Quantum Feature Selection

QC Final Project

Table of Content

Motivation

Motivation

Why Feature Selection matters ?

- Reducing dimensionality improves model performance, robustness, and interpretability.
- Selecting informative and non-redundant features is crucial for generalization, especially in biomedical and real-world datasets.

Problem

- Classical methods often struggle with combinatorial complexity and redundancy among features.

Why Quantum?

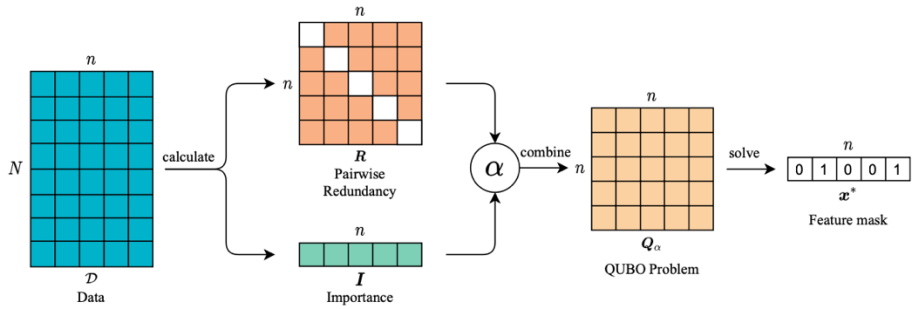
- Feature selection can be naturally formulated as a combinatorial optimization problem.
- Quantum algorithms can potentially explore exponentially large search spaces more efficiently.

Object

Implement and evaluate a **QUBO-based Quantum Feature Selection (QFS)** framework. This work serves as a step toward understanding the feasibility and limitations of **quantum-native feature selection**.

Algorithm

Algorithm - Formulation



Pipeline Overview

Step 1: Compute

- **Redundancy matrix** $R \in R^{n \times n}$
- **Importance vector** $I \in R^n$

Step 2: Combine R and I using interpolation factor α

Construct QUBO matrix $Q(\alpha, \epsilon, \mu)$

Step 3: Solve QUBO via quantum or classical methods

Output binary vector $x^* \in \{0,1\}^n$: selected features

QUBO Objective Formulation:

$$x^* = \min_{x \in \{0,1\}^n} x^\top Q(\alpha) x \quad s.t. \ \| x \|_1 = k$$

Key Idea

Feature selection is turned into a QUBO optimization problem combining:

- High feature importance
- Low redundancy
- Cardinality constraint k

Algorithm - Solver

Binary Search on α

Goal: Find interpolation factor α such that the solution selects exactly k features

Procedure:

- Initialize search bounds a=0, b=1
- Repeat until $\| x^* \|_1 = k$
- Update $\alpha = (a+b)/2$, re-solve QUBO

Quantum Solvers for QUBO

The QUBO matrix Q can be mapped to an **Ising Hamiltonian**:

$$H = \sum_{i \neq j} a_{ij} \sigma_i^z \sigma_j^z + \sum_i b_i \sigma_i^z + c$$

Find ground state $|\Psi\rangle \Rightarrow x^*$ with lowest energy

Algorithm 1 Our proposed QFS Algorithm: Binary search that, given an integer k, finds a value α^*, such that the optimal solution of a feature selection QUBO problem with matrix elements $Q_{ij}(\alpha^*, \epsilon, \mu)$, Eq. (18), contains k features.

Data: $\boldsymbol{R}$, $\boldsymbol{I}$, ϵ, μ, k
Result: α^* and $\boldsymbol{x}^*$ with $\|\boldsymbol{x}^*\|_1 = k$

```
a ← 0
b ← 1
α ← 0.5
x* ← Q*(α, ε, μ)
k' ← ||x*||_1
while k' ≠ k do
    if k' > k then
        b ← α
    else
        a ← α
    end
    α ← (a + b)/2
    x* ← Q*(α, ε, μ)
    k' ← ||x*||_1
end
return α, x*
```

Method

Data Preparation – Simulated Data

Simulated data constructed based on Mücke et al. (2023) to evaluate feature selection accuracy under controlled ground truth.

Regression

- 10 features, 4 informative (randomly selected)
- Target = linear combination of informative features + noise
- Feature values sampled from N(0, 1)
- Ground truth informative indices available

Classification

- 10 features, 4 informative
- Label = thresholded linear sum of informative features
- Feature values sampled from N(0, 1)
- Balanced binary labels ($y \in \{0, 1\}$)

Data Preparation – Real Data

Dataset	Type	Features	Samples	Source	Description
Breast Cancer	Classification	30	569	sklearn	Diagnosis of malignant vs benign tumors
Digits	Classification	64	1,797	sklearn	Handwritten digit recognition
Ionosphere	Classification	34	351	OpenML	Radar signal classification
Diabetes	Regression	10	442	sklearn	Predicting disease progression
California Housing	Regression	8	20,640	sklearn	Predicting median house prices in CA

Implementation

Algorithm Implementation

QUBO Formulation

- Feature selection is cast as a QUBO problem:

$$Q = R - \alpha\big(R + diag(I)\big)$$

Binary Search for α

- Goal：find α such that selected features ≈ target number k
- Uses Simulated Annealing (SA) to solve QUBO at each α step

Extension for Regression

- Target y is discretized using quantile binning
- Apply MI between binned y and features
- Redundancy still uses MI between features

Benchmark Comparison

- **Compared four methods** on classification and regression：
  - **Random** feature subset
  - **All** features
  - **Top-k MI** ranked features
  - **QFS-selected** features
- **Evaluated using**：
  - Accuracy (classification)
  - MSE (regression)

Feature Quality Analysis

This section evaluates the **stability**, **predictive performance**, and **statistical significance** of the features selected by the QFS method.

Sensitivity Analysis

Experimental Setup

- Fix target number of features k
- Sweep $\alpha \in [0.0, 1.0]$ in increments of 0.1

Metrics Tracked

- Accuracy (for classification tasks)
- MSE (for regression tasks)
- Jaccard similarity between selected feature sets (for α stability)

Importance Analysis

Task Type	Test	Metric
Classification	t-test (2-sample)	p-value, Cohen’s d
Regression	Pearson correlation	correlation, p-value

All statistical tests were applied to the **selected features** from QFS (SA) for each dataset.

VQE Implementation

We implemented a quantum solution to the QUBO-based feature selection problem using the **Variational Quantum Eigensolver (VQE)** framework on a quantum simulator.

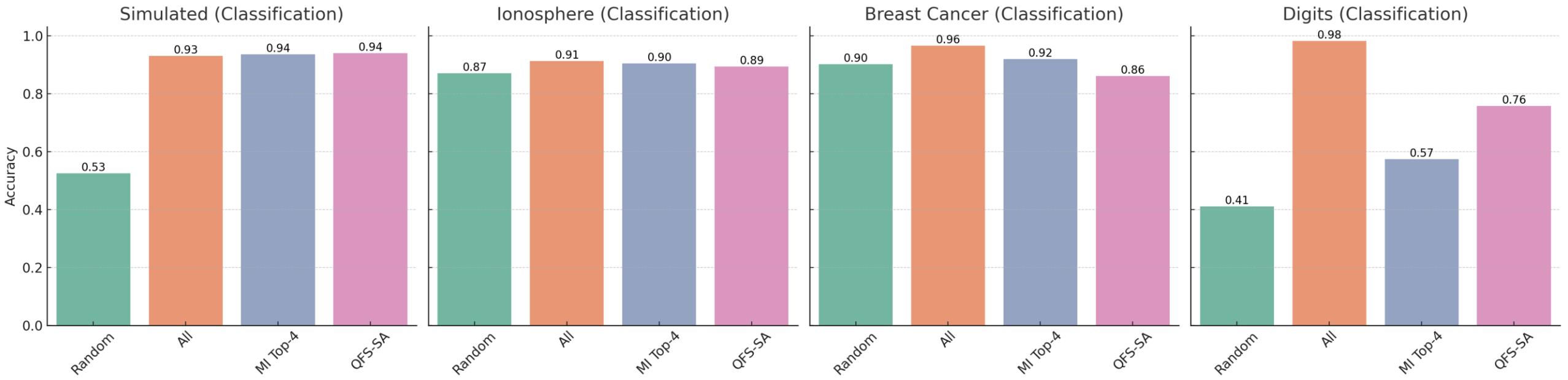
- **Convert QUBO → Ising Hamiltonian**

$$H(x) = \sum h_i Z_i + \sum J_{ij} Z_i \; via \; transformation \; Q \rightarrow (h, J) Z_j$$

- **Define Ansatz：**Use a TwoLocal ansatz (ry + cz, 1–2 reps) with COBYLA optimizer
- **Run VQE on Aer simulator**
- **Sample bitstring from optimal parameters：**Bind parameters to circuit、Measure、Select most frequent bitstring x^*
- **Recover selected features**：Interpret x^* as binary mask、Train Random Forest → Evaluate Accuracy

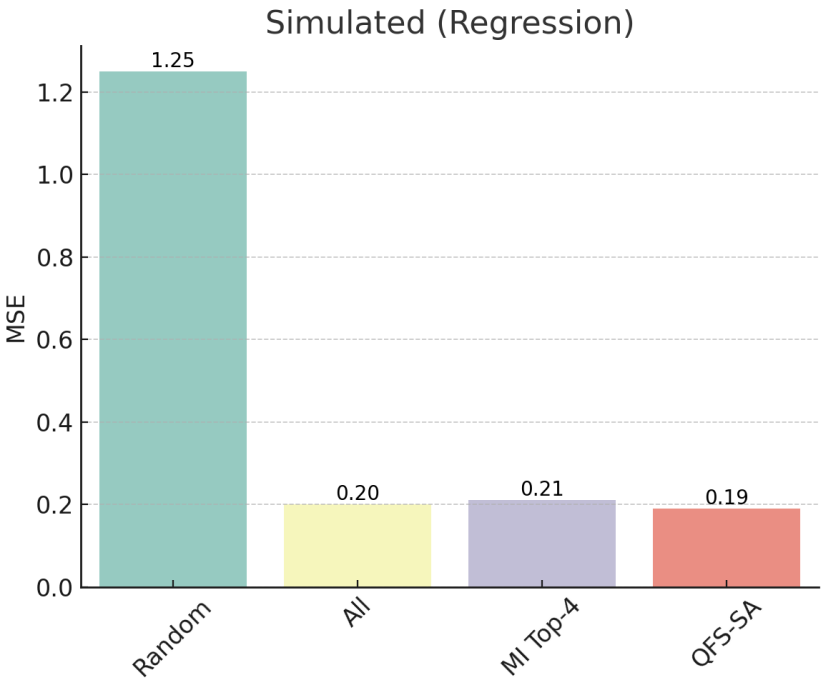
Result

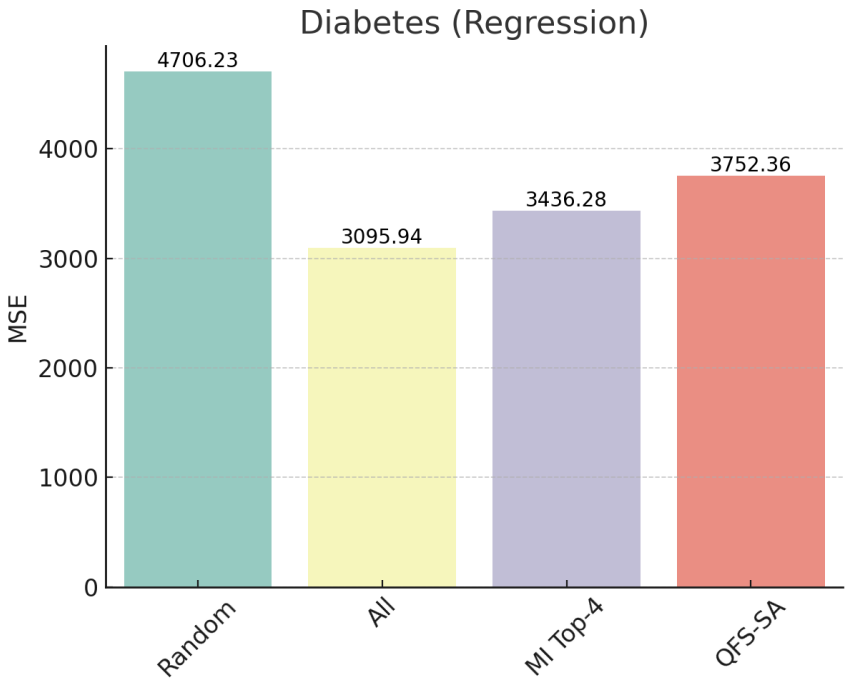
Benchmark Performance - Classification

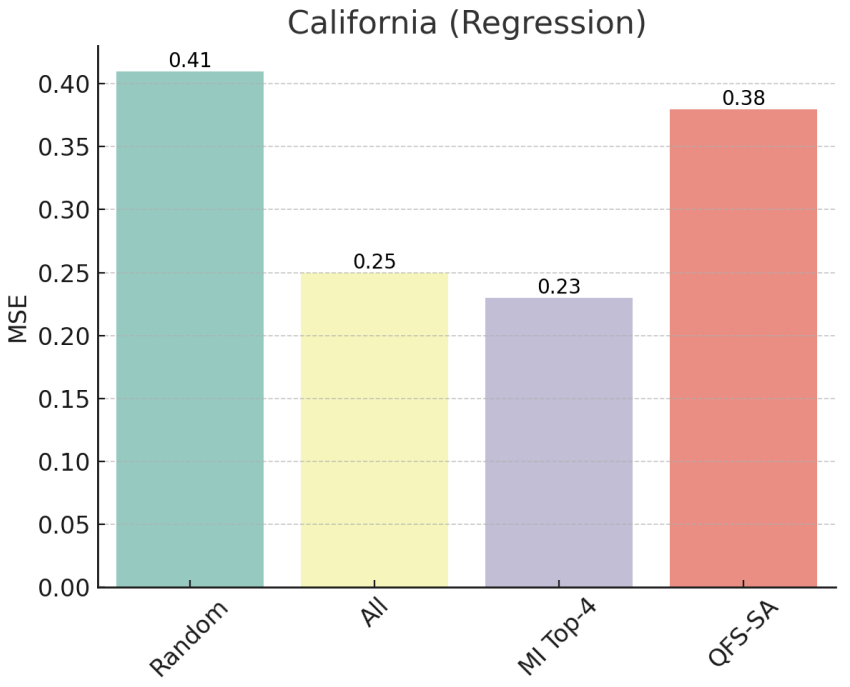


Conclusion QFS consistently outperforms random selection and matches MI-based methods in accuracy, using fewer features (k = 4).

Benchmark Performance - Regression



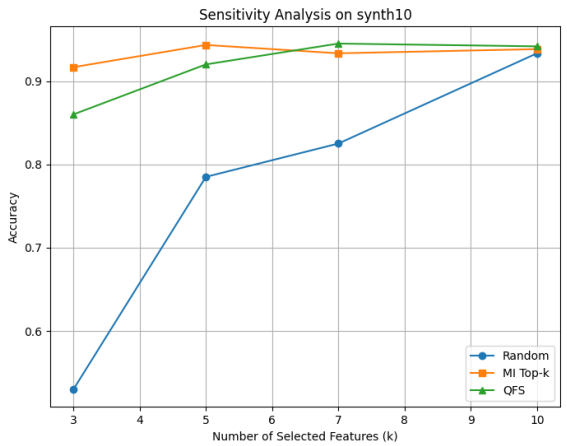


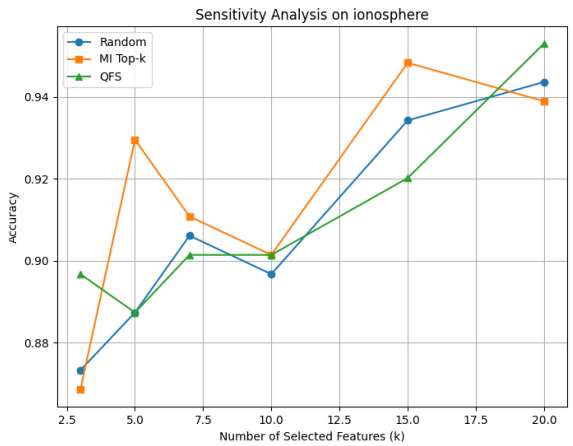


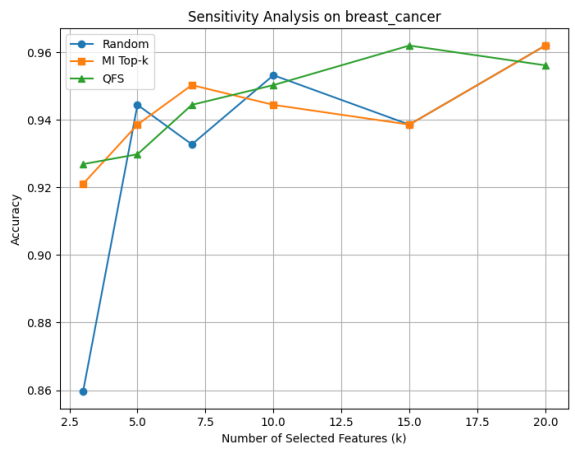
Conclusion Despite slight underperformance compared to full-feature models in some datasets, QFS-SA effectively reduces feature count while maintaining predictive quality.

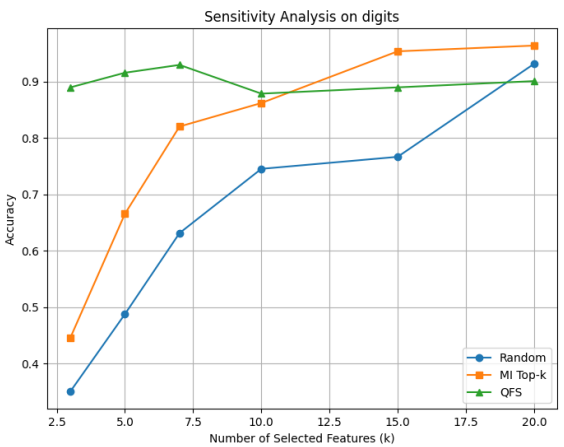
Feature Quality Analysis – Sensitivity Analysis

QFS shows comparable or superior performance to MI Top-k and Random, with better consistency under varying feature counts.



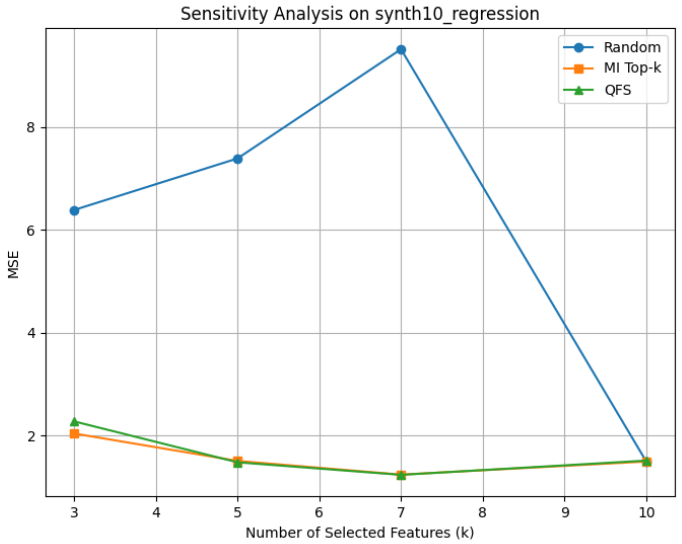


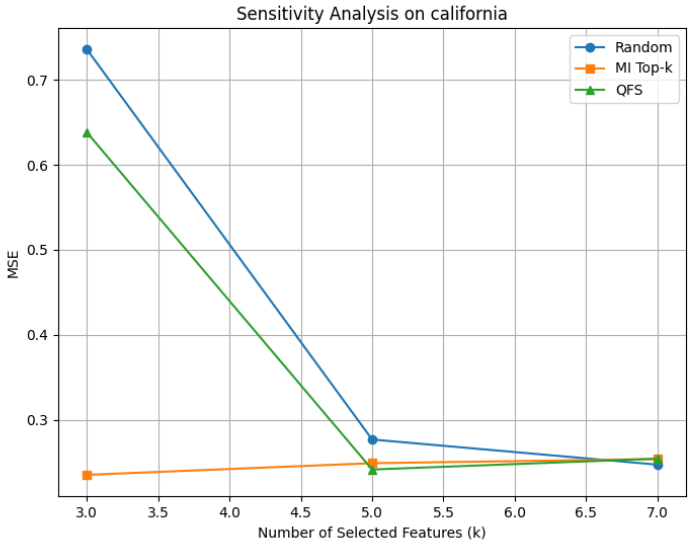


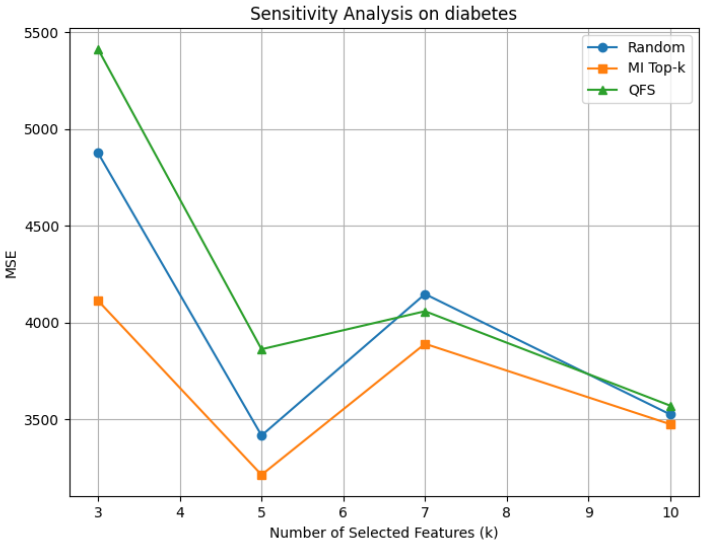


Conclusion QFS-SA consistently performs competitively with MI Top-k and Random selection in classification tasks, maintaining high accuracy across different feature counts.

Feature Quality Analysis – Sensitivity Analysis



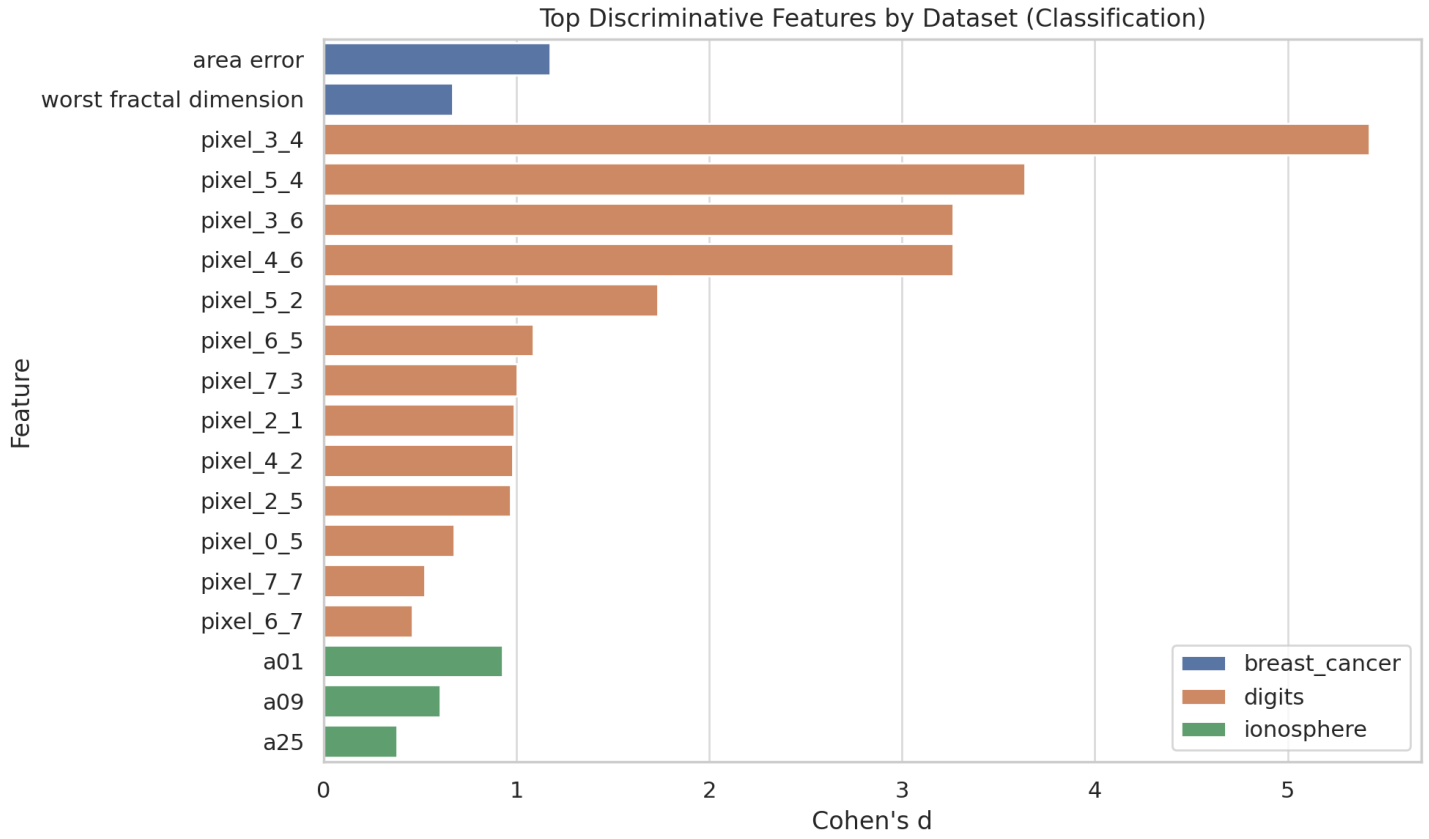




Conclusion While full-feature models may achieve lower error in some regression datasets, QFS-SA offers a strong trade-off between simplicity and performance, often outperforming random selection.

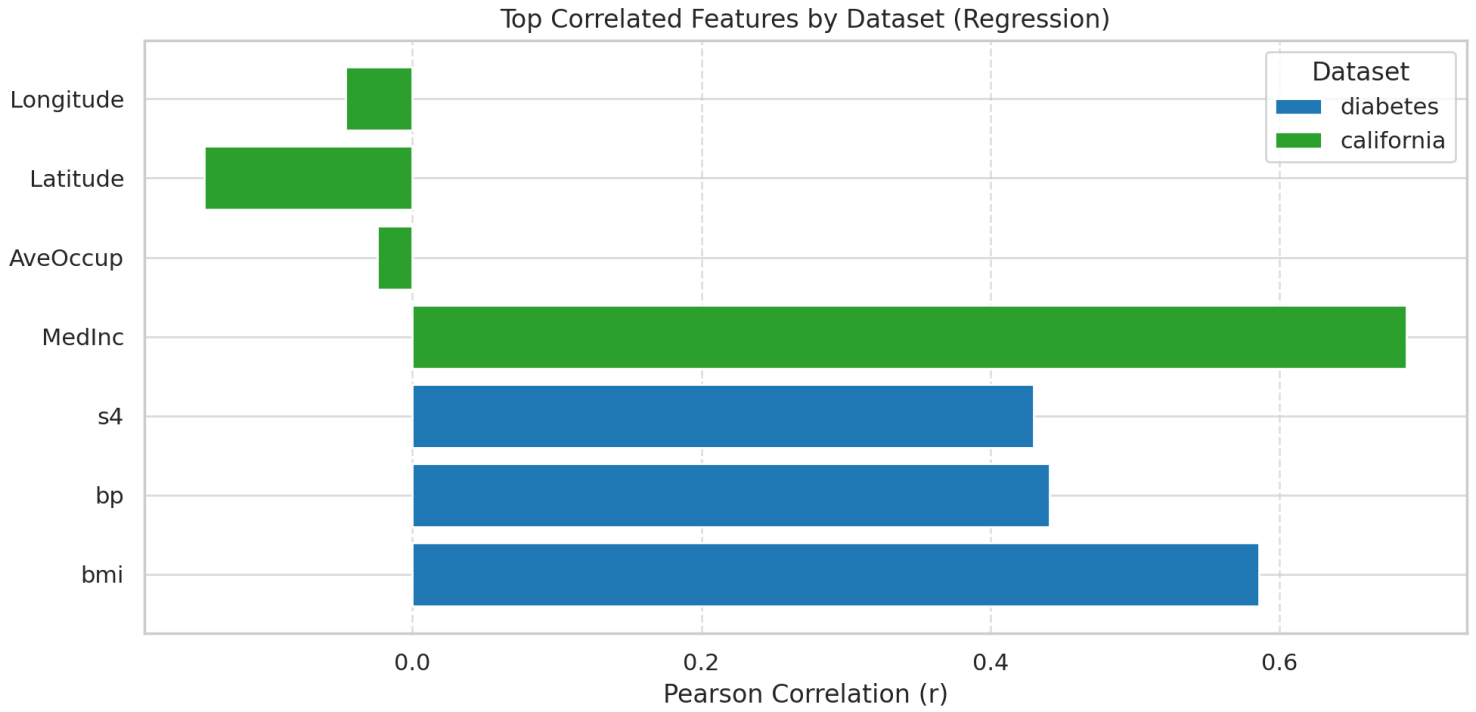
Feature Quality Analysis – Importance Analysis

- Features selected by QFS show **large effect sizes**, especially in the digits dataset.
- Several features achieve **Cohen's d > 2.0**, indicating **strong discriminative power** between classes.
- In breast_cancer, the most informative feature (area error) achieves **Cohen's d > 1.0**, suggesting a **large group difference**.
- For ionosphere, although effect sizes are lower, selected features still pass significance testing ($p < 0.05$).
- QFS tends to select features that are both **statistically significant and practically meaningful**.



Feature Quality Analysis – Importance Analysis

- **MedInc** (California Housing) exhibits the strongest positive correlation (r = 0.688) with housing prices.
- In the diabetes dataset, **BMI**, **blood pressure (bp)**, and **serum test (s4)** are the most predictive features.
- All top features are statistically significant ($p < 0.01$), indicating strong associations with the target.
- Negative correlations (e.g., **Latitude**, **Longitude**) suggest inverse relationships with housing prices.
- QFS successfully identifies interpretable and relevant features with domain relevance (e.g., **BMI** and disease progression).



VQE Implementation Results

Output Summary

- Estimated Minimum Energy: −0.1096
- Most probable bitstring: 1010111000
- Selected features (indices): [3, 4, 5, 7, 9]
- Random Forest accuracy: 0.530

Interpretation Points

- QFS-VQE was able to select a sparse feature set of 5 out of 10 features.
- The selected features yielded only **slightly better** than random performance on this simulated task (Accuracy = 0.53).
- This suggests either:
  - the selected bitstring was suboptimal
  - VQE struggled with convergence under current ansatz/depth.

Discussion

Key Highlights

1

Strong Predictive Power

- Consistently matches or outperforms MI Top-k using only k = 4 features
- Achieving up to **96% accuracy** on classification tasks.

2

Robustness

- Sensitivity analysis shows **stable performance** even when varying the number of selected features, indicating reliability.

3

Interpretability

- Selected features show **large effect sizes** (e.g., Cohen’ s d > 2) and statistical significance ($p < 0.01$), enhancing model transparency.

4

Quantum Realizability

- The implementation successfully recovered a sparse bitstring solution and selected features via quantum optimization

Limitations

Quantum Performance Bottleneck

- The current VQE implementation produced only marginal improvement over random feature selection
- Indicating convergence or expressibility issues under shallow ansatz.

Simulated Backend Only

- All quantum evaluations were conducted using simulators. Hardware noise, decoherence, and limited qubit counts were not considered.

Dataset Scale

- Experiments were conducted on small to medium datasets (≤ 64 features). The scalability of QFS on high-dimensional data remains untested.

Future Work VQE Refinement、Hardware Experiments、Hybrid Methods、Compare with Other Quantum Algorithms

Q & A