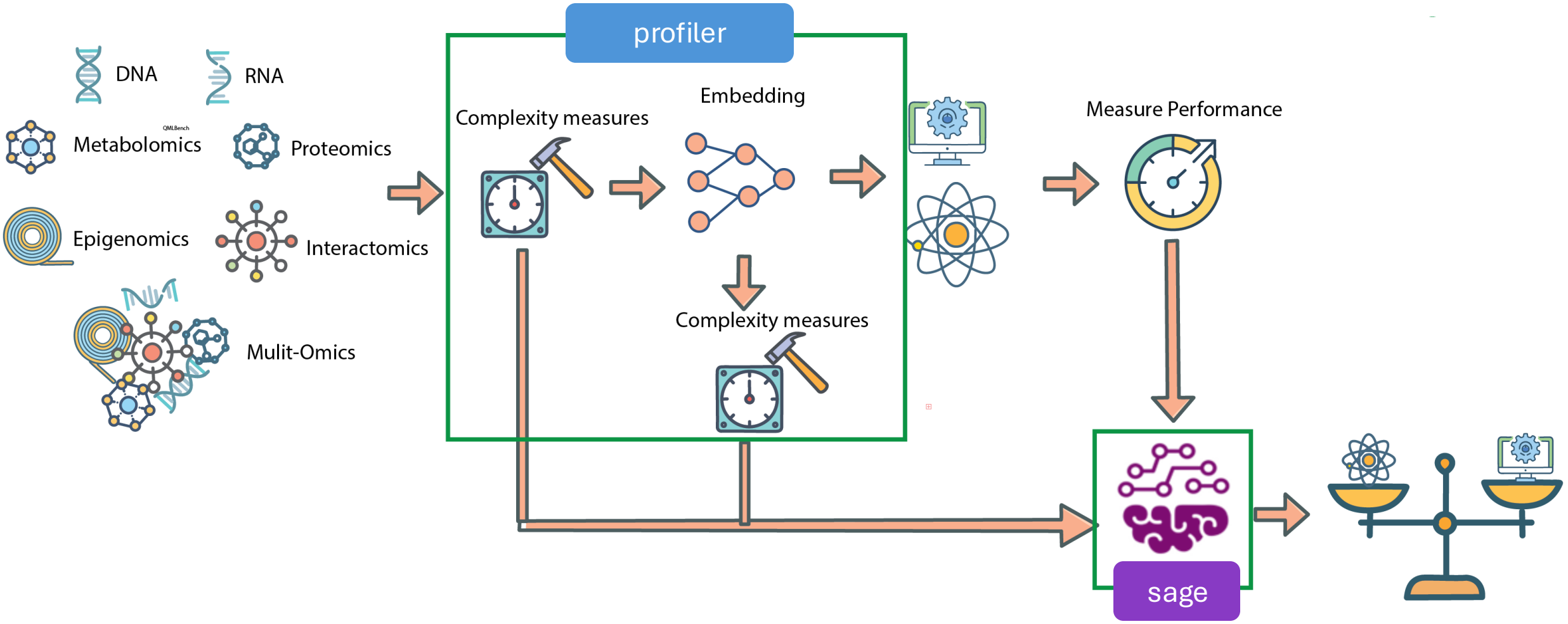
QBioCode: A Quantum-Classical Machine Learning Benchmarking tool for multi-omics data

QBioCode


DNA
RNA
Metabolomics
Proteomics
Epigenomics
Interactomics
Mulit-Omics
profiler
Complexity measures
Embedding
Complexity measures
Measure Performance
sage

QBioCode: Data Complexities

Dimensional

- Intrinsic Dimension (Rank)
- Manifold (Fractal Dimension)
- Volume
- Effective rank
- Eigenspectra

Distributional

- Kurtosis & Skewness
- Mutual Information
- Sparsity
- Condition Number

Geometric

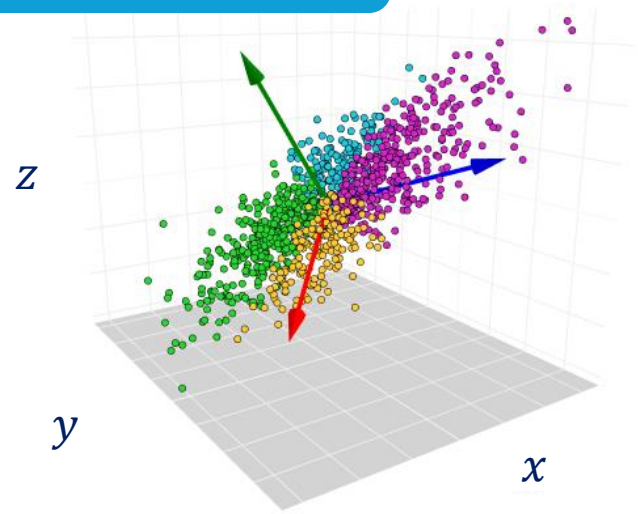
- Manifolds
- Clusters
- Density
- Topological Data Analysis
- Graph-based measures

Sampling

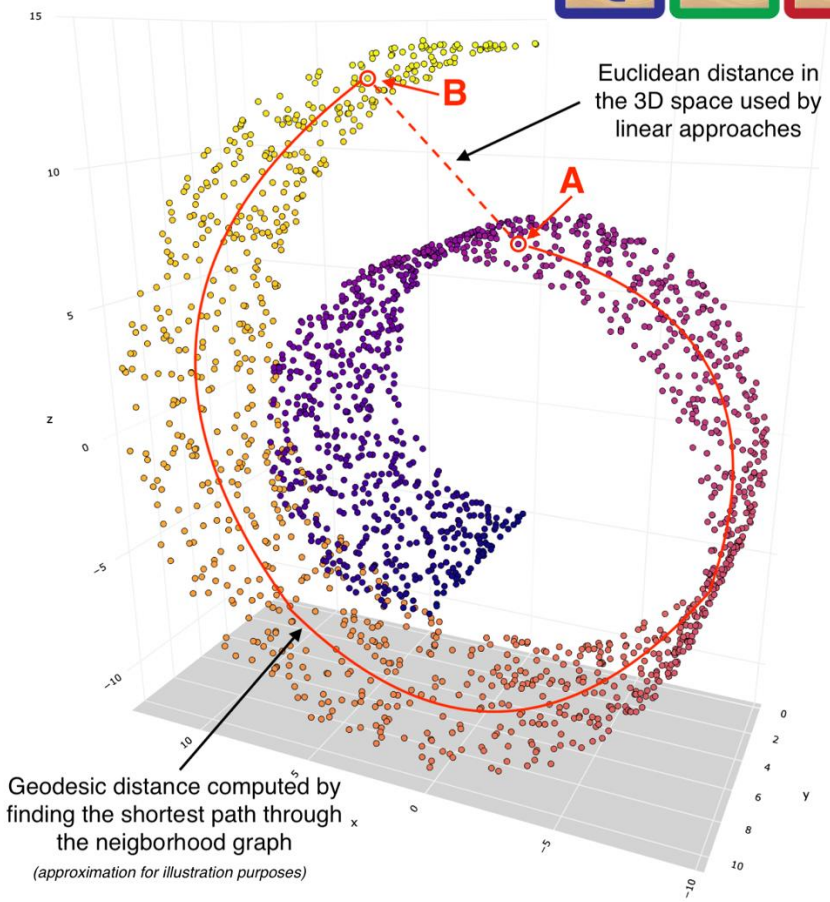
- Class imbalance ratio
- Class overlap measures
- Entropy
- Margin of separation between classes
- Sampling density variation

QBioCode : Embeddings

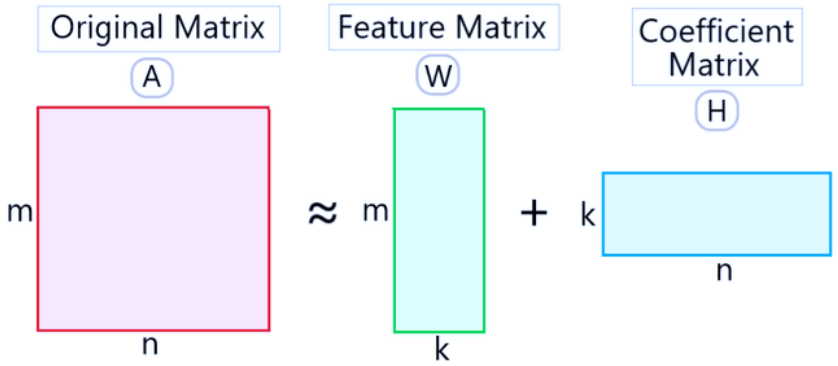
Principal Component Analysis (PCA)



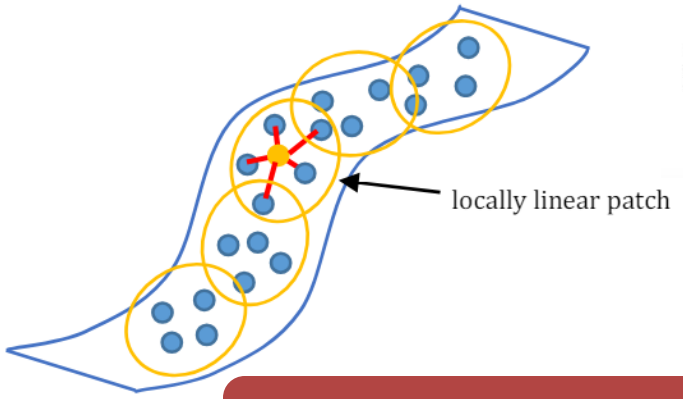
Spectral



Isomap

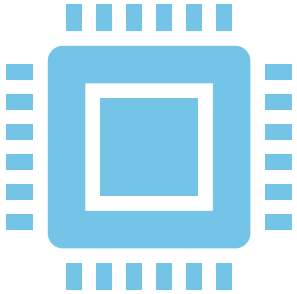


Non-negative Matrix Factorization (NMF)

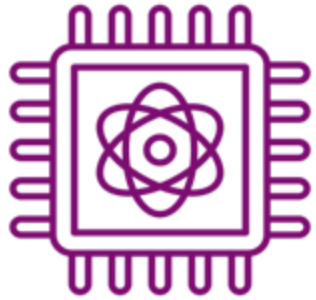


Locally Linear Embedding (LLE)

QBioCode: Models



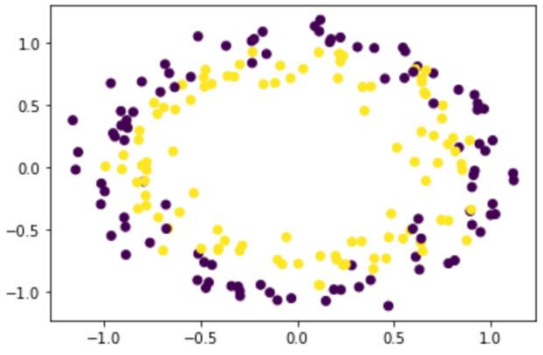
- Logistic Regression
- Support Vector Classifiers
- Naïve Bayes
- Random Forest
- XGBoost
- Multi-layer Perceptron



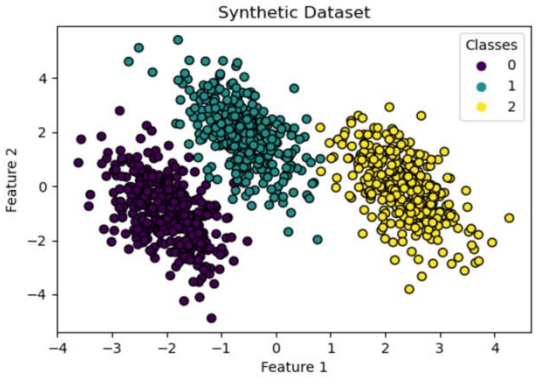
- Quantum Kernel Estimation
- Projected Quantum Kernel
- Quantum Support Vector Classifiers
- Variational Quantum Classifier / Quantum Neural Networks

Artificial Data Generation

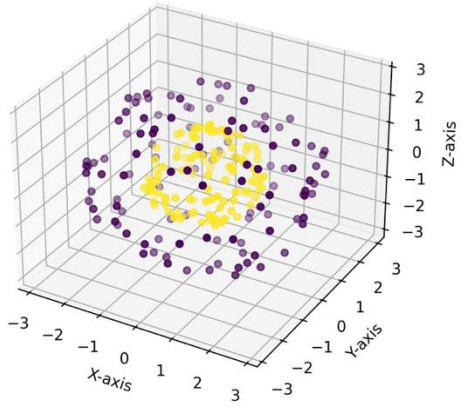
- To diversify datasets, we developed functions to generate artificial *non-linear* data based on user-defined combinations of data characteristic.
- These modules generate blobs, moons, circles, spheres, spirals, etc.



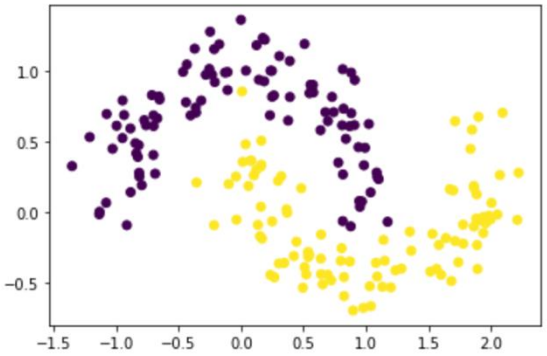
Circles



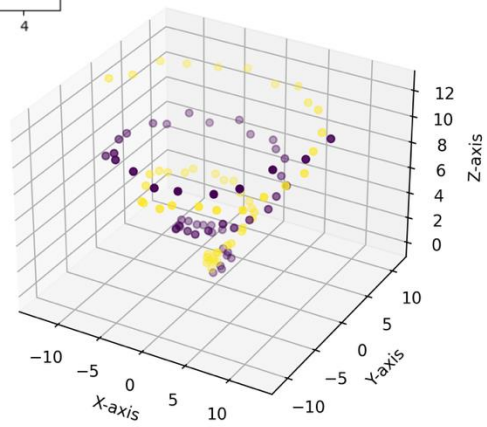
Blobs



Spheres



Half Moons

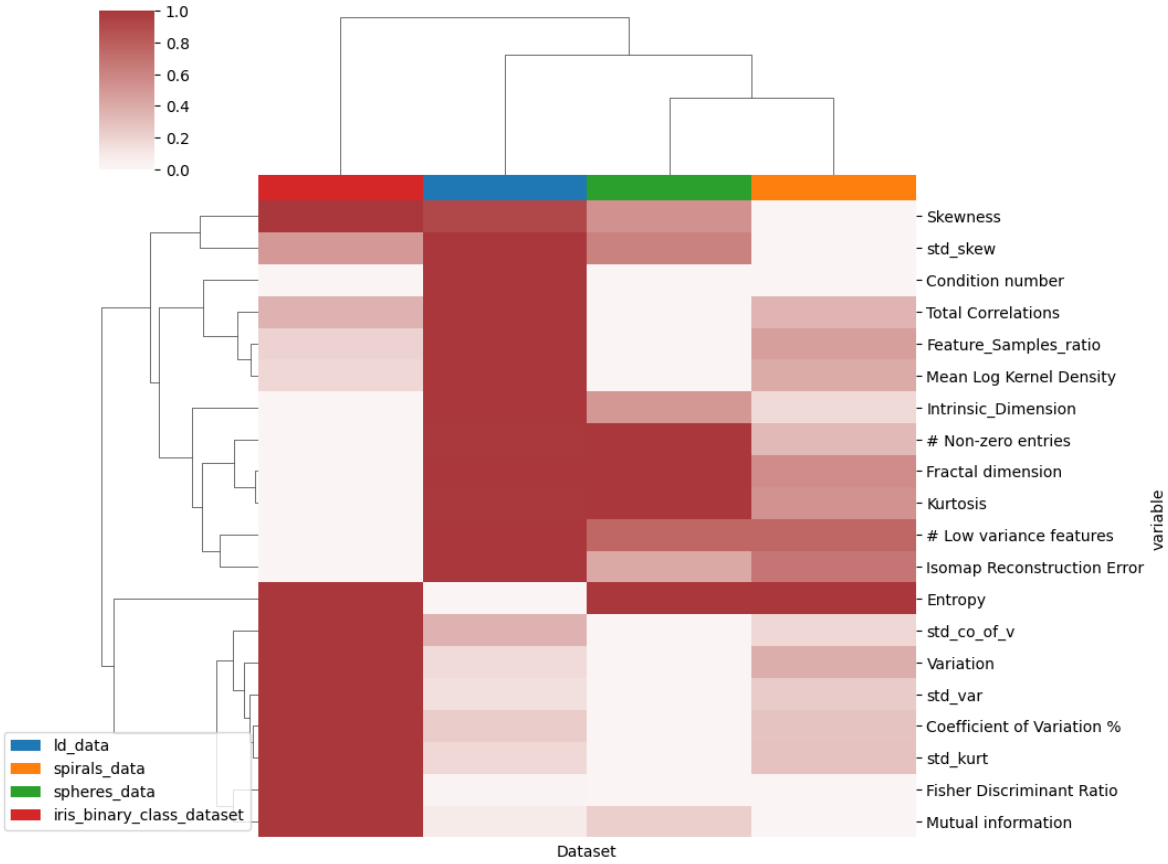


Spirals

Understanding the analyses

Hierarchical clustering heat maps

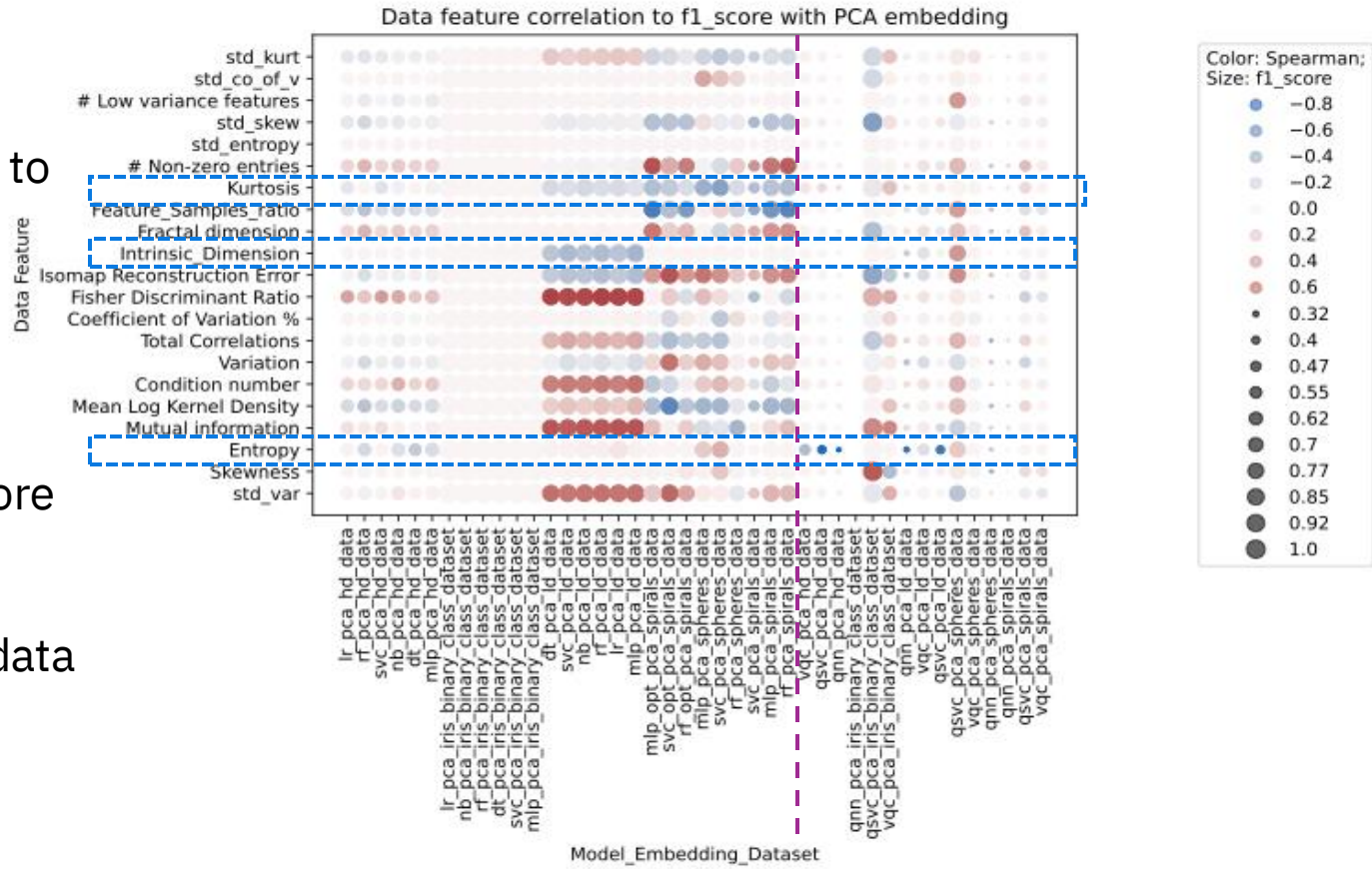
- What it's doing here:
  - Complexity measure range is normalized between 0 and 1.
  - Euclidean distance is calculated between columns and rows, clustering together those with the shortest distance → similar intensities for complexity measures.
  - The dendrogram branches create a pairing hierarchy.
  - Outlier has longest branch.



Understanding the analyses

Spearman Rank Correlations

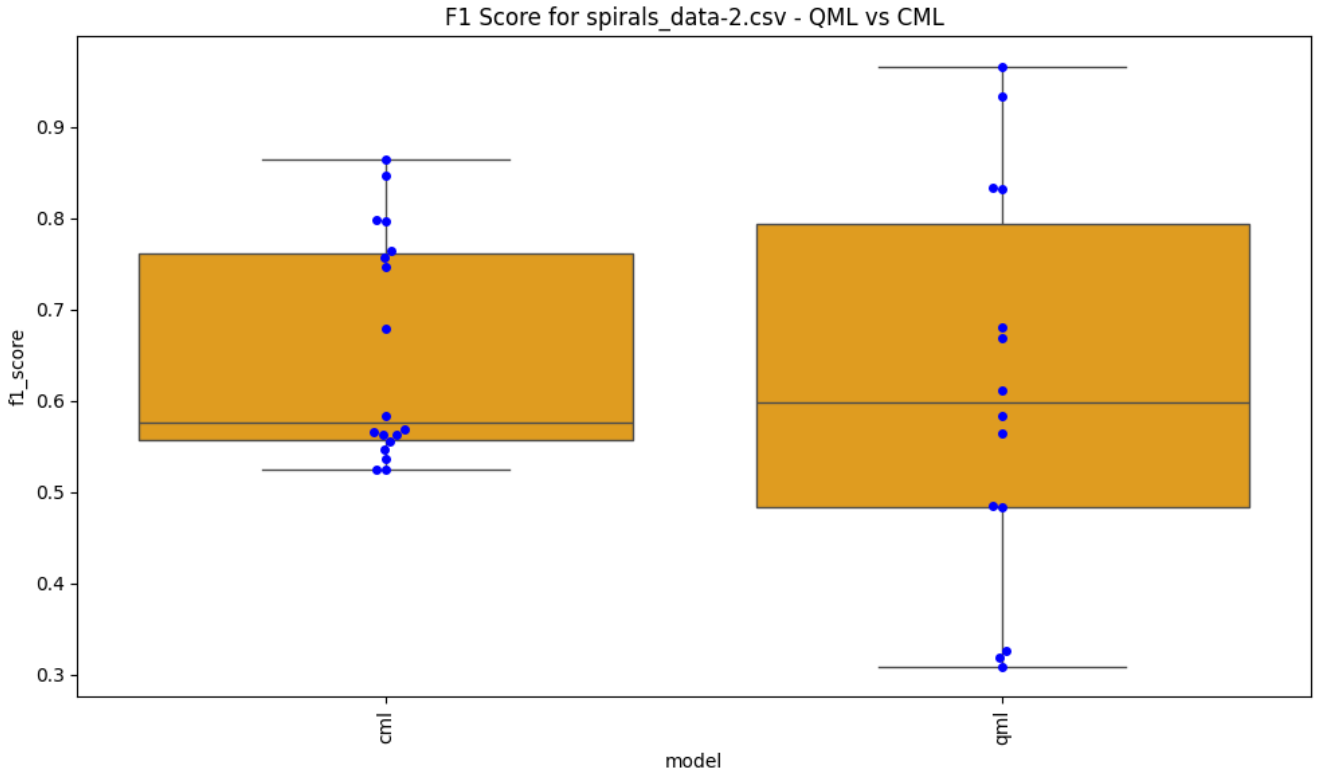
- What it's doing here:
  - Correlates data complexity measure to model performance (F1-score)
  - Red = positive correlated
  - Blue = anti-correlated
  - Size of sphere = magnitude of F1-score
  - The x-axis contains all classical and quantum ML models run on various data sets across linear and non-linear embeddings.
  - Helps answer:
    What complexity measures influence your model score the most?



Understanding the analyses

Box-and-whisker plots

- What it's doing here:
  - Plots distribution of median F1-scores per datasets, across all splits of data, per model
  - Top and bottom of box = upper and lower quartiles (Q3 and Q1)
  - Whiskers denote range in F1- scores
  - Helps answer:
    What is the locality, spread, and skewness groups in my data (F1-scores) based on their quartiles?



Understanding QSage

So, what is the big picture?

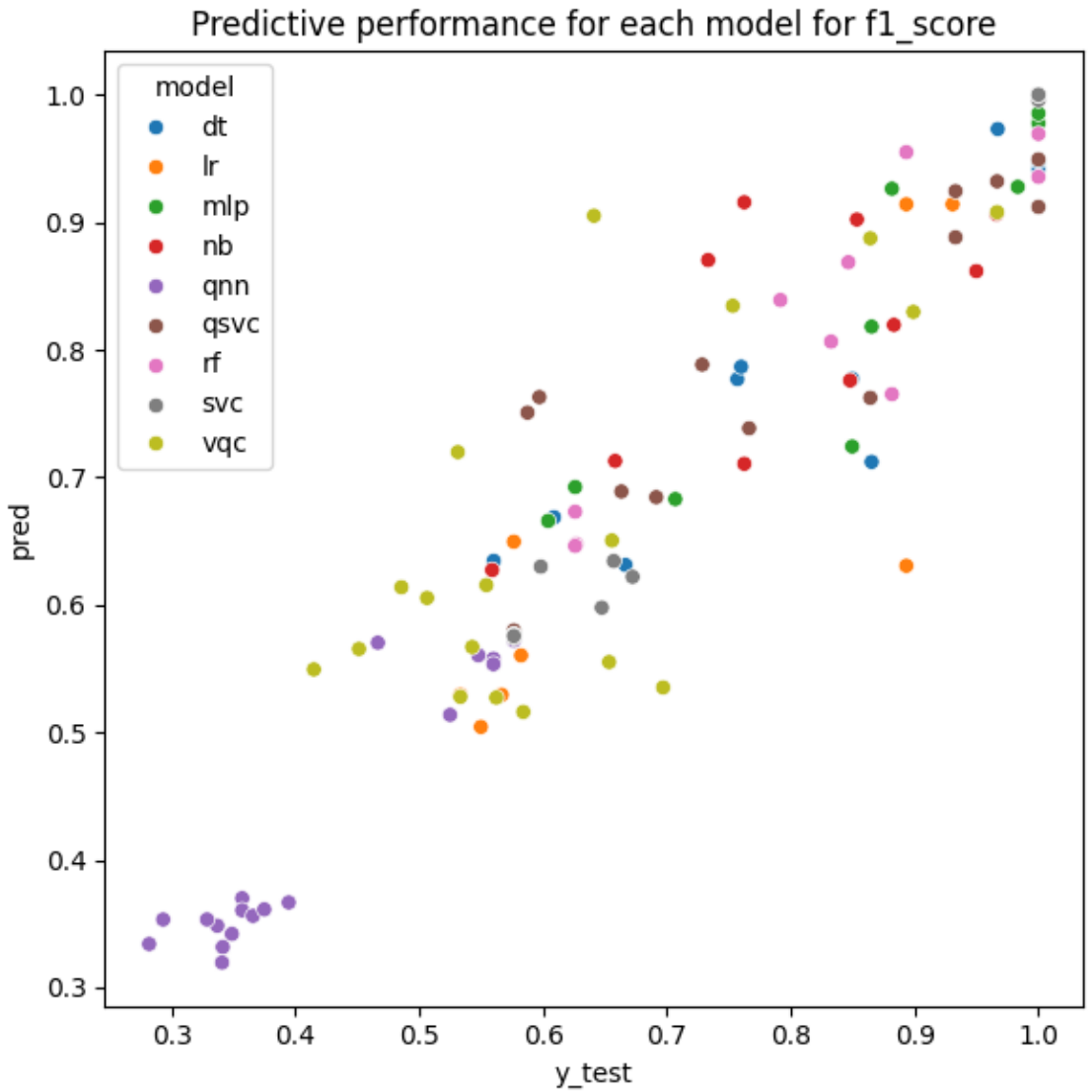
What do we do with all this stuff?

- *What if I were to tell you, what ML method to use, just by looking at your data?*

- We trained a new model on all of these correlations --> QSage

- QSage looks at the complexity metrics from your data and predicts the F1 score and outputs the model best suited for your data!

Predicts F1, AUC, and accuracy beforehand --> no need to run all model!

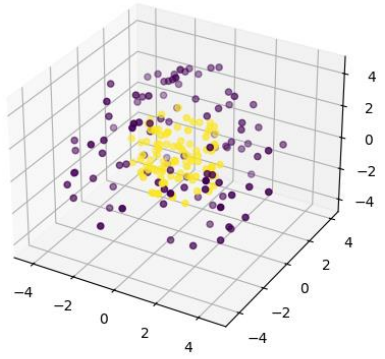


Examples

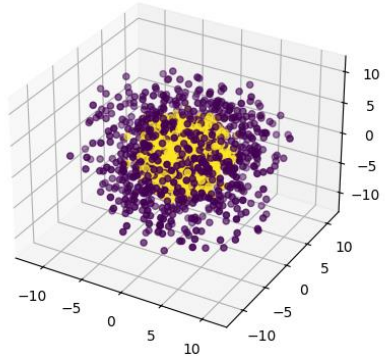
Examples: geometric shapes

- Let's look at higher dimensional artificial, geometric data sets (3D and beyond).
- Task – generate QML and CML models for these and compare performance.
- This data is periodic – can QML do well with these?

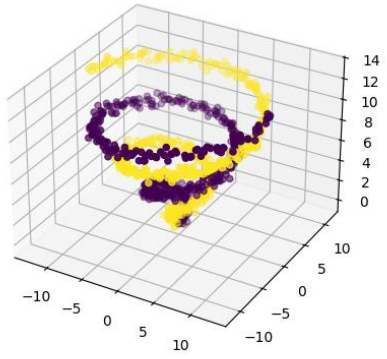
5 qubits/features



10 qubits/features

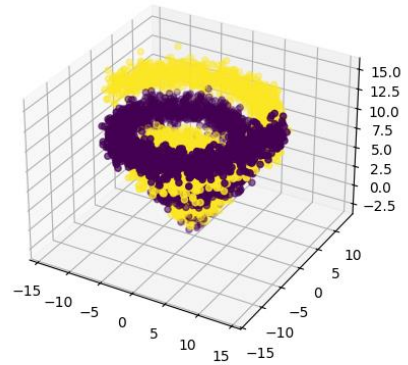


Low dimension and # of samples



3 qubits/features

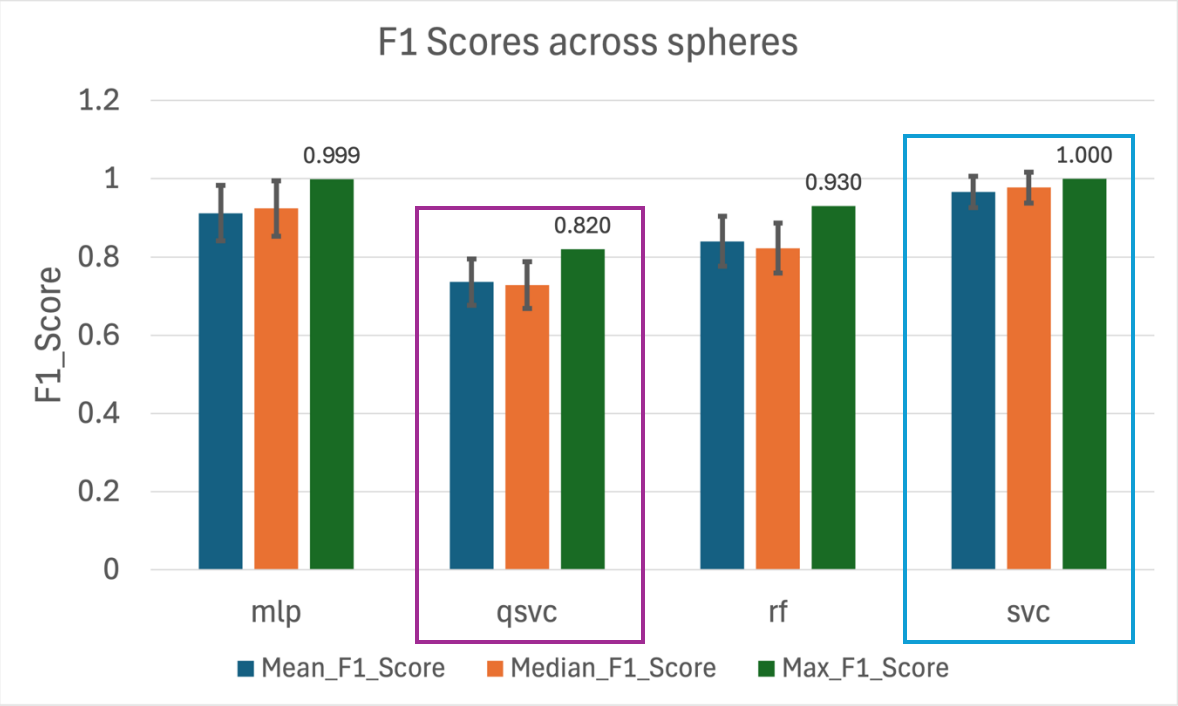
High dimension and # of samples

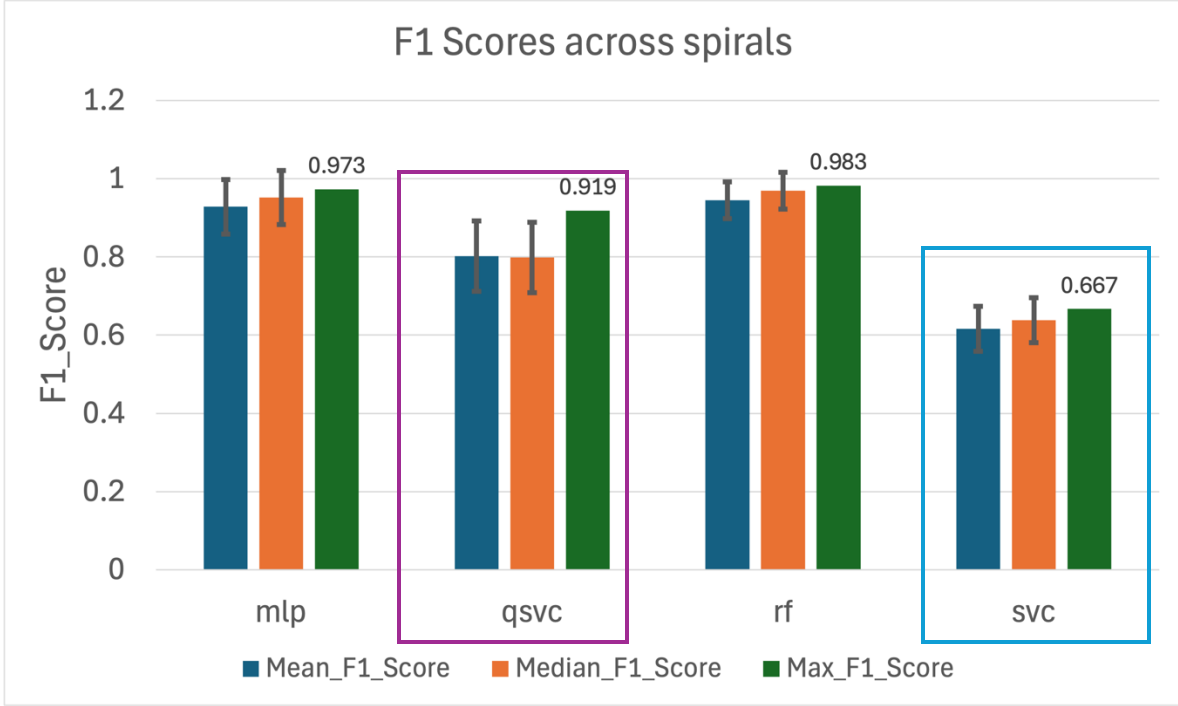


12 qubits/features

Examples: spheres and spirals

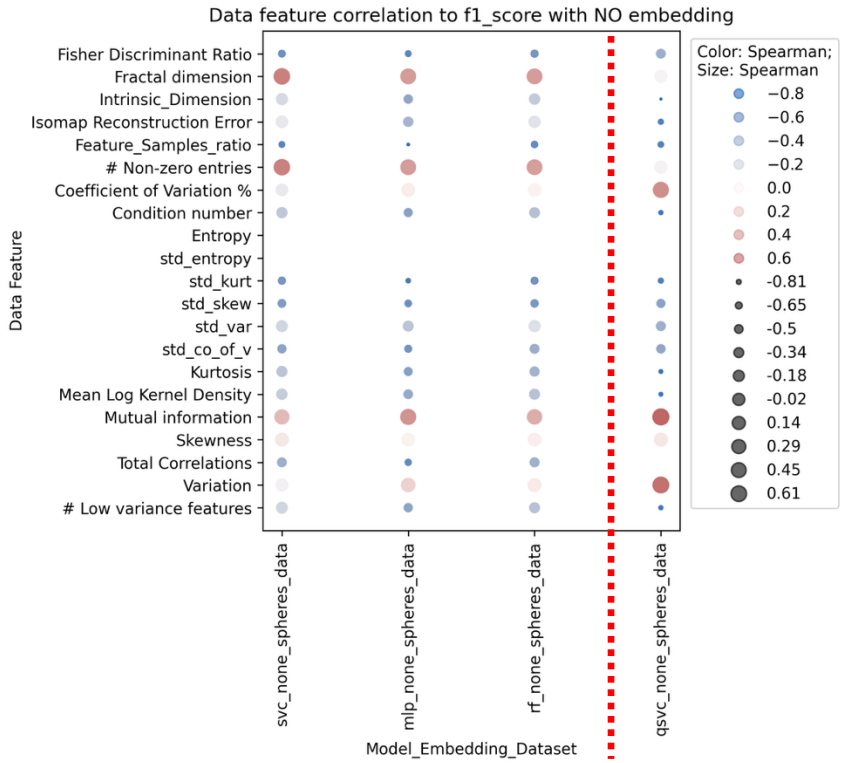
- Spirals seem QML friendly.
- SVC>QSVC with spheres, but it flips to QSVC>SVC with spirals
- RF improves with spheres, MLP is consistent across both



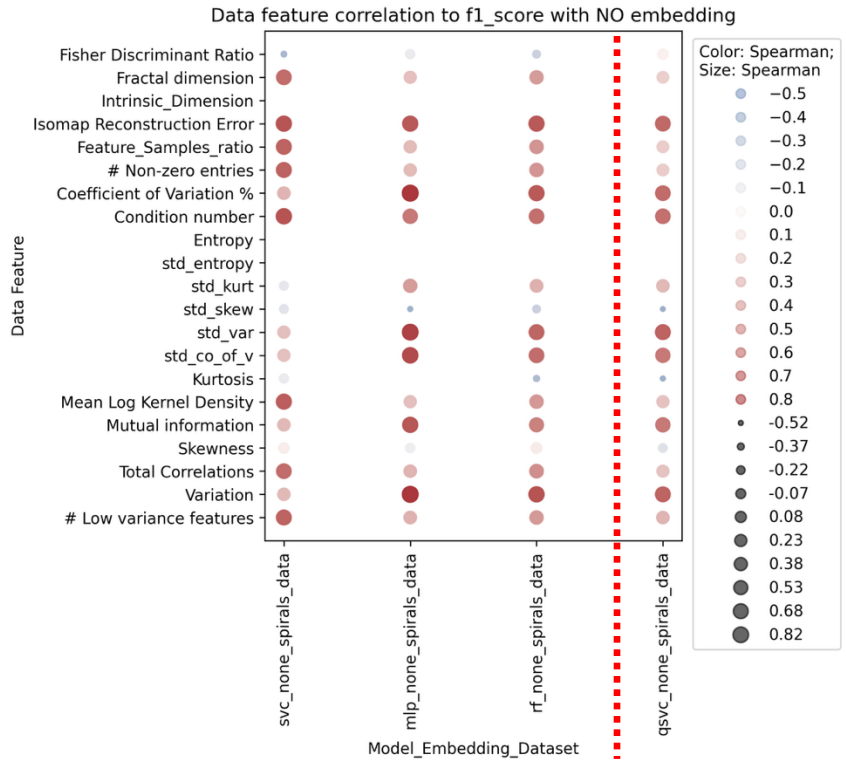


Examples: spheres and spirals

- Spheres: clear switch with Intrinsic dimension, Coeff of variance, and total correlations
- Spirals: correlation type switches (red vs blue) for Fischer Discrimination Ratio (measures imbalance) between CML and QSVC



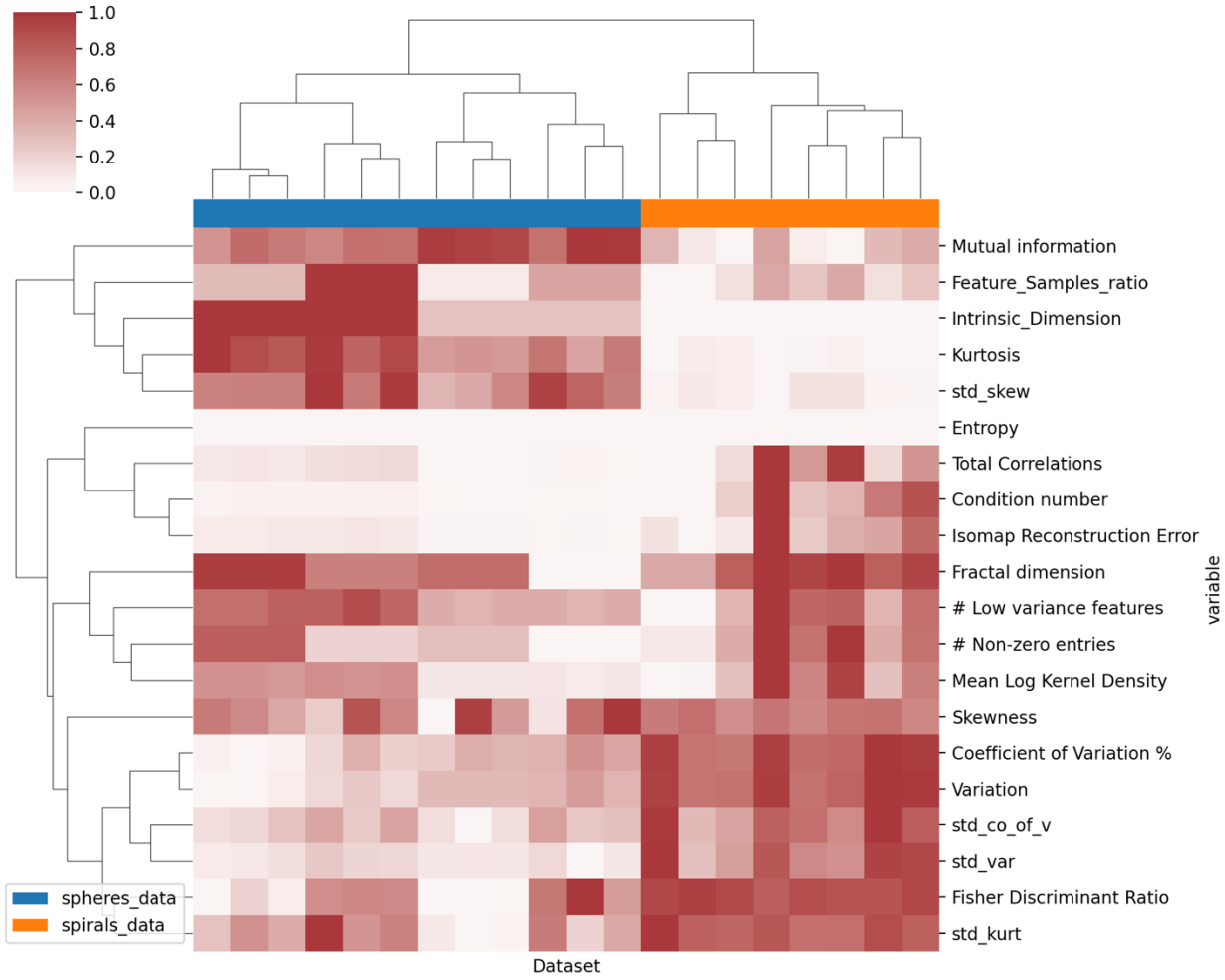
Spheres



Spirals

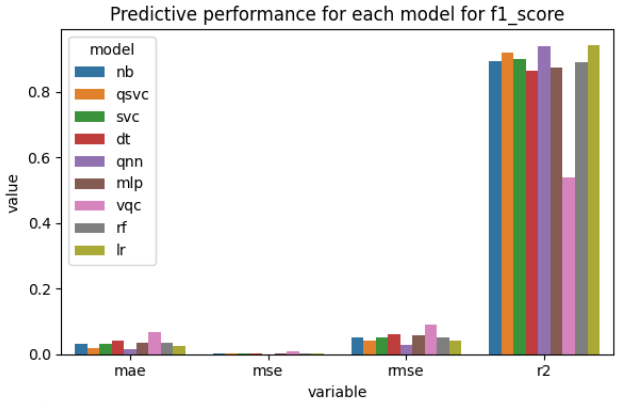
Examples: spheres and spirals

- Remember: on average QSVC>SVC with spirals.
- So, what is it about the spirals?
- There is a rather obvious disparity in a few areas



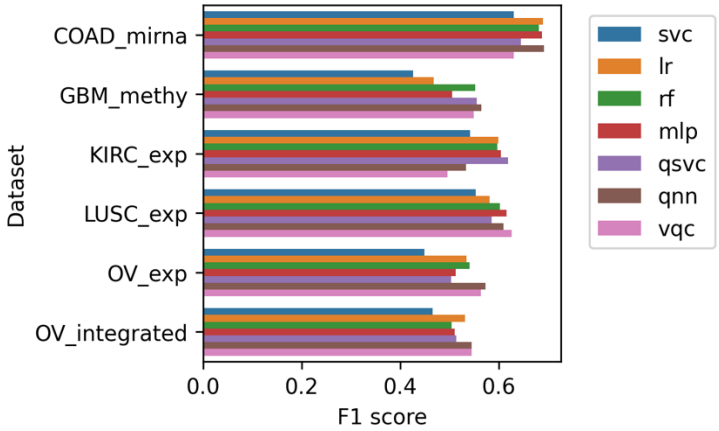
Examples: QSage

- Test datasets
  - Synthetic non-linear data
  - RNAseq from TCGA cancers
- For all data sets run n ML models:
  - Naïve Bayes (NB), Support Vector Classifier (SVC), Decision Trees (DT), Multi-layer Perceptron (MLP), Random Forest (RF), Logistic Regression (LR)
  - Quantum Support Vector Classifier (QSVC), Quantum Neural Network (QNN), and Variational Quantum Classifier (VQC)
- Embeddings used: None, PCA, NMF, IsoMap, LLE, Spectral
- Prediction metrics: F_1, Accuracy, and AUC
- Complexity measures for data and any embeddings generated:
  - # Features, # Samples, Feature_Samples_ratio, Intrinsic_Dimension, Condition number, Fisher Discriminant Ratio, Total Correlations, Mutual information, # Non-zero entries, # Low variance features, Variation, std_var, Coefficient of Variation %, std_co_of_v, Skewness, std_skew, Kurtosis, std_kurt, Mean Log Kernel Density, Isomap Reconstruction Error, Fractal dimension, Entropy, std_entropy
- QSage is a RF Regressor *predicting test labels* for each n ML model for each metric.

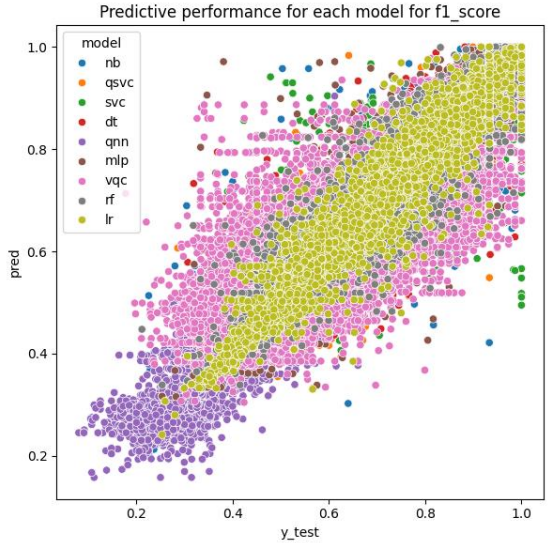


r^2 used as the metric to evaluate QSage performance

Training performance



Qprofiler was able to identify datasets where QML methods outperform CML methods*



* Results currently being hyperparameter optimized

Test performance on circles data

Models	Predicted F1	r^2 of prediction	F1 weighted by r^2	Actual F1	Rank by Predicted F1	Rank by Actual F1
qsvc	0.657	0.921	0.605	0.562	1.0	3.0
nb	0.649	0.893	0.579	0.518	2.0	5.0
mlp	0.571	0.875	0.499	0.582	3.0	1.0
rf	0.496	0.890	0.441	0.444	4.0	7.0
lr	0.453	0.943	0.427	0.580	7.0	2.0
svc	0.452	0.899	0.406	0.543	8.0	4.0
dt	0.456	0.865	0.395	0.403	6.0	8.0
qnn	0.333	0.941	0.314	0.333	9.0	9.0
vqc	0.464	0.538	0.250	0.494	5.0	6.0

QSVC predicted to have the best performance and in actuality was within 0.02% of the top performer

Let’s take it for a ride

So, let's try it out!

1) Go over the `config.yaml` file and learn how to change parameters

2) Activate your environment
   - If you haven't done so, set up your environment now following the instructions in the README.md

3) Run the main code. In your terminal, type:
   ```
   python qbiocode-profiler.py --config-name=config.yaml
   ```

4) Wait and watch the progress outputs being printed out.

5) We'll analyze the results and run your own data in the afternoon 😎.