SURYA GROUP OF INSTITUTION

VIKRAVANDI 605-652

PHASE 4 DEVELOPMENT PART 2

PREDICTION HOUSE PRICES USING MACHINE LEARNING

NAAN MUDHALVAN

PREPARED BY:

C.ANJUGAM

REG NO :422221106002

ECE DEPARTMENT

3RD YEAR 5TH SEM

AI_ PHASE 4:

To Develop the project development part 2 is build a model evaluation, model training by using Linear Regression and XG boost regression.

In this proposed system, we focus on predicting the house price values using machine learning algorithms like XG Boost regression model and Linear Regression . We proposed the system “House price prediction using Machine Learning” we have predicted the House price using XG boost regression and linear regression model. In this proposed system, we were able to train the machine from the various attributes of data points from the past to make a future prediction. We took data from the previous year stocks to train the model .The data set we used was from the official organization. Some of data was used to train the machine and the rest some data is used to test the data. The basic approach of the supervised learning model is to learn the patterns and relationships in the data from the training set and then reproduce them for the test data. We used the python pandas library for data processing which combined different datasets into a data frame. The raw data makes us to prepare the data for feature identification. The attributes were stories, no. of bed rooms, bath rooms, Availability of garage, swimming pool, fire place, year built, area in soft, sale price for a particular house. We used all these features to train the machine on XG boost regression and predicted the house price, which is the price for a given day. We also quantified the accuracy by using the predictions for the test set and the actual values. The proposed system gives the Predicted price

ALOGORITHM:

We used the python pandas library for data processing which combined different datasets into a data frame. The raw data makes us to prepare the data for feature identification. XG for regression builds an additive model in a forward stage wise fashion. It allows for the optimization of arbitrary differentiable loss functions. In each stage, a regression tree is fit on the negative XG of the given loss function. The idea of boosting came out of the idea of whether a weak learner can be modified to become better. A weak hypothesis is defined as one whose performance is at least slightly better than random chance. The Objective is to minimize the loss of the model by adding weak hypothesis using a XG descent like procedure. This class of algorithms was described as a stage-wise additive model. This is because one new weak learner is added at a time and existing weak learners in the model are frozen and left unchanged.

Step 1: Load the data set df = pd.read csv(“ml_house_data_set.csv”)

Step 2: Replace categorical data with one-hot encoded data

Step 3: Remove the sale price from the feature data

Step 4: Create the features and labels X and Y arrays.

Step 5: Split the data set in a training set (70%) and a test set (30%).

Step 6: Fit regression model.

Step 7: Save the trained model to a file trained_house_classifier_model.pkl

Step 8: Predict house worth using predict function

Model Building and evaluation:

Model 1 : Linear regression

Input;

model_lr=LinearRegression() model_lr.fit(X_train_scal,

Y_train)

Output:

LinearRegression()

Predicting Prices

Prediction1 = model_lr.predict(X_test_scal)

Evaluation of predicted data:

Input :

plt.figure(figsize=(12,6)) plt.plot(np.arange(len(Y_test)), Y_test,

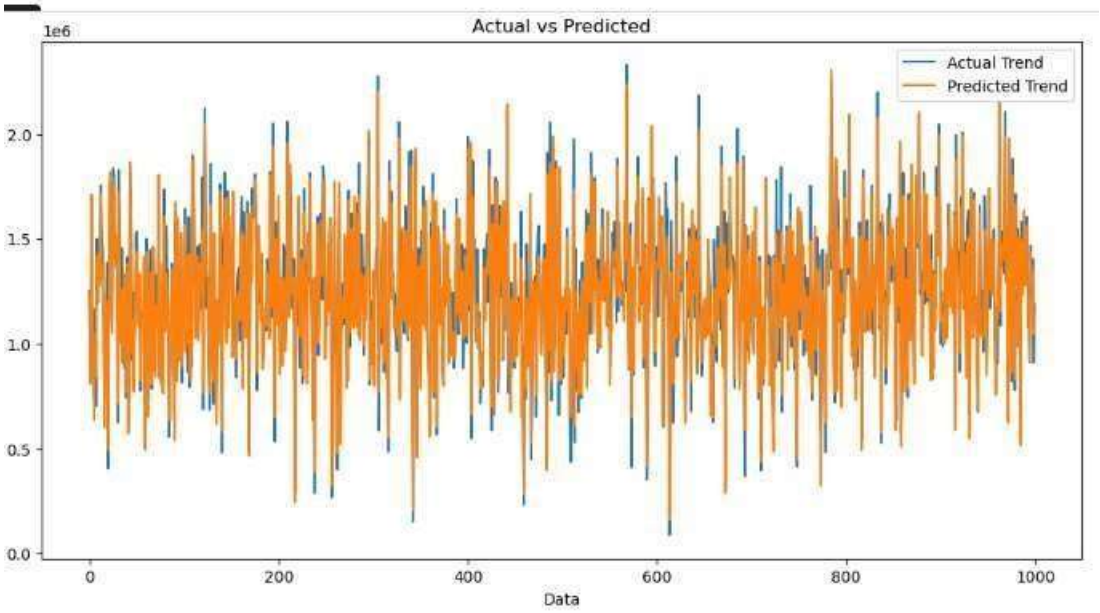
label='Actual Trend') plt.plot(np.arange(len(Y_test)), Prediction1,

label='Predicted Trend') plt.xlabel('Data') plt.ylabel('Trend') plt.legend()

plt.title('Actual vs Predicted')

Output:

Text(0.5, 1.0, 'Actual vs Predicted')



To Find a Mean absolute Error:

Input:

sns.histplot((Y_test-Prediction1), bins=50)

Output:

<Axes: xlabel='Price', ylabel='Count'>

Input:

print(r2_score(Y_test, Prediction1)) print(mean_absolute_error(Y_test,

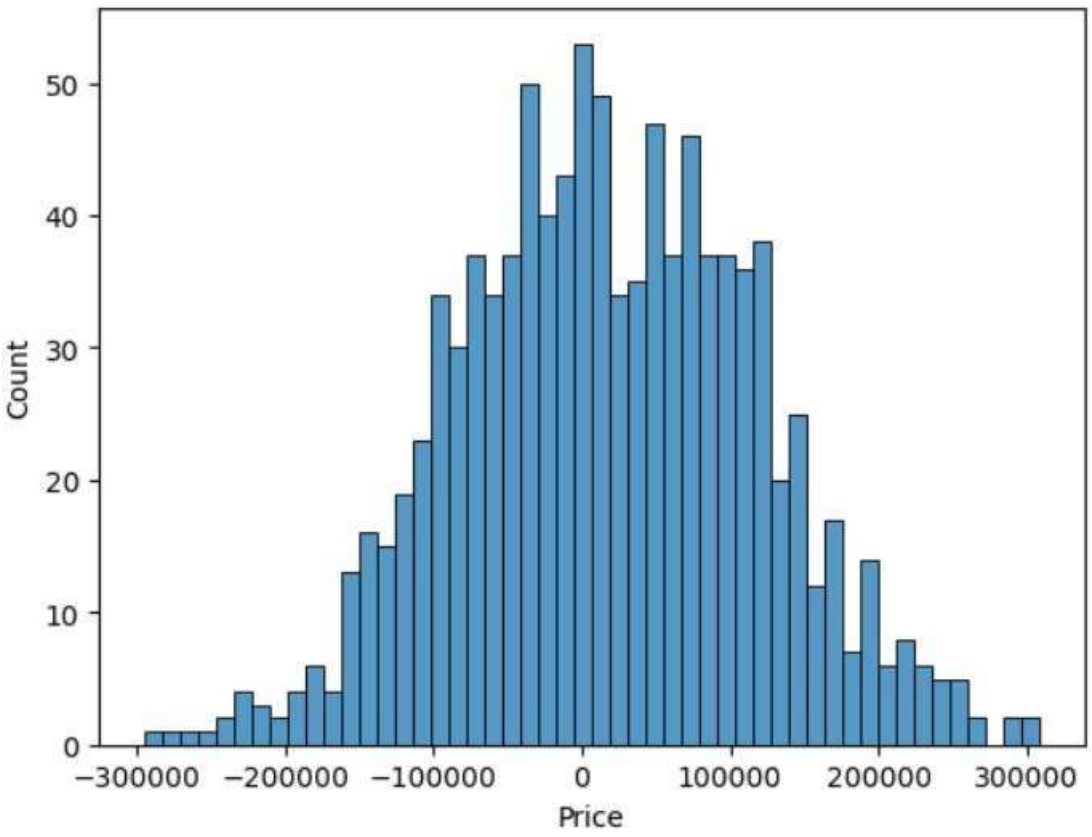
Prediction1)) print(mean_squared_error(Y_test, Prediction1))

Output:

0.9182928179392918

82295.49779231755

10469084772.975954



Model 2: XG Boost Regressor :

Input:

model_xg = xg.XGBRegressor() model_xg.fit(X_train_scal,

Y_train)

Output:

XGBRegressor (base_score=None, booster=None, callbacks=None, colsample_bylevel=None, colsample_bynode=None, colsample_bytree=None, early_stopping_rounds=None, enable_categorical=False, eval_metric=None, feature_types=None, gamma=None, gpu_id=None, grow_policy=None,

importance_type=None, interaction_constraints=None, learning_rate=None, max_bin=None, max_cat_threshold=None, max_cat_to_onehot=None, max_delta_step=None, max_depth=None, max_leaves=None, min_child_weight=None, missing=nan, monotone_constraints=None, n_estimators=100, n_jobs=None, num_parallel_tree=None, predictor=None, random_state=None, ...)

Predicting Prices :

```
Prediction5 = model_xg.predict(X_test_scal)
```

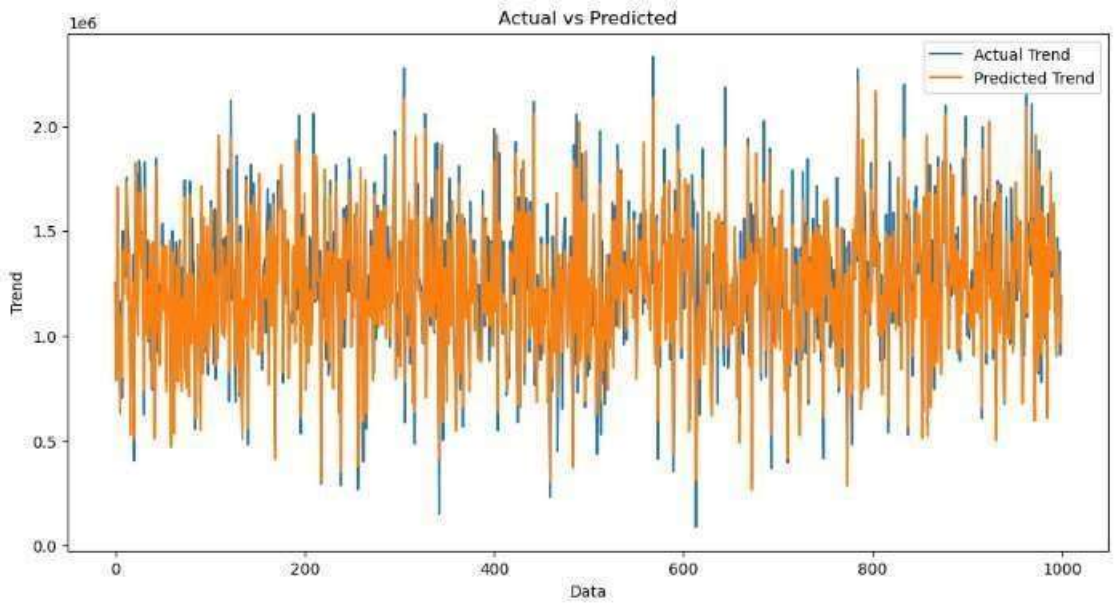
Evaluation of Predicting Prices:

Input:

```
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)), Y_test, label='Actual Trend')
plt.plot(np.arange(len(Y_test)), Prediction5, label='Predicted Trend')
plt.xlabel('Data') plt.ylabel('Trend') plt.legend()
plt.title('Actual vs Predicted')
```

Output:

Text(0.5, 1.0, 'APrediction5 = model_xg.predict(X_test_scal)

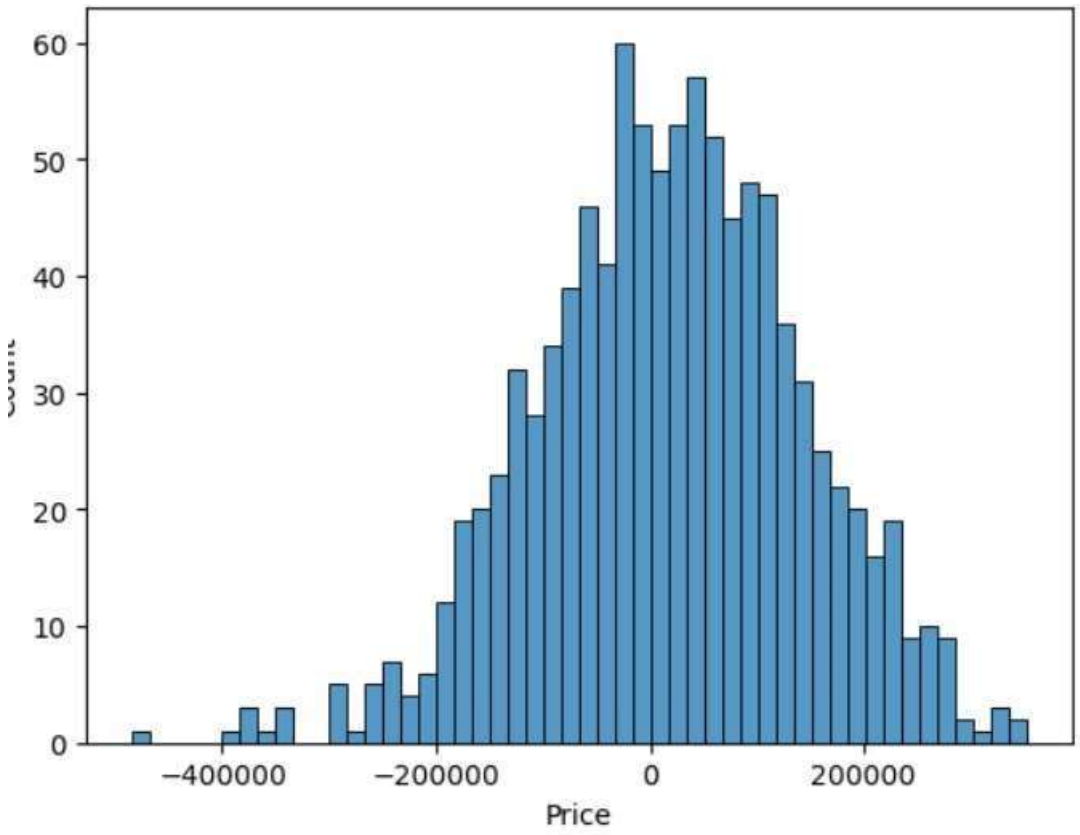


print(r2_score(Y_test, Prediction2)) print(mean_absolute_error(Y_test, Prediction2)) print(mean_squared_error(Y_test, Prediction2))

-0.0006222175925689744

286137.81086908665 128209033251.4034 sns.histplot((Y_test-Prediction4), bins=50)

<Axes: xlabel='Price', ylabel='Count'>



CONCLUSION:

This project entitled “House Price Prediction algorithm had all other algorithms regarding to model building and evaluation in this project