Conceptual

```
from notebookfuncs import *
```

1. For each of parts (a) through (d), indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answer.

(a) The sample size n is extremely large, and the number of predictors p is small.

- Flexible models, in general, perform better with more data and a smaller number of parameters to data point count.
- A flexible model fits the data better in the training data.
- Whether the flexible model performs better on test data depends on the true relationship underlying the data.
- If the true relationship is linear, then the flexible model will not perform as well on test data as the variance will tend to increase given that these are unseen observations not used to train the model.
- Generally, the flexible model will be better.

(b) The number of predictors p is extremely large, and the number of observations n is small.

- The inflexible model will perform better when the number of predictors are extremely large and the number of observations are few in number.
- The inflexible model will tend to overfit the data.
- This can be mitigated by use of well-designed syntehtic data but that's a topic yet to be covered or learnt.

(c) The relationship between the predictors and response is highly non-linear.

- A flexible model will be better since flexible models are better at capturing non-linearities in the data.

(d) The variance of the error terms, i.e.. $\sigma^2 = Var(\epsilon)$, is extremely high.

- When the variance of the error terms is very high, the flexible model is worse since it will tend to overfit the model i.e.. it will tend to explain part of the variance as well i.e, it will tend to fit too much of the noise.

2. Explain whether each scenario is a classification or regression problem, and indicate whether we are most interested in inference or prediction. Finally, provide n and p.

(a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

- The CEO Salary is a continuous variable.
- n = 500
- p = 3
- The problem is thus a regression problem.

(b) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

- Response variable is Success/ Failure of the new product
- n = 20
- p = 13
- The problem is a classification one.

(c) We are interested in predicting the % change in the USD/Euro exchange rate in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the USD/Euro, the %change in the US market, the % change in the British market, and the % change in the German market.

- Response is %change in USD/Euro exchange rate. This is a continuous variable.
- n = 52
- p = 3
- The problem is thus a regression problem.

3. We now revisit the bias-variance decomposition.

(a) Provide a sketch of typical (squared) bias, variance, training error, test error, and Bayes (or irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches. The x-axis should represent the amount of flexibility in the method, and the y-axis should represent the values for each curve. There should be five curves. Make sure to label each one.

```
def draw_bias_variance_plot():

  import numpy as np
  from matplotlib import pyplot as plt
  from scipy import interpolate

  # Set the figure size
  plt.rcParams["figure.figsize"] = [15.00, 10]
  plt.rcParams["figure.autolayout"] = True

  # Drae Test MSE curve
  nodes = np.array( [ [0, 135], [45, 90], [90, 135] ] )

  x = nodes[:,0]
  y = nodes[:,1]

  tck,u     = interpolate.splprep( [x,y] ,s = 0 , k=2)
  xnew,ynew = interpolate.splev( np.linspace( 0, 1, 100 ), tck,der = 0)

  plt.plot( xnew ,ynew, 'Orange', label="Test MSE" )

  # Draw Training  Error curve
```

```
nodes = np.array( [ [0, 125], [35, 75], [90, 25] ])
x = nodes[:,0]
y = nodes[:,1]

tck,u     = interpolate.splprep( [x,y] ,s = 0 , k=2)
xnew,ynew = interpolate.splev( np.linspace( 0, 1, 100 ), tck,der = 0)

plt.plot( xnew ,ynew, 'Green', label="Training Error" )


# Draw bias-squared
nodes = np.array( [ [0, 80], [70, 5], [90, 5] ] )

x = nodes[:,0]
y = nodes[:,1]

tck,u     = interpolate.splprep( [x,y] ,s = 0 , k=2)
xnew,ynew = interpolate.splev( np.linspace( 0, 1, 100 ), tck,der = 0)

plt.plot( xnew ,ynew, 'Red', label=r"$Bias^2$" )

# Draw Variance
nodes = np.array( [ [3, 5], [70, 25], [90, 80] ] )

x = nodes[:,0]
y = nodes[:,1]

tck,u     = interpolate.splprep( [x,y] ,s = 0 , k=2)
xnew,ynew = interpolate.splev( np.linspace( 0, 1, 100 ), tck,der = 0)

plt.plot( xnew ,ynew, 'Blue', label="Variance" )


ax = plt.gca()

# Hide X and Y axes label marks
ax.xaxis.set_tick_params(labelbottom=False)
ax.yaxis.set_tick_params(labelleft=False)

# Hide X and Y axes tick marks
ax.set_xticks([])
ax.set_yticks([])
```

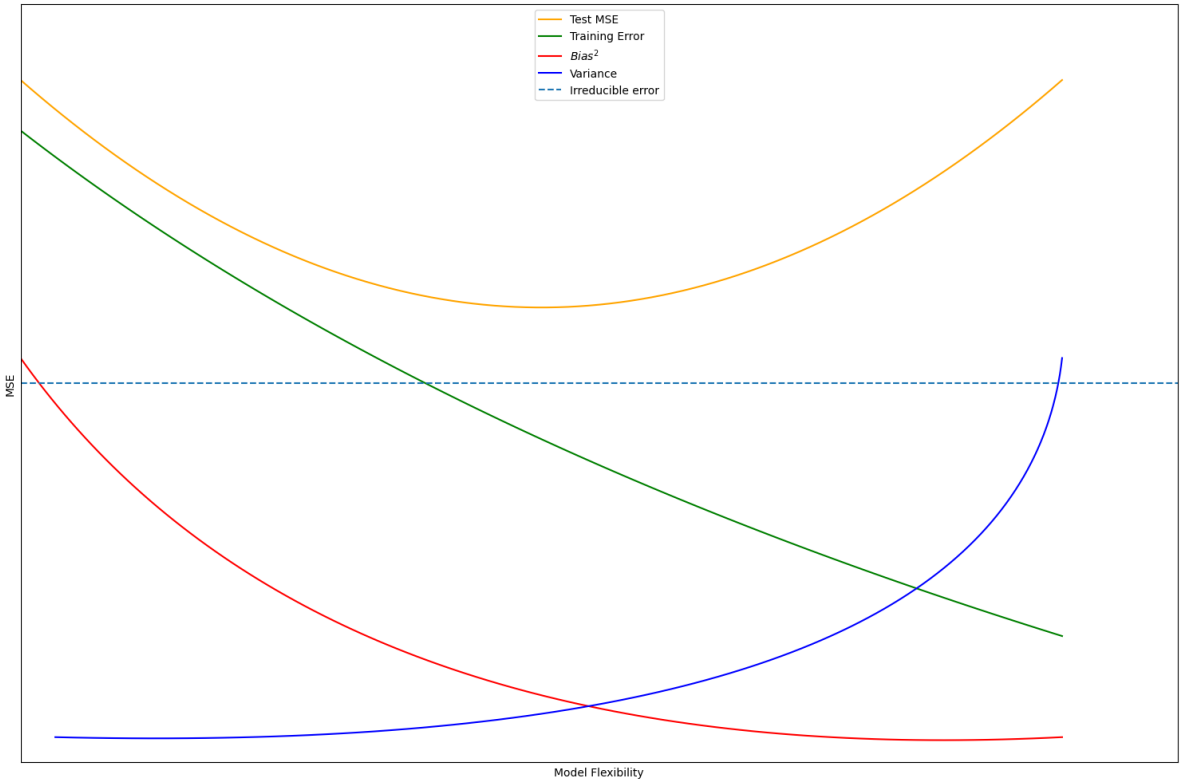
```
  # set x and y limits
  ax.set_xlim(0, 100)
  ax.set_ylim(0,150)

  plt.xlabel("Model Flexibility")
  plt.ylabel("MSE")

  # Draw irreducible error
  ax.axhline(75, linestyle='--', label="Irreducible error")

  plt.legend()
  plt.show();

draw_bias_variance_plot();
```



(b) Explain why each of the five curves has the shape displayed in part (a).

Test MSE

The Test MSE is U-shaped. As the model flexibility increases, the test MSE decreases initially before increasing as the model's flexibility increaases. This is because as the model's flexibility increases, it tends to overfit to the training data. This, however, leads to an increased error for out-of-sample values thus leading to an increase in the Test MSE.

Training Error

The training error decreases montonically as the flexibility increases. This is because as noted above, the model tends to fit to the data thus leading to closeness between the estimated values and the actual values. Thus, the training error decreaees montonically,

$Bias^2$

The Squared Bias decreases as the model flexibility increases. A more flexible model is more likely to approximate the true function and thus the bias decreases as the model's flexibility increases. Thus, the estimate is more likely to match the actual value.

Variance

The variance of the model increases with the model's flexibilitty since a more flexible model is prone to overfitting and thus may not work as well on unseen data. Unless the existing model is an extraordinarily good predictor of the population values, it is quite likely that the variance will increase as the model's flexbility increases.

Irreducible error

Irreducible error will always exist even if we somehow approximate the true function for the given values of X, the input features. This is because there will alwys be some fluctuations that cannot be explianed by the model. This is constant and is alsp termed Bayes error. This is caused by unknown variables and is the noise in the data.

$$Error = Bias^2 + Variance + IrreducibleError$$

4. You will now think of some real-life applications for statistical learning.

(a) Describe three real-life applications in which classification might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

- Predict the probability of a student getting a job through campus placements based on the following predictors: GPA, Aptitude Scores, Extra-curricular activities, internship count and state of economy that year.
- Loan Defaults: Classify loans into default or not based on factors such as credit score, outstanding balance, previous history of defaults, current income, total payments so far etc
- Fraud detection systems: Predict whether transaction is fraudulent or not based on factors such as size of transaction, time of day, first-time transaction of this type i.e.. new payee, country etc
- Disease prediction: In medicine, this analytics approach can be used to predict the likelihood of disease or illness for a given population. Healthcare organizations can set up preventative care for individuals that show higher propensity for specific illnesses.
- Churn prediction: Specific behaviors may be indicative of churn in different functions of an organization. For example, human resources and management teams may want to know if there are high performers within the company who are at risk of leaving the organization; this type of insight can prompt conversations to understand problem areas within the company, such as culture or compensation. Alternatively, the sales organization may want to learn which of their clients are at risk of taking their business elsewhere. This can prompt teams to set up a retention strategy to avoid lost revenue.

(b) Describe three real-life applications in which regression might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

- Predict batting average for a player in the IPL based on balls faced + highest score + runs scored + strike
- Predict a pitcher's performance
- We can infer the relationship between house price and factors such as area, number of rooms age, locality or zip code, amenities nearby, closeness to nearest station etc.

(c) Describe three real-life applications in which cluster analysis might be useful.

1. Industrial clustering analysis benefits from using a cluster analysis to identify agglomeration in a region or nation. https://www.nature.com/articles/s41598-023-27655-8

2. Identification of residential building topologies https://www.sciencedirect.com/science/article/pii/S2352710224004807
3. Cluster analysis to segregate cricketers into categories or types based on their performance data to enhance the selection process of players during the IPL auction. https://towardsdatascience.com/machine-learning-to-cluster-cricket-players-1d53beeb69b4

5. What are the advantages and disadvantages of a very flexible (versus a less flexible) approach for regression or classification? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible approach be preferred?

- Flexible approaches to regression or classification are able to capture non-linearities in the data better than a non-flexible (or linear approach).
- They are able to generalise better to unseen data when the true models are not linear in nature.
- One of their disadvantages is that they're hungry for data and have a lot more parameters than a simpler, linear approach.
- Linear approaches work better when the modelers have an idea of the domain and are thus able to generate parsimonious models which are usually good enough for their purposes.
- Interpretability is also simpler with less flexible models.
- It is difficult to understand a more flexible model.

A more flexible approach is preferred when the model is known to be complex and cannot be easily captured by a less flexible model.

A less flexible approach is preferred when the modelers seek interpretability, simplicity, speed, eficiency and have domain knowledge and expertise. Addtionally, when data quality is poor, a simpler model may function better than a more flexible model that may not learn as well and which adapts to the noise in the data.

References: 1> https://www.baeldung.com/cs/ml-flexible-and-inflexible-models

6. Describe the differences between a parametric and a non-parametric statistical learning approach. What are the advantages of a parametric approach to regression or classification (as opposed to a non-parametric approach)? What are its disadvantages?

- A parametric approach involves a two-step model-based approach.

1. We make an assumption about the functional form or shape of f. For example, we could assume that f is linear with p + 1 coefficients, $beta_0.\beta_1, ...\beta_p$. Then , we only have to estimate the p + 1 coeffcients of the function.

2. After we select the model, we need to choose a training method that fits the data. One of the possible ways to fit a linear model is Ordinary Least Squares (OLS).

- The parametric approach is simpler since we just have to estimate the p + 1 parameters in the above example.
- The disadvantage with a parametric approach is if it doesn't actually match the true, unknown f.
- Then, we have to opt for more flexible methods which involve a lot more parameters These models are more complex and could lead to overfitting the data i.e., they tend to adapt to the noise in the data.
- A non-parametric approach, on the other hand, does not make any assumptions about the functional form of f. Instead they seek an estimate of f that gets close to the data points wihout being too rough or wiggly. They can accurately fit many possible shapes of f.
- But since non-parametric models need more parameters, they will also require more data observations.

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Obs	X_1	X_1	X_1	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	-1	0	1	Green
6	1	1	1	Green

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

(a) Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$.

```
import numpy as np
from scipy.spatial import distance

X = np.array([[0,3,0], [2,0,0], [0,1,3], [0,1,2], [-1, 0,1],[1,1,1]], np.int32)
Y = np.transpose(np.array(["Red", "Red", "Red", "Green", "Green", "Green"]))
X0 = np.array([0,0,0])
```

```
euclidean_distances = [distance.euclidean(X0, point) for point in X]
print(euclidean_distances)
```

```
[3.0, 2.0, 3.1622776601683795, 2.23606797749979, 1.4142135623730951, 1.7320508075688772]
```

(b) What is our prediction with K = 1? Why?

```
index = np.argmin(euclidean_distances)
print(Y[index])
```

```
Green
```

(c) What is our prediction with K = 3? Why?

```
k = 3
idx = np.argpartition(euclidean_distances,k)
print(Y[idx[: k]])
```

```
['Green' 'Green' 'Red']
```

- With k = 3, our prediction will be the majority of the three closest neighbours of the point (0,0,0)
- In this case, it's Green, 2 out of 3.

(d) If the Bayes decision boundary in this problem is highly non-linear, then would we expect the best value for K to be large or small? Why?

- If the Bayes decision boundary is highly non-linear, we expect k to be small and best fit the decision boundary. That's because a small k would make the fit more flexible and would be drawn to the data boundaries better than a larger k which would smmoothen the data and probably have a more linear fit.
- The choice of k is crucial while using this method given that the Test Error is U-shaped and there has to be a balance struck in the Bias-Variance tradeoff.
- The choice of k = 1 minimizes the training error rate at the risk of overfitting to the seen data which will have high variance when exposed to unseen data.

```
allDone();
```

```
<IPython.lib.display.Audio object>
```