Full Name: Lee Taylor, ST Number: 190211479

CSC3831 Predictive Analytics
Machine Learning House Price Prediction

First Model – Linear Regression (LR)

I implemented a grid search which attempts to find the best hyperparameters for the linear regression model on the dataset. The grid search takes in as hyper-parameters; fit_intercept, copy_x, and n_jobs to search over. The grid search returned a list of results for each set of hyperparameters.

Fit Intercept	Copy X	N jobs	MSE	R^2
TRUE	TRUE	1	0.37206	0.621966
TRUE	TRUE	2	0.37206	0.621966
TRUE	TRUE	3	0.37206	0.621966
TRUE	TRUE	4	0.37206	0.621966
TRUE	FALSE	-1	0.37206	0.621966
TRUE	FALSE	1	0.372047	0.621979
TRUE	FALSE	2	0.372047	0.621979
TRUE	FALSE	3	0.372047	0.621979
TRUE	FALSE	4	0.372047	0.621979
FALSE	TRUE	-1	0.372046	0.621981
FALSE	TRUE	1	0.372046	0.621981
FALSE	TRUE	2	0.372046	0.621981
FALSE	TRUE	3	0.372046	0.621981
FALSE	TRUE	4	0.372046	0.621981
FALSE	FALSE	-1	0.372046	0.621981
FALSE	FALSE	1	0.372046	0.621981
FALSE	FALSE	2	0.372046	0.621981
FALSE	FALSE	3	0.372046	0.621981
FALSE	FALSE	4	0.372046	0.621981

These results suggest that the choice of hyperparameters has a minimal impact on the model's performance. The MSE scores are relatively close together and the R^2 scores are all above 0.6, indicating that the model is performing relatively well, but there is still room for improvement.

Second Model – Multi-Layer Perceptron (MLP)

Another grid search is performed to find the best hyperparameters for the multi-layer perceptron model. The different hyper-parameters tested were number of units, activation function, number of layers. The grid search returned 192 results; the top 9 models are shown below.

Units Per Layer	Activation Function	Layers	Optimizer	Epochs	Batch Size	MSE	MAE
128	relu	3	adam	10	64	0.203492	0.30959
512	relu	2	adam	10	64	0.204488	0.308437
256	relu	5	adam	10	8	0.204824	0.306975
512	relu	5	adam	10	32	0.205468	0.317355
256	relu	2	adam	10	32	0.206546	0.311331
128	relu	4	adam	10	16	0.206783	0.306865
256	relu	5	adam	10	16	0.2072	0.312006
512	relu	2	adam	10	32	0.207466	0.306897
128	relu	3	adam	10	16	0.207523	0.308899

The results show that more units, and layers does not necessarily lead to better performance. I then tested different epochs and optimizers with the hyperparameters of the best model from the first grid search.

Units Per Layer	Activation Function	Layers	Optimizer	Epochs	Batch Size	MSE	MAE
128	relu	3	adam	10	64	0.203516	0.30981
128	relu	3	adam	10	64	0.214913	0.326542
128	relu	3	adam	25	64	0.215211	0.305607
128	relu	3	adamax	25	64	0.215381	0.321018
128	relu	3	adam	50	64	0.218082	0.303142
128	relu	3	adamax	10	64	0.219988	0.327635
128	relu	3	rmsprop	50	64	0.220686	0.304911
128	relu	3	rmsprop	10	64	0.23416	0.334149
128	relu	3	rmsprop	25	64	0.253742	0.348849
128	relu	3	adagrad	50	64	0.257711	0.35942
128	relu	3	adagrad	25	64	0.283529	0.379947
128	relu	3	adagrad	10	64	0.305943	0.395837

Unexpectedly, increasing the epochs and changing the optimizer did not increase performance as shown the MSE increased with the different variations.

Third Model – Random Forest Regressor (RFR)

I performed a grid search on an RFR model searching the following hyperparameters; n_estimators, max_depth, min_samples_split, and min_samples_leaf. 20 of the 599 results are shown below.

The ten best RFR model hyper-parameters and results					
N_Estimators	**Max Depth**	**Min Samples Split**	**Min Samples Leaf**	**MSE**	**R^2**
300	None	2	2	0.192447	0.804463
300	30	2	2	0.192721	0.804184
100	None	2	2	0.193086	0.803814
100	50	2	2	0.193178	0.80372
300	30	5	2	0.19325	0.803647
300	40	2	2	0.193306	0.803591
200	50	2	2	0.193644	0.803246
300	50	5	1	0.193743	0.803146
300	None	5	2	0.193814	0.803074
300	50	5	2	0.193816	0.803072
The ten worst RFR model hyper-parameters and results					
10	10	2	8	0.244633	0.751439
10	10	15	4	0.244641	0.751431
10	10	10	2	0.246019	0.750031
10	10	5	1	0.24682	0.749217
10	10	5	8	0.247924	0.748095
10	10	20	4	0.248179	0.747836
10	10	20	1	0.249479	0.746516
10	10	15	2	0.250417	0.745562
10	10	2	1	0.252138	0.743814
10	10	10	1	0.252717	0.743225

These results show that a relatively high number of n_estimators and low amount of min samples for both split and leaf are suitable hyper-parameters to fit an RFR to this dataset. I then tested the best hyper-parameters from these results with more different hyper-parameters; ccp_alpha, min_impurity_decrease, oob_score, and warm_start. From the 144 results the top 10 are shown:

N estimators	Max Depth	Min Samples split	Min Sample Leaf	Bootstrap	OOB Score	Warm Start	Min Impurity decrease	CCP alpha	MSE	R^2
300	None	2	2	TRUE	TRUE	FALSE	0	0	0.192787	0.804118
300	None	2	2	TRUE	FALSE	TRUE	0	0	0.193311	0.803585
300	None	2	2	TRUE	FALSE	FALSE	0	0	0.194445	0.802433
300	None	2	2	TRUE	TRUE	FALSE	0.1	0	0.657525	0.331917
300	None	2	2	TRUE	FALSE	FALSE	0	0.1	0.662386	0.326978
300	None	2	2	TRUE	TRUE	TRUE	0.1	0.1	0.663693	0.32565
300	None	2	2	TRUE	FALSE	TRUE	0	0.1	0.664503	0.324827
300	None	2	2	TRUE	TRUE	FALSE	0.1	0.1	0.664552	0.324777
300	None	2	2	TRUE	FALSE	TRUE	0.1	0.1	0.665614	0.323698
300	None	2	2	TRUE	TRUE	TRUE	0	0.1	0.666029	0.323276

The top three results show no improvement whereas a majority of the results show deterioration in the accuracy of the model as shown by the high MSE score jumping from ~0.19 to ~0.66.

Fourth Model – Decision Tree Regressor (DTR)
Sample text

The ten best RFR model hyper-parameters and results						
Max Depth	**Min Samples Split**	**Min Samples Leaf**	**Min Weight Fraction Leaf**	**Max Leaf Nodes**	**MSE**	**R^2**
20	20	8	0	None	0.275792	0.719779
50	20	8	0	None	0.275795	0.719777
None	20	8	0	None	0.275797	0.719775
40	20	8	0	None	0.275809	0.719762
30	20	8	0	None	0.275891	0.719679
20	15	8	0	None	0.277281	0.718267
20	10	8	0	None	0.277293	0.718254
20	5	8	0	None	0.2773	0.718247
20	2	8	0	None	0.277375	0.718171
50	2	8	0	None	0.277416	0.718129
The ten worst RFR model hyper-parameters and results						
50	20	4	0.3	10	0.634208	0.355609
50	20	4	0.3	20	0.634208	0.355609
50	20	4	0.3	30	0.634208	0.355609
50	20	4	0.3	40	0.634208	0.355609
50	20	4	0.3	50	0.634208	0.355609
50	20	8	0.3	None	0.634208	0.355609
50	20	8	0.3	10	0.634208	0.355609
50	20	8	0.3	20	0.634208	0.355609
50	20	8	0.3	30	0.634208	0.355609
50	20	8	0.3	40	0.634208	0.355609
50	20	8	0.3	50	0.634208	0.355609