1098 lines (1098 sloc) | 38.9 KB

Feature Selection Advanced House Price Prediction

The main aim of this project is to predict the house price based on various features which we will discuss as we go ahead

Dataset to downloaded from the below link

https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data)

In [1]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline

## for feature slection

from sklearn.linear_model import Lasso
from sklearn.feature_selection import SelectFromModel

# to visualise al the columns in the dataframe
pd.pandas.set_option('display.max_columns', None)
```

In [21]:
```
dataset=pd.read_csv('X_train.csv')
```

In [22]:
```
dataset.head()
```

Out[22]:

	Id	SalePrice	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Cor
0	1	12.247694	0.235294	0.75	0.418208	0.366344	1.0	1.0	0.000000	0.333333	1.0	0.00	0.0	0.636364	0.4
1	2	12.109011	0.000000	0.75	0.495064	0.391317	1.0	1.0	0.000000	0.333333	1.0	0.50	0.0	0.500000	0.2
2	3	12.317167	0.235294	0.75	0.434909	0.422359	1.0	1.0	0.333333	0.333333	1.0	0.00	0.0	0.636364	0.4
3	4	11.849398	0.294118	0.75	0.388581	0.390295	1.0	1.0	0.333333	0.333333	1.0	0.25	0.0	0.727273	0.4
4	5	12.429216	0.235294	0.75	0.513123	0.468761	1.0	1.0	0.333333	0.333333	1.0	0.50	0.0	1.000000	0.4

In [23]:
```
## Capture the dependent feature
y_train=dataset[['SalePrice']]
```

In [25]:
```
## drop dependent feature from dataset
```

```
X_train=dataset.drop(['Id','SalePrice'],axis=1)
```

In [27]:
```
### Apply Feature Selection
# first, I specify the Lasso Regression model, and I
# select a suitable alpha (equivalent of penalty).
# The bigger the alpha the less features that will be selected.

# Then I use the selectFromModel object from sklearn, which
# will select the features which coefficients are non-zero

feature_sel_model = SelectFromModel(Lasso(alpha=0.005, random_state=0)) # remember to set the seed, the random state in this function
feature_sel_model.fit(X_train, y_train)
```

Out[27]:
```
SelectFromModel(estimator=Lasso(alpha=0.005, copy_X=True, fit_intercept=True, max_iter=1000,
   normalize=False, positive=False, precompute=False, random_state=0,
   selection='cyclic', tol=0.0001, warm_start=False),
        max_features=None, norm_order=1, prefit=False, threshold=None)
```

In [28]:
```
feature_sel_model.get_support()
```

Out[28]:
```
array([ True,  True, False, False, False, False, False, False, False,
       False, False,  True, False, False, False, False,  True, False,
       False,  True,  True, False, False, False, False, False, False,
       False, False,  True, False,  True, False, False, False, False,
       False, False, False,  True,  True, False,  True, False, False,
        True,  True, False, False, False, False, False,  True, False,
       False,  True,  True,  True, False,  True,  True, False, False,
       False,  True, False, False, False, False, False, False, False,
       False, False, False, False, False, False,  True, False, False,
       False])
```

In [33]:
```
# let's print the number of total and selected features

# this is how we can make a list of the selected features
selected_feat = X_train.columns[(feature_sel_model.get_support())]

# let's print some stats
print('total features: {}'.format((X_train.shape[1])))
print('selected features: {}'.format(len(selected_feat)))
print('features with coefficients shrank to zero: {}'.format(
    np.sum(sel_.estimator_.coef_ == 0)))
```

```
total features: 82
selected features: 21
features with coefficients shrank to zero: 61
```

In [30]:
```
selected_feat
```

Out[30]:
```
Index(['MSSubClass', 'MSZoning', 'Neighborhood', 'OverallQual', 'YearRemodAdd',
       'RoofStyle', 'BsmtQual', 'BsmtExposure', 'HeatingQC', 'CentralAir',
       '1stFlrSF', 'GrLivArea', 'BsmtFullBath', 'KitchenQual', 'Fireplaces',
       'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageCars', 'PavedDrive',
       'SaleCondition'],
      dtype='object')
```

In [35]:
```
X_train=X_train[selected_feat]
```

In [36]:
```
X_train.head()
```

Out[36]:

	MSSubClass	MSZoning	Neighborhood	OverallQual	YearRemodAdd	RoofStyle	BsmtQual	BsmtExposure	HeatingQC	CentralAir	1stFlrSF	GrLivArea
0	0.235294	0.75	0.636364	0.666667	0.098361	0.0	0.75	0.25	1.00	1.0	0.356155	0.577712
1	0.000000	0.75	0.500000	0.555556	0.524590	0.0	0.75	1.00	1.00	1.0	0.503056	0.470245
2	0.235294	0.75	0.636364	0.666667	0.114754	0.0	0.75	0.50	1.00	1.0	0.383441	0.593095
3	0.294118	0.75	0.727273	0.666667	0.606557	0.0	0.50	0.25	0.75	1.0	0.399941	0.579157
4	0.235294	0.75	1.000000	0.777778	0.147541	0.0	0.75	0.75	1.00	1.0	0.466237	0.666523

In [ ]: