

COMP30027 MACHINE LEARNING TUTORIAL

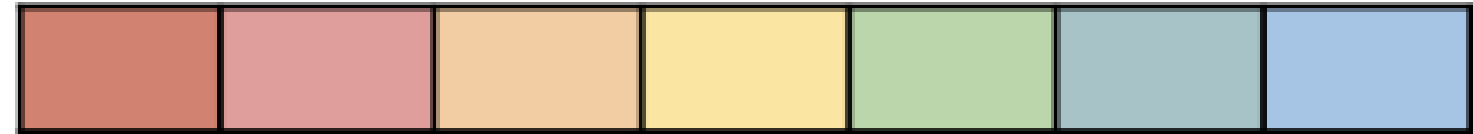
Workshop - 8

Feature Selection and Feature Evaluation

Feature selection is the process of choosing a subset of relevant and useful input features from the original dataset that contribute the most to the predictive performance of a model.

Feature evaluation is the process of assessing the importance or usefulness of each feature based on a on a certain metric or criterion. It helps you rank or score or score features before feature selection.

All Features



Feature Selection



Final Features



Why Feature Selection Matters

Reduced Training Time

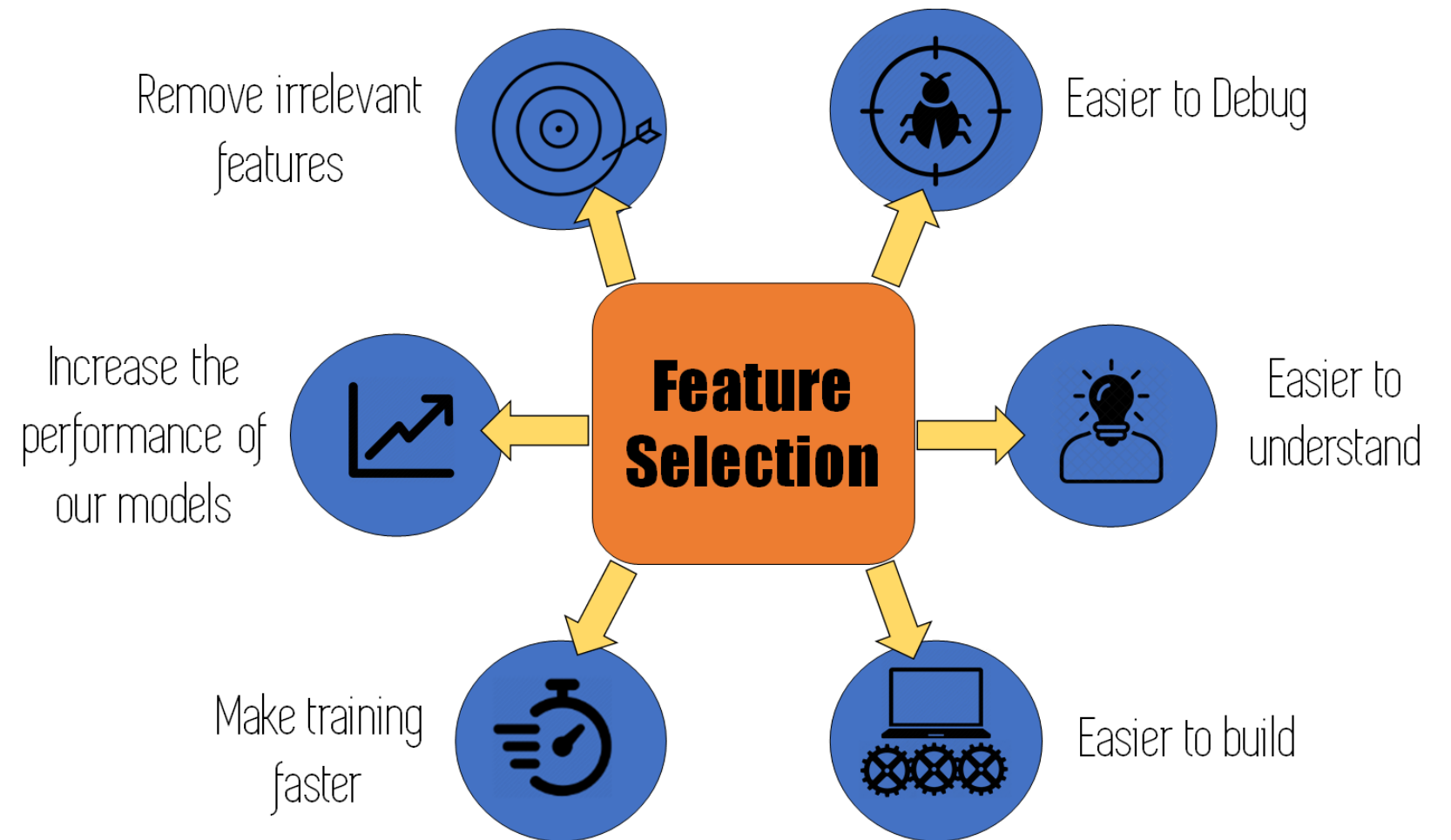
Fewer features mean faster model training

Reduces Overfitting

Too many features can cause the model to learn noise in the training data. By keeping only the most relevant features, the model focuses on the true patterns, not random fluctuations.

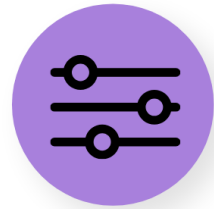
Improves Model Accuracy

Removing irrelevant or noisy features eliminates distractions for the model. A model trained on clean, informative features is more likely to make accurate predictions.



Feature selection is especially valuable when working with high-dimensional datasets where the signal-to-noise ratio can be challenging. challenging.

Supervised Feature Selection Techniques



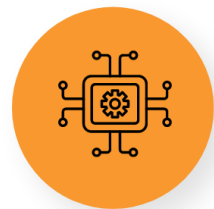
Filter-based Approach

Filter-based feature selection approaches are based on data intrinsic attributes such as feature correlation or statistics.



Wrapper-based Approach

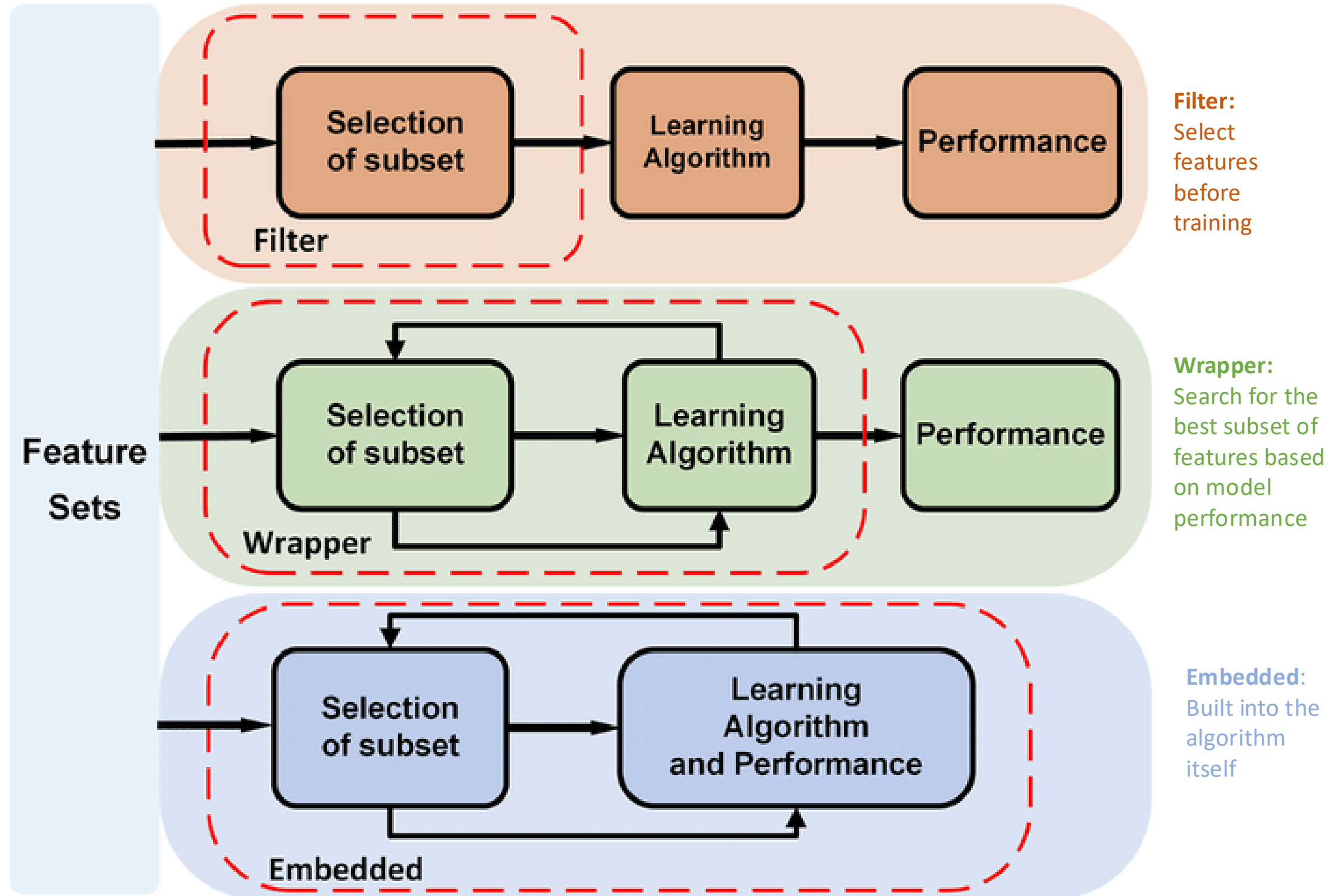
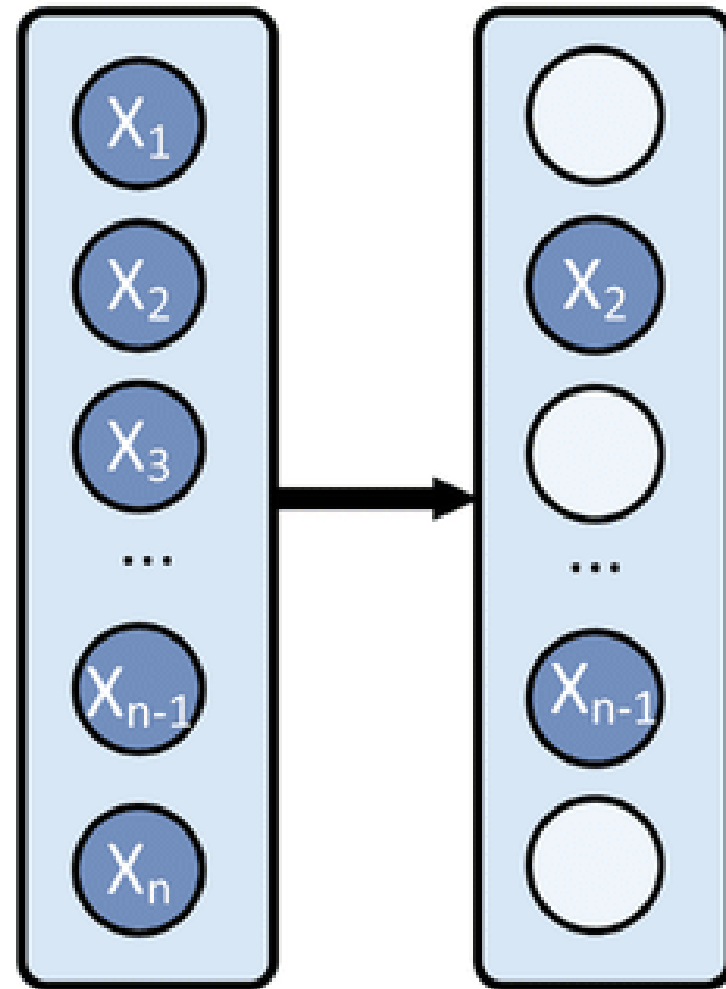
Wrapper-based feature selection approaches include assessing the importance of features using a specific machine learning algorithm.



Embedded Approach

Embedded feature selection approaches include the feature selection process as part of the learning algorithm.

Feature Selection



Popular Filtering Techniques

Pointwise Mutual Information (PMI)

Measures how much the presence of one feature tells us about another.

$$PMI(A = a, C = c) = \log_2 \frac{P(a, c)}{P(a)P(c)}$$

Mutual Information (MI)

Quantifies information gain between features and target variable.

$$MI(A, C) = \sum_{i \in \{a, \bar{a}\}} \sum_{j \in \{c, \bar{c}\}} P(i, j) \log_2 \frac{P(i, j)}{P(i)P(j)}$$

$0 \log_2 0$ is defined as 0

Chi-Square Test

Determines if features are independent independent of the target.

$$\chi^2 = \sum_{i \in \{a, \bar{a}\}} \sum_{j \in \{c, \bar{c}\}} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

- Independence: the following formula holds if attribute A is independent from class C

$$P(A, C) = P(A)P(C) \quad \begin{matrix} \text{Joint Prob.} \\ \text{Marginal Prob.} \end{matrix}$$

$P(C|A) = P(C)$

- If $\frac{P(A, C)}{P(A)P(C)} \gg 1$, attribute and class occur together much more often than randomly.
- If $\frac{P(A, C)}{P(A)P(C)} \approx 1$, attribute and class are independent, and they occur together as often as we would expect from random chance.
- If $\frac{P(A, C)}{P(A)P(C)} \ll 1$, attribute and class are negatively correlated.

Feature Selection and Feature Evaluation

$\mathbf{a_1}$	$\mathbf{a_2}$	\mathbf{c}
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

Which attribute, $\mathbf{a_1}$ or $\mathbf{a_2}$, is good?

Pointwise Mutual Information (PMI)

$$PMI(A = a, C = c) = \log_2 \frac{P(a, c)}{P(a)P(c)}$$

$P(a_1)$ means $P(a_1 = Y)$,
 Y is the “interesting” value
of a binary attribute

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

$$P(a_1) = \frac{2}{4}, P(c) = \frac{2}{4}, P(a_1, c) = \frac{2}{4}$$

$$PMI(a_1, c) = \log_2 \frac{\frac{1}{2}}{\frac{1}{2} \cdot \frac{1}{2}} = \log_2 2 = 1$$

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

$$P(a_2) = \frac{2}{4}, P(c) = \frac{2}{4}, P(a_2, c) = \frac{1}{4}$$

$$PMI(a_2, c) = \log_2 \frac{\frac{1}{4}}{\frac{1}{2} \cdot \frac{1}{2}} = \log_2 1 = 0$$

a_1 is better than a_2

Mutual Information (MI)

a₁	a₂	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

- Contingency Tables for toy example with attributes a_1 and a_2

a_1	$a = Y$	$a = N$	Total
$c = Y$	2	0	2
$c = N$	0	2	2
Total	2	2	4

a_2	$a = Y$	$a = N$	Total
$c = Y$	1	1	2
$c = N$	1	1	2
Total	2	2	4

MI for a_1

$$\begin{aligned}
 MI(A, C) &= P(a_1, c) \log_2 \frac{P(a_1, c)}{P(a_1)P(c)} + P(\bar{a}_1, c) \log_2 \frac{P(\bar{a}_1, c)}{P(\bar{a}_1)P(c)} + \\
 &\quad P(a_1, \bar{c}) \log_2 \frac{P(a_1, \bar{c})}{P(a_1)P(\bar{c})} + P(\bar{a}_1, \bar{c}) \log_2 \frac{P(\bar{a}_1, \bar{c})}{P(\bar{a}_1)P(\bar{c})} \\
 &= \frac{1}{2} \log_2 \frac{\frac{1}{2}}{\frac{1}{2} \cdot \frac{1}{2}} + 0 \log_2 \frac{0}{\frac{1}{2} \cdot \frac{1}{2}} + 0 \log_2 \frac{0}{\frac{1}{2} \cdot \frac{1}{2}} + \frac{1}{2} \log_2 \frac{\frac{1}{2}}{\frac{1}{2} \cdot \frac{1}{2}} \\
 &= \frac{1}{2} \cdot 1 + 0 + 0 + \frac{1}{2} \cdot 1 = 1
 \end{aligned}$$

$$MI(A, C) = \sum_{i \in \{a, \bar{a}\}} \sum_{j \in \{c, \bar{c}\}} P(i, j) \log_2 \frac{P(i, j)}{P(i)P(j)}$$

MI for a_2

$$\begin{aligned} MI(A, C) &= P(a_2, c) \log_2 \frac{P(a_2, c)}{P(a_2)P(c)} + P(\bar{a}_2, c) \log_2 \frac{P(\bar{a}_2, c)}{P(\bar{a}_2)P(c)} + \\ &\quad P(a_2, \bar{c}) \log_2 \frac{P(a_2, \bar{c})}{P(a_2)P(\bar{c})} + P(\bar{a}_2, \bar{c}) \log_2 \frac{P(\bar{a}_2, \bar{c})}{P(\bar{a}_2)P(\bar{c})} \\ &= \frac{1}{4} \log_2 \frac{\frac{1}{4}}{\frac{1}{2} \cdot \frac{1}{2}} + \frac{1}{4} \log_2 \frac{\frac{1}{4}}{\frac{1}{2} \cdot \frac{1}{2}} + \frac{1}{4} \log_2 \frac{\frac{1}{4}}{\frac{1}{2} \cdot \frac{1}{2}} + \frac{1}{4} \log_2 \frac{\frac{1}{4}}{\frac{1}{2} \cdot \frac{1}{2}} \\ &= 4 \cdot \frac{1}{4} \cdot 0 = 0 \end{aligned}$$

a_1 is better than a_2

Chi-Square Test

- Contingency Tables for toy example attribute a_1

Observed values	a_1	$a = Y$	$a = N$	Total
	$c = Y$	2	0	2
	$c = N$	0	2	2
	Total	2	2	4
Expected values (independent)	a_1	$a = Y$	$a = N$	Total
	$c = Y$	1	1	2
	$c = N$	1	1	2
	Total	2	2	4

χ^2 for a_1

$$\begin{aligned}
 \chi^2 &= \frac{(O_{a,c} - E_{a,c})^2}{E_{a,c}} + \frac{(O_{\bar{a},c} - E_{\bar{a},c})^2}{E_{\bar{a},c}} + \\
 &\quad \frac{(O_{a,\bar{c}} - E_{a,\bar{c}})^2}{E_{a,\bar{c}}} + \frac{(O_{\bar{a},\bar{c}} - E_{\bar{a},\bar{c}})^2}{E_{\bar{a},\bar{c}}} \\
 &= \frac{(2 - 1)^2}{1} + \frac{(0 - 1)^2}{1} + \frac{(0 - 1)^2}{1} + \frac{(2 - 1)^2}{1} \\
 &= 4
 \end{aligned}$$

Chi-Square Test

- Contingency Tables for toy example attribute a_2

Observed values	a_2	$a = Y$	$a = N$	Total
	$c = Y$	1	1	2
	$c = N$	1	1	2
	Total	2	2	4

Expected values (independent)	a_2	$a = Y$	$a = N$	Total
	$c = Y$	1	1	2
	$c = N$	1	1	2
	Total	2	2	4

- χ^2 for a_2

$$\chi^2 = \frac{(O_{a,c} - E_{a,c})^2}{E_{a,c}} + \frac{(O_{\bar{a},c} - E_{\bar{a},c})^2}{E_{\bar{a},c}} +$$

$$\frac{(O_{a,\bar{c}} - E_{a,\bar{c}})^2}{E_{a,\bar{c}}} + \frac{(O_{\bar{a},\bar{c}} - E_{\bar{a},\bar{c}})^2}{E_{\bar{a},\bar{c}}}$$

$$= 0$$

Higher χ^2 indicates dependency, so a_1 is more predictive than a_2

Q1

Given the following dataset, we wish to perform feature selection, where the class to predict is PLAY:

ID	Outlook	Temp	Humid	Wind	PLAY
A	S	H	H	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N

1. Which of $Humid = H$ and $Wind = T$ has the greatest *pointwise mutual information* with the class Y? What about class N?
2. Which of the attributes has the greatest mutual information for the PLAY class as a whole?

$$PMI(A, C) = \log_2 \frac{P(A \cap C)}{P(A)P(C)}$$

$$PMI(Humid = H, PLAY = Y) = \log_2 \frac{P(Humid = H \cap PLAY = Y)}{P(Humid = H)P(PLAY = Y)} = \log_2 \frac{(2/6)}{(4/6)(3/6)} = \log_2(1) = 0$$

$$PMI(Wind = T, PLAY = Y) = \log_2 \frac{P(Wind = T \cap PLAY = Y)}{P(Wind = T)P(PLAY = Y)} = \log_2 \frac{(0/6)}{(2/6)(3/6)} = \log_2(0) = -\infty$$

C. PMI(H=H, N):

- H = H and N: Rows A and B → Count = 2

$$P(H = H, N) = \frac{2}{6}, \quad P(N) = \frac{3}{6}, \quad P(H = H) = \frac{4}{6}$$

$$\text{PMI}(H=H, N) = \log_2 \left(\frac{2/6}{(4/6)(3/6)} \right) = \log_2(1) = 0$$

D. PMI(Wind = T, N):

- Wind = T and Play = N → Rows B, F → Count = 2
- Wind = T: Rows B and F → 2 total

$$P(Wind = T, N) = \frac{2}{6}, \quad P(Wind = T) = \frac{2}{6}, \quad P(N) = \frac{3}{6}$$

$$\text{PMI}(Wind=T, N) = \log_2 \left(\frac{2/6}{(2/6)(3/6)} \right) = \log_2(2) = 1$$

2. A general form of Mutual Information (MI) is as follows:

$$MI(X, C) = \sum_{x \in X} \sum_{c \in \{Y, N\}} P(c, x) PMI(x, c)$$

ID	Outlook	Temp	Humid	Wind	PLAY
A	S	H	H	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N

For *Outlook*, the formula is:

$$\begin{aligned}
 MI(Outlook) &= P(S, Y)PMI(S, Y) + P(O, Y)PMI(O, Y) + P(R, Y)PMI(R, Y) + P(S, N)PMI(S, N) + P(O, N)PMI(O, N) \\
 &\quad + P(R, N)PMI(R, N) \\
 &= \frac{0}{6} \log_2 \frac{(0/6)}{(2/6)(3/6)} + \frac{1}{6} \log_2 \frac{(1/6)}{(1/6)(3/6)} + \frac{2}{6} \log_2 \frac{(2/6)}{(3/6)(3/6)} + \frac{2}{6} \log_2 \frac{(2/6)}{(2/6)(3/6)} + \frac{0}{6} \log_2 \frac{(0/6)}{(1/6)(3/6)} + \frac{1}{6} \log_2 \frac{(1/6)}{(3/6)(3/6)} \\
 &= 0 + (0.1667)(1) + (0.3333)(0.4150) + (0.3333)(1) + 0 + (0.1667)(-0.5850) \\
 &= 0.541
 \end{aligned}$$

For *Temp*, the formula is:

$$\begin{aligned}
 MI(Temp) &= P(H, Y)PMI(H, Y) + P(M, Y)PMI(M, Y) + P(C, Y)PMI(C, Y) + P(H, N)PMI(H, N) + P(M, N)PMI(M, N) \\
 &\quad + P(C, N)PMI(C, N) \\
 &= \frac{1}{6} \log_2 \frac{(1/6)}{(3/6)(3/6)} + \frac{1}{6} \log_2 \frac{(1/6)}{(1/6)(3/6)} + \frac{1}{6} \log_2 \frac{(1/6)}{(2/6)(3/6)} + \frac{2}{6} \log_2 \frac{(2/6)}{(3/6)(3/6)} + \frac{0}{6} \log_2 \frac{(0/6)}{(1/6)(3/6)} + \frac{1}{6} \log_2 \frac{(1/6)}{(2/6)(3/6)} \\
 &= (0.1667)(-0.5850) + (0.1667)(1) + (0.1667)(0) + (0.3333)(0.4150) + 0 + 0 \\
 &= 0.208
 \end{aligned}$$

The Mutual Information for Humid is 0, and for Wind is 0.459

Consequently,

Outlook appears to be the **best attribute** to predict PLAY

Wind also seems quite good

Temp is not a very good predictor of PLAY

Humid is completely unhelpful.

Wrapper-Based Feature Selection

Choose subset of attributes that give best performance on the validation data



Forward Selection

Start with no features, add one at a time based on performance.
performance.



Backward Elimination

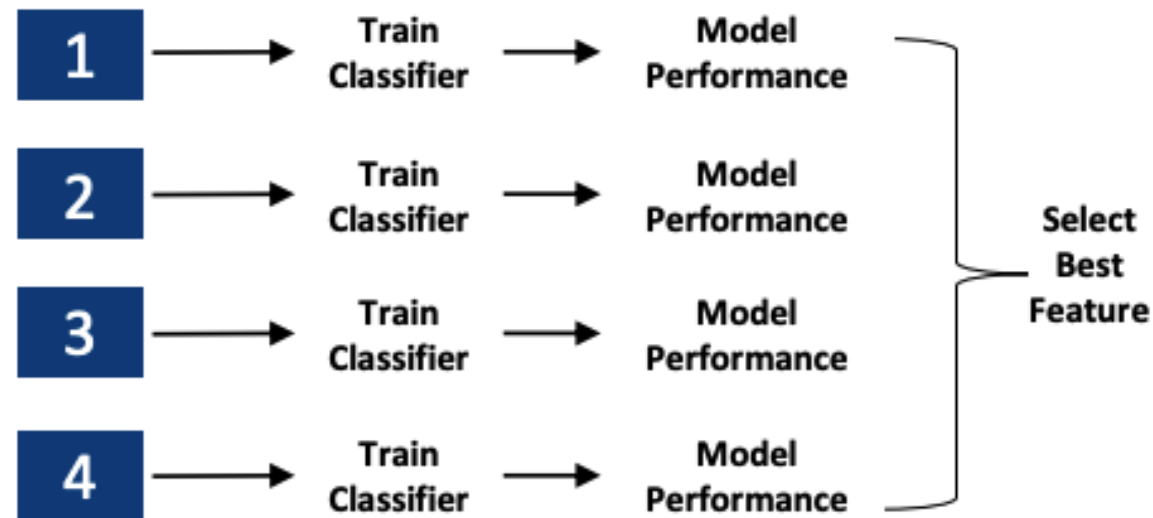
Start with all features, remove least important one by one.

- **Greedy Approach: Sequential Forward Selection**

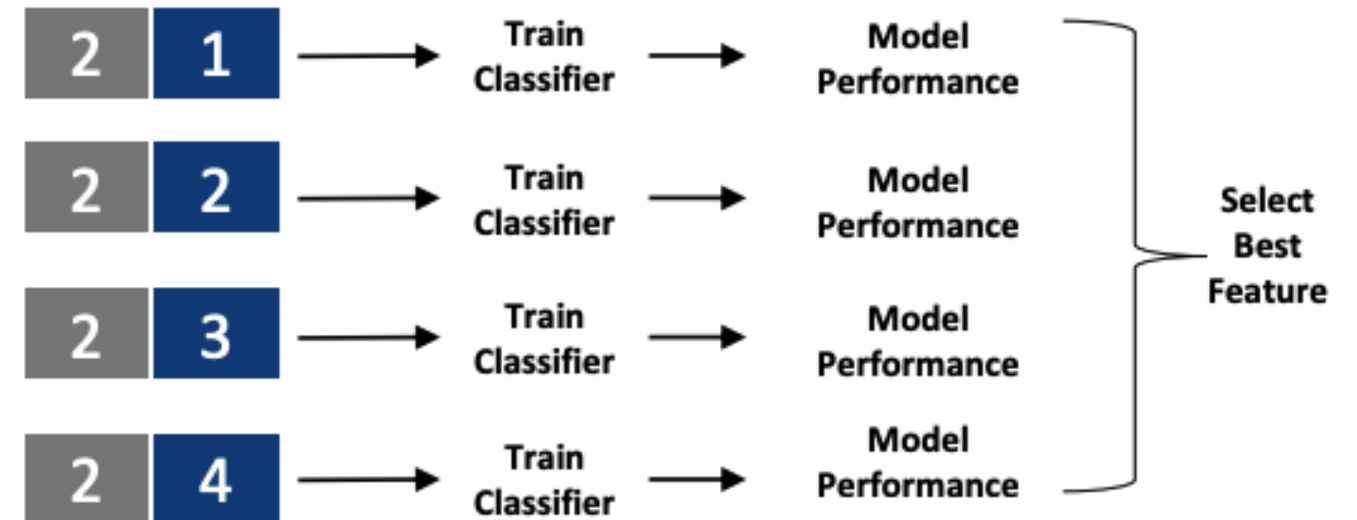
Original Feature Set

1 2 3 4

Round 1



Round 2



Round 3

Round 4

...

Until the termination condition is met or maximum number of features reached.

• Ablation Approach: Sequential backward selection

- 1 Start with original feature set $n=10$.

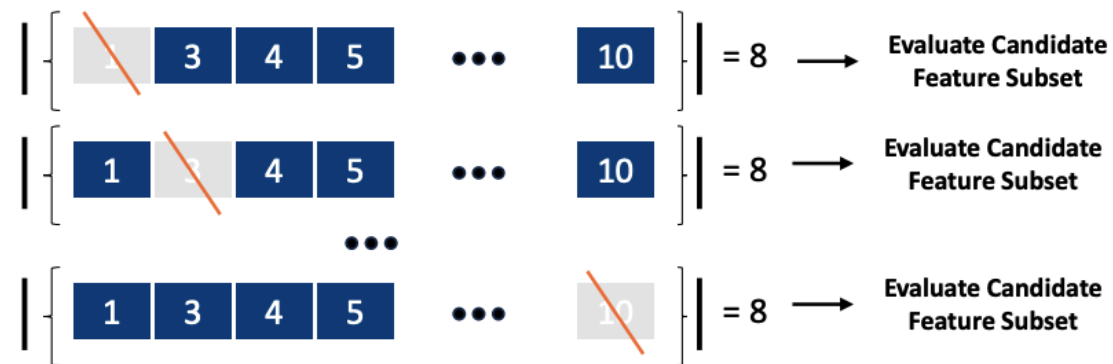
$$\left| \left[\begin{array}{cccc} 1 & 2 & 3 & 4 & \dots & 10 \end{array} \right] \right| = 10$$

- 2 **Iteration 1** – Generate all possible feature subsets of size $n=10-1$



- 3 Remove the Feature that is absent from the subset with highest evaluation score.
Suppose Subset 2 corresponds to the highest evaluation score.

- 4 **Iteration 2** Generate all possible feature subsets of size $(n-1) - 1 = 8$

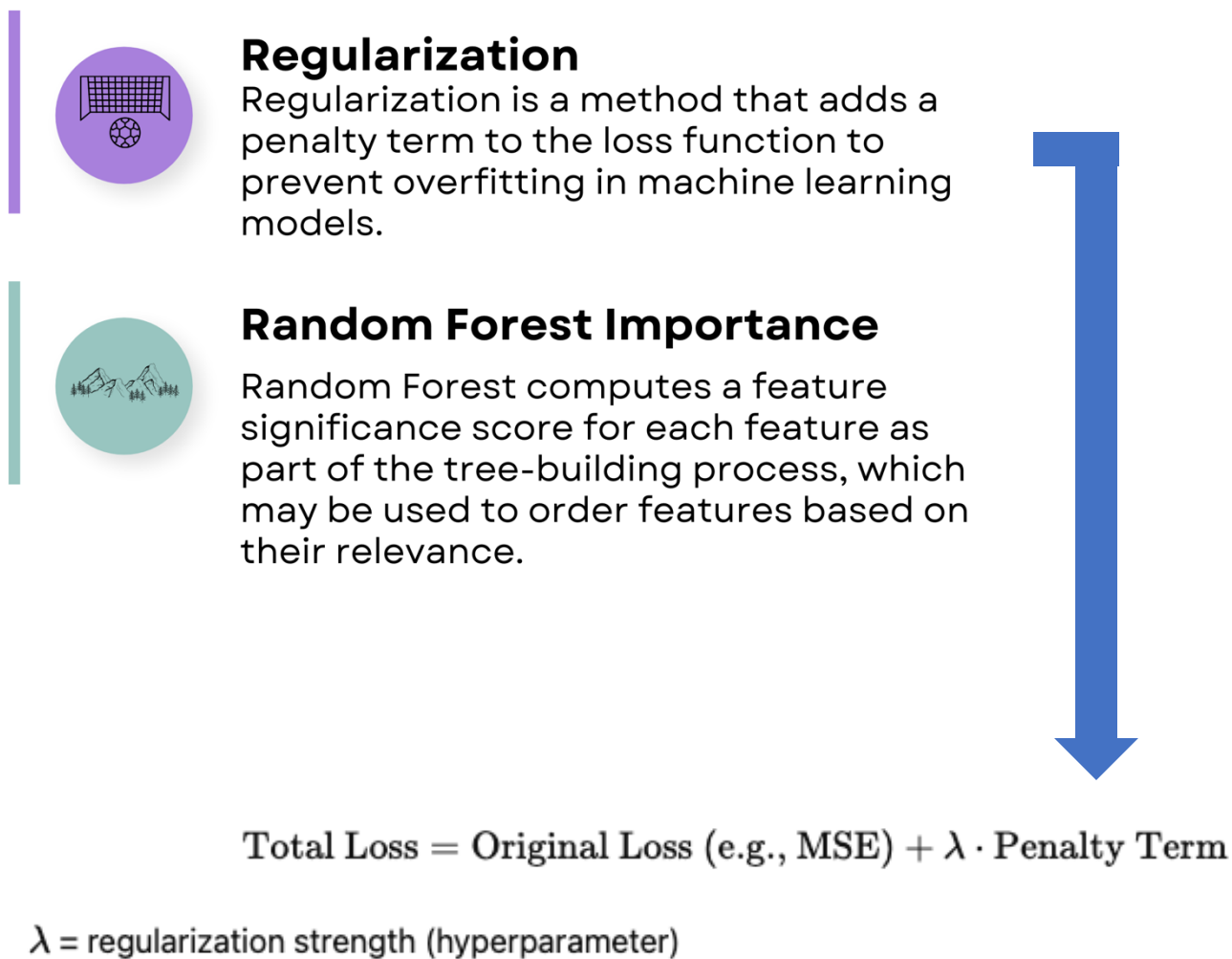


- 4 Repeat Step 3 and 4, until the subset contains only single feature
- 5 Considering all the iterations $1 \dots n-1$, the subset with highest evaluation score is selected as the final feature subset.

Embedding Methods

Model Training	Feature selection is done as part of the the model training process. Ex: random random forest, decision tree
Automatic Selection	The model automatically selects features by assigning importance scores while fitting the data
Optimized Output	Final model includes only significant features

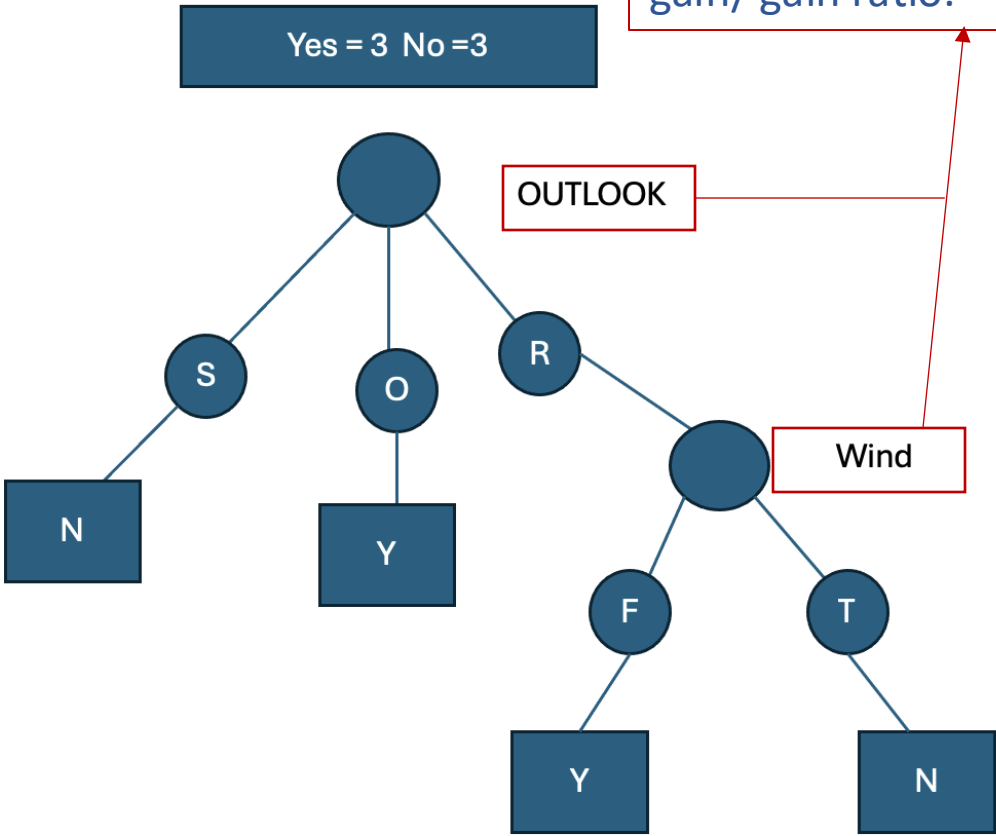
Embedded Approach



Method	Description
Lasso (L1 Regularization) $\lambda \sum w_i $	Shrinks some coefficients to zero → those features are eliminated
Ridge (L2 Regularization) $\lambda \sum w_i^2$	Penalizes large coefficients → reduces overfitting (but doesn't eliminate features)

Embedding Methods – Decision Tree

ID	Outlook	Temp	Humid	Wind	Play
A	S	H	H	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N



Comparing Selection Methods

Method	Speed	Accuracy	Use Case
Filter	Very Fast	Moderate	Large datasets, quick exploration exploration
Wrapper	Slow	High	Critical accuracy needs, smaller datasets
Embedding	Moderate	High	Balance of speed and accuracy

Model Evaluation: Bias and Variance

Bias

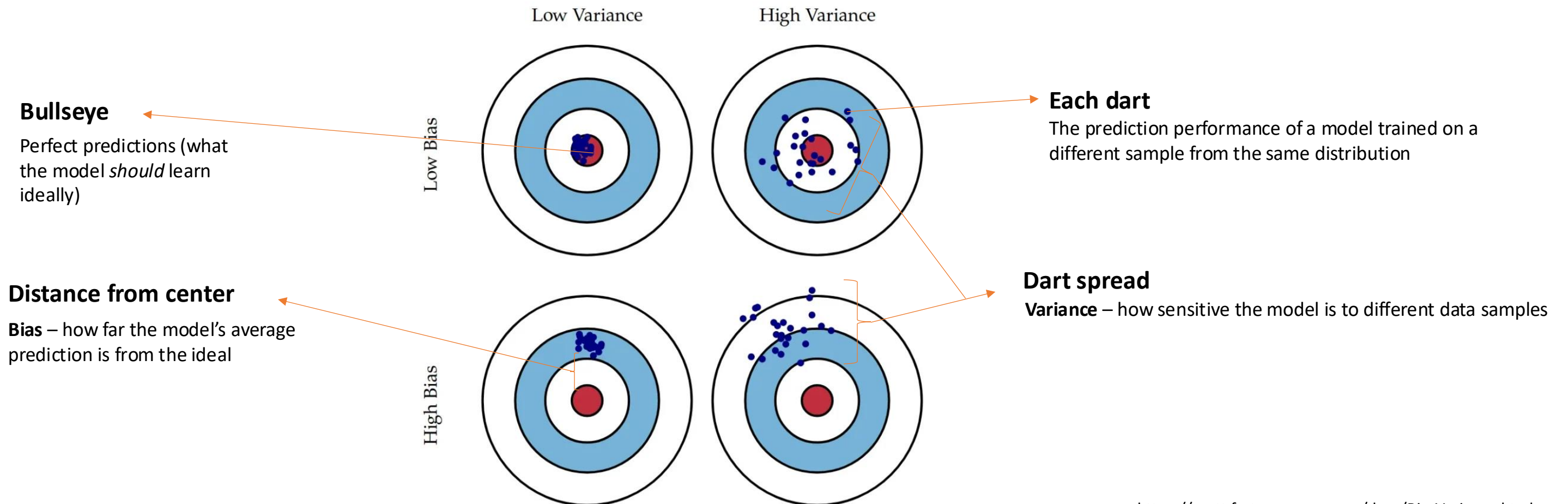
Error from wrong assumptions in the learning algorithm.

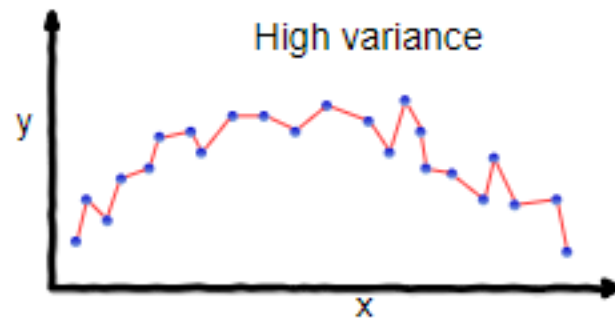
High bias leads to underfitting. The model misses relevant patterns.

Variance

Error from sensitivity to small fluctuations in training data.

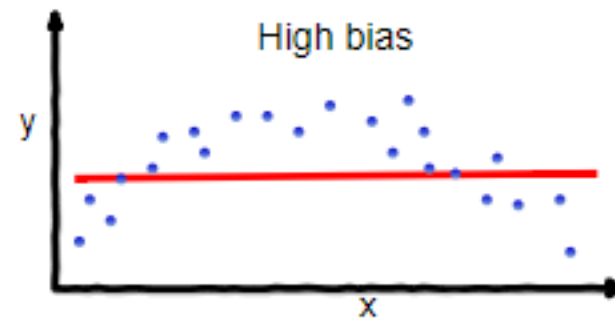
High variance causes overfitting. The model captures noise as signal.





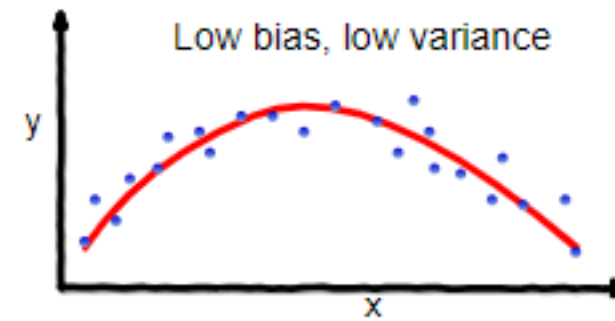
overfitting

Model is too complex.
It captures noise as signal.
Too many features causes great training performance but poor test results.



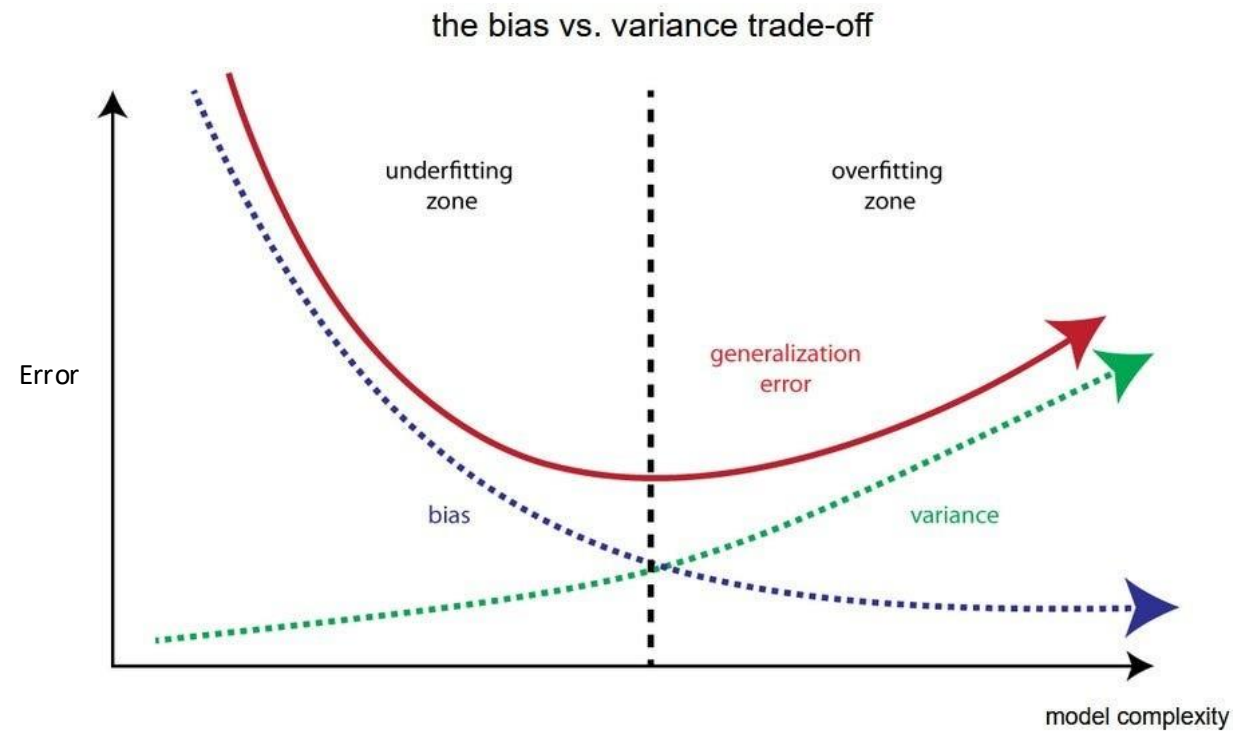
underfitting

Model is too simple.
It misses important patterns and relationships.
Insufficient features or complexity causes high error on both training and test data.

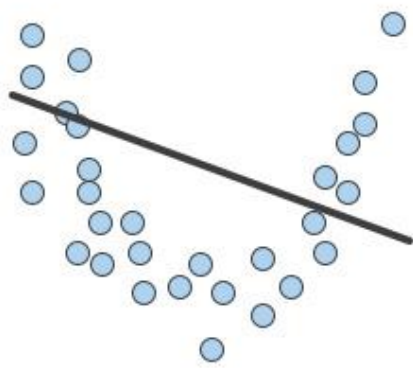
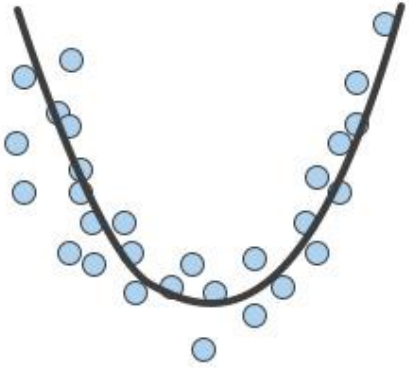

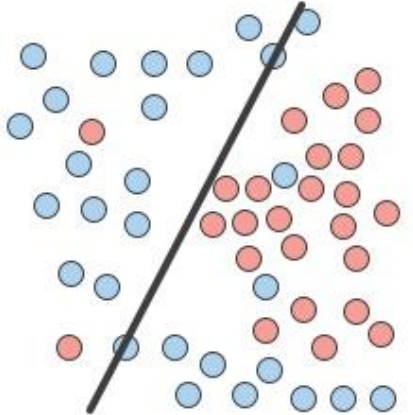
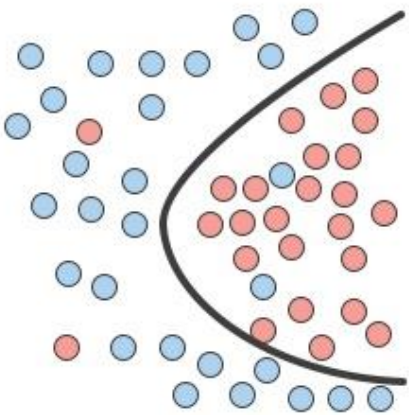
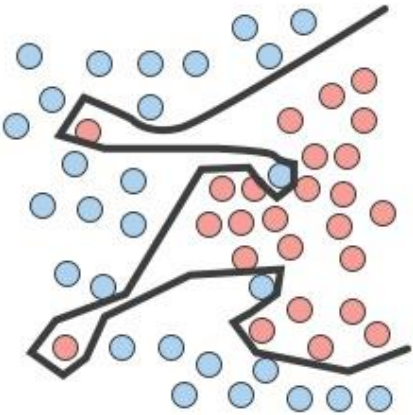

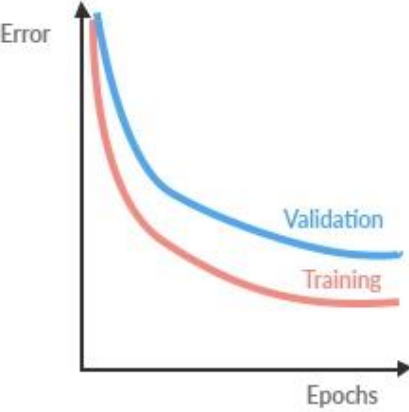
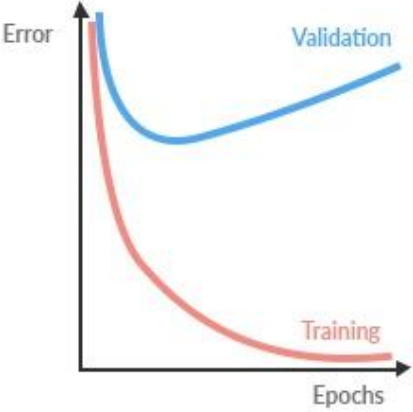


Good balance

Model captures true patterns.
It generalizes well to new data.
The right balance of features leads to good performance on test data.



Overfitting vs Underfitting

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"> • High training error • Training error close to test error • High bias 	<ul style="list-style-type: none"> • Training error slightly lower than test error 	<ul style="list-style-type: none"> • Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			

Q2

1. What is the difference between model **bias** and model **variance**?
2. Describe the behaviour of a classifier with high bias and low variance.
3. Describe the behaviour of a classifier with low bias and high variance.

1. **Model bias** is the propensity of a classifier to systematically produce the same errors. For example, "bias towards the majority class" - when the model predicts too many instances as the majority class.

If it doesn't produce errors, it is unbiased

If it produces random errors, it is also considered unbiased.

Model variance is the propensity of a classifier to produce different classifications using different training sets (randomly sampled from the same population). It is a measure of the inconsistency of the classifier, from training set to training set.

2. A classifier with **high bias** and **low variance** will be **consistently wrong (underfitting)** -- it will have a systematic error that makes its predicted labels different from the true labels, but relatively little random error.

3. A classifier with **low bias** and **high variance** will also make a lot of errors, but the **errors will not be consistent (overfitting)**. The classifier will make different types of errors depending on the training dataset; the error rate might be low on one set of data but high on another. The distribution of predictions should match the distribution of true labels (since the classifier is unbiased) , but which instances are assigned to which labels may be quite variable.

Evaluation Bias and Model Bias

Where the bias is introduced (training vs. testing) determines whether it's **model** or **evaluation** bias.

Model bias refers to the systematic error in the modelling/training process that results in a model that is unable to capture the true underlying relationship between the input features and the target variable.

Model bias can be caused by various factors, such as the choice of the model architecture, the selection of features, or the assumptions made by the model.

Model bias can be addressed by improving the modelling process, such as selecting more appropriate features or using a more complex model architecture.

Evaluation bias refers to the systematic error in the evaluation of a model that results in consistently overestimating or underestimating the true performance of the model.

Evaluation bias can be caused by various factors, such as the choice of evaluation metric, the sampling bias in the data used for evaluation, or the inappropriate assumptions made by the evaluator.

Evaluation bias can be mitigated by selecting appropriate evaluation metrics and data sampling techniques that are less biased and more representative of the model's true performance.

Evaluation Bias and Model Bias

Situation	Type of Bias
Training on imbalanced or skewed data	Model Bias
Testing on unrepresentative data	Evaluation Bias
Model performs poorly on unseen groups	Model Bias
Accuracy appears high due to easy test	Evaluation Bias

Scenario 1:

A disease detection model is trained only on data from adult patients aged 30–50. When tested on elderly patients, the model performs very poorly.

– Model bias

Scenario 2:

A gender classification model is trained on equal male and female images but evaluated only on images of males.

– Evaluation bias

Q4

Explain the difference between **evaluation bias** and **model bias**.

Q5

During training process, your model shows significantly different performance across different training sets.

1. What can be the reason?
2. How can we solve the issue?

1. When model performance changes significantly with small changes in the training set the model has **high variance**, or in other words has "**overfit**" the training data.
2. There are a few remedies to reduce variance of a model. Overfitting happens when the model is too complex relative to the size of the training dataset, so reducing the complexity of the model can help. This can be done through **feature selection** or **regularisation**. Alternatively, overfitting can be reduced by **increasing the number of training instances**, although this can may be difficult to do in real world problems where the training data may be limited.

Feature selection helps reduce overfitting by removing **irrelevant, redundant, or noisy** features that can mislead the model.

Regularization reduces overfitting by **penalizing large or complex model parameters**, forcing the model to **simplify itself** and focus on the most important patterns in the data.

Combating Overfitting

K-Fold Cross-Validation

Divide data into k subsets (folds)

Train on k-1 folds, validate on the remaining fold

Rotate and repeat k times

A model might perform great on the training set but poorly on some folds. Cross-validation exposes this by checking **multiple train-test splits**.

Bagging

Train multiple models on random data subsets

Combine predictions (usually by voting/averaging)

Example: Random Forest algorithm

Reduces variance: Single trees are unstable and prone to overfitting; bagging averages out their noise.

Q3

Explain how these strategies help reduce model **overfitting**:

- 1. Use of a validation set (e.g., cross-validation)
- 2. Model ensembling (e.g., random forests)

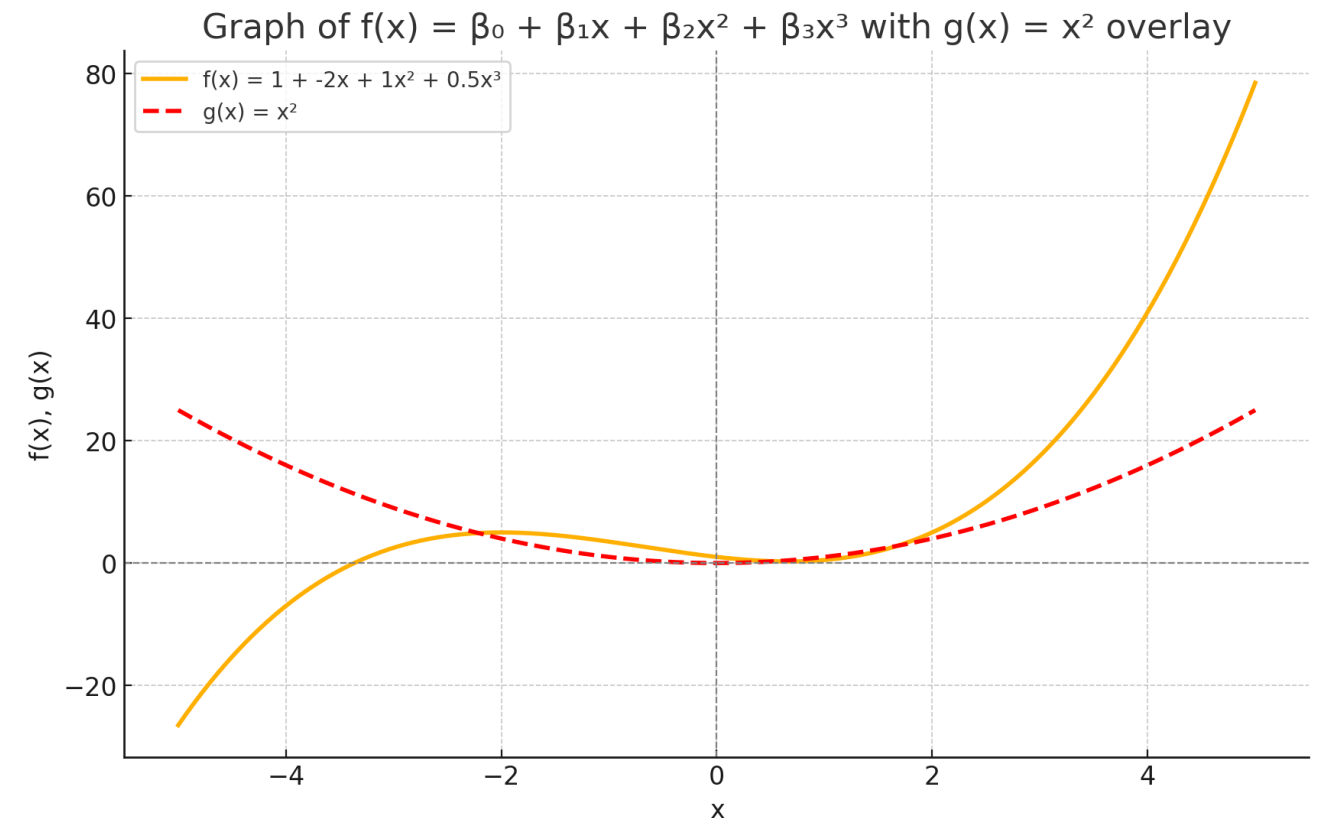
1.	Aspect	Explanation
	Early warning system	A validation set gives you feedback on how well your model is doing on unseen data . If training accuracy is high but validation accuracy is low → your model is overfitting.
	Guides model tuning	Helps you tune hyperparameters like tree depth, learning rate, number of layers, etc., to prevent over-complex models .
	Cross-validation advantage	Techniques like K-Fold Cross-Validation use multiple validation sets, reducing the risk of relying on a single "lucky" split. This leads to more robust and generalizable models .
2.	Technique	How It Helps
	Bagging (used in Random Forests)	Trains each model on a different subset of data (via bootstrapping), reducing model variance. It averages out overfitted predictions from any one model.
	Random Forests	Use multiple decision trees, each seeing different data/features. This prevents any single tree from overfitting , and final predictions are made by majority vote or averaging .
	Diverse Models	Since each model overfits in different ways , combining them cancels out individual errors and focuses on stable patterns.

Q6

Suppose you are given a dataset with single feature x and label y generated by a function of the form $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$. You intend to fit a regression model to this data. If your regression model involves polynomial terms up to x^2 , it will likely have **{low, high} bias** and **{low, high} variance**. (Select the correct word in each pair.)

x^2 doesn't have enough capacity to capture the full complexity of the data represented by $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$

The model will likely have **high bias** and **low variance**. The proposed model is not complex enough to fit the dataset and would probably **underfit**.



From the sample graphical representation above, we can see that x^2 **fit y poorly**, especially in regions where the cubic term has a significant effect