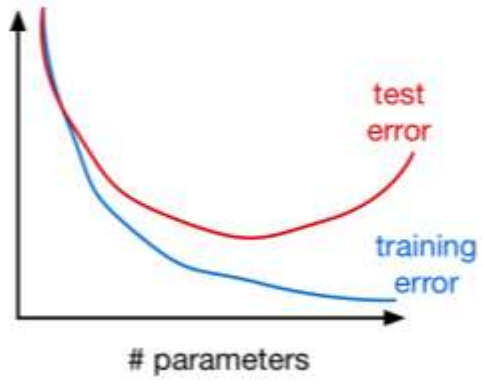
Short Answer and “True or False” Conceptual questions

Hw 1 Jason Bian

1a. \ Bias is the difference between the expected model parameter value and the true parameter value. Variance is the amount that the estimated ML function will change given different training data. Typically, bias^2 decreases as complexity increases and variance increases as complexity increases. This tradeoff between bias and variance when we increase complexity is called the bias-variance tradeoff.

1.b \ As stated above in the definition, typically bias^2 decreases as complexity increases and variance increases as complexity increases.

1.c \ False, feature selection dictates how well the model generalizes on the test set. Having too few features will lead to bad performance on the test set. Having too many features can also lead to bad performace on the test set.



1.d False, Hyperparameters should be tuned on the training set. Grid search is a basic hyper parameter tuning method. In Grid search, we build a model for each possible combination of all the hyperparameter values provided. Then, we evaluate each model and select the arhitecture which produces the best results.

1.e False, your training error can be both an overestimate and an underestimate of the test error. In the case where the test error is less than the training error, there must be sampling bias in the test data.

Maximum Likelihood Estimation (MLE)

2a.

$$x = [2, 4, 6, 0, 1]$$

$$P(x|\theta) = e^{-\theta}\frac{\theta^x}{x!}$$

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \log P(x|\theta)$$

$$f(x|\theta) = \sum_{n}^{j=1} ln(e^{-\theta}\frac{\theta^x}{x!})$$

$$f(x|\theta) = \sum_{n}^{j=1} [-\theta - ln(x_j!) + x_j ln(\theta)]$$

$$f(x|\theta) = -n\theta - \sum_{n}^{j=1} ln(x_j!) + ln(\theta)\sum_{n}^{j=1} x_j$$

$$\hat{\theta}_{MLE} = \frac{d}{d\theta} f(x|\theta)$$

$$\frac{d}{d\theta} f(x|\theta) = \frac{d}{d\theta}(-n\theta - \sum_{n}^{j=1} ln(x_j!) + ln(\theta)$$

$$\frac{d}{d\theta} f(x|\theta) = -n + \frac{1}{\theta}\sum_{n}^{j=1} x_j$$

$$\hat{\theta}_{MLE} = \frac{1}{n}\sum_{n}^{j=1} x_j$$

2b.

$$\hat{\theta}_{MLE} = 2.6$$

$$P(6|\theta) = e^{-2.6}\frac{2.6^6}{6!}$$

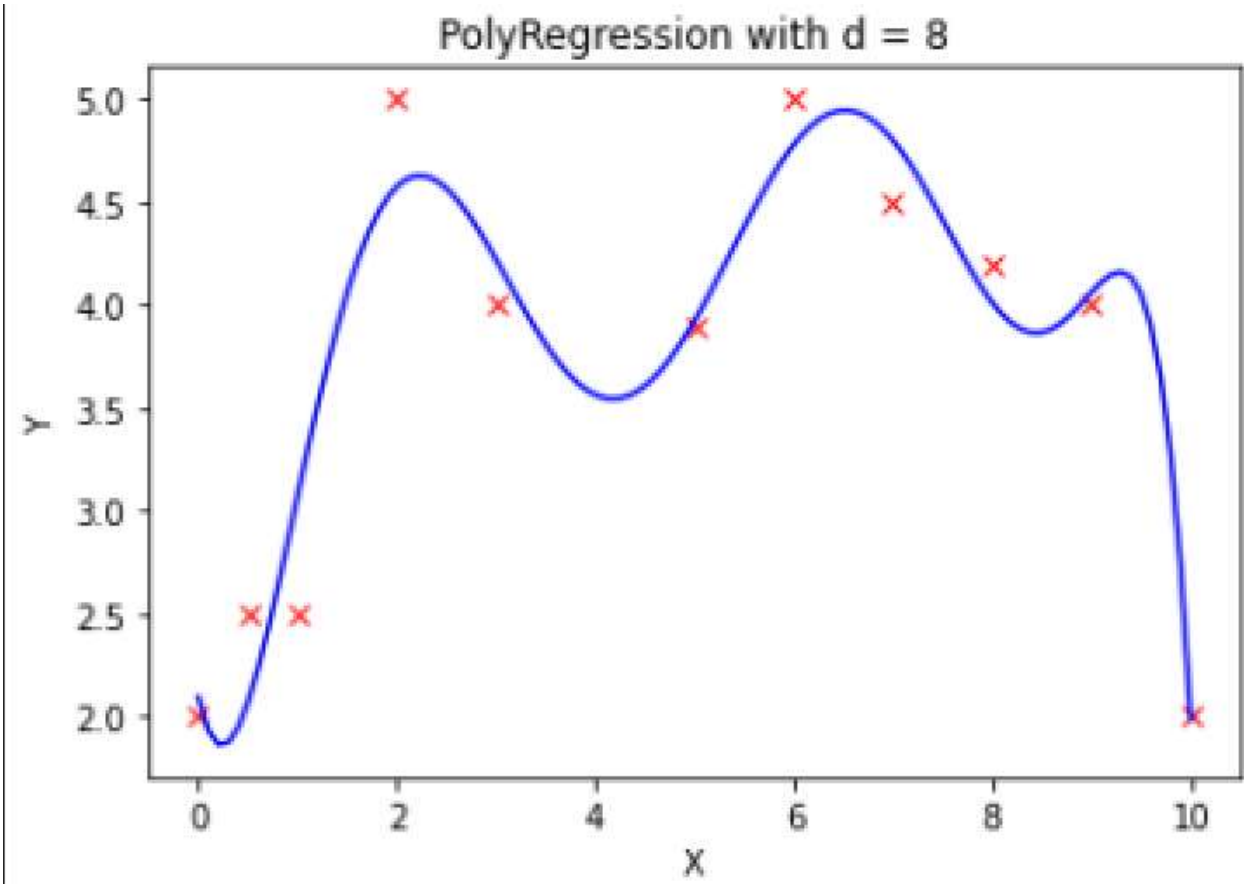
In [ ]:
```
import math

val = math.exp(-2.6) * ((2.6 ** 6)/(math.factorial(6)))
print(val)
```

```
0.03186705562552451
```

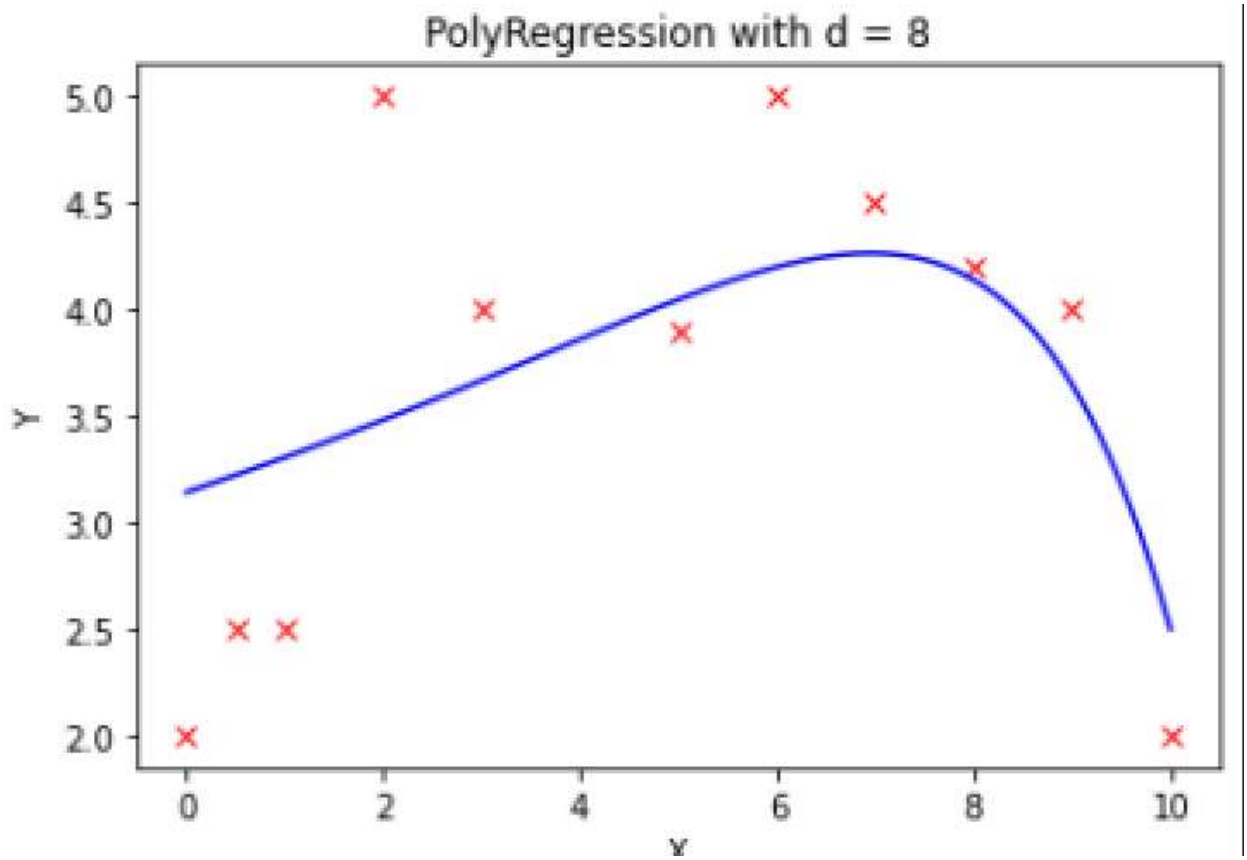
$$P(6|\theta) = 0.0318671$$

3a: As we increase the amount of regularization, the polynomial tends to fit the test set less and less:

For example, below is lambda = 0:



and below is lambda = 4:



4a:

We can write eq 1 into ten different equations by spliting e_j
into 10 equations that correspond to the 1 in each one-hot encoded value

$$\sum_{j=1}^{k}[||Xw_j - Ye_j||^2 + \lambda||w_j||^2] \text{ (eq 1)}$$

$$\sum_{j=1}^{k}[||Xw_{ij} - y_{ij}||^2 + \lambda||w_{ij}||^2]$$

$$\forall \text{i in } e_i \text{ (eq 2)}$$

If we take one e_i, we can rewrite the summation as the following matrixes:

$$= (X\widehat{W} - Y)^T(X\widehat{W} - Y) + \lambda\widehat{W}^T\widehat{W}$$

$$= (X\widehat{W})^T(X\widehat{W}) - (X\widehat{W})^TY - (X\widehat{W})Y + Y^TY + \widehat{W}^T\lambda I\widehat{W}$$

$$= -2(X\widehat{W})^TY + Y^TY + \widehat{W}^T(X^TX + \lambda I)\widehat{W}$$

Solving for $\widehat{W}$, we get

$$= \frac{\partial}{\partial\widehat{W}} = -2X^Ty + 2(X^TX + \lambda I)\widehat{W}$$

$$= (X^TX + \lambda I)\widehat{W} = X^TY$$

$$= (X^TX + \lambda I)X^TY$$

This should be the same result for each $e_i$value

In [ ]:
```
# Use one-hot encoding to remove the e term
# Take the resulting weights and add them all together
```

6a: 20 hours total