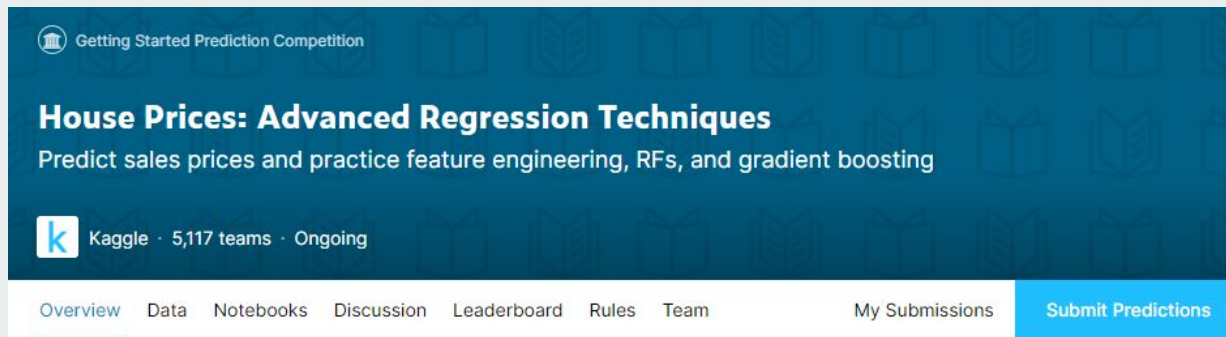




House Price Prediction



Presenters:

John Lee

Nathan Nusaputra

Ryan Sawasaki

August 05, 2020

Introduction

- House Price Prediction - Kaggle
- Regression Problem
- Steps:
 - Exploratory Data Analysis
 - Formulate Baseline
 - Linear Regression Model
 - Decision Trees with Ensembling Methods
 - Neural Network

Iowa

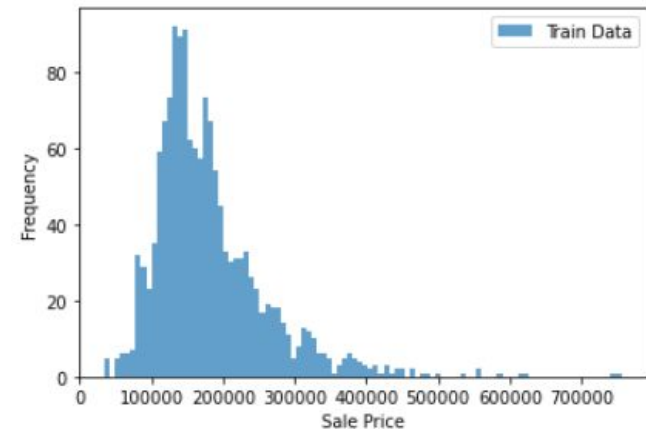


Exploratory Data Analysis

- Dataset:
 - Train Data - 1460 samples, 79 Features
 - Test Data - 1459 samples w/o labels
- Data:
 - 36 numerical data, 43 categorical
 - Many features missing data
 - No duplications

Number of train features with missing data: 19
Number of test features with missing data: 33

	feature	train_missing_count	test_missing_count
16	PoolQC	1453.0	1456.0
18	MiscFeature	1406.0	1408.0
1	Alley	1369.0	1352.0
17	Fence	1179.0	1169.0
10	FireplaceQu	690.0	730.0
0	LotFrontage	259.0	227.0
12	GarageYrBlt	81.0	78.0
13	GarageFinish	81.0	78.0
14	GarageQual	81.0	78.0
15	GarageCond	81.0	78.0
11	GarageType	81.0	76.0
5	BsmtCond	37.0	45.0
6	BsmtExposure	38.0	44.0
4	BsmtQual	37.0	44.0
8	BsmtFinType2	38.0	42.0
7	BsmtFinType1	37.0	42.0





Exploratory Data Analysis

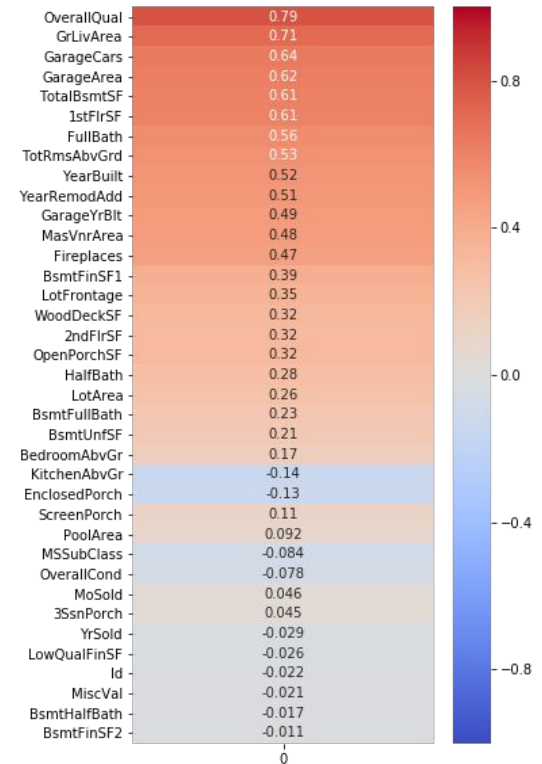
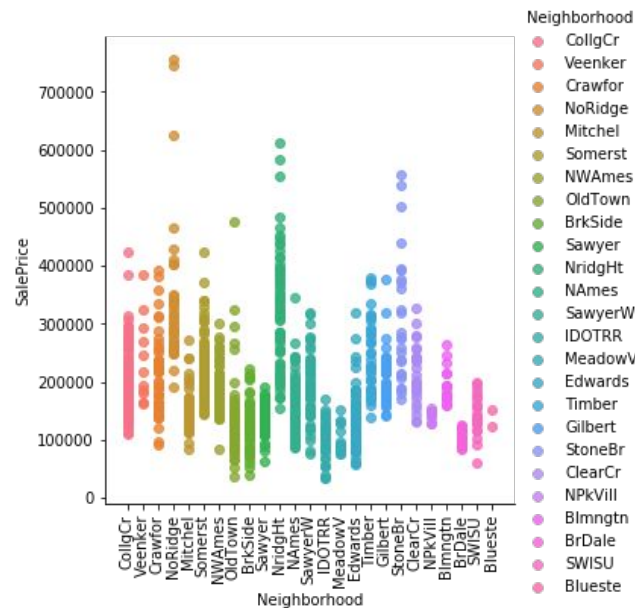
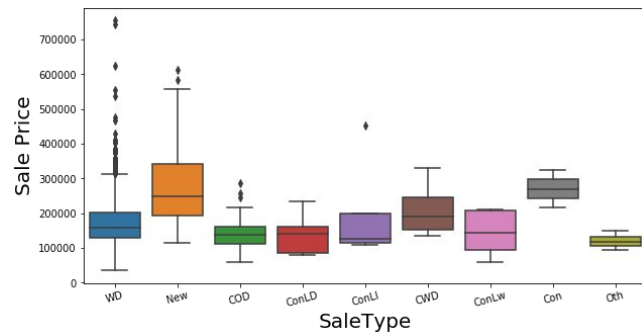
- Missing Data:
 - Drop features when missing from 50% of samples
 - Convert to 0 or 'None'
- Data formatting:
 - Ordinal Data: Mapped to Ranked Integers
 - Nominal Data: Convert to Dummies
 - Normalized Features w/ Min-Max Scaler
 - Train_test_split at 80/20

```
train_data: (1168, 222)
train_labels: (1168,)
dev_data: (292, 222)
dev_labels: (292,)
test_data (1459, 222)
```

```
train_df_raw2['BsmtQual']=train_df_raw2['BsmtQual'].map({'Ex':5,'Gd':4,'TA':3,'Fa':2,'Po':1,'NA':0,'None':0})
train_df_raw2['BsmtCond']=train_df_raw2['BsmtCond'].map({'Ex':5,'Gd':4,'TA':3,'Fa':2,'Po':1,'NA':0,'None':0})
```

Exploratory Data Analysis

- Correlation Graphics



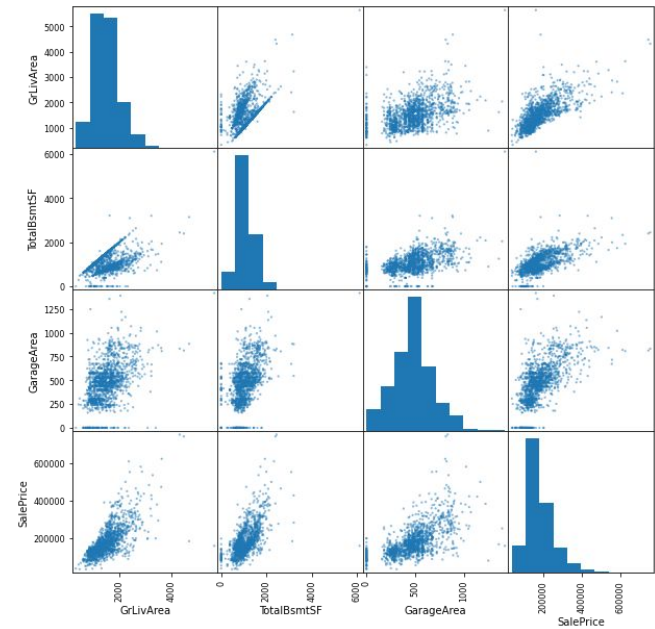


Baseline

- Baseline model: Linear Regression
- 3 features
 - Ground Floor Living Area
 - Total Basement Square Footage
 - Garage Area
- Estimated Function:

$$\text{SalePrice} = -20418.29 + 67.47 * \text{GrLivArea} + 46.55 * \text{TotalBsmtSF} + 104.50 * \text{GarageArea}$$

- Mean Squared Log Error (MSLE) used as evaluation metric



Baseline MSLE: 0.043

Linear Regression

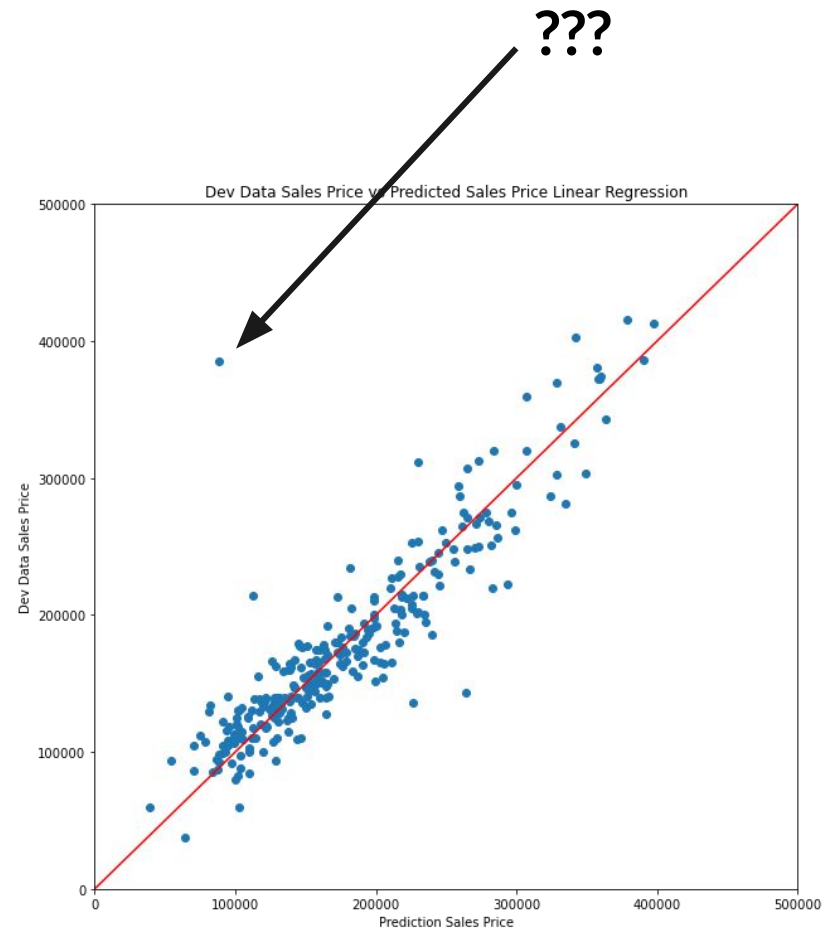
Additional EDA

- Outliers and negative predicted values
- Sale Condition feature included trades, short sales, foreclosures and sales between family members
- R^2 used as another metric for evaluation of linear regression models

R^2 : 0.748

Accuracy : 0.727

MSLE: 0.043 \Rightarrow **MSLE: 0.038**



Linear Regression

Feature Selection - Part 1

- Feature Selection based on correlation with Sale Price
- Created function to return features with a correlation greater than an input threshold
- Used for-loop to find the optimal threshold

R^2 : 0.940

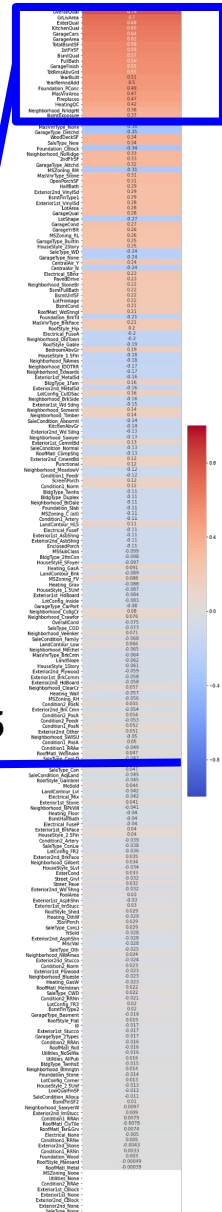
Accuracy : 0.863

MSLE: 0.038 \Rightarrow MSLE: 0.030

Correlation with Sale Price

OverallQual	0.79
GrLivArea	0.7
ExterQual	0.68
KitchenQual	0.65
GarageCars	0.64
GarageArea	0.62
TotalBsmtSF	0.59
1stFlrSF	0.59
BsmtQual	0.57
FullBath	0.56
GarageFinish	0.55
TotRmsAbvGrd	0.55
YearBuilt	0.51
YearRemodAdd	0.5
Foundation_PConc	0.49
MasVnrArea	0.47
Fireplaces	0.47
HeatingQC	0.42
Neighborhood_MridgHt	0.38
BsmtExposure	0.37
BsmtFinSF1	0.36

Threshold = 0.045



Linear Regression

Feature Selection - Part 2

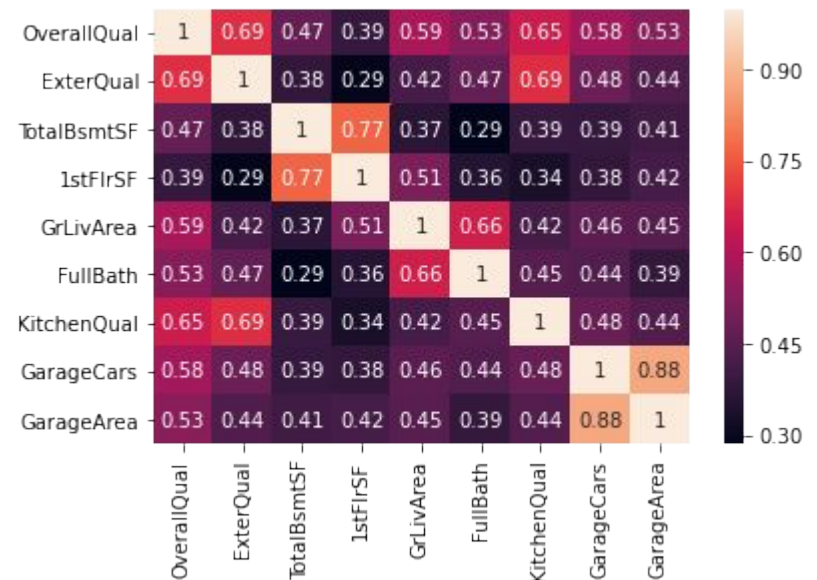
- Feature Selection based on multicollinear features
- Drop one of the multicollinear features
- Expectation that it would increase performance and reduce overfitting
- No significant improvement for this data set

R^2 : 0.939

Accuracy : 0.864

MSLE: 0.030 \Rightarrow **MSLE: 0.029**

Example Correlation Matrix



Linear Regression

Log - Linear Model

- Predicted vs Actual Sales Price showed a trend of increasing errors with increase in house price
- Good candidate for log-linear model
- Increased R^2 and accuracy, while closing the gap. Improved predictive model
- Reduced MSLE by over half

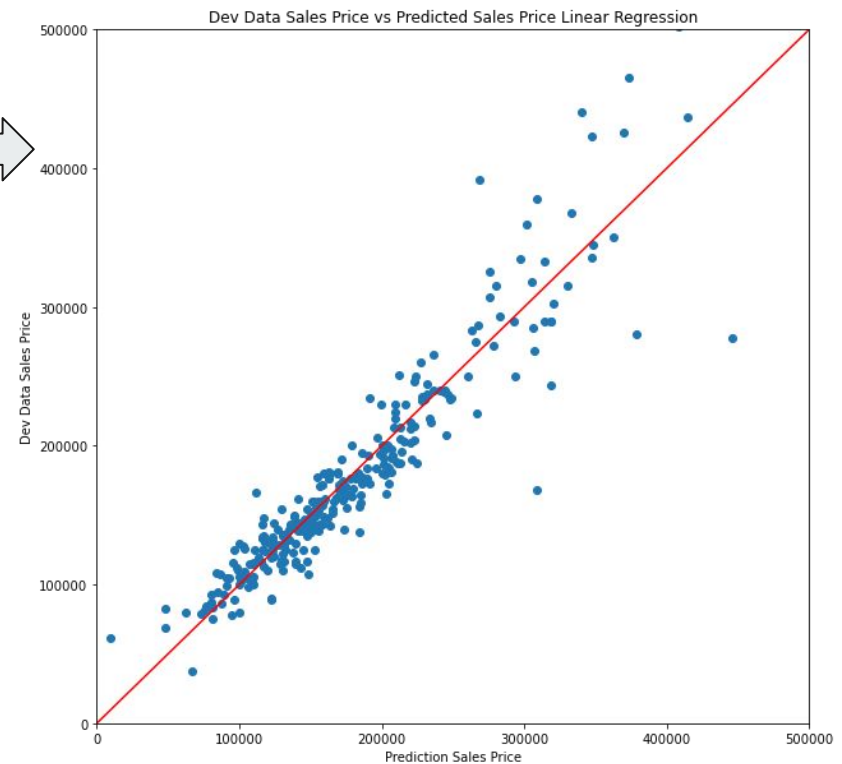
R^2 : 0.951

Accuracy : 0.921

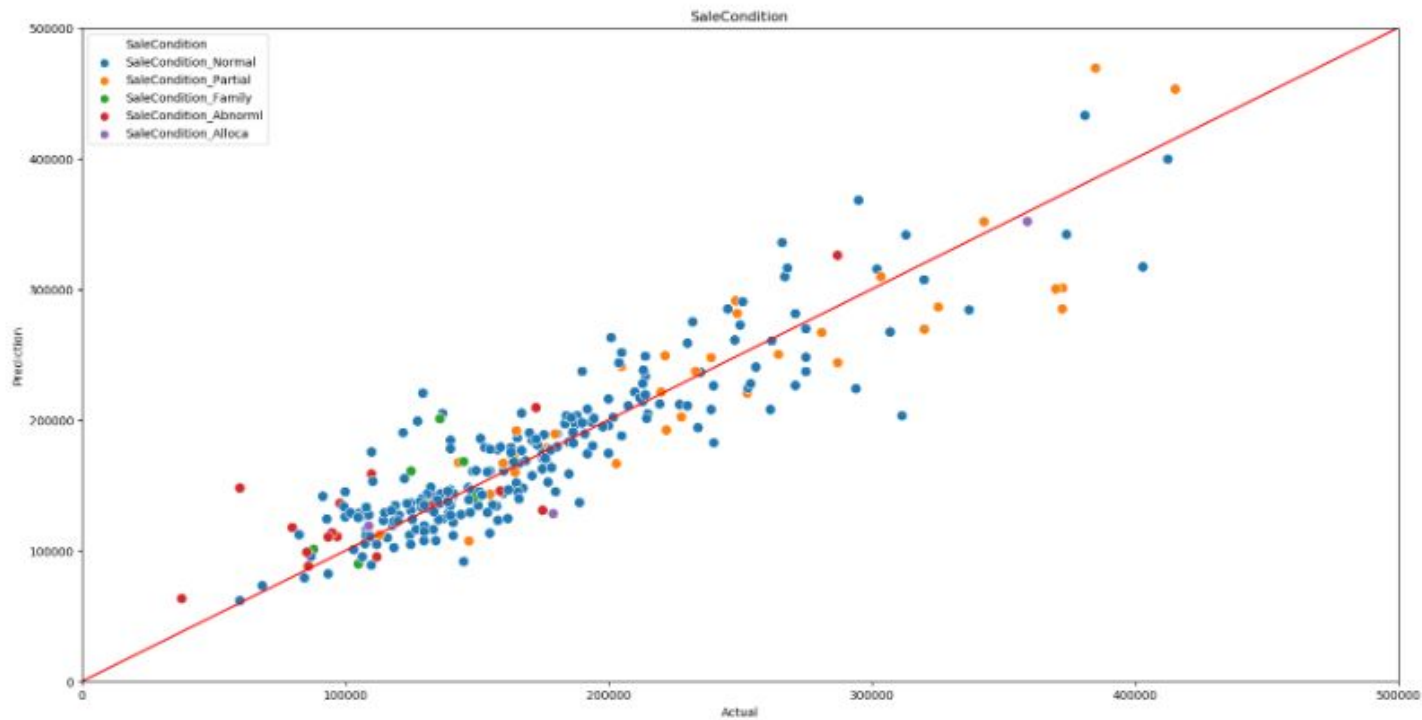
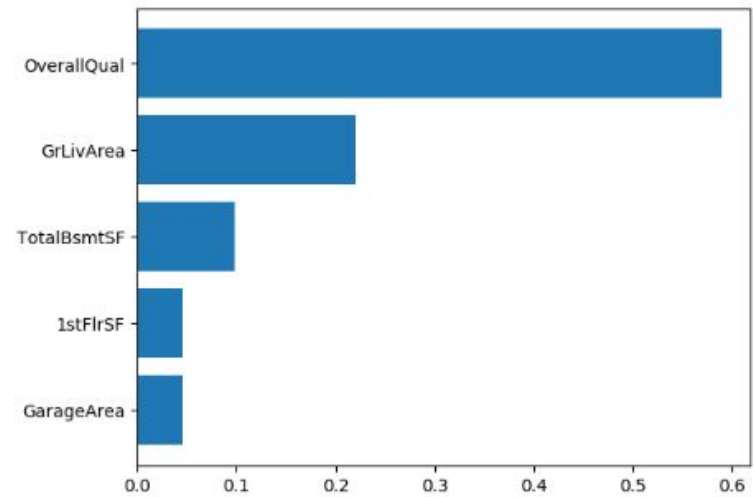
MSLE: 0.029 \Rightarrow **MSLE: 0.012**

Kaggle Score: 0.13361 (Top 36%)

Sales Price Predicted vs. Actual



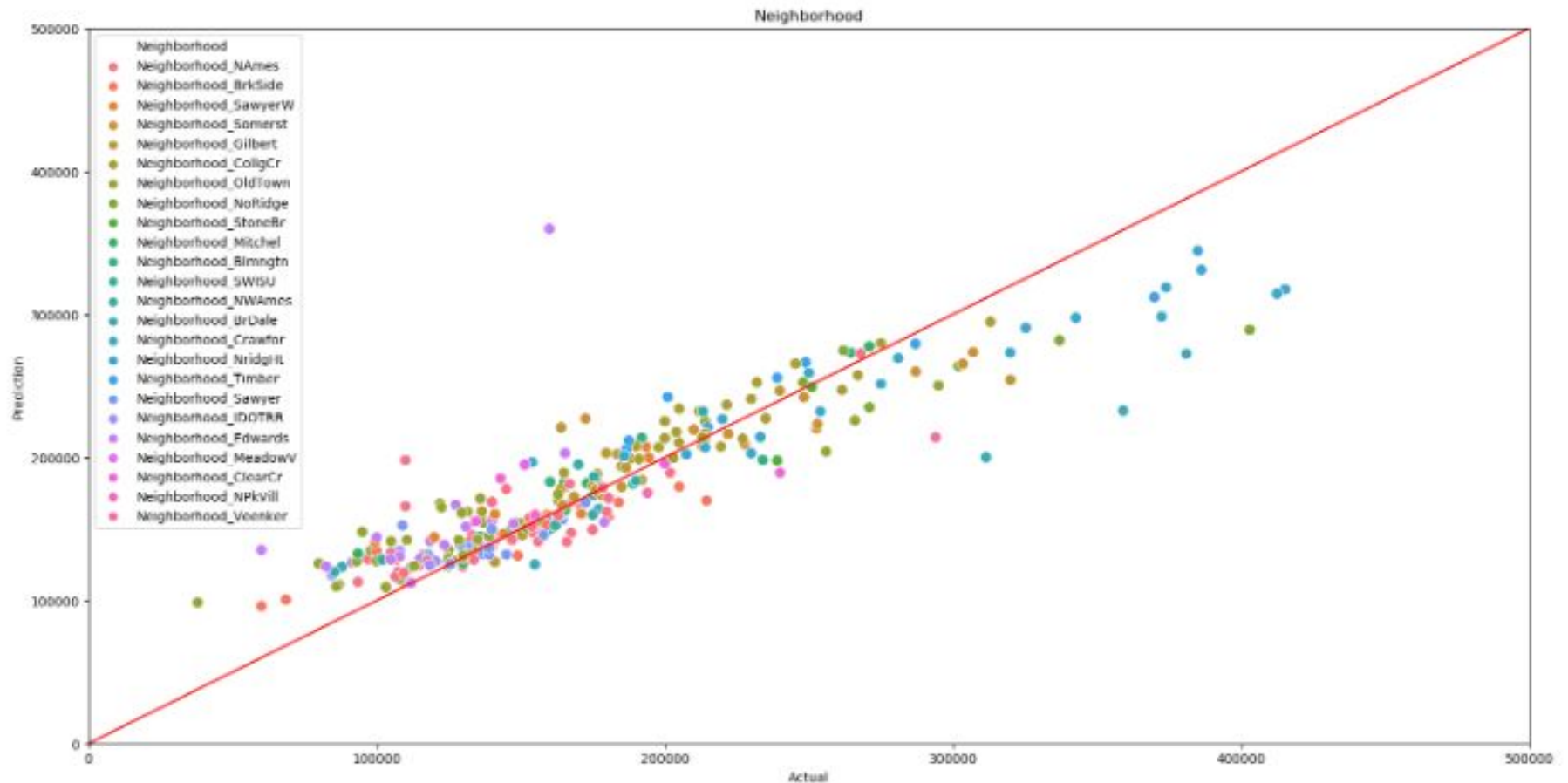
Gradient Boosting



MSLE: 0.028

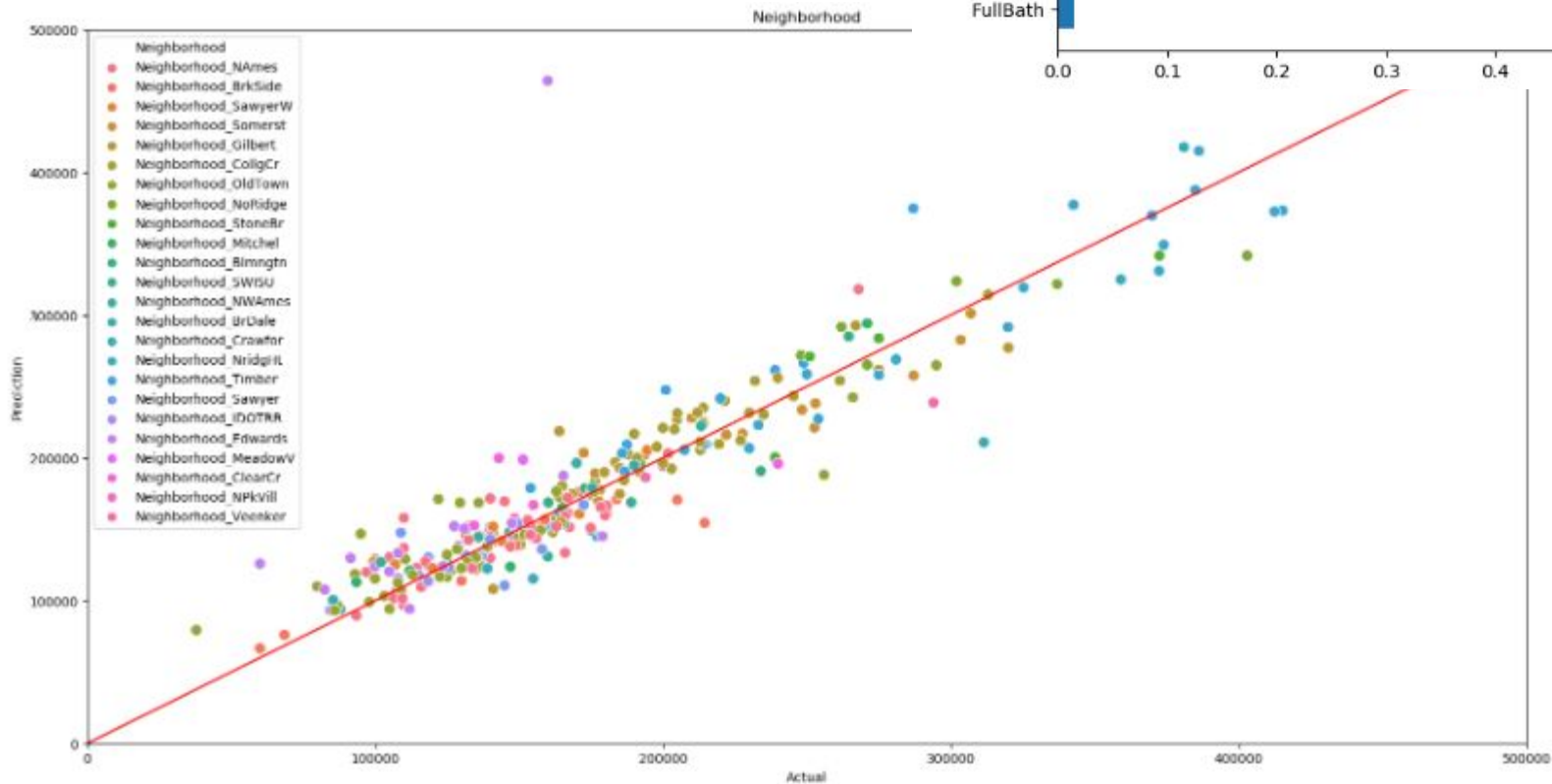
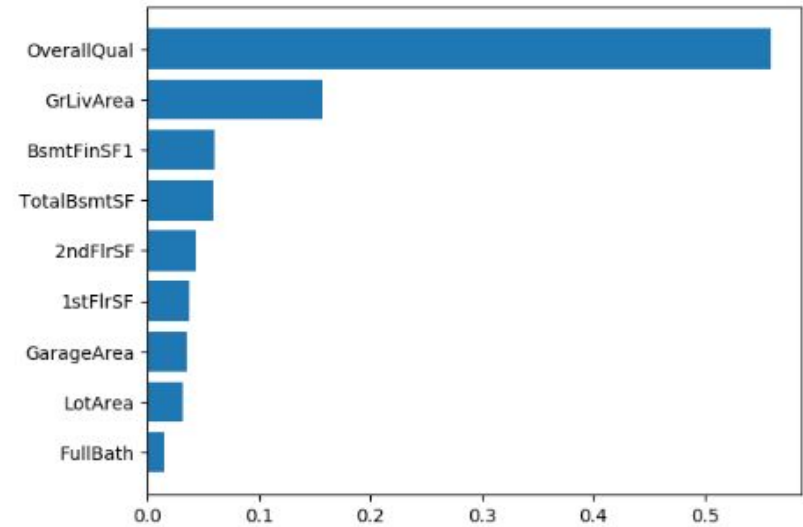
Bagging-Decision Tree

MSLE: 0.031 \Rightarrow MSLE: 0.030



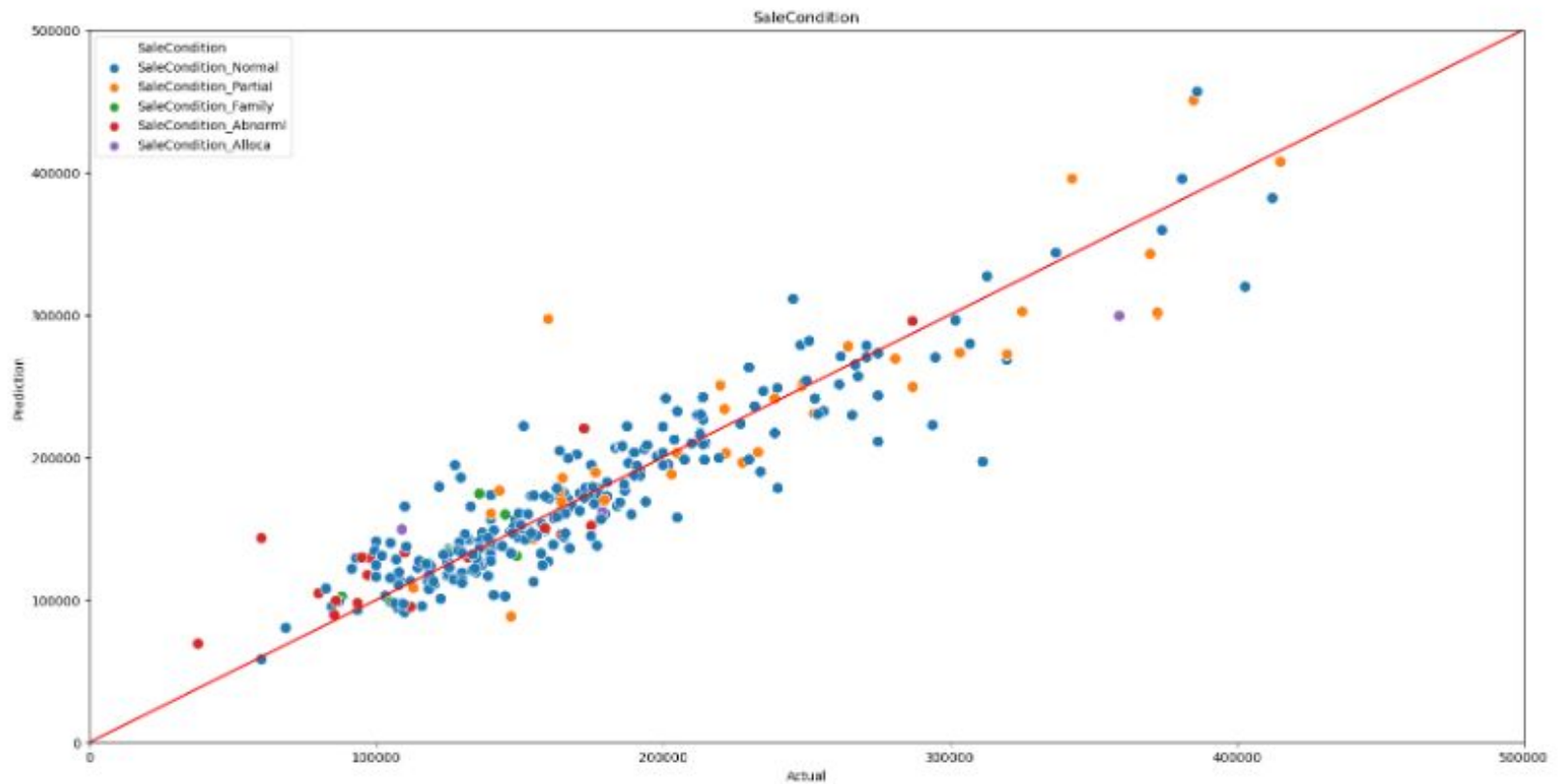
Random Forest

MSLE: 0.026



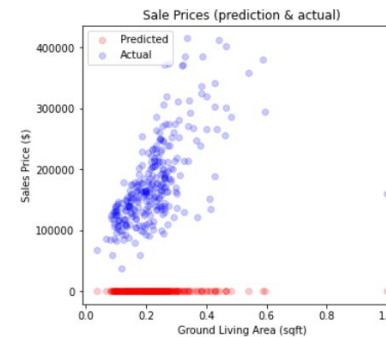
Extra Trees

MSLE: 0.024



Neural Network

- Varied parameters
- Considerations:
 - Training time
 - Overfitting and Evaluation (MSLE)
 - Dataset



Single Layer

- Epochs
- Batch size
- Optimizers and Learning Rates
- Activation functions

2-Layer

- Number of nodes
- Dropout, regularizations

3-, 4-Layer

- Applied findings from before

Neural Network - Epochs, Batch, Optimizer, Activation

	Batch	Epochs	Activation	Optimizer	Learning_Rate	Train_Time	MSLE
0	1	5	linear	SGD	default	3.778321	44.578516
1	1	5	linear	Adam	default	4.172123	57.942298
2	1	5	relu	SGD	default	4.187844	44.574707
3	1	5	relu	Adam	default	4.653705	57.665736
4	1	10	linear	SGD	default	8.051262	40.377461
5	1	10	linear	Adam	default	7.908419	44.010216
6	1	10	relu	SGD	default	7.706697	40.371325
7	1	10	relu	Adam	default	7.766160	43.302516
8	10	5	linear	SGD	default	0.946882	60.123714
9	10	5	linear	Adam	default	0.982346	89.829209
10	10	5	relu	SGD	default	1.134891	60.114417
11	10	5	relu	Adam	default	1.169575	90.051596
12	10	10	linear	SGD	default	1.558134	55.180212
13	10	10	linear	Adam	default	1.666607	80.932679
14	10	10	relu	SGD	default	1.649209	55.186514
15	10	10	relu	Adam	default	1.616487	80.934533

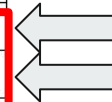
Neural Network - Epochs, Batch, Optimizer, Activation

	Batch	Epochs	Activation	Optimizer	Learning_Rate	Train_Time	MSLE
0	1	5	linear	SGD	default	3.778321	44.578516
1	1	5	linear	Adam	default	4.172123	57.942298
2	1	5	relu	SGD	default	4.187844	44.574707
3	1	5	relu	Adam	default	4.653705	57.665736
4	1	10	linear	SGD	default	8.051262	40.377461
5	1	10	linear	Adam	default	7.908419	44.010216
6	1	10	relu	SGD	default	7.706697	40.371325
7	1	10	relu	Adam	default	7.766160	43.302516
8	10	5	linear	SGD	default	0.946882	60.123714
9	10	5	linear	Adam	default	0.982346	89.829209
10	10	5	relu	SGD	default	1.134891	60.114417
11	10	5	relu	Adam	default	1.169575	90.051596
12	10	10	linear	SGD	default	1.558134	55.180212
13	10	10	linear	Adam	default	1.666607	80.932679
14	10	10	relu	SGD	default	1.649209	55.186514
15	10	10	relu	Adam	default	1.616487	80.934533





Neural Network - Epochs, Batch, Optimizer, Activation

	Batch	Epochs	Activation	Optimizer	Learning_Rate	Train_Time	MSLE
0	1	5	linear	SGD	default	3.778321	44.578516
1	1	5	linear	Adam	default	4.172123	57.942298
2	1	5	relu	SGD	default	4.187844	44.574707
3	1	5	relu	Adam	default	4.653705	57.665736
4	1	10	linear	SGD	default	8.051262	40.377461
5	1	10	linear	Adam	default	7.908419	44.010216
6	1	10	relu	SGD	default	7.706697	40.371325
7	1	10	relu	Adam	default	7.766160	43.302516
8	10	5	linear	SGD	default	0.946882	60.123714
9	10	5	linear	Adam	default	0.982346	89.829209
10	10	5	relu	SGD	default	1.134891	60.114417
11	10	5	relu	Adam	default	1.169575	90.051596
12	10	10	linear	SGD	default	1.558134	55.180212
13	10	10	linear	Adam	default	1.666607	80.932679
14	10	10	relu	SGD	default	1.649209	55.186514
15	10	10	relu	Adam	default	1.616487	80.934533



Neural Network - Learning Rate, Training Time



	Batch	Epochs	Activation	Optimizer	Learning_Rate	Train_Time	MSLE
0	1	5	relu	SGD	0.01	3.866009	44.575440
1	1	5	relu	SGD	0.10	3.911050	31.415750
2	1	5	relu	SGD	1.00	3.950958	20.682024
3	1	5	relu	SGD	10.00	4.005332	10.905133
4	1	5	relu	SGD	100.00	4.245734	3.059724
5	1	5	relu	SGD	1000.00	3.977953	1.784549
6	1	5	relu	SGD	10000.00	4.384743	11.329739

	Batch	Epochs	Activation	Optimizer	Learning_Rate	Train_Time	MSLE
0	10	1	relu	SGD	100	0.445323	2.71989
1	10	5	relu	SGD	100	0.965775	5.37876
2	10	10	relu	SGD	100	1.773728	9.71387
3	10	20	relu	SGD	100	2.773181	5.93881
4	10	50	relu	SGD	100	6.466962	2.89318
5	10 (PREDICTION)	100	N/A	N/A	N/A	12.554698	N/A
6	10 (PREDICTION)	1000	N/A	N/A	N/A	122.008003	N/A
7	10 (PREDICTION)	2000	N/A	N/A	N/A	243.622787	N/A
8	10 (PREDICTION)	3000	N/A	N/A	N/A	365.237571	N/A
9	10 (PREDICTION)	5000	N/A	N/A	N/A	608.467139	N/A
10	10 (PREDICTION)	10000	N/A	N/A	N/A	1216.541059	N/A



Neural Network - Learning Rate, Training Time

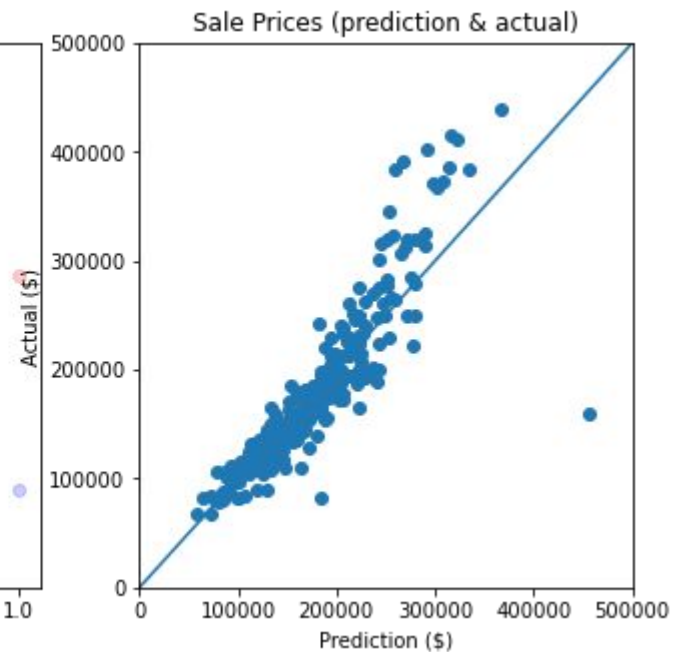
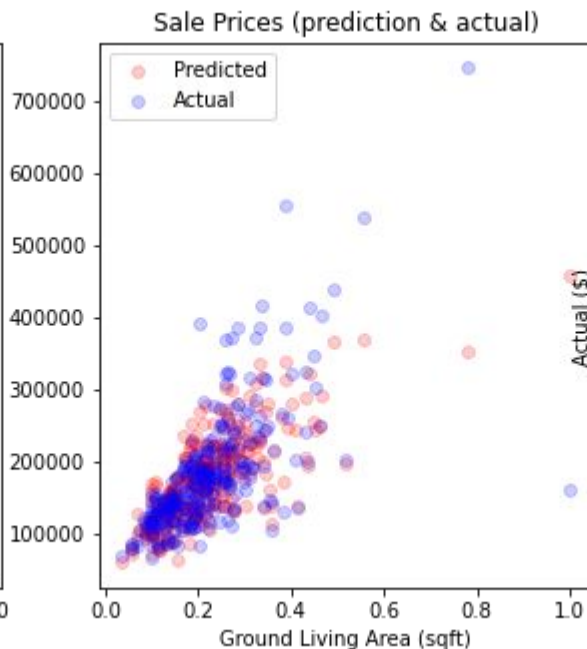
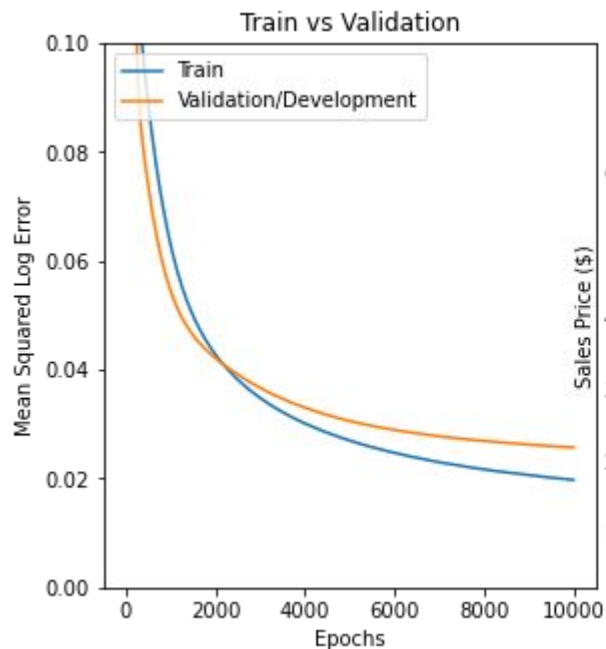
	Batch	Epochs	Activation	Optimizer	Learning_Rate	Train_Time	MSLE
0	1	5	relu	SGD	0.01	3.866009	44.575440
1	1	5	relu	SGD	0.10	3.911050	31.415750
2	1	5	relu	SGD	1.00	3.950958	20.682024
3	1	5	relu	SGD	10.00	4.005332	10.905133
4	1	5	relu	SGD	100.00	4.245734	3.059724
5	1	5	relu	SGD	1000.00	3.977953	1.784549
6	1	5	relu	SGD	10000.00	4.384743	11.329739

	Batch	Epochs	Activation	Optimizer	Learning_Rate	Train_Time	MSLE
0	10	1	relu	SGD	100	0.445323	2.71989
1	10	5	relu	SGD	100	0.965775	5.37876
2	10	10	relu	SGD	100	1.773728	9.71387
3	10	20	relu	SGD	100	2.773181	5.93881
4	10	50	relu	SGD	100	6.466962	2.89318
5	10 (PREDICTION)	100	N/A	N/A	N/A	12.554698	N/A
6	10 (PREDICTION)	1000	N/A	N/A	N/A	122.008003	N/A
7	10 (PREDICTION)	2000	N/A	N/A	N/A	243.622787	N/A
8	10 (PREDICTION)	3000	N/A	N/A	N/A	365.237571	N/A
9	10 (PREDICTION)	5000	N/A	N/A	N/A	608.467139	N/A
10	10 (PREDICTION)	10000	N/A	N/A	N/A	1216.541059	N/A


Neural Network - Single Layer

Mean Squared Log Error: 0.025

Kaggle Score: 0.1655 (Top 70%)



Neural Network - (2 layers) Nodes, Dropouts, Regularization



	Batch	Epochs	Activation	Optimizer	Learning_Rate	Train_Time	MSLE	Hidden_Layer_Nodes
0	1	100	relu	Adam	0.01	86.037419	0.029421	5
1	1	100	relu	Adam	0.01	87.316849	0.023116	10
2	1	100	relu	Adam	0.01	93.573366	0.023799	50
3	1	100	relu	Adam	0.01	89.124559	0.025354	100
4	1	100	relu	Adam	0.01	89.952543	0.026068	222
5	1	100	relu	Adam	0.01	115.683921	0.028017	500
6	1	100	relu	Adam	0.01	152.967657	0.028844	1000

	Batch	Epochs	Dropout	L1Reg	L2Reg	Train_Time	MSLE	Hidden_Layer_Nodes
0	1	100	No	No	No	89.754630	0.026004	10
1	1	100	0.5	No	No	88.080116	0.040753	10
2	1	100	No	1st Layer	No	88.080116	0.139776	10
3	1	100	No	2nd Layer	No	88.080116	0.066995	10
4	1	100	No	Both Layers	No	88.080116	0.372221	10
5	1	100	No	No	1st Layer	86.110852	0.139557	10
6	1	100	No	No	2nd Layer	85.142392	0.083193	10
7	1	100	No	No	Both Layers	85.257687	2.605657	10

Neural Network - Nodes, Dropouts, Regularization

	Batch	Epochs	Activation	Optimizer	Learning_Rate	Train_Time	MSLE	Hidden_Layer_Nodes
0	1	100	relu	Adam	0.01	86.037419	0.029421	5
1	1	100	relu	Adam	0.01	87.316849	0.023116	10
2	1	100	relu	Adam	0.01	93.573366	0.023799	50
3	1	100	relu	Adam	0.01	89.124559	0.025354	100
4	1	100	relu	Adam	0.01	89.952543	0.026068	222
5	1	100	relu	Adam	0.01	115.683921	0.028017	500
6	1	100	relu	Adam	0.01	152.967657	0.028844	1000

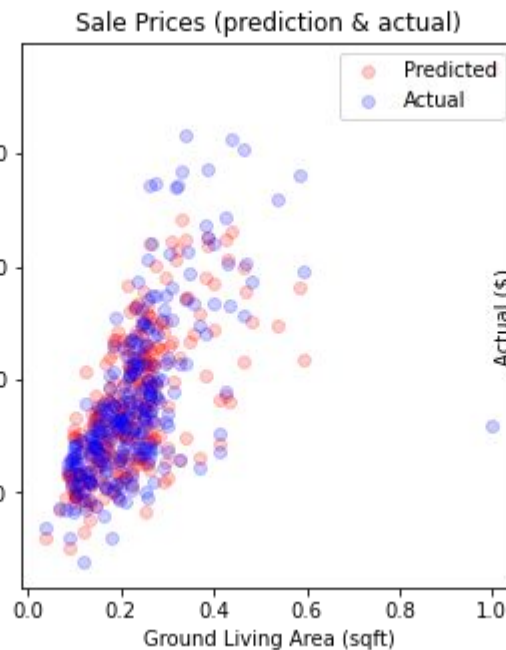
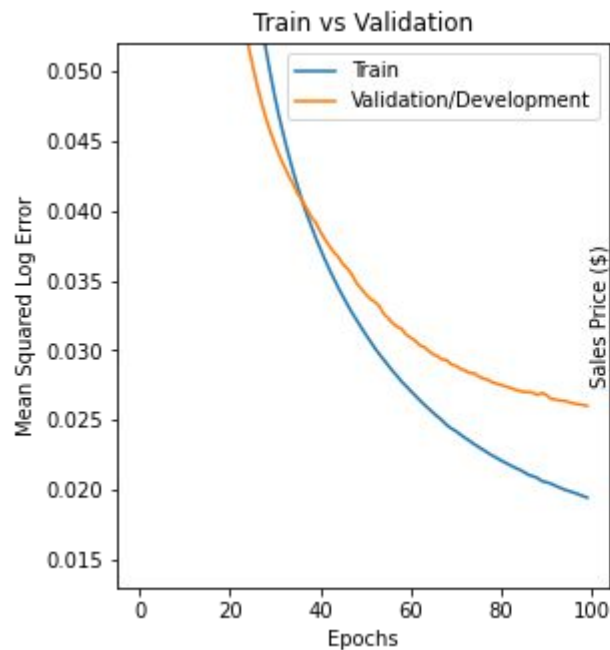
Lowest Error

	Batch	Epochs	Dropout	L1Reg	L2Reg	Train_Time	MSLE	Hidden_Layer_Nodes
0	1	100	No	No	No	89.754630	0.026004	10
1	1	100	0.5	No	No	88.080116	0.040753	10
2	1	100	No	1st Layer	No	88.080116	0.139776	10
3	1	100	No	2nd Layer	No	88.080116	0.066995	10
4	1	100	No	Both Layers	No	88.080116	0.372221	10
5	1	100	No	No	1st Layer	86.110852	0.139557	10
6	1	100	No	No	2nd Layer	85.142392	0.083193	10
7	1	100	No	No	Both Layers	85.257687	2.605657	10

Neural Network - Two Layer

Mean Squared Log Error: 0.026

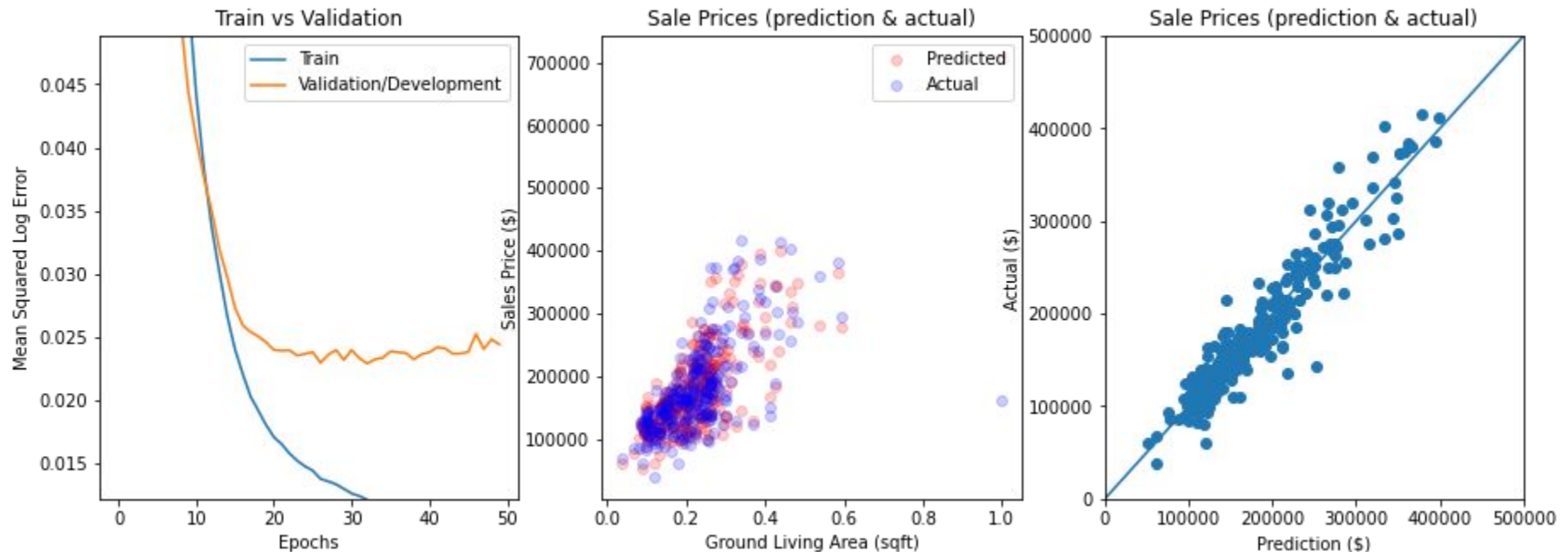
Kaggle Score: 0.1459 (Top 55%)



Neural Network - 3 & 4 layer

Mean Squared Log Error: 0.024

Kaggle Score: 0.1406 (Top 49%)





Conclusion

Linear Regression - 0.134

Extra Trees - 0.151

Neural Networks - 0.141

Final Ensemble (Weighted Average) - 0.128

Questions?

Top 27%

