Hybrid quantum-classical metaheuristics for automated machine learning applications

Von Dollen, D.J.

Chapter 3

Quantum-enhanced feature selection

In this chapter we approach the feature selection problem in machine learning and AutoML. We investigate various classical distance measures for encoding data in QUBO form which can be used by quantum systems to solve the feature subset selection problem. We test these methods of quantum-assisted subset selection over a variety of datasets and use classical estimators for the real-world application of vehicle price prediction. This has been published in:

[228] D. Von Dollen, D. Weimer, F. Neukart, and T. Bäck. Predicting vehicle prices via quantum-assisted feature selection. *International Journal of Information Technology*, 15(6):2897–2905, 2023. doi: 10.1007/s41870-023-01370-z. URL `https://doi.org/10.1007/s41870-023-01370-z`.

3.1 Introduction

Data pre-processing techniques, along with exploratory analysis and feature engineering, are standard steps within pipelines for predictive model training and deployment [112]. In practice, before feeding data to a learning algorithm, it is common to cleanse data sets to improve the predictive performance and generalizability of models.

Feature selection (FS) is a preprocessing technique [222], in which a subset of features is selected from a data set that may contain more relevant information than the total set of features, due to collinearity, redundant or constant features within the to-

tal data set. Benefits to utilizing FS within a preprocessing step include reduced data dimensionality, better scaling for learning algorithms in practice, and reduced costs in regard to model training and fitting. Feature selection differs from other techniques, such as feature engineering, in that it does not create a projection of linear combinations of features into a new subspace, or a change of basis using an eigendecomposition of a covariance data matrix, as in the case of Principal Component Analysis [159], but rather filters down and eliminates features with low information in regard to a target variable. This is an important property, as the sparse representation and preservation of the original feature space allows for greater visibility and explainability when accounting for predictive model output [100]. Feature selection techniques may leverage an output from an estimator or model in the selection process, also known as *wrapper* methods [147] or may use heuristics and select an optimal subset, also known as *filter* methods. Feature selection methods have been used in a variety of applications, including areas such as character recognition and healthcare[196, 53, 194, 144]. In this work, we examine using quantum-assisted methods to find combinations of features of a subset of size k, which may show improved performance characteristics and a decrease in the input feature space for our learning algorithms. Feature subset selection is a known NP-Hard combinatorial optimization problem [39].

Our contribution is as follows: framing the combinatorial search as a binary quadratic model, we examine using a quantum device to search the combinatorial space of finding optimal feature subsets of size k. We investigate various distance and correlation measures, which we compute classically for the formulation of the binary quadratic optimization problem. We apply the heuristic of Maximal Relevancy Minimal Redundancy (MRMR) [85] as an approach to the feature selection problem, with the idea that we want to find a subset of features that may have a strong predictive signal in regards to a target variable (Maximal Relevancy), but low pairwise feature correlation between independent variables (Minimal Redundancy). We train and compare two types of regression models using quantum-assisted feature selection along with other benchmark selection methods including all features, greedy selection, and a wrapper method, over two data sets with continuous target values. We examine model predictive performance using this methodology and apply it to a real-world problem of data preprocessing for predictive models for vehicle prices. We show that by using quantum-assisted routines, we find combinations of features that increase the predictive quality of models on validation sets of data, and improve upon our benchmarks of all features, greedy feature selection, and recursive feature elimination.

3.1.1 Related Work

The problem of feature selection has been well studied in the literature [18, 140]. Recent approaches apply mutual information-based measures for feature selection for supervised learning in classification applications [148]. In recent years, new correlation measures have been proposed which may have more expressive power in measuring relationships between variables. Examples of this are the maximum information coefficient (MIC) [172] and the generalized mean information coefficient (GMIC) [125], which introduce the concept of *equitability* in variable relationships and may be more robust when dealing with non-linear relationships than correlation statistics such as the Pearson correlation coefficient, which assume linear relationships.

In the current era of quantum computing, the availability of devices that leverage quantum processing units (QPUs) has come online, and applications that leverage these devices have been developed for solving real-world problems within various industries. For example, in [145], the authors leveraged a quantum annealing system to optimize traffic flows around the city of Beijing, and the authors in [205] showed how to price options using quantum algorithms run on a gate-model quantum chip.

The feature subset selection problem was formulated by the authors in [173] as a quadratic program, where mutual information and Pearson's correlation coefficient were used to calculate the matrix $\mathbf{Q}$, or a symmetric positive semidefinite matrix representing quadratic terms to minimize the objective function of multiple variables. There have been research efforts to apply quantum annealing to searching feature space for optimal subsets using mutual information-based formulations of interactions and linear terms for Ising spin-glass models and quadratic unconstrained binary optimization [137, 195]. Some of these efforts have included complexity considerations in regard to scaling for the quantum-assisted feature subset selection problem, claiming performance gains of $O(1/m^2)$ versus $O(mn^2)$ for classical computation [195]. This work in particular claims a bound for the quantum-assisted routine given by the size of the minimum gap $g(t) = E_0 - E_1$ in the energy eigenvalues during the annealing regime, with a resulting time complexity bounded by the adiabatic evolution time to maintain the ground state, of $T = O\left(\frac{1}{g_{\min}^2}\right)$ where T is the upper time limit.

In regard to the price prediction problem, this has been well studied in the literature for machine learning research [166, 139, 150]. In [166], the authors built predictive models for the prediction of the price of used cars in Mauritius. The models included naive Bayes and decision tree algorithms, with reported accuracy rates in a range of 60-70%, and achieved a mean error of 51000 and 27000 using linear regression. Mon-

burinon et al. tested various regression models for price prediction for German used cars in [139], with gradient-boosted decision trees outperforming random forest and multiple regression (mean squared error = 0.28). In [150] the authors applied feature selection techniques for multiple regression models to predict prices and reported the score with respect to the fitness of the model ($R^2 = 0.9861$).

3.2 Methods

Consider a data set:

$$\mathcal{D} = \mathbf{X} \in \mathbb{R}^{n \times m}, \mathbf{y} \in \mathbb{R}^n$$

where $\mathbf{X}$ is a data matrix of size $n \times m$ of n rows and m columns, and $\mathbf{y}$ is a column vector of size $n \times 1$ of n rows, and 1 column. We wish to find an approximate functional mapping or hypothesis of $h(\mathbf{X}) \simeq \mathbf{y}$ using various learning algorithms. This is also known as supervised learning, and our goal is to minimize a generalization error for a given loss function between predictions and a target variable in order to make predictions given new data or find the best hypothesis from the hypothesis space which the given learning algorithm encompasses.

In the feature subset selection problem we wish to find a subset of features $\mathbf{X_k}$ where each row vector $\mathbf{x_i}$ in $\mathbf{X}$ (where $i = \{1, 2, ..., n\}$) is of reduced column size k, i.e each row vector is filtered down as in $\{x_{i1}, x_{i2}, \ldots, x_{ik}\}$ where $k < m$. We may investigate various values of k such that the loss for our function $h(\mathbf{X_k}) \simeq \mathbf{y}$ is less than or equal to $h(\mathbf{X}) \simeq \mathbf{y}$ cross-validated on a holdout test set of data. Essentially, the problem of selecting the feature subset selection problem is one of choosing the optimal subset of features or columns of $\mathbf{X}$ of size k. In this problem, we assume that there exists an optimal subset of size k, which may not be the case for all data sets, where the optimal set $h(\mathbf{X_k}) = h(\mathbf{X})$ or where $k = m$.

3.2.1 Relevancy, Correlation and Distance Measures

To compare features with a target variable and investigate pairwise relationships between the set of features $\mathbf{X}$, we must first define our distance functions. These functions are used as input to formulate the binary optimization problem for selection of feature subsets. The selected features are then used as input for our predictive models. In our experiments, we compare each distance function when used to model linear and quadratic terms for a binary quadratic model. Note that we compute these distances classically before using them as input in the form of quadratic terms of a

binary quadratic model, which we sample using a quantum annealer to get subsets of features. We follow this process in order to understand which distance measures may be attributed to higher predictive power for models.

We examined the maximum information coefficient (MIC), the generalized mean information coefficient (GMIC), the mutual information coefficient (MI), and the Pearson correlation coefficient (PCC) for this study. Let us define each in the following:

Mutual information (MI) is defined as:

$$MI(\mathbf{x}, \mathbf{y}) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) \ln \left(\frac{P(x, y)}{P(x)P(y)} \right) \tag{3.1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are data vectors from which the probability distributions are estimated, $\mathcal{X}$ and $\mathcal{Y}$ are their respective sample spaces, $P(x, y)$ is the joint probability mass function estimated from the data, and $P(x)$ and $P(y)$ are the marginal probability mass functions. For continuous variables, the summations are replaced by integrals.

In our case, we are looking at mixes of purely continuous variables as well as continuous and discrete variables, with which there are various strategies for binning continuous values to estimate probabilities for the MI calculation. These include kernel density estimation, binning continuous to discrete variables, and clustering algorithms. For our method, we use the k-nn binning strategy from [176] to estimate the MI:

$$I(\mathbf{x}, \mathbf{y}) = \psi(N) - \langle \psi(n_x) \rangle + \psi(\kappa_{nn}) - \langle \psi(n_y) \rangle \tag{3.2}$$

where ψ is the digamma function, N is the total number of samples, κ_{nn} is the number of nearest neighbors, and n_x and n_y are the number of neighbors within the κ_{nn}-distance in the $\mathbf{x}$ and $\mathbf{y}$ marginal spaces, respectively. Angle brackets $\langle \cdot \rangle$ denote the expectation over all samples, per the standard defined in [200]. For more details, see [200].

Various algorithms have been proposed to handle a mix of discrete and continuous and purely continuous algorithms based on mapping mutual information to a grid. MIC and GMIC belong to this category.

MIC is defined as [125]. First, we define a *Characteristic* Matrix C:

$$C(\mathbf{x}, \mathbf{y})_{i,j} = \frac{I^*(\mathbf{x}, \mathbf{y})_{i,j}}{\log_2 \min\{i, j\}} \tag{3.3}$$

where $I^*(\mathbf{x}, \mathbf{y})_{i,j}$ represents the maximum mutual information over all $i \times j$ grids $G_{i,j}$ partitioning the space of $(\mathbf{x}, \mathbf{y})$, with i bins for $\mathbf{x}$ and j bins for $\mathbf{y}$.

Then we obtain the MIC using this characteristic matrix:

$$MIC(\mathbf{x}, \mathbf{y}) = \max_{i \cdot j < B(n)} \{C(\mathbf{x}, \mathbf{y})_{i,j}\} \tag{3.4}$$

where $B(n) = n^{0.6}$ is the maximum grid size as recommended in the original text [172], and n is the number of data points. Note that we consider $i \cdot j$ as a product denoting the total number of cells in the grid.

We can define GMIC by taking the characteristic matrix and extending it to find the maximal characteristic matrix:

$$C^*(\mathbf{x}, \mathbf{y})_{i,j} = \max_{k \cdot l < i \cdot j} \{C(\mathbf{x}, \mathbf{y})_{k,l}\} \tag{3.5}$$

where the maximum is taken over all grid sizes (k, l) such that $k \cdot l < i \cdot j$. We may then use this to define the GMIC measure as defined in [125].

$$GMIC(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{Z_{\text{norm}}} \sum_{i \cdot j < B(n)} (C^*(\mathbf{x}, \mathbf{y})_{i,j})^p \right)^{1/p} \tag{3.6}$$

where $p \in [-\infty, \infty]$ is a control parameter for the mean. For our work, we set $p = -1$. Z_{norm} is a normalization constant equal to the number of grid pairs (i, j) where $i \cdot j \leq B(n)$ as described in the original work [125].

There exists a lively debate in the literature as to the significance of the statistical power of these methodologies [107]. We note that we incur additional overhead with regard to computational cost in allocating grids, as well as tuning parameters such as $B(n)$ in the case of MIC and p in GMIC using these methods.

In our study, we compared the efficacy of using the Pearson correlation coefficient (PCC) as a distance measure for the QUBO formulation. This statistical measure is ubiquitous in science and engineering for measuring linear relationships between variables.

Pearson correlation coefficient is defined as [132]:

$$PCC(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.7}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ are vectors of the observations n, and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ are the sample means of $\mathbf{x}$ and $\mathbf{y}$, respectively.

Information-theoretic measures offer key properties that may be advantageous for

feature selection over correlation-based approaches. Unlike PCC, which assumes linear dependence between variables, mutual information and its variants (MIC, GMIC) provide a model-free approach to quantifying statistical dependence without assumptions about data distribution or linearity. Although computationally intensive, the grid construction strategies employed by MIC and GMIC naturally handle mixed discrete-continuous variable spaces. For feature subset selection formulated as combinatorial optimization, these properties may more easily identify informative feature combinations, which may not be captured by linear methods assuming linear dependence.

3.2.2 QUBO Formulation

Given that feature selection is NP-Hard, we encode it as a QUBO problem, which can be solved using quantum annealing. In order to formulate our problem as a quadratic binary optimization model (QUBO), we first need to calculate a distance measure for each column vector $\mathbf{X}_i$ in $\mathbf{X}$ versus $\mathbf{y}$. In this case, we use the absolute value of the distance measure and negate it as we wish to find a minimum for our optimization problem. These values then become the linear terms along the diagonal of the $\mathbf{Q}$ matrix.

With these linear terms, we encode the *maximal relevancy* portion of our heuristic, although we negate the values as optimization takes the form of finding the minimum of the domain. By taking the absolute value of the distance function, we give greater weight to features that have higher relevance or correlation with the target variable by treating positive and negative correlations equally.

Then we calculate the distance between each pairwise column vector $(\mathbf{X}_i, \mathbf{X}_j)$ indexed at i and j in $\mathbf{X}$. This allows us to formulate the interactions for the quadratic terms for the binary quadratic model, which become values along the upper diagonal of the matrix $\mathbf{Q}$. This encodes the *minimum redundancy* characteristic of our heuristic. We want to find combinations of features that are not correlated or are more distant from each other while retaining relevance to the target variable.

$$\mathbf{Q}_{ij} = \begin{cases} -\text{distance}(\mathbf{X}_i, \mathbf{y}), & \text{if } i = j \\ \text{distance}(\mathbf{X}_i, \mathbf{X}_j), & \text{if } i < j \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

We combine these to obtain our QUBO formulation for our optimization problem. For clarity, we use the $\mathbf{o}$ term here to represent a vector of qubit values. We include

a scaling parameter α that we use to scale the domain of the optimization landscape.

$$E(\mathbf{o}) = \alpha \sum_{i \leq j} \mathbf{o}_i \mathbf{Q}_{ij} \mathbf{o}_j \qquad \mathbf{o} \in \{0, 1\} \tag{3.9}$$

Finally, we impose a penalty term such that the resulting sample from our objective function only has k qubits turned "on", or has a value of 1. We introduce a scaling parameter for the penalty term, λ_s, to enforce this constraint.

$$E(\mathbf{o}) = \alpha \sum_{i \leq j} \mathbf{o}_i \mathbf{Q}_{ij} \mathbf{o}_j + \lambda_s \left(\sum_i \mathbf{o}_i - k \right)^2 \tag{3.10}$$

We then use this formulation to follow an annealing schedule and sample solutions to find a minimum energy E. In our resulting vector $\mathbf{o}$, which is of length m, our constraint ensures that k qubits have a value of 1, and the rest 0. We then use this vector to filter the data set $\mathbf{X}$ to obtain $\mathbf{X_k}$, where the k columns are filtered from the index location where $\mathbf{o}_i = 1$ for $\{o_1, o_2, \ldots, o_m\}$.

3.2.3 Data Sets

For our experiments, we tested our feature selection algorithms on two data sets, one synthetic and one drawn from real-world samples. The first dataset that we used was the Friedman 1 dataset [70]. Here, the data are generated by the function:

$$\mathbf{y} = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \epsilon \tag{3.11}$$

for each row, $x_i \in \mathbf{X}$ where ϵ is some random, normally distributed noise, and the rest of the features are independent and drawn from a uniform distribution in the interval of $[0, 1]$, except for the first five features, which were generated according to Eq. (3.11). We generated 100 instances of this data for our train/test 70/30 percentage split, with a feature size of 50. We specifically tested this data set in order to study the difference in performance using various mutual information-based measures within the QUBO formulation of the optimization problem, since the generating function contains a non-linear term, and one of the reported gains in using these distance measures is in measuring non-linear relationships. Another advantage of experimenting with this data set is that the optimal subset of features is known in advance, and therefore we may determine how accurate the feature selection algorithms are in response to the optimum.

The next data set that we tested was for vehicle price prediction using the open source Auto data set from the UCI machine learning repository [57, 183]. In this data set, we have prices for 205 automobiles, along with other features such as fuel type, engine type, and engine size. We encoded all categorical and nominal features using ordinal encoding, which preserved the attribute size of 26. This data set also included the price of the target variable in the range of $[5118, \ldots, 45400]$.

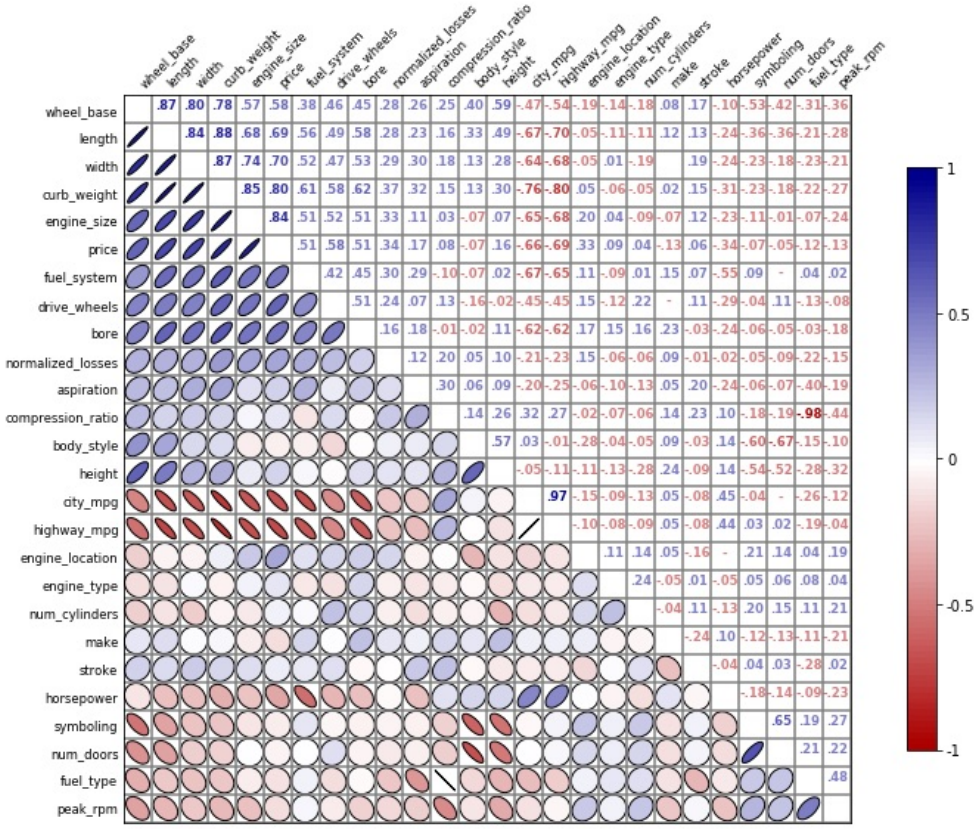

Figure 3.1: Correlation Plot for UCI Auto data, using Pearson Correlation Coefficient as a distance measure between features. Higher values of positive correlation are indicated in dark blue and negative correlation in red. In the feature subset selection problem, we wish to select a subset of features such that the relevancy is maximized with regard to a target variable, in this case, price, and the feature-to-feature correlation amongst independent variables is minimized.

In performing an exploratory analysis of the Auto data, we examined the correlation between features and target variables. In reviewing Fig. 3.1 for the UCI Auto

data, we notice that there are strongly correlated features with the target value of the price. In terms of positive correlation, we see curb weight (PCC= 0.80) and engine size (PCC = 0.84) to have a strong linear relationship, and city and highway miles per gallon (PCC = -0.66, PCC= -0.69) as having a strong negative correlation.

3.2.4 Estimators

During the training/testing regime, we used a 70/30 percentage split between training and test sets. We measured the performance of two different types of predictive model for the underlying regression problem. These models included the following:

Multiple linear regression (LR): We used a multiple linear regression model to estimate predicted values of $\hat{\mathbf{y}}$ [68]. Multiple linear regression is defined as:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon} \tag{3.12}$$

Here, n is the number of samples, $\mathbf{y}$ is the vector of observations, $\mathbf{X}$ is the data matrix, $\mathbf{w}$ is the vector of coefficients (weights) to be estimated, $\boldsymbol{\epsilon}$ is the vector of errors, $\hat{y}_i$ are the predicted values, and y_i are the true observed values. The goal is to find $\mathbf{w}$ that minimizes $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$. Once trained on the data, the predictions for the new test data $\mathbf{X}_{\text{test}}$ are obtained as $\hat{\mathbf{y}}_{\text{test}} = \mathbf{X}_{\text{test}}\mathbf{w}$. We use the term multiple linear regression to clarify that we have two or more independent variables mapping to a target variable.

To evaluate model performance, we use the mean absolute error (MAE), which measures the average absolute difference between predicted and actual values:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{3.13}$$

where n is the number of samples, $\hat{y}_i$ are the predicted values, and y_i are the true observed values. Lower MAE indicates better model fit.

Gradient boosted regression trees (GBR): We looked at a separate regression model, gradient-boosted ensembles of regression trees, to benchmark vs. LR. The gradient-boosted algorithm works by sequentially creating trees from the features of the training data sets, and the structure of the combined trees forms an ensemble that outputs a prediction given new data. For further reading, insightful discussion, and definitions for tree-based learning methods, please see [26]. Performance criteria for our trained models included the mean absolute error (MAE). Model validation was performed using cross-validation on a randomized holdout set of three folds.

In examining the Friedman 1 data set, we designed a performance measure in order to test whether the FS algorithm selected the features of interest from the set, which we knew to be the optimal subset. We call this performance measure *Subset accuracy* which we define using a *hit score* where we take the cardinality of the intersection of the set of the index of selected features $\mathbf{k}_{\text{selected}}$ and set of the index of optimal features $\mathbf{k}_{\text{opt}}$ divided by the cardinality of the optimal feature set. We add length score as a component of subset accuracy as we want to penalize solutions that may have a higher or lower cardinality than the optimal subset.

$$\text{hit score} = \frac{|\mathbf{k}_{\text{selected}} \cap \mathbf{k}_{\text{opt}}|}{|\mathbf{k}_{\text{opt}}|} \tag{3.14}$$

We then calculate a *length score* taking the cardinality of $\mathbf{k}_{\text{selected}}$ minus the cardinality of $\mathbf{k}_{\text{opt}}$ and subtracting this value from 1.

$$\text{length score} = 1 - \frac{|\mathbf{k}_{\text{selected}}| - |\mathbf{k}_{\text{opt}}|}{|\mathbf{k}_{\text{opt}}|} \tag{3.15}$$

We then simply sum the two and divide by two to obtain *Subset accuracy*:

$$\text{subset accuracy} = \frac{(\text{hit score} + \text{length score})}{2} \tag{3.16}$$

3.2.5 Baselines, Filter and Wrapper Methods

For each quantum-assisted feature selection method, we bootstrapped each run with 10 result sets, and for each result, queried the quantum processing unit (QPU) for 10000 shots. We evaluated each set of results and took the best overall result, which we averaged over each cross-validation fold. For each data set, we tested the quantum-assisted feature selection methods and compared them versus the following methods, each of which we cross-validated using 3 folds with replacement:

- All features (All): We initially fit the estimator over all features in the training set of $\mathbf{X}$ and $\mathbf{y}$ in order to understand the test error and establish a baseline for performance criteria.

- Greedy Ranked Method (GR): We devised a simple ranking algorithm, where for each feature column vector $\mathbf{x_k} \in \mathbf{X_k}$, we calculated the $MIC(\mathbf{x_k}, \mathbf{y})$, and then sorted the features based on this relevancy criterion and selected the top k features.

- Recursive Feature Elimination (RFE): We also tested a wrapper style feature selection algorithm, in this case, Recursive Feature Elimination (RFE) as introduced in [83]. We did so in order to benchmark our quantum-assisted method with the RFE wrapper method. For further implementation details please see [83].

3.2.6 Parameters for Experimentation

For each data set and each estimator, we manually tuned the parameters α and λ_s within Eq. (3.8) to values of 1000 and 10. This can be explained by the parameter $\alpha = 1000$ producing a scaling effect, creating a more rugged objective landscape to optimize over. The parameter $\lambda_s = 10$ had the effect of constraining the set of results such that the cardinality of the feature sets was close to k. An interesting result occurred when testing various settings for λ_s; scaling λ_s sometimes led to finding values for k which were improvements in regard to model fit and predictive performance over restricting output to a predefined k. While we did not include optimizing over these hyper-parameters in the scope of this work, we believe that future work could entail investigating this point in further detail.

3.2.7 Implementation Details

The Python code for the experiments was generated using the scikit-learn [160] library, pandas [233], and scipy [224]. Access to the D-Wave quantum annealing machine was obtained using the Leap software and API and the dimod library. An important point to note is that D-Wave provides a set of software tools to embed the binary quadratic model on the QPU. As the QPUs do not have full connectivity, a minor embedding must be created using chains of physical qubits. In this case, groups of physical qubits represent one logical qubit with full connectivity, which are coupled together using a chain strength. Chain strength is an additional parameter that can be optimized during the quantum subroutine. For our experiments, we used the automated minor embedding tools provided by D-Wave for the Advantage 1.1 sampler, which automatically creates minor embeddings on the QPU. The Advantage 1.1 sampler contains 5760 physical qubits, which can be connected directly up to a degree of 15, and was the sampler used for our experiments. We believe that future work could also examine the optimization of chain strength parameters and embeddings, which was beyond the scope of this work at this time, and we refer the interested reader to the D-Wave Ocean documentation for more details [210]. As we are currently in the NISQ era, we do have

limits in regard to the number of features that we can encode for one optimization run. We can also note that while D-Wave provides hybrid tools for problems that are too large to fit on a single QPU, our problem sizes were small enough to fit on current generation QPUs. Some statistical measures were calculated using the minepy library [3].

3.3 Results and Discussion

For our experiments, we looked at performance measures in terms of prediction error (MAE) and feature selection subset size k over cross-validated held out test data for each of the two data sets. For the quantum-assisted routines, we took the average over the cross-validation folds of the best sample from a bootstrapped sample set of 10 samples.

The results for the Friedman 1 data set (in Table 3.1) and the UCI Auto data set (in Table 3.2) show optimal results obtained using PCC as the correlation statistic within the QUBO formulation for the quantum-assisted routine. This is reasonable since both mappings of data sets to their target variables are continuous, and while there may be some nonlinear variables in the data, the results imply that the mappings of inputs to the target variable are mostly linear and efficiently captured under the linear assumptions of PCC.

For the Friedman 1 data set, the quantum-assisted routine using the Pearson correlation coefficient within the QUBO formulation and multiple linear regression as the learning algorithm (**QPCC-LR**) achieved the best performance with the lowest mean absolute error averaged, $MAE = 2.27$, an optimal size for $k = 5$, with a Subset Accuracy score of 0.9 as shown in Table 3.1. Although this predictive algorithm achieved the lowest error, note that the gradient-boosted decision trees using PCC (**QPCC-GBR**) performed comparatively well. The subset accuracy scores ($SA = 0.9$) for both indicate that the feature selection algorithm performed well in selecting the optimal features for prediction, although it did not achieve a perfect score of 1. This could be explained by the feature selection algorithm that excludes features that may be members of the generating subset, but also have low correlation to the target variable due to nonlinearity or noise.

For the UCI Auto data set the best score was obtained using the quantum-assisted routine using the Pearson correlation coefficient within the QUBO formulation and using gradient-boosted regression trees as a learning algorithm (**QPCC-GBR**) achieving the lowest $MAE = 1471$ and size for $k = 18$ as shown in Table 3.2. Although the

Friedman 1 Data Set			
FS method	*MAE*	k	*SA*
QMI-LR	2.58 ± 0.12	5	0.9
QMIC-LR	2.42 ± 0.22	5	0.8
QGMIC-LR	2.63 ± 0.11	5	0.9
QPCC-LR	$\mathbf{2.27 \pm 0.12}$	**5**	**0.9**
All-LR	3.461 ± 0.44	26	-
GR-LR	2.73 ± 0.19	12	0.4
RFE-LR	3.22 ± 0.40	12	0.4
QMI-GBR	3.24 ± 0.39	5	0.59
QMIC-GBR	2.92 ± 0.55	5	0.8
QGMIC-GBR	2.99 ± 0.26	5	0.8
QPCC-GBR	2.42 ± 0.30	5	0.9
All-GBR	2.96 ± 0.42	26	-
GR-GBR	2.93 ± 0.52	12	0.4
RFE-GBR	2.98 ± 0.39	12	0.4

Table 3.1: Table of Results for Friedman 1 Data Set. We include the Subset Accuracy measure (SA) here to understand the ratio of how many optimal features were selected out of the optimal subset $\mathbf{k}_{\text{opt}}$. Note we did not include SA results for All features, as no features were selected for these estimators. In the figure above, we see all of the quantum-assisted routines achieving approximately similar performance, with gains for QPCC-LR (Quantum-Assisted Pearson correlation coefficient-Multiple Linear Regression) with the lowest averaged (12 trials) $\mathbf{MAE} = \mathbf{2.27}$ (p-value=0.00001) and optimal size for $k = 5$, with a Subset Accuracy score of 0.9. Values shown as mean +/- standard deviation over trials.

other distance measures did not outperform PCC, some were comparable, as in the case of MIC in application to LR for the Friedman data set, or MI for GBR on the Auto Data set. This suggests that these measures may be as robust as PCC for certain types of applications and data sets, and it may be that there are other data sets with which these statistics outperform PCC and emphasize the *minimum redundancy* component of the MRMR heuristic and may capture more information from nonlinear relationships. It is also interesting to note that GBR performed better than LR on the UCI Auto data set. This could be attributed to the tabular nature of the data and the tree structure of the algorithm for GBR. This could be attributed to the GBR finding better partitions or splits in the training data than the LR for price estimation. In general, the quantum-assisted feature selection methods achieved results that outperformed our baselines of all features, greedy selection, and recursive feature selection on both data sets.

UCI Auto Data Set		
FS method	**MAE**	k
QMI-LR	2669.1 ± 417	19
QMIC-LR	2684.8 ± 484	20
QGMIC-LR	2669 ± 482	20
QPCC-LR	2622.3 ± 463	18
All-LR	2690.5 ± 460	26
GR-LR	2980.3 ± 445	12
RFE-LR	3355.7 ± 527	12
QPCC-GBR	$\mathbf{1471 \pm 135.6}$	**18**
QMI-GBR	1491.5 ± 142	18
QGMIC-GBR	1491.5 ± 181	19
QMIC-GBR	1536.2 ± 168	20
All-GBR	1546.2 ± 154	26
GR-GBR	1707.2 ± 168	12
RFE-GBR	1678.3 ± 143	12

Table 3.2: Table of Results for UCI Auto Data set. For this experiment, in the table above, we see the best score for the quantum-assisted routine using the Pearson correlation coefficient, with a gradient-boosted decision tree learning algorithm obtaining the lowest $MAE = 1471$ (p-value = 0.001, 12 trials) and size for $k = 18$. Values shown as mean +/- standard deviation over trials.

3.4 Conclusion

In conclusion, we show that by leveraging quantum-assisted routines within machine learning training and testing regimes, we achieve solutions that beat our defined benchmarks with regard to model performance on cross-validated data. We also show that the quantum-assisted feature selection routines are able to filter subsets of data with high subset accuracy. We also uncovered that quantum-assisted routines may show the additional benefit of discovering values of k, given some slight tuning of α and λ_s, which show performance improvements. Finally, we show that we can increase model performance while reducing the input dimensionality of the data to the predictive models.

Future work could investigate automatically optimizing α and λ_s using Bayesian optimization or another methodology. This investigation could also include optimizing the chain strengths of the minor embeddings from the binary quadratic model to the QPU on the quantum annealer.

While we limit the scope of this work to focus on using the quantum annealer as the device for our quantum-assisted routine, this problem could be formulated to

run as an input problem Hamiltonian for a variational quantum algorithm on a gate model chip. We also note that in this work we compute distance measures classically, and there could be some value in further investigation of encoding data onto a gate model QPU and computing distance on the QPU as a step before running a quantum approximate optimization algorithm, to get solution samples of subsets of features. Future work could also explore this point in more detail and investigate the limits of feature selection problem sizes that can currently be fit on NISQ-era chips.

3.5 Summary

In this chapter we investigated solutions to the feature subset selection problem using quantum annealing as part of the selection heuristic in response to RQ1 and RQ2. As many AutoML and data science processes identify feature selection and engineering as a crucial step in building machine learning pipelines [62], this is relevant to our research direction. The work in this chapter highlights how quantum computation may be used as a component in the preprocessing steps of building machine learning pipelines as part of an AutoML process.