

# ANÁLISE DE COMPORTAMENTO DO SVM QUÂNTICO

## QUANTUM SVM BEHAVIOUR ANALYSIS

**Eduardo Pioli do Amaral**

Universidade Federal de São Carlos – UFSCAR

[eduardoamaral@estudante.ufscar.br](mailto:eduardoamaral@estudante.ufscar.br)

<https://orcid.org/0009-0008-4465-8636>

**Diego Saqui**

Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais

- IFSULDEMINAS - Campus Muzambinho

[diego.saqui@muz.ifsuldeminas.edu.br](mailto:diego.saqui@muz.ifsuldeminas.edu.br)

<https://orcid.org/0000-0002-9927-7063>



DOI: 10.18406/2359-1269v11n32024399

## Resumo

A computação quântica é tida como possível substituta da computação clássica em certos problemas, o que leva a proposição de versões quânticas de algoritmos convencionais. Este estudo exploratório busca verificar a acurácia de um *Quantum Support Vector Machine* (QSVM) de classificação binária e comparar com sua versão clássica (SVM), a fim de explorar seu funcionamento, verificando suas limitações e parâmetros mais influentes. Para tal, foi aplicada uma metodologia quantitativa, na qual gerou-se conjuntos de dados artificiais de disposições únicas de valores e proporções de dados e dimensões. Aplicou-se o SVM clássico nestes conjuntos e montou-se, posteriormente, o SVM quântico para executá-lo nos mesmos conjuntos de dados, de forma simulada. Extraíu-se os dados de funcionamento e resultados de ambos os lotes de execuções para compará-los entre si, sendo estes o *F1-Score*, *Recall Score* e Curva ROC para acurácia e teste de McNemar para discriminação estatística. Por fim, analisou-se os resultados do circuito do SVM quântico, a fim de pontuar não só os conjuntos de dados mais compatíveis com sua estrutura, como também possíveis hiperparâmetros a serem ajustados, ou sub-estruturas a serem consideradas para alteração.

**Palavras-chave:** Classificação. Computação Quântica. Máquina de Vetor Suporte.

## Abstract

*Quantum computing is taken as a possible substitute for classic computing in certain problems, which leads to proposals of quantum versions of conventional algorithms. This exploratory study aims to verify the accuracy of a Quantum Support Vector Machine (QSVM) for binary classification and compare it with its classic version (SVM), as a way to explore its inner workings, verifying its limitations and most influential parameters. For such, a quantitative methodology was applied, in which datasets of unique data distributions and relations of data and dimensions were generated. The classical SVM was applied to the datasets, and the quantum SVM was then assembled, and used, in a simulated way, to classify the same datasets. After both batches of executions, their data compared with one another, with them being the F1-Score, Recall Score and ROC curve for accuracy and McNemar's test for statistical discrimination. Then the QSVM results were analyzed, to distinguish the datasets that had higher compatibility, as well as hyperparameters that could be adjusted, or structures to consider altering.*

**Keywords:** Classification. Quantum Computing. Support Vector Machine.

## Introdução

Uma vertente da computação que tem se destacado é a quântica, uma complexa e inovadora forma de se apropriar dos efeitos aos quais os objetos quânticos estão sujeitos. A construção de computadores é meticulosa e variada; o *System One* da *International Business Machines Corporation* (IBM), por exemplo, possui os chamados *qubits*, estruturas feitas de materiais supercondutores chamados *superconducting transmon qubits*. Para simular as medições binárias comuns em sistemas digitais, dois níveis de energia distintos destes materiais são isolados, formando *de facto* o *qubit* (IBM Quantum, 2016).

A computação quântica teve seu início prático em meados da década de 80, com as falas de, principalmente, Feynman (1982, p.474-475, tradução do autor). Em suas palavras:

*Agora acontece que, até onde eu posso dizer, você pode simular isso com um sistema quântico, com elementos de computadores quânticos. Não é uma máquina de Turing, mas uma máquina de um tipo diferente. Se nós desprezarmos a continuidade do espaço e torná-lo discreto, e por aí vai, como uma aproximação (do mesmo modo em que nos permitimos no caso clássico), certamente parece ser verdade que todas as variadas teorias de campo têm o mesmo tipo de comportamento, e podem ser simuladas de todas as formas, aparentemente, com pequenos látices trabalhados de spins e outras coisas (FEYNMAN, 1982, p. 474-475).*

Desde então, pesquisas acerca do tema foram gradativamente surgindo. Até que um importante salto foi realizado na área com a proposição de lógicas que serviriam de base para as portas quânticas como a lógica de superposição, que é descrito como a computação em “diferentes universos”, ou a alteração de fases que manipulam as probabilidades (DEUTSCH, 1985, p. 109-112).

A perspectiva unicamente acadêmica acerca da computação quântica logo mudou por conta de um artigo que demonstrava a obtenção dos fatores primos de um número inteiro, em tempo poli-logarítmico  $O(n)$ . Tal fatoração é o mecanismo principal implementado pelo sistema de encriptação Rivest-Shamir-Adleman (RSA) (SHOR, 1994, p. 130-131), ou seja, a quebra da criptografia popularizou da área da computação quântica.

A superposição já denota as implicações que um computador quântico pode ocasionar. Isso os torna um objeto de estudo promissor, afinal, conforme já citado, um *qubit* pode apresentar não só os valores binários convencionais, como pode também entrar em superposição (descrito como estando em 1 e 0 ao mesmo tempo). *A posteriori*, computadores quânticos possuem o potencial de superar os limites de computadores convencionais, como é o caso da busca desestruturada. Os algoritmos clássicos não conseguem realizar a busca com complexidade menor que  $O(n)$ , mas o algoritmo de Grover realiza em  $O(\log n)$  através de probabilidades e estados de fase (GROVER, 1996, p. 213-214).

Uma técnica comum nas áreas de aprendizado de máquina (do inglês *machine*

*learning* - ML) é o emprego de modelos computacionais que, a partir de dados prévios, são capazes de aprender como realizar alguma tarefa específica. Mitchell (1997, p. 14-15) descreve o ML como envolvendo a busca num amplo espaço de hipóteses possíveis para determinar a que melhor se adequa aos dados observados e a qualquer conhecimento prévio do aprendiz. Os três tipos mais comuns são o aprendizado supervisionado, por reforço e o não-supervisionado. Baseado na contextualização apresentada, nota-se que o uso de mecanismos quânticos para potencializar o ML torna-se desejável.

Existem exemplos em que houve êxito em ao menos fundir algoritmos clássicos com técnicas quânticas, como é o caso de um artigo que prova de forma extensiva que é possível utilizar da computação quântica para potencializar o algoritmo *AdaBoost*, amplamente utilizado para fortalecer um algoritmo de aprendizado, tornando modelos imprecisos em acurados, conquistando uma melhoria quadrática em sua performance; tal método ficou conhecido como *Quantum Boosting* (ARUNACHALAM; MAITY, 2020, p. 14-33). Dada a extensão do ML, o escopo deste trabalho se limita à classificação, subcategoria do aprendizado supervisionado.

Consoante com o que foi previamente apresentado, o presente trabalho busca sondar a eficiência de um modelo convencional de classificador em relação à sua alternativa quântica, e por fim, analisar possíveis alterações a serem exploradas. Para isso, os *datasets* usados serão gerados de forma artificial compreendendo estruturas e valores distintos.

Isso será feito através de um algoritmo de classificação quântica chamado QSVM, que será mais detalhado posteriormente. Mas, em suma, ele replica o funcionamento de um *Support Vector Machine* (SVM), um classificador tradicional popular, com a diferença de que o QSVM usará um kernel quântico, baseado em circuitos quânticos.

## Superposição quântica

A priori, objetos quânticos são aqueles que podem ser descritos através da mecânica quântica, e portanto sua natureza passa a ser distintamente probabilística, como é o caso de, *inter alia*, átomos e elétrons; além do fato de se comportarem como partículas individuais e ondas concomitantemente (BORN, 1926; SCHRÖDINGER, 1926). A partir disso, um *qubit* tem acesso a muitas qualidades destes objetos, como a superposição quântica.

A superposição quântica descreve que uma partícula pode existir simultaneamente em todos os seus possíveis estados, até que uma interação ou medição qualquer atue com ela e, conseqüentemente, há o que é chamado de colapso de função de onda (DIRAC, 1930, p. 10-14). O colapso tem este nome pois partículas, apesar de objetos individuais, possuem o comportamento concomitante de onda, o que também possibilita o estranho comportamento que apresentam (DE BROGLIE, 1923, p. 540).

## Emaranhamento quântico

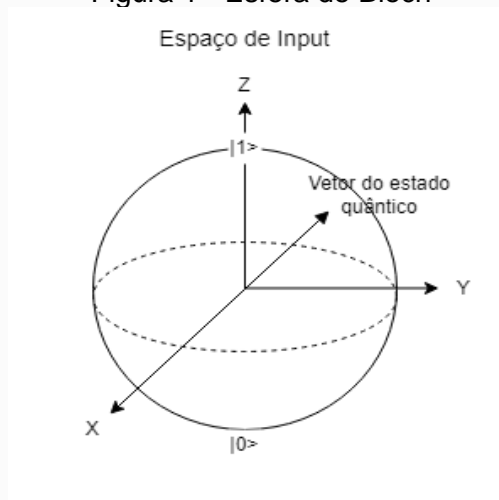
Um dos conceitos chave da mecânica quântica, que é utilizado em larga escala neste trabalho, é o emaranhamento quântico. O fenômeno de alta complexidade ainda não é totalmente compreendido, porém, a ideia fundamental por trás dele é: duas (ou mais) partículas são consideradas emaranhadas quando não podem ser descritas em sua totalidade de forma individual, ou seja, elas são interdependentes. Este emaranhamento porém é quebrado quando uma destas partículas é medida, uma vez que, como ela foi completamente descrita, seu par ou conjunto de pares devem ser igualmente descritos (PERES, 1995, p. 116-117).

No contexto deste trabalho, o emaranhamento é utilizado largamente em circuitos quânticos. No QSVM, o emaranhamento é realizado entre as dimensões do conjunto de dados em questão, ou seja, as features tornam-se interdependentes. Há várias arquiteturas padrões na biblioteca *Qiskit*, e são essas a serem empregadas durante os testes com o algoritmo.

## Amplitude e fase quântica

Há dois outros conceitos cruciais para o funcionamento de um circuito quântico, são eles a amplitude de probabilidade, ou seja, as chances de um *qubit* ser medido em um estados binários; e a fase quântica, uma consequência da dualidade onda-partícula da matéria, mais especificamente do comportamento de onda, onde a fase é uma fração do ciclo da mesma.

Figura 1 - Esfera de Bloch



Fonte: Autoria própria (2024)

Como um estado quântico é complexo, ou seja, descrito por uma parte real e uma complexa, a fase quântica abrange infinitas configurações de sua amplitude, ou seja, múltiplas formas de descrever as mesmas probabilidades. A Figura 1 mostra uma representação do estado quântico.

Pode-se observar que qualquer rotação nos eixos  $x$  ou  $y$  irá afetar somente a amplitude de probabilidades, que está entre o estado 0 e 1 respectivamente. Porém, uma rotação no eixo  $z$  causa uma mudança de fase, o que não afeta a amplitude, somente a direção do vetor (IBM Quantum, 2021).

## ***Support Vector Machine***

Antes de aprofundar um pouco sobre o SVM, vale ressaltar o que qualifica um aprendizado de máquina como supervisionado, elucidado pela descrição a seguir:

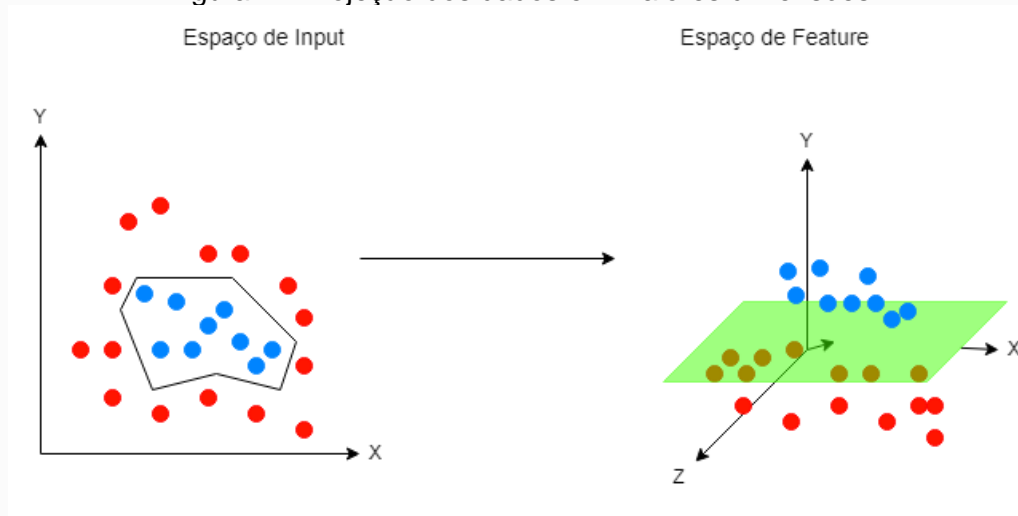
Aprendizado supervisionado: Primeiramente é utilizado um processo de treinamento, onde o modelo recebe um conjunto de dados de exemplo, ou seja, as entradas e suas respectivas saídas esperadas, e o modelo irá procurar dentre as hipóteses possíveis qual a melhor se encaixa na separação dos dados seguindo suas saídas esperadas. Espera-se que a hipótese encontrada não funcione só nos exemplos atuais, mas também em novas entradas. Para medir o quão ideal é a hipótese, são fornecidos dados de teste para avaliação do desempenho do algoritmo (NORVIG; RUSSEL, 2009, p. 695-696).

O SVM é um algoritmo de ML supervisionado aplicado a problemas de classificação e regressão. Originalmente, o método se baseava em criar uma função que descrevia um limite de decisão, de margem máxima, que separasse os dados linearmente em dois grupos. Ou seja, o SVM inicial contemplava apenas a classificação binária e a classificação linear dos dados propostos (CORTES; VAPNIK, 1995, p. 273-276).

Para evitar a linearidade do SVM foi proposto o *kernel trick*, que envolve substituir os produtos vetoriais do SVM por uma função *kernel*, não-linear e ponderada. Isso permite mapear os dados em dimensões superiores e encontrar hiperplano de maior margem separando os dados em dois ou mais grupos, conforme mostrado na figura 2. Um efeito indireto deste método também é ignorar os padrões atípicos que podem existir no *dataset*, reduzindo os impactos dos ruídos (BOSER; GUYON; VAPNIK, 1992, p. 145-149).



Figura 2 - Projeção dos dados em maiores dimensões



Fonte: Autoria própria (2024)

## Avaliação de acurácia

Um dos métodos de se avaliar uma classificação binária é através do *F1-Score* que é calculado através do *Recall* e *Precision*, respectivamente medidas de proporção de positivos verdadeiros e proporção de positivos previstos que batem com os positivos verdadeiros, e posteriormente é feita uma média harmônica do cálculo (POWERS, 2020, p. 37-40).

O *Recall Score*, também conhecido por *Sensitivity* é a taxa de verdadeiros positivos encontrados por um algoritmo, com todos os valores positivos (independentes da inferência do mesmo algoritmo); é uma métrica adequada para casos em que falsos negativos possuem maior impacto negativo, e em classificação binária.

Outro método comumente empregado é o da Curva ROC, ou *Receiving Operator Characteristic curve*, que demonstra de forma gráfica o quão bom é o modelo em classificar, baseando-se no seu limite de discriminação. O método se baseia na taxa de positivos verdadeiros e falsos positivos, sendo que quanto mais a curva se distorce para cima, melhor é o modelo de classificação (FAWCETT, 2006, p. 861-863).

## Teste de McNemar

O teste de McNemar é um teste estatístico baseado em uma tabela de contingência, ou seja, nas frequências de variáveis específicas, com atributos dicotômicos. Contextualizando para este trabalho, o teste usa os resultados previstos do SVM com os QSVM e suas diferentes configurações utilizadas.

Então, as variáveis dicotômicas são os acertos, isto é, se a classe prevista acertou, e se ambos os algoritmos concordam com o resultado previsto.

Através da chamada “hipótese nula” e um valor de tolerância arbitrário (neste trabalho foi escolhido 0.05) o teste calcula primeiramente o qui-quadrado, e posteriormente o valor  $p$  através de uma distribuição binomial. A hipótese nula sugere que não há diferença significativa entre os dois algoritmos quando executados num dado conjunto de dados (MCNEMAR, 1947).

Caso o valor  $p$  seja menor que a tolerância, há o descarte da hipótese nula, sugerindo então a diferença significativa entre a execução dos algoritmos, porém sem inferir nada sobre a eficiência da execução em si.

## **Quantum Support Vector Machine**

O SVM apropriou-se de uma ferramenta para elevar seu poder de classificação, a função *kernel*, responsável por cálculos cruciais do algoritmo. A partir deste conceito, Havlicek et al. (2018, p. 2-19) propuseram um *kernel* utilizando as propriedades da computação quântica. Este utiliza um mapa de estados quânticos derivados dos dados clássicos, e estima a fidelidade dos *qubits*; posteriormente, através da aplicação de portas controladas que manipulam a fase dos *qubits* e os deixam emaranhados, todos os estados são medidos e a sobreposição dos pontos é dada pela frequência de obtenção de instâncias onde todos os valores são nulos.

O mapa de *features* é outra técnica utilizada para a estimação do *kernel* quântico. Também utilizado no artigo de Havlicek et al. (2018, p. 2-19), o método constrói um circuito de  $n$  *qubits* representando as  $n$  *features* do *dataset*, decompondo os valores clássicos em estados quânticos através de uma versão quântica da *Singular Value Decomposition* (SVD) (GHOSH, 2018, p. 3-7).

## **Metodologia**

Toda metodologia foi realizada através do *Google Colaboratory*, uma plataforma em nuvem de desenvolvimento rápido em Python através de *notebooks*, que permite acesso à outros serviços *cloud* como RAM e GPU.

O *Qiskit* é a principal ferramenta do trabalho, permitindo a construção, manipulação e execução de circuitos e algoritmos quânticos, além da ponte entre o *Colab* e os serviços em nuvem da *IBM Quantum*.

As demais bibliotecas utilizadas foram o *Matplotlib* para visualização dos dados, *Numpy* para funcionalidades matemáticas e *pandas* para manipulação dos dados. O *scikit-learn* é amplamente utilizado para geração de conjuntos de dados e seu tratamento, uso do SVM e a maior parte das métricas de acurácia. Por fim, para realizar o teste de McNemar, foi utilizado o *Statsmodels*.



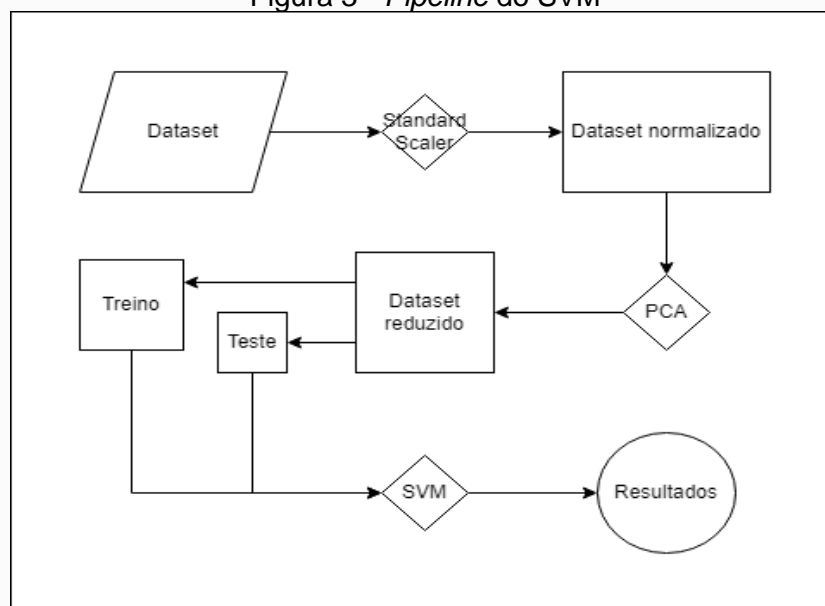
## Criação dos datasets

O `make_classification`, do `scikit-learn`, é um algoritmo que gera agrupamentos de dados formados por uma distribuição normal em torno dos vértices de um hiper-cubo, sendo que cada classe possui, por padrão, a mesma quantidade destes grupos. O algoritmo também possui diversos parâmetros configuráveis que mudam o comportamento da criação do conjunto de dados, como por exemplo determinar dimensões redundantes, introduzir ruído, entre outros. Os seguintes datasets foram gerados por meio desta funcionalidade:

- **Dataset *maxmin*:** O primeiro conjunto de dados almeja testar o QSVM quando há muitas dimensões, porém poucos dados a serem trabalhados. O *maxmin* possui 20 dimensões (+1 de classe) e apenas 30 instâncias.
- **Dataset *minmax*:** O segundo conjunto é o oposto, possuindo poucas dimensões e muitos dados. O *minmax* conta com 4 dimensões (+1 de classe), porém 9000 instâncias.
- **Dataset *unbalanced*:** O terceiro conjunto, *unbalanced*, foi criado com 500 instâncias e 5 dimensões (+1 de classe), porém 80% dos seus dados foram gerados como pertencentes a uma classe só, ou seja, o *dataset* gerado é enviesado.
- **Dataset *noisy*:** Por último, o conjunto *noisy*, possui a mesma quantidade de instâncias e dimensões que o anterior, porém este possui metade de suas instâncias atribuídas a qualquer uma das duas classes aleatoriamente, criando assim um *dataset* com ruído.

## Pipeline do SVM

Figura 3 - Pipeline do SVM



Fonte: Autoria própria (2023)

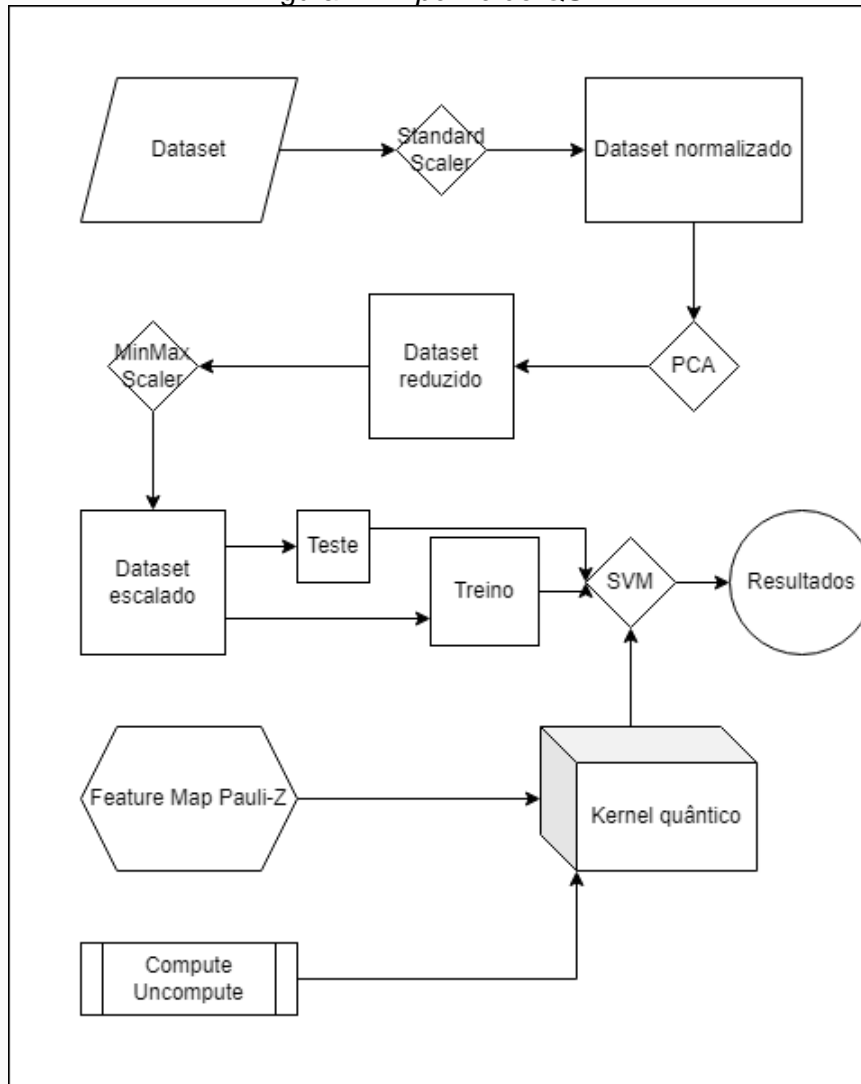
Nota-se pela Figura 3 que foi realizada a normalização dos dados, um procedimento padrão para evitar o *overfitting* através da média e reduzir a influência dos *outliers* no resultado. Então, aplica-se o *Principal Component Analysis* (PCA) reduzir a dimensionalidade dos dados, com propósito de padronizar as execuções com as do QSVM. Os dados então foram divididos em 80% para treino e os 20% para teste (abordagem chamada de *Hold-Out*); e por fim foram inseridos no SVM do *scikit-learn*, com *kernel* padrão *rbf*.

### **Pipeline do QSVM**

A pipeline do QSVM difere um pouco devido à preparação dos dados para serem quantizados (exemplificado na Figura 4). *A priori* deve ser criado uma *feature map* pelo circuito *Pauli-Z* de evolução de segunda ordem; este mapa serve para dispor os dados no circuito quântico, gerenciando as dimensões, o padrão de emaranhamento das *features* e o número de repetições do processo de evolução.



Figura 4 - Pipeline do QSVM



Fonte: Autoria própria (2023)

Um método comum na computação quântica é o *Uncompute*, que lida com o lixo gerado durante a execução do circuito. O descarte de *qubits* é equivalente a medi-los, ou seja, antes que o circuito conclua sua operação, haverá colapsos de função de onda o que impactará no resultado. Para isso, a *pipeline* implementa uma funcionalidade do *Qiskit*, o *ComputeUncompute* para calcular a fidelidade do circuito mediante o método descrito.

Com o mapa de *features* e a fidelidade do circuito, finalmente é construído o *kernel* quântico que possui, na prática, a mesma funcionalidade do *kernel* clássico no SVM; projetar os dados em um espaço dimensional superior para separá-los com um hiperplano.

Com estes modelos prontos é criado um modelo do *PegasosQSVC*, o QSVM utilizado no trabalho. Este modelo recebe o *kernel*, o parâmetro de regularização positiva (valor 500.0) que dita o quão resistente a *overfitting* ou o quanto de performance o QSVM possuirá; e também quantas vezes o algoritmo irá rodar, neste caso 100 vezes.

Os dados também devem ser tratados para melhor funcionamento, neste caso, a mesma divisão dos dados em treino e teste, e a mesma normalização realizada na *pipeline* de SVM também ocorrem nesta. O PCA é utilizado para atender as limitações de *qubits* de um computador quântico moderno, especialmente em planos gratuitos como o da IBM que apenas disponibiliza 7 *qubits*; e também devido ao impacto drástico do número de *qubits* de dimensões no tempo de execução dos algoritmos. Também deve ser aplicado o *MinMaxScaler*, a ideia por trás deste método é transformar o menor valor do conjunto de dados em 0 e o maior valor em 1, com os valores restantes sendo transformados para adequar a essa faixa binária. Isso se deve ao fato do QSVM ser um circuito que lida com valores binários, além dos valores probabilísticos.

### **Métricas de avaliação**

Após ambos algoritmos serem executados nos mesmos *datasets*, os resultados são extraídos e os cálculos das métricas de acurácia ocorrem, seguido do *plot* dos valores, representando os resultados iniciais do trabalho.

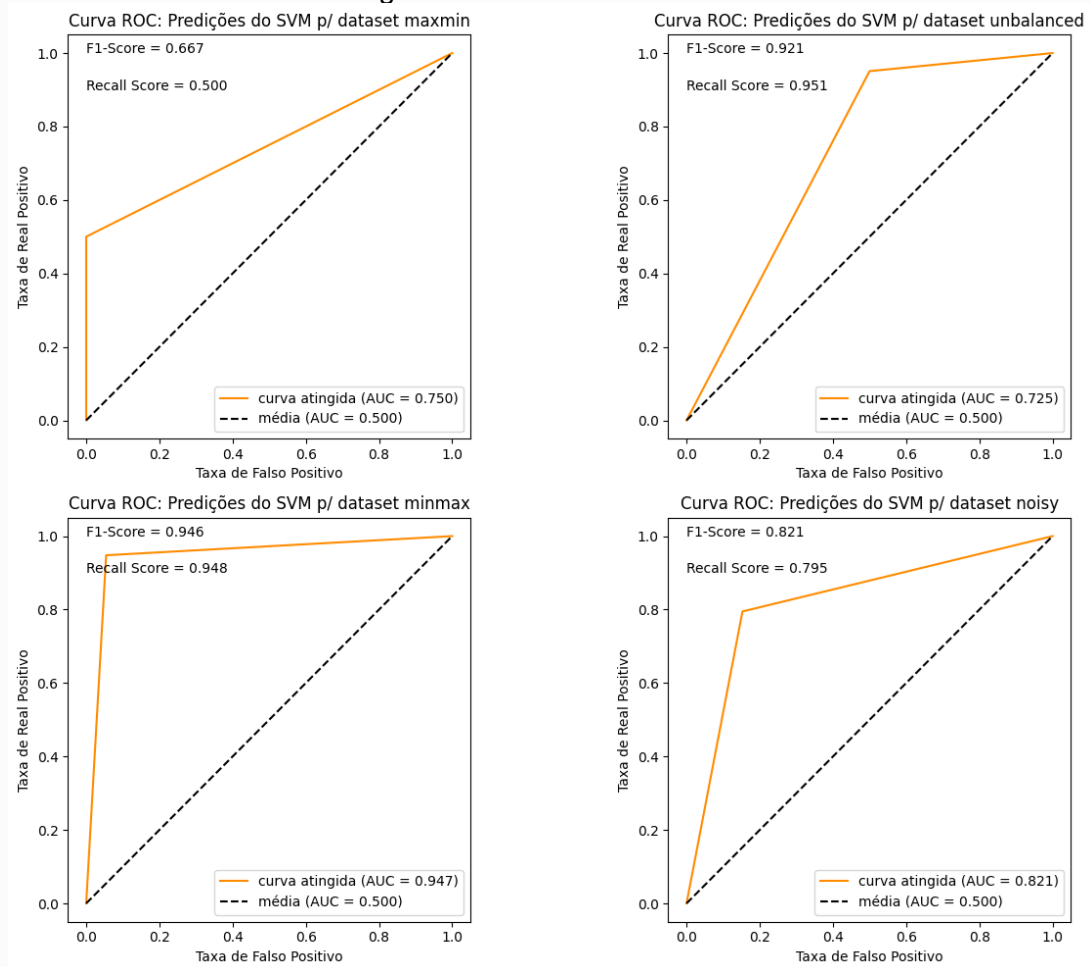
No caso, as métricas calculadas inicialmente são o *Recall Score* representando, dentre todos os resultados positivos (independente se o algoritmo os considera positivos ou negativos), a parcela de verdadeiros positivos que foram obtidos; o *F1-Score* que une o *Recall Score* com o *Precision Score* formando uma média harmônica, que ajuda a perceber se há equilíbrio entre as duas; e por fim, a curva ROC, a taxa de positivos verdadeiros contra a de falsos positivos.

É utilizada a métrica de McNemar para diferenciar estatisticamente a eficiência entre os classificadores sem recorrer ao *backpropagation*.

### **Considerações finais**

Os resultados da figura 5 correspondem ao modelo do SVM básico, e servirá de base de comparação para os modelos do QSVM exibidos em sequência na figura 6 (QSVM Linear e Linear Reverso), figura 7 (QSVM Completo), figura 8 (QSVM emparelhado) e figura 9 (QSVM circular e SCA).

Figura 5 - Resultados do SVM

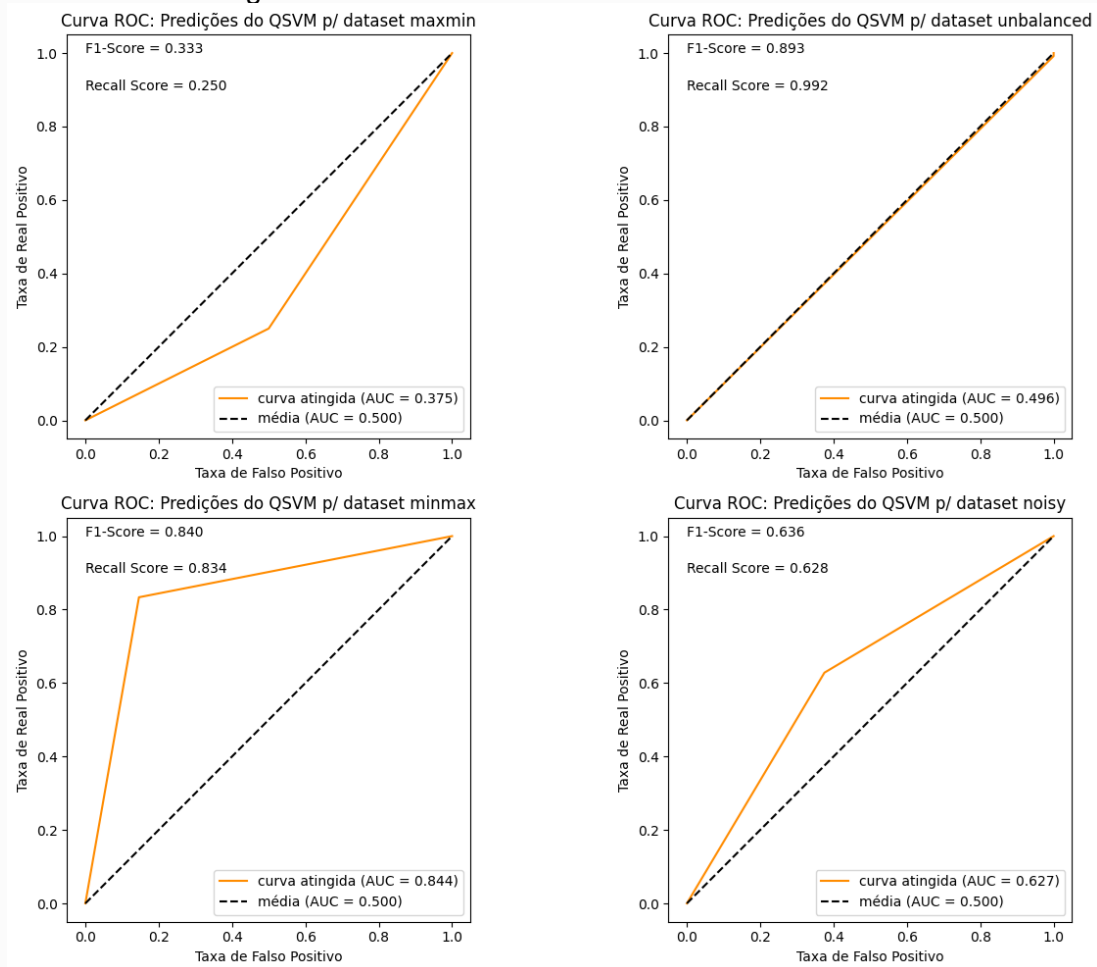


Fonte: Autoria própria (2023)

**Dataset maxmin:** O SVM apresentou boa distinção entre positivos e negativos, de acordo com a curva ROC; e a habilidade de acertar corretamente os positivos se mostrou mediana como indica o *Recall Score*, suportado pelo *F1-Score*. Já nos QSVM lineares, pela curva ROC, há baixa qualidade de discriminação; e o *F1-Score* mostra uma performance ruim tanto no *Recall* quanto no *Precision*.

**Dataset minmax:** O SVM apresentou muito boa discriminação entre positivos e negativos, e todas as métricas indicam o mesmo. Já nos QSVM lineares, todas as métricas possuem o mesmo padrão visto no SVM, porém desta vez com pequena perda de performance geral.

Figura 6 - Resultados QSVM linear e linear reverso



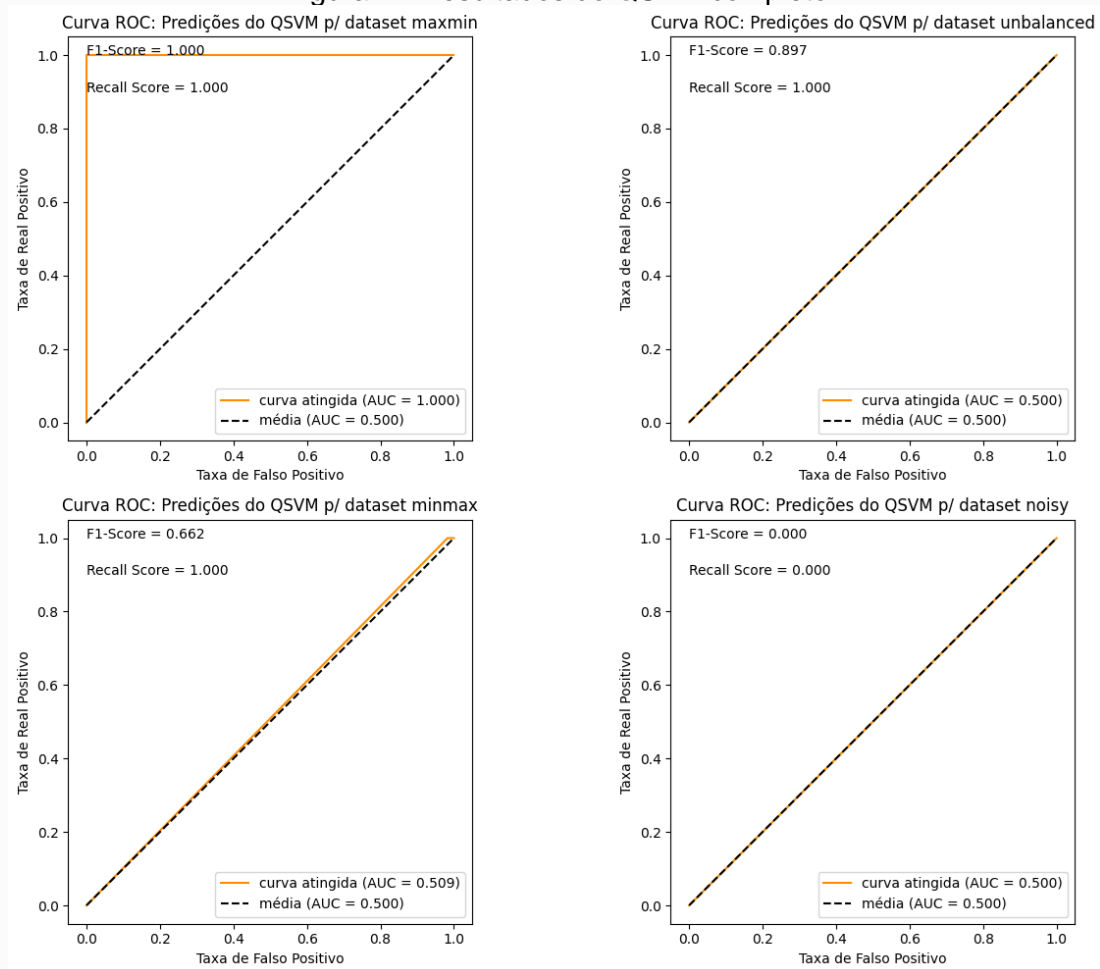
Fonte: Autoria própria (2023)

**Dataset unbalanced:** O SVM sugere uma eficiência quase perfeita através da *F1-Score* e o *Recall Score*, porém a curva ROC mostra que o modelo possui certo viés, uma vez que seu resultado está abaixo dos resultados das outras métricas. Já nos QSVM lineares, o padrão se repete com bons valores nas métricas, porém com baixo valor na curva ROC.

**Dataset noisy:** O SVM apresenta resultados similares aos do terceiro *dataset*. Já nos QSVM lineares, todos os valores decrescem em grande quantidade, mostrando desempenho medíocre.



Figura 7 - Resultados do QSVM completo



Fonte: Autoria própria (2023)

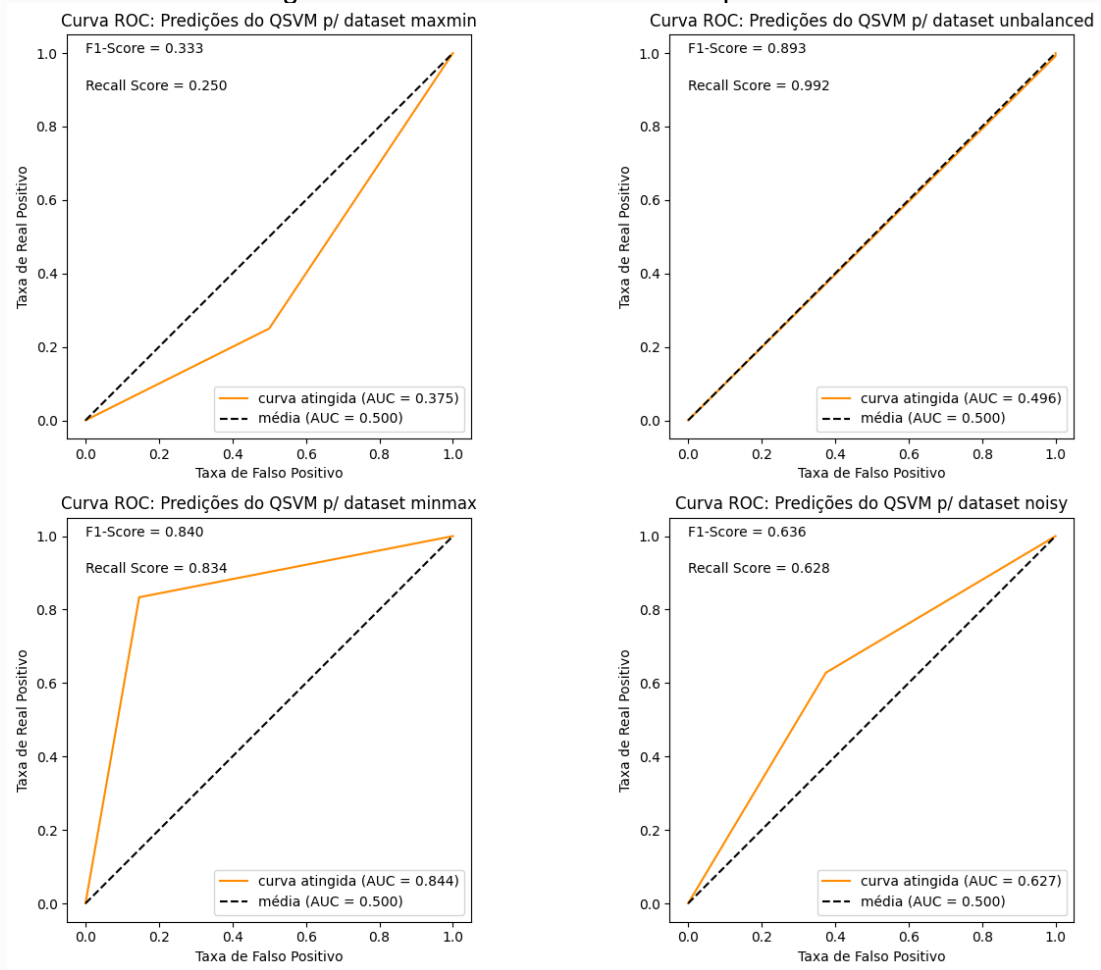
**Dataset maxmin:** Todas as métricas retornaram valor perfeito, o que é incomum e inesperado por possivelmente demonstrar uma capacidade total de discriminação. Há uma boa possibilidade que isso indique enviesamento.

**Dataset minmax:** Houveram bons resultados para a *Recall*, porém a curva ROC aliada à *F1-Score* mostram que mesmo assim os resultados foram verdadeiramente improdutivos.

**Dataset unbalanced:** Seguindo a mesma tendência do item anterior, vemos métricas numéricas como muito boas, mas a curva ROC indica o oposto; que há muito viés no conjunto de dados.

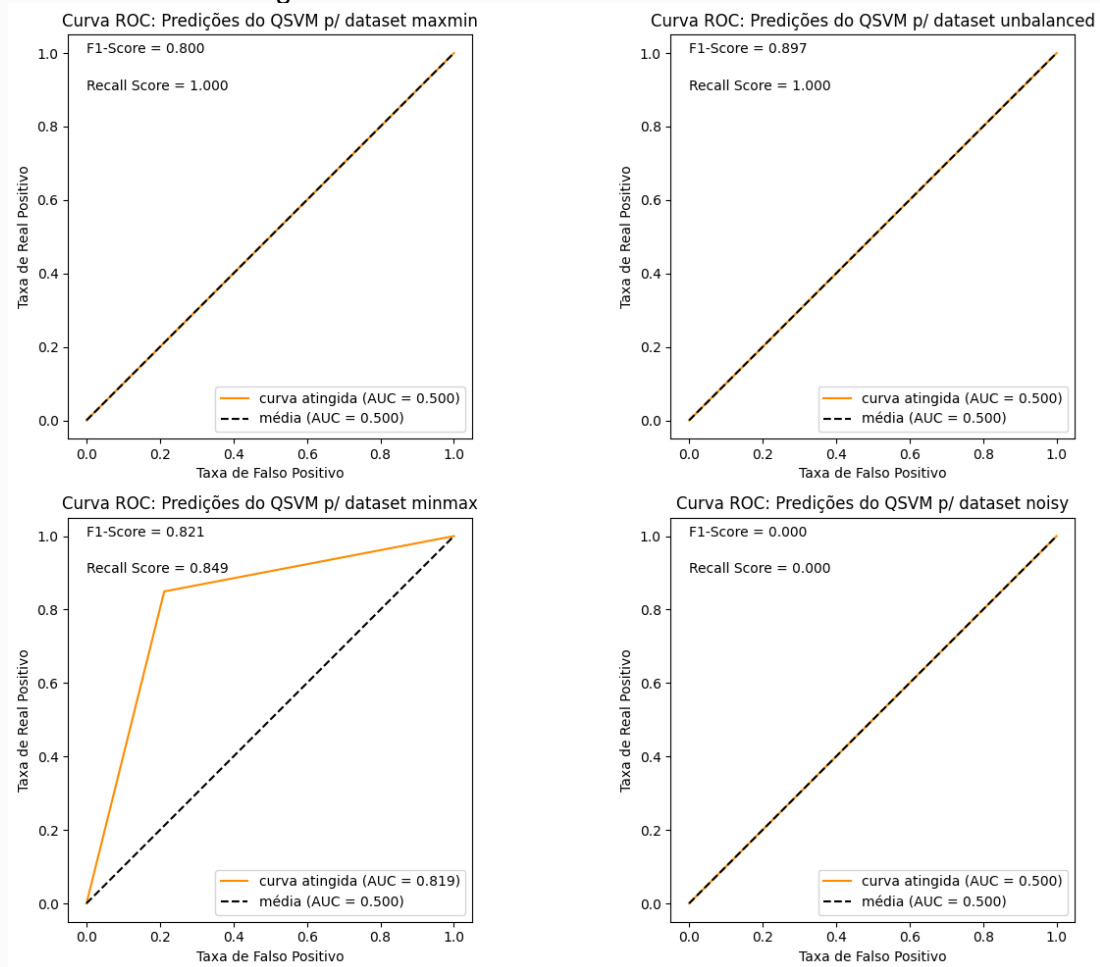
**Dataset noisy:** Por fim, pode-se notar que para o QSVM, temos notas realmente zeradas, o que possivelmente indica total incapacidade em aprender com conjuntos em que o ruído está muito forte.

Figura 8 - Resultados do QSVM emparelhado



Fonte: Autoria própria (2023).

Figura 9 - Resultados do QSVM circular e SCA



Fonte: Autoria própria (2023).

Todos os resultados destes modelos seguem os mesmos padrões observados nos modelos lineares e completo. Aqueles que não se assemelham a eles apresentaram forte enviesamento.

Conforme mostrado na Tabela 1, de acordo com a tolerância escolhida pode-se inferir quais modelos, em quais conjuntos de dados, há o descarte da hipótese nula. Para os *datasets maxmin* e *unbalanced* vemos valores acima de 0.05, ou seja, a hipótese se mantém, o que significa que durante a execução nestes conjuntos citados, tanto o SVM quanto todos os modelos do QSVM não apresentaram diferença significativa, sendo virtualmente semelhantes.

O mesmo não ocorre para os *datasets minmax* e *noisy*, onde todos os valores são extremamente menores que a tolerância, indicando uma execução estatística significativamente distinta entre os modelos do QSVM com o SVM para estes conjuntos, sinalizando que há uma diferença no comportamento.

Tabela 1 - Resultados do teste de McNemar

Emaranhamento	maxmin	minmax	unbalanced	noisy
Linear	0.500	2.074e <sup>-36</sup>	0.078	8.171e <sup>-5</sup>
Linear reverso	0.500	2.074e <sup>-36</sup>	0.078	8.171e <sup>-5</sup>
Completo	0.500	7.573e <sup>-193</sup>	0.115	9.089e <sup>-10</sup>
Emparelhado	0.500	2.074e <sup>-36</sup>	0.078	8.171e <sup>-5</sup>
Circular	1.000	2.074e <sup>-36</sup>	0.078	8.171e <sup>-5</sup>
SCA	1.000	5.075e <sup>-44</sup>	0.115	9.089e <sup>-10</sup>

Fonte: Autoria própria (2024).

## Considerações finais

Com a conclusão deste trabalho pôde-se demonstrar que a arquitetura de emaranhamento do circuito quântico utilizado influencia amplamente sua execução e seu comportamento, uma vez que a maioria apresentou ao menos alguma peculiaridade nos resultados em relação aos demais. Porém não foi possível notar alguma arquitetura ideal, uma vez que, de modo geral, todos os resultados foram piores que os do SVM.

Devido à complexidade do QSVM e a quantidade de variáveis em torno de sua execução, há mais caminhos e hiperparâmetros a serem explorados, como o parâmetro  $C$ , responsável por lidar com o *overfitting* ou a performance geral, ou configurações customizadas de emaranhamento, por exemplo, cada problema e conjunto necessitando um padrão próprio.

De modo geral, como trabalhos envolvendo classificadores utilizam grandes conjuntos de dados adequadamente tratados, o QSVM ainda não deve ser descartado, uma vez que o *dataset* mais viável, o *minmax*, apresentou resultados que se aproximaram dos resultados do SVM o suficiente.

*Datasets* compatíveis também são uma área de estudo, uma vez que o QSVM, conforme a literatura, em especial o artigo de Havlicek et al. (2018), apresenta êxito em *datasets* matematicamente complexos, ou seja, em que as relações estatísticas entre os dados sejam altamente intrincadas, algo que transcende a natureza das dimensões dos dados, ou o sentido de cada *feature*.

Em relação ao futuro do QSVM, além das alterações do *kernel* apresentadas, há trabalhos atuais que cogitam o desenvolvimento de um QSVM puramente quântico, ou seja, não só o *kernel*, como todo processo de classificação seguinte, como é o caso do artigo de Kariya e Behera (2021).

## Referências

ARUNACHALAM, Srinivasan; MAITY, Reevu. **Quantum Boosting**. ArXiv: quant-ph, 2002.05056, Ago. 2020. Disponível em: <https://arxiv.org/abs/2002.05056>. Acesso em: 15 Fev. 2023.

BORN, Max. Zur Quantenmechanik der Stoßvorgänge. **Zeitschrift für Physik A Hadrons and Nuclei**, Berlin, v. 37, n. 12, p. 863-867, Dec. 1926. Disponível em: <https://link.springer.com/article/10.1007/BF01397477>. Acesso em: 16 Jan. 2023.

BOSER, Bernhard E.; GUYON, Isabelle M.; VAPNIK, Vladimir N. A training algorithm for optimal margin classifiers. **COLT '92: Proceedings of the fifth annual workshop on Computational learning theory**, New York, p. 144-152, Jul. 1992. Disponível em: <https://dl.acm.org/doi/10.1145/130385.130401>. Acesso em: 14 Mai. 2023.

CORTES, Corinna; VAPNIK, Vladimir. Support-Vector Networks. **Machine Learning**, [s.l.], v. 20, n. 3, p. 273-297, Set. 1995. Disponível em: <https://link.springer.com/article/10.1007/BF00994018>. Acesso em: 15 Jan. 2023.

DE BROGLIE, Louis. Waves and Quanta. **Nature**, London, v. 112, n. 2815, p. 540, Out. 1923. Disponível em: <https://www.nature.com/articles/112540a0>. Acesso em: 16 Jan. 2023.

DEUTSCH, David. Quantum theory, the Church-Turing principle and the universal quantum computer. **Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences**, London, v. 400, n. 1818, p. 97-117, Jul. 1985. Disponível em: <https://royalsocietypublishing.org/doi/10.1098/rspa.1985.0070>. Acesso em: 20 Jan. 2023.

DIRAC, Paul A. M. **The Principles of Quantum Mechanics**. 3. ed. Oxford: Oxford University Press, 1930.

FAWCETT, Tom. An introduction to ROC analysis. **Pattern Recognition Letters**, Amsterdam, v. 27, n. 8, p. 861-874, Jun. 2006. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S016786550500303X>. Acesso em: 14 Mai. 2023.

FEYNMAN, Richard P. Simulating Physics with Computers. **International Journal of Theoretical Physics**, Pasadena, v. 21, n. 6-7, p. 467-488, Mai. 1982. Disponível em: <https://link.springer.com/article/10.1007/BF02650179>. Acesso em: 20 Jan. 2023.

GHOSH, Kumar J. B. **Encoding classical data into a quantum computer**. ArXiv: quant-ph, 2107.09155, Jul. 2021. Disponível em: <https://arxiv.org/abs/2107.09155>. Acesso em: 2 Jan. 2023.

GROVER, Lov. K. A fast quantum mechanical algorithm for database search. **STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing**, Philadelphia, p. 212-219, Jul. 1996. Disponível em: <https://dl.acm.org/doi/10.1145/237814.237866>. Acesso em: 20 Jan. 2023.

HAVLICEK, Vojtech. et al. **Supervised learning with quantum enhanced feature spaces**. ArXiv: quant-ph, 1804.11326, Jun. 2018. Disponível em: <https://arxiv.org/abs/1804.11326>. Acesso em: 2 Jan. 2023.

IBM. **IBM Quantum**, 2016. Learn quantum computing: a field guide - The qubit. Disponível em: <https://quantum-computing.ibm.com/composer/docs/ixq/guide/the-qubit>. Acesso em: 15 Jan. 2023.

IBM Quantum. **Qiskit: An open-source framework for quantum computing**. Versão 0.29.1. [S.l.]: IBM Quantum, 2021. Disponível em: <https://qiskit.org/documentation>. Acesso em: 18 Abr. 2023.

IBM Quantum Documentation. **Introduction**. [s.n.], [s.l.]: IBM Quantum, [s.d.]. Disponível em: <https://docs.quantum.ibm.com/start>. Acesso em: 13 Dez. 2023.

IBM Quantum Documentation. **Circuit library**. [s.n.], [s.l.]: IBM Quantum, [s.d.]. Disponível em: <https://docs.quantum.ibm.com/build/circuit-library>. Acesso em: 13 Dez. 2023.

KARIYA, Anekait; BEHERA, Bikash K. **Investigation of Quantum Support Vector Machine for Classification in NISQ era**. ArXiv: quant-ph, 2112.06912, Dez. 2021. Disponível em: <https://arxiv.org/pdf/2112.06912.pdf>. Acesso em: 13 Dez. 2023.

LIU, Cheng. et al. **Improving the Success Rate of Quantum Algorithm Attacking RSA Encryption System**. ResearchGate, Mar. 2022. Disponível em: [https://www.researchgate.net/publication/359643607\\_Improving\\_the\\_Success\\_Rate\\_of\\_Quantum\\_Algorithm\\_Attacking\\_RSA\\_Encryption\\_System](https://www.researchgate.net/publication/359643607_Improving_the_Success_Rate_of_Quantum_Algorithm_Attacking_RSA_Encryption_System). Acesso em: 13 Dez. 2023.

LLOYD, Seth; MOHSENI, Masoud; REBENTROST, Patrick. Quantum Support Vector Machine for Big Data Classification. **Physical Review Letters**, Ridge, v. 113, n. 13, p. 130503, Set. 2014. Disponível em: <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.113.130503>. Acesso em: 1 Jun. 2023.

MCNEMAR, Quinn. Note on the sampling error of the difference between correlated proportions or percentages. **Psychometrika**. [s.l.], v. 12, n. 2, p. 153-157. Jun. 1947. Disponível em: <https://link.springer.com/article/10.1007/BF02295996>. Acesso em: 10 Nov. 2023.

MITCHELL, Tom M. **Machine Learning: A multistrategy approach**. 1. ed. New York: McGraw-Hill Science, 1997.

NORVIG, Peter; RUSSELL, Stuart. **Artificial Intelligence: A Modern Approach**. 3. ed. Upper Saddle River: Prentice Hall, 2009.

PERES, Asher. **Quantum Theory: Concepts and Methods**. 1995. ed. [s.l.]: Springer, 1995.

POWERS, David M. W. **Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation**. ArXiv: cs.LG, 2010.16061, Out. 2020. Disponível em: <https://arxiv.org/abs/2010.16061>. Acesso em: 14 Mai. 2023.

SCHRÖDINGER, Erwin. An Undulatory Theory of the Mechanics of Atoms and Molecules. **The Physical Review**, College Park, v. 28, n. 6, p. 1049-1070, Dez. 1926. Disponível em: <https://journals.aps.org/pr/abstract/10.1103/PhysRev.28.1049>. Acesso em: 16 Jan. 2023.



SHOR, Peter W. Algorithms for quantum computation: discrete logarithms and factoring. **Proceedings 35th Annual Symposium on Foundations of Computer Science**, Santa Fe, v. 35, p. 124-134, Nov. 1994. Disponível em: <https://ieeexplore.ieee.org/document/365700>. Acesso em: 20 Jan. 2023.

ZEH, Heinz D. On the interpretation of measurement in quantum theory. **Foundations of Physics**. [s.l.], v. 1, p. 69-76. Mar. 1970. Disponível em: <https://link.springer.com/article/10.1007/BF00708656>. Acesso em: 13 Dez. 2023.