**Task 4 QSVM**

Your objective is to design and evaluate quantum machine learning models for a binary classification problem using the Iris dataset.

Requirements

* + Create at least two different circuit-based proposals, each with distinct architectures and layers.
  + Clearly describe the design choices for each proposal (e.g., number of layers, type of parameterized gates, entanglement structure).
  + Use the Iris dataset restricted to a binary classification task (selecting two classes).
  + Apply appropriate classical preprocessing (e.g., normalization, dimensionality reduction, or feature selection).
  + Explain the rationale behind the chosen preprocessing methods.
  + Evaluate the expressibility of each circuit to identify its representational limits.
  + Discuss how the architecture affects the ability of the circuit to represent complex decision boundaries.
  + Compare their performance and analyze strengths and weaknesses.
  + Provide reasoning based on expressibility, accuracy, and practical considerations (e.g., scalability, circuit depth, or noise resilience).

**Core Definitions**

**1. Quantum Support Vector Machine (QSVM)**

A **Support Vector Machine (SVM)** is a classical supervised machine learning algorithm used for classification and regression. In a classification context, it finds the optimal **hyperplane** that separates the data points of different classes in the feature space, maximizing the margin between the classes.

The standard SVM relies on the kernel trick, where a kernel function, K(xi​,xj​), implicitly maps the input data points xi​ and xj​ into a high-dimensional feature space Φ:



The **Quantum Support Vector Machine (QSVM)** replaces this classical kernel function with a **Quantum Kernel** (or **Quantum Feature Map**).

* Quantum Feature Map (Φ(x)): This is a parameterized quantum circuit, UΦ​(x), that encodes the classical input data x into a quantum state ∣ψ(x)⟩ in the Hilbert space.



* Quantum Kernel (KQ​(xi​,xj​)): The kernel is computed as the overlap (fidelity) between the quantum states generated by two data points:



This overlap is classically hard to compute for certain feature maps, providing the potential for a **quantum advantage** by implicitly leveraging the exponentially large Hilbert space to find better, non-linear separations. The rest of the SVM training (finding the optimal hyperplane) remains a classical convex optimization problem.

**2. Quantum Circuit-Based Proposals (Quantum Feature Maps)**

These are the core components of the QSVM kernel. A proposal is defined by its **architecture** and **layers**.

* **Architecture (Ansatz)**: The specific structure of the quantum circuit, which includes:
  + **Data Encoding Gates**: Parameterized single-qubit rotation gates where the rotation angle is a function of the input data features.
  + **Entangling Gates**: Multi-qubit gates that link different qubits and introduce entanglement. This is crucial for capturing correlations between input features.
  + **Number of Qubits**: Determined by the number of features after pre-processing.
* **Layers**: A **layer** typically consists of one set of data-encoding gates followed by one set of entangling gates. The number of layers (or circuit depth) directly impacts the complexity, expressibility, and potential noise resilience of the circuit.

**3. Expressibility of a Quantum Circuit**

**Expressibility** (or **Power**) is a metric that quantifies the **representational capacity** of a Parameterized Quantum Circuit (PQC).

* **Definition**: It measures how uniformly the set of quantum states that a circuit can generate, , covers the entire reachable area of the Hilbert space (the distance to the uniform distribution).
* **High Expressibility**: A highly expressible circuit can generate a wide range of quantum states and, therefore, can potentially realize a greater variety of complex kernel functions and decision boundaries.
* **Trade-off**: While high expressibility is desirable for finding complex solutions, it can lead to **barren plateaus**—regions in the optimization landscape where the gradients are exponentially small—making the model difficult to train. Low expressibility suggests limited complexity and may result in an under-fitting model.

The expressibility of a circuit is typically calculated by measuring the statistical distance (e.g., the 3-design fidelity) between the set of states generated by the circuit and the entire space of possible states (the Haar measure), or by measuring its **effective dimension**.

Step-by-Step Approach

Refer qiskit code for more information (“QSVM\_MK.ipynb”)

1. **Dataset Selection and Preprocessing**

* **Binary Classification**: Use only two classes from the Iris dataset (e.g., Setosa vs Versicolor).
* **Feature Selection**: Choose 2 features (e.g., petal length and petal width) for simplicity.
* **Normalization**: Apply StandardScaler to ensure features are zero-mean/unit-variance—important for stable angle encoding.

2. **Circuit Proposal 1: Shallow Ansatz**

* **Encoding**: AngleEmbedding (RY rotations).
* **Layers**: 2 layers.
* **Parameterized Gates**: RY rotations.
* **Entanglement**: Linear entanglement (CNOT between adjacent qubits).
* **Pros**: Low depth, noise-resilient.
* **Cons**: Limited expressibility.

3. **Circuit Proposal 2: Deep Ansatz**

* **Encoding**: AngleEmbedding (RY + RX).
* **Layers**: 4 layers.
* **Parameterized Gates**: RY, RX, and CZ.
* **Entanglement**: Full entanglement (CNOT between all pairs).
* **Pros**: High expressibility.
* **Cons**: More prone to noise, harder to train.

4. **Expressibility Evaluation**

Use metrics like **Hilbert-Schmidt Test** or visualize decision boundaries to compare how well each circuit captures complex patterns.

5. **Performance Comparison**

* **Accuracy**: Use cross-validation.
* **Scalability**: Shallow circuits scale better.
* **Noise Resilience**: Fewer gates = less noise.
* **Training Stability**: Deep circuits may suffer from barren plateaus.