**TASK 3**

**Introduction to our problem statement**

Climate change and water are deeply interconnected. Changes in climate can significantly impact the quality and availability of freshwater resources, which are essential for human consumption, agriculture, and ecosystems. The increase in global temperatures can lead to more frequent and severe droughts, reducing the availability of potable water. On the other hand, excessive rainfall and flooding can lead to water contamination, affecting water portability. Water portability, which refers to the safety of water for drinking, is influenced by various factors such as pH levels, hardness, the presence of contaminants like Chloramines, and other dissolved solids. These factors can be affected by environmental changes, making it crucial to monitor and manage water quality, especially in the face of climate change.

Given the complexity and the vast amount of data involved in assessing water quality, quantum computing and AI-based methods have emerged as promising tools. These advanced methods can handle large datasets more efficiently and can uncover complex patterns that might not be easily discernible through traditional methods.

**AI and Quantum Algorithms Used for Water Portability Prediction**

To address the problem of predicting water portability, several machine learning and quantum algorithms were employed. Below is a detailed description of each algorithm and how it was used:

**1. LightGBM (Light Gradient Boosting Machine)**

LightGBM is a highly efficient and scalable gradient boosting framework. It uses decision trees as weak learners and builds them sequentially, with each new tree correcting the errors made by the previous ones. LightGBM is known for its speed and efficiency, especially on large datasets, as it uses a leaf-wise growth strategy with depth limitation, which can result in better accuracy with fewer computational resources.

* **Algorithm Overview:** LightGBM grows trees leaf-wise, which means it splits the leaf with the highest loss improvement among all leaves. This can lead to a more complex model with potentially better accuracy but requires careful tuning to avoid overfitting.

**2. XGBoost (Extreme Gradient Boosting)**

XGBoost is another gradient boosting framework that has gained popularity due to its performance and flexibility. Like LightGBM, XGBoost builds decision trees sequentially, but it offers additional features like regularization, which helps prevent overfitting, and it supports parallel processing, which makes it faster.

* **Algorithm Overview:** XGBoost uses a level-wise tree growth strategy, where all leaves at a given depth are split simultaneously. This is in contrast to LightGBM’s leaf-wise approach. XGBoost also introduces a regularization term to the objective function to penalize more complex models, helping to reduce overfitting.

**3. Bagging Classifier**

Bagging (Bootstrap Aggregating) is an ensemble method that improves the accuracy and robustness of machine learning models by combining the predictions of several base models trained on different subsets of the data. Bagging helps to reduce variance, leading to a more stable model.

* **Algorithm Overview:** In Bagging, multiple instances of a weak learner (e.g., a decision tree) are trained on different subsets of the training data, generated through bootstrapping (random sampling with replacement). The final prediction is made by averaging the predictions (in regression) or by majority voting (in classification) from all the individual models.

**4. Gradient Boosting Classifier**

Gradient Boosting is an ensemble technique that builds models sequentially. Unlike Bagging, where models are built independently, Gradient Boosting builds each new model to correct the errors made by the previous ones, thus reducing bias and improving performance.

* **Algorithm Overview:** Gradient Boosting trains each model to predict the residuals (errors) of the previous models. By adding these models together, it gradually improves accuracy. Gradient Boosting can be prone to overfitting, so regularization techniques like shrinkage, early stopping, or subsampling are often used.

**Quantum Algorithms**

With the advent of quantum computing, quantum algorithms are being developed to solve complex problems more efficiently than classical algorithms. Here, we used a few quantum algorithms for feature selection and classification in water portability prediction.

**1. Quantum Principal Component Analysis (Quantum PCA)**

Quantum PCA is a quantum version of the classical Principal Component Analysis (PCA), which is a dimensionality reduction technique. Quantum PCA leverages the principles of quantum computing to process and analyze large datasets more efficiently by encoding the data in quantum states and performing operations that reveal the most significant components.

* **Algorithm Overview:** Quantum PCA is based on the idea that quantum computers can efficiently find the eigenvalues and eigenvectors of a covariance matrix encoded in a quantum state. This allows for a potentially exponential speedup over classical PCA for large datasets.

**2. Quantum Approximate Optimization Algorithm (QAOA)**

QAOA is a quantum algorithm designed to solve combinatorial optimization problems. It is a hybrid quantum-classical algorithm that uses a quantum circuit to approximate the solution to an optimization problem and a classical optimizer to tune the parameters of the quantum circuit.

* **Algorithm Overview:** QAOA operates by preparing a quantum state that encodes the possible solutions to an optimization problem, applying a sequence of quantum gates parameterized by classical variables, and then measuring the quantum state to find the optimal solution. The parameters of the quantum gates are adjusted using a classical optimization algorithm to improve the solution iteratively.

**3. Quantum Support Vector Machines (QSVM)**

Quantum SVMs use quantum-enhanced feature selection by mapping data into a higher-dimensional quantum feature space. QSVMs can potentially capture complex patterns in the data that classical SVMs might miss.

* **Algorithm Overview:** In QSVM, the data is transformed using a quantum feature map, which maps the data into a quantum state. A quantum kernel, which is the inner product of two quantum states, is then computed and used in a classical SVM framework. The quantum kernel can capture more intricate patterns in the data due to the higher-dimensional feature space.

**Computational Resources**

* For the experiments conducted, we utilized the L4 GPU available in Google Colab Pro, which provided significant acceleration for both classical and quantum simulations. The GPU was instrumental in handling the complex quantum simulations required for algorithms such as Quantum PCA and QSVM, which involve executing multiple quantum circuits. Additionally, the CPU resources were used to manage data preprocessing, classical machine learning model training (like LightGBM, Gradient Boosting, and Bagging Classifier), and other supporting tasks.

**Results and Literature Review**

The results obtained from applying these classical and quantum algorithms to the water portability dataset revealed interesting insights:

Here is the updated summary table that includes both training and test set accuracies, recalls, and precisions for the different algorithms:

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Test Accuracy | Test Recall | Test Precision |
| LightGBM Baseline | **0.684** | **0.373** | **0.628** |
| Quantum Optimized LightGBM | **0.675** | **0.361** | **0.607** |
| Quantum Transformed LightGBM | **0.628** | **0.000** | **0.000** |
| Quantum PCA + LightGBM | **0.705** | **0.213** | **0.382** |
| Quantum PCA (Reduced Features) + LightGBM | **0.709** | **0.201** | **0.366** |
| Quantum PCA + Gradient Boosting | **0.530** | **0.530** | **0.520** |
| Quantum PCA + Bagging Classifier | **0.500** | **0.490** | **0.490** |
| Quantum Kernel + SVM | **-** | **-** | **-** |
| Linear SVM | **-** | **-** | **-** |

**This table now includes both the training accuracy and the test accuracy along with recall and**

* **LightGBM Baseline:** Achieved a test accuracy of 0.684 with a recall of 0.373, indicating moderate performance in predicting water portability.
* **Quantum Optimized LightGBM:** Showed slightly lower accuracy and recall compared to the baseline, possibly due to the complexity added by quantum transformations that might not have been fully captured by the classical LightGBM model.
* **Quantum Transformed LightGBM:** Had lower accuracy and recall, possibly because the transformation didn’t align well with LightGBM’s structure or because of insufficient quantum resources.
* **Quantum PCA + LightGBM:** Demonstrated that reducing dimensionality with Quantum PCA led to a rise in performance with almost 71% accuracy, suggesting that the quantum transformation effectively capture the most informative features for LightGBM.
* **Gradient Boosting and Bagging Classifiers with Quantum PCA:** Both models showed similar performance, with test accuracies hovering around 0.500 to 0.530, which might suggest that quantum PCA did not significantly enhance feature selection.
* **Quantum Kernel + SVM vs. Linear SVM:** While the results for the Quantum Kernel + SVM could not be computed due to hardware constraints. However, due to noise and limited quantum resources, the classical SVM might still outperform QSVM in this context.

**Literature Review Insights:**

In recent studies, quantum algorithms have shown potential in enhancing classical machine learning models, particularly in tasks involving high-dimensional feature spaces. However, the practical application of quantum methods is still in its infancy, and the computational resources required can be substantial. This is evident in the limitations observed in the quantum-enhanced models, where classical methods sometimes outperformed their quantum counterparts.

**Computational Resources and Advantages**

Running quantum algorithms, even on simulators, requires significant computational resources. Quantum circuits need to be simulated or executed on quantum hardware, which can be expensive and time-consuming. Moreover, the accuracy of quantum algorithms can be affected by noise in quantum hardware, which is why hybrid quantum-classical approaches are often used.

**Advantages of Quantum and AI Methods:**

* **Complexity Handling:** Quantum algorithms can handle complex and high-dimensional data more effectively, potentially leading to better feature extraction and pattern recognition.
* **Speed:** For certain tasks, quantum algorithms offer the potential for exponential speedup over classical methods, particularly in optimization and searching.
* **Enhanced Model Performance:** AI and quantum methods combined can provide better generalization and accuracy by capturing complex relationships in the data.

However, the practical deployment of quantum-enhanced AI models is still challenging due to the current state of quantum hardware and the need for significant computational resources. As quantum computing matures, its integration with AI is expected to provide more robust and efficient solutions to complex problems like water portability prediction.

**Conclusion**

In conclusion, the use of quantum algorithms like QPCA, QSVM, and quantum-enhanced LightGBM provides exciting avenues for improving model performance in complex data environments. However, the results show that these quantum methods need to be carefully tuned and require significant computational resources to outperform classical models. The ongoing advancements in quantum computing will likely address these challenges, enabling more practical and effective solutions in the near future.