Machine Learning Model Building Pipeline: Feature Selection

In [1]:
```
# importation des librairies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.linear_model import Lasso
from sklearn.feature_selection import SelectFromModel
pd.pandas.set_option('display.max_columns', None)
```

In [2]:
```
# chargement des datasets
X_train = pd.read_csv('Data/xtrain.csv')
X_test = pd.read_csv('Data/xtest.csv')

X_train.head()
```

Out[2]:

	id_mutation	id_parcelle	id_bien	date_mutation	adresse_nom_voie	nom_commune	valeur_fonciere	nature_mutation	code_
0	2017-1381514	95018000AV0057	95018000AV0057-95	2017-05-16	RUE DE ST QUENTIN	Argenteuil	12.354493	0.666667	
1	2017-131542	132098460A0288	132098460A0288-13	2017-04-07	RUE ANTOINE FORTUNE MARION	Marseille 9e Arrondissement	13.075272	0.666667	
2	2017-1162525	83038000AB0022	83038000AB0022-83	2017-05-22	SAINTE ANNE	Châteaudouble	11.652687	0.666667	
3	2019-173403	44109000NY0325	44109000NY0325-44	2019-03-29	RUE FELIX LEMOINE	Nantes	9.510445	0.666667	
4	2017-242501	22011000AB0237	22011000AB0237-22	2017-04-27	LE BOURG	Boqueho	8.006368	0.666667	

In [3]:
```
# création de la target
y_train = X_train['valeur_fonciere']
y_test = X_test['valeur_fonciere']

# suppression des variables non sélectionnées
X_train.drop(['id_mutation', 'id_parcelle', 'id_bien', 'date_mutation', 'adresse_nom_voie', 'nom_commune', 'valeur_fonciere', 'longitude', 'latitude', 'code_type_local_na', 'surface_reelle_bati_na', 'nombre_pieces_principales_na', 'surface_terrain_na', 'longitude_na', 'latitude_na'], axis=1, inplace=True)
X_test.drop(['id_mutation', 'id_parcelle', 'id_bien', 'date_mutation', 'adresse_nom_voie', 'nom_commune', 'valeur_fonciere', 'longitude', 'latitude', 'code_type_local_na', 'surface_reelle_bati_na', 'nombre_pieces_principales_na', 'surface_terrain_na', 'longitude_na', 'latitude_na'], axis=1, inplace=True)
```

Feature Selection

Let's go ahead and select a subset of the most predictive features. There is an element of randomness in the Lasso regression, so remember to set the seed.

In [4]:
```
sel_ = SelectFromModel(Lasso(alpha=0.005, random_state=123))
sel_.fit(X_train, y_train)
```

Out[4]:
```
SelectFromModel(estimator=Lasso(alpha=0.005, random_state=123))
```

In [5]:
```
# permet de visualiser les variables selectionnées
sel_.get_support()
```

Out[5]:
```
array([ True,  True, False,  True,  True,  True, False,  True])
```

In [6]:
```
# liste des colonnes selectionnées
selected_feat = X_train.columns[(sel_.get_support())]

# 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: 8
selected features: 6
features with coefficients shrank to zero: 2
```

In [7]:
```
selected_feat
```

Out[7]:
```
Index(['nature_mutation', 'code_departement', 'code_type_local', 'type_local',
       'surface_reelle_bati', 'surface_terrain'],
      dtype='object')
```

Identify the selected variables

In [8]:
```
# autre manière de trouver les colonnes à sélectionner
selected_feats = X_train.columns[(sel_.estimator_.coef_ != 0).ravel().tolist()]
selected_feats
```

Out[8]:
```
Index(['nature_mutation', 'code_departement', 'code_type_local', 'type_local',
       'surface_reelle_bati', 'surface_terrain'],
      dtype='object')
```

In [9]:
```
# sauvegarde de la feature selection
pd.Series(selected_feats).to_csv('selected_features.csv', index=False)
```

That is all for this notebook. In the next video, we will go ahead and build the final model using the selected features. See you then!