An Experimental Study of Machine Learning Evaluation Methods

A thesis submitted for the degree
Bachelor of Advanced Computing

12 pt Honours project, S2/S1 2021–2022

By:
Meilin Guo

Supervisor:
Kelvin(Yang) Li

School of Computing
College of Engineering and Computer Science (CECS)
The Australian National University

May 2024

Declaration:

I declare that this work:

- upholds the principles of academic integrity, as defined in the University Academic Misconduct Rules;
- is original, except where collaboration (for example group work) has been authorised in writing by the course convener in the class summary and/or Wattle site;
- is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener;
- gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used;
- in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling.

May, Meilin Guo

Acknowledgements

Thanks a lot for Kevin Korb's supervision, who is from Monash University and also one of the group who proposed BIR. Appreciate a lot for his work! Many thanks to my supervisor Kelvin(Yang) Li!

Abstract

Evaluation metric performs a important position in attaining the most reliable classifier for the duration of the class training. Thus, a choice of appropriate assessment metric is an essential key for discriminating and acquiring the most reliable classifier. However, while predicted accuracy is commonly used for evaluation, it can hardly tell the truth for every classifiers. Another metric, Bayesian Information Reward(BIR), was proposed in 2004 based on information theory is technically useful for all kinds of classifiers. In this project, a variety of classifiers and metrics will be tested in the experimental study and show the superior results under BIR.

Table of Contents

Chapter 1

Introduction

Both theoretical breakthroughs and practical applications of machine learning (ML) are exploding. Unfortunately, there hasn't been a parallel increase in understanding of how to assess machine learning algorithms. While it's clear that Deep Learning programmes have beaten the world's top people in cognitive games like Chess and Go, it's less clear what makes one machine learning algorithm superior to another in difficult prediction and classification tasks like cancer diagnosis or consumer credit scoring.

Instead, the machine learning community employs a patchwork of assessment methods, most of which are based on prediction accuracy, which is calculated by counting accurate classifications and dividing by the total number of classifications attempted. Despite the fact that the insufficiency of prediction accuracy and its cousins (such as the F score) has been recognised for decades, this is still the case. Cognitive psychology, in particular, has spent a lot of time trying to understand human thinking in the face of ambiguity, and has long recognised that human judgments, no matter how precise, have a strong propensity to be miscalibrated [48]. That is, even if people get the answer correctly 90% of the time, say in sports betting, they will frequently be overconfident, declaring that there is a 99 percent chance they are correct. For maximising expected value in classifications and evaluation, which is the Bayesian ideal criterion for assessing performance under uncertainty, both being correct as much as possible and knowing precisely how accurate "as much as possible" happens to be in any particular circumstance are required. This is, in reality, an Information Theory theorem[46].

When the predicted underlying truth of the system is known, Kullback-Leibler divergence provides an optimal assessment metric for probabilistic predictors[42]. KLD reports the average number of additional bits necessary to report samples from the true distribution using the trained distribution given a true probability density function and a learnt or candidate probability distribution. Of course, it is 0 if the learnt distribution is the genuine distribution, and it increases as the predicted distribution shifts away from reality. Many people in the field of machine learning are familiar with KLD, and it is frequently utilised when it is accessible. Unfortunately, it is not accessible in many supervised learning jobs since the underlying reality is unknown. In supervised learning tasks, for example, where data instances are pre-classified by experts as well as machine learning algorithms must learn from a random choice of 80% and be tested on the remaining 20%, the human experts may have some idea of what the underlying truth is like, but it is not widely in terms to machine learning researchers. KLD cannot be calculated.

A Bayesian alternative for binomial classification assuming 50-50 prior probabilities was developed early on (before machine learning actually existed as a discipline) by IJ Good [49]. More recently, Hope and Korb generalized this to any multinomial classification task and with any prior probability distribution over the classes, calling it Bayesian Information Reward (BIR)[45]. The theoretical superiority of BIR to predictive accuracy in assessing probabilistic predictions has always been clear. Regardless, BIR has been loudly ignored by the machine learning community. Quite possibly, the reason is that theoretical virtues, however clear in the abstract, do not communicate very well with those interested in day-to-day practice. This project aims to bridge that gap, by filling in with experimental evidence that cannot be swept under the carpet.

We'll evaluate a range of machine learning methods, including the widely used logistic regression algorithm, neural networks, and decision tree learners, as well as Bayesian network causal discovery techniques (probabilistic graphical models). All of these models are widely used, with proponents often claiming superiority for a variety of learning tasks, generally with just prediction accuracy data to back up their claims. We'll look into when BIR is better in identifying better learners than predictive accuracy, and when it isn't. The experimental article will have a strong influence on the machine learning community by laying out the practical evidence that BIR gets closer to the KLD true distribution than predicting accuracy over a wide range of instances.

Chapter 2

Background

2.1 What is Classification

The prediction problem is tackled utilising cutting-edge mathematical approaches from an algorithmic viewpoint. There are several techniques, but they all have share one fact: they all use available data (X variables) to determine the best prediction $\hat{Y}$ of the outcome variable Y. In multi-class classification, the response variable Y and the prediction Y may be thought of as two discrete random variables with values in the range $\{1, \cdots, K\}$, with each number representing a distinct class[4]. Fig 2.1 shows that the classification algorithm predict results given the data.

The algorithm calculates the likelihood that a certain unit belongs to one of many classes, and then uses a classification criteria to assign each individual to a single class. While there are other options in the multi-class scenario, the highest probability value and SoftMax are the most commonly used strategies.

From a perspective from Bayesian Inference, we regard θ as parameters of a classifier, then $p(C_k|X) = \frac{p(X|C_k) \cdot p(C_k)}{P(X)}$, where the fraction here means the likelihood. A classification problem is to compute $p(x, t)$ from a set of data, that is, take a specific action

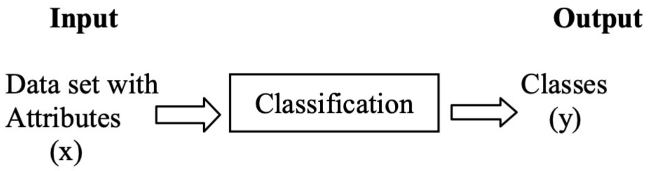

Figure 2.1: Classification from x to y[44]

based on our understanding of the value t is likely to take[43].

2.2 A Brief Review on BIR

Before the project details, it is essential to have a brief review on the metric Bayesian Information Reward, that was proposed by Lucas R. Hope and Kevin B. Korb in 2004[45] and the main content of this section is based on the paper as well.

2.2.1 Motivation

There are two key components of gambling success, and we'd like our metric to be maximum when they're both maximised:

Property 1: Domain knowledge, which may be quantified by the likelihood of accurately asserting $x_i = T$ or $x_i = F$, i.e., predictive accuracy. In sports betting, for example, the more often you can choose the winning team, the better.

Property 2: Calibration, or the bettor's proclivity to set $P(x_i = T)$ as near to the objective probability as possible (or, actual frequency). Theorem 6.1.2 in Cover and Thomas's Elements of Information Theory [46] proves that perfect calibration maximises betting payoff.

2.2.2 Good's score

I.J. Good proposed the original information reward (IR) as fair betting costs — the cost of buying a bet that makes the expected value of the transaction zero[50].If the IR of a binary classification is probability $p6$,then IR is as Eq.2.1 and Eq.2.2 show. When compared to a uniform prior, Good's IR rewarded binary classifications that were informative. Given a uniform prior, when the learner reports a posterior probability of 0.5, it is not communicating any information and so earns no reward. Good's approach is extended to multi-class classification tasks, and the reward function is relativized to non-uniform prior probabilities.

$$I^{+} = 1 + \log_2 p^{'} \ (for\ correct\ classification)(2.1)$$

$$I^{-} = 1 + \log_2 (1 - p^{'}) \ (for\ misclassification)(2.2)$$

2.2.3 Kononenko and Bratko's Metric

Kononenko and Bratko's[51] metric relativizes reward in terms of previous probability. This contradicts the ostensible information-theoretic foundation: according to Shannon, a reward for a specific true prediction can only be limited, whereas a penalty for such a prediction going wrong can only be infinite. There will be no valid information-theoretic explanation of their reward function if these are balanced. Kevin and Hope do agree, though, that the type of cost-neutral reward they are looking for needs to be highly relativized to prior likelihood. The Kononenko and Bratko's reward function is as Eq.2.3 and Eq.2.4 show, where $p^{'}$ is the estimate probability and p is the prior.

$$I_{KB}^{+} = \log_2 p - \log_2 p^{'} \ (for\ correct\ classification)(2.3)$$

$$I_{KB}^{-} = -\log_2 (1-p) - \log_2 1 - p^{'} \ (for\ misclassification)(2.4)$$

This metric is solely used to evaluate the genuine class. Because the probability of other classes are not taken into account, a miscalibrated judgement of the alternative classes will not be punished in multinomial classification. As a result, KLD cannot be minimised. For all of these reasons, we believe the Kononenko and Bratko functions are insufficient.

2.2.4 Bayesian Information Reward(BIR)

Good's IR does not fully handle the concept of fair fees, which states that you should only be compensated for an informed forecast. This expert is slacker and just says that each patient does not have the cancer, with a confidence level of 0.9. For this technique, Good's anticipated reward per patient is 0.9(1 + log2 0.9) + 0.1(1 + log2 0.1) = 0.531[45], therefore the expert is well compensated for the misinformed method! Only when the antecedent is constant and the job binary are Good's fair fees genuinely fair.

The BIR formula for a classification into classes $\{C_1, ..., C_k\}$ with predicted probabilites $p_i^{'}$ and prior probabilities p_i, where $i \in 1, \cdots, k$, is

$$IR_B = \frac{\sum_i I_i}{k}$$

where $I_i = I_i^{+}$ if classifies correctly and $I_i = I_i^{-}$ if misclassifies, and I_i^{+} and I_i^{-} are as Eq.2.5 and Eq.2.6 show.

$$I_i^{+} = \log \frac{p_i^{'}}{p_i} \ (for\ correct\ classification) \tag{2.5}$$

$$I_i^{-} = \log \frac{1-p_i^{'}}{1-p_i} \ (for\ misclassification) \tag{2.6}$$

When p = p, the reward is obviously 0. The measure is finitely constrained in the positive direction, because prior probabilities are never zero, and misclassifications deserve an infinite negative reward, according to BIR. Finally, in the long term, accurate probabilities are now rewarded the most. This evidence is not included, however it is structurally comparable to the proof in Section 10.8 in[52].

In Hope and Kevin's paper, they has shown some results based on bayesian models using artificial data, including Naive Bayes, Tree Augmented Naive models(TAN), Averaged One-Dependence models(AODE) and CaMML. I will continue and extend the research scope into real data sets to explore more and see whether we could prove that BIR has its own advantages.

Chapter 3

Related Work

After some background information, some related state-of-art researches related to our project will be demonstrated in this section.

Other related works can be dating back to the early 2000s [1], that involves an expirical study among all kinds of metrics. They proposed another metric as well, SAR. However,they proposed it based on the 2D-correlation analysis upon the study results. In other words, no theoretical support for SAR, only a linear combination of existing metrics in a way the evaluation ability would be better.However, this is the pioneer among all works focusing on machine learning metrics.

Another thorough review upon measurements would be the paper proposed in 2015[2]. This paper also suggests five important aspects that must be taken into consideration in constructing a new discriminator metric. So far, people started to discuss how to construct a better suitable metric than the existing ones to break the cover under predicted accuracy.

The third paper is a more detailed review on only multi-class classification problems without any experiments. In addition, [5] proposed a new feature selection metric called ‘Bi-Normal Separation’ (BNS) and shows that it performs the best under a combine of BNS and f1-score. The last study examines twenty-four performance metrics used in the full range of Machine Learning classification problems, including binary, multi-class, multi-labelled, and hierarchical classification. [6]

The common gap between theoretical and experimental is not studied under any related start-of-art work. The review on supervised machine leaning metrics under binary and multi-class problems is not studies at all and such few studies show the comparisons under experimentation on experiments. The motivation of this project would be make a more thorough comparison, involving as much data sets, classifiers and metrics as we can during experimental parts and shows whether BIR can do a better job.

Chapter 4

Methodology

After a thorough review on state-of-art empirical studies about classification performance metrics, three main three parts of my work will be introduced here, including metrics, classifiers, data sets along with other essential techniques.

4.1 Metrics

The most important part in this experiment is which metrics we want to compare and why we chose them. Next in this section we will describe them in detail.All indicators are divided into three major categories, and the most representative measurements for each category are selected to conduct experiments, and measurements that do not belong to any of the three major categories in our experiments will be included as well.

4.1.1 Threshold Types

- **Predicted Accuracy and Confusion Matrix**

  Table 4.1 shows the confusion matrix on binary cases. For multi-class problem, we treat each class binary and combine the results

Table 4.1: The Confusion Matrix on Binary Cases.

	Actual Positive Class	**Actual Negative Class**
Predicted Positive Class	True Positive(tp)	False Negative(fn)
Predicted Negative Class	False Positive(fp)	True Negative(tn)

Some indicators are introduced based on confusion matrix. The mostly widely used one is predicted accuracy. If $\hat{y}_i$ is the predicted value of the i-th sample and

y_i is the corresponding true value then predicted accuracy over $n_{samples}$ is defined as Eq 4.1.

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i) \tag{4.1}$$

However, One of the primary drawbacks of accuracy is that it provides values that are less distinct and discriminable[34][36], since it treats all cases the same and frequently favours the majority class [35]. Some more metrics are popular as well. Precision(Eq 4.2) and recall(Eq 4.3) both contribute equally to the F1-score(Eq 4.4), and the harmonic mean may be used to discover the optimum trade-off between the two [37].Here we use accuracy to represent the confusion matrix collective.

$$precision = \frac{tp}{tp + fp} \tag{4.2}$$

$$recall = \frac{tp}{tp + fn} \tag{4.3}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{4.4}$$

4.1.2 Relative Order Types

Metrics are dependent on the order of predicted classes.

- **Receiver Operating Curve (ROC) and Area under the ROC Curve (AUC)** For measuring classifier performance and discriminating an ideal solution during classification training, the AUC has been shown to be conceptually and experimentally superior than the accuracy metric [39]. The Receiver Operating Characteristic (ROC) curve was developed by engineers during World War II for detecting enemy objects in battlefields[41]. The ROC curve is defined as a plot of Se(c) versus 1Sp(c) forc,or equivalently as a plot of Eq 4.5 over $t \in [0, 1]$, where $F^{-1}(1 - t) = \inf\{\mathrm{x} \in \mathrm{R} : \mathrm{F(x)} \geq 1 - \mathrm{t}\}$. The AUC is given by Eq 4.6. The ROC analysis can be used to: (i) evaluate a continuous marker's discriminatory ability to correctly assign two-group subjects; (ii) find an optimal cut-off point to least misclassify two-group subjects; (iii) compare the efficacy of two (or more) diagnostic tests or markers; and (iv) investigate inter-observer variability when two or more observers measure the same continuous variable[42].

$$ROC(t) = 1 - G(F^{-1}(1 - t)) \tag{4.5}$$

$$AUC = \int_0^1 ROC(u)du \tag{4.6}$$

- **Others** The average accuracy is generally calculated as the sum of the precisions at eleven recall levels that are evenly spaced. Lift is a term commonly used in marketing analysis to describe how much better a classifier predicts positives than a baseline classifier that predicts positives at random[47]. These are not suitable for classification tasks. Hence, we only test on ROC_AUC, the most representative indicator.

4.1.3 Probability Types

This section discusses probability measures that are based on expected values rather than how they fall relative to a threshold or to each other.

- **Mean Squared Error.** MSE over $n_{samples}$ is defined as Eq 4.7.

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i) \tag{4.7}$$

where $\hat{y}_i$ is the

- **Calibration.** Another version: chart calibration plot [40] Here version: doc Such classifiers provide a score (from 0 to 1) or probability to each example, which should indicate the genuine probability that the example belongs to the positive class. Expected probabilities (scores) are properly calibrated, which means that a percentage of occurrences with predicted probability p really happen[42].

$$\sqrt{\sum_{i=1}^{n} \frac{(p_i - \phi_i)^2}{n}}$$

Interpret with n the number of predictions made, p_i the prediction probability, phi_i the actual probability. This eliminates any need for bucketing; buckets of predictions can be treated as n_b individual predictions. The actual probability can be estimated from prior known outcomes, or may be delivered by a physical theory, or may be estimated by human experts. This is better than ECE above.

Calibration is then measured by how close to zero this gets. Division by n guarantees this is not overly sensitive to sample size. (Of course, it will vary highly with small ‘ and little with larger samples.)

- **Brier Score.**

$averaged(prediction - observation)^2$

This is proper, but does a really poor job, especially with the extreme predictions 0 or 1. These correspond to infinite odds, something which a Bayesian ought to be reluctant to take, since it implies losing everything if you are wrong once.

4.1.4 Other Types

Other types of significant metrics with respect to classification problems are shown here, including BIR, Expected Value and Kullback–Leibler Divergence(KLD).

- **BIR.** As discussed in the Background section.
- **Cumulative/Average Expected Value.**

This works if and only if predicted outcomes have known values or rewards, and so it's a special case. But on Bayesian grounds, when it is available, it is the best way of assessing predictions. Some datasets have at least rough values. For example, it's entirely clear that the preference order for predicting some cancers would be $TN > FP > TP > FN$. Predictive models can be assessed on their average outcomes' values.

One of the most difficult and most critical parts of implementing data science in business is quantifying the return-on-investment or ROI. As a data scientist in an organization, it's of chief importance to show the value that your improvements bring.

To calculate an expected value one simply multiplies each outcome by its given probability and add all those results together to one value. In other words it is calculated by multiplying each of the possible outcomes by the likelihood each outcome will occur and then adding all of those values to one key value. Expectation, the mean or the first moment are all terms also used instead of expected value itself.

- **Kullback-Leibler Divergence(KLD)** We may use the Kullback-Leibler divergence to quantify the amount to which the posterior distribution of the model varies from the real distribution if we have the true probability distribution of the target variable. To put it another way, we may make use of

$$\mathrm{KLD}(p,q) = \sum_{x \in X} p(x) log \frac{p(x)}{q(x)} \tag{4.8}$$

The two features of betting reward (refer to background) are reversed in Kullback-Leibler divergence: decreasing KLD maximises betting reward, and vice versa. Divergence is clearly reduced when q = p, implying that the model is fully calibrated. But it also entails Property 1: if you know the correct probability distribution in the model, there is no further probabilistic information to be acquired. As a result, KLD is an excellent statistic for assessing machine learning. In reality, it has the serious issue of requiring access to the correct probability distribution, which implies there is no meaningful learning challenge. KLD, on the other hand, serves as a good standard for evaluating other measures[45].

4.2 Classifiers

Only supervised classifiers are involved in this project. The classifiers we used are: Prior Knowledge, Logistic Regression, Support Vector Machine(SVM), Naive Bayes, Decision Tree and Random Forests. We tend to test on Artificial Neural Network(ANN) as well, since neural network started to performs well on complex data and leads another trend in machine learning.

Prior Knowledge here refers to the concept that, we only use the prior knowledge to make predictions, without any trace of training. The priors usually come form "the background" or the memories before this event from a view of humanity. For a clarification task, the priors could be given in advance or solely from the data set. Here we chose to use the proportions of classes among the whole data to be the prior, regardless of the train set and test set sampled. Therefore, it only predict to the majority class since it takes the largest proportion of the data set. Therefore ,we expect Prior Knowledge classifier to be the "baseline" and other classifiers are expected to perform no worse than the baseline. Other classifiers are widely used in classification tasks.

4.3 Data Sets

The data sets I used here are from UCI Machine Learing Repository: Abalone, Letter Recognition, Cover Type and Wine Equality.

4.4 Other Techniques

Other technique used are listed here.

4.4.1 Smoothing

Laplace Smoothing is used here to prevent exact probabilities of zeros coming from bayesian classifiers: Random Forest, Naive Bayes and Decision Tree to compute BIR since the probabilities are the denominator. The formula is

$$\hat{\theta}_i = \frac{x_i + a}{N + \alpha d}$$

where I set α to be 1.

4.4.2 Sampling

Many sampling approaches are popular in the field of machine learning. Furthermore, improving the performance of the network model requires researchers to consider sampling approach, proper distance metric, and network structure.[38]

Proportional sampling is used in train set since it preserves the original proportions of classes, which makes the classifiers to learn better. A random sampling is used for test data so that it keeps the randomness for testing.

4.4.3 Confidence Interval

The purpose to apply confidence interval is to get a interval of each measurement under each sampling size of each classifiers for several times of repetitions instead of the average value. It could be more accurate and we compare the metrics under the observation among overlapping intervals(discussed more in the next section).

Chapter 5

Evaluation

After a detailed background and theoretical support about this project, we started to design and implement many experiments under the goal of 'comparing metrics'. This section focuses on the experimental design, procedure and results, as well as some interesting observations and discussions based on the results.

5.1 Implementation

Detailed project implementations will be demonstrated in this section. First, some clarifications are made before the real empirical process. Then, the packages and methods used in the experiments will be stated, including pre-processing for each data set, how to define classifiers' and metrics' functions and the sampling methods we used here. Finally, the whole experimental procedures are displayed by an algorithm.

5.1.1 Clarifications

Dot points of clarifications for the whole project are illustrated in this subsection.

- **Setup and Project Scope.** The whole experiments are tested only on CPUs on IOS operating system.All steps are done using Python programming language on JetBrains PyCharm IDE. This project mainly aimed for supervised learning techniques on binary or multi-label problems because of time limitation. Random seed for Logistic Regression, Decision Tree, Random Forest and SVM classifiers is set to 1989 to ensure the reproducibility of this entire project. All data are numeric, that means no attribute column in any data is categorical, that mainly

- **No Validation Set or Cross Validation.** For normal classification tasks, evaluating the model on a non redundant data set from either train or test data, namely validation set, is crucial to tune parameters of this model, while performance on test data basically only shows the generalization and portability. Among a range of techniques to separating such data set, k fold cross validation[c] is the more popular one since it largely reduced computation cost. Especially, bias-corrected versions of cross-validation proposed by Burman[a] and Yanagihara et al[b] overcome the potential bias problem. Back to the experiments for this topic, there is no point to adjusting the "settings" of any classifier, while, on the contrary, we need the constant classifiers for all data, thus to compare discrimination ability for a set of metrics on the identical classifiers.

- **Identical Classifiers.** Since the procedure to fine tuning parameters in all classifiers is abandoned, to ensure the identity of classifiers on different data sets, unique functions are built for each classifier and revoke them each time for different classifiers on a variety of data sets respectively.

- **Fixed Test Set Size.** Normally, train/test split ratio is fixed and a relatively common ratio is between 0.2 to 0.3[a]. However, in this experiments, the test size for each data set is constant respectively. Normally, it is the half size of the whole data set and the maximum size is 10,000 due to computation resources limitation. The reason I chose to set a constant test size is to cover the randomness and all type of cases that do not overlap with the training set of as much as possible.

  [a]Crowther, Patricia S., and Robert J. Cox. "A method for optimal division of data sets for use in neural networks." International conference on knowledge-based and intelligent information and engineering systems. Springer, Berlin, Heidelberg, 2005.

- **Repeat experiments** In this project, each experiment is repeated for ten times to compute confidence intervals. That is because as the time the project repeat increase, the shorter the intervals. After testing for five, ten, fifteen and twenty times respectively, the plots for ten repeats display the most appropriate interval length.

- **No OneHot Encoding** A common used approach to deal with categorical attributes in Python is to apply OneHot or Label Encoding to convert one multi-label column into multiple binary columns so that built-in learners can accept data. However, this kind of approaches completely ignore the dependencies between attributes so as to influence classifier learning, especially for classifiers based on Bayesian Inference including Naive Bayes, Decision Tree and Random Forest[a]. Therefore, only numeric data sets are involved in experiments, and that leads to fewer options for data sets as well. [a] An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing

- **No Normalization.** Normalisation of data is the scaling of data to a scale. Many experiments related to machine learning use max-min normalisation or any other standardization approach to bring all the data about the metric down to between 0 and 1, to remove the unit constraints of the data and transform it into a dimensionless pure value, so that metrics of different units or magnitudes can be compared and weighted[70]. We did not perform any kind of normalisation on the metric used to make the graph after all the experiments, because it did not play a substantial role in this experiment.

[70]Jain, Anil, Karthik Nandakumar, and Arun Ross. "Score normalization in multimodal biometric systems." Pattern recognition 38.12 (2005): 2270-2285.

- **Unfinished Work.** A typical three-layer feed-forward neural network is prepared to test, using "setting..". However,the results are not interesting since if we fix the hyper parameters for the network and abandon the process to tuning the model, like cross validation, the results for all data sets are too plain. In addition, it is almost impossible to find such a set of hyper parameters to show excellent discrimination among all data. Hence, the code part is finished, though, due to the time limitation, the experiments on ANN are dropped. For expected value, it is hard to construct such a weight matrix for all data, hence, it is dropped a well. The same as KLD, no artificial data is tested so far. Hopefully, we could finish them in the future work.

5.1.2 Basic Steps

Some basic steps, including processing data, train/test split and set-up classifiers and metrics are demonstrated in this subsection. All steps are sealed inside functions. Therefore, when I do those steps, I call the corresponding functions.

- **Pre-processing.** For Abalone, the regression problem is converted into a classification problem by grouping values in label column, that is value 1-5 falling into class 0, 6-10 into class 1, 11-15 into class 2, 16-20 into class 3 and 20-29 into class 4 respectively. Also convert Sex into 0 male and 1 female.

  Class 9 in Wine Equality-white data is dropped since it is too small (with only five instances). Therefore the classes of white wine are identical to those of red wine. Also, since they are from the same data set, the attributes are identical as well. Hence, Wine Equality data set is a combination of Wine Equality-white and Wine Equality-red data sets.

  For Letter Recognition, we encode 26 letters into class [0-25] and each class has approximately 800 cases.

  The Table 5.1 shows description after processing data and what pre processing steps have been taken.

- **Sampling Data.** The train sizes and constant test size are shown in Table 5.2. They are dependent on the size of whole data set respectively.

- **Classifier Set-up Functions.** Among all the classifiers and metrics I implemented in the codes, I use several built-in functions in Python packages. Parameters for each differ from default ones are listed as well, as Table 5.3 shows.

Table 5.1: Data set Description after Pre-processing.

	Shape	Balanced	No. Classes	Pre-processing
Cover Type	(581012,55)	False	7	-
Abalone	(4177,9)	False	6	group values, encode Sex
Letter Recognition	(20000,17)	True	26	label encoding 26 letters
Wine Equality-red	(1599,11)	False	6	-
Wine Equality-white	(4893,11)	False	6	drop class 9
Wine Equality	(6488,11)	False	6	-

Table 5.2: No. Train Set Instances and Constant Test Set Size on Data Sets.

	Train Set Sizes	Test Set Size
Cover Type	[500,1000,5000,10000]	10000
Abalone	[500,1000,2000]	2000
Letter Recognition	[500,1000,5000,10000]	10000
Wine Equality-red	[500,700]	700
Wine Equality-white	[500,1000,2000]	2000
Wine Equality	[500,1000,4000]	2000

- **Metric Set-up Functions.** Only MSE, ROC_AUC and accuracy are implemented using Python Packages, as Table 5.4 shows. The other three metrics are constructed according to formulas listed in the previous section.

5.1.3 Experimental Procedures

The below Algorithm 1 shows the main procedure shared for all data sets. Briefly speaking, we obtain ten scores for each measurement for each train size using each classifier on each data set.

5.2 Results

In this section, I will show the result plots for each data set first. Then I will discuss more behind the results before we make a concrete conclusion. In addition, we make a prediction that BIR will stand out among all the metrics.

5.2.1 Plots

Plots under each data sets are as Figure 5.2.1 - Figure 5.2.1 shows. Each figure corresponds to the results on one specific data set. There are six plots in each figure, where the vertical coordinates represent a specific metric, and the horizontal coordinates are the different train sample sizes in this data set. There are six lines in each figure, which

Table 5.3: Classifiers Set-up Functions.

	Package	Parameters
Prior Knowledge	-	-
Logistic Regression	sklearn.linear_model.LogisticRegression	max_iter = 1000
SVM	sklearn.svm.SVC	probability = True
Naive Bayes	sklearn.naive_bayes.ComplementNB	-
Decision Tree	sklearn.tree.DecisionTreeClassifier	max_depth=4
Random Forest	sklearn.ensemble.RandomForestClassifier	-

Table 5.4: Metric Set-up Functions

	Package	Parameters
MSE	sklearn.metrics.mean_squared_error	-
Accuracy	sklearn.metrics.accuracy_score	-
ROC_AUC	sklearn.metrics.roc_auc_score	multi_class='ovr'

represent the results and variations of a classifier on different sizes of training sets. We can see that each data is represented by a column, because we use 95% confidence interval for more accurate representation. Also, to make it easier to visually compare the variation of the lines in different plots, we inverted the vertical coordinates of the plots representing MSE, Calibration, and Brier Score by 180 degrees, because these three criteria are the less the better, while the other three criteria (BIR, ROC_AUC and Acuracy) are in the opposite way. Therefore, for all plots, the higher the line the better the performance of the classifier under that metric.

5.2.2 Discussions

Generally speaking, BIR is more discriminatory than other metrics in some cases, but less discriminatory for multi-class problems. It usually favors Bayesian type classifiers(Naive Bayes, Random Forest and Decision Tree) alot, then SVM and finally gives a very hard penalty on prior knowledge.

Among all the metrics, only BIR gives the worst score on Prior Knowledge, which are treated as "the baseline" so are expected to have the worst scores. BIR gives almost 10 times lower score on classification model only dependent the information from the data set.

The result for SVM classifier (shown as the black line) biases a lot under BIR metric than other metrics for wine equality data set. This is mainly because SVMs are not suitable for multi-class data sets since it draws decision boundary to discriminate so that it is more suitable for binary data (it takes more time to classify as well). Therefore, it may be very hard to find a hyperplane to separate classes for all the data sets and the changes

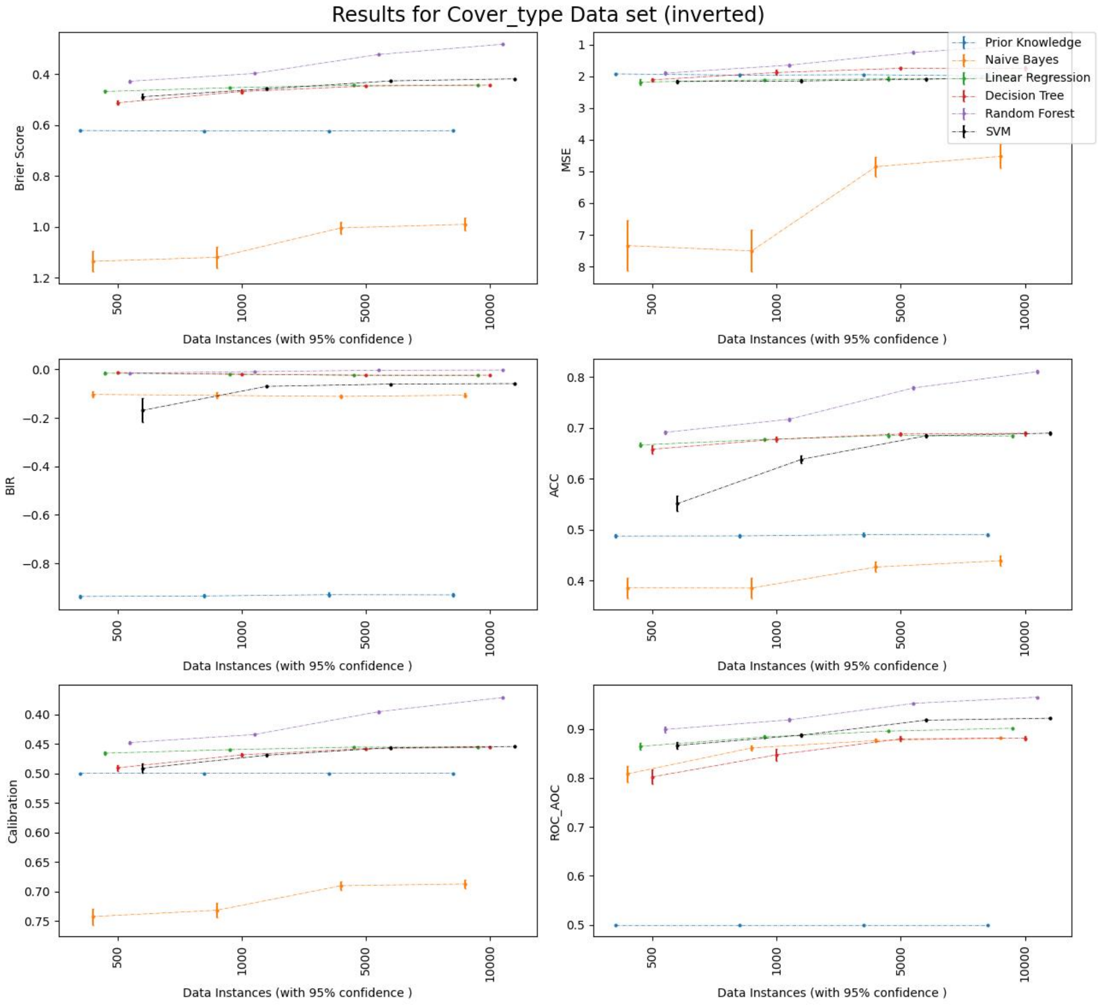

Figure 5.1: Cover Type data set

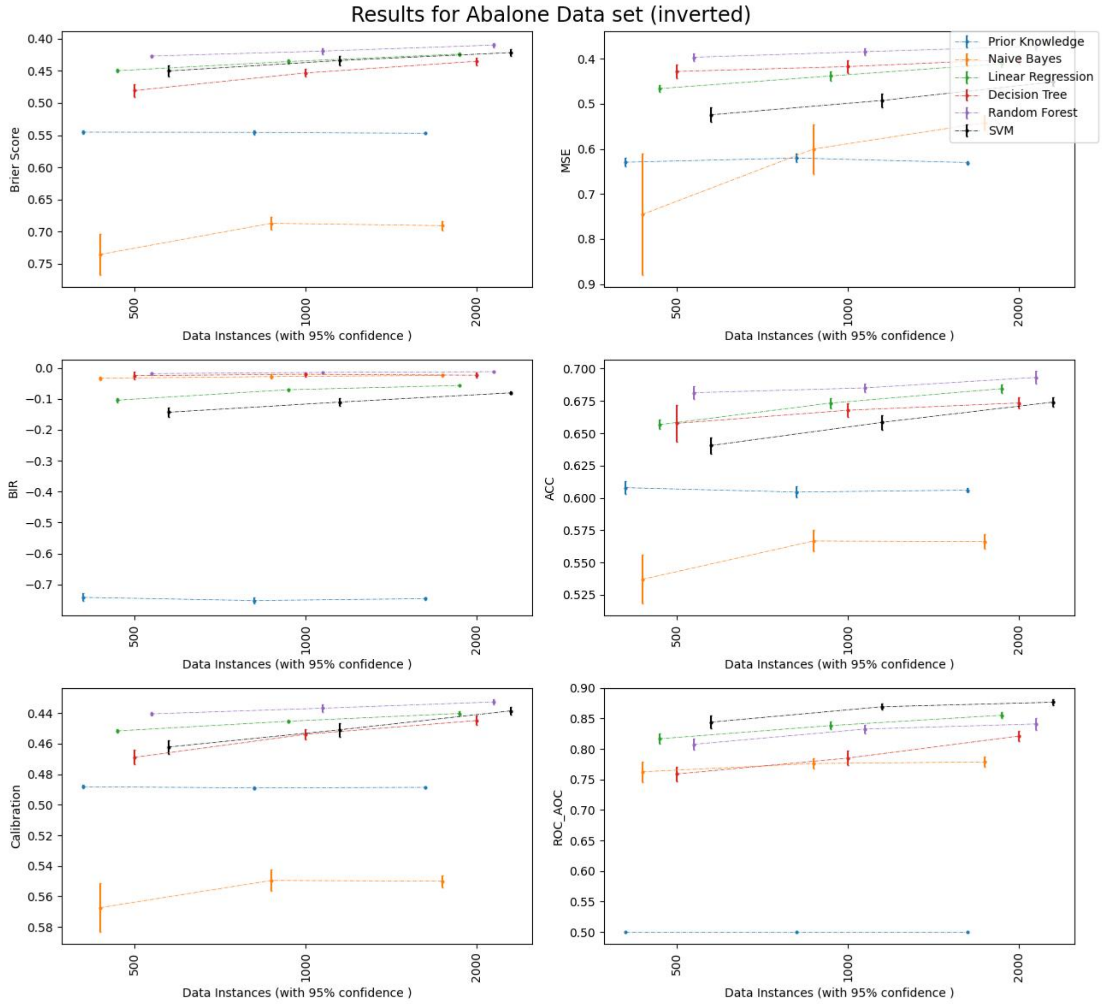

Figure 5.2: Abalone data set

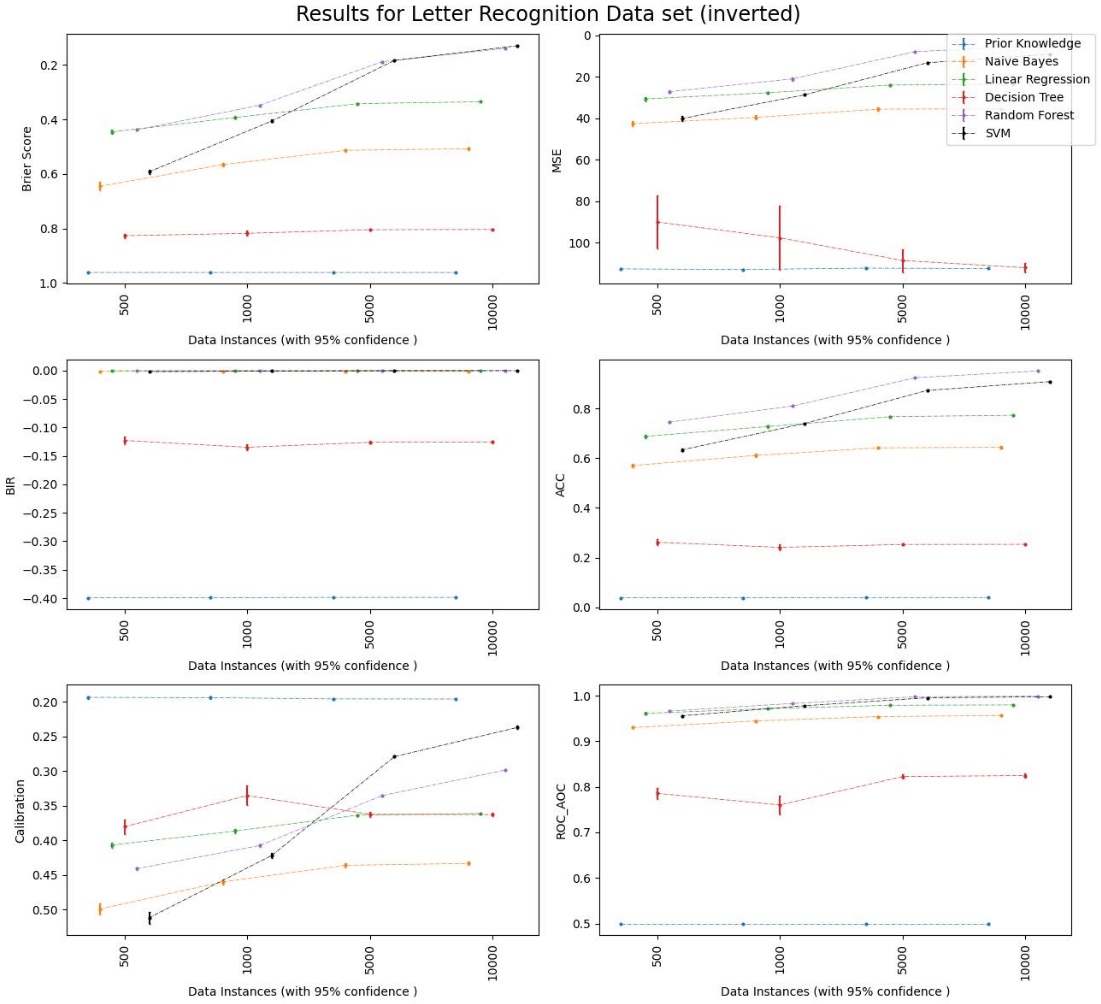

Figure 5.3: Letter Recognition data set

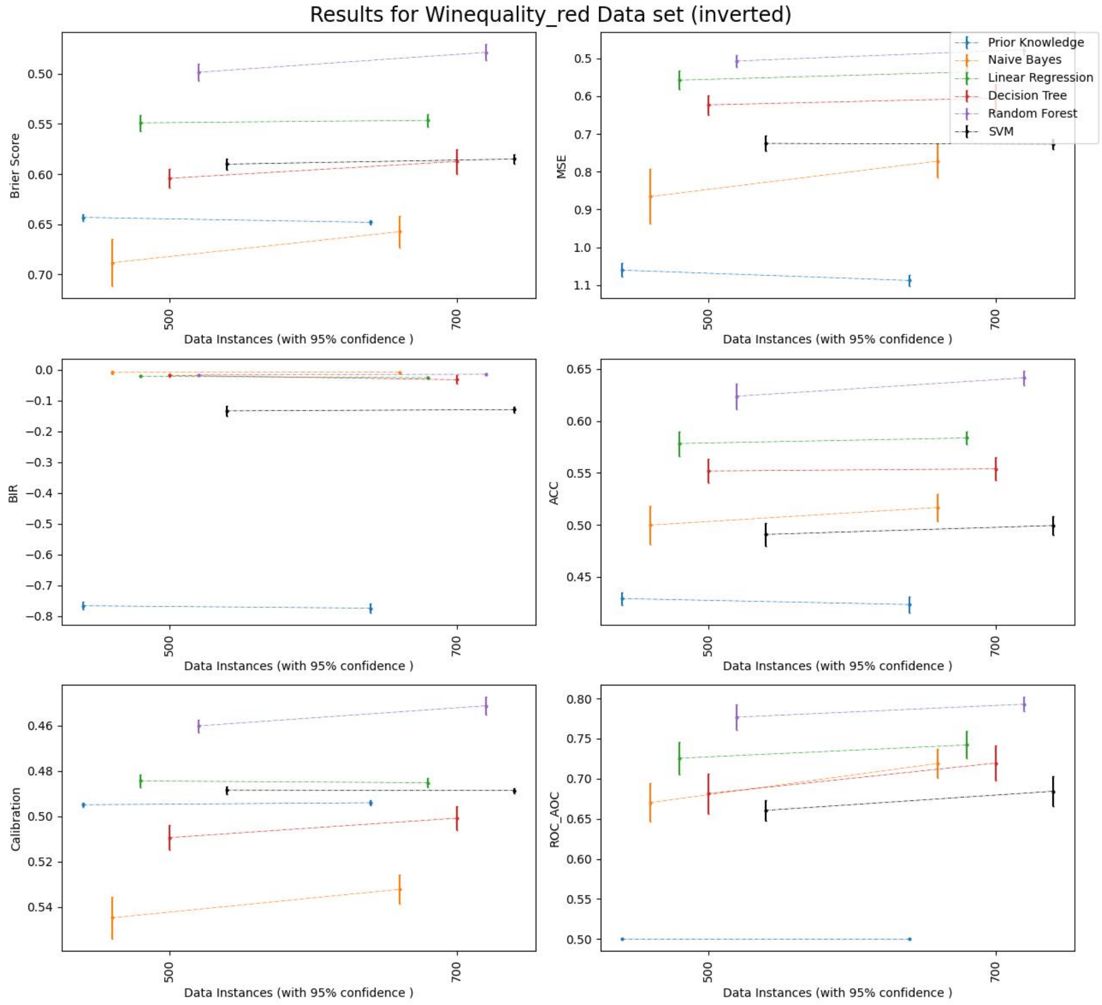

Figure 5.4: Wine Equality-red data set

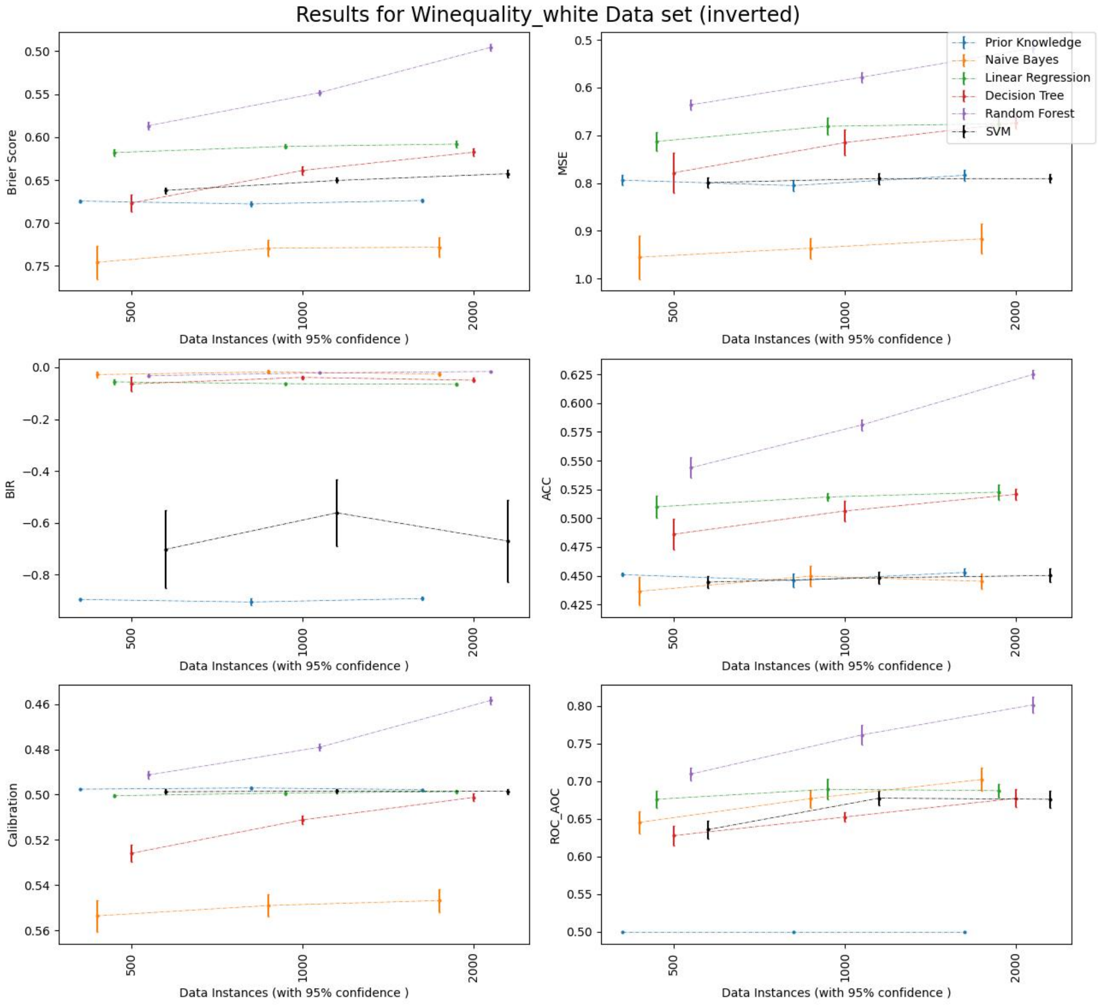

Figure 5.5: Wine Equality-white data set

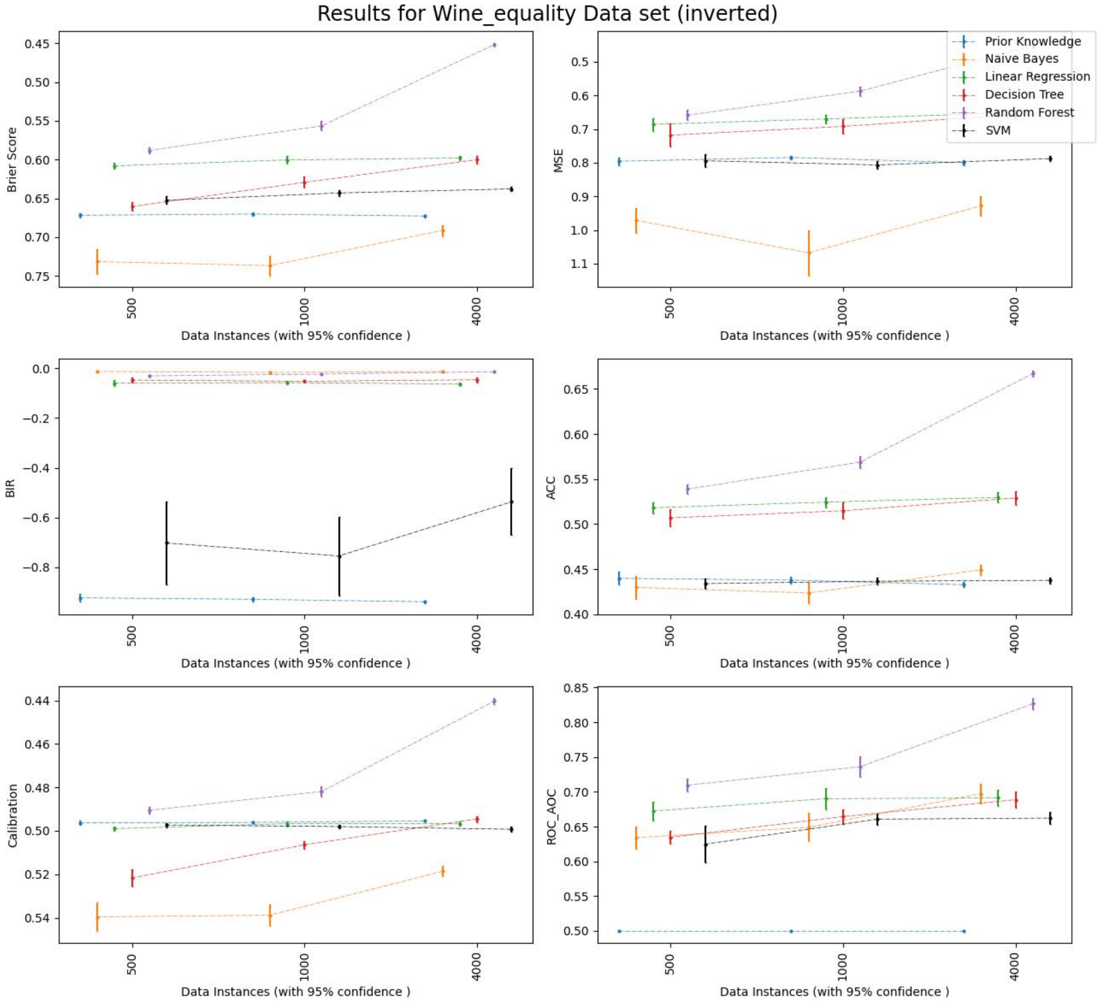

Figure 5.6: Wine Equality data set

Algorithm 1: Experimental Procedures

```
for all data sets do
    Compute prior knowledge from the whole data set
    for all sample sizes do
        for all classifiers do
            while repeat ten times do
                Re-sample train and test data
                Call classifier and get results given sample size, test set size, the priors and
                the data, including the predicted values, the predicted probabilities and the
                order of classes on train and test data
                Call Laplace Smoothing on the predicted probabilities
                evaluate on all metrics and store the scores
            end while
            Get ten scores for all metrics after ten times of repetition
            Compute CI for each metric
        end for
    end for
    Plot results
end for
```

of the scores may be more.

We have discussed that calibration is significant among unbalanced data. Take the balanced data, letter recognition data set, as an example, the plot trends of accuracy is much different to that of plot calibration: as the sample size increases, calibration under SVM jumps while accuracy of SVM stays steadily. We know that SVM is built under the maximum margin which takes no account of the proportions of classes. Hence, it should performs bad when only few cases are covered, i.e. the sample size is quite small. As more cases are invloved, it starts to show what it really can do.

Since BIR is a ”combination” concept of accuracy and calibration, it should've be similar to the trends for both accuracy and calibration. However, from the plots, it is not that similar to any of them. This can be interpreted by the reward/penalty for each class in BIR, instead of the average sum of some loss for each data instance.

Some bad news about BIR, since the scores for prior classifiers are too low, we can hardly tell the differences among other classifiers(overlap too much, especially for letter recognition data set). Hence, in this case, it shows the drawback of this metric: not discriminate enough for all data set.

Chapter 6

Concluding Remarks

After analysing those experiments on many supervising classifiers on a variety of data sets, we comes to the conclusions and potential future works later for other people to continue this project due to the time limitation.

6.1 Conclusion

We discussed several metrics for evaluating machine learners. The precision is rough, and the domain knowledge is only optimized, but the calibration is neglected. Real world data, on the other hand, don't apply and are ideal. We found deficiencies in other information theory measures, including our BIR measure. We develop new metrics that are maximized through a combination of domain knowledge and perfect calibration. This information encourages learners to evaluate estimates of the distribution of the class as a whole, rather than a single classification. When cost-sensitive measurements are lacking, it provides a viable alternative to rewarding calibration.

We applied many experiments and found that BIR is more discriminatory than other indicators in some cases, but less discriminatory for multi-class issues. It generally favors Bayesian classifiers (Naive Bayes, random forests, and decision trees), then support vector machines, and finally strict penalties for prior knowledge. Of all the indicators, only BIR gave the worst score on prior knowledge, which is considered "baseline" and therefore expected to have the worst score. BIR gives a classification model almost 10 times lower, relying only on information from the data set.

In general, BIR shows good performance as other commonly used metrics and more experiments should be done to make a precise conclution.

6.2 Future Work

Combined with unfinished work and what we expect the following work could accomplish based on observations from this project, some potential future work derived.

Firstly, instead of report accuracy and calibration separately, it is better to find a more proper way to "combine" them, like introduce a new metric based on their definitions. Therefore, we can compare BIR with that metric as well. Secondly, generating artificial data to compare the predicted distribution and true distribution using KLD is another essential way to make comparisons between a variety of metrics. Thirdly, include Expected Value and ANN to more fully compare and analyze if possible. In addition, in order to study the relationship between the variation of BIR and accuracy and calibration, respectively, in a more detailed and reasonable way, we can do an analysis like this: use as many as data sets and generate as many as Bayes Networks to representing dependencies, to make a two-dimensional graph with horizontal and vertical coordinates of the variation of calibration and accuracy, respectively, and study the variation of BIR following them in different combinations of values. Moreover, test with a relatively large number of real data sets can make the experimental data more convincing.

Appendix A

Appendix: Mistakes on Confidence Interval Computation

This appendix is to show a mistake I made when computing confidence intervals. When I am coding, as we can see from the experimental processes, I used a lot of loops when writing the code to make the experiment automatically generate loops on a data set to classify different classifiers at different sample sizes and print the plots. This has made my testing process very efficient to a large extent. However, because I am very incompetent at degugging and testing code, I made a mistake: I wrote the wrong indent when initializing the list of six lists representing different metric for each classifier at different sample sizes. Then when I calculate the confidence interval for all lists, it is not just for 10 scores, but for all the data of the same metric on all sample sizes and all classifiers that I have done before, and the result will be very small for each classifier with different sample sizes. And I made another serious mistake by confusing the variable names, so that the Brier Score and BIR (the first chart in the first row and the first chart in the second row) accept the opposite data in each data set. At the time I discovered this problem, there were only a few days left before the deadline for this project, which then amounted to incorrect results for all the previous ones I had done and analyzed. This revealed my lack of knowledge on testing and debug test results. I will try to learn such knowledge in the future to avoid such mistakes. Here is a comparison of the experimental results on letter-recognition data set before and after correcting these problems, as Fig A and Fig A.

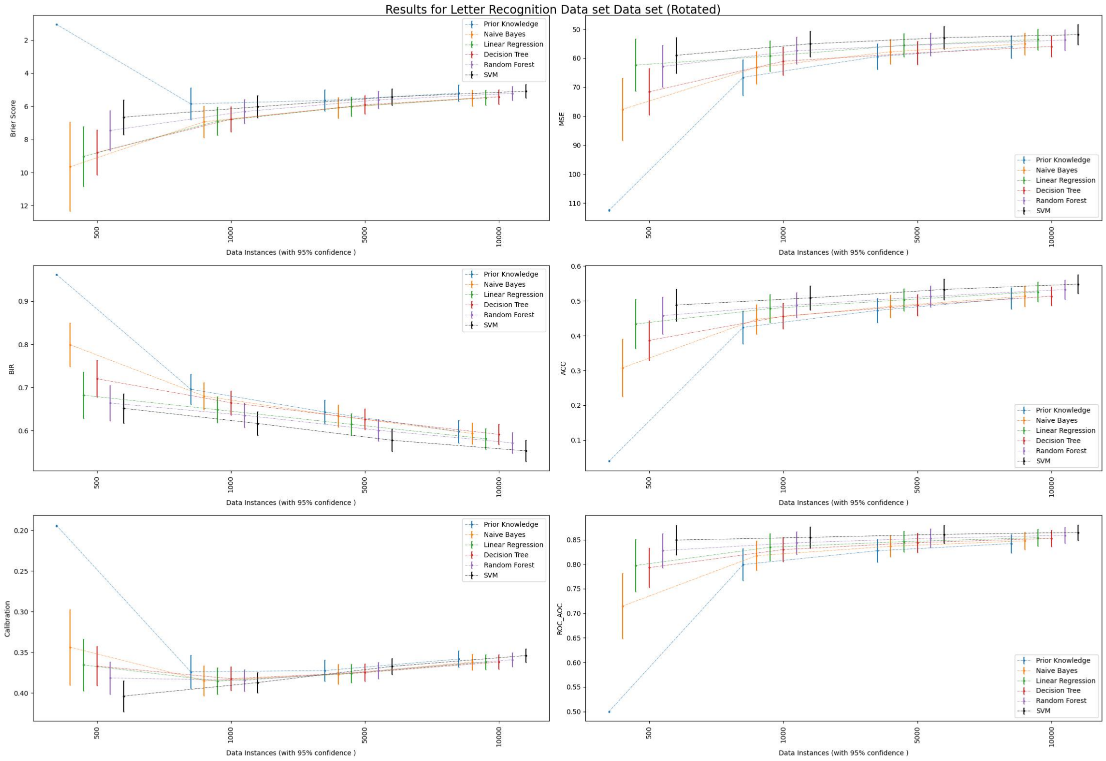

Figure A.1: Letter Recognition data set(changed before)

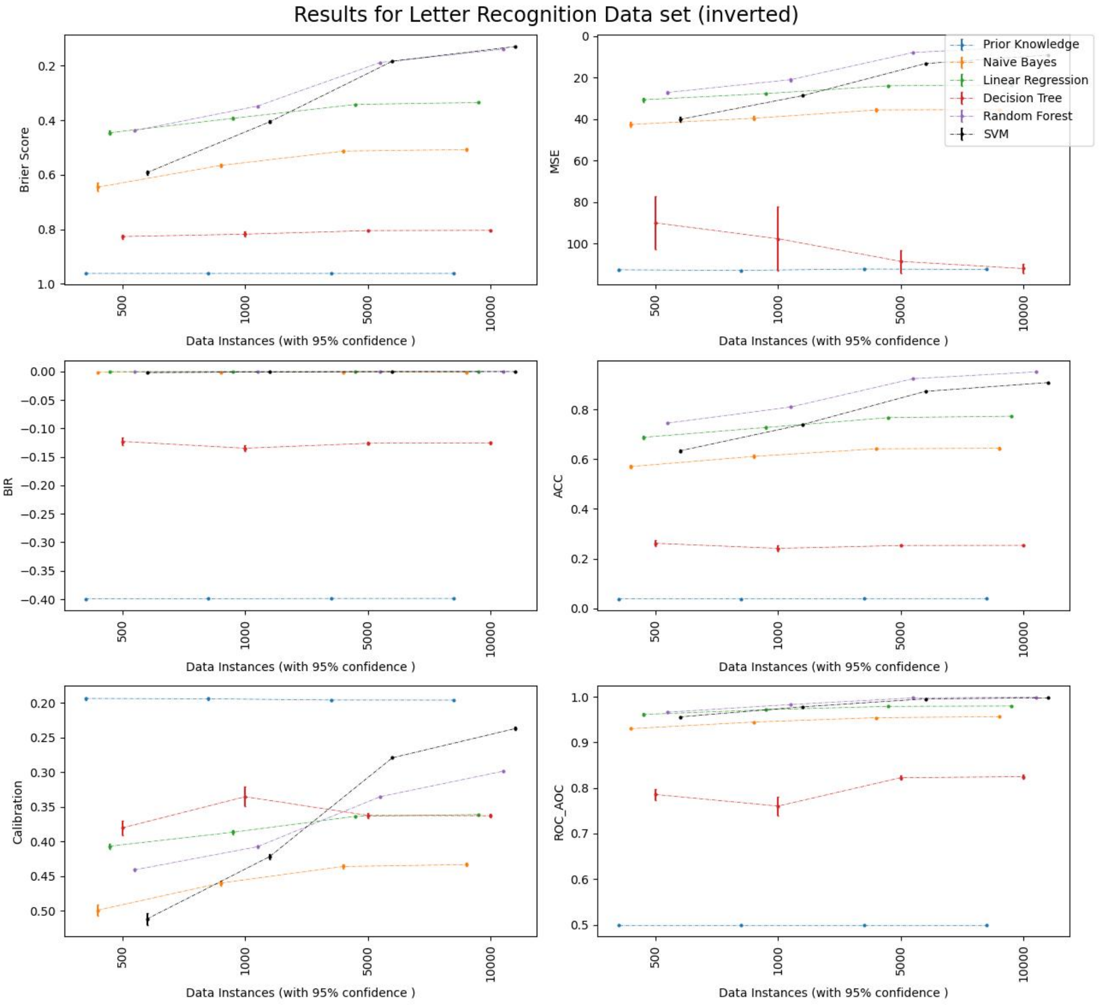

Figure A.2: Letter Recognition data set(changed after)