



# Group project: House price prediction

Boris Korotkov  
Robert Shaheen  
Vladimir Taubes

SCS\_3253\_009 Machine Learning



# Data source: ongoing Kaggle competition



## House Prices: Advanced Regression Techniques

Predict sales prices and practice feature engineering, RFs, and gradient boosting  
4,512 teams · Ongoing

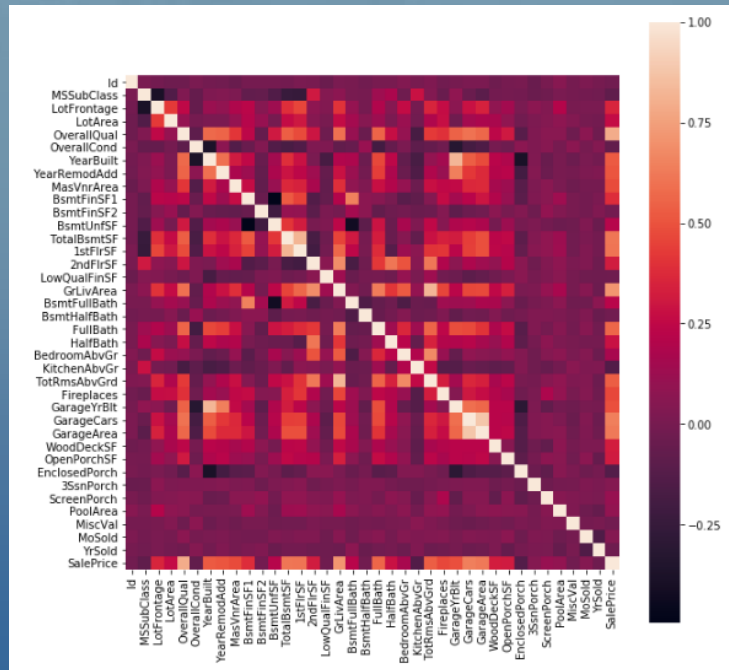
[Overview](#)[Data](#)[Kernels](#)[Discussion](#)[Leaderboard](#)[Rules](#)

There are 79 explanatory variables, 1460 rows in train dataset and 1459 rows in test dataset



# Data descriptive analysis

## Numeric columns correlation



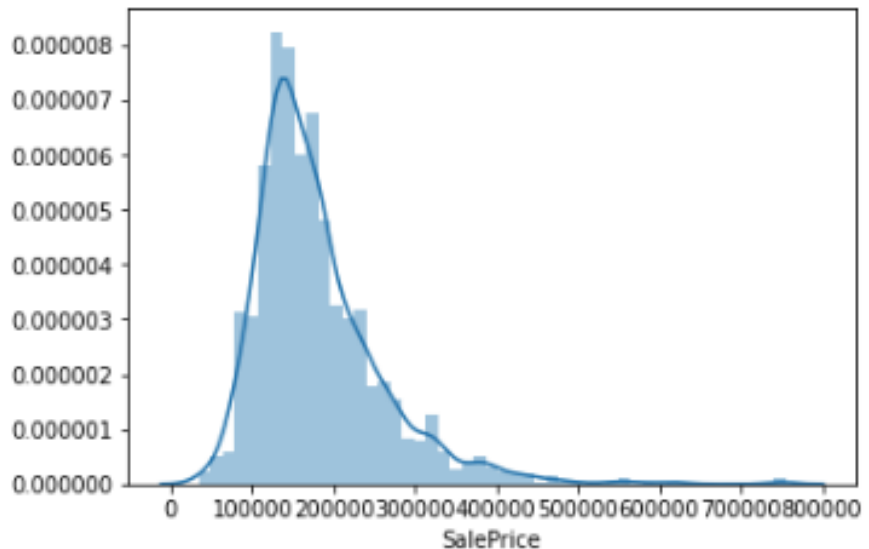


# Data descriptive analysis (cont.)

## Sales price distribution

```
df_train['SalePrice'].describe()
```

count	1460.000000
mean	180921.195890
std	79442.502883
min	34900.000000
25%	129975.000000
50%	163000.000000
75%	214000.000000
max	755000.000000
Name: SalePrice, dtype: float64	

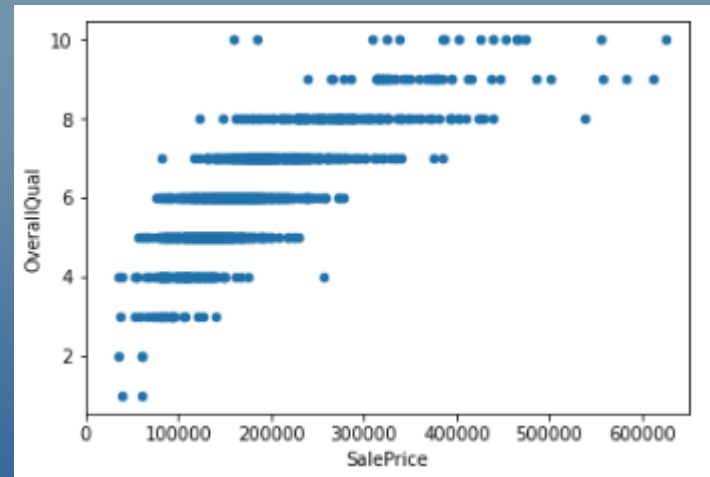
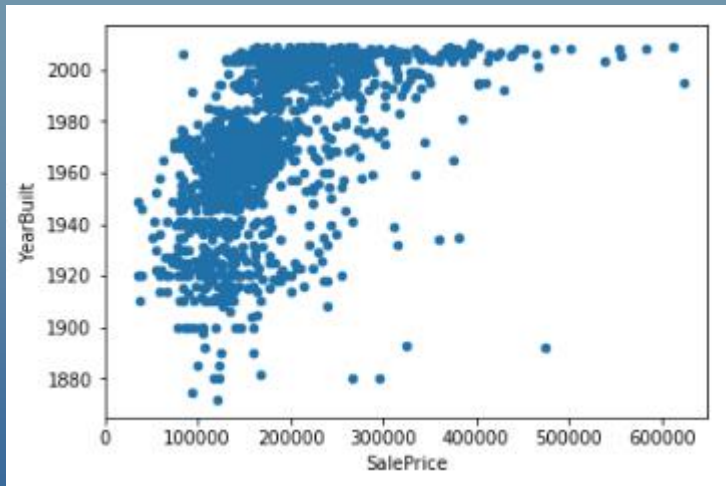






# Data descriptive analysis (cont.)

Sales price over year built and over house quality





# Data preparation

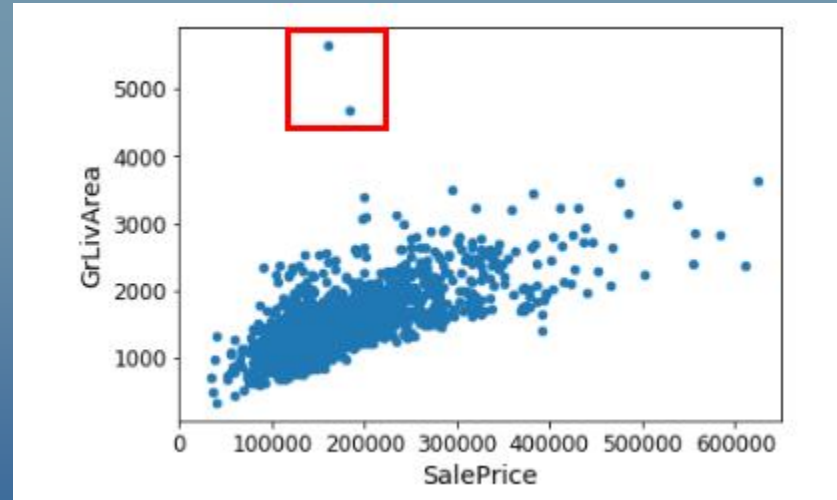
New features added:

- Years between renovation and sold date
- Above ground living area over Lot area
- Above ground living area over number of bedrooms
- Overall quality over Overall condition



## Data preparation (cont.)

- We noticed that source data has some significant outliers and removed two data point from the training dataset to increase accuracy of prediction.





## Data preparation (cont.)

- Data transformation of numerical and categorical features using:
  - Pipeline
  - SimpleImputer
  - MinMaxScaler
  - OneHotEncoder
- The feature number after transformation grew up from 79 to 291














# Base model selection

The following regression models have been selected for initial assessment:

- Linear Regression
- SVR
- Ridge
- Lasso
- ElasticNet
- SGDRegressor
- KNeighbors Regressor
- Decision Tree Regressor
- Random Forest Regressor

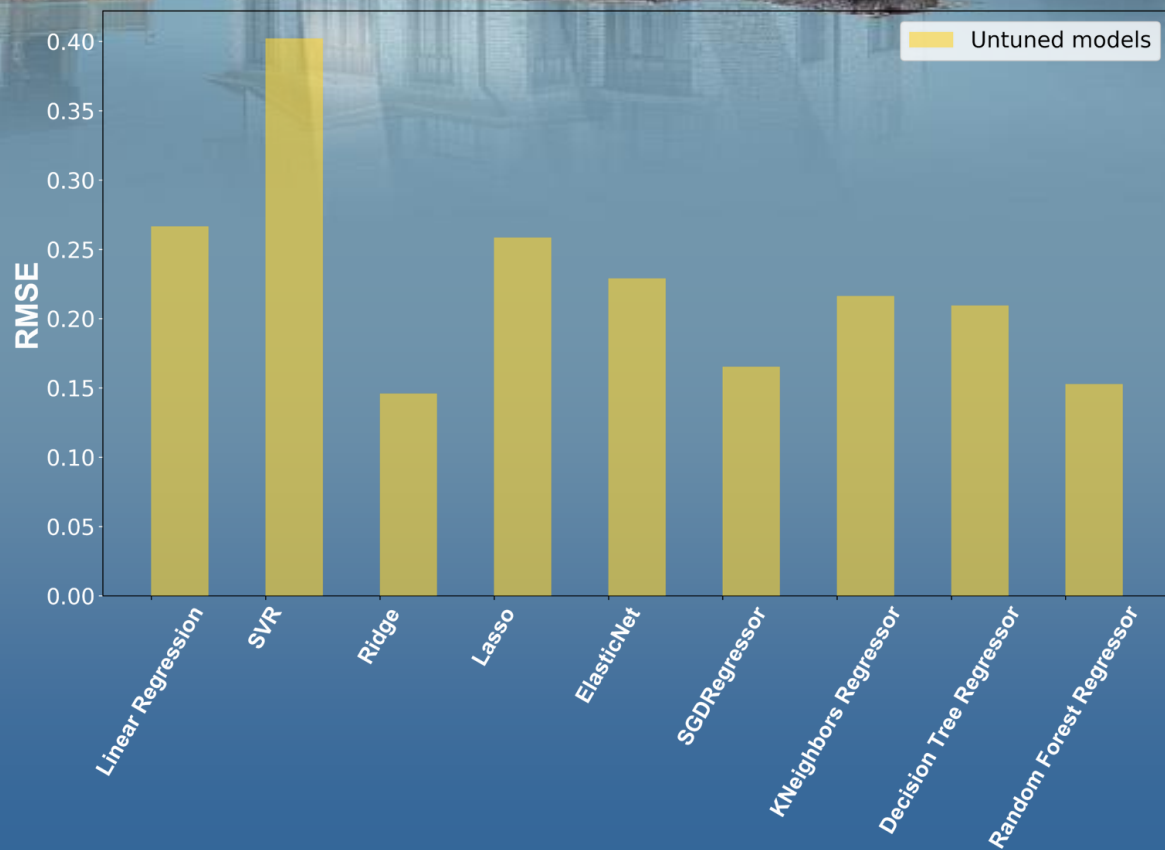
# Performance Metric

$RSME(\log(\text{SalePrice}), \log(\text{predicted SalePrice}))$

4	new	Ketian		0.10387	1	3d
5	▼ 3	Mohammed Amro		0.10567	1	13d
6	▼ 3	3rd Ring House	 	0.10677	1	21d
7	new	Serendipity_	   	0.10874	1	4d
8	▼ 4	Dmitry Starchenko		0.10915	1	10d



# RMSE untuned models





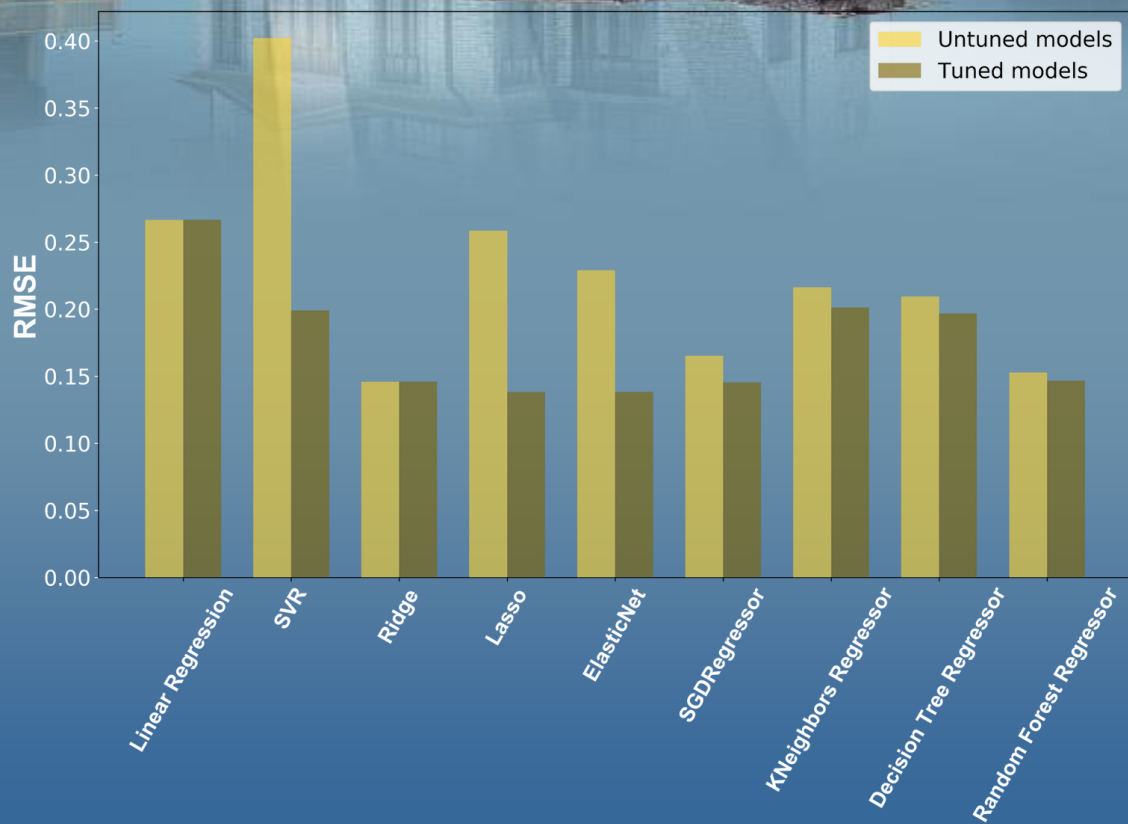
# Tuning hyperparameters

Model name (sklearn)	Tuned hyperparameters
LinearRegression	<code>{'fit_intercept': True}</code>
SVR	<code>{'max_depth': 16, 'max_features': 125, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 1000}</code>
Ridge	<code>{'alpha': 1.0, 'fit_intercept': True, 'solver': 'auto'}</code>
Lasso	<code>{'alpha': 71, 'max_iter': 100}</code>
ElasticNet	<code>{'alpha': 100, 'l1_ratio': 1.0}</code>
SGDRegressor	<code>{'eta0': 0.001, 'learning_rate': 'constant', 'loss': 'squared_epsilon_insensitive', 'max_iter': 50, 'penalty': 'none', 'power_t': 0.5}</code>
K-Neighbours Regressor	<code>{'algorithm': 'auto', 'n_neighbors': 7, 'p': 1, 'weights': 'distance'}</code>
Decision Tree Regressor	<code>{'max_depth': 9, 'max_features': 'auto', 'max_leaf_nodes': None, 'min_samples_leaf': 2, 'min_samples_split': 18}</code>
Random Forest Regressor	<code>{'max_depth': 16, 'max_features': 125, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 1000}</code>





# RMSE tuned models



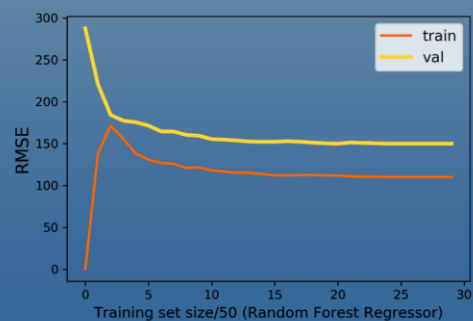
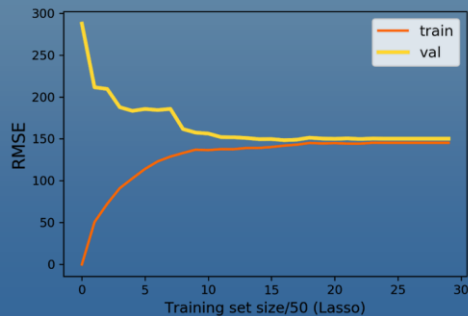
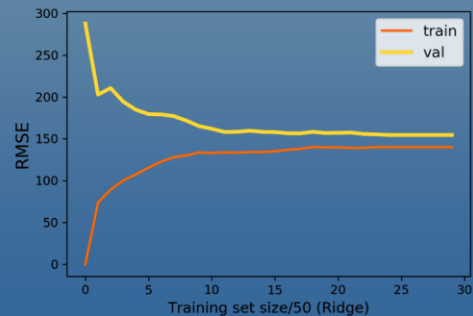
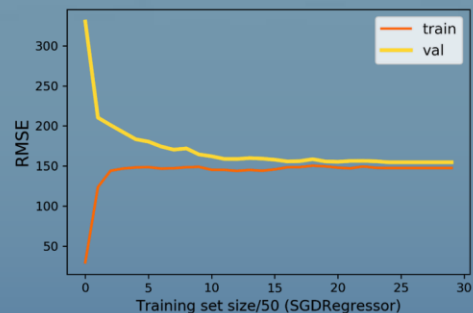
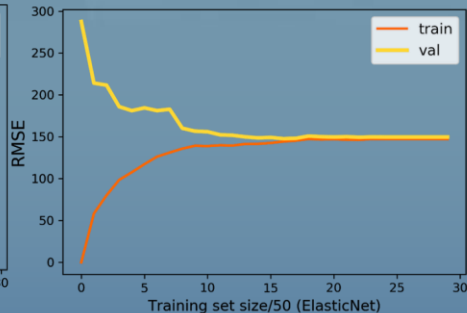
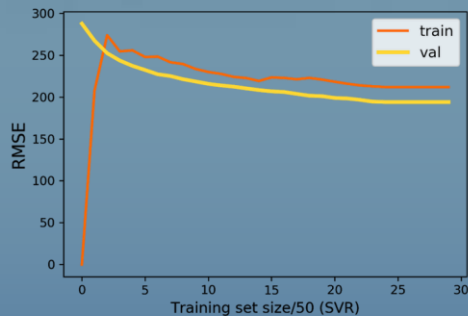
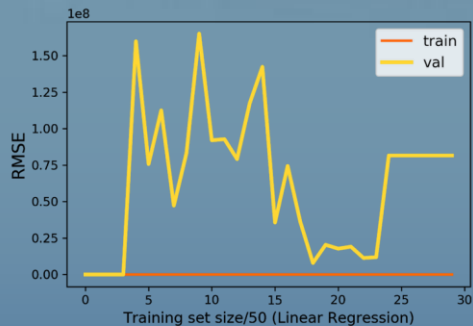


# RMSE tuned models

Model name (sklearn)	Untuned RMSE	Tuned RMSE	Performance Increase	Untuned r2	Tuned r2
LinearRegression	0.26636	0.26639	0.00%	0.57903	0.87793
SVR	0.40208	0.19893	↑50.53%	0.04099	0.74334
Ridge	0.14586	0.14586	0.00%	0.87380	0.89722
Lasso	0.25845	0.13809	↑46.57%	0.60378	0.90787
ElasticNet	0.22906	0.13824	↑39.65%	0.68876	0.90916
SGDRegressor	0.16523	0.14529	↑12.07%	0.83804	0.89647
K-Neighbours Regressor	0.21635	0.20128	↑6.96%	0.72234	0.78280
Decision Tree Regressor	0.20935	0.19671	↑6.04%	0.74002	0.76758
Random Forest Regressor	0.15267	0.14660	↑3.98%	0.86175	0.90821



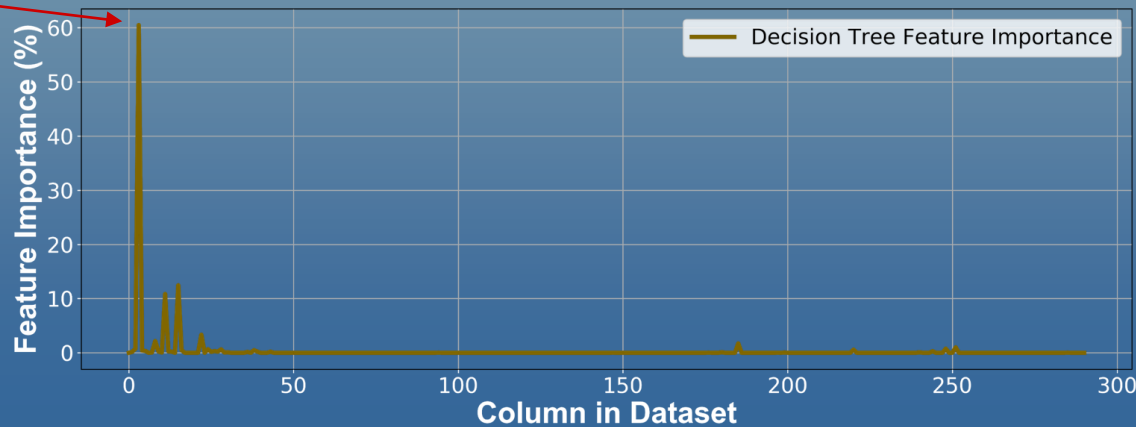
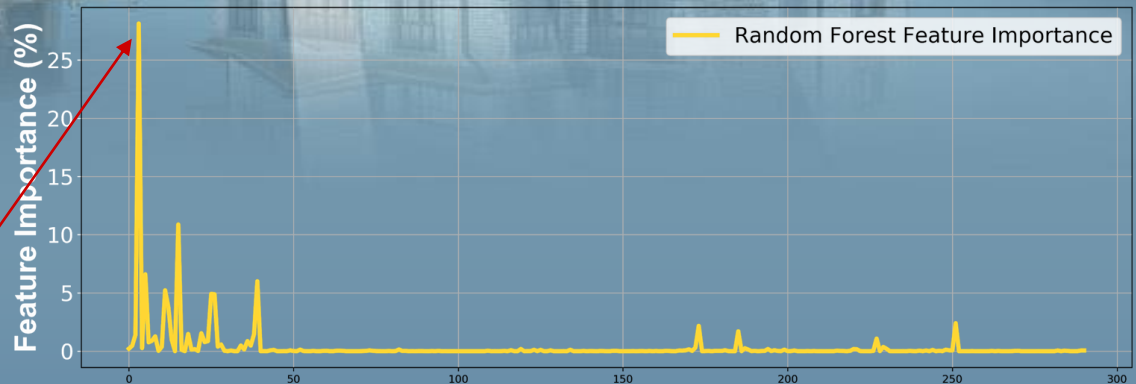
# Learning curves for select tuned models





# Feature importance

Most important feature: LotArea







# Aggregate Learning

The following models were selected for ensemble learning:

- Bagging Regressor
- Gradient Boosting Regressor
- Stacking



# Bagging Regressor

## Model name (sklearn)

BaggingRegressor

## Tuned hyperparameters

```
{'base_estimator': SVR(C=113564, cache_size=200, coef0=0.0,
degree=3, epsilon=0.1, gamma=0.0007790692366582295,
kernel='rbf', max_iter=-1, shrinking=True, tol=0.001,
verbose=False), 'bootstrap': True, 'max_features': 1.0,
'max_samples': 1.0, 'n_estimators': 7}
```

Model name (sklearn)	Tuned RMSE	Tuned r2
BaggingRegressor	0.19855	0.74745

↓43.8%



# Gradient Boosting Regressor

## Model name (sklearn)

GradientBoostingRegressor

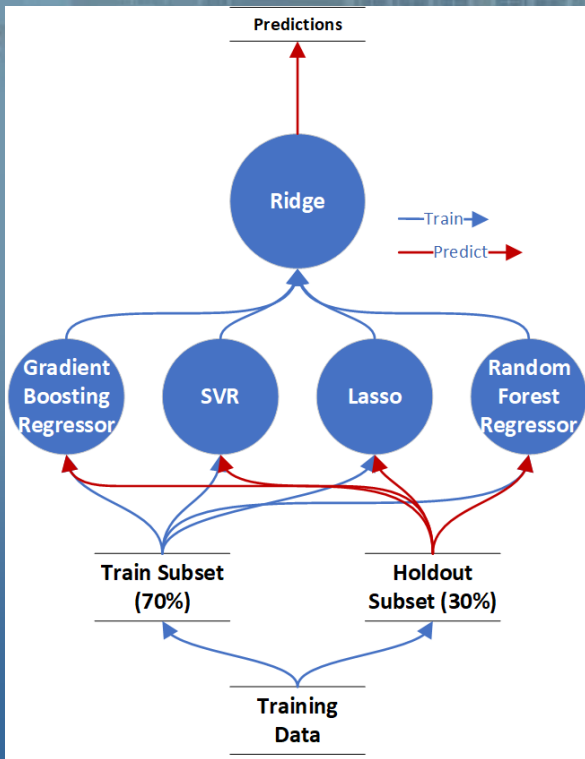
## Tuned hyperparameters

```
{'learning_rate': 0.1, 'max_depth': 6, 'max_features': 'auto',  
'max_leaf_nodes': 10, 'min_samples_leaf': 2, 'min_samples_split':  
50, 'n_estimators': 300, 'subsample': 0.9, 'warm_start': True}
```

Model name (sklearn)	Tuned RMSE	Tuned r2
GradientBoostingRegressor	0.11468	0.91971

↑17.0%

# Stacking Regressor



Model name (sklearn)

Tuned RMSE

Tuned r2

GradientBoostingRegressor

0.11468

0.91971

↑17.1%










# Competition Public Leaderboard

The Neural network was used for final prediction using test dataset and the result was submitted to Kaggle public leaderboard.

The submission rank was 1340 out of 4,632, top 28.9%. Not bad!

1338	▼ 177	AT073001		0.12487	4	1mo
1339	▲ 882	ywleung		0.12488	17	2d
1340	new	shahroberto	  	0.12491	14	15h

*The top 10 score was about 0.11*



Questions?