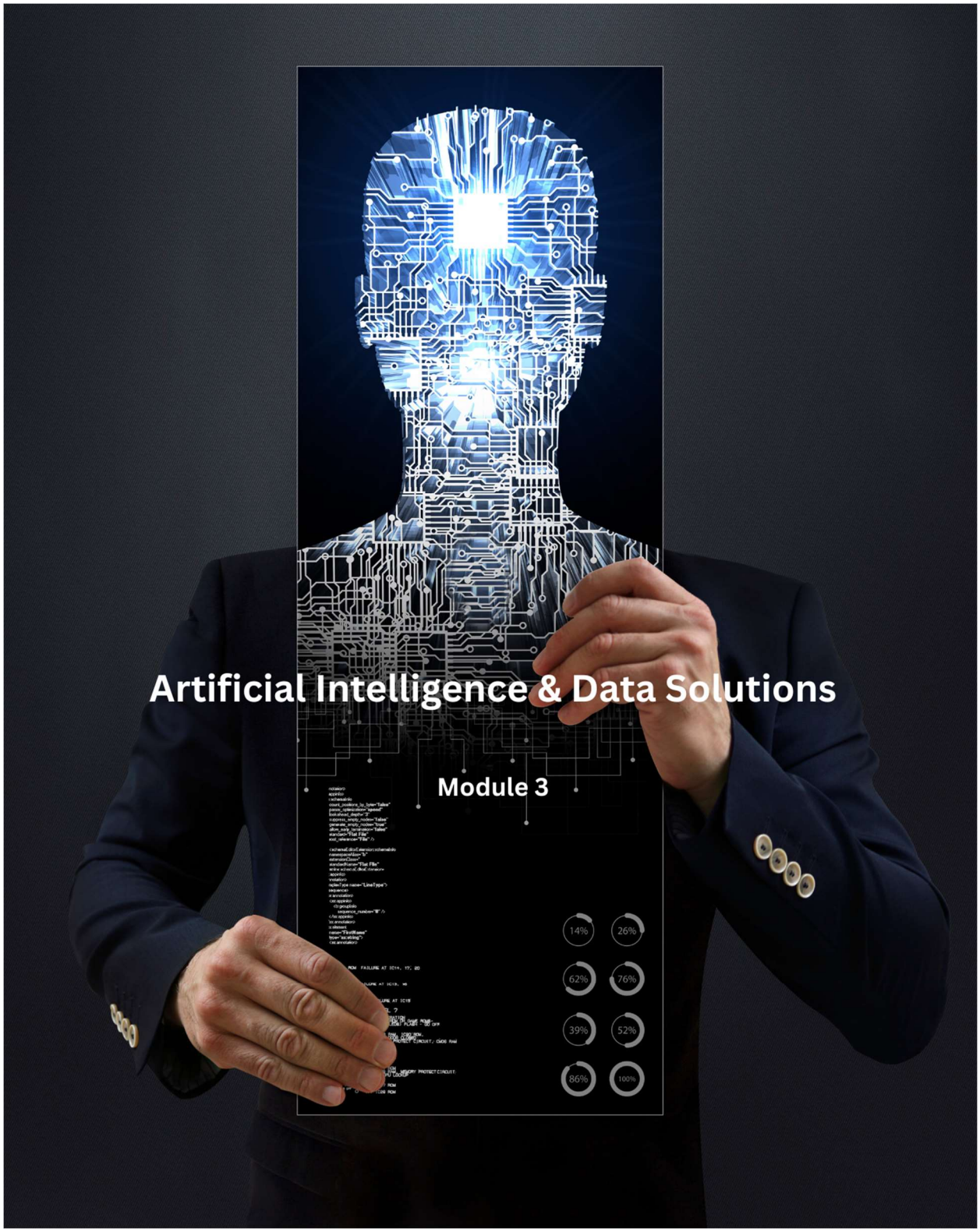

Artificial Intelligence & Data Solutions
Module 3

Module 3: Core Machine Learning and Predictive Modelling

3.1 Supervised Learning: Regression and Classification Architectures

The Supervised Learning Framework

Supervised learning forms the cornerstone of predictive modeling. In this paradigm, an algorithm learns a mapping function (f) that projects input feature vectors (X) onto target output labels (Y), based on a historical training dataset containing known pairs:

\(Y=f(X)+\epsilon \)

Where ε represents random, irreducible background noise. The nature of the target variable dictates the machine learning task:

- **Regression Tasks:** The target variable (Y) is continuous and scalar (e.g., predicting real estate values or equity prices).
- **Classification Tasks:** The target variable (Y) is discrete and categorical (e.g., classifying medical images or identifying fraudulent financial transactions).

1. Linear and Polynomial Regression Models

Linear regression assumes a linear relationship between input features and the continuous target variable. The multivariate linear model is formulated as:

\(\^{y}=\beta _{0}+\beta _{1}x_{1}+\beta _{2}x_{2}+\dots +\beta _{n}x_{n}\)

Where $\hat{y}$ is the predicted value, β_0 is the intercept term, and \(\beta _{i}\) are the model parameters (weights).

Loss Optimization: Ordinary Least Squares (OLS)

The model parameters are calculated by minimizing the **Residual Sum of Squares (RSS)**, which squares the differences between actual values (\(y_{i}\)) and predictions (\(\^{y}_{i}\)):

\(\text{RSS}=\sum _{i=1}^{m}(y_{i}-\^{y}_{i})^{2}\)

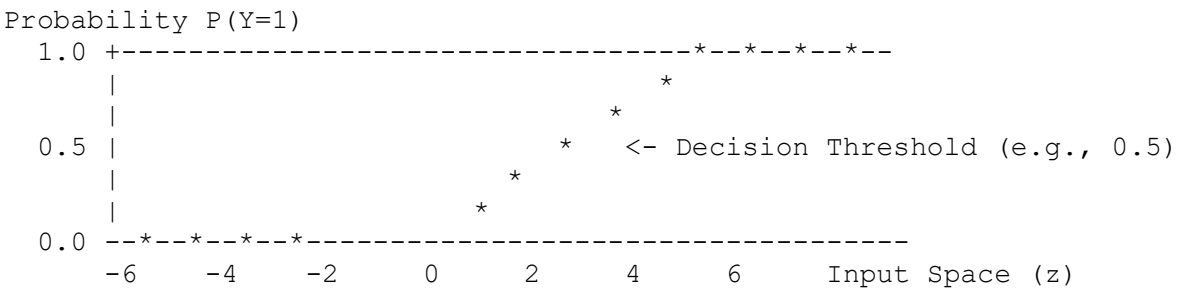
When the underlying data distribution is curved or non-linear, features can be projected into a higher-dimensional space using **Polynomial Regression** transformations, allowing linear estimators to map non-linear boundaries.

2. Logistic Regression for Binary Classification

Despite its name, Logistic Regression is a classification algorithm. It maps linear combinations of features to a probability value between 0 and 1 using the **Sigmoid (Logistic) Function**:

$$\sigma (z)=\frac{1}{1+e^{-z}}$$

Where $z = \beta^T X$. The output σ(z) represents the conditional probability $P(Y=1 \vert X)$.



Loss Optimization: Cross-Entropy Loss

Because the Sigmoid function introduces non-linearities, using OLS would result in a non-convex loss function with multiple local minima. Instead, Logistic Regression models optimize parameters using **Binary Cross-Entropy Loss (Log Loss)**:

$$\text{Loss}=-\frac{1}{m}\sum _{i=1}^{m}\left[y_{i}\log (\^{y}_{i})+(1-y_{i})\log (1-\^{y}_{i})\right]$$

3. Support Vector Machines (SVM)

Support Vector Machines seek to find an optimal **Hyperplane** in an n-dimensional space that maximizes the margin—the physical distance between the closest data points of opposing classes, known as **Support Vectors**.

The Kernel Trick

When classes are non-linearly separable in their original input dimensions, SVMs use mathematical **Kernel Functions** (e.g., Radial Basis Function [RBF] or Polynomial Kernels). These functions compute the dot product of data points in a higher-dimensional feature space without explicitly calculating the coordinates of the points in that space, bypassing heavy computational loads.

4. Decision Trees and Ensemble Architectures

Decision Trees recursively partition feature spaces based on statistical impurity metrics.

- **Information Gain:** Measures the reduction in **Entropy** (a metric of data chaos) after a split:

$$\text{Entropy}(S)=-\sum _{i=1}^{c}p_{i}\log _{2}p_{i}$$

- **Gini Impurity:** Evaluates the probability of a randomly chosen element being incorrectly labeled if it were randomly labeled according to the distribution in the subset.

Overcoming Overfitting via Ensembles

Standalone decision trees are prone to overfitting because they grow deep blocks that fit training noise too closely. Ensemble methods combine multiple weaker trees to create highly robust models:

Ensemble Typology	Operational Mechanism	Key Advantage
Random Forests (Bagging)	Trains numerous independent trees in parallel using random subsets of data (Bootstrap Aggregation) and feature spaces. Predictions are combined via majority voting or averaging.	Reduces overall model variance and prevents overfitting.
Gradient Boosting (GBM/XGBoost)	Trains trees sequentially. Each new tree is designed to fit the residual errors (loss gradients) generated by the preceding trees.	Reduces model bias, delivering high predictive accuracy on structured data.

3.2 Unsupervised Learning, Clustering, and Statistical Evaluation

The Unsupervised Learning Framework

Unsupervised learning handles datasets that lack target output labels (Y). The algorithm evaluates unlabeled feature vectors (X) to discover hidden structures, underlying distributions, and natural groupings within the data.

1. K-Means Clustering Optimization

K-Means partitions a dataset into K distinct, non-overlapping clusters. The algorithm works by minimizing the **Within-Cluster Sum of Squares (WCSS)**, also known as inertia:

$$\text{WCSS}=\sum _{k=1}^{K}\sum _{i\in C_{k}}||x_{i}-\mu _{k}||^{2}$$

Where $\mu _{k}$ represents the centroid (mean vector) of cluster C_{k}.

The Iterative Execution Loop

1. **Initialization:** Randomly assign K initial cluster centroids across the feature space.
2. **Assignment Step:** Calculate the Euclidean distance from every data point to each centroid, assigning each point to its nearest cluster.
3. **Update Step:** Compute the mean coordinates of all data points within each cluster, moving the centroids to these new mean locations.
4. **Convergence:** Repeat the assignment and update steps until centroid locations stabilize or the maximum iteration threshold is reached.

Determining Optimal Clusters: The Elbow Method

To find the ideal number of clusters (K), data scientists plot WCSS against a range of K values. As K increases, WCSS naturally drops toward zero. The optimal choice is indicated by an "Elbow" on the line plot—the point where the rate of WCSS reduction decreases significantly, showing diminishing returns for adding more clusters.

2. Hierarchical and Density-Based Clustering (DBSCAN)

- **Hierarchical Clustering:** Builds a tree of clusters (Dendrogram) using either a bottom-up approach (Agglomerative) or a top-down approach (Divisive).
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** Groups points based on the spatial density of neighboring points. It classifies points into *Core Points*, *Border Points*, and *Noise*. Unlike K-Means, DBSCAN does not require initializing K in advance and can discover clusters of arbitrary, complex geometric shapes while automatically isolating outliers.

3.3 Model Evaluation, Generalisation, and Hyperparameter Tuning

The Challenge of Generalisation

The primary objective of machine learning is to perform well on unseen, real-world data, not simply to score highly on training data. Model performance challenges are categorized using the Bias-Variance trade-off:

```
[ UNDERFITTING ]                    [ ROBUST OPTIMUM ]                    [
OVERFITTING ]
- High Bias / Low Variance          - Balanced Fit                        -
Low Bias / High Variance
- Model is too simple              - Captures true patterns              -
Captures training noise
- Fails on training data            - Generalises well to new data       -
Fails on validation data
```

- **Bias:** Errors originating from erroneous assumptions in the machine learning algorithm. High bias causes the model to underfit the data, missing key relationships between features and outputs.
- **Variance:** Errors originating from an algorithm's high sensitivity to small fluctuations in the training dataset. High variance causes the model to overfit the data, learning random background noise as if it were a valid structural pattern.

Regularization Methodologies

To curb overfitting and control variance in high-dimensional datasets, regularization penalties are added directly to the model's loss function:

- **L1 Regularization (Lasso Regression):** Adds an absolute value penalty of the weights ($\lambda \sum \vert \beta_i \vert$) to the loss function. This drives less important feature weights to zero, performing automatic feature selection.
- **L2 Regularization (Ridge Regression):** Adds a squared value penalty of the weights ($\lambda \sum \beta_i^2$) to the loss function. This shrinks feature weights close to zero, minimizing the impact of highly correlated features without removing them.

Cross-Validation Workflows

To evaluate how well a model generalizes before deployment, data platforms use **K-Fold Cross-Validation**:

```
Raw Data Vector ---> [ 20% Holdout Test Dataset ]
               ---> [ 80% Training/Validation Dataset Split into K Folds ]

Iteration 1: [ Fold 1: VAL  ] [ Fold 2: TRAIN ] [ Fold 3: TRAIN ] [ Fold
4: TRAIN ]
Iteration 2: [ Fold 1: TRAIN ] [ Fold 2: VAL  ] [ Fold 3: TRAIN ] [ Fold
4: TRAIN ]
Iteration 3: [ Fold 1: TRAIN ] [ Fold 2: TRAIN ] [ Fold 3: VAL  ] [ Fold
4: TRAIN ]
```

The training dataset is split into K equal-sized subsets (folds). The model is trained K times, each time using a different fold as the validation set while combining the remaining K-1 folds for training. The final evaluation score is calculated by averaging the performance across all K iterations.

Comprehensive Classification Metrics

Evaluating classification performance using simple accuracy can be misleading when working with imbalanced datasets (e.g., fraud datasets where only 0.1% of transactions are fraudulent). Instead, systems are audited using a **Confusion Matrix**, which tracks four distinct prediction outcomes:

$$\text{Confusion Matrix}=\left[\begin{matrix}\text{True Positives (TP)}&\text{False Positives (FP)}\\ \text{False Negatives (FN)}&\text{True Negatives (TN)}\end{matrix}\right]$$

- **Precision:** Measures the accuracy of positive predictions. It answers: *Of all items predicted as positive, how many were actually positive?*
  $$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}}$$

- **Recall (Sensitivity):** Measures the model's ability to find all positive instances. It answers: *Of all actual positive items, how many did the model correctly identify?*
  $$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}}$$

- **F1-Score:** The harmonic mean of Precision and Recall, providing a balanced metric for evaluating uneven data distributions:
  $$\text{F1-Score}=2\times \frac{\text{Precision}\times \text{Recall}}{\text{Precision}+\text{Recall}}$$

Legal Notice & Terms of Use

1. Copyright and Intellectual Property Notice

© 2026 Avenbury College. All Rights Reserved.

All course materials, including but not limited to text, structures, curriculum design, digital assets, diagrams, and methodologies contained within this Free Course Initiative, are the exclusive intellectual property of **Avenbury College** and are protected under the United Kingdom Copyright, Designs and Patents Act 1988 and international intellectual property treaties.

2. Free Education Initiative & Access Terms

This course is published strictly as a Free-to-Access Educational Resource for the public. It is designed to support aspiring students, professionals, and functional leaders worldwide by removing financial barriers to high-quality education. There are absolutely no fees, subscriptions, or hidden charges required to access, read, or study this material directly on our official platform. Please note that while access is free, the content remains the proprietary property of the College and is not released under an open-source or public-domain license.

3. Permitted and Restricted Usage (Terms of Distribution)

While we encourage the widespread use of this material for personal development and academic growth, strict conditions apply to protect institutional integrity:

- **Personal and Educational Use:** You are permitted to read, study, and reference this text solely for your personal, non-commercial education. Downloading or printing is limited to individual, private study use only.
- **Mandatory Attribution:** If you quote, reference, or summarize parts of this course on external websites, blogs, academic papers, or social platforms, you must provide explicit and visible attribution to **Avenbury College**, alongside a direct hyper-link back to our official course page.
- **Strict Prohibition on Plagiarism and Re-publishing:** You are strictly prohibited from copying, replicating, scraping, or re-publishing this content in its entirety or in substantial parts on any other website, learning management system (LMS), or digital platform.
- **AI Scraping Restriction:** Data mining, robots, or similar data gathering and extraction tools are strictly prohibited from scraping this content for the purpose of training artificial intelligence (AI) models without express written consent.
- **Commercial Restriction:** Selling, white-labeling, or using this text to generate commercial revenue under another brand name is strictly illegal and will result in immediate legal action under UK copyright enforcement acts.
- **Permissions Contact:** For any requests regarding licensing, external reuse, or distribution permissions, please contact our administrative team at: **info@avnc.co.uk**.

4. Educational Disclaimer

The information contained in this foundational course is for general educational and informational purposes only. While we strive for academic excellence, **Avenbury College** makes no representations or warranties of any kind regarding specific career outcomes, financial success, or professional performance resulting from the study of these free materials.