Test-train split

Create training and test splits in using a split ratio of 30%:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

Assumption: By training and test split ratio it means 70% training and 30% test. There's no ratio function for train_test_split so I used the first function in the documentation which is test_size, and used 0.3 due to 30%.

Create linear regression object:

```
LinearRegression = LinearRegression()
```

Fit the model on to the instantiated object itself:

```
LinearRegression.fit(X_train, y_train) # Using fit we put training data into the model.
✓ 0.1s
LinearRegression
LinearRegression()
```

Check the intercept and coefficients: ->

```
Intercept: -2639897.157447537
Coefficients: [2.16053752e+01 1.65383679e+05 1.20959919e+05 8.67234961e+02
 1.52685340e+01]
```

Sort the features based on the t-statistic:

```
metric_calculations(X_train, y_train, LinearRegression) # Assumptions: We just use the above function and by model it means the object.
✓ 0.0s
```

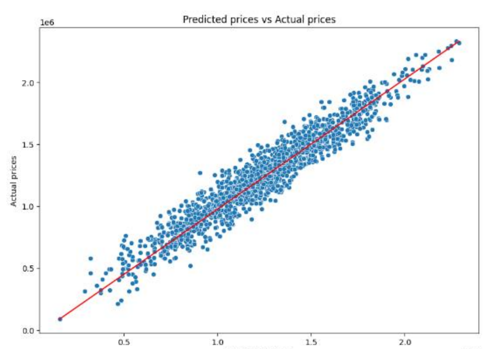
	Coefficients	Standard Error	t-statistic
Avg. Area Income	21.605375	0.161641	133.663013
Avg. Area House Age	165383.679490	1721.176995	96.087549
Avg. Area Number of Rooms	120959.919418	1690.030744	71.572615
Avg. Area Number of Bedrooms	867.234961	1357.935779	0.638642
Area Population	15.268534	0.171699	88.926203

Do predictions with the linear model:

```
y_prediction = LinearRegression.predict(X_test)
```

Assuming we use X_test since the source code documents predict() to be this: predict(self, X), and we use X_test and not X_train because the model was trained on X_train and we want to predict with new data to avoid overfitting

Plot predictions against the ground truth values:



What can be determined from this plot:

It can be determined that the model works and isn't overfitted. That the trained data works with the predicted data, i.e. future predictions can be made confidently.

```
Mean absolute error: 81400.85819674339
Mean squared error: 10429346272.554321
Root mean squared error: 102124.17085369321
R2 value: 0.920013906373715
```

Optional plots: I did them both, please check the .ipynb