**Functions**

1. Reading and creating train and test data

import pandas as pd

data = pd.read\_csv('AmesHousing.txt', delimiter="\t")

train = data[0:1460]

test = data[1460:]

print(train.info())

target = 'SalePrice'

1. Displaying the data

import matplotlib.pyplot as plt

# For prettier plots.

import seaborn

fig = plt.figure(figsize=(7,15))

ax1 = fig.add\_subplot(3, 1, 1)

ax2 = fig.add\_subplot(3, 1, 2)

ax3 = fig.add\_subplot(3, 1, 3)

train.plot(x="Garage Area", y="SalePrice", ax=ax1, kind="scatter")

train.plot(x="Gr Liv Area", y="SalePrice", ax=ax2, kind="scatter")

train.plot(x="Overall Cond", y="SalePrice", ax=ax3, kind="scatter")

plt.show()

## Finding correlations in variables

train[['Garage Area', 'Gr Liv Area', 'Overall Cond', 'SalePrice']].corr()

1. Using scikit learn

from sklearn.linear\_model import LinearRegression

lr = LinearRegression()

lr.fit(train[['Gr Liv Area']], train['SalePrice'])

print(lr.coef\_)

print(lr.intercept\_)

a0 = lr.intercept\_

a1 = lr.coef\_

1. Checking the accuracy with train and test model

import numpy as np

lr = LinearRegression()

lr.fit(train[['Gr Liv Area']], train['SalePrice'])

from sklearn.metrics import mean\_squared\_error

train\_predictions = lr.predict(train[['Gr Liv Area']])

test\_predictions = lr.predict(test[['Gr Liv Area']])

train\_mse = mean\_squared\_error(train\_predictions, train['SalePrice'])

test\_mse = mean\_squared\_error(test\_predictions, test['SalePrice'])

train\_rmse = np.sqrt(train\_mse)

test\_rmse = np.sqrt(test\_mse)

print(train\_rmse)

print(test\_rmse)

1. Adding more features

cols = ['Overall Cond', 'Gr Liv Area']

lr.fit(train[cols], train['SalePrice'])

train\_predictions = lr.predict(train[cols])

test\_predictions = lr.predict(test[cols])

train\_rmse\_2 = np.sqrt(mean\_squared\_error(train\_predictions, train['SalePrice']))

test\_rmse\_2 = np.sqrt(mean\_squared\_error(test\_predictions, test['SalePrice']))

print(train\_rmse\_2)

print(test\_rmse\_2)