## Ames Housing Dataset

I decided to use Ames Housing dataset for this project because it comes with the features and target variables that will be useful in applying supervised regression model. The dataset is available in this course's second lab and the useful information from the author who prepared the dataset is available at http://jse.amstat.org/v19n3/decock.pdf

## Main Objective: Prediction

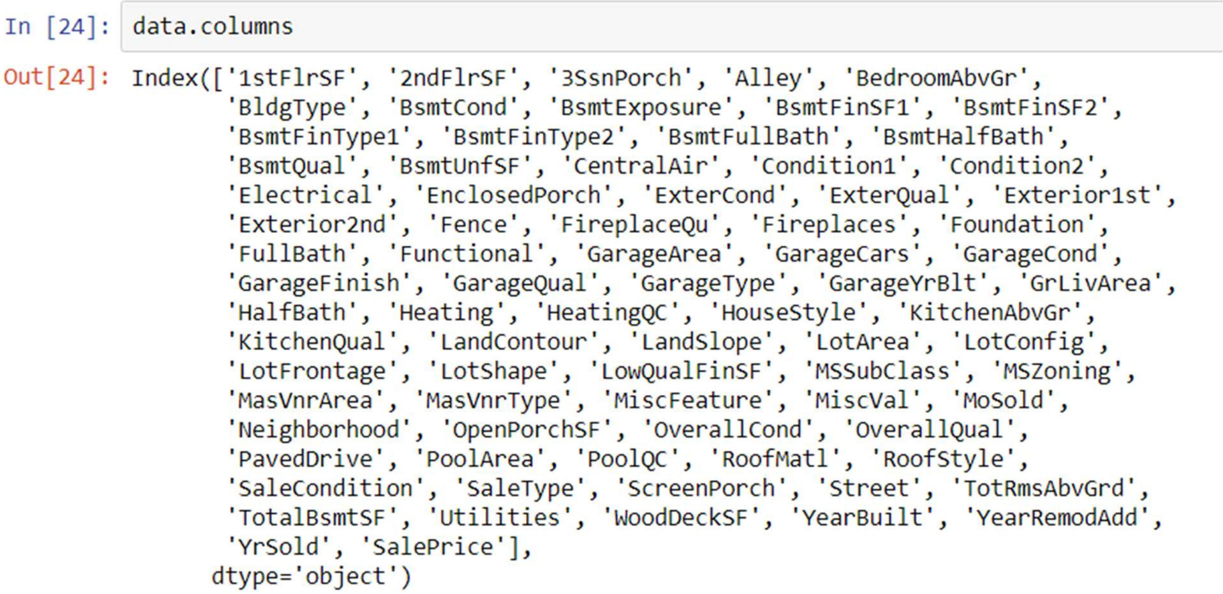
I will be using supervised regression model in predicting price of next purchase of houses based on the purchase history that is available in Ames Housing dataset. In other words, my focus would be more on predicting the value of y rather than interpreting the parameters Ω.

## Dataset Summary

Here are some attributes of the dataset:

| | |
|---|---|
| Number of rows | 1379 |
| Number of columns | 80 |
| Number of columns with object data types | 43 |
| Number of columns with float data types | 21 |
| Number of columns with int data types | 16 |

Here is the list of columns:



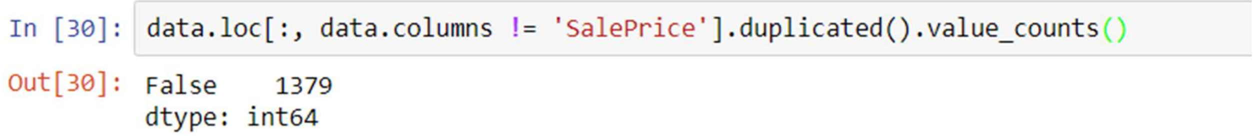
```
In [24]: data.columns

Out[24]: Index(['1stFlrSF', '2ndFlrSF', '3SsnPorch', 'Alley', 'BedroomAbvGr',
                'BldgType', 'BsmtCond', 'BsmtExposure', 'BsmtFinSF1', 'BsmtFinSF2',
                'BsmtFinType1', 'BsmtFinType2', 'BsmtFullBath', 'BsmtHalfBath',
                'BsmtQual', 'BsmtUnfSF', 'CentralAir', 'Condition1', 'Condition2',
                'Electrical', 'EnclosedPorch', 'ExterCond', 'ExterQual', 'Exterior1st',
                'Exterior2nd', 'Fence', 'FireplaceQu', 'Fireplaces', 'Foundation',
                'FullBath', 'Functional', 'GarageArea', 'GarageCars', 'GarageCond',
                'GarageFinish', 'GarageQual', 'GarageType', 'GarageYrBlt', 'GrLivArea',
                'HalfBath', 'Heating', 'HeatingQC', 'HouseStyle', 'KitchenAbvGr',
                'KitchenQual', 'LandContour', 'LandSlope', 'LotArea', 'LotConfig',
                'LotFrontage', 'LotShape', 'LowQualFinSF', 'MSSubClass', 'MSZoning',
                'MasVnrArea', 'MasVnrType', 'MiscFeature', 'MiscVal', 'MoSold',
                'Neighborhood', 'OpenPorchSF', 'OverallCond', 'OverallQual',
                'PavedDrive', 'PoolArea', 'PoolQC', 'RoofMatl', 'RoofStyle',
                'SaleCondition', 'SaleType', 'ScreenPorch', 'Street', 'TotRmsAbvGrd',
                'TotalBsmtSF', 'Utilities', 'WoodDeckSF', 'YearBuilt', 'YearRemodAdd',
                'YrSold', 'SalePrice'],
               dtype='object')
```

'SalePrice' is the target column.

## Data Exploration

**Find duplicates:**

No duplicates were found in the dataset for the all the feature columns



```
In [30]: data.loc[:, data.columns != 'SalePrice'].duplicated().value_counts()

Out[30]: False    1379
         dtype: int64
```

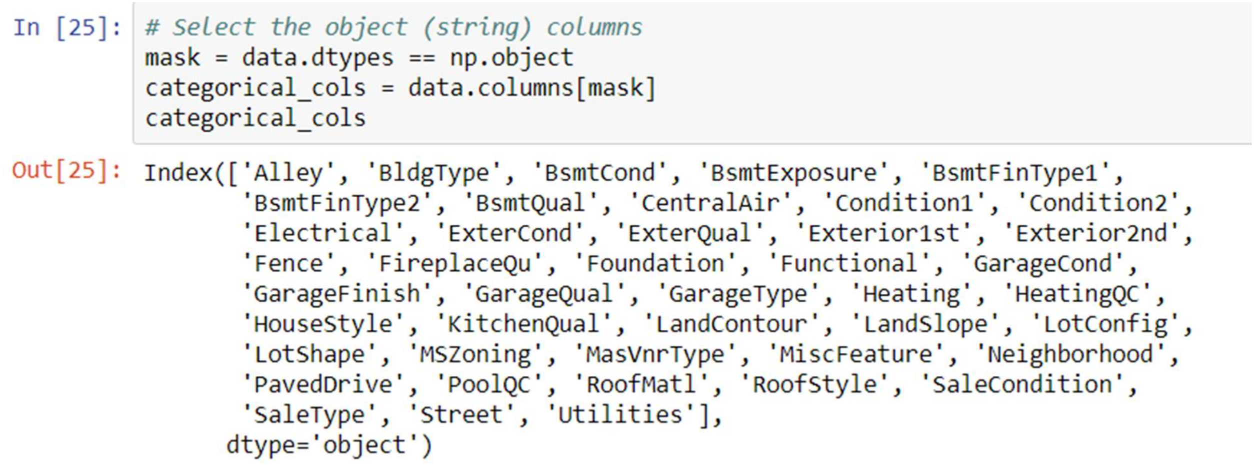
## Data Cleaning

The Ames Housing dataset is already clean and well maintained by the author and does not contain significant number of missing or duplicate data that would require data cleansing steps.

## Feature Engineering

**Categorical columns for One-Hot Encoding:**

Following categorical columns were found in the dataset



```
In [25]: # Select the object (string) columns
         mask = data.dtypes == np.object
         categorical_cols = data.columns[mask]
         categorical_cols

Out[25]: Index(['Alley', 'BldgType', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1',
                'BsmtFinType2', 'BsmtQual', 'CentralAir', 'Condition1', 'Condition2',
                'Electrical', 'ExterCond', 'ExterQual', 'Exterior1st', 'Exterior2nd',
                'Fence', 'FireplaceQu', 'Foundation', 'Functional', 'GarageCond',
                'GarageFinish', 'GarageQual', 'GarageType', 'Heating', 'HeatingQC',
                'HouseStyle', 'KitchenQual', 'LandContour', 'LandSlope', 'LotConfig',
                'LotShape', 'MSZoning', 'MasVnrType', 'MiscFeature', 'Neighborhood',
                'PavedDrive', 'PoolQC', 'RoofMatl', 'RoofStyle', 'SaleCondition',
                'SaleType', 'Street', 'Utilities'],
               dtype='object')
```

**Extra columns for one-hot encoding:**

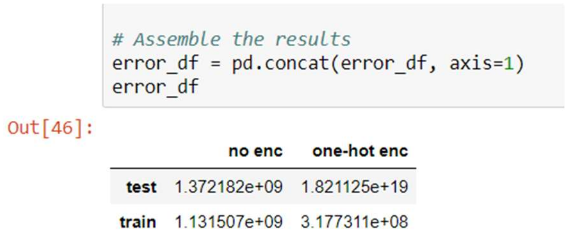
There are 215 new columns where one-hot encoding is applied after dropping the original columns. The categorical columns with only one unique value are not one-hot encoded. Also, a copy of the original dataset is made before applying one-hot encoding.

## Summary of Training

### Simple Linear Regression

The categorical columns were dropped the from the original dataset before using that for fitting to linear regression model. Before fitting the model, the original dataset and one-hot encoded dataset were split into training and testing datasets with testing dataset size of 30% for each respectively.

After fitting linear regression model to both datasets, mean squared error was calculated that showed the model was overfitting on one-hot encoded dataset. The error on one-hot encoded testing dataset was too high as compared to one-hot encoded training dataset.

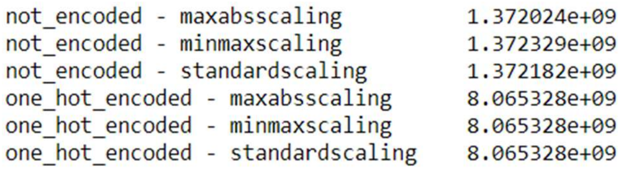


```
# Assemble the results
error_df = pd.concat(error_df, axis=1)
error_df
```

Out[46]:

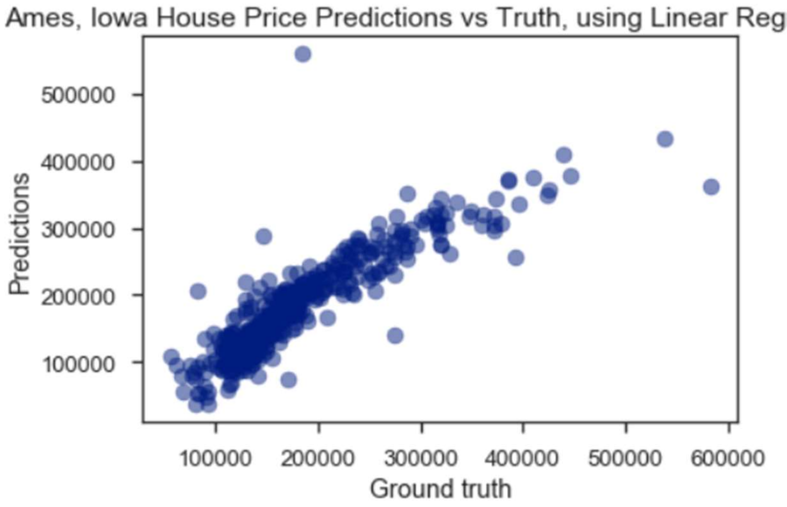
| | no enc | one-hot enc |
|---|---|---|
| test | 1.372182e+09 | 1.821125e+19 |
| train | 1.131507e+09 | 3.177311e+08 |

After scaling float data type features, the error reduced for one-hot encoded test dataset, however, it does not have any effects on original dataset.



```
not_encoded - maxabsscaling          1.372024e+09
not_encoded - minmaxscaling          1.372329e+09
not_encoded - standardscaling        1.372182e+09
one_hot_encoded - maxabsscaling      8.065328e+09
one_hot_encoded - minmaxscaling      8.065328e+09
one_hot_encoded - standardscaling    8.065328e+09
```

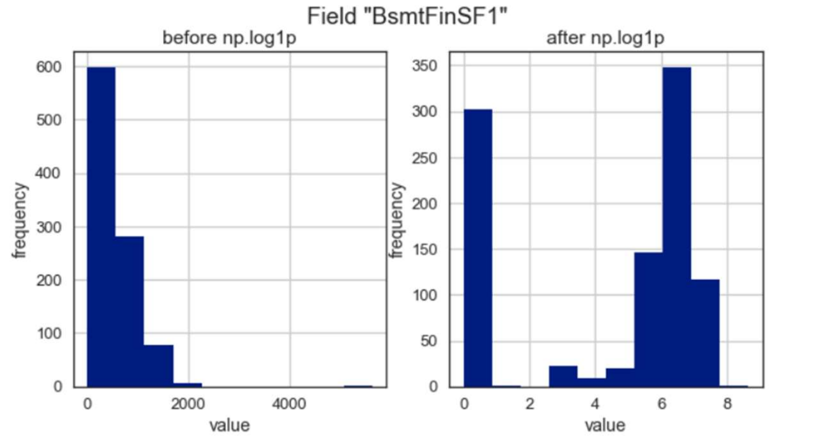
The plot for actual vs. predicted values reflects the model did predict values fairly close to actual values with only few exceptions.

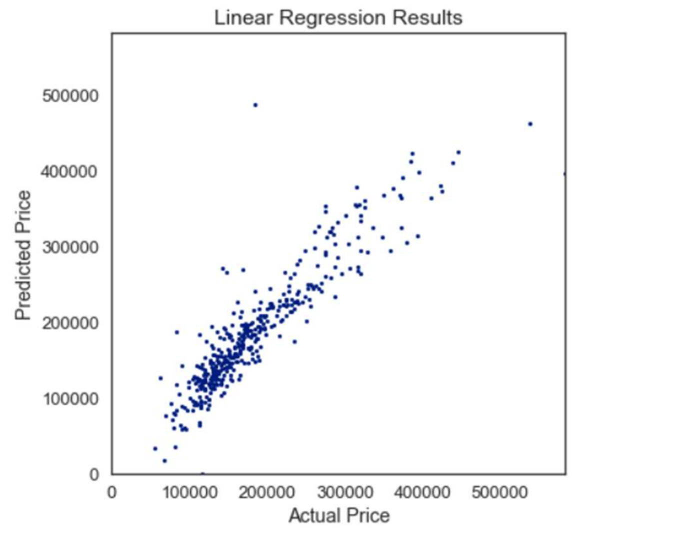


**Adding Polynomial**

By looking at the Ames Housing dataset, it appears to have some non-linear features. However, the course lab used a different scenario to show how PolynomialFeatures class is used. It is thus assumed that for the non-linear features (float data types) that have skew of more than 0.75, log transformation can be applied and that can result in predictions close to actual sale prices.

Below graphs show that after applying log transformation, the feature 'BsmtFinSF1' now has values that shows normal distribution.





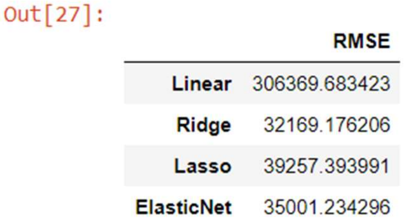
**Using Regularization**
Ridge regression is applied first in order to reduce magnitude of coefficients using L2 regularization. Used following range of alphas:
alphas = [0.005, 0.05, 0.1, 0.3, 1, 3, 5, 10, 15, 30, 80]

Lasso is applied next and it uses L1 regularization. Used following range of alphas:
[1e-5, 5e-5, 0.0001, 0.0005]

Finally, Elastic Net is applied that is hybrid of both L1 and L2 regularizations.

## Model Recommendation

Based on RSME calculated for Linear, Ridge, Lasso and Elastic Net, it is concluded that Ridge regression produces least RMSE



Out[27]:

| | RMSE |
|---|---|
| **Linear** | 306369.683423 |
| **Ridge** | 32169.176206 |
| **Lasso** | 39257.393991 |
| **ElasticNet** | 35001.234296 |

## Key Findings and Insights

As feature engineering steps were applied such one-hot encoding and later on scaling of features with float data types, the model started to overfit the dataset. After applying regularization techniques, especially Ridge, the decreased the RMSE factor and made overall model fairly good at predicting sale prices.

## Next Steps

Ames Housing dataset was used in different labs and demos. The next step would be to cover all the topics discussed in the course and create a one python notebook to provide end to end solution for predicting housing prices using linear regression. Another next step would be to perform interpretation on this dataset and research on parameters Ω and features that contributes to affecting house sale prices.