Machine Learning Engineer Nanodegree

Capstone Proposal-Kaggle House Prices: Advanced Regression Techniques

Süleyman Diker July 07st, 2018

Proposal

Domain Background

Machine learning has different learning methods, including unsupervised and supervised machine learning.Instead of going through a single method, we will proceed using the ensemble method of multi-learning algorithm.Estimating a house value involves a multifaceted learning method that accommodates many variables.The home is a sector with a very large value because it is from the basic needs of a person.And the most accurate estimation of this value constitutes very serious importance and value for this sector.

The dataset we use under this project is Ames Housing dataset.This dataset has been open to the public since 1978 and has benefited many investigations and reviews.

Problem Statement

Within this problem, we will try to estimate housing prices in the most accurate way through some algorithms using Ames Housing data set.We will use all 81 features in our data set or some of them to train our model.

We will use the following regression algorithms to increase our accuracy rates in our estimates: LinearRegression, DecisionTreeRegressor, SVR, ElasticNet, Lasso, Ridge, LassoLars, BayesianRidge, GradientBoostingRegressor, ExtraTreesRegressor, BaggingRegressor,AdaBoostRegressor, XGBRegressor.After selecting the best models, according to their performances, an ensemble generation will be implemented. Then, to determine if any improvement was made, the performance metrics of the ensemble will be calculated and compared with the ones of the benchmark model.

Datasets and Inputs

The Ames Housing dataset from Kaggle will be used in our project.This dataset is divided into two parts; the training dataset has 81 data points and 1460 data points; the test dataset has 80 data points and 1459 data points.The SalePrice feature has been removed from the test dataset. Because the feature SalePrice is a feature we need to guess already.The train dataset will be divided into a training and testing set, using ‘train_test_split’ from sklearn.cross_validation to shuffle and split the features and gross data into the training and the testing sets.

Below are the 81 features:

In [6]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')

print ("Train data shape:", train.shape)
print ("Test data shape:", test.shape)

print("================================")
print(train.dtypes)
```

```
('Train data shape:', (1460, 81))
('Test data shape:', (1459, 80))
================================
Id                   int64
MSSubClass           int64
MSZoning            object
LotFrontage        float64
LotArea              int64
Street              object
Alley               object
LotShape            object
LandContour         object
Utilities           object
LotConfig           object
LandSlope           object
Neighborhood        object
Condition1          object
Condition2          object
BldgType            object
HouseStyle          object
OverallQual          int64
OverallCond          int64
YearBuilt            int64
YearRemodAdd         int64
RoofStyle           object
RoofMatl            object
Exterior1st         object
Exterior2nd         object
MasVnrType          object
MasVnrArea         float64
ExterQual           object
ExterCond           object
Foundation          object
                    ...
BedroomAbvGr         int64
KitchenAbvGr         int64
KitchenQual         object
TotRmsAbvGrd         int64
Functional          object
Fireplaces           int64
FireplaceQu         object
GarageType          object
GarageYrBlt        float64
GarageFinish        object
GarageCars           int64
GarageArea           int64
GarageQual          object
GarageQual        object
GarageCond        object
PavedDrive        object
WoodDeckSF         int64
OpenPorchSF        int64
EnclosedPorch      int64
3SsnPorch          int64
ScreenPorch        int64
PoolArea           int64
PoolQC            object
Fence             object
MiscFeature       object
MiscVal            int64
MoSold             int64
YrSold             int64
SaleType          object
SaleCondition     object
SalePrice          int64
Length: 81, dtype: object
```

Some data in the dataset have deficiencies, either this missing data will be deleted or replaced.It would be more reasonable to delete the data of features that have too much missing value.For example ‘SaleCondition’ (the different types of sales), ‘Street’, ‘Alley’, etc.If some data is not normally distributed, especially if the mean and median vary significantly, a non-linear scaling may be applied (for example the feature ‘SalePrice’ will be logarithmically scaled). To the categorical variables some feature encoding will be applied, One Hot Encoder29 may be applied, to create a group of n dummy features30 (Yes/No (1/0) variables). Also, larger five and ten point quality scales and some discrete variables may be collapsed into fewer categories. Analyzing the correlation between the selected features and ‘SalePrice’, we will perform a Feature Selection of the most significant features.

Solution Statement

Once the data preparation processes are complete we will use these regression algorithms for the best guess: LinearRegression, DecisionTreeRegressor, SVR, ElasticNet, Lasso, Ridge, LassoLars, BayesianRidge, GradientBoostingRegressor, ExtraTreesRegressor, BaggingRegressor, AdaBoostRegressor, XGBRegressor.

To tune these models the grid search technique will be used. After selecting the best models, according to their performances, an ensemble generation will be implemented. Then, to determine if any improvements were made, the performance metrics of the ensemble will be calculated and compared with the ones of the benchmark model.

Benchmark Model

The House Prices: Advanced Regression Techniques competition uses RMSLE as evaluation metric.

In Kaggle, the RMSLE values of the teams in this competition are as follows. The values will be verified comparatively.

mean: 0.328228

min : 0.038390

25% : 0.124012

50% : 0.140710

75% : 0.177783

Evaluation Metrics

We can use the following metrics to measure the performance of the model:

Coefficient of Determination: The coefficient of determination, R2, is used to analyze how differences in one variable can be explained by a difference in a second variable. For example, when a person gets pregnant has a direct relation to when they give birth.More specifically, R-squared gives you the percentage variation in y explained by x-variables. The range is 0 to 1 (i.e. 0% to 100% of the variation in y can be explained by the x-variables.The coefficient of determination, R2, is similar to the correlation coefficient, R. The correlation coefficient formula will tell you how strong of a linear relationship there is between two variables. R Squared is the square of the correlation coefficient, r (hence the term r squared).

Mean Squared Error: The mean squared error tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the "errors") and squaring them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It's called the mean squared error as you're finding the average of a set of errors.

Root Mean Square Error: Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

Project Design

This project will use these libraries: NumPy, MatplotLib, Seaborn, Pandas, Sklearn, XGBoost, SciPy. Our data set will be divided into training data set and test data set using train_test_split method in sklearn.cross_validation.We will analyze the discovery data to better understand our data.(Exploratory Data Analysis) Entries with missing data will be remove or replace with some arbitrary value. Some variables may be dropped from the dataset,for example 'SaleCondition', 'Street', 'Alley', etc.On the numeric variables some normalization may be used.If some data is not normally distributed, especially if the mean and median vary significantly, a non-linear scaling may be applied (for example the feature 'SalePrice' will be logarithmically scaled). To the categorical variables some feature encoding will be applied, One Hot Encoder may be applied, to create a group of n dummy features (Yes/No (1/0) variables). Also, larger five and ten point quality scales and some discrete variables may be collapsed into fewer categories. After analyzing the correlation between the features and 'SalePrice", we will perform a Feature Selection of the most significant features. Then Model Selection will be performed, the regression techniques used may be: LinearRegression, DecisionTreeRegressor, SVR, ElasticNet, Lasso, etc. To tune the models, the grid search technique will be used. The results from these regressions will be analyzed and the best models will be chosen. Once the best models are selected, according to their performances, an ensemble generation will be implemented. The performance metrics calculated and compared with the ones of the benchmark model, to determine if the target was reached.