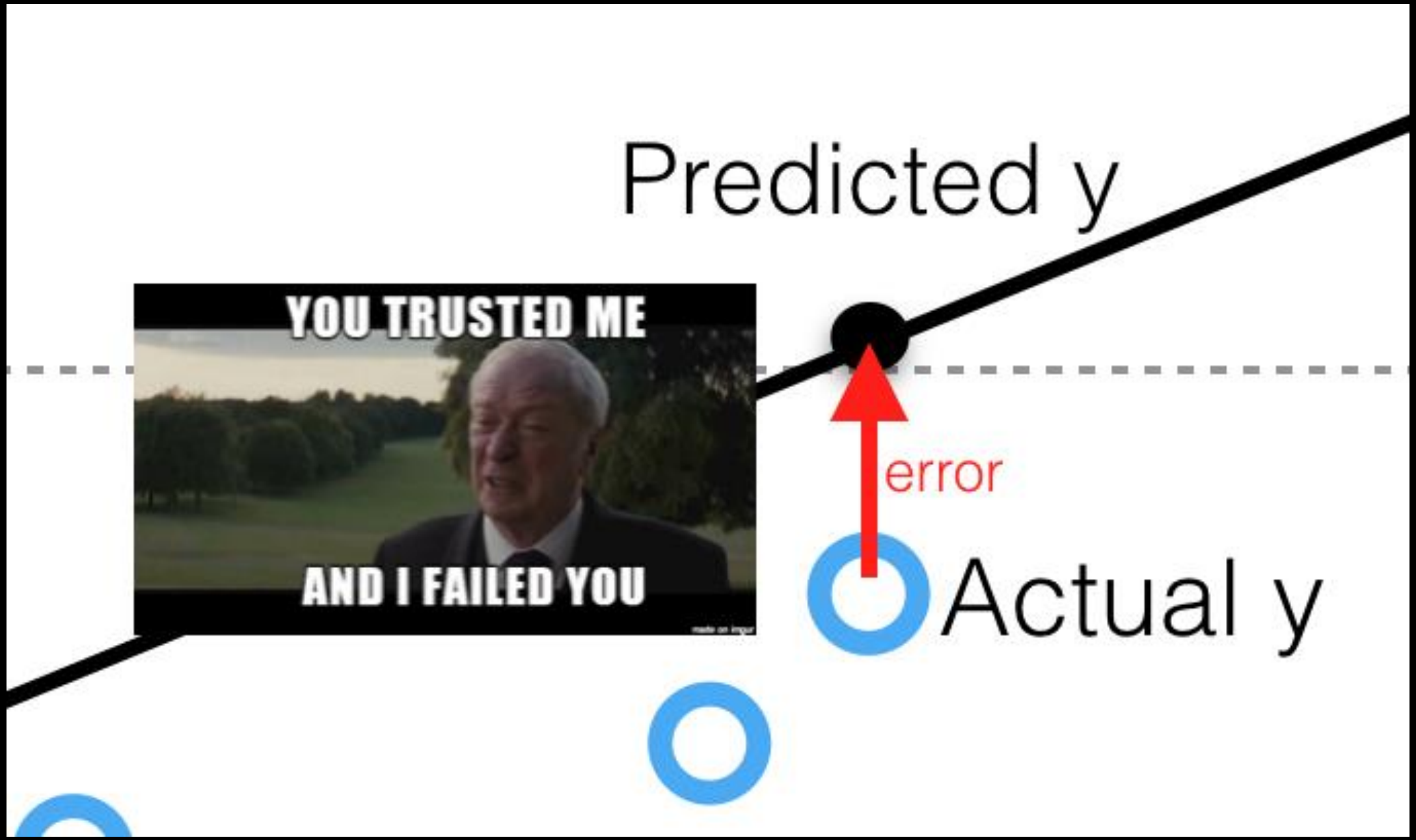
REGRESSION MODEL

CAR PRICE PREDICTOR WEB APP

Steven L Truong

Friday, 16/04/2021


Predicted y
YOU TRUSTED ME
AND I FAILED YOU
error
Actual y

INTRODUCTION

❖**MOTIVATION**:

- Buy and sell used cars is always a big decision.
- Create the tools to predict the as closest car’s price as possible.

INTRODUCTION

❖ **MOTIVATION**:

- Buy and sell used cars is always a big decision.
- Create the tools to predict the as closest car's price as possible.

❖ **OBJECTIVES**:

- Build predictive models using data gathered from the internet.
- Conclude the best model ready for production.

INTRODUCTION

❖ **MOTIVATION**:

- Buy and sell used cars is always a big decision.
- Create the tools to predict the as closest car's price as possible.

❖ **OBJECTIVES**:

- Build predictive models using data gathered from the internet.
- Conclude the best model ready for production.

❖ **GOALS**:

- Write the web app and deploy the model to the cloud.

METHODOLOGY

- **Data source:**
  - Scrape from cars.com

METHODOLOGY

- **Data source:**
  - Scrape from cars.com

USED

2019 Chevrolet Equinox 1LT

34,354 miles

$19,995 GOOD DEAL

HOT CAR

DELIVERY AVAILABLE | VIRTUAL APPOINTMENTS

Sold by Cash for Cars San Jose, CA 95128

(4.5) 2 Reviews

Basics

Fuel Type: Gasoline
City MPG: 26
Highway MPG: 32
Drivetrain: FWD
Engine: 1.5L I4 16V GDI DOHC Turbo
Mileage: 34,354
Exterior Color: Summit White
Interior Color: Jet Black
Stock: 3558
Transmission: 6-Speed Automatic
VIN: 3GNAXKEV4KL116752

Show more details

Convenience

- Keyless Start
- USB Port

Entertainment

- Bluetooth
- Premium Sound System
- Apple CarPlay/Android Auto

Safety

- Backup Camera
- Brake Assist
- Stability Control

Exterior

- Alloy Wheels

METHODOLOGY

Data Scraping and Preparation

- Use BeaufulSoup to scrape data from cars.com
- Clean the data to be ready for EDA

METHODOLOGY

Data Scraping and Preparation

- Use BeaufulSoup to scrape data from cars.com
- Clean the data to be ready for EDA

Exploratory Analysis

- Exploratory Data Analysis
- Look at the features' correlations for insights before modeling.

METHODOLOGY

Data Scraping and Preparation

- Use BeaufulSoup to scrape data from cars.com
- Clean the data to be ready for EDA

Exploratory Analysis

- Exploratory Data Analysis
- Look at the features' correlations for insights before modeling.

Modeling

- Build baseline models.
- Cross validation and choose the final model.

DATA CLEANING

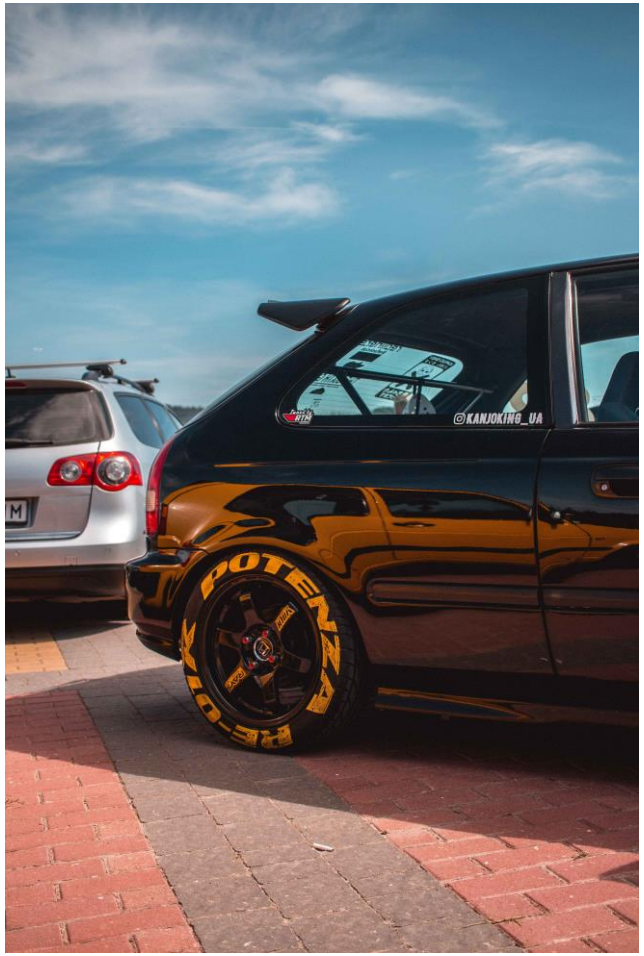
DATA CLEANING

DATA CLEANING

Or we could say “cars cleaning”

DATA CLEANING

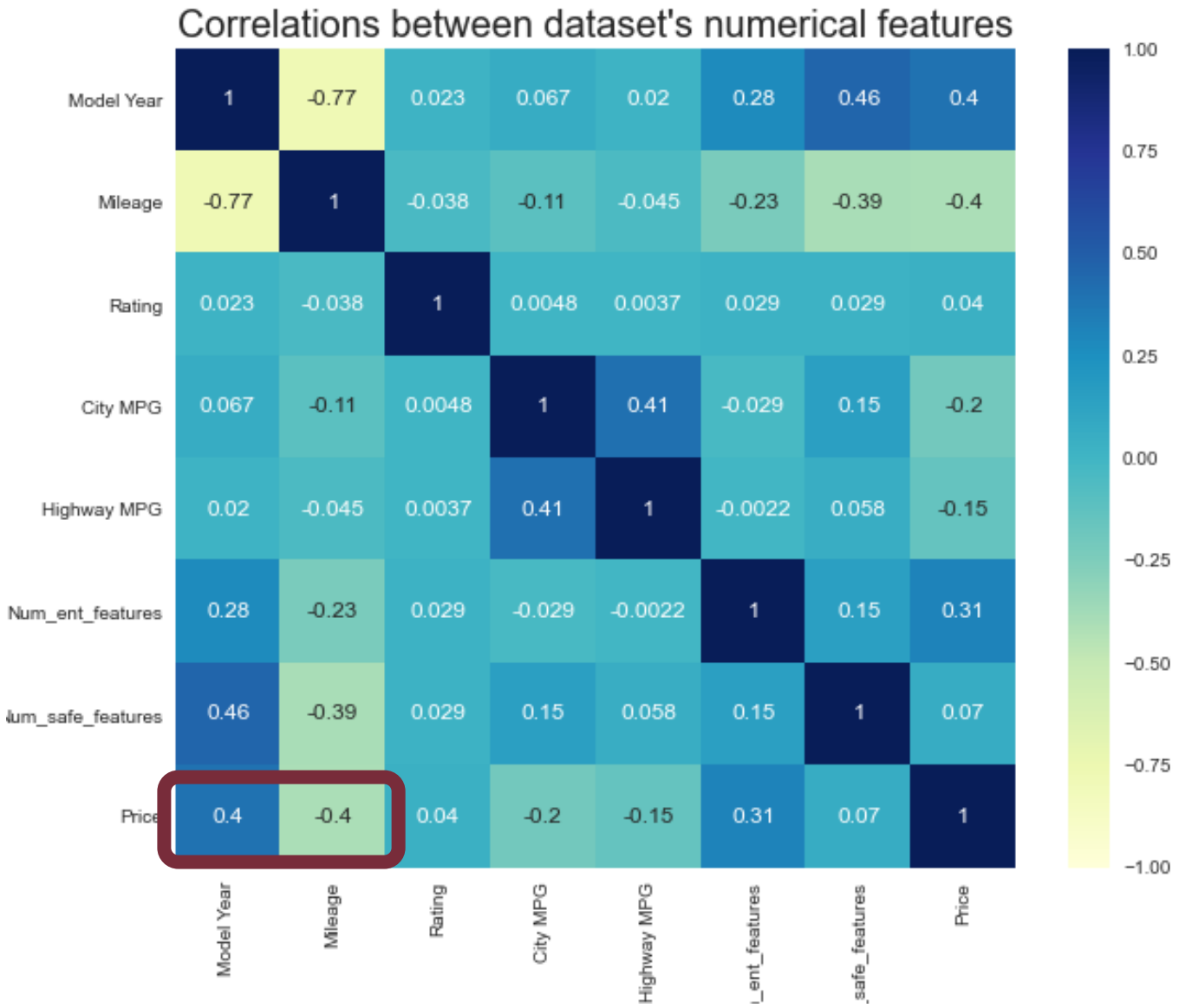
Or we could say “cars cleaning”



- 187,168 raw data points were scraped.
- Clean data set has 122,351 rows and 18 columns

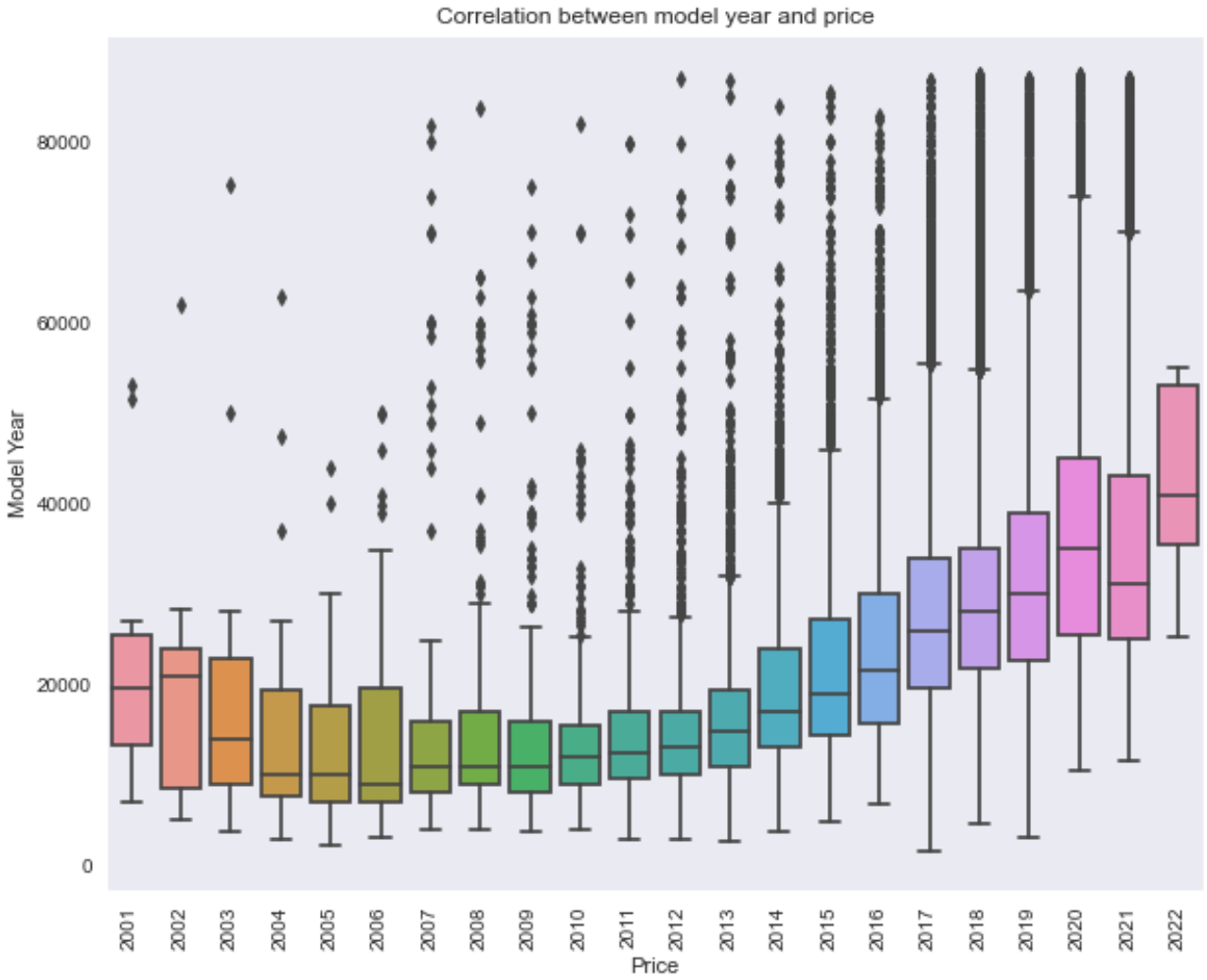
EDA

- Price is positively correlated with Model Year and negatively correlated with Mileage.
- Slightly positively correlated with num_ent_features.
- Not so much for the rest of the features.



EDA

- Price is positively correlated with Model Year.
- There are outliers all over the place.
- Generally speaking, the newer the more expensive car.



MODELS

Pre features engineered.

MODELS

Pre features engineered.

- Linear Regression Model:
  - R^2 for test set: 0.290
- Polynomial Regression Model:
  - R^2 for test set: 0.474

MODELS

Pre features engineered.

- Linear Regression Model:
  - R^2 for test set: 0.290
- Polynomial Regression Model:
  - R^2 for test set: 0.474
- Random Forest Regressor:
  - R^2 for test set: 0.577
- Gradient Boosted Regressor:
  - R^2 for test set: 0.605
- Extreme Gradient Boosting (XGBoost):
  - R^2 for test set: 0.729

MODELS

Pre features engineered.

- Linear Regression Model:
  - R^2 for test set: 0.290
- Polynomial Regression Model:
  - R^2 for test set: 0.474
- Random Forest Regressor:
  - R^2 for test set: 0.577
- Gradient Boosted Regressor:
  - R^2 for test set: 0.605
- Extreme Gradient Boosting (XGBoost):
  - R^2 for test set: 0.729

In general, they all underfit

MODELS

Work with categorical features!

MODELS

Work with categorical features!

Intuitively, car's brand (make) determines the product's price, so let's work on that.

MODELS

Work with categorical features!

- Linear Regression Model:
  - R^2 for test set: 0.505
- Polynomial Regression Model:
  - R^2 for test set: 0.695
- Extreme Gradient Boosting (XGBoost):
  - R^2 for test set: 0.870

MODELS

Work with categorical features!

- Linear Regression Model:
  - R^2 for test set: 0.505
- Polynomial Regression Model:
  - R^2 for test set: 0.695
- Extreme Gradient Boosting (XGBoost):
  - R^2 for test set: 0.870

We have better results, can we improve our performance?

MODELS

Work with categorical features!

Let's dummify the entire dataset!

MODELS

L1/L2 Regularization

K-Fold Cross-Validation!

Parameters tuning

- Linear Regression Model:
  - R^2 for test set: 0.869
  - RMSE = 4787.90
- Lasso Model:
  - R^2 for test set: 0.866
- Extreme Gradient Boosting (XGBoost):
  - R^2 for test set: 0.955
  - RMSE = 2788.11
- Ridge Model:
  - R^2 for test set: 0.865

PREDICTION

2017 Chevrolet Camaro 2SS

44,953 Mileage, Gasoline engine

City MPG 16 – Highway MPG 25

RWD – Engine 6.2L V8 – 8 speed Manual

Linear Regression Model predicts

$35,235

Extreme Gradient Boosting (XGBoost) predicts

$35,893

PREDICTION

2017 Chevrolet Camaro 2SS

44,953 Mileage, Gasoline engine

City MPG 16 – Highway MPG 25

RWD – Engine 6.2L V8 – 8 speed Manual

Linear Regression Model predicts

$35,235

Extreme Gradient Boosting (XGBoost) predicts

$35,893

True value

$38,395

PREDICTION

2018 INFINITY Q60 3.0t LUXE

18, 719 Mileage, Gasoline engine

City MPG 19 – Highway MPG 27

AWD – Engine 3.0 V6 – 7 speed Automatic

Linear Regression Model predicts

$37,604

Extreme Gradient Boosting (XGBoost) predicts

$35,340

PREDICTION

2018 INFINITY Q60 3.0t LUXE

18, 719 Mileage, Gasoline engine

City MPG 19 – Highway MPG 27

AWD – Engine 3.0 V6 – 7 speed Automatic

Linear Regression Model predicts

$37,604

Extreme Gradient Boosting (XGBoost) predicts

$35,340

True value

$32,500

CONCLUSION

L1/L2 Regularization

K-Fold Cross-Validation!

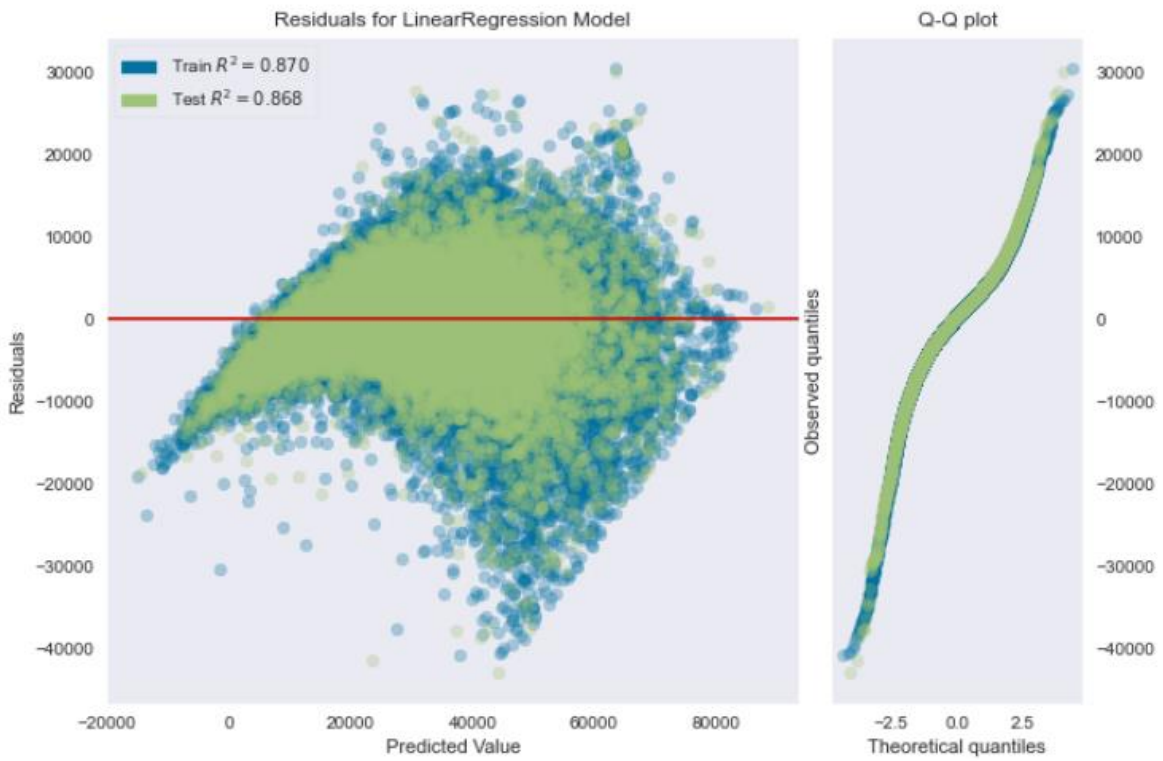
Parameters tuning

- Linear Regression Model:
  - R^2 for test set: 0.869
  - RMSE = 4787.90
- Lasso Model:
  - R^2 for test set: 0.866

- Extreme Gradient Boosting (XGBoost):
  - R^2 for test set: 0.955
  - RMSE = 2788.11
- Ridge Model:
  - R^2 for test set: 0.865

RESIDUALS

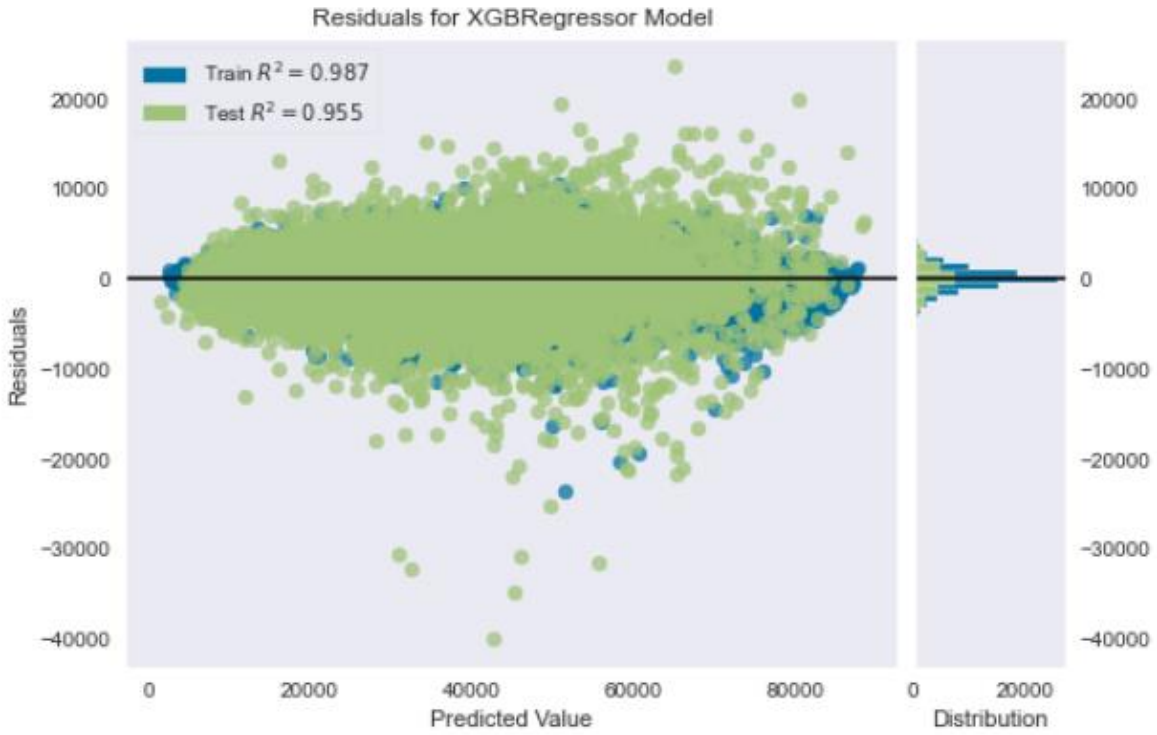
- Linear Regression Model:
  - R^2 for train set: 0.871
  - R^2 for test set: 869
  - RMSE = 4787.90



RESIDUALS

- Extreme Gradient Boosting (XGBoost):
  - R^2 for train set: 0.987
  - R^2 for test set: 0.955
  - RMSE = 2788.11

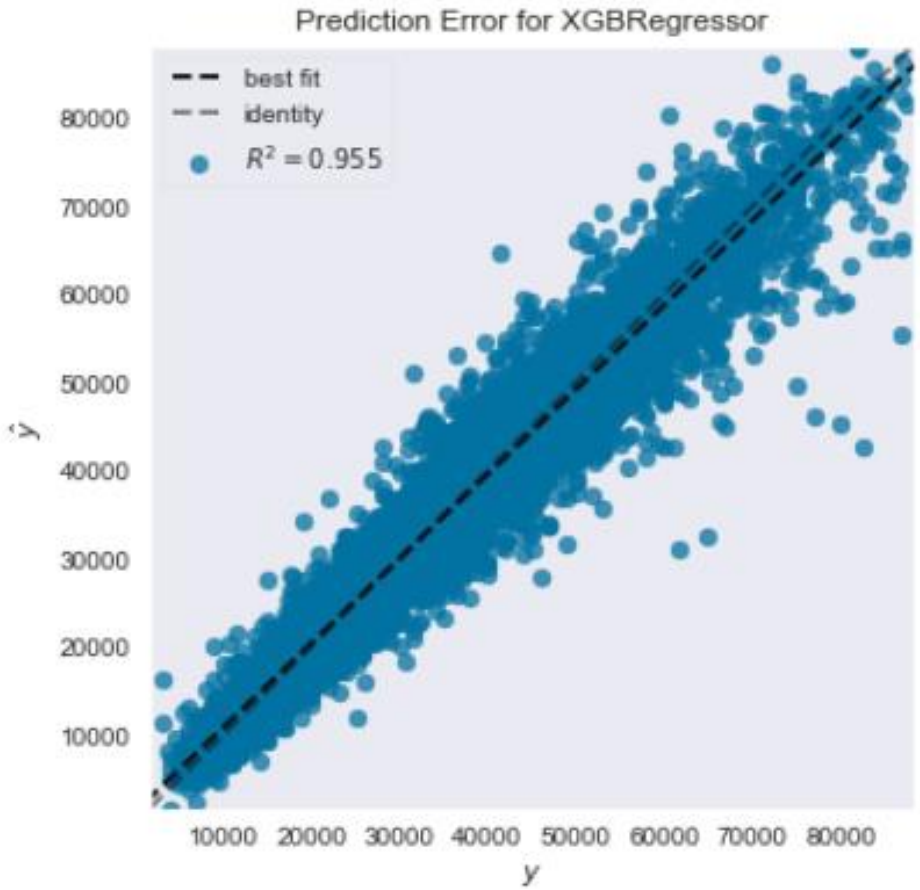
Residuals Plot



PREDICTION ERROR

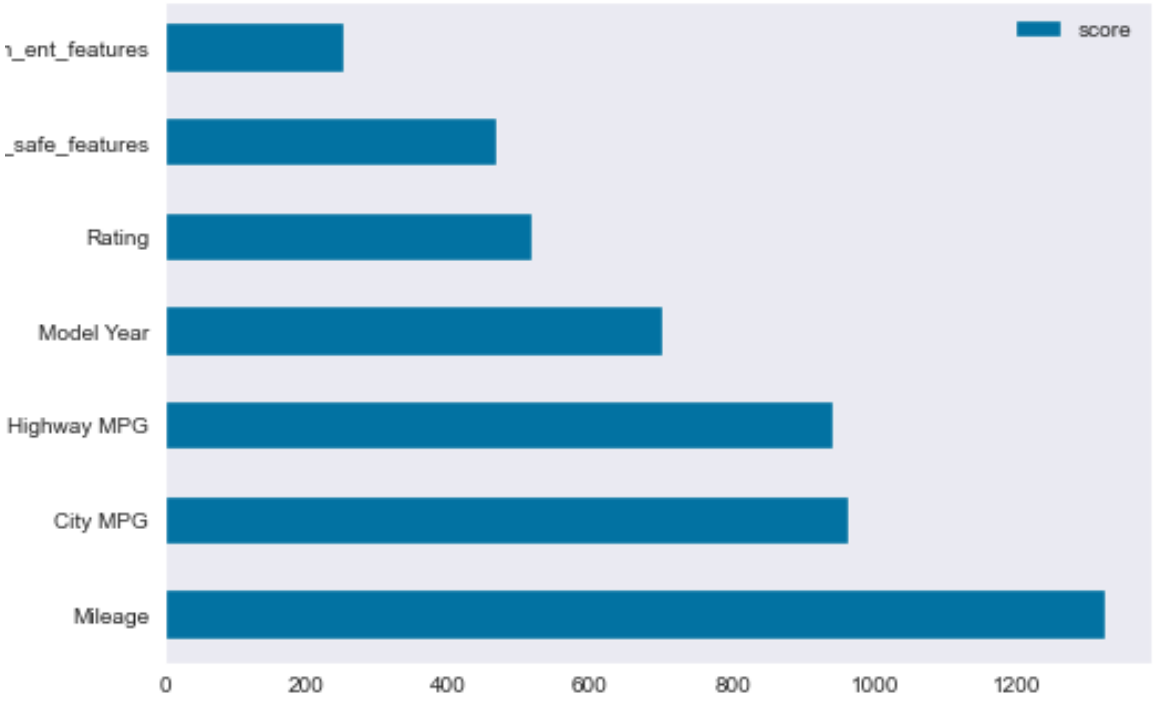
- Extreme Gradient Boosting (XGBoost):
  - R^2 for train set: 0.987
  - R^2 for test set: 0.955
  - RMSE = 2788.11

Prediction Error Plot



FEATURE IMPORTANCE

- Extreme Gradient Boosting (XGBoost):
  - R^2 for train set: 0.987
  - R^2 for test set: 0.955
  - RMSE = 2788.11



RECAP

RECAP

Linear Model

0.290 → 1 cat feature → 0.505 → L1/L2/5-Fold CV / All cat features → 0.869

RECAP

Linear Model

0.290 → 1 cat feature → 0.505 → L1/L2/5-Fold CV / All cat features → 0.869

Gradient Boosting

0.729 → 1 cat feature → 0.870 → All cat features / Params tuning → 0.955

WHAT'S NEXT?

Orignal question?

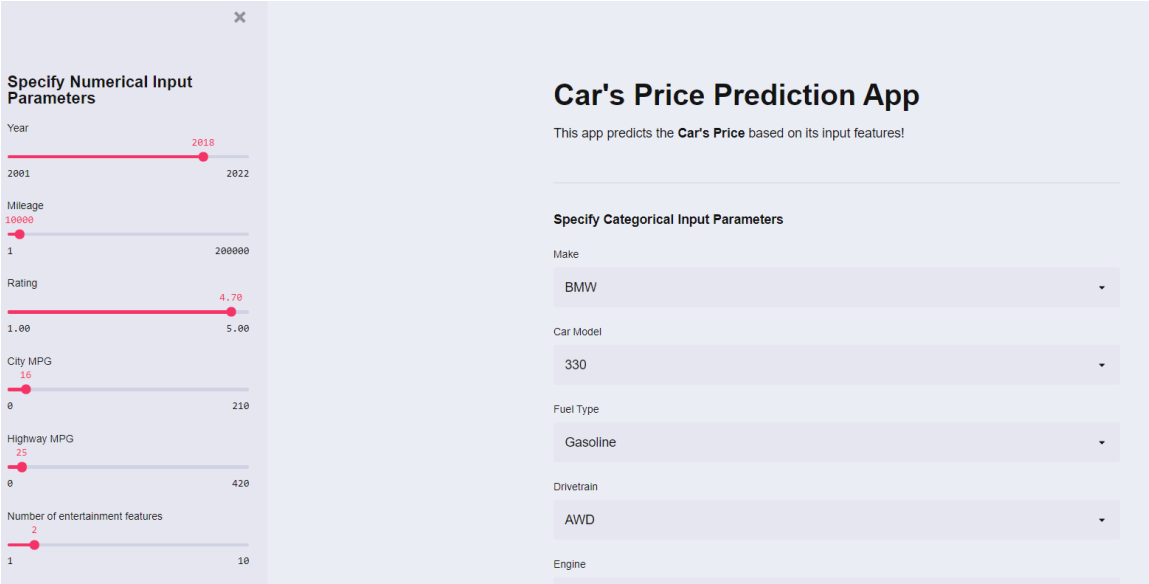
BUILD THE INTERACTIVE WEB APP AND DEPLOY IT TO THE CLOUD!

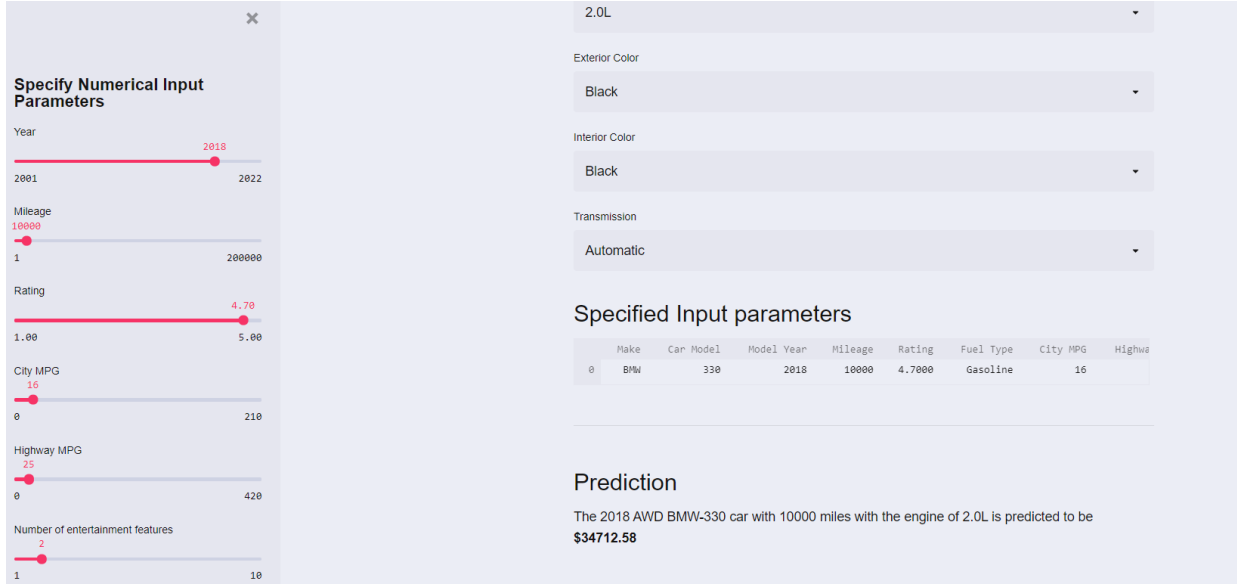
BUILD THE INTERACTIVE WEB APP AND DEPLOY IT TO THE CLOUD!

https://car-predictor-regression.herokuapp.com/

BUILD THE INTERACTIVE WEB APP AND DEPLOY IT TO THE CLOUD!

https://car-predictor-regression.herokuapp.com/





FUTURE WORK

- Incorporate geographic features to determine the price based on location.
- Explore parameters for XGBoost to get better models.

THANK YOU

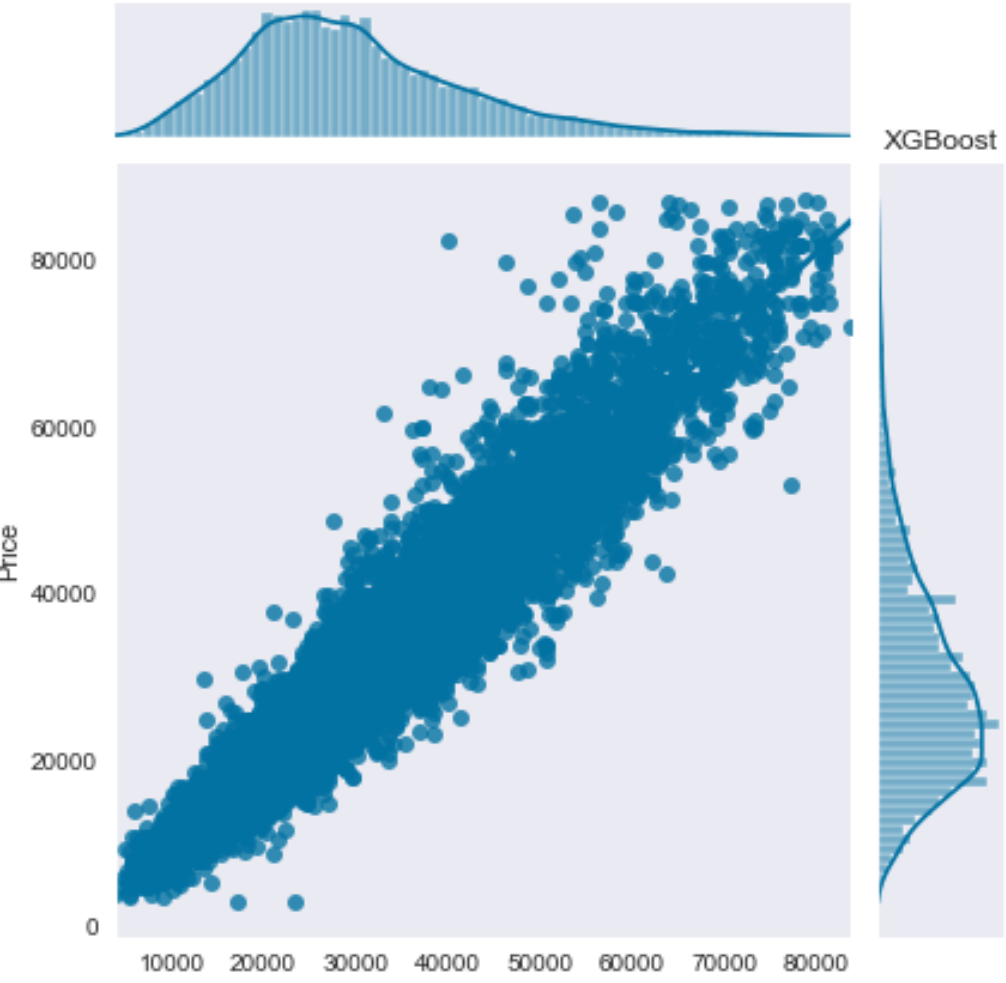
STEVEN L TRUONG

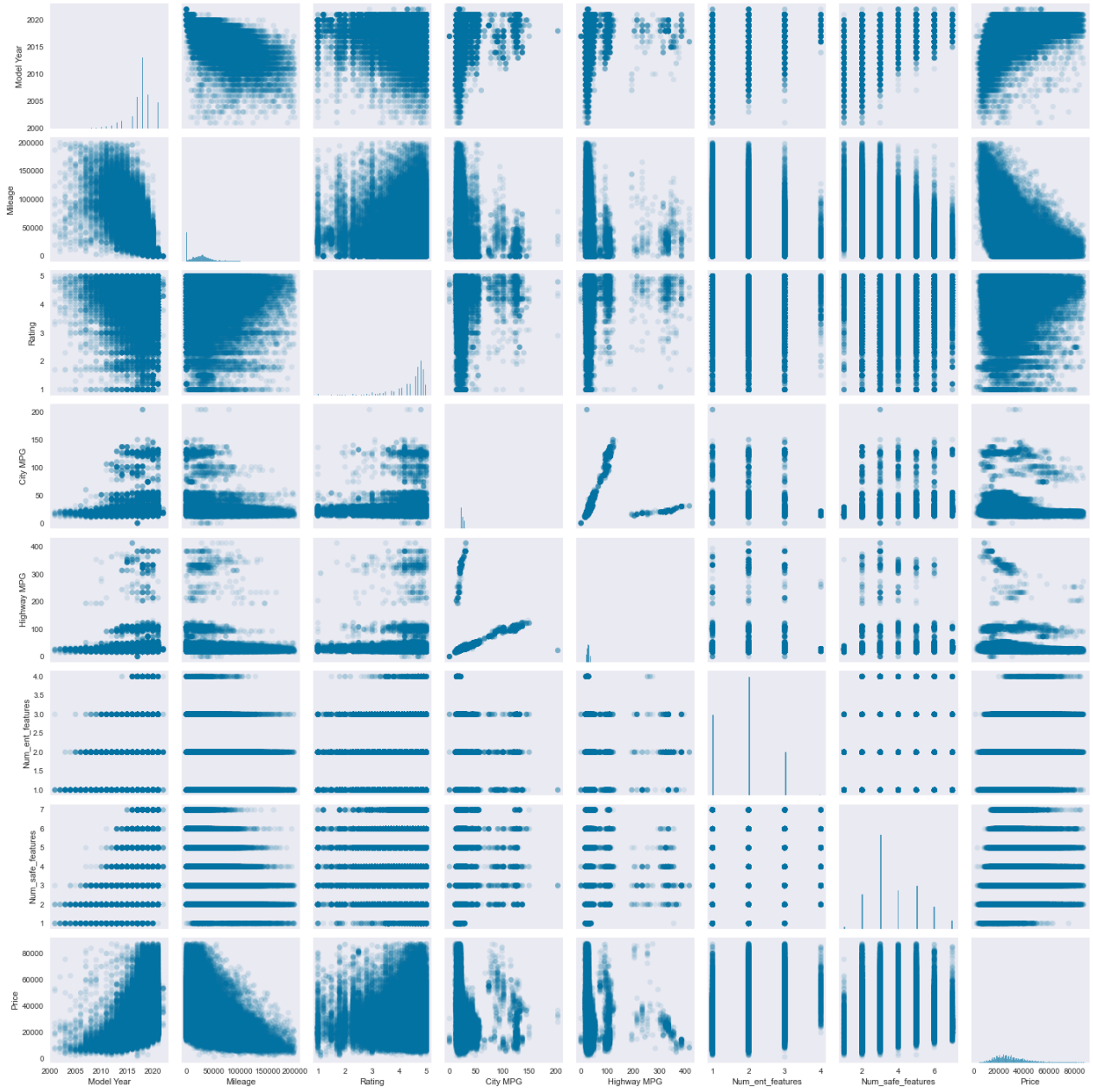
https://github.com/luongtruong77

tqluong77@gmail.com

QUESTIONS?

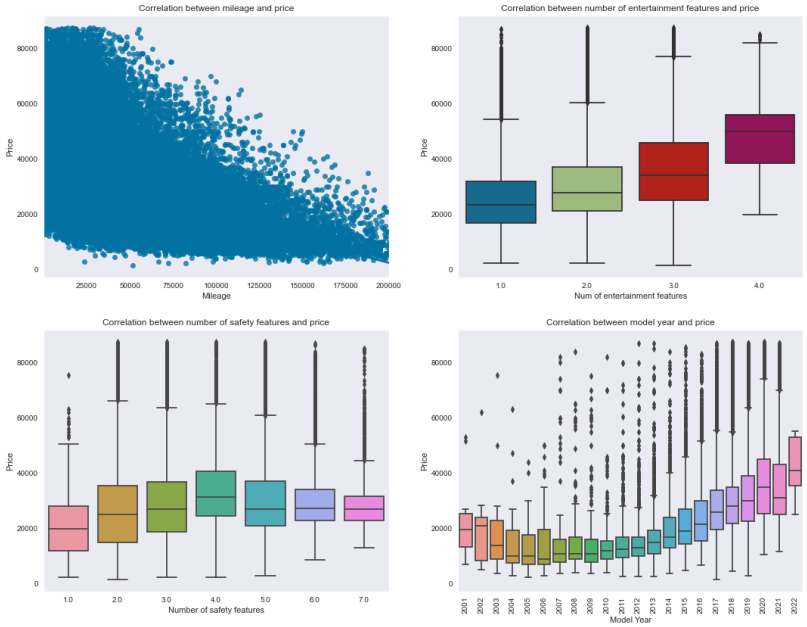
APPENDIX


XGBoost
Price
0
20000
40000
60000
80000
10000
20000
30000
40000
50000
60000
70000
80000


Model Year
Mileage
Rating
City MPG
Highway MPG
Num_ent_features
Num_safe_features
Price

APPENDIX

Parameters of XGBoost to achieve the best results so far.




```
RMSE:  2788.11

R-Squared:  0.9553
```

```
1  xgb_reg_2000
```

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.3, max_delta_step=0, max_depth=6,
             min_child_weight=1, missing=nan, monotone_constraints='()',
             n_estimators=2000, n_jobs=0, num_parallel_tree=1, random_state=0,
             reg_alpha=1, reg_lambda=1, scale_pos_weight=1, subsample=1,
             tree_method='exact', validate_parameters=1, verbosity=None)
```

Documentation on how to use streamlit to build the interactive app : https://streamlit.io/