Quantum Binary Classification on the Palmer Penguins Dataset: A Comparative Study of VQC and QSVM Architectures

Asadullah Bin Rahman

October 31, 2025

Abstract

This study evaluates quantum machine learning models for binary classification using the Palmer Penguins dataset. Two quantum circuit architectures are implemented: a Variational Quantum Classifier (VQC) with trainable parameters and a Quantum Support Vector Machine (QSVM) employing kernel-based learning. All three species pairs—Adelie versus Chinstrap, Adelie versus Gentoo, and Chinstrap versus Gentoo—are analyzed with both exploratory data analysis (EDA)-driven and PCA-based feature selection applied. Circuit expressibility is assessed, performance metrics are compared, and the viability of these quantum approaches is evaluated against classical baselines. Findings indicate that competitive accuracy is achieved by quantum methods on easily separable tasks, but no advantage is observed over classical RBF-SVM on this dataset

1 Introduction

The convergence of quantum computing and artificial intelligence in quantum machine learning (QML) holds promise for tackling complex classification problems with potential computational benefits. In this work, we apply QML to biological data through binary classification of penguin species, drawing on the Palmer Penguins dataset to explore practical implementations.

Our primary aims include designing two quantum circuit architectures with varied encoding and ansatz strategies, applying suitable preprocessing techniques, evaluating circuit expressibility and representational capacity, comparing quantum methods to classical baselines across species pairs, and analyzing strengths, weaknesses, and deployment considerations.

2 Dataset and Preprocessing

The Palmer Penguins dataset[1] comprises measurements from 344 penguins across three species: Adelie (152 samples), Chinstrap (68 samples), and Gentoo (124 samples). It features four numerical attributes—bill length (mm), bill depth (mm), flipper length (mm), and body mass (g)—which we use for binary classification across all species pairs: Adelie versus Chinstrap (220 samples, moderate difficulty), Adelie versus Gentoo (276 samples, easy separation), and Chinstrap versus Gentoo (192 samples, easy separation).

Preprocessing begins with imputing missing values using species-specific medians to maintain inter-species distinctions, as defined by:

$$x_{ij}^{\text{imputed}} = \begin{cases} x_{ij} & \text{if not missing} \\ \text{median}(x_{\cdot j} \mid \text{species}_i) & \text{otherwise} \end{cases}$$

For feature selection, we adopt two approaches. In the EDA-driven method, pairplot visualizations and correlation analyses guide the choice of the two most discriminative features for each pair: flipper length and body mass for Adelie versus Chinstrap, bill length and flipper length for Adelie versus Gentoo, and bill depth and body mass for Chinstrap versus Gentoo. These selections maximize class separation while incorporating biological insights into species differences, as illustrated in the feature distribution plot (Figure 1).

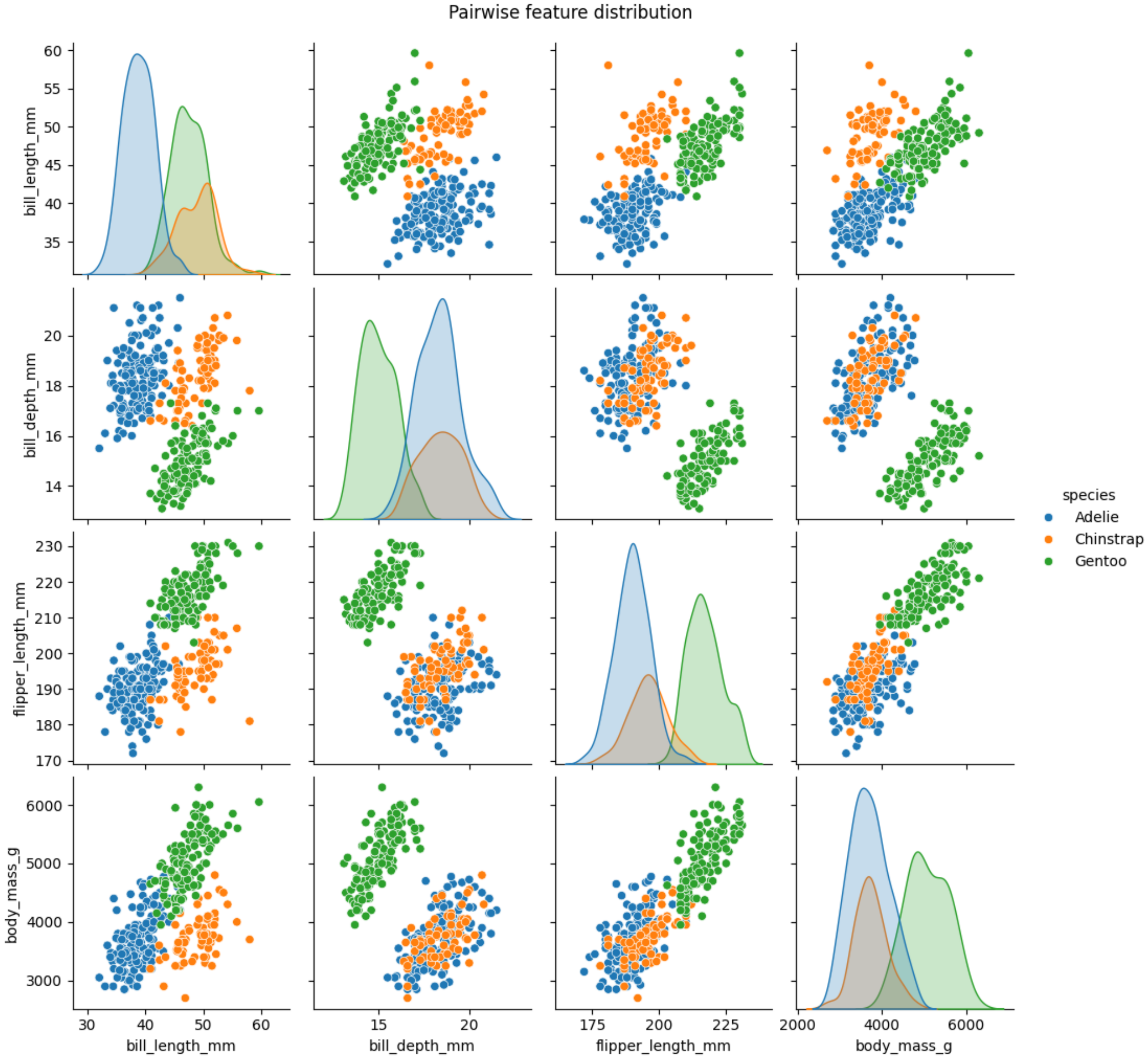

Figure 1: Feature distribution across species pairs.

Alternatively, PCA reduces all four features to two dimensions, capturing approximately 88% of the variance (explained ratios: 0.69 and 0.19) via:

$$\mathbf{X}_{\text{PCA}} = \mathbf{X}_{\text{standardized}} \cdot \mathbf{W}_{2D},$$

where $\mathbf{W}_{2D}$ comprises the top two principal components. This data-driven reduction preserves maximum variance and aligns with the two-qubit constraint for efficient quantum encoding.

Both feature sets are normalized to [0, 1] using MinMax scaling:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}},$$

ensuring compatibility with quantum angle encoding.

3 Quantum Circuit Architectures

3.1 Variational Quantum Classifier (VQC)

The VQC employs a two-stage design: state preparation followed by a parameterized ansatz. For state preparation, amplitude encoding embeds 2-feature inputs $\mathbf{x} = [x_1, x_2]$ by padding to four dimensions and normalizing: $\mathbf{x}_{\text{padded}} = [x_1, x_2, 0.1, 0.1]$, $\mathbf{x}_{\text{norm}} = \mathbf{x}_{\text{padded}}/\|\mathbf{x}_{\text{padded}}\|$. Rotation angles are then computed as:

$$\beta_0 = 2 \arcsin\left(\sqrt{x_2^2}/\sqrt{x_1^2 + x_2^2}\right), \quad \beta_1 = 2 \arcsin\left(\sqrt{x_3^2}/\sqrt{x_2^2 + x_3^2}\right), \quad \beta_2 = 2 \arcsin\left(\sqrt{x_2^2 + x_3^2}/\|\mathbf{x}\|\right).$$

The circuit applies a sequence of RY and CNOT gates to encode these angles.

The ansatz consists of three layers, each featuring single-qubit rotations $Rot(\phi_i, \theta_i, \omega_i)$ on each qubit and a CNOT for entanglement, yielding 18 trainable parameters plus a bias. For layer ℓ, the unitary is $U_\ell(\boldsymbol{\theta}_\ell) = CNOT_{0,1} \cdot \prod_{i=0}^{1} Rot(\phi_i^\ell, \theta_i^\ell, \omega_i^\ell)$. Measurement yields the expectation $\langle Z_0 \rangle + b$, with classification based on its sign.

Training uses Nesterov Momentum optimization ($\eta = 0.1$) to minimize mean squared error over 200 epochs with batch size 32. Parameters initialize from $\mathcal{N}(0, 0.01^2)$, bias at zero. This design balances expressibility through entanglement and rotations while mitigating barren plateaus in a three-layer setup.

3.2 Quantum Support Vector Machine (QSVM)

QSVM leverages kernel-based learning, encoding 2D features via angle embedding with RX gates: $|\psi(\mathbf{x})\rangle = RX(x_1) \otimes RX(x_2)|00\rangle$. The kernel measures state overlap: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = |\langle \psi(\mathbf{x}_i)|\psi(\mathbf{x}_j)\rangle|^2$, implemented by preparing $|\psi(\mathbf{x}_i)\rangle$, applying the adjoint of $\mathbf{x}_j$'s encoding, and measuring the all-zero probability.

Kernel matrices for training and testing feed into a classical SVM solver minimizing $\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij} - \sum_i \alpha_i$ subject to constraints. This shallow, hardware-efficient design avoids quantum training, relying on classical optimization in a high-dimensional Hilbert space.

4 Expressibility Analysis

For VQC, the 18-parameter, three-layer ansatz generates a subset of 2-qubit unitaries, with expressibility tied to Jacobian rank and depth. Single-qubit rotations span SU(2), while CNOTs enable entanglement, though measurement limits to $\langle Z_0 \rangle$. Empirical training shows instability, with accuracy fluctuating on hard tasks and stabilizing near perfection on easy ones, suggesting sensitivity to initialization and potential barren plateaus.

QSVM maps data to a Hilbert space via $\kappa(\mathbf{x}, \mathbf{x}') = \cos^2\left(\frac{x_1 - x_1'}{2}\right)\cos^2\left(\frac{x_2 - x_2'}{2}\right)$, akin to RBF but with trigonometric periodicity. It supports smooth non-linear boundaries but lacks tunable parameters, limiting flexibility compared to classical kernels.

5 Experimental Results

Using 5-fold stratified cross-validation, we report accuracy, F1-score, and training time, benchmarking against classical RBF-SVM ($C = 1.0$, $\gamma =$auto).

Pair	**Features**	**SVM Acc**	**VQC Acc**	**QSVM Acc**	**SVM F1**	**VQC F1**	**QSVM F1**
Ade vs Chi	EDA	0.736 ± 0.023	0.696 ± 0.011	0.691 ± 0.011	0.824	0.816	0.810
Ade vs Chi	PCA	0.827 ± 0.037	0.686 ± 0.009	0.841 ± 0.032	0.882	0.814	0.890
Ade vs Gen	EDA	0.989 ± 0.009	0.899 ± 0.082	0.993 ± 0.009	0.990	0.891	0.994
Ade vs Gen	PCA	1.000 ± 0.000	0.993 ± 0.009	1.000 ± 0.000	1.000	0.993	1.000
Chi vs Gen	EDA	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000	1.000	1.000
Chi vs Gen	PCA	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000	1.000	1.000

Table 1: 5-fold cross-validation results (mean ± std).

All methods excel on easy tasks (near 100% accuracy), but on the moderate Adelie versus Chinstrap pair, classical SVM leads (73–83%), with quantum variants at 69–84%. EDA features aid hard tasks, while PCA suits easy ones. Training times reveal classical SVM's efficiency (0.01 s per fold) versus QSVM (73.6 s) and VQC (198 s), highlighting quadratic and iterative overheads.

6 Discussion

No quantum advantage emerges, attributable to the dataset's low dimensionality and near-linear separability, where classical methods suffice without exponential costs. Feature engineering outweighs model type, with EDA often outperforming PCA by preserving discriminative traits.

VQC offers adaptability through parameters but suffers instability and high costs, while QSVM provides stability via shallow circuits but scales poorly. Deeper circuits do not assure better performance; QSVM's single-layer design outperforms VQC on hard tasks.

Noise considerations favor QSVM's low gate count for NISQ devices, though VQC may benefit from future fault-tolerant hardware. Overall, classical SVM proves optimal for this task in accuracy, speed, and reliability.

7 Conclusion

Quantum methods match classical performance on simple separations but lag on moderate challenges, with prohibitive computational demands. QSVM suits NISQ applications better than VQC due to its stability and shallower depth. This study underscores the need for problems with inherent quantum structures to realize QML advantages and highlights the enduring value of domain-informed preprocessing.

Acknowledgments

We acknowledge the Palmer Penguins dataset (Horst et al., 2020) and PennyLane framework. Code is available at Google Colab.

References

[1] Horst, A. M., Hill, A. P., & Gorman, K. B. (2020). *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*. R package version 0.1.0.

[2] Schuld, M., & Killoran, N. (2019). Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, 122(4), 040504.

[3] Havlíček, V., Córcoles, A. D., et al. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747), 209–212.

[4] McClean, J. R., Boixo, S., et al. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1), 4812.

[5] Cerezo, M., Sone, A., et al. (2021). Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1), 1791.

[6] Bergholm, V., Izaac, J., et al. (2018). PennyLane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968.*

[7] Schuld, M., & Petruccione, F. (2021). *Machine Learning with Quantum Computers*. Springer International Publishing.

[8] Biamonte, J., Wittek, P., et al. (2017). Quantum machine learning. *Nature*, 549(7671), 195–202.