Predicting Value of Used Cars

Nick Lamm
Springboard Data Science Capstone Project

Problem:

- Cars are assets
- Valuation of cars is difficult
- Many listings therefore may sit for potentially years

Who cares?

- Car sales sites clogged with bad listings
- Individuals who want the most out of their assets
- Auto traders making sure they come out ahead

Factors involved

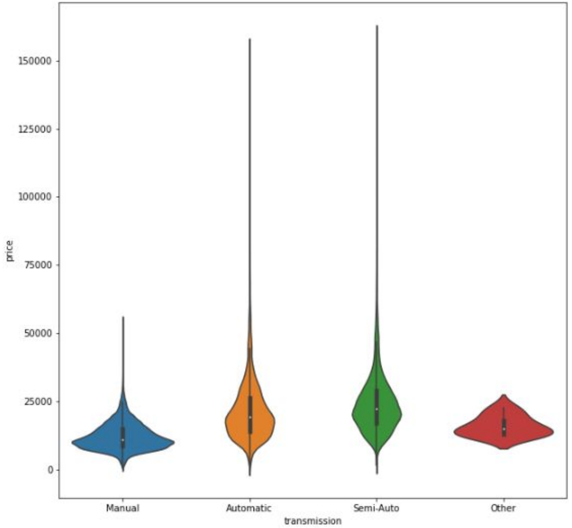
Average model price

Weights:
- Mileage
- Age
- Vehicle tax
- Optional features

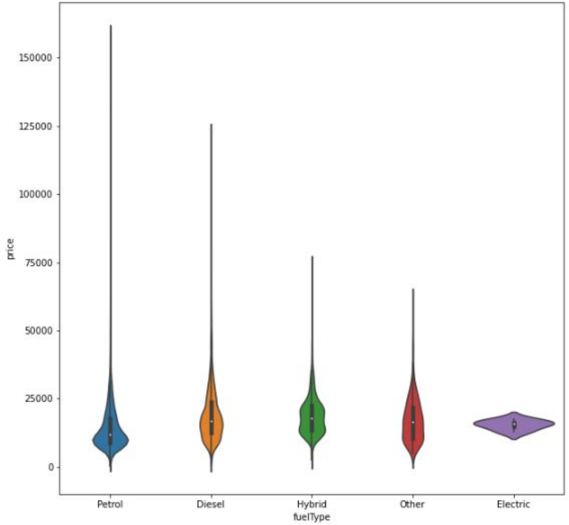
Estimated price

Data analysis

Price by transmission type



Price by fuel type



Categorical data

Sample record:

	make	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	audi	A1	2017	12500	Manual	15735	Petrol	150	55.4	1.4

```
make
unique values: 9

model
unique values: 194

transmission
unique values: 4

fuelType
unique values: 5
```

Sample record after one-hot encoding (partial shown):

	model	year	price	mileage	tax	mpg	engineSize	make_audi	make_bmw	make_ford	...	make_vw	transmission_Automatic	transmission_Manual	tr
0	A1	2017	12500	15735	150	55.4	1.4	1	0	0	...	0	0	1	

Dealing with models

- We want to avoid adding any more dimensions, let alone 195
- Solution: calculate the average price for each model

ype_Hybrid	fuelType_Other	fuelType_Petrol	avgModelPrice
0	0	0	2490.0

Training models

- Supervised learning
- Regression model
- High dimensional

price
12500
16500
11000
16800
17300

```
In [3]: data.shape
Out[3]: (97443, 25)
```

Potential models:

- Ridge regression
- Support vector regression
- Random forest
- Gradient boosting

Model comparison

	explainedVariance	maxError	MSE	r2
ridge	0.84777	100575.31161	14997268.358761	0.84777
SVR	0.811553	104391.173484	18565238.680877	0.811553
RandomForest	0.986809	67249.28	1299812.435553	0.986806
GradientBoost	0.922231	83764.217713	7682319.608025	0.92202

```
In [38]: %%time
         #random forest performs best across the board but is the most costly
         rfrModel.fit(X_train,y_train)

         Wall time: 19.2 s

Out[38]: RandomForestRegressor()

In [39]: %%time
         gbrModel.fit(X_train,y_train)

         Wall time: 7.36 s

Out[39]: GradientBoostingRegressor(loss='huber')
```

Conclusion

- Models are accurate when used to test average listings but can be hugely off for certain atypical listings
- This is due to further data that is much more difficult to collect including but not limited to:
    - Owner/usage history, beyond simply mileage
    - Cosmetic condition
    - Convenience of transaction to buyer/seller
    - Subjectivity
- Therefore any model trained on this data will be useful only as an advisor to price.