Ames Housing

Modeling and Analysis

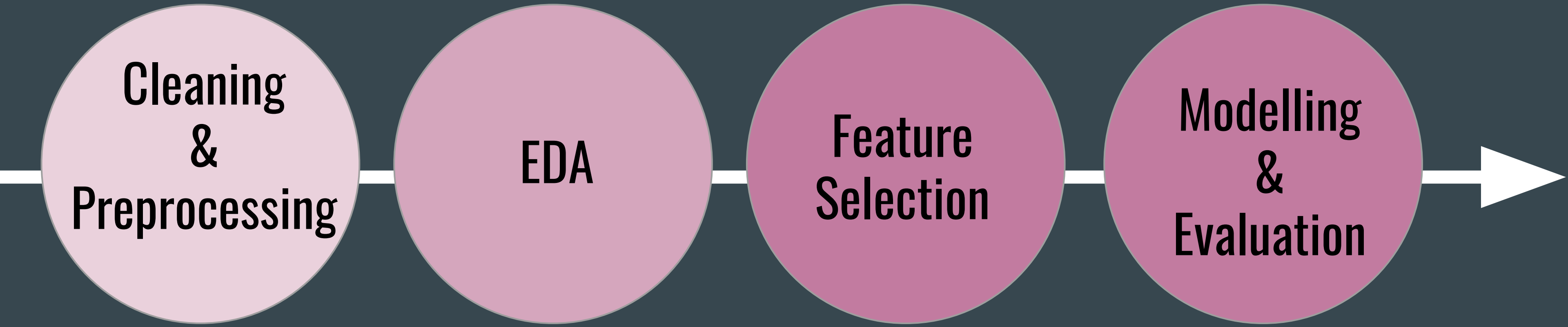
Problem statement

Determine the best model for predicting Sale Price for houses in Ames (R^2 of at least 0.81, and should generalize well to new data within the Ames area)

Use the model to answer the following:

1. What features add the most value to a house, and which hurt house values most?
2. With a set of features, what is the expected sale price of a house?
3. Given a budget, what kind of house would one be able to afford?

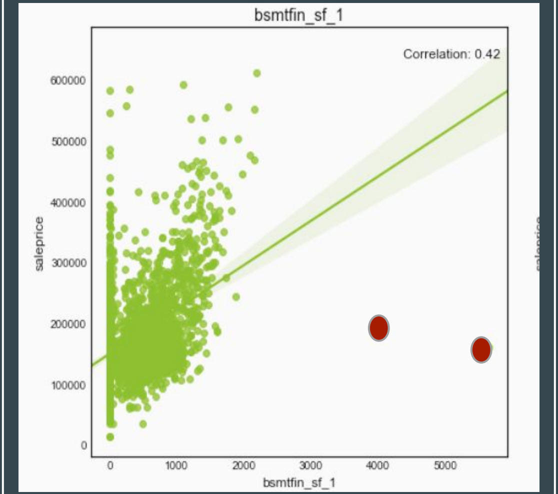
Workflow


Cleaning
&
Preprocessing
EDA
Feature
Selection
Modelling
&
Evaluation

Data Cleanliness and Encoding

Outliers

- Eliminate Outliers.



"Missingness"

- Replace with 0 / "NA" if NaN.
- Simple / Iterative Imputation if truly missing.

Dummying

- Categorical variables were one-hot encoded.
- Variables like "Bsmt Qual" were transformed as Likert scales.

Code Snippets

1

Drop columns with >80% zero or a single value

```
col_to_drop = ['alley','miscval','lowqualfinsf','street',\
               'utilities','condition2','roofmatl',\
               'heating','centralair','electrical',\
               'paveddrive','fence','saletype','bsmthalfbath',\
               'bsmtfintype2','bsmtfinsf2','bsmtcond','extercond',\
               'garagequal']
dropcol(df, col_to_drop)
```

2

Create ‘presence-absence’ columns

```
# PORCH
col_porch = ['3ssnporch','enclosedporch','openporchsf','screenporch']
df['porchpres'] = df[col_porch].sum(axis=1)\
                               .apply(lambda x: 1 if x > 0 else 0)
dropcol(df, col_porch)
```

3

Convert ordinal to numerical

```
def map_new_vals(colname,dictionary):
    df[colname] = df[colname].map(dictionary)
lotshape_di = {'Reg': 0,
               'IR1': 1,
               'IR2': 2,
               'IR3': 3}
map_new_vals('lotshape', dictionary = lotshape_di)
```

4

Add new columns

```
# AGE SOLD
for index, val in enumerate(df['yearbuilt']):
    if val == df.loc[index, 'yrsold']:
        df.loc[index, 'age_sold'] = 0
    else:
        df.loc[index, 'age_sold'] = df.loc[index,'yrsold'] - val
```

5

Impute missing values

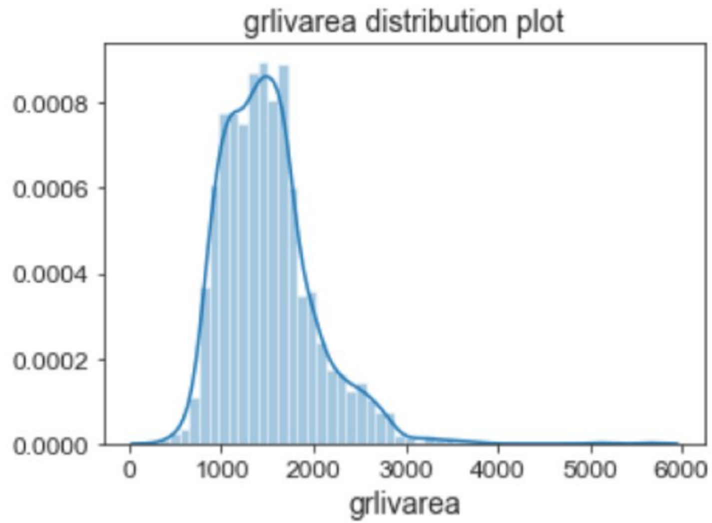
```
#This function uses sklearn's iterative fill to imput missing v
imp = IterativeImputer(missing_values = np.nan, estimator = est
rs = RobustScaler()
rs.fit_transform(X)
imp.fit(X)
```

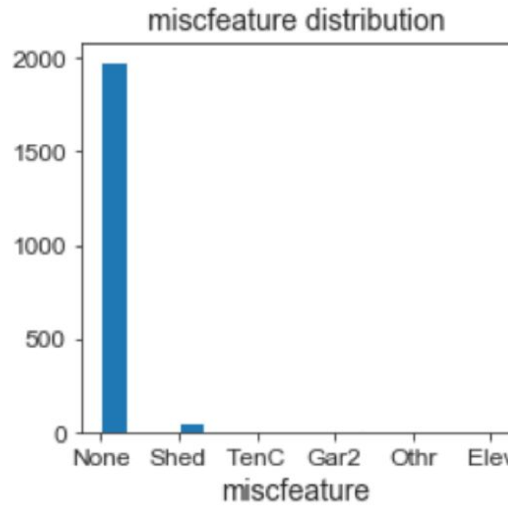
6

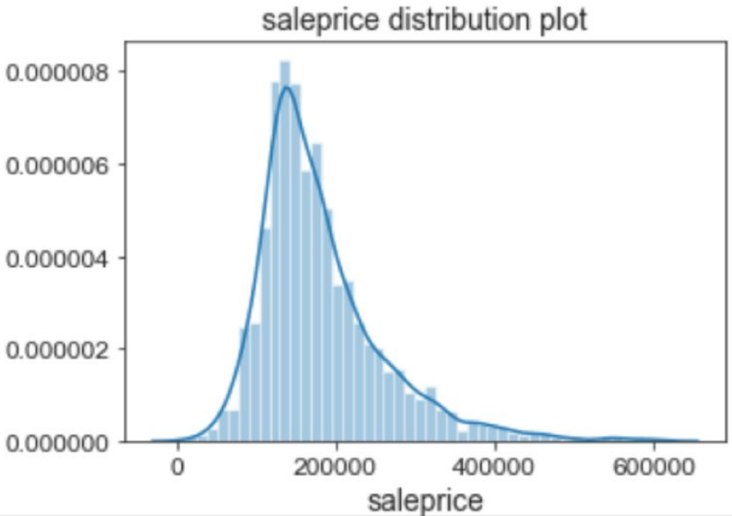
Remove outliers

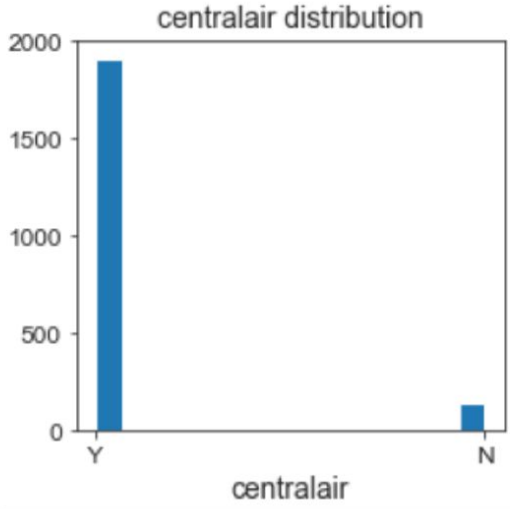
```
df.drop(df[df['grlivarea'] > 4_500].index, inplace = True)
df.drop(df[df['lotfrontage'] > 300].index, inplace = True)
df.drop(df[df['lotarea'] > 100_000].index, inplace = True)
```

Heavily skewed columns

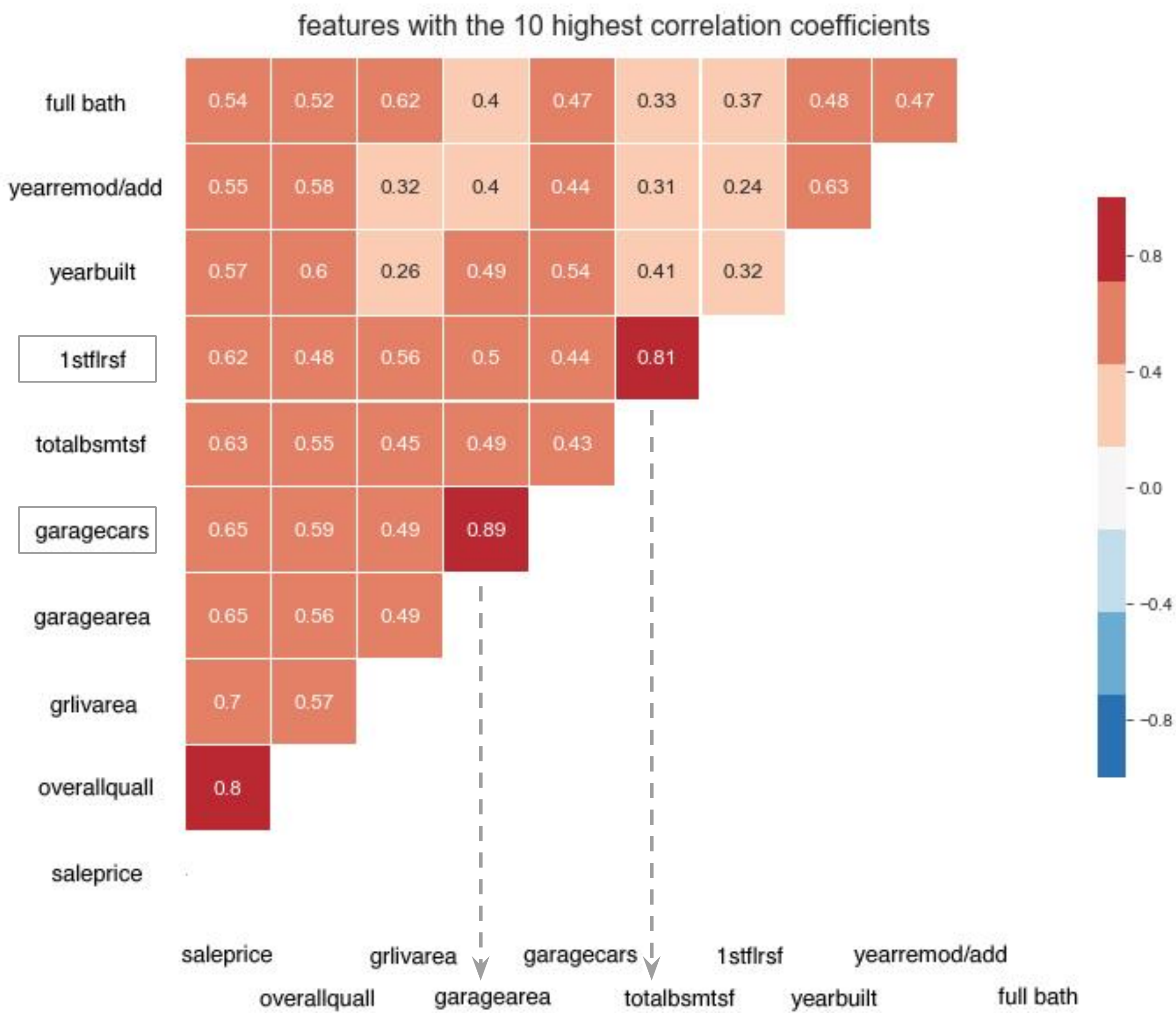








Multicollinearity



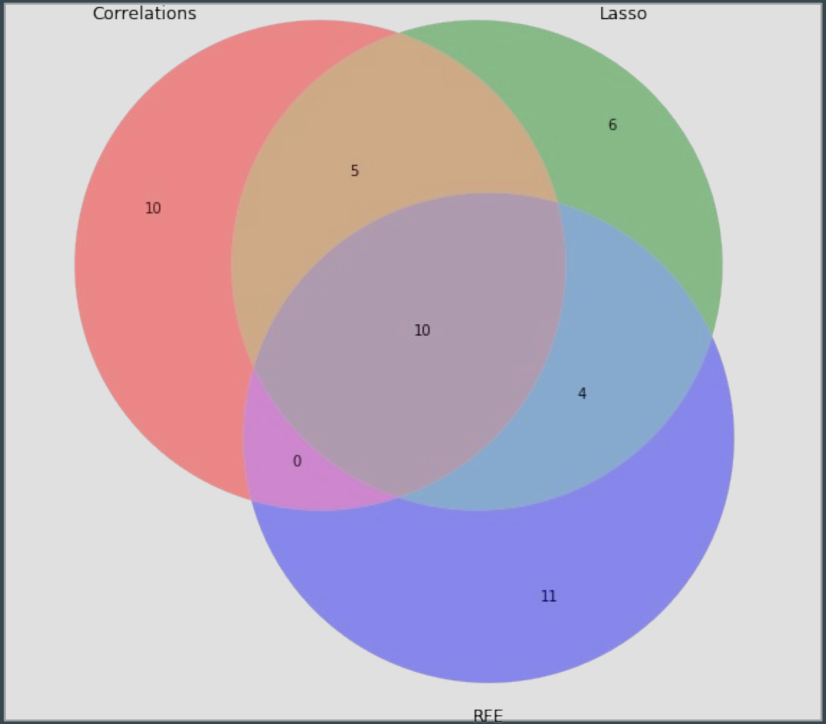
Feature Selection

Overlap of Feature Selection Methods

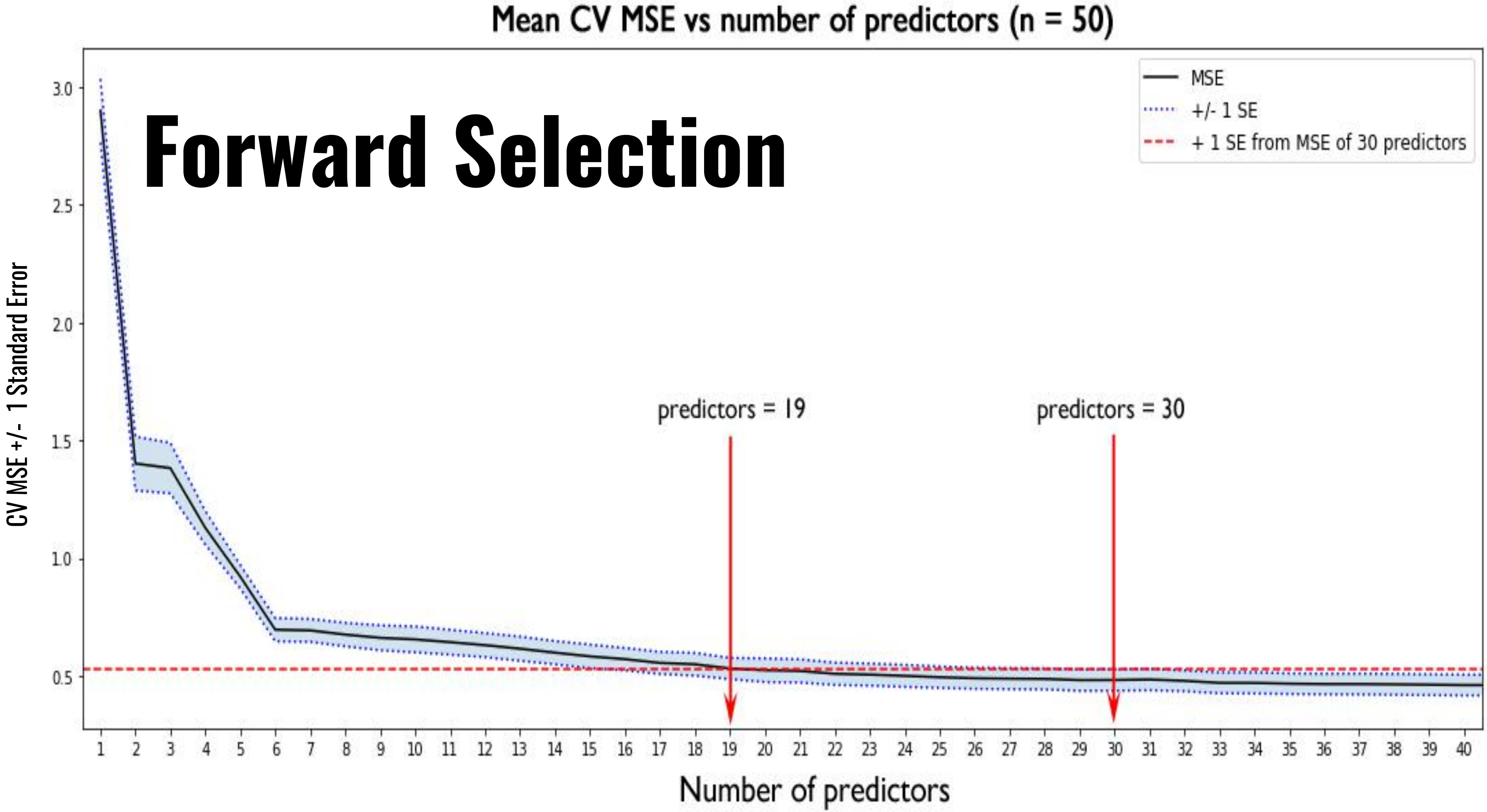
Three feature selection methods were used:

1) Filter (by correlation)
2) Wrapper (Recursive Feature Elimination)
3) Embedded (Lasso)

The features from all three methodologies were compared, and returned a list of 10 features that were shared.



Forward Selection



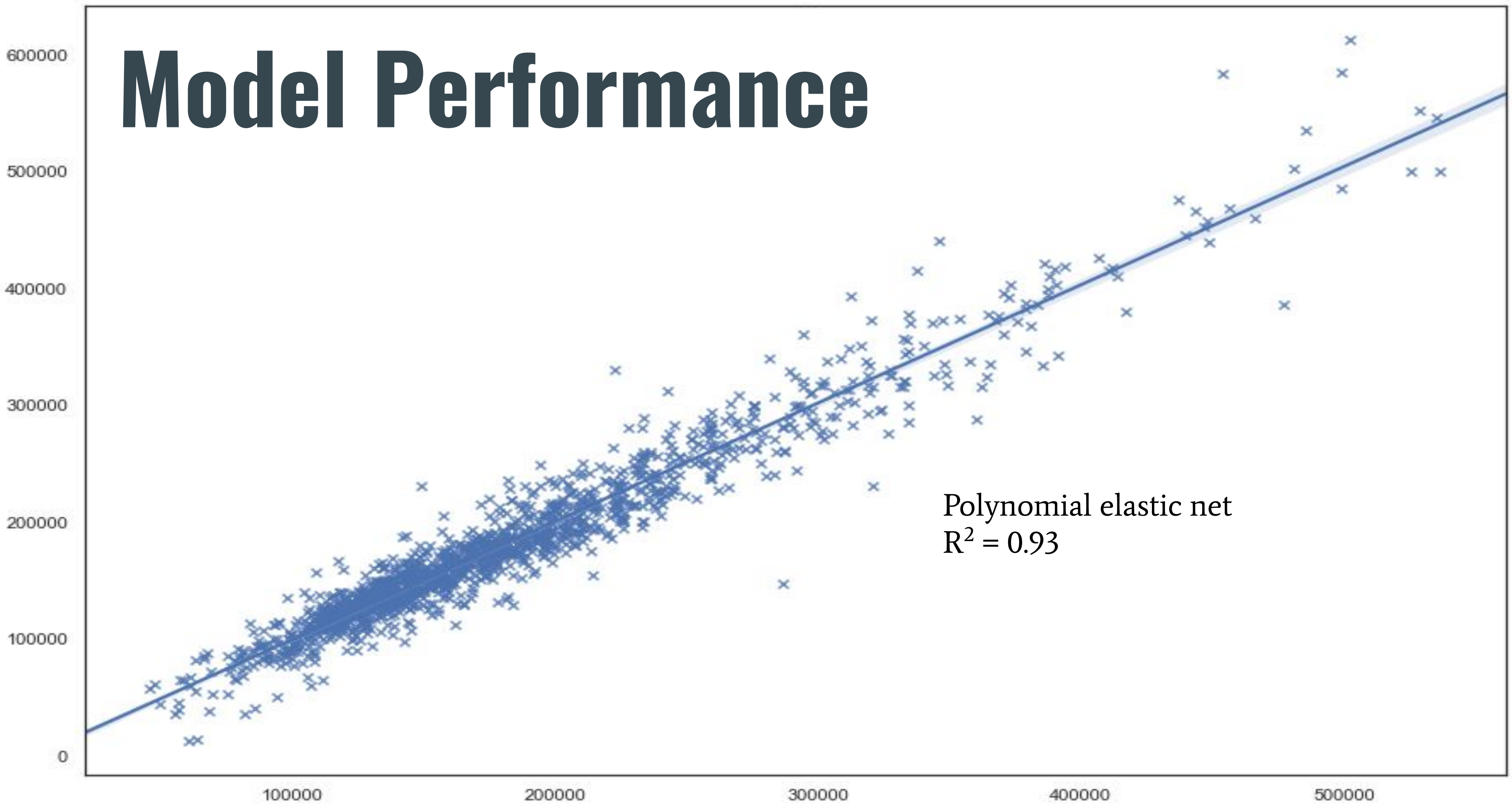
Comparing Model R2

Feature Selection Method

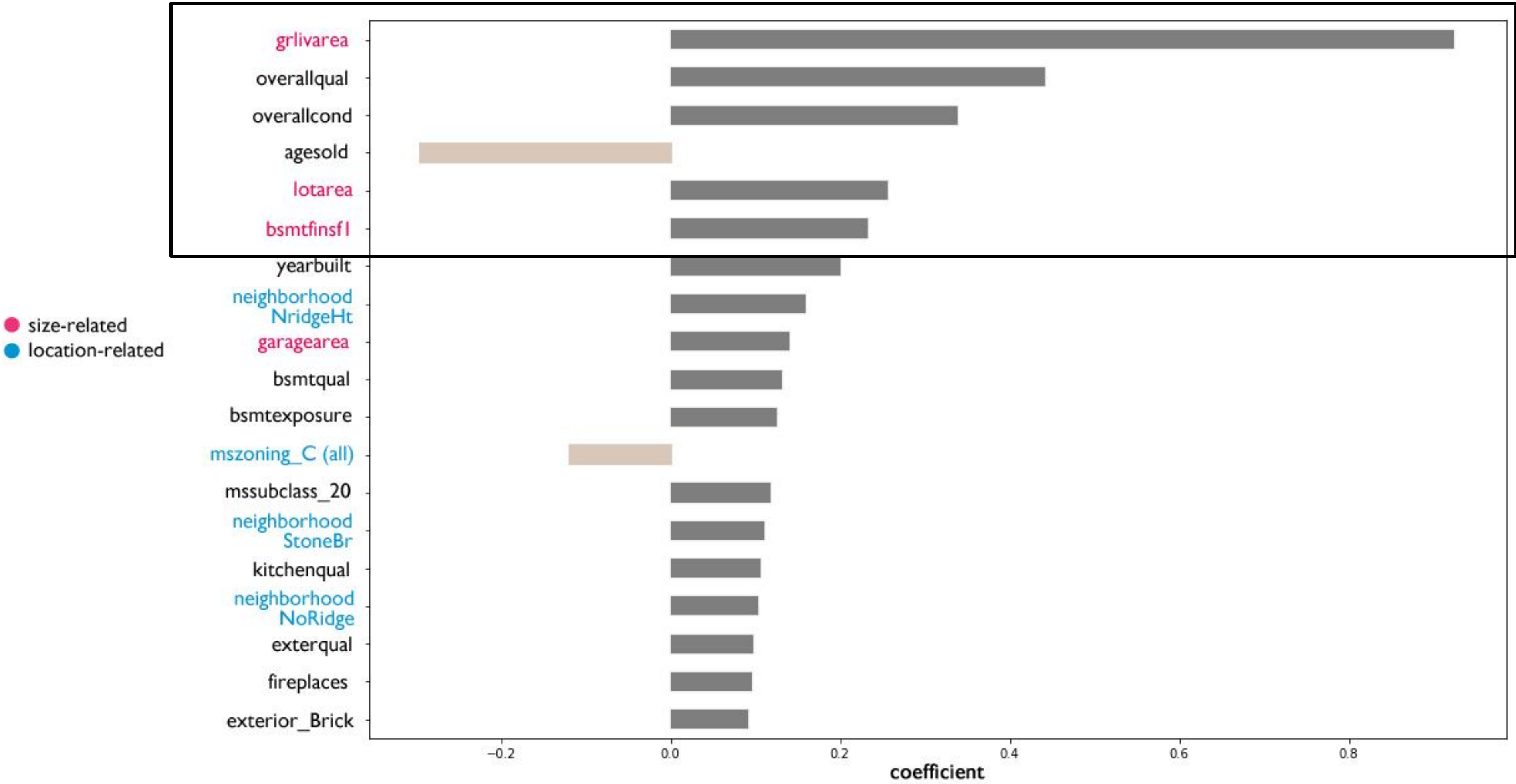
Regularization Models

	Filter	Embedded	Wrapper	Combined	Forward Selection
Linear	-	-	-	-	0.81
Ridge	0.87	0.89	0.89	0.86	0.918
Lasso	0.87	0.90	0.89	0.86	0.918
Enet	0.87	0.90	0.89	0.86	0.919
Poly Enet	-	0.93	-	0.90	-

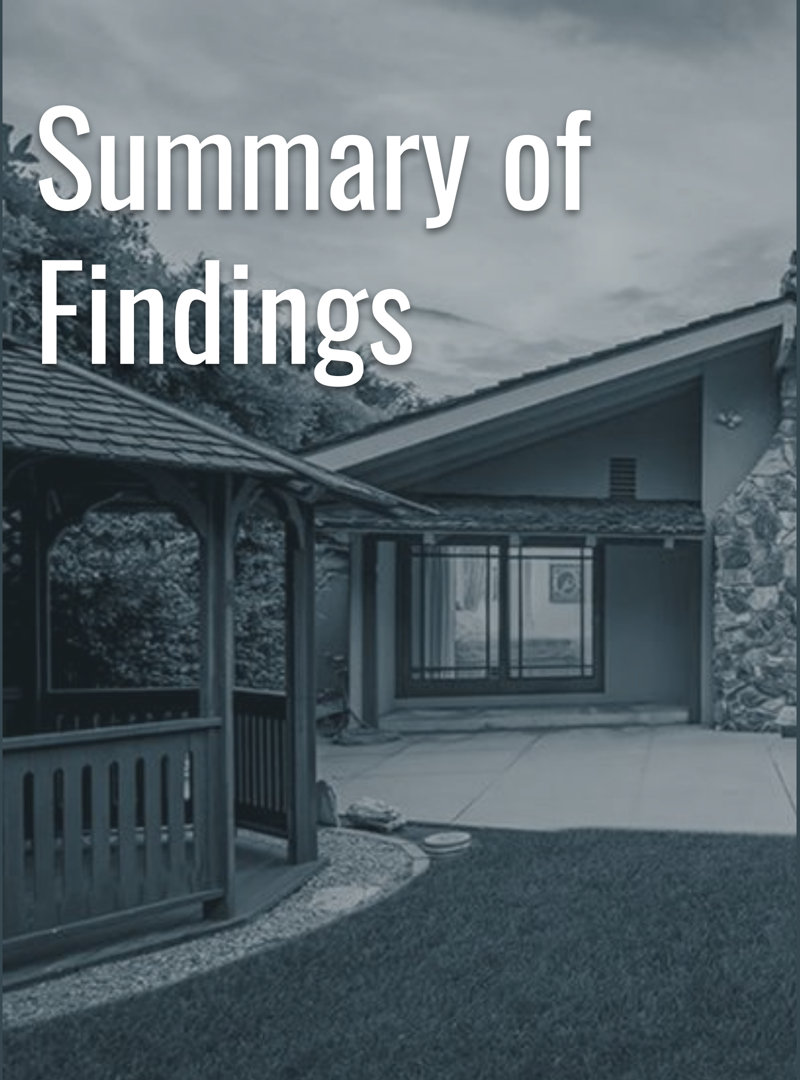
Model Performance



19 FEATURES WITH THE HIGHEST COEFFICIENTS



Summary of Findings



- The ElasticNet model was the best performing in terms of both R^2 and MSE.
- Square feet area, condition, age, and the location of the house are the most important determinant factors of sale price
- House buyers should invest in Northridge Heights, Stone Brook, and Northridge
- People looking to sell should do it sooner rather than later
- To increase the value of a home:
    - Repaint/remodel the interior and exterior finish
    - Renovate the kitchen
    - Add a fireplace (if not already present)
    - Renovate the garage if it is in bad condition
    - Renovate the house if it had been severely damaged

Data Dictionary

Feature	Type	Description	Analysis
lot_frontage	Continuous	Lot size in square feet	330 missing values - fill using imputer
alley	Nominal	Type of alley access to property	NAN represents no alley access - replace NAN with 0.
mas_vnr_type	Nominal	Masonry veneer type	NAN represents missing values - fill using imputer
mas_vnr_area	Continuous	Masonry veneer area in square feet	NAN represents missing values - fill with most frequent (which is 0)
bsmt_qual	Ordinal	Evaluates the height of the basement	NAN represents no basement - replace NAN with 0
bsmt_cond	Ordinal	Evaluates the general condition of the basement	NAN represents no basement - replace NAN with 0
bsmtfin_type_1	Ordinal	Rating of basement finished area	NAN represents no basement - replace NAN with 0
bsmtfin_sf_1	Continuous	Type 1 finished square feet	1 missing value - replace with 0 (i.e. assume no basement)
bsmtfin_type_2	Ordinal	Rating of basement finished area (if multiple types)	NAN represents no basement
bsmt_unf_sf	Continuous	Unfinished square feet of basement area	1 missing value - replace with 0 (i.e. assume no basement)
total_bsmt_sf	Continuous	Total square feet of basement area	1 missing value - replace with 0 (i.e. assume no basement)
bsmt_full_bath	Discrete	Basement full bathrooms	2 missing values - replace with 0 (i.e. assume no basement)
bsmt_half_bath	Discrete	Basement half bathrooms	2 missing values - replace with 0 (i.e. assume no basement)
fireplace_qu	Ordinal	Fireplace quality	NAN represents no fireplace - replace with 0
garage_type	Nominal	Garage location	NAN represents no garage - replace with 0
garage_yr_blt	Discrete	Year garage was built	NAN represents no garage - keep as is, as we will create a new column to capture garage age
garage_finish	Ordinal	Interior finish of the garage	NAN represents no garage - replace with 0
garage_cars	Discrete	Size of garage in car capacity	1 missing value - replace with 0
garage_area	Continuous	Size of garage in square feet	1 missing value - replace with 0
garage_qual	Ordinal	Garage quality	NAN represents no garage - replace with 0
garage_cond	Ordinal	Garage condition	NAN represents no garage - replace with 0
pool_qc	Ordinal	Pool quality	NAN represents no pool - replace with 0
fence	Ordinal	Fence quality	NAN represents no fence - replace with 0
misc_feature	Nominal	Miscellaneous feature not covered in other categories	NAN represents none - replace with 0