The Price of #vanlife

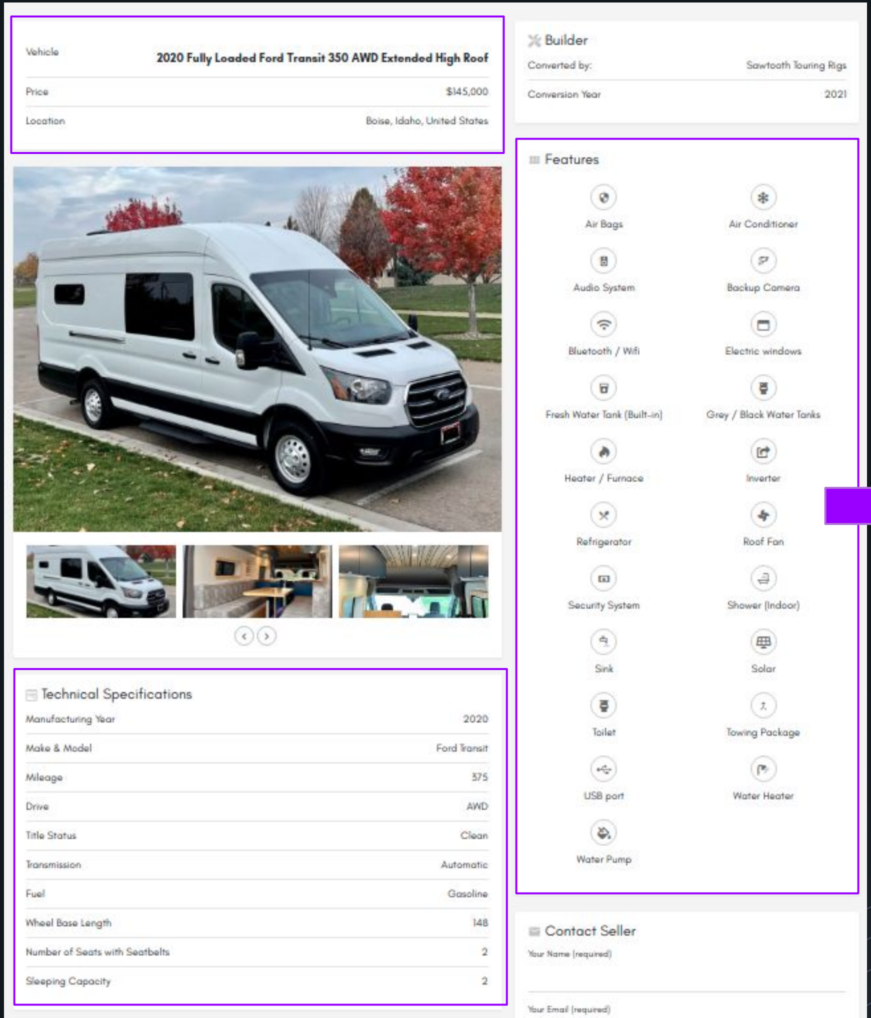
Regression Analysis by Matt Edrich

Motivation

- After purchasing an (older) Sprinter van in 2019 and converting it for #vanlife, I’ve realized it just isn’t for me!
- I did the conversion myself, and used a smorgasbord of components; the van itself has ‘certain realities’ (like mileage, age, drivetrain, and so forth)
- Determining the fair-market value of a converted van is really difficult!
- Can linear regression be used to build an model that can take into account the multitude of features a campervan may have and predict an accurate price?

Design

1. Create a web scraping pipeline to scrape vanlifetrader.com
2. Explore the raw data via Pandas and clean/preprocess data to prepare it for regression analysis
3. Conduct OLS fits on preliminary ‘clean’ training data
4. Apply a range of feature selection techniques and regression approaches to generate several models
5. Evaluate models on data held out for testing
6. Engineer features and retest



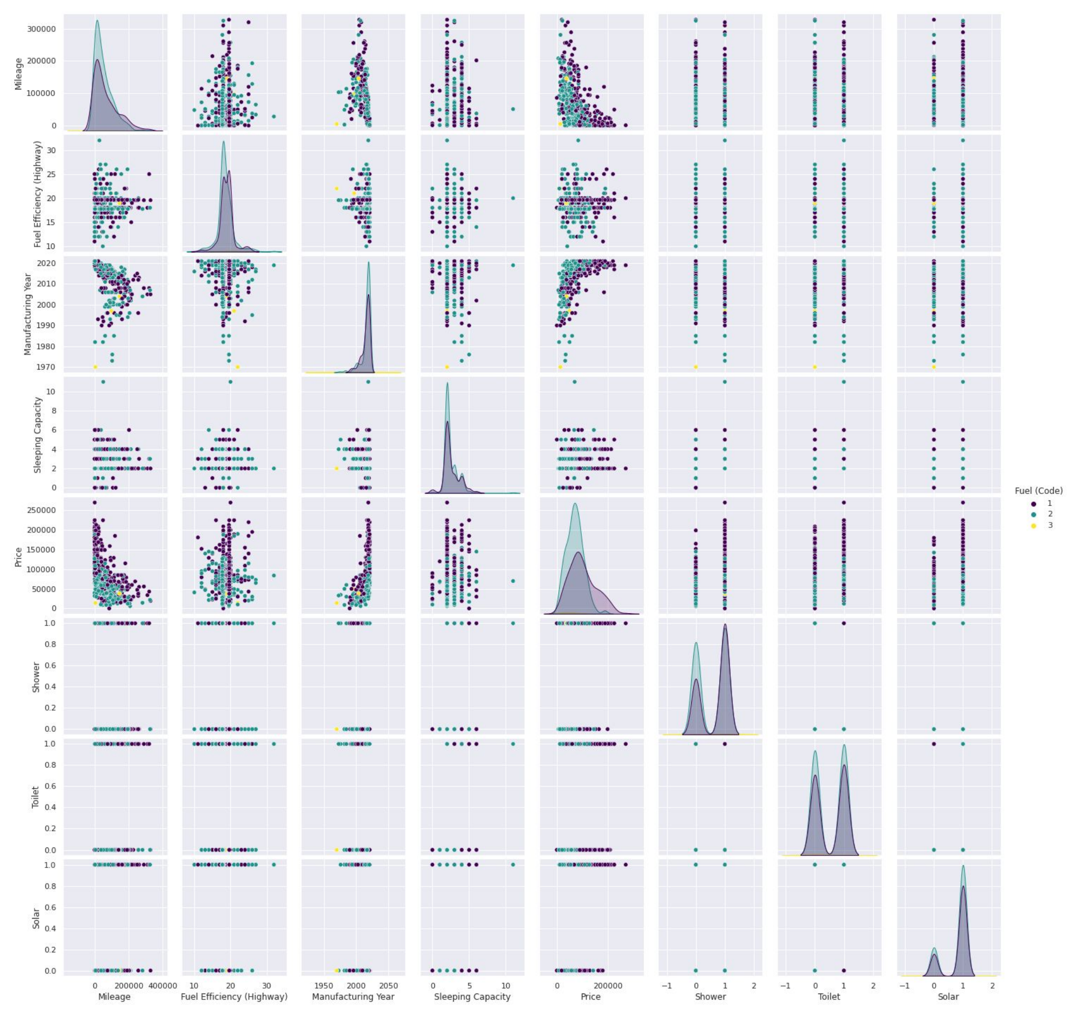

```
[82]: vans.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 593 entries, 0 to 592
Data columns (total 67 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Price                           593 non-null    float64
 1   Manufacturing Year              593 non-null    int64
 2   Mileage                         593 non-null    float64
 3   Fuel Efficiency (Highway)       593 non-null    float64
 4   Wheel Base Length               593 non-null    float64
 5   Number of Seats with Seatbelts  593 non-null    int64
 6   Sleeping Capacity               593 non-null    int64
 7   Air Bags                        593 non-null    int64
 8   Air Conditioner                 593 non-null    int64
 9   Audio System                    593 non-null    int64
 10  Backup Camera                   593 non-null    int64
 11  Bluetooth / Wifi                593 non-null    int64
 12  Electric windows                593 non-null    int64
 13  Exterior Lights                 593 non-null    int64
 14  Fresh Water Tank (Portable)     593 non-null    int64
 15  Generator                       593 non-null    int64
 16  Grey / Black Water Tanks        593 non-null    int64
 17  Heater / Furnace                593 non-null    int64
 18  Inverter                        593 non-null    int64
 19  Refrigerator                    593 non-null    int64
 20  Roof Fan                        593 non-null    int64
 21  Roof Rack                       593 non-null    int64
 22  Shower (Outdoor)                593 non-null    int64
 23  Sink                            593 non-null    int64
 24  Solar                           593 non-null    int64
 25  Toilet                          593 non-null    int64
 26  USB port                        593 non-null    int64
 27  Water Heater                    593 non-null    int64
 28  Water Pump                      593 non-null    int64
 29  Fresh Water Tank (Built-in)     593 non-null    int64
 30  Shower (Indoor)                 593 non-null    int64
 31  Stove                           593 non-null    int64
 32  Towing Package                  593 non-null    int64
 33  Heated seats                    593 non-null    int64
 34  Offroad Lights                  593 non-null    int64
 35  Awning                          593 non-null    int64
 36  Cooler                          593 non-null    int64
 37  Offroad Tires                   593 non-null    int64
 38  Security System                 593 non-null    int64
 39  Suspension Mods                 593 non-null    int64
 40  Tow Winch                       593 non-null    int64
```

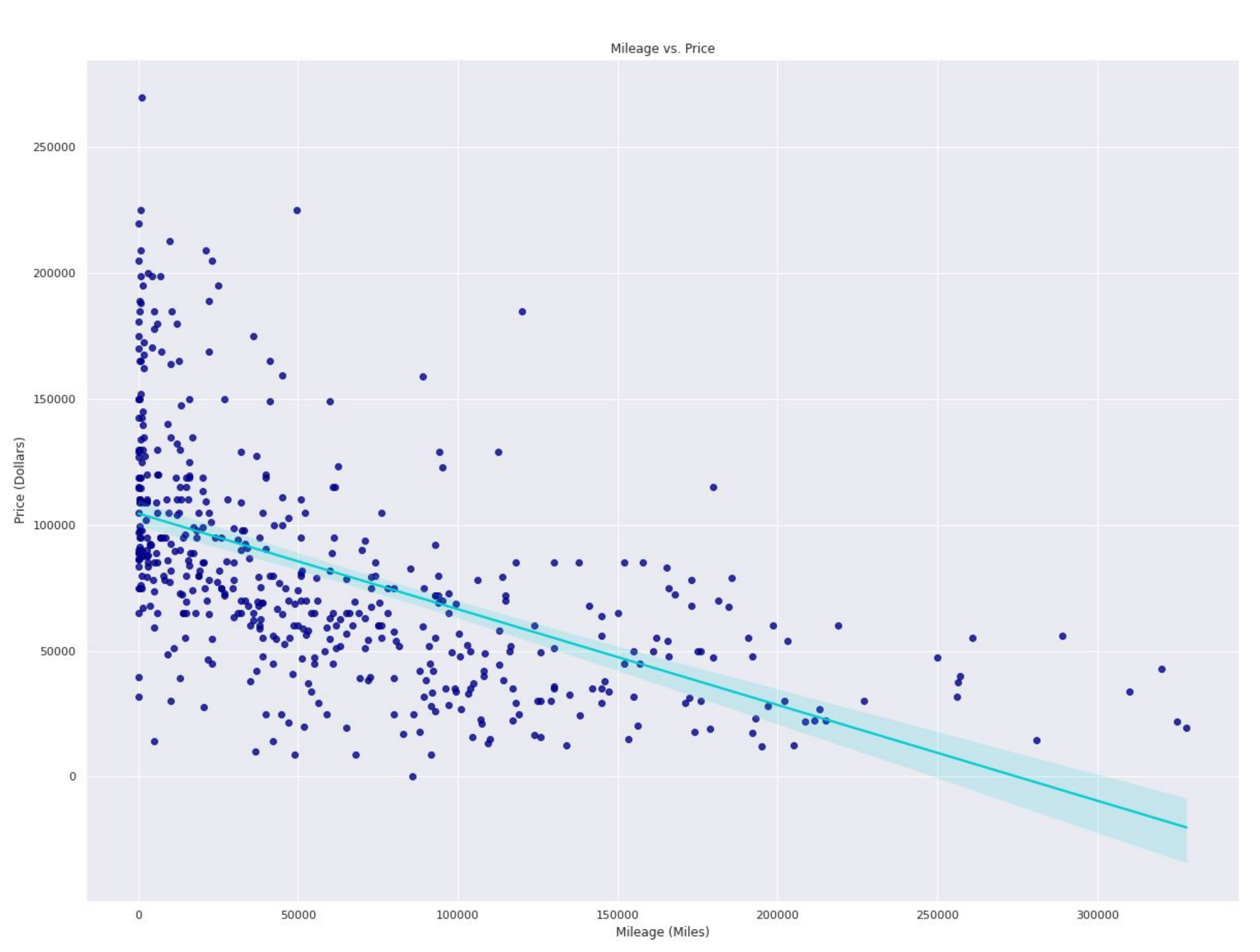
Data Cleaning

- I began with a total of 56 features
  - 40 binary
  - 7 categorical
  - 8 numeric (9 including the target)
- NaN values:
  - Mean imputation for MPG, mileage based on model year
  - Mode imputation for overall size based on model year
- **So much feature engineering**
  - Dummy variables for categorical data
  - Groupings of binary features into “feature packages”
  - Ordination of feature packages and fields like ‘drivetrain’

```
55  Shower                        593 non-null    int64
56  Fuel (Code)                   593 non-null    int64
57  Sprinter                      593 non-null    int64
58  Promaster                     593 non-null    int64
59  Transit                       593 non-null    int64
60  Manufacturing Year Binned     593 non-null    object
61  Age                           593 non-null    int64
62  Wheelbase Binned              593 non-null    object
63  Size                          593 non-null    int64
64  West                          593 non-null    int64
65  Midwest                       593 non-null    int64
66  South                         593 non-null    int64
67  Northeast                     593 non-null    int64
68  Fuel Dummy                    593 non-null    int64
69  Plumbing Score                593 non-null    int64
70  Gadget Score                  593 non-null    int64
71  Creature Comfort Score        593 non-null    int64
72  Plumbing Amenities            593 non-null    object
73  Plumbing                      593 non-null    int64
74  Gadget Amenities              593 non-null    object
75  Gadgets                       593 non-null    int64
76  Comfort Amenities             593 non-null    object
77  Creature Comfort              593 non-null    int64
78  Drivetrain                    593 non-null    int64
```

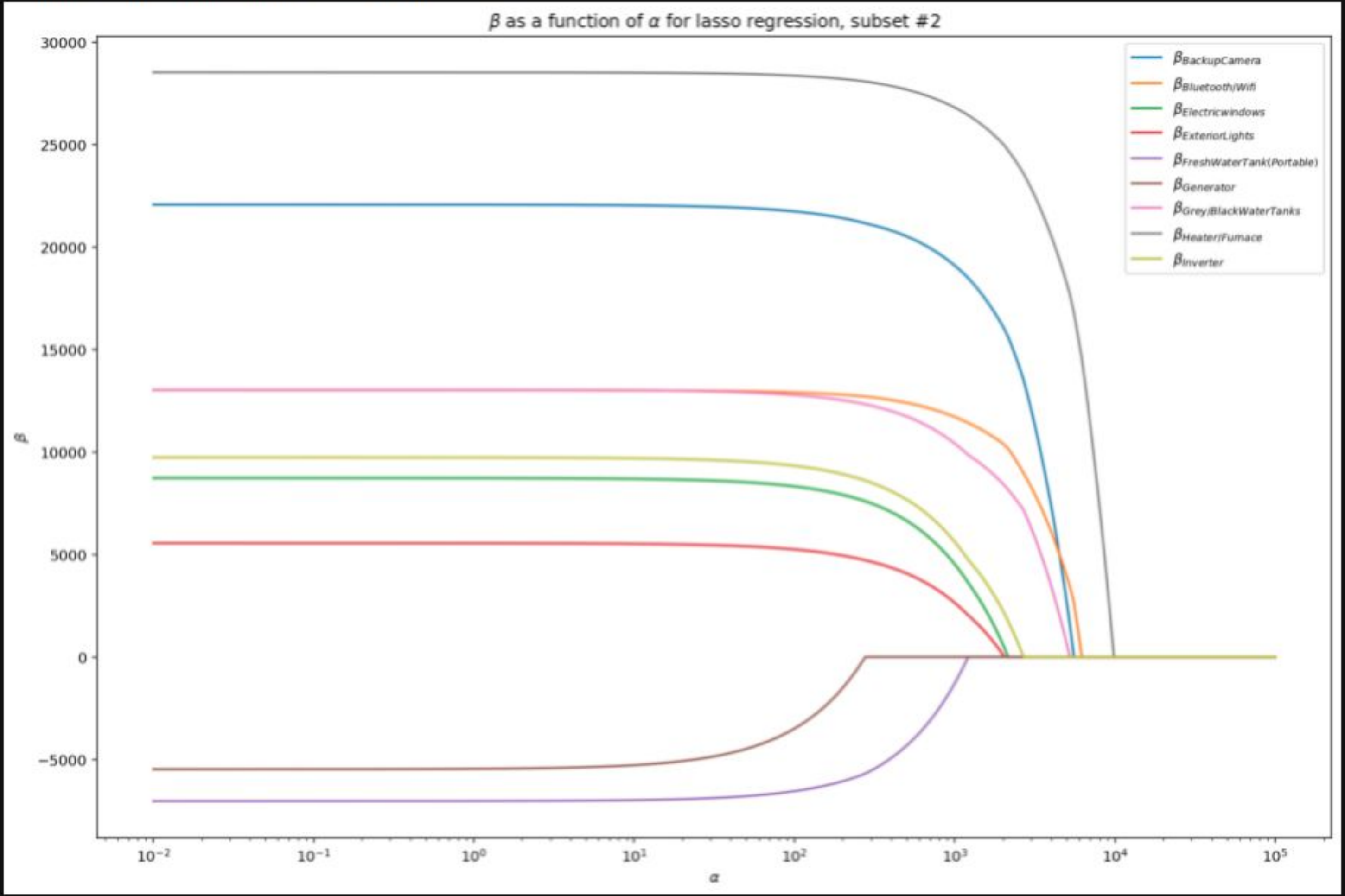
The large number of binary features I began with made pair plots and correlation matrices less than helpful!



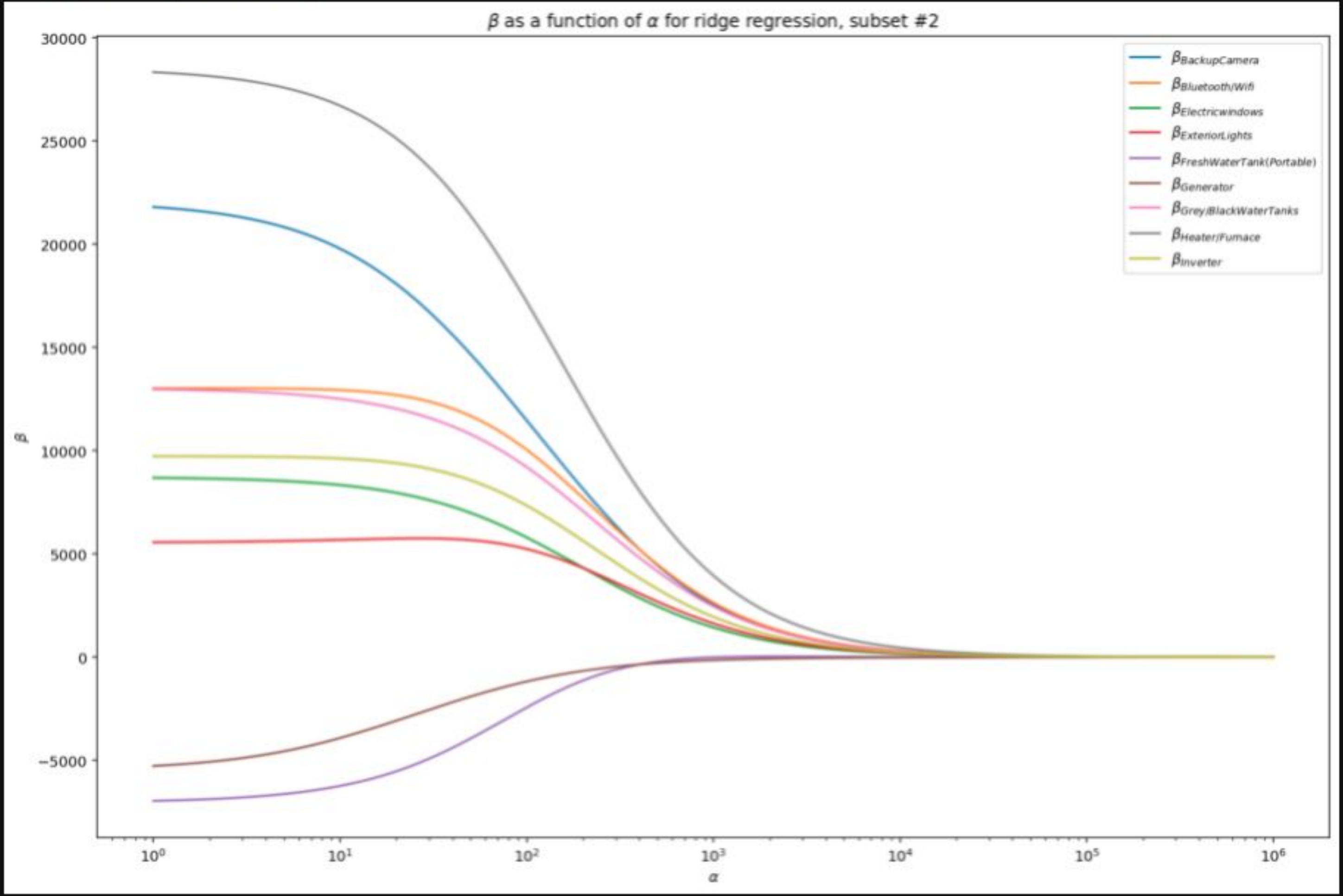


A good sanity check!

```
[9]: <matplotlib.legend.Legend at 0x7f7769e7e6d0>
```



[26]: <matplotlib.legend.Legend at 0x7f776c806160>



OLS Regression Results

Dep. Variable:	Price	R-squared:	0.745
Model:	OLS	Adj. R-squared:	0.739
Method:	Least Squares	F-statistic:	122.6
Date:	Sat, 06 Nov 2021	Prob (F-statistic):	2.20e-129
Time:	20:23:15	Log-Likelihood:	-5438.5
No. Observations:	474	AIC:	1.090e+04
Df Residuals:	462	BIC:	1.095e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-2.769e+06	3.42e+05	-8.089	0.000	-3.44e+06	-2.1e+06
Manufacturing Year	1397.6805	169.695	8.236	0.000	1064.210	1731.151
Heater / Furnace	9219.5179	2504.850	3.681	0.000	4297.207	1.41e+04
Water Heater	1.462e+04	2793.368	5.233	0.000	9128.408	2.01e+04
Roof Rack	5991.7047	2301.622	2.603	0.010	1468.759	1.05e+04
Fresh Water Tank (Built-in)	9377.6583	2935.746	3.194	0.001	3608.589	1.51e+04
Water Pump	1.09e+04	3180.475	3.426	0.001	4646.292	1.71e+04
Suspension Mods	7906.7936	3233.139	2.446	0.015	1553.313	1.43e+04
Sprinter	1.799e+04	2480.662	7.253	0.000	1.31e+04	2.29e+04
West	5899.8842	2237.336	2.637	0.009	1503.268	1.03e+04
Drivetrain	1.664e+04	1632.091	10.192	0.000	1.34e+04	1.98e+04
Mileage	-0.1821	0.021	-8.659	0.000	-0.223	-0.141

Omnibus:	66.653	Durbin-Watson:	2.033
Prob(Omnibus):	0.000	Jarque-Bera (JB):	133.904
Skew:	0.789	Prob(JB):	8.38e-30
Kurtosis:	5.071	Cond. No.	2.73e+07

```
R^2 for the training data is 0.7448396662688093
R^2 for the test data is 0.7150102188719344
The difference is 0.029829447396874875.
This model does not seem to be overfit!


The Mean Absolute Error is 16962.39174471206
The Root Mean Squared Error is 21896.350075077204
The normalized RMSE is 0.12388179779004767
```

	variable	vif
0	const	112389.008418
1	Manufacturing Year	1.568256
2	Heater / Furnace	1.305546
3	Water Heater	1.623137
4	Roof Rack	1.104949
5	Fresh Water Tank (Built-in)	1.685184
6	Water Pump	1.642191
7	Suspension Mods	1.191027
8	Sprinter	1.227141
9	West	1.042382
10	Drivetrain	1.402993
11	Mileage	1.580645

Conclusion & Next Steps

- My best performing model is “okay”:
  - Explaining roughly 70% of target variance in testing
  - Not overfit
  - Acceptable variable inflation values
  - **The error range is ~ $17,000 - $22,000**
- With more time I will:
  - Create interaction terms between variables
  - Engineer features that have non-linear relationships with the target
  - Find ways to include the categorical features that are eluding me currently
  - Cross validate this model and future versions of it
- Questions?