Implementation for Kaggle Housing Dataset

Dean Katsaros

March 24, 2024

Initial Data Exploration

This example was first done for the Kaggle Housing Prices dataset.

- Upon reading in the dataset and checking the values, we have ~ 1500 data points each consisting of a collection of features of a house and it's selling price.

Initial Data Exploration

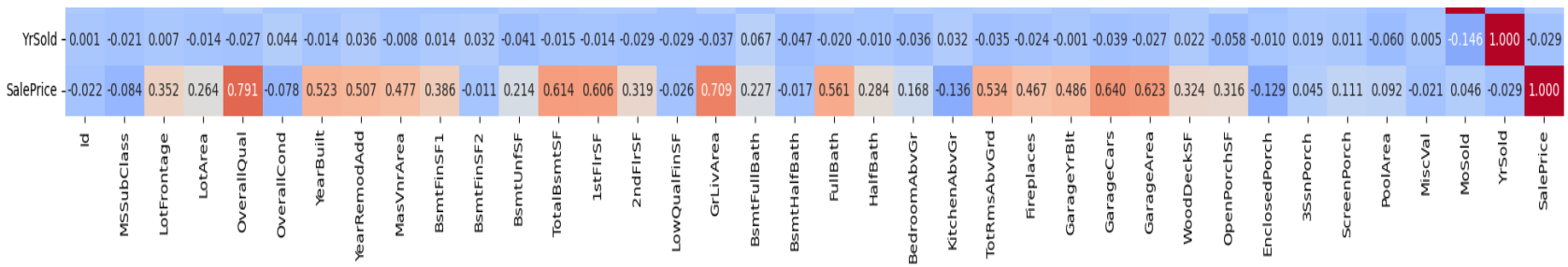
This example was first done for the Kaggle Housing Prices dataset.

- Upon reading in the dataset and checking the values, we have ~ 1500 data points each consisting of a collection of features of a house and it's selling price.
- We want to predict the value of the column 'SalePrice' for a given test set.

Initial Data Exploration

This example was first done for the Kaggle Housing Prices dataset.

- Upon reading in the dataset and checking the values, we have ~ 1500 data points each consisting of a collection of features of a house and it's selling price.
- We want to predict the value of the column 'SalePrice' for a given test set.
- Correlations:



The most correlated factors to SalePrice are OverallQual (0.791), YearBuilt (0.523), YearRemodAdd (0.507), MasVnrArea (0.477), TotalBsmtSF (0.614), 1stFlrSF (0.606), GrLivArea (0.709), FullBath (0.561), TotRmsAbvGrd (0.534), Fireplaces (0.467), GarageYrBlt (0.486), GarageCars (0.640), and GarageArea (0.623)

Cleaning the Data.

- Firstly, we see that there are many columns with a large number of null values. We will delete columns missing more than 60% of their data, which are 'PoolQC', 'MiscFeature', 'Alley', and 'Fence'. The motivation here being that these columns are underreported and likely not important to the ultimate predictions desired.

Cleaning the Data.

- Firstly, we see that there are many columns with a large number of null values. We will delete columns missing more than 60% of their data, which are 'PoolQC', 'MiscFeature', 'Alley', and 'Fence'. The motivation here being that these columns are underreported and likely not important to the ultimate predictions desired.
- For columns with less than 10% missing, we will fill in the missing values using basic imputation based on either the most frequent values or via an iterative multivariate imputation scheme in the cases of categorical / numerical data respectively. The frequentist approach should suffice here because of the high percentage of the data still present.

Cleaning the Data.

- Firstly, we see that there are many columns with a large number of null values. We will delete columns missing more than 60% of their data, which are 'PoolQC', 'MiscFeature', 'Alley', and 'Fence'. The motivation here being that these columns are underreported and likely not important to the ultimate predictions desired.
- For columns with less than 10% missing, we will fill in the missing values using basic imputation based on either the most frequent values or via an iterative multivariate imputation scheme in the cases of categorical / numerical data respectively. The frequentist approach should suffice here because of the high percentage of the data still present.
- This leaves data with more than 10% and less than 60% of their data missing. For this data, we will use a random forest classifier to *predict* the missing values from the rest of the data.

Cleaning the Data.

Importantly, these thresholds are chosen from computing the number of nulls in each column and looking at these numbers:

PoolQC	99.520548
MiscFeature	96.301370
Alley	93.767123
Fence	80.753425
MasVnrType	59.726027
FireplaceQu	47.260274
LotFrontage	17.739726
GarageYrBlt	5.547945
GarageCond	5.547945
GarageType	5.547945
GarageFinish	5.547945
GarageQual	5.547945
BsmtFinType2	2.602740
BsmtExposure	2.602740
BsmtQual	2.534247
BsmtCond	2.534247
BsmtFinType1	2.534247
MasVnrArea	0.547945
Electrical	0.068493
Id	0.000000
Functional	0.000000
Fireplaces	0.000000
KitchenQual	0.000000

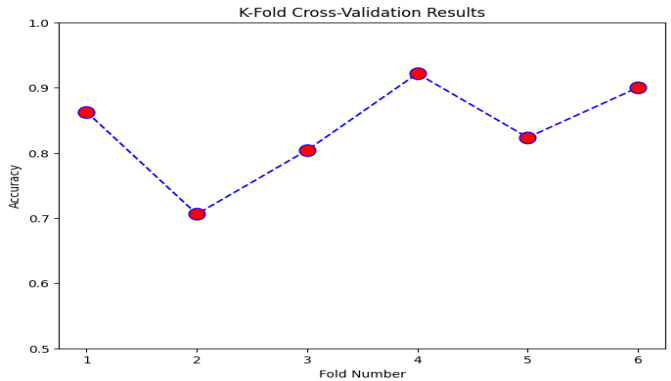
Notice how we seem to be missing either *far more than* 60%, or less than 6%, leaving us to predict only those values between $\sim 17\%$ and 60%. These thresholds would also be good choices.

Cleaning the Data.

- After imputation, we will be left to predict the 'FireplaceQu', 'LotFrontage', and 'MasVnrType' columns which are each missing between 10% and 60% of their values using machine learning, in this case a random forest classifier. The cross-validation values from a stratified k-fold cross-validator are reported.

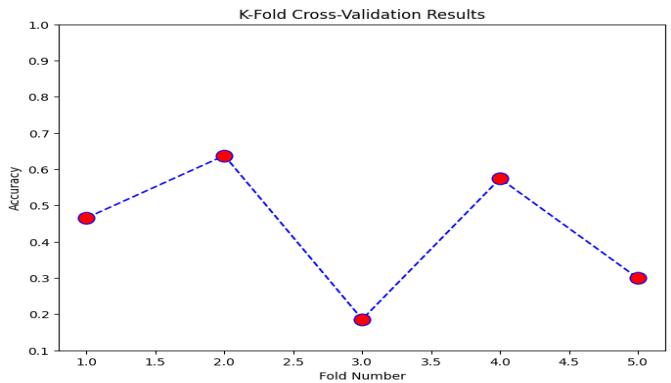
Cleaning the Data.

- Cross-validation for 'MasVnrType' predictions:



The average is ~ 0.836, meaning the model did not significantly overfit in this fitting.

- Cross-validation for 'MasVnrType' predictions:



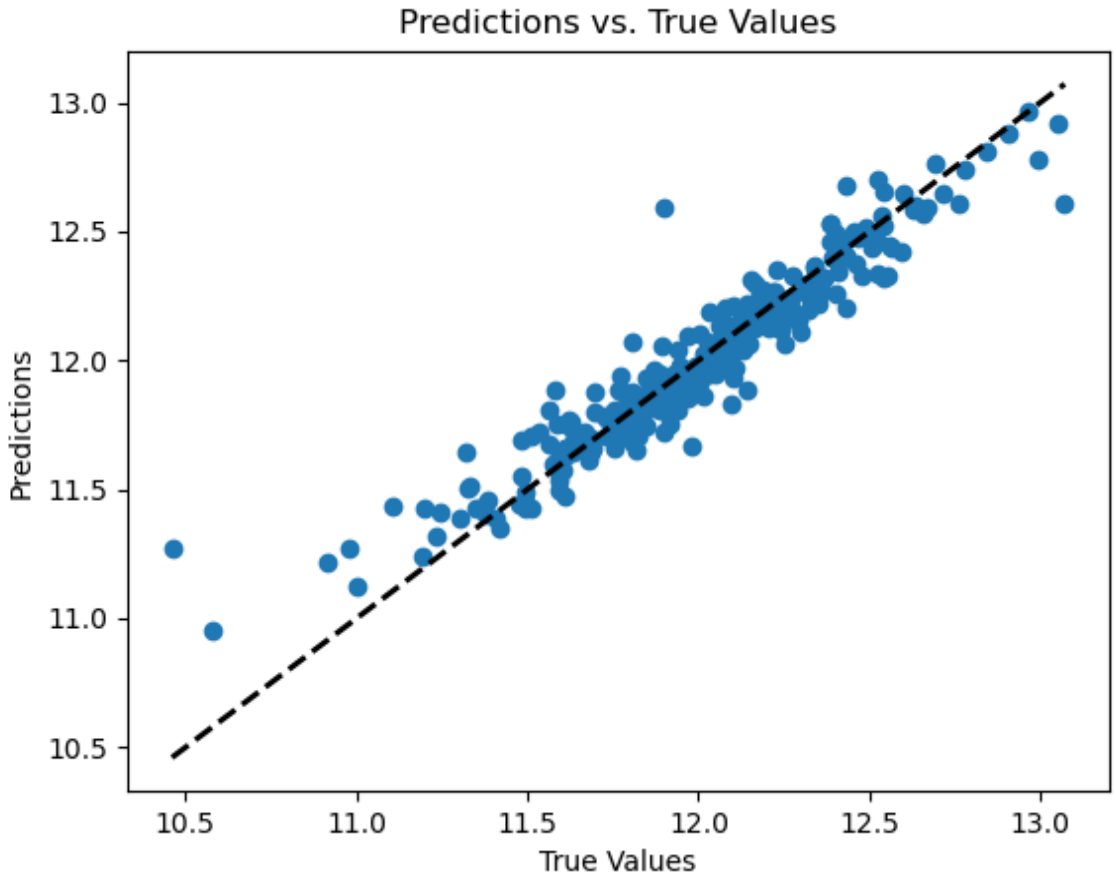
The average is ~ 0.433, meaning the model exhibited overfitting and may give erratic predictions of LotFrontage. This is a possible source of error in the final predictions!

Predictions.

- Now that all the missing data is dealt with, we will utilize the CatBoost regressor to predict the missing values. The model is trained on a standard 80:20 train:test split for 100 iterations. After about 90 iterations, the learning errors roughly converge to about 0.08. Adding iterations risks overfitting.

Predictions.

- Now that all the missing data is dealt with, we will utilize the CatBoost regressor to predict the missing values. The model is trained on a standard 80:20 train:test split for 100 iterations. After about 90 iterations, the learning errors roughly converge to about 0.08. Adding iterations risks overfitting.



Predictions.

- From here, we predict the sale prices for the houses in the test data set using the model above.

Predictions.

- From here, we predict the sale prices for the houses in the test data set using the model above.
-

Mean Squared Error:	0.0152
Mean Absolute Error:	0.0833
R^2 score:	0.9018

Appendix A: Full heatmap

Correlation Heatmap: