Balancing Interpretability and Accuracy in Predictive Modeling: A Case Study on Used Car Price Estimation

The aim of this project is to predict the the price of a used vehicle on craigslist (data). The project includes the following parts:

1. data visualization
2. data preprocessing and imputation using pipelines
3. grid search on Linear Regression, Elastic Net, and LinearSVR to build a baseline linear model
4. feature engineering (e.g. imputation, encoding) and grid search on Random Forst and Gradient Boosting models and hyperparameters to find the best model
5. feature selection using permutation importance and auto feature selection (Random Forst)
6. build an explainable model that is nearly as good as the best model

The final model is a Gradient Boosting Regressor model. The test accuracy is 79%.

Load packges and data

Due to limited computation resources, we only sampled 10% data (~50k records) from the orginal data.

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import missingno as msno
import math

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import ElasticNet
from sklearn.linear_model import Lasso
from sklearn.svm import LinearSVR
from sklearn.tree import DecisionTreeRegressor

from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV

from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OrdinalEncoder
from category_encoders import TargetEncoder
from sklearn.compose import TransformedTargetRegressor
from sklearn.compose import make_column_transformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import PolynomialFeatures
from sklearn.feature_selection import VarianceThreshold

from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.inspection import permutation_importance
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import RandomForestRegressor
#import shap
from sklearn.feature_selection import SelectFromModel
#from sklearn.pipeline import make_pipeline
```

In [ ]:

```
# sampling
vehicles = pd.read_csv("./data/vehicles.csv")
n = math.ceil(vehicles.shape[0] * .03)
vehicles = vehicles.sample(n, replace=False).reset_index(drop=True)
vehicles.to_csv("./data/sample_vehicles.csv", index=False)
print(n)
```

```
12807
```

In [ ]:

```
# load sample data
vehicles = pd.read_csv("./data/sample_vehicles.csv")
```

Exploratory data analysis and data cleaning

Missing data

We found that the missing correlations of a feature pair are so low that we consider it missing at random. This justifies a complete case analysis in further process. In the follow plots:

- Left: White indicates missing
- Right: Perfectly incorrelated is represented as 0

Data cleaning

We conducted the following steps to clean the data and identify features:

1. **Columns**: Keep only features that we think are useful for price prediction for further inspection. Almost all data points have unique `id`, `url`, `image_url`, `VIN`, `model`, and `description`. If these features are included, a model may associate the value to the price and will not be able to generalize well. `description` also leaks target information because it contains the price in the text. We belive that the variation in `state` is more informative than in `region`. We keep the following 13 features:
    - `year`, `manufacturer`, `condition`, `cylinders`, `fuel`, `odometer`, `transmission`, `drive`, `size`, `type`, `paint_color`, `state`, `title_status`.

    And dropped the following 12 features for the above reasons and also because too much missing values (e.g. `county` has all values missing):

    - `id`, `url`, `region_url`, `VIN`, `image_url`, `description`, `county`, `lat`, `long`, `posting_date`, `region`, `model`.
2. **Rows**: Remove rows with outliers and unreasonable values. It is unreasonable for selling a (used) car for less than 1,000 or greater than 100,000; those with odometer higher than 1,000,000 are also treated as outliers. They are not a good representation of the data.

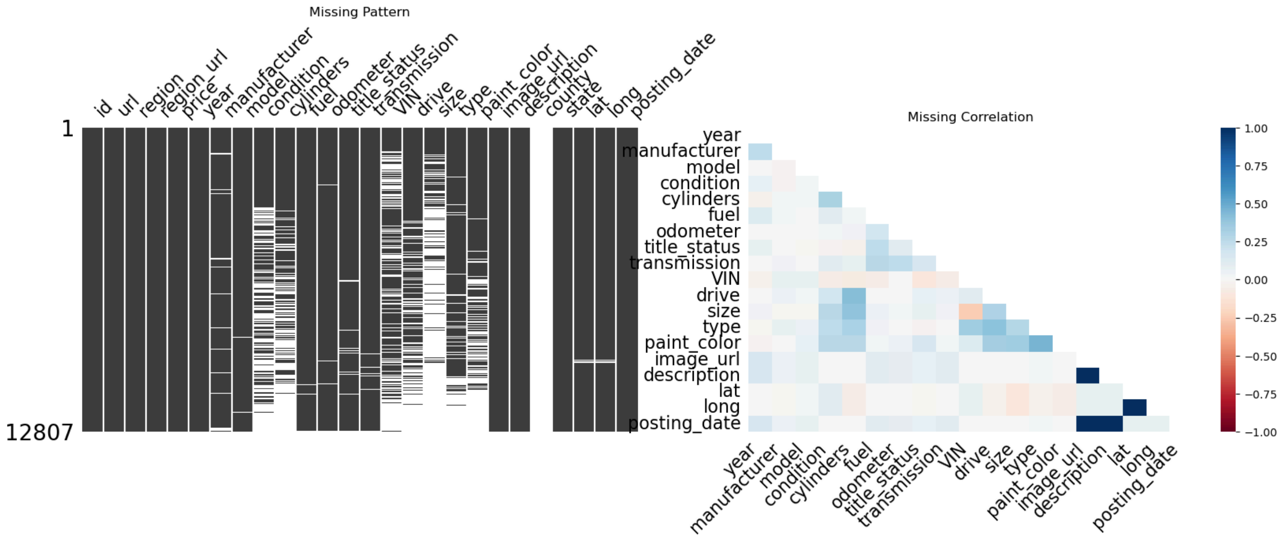
3. Convert `cylinders` (string) to numeric.

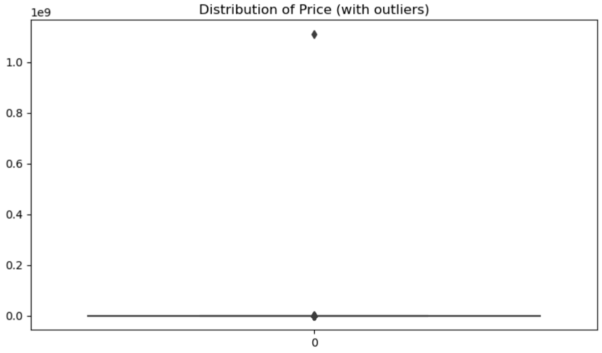
In [ ]:

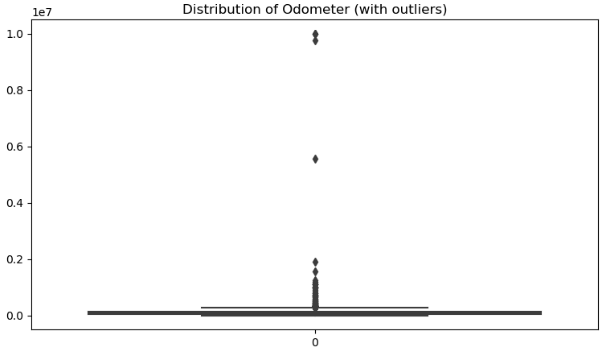
```
# check missing
## percent missing in each feature
missing_per=(vehicles.isnull().sum()[vehicles.isnull().sum()>0]/vehicles.shape[0]).sort_values(axis='index',ascendi
print("missing percentage:\n{}".format(missing_per))

fig,ax = plt.subplots(1,2, figsize=(20,5))
msno.matrix(vehicles, sort="descending", sparkline=False, ax=ax[0]) # White = missing
msno.heatmap(vehicles, figsize=(10,5), ax=ax[1]) # Ideally want 0 missing correlation
ax[0].set_title('Missing Pattern');
ax[1].set_title('Missing Correlation');

# check outlier
fig,ax=plt.subplots(1,2,figsize=(20,5))
sns.boxplot(vehicles.price,ax=ax[0]);
sns.boxplot(vehicles.odometer,ax=ax[1]);
ax[0].set_title("Distribution of Price (with outliers)");
ax[1].set_title("Distribution of Odometer (with outliers)");
```

```
missing percentage:
county          1.000000
size            0.715937
cylinders       0.417428
condition       0.407902
VIN             0.377372
drive           0.307800
paint_color     0.307098
type            0.224252
manufacturer    0.041540
title_status    0.018271
long            0.015538
lat             0.015538
model           0.012337
odometer        0.010229
transmission    0.007262
fuel            0.006949
year            0.002733
description     0.000078
image_url       0.000078
posting_date    0.000078
dtype: float64
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/seaborn/matrix.py:260: FutureWarning: Format s
trings passed to MaskedConstant are ignored, but in future may error or produce different behavior
  annotation = ("{:" + self.fmt + "}").format(val)
```







In [ ]:
```
# data cleaning
pd.options.mode.chained_assignment = None  # mute warnings (default='warn')

# drop missing rows and outliers
vehicles = vehicles[~((vehicles.price<1e3)|(vehicles.price>1e5)|(vehicles.odometer>1e6))]
```

Identify features through visualization

We consider the features as:

- Continuous: `year`, `odometer`
- Categorical: `title_status`, `fuel`, `transmission`, `drive`, `type`, `manufacturer`, `paint_color`, `state`
- Ordinal: `condition`, `size`, `cylinders`

We visualize univariate relationships between each feature and the target `price`. Some interesting features are:

- `cylinder`:
  - Considering only common cylinders (4, 6, and 8), a car with 8 cylinders portrays higher price
  - 12 cylinder has distinctively high price (we found later that because there is only a couple of data points)
- `condition`:
  - Natural ordering intuitively reflect the price well. (New is the highest and salvage is the lowest.)
- `size`:
  - Full-size cars have the highest median price
- `year`:
  - A newer model has a higher median price.
- `odometer`:
  - A car with high odometer has a lower median price.

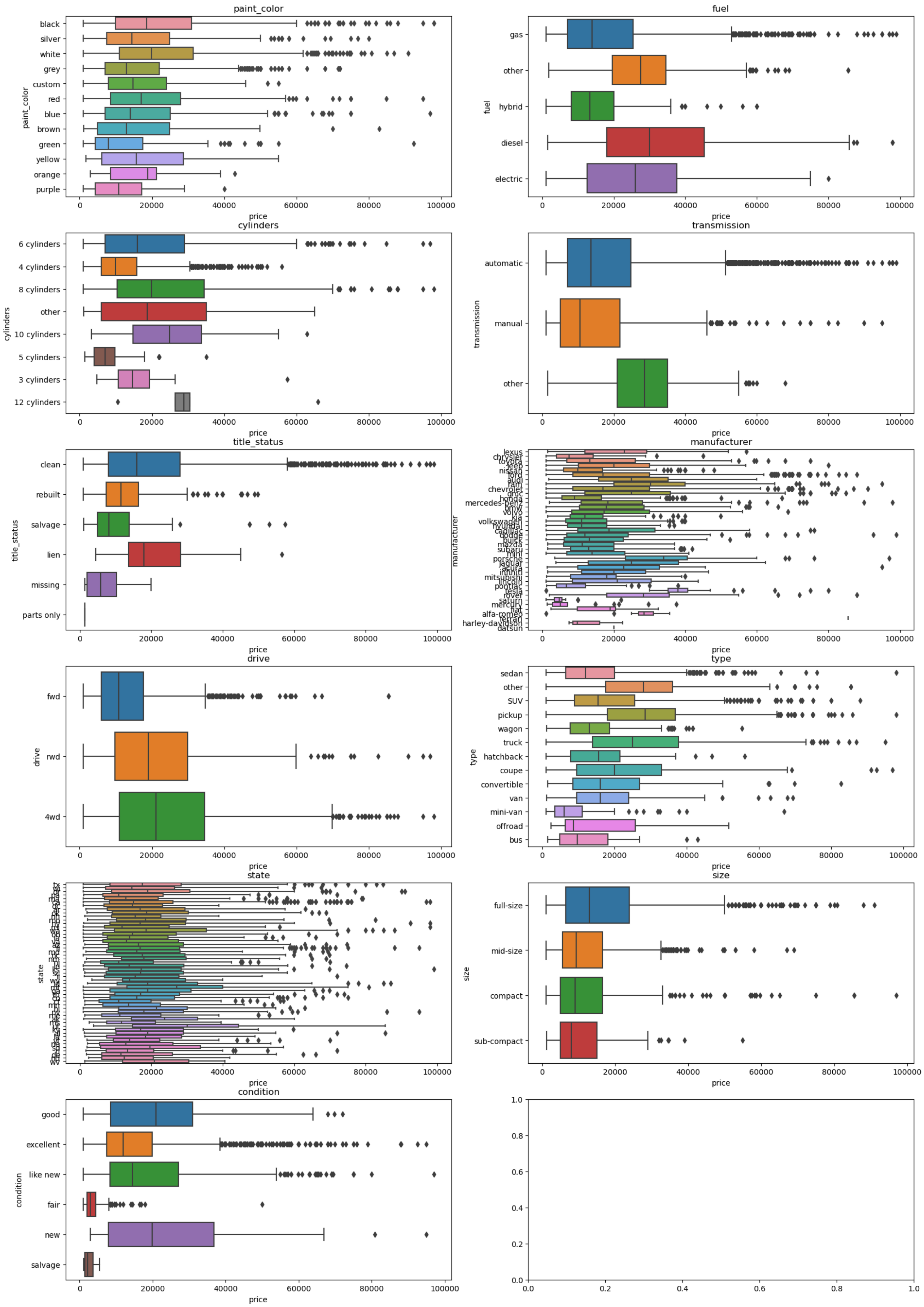
No patterns are observed in other features due to too few data points (e.g. manufacturer Tesla has significantly high price). However, we decide to keep most of them for further inspection because they are intuitive predictors for the selling price.

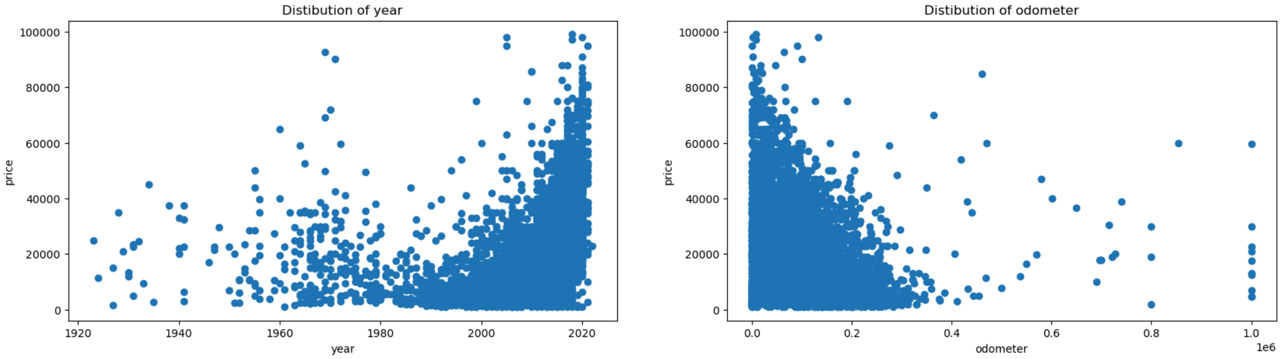
In [ ]:
```
categorical = ['fuel','transmission', 'drive','type',
               'manufacturer','paint_color', 'state', 'title_status']
```

```
ordinal = ['condition','size','cylinders']
continuous = ['year','odometer']
```

In [ ]:

```
# boxplot for categorical variables using seaborn
feature = list(set(categorical+ordinal))
fig,ax = plt.subplots(6,2,figsize=(20,30))
for i in range(0,6):
    if i != 5:
        for j in range(0,2):
            sns.boxplot(y=vehicles[feature[i*2+j]], x=vehicles.price, ax=ax[i,j],orient="h").set_title(feature[i*2+
    else:
        sns.boxplot(y=vehicles[feature[i*2]], x=vehicles.price, ax=ax[i,0],orient="h").set_title(feature[i*2])

# continuous variables
fig,ax=plt.subplots(1,2,figsize=(20,5))
for i in range(0,2):
    ax[i].scatter(vehicles[continuous[i]],vehicles.price)
    ax[i].set_xlabel(continuous[i])
    ax[i].set_ylabel('price')
    ax[i].set_title('Distibution of {}'.format(continuous[i]))
```


paint_color
black
silver
white
grey
custom
red
blue
brown
green
yellow
orange
purple
fuel
gas
other
hybrid
diesel
electric
cylinders
6 cylinders
4 cylinders
8 cylinders
other
10 cylinders
5 cylinders
3 cylinders
12 cylinders
transmission
automatic
manual
other
title_status
clean
rebuilt
salvage
lien
missing
parts only
manufacturer
lexus
chrysler
toyota
jeep
nissan
ford
audi
ram
chevrolet
gmc
honda
mercedes-benz
bmw
volvo
kia
volkswagen
hyundai
cadillac
mini
dodge
buick
mazda
subaru
mini
porsche
jaguar
acura
infiniti
mitsubishi
lincoln
pontiac
tesla
rover
saturn
mercury
fiat
alfa-romeo
ferrari
harley-davidson
datsun
drive
fwd
rwd
4wd
type
sedan
other
SUV
pickup
wagon
truck
hatchback
coupe
convertible
van
mini-van
offroad
bus
state
size
full-size
mid-size
compact
sub-compact
condition
good
excellent
like new
fair
new
salvage
price
0
20000
40000
60000
80000
100000
0.0
0.2
0.4
0.6
0.8
1.0



Preprocessing and Baseline model

Baseline model

We use a Linear regression model as the baseline model because it is simple to implement. For the baseline model, only the following features are being used:

- Continuous: `year`, `odometer`
- Categorical: `title_status`, `fuel`, `transmission`, `drive`, `state`, `type`
- Ordinal: `condition`, `size`, `cylinders`

`manufacturer`, `paint_color` are removed from the training data for the baseline model because they have many unique values and one-hot encoding those features will make the model have much more variable and less explainable. Using only six categorical variables allows for a reduced number of coefficients, which is good for a baseline model that serves as an initial approach to the data.

Preprocessing

We preprocessed the data in following steps:

- drop outliers in `price` (less than 1,000 or greater than 100,000) and `odometer` (higher than 1,000,000)
- deal with missing values (imputation with mean or mode)
- standardize `odometer` (we did not standardize `year` because it does not make sense)
- target encode `state`
- one-hot encode other categprical variables (`title_status`, `fuel`, `transmission`, `drive`, `type`)

In [ ]:
```
from joblib import cpu_count
print(cpu_count())    # use this to check number of cores for multiprocessing in Grid Search and Random Forest

n_jobs = min(10, cpu_count())    # or None if don't want to use multiprocessing
```

```
11
```

In [ ]:
```
def pipe_preprocess(impute=True, continuous=[], target=[], categorical=[], ordinal=[], ord_cat=[],other_imp=[]):
    if impute:
        cat_preprocessing = Pipeline([('cat_imp',SimpleImputer(missing_values=np.nan,
                                                               strategy='most_frequent')),
                                      ('cat_pre',OneHotEncoder(handle_unknown ='ignore', drop='first'))])  # this w
        # check this post https://github.com/scikit-learn/scikit-learn/issues/18072
        # and https://github.com/scikit-learn/scikit-learn/pull/19041
        ord_preprocessing=Pipeline([('ord_imp',SimpleImputer(missing_values=np.nan,
                                                               strategy='most_frequent')),
                                     ('ord_pre',OrdinalEncoder(categories=ord_cat, handle_unknown = 'use_encoded_va
        cont_preprocessing=Pipeline([('cont_imp',SimpleImputer(missing_values=np.nan,
                                                               strategy='mean')),
                                     ('cont_pre',StandardScaler())])
        imp_preprocessing=Pipeline([('imp',SimpleImputer(missing_values=np.nan,
                                                         strategy='most_frequent'))])
    else:
        cat_preprocessing = Pipeline([('cat_pre',OneHotEncoder(handle_unknown ='ignore'))])
        ord_preprocessing = Pipeline([('ord_pre',OrdinalEncoder(categories=ord_cat, handle_unknown = 'use_encoded_v
        cont_preprocessing = Pipeline([('cont_pre',StandardScaler())])
        imp_preprocessing = None

    preprocess = make_column_transformer((cat_preprocessing,categorical),
                                         (ord_preprocessing,ordinal),
                                         (cont_preprocessing,continuous),
                                         (imp_preprocessing,other_imp),
                                         (TargetEncoder(),target),
                                         remainder='passthrough')
    return preprocess

def grid_result(param_grid, X_trainval, y_trainval, preprocess, cv=3, n_jobs=10, verbose=2):
    """
    Process and apply a grid search on X_trainval and y_trainval.
    Args:
    param_grid: dictionary of parameter settings to be grid searched on
    X_trainval, y_trainval: training and validation data
    preprocess: preprocess pipeline
    cv; # of cross-validation fold
    Return:
    grid object that is fitted on X_trainval and y_trainval
    """
    pipeline = Pipeline([('preprocess',preprocess),
                         ('regressor',LinearRegression())])

    grid = GridSearchCV(pipeline, param_grid, return_train_score=True, cv=cv, n_jobs=n_jobs, verbose=verbose)
    grid.fit(X_trainval, y_trainval)
    return grid
```

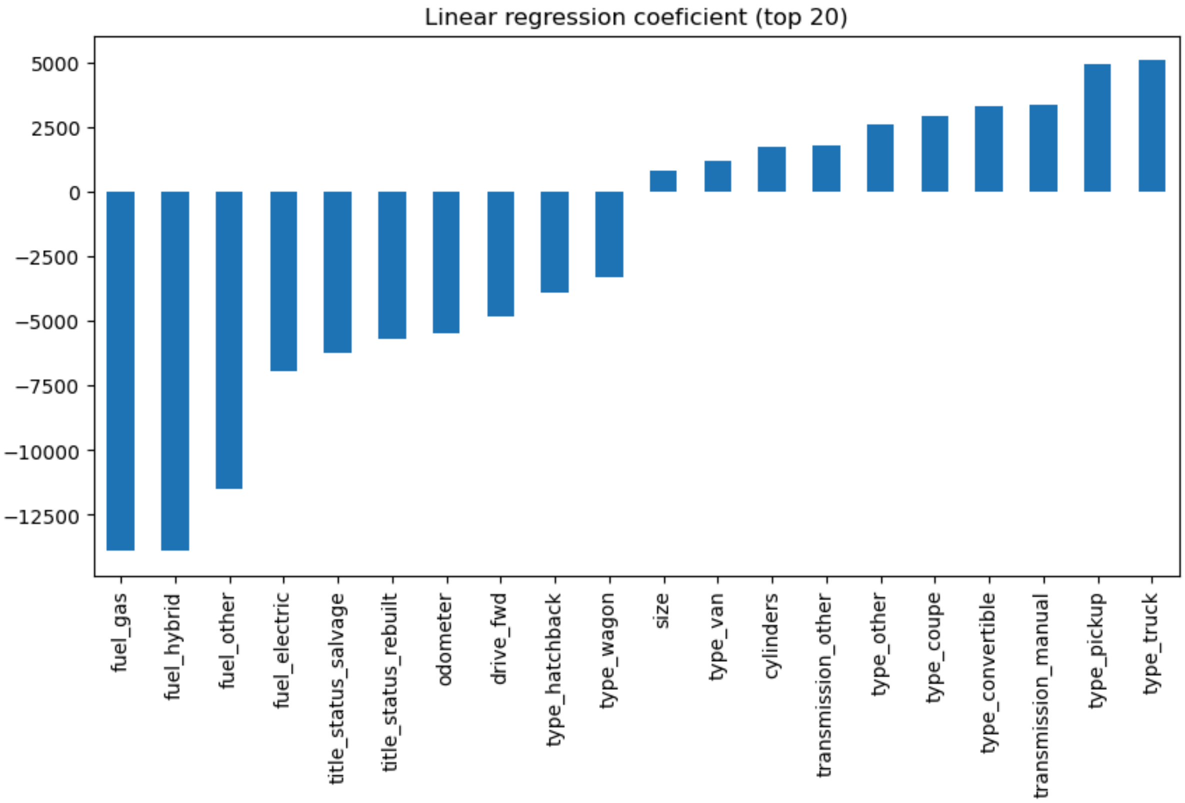
In [ ]:

```
continuous = ['odometer', 'year']
ordinal = ['condition','size', 'cylinders']
categorical = ['fuel','transmission','drive','title_status','type']
target = ['state']

# reorder ordinal variables for easier interpretation of the model
ord_cat = [['salvage', 'fair', 'like new', 'new', 'good', 'excellent'],
           ['sub-compact','compact', 'mid-size', 'full-size'],
           ['3 cylinders', '4 cylinders', '5 cylinders', '6 cylinders', '8 cylinders', '10 cylinders', '12 cylinder

y = vehicles.price
X = vehicles[categorical + ordinal + continuous + target]

## Set global training and test sets
global_X_trainval, global_X_test, global_y_trainval, global_y_test = train_test_split(X, y, random_state=1)

log_ols = TransformedTargetRegressor(LinearRegression(), func=np.log, inverse_func=np.exp)
param_grid=[{'regressor':[LinearRegression()]}]
preprocess = pipe_preprocess(impute=True, continuous=['odometer'], target=target, categorical=categorical, ordinal=
grid_cp = grid_result(param_grid, global_X_trainval, global_y_trainval, preprocess, cv=5, n_jobs=n_jobs)

print('The best model:\n {model}\nCV score: {score:.3f}'\
      .format(model=grid_cp.best_params_,score=grid_cp.best_score_))
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[CV] END .......................regressor=LinearRegression(); total time=   0.0s
[CV] END .......................regressor=LinearRegression(); total time=   0.0s
[CV] END .......................regressor=LinearRegression(); total time=   0.0s
[CV] END .......................regressor=LinearRegression(); total time=   0.0s
[CV] END .......................regressor=LinearRegression(); total time=   0.0s
The best model:
 {'regressor': LinearRegression()}
CV score: 0.514
```

In [ ]:

```
col_names = grid_cp.best_estimator_.named_steps['preprocess'].transformers_[0][1].named_steps['cat_pre'].get_featur
col_names = list(col_names) + ordinal+['odometer']+['year']+target
coef = grid_cp.best_estimator_.named_steps['regressor'].coef_
# coef = grid_cp.best_estimator_.named_steps['regressor'].regressor_.coef_

coef_plot = pd.DataFrame({"Coef":coef}, index = col_names)
coef_plot.sort_values(['Coef']).iloc[[*range(10), *range(-10, 0)]].plot.bar(figsize=(10,5), title='Linear regressio
```



In general, the results make sense. The most important factors seems to be the type and status of a car, which is not surprising given this is characteristic indicates the utility and quality of a car. Truck and pickup car usually have a larger size and tend to be more expensive. Cars with only parts available reduce the value of them. Although it is weird that cars without a title status are more expensive than the baseline (clean title status). It might be because not many sample points are included for this category (19 records, less than 0.1%). The rest of the coefficients also make sense, whith the exception of electric cars being cheaper than the diesel car. We suspect this is because many of these vehicles are not actually cars, but golf carts and similar vehicles.

Feature Engineering

From the previous task, `cylinders` , `condition` , and `size` are three columns with the most missing values. At this step, we are curious whether we should

1. impute missing categorical values with the most frequent value.
2. create a new category indicating the missing data.

It turns out that the second methods gives a better validation score (0.571) than the one using the first method (0.566). It makes sense considering the context because having a NaN might give an extra piece of information (e.g. if someone does not include the condition of the car, it may be that she is hiding it and therefore we would expect a lower price).

Based on the baseline model, we made a few modification to the features and pre-processing methods:

- transform output `price` into log form, as price seems to have a log-normal distribution
- included `manufacturer` , `paint_color` and one-hot encode them
- dealing with missing values (create an NaN category)
- rebased year to 1900 to make the value smaller
- include the interaction between ordinal variables (interaction among `cylinders` , `condition` , and `size`)

After the imputation, we also try

1. deriving `paint_color_group` from `paint_color` , which describes car colors into most common, somewhat common, and uncommon color.
2. deriving `use_type` from `type` , which categorized car types into family used and non-family used.
3. deriving a feature that describes the length of description entry

However, all of the approaches (code not shown) do no better than the current feature engineering.

In [ ]:
```
def pipe_preprocess(impute=True, continuous=[], target=[], categorical=[], ordinal=[], ord_cat=[], other_imp=[]):
    if impute:
        cat_preprocessing = Pipeline([('cat_imp',SimpleImputer(missing_values=np.nan, strategy='most_frequent')),
                                      ('cat_pre',OneHotEncoder(handle_unknown ='ignore', drop='first'))])  # this w
        # check this post https://github.com/scikit-learn/scikit-learn/issues/18072
        # and https://github.com/scikit-learn/scikit-learn/pull/19041
        ord_preprocessing=Pipeline([('ord_imp',SimpleImputer(missing_values=np.nan, strategy='most_frequent')),
                                    ('ord_pre',OrdinalEncoder(categories=ord_cat, handle_unknown = 'use_encoded_val
                                    ('ord_poly',PolynomialFeatures(interaction_only=True, include_bias=False))])
        cont_preprocessing=Pipeline([('cont_imp',SimpleImputer(missing_values=np.nan, strategy='mean')),
                                     ('cont_pre',StandardScaler())])
        imp_preprocessing=Pipeline([('imp', SimpleImputer(missing_values=np.nan, strategy='most_frequent'))])
    else:
        cat_preprocessing = Pipeline([('cat_pre',OneHotEncoder(handle_unknown ='ignore'))])
        ord_preprocessing = Pipeline([('ord_pre',OrdinalEncoder(categories=ord_cat, handle_unknown = 'use_encoded_v
        cont_preprocessing = Pipeline([('cont_pre',StandardScaler())])
        imp_preprocessing = None

    preprocess = make_column_transformer((cat_preprocessing,categorical),
                                         (ord_preprocessing,ordinal),
                                         (cont_preprocessing,continuous),
                                         (imp_preprocessing,other_imp),
                                         (TargetEncoder(),target),
                                         remainder='passthrough')

    return preprocess
```

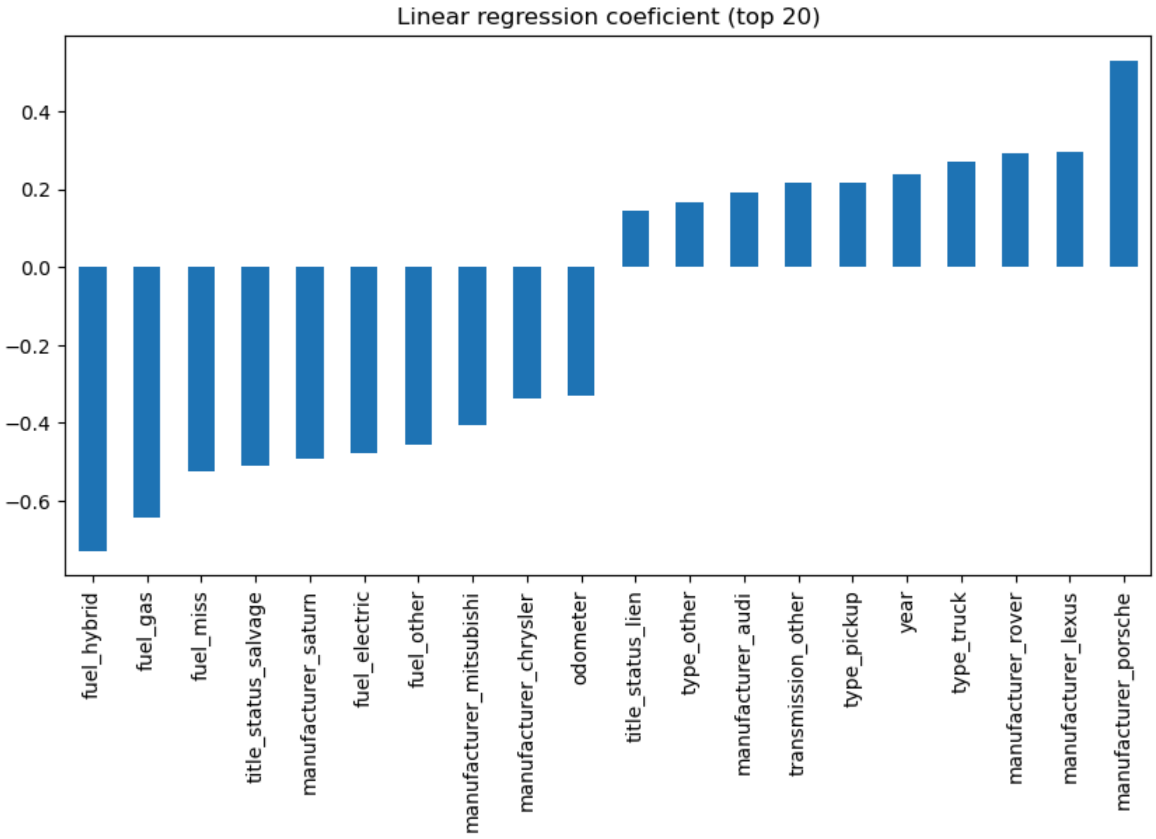
In [ ]:
```
continuous = ['odometer', 'year']
ordinal = ['condition','size', 'cylinders']
categorical = ['fuel','transmission','drive','title_status','type',
               'manufacturer','paint_color']
target = ['state']

y = vehicles.price
X = vehicles[categorical + ordinal + continuous + target]
X['year'] = X['year'] - 1900

X[ordinal+categorical] = X[ordinal+categorical].fillna('miss')
# # reorder ordinal variables for easier interpretation of the model
# X['condition'] = X['condition'].astype('category').cat.reorder_categories(['miss','salvage', 'fair', 'like new',
# X['size'] = X['size'].astype('category').cat.reorder_categories(['miss','sub-compact','compact', 'mid-size', 'ful
# X['cylinders'] = X['cylinders'].astype('category').cat.reorder_categories(['miss','3 cylinders', '4 cylinders', '

## Set global training and test sets
global_X_trainval, global_X_test, global_y_trainval, global_y_test = train_test_split(X, y, random_state=1)
```

In [ ]:
```
log_ols = TransformedTargetRegressor(LinearRegression(),func=np.log,inverse_func=np.exp)
# log_en = TransformedTargetRegressor(ElasticNet(tol = 1),func=np.log,inverse_func=np.exp)
# log_svr = TransformedTargetRegressor(LinearSVR(tol = 1),func=np.log,inverse_func=np.exp)

# param_grid=[{'regressor':[log_ols]},
#             {'regressor':[log_en],
#              'regressor__regressor__alpha': np.logspace(-5,1,7),
#              'regressor__regressor__l1_ratio': [0,0.01, .1, .5, .98, 1]},
#             {'regressor':[log_svr],
#              'regressor__regressor__C':np.logspace(-5,1,7)}]
param_grid=[{'regressor':[log_ols]}]

preprocess = pipe_preprocess(impute=True, continuous=continuous, target=target,
                             categorical=categorical, ordinal=ordinal, ord_cat=ord_cat)
grid_cp = grid_result(param_grid, global_X_trainval, global_y_trainval, preprocess, cv=5, n_jobs=n_jobs)

print('The best model:\n {model}\nCV score: {score:.3f}'\
      .format(model=grid_cp.best_params_,score=grid_cp.best_score_))
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=LinearRegression()); total time=   0.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=LinearRegression()); total time=   0.0s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=LinearRegression()); total time=   0.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=LinearRegression()); total time=   0.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=LinearRegression()); total time=   0.0s
The best model:
 {'regressor': TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=LinearRegression())}
CV score: 0.569
```

```
In [ ]: col_names = list(grid_cp.best_estimator_.named_steps['preprocess'].transformers_[0][1].named_steps['cat_pre'].get_f
                    list(grid_cp.best_estimator_.named_steps['preprocess'].transformers_[1][1].named_steps['ord_poly'].get_
                    ['odometer']+['year']+target
        # coef = grid_cp.best_estimator_.named_steps['regressor'].coef_
        coef = grid_cp.best_estimator_.named_steps['regressor'].regressor_.coef_

        coef_plot = pd.DataFrame({"Coef":coef}, index = col_names)
        coef_plot.sort_values(['Coef']).iloc[[*range(10), *range(-10, 0)]].plot.bar(figsize=(10,5), title='Linear regressio
```



Grid Search with Random Forest and Gradient Boosting

In addition to Linear Regression, we train models with

- GradientBoosting and
- Random Forest

and conducted Grid Search on the hyperparameters (code shown below).

The R^2 of validation set improves from 0.57 to 0.79 when we use gradient boosting with the following hyperparameters

- loss = `huber`
- learning_rate = `0.1`
- max_depth = `9`

The test score is 0.791.

We further tried to introduce new features: "**length of description**" and "**spaces in description**" into the model, but they did not significantly improves the model performance. (Code not shown).

```
In [ ]: import warnings
        warnings.filterwarnings("ignore")

        log_gbr=TransformedTargetRegressor(GradientBoostingRegressor(warm_start = True),func=np.log,inverse_func=np.exp)
        log_rfr=TransformedTargetRegressor(RandomForestRegressor(warm_start = True),func=np.log,inverse_func=np.exp)

        param_grid=[{'regressor':[log_gbr],
                     'regressor__regressor__loss': ['squared_error', 'absolute_error', 'huber'],
                     'regressor__regressor__learning_rate': [0.001, 0.01, 0.1, 0.5],
                     'regressor__regressor__max_depth': [7,9,11]},
                    {'regressor':[log_rfr],
                     'regressor__regressor__max_depth': [7,9,11],
                     'regressor__regressor__n_estimators': [100, 150, 200]}]

        best_grid = grid_result(param_grid, global_X_trainval, global_y_trainval, preprocess, cv=5, n_jobs=n_jobs)
```

```
Fitting 5 folds for each of 45 candidates, totalling 225 fits
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.6s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.3s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.2s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.2s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.2s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.7s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.5s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.6s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  14.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.4s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.5s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  14.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  14.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   4.5s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  14.8s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.001, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  14.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.1s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   3.2s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   6.2s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   5.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.8s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.5s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   7.7s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   7.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.1s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   3.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   3.1s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   3.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   3.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   3.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  15.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  15.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  14.9s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  15.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.01, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=  15.5s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   4.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   4.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   4.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   4.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   4.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.6s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   3.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   3.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   4.0s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   3.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   4.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   5.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   2.9s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   5.5s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   5.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   5.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   5.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   3.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   5.1s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   5.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   5.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   5.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   5.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.1s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=7; total time=   2.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   8.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   8.7s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   8.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   8.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   2.9s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.1, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   8.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   2.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   2.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   2.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=9; total time=   3.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.2s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=7; total time=   2.1s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   3.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   3.7s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   3.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   3.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=squared_error, regressor__regressor__max_depth=11; total time=   3.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   2.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   2.7s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   2.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   2.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=9; total time=   2.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   3.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   3.8s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   3.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   3.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=absolute_error, regressor__regressor__max_depth=11; total time=   3.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   2.7s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   2.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   2.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   2.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=7; total time=   2.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   4.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   4.4s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   4.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   4.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=9; total time=   4.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=100; total time=   1.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=100; total time=   1.9s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=100; total time=   1.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=100; total time=   1.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=100; total time=   1.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   6.4s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   6.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   6.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   6.5s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), regressor__regressor__learning_rat
e=0.5, regressor__regressor__loss=huber, regressor__regressor__max_depth=11; total time=   6.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=150; total time=   2.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=150; total time=   2.8s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=150; total time=   2.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=150; total time=   2.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=150; total time=   2.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=200; total time=   3.7s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=200; total time=   3.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=200; total time=   3.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=200; total time=   3.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=7, reg
ressor__regressor__n_estimators=200; total time=   3.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=100; total time=   3.0s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=100; total time=   3.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=100; total time=   2.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=100; total time=   3.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=100; total time=   3.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=150; total time=   4.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=150; total time=   4.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=150; total time=   4.3s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=150; total time=   4.4s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=150; total time=   4.5s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=200; total time=   5.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=200; total time=   5.9s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=200; total time=   6.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=200; total time=   6.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=9, reg
ressor__regressor__n_estimators=200; total time=   6.2s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=100; total time=   4.4s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=100; total time=   4.5s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=100; total time=   4.5s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
```

```
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=100; total time=   4.6s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=100; total time=   4.5s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=150; total time=   6.8s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=150; total time=   6.7s
```

```
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=150; total time=   6.7s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=150; total time=   6.5s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=150; total time=   6.6s
/Users/yc/Software/miniconda3/envs/py312/lib/python3.12/site-packages/sklearn/preprocessing/_encoders.py:241: UserWa
rning: Found unknown categories in columns [5] during transform. These unknown categories will be encoded as all zer
os
  warnings.warn(
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=200; total time=   8.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=200; total time=   8.0s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=200; total time=   7.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=200; total time=   7.9s
[CV] END regressor=TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=RandomForestRegressor(warm_start=True)), regressor__regressor__max_depth=11, re
gressor__regressor__n_estimators=200; total time=   7.8s
```

In [ ]:

```
print('The best model:\n{model}.\n\nCV score: {score:.3f}'\
      .format(model=best_grid.best_params_,score=best_grid.best_score_))
print("\nTest Score: {test_score:.3f}".format(test_score = best_grid.best_estimator_\
                                             .score(global_X_test, global_y_test)))
```

```
The best model:
{'regressor': TransformedTargetRegressor(func=<ufunc 'log'>, inverse_func=<ufunc 'exp'>,
                           regressor=GradientBoostingRegressor(warm_start=True)), 'regressor__regressor__learning_ra
te': 0.1, 'regressor__regressor__loss': 'huber', 'regressor__regressor__max_depth': 9}.

CV score: 0.744

Test Score: 0.791
```

Feature selection

Permutation importance

We use permuation importance to identify features that are most influential to our best model (gradient boosting).

- `Year` is the most important feature to the model, followed by `odometer` and `drive`.
- `transmission`, `size`, and `paint_color` have the least degree of importance.

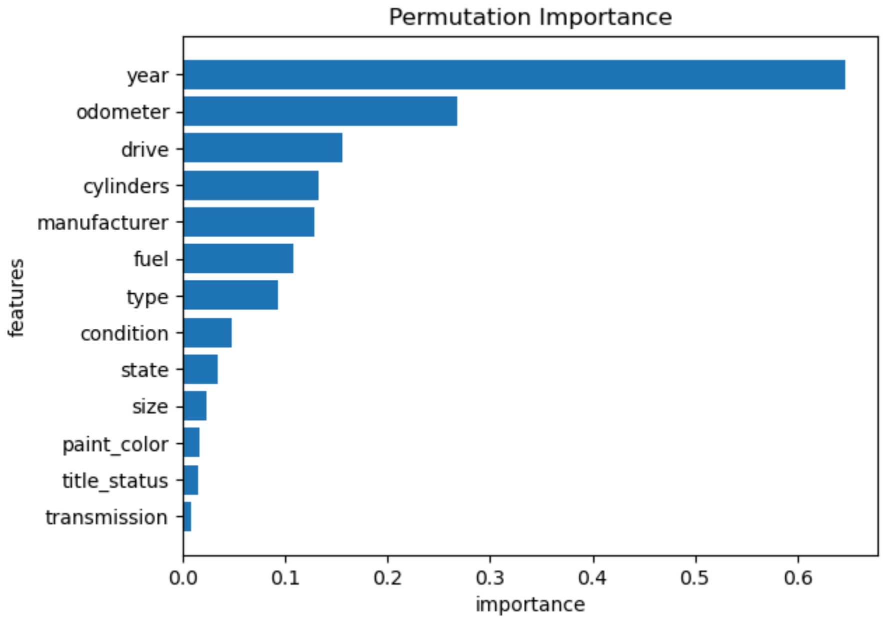
However, with pipelines, permutation importanace could only check categorical features as a whole. For example, it could only check the importance of `cylinders` not `12 cylinders` specifically.

In [ ]:

```
# specify the best model to aviod running grid search again next time
current_best_model = TransformedTargetRegressor(GradientBoostingRegressor(warm_start = True,
                                                                          learning_rate = 0.1,
                                                                          loss = 'huber',
                                                                          max_depth = 9),
                                                func=np.log,inverse_func=np.exp)

preprocess = pipe_preprocess(impute=True, continuous=['odometer'], target=target,
                             categorical=categorical, ordinal=ordinal, ord_cat=ord_cat, other_imp=['year'])
pipeline = Pipeline([('preprocess',preprocess), ('regressor',current_best_model)])
pipeline.fit(global_X_trainval, global_y_trainval)

p_impt = permutation_importance(pipeline, global_X_trainval, global_y_trainval,
                                n_repeats = 5, random_state = 0, n_jobs=n_jobs)
```

In [ ]:

```
p_impt_result = pd.DataFrame({"Feature": global_X_trainval.columns.to_list(),
                              "importances_mean": p_impt['importances_mean'],
                              "importances_std": p_impt['importances_std']}).sort_values(by=['importances_mean'])
plt.barh(p_impt_result['Feature'], p_impt_result['importances_mean'])
plt.title("Permutation Importance");
plt.xlabel("importance");
plt.ylabel("features");
```



Use Random Forest to select features

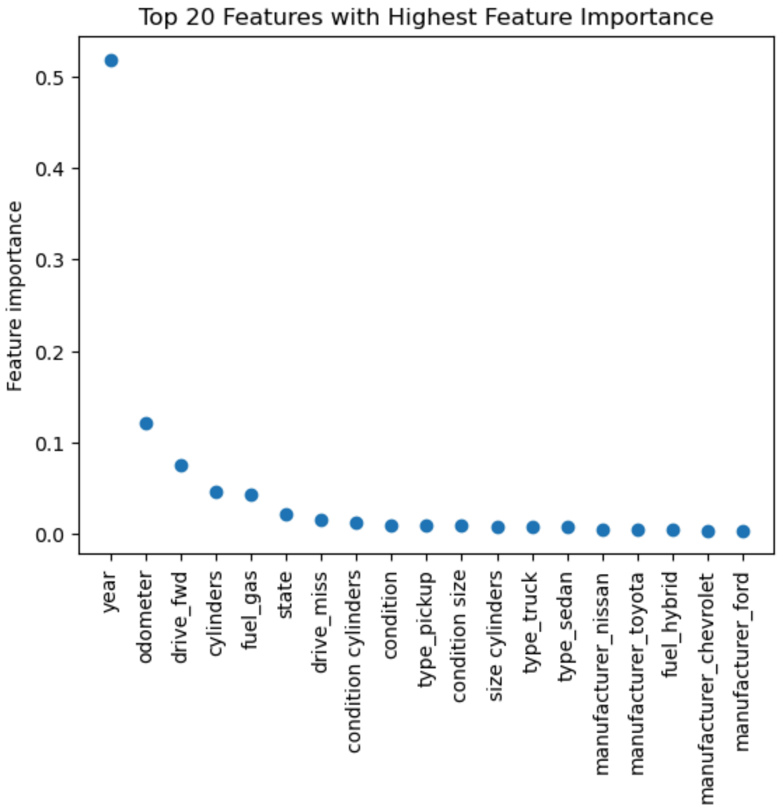
We use random forest to select features for the best model (Gradient Boosting). After the pre-processing, there are 90 features in the best Gradeint Boosting model.

In [ ]:

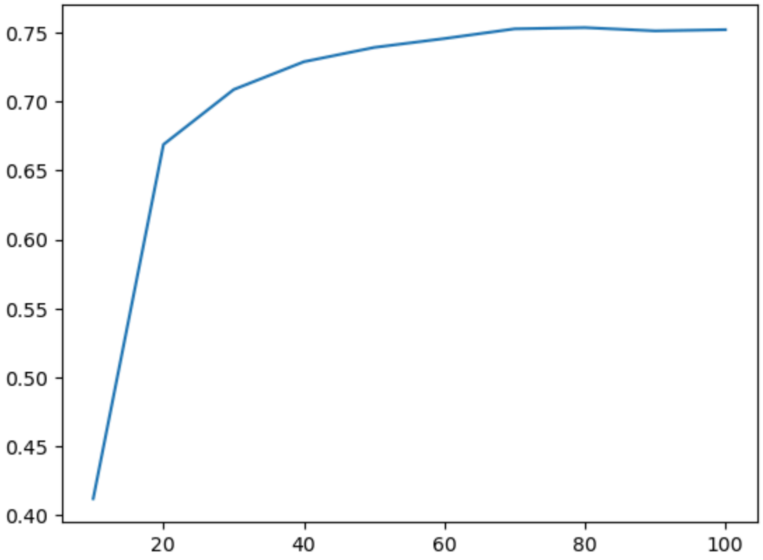
```
## Best model from grid search
# best_model = best_grid.best_estimator_.named_steps['regressor']
best_model = pipeline.named_steps['regressor'].regressor_
```

In [ ]:

```
col_names = list(pipeline.named_steps['preprocess'].transformers_[0][1].named_steps['cat_pre'].get_feature_names_ou
            list(pipeline.named_steps['preprocess'].transformers_[1][1].named_steps['ord_poly'].get_feature_names_o
            ['odometer']+['year']+target
coef = pd.Series(best_model.feature_importances_, index = col_names)
coef = coef.sort_values(ascending = False)[0:19]
plt.scatter(coef.index, coef)
plt.xticks(rotation=90)
plt.gca().set(ylabel='Feature importance', title='Top 20 Features with Highest Feature Importance')
plt.show()
```



In [ ]:
```
from sklearn.feature_selection import RFECV
rfecv = Pipeline([('pre_processing',preprocess),
                 ('rfecv',RFECV(estimator=best_model, step=10, cv=5, n_jobs=n_jobs))])
rfecv.fit(global_X_trainval, global_y_trainval)
scores = rfecv.named_steps['rfecv'].cv_results_['mean_test_score']
plt.plot(np.array(range(1, len(scores) + 1))*10, scores);
```



In [ ]:
```
import warnings
warnings.filterwarnings("ignore")

# Find the number of features to keep at various thresholds
thresholds = [1e-1, 1e-2, 5e-3, 1e-3, 0]
result = []
pipe_list = []
select_rf_list = []

X_trainval_processed = preprocess.transform(global_X_trainval)
for i in thresholds:
    select_rf = SelectFromModel(RandomForestRegressor(random_state = 0), threshold=i)
    pipe_rf = Pipeline([('preprocess',preprocess),
                        ('feature_selection',select_rf),
                        ('regressor', best_model)])
    pipe_list.append(pipe_rf)

    mean_score = np.mean(cross_val_score(pipe_rf, global_X_trainval, global_y_trainval, cv=5, n_jobs=n_jobs))

    select_rf.fit(X_trainval_processed, global_y_trainval)
    select_rf_list.append(select_rf)

    result.append((select_rf.transform(X_trainval_processed).shape[1] , mean_score))
```

In [ ]:

```
result_df = {"threshold": thresholds, "mean_R2": [r for (_,r) in result],
             "num_feature": [n for (n,_) in result]}
pd.DataFrame(result_df)
```

Out[ ]:

	threshold	mean_R2	num_feature
0	0.100	0.423767	2
1	0.010	0.684176	12
2	0.005	0.705957	20
3	0.001	0.747111	58
4	0.000	0.752800	89

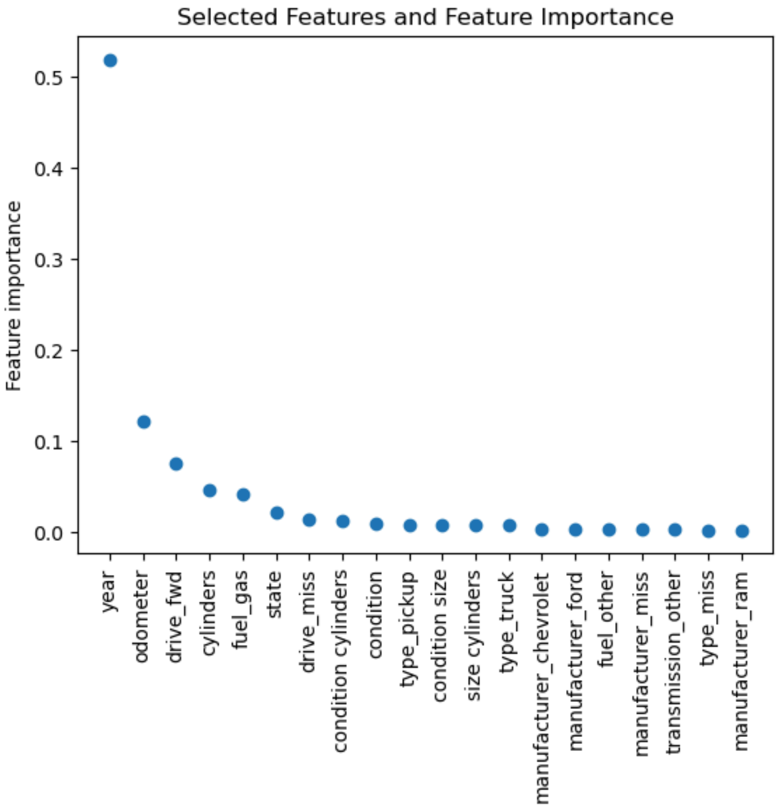
There are initially 92 features in total. Removing subsets of features lowers the model performace. Using SelectFromModel(), we find that there are 21 features that, when included in the model, can perform as acceptably well ($R^2 = 0.70$) as including all features ($R^2 = 0.76$).

In [ ]:

```
#' Get feature names
#' order matters!
col_names = list(pipeline.named_steps['preprocess'].transformers_[0][1].named_steps['cat_pre'].get_feature_names_ou
            list(pipeline.named_steps['preprocess'].transformers_[1][1].named_steps['ord_poly'].get_feature_names_o
            ['odometer']+['year']+target

picked_index = 2
selected_features = []
for pipe in pipe_list:
    mask = pipe.named_steps['feature_selection'].get_support().tolist()
    coef = pipe.named_steps['regressor'].feature_importances_
    selected_features.append(pd.DataFrame({'Feature': np.array(col_names)[mask], 'Coef': coef[mask]}))
```

In [ ]:

```
coef = selected_features[picked_index].set_index('Feature')['Coef']
coef = coef.sort_values(ascending = False)
plt.scatter(coef.index, coef)
plt.xticks(rotation=90)
plt.gca().set(ylabel='Feature importance', title='Selected Features and Feature Importance')
plt.show()
```

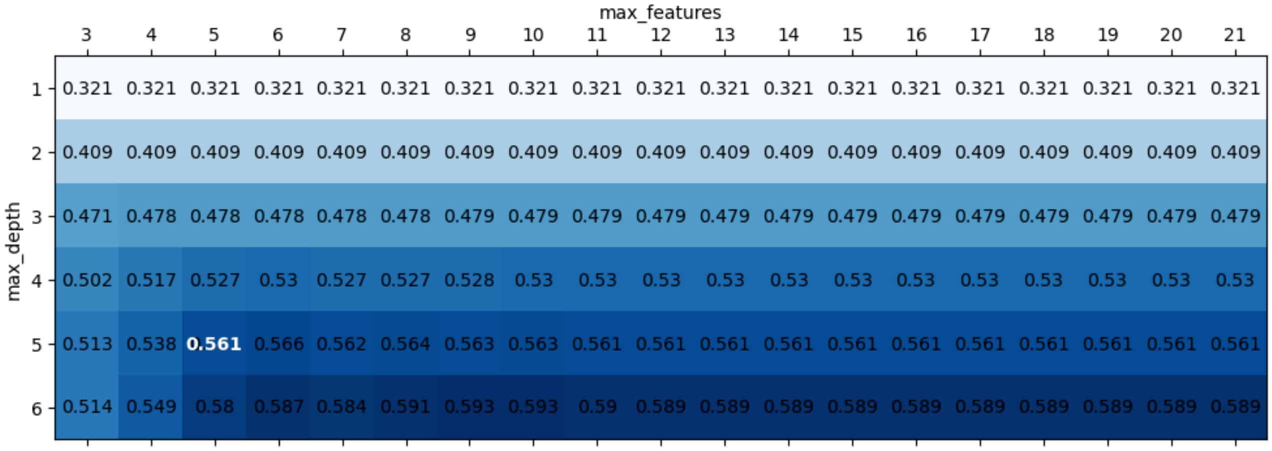


An Explanable Model - Decision Tree Regressor

In order to build an explanable model, We use a Decision Tree to predict the price. We still use random forest to select features to include in the model, and we conducted Grid Search on the number of max features in feature selection and the max depth of the Decision Tree. The result shows that with only 5 features and a Decision Tree of depth 5, we could have a model that achieves $R^2 = 0.576$. The 5 selected features are `fuel_gas`, `drive_fwd`, `cylinders`, `odometer`, `year`.

In [ ]:
```
import warnings
warnings.filterwarnings("ignore")

select_rf = SelectFromModel(RandomForestRegressor(random_state = 0))
pipe_dt = Pipeline([('preprocess',preprocess),
                    ('feature_selection',select_rf),
                    ('regressor', DecisionTreeRegressor(max_depth=6))])
param_grid = [{'feature_selection__max_features':range(3,22),
               'regressor__max_depth':range(1,7)}]
dt_best = GridSearchCV(pipe_dt, param_grid, return_train_score=True, cv=5, n_jobs=n_jobs)
dt_best.fit(global_X_trainval, global_y_trainval);
```

In [ ]:
```
results = pd.DataFrame(dt_best.cv_results_)[['param_feature_selection__max_features','param_regressor__max_depth','
results = results.pivot_table(index='param_regressor__max_depth', columns='param_feature_selection__max_features',v
fig, ax = plt.subplots(figsize=(12,12))
ax.matshow(results, cmap=plt.cm.Blues)
for i in range(0, results.shape[0]):
    for j in range(0, results.shape[1]):
        ax.text(j, i, str(round(results.iloc[i,j],3)),va='center', ha='center')

ax.set_xticks(range(len(results.columns)))
ax.set_xticklabels(results.columns)
ax.set_xlabel('max_features')
ax.xaxis.set_label_position('top')

ax.set_yticks(range(len(results.index)))
ax.set_yticklabels(results.index)
ax.set_ylabel('max_depth');
ax.text(2, 4, str(round(results.iloc[4,2],3)), color='white', weight='bold', va='center', ha='center');
```

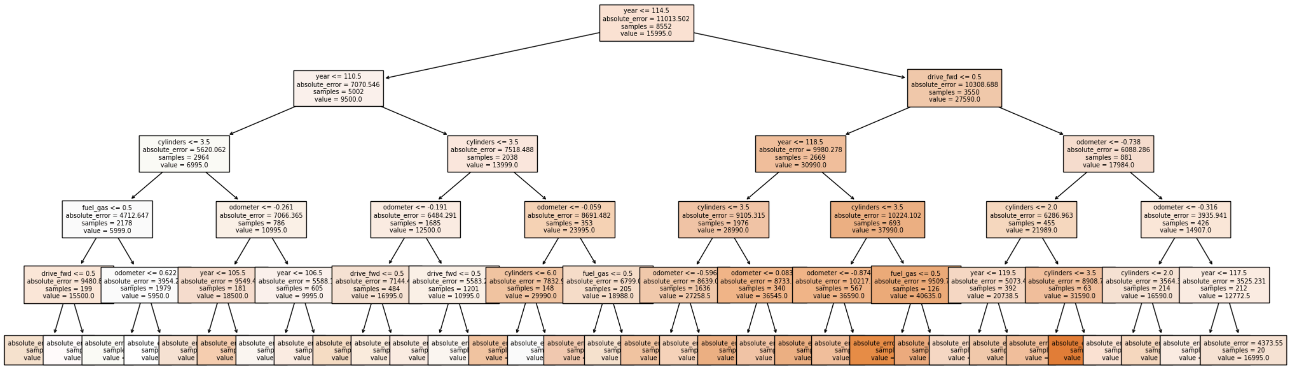


max_depth	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321	0.321
2	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409	0.409
3	0.471	0.478	0.478	0.478	0.478	0.478	0.479	0.479	0.479	0.479	0.479	0.479	0.479	0.479	0.479	0.479	0.479	0.479	0.479
4	0.502	0.517	0.527	0.53	0.527	0.527	0.528	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
5	0.513	0.538	**0.561**	0.566	0.562	0.564	0.563	0.563	0.561	0.561	0.561	0.561	0.561	0.561	0.561	0.561	0.561	0.561	0.561
6	0.514	0.549	0.58	0.587	0.584	0.591	0.593	0.593	0.59	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589	0.589

```
In [ ]: select_rf = SelectFromModel(RandomForestRegressor(random_state = 0, n_jobs=n_jobs), max_features=5)
        pipe_explainable = Pipeline([('preprocess',preprocess),
                                    ('feature_selection',select_rf),
                                    ('regressor', DecisionTreeRegressor(max_depth=5,criterion='absolute_error'))])
        pipe_explainable.fit(global_X_trainval, global_y_trainval);
```

```
In [ ]: from sklearn.tree import plot_tree
        fig, ax = plt.subplots(figsize=(25, 8))
        col_names = list(pipe_explainable.named_steps['preprocess'].transformers_[0][1].named_steps['cat_pre'].get_feature_
                    list(pipe_explainable.named_steps['preprocess'].transformers_[1][1].named_steps['ord_poly'].get_feature
                    ['odometer']+['year']+target
        mask = pipe_explainable.named_steps['feature_selection'].get_support().tolist()
        print('Selected features:',np.array(col_names)[mask])
        plot_tree(pipe_explainable.named_steps['regressor'], filled=True, feature_names = np.array(col_names)[mask], ax=ax,
        plt.show()
```

```
Selected features: ['fuel_gas' 'drive_fwd' 'cylinders' 'odometer' 'year']
```



```
In [ ]:
```