LIB AI Session - 3

Date - 7th June, 2025

Introduction to Classification

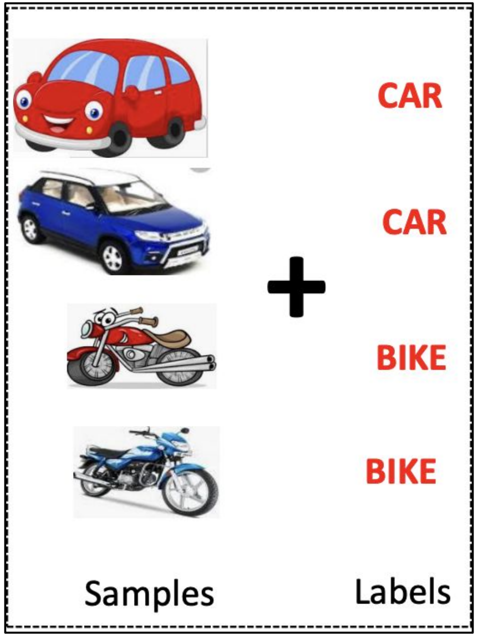
Presented by -
Deepanshi Seth
Lead Data Scientist
Games24x7 Pvt Ltd

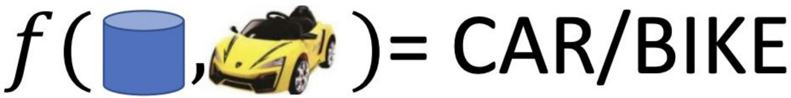
Today’s Agenda

- What is Classification?
- Key metrics for Classification
  - Confusion Matrix
  - Accuracy
  - Precision - Recall
  - ROC Curve (discussed after Logistic Regression)
- Algorithms for Classification
  - k-NN
  - Logistic Regression
  - Decision Trees
  - Random Forest
  - SVM
- Hands-on Exercise

What is Classification?

- ➢ Type of Supervised Learning (for each input in training data, output is known)
- ➢ Objective - Predict the category labels (discrete or nominal outputs)





Given a dataset D = {x1, x2,..xn} and a set of Class labels C = {c1, c2,...ck}, the task of classification is devise a mapping function f : D -> C

What is Classification?

- Binary vs Multi-Class Classification
- Typical applications -
  - Credit/Loan approval
  - Medical diagnosis - if a tumor is cancerous or benign
  - Fraud detection
  - Customer churn
  - Sentiment Category Analysis (Email categorisation)

Key Evaluation Metrics

Confusion Matrix

Actual / Predicted Class	Class 0 (Negative)	Class 1 (Positive)
Class 0	True Negative (TN)	False Positive (FP)
Class 1	False Negative (FN)	True Positive (TP)

- **Accuracy** : Percentage of rows correctly classified
  (TP + TN) / All
- **Error Rate** : 1- Accuracy
  (FP + FN) / All
- **Sensitivity** : True Positive recognition rate
  TP / P
- **Specificity** : True Negative recognition rate
  TN / N

Key Evaluation Metrics

Actual / Predicted Class	Class 0 (Negative)	Class 1 (Positive)	Total
Class 0	2588 (TN)	412 (FP)	3000
Class 1	46 (FN)	6954 (TP)	7000
Total	2634	7366	10000

- **Accuracy : (TP + TN) / All** (6954+2588)/10000 = 95%
- **Error Rate : (FP + FN) / All** (412+46)/10000 = 5%
- **Sensitivity : TP / P** 6954/7000 = 99%
- **Specificity : TN / N** 2588/3000 = 86%

Actual / Predicted Class	Class 0 (Negative)	Class 1 (Positive)	Total
Class 0 - non fraud	9900 (TN)	0 (FP)	9900
Class 1 - fraud	100 (FN)	0 (TP)	100
Total	10000	0	10000

Actual / Predicted Class	Class 0 (Negative)	Class 1 (Positive)	Class 2	Total
Class 0 - non fraud	9900 (TN)	0 (FP)		9900
Class 1 - fraud	100 (FN)	0 (TP)		100
Class 2				
Total	10000	0		10000

Accuracy can be misleading!

Detect Fraudulent Transactions

- ➢ Total Transactions = 10000
- ➢ Fraudulent Transaction = 100
- ➢ Fraud occurs only in 1% of the data (High Class Imbalance)

Model Performance

- ➢ Predicted Fraud = 0
- ➢ Predicted Non Fraud = 10000

Accuracy = Correct Predictions / Total Predictions = 9900/10000 = **99% (WOW!!)**

But the model never detected fraud

So despite high accuracy, the model is useless!

Key Evaluation Metrics

Confusion Matrix

Actual / Predicted Class	Class 0 (Negative)	Class 1 (Positive)
Class 0	True Negative (TN)	False Positive (FP)
Class 1	False Negative (FN)	True Positive (TP)

- ➢ **Precision** : What % of data that classifier predicted as Positive is actually Positive
  TP/(TP+FP)
- ➢ **Recall** : What % of Positive data did the classifier label as Positive
  TP/(TP+FN)
- ➢ **F1 Score** : Harmonic mean of Precision and Recall
  2*Precision*Recall/(Precision+Recall)
- ➢ **Weighted F Score** : Assign β times more weight to Recall as to Precision

$$F_\beta = (1+\beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision} + \text{Recall})}$$

- β>1 : Recall is more important
- β=1 : Balanced Precision and Recall
- β<1 : Precision is more important

Key Evaluation Metrics

Actual / Predicted Class	Class 0 (Negative)	Class 1 (Positive)	Total
Class 0	2588 (TN)	412 (FP)	3000
Class 1	46 (FN)	6954 (TP)	7000
Total	2634	7366	10000

- **Precision** : **TP/(TP+FP)** 6954/(6954+412) = 94%
- **Recall** : **TP/(TP+FN)** 6954/(6954+46) = 99%
- **F1 Score** : 2*Precision*Recall/(Precision+Recall) 2*0.94*0.99/(0.94+0.99) = 96%

Choosing between Precision and Recall

Depends on the specific application

- ➢ **In a Medical Diagnosis System - High Recall could be crucial**

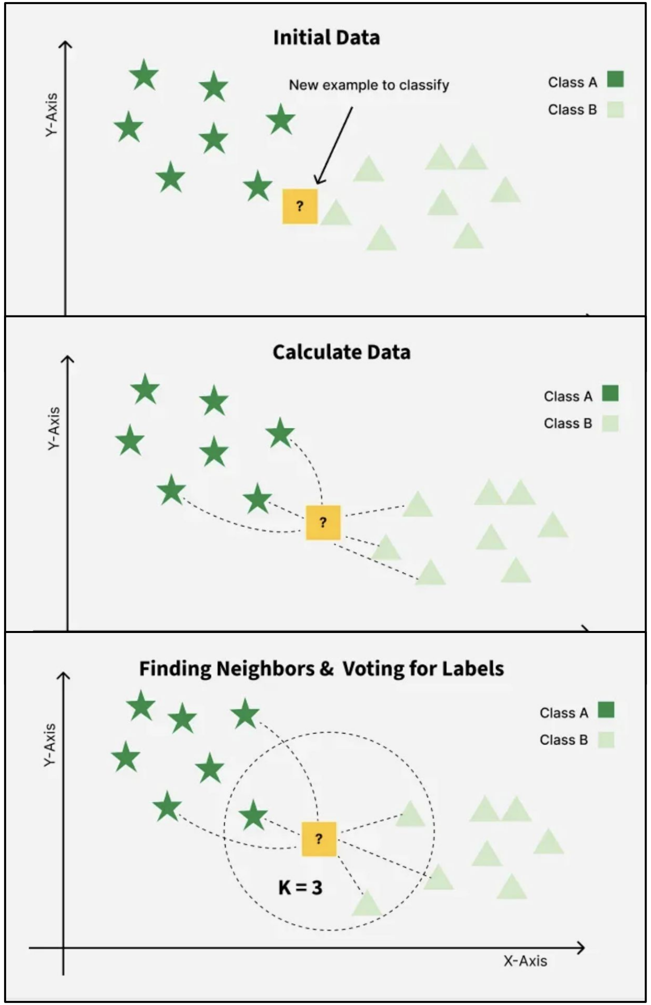
Catching as many positive cases (diseases) as possible, even if it leads to some false positives (unnecessary tests)

- ➢ **In a Financial Fraud Detection System - High Precision could be crucial**

Minimizing false positives (wrongly declined transactions) to avoid inconvenience to the customers

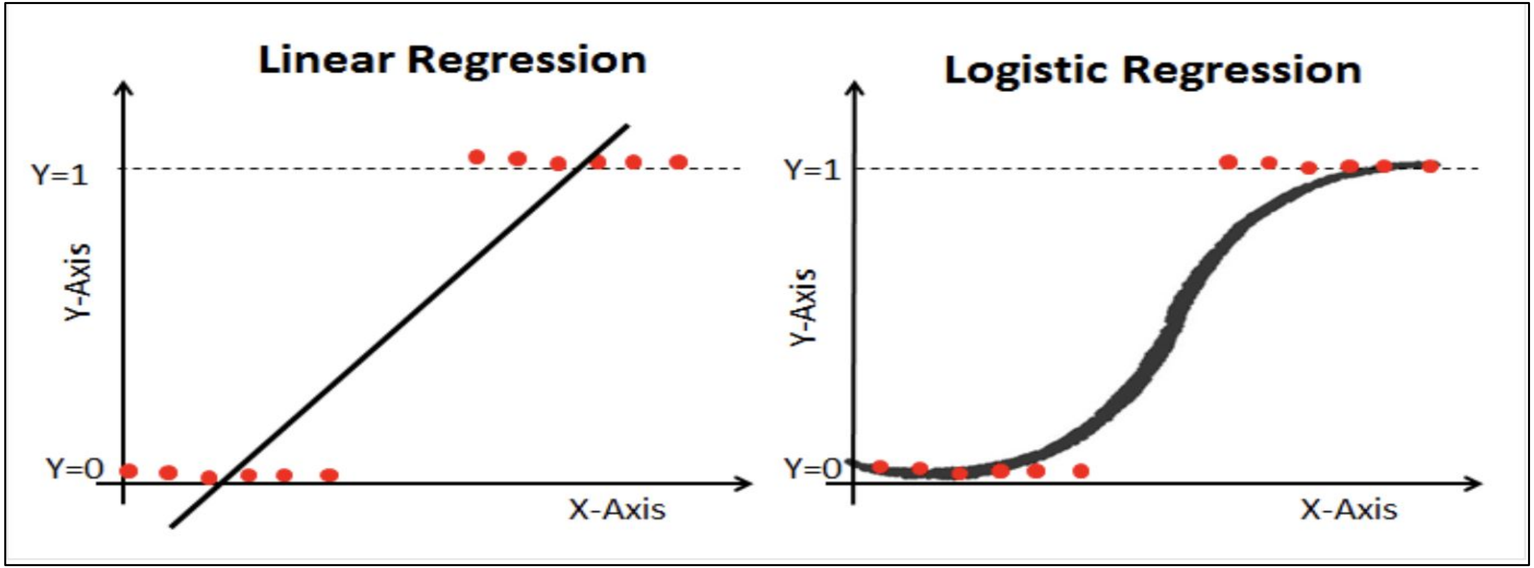
k-NN (k Nearest Neighbours)

- Lazy learner algorithm
  - Compute distance from test point to all training points
  - Select k closest (nearest) neighbours
  - Use majority vote for classification
- Key Hyperparameter - k
- Small k -> noisy, Large k -> stable (can miss patterns)
- Common Distance metrics - Euclidean (default), Cosine
- **Advantages -**
  - Simple to use
  - No training
  - Few parameters
- **Disadvantages -**
  - Slow with large data
  - Struggles with many features
  - Can overfit



Logistic Regression

- ➢ Used for Binary classification (label = 0 or 1)
- ➢ Special case of Linear Regression where target variable is categorical in nature
- ➢ Predicts the probability of occurrence of a binary event using logit function

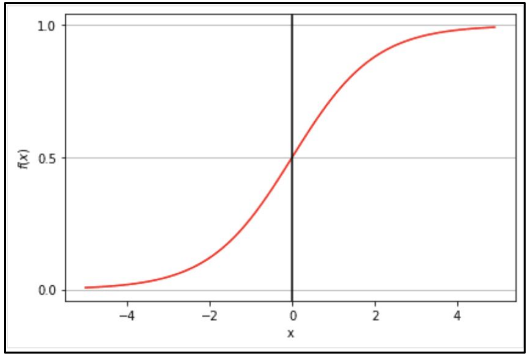


Logistic Regression

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots.. + \beta_n X_n$$

Sigmoid Function -

$$p = \frac{1}{1 + e^{-y}}$$



- Sigmoid Function (also called as Logistic function), gives an ‘S’ shaped curve that can take any real number and map it to a value between 0 and 1
- If the curve goes to positive infinity, y will be predicted 1
- If the curve goes to negative infinity, y will be predicted 0
- Usually 0.5 is considered as the threshold

Logistic Regression

Pros -

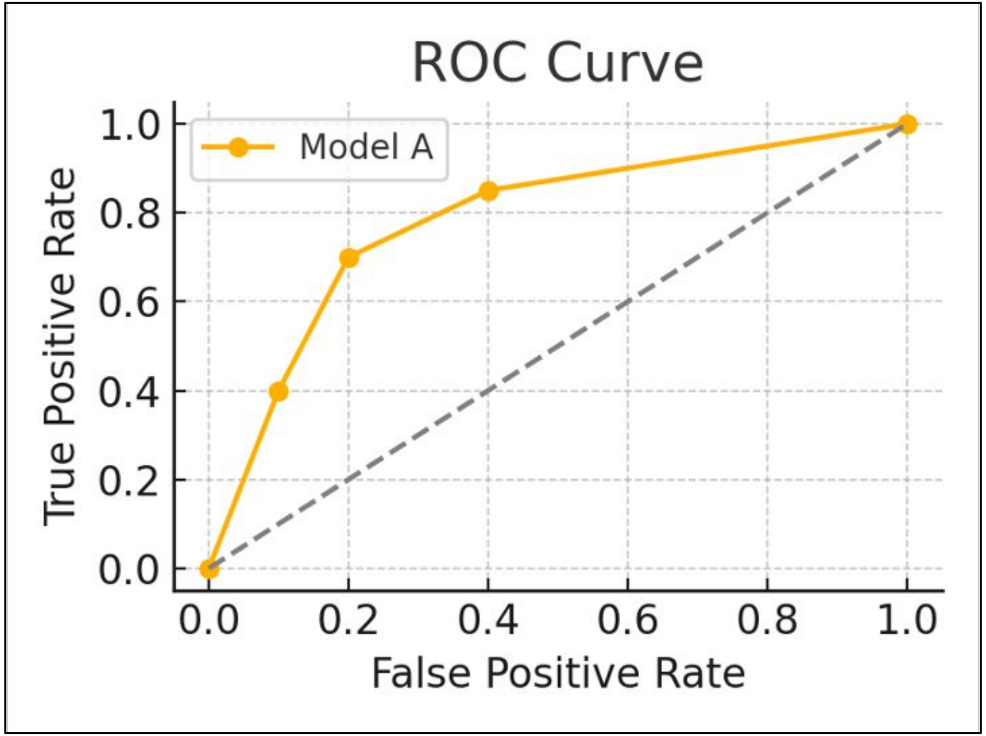
- ➢ Does not require high computation power
- ➢ Highly interpretable
- ➢ Easy to implement

Cons -

- ➢ Not able to handle large number of categorical features
- ➢ Cannot handle collinear features
- ➢ Cannot handle non-linear separations

https://mlu-explain.github.io/logistic-regression/

AUC - ROC Score



ROC Curve - TPR vs FPR at various thresholds

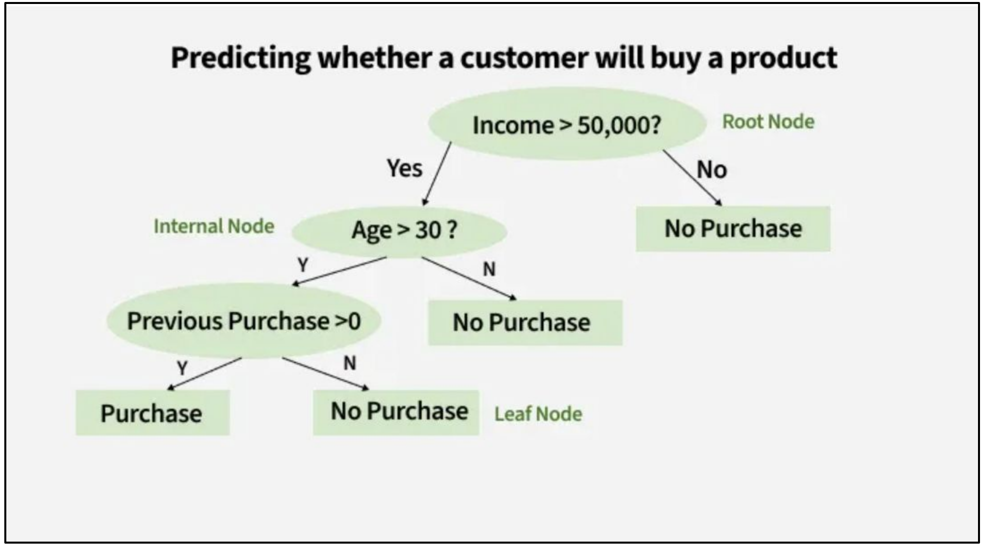
AUC - Area under the curve

- Measures the model's ability to distinguish between classes
- Ranges from 0(worst) to 1(perfect)
- Higher AUC - better model performance
- AUC = 0.5 implies random guessing

Top Left point is the optimal threshold - Maximise TPR and Minimise FPR

Decision Tree

- Constructs a flowchart like structure where -
  - Internal nodes represent feature based decision rules
  - Leaf nodes represent outcome labels
- Recursive Partitioning
  - Repeats splits until a stopping criterion is met (eg max depth, min samples)
- Prediction
  - Traverse from root to a leaf based on the feature values



Decision Tree - Splitting Criteria

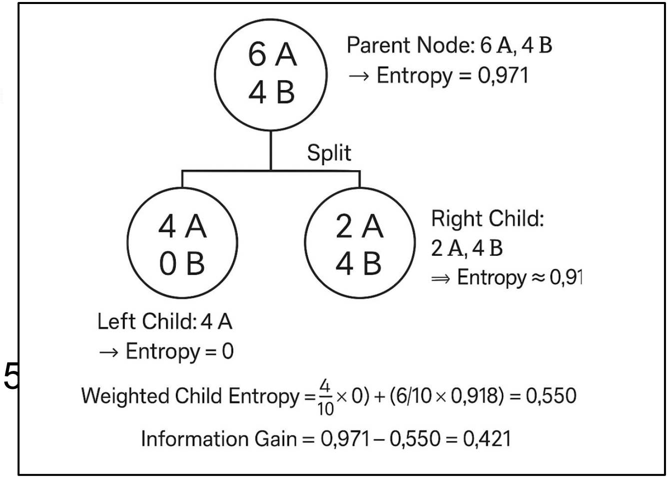
- **Gini Impurity**

  $$Gini = 1 - \sum_{i=1}^{C} p_i^2$$

  - Lower Gini = purer node
  - Example, Class A = 6, Class B = 4 -> Gini = 1 - (0.6*0.6 + 0.4*0.4) = 0.48
- **Entropy and Information Gain**

  $$Entropy = -\sum p_i \log_2(p_i) \; IG = Entropy_{parent} - \sum_k \frac{n_k}{n} Entropy_k$$

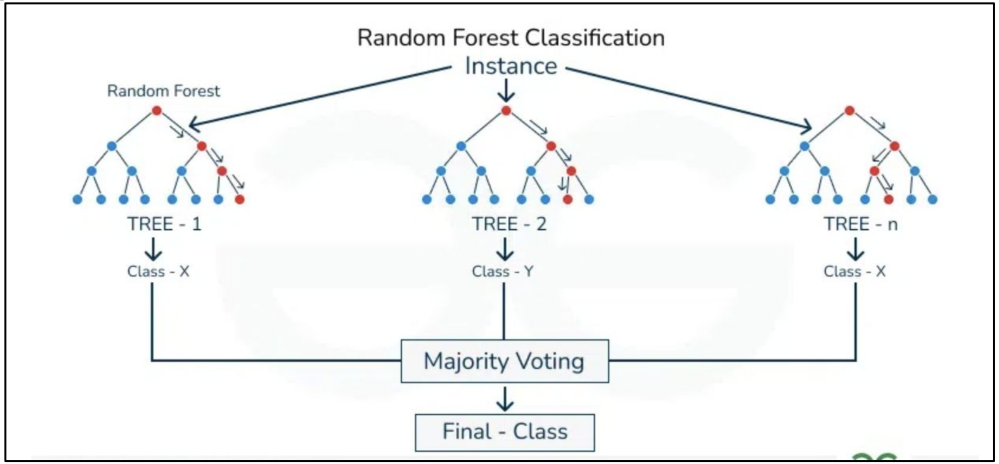
  - Parent Node: 6A, 4B -> Entropy = 0.971
  - Split:
    - Left Child: 4A, 0B -> Entropy = 0
    - Right Child: 2A, 4B -> Entropy = 0.918
  - Weighted Child Entropy = (4/10 × 0) + (6/10 × 0.918) = 0.5
  - **Information Gain** = 0.971 - 0.550 = 0.421



https://mlu-explain.github.io/decision-tree/

Random Forest

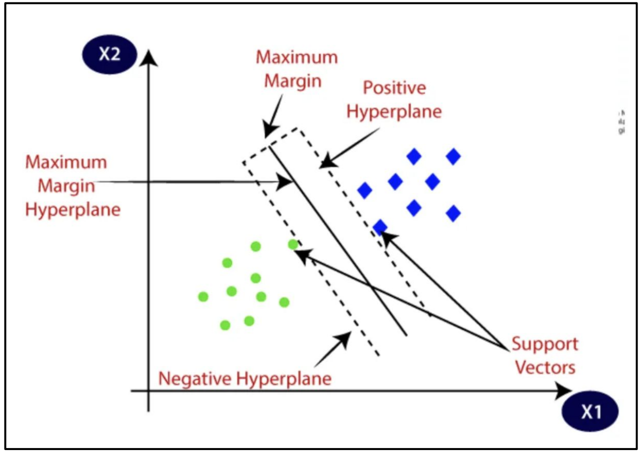
- ➢ Ensemble method that combines predictions from multiple decision trees
- ➢ Each tree is trained on random sample + random feature subset
- ➢ Prediction - Majority vote for classification
- ➢ Pros -
  - ○ More robust than single trees
  - ○ Reduces variance (overfitting)
  - ○ Can handle large feature sets
- ➢ Cons -
  - ○ Slower to predict compared to a single tree
  - ○ Less interpretable

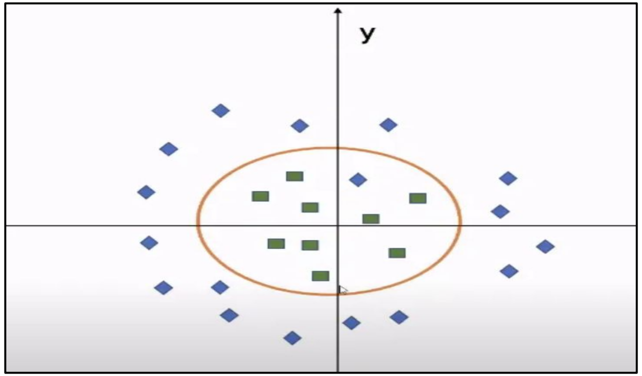


Will be covered in detail in one of the future sessions

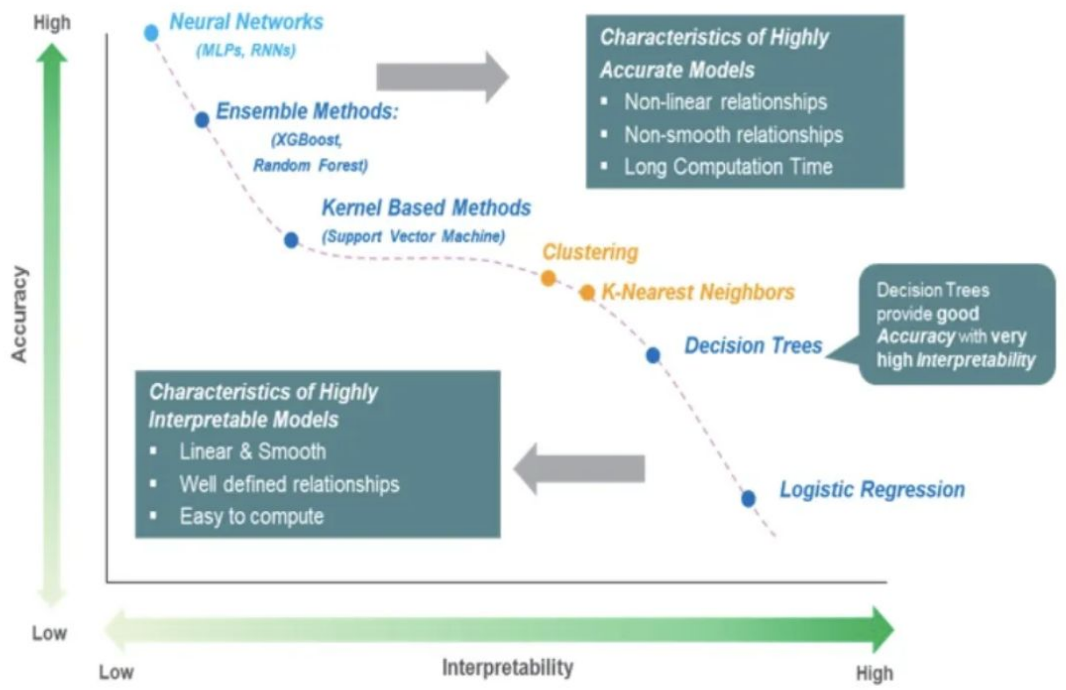
SVM (Support Vector Machine)

- ➢ Constructs a maximum margin hyperplane to separate classes
  - ○ **Hyperplane** - A boundary that separates classes
  - ○ **Margins** - Distance between the support vectors and hyperplanes
  - ○ **Support Vectors** - Critical points that influence the boundary
- ➢ **Linear SVM** - works when data is linearly separable
- ➢ **Kernel SVM** - maps data to higher dimensions using kernels
- ➢ Objective Function - **Maximise margin while minimising classification**
- ➢ Pros -
  - ○ Powerful in high-dimensional spaces
  - ○ Works well for clear margin separation
- ➢ Cons -
  - ○ Not suited for large datasets (slow training)
  - ○ Requires tuning of kernel and regularisation





Model Performance vs Interpretability



Summary

- **Logistic Regression**: Simple, interpretable
- **Decision Tree**: Interpretable, may overfit
- **Random Forest**: Accurate, less interpretable
- **SVM**: Powerful, needs tuning
- **KNN**: Intuitive, sensitive to data scale and choice of *k*

Thankyou!