

May Mon Thant
Mentor - Nu Wai Thet

Question (1)

a) Which model gives you better performance: Multiple Linear Regression or Polynomial Regression?


Provide evidence from your analysis (such as R-squared, Mean Squared Error, or other performance metrics) and explain your reasoning. Be sure to compare the results from both models and justify why one might perform better than the other based on the dataset and problem you're solving.

To answer the question (1) we need to compare the two models, multiple regression and Polynomial regression.

There is one thing that we need to consider is the size of the dataset. In our current dataset , there are **50** rows, but I think we need to get more training data to analyze the patterns more. If we have more data, our polynomial model can give better results because for now it has some error or irrelevant results in testing and which we can say we have less training data.


In the code, the best order 2 is saved as the deployment model. So here, I would like to start with comparing two values from multiple linear regression and the best model from polynomial regression.

I have to create a multiple regression model based on the df1 datasets and this is the results.

```
0s ✓  from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

Y_pred = lr.predict(X_scaled)
rscore = r2_score(Y, Y_pred)
mse = mean_squared_error(Y, Y_pred)
mae = mean_absolute_error(Y, Y_pred)

print('r2-score:', rscore, '\nmean squared error:', mse, '\nmean absolute error:', mae)
```

```
 r2-score: 0.0
mean squared error: 10547390952.166204
mean absolute error: 78355.04155124653
```

Model	R Score	Mean Squared Error (MSE)	Mean Absolute Error (MAE)
Multiple Linear Regression	0.0	10547390952.166204	78355.04155124653
Polynomial Regression (order = 2)	0.428	60.99	157,625.27

In MLR, the two features are used to predict the current price, which are sqrt and bedrooms, but in polynomial regression only sqrt is used.

But I think this does not give any significant difference because when we checked the coef of bedrooms it gives minus value. Square feet alone can predict the price.

If we compare each value from the two models, MLR can explain 0% of the variation and the polynomial model can explain 42 % (obviously poly better) But poly models have lower MAE than MLR models.

In the last assignment, we used MLR to explain how the target price is influenced by features such as square ft and bedrooms. However, this time I want to focus on the prediction price which indicates the small error with one feature.

But in the previous assignment, the MLR model with a full dataset can explain 50% of the data. Even if I will choose Poly here because Poly can predict more accurately with one feature.

```
[ ] new_sqft = np.array([[100]])
    scaled_input = scaler.transform(new_sqft)
    poly_input = poly.transform(scaled_input)

    predicted_price = model.predict(poly_input)
    print("Predicted price: {:.2f}".format(predicted_price[0]))
```

➡ Predicted price: -60631.98

I have tried with sqrt 100 here and it gives me a negative result.

In real life, the price of house data is changing over time and somehow I think it doesn't always go in the straight line. It also depends on the place like rural, urban or suburban areas. On the same sqrt, the price of the house is not the same as in Yangon and Patheingyi. So, in this case, using a polynomial regression model is the good choice.

b)How do you decide the optimal degree for Polynomial Regression in this case?

Explain how you determined the degree of the polynomial and what criteria you used to decide whether a higher degree improves the model's performance. Discuss the potential risks of choosing too high or too low a degree for the polynomial.

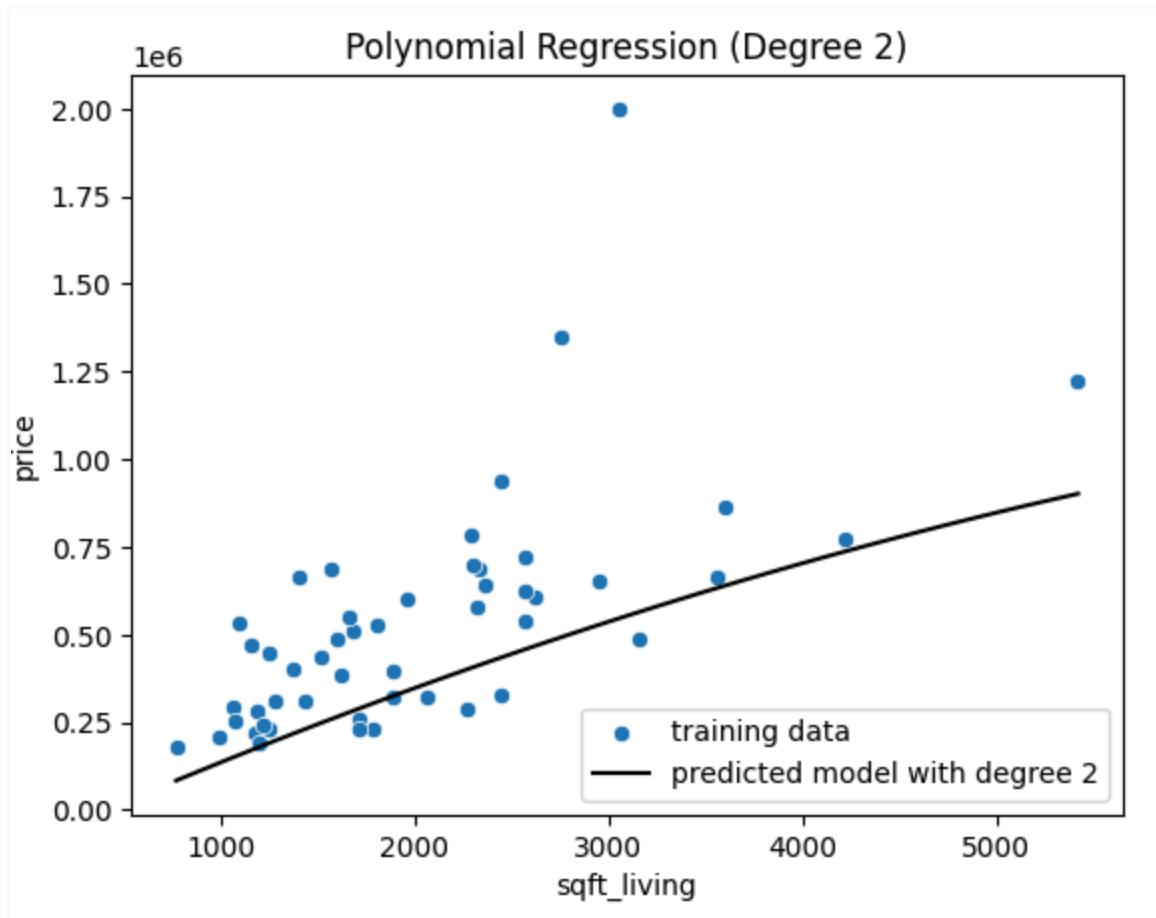
First, I have started with order one which is actually linear regression and then with 5.

What I understand about degree or order is using it to find more complex patterns.

If our degree is too low, we cannot observe patterns in the dataset. If we use too a highdegree, our model can only memorize the patterns in our dataset and cannot properly work well when working with real world data. Models can be easily overfit in this small dataset.

I have tested with 4 degrees and chose degree 2 as the best one.

Among the four degrees, degree 10 has the best performance with an R score 53%. However, as our dataset is small enough to catch up the data in order 2. In this case, I can also avoid the problem of overfitting in our small dataset. And also MAE is the lowest in degree 2 which is relevant with our goal of predicting the accurate price. It is also easy to understand the patterns and visualize the results in degree 2.



Degree	Rscore	MSE	MAE
1	0.4173	62,185,849,719.22	161,171.37
2	0.4285	60,999,286,388.34	157,625.27
5	0.4780	55,715,230,172.06	161,830.17
10	0.5353	49,596,225,121.88	158636.3