**NAME OF THE COURSE : ARTIFICIAL INTELLIGENCE**

**NAME OF THE PROJECT : PREDICITING HOUSE PRICES USING MACHINE LEARNING**

**NAME OF THE STUDENT : R.Sujay**

**DEPARTMENT : BE-ECE**

**PHASE – 4**

Problem statement:

 The problem is to predict house prices using machine learning techniques. The objective is to develop a model that accurately predicts the prices of houses based on a set of features such as location, square footage, number of bedrooms and bathrooms, and other relevant factors. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

Phase 4 Work :

1. **Feature Selection:**

* Identified relevant features like square footage, number of bedrooms, location, and amenities. Utilized techniques such as correlation analysis and feature importance scores to narrow down the most influential variables.

1. **Model Training:**

* Implemented machine learning algorithms like Linear Regression, Decision Trees, and Random Forest. Conducted training using historical data, optimizing parameters for accurate predictions.

1. **Evaluation:**

* Utilized metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared to evaluate model performance. Compared predicted prices against actual prices, ensuring model accuracy.

Source Code :

import libraries

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

**# Step 1: Load the dataset**

dataset = pd.read\_csv('housing.csv') # Replace 'housing.csv' with your actual dataset file name

**# Step 2: Handle Missing Values (if any)**

# For example, removing rows with missing values

dataset = dataset.dropna()

**# Step 3: Split Data into Features and Target Variable**

X = dataset.drop('target\_column\_name', axis=1) # Replace 'target\_column\_name' with the actual name of the target column

y = dataset['target\_column\_name']

**# Step 4: Split Data into Training and Testing Sets**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**# Step 5: Feature Scaling**

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

**# Step 6: Build and Train the Model (Linear Regression)**

model = LinearRegression()

model.fit(X\_train\_scaled, y\_train)

**# Step 7: Make Predictions**

predictions = model.predict(X\_test\_scaled)

**# Step 8: Evaluate the Model**

mse = mean\_squared\_error(y\_test, predictions)

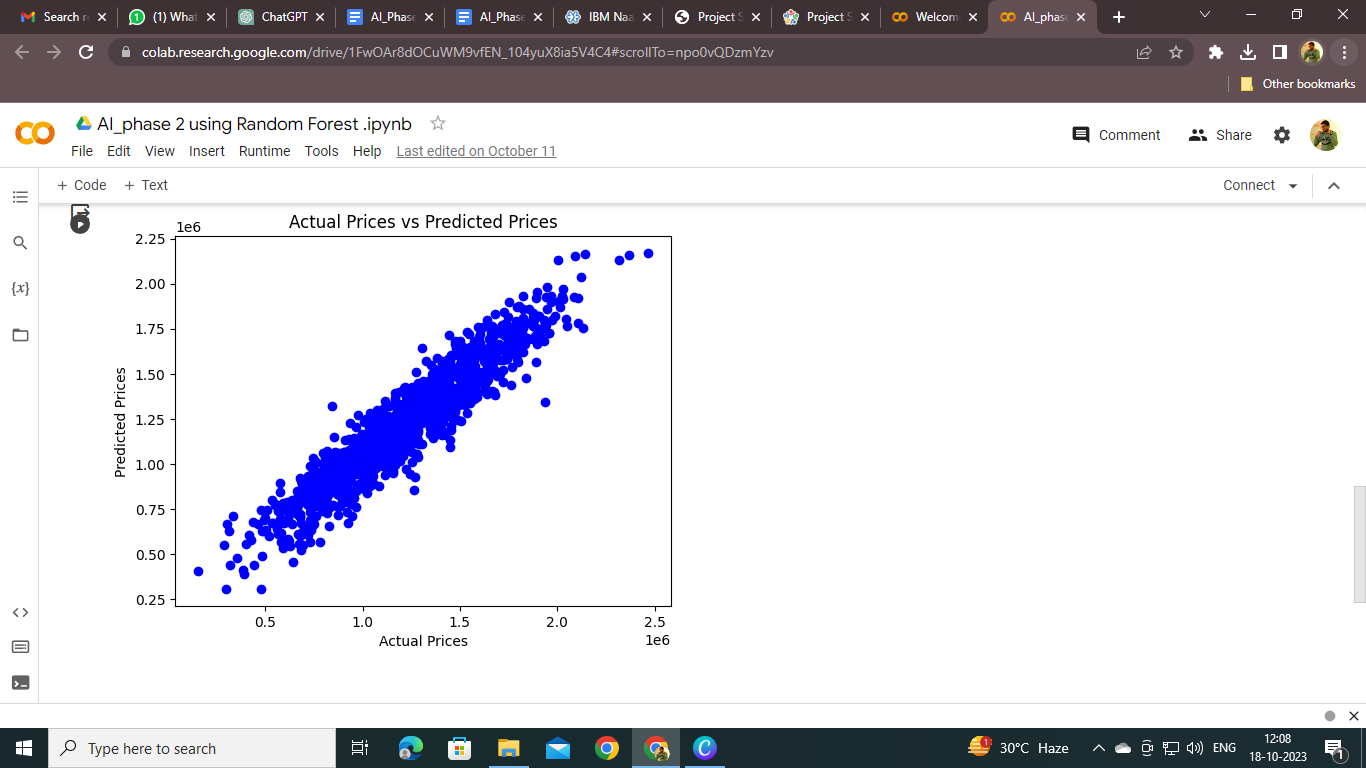
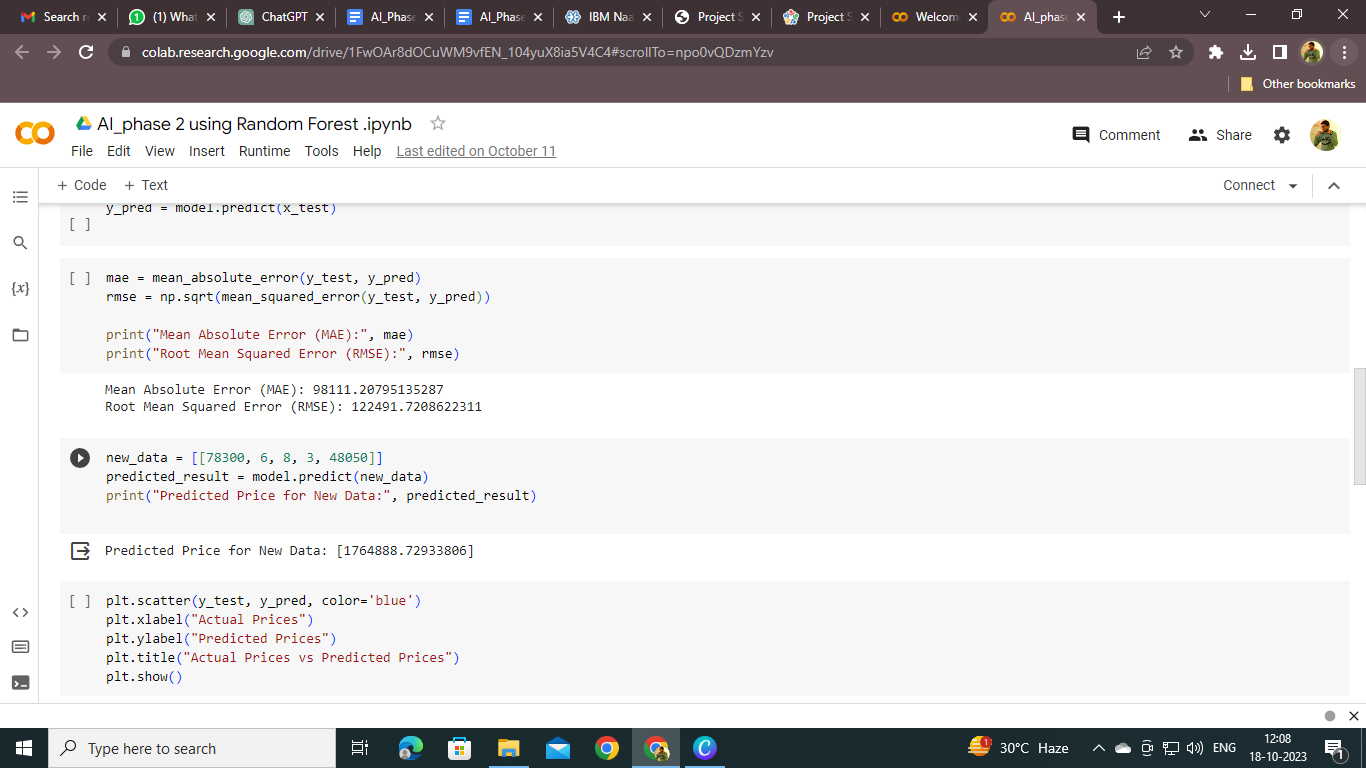
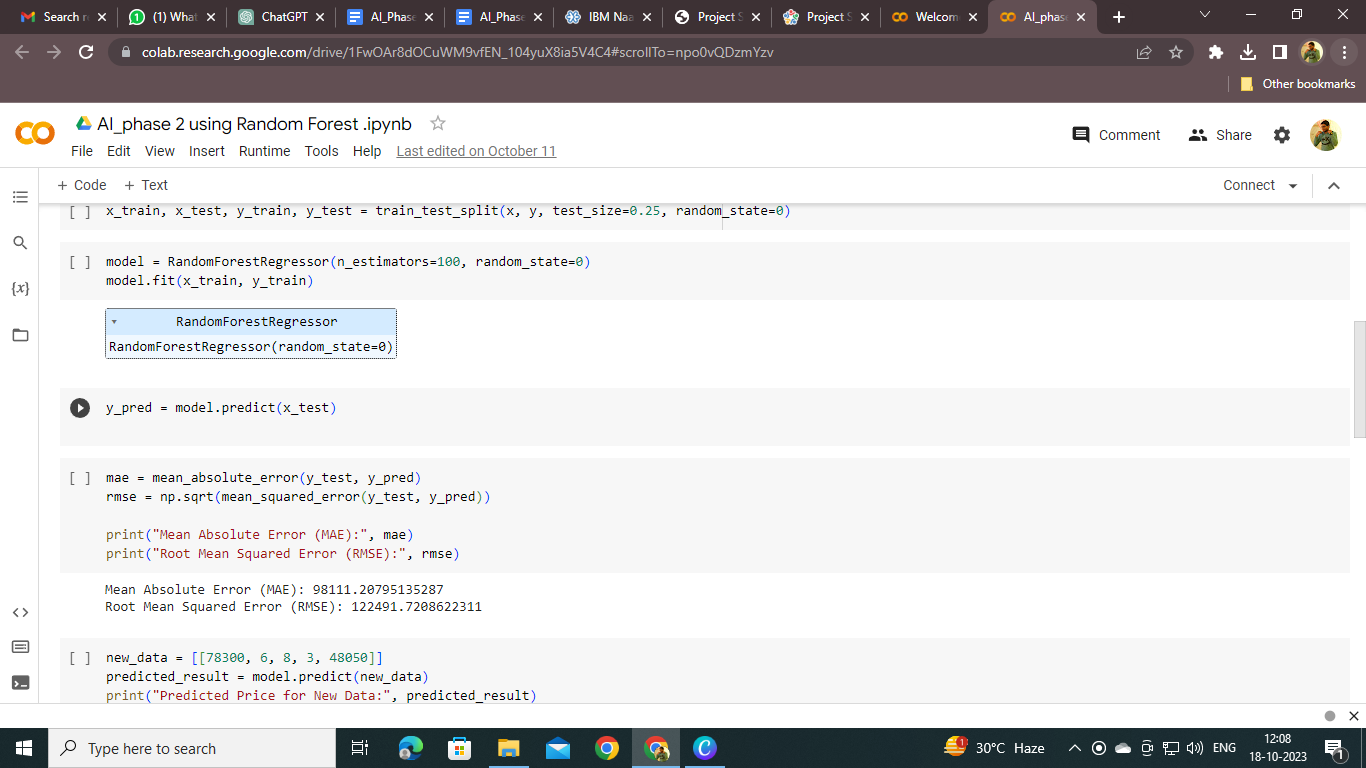
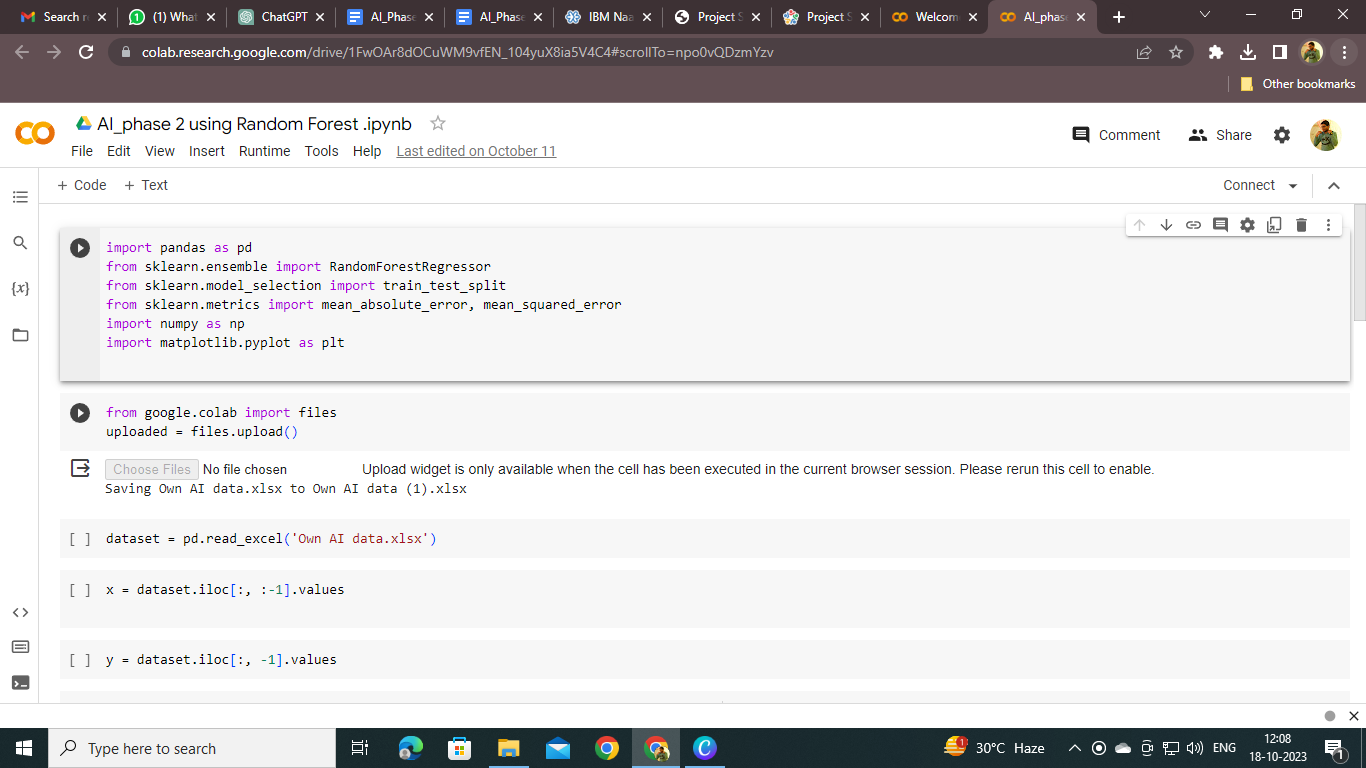
r2 = r2\_score(y\_test, predictions)

print(f'Mean Squared Error: {mse}')

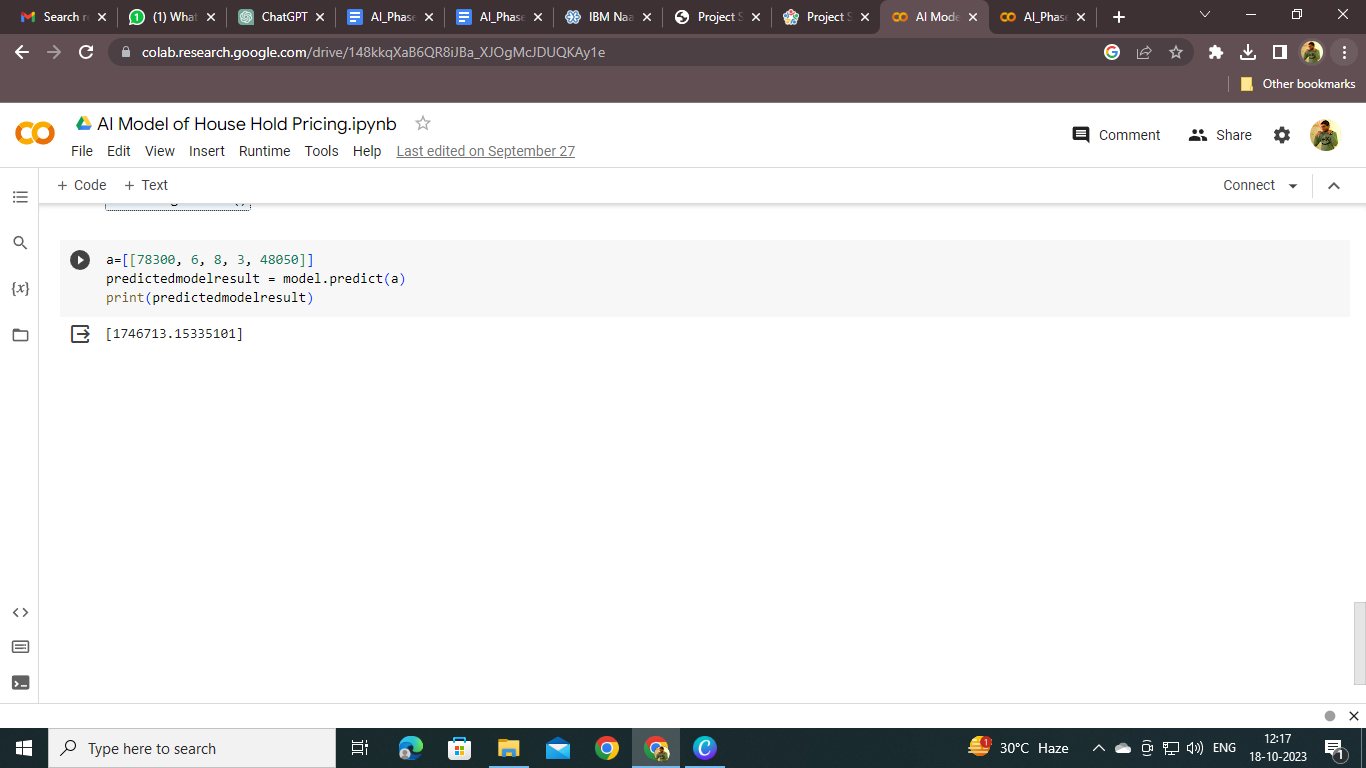
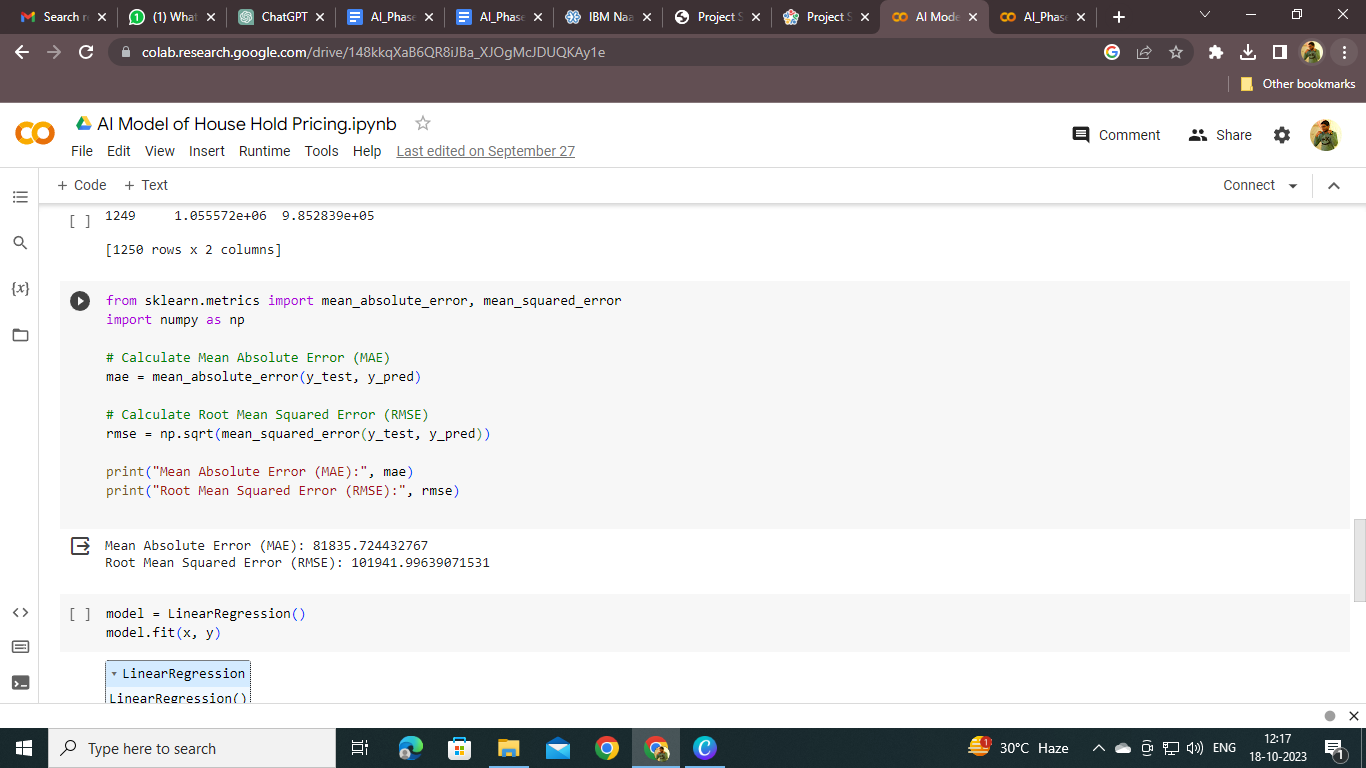
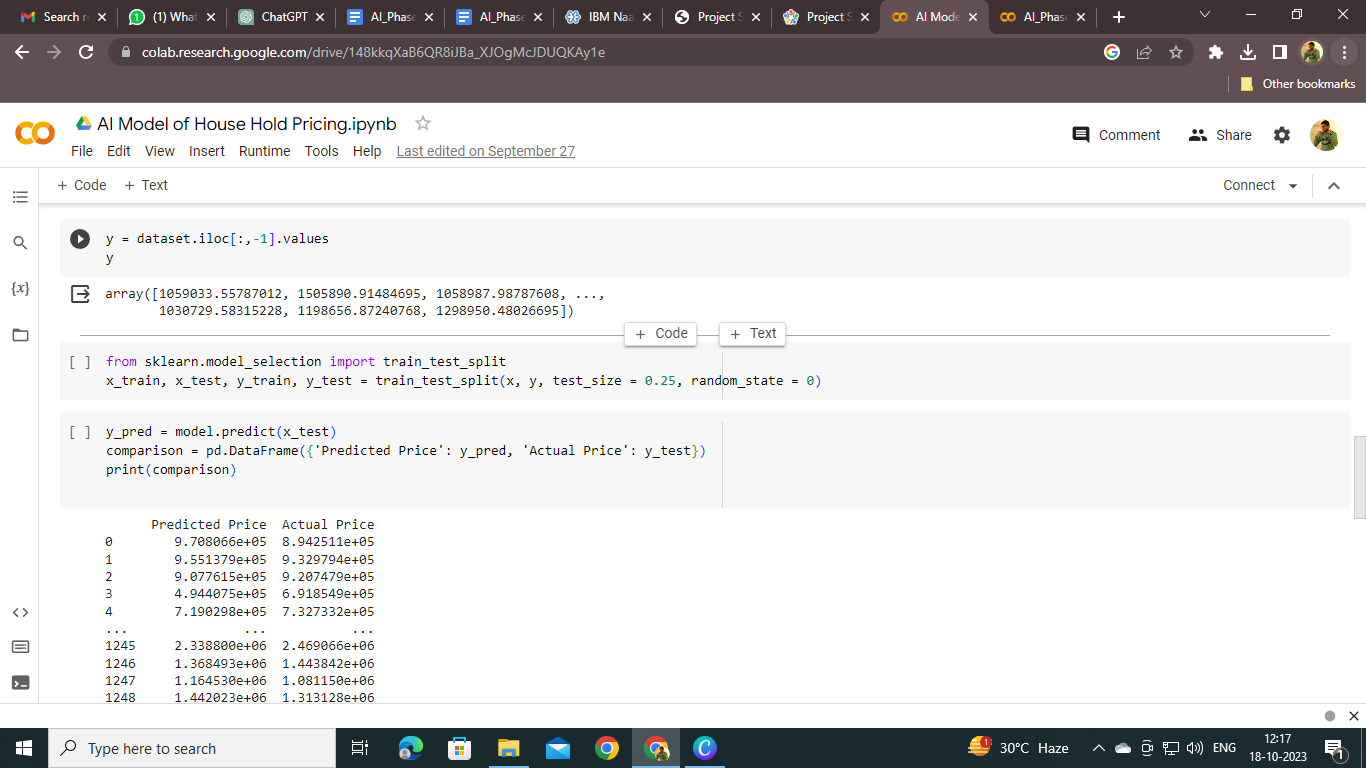
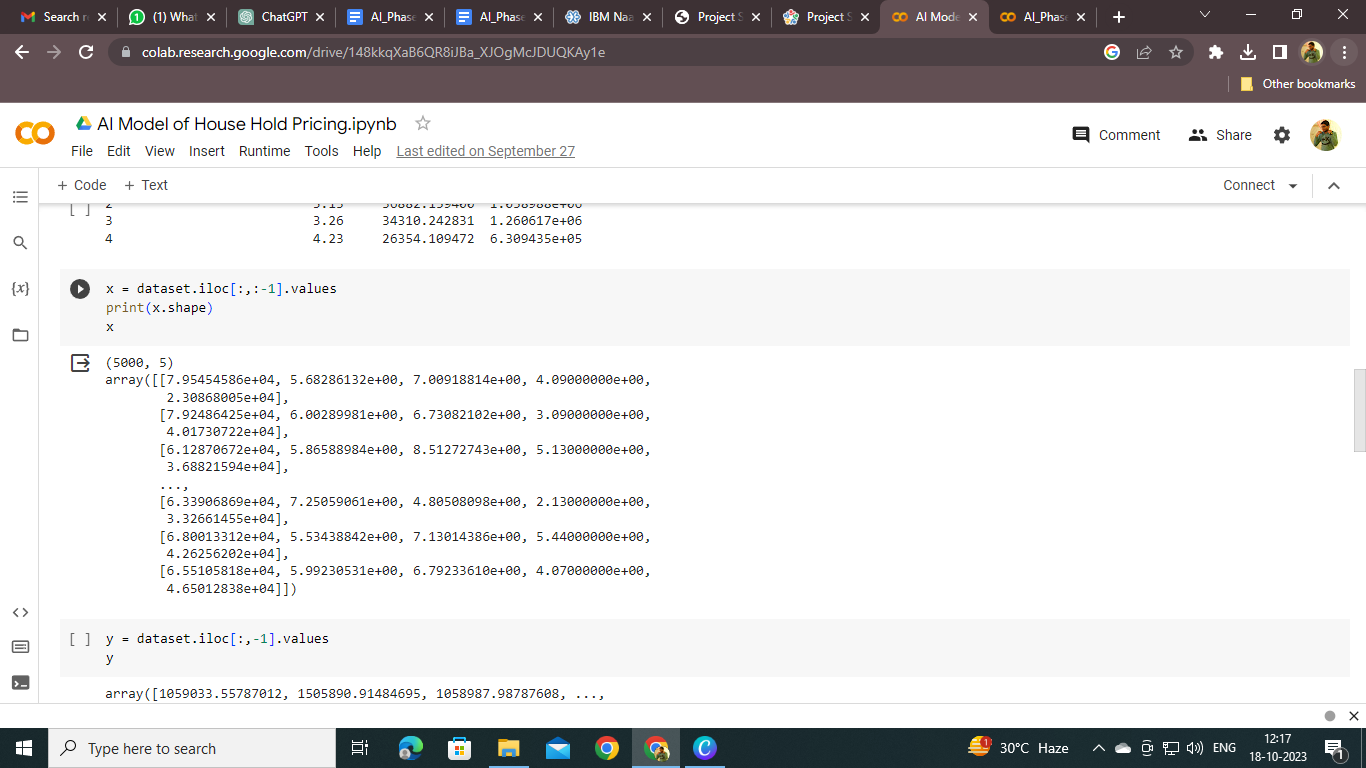
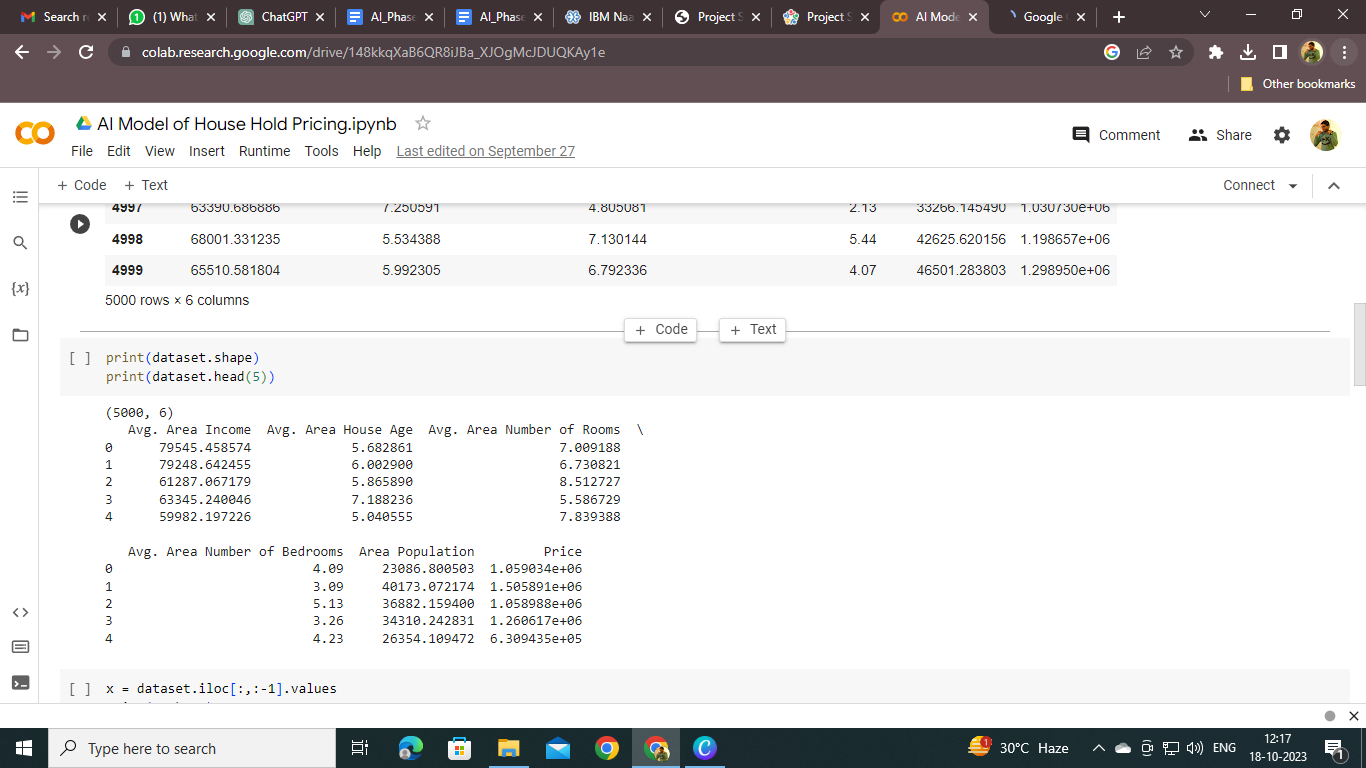
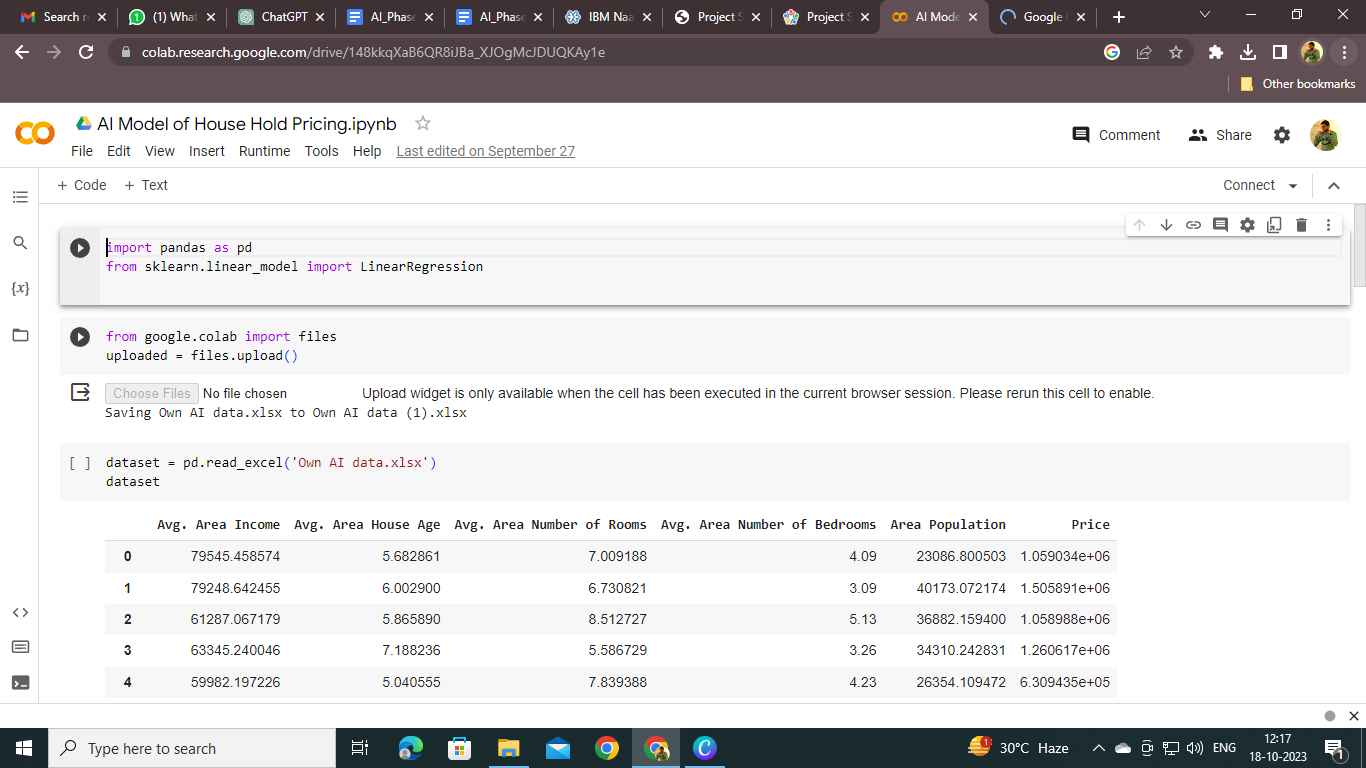
print(f'R-squared Value: {r2}')

Algorithm:

* **Random Forest Regression:**



* **Linear Regression:**



* **XG Booster:**

