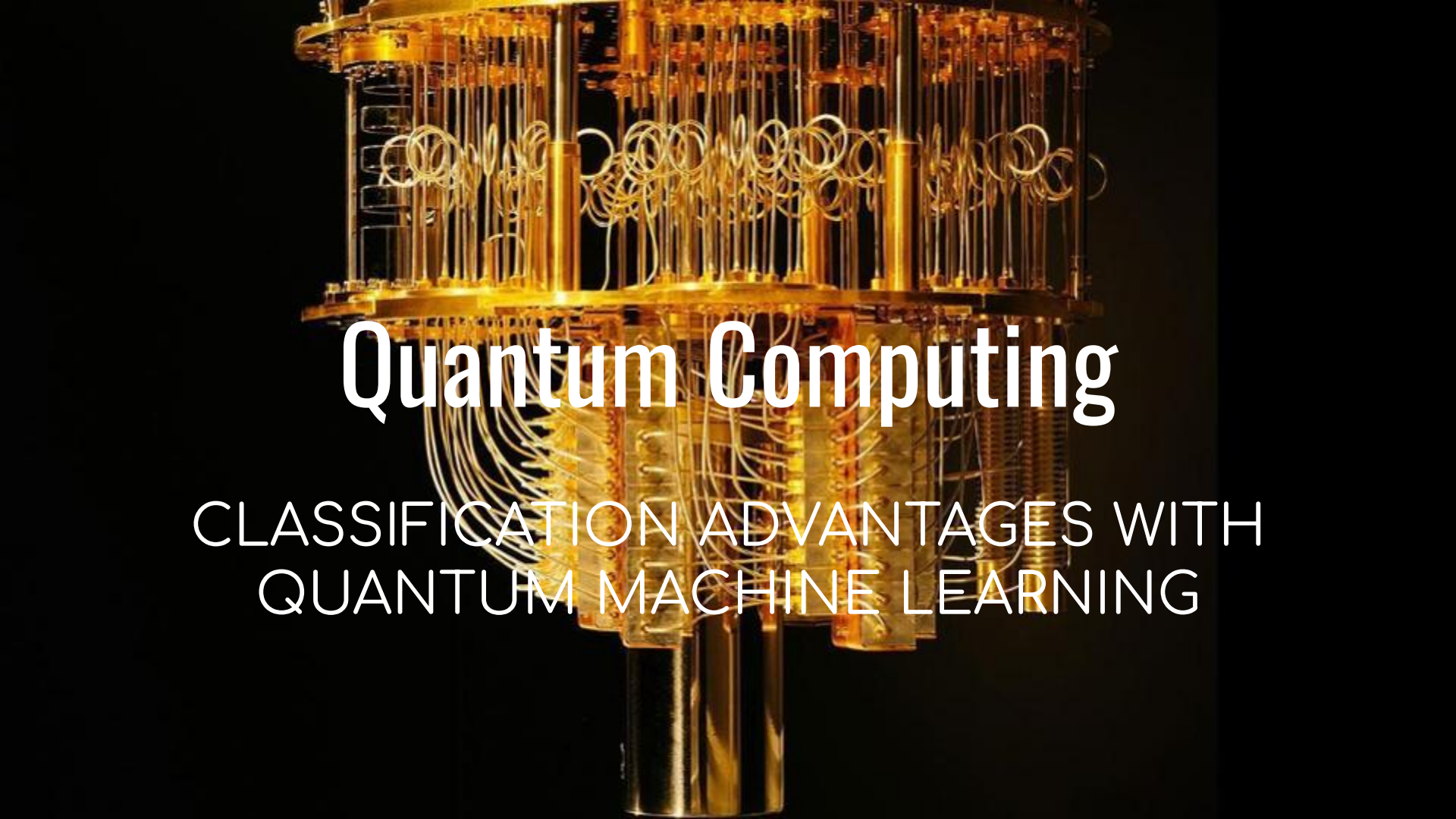
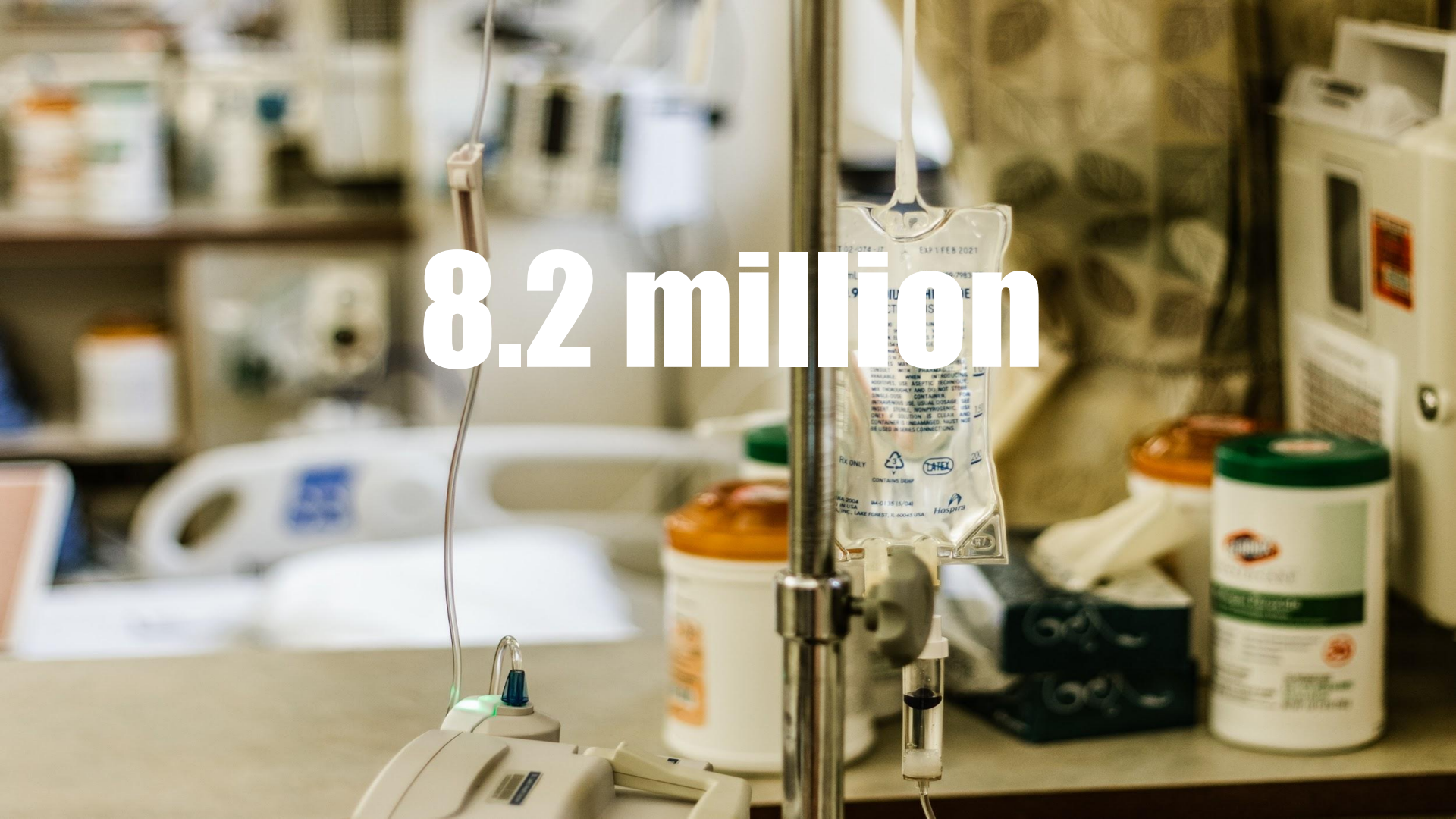
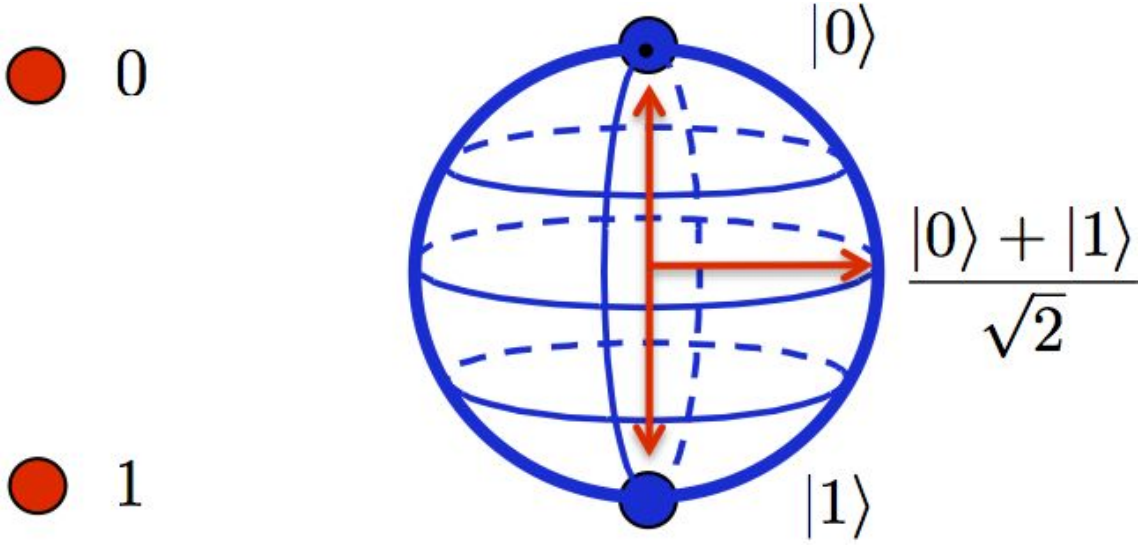


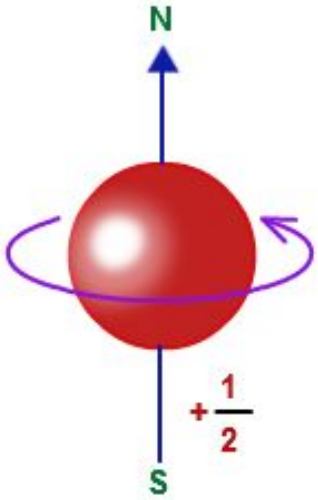
Quantum Computing

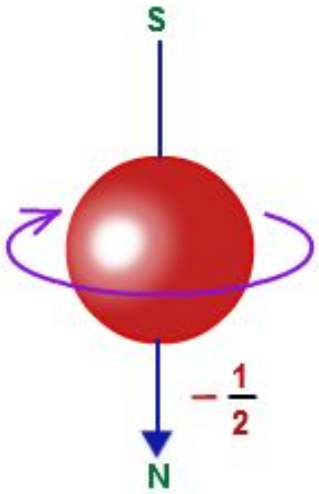
CLASSIFICATION ADVANTAGES WITH QUANTUM MACHINE LEARNING

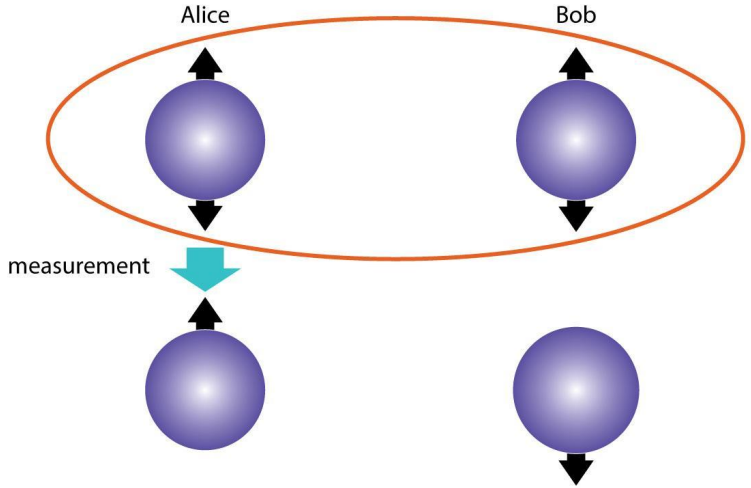

8.2 million

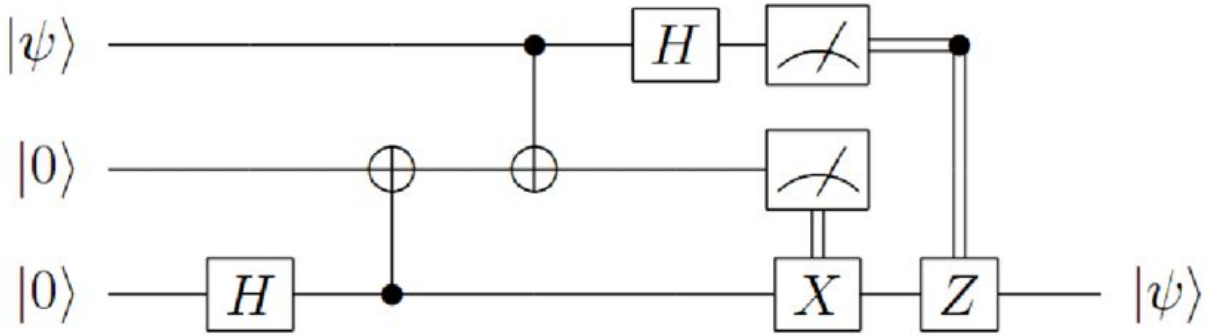


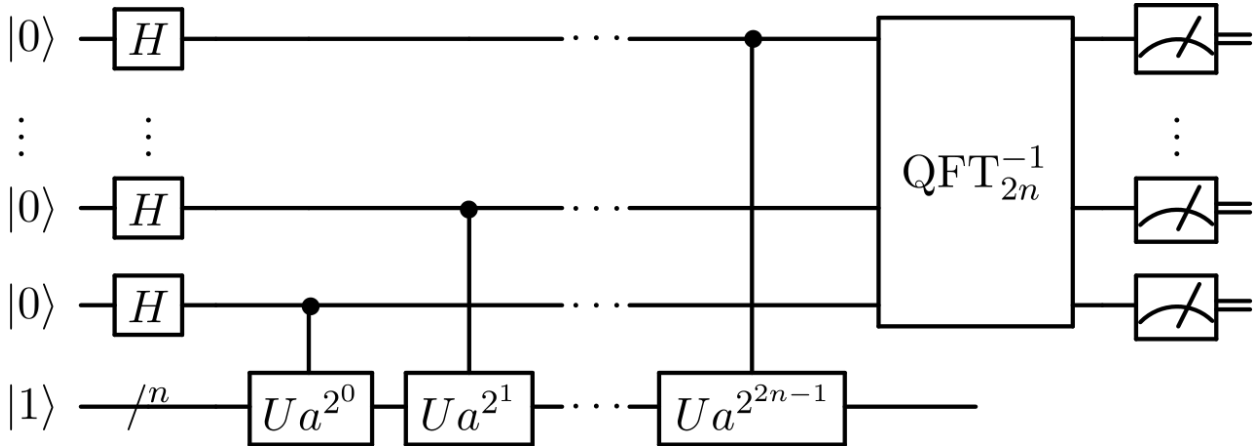
Classical Bit **Qubit**

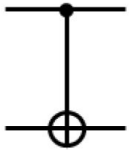

N
S
+ 1/2

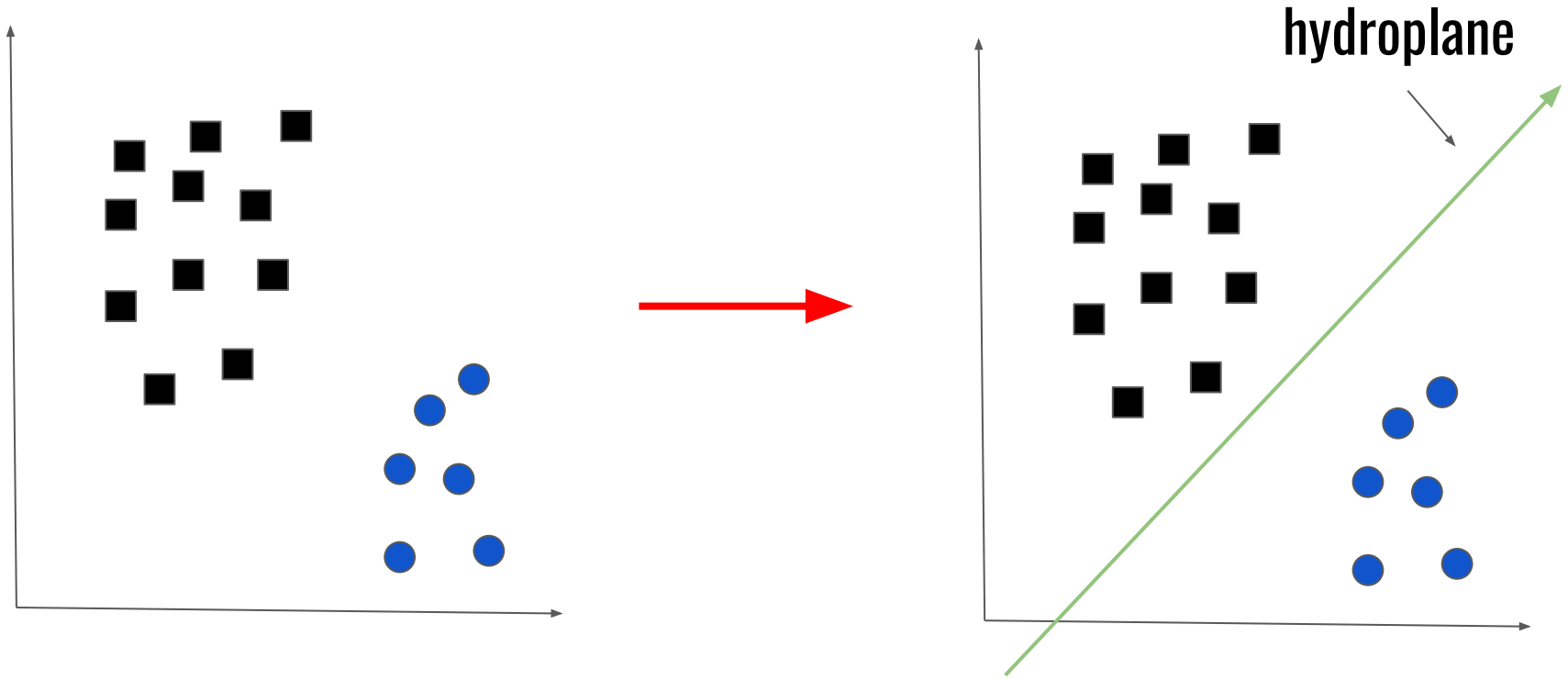

S
N
− 1/2

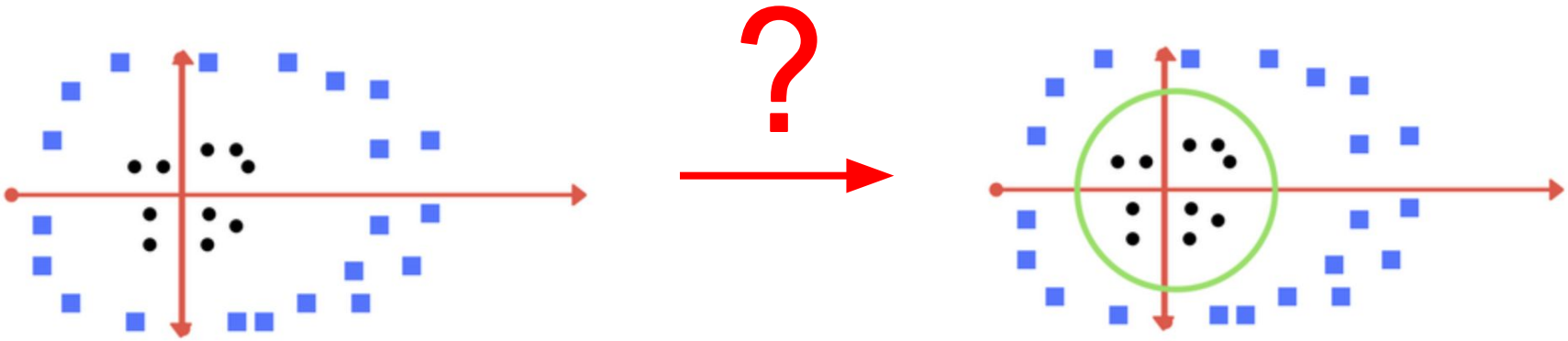

Alice
Bob
measurement

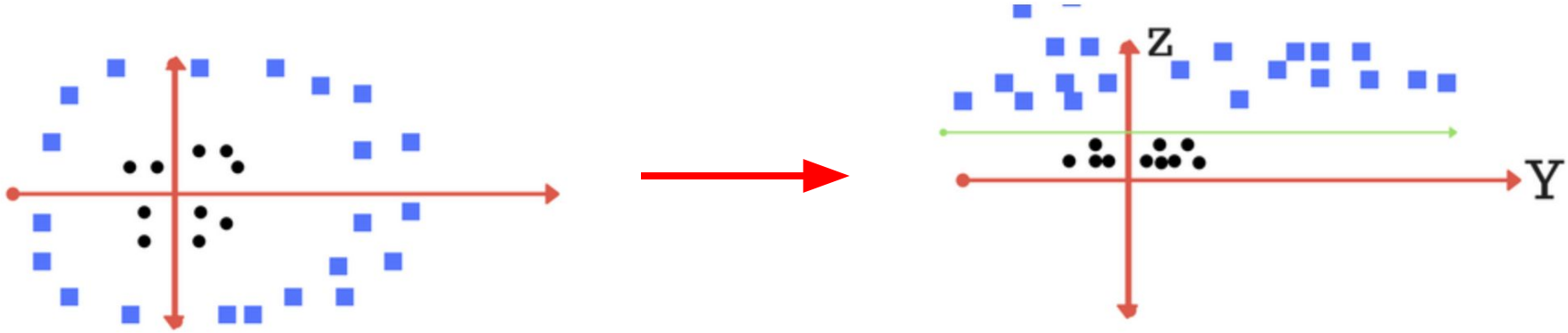

$|\psi\rangle$
$|0\rangle$
$|0\rangle$
H
H
X
Z
$|\psi\rangle$

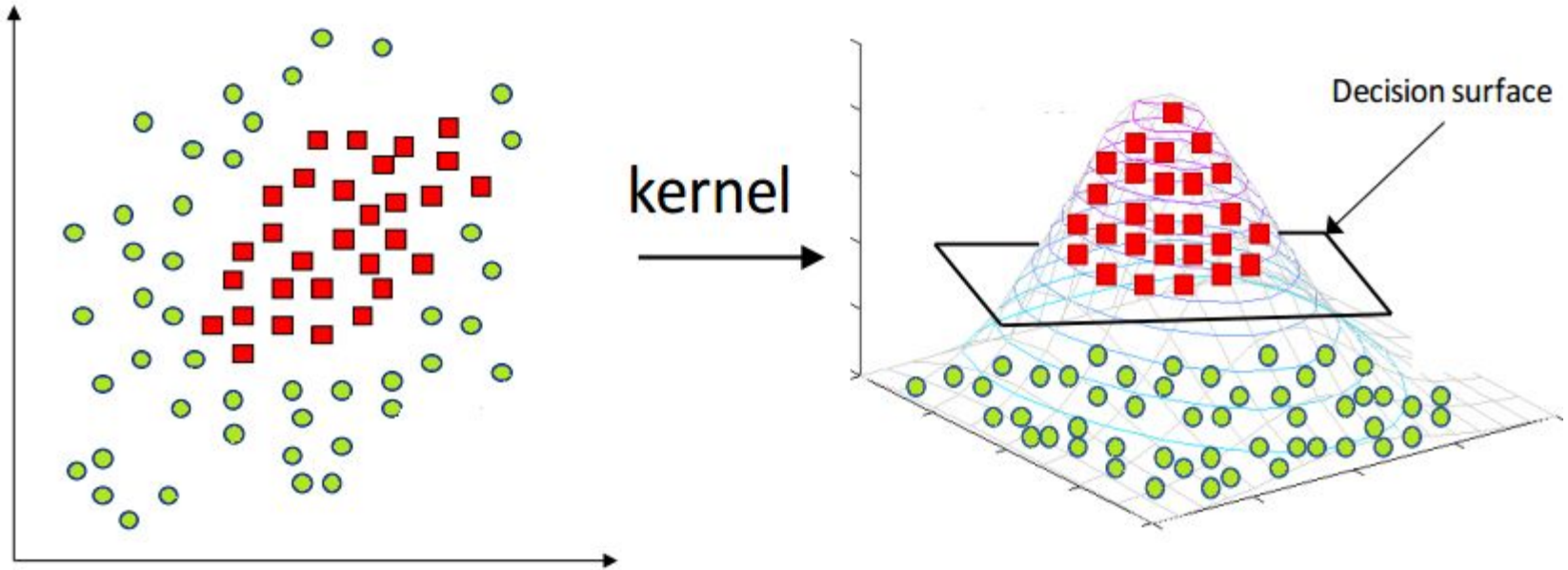

$|0\rangle$
$|0\rangle$
$|0\rangle$
$|1\rangle$
H
H
H
n
Ua^{2^0}
Ua^{2^1}
$Ua^{2^{2n-1}}$
QFT_{2n}^{-1}

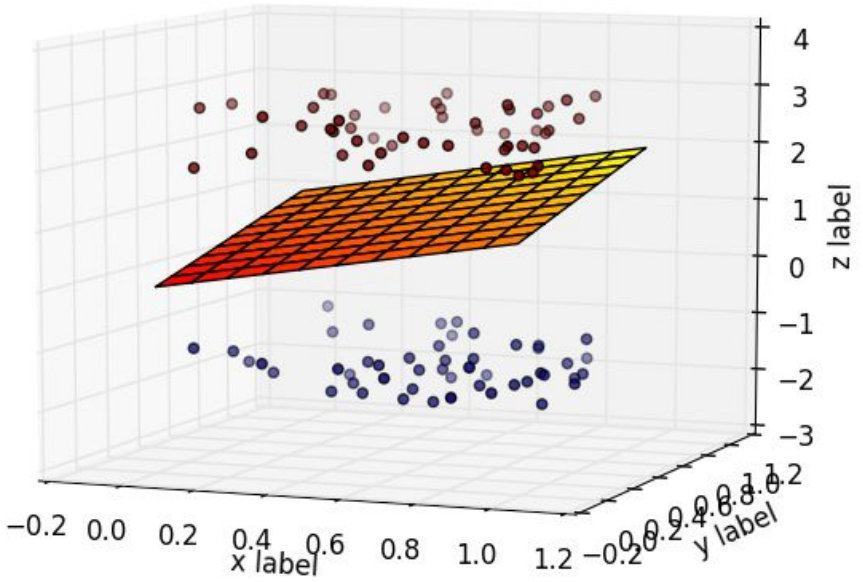
Gate	Notation	Matrix
NOT (Pauli-X)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
CNOT (Controlled NOT)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

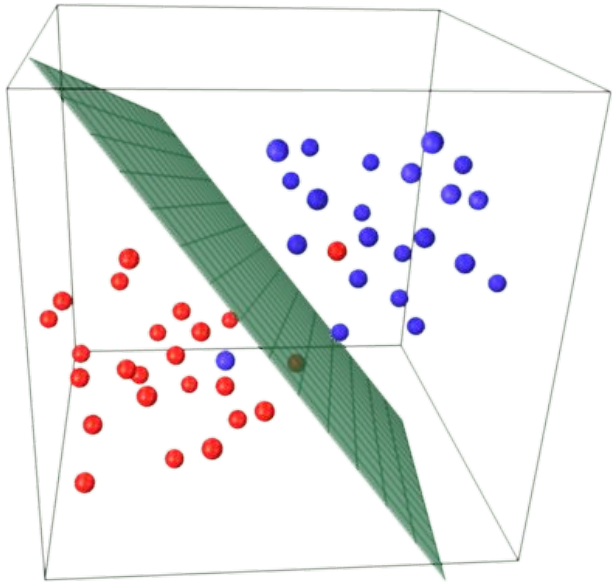

hydroplane



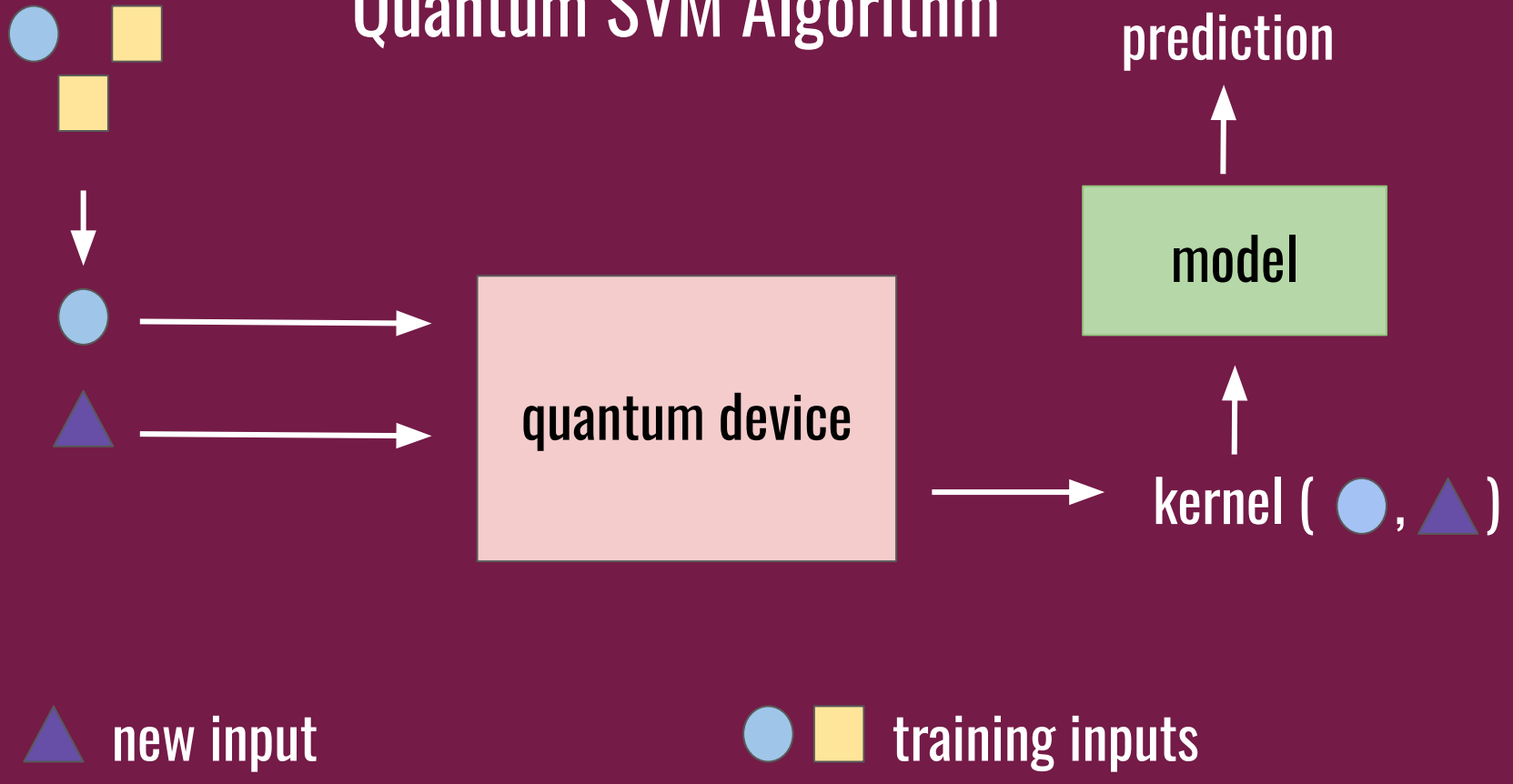

Z
Y

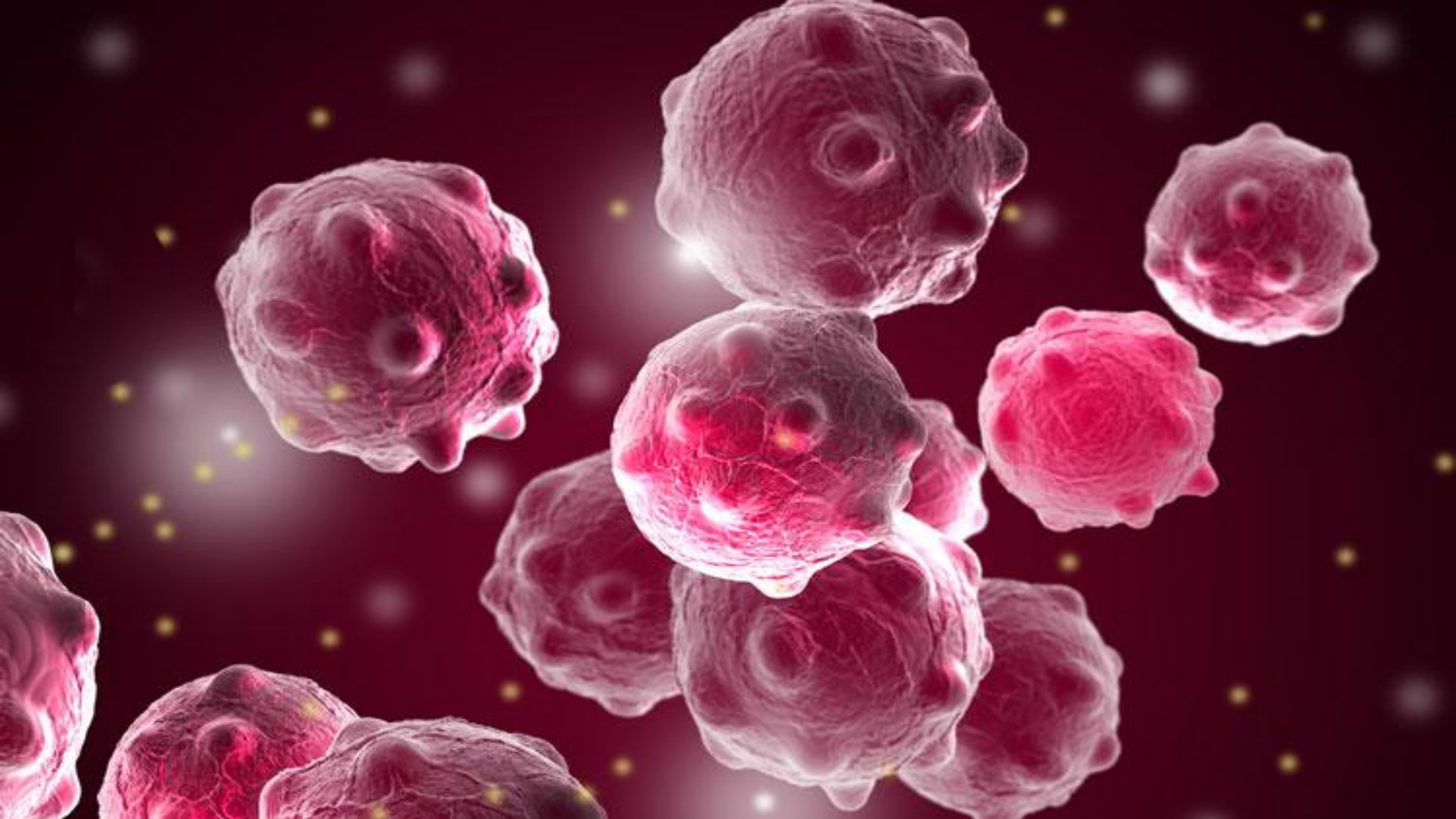

kernel
Decision surface

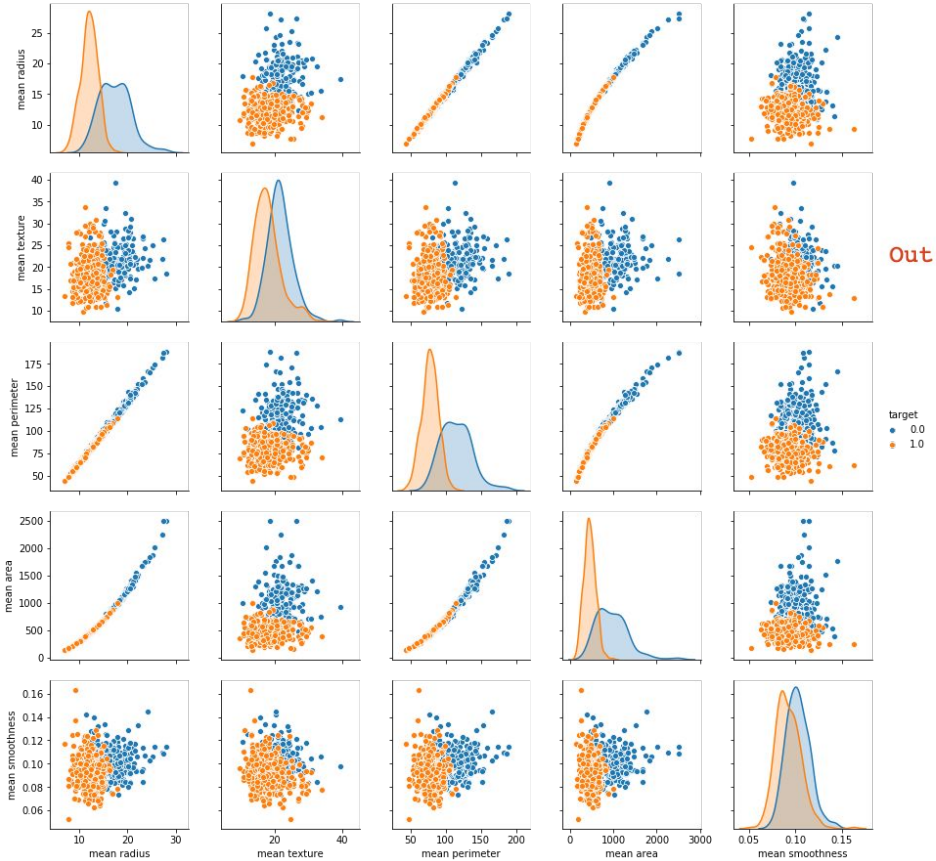

4
3
2
1
0
−1
−2
−3
z label
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
x label
y label



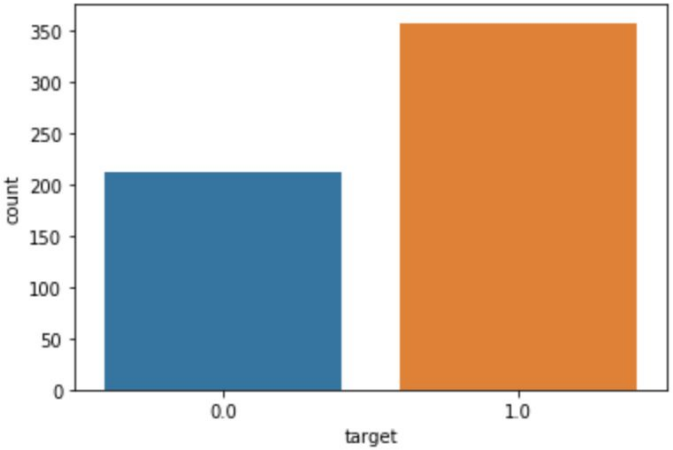
Quantum SVM Algorithm





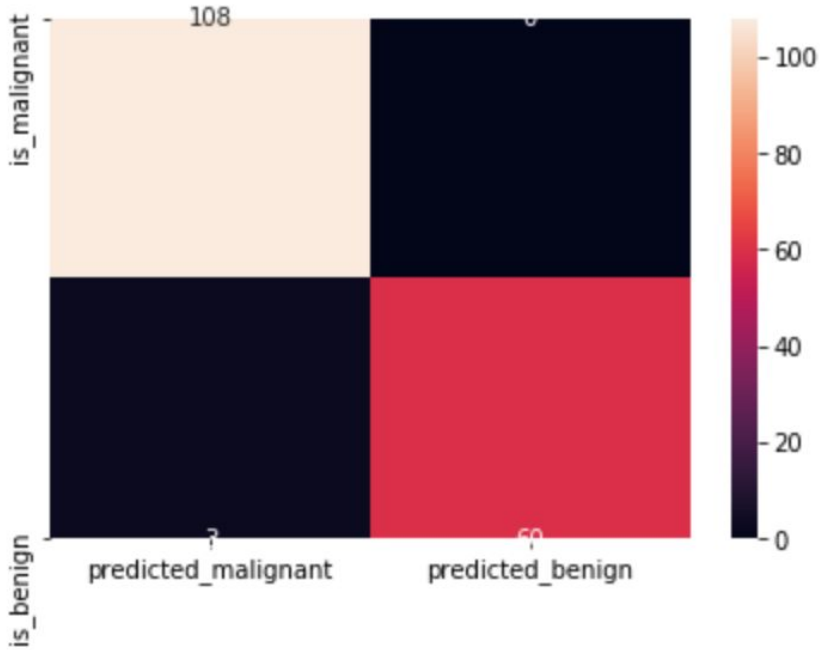


Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x1a28651dd0>



	predicted_malignant	predicted_benign
is_malignant	108	0
is_benign	3	60

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1a26ca1e10>

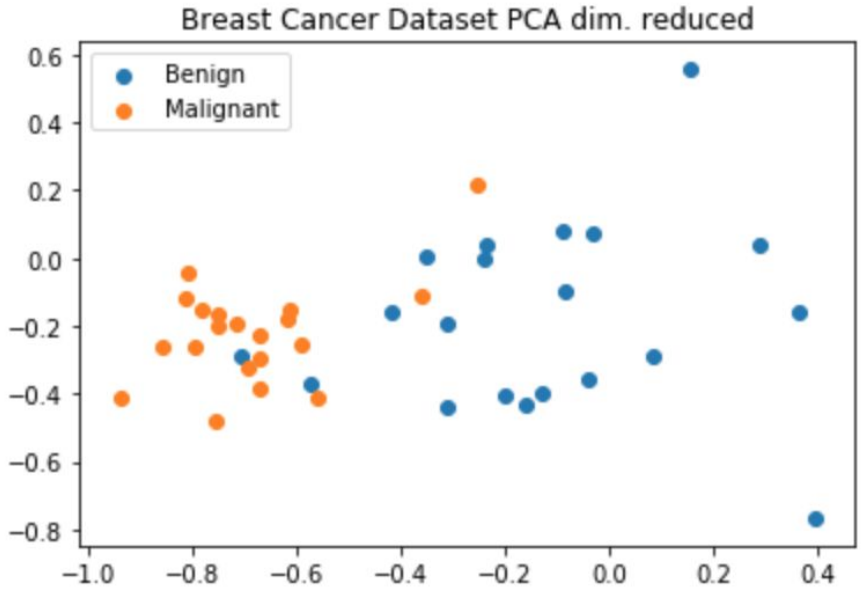


```
from qiskit_aqua.utils import split_dataset_to_data_and_labels

n = 2 # How many features to use (dimensionality)
training_dataset_size = 20
testing_dataset_size = 10

sample_Total, training_input, test_input, class_labels = breast_cancer(training_dataset_size, test
ing_dataset_size, n)

datapoints, class_to_label = split_dataset_to_data_and_labels(test_input)
print(class_to_label)
```




```
{'Benign': 0, 'Malignant': 1} {0: 'Benign', 1: 'Malignant'}
```

In [5]:
```
from qiskit_aqua.input import SVMInput
from qiskit_qcgpu_provider import QCGPUProvider
from qiskit_aqua import run_algorithm

params = {
    'problem': {'name': 'svm_classification', 'random_seed': 10598},
    'algorithm': { 'name': 'QSVM.Kernel' },
    'backend': {'name': 'qasm_simulator', 'shots': 1024},
    'feature_map': {'name': 'SecondOrderExpansion', 'depth': 2, 'entanglement': 'linear'}
}

backend = QCGPUProvider().get_backend('qasm_simulator')

algo_input = SVMInput(training_input, test_input, datapoints[0])
%time result = run_algorithm(params, algo_input)
%time result = run_algorithm(params, algo_input, backend=backend)
```

```
CPU times: user 1min 10s, sys: 4.27 s, total: 1min 15s
Wall time: 1min 46s
CPU times: user 1min 2s, sys: 5.27 s, total: 1min 7s
Wall time: 1min 30s
```

```
seed = 10598

feature_map = SecondOrderExpansion(feature_dimension=feature_dim, depth=2, entanglement='linear')
qsvm = QSVM(feature_map, training_input, test_input)

backend = BasicAer.get_backend('qasm_simulator')
quantum_instance = QuantumInstance(backend, shots=1024, seed_simulator=seed, seed_transpiler=seed)

result = qsvm.run(quantum_instance)

print("testing success ratio: ", result['testing_accuracy'])
```

```
testing success ratio:  0.8
```

ProteinQure

designing novel drugs with machine learning and quantum computing