

```

import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt

train_data = pd.read_csv('train.csv')
test_data = pd.read_csv('test.csv')
sample_submission = pd.read_csv('sample_submission.csv')
data_description = open('data_description.txt', 'r').read()

print(train_data.head())
print(train_data.isnull().sum())

X = train_data[['GrLivArea', 'BedroomAbvGr', 'FullBath', 'HalfBath']]
y = train_data['SalePrice']

categorical_cols = ['MSZoning', 'Street', 'Alley', 'LotShape',
'LandContour', 'Utilities', 'LotConfig', 'LandSlope', 'Neighborhood',
'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'RoofStyle',
'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'ExterQual',
'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure',
'BsmtFinType1', 'BsmtFinType2', 'Heating', 'HeatingQC', 'CentralAir',
'Electrical', 'KitchenQual', 'Functional', 'FireplaceQu', 'GarageType',
'GarageFinish', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature',
'SaleType', 'SaleCondition']
for col in categorical_cols:
    test_data[col] = test_data[col].fillna(train_data[col].mode()[0])

le = LabelEncoder()
for col in categorical_cols:
    train_data[col] = le.fit_transform(train_data[col])
    test_data[col] = le.transform(test_data[col])

X = pd.concat([X, train_data[categorical_cols]], axis=1)
X_test = pd.concat([test_data[['GrLivArea', 'BedroomAbvGr', 'FullBath',
'HalfBath']], test_data[categorical_cols]], axis=1)
X_test = X_test[X.columns]
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,
random_state=42)

lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

print("Coefficients:")
print("Square Footage:", lr_model.coef_[0])
print("Number of Bedrooms:", lr_model.coef_[1])
print("Number of Full Bathrooms:", lr_model.coef_[2])
print("Number of Half Bathrooms:", lr_model.coef_[3])
print("Intercept:", lr_model.intercept_)

def predict_price(square_footage, bedrooms, full_bathrooms,
half_bathrooms):
    return lr_model.coef_[0] * square_footage + lr_model.coef_[1] *
bedrooms + lr_model.coef_[2] * full_bathrooms + lr_model.coef_[3] *
half_bathrooms + lr_model.intercept_

```

```
print("Predicted Price for a 2000 sqft house with 3 bedrooms, 2 full  
bathrooms, and 1 half bathroom:")  
print(predict_price(2000, 3, 2, 1))  
  
y_pred_test = lr_model.predict(X_test)  
  
submission = pd.DataFrame({'Id': test_data['Id'], 'SalePrice':  
y_pred_test})  
submission.to_csv('submission.csv', index=False)  
  
plt.scatter(X_train['GrLivArea'], y_train)  
plt.xlabel('Square Footage')  
plt.ylabel('Sale Price')  
plt.title('Linear Regression')  
plt.show()
```