3-2020

Algorithm Selection Framework: A Holistic Approach to the Algorithm Selection Problem

Marc W. Chalé

Algorithm Selection Framework: A Holistic Approach to the Algorithm Selection Problem

THESIS

Marc W. Chalé, Captain, USAF

AFIT-ENS-MS-20-M-137

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-20-M-137

AFIT/ENS ALGORITHM SELECTION FRAMEWORK: A HOLISTIC APPROACH TO THE ALGORITHM SELECTION PROBLEM

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Marc W. Chalé, BS, MS

Captain, USAF

March 2020

AFIT-ENS-MS-20-M-137

AFIT/ENS ALGORITHM SELECTION FRAMEWORK:

A HOLISTIC APPROACH TO THE ALGORITHM SELECTION PROBLEM

THESIS

Marc W. Chalé, BS, MS
Captain, USAF

Committee Membership:

Dr. J. D. Weir
Chairman

Maj N. D. Bastian, PhD
Member

AFIT-ENS-MS-20-M-137

Abstract

A holistic approach to the algorithm selection problem is presented. The "algorithm selection framework" uses a combination of user input and meta-data to streamline the algorithm selection for any data analysis task. The framework removes the conjecture of the common trial and error strategy and generates a preference ranked list of recommended analysis techniques. The framework is performed on nine analysis problems. Each of the recommended analysis techniques are implemented on the corresponding data sets. Algorithm performance is assessed using the primary metric of recall and the secondary metric of run time. In six of the problems, the recall of the top ranked recommendation is considered excellent with at least 95 percent of the best observed recall; the average of this metric is 79 percent due to two poorly performing recommendations. The top recommendation is Pareto efficient for three of the problems. The framework measures well against an a-priori set of criteria. The framework provides value by filtering the candidate of analytic techniques and, often, selecting a high performing technique as the top ranked recommendation. The user input and meta-data used by the framework contain information with high potential for effective algorithm selection. Future work should optimize the recommendation logic and expand the scope of techniques for other types of analysis problems. Further, the results of this proposed study should be leveraged in order to better understand the behavior of meta-learning models.

Acknowledgements

My graduate studies at AFIT has been often gratifying and rarely easy. I thank my instructors and committee members for pushing me in the classroom and mentoring me always. I thank my family and friends for fostering an invaluable network of support. It has been a pleasure to collaborate with Lt Clarence Williams, Capt Brandon Hufstetler and Professor Mark Gallagher. Thank you to everyone who has helped me be my best.

Marc W. Chalé

Contents

Page

List of Figures

List of Tables

AFIT/ENS ALGORITHM SELECTION FRAMEWORK:
A HOLISTIC APPROACH TO THE ALGORITHM SELECTION PROBLEM

I. Introduction of the Problem

1.1 Introduction to Operations Research

Operations research (OR) emerged during World War II as the British military tasked scientists to develop a disciplined approach to problem solving. The modern definition of OR is the science of determining the best decision under a constrained system in order to optimize a goal. OR projects typically incorporate mathematical modelling, a quantitative representation of a real-world system [1].

1.2 Rise of Meta-models

There are three overarching approaches to developing mathematical models: *physics based, data-driven*, and a *hybrid*. Physics based models are used when the underlying nature of the system is well understood. They require well refined parameter settings in order for the model to be useful and they may be computationally expensive to execute. The hybrid approach to modelling requires some system expertise to properly employ, however, it also leverages system data to formulate the model. Finally, data driven models are produced solely from system data without regard to system knowledge. Data driven models are also known as meta-models because they are a higher abstraction of the relationship between systems input and response [2]. An overview of predominant meta-models is presented in Chapter 2.

Learning algorithms may be used to formulate a meta-model. Selection of the best

learning algorithm, including hyper-parameters, for a particular problem instance is a difficult and time consuming task [3]. [4] has confirmed conclusions of [5] and [6] that meta-models' performance varies among problem types and problem instances. [7] uses *The Extended Bayesian Formalism* to show that given a set of learning algorithms and problems, each algorithm will outperform the others for some (equally sized) subset of problems. This phenomena has driven researchers to a trial-and-error strategy of identifying the best meta-model for a given problem. The preferred meta-model is selected by comparison of model performance metrics such as accuracy [2]. Unfortunately, the computational run time and human investment required to select a learning algorithm by trial-and-error is generally prohibitive of finding the optimal choice.

1.3 Problem Statement

Cui et al. successfully implemented a meta-learning approach within Rice's framework [8] for "The Algorithm Selection Problem." This paper explores an alternate yet related approach to the algorithm selection problem. Within, an approach is presented that builds on Rice's framework by employing *rules of thumb*, inspired either by literature or developed independently. The research problem is to create an algorithm selection technique for the human analysts that also develops the theoretical intuition for meta-learners by characterizing the problem and referencing a taxonomy of analysis techniques. A primary goal of this paper is to explore the nature of tangible recommendation systems in order to understand black box recommendation systems such as [2]. Further, the paper will identify whether components of the new system can be combined with meta-learners to automatically provide better recommendations.

1.4 Overview of Contents

Chapter 2 of this thesis provides a review of previous work in meta-learning and introduces the applicable meta-models and their performance metrics. Chapter 3 outlines the methodology used to demonstrate the metrics in the meta-learning recommendation framework. The criteria for an acceptable solution is discussed in Chapter 3. Chapter 4 presents the experimental results and Chapter 5 draws conclusions from the research and suggests areas for future work.

II. Literature Review

2.1 Machine Learning

[9] presented a breakthrough to the artificial intelligence community in 1957 by publishing a mathematical model of a neuron. This model, called the perceptron, could be trained to detect patterns in data, in turn automating decisions. In 1969, [10] published influential findings that machine learning methods, notably perceptrons, were incapable of performing complex classification tasks. This news discouraged advancements in the artificial intelligence field until 1986 when [11] succeeded to show the excellent performance of backwards propagating neural networks. The introduction of the backwards propagating neural network marks the beginning of the modern era in machine learning. Today, machine learning algorithms are used to perform a variety of real world problems ranging from speech recognition [12] to military search and rescue [13]. The United States Department of Defense recognizes the military applications of machine learning. In fact, the 2018 National Defense Strategy calls for accelerated modernization of *advanced autonomous systems*, to include artificial intelligence and machine learning in order to achieve a competitive military advantage over adversaries [14].

2.2 The Taxonomy of Analysis Techniques

Many analysts in industry and academia have offered taxonomies to categorize and describe analysis techniques. These products communicate the capabilities and limitations of techniques and group them by a common trait. Taxonomies in the literature vary by size, format, intended audience, and purpose. Two existing taxonomies are referenced within due to their wide scope, high level of refinement, and possible application to the algorithm selection problem. [15] shows that the algorithm

recommendation system is closely linked to the taxonomy.

Field Guide to Anlaysis.

This taxonomy is also very comprehensive, addressing techniques related to classical statistics, statistical learning, machine learning, simulation, optimization, and operations research. The techniques are categorized by the types of problems they solve. [15] demonstrated that a thorough framework relies on comprehensive taxonomy.

[15] presents *Learning* techniques as one of three classes within the universe of data analytics. The highest level of discrimination within the class of learning algorithms is the *category of analysis.* Learning analytics are broken into three categories: *regression, clustering, and classification* [15]. Regression algorithms assign a continuous numerical response to each input data point. Clustering and classification algorithms assign a class membership to each data point. Clustering techniques follow an unsupervised learning style and classification follows a supervised learning style [16].

Learning techniques may be categorized into three *learning styles: unsupervised*, *supervised* and *semi-supervised*. Unsupervised learning models are used when no prior information of class membership is available. The supervised learning approach utilizes a training data set in which all observations are labeled with membership. Semi-supervised learning models are ideal when only some observations contain labels. These models yield more accurate results than unsupervised methods [15].

Offline, reinforcement, and *online* are the three training styles. An offline training style describes methods for which all training is performed in one training event. Alternatively, online models are trained additively in subsequent training events each of which update the model. Although an online model is deployed once, it may improve over time as it gains experience such as feedback on its prior performance.

Reinforcement learning is a special case of online learning. This training style adapts to features in its environment continuously. The model learns to respond to achieve long term goals by responding to changes in the environment. Advancements in deep learning has allowed reinforcement learning to impact dynamic optimization problems such as day trading and navigation [15].

McGarigal.

[17] offers a taxonomy which discriminates techniques by the form data is input and output from the model. According to theory, the form of the model output indicates the type of problems the technique can effectively solve. Accordingly, the key to algorithm selection lies in understanding capabilities of each algorithm in respect to problem characteristics, analysis objectives, data compatibility, data sampling, and the underlying mathematical structure of the model. This taxonomy seeks to provide such information. The taxonomy however was limited to classical statistics and statistical learning techniques.

2.3 Algorithm Selection Frameworks

Booz Allen Hamilton.

[15] asserts that algorithm selection is an art and not a mechanical "repeatable process." Further, each analysis problem contains "hidden dependencies or constraints" that require human judgment to mitigate. The process leverages a fractal analytical model to decompose features of the overarching analysis problem into smaller, more tangible analysis problems. The factors of the fractal analytical model are *data, goal*, and *action*. Practitioners apply the fractal decomposition process until a specific analytic technique is identified. The process relies on analyst judgment to select analytical techniques. The fractal decomposition model is supplemented by five guiding

factors: "compound analytic goals that create natural segmentation natural orderings of analytic goals, data types that dictate processing activities, requirements for human-in-the-loop feedback, need to combine multiple data sources." [15]

Big Data Sources.

[18] provides an algorithm selection framework that emphasizes data compatibility. Increasingly, data is drawn from non-conventional sources such as urban sensors, websites, and mobile applications. In fact 95 percent of big data is unstructured and is obtained in various volumes, velocities, varieties, and veracities. [18] provides guidance on how to analyze such data. Fist, *data governance layer* is a construct that describes physical, legal, and ethical ability to obtain and use data. Next, the *data analysis layer* provides guidance to store, integrate, pre-process, and analyze data as well as publish the results. Finally, the *persistence layer* describes the need to maintain and update the data over a long time horizon.

INFORMS Body of Knowledge.

[19] affirms that algorithm selection is crucial for effective analysis, however, the source sidesteps providing any specific strategy. The major contribution from the Body of Knowledge is linking *categories of analysis* to characteristics of a problem. Techniques within the category *descriptive analysis* explore patterns and trends of historical data. These techniques answers the question "what happened?" with tools such as summary reports, visualizations, and models. Predictive techniques anticipate trends in the future. These techniques address "what could happen?" with statistical methods and data mining methods including machine learning. The most sophisticated models, albeit insightful analysis, belong to the category *prescriptive analysis.* These techniques identify ways to change actions and improve operational

outcomes [19]. Evidently, techniques fall within categories of analysis and problems can be characterized by these categories.

2.4 Classification Techniques

Classification is a supervised machine learning task which seeks to associate input data to its true class, when the set of all classes is known a priori. The general procedure for constructing a classification model includes two step: building the classifier model from training data, and evaluating the model with test data. Only once the model is shown to perform adequately in the test data set should it be used to make new predictions.

The construction of the classifier is sometimes referred to as the learning step. The training set for the learning step consists of a database of ordered tuples, referred to as X, and label attribute, A, associated with each tuple. The label attribute, A, is a nominal variable which dictates the true class of each tuple in X [16].

Support Vector Machine.

Support Vector Machine (SVM) is a supervised learning classification technique proposed by [20] in 1992. Since its publication, SVM has become extremely popular for its outstanding performance classifying records in comparison to more computationally costly methods such as neural networks [21].

The SVM model is trained to fit a separating hyperplane to distinguish observations training data by class membership. The hyperplane serves as a decision boundary to predict the class of unobserved data points. SVM decision boundaries are resistant to training bias because the loss function, which drives separating hyperplane, minimizes classification error and maximizes the buffer between points in each class [20].

Any labelled data sets which are not linearly separable can be mapped to a higher dimension where linear separation is possible using a Kernel function. Proper selection of the Kernel ensures a model that classifies well, but is not overly complex and therefore biased. The SVM problem which incorporates a kernel transformation can be solved very efficiently using Lagrangian optimization[21].

Multi-class categorization, in which observations may belong to one of n nominal classes, and multi-label categorization in which an entity can simultaneously belong to multiple classes, are both extensions to SVM of the base SVM. [21]

K-Nearest Neighbor.

K-nearest neighbor was first demonstrated in the early 1950's however it wasn't until the 1960s that advancements in computing technology allowed the technique to be employed for pattern recognition on larger data sets. The method compares an unlabeled tuples to labelled tuples in the training set according to a distance metric such as euclidean distance. The unlabeled tuple is assigned the most frequently encountered label among the k nearest labelled tuples according to the metric. Nominal attributes must be converted to numerical values via one hot encoding. The value of k is generally set to one and attractively increased until desirable classification performance is achieved. Therefore, training a K-nearest neighbor model is computationally intensive. Some algorithms train on sub-samples of the available data to save time. K-nearest neighbor is also robust to missing data by making assumptions about the distance for missing attributes [16]. According to [16], Equation 1 provides the n-dimensional euclidean distance between tuple $\mathbf{X_1}$ and $\mathbf{X_2}$ as

$$dist(\mathbf{X}_1,\ \mathbf{X}_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}. \tag{1}$$

The Naïve Bayes algorithm.

The Naïve Bayes Classifier algorithm warrants special attention due to its high performance which rivals neural networks and decision trees in some applications. The Naïve Bayes learner is designed to determine the best hypothesis h from a space H hypotheses given the observed data D. In the context of classification h is a hypothesized class and D is a training set. The algorithm searches all hypotheses in H for the hypotheses with the greatest value $P(h|D)$ which is known as the maximum a posteriori [12].

The Naïve Bayes Classifier models the likelihood of each hypothesis under all observed attribute settings. For each unobserved tuple, the model predicts the most likely target $v \in V$ associated with an unlabeled tuple with attributes $\langle a_1, a_2, ...a_n \rangle$. The target v corresponding to the greatest likelihood for a record is the predicted class [12].

The probability of each target $h \in H$ is easily calculated for a given data set. It is impractical, however, to solve for $P(a_1, a_2, ...a_n|h_j)$ because most training sets do not contain sufficient instances of $a_1, a_2, ...a_n$ to drive an estimate of $P(a_1, a_2, ...a_n|h_j)$ with good confidence. Therefore, it is necessary to make the naïve assumption that values of the attributes are conditionally independent of for the given class target, v. This statement of conditional independence is expressed mathematically in Equation 2.

$$P(a_1, a_2, ...a_n|v_j) = \prod_i P(a_i|v_j) \tag{2}$$

Equation 3 shows that the Naive Bayes classifier determines the hypothesized target class of maximum likelihood given a set of attributes.

$$h_{NB} = argmax_{h_j \in V} P(h_j)P(a_i|h_j) \tag{3}$$

[12]

Decision Trees.

Decision trees are a supervised classification technique that partitions training records into branches based on the attributes which contain the most predictive information. The objective during training is to partition observations until each partition is pure, that is it contains observations of only one class. The structure of the partitioning is known as a decision tree, and the decision tree generated from training data is used to classify unlabeled observations. The technique is popular because it performs well in many applications and the model is intuitive to interpret. According to [16], Decision tree methods were developed in parallel by two groups in the late 1970's into the 1980's, namely [22] and [23]. Each algorithm follows a similar classification strategy but differs in the attribute selection heuristic [16]. Decision trees are commonly applied in fields such as biology, engineering, chemistry, finance, and medical research, [24] and may be performed using both categorical and continuous attributes [16]. The two predominant strategies for attribute selection are the Gini Criterion which was introduced in [23] and Information Gain which was introduced in [22]. The Gini Criterion branches on the attribute which minimizes impurity in the resulting partitions [23] and the Information Gain heuristic splits on the attribute which maximizes the information gained at a branch. Information is a quantification of the tree's failure to provide pure classifications [22]. [16] provides analysis of the strengths and weaknesses for each attribute selection method.

Multi-layer Perceptron.

There are many varieties of artificial neural networks. The rudimentary feed forward perception model was introduced in [9]. The features common to a feed

forward ANN include layered sets of nodes (also called neurons) connected by arcs. The connected structure of the network is known as its topography. Nodes that define the topography are organized into layers, or neurodes. Data features from the input data are processed through the input layer. Each node in the input layer performs a function, known as the activation function, on the inputs. The weighted outputs of this layer are the input of the nodes of the next neurode. Equation 4 demonstrates that the input value of node j is the weighted sum of the outputs of all connected nodes in the preceding neurode i, plus a bias scalar, θ, which is a free variable [16].

$$I_j = \sum_i w_{ij} \cdot O_i + \theta_j \tag{4}$$

The backwards propagating neural network, introduced in 1986 by [11], is a more advanced model that communicates error information to predecessor neurodes. The backwards propagating ANN overcomes learning limitations of the feed forward ANN. ANN models are robust to errors in training data. Although training times may be high for large data sets, trained models are evaluated very quickly. Often, it is impossible, albeit unnecessary, for a human to interpret meaning from the weights assigned to a trained ANN. Therefore, ANN is considered a black box classification technique [12].

2.5 Regression Techniques

Regression is a statistical method that estimates a relationship between predictor variables and response variables. Regression estimates each predictor variable's contribution to a response by generating a coefficient for each predictor. Some regression estimate interaction coefficients, that is multiple variable's combined contribution to the response. The best regression models are shown to follow the theoretical true relationship between the predictor variables and response, sometimes called the physics

model. Even the true model fails to predict noise, or random deviations from the model. Regardless, regression is an effective approach for predicting event outcome and for many fields of science and engineering. Under the strong assumption of causal relationship, regression can be used for controlling engineering systems [25].

Support Vector Regression.

[26] introduced support vector regression (SVR) as an extension of SVM methods. The SVR algorithm produces a function $F(x)$ which models $G(x)$, the true relationship between predictor data point x and the response, y.

$F(X, \hat{w})$ is a reparameterization of F where$\hat{w}$ is the normal vector defining the optimal hyperplane, and x is once again a point in the input space.

Equation 5 is common choice of F which takes the form

$$F_1(X, \hat{w}) = z^t \cdot \hat{w}, \tag{5}$$

where z^t is defined in Equation 6 by

$$z^t = [x_1^2, ..., x_d^2, ..., x_i x_j, ...x_{d-1} x_d, x_1, ..., x_d, 1], \tag{6}$$

for all $d \in \{1...N\}$. The number of features in z is defined in Equation 7 as the combinatorial expression in

$$f = \sum_{i=d-1}^{p+d-1} C_{d-1}^i \tag{7}$$

[27] proposes an alternative form of F in Equation as 8 where

$$F_2(X, \hat{w}) = \sum_{j=1}^{N} (\alpha_i * -\alpha_i *)(v_i^t x + 1)^p + b, \tag{8}$$

such that α_i*, α_i* and b are free variables and p indicates the order of the polynomial model.

The goal is to select $\hat{w}$, the optimal normal vector such that $F(x, \hat{w})$ is the best possible estimate of $G(x)$. To do so, a loss function, is introduced, for example, L such that $L[\cdot] = [\cdot]^2$. In this case, Equation 9 shows the primal objective function of SVR is defined as the quadratic

$$U \sum_{j=1}^{N} L[y_j - F(v_j, \hat{w})] + ||\hat{w}||^2, \tag{9}$$

where U is a regularizer constant v_i are support vectors, ie.input data points which fall outside of the acceptable buffer region. y_i are the corresponding observed values to $G(x)$ including noise. The regularizer constant is a tunable hyperparameter that sets the relative importance of reducing prediction error verse generalizing the function. Geometrically, the first term is the sum of squares between predicted response and observed value of the response; the second term is the distance between the F and the boundary of the acceptable buffer region.

Linear Regression.

Simple linear regression is a technique used to model one predictor variable's relationship to a single response variable. y denotes the predicted response variable, β_0 denotes the estimated intercept, β_1 denotes the estimated regression coefficient for the estimator variable, and x denotes the input data point. β_0 and β_1 are commonly estimated using the method of least squares [25]. The simple linear regression formula, Equation 10, predicts the response value as a function of the predictor variable on the surface of a line

$$y = \beta_0 + \beta_1 x + \varepsilon \tag{10}$$

The multiple regression model with k predictor variables and interaction terms can also be generated using the method of least squares. This type of model, seen in Equation 11 follows the same form as the simple regression model with the additional predictor variable terms.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \varepsilon \tag{11}$$

The multiple linear regression formula predicts the response value as a function of all predictor variables on the surface of a hyperplane. The addition of the interaction term in Equation 12 yields a more complex multiple regression model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_{12} x_1 x_2 + ... + \varepsilon. \tag{12}$$

A regression models should always be built according to the principle of parsimony, that is with the minimum order function that represents the data and contextual knowledge of the system. A simple model is always preferred to a complex model. In the case of a curvilinear relationship between predictor and response variables, a transformation may mitigate the need for increasing the order of the model [25].

2.6 Clustering Techniques

Clustering is the unsupervised approach to machine learning which partitions observed data points into groups, or clusters, based on perceived similarities. In contrast to classification techniques, the clustering approach to machine learning can be performed on unlabeled data. It has been described as "automatic classification" because analyst does not require domain knowledge or knowledge of class characteristics or grouping logic. Some algorithms do require the number of desired clusters as an input. Clustering is commonly used to identify outlier data points. Clustering has been

applied successfully to management science, information security, medicine, and web search [16].

Distance metrics such as euclidean distance quantify the similarity of points in a data set. Clustering algorithms use four methods, *partitioning, hierarchical, density based,* and *grid based*, to assign tuples to the correct cluster. See chapter 10.1 of [16] for detailed comparison of these methods.

Clustering techniques generally perform well on a range of data types including nominal, ordinal, and binary. Recent research has shown potential applications with less conventional data such as "graphs, sequences, images, and documents". Some clustering techniques such as partitioning generate only spherical decision boundaries while others such as hierarchical are more robust to an arbitrary decision surface. Although clustering techniques typically handle high dimensional data, analysts must use caution to avoid biased results due to inclusion of immaterial factors [16].

K-Means.

The K-means clustering algorithm is the most fundamental algorithm among partitioning clustering methods. Initially, all observations $\in D$ are randomly assigned to k clusters. The centroid, c_i, of each cluster, C_i, is typically calculated as either the m-dimensional mean or medioid of all points in C_i. The algorithm seeks to minimize the within-cluster variation for all $C_i \in D$. This is an NP-hard problem that can be approximately solved using a greedy heuristic. At each iteration, observations $\mathbf{p}$ are reassigned to the nearest centroid, and the centroid location is updated. This process is repeated until a stopping criteria is met [16].

2.7 Data Reduction Techniques

The *Curse of dimensionality* describes a data set which contains too many variables to be interpretable or useful for analysis [28]. Data reduction can be used during pre-processing to reduce the time complexity of algorithms that increase rapidly with the number of variables [28]. Data reduction methods are used to reduce the number of variables in the data while retaining the integrity of the information represented. If any information is lost, the data reduction is described as lossy. If the original data can be regenerated fully from the compressed data, the compression is known as lossless [16]. Data reduction also improves results when the number of variables is nearly as great as the number of observations or when there is high correlation among variables [28]. Principal component analysis is among the most widely used ordination technique and is discussed below.

Principal Component Analysis.

Principle component analysis (PCA) is a multivariate technique which describes the variance present in the original variables in a new set of orthogonal components. The algorithm performs a change of basis operation which identifies the basis which contains the most information in the fewest dimensions. It works especially well when there is multicollinearity present in the original data. The ideal outcome of PCA is that a small subset of the new components, the principal components, contain enough information that the remainder of components can be omitted from the model. The result of PCA can be used as an exploratory tool to determine underlying trends in a system. For example, an economist may use PCA to determine the key sectors that serve as an indicator of a greater economic behavior [28]. The steps to perform PCA are outlined in Equation 13.

$$C = X^T X \tag{13}$$

If the variables of the input data are presented in drastically different scales, the correlation matrix should be used instead of covariance matrix. The eigenvecters of C are generated by eigenvalue decomposition. The eigenvectors v_i are arranged into matrix A in order of descending eigenvalues, λ_i, where the eigenvalue quantifies the variance explained in each new component i [13]. The proportion of variance explained by each of k total components is expressed in Equation 14 as

$$\frac{\lambda_{\hat{i}}}{\sum_{i=1}^{k} \lambda_i}. \tag{14}$$

There are several acceptable methods to determine the number of retained principal components. Commonly, the number of principal components chosen is the fewest that accounts for a predetermined proportion of retained variance. Alternatively, only components that account for a greater than average amount of variance are retained. Another common method is to plot the eigenvalues in descending order and selecting a cutoff point at the elbow of the *scree plot* [28].

2.8 Performance Metrics for Machine Learning Techniques

Performance metrics are used to assess the quality of a classifying model and are the basis of algorithm selection [16]. This section addresses how prominent performance metrics are applied in the field of machine learning. These metrics are the fundamental to assessing the performance of algorithm selection.

Normalized Root Square Mean Error.

Normalized route mean square error is used to quantify the similarity between the observed response $y_i \in Y$ and their predicted response $\hat{y}_i$ [2]. Route mean square error takes the same form as population standard deviation, a metric of the spread in a data set; however, according to [29], RSME is distinguished in that the reference point of comparison is an observed value and not the set's mean. [2] proposes the normalized root mean square error, shown in Equation 15, which is scaled by the range of values in Y.

$$NRMSE = \sqrt{\frac{1}{N} \frac{\sum_{i=1}^{N}}{(y_{max} - y_{min})}} \tag{15}$$

Area Under the Curve - Receiver Operator Curve.

Area under the curve - Receiver Operator Characteristic (AUC-ROC) was originally developed to convey the tradeoff between a true positive and a false positive detection rate in radar systems during World War II. Today it is commonly used to describe the performance of classification models [16]. Calculating AUC-ROC requires that the model outputs the probability that each tuple belongs to each class. Therefore this metric is naturally suitable for decision trees and Naïve Bayes Classifiers, though it can be extended to other machine learning techniques. [30] has shown precedent by successfully using AUC-ROC as a performance metric in meta-learners [16].

To calculate AUC-ROC, tuples from the test data are sorted in descending order of likelihood membership to positive class. Next, the true positive rate, also known as sensitivity, is calculated as $TP = \frac{TP}{P}$. The TP rate (TPR) is plotted as a function of false positive rate $FP = \frac{FP}{N}$. The curve begins at the vertical axis, TPR, where

the horizontal axis is set to zero. A point is plotted at the origin. Iterating down the sorted list, a point is plotted for each tuple. If the tuple is correctly labeled, the points appear above the previous. If the tuple is incorrectly labeled, its corresponding plot point appears to the right of the previous point [16]. [31] notes that since there is an inverse relationship between TP and FP, the ROC plot indicates the nature of the tradeoff. The ideal model would show a large TP at every value of FP resulting in an AUC-ROC score of nearly 1.0. Conceptually, the AUC-ROC score is probability that a classifier will rank a randomly chosen positive observation higher than a randomly chosen negative observation. A score greater than 0.5 indicates that the model classifies better than a random classifier. Identifying a desirable AUC-ROC score is ultimately a business decision based on judgement [31] .

Algorithm Run Time.

An algorithm is defined as a process of discrete steps used to solve a specific problem. Algorithms typically perform operations on an input data and output a solution. Complexity classes are metrics that quantify the computational performance of an algorithm. Space complexity refers to the amount of memory the algorithm requires to store data throughout each step. Space complexity is not a major concern in many cases due to the large memory capacities in modern computers. Time complexity, the duration required to complete a computing task is, however, a consideration for algorithm selection [32]. Time complexity is typically defined within the construct of a theoretical random access machine (RAM). The RAM counts every primitive operation performed within an algorithm such as addition, multiplication, assignment, ect. Running time, is the number of primitive operations required to perform all tasks in an algorithm for a specific problem, and is closely related to time complexity. It is assumed that running time for an algorithm of n primitive operations is cn, where c

is a constant related to a computer's rate of performing primitive operations. There is variability however in running time of algorithms on equally sized problems. For instance, a sorting algorithm may require fewer subroutines for a data set that is nearly sorted than for a data set that is completely random. Therefore, $\Omega(n)$, pronounced Big-Oh notation, is used to describe the worst case computational complexity of an algorithm on any problem with size n. Computational complexity is the standard proxy for runtime when comparing algorithm performance [32].

Accuracy.

Accuracy, sometimes referred to as recognition rate, provides the data analyst with the overall proportion of correct classifications by a model. For a binary classification problem we define the two classes as positive and negative. Accuracy is defined in Equation 16 as the sum of true positive and true negative classifications divided by the total number of observations which were classified [16].

$$A = \frac{TP + TN}{P + N} \tag{16}$$

Alternatively, this information can be reported as the error rate where $errorrate = 1 - accuracy$. The accuracy metric does not account for potential imbalance of positive and negative tuples in the test data [16]. Imagine 99% of tuples in the test set are dogs, and 1% are cats. The model may correctly classify all dogs, and incorrectly classify all cats but still reflect 99% accuracy. Optimizing a learning model via a loss function related to accuracy may incentivize a base learning algorithm to develop a bias toward the class of higher instances. In this case the balanced accuracy metric is preferred because it is centered about each class [33]. [33] shows that balanced accuracy can move beyond point estimates and provide confidence intervals of classification performance in the population of data sets. The formula for the commonly

used point estimate is shown in Equation 17.

$$A_{adj} = \frac{1}{2}(A_P + A_N) \tag{17}$$

Recall.

The recall, also known as sensitivity, of a classifier is the true positive rate of detection for the positive class. The formula for sensitivity is shown in Equation 18.

$$recall = \frac{TP}{TP + FN} \tag{18}$$

Unlike accuracy, which reflects the classifying performance for all classes, recall reports performance for only one class which may be of particular importance. For example, recall is of more importance than accuracy and balanced accuracy in a model that predicts cancer because failure to identify a true positive tuple results in an undiagnosed cancer patient. Similarly, specificity quantifies the rate of true negatives [16]. It is defined by Equation 19.

$$specificity = \frac{TN}{TN + FN} \tag{19}$$

Model dimensionality.

Incorporating superfluous complexity to an analytical model detriments its statistical legitimacy and makes it difficult to interpret. A model with excessive complexity *will* properly represent the training data but *lacks* the statistical properties to predict the response of unobserved [21]. [25] instructs that a [regression] model should always be built to the least complexity that accurately represents the system. Note that a regression model of *n-1* polynomial terms can always be fit through n data points. Although such a model will exhibit low error on the training data set, it does

not represent the underlying nature of the system and does not serve as an effective predictor [25].

Overfitting.

Overfitting is the phenomena characterized by an analytical model that is well suited to describe the training data but is unable to perform well on data not observed in the training set. Empirical risk is defined as the optimal value of the loss function regularized by the number of observations in the set. Structural risk is defined as the difference in the empirical risk yielded by the training data set and a test set. A model is said to exhibit *overfitting* if the structural risk is very high. This indicates the model lacks the underlying statistical nature of the data will perform poorly as a predictor [21].

There are two causes of overfitting. First, a model with excessive complexity tends to describe data well but is unable to effectively predict using *any* unseen data-regardless of statistical similarities between training data set and validation data set. Second, a model trained on data which lacks the required statistical information is unable to predict with unseen data [21].

2.9 Meta Learning

Background of Meta-Learning.

Rice's algorithm selection framework was presented in 1976 [8]. The framework is performed by employing all algorithms under consideration on all problems in a problem set. One or more performance metrics are chosen, and the performance of each algorithm on each problem is reported. Upon completion of the process, the preferred algorithm for each problem is taken as the one with the best performance metrics [8]. [34] presents a modern depiction of Rice's framework as phase 1 in figure

1.

The classic approach of learning algorithms is known as base learning. That is a machine learning algorithm which builds a data driven model for a specific application [35]. Meta-learning however, is an approach introduced by [36] which algorithms learn on the learning process itself. A meta-learning algorithm extracts meta-features $f(x) \in$ space F from a problem $x \in$ problem space P. The meta-model is trained to recommend the best known base learning algorithm $a \in A$ to solve x. Additional works such as [37] and [38] further contributed to the theory of meta-learning recommendation systems[35].

In 2014, [39] proposes the concept of applying meta-learning to Rice's model. It was not until 2016, however, that [2] implemented the concept. Figure 1 demonstrates that Cui et al. trained a meta-learning model to correlate problem features to algorithm performance and that the trained model could be used to recommend the algorithm for unobserved problems within Rice's framework. The meta-learner correctly recommended the best algorithm in 91 percent of test problems. Further, it demonstrated that time to perform algorithm selection could be reduced from minutes to seconds compared to trial and error techniques [2].

Recent Work In Meta-Learning.

[40] proposed *landmarking* as a novel training strategy for metal learners. In lieu of training via feature extraction, landmarking determines the geometrical location of a problem instance in the space of all possible problems by testing each problem instance's performance against a baseline learning algorithm. The meta-learner recommends a learning algorithm to be paired with each actual problem instance. Initial findings indicate some success, and may warrant future research [40].

Ler et al. has explored the use of clustering analysis to produce meta-features repre-

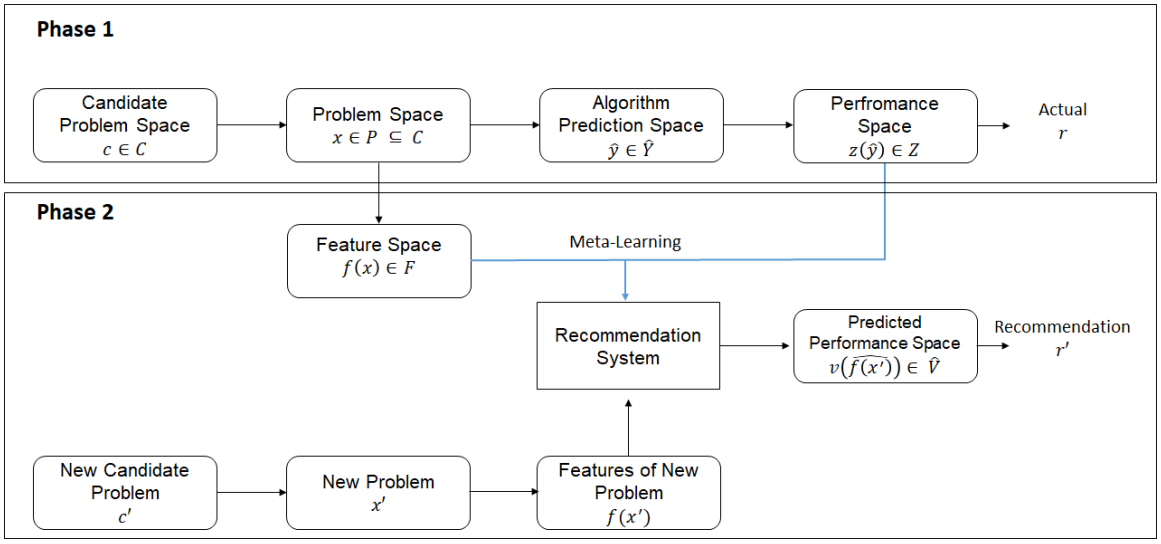

Figure 1. The meta-learner adaptation of Rice's framework [34]

sentative of data complexity. Ler shows that *purity ratio, size distance*, and *volume distance* are representative of data complexity and that data complexity is correlated to base learner performance [30]. Similar results were achieved by [41].

Evaluation of Recommendation System Performance.

Analysis is performed to rate the quality of the recommendation system based on the performance metrics listed above. Evaluation of recommendation system performance is a major driver for improving recommendation systems such as [2]. The following techniques address this topic.

Spearman's Rank Correlation Coefficient.

Spearman's Rank Correlation Coefficient, Equation 20, is a measure of similarity between two ranking schemes for members of a set [2]. [2] utilizes the Spearman's Rank Correlation Coefficient to measure agreement between the meta-learner's predicted ranking of algorithms by performance and the observed algorithm ranking

by performance for a problem. d_i is defined as the difference in assigned rank for algorithm A_i and N is the number of elements i in I.

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{20}$$

Perfect rank matching between two ranking schemes produces a SRC of 1, while two opposite ranking schemes produces a SRC of -1, and two uncorrelated ranking schemes are characterized by a SRC of 0. In the case that no ties are present, Spearman's coefficient produces an equivalent value to the widely used Pearson's correlation coefficient when calculated for the ranking scheme, but Spearman's is preferred due to computational simplicity [42] and relaxed statistical assumptions [43]. Likewise, Spearman's coefficient is preferred in the case of moderate ties or many ties because the difference in the two statistics is negligible [42].

Hypothesis testing is used to determine if the calculated correlation is statistically significant. The null hypothesis states the paired random variates are mutually independent, ie. the correlation is 0; the alternative hypothesis explicitly states the type of dependency. "either (a) there is a tendency for the larger values of X to be paired with the larger values of Y, or (b) there is a tendency for the smaller values X to be paired with the larger values of Y [42]." The test is performed by selecting ρ as the test statistic. The critical values for testing the null hypothesis are presented in the table of quantile of the Spearman's statistic as a function of n and p, the quantile of the standard normal variable. That is to say the table is a measure of how extreme the statistic is at a specified confidence. For a two-tailed test, reject the null hypothesis if the test statistic is either greater than the corresponding critical value for $p = 1 - \alpha/2$, or less than the symmetrical critical value for $p = \alpha/2$.

Hit Ratio.

The hit ratio, proposed by [35], is defined as the percentage of trials a meta-learner correctly recommends the best performing algorithm for a problem. The metric is akin to the true positive rate, or recall of a classifying base learner. The metric is shown in Equation 21

$$R_{hit} = \frac{\sum_{i=1}^{n} h_i}{n}, \tag{21}$$

where n is the number of problems. h_i is 1 if the recommendation is correct and 0 otherwise.

III. Methodology

This section communicates the methodology to solve the problem statement: "create an algorithm selection technique for human analysts that also develops the theoretical intuition for meta-learners." Notably, the criteria was identified prior to designing the solution and prior to performing any experimentation. This methodology includes a criteria and an outline of the experimental strategy.

3.1 Criteria

A criteria was defined to include all of the desirable traits of a solution to the problem statement. Alternative solutions are referred to as frameworks of the algorithm selection problem. Therefore, the following traits define the criteria for the algorithm selection framework under development:

Framework.

- Leverages a taxonomy. The framework must discriminate machine learning techniques by both their intended applications and their internal mechanics. The framework therefore must interface with a comprehensive taxonomy containing all the algorithms under consideration.
- Maps to specific recommendation(s). The framework should produce a rank ordered list of the specific algorithms appropriate for each problem, not just a set of acceptable choices or a statement of guidance. The framework is not, however, required to set hyper-parameters or provide tuning guidance. Therefor it is acceptable to consider default parameter settings for all algorithms.
- Recommended algorithm performs well. The framework must recommend algorithms that are applicable to the intended task. Furthermore, the recommended

algorithms are expected to produce high quality results according to an appropriate performance metric such as recall, MSE, or accuracy. An *excellent* recommendation is defined as one which produces results within 5 percent of the best observed performance. A *good* recommendation is within 10 percent; a *satisfactory* recommendation is within 20 percent, and a *poor* recommendation is not within percent of the best observed performance.

- Rigorous and repeatable process. The framework should remove subjectivity from the algorithm selection process. The framework should produce the same recommendation each time it is implemented on a particular data set. The recommendation should be made based on known information, not the practitioner's intuition.

- Fast implementation. The time required for an analyst to perform the the algorithm selection should be negligible in the scope of the project. Specifically, the recommendation time must be an order of magnitude shorter in duration than the analysis algorithm.

- Aids a human analyst. The framework should be easy for a human analyst to implement without any ancillary training. It should mitigate the conventional trial and error procedure for algorithm selection.

- Supports meta-learning problem. The algorithm selection framework must employ a logical decision process in the most efficient way possible. Studying this logic will provide insight onto the logic of a black box meta-learner may be using to recommend an algorithm. Therefore, if advantageous, aspects of the framework can be incorporated into a meta-learner hybrid model.

Taxonomy.

- Distinguishes techniques by application. The taxonomy must describe algorithms by application as one aspect of providing a quick and intuitive reference for algorithm selection.

- Distinguishes techniques by mechanism. The taxonomy must characterize algorithms by their mathematical model to provide information for predicting algorithm performance. The analyst may assess the compatibility of the mathematical model with aspects of the problem characterization.

- Distinguishes techniques by training style. The taxonomy shall identify the compatible training styles for each technique to inform whether the training data may be provided in a single event, or in successive events.

- Addresses data characteristics. A characterization of the feasible, and ideal data features that are compatible for each technique will aid the alignment of techniques to problems. Proper alignment will facilitate good performance metrics.

- Hierarchical structure. A hierarchically structured taxonomy is necessary to clearly organize the taxonomy, to encapsulate the necessary information, and to allow growth over time.

- Comprehensive. The taxonomy must include all commonly used techniques in order be a useful reference to the analyst.

- Expandable. The taxonomy needs to grow as new techniques emerge and as new technique attributes are deemed necessary to characterize.

Table 1 compares the strengths and weaknesses of several existing frameworks of analytic against the criteria above. A green colored box indicates the criterion is

fully met; yellow indicates a criterion is partially met; red indicates that a criterion is poorly met or not addressed at all. Note that none of the frameworks provide a sufficiently rigorous and repeatable recommendation; none of the frameworks aid the understanding of meta-learning recommendation systems. The Analytics Body of Knowledge Framework leverages a taxonomy of analytical techniques which leads the analyst to only consider a subset of applicable techniques for each problem. This often results in analysis that properly solves the correct problem, but does not necessarily identify the best performing technique. The Analytics Body of Knowledge Framework did not, however remove subjectivity from the recommendation, and was therefore not repeatable. The framework presented in the Field Guide provides the most guidance for matching a problem to technique. Still, this guidance is largely unspecific, lacking quantitative metrics. Conflicting recommendations could be generated from this guidance depending on its interpretation. Further, the framework from the Field Guide occasionally leads the analyst to techniques which would not be appropriate for the analysis problem.

Table 2 compares two existing taxonomies of analysis techniques. Neither of the alternatives sufficiently categorize techniques by both their application and their mechanism. The taxonomies do properly address learning style. Finally, the taxonomies do not cover a comprehensive scope of all relevant analysis techniques in the universe of analysis.

3.2 Proposed Framework

The proposed framework is derived from discussions regarding how most analysts select a machine learning algorithm for a problem. Evidently, many analysts become comfortable with only a small fraction of the available analysis techniques. They often neglect to consider all appropriate algorithms for a problem. Therefore, the framework

Table 1. Comparison of the frameworks reviewed

	Field Guide	Big Data Sources	Analytics Body of Knowledge
Leverages a taxonomy			
Maps to specific recommendation(s)			
Recommended technique performs well			
Rigorous and repeatable process			
Fast implementation			
Aids a human analyst			
Supports meta-learning problem			

is built to guide the analyst to the correct technique agnostic to any personal bias. This approach follows from the stated criteria. The framework should be implemented within the analysis process in order to identify the appropriate analytical approaches and recommend specific analytical techniques. Figure 2 shows that within the analysis process, four factors are identified which drive the analytical approach and analytical technique selection. They are the input to the algorithm selection framework.

Characterizing the Problem.

The framework is a mechanism to characterize an analysis problem and to determine the algorithms that best matches the problem characterization. The four factors each drive analytical approach selection and analytical technique selection in a different way. The factor *assigned task* pertains to the problem provided by management

Table 2. Comparison of the taxonomies reviewed

	McGarigal	Field Guide
Distinguishes technique by application		
Distinguishes techniques by mechanism		
Distinguishes techniques training style		
Addresses data (problem) constraints		
Hierarchal structure		
Comprehensive		
Expandable over time		

or a decision maker. The analyst must decipher the intent of the assignment from the lexicon of the manager into specific analytical terms, which are listed under the *Task*. This list of terms, called *considerations* is shown in Figure 3 for each factor. The considerations for the factor *data* describe the different formats analysts commonly receive data for analysis problems. The *data* factor is important because it relates to the problem's compatibility with the mathematical mechanics of the analysis technique. Likewise, the considerations for the *resources* factor help the analyst identify which algorithms are compatible with the available resources. Finally, the factor of *analyst skill* characterizes the human analyst's abilities, which also impacts algorithm selection. Education level is used as a coarse proxy for analyst skill level [19]. In reality, work experience and problem solving skills are also relevant considerations but they are not addressed in this framework due to the subjectivity involved in capturing

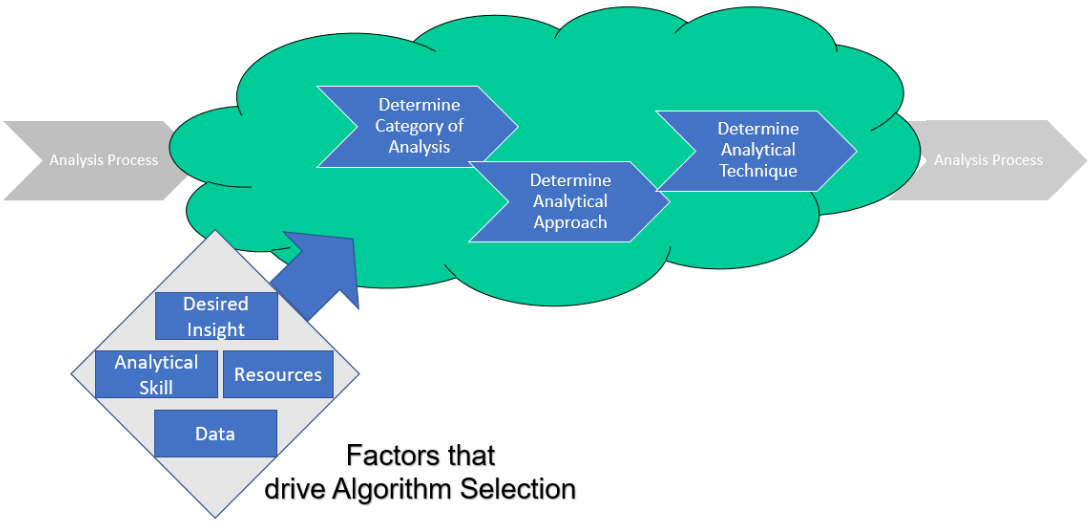

Figure 2. The factors identified in this this research are superimposed with the stages of analysis which they impact, ie. determine analysis approach and determine technique

them. The analyst should refer to Figure 3 to evaluate and record the considerations for each factor prior to beginning step 1.

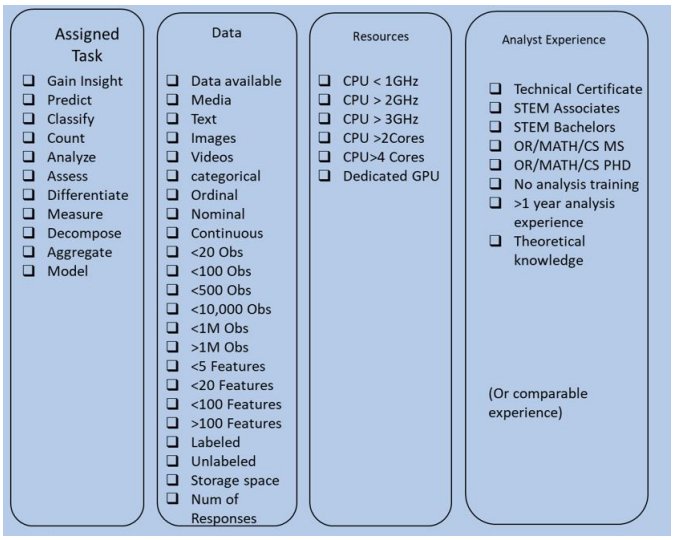

Figure 3. The considerations are shown for each factor which drives analytical approach and analytical technique selection

Step 1: Map Problem to Category and Approach.

Step 1 leverages information from the *problem characterization* to identify the appropriate *analytical approaches*. Each *consideration* selected from the *assigned task* factor maps to one or more *categories of analysis*. The *categories of analysis* describe the general goal of the analysis problem [19]. Each *category of analysis* can be implemented by certain *analytical approaches*. The *analytical approach* a technique class referring to the specific type of response the techniques produce. Therefore, the framework leverages a hierarchical taxonomy that groups techniques grouped by both categories of analysis and analytical approaches. Figure 4 shows the mapping from *assigned task* to *category of analysis*, and the mapping of *category of analysis* to *analytical approach*. An alternate representation is shown in Table 3 where the colored boxes indicate compatibility between the category of analysis and analytical approach.

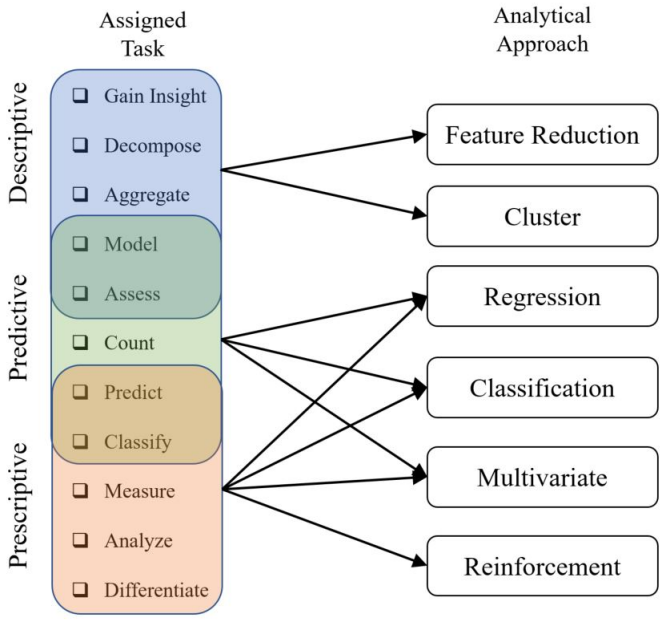

Figure 4. The 11 possible assigned tasks all into one or more of the categories of analysis which are listed on the far left. Each category of analysis maps to one or more analytical approach on the far right.

Table 3. The matrix view of mapping from category of analysis to analytical approach.

	Category of Analysis		
Analytical Approach	Descriptive	Predictive	Prescriptive
Cluster			
Regression			
Classification			
Feature Reduction			
Multivariate			
Reinforcement			

An excerpt of the proposed taxonomy is presented in Figure 5. The taxonomy is built with an object-oriented structure to promote flexibility and expandability. As an example, techniques are shown within the *regression* and *classification analytical approaches*. The text *predictive* and *descriptive* appears at the bottom edge of the regression panel to indicate that regression techniques produce results suitable for either of these two *categories of analysis*. Applicable considerations are listed below each factor on the panel for each technique. Compatible training styles are listed to the right of the technique name. The object oriented structure allows new techniques to be easily added and new attributes to be included as necessary.

Step 2: Score Techniques.

The framework thus far identifies a subset of techniques which are compatible for the problem according to application. Next, the framework leverages the remaining three factors *data, resources* and *experience* to discern aspects of technique com-

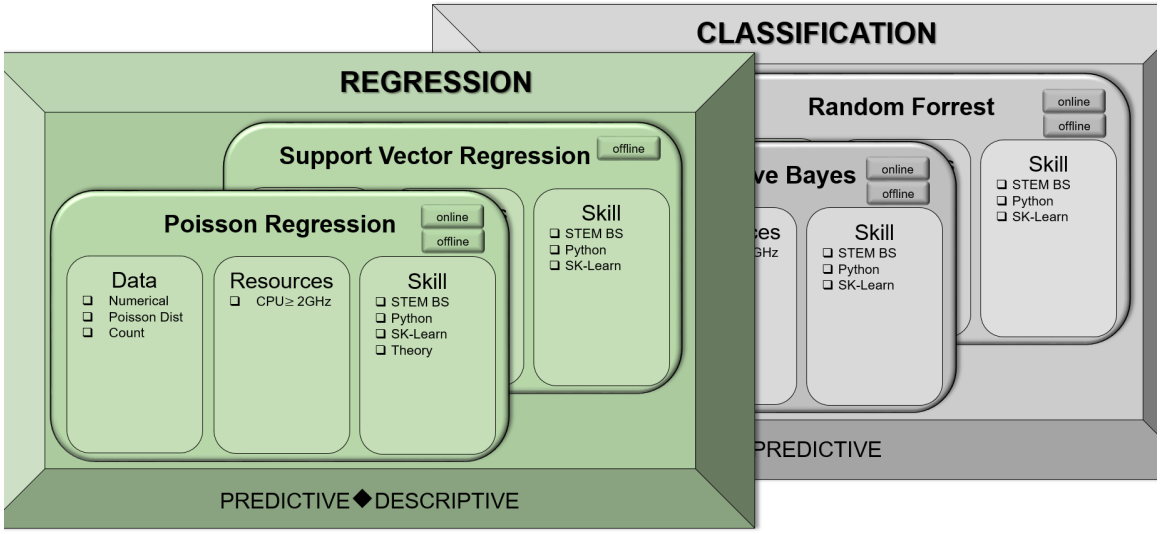

Figure 5. A portion of the proposed taxonomy is hi-lighted to show the structure of the taxonomy

patibility relating to the mechanics of the mathematical model. The techniques are ranked ordered by level of compatibility for each of the remaining three factors. The most preferred technique for each factor is assigned the highest ordinal score and ties are resolved by providing the average score of the tied scores.

Step 3: Rank Recommendations.

The final recommendation score $s^{(j)}$ for each technique j is shown below as the product of the score $s_k^{(j)}$ for each factor k. The weights of the scores for each factor are assumed equivalent for this study. The techniques are then ranked by their final score where the highest number is most preferred.

$$s^{(j)} = \prod_{k=1}^{3} s_k^{(j)} \tag{22}$$

A decision tree is used to assign the ordinal scores for each technique within each factor for data and resources. The decision tree is built from features of the data.

Notably, the features pertaining to data also impact the compatibility of techniques in respect to resources. Therefore, it is justified to use the same decision tree, Figure 6, to adjudicate the scores for both factors. A separate decision tree could be produced for each factor, however, it is not necessary to prove the concept. The ranking logic for analyst skill is omitted from this study due to analysis automation, which is discussed in Chapter 4.

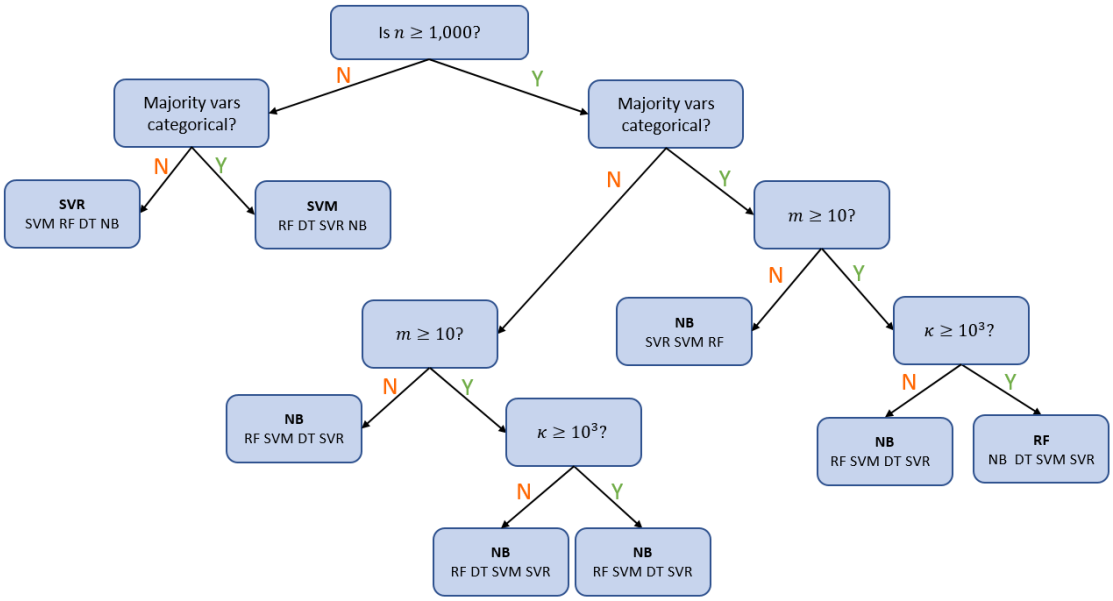

Figure 6. Decision tree used to assign a preference rank for each technique in regards to the data factor

3.3 Data Sets

The experimental process is performed using nine unique data sets. The data sets are pre-processed such that a binary target was placed into the first column. The following list outlines the assigned task for each data set and references the source.

1. Heart: Predict presence of heart disease from 13 predictor variables [44]
2. Framingham: Predict presence of heart disease in the Framingham study from 15 predictor variables [45]
3. Spam: Predict if an email is spam based on six predictor variables [46]
4. Loan: Predict whether a consumer purchases a loan from Thera Bank based on 12 predictor variables [47]
5. PMESII: Predict whether the sumintensityofwar metric surpassed a threshold of five for each country each year. Data is a compilation of AFIT's *Political Military Economic, Social, Information, Infrastructure (PMESII)* data set [48], the *Correlates of War* data set[49], and the *Armed Conflict Data* data set totaling 408 predictor variables
6. Cancer: Predict whether a patient has breast cancer from 30 predictor variables collected in a fine needle aspirate procedure [50]
7. Urinalysis: Predict whether a patient is experiencing formation of calcium oxalate crystal based on six predictor variables [51]
8. Colleges: Data used to predict whether a college is public or private based on 17 predictor variables [52]
9. Election: Data used to predict the electoral vote for each state in the United States based on five predictor variables [53]

Efficiency.

Recall efficiency is presented as a single value to assess the success of a recommendation for the metric of recall. Equation 13 demonstrates that the recall efficiency is calculated as the recall of the top recommended technique divided by the recall of the technique with the best observed recall. A technique obtains a recall efficiency of 100 percent if it generates the best observed recall. Otherwise, the recall efficiency for a technique is greater than 0 percent and less than 100 percent, commensurate to its observed recall.

$$E_R = \frac{R_{bestRec}}{R_{bestobs}} \tag{23}$$

3.4 Software and Packages

This research is implemented entirely within the Spyder 3.1.2 integrated development environment using the Python 3.6.0 kernel. The open source Python library *sci-kit learn* is used to access the machine learning functions *DecisionTreeClassifier, RandomForestClassifier, MultinomialNB* with *MultinomialNB, svm.SVC, SVR, classification report*, and *mean squared error*. Additional functions were used for data pre-processing. The experimental design section will detail how these functions were implemented as classifiers.

Workstation Specifications.

A Dell Precision 5540 mobile workstation was used for all computations and analysis in this study. The workstation ran Windows 10 Enterprise and has an intel i9-9980 CPU running at 4.8GHz

IV. Experimental Results and Analysis

4.1 Experimental Results

The recommendation system is implemented on nine binary classification data sets according to the framework described in Chapter 3. Problem characterization is performed by the research team for each of nine data sets by reading the data set description and adjudicating an assigned task to each data sets. In step 2, all nine data sets are mapped to predictive category of analysis. Therefore, the analytical approaches assigned to each data set are *regression*, *classification*, and *multivariate*. Figure 7 shows that scores of one through five are assigned to the factors *data* and *resources* in accordance with the decision tree shown in Figure 6. All techniques are assigned the same score for *analytical skill*, effectively nullifying the factor. All factors are weighted equally for the final recommendation score. The eligible analytical techniques were ranked from one, highly recommended, to five, least recommended, based on their final recommendation score. Figure 7 reveals that the taxonomy is not comprehensive. The four classification techniques, two multivariate techniques, and two regression techniques used in this study are representative of how the framework is applied to an expandable taxonomy of techniques.

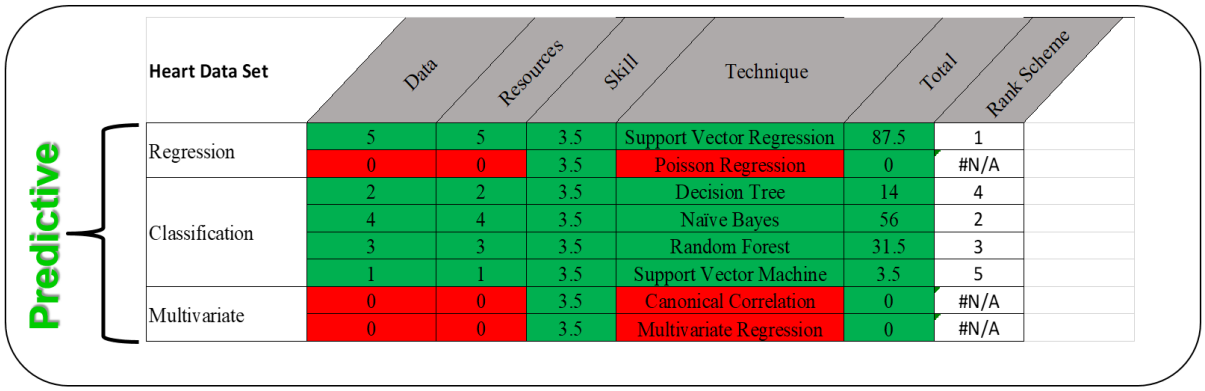

Figure 7. Step 2, scoring, is performed for each factor. Step 3, overall technique ranking, is performed for the Heart data set.

Table 4 reports the framework's recommendations and several analysis metrics. The first line of Table 4 provides the recommended ranking scheme for each data set. The rank is provided for decision tree, random forest, Naïve Bayes, support vector machine, and support vector regression techniques in order, as these were the recommended techniques for each problem. The next line reports the rank scheme for each data set according to the observed recall. The Spearman's coefficient of rank correlation between the two rank schemes is reported in the next line of the table. Eight data sets demonstrate positive correlation and one data set demonstrates a negative correlation; none of these figures are statistically significant using a two tailed hypothesis test and 5 percent significance. The metric of correlation describes the consistency in ranking for *both high* performing *and low* performing techniques. Since in practice, the framework need only implement the top performing technique, the correlation for lower rankings is immaterial. Therefore, the performance of the recommendation system is best understood by assessing the performance of the top recommendation. Accordingly we attribute greater consideration to the True Hit Ratio which conveys that a perfect agreement between *top recommended* and *top performing* techniques is observed for four of nine data sets. Additionally, the "The Good Hit Ratio" conveys a good (or better) top recommendation for seven of nine

data sets. The average recall efficiency for top the recommendation is reported "poor" as 79 percent, though this figure is skewed downward by two extremely low recall efficiencies. The worst recall efficiency is recorded for the Spam data set for which the Naïve Bayes algorithm is the most highly recommended technique and also the worst performing. Interestingly, Naïve Bayes is a Pareto efficient solution for the Spam data set when considering the secondary metric of run time. Three of nine recommendations are Pareto efficient. Finally, run time is reported in Table 4 for the top recommended technique as well as for the technique with the best observed recall.

Table 4. Table of the complete results

	Heart	Fram	Spam	Loan	PMESII	Cancer	Urinalysis	Colleges	Election	†Avg/*Cnt
Recommended Rank Scheme	4 3 2 5 1	3 1 2 4 5	4 2 1 3 5	3 1 2 4 5	3 1 2 4 5	3 1 2 4 5	4 2 1 3 5	3 1 2 4 5	4 2 1 3 5	
Observed Rank Scheme	5 4 3 2 1	2 3 4 1 5	2 1 5 3 4	4 1 1 5 1	2 1 4 3 5	5 4 2 3 1	1 2 5 3 4	4 3 1 5 2	5 4 1 3 1	
Spearman Rank Correlation	0.4000	0.10	-0.10	0.34	0.70	-0.50	-0.30	0.20	0.10	0.10†
Recall of Top Recommended	0.91	0.54	0.05	1.00	0.99	0.96	0.14	0.95	1.00	0.72†
Best Observed Recall	0.91	0.57	0.79	1.00	0.99	1.00	0.71	0.99	1.00	0.89†
True Hit	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	4/9 †
Recall Efficiency of Recommended	100%	93%	6%	100%	100%	96%	20%	96%	100%	79% †
Good Recall Efficiency ($\geq$ 90%)	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	7/9 †
Name of Top Recommended	SVR	RF	NB	RF	RF	RF	NB	RF	NB	RF:5*
Name of Best Observed Recall	SVR	SVM	RF	RF, NB, SVR	RF	SVR	DT	NB	NB SVR	SVR:4*
Run Time of Top Recommended	0.002	0.030	0.001	0.009	0.260	0.012	0.001	0.013	0.001	0.037†
Run Time Best Observed Recall	0.002	0.330	0.025	0.009, 0.001, 0.001	0.263	0.007	0.001	0.001	0.001, 0.0002	0.053†
Pareto Efficient Recall/Run Time	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	3/9 †

Two plots are generated to visually convey the performance of the recommendation system on each data set. The bar plots, Figures 8, 10, and 12, compare mean recall for each of the recommended techniques. The most highly recommended technique is represented with a red bar and all other recommended techniques are shown in blue. The 95 percent two tailed confidence interval for mean recall is represented with whiskers emanating from the top of the bar.

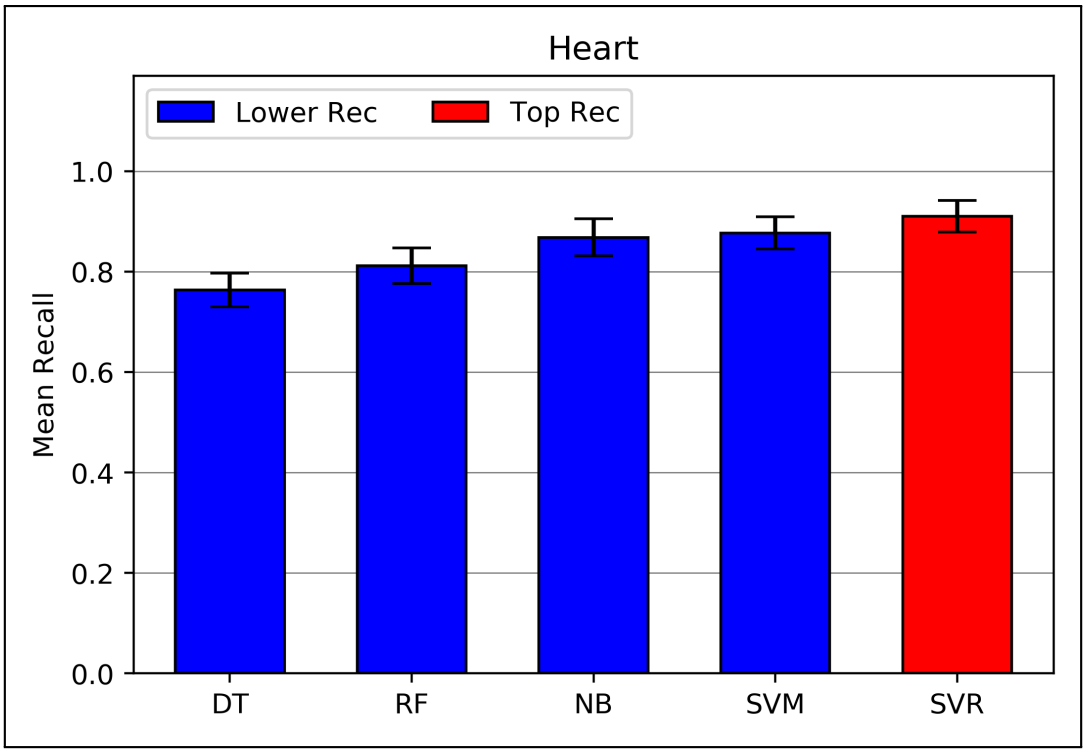

Figure 8. Mean recall for the Heart data set. SVR is a hit.

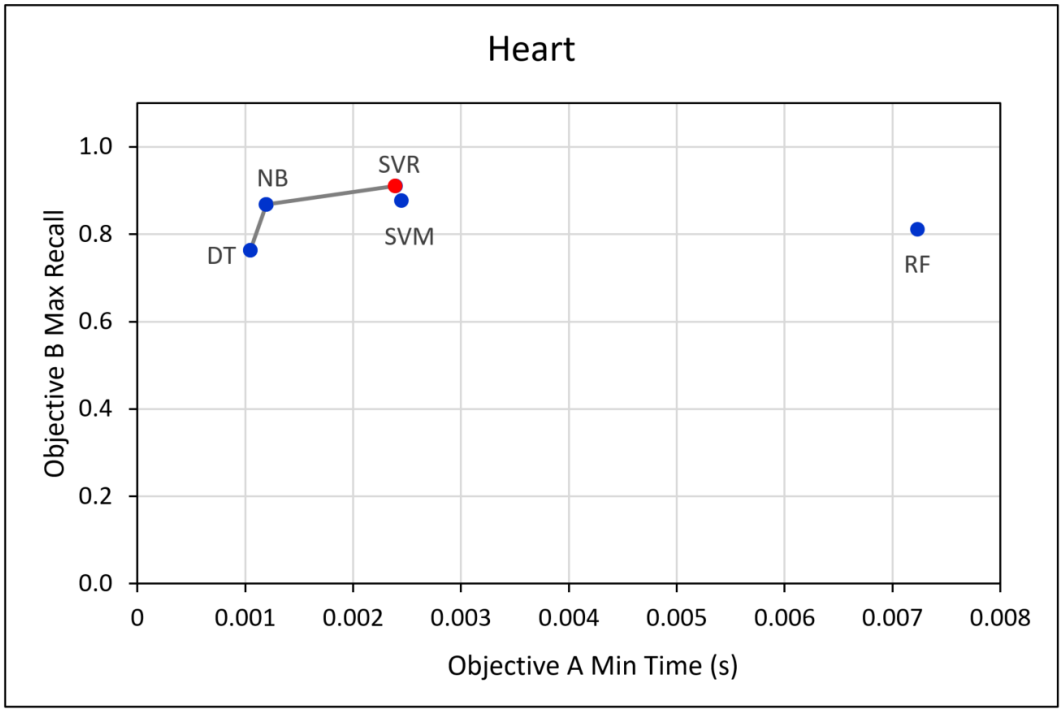

Figure 9. Mean recall and mean run time for the Heart data set. SVR is Pareto efficient because it dominates recall.

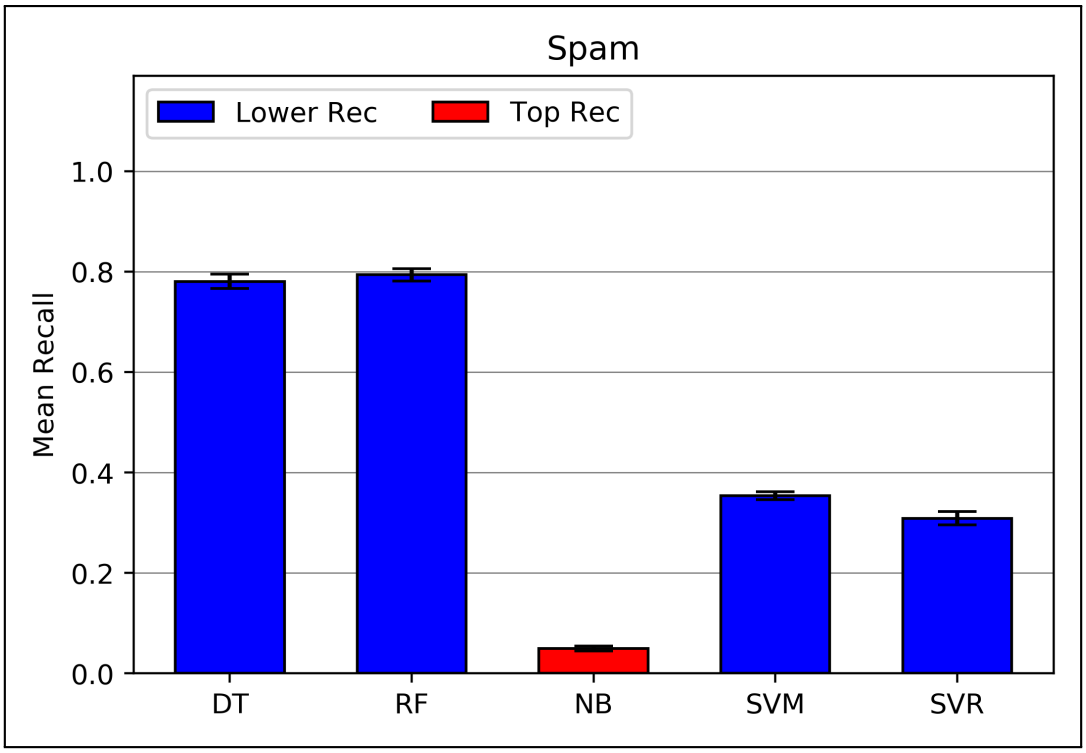

Figure 10. Mean recall for the Spam data set. Naïve Bayes is the top recommendation and produces the worst recall of all recommendations.

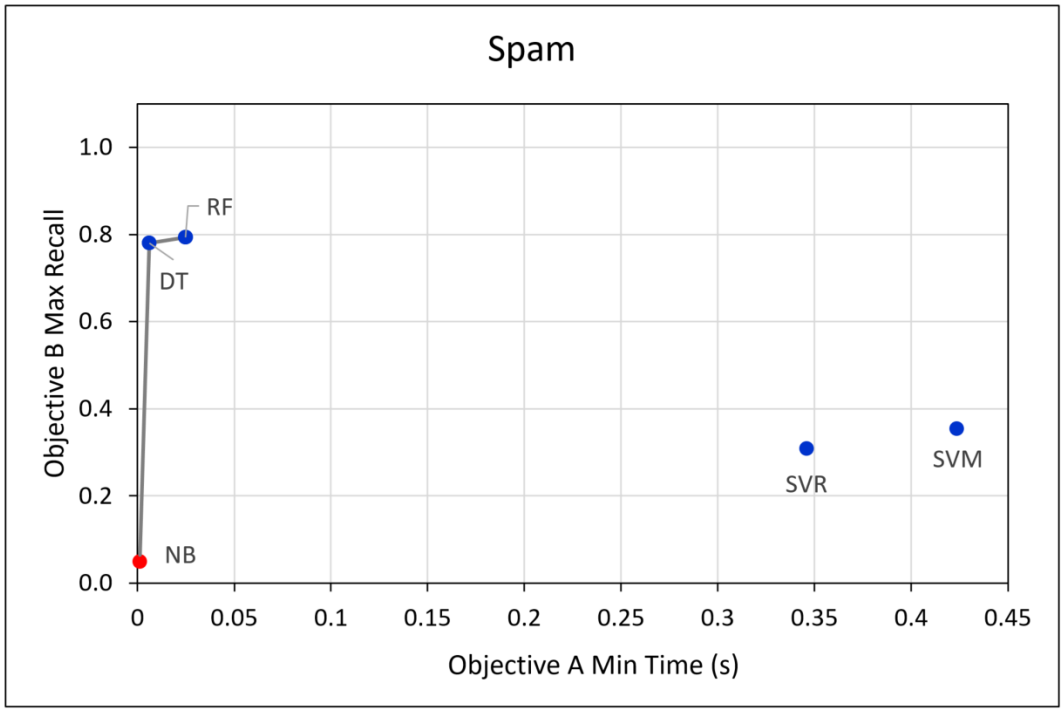

Figure 11. Naïve Bayes is a Pareto efficient solution because it dominates run time.

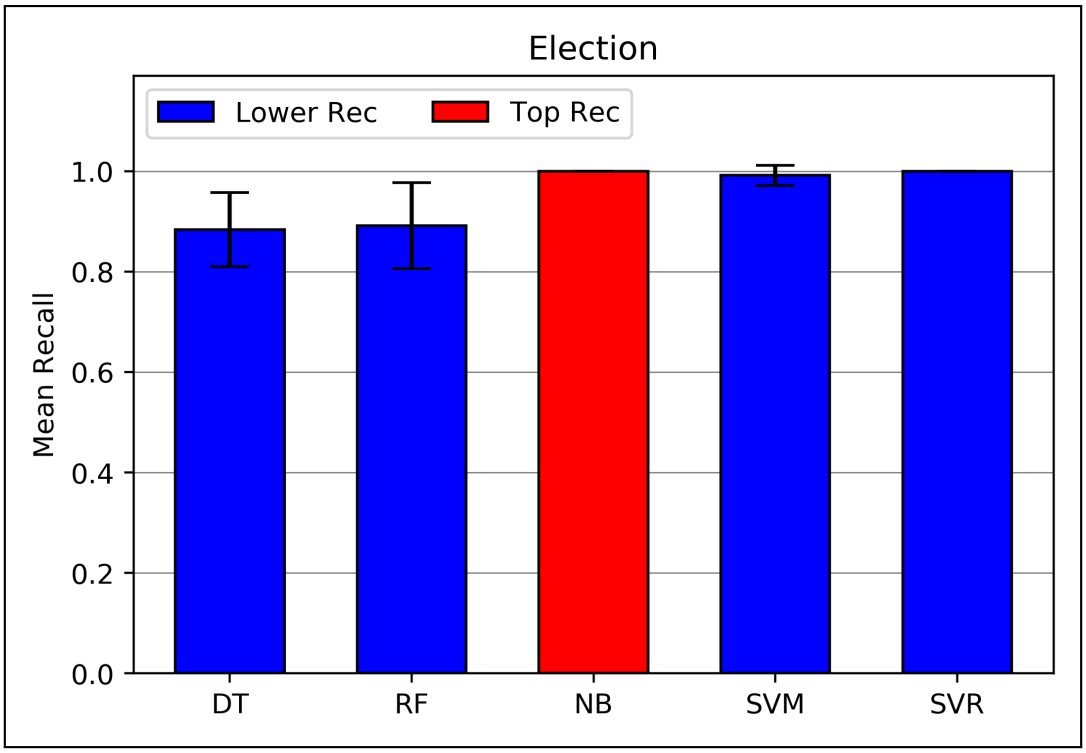

Figure 12. Mean recall for the Election data set. Naïve Bayes is a hit.

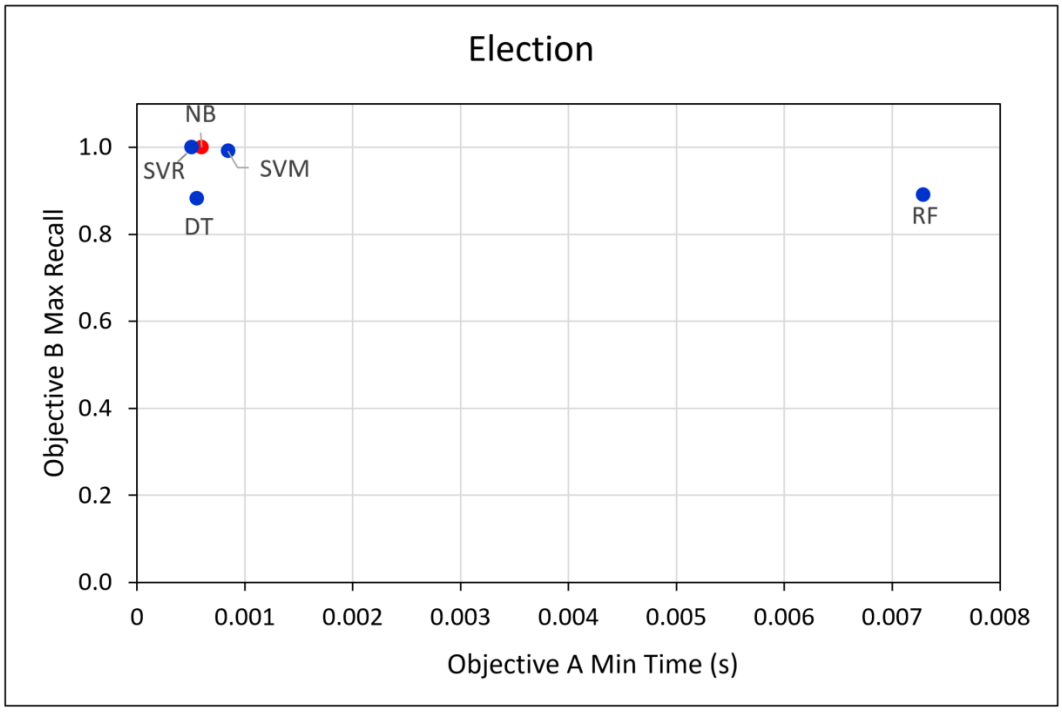

Figure 13. Despite producing the ideal recall and a fast run time, the recommended technique, Naïve Bayes, is not a Pareto efficient solution

The multi-objective plots, Figures 9, 11, and 13, present the primary objective *maximize recall* plotted against the secondary objective *minimize run time*. The most highly recommended technique is identified with a red dot; all other recommended techniques are represented with a blue dot. If a Pareto frontier exists, it is represented as a grey line. Otherwise, one solution is shown that dominates both objectives.

The recall plot and the multi-objective plot of the heart data set are shown in Figure 8 and Figure 9 respectively. This set of results demonstrates an excellent overall outcome from the recommendation framework. In Figure 8 the bar corresponding to SVR is red to indicate it is the top recommended analysis technique. The height indicates SVR generated better recall than all other recommendations. The whiskers at the top of the bar indicate the 95 percent confidence interval about the mean of recorded recall in 20 trials. Figure 9 represents SVR as a red point, also to indicate it as the top recommended technique for the Heart data set. Since no technique outperforms SVR on both objectives, SVR is Pareto efficient. The Pareto frontier is shown in gray, connecting each Pareto efficient recommendation.

The recall plot and the multi-objective plot of the spam data set are shown in Figure 10 and Figure 11 respectively. These results demonstrate that the framework does occasionally provide a poor recommendation. Here, the top recommended technique, Naïve Bayes, yields the worst recall of all recommend techniques with a recall efficiency of just 6 percent. This recommendation is shown to be Pareto efficient in Figure 11 due to its very low run time. Still, it is a bad recommendation.

Naïve Bayes is the top recommendation for 3/9 data sets. In two of these instances, Naïve Bayes performed very poorly with recall efficiencies of 6 percent and 20 percent respectively, which indicates that the ranking logic to recommend Naïve Bayes may not be optimized.

The recall plot and the multi-objective plot of the election data set are shown in

Figure 12 and Figure 13, respectively. For this data set, the framework succeeds in recommending an algorithm, Naïve Bayes, with perfect recall. In fact, three of the recommended analytic techniques exhibited excellent mean recall, and their recall was statistically equivalent at a confidence of 95 percent using a two tail hypothesis test. The 95 percent confidence interval for all five recommended algorithms indicate the mean recall may be excellent. The reader may therefore ask whether the recommendation framework is necessary for this data set. A posteriori it is revealed that three of the five recommendations are excellent. Still, the framework is necessary to systematically identify the five techniques. There is a statistical difference between the mean recall of the top recommendation and the two worst performing recommendations; this difference may be practically significant. Noting that an election model is most useful for several states that are difficult to predict, it would behoove a newspaper editor to call election results using the model that exhibits perfect true positive rate and an imperfect false negative rate. It is always preferred to recommend the *best* performing technique even as several perform generally well.

Interestingly, several data sets which yield hits are not Pareto efficient if multiple techniques reflect perfect recall. This occurs if the recommendation is dominated by other solutions for run time. The effect is observed for the Loan data set and the Election data set in Figures 17 and 13, respectively. In both cases the recall and run time for the recommended technique are not practically different from the dominating technique. Charts depicting the results of all other data sets are included in Appendix B.

4.2 Evaluation of Taxonomy to Criteria

The taxonomy included in this study demonstrates that a recommendation framework benefits from leveraging a taxonomy. Table 5 shows it evaluated generally well

against the taxonomy criteria presented in Chapter 3 and serves the intended purpose within the scope of this research. The taxonomy excels at distinguishing algorithms by intended application. It exhibits a minor shortcoming of distinguishing techniques by mechanism. While the taxonomy does address the spirit of this criteria by characterizing each technique for compatibility with data, skill, and resources of a problem, there is no specific classification of techniques by mechanism. The taxonomy receives full credit for addressing the compatibility of data with each technique through its analysis of data meta-features. The taxonomy does specify applicable training style for each technique. The taxonomy is constructed with a hierarchical relationship between characteristics. An object oriented structure places each technique within an analytic approach, which in turn is mapped to a broader category of analysis. Attributes can be assigned to objects at any level of the hierarchy. The example used in this study is not comprehensive and must be expanded to incorporate all prevalent analysis techniques. Ultimately, the proposed taxonomy serves as a template and proof of concept. Fortunately, the taxonomy can be easily expanded in regards to breadth and depth. Future revisions of the taxonomy should include more techniques and more attributes for each technique. In particular, the taxonomy must be adapted to include techniques other than machine learning algorithms.

Table 5. Comparison of the reviewed taxonomies to the proposed

	McGarigal	Field Guide	Chalé
Distinguishes technique by application			
Distinguishes techniques by mechanism			
Distinguishes techniques training style			
Addresses data (problem) constraints			
Hierarchal structure			
Comprehensive			
Expandable over time			

4.3 Evaluation of the Framework

Analysis of Results.

The novel framework demonstrates several major improvements over existing frameworks. It successfully meets the intent of each criteria except *recommendation performance*. Table 4 shows the framework performs inconsistently across the data sets. On average, the Spearman's coefficient of rank correlation demonstrates a slight positive correlation. The highest level of utility for an algorithm recommendation system is to correctly rank order compatible recommendations by performance. This is a difficult task to optimize. We observe the framework leverages predictive information but the decision logic is not optimized. Therefore, the criterion "performs well" is partially met. The recommendations reflect an average recall efficiency of 79 percent, which is considered poor. Notably, the recommendations for seven of nine data sets have at least good recall efficiencies. Six of nine have excellent recall efficiencies. The framework is beneficial even when it does not produce a hit.

The framework consistently filters techniques that are incompatible with the problem characterization. Further, the framework identifies five viable options, some of which perform excellently.

Evaluation of Framework to Criteria.

Table 6. Evaluation of Framework criteria for three reviewed frameworks and the proposed framework.

	Field Guide	Big Data Sources	Analytics Body of Knowledge	Chalé
Leverages a taxonomy				
Maps to specific recommendation(s)				
Recommended technique performs well				
Rigorous and repeatable process				
Fast implementation				
Aids a human analyst				
Supports meta-learning problem				

V. Conclusion

The proposed framework measures well against the stated criteria. The framework successfully filters inadequate analysis techniques from each problem and recommended good techniques in most cases. Although the framework's rank scheme of recommended techniques is positively correlated with the ranking of best observed techniques, the correlation is low. Fundamentally, the meta-data and user input collected by the framework does contain information capable of consistently predicting the a good analysis technique for a problem. The process of problem characterization fits well into the framework but does require further refining. The decision tree used to generate rank schemes provided intelligible recommendation logic. The factor of *analytical skill* proved to be of no importance due to the automation incorporated into the framework. Future work should use the Gini criterion to optimize the recommendation logic and should expand the scope of techniques into other types of analysis problems. The results of this proposed study should be leveraged in order to better understand the behavior of meta-learning models. Aspects of the recommendation framework, such as technique filtering process, may be incorporated into future meta-learning ventures.

VI. Appendix A

Table 7 lists the AFIT theses referenced in the literature review of this paper.

Table 7. AFIT theses reviewed during research

Thesis Title	**Author**	**Advisor**
Spectral Textile Detection in the VNIR/SWIR Band	A James A. Arneal, Second Lieutenant	Lt Col Jeffrey D Clark, PhD
A Metamodel Recommendation System Using Meta-learning	Megan K. Woods, CTR	Professor Jeffery Weir

VII. Appendix B

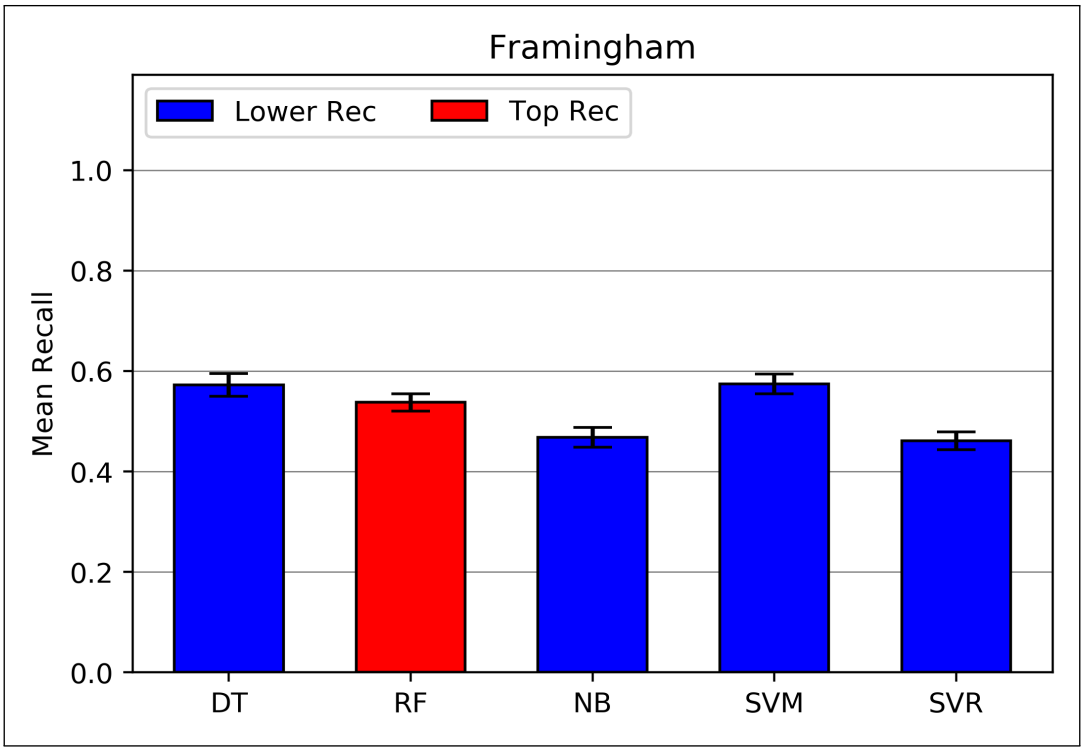

Figure 14. Mean recall for the Framingham data set

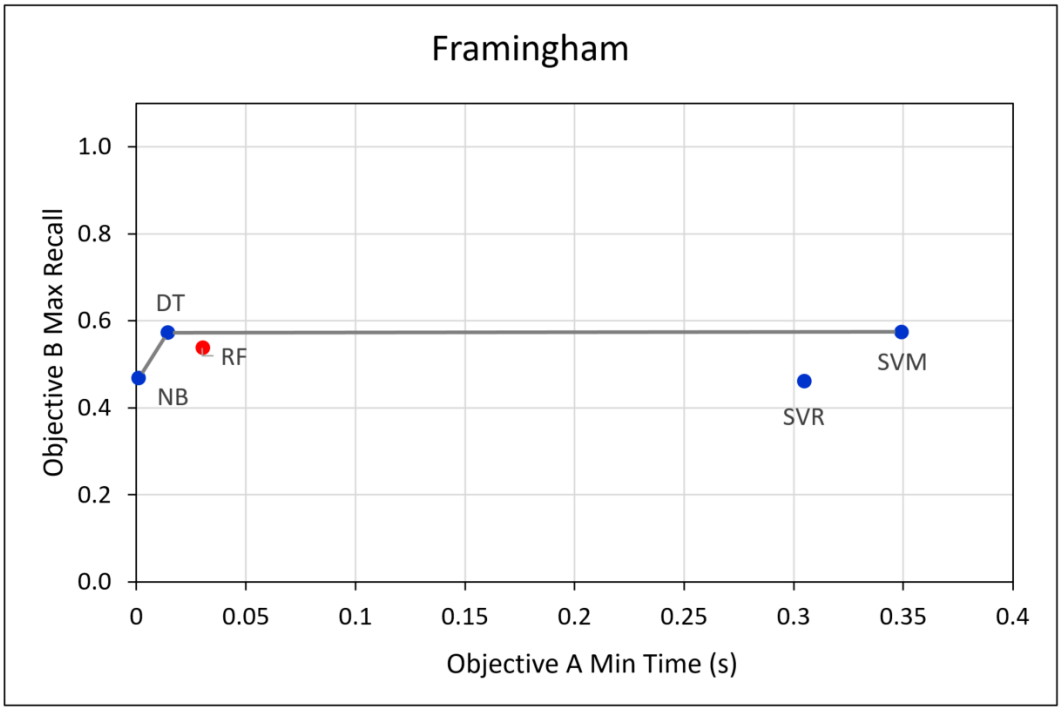

Figure 15. Mean recall and mean run time for the Framingham data set

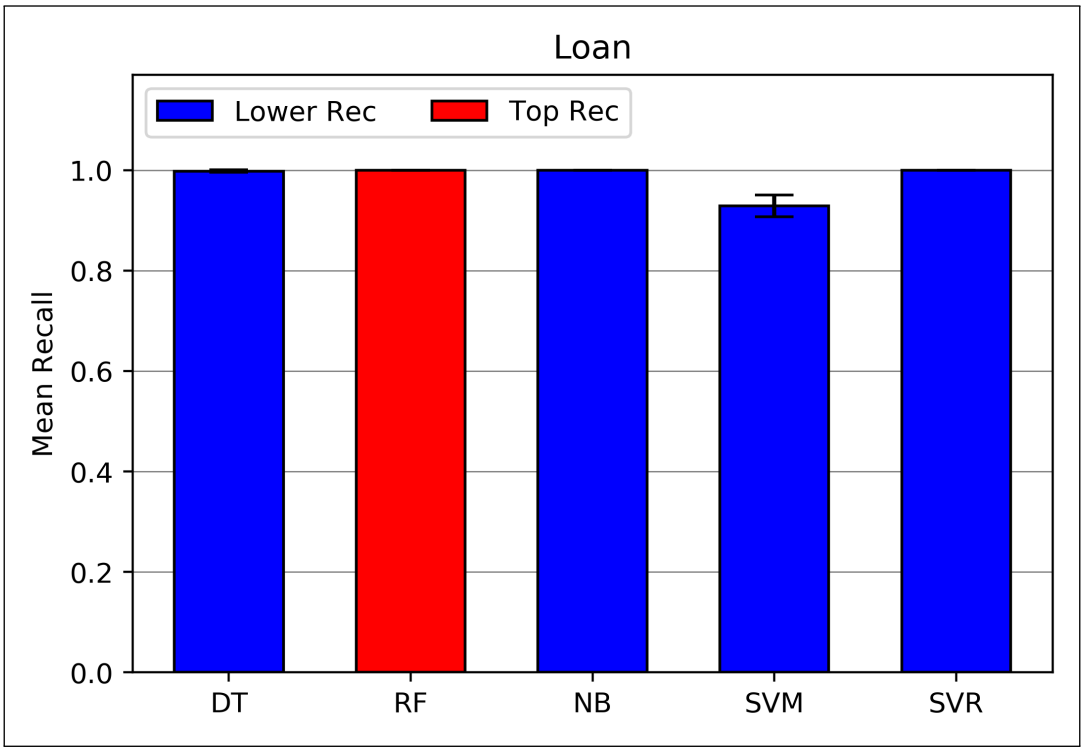

Figure 16. Mean recall for the Loan data set

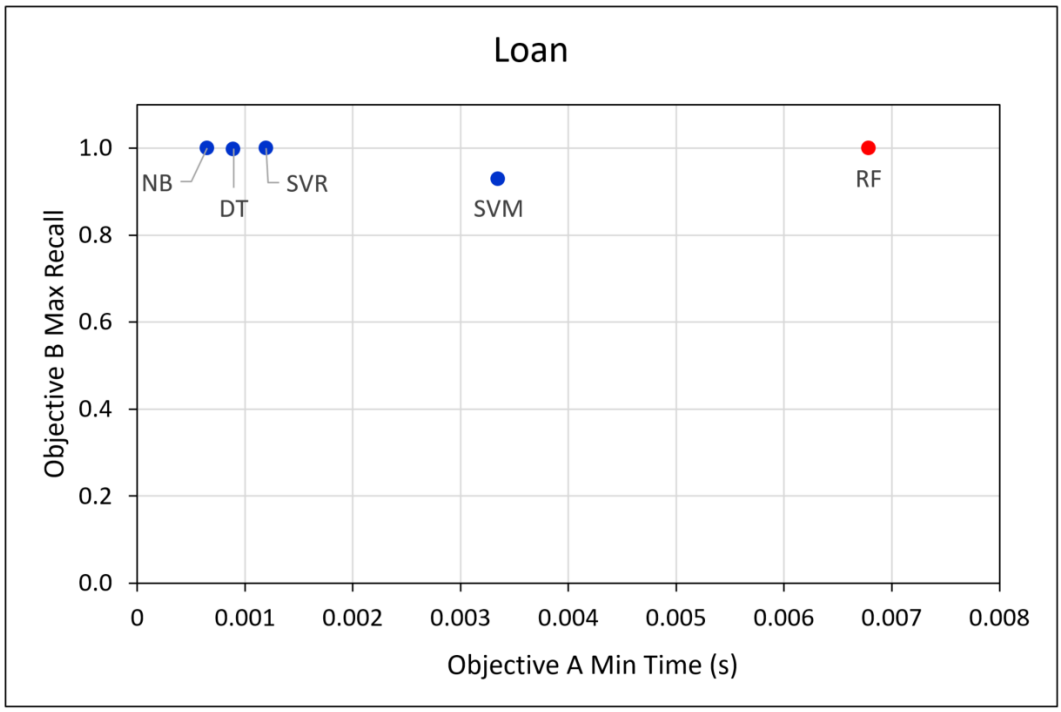

Figure 17. Mean recall and mean run time for the Loan data set

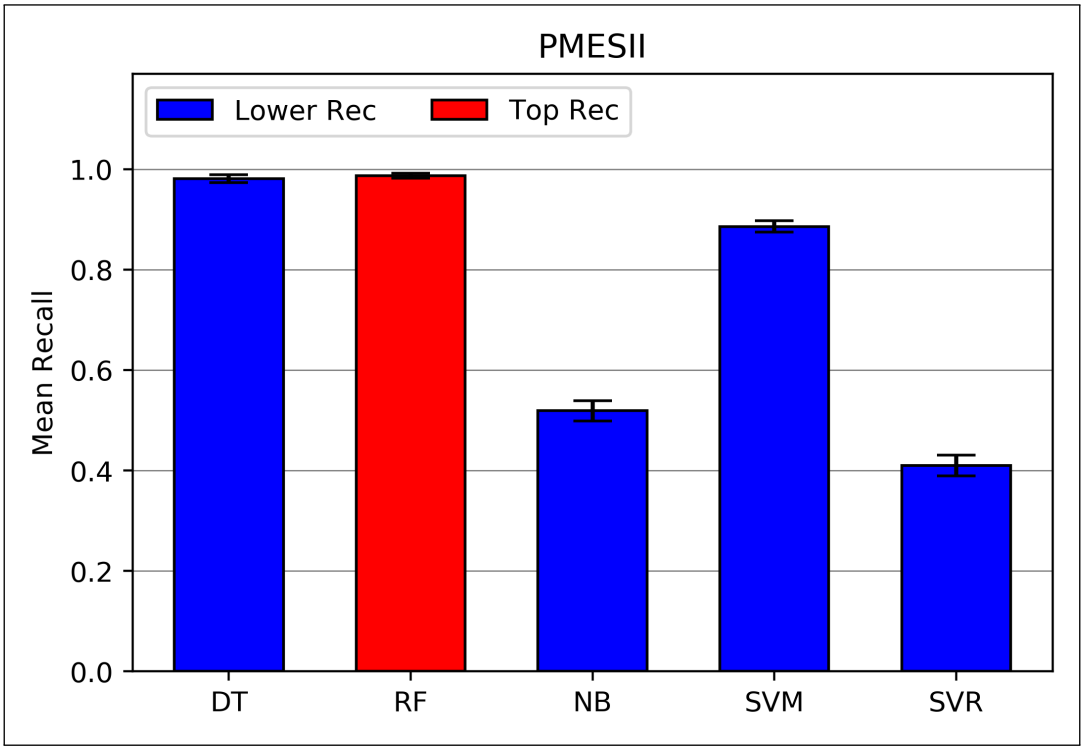

Figure 18. Mean recall for the PMESII data set

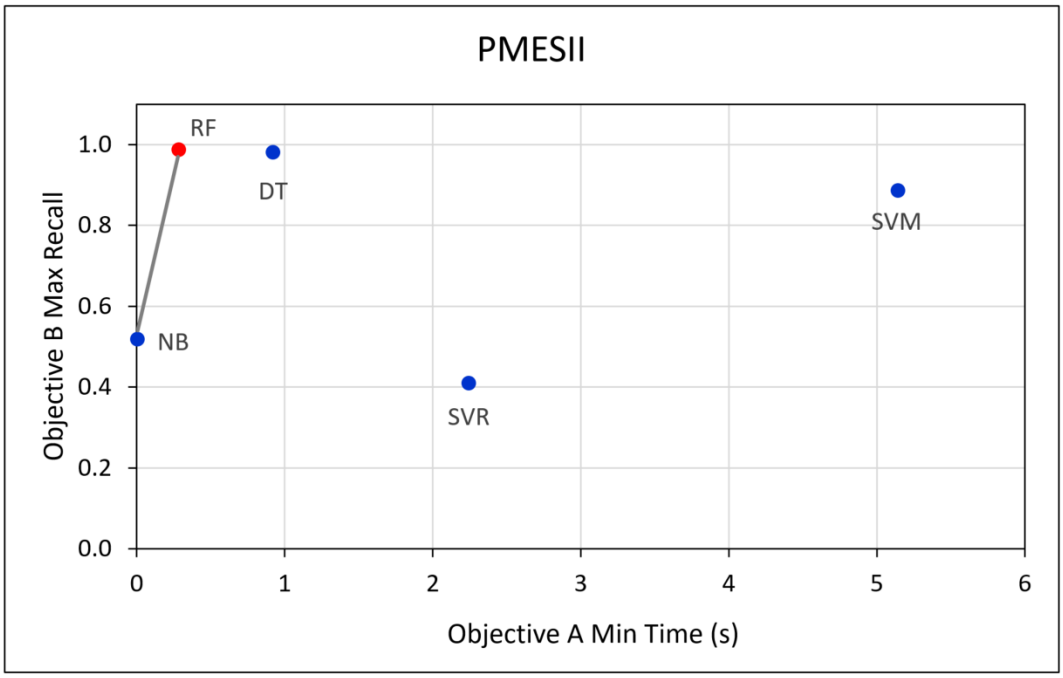

Figure 19. Mean recall and mean run time for the PMESII data set

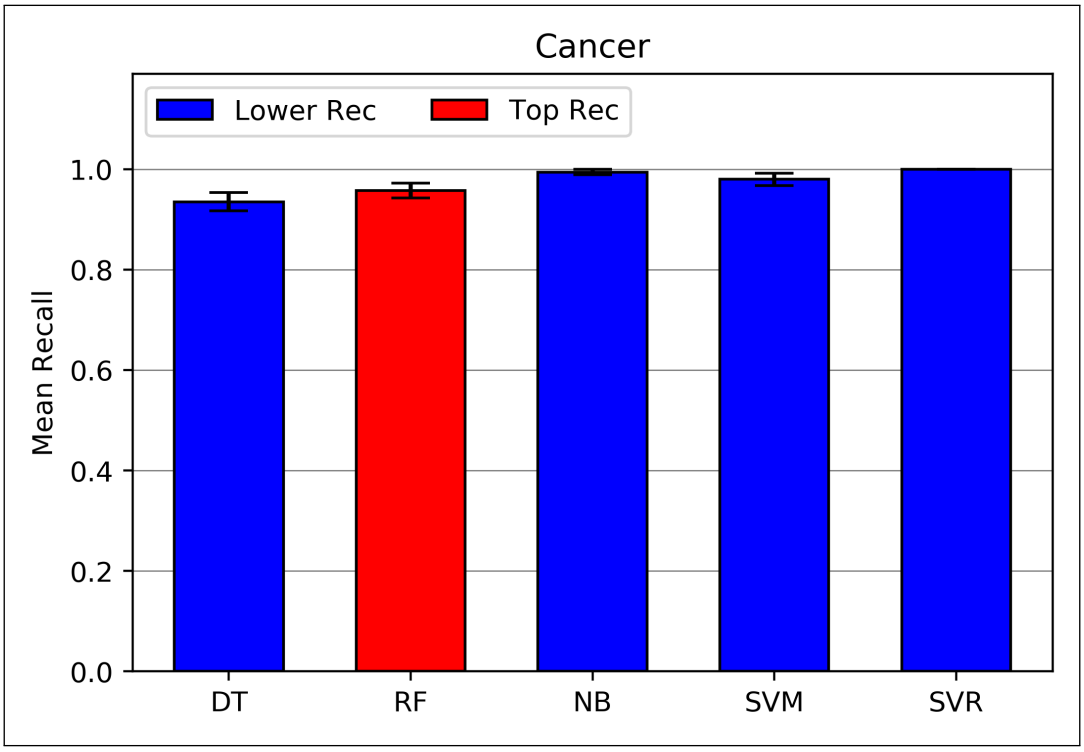

Figure 20. Mean recall for the Cancer data set

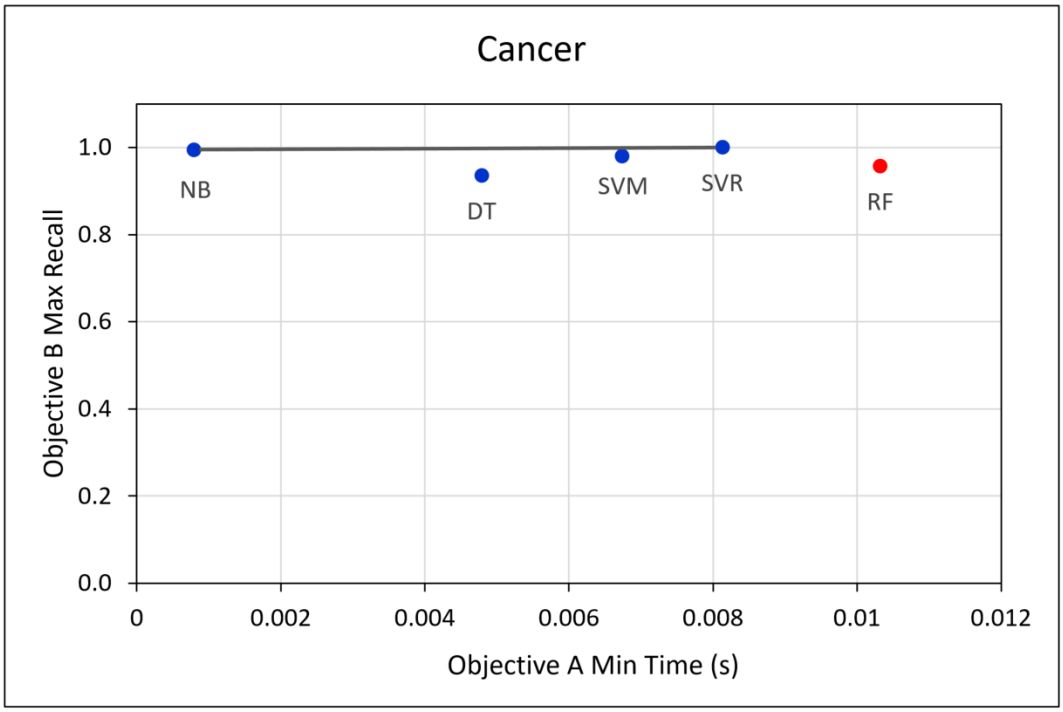

Figure 21. Mean recall and mean run time for the Cancer data set

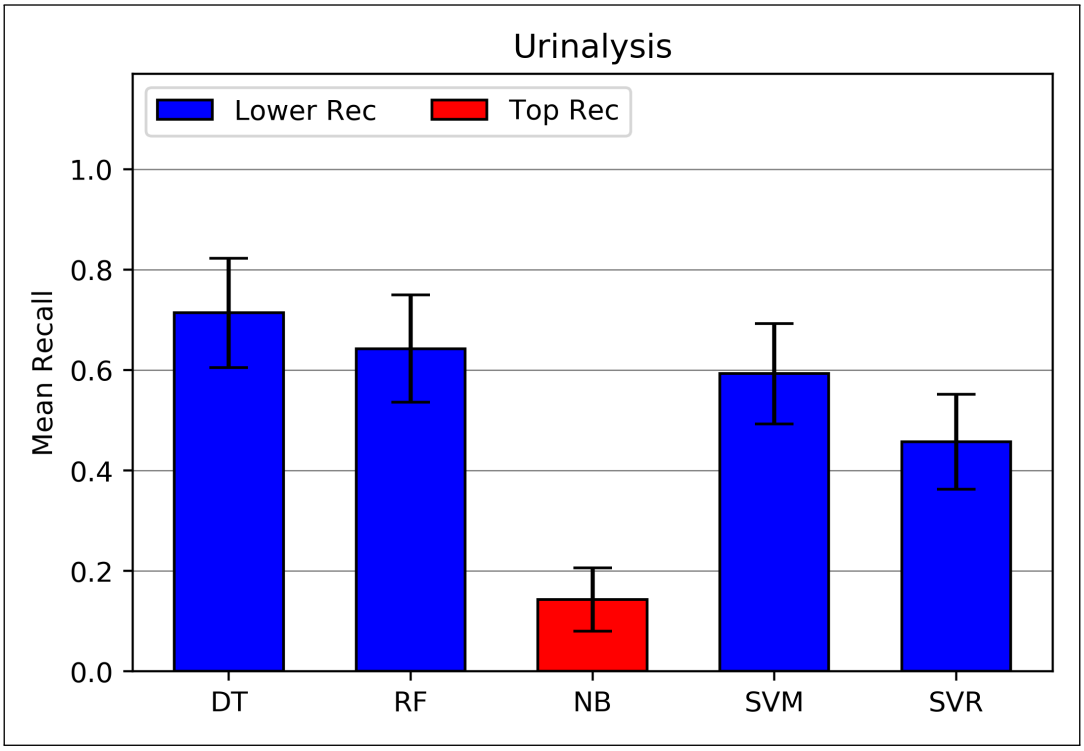

Figure 22. Mean recall for the Urinalysis data set

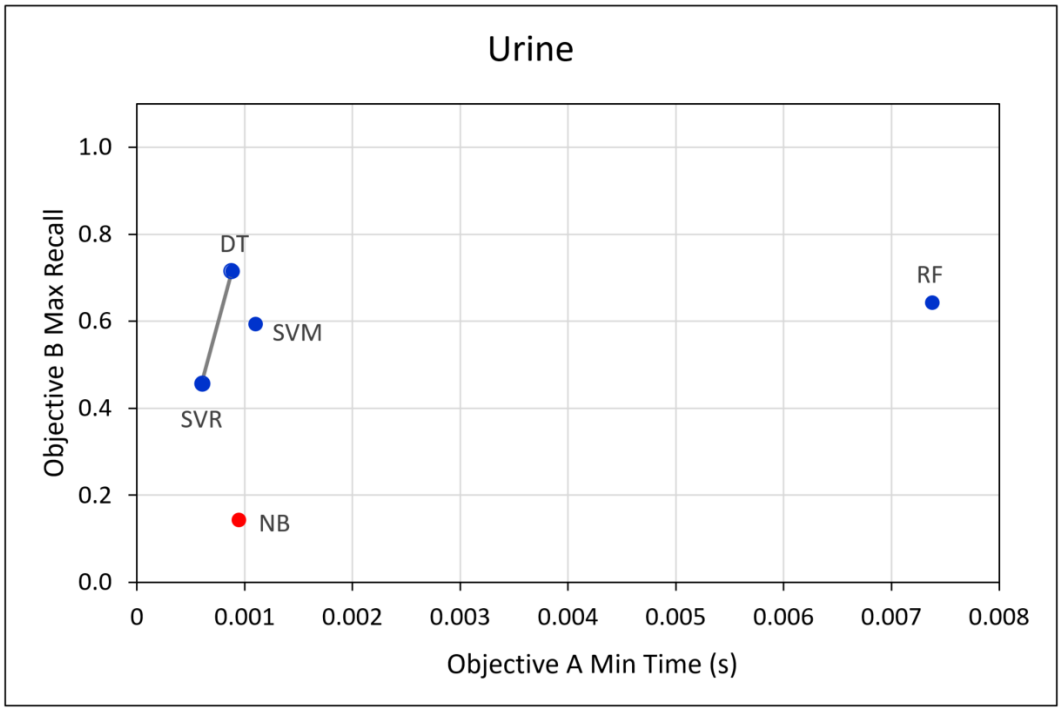

Figure 23. Mean recall and mean run time for the Urinalysis data set

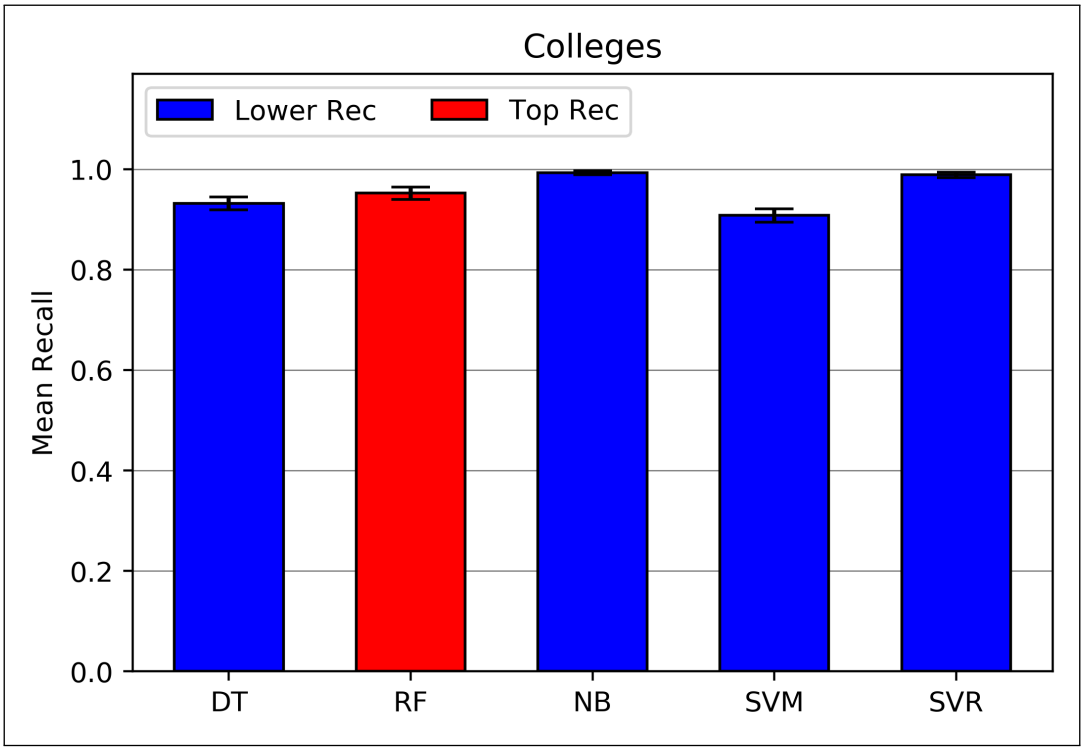

Figure 24. Mean recall for the Colleges data set

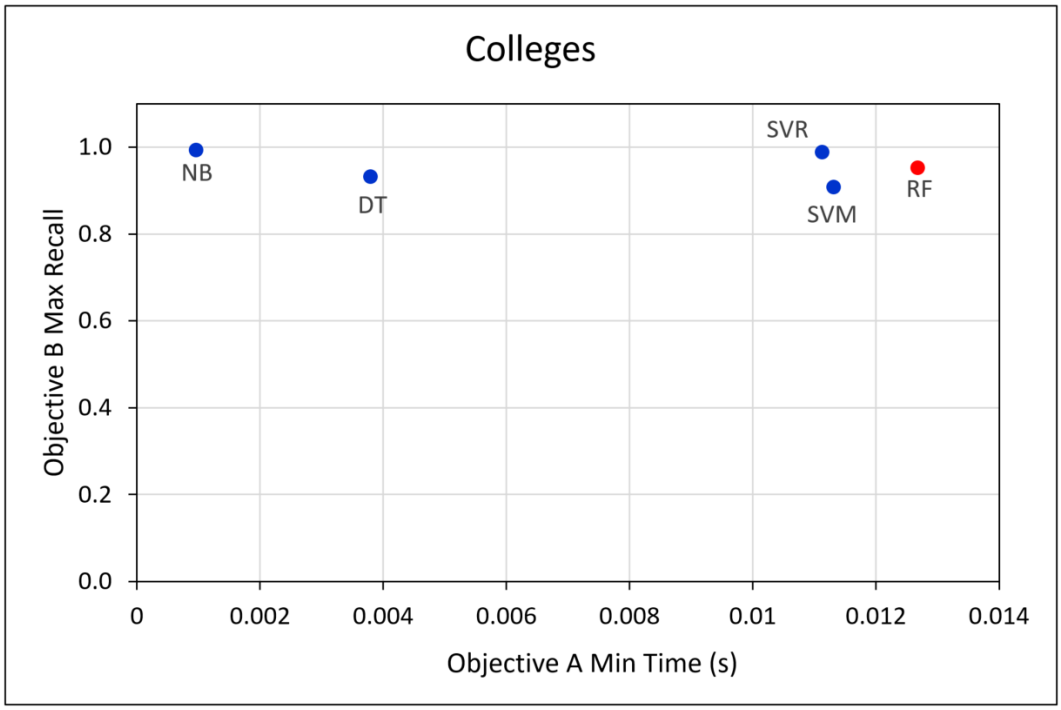

Figure 25. Mean recall and mean run time for the Colleges data set

VIII. Appendix C

8.1 Ranking Recommended Techniques for Data Factor

Characterize the Data Set

```
#Read in data set
filename=filename_str
data = pd.read_csv(filename)

#Assign response variable as y
firstcolname=list(data)[0]
y=data[[firstcolname]]
del data[firstcolname]

#Minimax normalization of data
min_max_scaler = preprocessing.MinMaxScaler()
final_data = min_max_scaler.fit_transform(data)

#Get n and m
n,m =np.shape(final_data)

if n>= 10^3:
    big_set=True
else:
    big_set=False

if m>= 10:
    many_vars=True
else:
    many_vars=False

#Get MajVarsCat
type_vect=np.zeros((1, m))

for i in range(0, m): #Py indexing to generate num_reps iterations

    if data.ix[:,i].nunique()>=12:
        type_vect[0,i]=1

    if np.mean(type_vect) >=.5: #tests whether the majority of
       columns have many levels
        data_categorical = True
    else:
        data_categorical = False


#Get Condition
if LA.cond(final_data)>=10^5:
    ill_cond=True
else:
    ill_cond=False
```

Assign Preference Ranks to Techniques

```
#Generate Recommendation

#Big set (Right side of tree)
if big_set==True and many_vars==True and data_categorical==True
    and ill_cond == True:
     Ranks=[3,1,2,4,5]      #RF NB DT SVM SVR

elif big_set==True and many_vars==True and data_categorical==False
    and ill_cond == True:
     Ranks=[4,3,2,5,1]      #SVR RF NB DT SVM

elif big_set==True and many_vars==True and data_categorical==False
    and ill_cond == True:
     Ranks= [3,1,2,4,5]     #RF NB DT SVM SVR

elif big_set==True and many_vars==True and data_categorical==False
    and ill_cond == False:
     Ranks= [3,2,1,4,5]     #NB RF DT SVM SVR

#Small set (Left side of tree)
elif big_set==False and data_categorical==True:# and many_vars==
    True #and ill_cond == False:
     Ranks= [3,2,5,1,4]     #SVM RF DT SVR SVM SVR

elif big_set==False and data_categorical==False: # and many_vars==
    True #and ill_cond == False:
     Ranks= [4,3,5,2,1]     #SVR SVM RF DT NB

else:
     Ranks= [4,2,1,3,5]     #SVR SVM RF DT NB

rank_array = Ranks
```

Return Object of Rank Scheme and Data Characterization

```
#Return an object that reports characterization and recommended
    rank scheme for the data set

class result:
     def __init__(self, ranks, ranks_df, bigset, manyvars,
         categorical, illcond):
          self.ranks = ranks
          self.ranksdf = Ranks
          self.bigset = bigset
          self.manyvars = manyvars
          self.categorical = categorical
          self.illcond = illcond

result_obj = result(rank_array, Ranks, big_set, many_vars,
    data_categorical, ill_cond)

return result_obj
```

Split Data to Training and Test Sets

```
#Read in new data set
data = pd.read_csv(filepath_str)

#Assign response variable as y
firstcolname=list(data)[0]
y=data[[firstcolname]]
del data[firstcolname]

#Minimax normalization of data
min_max_scaler = preprocessing.MinMaxScaler()
final_data = min_max_scaler.fit_transform(data)

# %% Enter loop for each rep
for i in range(1, num_reps+1): #weird python indexing will generate
    num_reps iterations
    seed = 18+i # fix random seed for reproducibility
    np.random.seed(seed)

    #Split the final data into train/test
    x_final_train, x_final_test, y_final_train, y_final_test =\
    train_test_split(final_data, y, test_size=0.2, random_state=
        seed, stratify=y)
    y_final_train=y_final_train.values.ravel()

    #Enter modelling module
```

Create Metamodels of Data Set

```
print('1/5: Creating Decision Tree Classifier', flush=True)
r=1 #Index number of technique
start_dt = time.time() #Record time Decision Tree begins

# Instantiate a DecisionTreeClassifier
dt_final = DecisionTreeClassifier(random_state=seed)
#defaults: max_depth default is until pure. default criterion is
    gini

# Fit dt to the training set
dt_final.fit(x_final_train, y_final_train)

#Predict the class of each observation of a dataset
y_pred_final_DT = dt_final.predict(x_final_test)

#Record time decision tree completes
now=time.time()
durationmin_dt = round((now-start_dt)/60)
durationsec_dt = round((now-start_dt)%60)
duration_mat[r-1,i-1]=(now-start_dt) #seconds

print('The Decision Tree model and predictions have been generated
    in', '%2.2d:%2.2d' % (durationmin_dt, durationsec_dt), flush=
    True)
```

Bibliography

1. Wayne L Winston and Jeffrey B Goldberg, *Operations Research: Applications and Algorithms*, vol. 3, Thomson Brooks/Cole Belmont, 2004.

2. Can Cui, Mengqi Hu, Jeffrey D. Weir, and Teresa Wu, "A Recommendation System for Meta-modeling: A Meta-learning Based Approach," *Expert Systems With Applications*, vol. 46, pp. 33–34, 2016.

3. Michael R. Smith, Logan Mitchell, Christophe G. Giraud-Carrier, and Tony R. Martinez, "Recommending Learning Algorithms and Their Associated Hyperparameters," *CoRR*, vol. abs/1407.1890, pp. 1–2, 2014.

4. Can Cui, Teresa Wu, Mengqi Hu, Jeffery D Weir, and Xianghua Chu, "Accuracy vs. Robustness: Bi-Criteria Optimized Ensemble of Metamodels," in *Proceedings of the Winter Simulation Conference 2014*. IEEE, 2014, pp. 616–627.

5. Timothy W Simpson, Jesse Peplinski, Patrick N Koch, and Janet K Allen, "On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments," *Design Theory and Methodology*, vol. 14, pp. 14–17, 1997.

6. G Gary Wang and Songqing Shan, "Review of Meta-Modeling Techniques in Support of Engineering Design Optimization," *Journal of Mechanical design*, vol. 129, no. 4, pp. 370–380, 2007.

7. David H Wolpert, "The Supervised Learning No-Free-Lunch Theorems," *Soft Computing and Industry*, pp. 25–42, 2002.

8. John Rice, "The Algorithm Selection Problem," *Advances in Computers*, vol. 15, pp. 75–152, 1976.

9. Frank Rosenblatt, "The Perceptron, a Perceiving and Recognizing Automation," Tech. Rep., Cornell Aeronautical Laboratory, 1957.

10. Marvin Minsky and Seymour A Papert, *Perceptrons: An Introduction to Computational Geometry*, MIT press, 1969.

11. David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning Representations by Back-Propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

12. Tom M. Mitchell, *Machine Learning*, McGraw Hill, New York, 1997.

13. James Arneal, "Spectral Textile Detection in the VNIR/SWIR Band," M.S. thesis, Air Force Institute of Technology, 2015.

14. Department of Defense, Washington, DC, *Summary of the 2018 National Defense Strategy of the United States of America*, Jan 2018, Available at *https://dod.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf*.

15. Fred Blackburn, Josh Sullivan, Peter Guerra, Angela Zutavern, Steve Escaravage, Ezmeralda Khalil, Steven Mills, Alex Cosmas, Brian Keller, Stephanie Beben Kirk Borne, Drew Farris, Paul Yacci, Charles Glover, Michael Kim, Stephanie Rivera, and Aaron Sander, *The Field Guide to Data Science*, Booz Allen Hamilton, McLean, Virginia, 2015.

16. Jiawei Han, Micheline Kamber, and Jian Pei, *Data Mining Concepts and Techniques Third Edition*, Morgan Kaufmann, Waltham, MA, 2012.

17. Kevin McGarigal, Samuel A Cushman, and Susan Stafford, *Multivariate Statistics for Wildlife and Ecology Research*, Springer Science & Business Media, 2013.

18. Desamparados Blazquez and Josep Domenech, "Big Data Sources and Methods for Social and Economic Analyses," *Technological Forecasting and Social Change*, vol. 130, pp. 99–113, 2018.

19. James J Cochran, *INFORMS Analytics Body of Knowledge*, John Wiley & Sons, 2018.

20. Bernhard Boser, Isabelle Guyon, and Vladimir Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Proceedings of the Workshop on Computational Learning Theory*, vol. 5, pp. 144–152, 1992.

21. José Luis Rojo-Álvarez, Manel Martínez-Ramón, Jordi Muñoz-Marí, and Gustau Camps-Valls, *Digital Signal Processing with Kernel Methods*, John Wiley & Sons, Inc., Hoboken, NJ, 2018.

22. J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

23. Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone, "Classification and regression trees," *Group*, vol. 37, no. 15, pp. 237–251, 1984.

24. Xindong Wu and Vipin Kumar, *The Top Ten Algorithms in Data Mining*, CRC press, 2009.

25. Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining, *Introduction to Linear Regression Analysis*, vol. 821, John Wiley & Sons, 2012.

26. Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik, "Support Vector Regression Machines," in *Advances in Neural Information Processing Systems*, 1997, pp. 155–161.

27. Vladimir N Vapnik, *The Nature of Statistical Learning*, Springer Science, New York, New York, 1995.

28. Brian Everitt and Torsten Hothorn, *An Introduction to Applied Multivariate Analysis With R*, Springer Science & Business Media, 2011.

29. John F. Kennedy, *Mathematics of Statistics*, Chapman & Hall Ltd, Boston, MA, 1939.

30. D. Ler, H. Teng, Y. He, and R. Gidijala, "Algorithm selection for classification problems via cluster-based meta-features," in *2018 IEEE International Conference on Big Data (Big Data)*, Dec 2018, pp. 4952–4960.

31. Kevin Markham, "ROC Curves and Area Under the Curve (AUC) Explained," Data School, Nov 2014, Available at *https://www.dataschool.io/roc-curves-and-auc-explained/*, Accessed 04 Oct 2019.

32. Christos H Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.

33. Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann, "The Balanced Accuracy and its Posterior Distribution," in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 3121–3124.

34. Megan K. Woods, "A Metamodel Recommendation System Using Meta-Learning," M.S. thesis, Air Force Institute of Technology, 2020.

35. Can Cui, Mengqi Hu, Jeffery D. Weir, and Teresa Wu, "A Recommendation System for Meta-Modeling: A Meta-Learning Based Approach," *Expert Systems with Applications*, vol. 46, pp. 33–44, 2016.

36. Paul E Utgoff, "Shift of Bias for Inductive Concept Learning," *Machine Learning: An Artificial Intelligence Approach*, vol. 2, pp. 107–148, 1986.

37. Larry A Rendell, Raj Sheshu, and David K Tcheng, "Layered Concept-Learning and Dynamically Variable Bias Management," in *IJCAI*, 1987, pp. 308–314.

38. Pavel Brazdil, João Gama, and Bob Henery, "Characterizing the Applicability of Classification Algorithms Using Meta-Level Learning," in *European Conference on Machine Learning*. 1994, pp. 83–102, Springer.

39. Michael R. Smith, Logan Mitchell, Christophe Giraud-Carrier, and Tony Martinez, "Recommending Learning Algorithms and Their Associated Hyperparameters," in *CEUR Workshop Proceedings*. 2014, pp. 39–40, Aachen University.

40. Bernhard Pfahringer, Hilan Bensusan, and Christophe G Giraud-Carrier, "Meta-Learning by Landmarking Various Learning Algorithms," in *ICML*, 2000, pp. 743–750.

41. Kyle Hsu, Sergey Levine, and Chelsea Finn, "Unsupervised Learning via Meta-Learning," in *Conference Proceedings ICLR*. 2018, pp. 1–24, International Conference on Learning Representations.

42. W. J Conover, *Practical Nonparametric Statistics*, John Wiley & Sons, Inc, New York, New York, 1971.

43. Myles Hollander, Douglas A Wolfe, and Eric Chicken, *Nonparametric Statistical Methods*, vol. 751, John Wiley & Sons, 2013.

44. "Heart Data Set," World Wide Web Page, Available at *https://www.kaggle.com/ronitf/heart-disease-uci*, Accessed 3 Jan 2020.

45. "Framingham Data Set," World Wide Web Page, Available at *www.kaggle.com/dileep070/heart-disease-prediction-using-logistic-regression#framingham.csv*, Accessed 3 Jan 2020.

46. "Spam Data Set," World Wide Web Page, Available at *http://vincentarelbundock.github.io/Rdatasets/doc/DAAG/spam7.html*, Accessed 3 Jan 2020.

47. "Bank Personal Loan Data Set," World Wide Web Page, Available at *https://www.kaggle.com/itsmesunil/bank-loan-modelling#Bank_Personal_Loan_Modelling.xlsx*, Accessed 3 Jan 2020.

48. "PMESII Data Set," World Wide Web Page, Available at *https://www.milsuite.mil/book/groups/analyst-stack-exchange/projects/programming-in-r/content?filterID=contentstatus%5Bpublished%5DsortKey=co centActivityDateDesc*, Accessed 3 Jan 2020.

49. "Correlates of Wat Data Set," World Wide Web Page, Available at *https://correlatesofwar.org/data-sets*, Accessed 3 Jan 2020.

50. "Breast Cancer Data Set," World Wide Web Page, Available at *https://goo.gl/U2Uwz2*, Accessed 3 Jan 2020.

51. "Urine Data Set," World Wide Web Page, Available at *http://vincentarelbundock.github.io/Rdatasets/csv/boot/urine.csv*, Accessed 3 Jan 2020.

52. "College Data Set," World Wide Web Page, Available at *http://vincentarelbundock.github.io/Rdatasets/csv/ISLR/College.csv*, Accessed 3 Jan 2020.

53. "2008 Election Data Set," World Wide Web Page, Available at *http://vincentarelbundock.github.io/Rdatasets/csv/Stat2Data/Election08.csv*, Accessed 3 Jan 2020.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
26-03-2020	Master's Thesis	SEP 2018 - MAR 2020

4. TITLE AND SUBTITLE

Algorithm Selection Framework:
A Holistic Approach to the Algorithm Selection Problem

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Chalé, Marc , W. Capt, U.S. Air Force

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-MS-20-M-137

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The "algorithm selection framework" uses a combination of user input and meta-data to streamline the algorithm selection for any data analysis task. The framework removes the conjecture of the common trial and error strategy and generates a ranked list of recommended analysis techniques. In seven of nine sample problems, the recall of the top ranked recommendation was considered "good" with at least 90% of the best observed recall. Pareto efficient recommendations for recall and run time were generated for three of the problems. The framework measured well against the pre-defined criteria. The framework successfully used information in the problem to recommend appropriate algorithms.

15. SUBJECT TERMS

meta-learning, machine learning, algorithm selection, data analysis, artificial intelligence

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Jeffery D. Weir, Ph.D., AFIT/ENS
U	U	U	UU	82	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4523; jeffery.weir@afit.edu