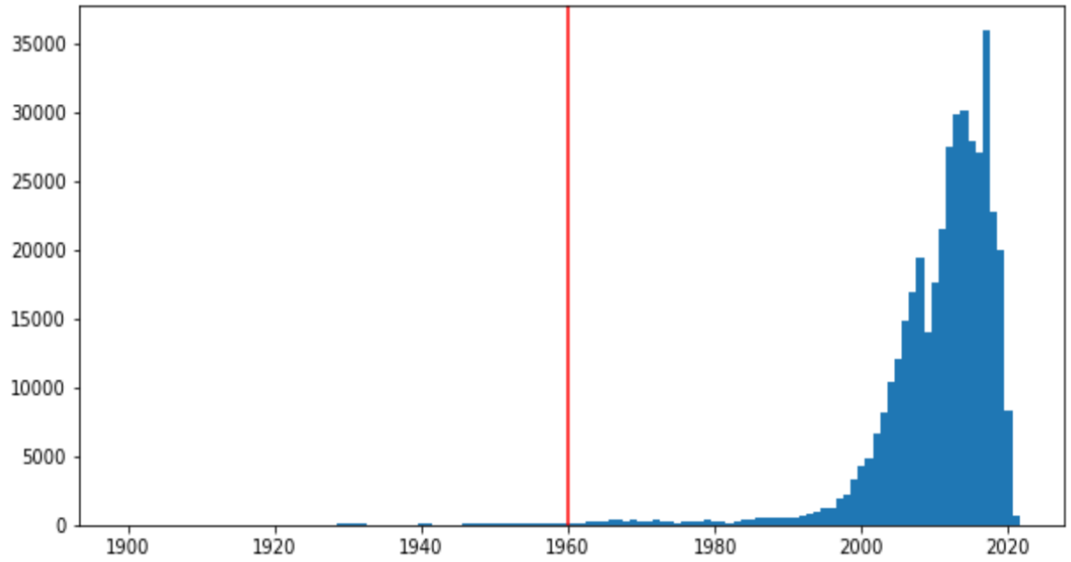
In the initial dataset, there are a lot of **locations** that **don't make sense** (i.e., in Antarctica or in the ocean)

Users can **drop a pin** anywhere in the world

We decided to use **'state' and 'region'** instead of lat & long as an indicator of **location**

Appendix 2

Year Outliers



Based on the graph we built (set the 'year' as x and 'price' as y), we find that there are very few cars built before 1960, which are the outliners of this feature.

Appendix 3

Handling **price outliers**

The **maximum** of the price is **$ 36.15 M**, and the **minimum** is **$ 0**. in the dataset, there were a lot of **dirty values** such as 999,999,999, 1,234,567 and 11,111,111.

The maximum price might be a mistake (over one billion US dollars) which should be deleted. we can also conclude that the data of price is dirty, and we need to refine it if we want to derive some valuable insights.

Solutions

Given our analysis of the price outliners, we decide to set the benchmark for **over-priced** cars as more than **3 times the average** for this **model**.

For those instances with **over $ 500k** or **under $ 100** price, we decided to delete overpriced /underpriced cars

Appendix 4

Variable	Number of Missing Values	% of Missing Values
Size	321 348	70%
Condition	192 940	42%
Vin	187 549	41%
Cylinders	171 140	37%
Paint_color	140 843	31%
Drive	134 188	29%
Type	112 738	25%
Odometer	55 303	12%
Manufacturer	18 220	4%
Lat	7 448	2%
Long	7 448	2%
Model	4 846	1%
Fuel	3 237	1%

Variable	Number of Missing Values	% of Missing Values
Title_status	2 577	1%
Transmission	2 442	1%
Year	1 050	0%
Description	70	0%
Posting_date	28	0%
Image_url	28	0%
State	0	0%
Price	0	0%
Region_url	0	0%
Region	0	0%
Url	0	0%
Id	0	0%

Appendix 5

Creation of the Region Variable

Evaluation results on the testing dataset:

'R2': 73.81%, 'mae': 4115.3, 'mse': 41603557.55, 'mape': 160.2

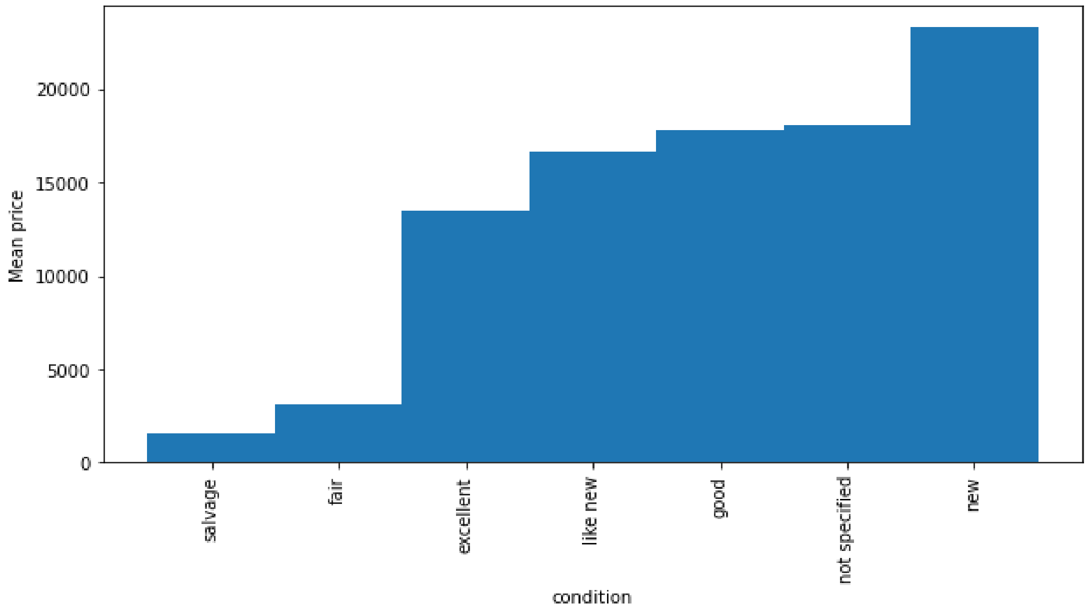
Evaluation results on the training dataset:

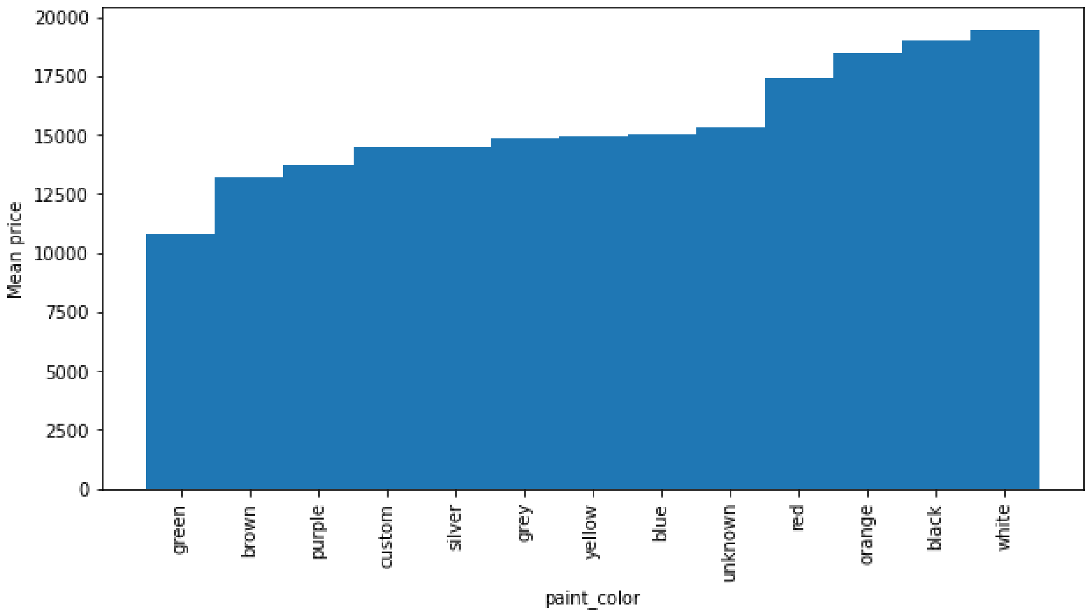
'R2': 73.45%, 'mae': 4130.22, 'mse': 42345093.72, 'mape': 163.79

Here, **dropping "type" and binning "state"** into regions had **no real impact** on performance, but make the model **more interpretable** and simplify the following analysis.

Appendix 6

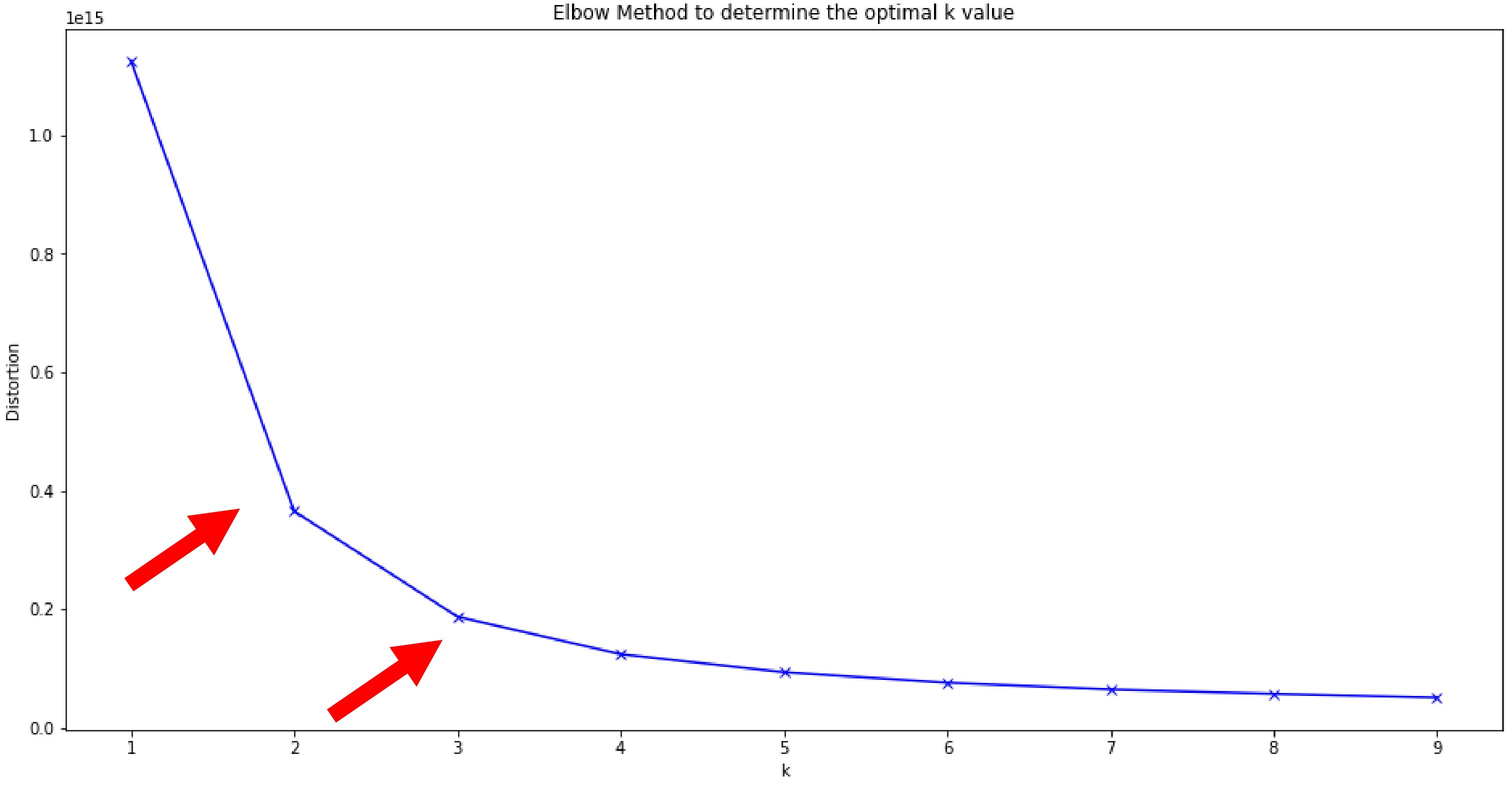
Price vs features analysis





Further **categorizing** features had **no significant impact** on performance

Appendix 7



cluster	price	odometer
0	19990	46176.0
1	7495	140000.0

cluster	price	odometer
0	23319	33305.0
1	5990	170018.5
2	10795	103075.0

Appendix 8

Calculation of Aggregated Absolute Error

The **‘Middle Range**’ cluster RF has **140k instances** and a **MAE** that is **$ 320 higher** than the general RF
The **‘Low Range’** cluster RF has **170k instances** and a **MAE** that **is $ 396 lower** than the general RF

⇒Net effect = 140k * (+ $ 320) + 170k * (- $ 396)
⇒Net effect = $ 44,800,000 - $ 67,320,000
⇒Net effect = -$ 22,520,000

So, the **‘Net Absolute Error’ dropped by $ 22.52 M** when comparing the net cluster RF performance to the general RF performance

Appendix 9

Selection of Feature Coefficients

odometer	123	-0.04554
drive = 4wd	ABC	0
drive = fwd	ABC	-1518.76000
drive = rwd	ABC	-2765.39000
age_group = new	ABC	5689.94000
age_group = 1	ABC	0
age_group = 2	ABC	-3244.61000
age_group = 3	ABC	-4686.01000
age_group = 4	ABC	-6321.62000

Appendix 9

Selection of Feature Coefficients

fuel = diesel	ABC	0
fuel = other	ABC	-7069
fuel = electric	ABC	-7239.41000
fuel = hybrid	ABC	-7336.13000
fuel = gas	ABC	-7766.40000