```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

housing = pd.DataFrame(pd.read_csv(r'C:\Users\kjbua\OneDrive\Documents\School\Machine Learning\Housing.csv'))
housing.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished

[43]:
```
m = len(housing)
m
```

[43]: 545

[44]:
```
housing.shape
```

[44]: (545, 13)

[45]:
```
varlist = ['area', 'bedrooms', 'bathrooms', 'stories', 'parking']

def binary_map(x):
    return x.map({'yes': 1, "no": 0})

housing[varlist] = housing[varlist].apply(binary_map)

housing.head()
```

[45]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	NaN	NaN	NaN	NaN	yes	no	no	no	yes	NaN	yes	furnished
1	12250000	NaN	NaN	NaN	NaN	yes	no	no	no	yes	NaN	no	furnished
2	12250000	NaN	NaN	NaN	NaN	yes	no	yes	no	no	NaN	yes	semi-furnished
3	12215000	NaN	NaN	NaN	NaN	yes	no	yes	no	yes	NaN	yes	furnished
4	11410000	NaN	NaN	NaN	NaN	yes	yes	yes	no	yes	NaN	no	furnished

[47]:
```
from sklearn.model_selection import train_test_split


np.random.seed(0)
df_train, df_test = train_test_split(housing, train_size = 0.7, test_size = 0.3)
df_train.shape
```

[47]: (381, 13)

[48]:
```
df_test.shape
```

[48]: (164, 13)

[49]:
```
num_vars = ['area', 'bedrooms', 'bathrooms', 'stories', 'parking']

df_Newtrain = df_train[num_vars]
df_Newtest = df_test[num_vars]
df_Newtrain.head()
```

[49]:

	area	bedrooms	bathrooms	stories	parking
454	NaN	NaN	NaN	NaN	NaN
392	NaN	NaN	NaN	NaN	NaN
231	NaN	NaN	NaN	NaN	NaN
271	NaN	NaN	NaN	NaN	NaN
250	NaN	NaN	NaN	NaN	NaN

[54]:
```
df_Newtrain.shape
```

[54]: (381, 5)

[56]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler

scaler = MinMaxScaler()

df_Newtrain[num_vars] = scaler.fit_transform(df_Newtrain[num_vars])

df_Newtrain.head(5)
```

[56]:

	area	bedrooms	bathrooms	stories	parking
454	NaN	NaN	NaN	NaN	NaN
392	NaN	NaN	NaN	NaN	NaN
231	NaN	NaN	NaN	NaN	NaN
271	NaN	NaN	NaN	NaN	NaN
250	NaN	NaN	NaN	NaN	NaN

[53]:
```
y_Normtrain = df_Newtrain.pop('price')

X_Normtrain = df_Newtrain.copy()
```

[ ]:
```
X_Normtrain.head()
```

[ ]:
```
y_Normtrain.head()
```

[ ]:
```
Y_Normtrain.head()
```

[ ]:
```
Y = y_Normtrain.values
Y
```