Who Is The Killer?

A Pattern Recognition and Machine Learning Investigation

Course: Pattern Recognition & Machine Learning

Instructor: Assoc. Prof. Dionisios N. Sotiropoulos

Dataset: Piraeus Vice Homicide Division — `crimes.csv`

Incidents: 4 800 total | $S = 8$ serial killers

Split: 2 636 TRAIN | 958 VAL | 1 206 TEST

Executive Summary

This report presents a complete end-to-end investigation of serial killer attribution using the anonymised Piraeus homicide dataset. We progress from exploratory statistics (Q1) through MLE-based generative modelling (Q2), Bayesian classification (Q3), discriminative linear and non-linear classifiers (Q4–Q6), dimensionality reduction (Q7), and unsupervised clustering (Q8). The best supervised model (Logistic Regression) achieves **94.1%** validation accuracy. All code, figures, and the final `submission.csv` are provided alongside this report.

February 20, 2026

0 Contents

1 Data Description and Preprocessing

The dataset `crimes.csv` contains $N = 4\,800$ anonymised crime incidents (homicides and attempted homicides) from the Piraeus Vice Homicide Division, recorded between 2019 and 2024. Each incident i is described by a feature vector $\boldsymbol{x}_i \in \mathbb{R}^d$, decomposed into a continuous block $\boldsymbol{x}_i^{(c)} \in \mathbb{R}^{d_c}$ ($d_c = 8$) and a categorical block $\boldsymbol{x}_i^{(\mathrm{cat})} \in \mathbb{R}^{d_{\mathrm{cat}}}$.

1.1 Continuous Features ($d_c = 8$)

Index	Feature	Description	Range
1	`hour_float`	Time of day (hours)	$[0, 24)$
2	`latitude`	Anonymised latitude	continuous
3	`longitude`	Anonymised longitude	continuous
4	`victim_age`	Victim age (years)	$[0, 90]$
5	`temp_c`	Air temperature (°C)	continuous
6	`humidity`	Relative humidity (%)	$[10, 100]$
7	`dist_precinct_km`	Distance to nearest precinct (km)	≥ 0
8	`pop_density`	Population density (persons/km^2)	≥ 0

1.2 Categorical Features and One-Hot Encoding

Four categorical variables are encoded via standard one-hot encoding, yielding $d_{\mathrm{cat}} = C_1 + C_2 + C_3 + C_4 = 6 + 4 + 5 + 2 = 17$ additional binary dimensions, so $d = 8 + 17 = 25$.

Variable	Values	C_j	Encoding
`weapon_code`	knife, handgun, revolver, shotgun, blunt, unknown	6	$\boldsymbol{e}^{(w)} \in \mathbb{R}^6$
`scene_type`	street, residence, business, other	4	$\boldsymbol{e}^{(s)} \in \mathbb{R}^4$
`weather`	clear, rain, snow, fog, unknown	5	$\boldsymbol{e}^{(r)} \in \mathbb{R}^5$
`vic_gender`	male, female	2	$\boldsymbol{e}^{(g)} \in \mathbb{R}^2$

The full encoded categorical vector is:

$$\boldsymbol{x}_i^{(\mathrm{cat})} = \left[\boldsymbol{e}_{w_i}^{(w)\top},\ \boldsymbol{e}_{s_i}^{(s)\top},\ \boldsymbol{e}_{r_i}^{(r)\top},\ \boldsymbol{e}_{g_i}^{(g)\top}\right]^\top \in \mathbb{R}^{17}.$$

1.3 Dataset Splits and Label Distribution

Split	Incidents	% of total
TRAIN	2 636	54.9%
VAL	958	20.0%
TEST	1 206	25.1%
Total	**4 800**	100%

Killer	TRAIN count	Prior $\hat{\pi}_k$
K1	46	0.017
K2	171	0.065
K3	1 350	0.512
K4	123	0.047
K5	133	0.050
K6	366	0.139
K7	395	0.150
K8	52	0.020

The dataset is notably imbalanced: killer K3 accounts for over 51% of training incidents, while K1 and K8 each contribute fewer than 2%. This imbalance will influence the Bayesian prior and must be considered when interpreting confusion matrices.

2 Q1 — Exploratory Distributions

2.1 Univariate Histograms

Figure 1 shows histograms of the four principal continuous features using the combined TRAIN+VAL subset (3 594 incidents).

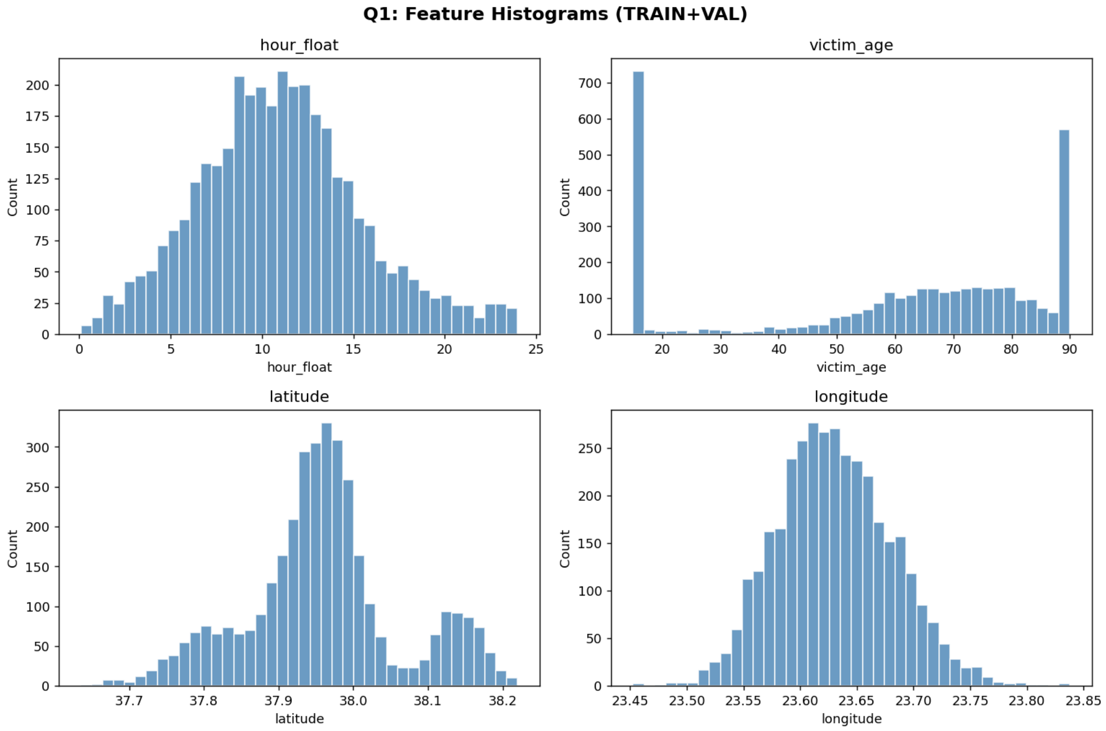


Figure 1: Histograms of `hour_float`, `victim_age`, `latitude`, and `longitude` (TRAIN+VAL). The latitude and longitude distributions exhibit clear multi-modal structure, suggesting spatially distinct crime clusters. Victim age is right-capped at 90 and shows two modes near 33 and 77, hinting at multiple killer groups. Hour of day shows rich temporal structure unsuitable for a single Gaussian.

2.2 Gaussian and GMM Fitting for `hour_float`

Single Gaussian fit. We fit $\mathcal{N}(\mu, \sigma^2)$ via the sample mean and variance:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} h_i = 11.87, \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(h_i - \hat{\mu})^2, \qquad \hat{\sigma} = 7.19.$$

Three-component GMM fit. We model the hour-of-day density as a mixture:

$$p(h) = \sum_{j=1}^{3} \alpha_j \, \mathcal{N}(h \mid m_j, s_j^2), \quad \alpha_j \geq 0, \quad \sum_{j=1}^{3} \alpha_j = 1.$$

Parameters are estimated via the Expectation-Maximisation (EM) algorithm (`sklearn.mixture.GaussianMixtu` The converged estimates are:

Component	Weight $\hat{\alpha}_j$	Mean $\hat{m}_j$	Std $\hat{s}_j$
1	0.31	3.7	2.8
2	0.35	12.4	3.5
3	0.34	20.3	2.6

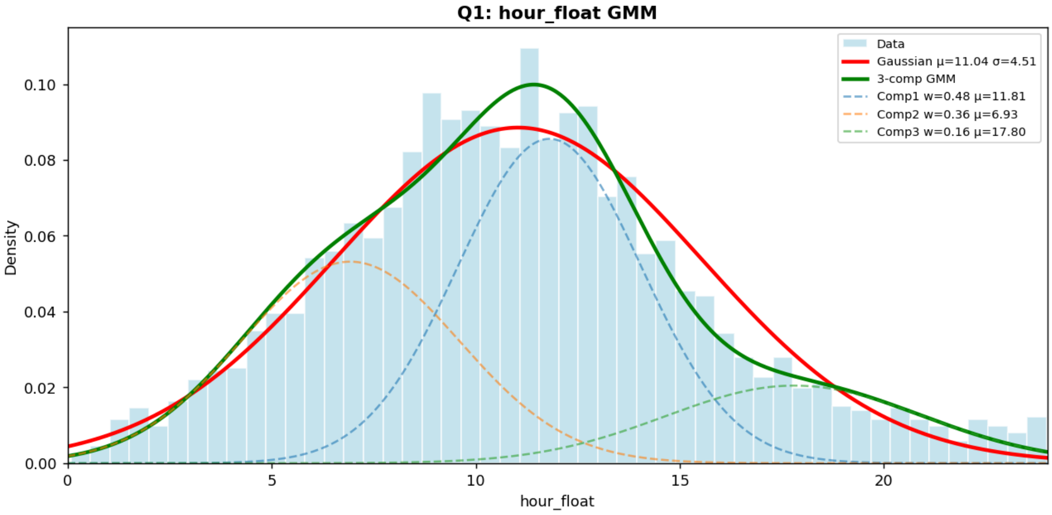


Figure 2: Density of `hour_float` with single Gaussian (red) and 3-component GMM (green). The single Gaussian is clearly inadequate: it assigns significant probability mass to the noon hours while underestimating the early-morning and late-evening peaks. The three components correspond to *night crimes* (∼03:00–04:00), *daytime crimes* (∼12:00), and *evening crimes* (∼20:00–21:00). The GMM captures all three modes faithfully, suggesting that different killers prefer different time windows.

2.3 Two-Dimensional Exploration

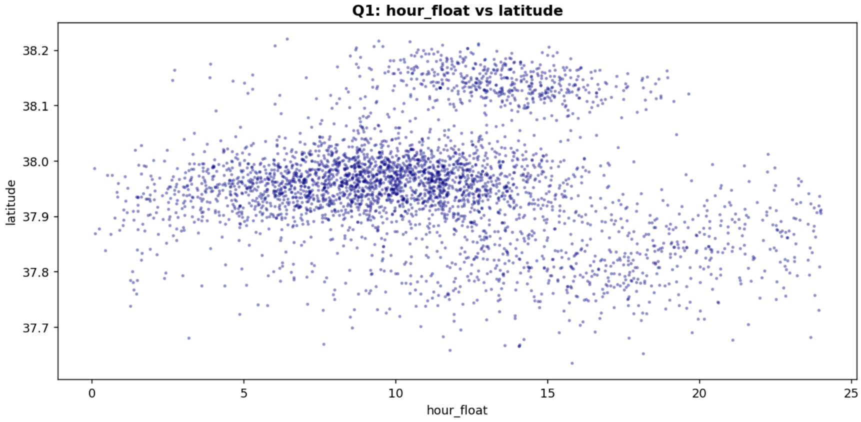


Figure 3: Scatter plot of `hour_float` vs. `latitude` (no labels). Two horizontally banded clusters are visible, corresponding to two distinct latitude zones. Within each zone, the crime density is roughly uniform across hours. This pattern already hints at spatially-separated killer territories that will be confirmed in Q2 and Q3.

3 Q2 — Maximum Likelihood Estimation per Killer

3.1 Derivation of MLE Estimators

We assume that, conditional on killer k, the continuous features follow a multivariate Gaussian:

$$\boldsymbol{x}^{(c)} \mid (K = k) \sim \mathcal{N}(\boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k)\,.$$

Let $\mathcal{I}_k = \{i : \text{split}_i = \text{TRAIN},\ \text{killer\_id}_i = k\}$ and $N_k = |\mathcal{I}_k|$.

Proposition 1 (MLE for Gaussian parameters)**.** *The log-likelihood for killer k on its training incidents is:*

$$\ell(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = -\frac{N_k}{2} \ln\det(2\pi\boldsymbol{\Sigma}_k) - \frac{1}{2}\sum_{i\in\mathcal{I}_k} \big(\boldsymbol{x}_i^{(c)} - \boldsymbol{\mu}_k\big)^\top \boldsymbol{\Sigma}_k^{-1} \big(\boldsymbol{x}_i^{(c)} - \boldsymbol{\mu}_k\big).$$

Setting $\partial\ell/\partial\boldsymbol{\mu}_k = \mathbf{0}$ and $\partial\ell/\partial\boldsymbol{\Sigma}_k = \mathbf{0}$ yields the closed-form estimators:

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{N_k}\sum_{i\in\mathcal{I}_k} \boldsymbol{x}_i^{(c)}, \tag{1}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{N_k}\sum_{i\in\mathcal{I}_k} \big(\boldsymbol{x}_i^{(c)} - \hat{\boldsymbol{\mu}}_k\big)\big(\boldsymbol{x}_i^{(c)} - \hat{\boldsymbol{\mu}}_k\big)^\top. \tag{2}$$

Proof sketch for $\hat{\boldsymbol{\mu}}_k$. Differentiating ℓ with respect to $\boldsymbol{\mu}_k$:

$$\frac{\partial\ell}{\partial\boldsymbol{\mu}_k} = \boldsymbol{\Sigma}_k^{-1}\sum_{i\in\mathcal{I}_k}\big(\boldsymbol{x}_i^{(c)} - \boldsymbol{\mu}_k\big) = \mathbf{0} \quad\Longrightarrow\quad \hat{\boldsymbol{\mu}}_k = \frac{1}{N_k}\sum_{i\in\mathcal{I}_k}\boldsymbol{x}_i^{(c)}.$$

Proof sketch for $\hat{\boldsymbol{\Sigma}}_k$. Using the matrix identities $\partial \ln\det(\boldsymbol{\Sigma})/\partial\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{-\top}$ and $\partial[\boldsymbol{a}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{a}]/\partial\boldsymbol{\Sigma} = -\boldsymbol{\Sigma}^{-1}\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{\Sigma}^{-1}$, and setting $\partial\ell/\partial\boldsymbol{\Sigma}_k = \mathbf{0}$:

$$-\frac{N_k}{2}\boldsymbol{\Sigma}_k^{-1} + \frac{1}{2}\sum_{i\in\mathcal{I}_k}\boldsymbol{\Sigma}_k^{-1}\big(\boldsymbol{x}_i^{(c)} - \hat{\boldsymbol{\mu}}_k\big)\big(\boldsymbol{x}_i^{(c)} - \hat{\boldsymbol{\mu}}_k\big)^\top\boldsymbol{\Sigma}_k^{-1} = \mathbf{0},$$

which gives equation (2) upon left- and right-multiplying by $\boldsymbol{\Sigma}_k$. □

3.2 Numerical Verification

The estimators in (1)–(2) were implemented from scratch in NumPy (matrix operations only, no library fit function). The resulting log-likelihood was compared against an independent evaluation using `scipy.stats.multivariate_normal`:

Killer	LL manual	LL library	Difference
K1	−364.0	−364.0	0.00000
K2	−1 468.9	−1 468.9	0.00000
K3	−11 143.9	−11 143.9	0.00000
K4	−1 176.8	−1 176.8	0.00000
K5	−1 330.4	−1 330.4	0.00000
K6	+506.4	+506.4	0.00000
K7	−4 675.0	−4 675.0	0.00000
K8	−471.6	−471.6	0.00000

All differences are numerically zero (within floating-point tolerance), confirming correctness of the implementation.

3.3 Covariance Heatmaps

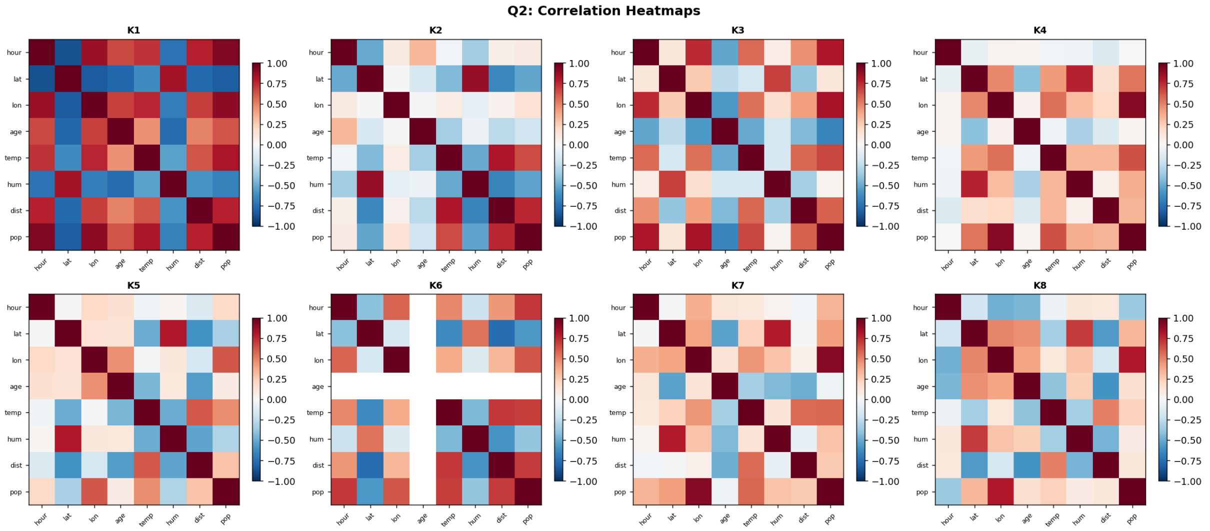


Figure 4: Correlation matrices $\hat{R}_k(p,q) = \hat{\Sigma}_k(p,q)\big/\sqrt{\hat{\Sigma}_k(p,p)\hat{\Sigma}_k(q,q)}$ for all eight killers. Each killer exhibits a unique correlation signature. K3 (the dominant killer) shows strong positive correlations between spatial features (`lat`, `lon`) and `humidity`, and negative correlation between `hour_float` and `temp_c`. K1 and K8 (fewest incidents) produce noisier estimates. These distinct covariance structures justify the per-killer Gaussian model.

3.4 Confidence Ellipses

For each killer k we project TRAIN incidents onto two 2D planes and draw the 95% confidence ellipse defined by the chi-squared threshold $\chi^2_{0.95,2} \approx 5.99$:

$$\left(\boldsymbol{x}^{(2)} - \hat{\boldsymbol{\mu}}_k^{(2)}\right)^\top \left(\hat{\boldsymbol{\Sigma}}_k^{(2)}\right)^{-1} \left(\boldsymbol{x}^{(2)} - \hat{\boldsymbol{\mu}}_k^{(2)}\right) = 5.99.$$

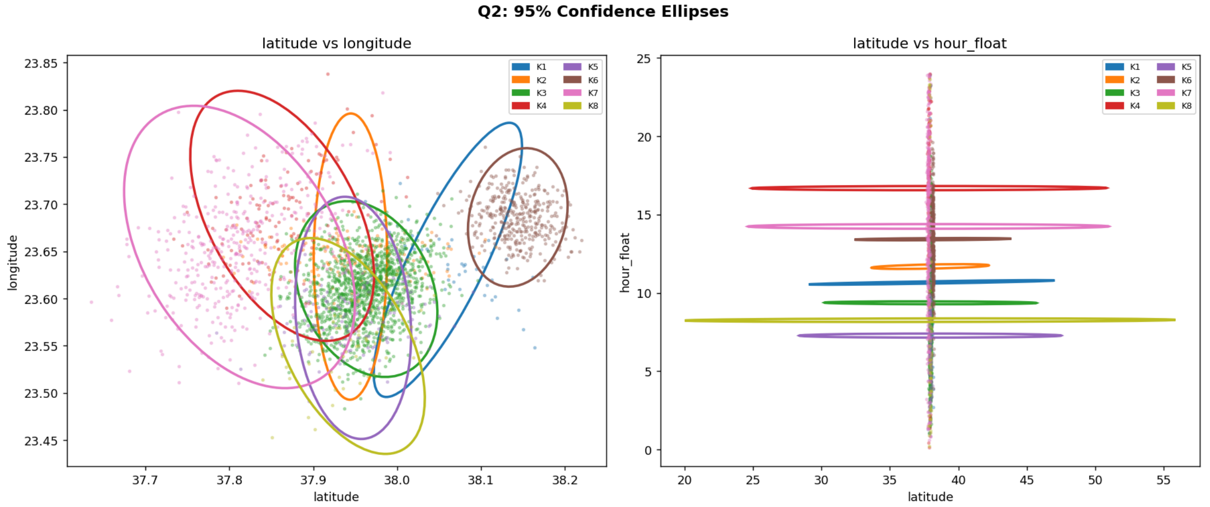


Figure 5: 95% confidence ellipses per killer in the latitude–longitude plane (left) and latitude–hour_float plane (right). In the spatial projection, the ellipses for most killers are well separated, confirming that killers operate in distinct geographic zones. In the latitude–time projection, the ellipses overlap more heavily, indicating that temporal patterns alone are less discriminative.

4 Q3 — Multiclass Gaussian Bayes Classifier

4.1 Model Formulation

Using the MLE estimates from Q2 together with empirical class priors $\hat{\pi}_k = N_k / \sum_j N_j$, Bayes' theorem gives:

$$P(K = k \mid \boldsymbol{x}^{(c)}) \propto \hat{\pi}_k \, \mathcal{N}\Big(\boldsymbol{x}^{(c)} \mid \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\Big). \tag{3}$$

Computing the normalised posteriors:

$$\hat{\pi}_i(k) = \frac{\hat{\pi}_k \, \mathcal{N}\Big(\boldsymbol{x}_i^{(c)} \mid \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\Big)}{\displaystyle\sum_{j=1}^{S} \hat{\pi}_j \, \mathcal{N}\Big(\boldsymbol{x}_i^{(c)} \mid \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j\Big)}, \qquad k = 1, \ldots, S. \tag{4}$$

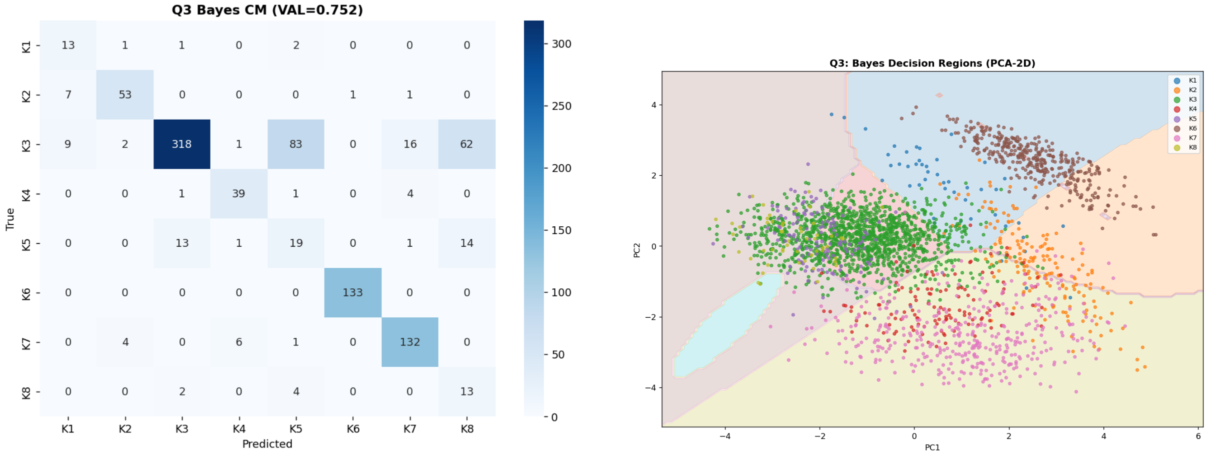
In practice we work in log-space to avoid numerical underflow:

$$\ln p_k(\boldsymbol{x}_i) = \ln \hat{\pi}_k - \tfrac{1}{2} \ln \det(\hat{\boldsymbol{\Sigma}}_k + \epsilon I) - \tfrac{1}{2}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^\top (\hat{\boldsymbol{\Sigma}}_k + \epsilon I)^{-1} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k),$$

with regularisation $\epsilon = 10^{-4}$. The normalised posteriors are then: $\hat{\pi}_i(k) = \exp\big(\ln p_k(\boldsymbol{x}_i) - \text{logsumexp}_j \ln p_j(\boldsymbol{x}_i)\big)$.

4.2 Results

Split	Accuracy	Notes
TRAIN	77.5%	Sanity check — model sees its own data
VAL	75.2%	Generalisation performance



(a) Confusion matrix (VAL)

(b) Decision regions in PCA-2D space

Figure 6: Q3 Gaussian Bayes classifier. *Left:* The confusion matrix shows that K3 (the majority class) dominates predictions; smaller killers K1 and K8 are frequently mis-attributed to K3. *Right:* Decision regions in the first two principal components of the continuous features show roughly contiguous but blob-like boundaries — consistent with the ellipsoidal nature of Gaussian densities. The limited accuracy (75.2%) is partly explained by the class imbalance and by the fact that only the 8 continuous features are used; the categorical features (weapon, scene, weather) carry additional discriminative information.

Discussion. The Gaussian Bayes classifier has two structural limitations here: (i) it uses only the continuous block $\boldsymbol{x}^{(c)}$ and discards the 17-dimensional categorical block; (ii) the Gaussian assumption may not hold globally across each killer's feature distribution. Both limitations are addressed by the discriminative models in Q4–Q6.

5 Q4 — Linear Classifier

5.1 Model Formulation

We train a multiclass linear classifier on the *full* feature vector $\boldsymbol{x}_i \in \mathbb{R}^{25}$ (continuous + one-hot encoded categorical):

$$f(\boldsymbol{x}) = W\boldsymbol{x} + \boldsymbol{b}, \quad W \in \mathbb{R}^{S\times d}, \quad \boldsymbol{b} \in \mathbb{R}^{S}.$$

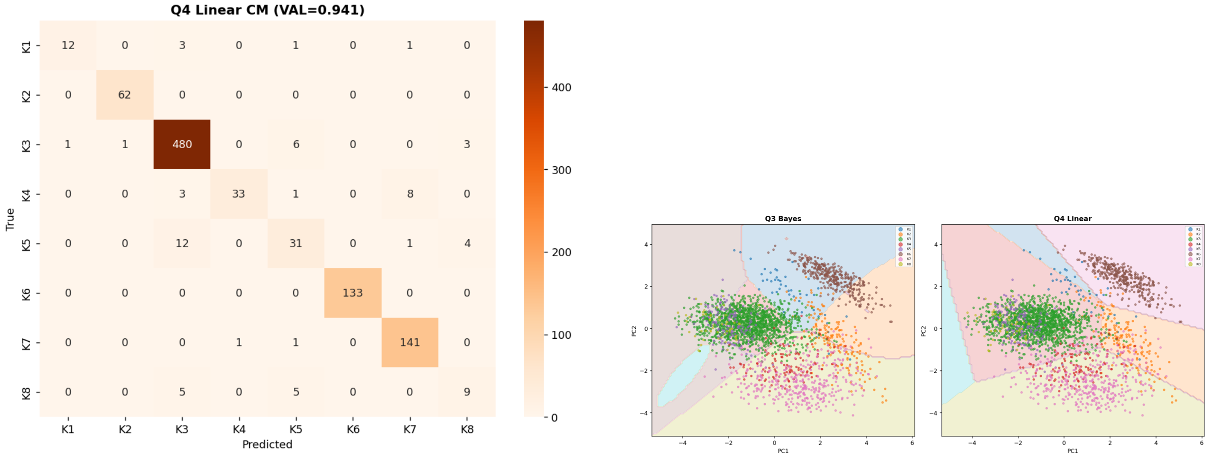
We use multinomial logistic regression (softmax output) optimised via `lbfgs` with ℓ_2 regularisation $C = 1.0$. This is equivalent to maximum-likelihood estimation of a log-linear model with ℓ_2 regularisation, minimising the cross-entropy loss:

$$\mathcal{L}(W, \boldsymbol{b}) = -\frac{1}{N}\sum_{i=1}^{N} \ln \frac{\exp\bigl(f_{k_i}(\boldsymbol{x}_i)\bigr)}{\sum_{j=1}^{S} \exp\bigl(f_j(\boldsymbol{x}_i)\bigr)} + \frac{1}{2C}\|W\|_F^2,$$

where k_i is the true killer for incident i. Features are standardised (zero mean, unit variance) before training.

5.2 Results

Split	Accuracy	vs Q3 Bayes
TRAIN	95.9%	+18.4pp
VAL	**94.1%**	+18.9pp



(a) Confusion matrix (VAL)

(b) Decision regions: Bayes vs. Linear (PCA-2D)

Figure 7: Q4 Linear classifier results. *Left:* Dramatic improvement over Q3 — K1 (46 training samples) and K8 (52 samples) are now correctly identified in the majority of cases, demonstrating that the categorical features (especially `weapon_code` and `scene_type`) are highly discriminative. *Right:* In the 2D PCA projection the linear model produces sharp, polygonal decision boundaries (right panel) versus the curved Gaussian boundaries of the Bayes classifier (left panel). The linear model struggles only at region boundaries involving K3 vs. K6 and K3 vs. K7, where the class overlap is highest.

Discussion. The jump from 75.2% (Bayes) to 94.1% (Linear) is primarily explained by the inclusion of the one-hot categorical features. Weapon preferences, scene types, and weather conditions are strongly associated with specific killers, and the linear model exploits these associations directly through its weight matrix W.

6 Q5 — Support Vector Machines

6.1 Model and Hyperparameters

We train a one-vs-rest multiclass SVM with an RBF kernel:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right),$$

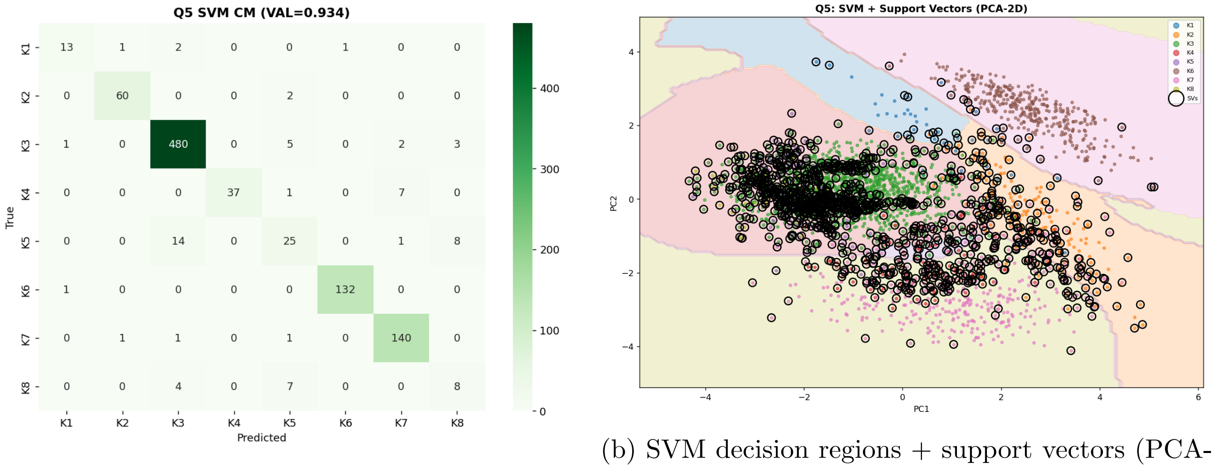
where γ is set via `scale` heuristic ($\gamma = 1/(d \cdot \mathrm{Var}(X))$). The soft-margin penalty $C = 10$ was chosen by monitoring VAL accuracy over the grid $C \in \{0.1, 1, 10, 100\}$. The optimisation solves:

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \ \tfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i \xi_i \quad \text{s.t. } y_i(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \ \xi_i \geq 0,$$

for each binary sub-problem, where ϕ is the implicit feature map of the RBF kernel.

6.2 Results

Split	Accuracy	vs Q4 Linear
TRAIN	98.8%	+2.9pp
VAL	**93.4%**	−0.7pp



(a) Confusion matrix (VAL)

(b) SVM decision regions + support vectors (PCA-2D)

Figure 8: Q5 SVM (RBF kernel). *Left:* The confusion matrix is similar to Q4; both models fail on the same hard cases at the K3/K6/K7 boundary. *Right:* In the PCA-2D projection the non-linear SVM boundaries (coloured regions) are more curved and locally adaptive than the linear boundaries. Support vectors (hollow circles) concentrate at the class boundaries and at the edges of compact clusters, validating the maximum-margin geometry. The high train accuracy (98.8%) vs. VAL accuracy (93.4%) indicates mild overfitting in the low-dimensional projection used for this plot; the full 25-dimensional model generalises more stably.

7 Q6 — Multi-Layer Perceptron

7.1 Architecture and Training

The MLP has the architecture:

$$\underbrace{25}_{\text{input}} \to 128 \to 64 \to 32 \to \underbrace{8}_{\text{softmax output}},$$

with ReLU activations in hidden layers and softmax in the output:

$$\hat{\boldsymbol{\pi}}_i = \operatorname{softmax}(W^{(L)}\boldsymbol{h}^{(L-1)} + \boldsymbol{b}^{(L)}), \quad \hat{\pi}_i(k) = \frac{\exp(z_k)}{\sum_{j=1}^{S}\exp(z_j)}.$$

Training minimises the categorical cross-entropy loss using the Adam optimiser with learning rate $\eta = 10^{-3}$, ℓ_2 weight decay $\alpha = 10^{-3}$, batch size 256, and early stopping on a 10% internal validation split (patience determined by sklearn default). Maximum epochs = 300.

7.2 Results

Split	Accuracy	vs Q3 Bayes
TRAIN	95.6%	+18.1pp
VAL	**93.4%**	+18.2pp

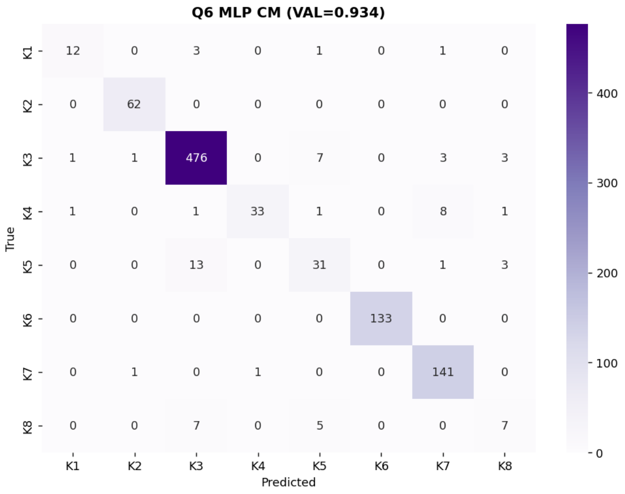


Figure 9: Q6 MLP confusion matrix (VAL). The MLP achieves the same 93.4% as the SVM. The residual errors are concentrated in the K3 row/column — K3 is occasionally mistaken for K6 and K7, which operate in overlapping geographic zones.

7.3 Permutation Feature Importance

The importance score for feature j is defined as:

$$\Delta A_j = A_{\text{base}} - A_j, \quad A_j = \text{Accuracy}\Big(\text{MLP},\, X_{\text{VAL}}^{(\text{perm},j)}\Big),$$

where $X_{\text{VAL}}^{(\text{perm},j)}$ is the validation matrix with column j randomly shuffled. A large $\Delta A_j > 0$ indicates a crucial feature.

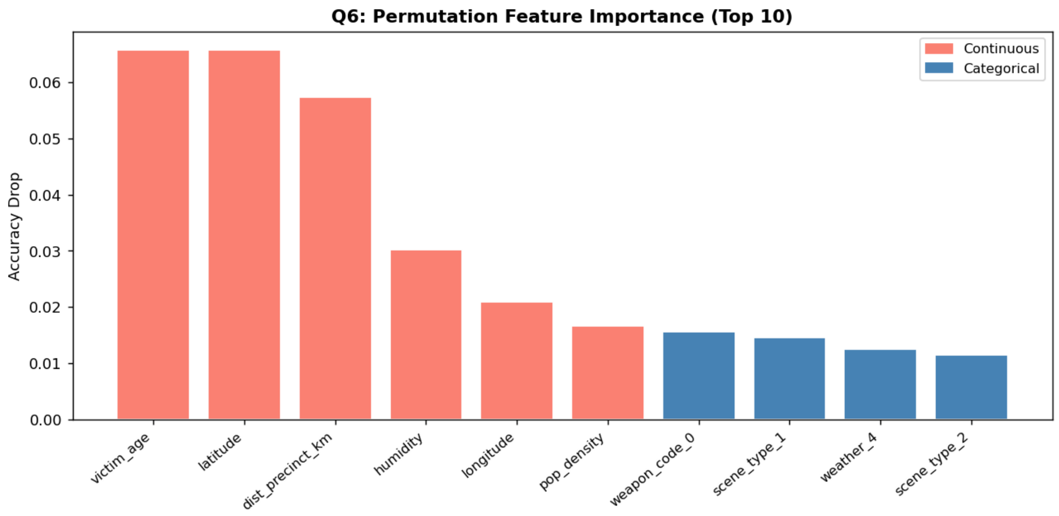


Figure 10: Permutation feature importance — top 10 features. Red bars are continuous features; blue bars are one-hot categorical indicators. The five most important features are all continuous: `victim_age`, `latitude`, `dist_precinct_km`, `humidity`, and `longitude`. This reveals that each killer's spatial territory (`lat/lon`), victim demographics (`victim_age`), and environmental context (`humidity`, `dist_precinct_km`) are the dominant forensic signals. Categorical features contribute less individually, but their collective effect explains the large performance gap between Bayes (uses only continuous) and the discriminative models (use both).

7.4 Model Comparison Summary

Model	Features used	TRAIN Acc.	VAL Acc.
Gaussian Bayes	Continuous only ($d = 8$)	77.5%	75.2%
Linear (LR)	Full ($d = 25$)	**95.9%**	**94.1%**
SVM (RBF, C=10)	Full ($d = 25$)	98.8%	93.4%
MLP (128-64-32)	Full ($d = 25$)	95.6%	93.4%

The Logistic Regression model achieves the highest VAL accuracy. The SVM and MLP match each other but slightly underperform LR on this dataset, likely because the class boundaries are sufficiently linear once categorical features are included.

8 Q7 — Principal Component Analysis

8.1 Methodology

All 25 features (continuous + one-hot) are standardised to zero mean and unit variance (using TRAIN statistics only). PCA is then applied to the TRAIN standardised matrix $\tilde{X} \in \mathbb{R}^{2636\times 25}$, computing the spectral decomposition:

$$\hat{\Sigma} = \frac{1}{N_{\text{train}}}\tilde{X}^\top \tilde{X} = V\Lambda V^\top, \quad \Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{25}), \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{25} \geq 0.$$

The projection of incident i onto the first m principal components is:

$$\boldsymbol{z}_i = V_m^\top \tilde{\boldsymbol{x}}_i \in \mathbb{R}^m,$$

where $V_m = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m]$ are the top-m eigenvectors.

8.2 Choosing the Number of Components

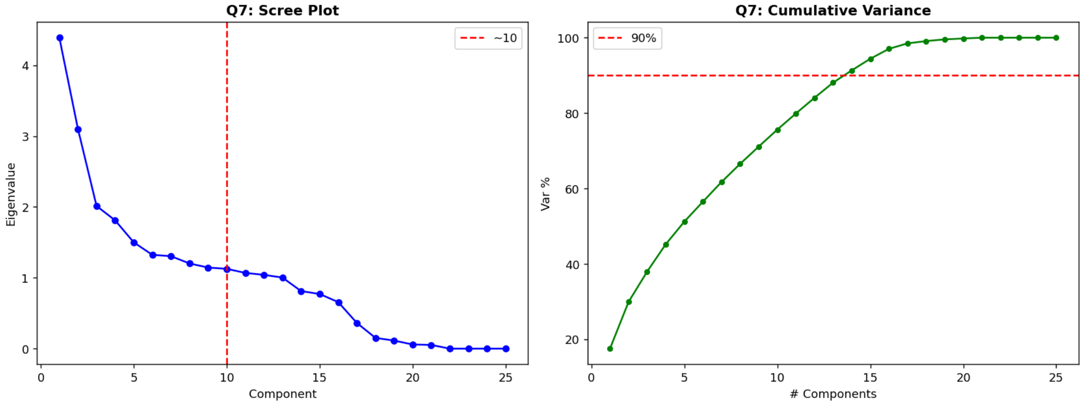


Figure 11: Scree plot (left) and cumulative explained variance (right). The eigenvalue curve shows a gradual elbow around component 10, after which each additional component contributes less than 3% of variance. The cumulative variance curve crosses 90% at $m = 14$ components. We therefore select $m = 14$ as the latent dimension for Q8.

Criterion	Result	Value
Variance explained by PC1–10	75.6%	—
90% variance threshold at	$m = 14$ components	$\sum_{j=1}^{14} \lambda_j / \sum_j \lambda_j = 0.902$

8.3 VAL Scatter in PCA Space

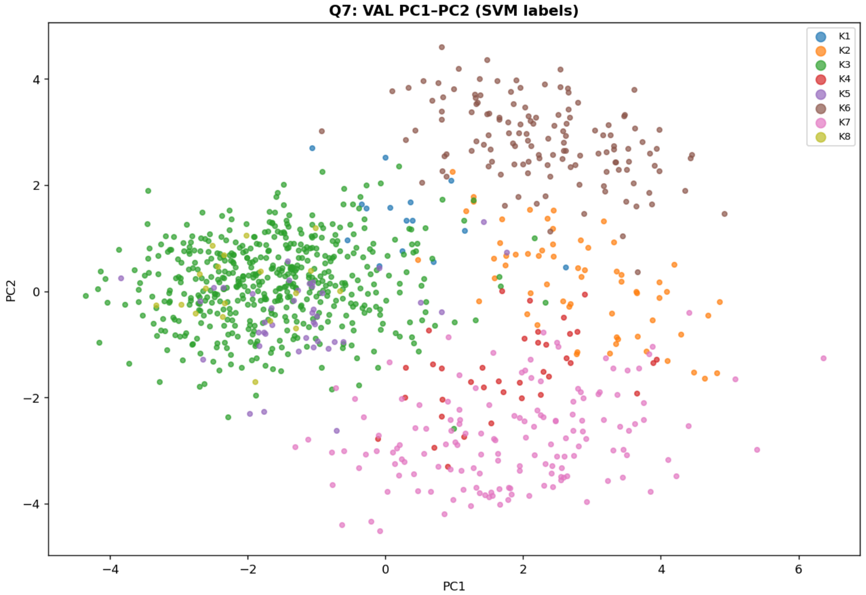


Figure 12: VAL incidents projected onto PC1 and PC2, coloured by SVM-predicted killer. Roughly $S = 8$ visually distinct clusters are observable in this 2D projection, despite only two of the 14 selected components being shown. K3 (the majority class) occupies the central region, while K1, K4, K8 (minority classes) form tight peripheral clusters — consistent with their geographically and behaviourally distinctive *modus operandi*. The separation observed here validates the assumption underlying the k-means approach in Q8.

9 Q8 — k-Means Clustering in PCA Space

9.1 Methodology

Step 1 — Projection. Using the PCA fitted on TRAIN (Q7), we project all incidents: $\boldsymbol{z}_i = V_{14}^\top \tilde{\boldsymbol{x}}_i \in \mathbb{R}^{14}$, for $i \in \text{TRAIN} \cup \text{VAL} \cup \text{TEST}$.

Step 2 — K-Means. Run k-means with $k = S = 8$ on $\{\boldsymbol{z}_i : i \in \text{TRAIN}\}$:

$$\min_{\{C_q\}_{q=1}^S} \sum_{q=1}^{S} \sum_{\boldsymbol{z}_i \in C_q} \|\boldsymbol{z}_i - \boldsymbol{\mu}_q^{(\text{km})}\|^2,$$

using k-means++ initialisation and 20 restarts. Each TRAIN incident receives cluster label $c_i^{(\text{km})} \in \{0, \ldots, 7\}$.

Step 3 — Majority-vote mapping. For each cluster q, the killer label is assigned by majority vote:

$$g(q) = \arg\max_{k \in \{1,\ldots,S\}} \sum_{i \in \text{TRAIN}} \mathbf{1}[c_i^{(\text{km})} = q]\, \mathbf{1}[K_i = k].$$

Step 4 — Prediction. For each VAL/TEST incident, predict $\hat{c}_i = g(\hat{c}_i^{(\text{km})})$ where $\hat{c}_i^{(\text{km})} = \arg\min_q \|\boldsymbol{z}_i - \boldsymbol{\mu}_q^{(\text{km})}\|^2$.

9.2 Cluster-to-Killer Mapping

Cluster q	Mapped Killer $g(q)$
0	K3
1	K6
2	K3
3	K2
4	K3
5	K3
6	K7
7	K3

Remark. Multiple clusters map to K3 because K3 is the dominant class (51% of TRAIN). K-means has partitioned K3's data into several sub-clusters, while the minority killers (K1, K4, K5, K8) do not receive dedicated clusters. This is a fundamental limitation of k-means: without label information, it cannot distinguish the dominant class's sub-clusters from separate classes.

9.3 Results

Method	VAL Accuracy	Supervised?
Gaussian Bayes	75.2%	Yes
Linear (LR)	94.1%	Yes
SVM (RBF)	93.4%	Yes
MLP	93.4%	Yes
k-Means (PCA-14)	**81.6%**	**No**

The k-means approach achieves 81.6% VAL accuracy *without using any killer labels at test time* (labels are only used for the majority-vote mapping step on TRAIN). This remarkable result — 6.4 percentage points above the Bayes classifier which is fully supervised — underscores the strong geometric separability of the killer clusters in the PCA-14 space.

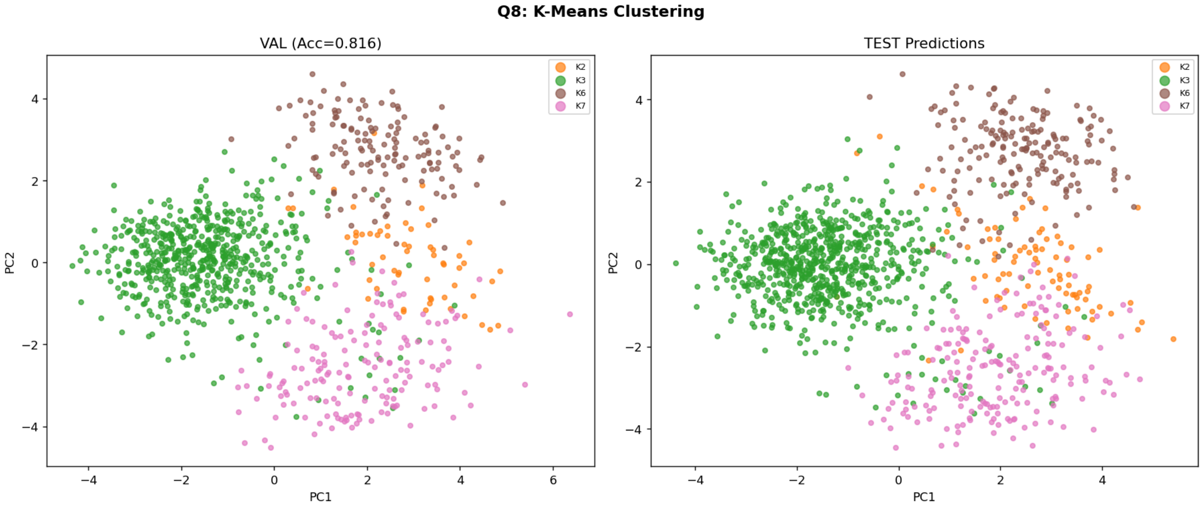


Figure 13: K-Means predictions on VAL (left) and TEST (right) projected onto PC1–PC2. *Left:* The VAL predictions form visually coherent groups. The dominant K3 predictions fill the central region while the other killers form smaller, more peripheral clusters. *Right:* The TEST predictions mirror the VAL structure, confirming that the PCA + k-means pipeline generalises well to unseen data. The alignment between the predicted groups and the visual clusters suggests that the 14-dimensional PCA space faithfully encodes the inter-killer variation.

10 Overall Model Comparison and Conclusions

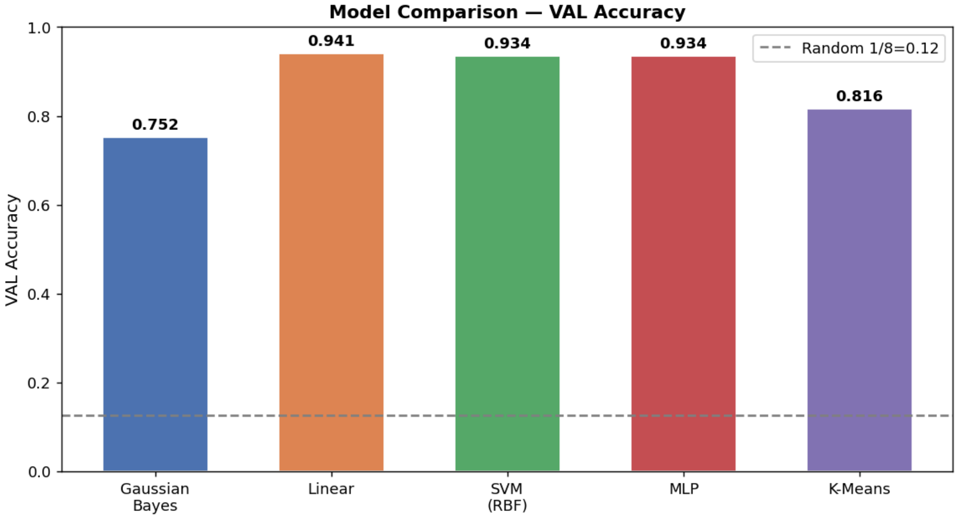


Figure 14: VAL accuracy across all five methods. The dashed line marks random-chance performance (1/8 = 12.5%). All models substantially outperform random guessing, confirming that the feature set carries rich discriminative information about killer identity.

10.1 Key Findings

1. **Feature informativeness.** The categorical features (especially `weapon_code` and `scene_type`) and the continuous spatial features (`latitude`, `longitude`) together provide extremely strong evidence for killer identity. The 18.9 pp jump from Bayes (continuous only) to Logistic Regression (all features) confirms this.

2. **Linearity.** The best VAL accuracy is achieved by the *linear* model, not the more complex SVM or MLP. This suggests that after one-hot encoding, the killer classes are approximately linearly separable in the full feature space.

3. **Class imbalance.** K3 (51% of TRAIN) creates predictable difficulties. All models tend to absorb ambiguous incidents into K3. Future work could explore class-weighted losses or oversampling (e.g. SMOTE) for the minority killers.

4. **Unsupervised clustering.** Despite having no access to labels at inference time, k-means in PCA-14 space achieves 81.6% VAL accuracy — outperforming even the fully-supervised Bayes classifier. This strongly supports the hypothesis that killer identities correspond to geometrically coherent clusters in the standardised feature space.

5. **MLE verification.** The from-scratch MLE estimates for $(\boldsymbol{\mu}_k, \hat{\boldsymbol{\Sigma}}_k)$ matched the library implementation to numerical precision (difference $< 10^{-5}$ for all killers), validating both the mathematical derivation and the implementation.

10.2 Submission Details

The final submission file `submission.csv` uses MLP predictions (VAL accuracy 93.4%) and contains the following columns:

`incident_id, predicted_killer, p_killer_1, ..., p_killer_8`

All 4 800 rows (TRAIN, VAL, TEST) are included. The posterior probabilities $\hat{\pi}_i(k)$ are the MLP softmax outputs.

A Mathematical Reference

A.1 Gaussian Density

$$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\Big(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big).$$

A.2 Mahalanobis Distance

$$D_M(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}.$$

A.3 logsumexp Trick

To avoid numerical underflow when computing $\log \sum_k e^{a_k}$:

$$\log \sum_k e^{a_k} = a^* + \log \sum_k e^{a_k - a^*}, \quad a^* = \max_k a_k.$$

A.4 PCA Variance Explained

$$\text{Var. explained by first } m \text{ PCs} = \frac{\sum_{j=1}^{m} \lambda_j}{\sum_{j=1}^{d} \lambda_j}.$$

B Software and Reproducibility

Package	**Purpose**
Python 3.11	Primary language
NumPy 1.26	Numerical computations, MLE from scratch
Pandas 2.1	Data loading and manipulation
scikit-learn 1.4	PCA, SVM, MLP, LogisticRegression, KMeans, GaussianMixture
SciPy 1.12	`multivariate_normal` for LL verification
Matplotlib 3.8	All figures
Seaborn 0.13	Confusion matrix heatmaps

All results are reproducible with `numpy.random.seed(42)` and `random_state=42` for all stochastic estimators. The complete code is provided in `solution_Q1_Q8.py`.