Project 1

Team Members: Shu Xu (shuxu3@illinois.edu), Yan Han (yanhan4@illinois.edu), Amrit Kumar(amritk2@illinois.edu)

Section 1: Technical Details

1.1 Data Preprocessing

These data preprocessing steps are performed on both training and testing data.

1. Filled missing values in feature ‘Garage_Yr_Blt’ with 0.
2. Removed features ‘PID’, 'Condition_2', 'Heating', 'Latitude', 'Longitude', 'Low_Qual_Fin_SF', 'Misc_Feature', 'Pool_Area', 'Pool_QC', 'Roof_Matl', 'Street', 'Utilities'. These features are either highly imbalanced or not informative.
3. To reduce the influence of outliers, we conducted winsorization on the following features: 'BsmtFin_SF_2', 'Bsmt_Unf_SF', 'Enclosed_Porch', 'First_Flr_SF', 'Garage_Area', 'Gr_Liv_Area', 'Lot_Area', 'Lot_Frontage','Mas_Vnr_Area', 'Misc_Val', 'Open_Porch_SF', 'Screen_Porch', 'Second_Flr_SF', 'Three_season_porch', 'Total_Bsmt_SF', 'Wood_Deck_SF'. We performed cross validation test on the cap value and found that 97.4 percentile gives the best performance.
4. The categorical features are converted to one-hot vectors before fed into the model.
5. The response variable Sale_Price is converted to log scale.

1.2 Model Training

The two models we have chosen are linear regression with Lasso regularization and gradient boosting tree. They both use the same data preprocessed as described above. All features are standardized using sklearn’s StandardScaler before being fed into the models.

For the linear regression model, we used Lasso from scikit-learn and performed grid search to find the optimal regularization strength (alpha). We performed 10 fold cross validation on 20 different alpha values ranging from 10e-5 to 10e0.1, and chose the one with the lowest mean squared error. The best alpha is 0.026 and is used to train a final model with all the training data.

For the boosting tree model, we used XGBRegressor from xgboost and performed grid search with 10 fold cross validation to tune the number of estimators and max depth of each estimator. The criteria is also mean squared error. The optimal combination we found is max_depth = 2 and n_estimators = 420. We then used these parameters to train a final model.

Section 2: Performance Metrics

The performance of the two models are evaluated using the RMSE of log(Sale_Price) on 10 folds. Training and prediction are done on Macbook Pro, M2 chip, 8GB memory.

Fold	Lasso RMSE	Train + predict time	XGBoost RMSE	Train + predict time
1	0.1238	0.106	0.1169	0.519
2	0.1174	0.089	0.1202	0.515
3	0.1219	0.121	0.1175	0.431
4	0.1296	0.090	0.1208	0.442
5	0.1121	0.122	0.1197	0.535
6	0.1338	0.100	0.1328	0.529
7	0.1265	0.110	0.1316	0.442
8	0.1201	0.095	0.1325	0.463
9	0.1302	0.098	0.1322	0.472
10	0.1233	0.117	0.1318	0.439