QINTEL Finance v0

Quantum-Integrated Intelligence Engine for Financial Filtering

QINTEL Finance is a quantum-inspired data filtering engine designed to simulate quantum-style decision logic for financial datasets. It lets users define custom filtering conditions over stock market or investment data — and runs a simulated quantum circuit that evaluates those conditions using Qiskit.

□ What Does It Do?

- Loads a .csv dataset of financial/investment/stocks data.
- Lets you select **which columns (factors)** to evaluate.
- Lets you define **custom filtering rules** (e.g., Price < 100, Volume > 5000, or == True).
- Simulates a quantum circuit that uses X-gates to encode logic and filters for all-1 match vectors.
- Outputs the list of **matching stocks** into a new .csv file.

⚛ The Quantum Circuit (Simplified Overview)

For N factors selected:

- The engine creates an N-qubit register.
- It **flips qubits using X-gates** based on whether a stock matches the filtering condition on each factor.
- If all N qubits are in state $|1\rangle$, the stock is considered a match.
- The circuit includes a final measurement step (not logically required but completes the operation).

This system mimics the idea of using quantum states to represent matching data points — and while it runs classically, the logic structure allows for future conversion to Grover-style quantum search.

📦 How to Use

1. Prepare Your Dataset

- Use a .csv file with financial data.

- Each column should represent a **stock attribute** (like Price, Volume, RSI, etc.).
- Optionally include a "Stock Name" column for easy labeling.

2. Run the Script

```
python QINTEL_Finance_v0.py your_dataset.csv
```

3. Answer Prompts

- Frequency (daily / weekly / monthly)
- Column indices to use as filter factors
- For each column, enter a condition like:

Example:-

< 100

== True

> 3000

4. Output

A filtered list of stocks that match **all** the specified conditions will be saved as:

daily_qintel_output.csv (or weekly/monthly based on your input)

✅ Dataset Format Rules

- Accepted formats: **.csv**
- Headers: Column names should not have special characters.
- **Boolean values** must be written as True or False (capitalized).
- Conditions like '== True', '> 100', or '< 50.5' are supported.
- String-based filtering is currently **not supported** (and gets ignored in the current version).

🚀 Future Plans

- Optimize the logic for faster batch filtering using quantum sampling techniques.
- Upgrade the simulation model to support **actual Grover's algorithm** for search.
- Add support for **multi-condition filtering per factor** (e.g., between 100 and 200).

- Include **string-based matching**, fuzzy logic, and symbolic AI extensions.
- Convert filtering engine into a **full QML pipeline** for reinforcement-style investment modeling.

⚠ Disclaimer

If you encounter any bugs, errors, or unexpected behavior — please know that:

- This was developed by a 19-year-old quantum researcher with no formal lab access or hardware simulation tools.
- Any issues will be **fixed in future versions**.
- Your patience and support during this early development stage are deeply appreciated.

We intend to make QINTEL faster, more intelligent, and **more quantum-powered than ever** in upcoming releases.