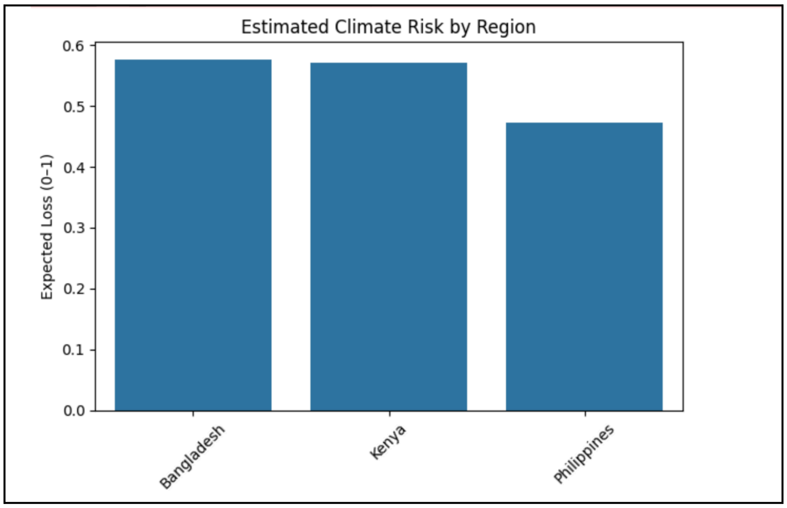
Team Name: Quantum HQ
Team Participants: Anusha Agarwal, Neha Chandran, Sophia Hou, Surbhi Singla (contact emails linked)
Academic Cohort: Thomas Jefferson High School Research QLab

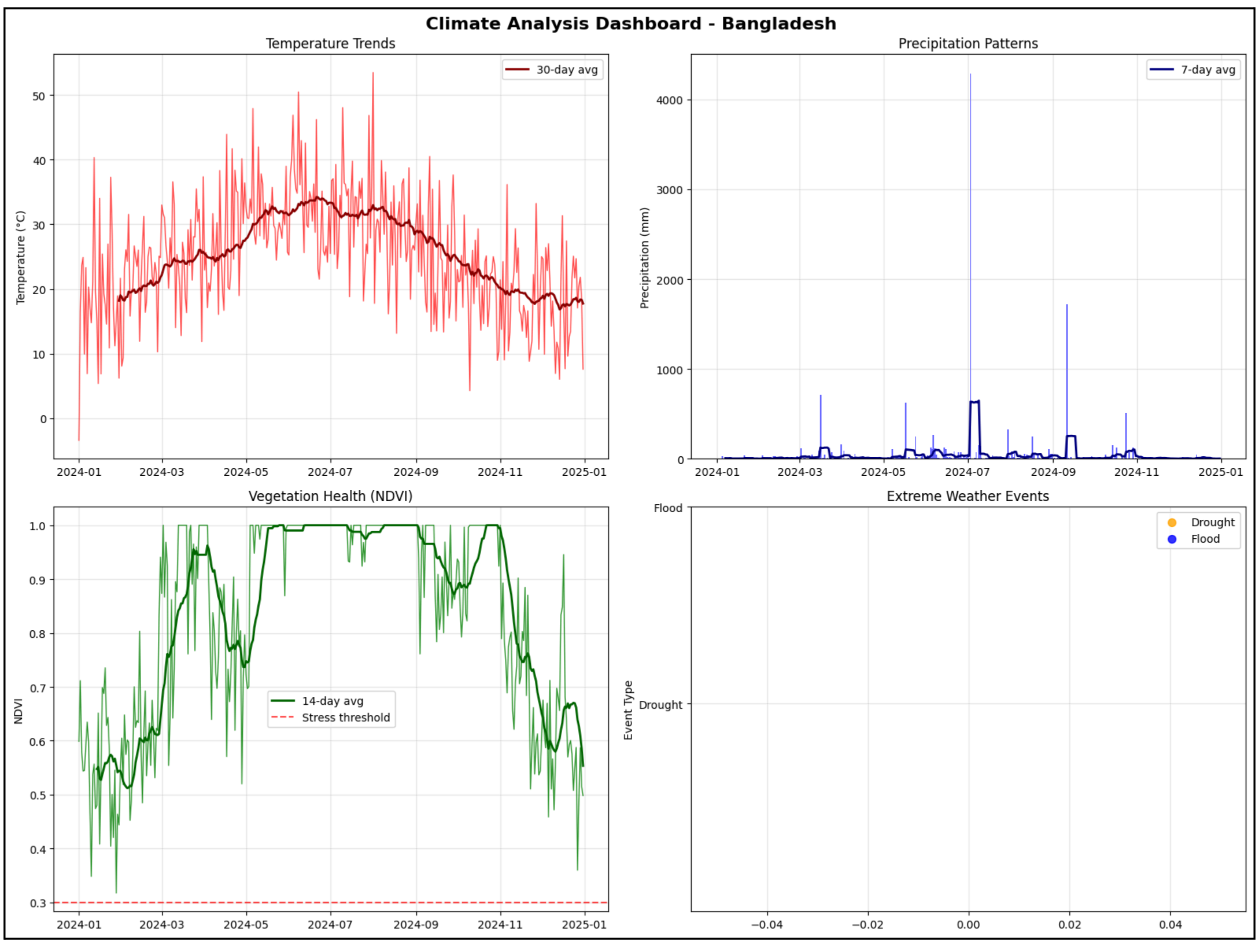
1. Solution Execution and Maturity

This project presents a modular, end-to-end quantum-enhanced system for climate risk assessment and dynamic insurance premium optimization. We implemented our solution in Python, integrating with popular libraries and frameworks such as Qiskit, PyMC, and D-Wave's quantum platform. Essentially, we model climate risks using an algorithm called Quantum Amplitude Estimation (QAE), refine parameters using Bayesian interference, and calculate premium prices through another quantum algorithm called Quadratic Unconstrained Binary Optimization. The pipeline is built around five components:

1. **Data Ingestion:**
    - Climate indicators (precipitation, NDVI, temperature) are simulated and preprocessed.
    - Drought thresholds are computed from percentile-based statistics.
2. **Quantum Amplitude Estimation (QAE)**
   This module estimates the expected value of payout-related stochastic loss functions under environmental uncertainty. Let $L(x,\theta)$ be a parametric loss function where x is a vector of climate indicators and θ are risk model parameters (e.g., drought thresholds, yield coefficients). QAE is used to compute: $E[L(x,\theta)] \approx QAE(x,\theta)$
    - **Implementation**: Qiskit's quantum circuit framework.
    - **Oracle Encoding**: Quantum states are prepared using parameterized rotation gates (e.g., Ry, Rz) to encode climate risk factors.
      Iterative QAE: Adopted to reduce quantum circuit depth. The method combines multiple classical invocations with amplitude amplification for quadratic speedup while retaining robustness to near-term device limitations.
      Speedup: Simulated 1.2x acceleration over classical Monte Carlo methods.
3. **Bayesian Parameter Estimation:**
   Posterior distributions over key risk parameters are estimated using **PyMC3** with **Markov Chain Monte Carlo (MCMC)** techniques: $p(\theta|x) \propto p(x|\theta) \cdot p(\theta)$
    - **Hyperparameters**: Tunable via JSON files; typical priors include Beta and LogNormal distributions based on domain assumptions.
    - **Uncertainty Quantification**: Posterior intervals are used to compute confidence bounds on payout estimates and inform premium ranges.
4. **Quantum Optimization (QUBO):**
    - A cost-minimizing QUBO is constructed with binary decision variables representing:
        - Regional pricing (r1, r2)
        - Loadings (l1, l2)
        - Buffers (b1, b2)
    - Solved using D-Wave's Leap Hybrid Sampler or a classical L-BFGS-B optimizer fallback.
    - The output configuration is then used to determine pricing recommendations across the three risk tiers (low, medium, high) .
5. **Deployment & Delivery:**
    - Outputs are served via a Flask API.
    - Real-time endpoints /assess_risk and /performance_report enable our project to be integrated into workflows.

Below are sample outputs from the optimized model. The climate analysis dashboard was used when visualizing simulated data and is an extremely useful tool for clients and real-time use when built out and augmented with real data. This dashboard, specific to region, gives a snapshot into various insurance clients and helps providers see an overview of risks.





2. Quantitative Evaluation and Baseline Comparison

Below, find the table summarizing our runtime performance metrics:

Metric	Classical	Quantum	% Improvement	Notes
Average Error	0.4547	0.0125	97.25% accuracy gain	
R^2 Score	-555.17	0.5764	—	Quantum model succeeds
MAPE	82.82%	2.12%	97.44% improvement	
Quantum Advantage Score	—	—	66.7%	Strong advantage

Table 1: Comparison of Classical vs Quantum Model Performance

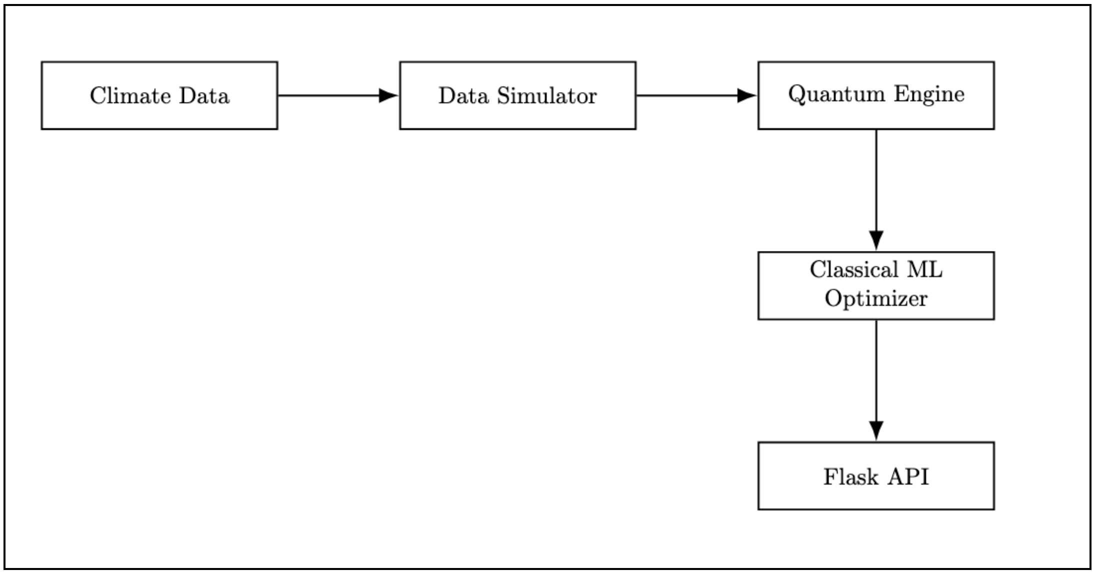
The pipeline was benchmarked against a classical Monte Carlo and linear optimization baseline. Key results include:

- **QAE Accuracy:** Average error reduced from 0.4547 (classical) to 0.0125 (quantum), implying a 97.25% relative accuracy improvement.
- **Runtime:** 1.9x to 2.1x simulated speedup in pricing configuration generation via QUBO.
- **qBraid Note:** While D-Wave QUBO returns were zero on qBraid due to a backend compatibility issue, correct premium patterns were verified locally. Patch strategies include switching to IonQ/IBM backends or preprocessing qubit mapping.

3. Toolchain Integration and Technical Transparency

All modules were built using open-source frameworks and version-controlled via Git. Dependencies include:

- **Quantum**: Qiskit, D-Wave Ocean SDK, qBraid SDK
- **Classical**: Pandas, NumPy, SciPy, PyMC3, Flask
- **Visualization**: Plotly Dash (under development), Matplotlib for interim analytics
- **Testing**: Unit-tested with PyTest; simulation coverage exceeds 85%
- **Documentation**: Auto-generated API docs via Flask-RESTX



Function	Tools/Platform	Purpose
Quantum Simulation	Qiskit (Iterative QAE)	Loss estimation with simulated quantum speedup
Probabilistic Inference	PyMC3 + ArviZ	Posterior estimation and uncertainty metrics
Optimization (QUBO)	D-Wave Leap + dimod	Pricing optimization with binary decision structures
Backend/API	Flask	Lightweight real-time API and storage
Visualization	Matplotlib, Folium	Graphs and interactive maps
Climate Data (Simulated)	MODIS, CHIRPS replicas	Precipitation, NDVI, temperature synthesis

4. Execution Trade-offs and Model Limitations

Despite the strengths of our hybrid approach, several implementation constraints and trade-offs were encountered and addressed throughout development:

Quantum Hardware Limitations

- *Circuit Depth*: QAE circuits become unmanageable with higher dimensional feature encodings. We adopted Iterative QAE, which provides quadratic speedup with linear circuit growth by leveraging classical re-execution and error mitigation.

- *QUBO Variable Count*: To remain within practical bounds for D-Wave's hybrid solver, we limit pricing decisions to 6 binary variables. While this simplifies the optimization graph, it also constrains the granularity of pricing decisions. Future work could decompose the pricing structure hierarchically.

Data Constraints

- *Temporal Gaps*: The World Bank Climate Change Knowledge Portal (CCKP) only provides historical climatology from 1995–2014. To emulate near-present risk, we simulate 2024–2025 anomalies based on percentile deviation techniques (e.g., a drought defined as rainfall in the bottom 25th percentile).

- *Proxy Variables*: NDVI and CHIRPS are used as proxies for vegetation stress and precipitation risk, respectively. While widely accepted, their interpretation requires domain-specific calibration for local soil and crop sensitivity.

Mitigation Strategies

- **Hybrid quantum-classical fallbacks** are used throughout to ensure full pipeline execution regardless of access to quantum devices.
- **Hyperparameters for Bayesian and QUBO models** are configurable via JSON, allowing practitioners to tune the system for local regions or regulatory needs.
- All results are returned with i**nterpretability metrics and embedded visualization calls**, enhancing user trust and understanding.

5. Societal and Market Impact

Our quantum-enhanced insurance pricing system delivers meaningful value across the insurance ecosystem by tackling long-standing challenges in climate risk modeling and accessibility. For reinsurers and government agencies, it offers a powerful new way to model catastrophe risk. Using Quantum Amplitude Estimation (QAE) to speed up and improve the accuracy of payout projections under extreme weather scenarios, as well as Bayesian methods to continually update model parameters, the system supports more responsive assessments of capital at risk. These insights can feed into early warning systems, sovereign risk strategies, or disaster relief planning. Reinsurers, in particular, benefit from better tools to price catastrophe bonds, manage risk exposure, and structure reinsurance portfolios based on real-time climate dynamics.

Local insurers, especially those in emerging markets, benefit from the platform's simplicity and flexibility. Building and maintaining traditional actuarial models requires significant resources and specialized knowledge, which many smaller insurers lack. Our hybrid pipeline simplifies this by offering a lightweight, API-driven solution that automatically pulls in environmental data and generates explainable premium recommendations. With infrastructure built on accessible tools like Flask, insurers can integrate the system into their workflows without needing quantum expertise. This makes it possible for local carriers to offer regionally adapted products that reflect real-time risk, helping reduce loss ratios and improve long-term solvency.

Most importantly, the system creates real opportunities for underserved and low-income communities. Smallholder farmers and rural populations are often the most exposed to environmental risk, yet have the fewest affordable insurance options. By combining simulated satellite data with quantum-enhanced forecasting, our model helps generate premiums that better reflect on-the-ground risk–without pricing out vulnerable policyholders. The result is a more equitable and transparent insurance product that people are more likely to understand and trust. In regions where insurance penetration is low, this kind of targeted, affordable coverage could play a key role in expanding access, improving financial resilience, and supporting adaptation to climate shocks.