**Problem Definition**

This project aims to predict house sale prices based on attributes like square footage, number of bedrooms, location etc. using a Seattle house dataset. It explores the data, builds regression models with LinearRegression, XGBoost, GradientBoostingRegressor and compares their performance.

Accurate home price prediction can help home buyers and sellers make better decisions.

**Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

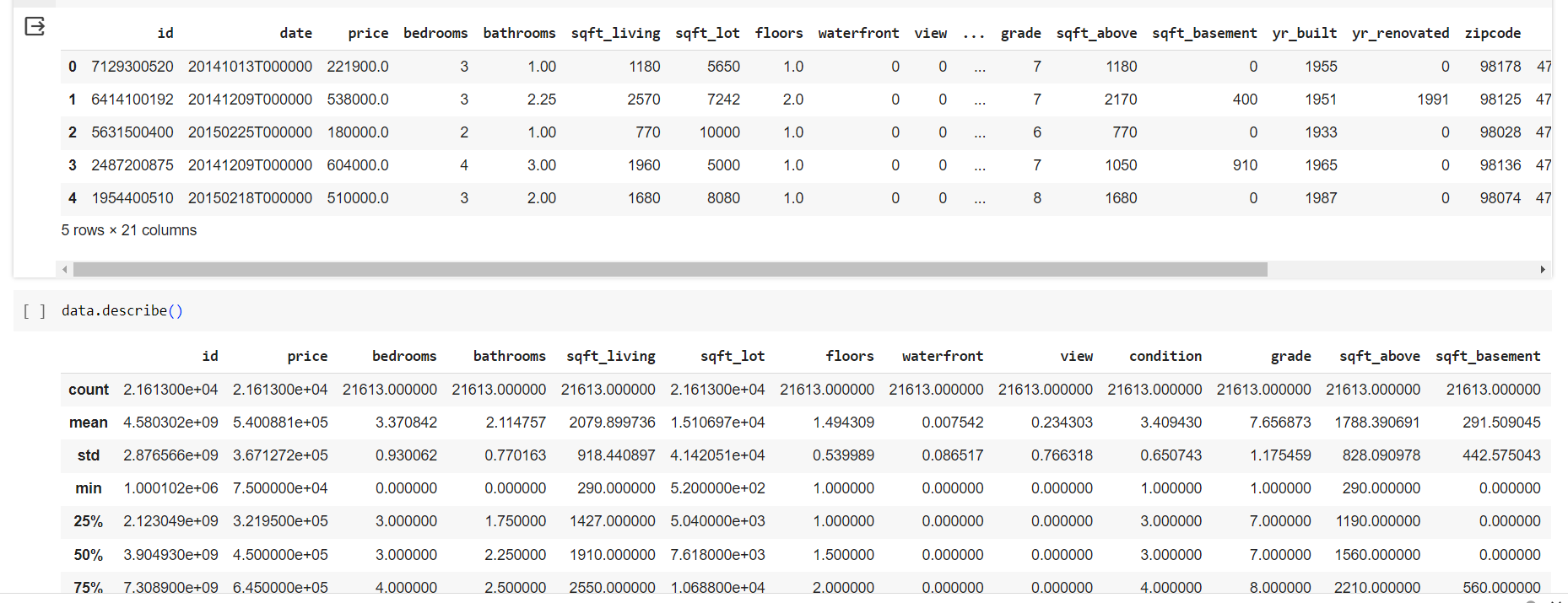
import seaborn as sns

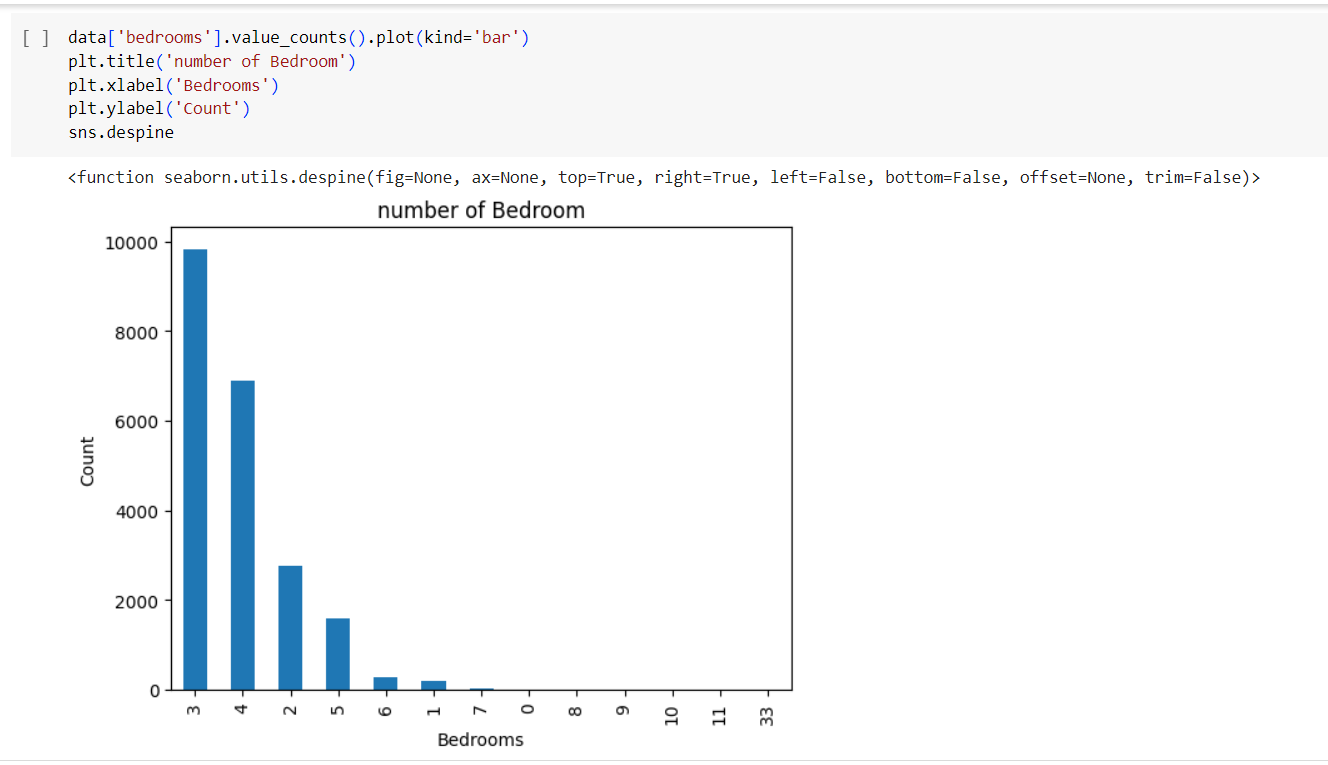
import mpl\_toolkits

%matplotlib inline

data = pd.read\_csv("/content/kc\_house\_data.csv")

data.head()

****

****

plt.figure(figsize=(10,10))

sns.jointplot(x=data.lat.values, y=data.long.values, size=10)

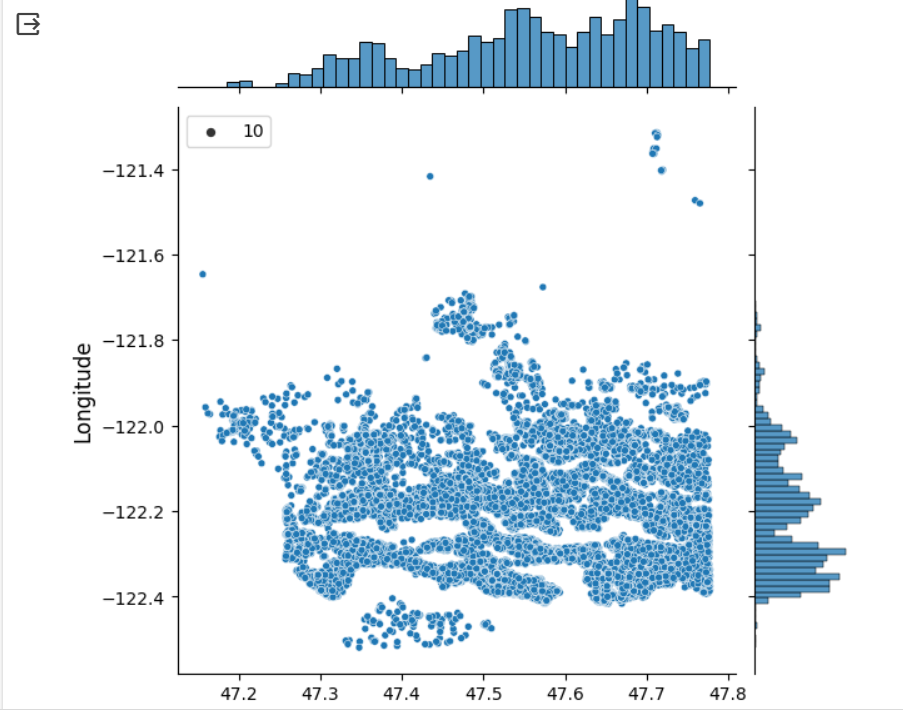
plt.ylabel('Longitude', fontsize=12)

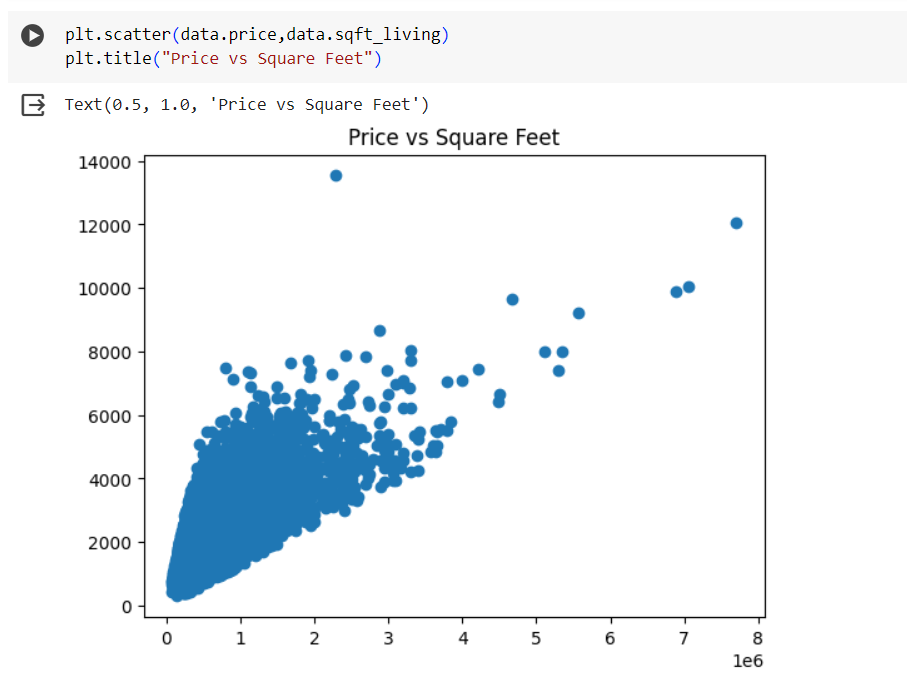
plt.xlabel('Latitude', fontsize=12)

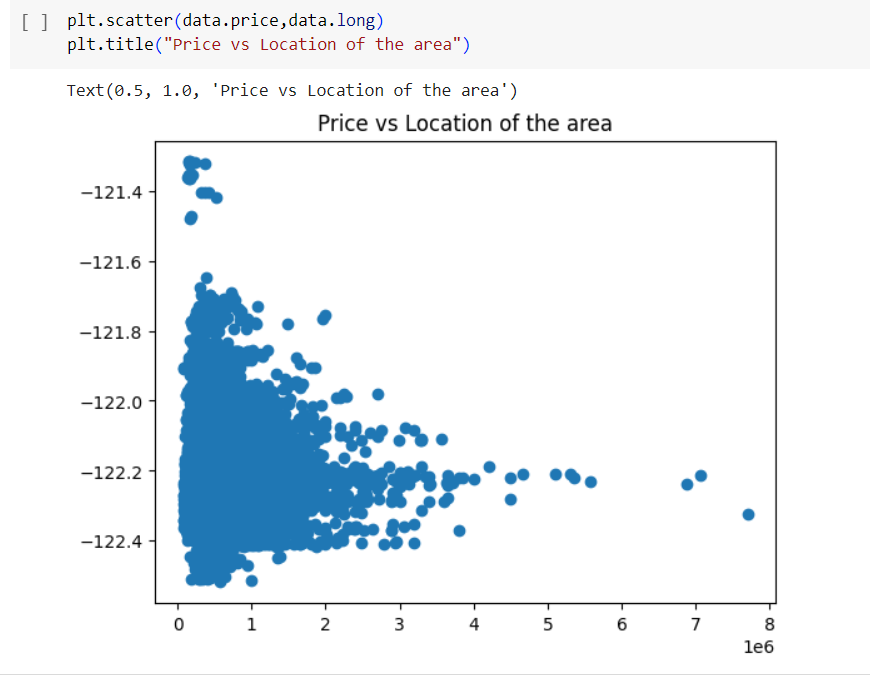
plt.show()

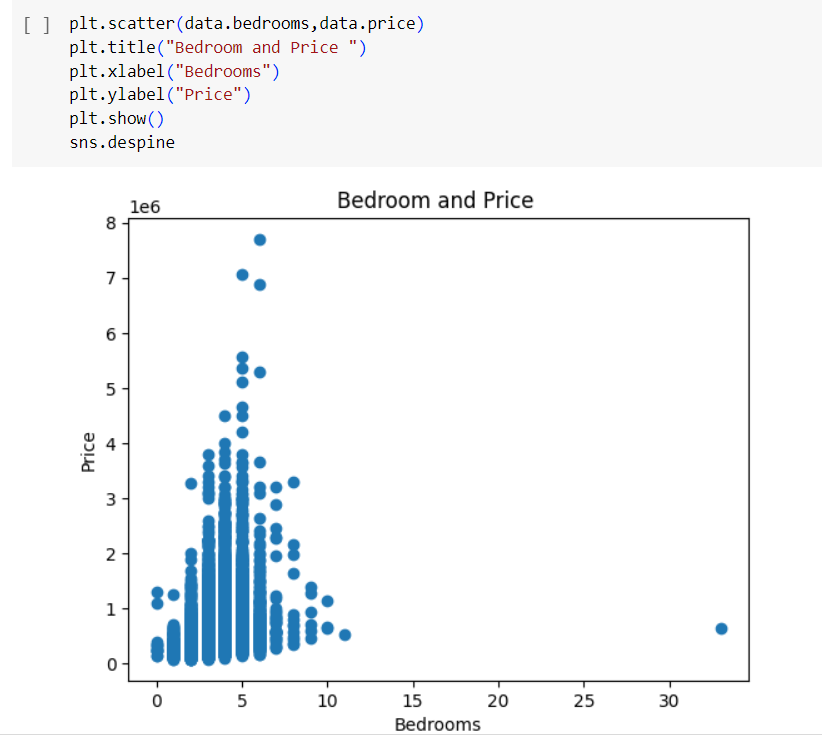
plt1 = plt.figure()

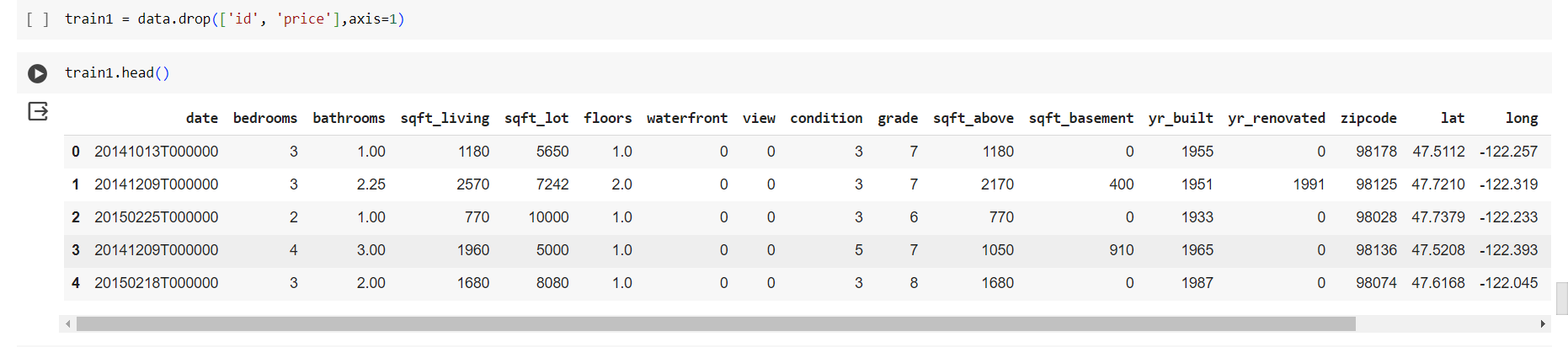
sns.despine

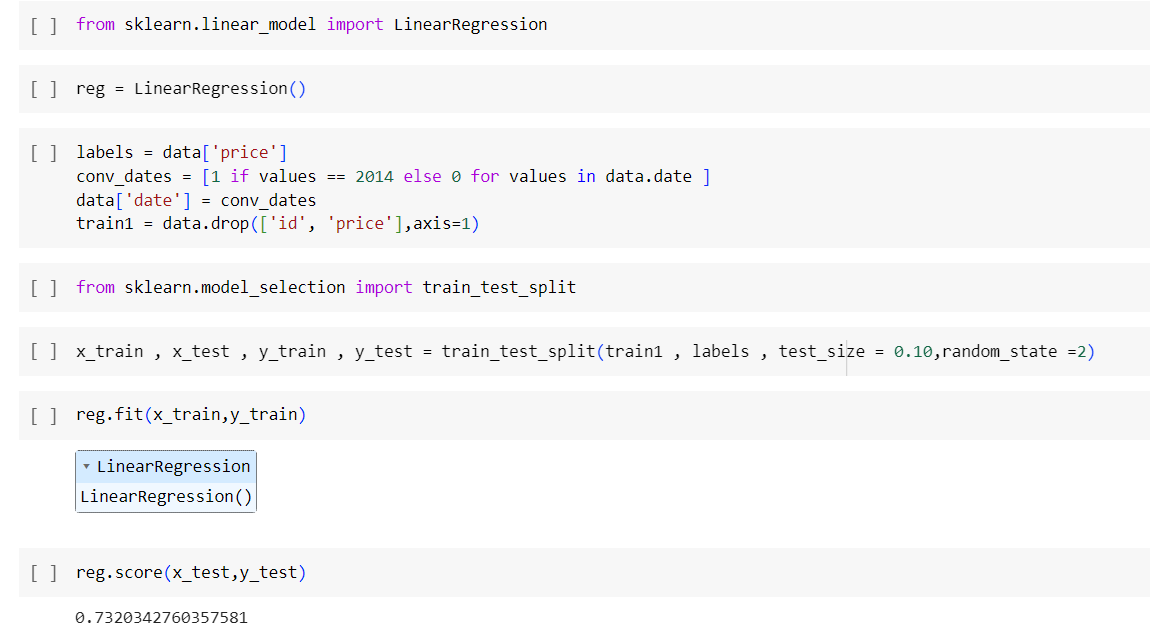
****

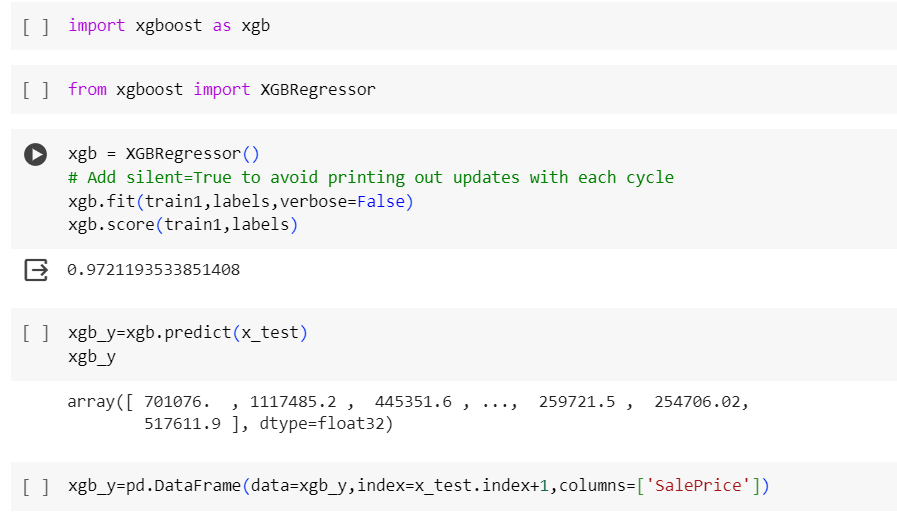
****

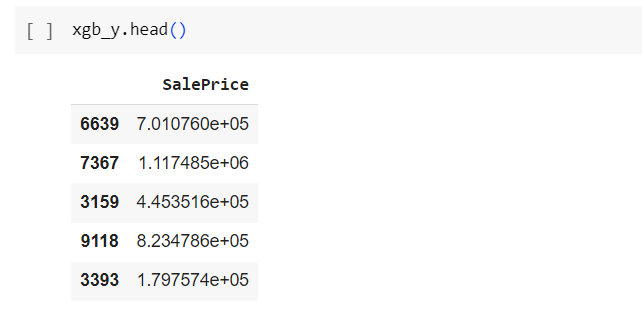
****

****

****

****

****

****

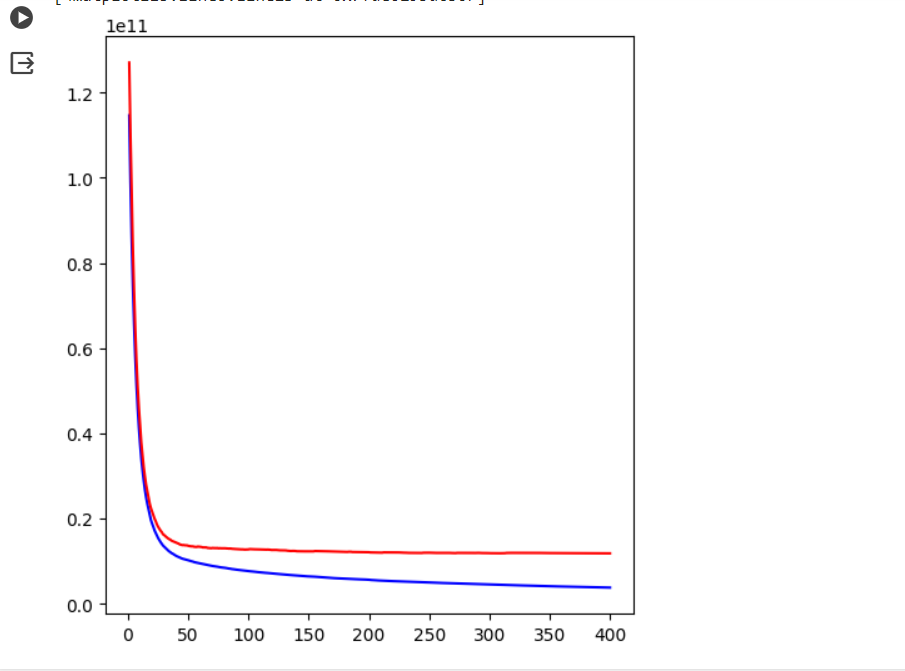
****

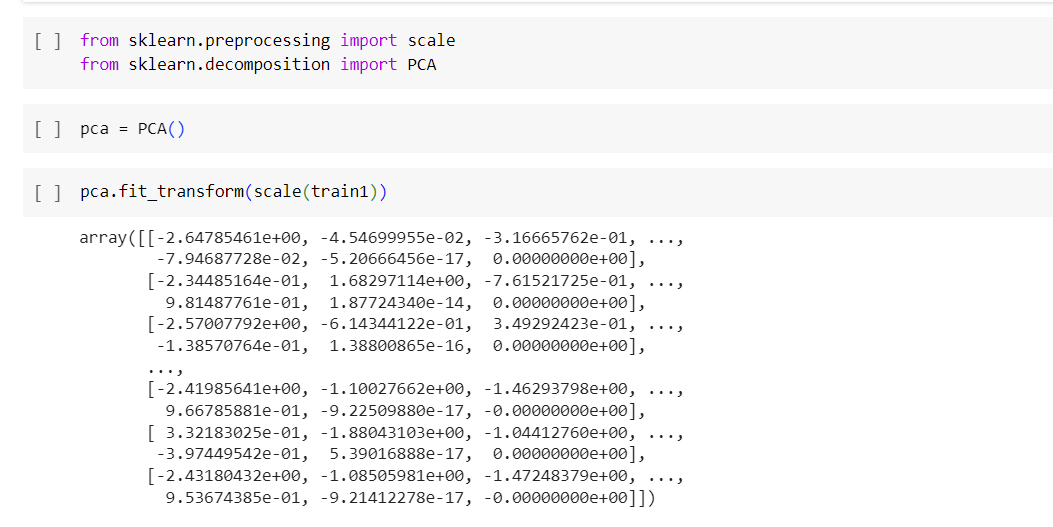
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

plt.plot(testsc,clf.train\_score\_,'b-',label= 'Set dev train')

plt.plot(testsc,t\_sc,'r-',label = 'set dev test')

****

****

**Code Description**

**Imports and Data Loading**

* Imports core data science libraries like numpy, pandas, matplotlib, seaborn, sklearn
* Loads kc\_house\_data.csv dataset

The dataset is named kc\_house\_data.csv. It is loaded from a csv file containing data on house sales.

Each row in the dataset likely represents a single house sale, with the following columns:

id: Unique ID for each home sale

price: The sale price of the home (target variable to predict)

bedrooms: Number of bedrooms

bathrooms: Number of bathrooms

sqft\_living: Square footage of living spaces

sqft\_lot: Square footage of the lot/land

floors: Number of floors

waterfront: Binary indicator for whether the home was waterfront (0 or 1)

view: Type of view from home

condition: Condition of the home

grade: Overall grade given to the housing unit

sqft\_above: Square footage of house apart from basement

sqft\_basement: Square footage of the basement

yr\_built: Year when built

yr\_renovated: Year when renovated

zipcode: Zip code of the home

lat: Latitude coordinate

long: Longitude coordinate

date: Date when the home sale occurred

* Prints first 5 records and statistical description

**Data Exploration**

* Visualizations to understand data distributions:
  + Histogram for number of bedrooms
  + Scatter plots between price, sqft\_living, location coordinates
  + Scatter plots between price and key attributes like bedrooms, waterfront
* Splits features & labels into X & y variables

**Model Building**

* Builds a baseline LinearRegression model
* Tries XGBRegressor, tunes hyperparams through cross-validation
* Builds a GradientBoostingRegressor with tuned hyperparameters
* Compares model test scores

**Preprocessing & Dimensionality Reduction**

* One-hot encodes categorical variable for date
* Scales features and applies PCA identifying main components in data

**Challenges**

Hyperparameter Tuning: Though some hyperparameter tuning is done - the space searched seems limited. Fine tuning models by exhaustively exploring hyperparameters can yield gains but takes patience and compute resources.

Overfitting: Fitting complex nonlinear models like GradientBoosting, XGBoost have a tendency to overfit. Lack of validation data or regularization like early stopping increases the risk.

Class Imbalance: Certain target classes if significantly lower in proportion (outliers) can skew model performance. Strategies like upsampling minority classes may be needed.