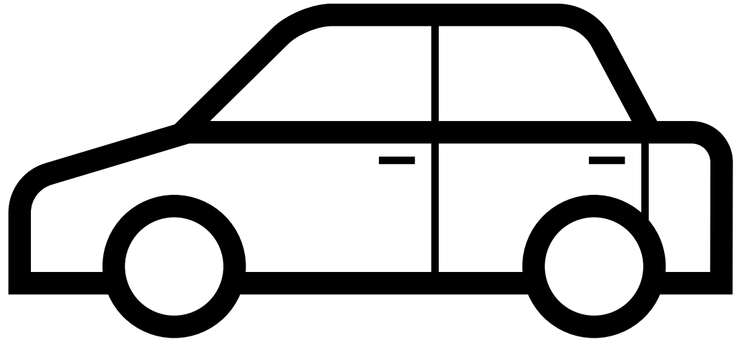
USED CAR ANALYSIS

PRESENTED BY DEV, MASON, AND LIAM

Research Question



Question:

Can we build a model that accurately predicts the selling price of used Ford Vehicles?

Data

```
RangeIndex: 2499 entries, 0 to 2498
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    2499 non-null   int64
 1   price         2499 non-null   int64
 2   brand         2499 non-null   object
 3   model         2499 non-null   object
 4   year          2499 non-null   int64
 5   title_status  2499 non-null   object
 6   mileage       2499 non-null   int64
 7   color         2499 non-null   object
 8   vin           2499 non-null   object
 9   lot           2499 non-null   int64
 10  state         2499 non-null   object
 11  country       2499 non-null   object
 12  condition     2499 non-null   object
dtypes: int64(5), object(8)
memory usage: 253.9+ KB
```

Unprocessed Data

```
Index: 745 entries, 9 to 2203
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   price     745 non-null    int64
 1   mileage   745 non-null    int64
 2   color     745 non-null    category
 3   region    745 non-null    category
 4   age       745 non-null    int64
 5   is_f-150  745 non-null    int64
dtypes: category(2), int64(4)
memory usage: 31.1 KB
```

Processed Data

Decision Tree

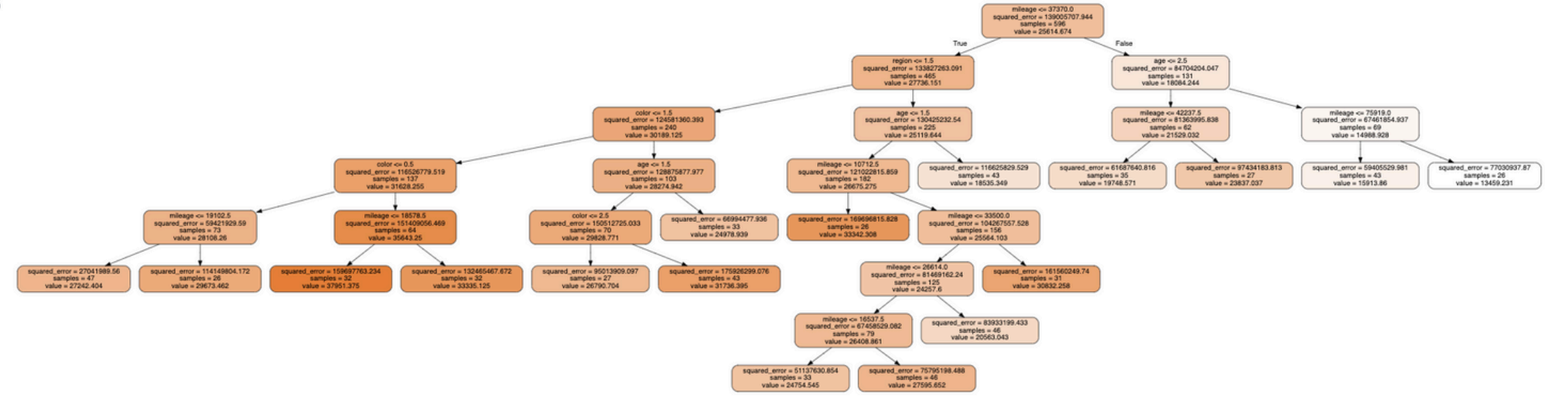
Process

- Data Prep
  - Ordinal Encode cat. vars.
- Grid Search
  - Initially tried Depth - performed terribly with optimal depth of 2
  - Then tried min_samles_leaf - resulted in a more complex tree with slightly improved performance

Results

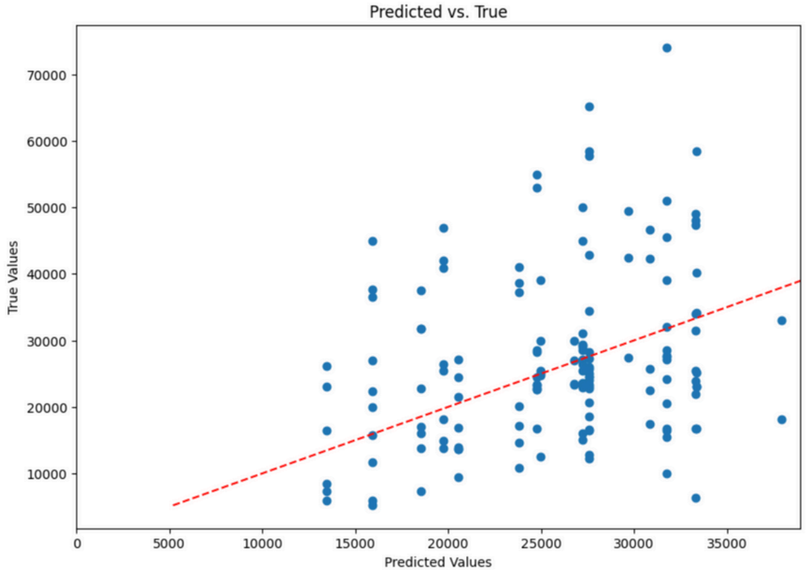
- Model really only predicts values near the true mean
- Given our data's lack of variables, decision trees are not the best model to predict price

```
DecisionTreeRegressor(min_samples_leaf=np.int64(26), random_state=30)
```



RMSE: 12468.90

R^2: 0.0362 (yikes)



Linear Regression

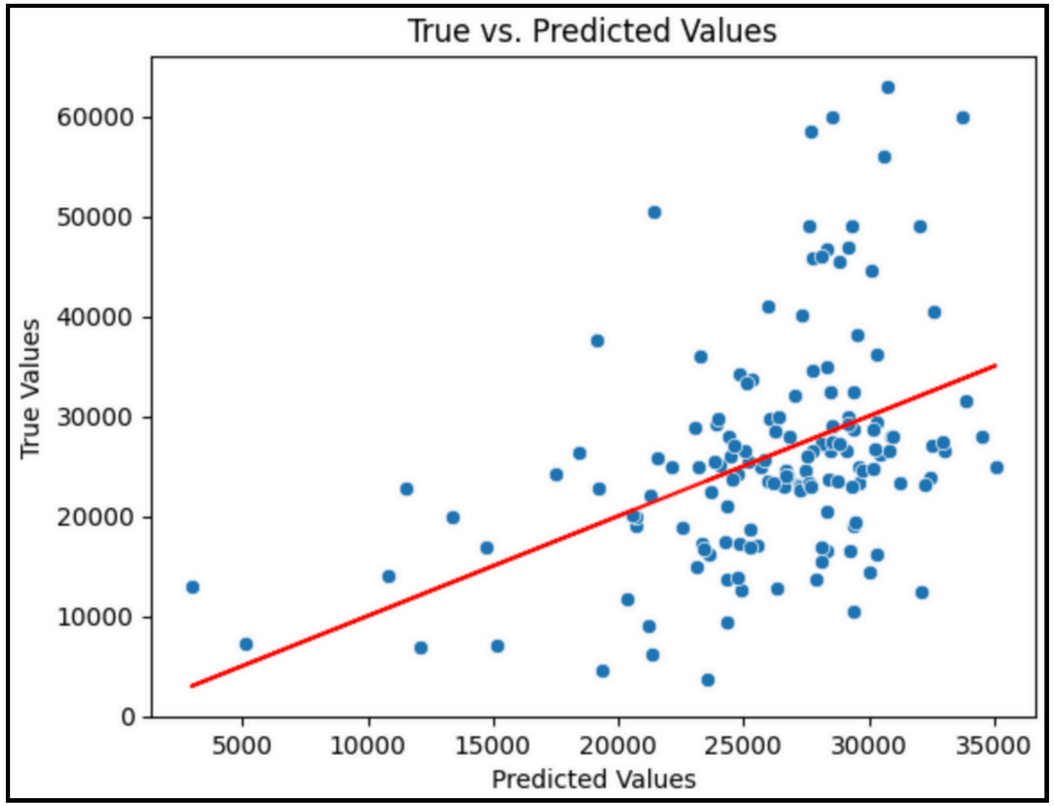
Process

- Data Prep
  - Data Cleaning
  - One-Hot Encoding
- Linear Regression

Results

- Intercept = **35968**
- Coefficient Significance
- Plot True Values vs Predicted Values
- Plot Residuals

	Coefficient
mileage	-12727.271494
age	-19322.967653
is_f-150	1123.478403
color_blue	-1722.475093
color_gray	-4539.654439
color_off-white	1672.448957
color_other	-2017.103434
color_red	-4415.254704
color_silver	-6964.419820
color_white	-4188.232163
region_Northeast	1279.651486
region_South	-2186.939664
region_West	-1162.949331



R2 Score: 0.156

RMSE: 10299

Polynomial Regression

	Degrees	Train R2 Score	Test R2 Score	Train RMSE Score	Test RMSE Score
0	2	0.293552	1.524941e-01	10235.965806	1.031921e+04
1	3	0.435662	-3.525838e+00	9148.679850	2.384649e+04
2	4	0.564678	-8.109953e+04	8035.152523	3.192176e+06
3	5	0.692176	-3.754630e+07	6756.780101	6.868447e+07
4	6	0.752294	-1.677408e+10	6061.183614	1.451758e+09
5	7	0.777700	-7.098280e+11	5741.938106	9.443903e+09
6	8	0.784942	-2.921340e+13	5647.629831	6.058512e+10

2 Degrees

R2 Score: 0.152

RMSE: 10236

Process

- Same data prep as linear regression
- For loop to exand the features by n degrees

Results

- Model with degrees = 2 performed best
- Significant overfitting
  - Training scores improved
  - Test scores became extreme

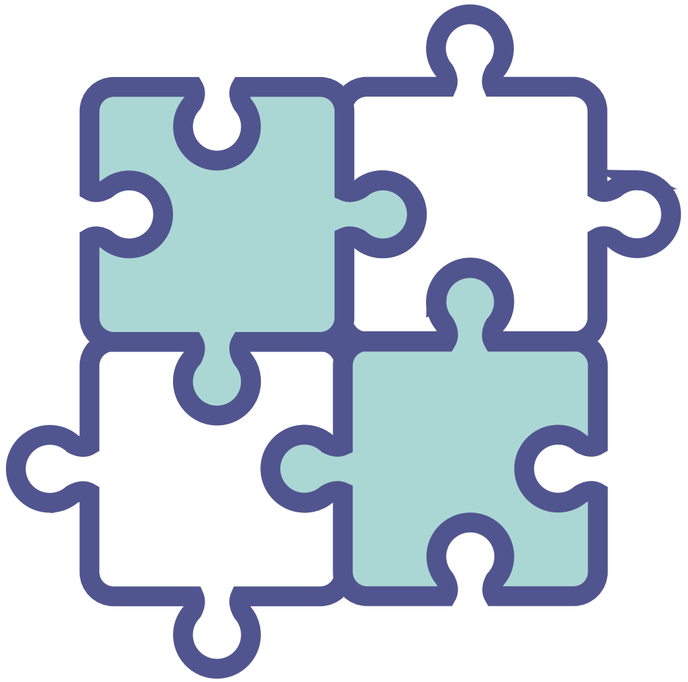
CONCLUSION

Key Takeaways

- Data did not have many useful features to make great predictions
- Linear Regression was the best performing model
    - 2-Degree Polynomial Regression performed similarly

Future Recommendations

- More data scraping or feature engineering in order to build a better model
- Focus on a single car make and model
    - Ex. Ford F-150s



Questions?