AQSVC to determine credit risk using Statlog German Credit Data

Pratham Gujar

January 2025

1 Abstract

Using my algorithm, I found that using 2 repetitions with linear entanglement and 4 qubits yielded the best result of 0.67 minority recall. Different results could be achieved by changing the test/train split but it will not vary by too much.

2 Introduction

AQSVC is a novel technique used to automatically determine hyperparameters for quantum feature maps. It takes in a quantum feature map and evaluates its kernel matrix. Using this, the algorithm uses SVC on the generated quantum feature map circuit to train the machine learning model. For the credit risk assessment accuracy, I am most interested in the minority recall as it evaluates how well the model identified the minority target value. This is crucial since the minority target value coincided with individuals deemed as credit risks. Correctly identifying more than 50% of the credit risks would significantly reduce any chances of providing services to individuals who may cause us more loss than gain.

For this project, a **ZZFeatureMap** was used for its pairwise interactions between qubits. This allowed me to consider potential interactions between features and how they affect the target outcome. The hyperparameters optimized for the feature map were repetitions (**reps**), entanglement (**entanglement**), and number of qubits (**n_dims**). Altering these affects the way feature variables are mapped into the quantum circuit. Using an optimization wrapper function, we can find the optimal hyperparameters to maximize our model training.

3 Tools

For this, I utilized Qiskit for modeling my machine learning implementation. I also used a preprocessed version of the Statlot German Credit data where the alphanumerical values in some columns are restructured into their own individual columns with numerical values. For data handling, I used pandas, numpy, and sklearn. Much of the preprocessing utilizes functions from sklearn.

4 Technique

First, I filtered the German Credit Data to find features more closely correlated to the minority target value. I tweaked this to increase minority recall and maximize the overall accuracy score for the SVC fitting. This is found in the **adjusted_features()** function.

I preprocess the restructured German Credit Data using PCA to reduce the dimensionality to match the number of qubits. For example, if the data is reduced to 4 dimensions, then there are 4 qubits to represent the features.

The features are then normalized using a Z-score normalization to have a mean of 0 and a standard deviation of 1. The features are then scaled to -1 and 1 to facilitate machine learning fitting. Finally, a sample of the training and testing data is selected to minimize computational computing.

I chose a **ZZFeatureMap** to map the features to the quantum map. This handles pairwise interactions between qubits which considers interactions between features and how they affect the produced target. I create a quantum kernel using the feature map and the kernel is evaluated when using the **qsvc** function. The evaluated kernel matrix is used in the generated quantum circuit to find the best model to predict test data. Refer to the **run_qsvc()** function to see the implementation.

The **optimize_qsvc()** function wraps around the **run_qsvc()** function to try various quantum hyperparameters such as repetitions **reps**, entanglement **entanglement**, and the number of qubits **n_dims**. There are default values assigned if the user does not feed in **test_params**.

5 Results

With my strategy, I determined that 2 repetitions of the feature circuit, with a linear entanglement, and 4 qubits yielded the best minority recall of 0.67. The testing was not comprehensive since I was limited by computational power and there could be a better configuration. However, I noticed that as the qubit number deviated from 4, the minority score decreased until reaching 0. Thus, it appears that 4 qubits may already be an optimized hyperparameter for the QSVC solution.

6 Conclusion

My code found a configuration where the minority recall is above 0.5 without sacrificing much from the accuracy score (0.75). The **ZZFeatureMap** provided the best results as it considered interactions between features and mapped the features the best. A custom Pauli Map could be implemented to better map the features but more investigation is required. The results obtained did satisfy the condition for achieving a minority score of 0.5.