Comparative Analysis of Various Machine Learning Methods

Project Final Report

Team Members: Yee Chuen Teoh, Runlong Zhang

Abstract

This paper details the implementation of various machine-learning models that we have addressed in class. These include but are not limited to support-vector machines, and logistic regression. For each of the models we implement, we also incorporate various optimization techniques in order to increase the performance of corresponding models. Some of the techniques we will show include bagging, boosting, feature selection via entropy/information gain, feature space projection, hyper-parameter tuning, etc. Furthermore, we also make hypotheses on the correlation factor between the model, optimization techniques used as well as quality of the data. This paper aims to provide insight into the efficacy of different optimization techniques on various models and show their impact on performance gain. The efficacy of these models is assessed through their performance on a diverse range of datasets, encompassing predictions related to breast cancer, Alzheimer's disease handwriting, spam email classification, and water potability determination.

1 Introduction

There exist many novel machine-learning models for dealing with classification tasks, some of which include perceptron, support-vector machines, logistic regression, and so on. We aim to implement and explore the aforementioned models and various techniques which could potentially improve the models. Our experiments are conducted on datasets with notable differences to better accentuate the niches of each model. Our experimental setup for the models will include a base case for each model -this will be the model with some pre-defined hyper-parameters based on intuition- individual techniques applied to the base case, and then finally a combination of techniques to optimize the model performance. We expect that the base case of each model applied on each dataset will have sub-optimal performance on average and reasonable performance on easy -linearly separable- data.

Support-vector Machines or SVMs are a basic classification model in machine learning that attempts to maximize a margin around the decision boundary to make confident classifications. In this paper, we utilize SVM to classify various datasets such as water potability, spam emails, handwriting for Alzheimer's, and breast cancer. We explore the various properties of SVM, such as the Kernel function, C, etc. The datasets we utilized in our preliminary results are all tailored toward classification problems. As such, we expect the accuracy of our model to be around 90% with some minor tuning/optimization. However, this is highly dependent on the quality of the dataset as well as the linear separability of the dataset. Therefore, there is a possibility that the results may be lower than our expectations.

Logistic Regression is a linear classification model that leverages a logistic function, such as $sigmoid(z) = \frac{1}{1+e^{-z}}$, to characterize the likelihood of a particular class or event. The distinctive feature of the logistic regression model is its capacity to accommodate uncertainty in classification, allowing for output values between 0 and 1, which signify degrees of uncertainty. Logistic regression is extensively utilized across various industries for predictive and classification tasks, including applications such as fraud detection and disease prediction. This research project dives into an exploration of different logistic regression variants, including Binomial logistic regression (BLR) and Multinomial logistic regression (MLR), with the aim of classifying four diverse type of datasets including spam email, water potability, handwriting for Alzheimer's, breast cancer and more. The study further involves the identification of the best-performing logistic regression model, which will subsequently undergo a comparative analysis against alternative classification models.

The datasets utilized in our experiments -handwriting, breast cancer, water potability, and spam email- each offer something that the others do not. The handwriting data have 400+ features for each sample with only around 170 samples in total. The low sample count and high feature count of this dataset allows us to experiment with feature reduction as well as robustness of the model with limited input. The breast cancer dataset is the most balanced of the datasets, with around 600 samples and around 30 features; we expect this dataset to require the least amount of fine-tuning in order to achieve the best performance due to the dimensions of the dataset. The third dataset, water potability, contains samples with missing values. This presents the challenge of how we should handle these missing values in order to optimize performance. Lastly, the spam email dataset contains features which are purely text, this is unique in that we must perform data preprocessing and handle the text features before the data is usable in our models.

2 Related Work

This section represents some related works done in the research space related to comparative analysis of different machine learning techniques on their accuracy. In the paper [Bansal et al. (2022)], the authors provides a comprehensive analysis of five prominent machine learning algorithms: K-Nearest Neighbor (KNN), Genetic Algorithm (GA), Support Vector Machine (SVM), Decision Tree (DT), and Long Short Term Memory (LSTM) networks. It offers a detailed discussion on the origin, definitions, methodologies, and real-world applications of these algorithms, shedding light on their often-overlooked historical context. In the research, GA and LSTM is compared on their accuracy in daily tourist flow prediction, and comparison analysis is done between KNN, SVM and DT in student performance prediction. The paper assesses their accuracy, robustness, and reliability, with particular emphasis on the LSTM network and SVM algorithm, which exhibited superior performance. Additionally, the paper addresses key questions surrounding these algorithms and presents a thorough comparison of their capabilities. It also highlights the future prospects of machine learning and artificial intelligence, envisioning their roles in technological advancement and humanitarian domains.

In the next study [Shou-Tung et al. (2009)], the authors of the study investigate the diagnostic capabilities of Logistic Regression Analysis (LRA), Support Vector Machine (SVM), and Neural Network (NN) models within computer-aided diagnostic (CAD) systems for breast ultrasonography. Utilizing a database containing 3D power Doppler imaging data of both benign and malignant solid breast tumors, the research aimed to show on the potential applications of CAD systems, such as three-dimensional power Doppler imaging, vascularity evaluation, and solid mass differentiation. The analysis involved a comparative analysis of these models, measuring their diagnostic performance through the area under the Receiver Operating Characteristic (ROC) curve (Az values). The results revealed that, irrespective of whether non-harmonic or harmonic 3D power Doppler imaging was used, LRA, SVM, and NN demonstrated similar diagnostic performances. However, LRA displayed a slight advantage in sensitivity compared to the other models, offering valuable insights for the future development of CAD systems for breast ultrasonography. Although our study does not incorporate datasets involving 3D imaging, this paper provides valuable insights into the methodologies and techniques for conducting comparative analyses between logistic regression and SVMs.

In the last study [Widyahastuti et al. (2017)], in response to the growing emphasis on enhancing educational quality and students' learning outcomes, this study delves into the prediction of students' performance by analyzing their learning behavior. The primary objective is to offer predictions for students' final examination results using two distinct approaches: linear regression and multilayer perceptron, implemented in WEKA. This comparative analysis focuses on evaluating their accuracy, performance, and error rates to assess their practicality in educational contexts. The dataset is derived from the extraction and analysis of e-learning logs from discussion forums and attendance records. The findings reveal that multilayer perceptron outperforms linear regression in terms of predictive accuracy for final examination results, suggesting its potential superiority in aiding educators, school authorities, and relevant stakeholders in making informed policy decisions to enhance educational outcomes. In our research, we will be implementing linear regression and perceptron in Python instead of using WEKA. However, the insights provided by this paper regarding their expected performance and their significance in understanding learning behavior remain highly valuable to our study.

3 Methods

We employ a wide range of techniques in our experiment to gauge the performance of our model on the datasets. Some techniques are more trivial such as shuffling, and others are more complex like data reconstruction. While there isn't a set of rules defined on what we included for our methodology/techniques, the techniques we did include are expected to boost performance in at least some of our testing cases. We start with more trivial techniques in the beginning and we discuss more complex methods in the later parts of this section.

3.1 Data-parsing

Due to the nature of asymmetrical nature of our datasets, the datasets are first parsed into usable form with all feature values stored in a 2D matrix X, and y with all the labels. For the handwriting and breast cancer dataset, we were able to utilize all of the features and feature values provided in the dataset. The parsing for these two datasets were straight forward in that we only needed to transfer the information into a numpy array. The water potability dataset however, required more attention as some feature values do not exist. We handled these invalid values by replacing them with 0's inside our X matrix, with also the option to replace invalid values with medians or means of that feature column. Lastly, due to the complexity involved in parsing text and assigning them numerical representations, we decided to parse out the text-based feature columns in the spam email dataset as text parsing is not the focus of this experiment.

3.2 Shuffling

One challenge that exists across all models and datasets is the validity of the results from said model. While employing various techniques on different models may yield favorable results, we must ensure that the result we achieve can be achieved via the same parameters consistently. This is especially important in more unstable datasets or datasets of smaller sizes such as the handwriting dataset. This is because smaller datasets result in fewer training examples which are then in turn more prone to noise and outliers. By implementing shuffling to address consistency and robustness we can evaluate the true performance of the model under specific conditions and parameters. The implementation of shuffling is simple and straight forward, simply reposition vectors at certain indices to other indices while maintaining the same label.

3.3 Normalization

While feature values in our datasets often have meaning to an expert recording the data, those same meaning does not apply to a machine learning model. As such, recorded data can sometimes exhibit huge discrepancies between the ranges in the data values. An example would be one feature could have a value range of 1-10 within the dataset, and another feature could have a value range of 10,000-100,000 in the dataset. This extreme value range can sometimes be detrimental to obtaining an accurate model. In order to resolve this issue, we employ normalization on our data in order to squash the value ranges of features down to 0-1. This ensures that predictions done on the model will not have its result significantly altered by one extreme deviation of one feature value. The specific implementation for our normalization method is that we find the lowest value within a feature column, this value will be squashed down to 0, we then find the max value within the same feature column, and that value will be squashed down to 1, every other value in between will be calculated by the formula $\frac{v-min}{max}$, where v is the value in question, min is the min value of the feature column, and max is the max value of the feature column. Note that here we use the word "squash" loosely as "squashing" could potentially increase the value of a feature if the max value of a feature column is less than 1.

3.4 Data Reconstruction

Although predictions made with perfect information are often considered to be superior to predictions made with imperfect information, oftentimes we may find ourselves with imperfect information. An example of this in our experiments is the water potability dataset where there exist samples with some feature values missing. A naive approach to this may be to simply ignore the missing values by parsing them as 0's. However, we aim to provide the groundwork for an improved way of handling non-existing data values. We believe that by providing a more advanced way of handling missing values, we can improve our model's robustness as well as its general performance in terms of predictions. Our implementation of the data reconstruction method is as follows; we first construct the correlation matrix such that the matrix at row i and column j corresponds to the correlation between feature i and feature j. This calculation is done following the formula for Pearson's r. After which, we construct a "coefficient matrix" based on the mean values of each feature column. To go into more detail; the coefficient matrix at row i and column j is a real number representing the relative size of the mean of feature i compared to feature j. Once the correlation matrix and coefficient matrix are constructed, we iterate through our dataset and find missing values which are then reconstructed by finding a value -that exists within the same sample- with the highest correlation to the missing value and is then multiplied by the corresponding coefficient matrix value. Although there may be more complex and better ways to perform data reconstruction(such as utilizing the standard deviation of existing values and probability to approximate the missing value), we believe that the current implementation can serve as a baseline for future experiments.

3.5 Data Corruption

To keep our experiments uniform across all datasets, we introduced a novel technique to simulate data corruption on datasets that do not contain invalid or missing values. While this method in itself does not -and should not- help improve the model, this method can offer valuable insight into the performance of certain models and technique combinations when exposed to uncertainty regarding data integrity(ie. perhaps a certain technique may increase performance but in return make the model more susceptible to noise). To better simulate realism, our implementation includes a threshold parameter for the percentage of samples that are corrupted, a threshold parameter for how many values in each sample should be corrupted, and finally a boolean variable to introduce varying degrees or uniform degrees of corruption across samples(ie. one sample may be 20% corrupt, whereas another sample may only be 5% corrupt). We will see later in the experimental results section that, while our data corruption method provides a means to do uniform testing across our datasets, the method also has a unique side effect in being able to expose redundant feature space.

3.6 Kernel Functions - Radial Basis, Polynomial, Sigmoid

Traditionally used in SVM to promote linear separability in the dataset, we decided to implement and experiment with Kernel functions outside of SVM. Our reasoning behind this decision is that Kernel functions serve as a means to calculate the similarities between samples of the dataset. While this is typically employed in SVM to allow for better decision boundaries, the nature of the function is to simply transform data. Therefore, we believe that if Kernel function may promote increased performance in SVMs, then we may achieve similar results when applying Kernel functions to other models. As the name of this section suggests, our experiments included three variants of Kernel functions - radial basis, polynomial, and sigmoid. All three aforementioned Kernel functions follow the standard formula (RBF: $exp(-\frac{\|x_1-x_2\|^2}{2\sigma^2})$, Polynomial: $(x_1 \cdot x_2 + c)^d$, Sigmoid: $tanh(\gamma * x_1 \cdot x_2 + c)$) with adjustable parameters for γ, c, and d.

3.7 Ensemble Learning

Ensemble learning is an integral part of any machine learning experiment when dealing with performance and reliability. We incorporated both bagging and boosting in our experiments in hopes of bolstering both our model accuracy as well as reliability. In classification problems, ensemble learning becomes especially important when performing predictions on hard or non-linearly separable datasets. The implementation of our bagging method follows the unweighted scheme of training a set number k of weak classifiers -in parallel- all of which then cast an equal vote on the prediction of a sample. On the other hand, our boosting method utilizes the weighted scheme of implementation following the Adaboost algorithm addressed in lectures. Both bagging and boosting are implemented with adjustable parameters to set the number of weak models to train, the number of training samples for each model, and all sampling is done with replacement. Furthermore, we also introduced another hyperparameter in both bagging and boosting which will be addressed in the following section.

3.8 Offset

The additional hyperparameter we introduced in our ensemble methods is named ”offset”. This hyperparameter takes into effect during the prediction stage of our model and takes on a float value between 0.0 to 1.0. The function of this hyperparameter is to offset each weaker classifier's accuracy by the specified amount which is then multiplied by the prediction value of the corresponding weaker classifier. While this hyperparameter can take on any value between 0.0 to 1.0, our experiments mainly focused on an offset value of 0.5. The rationale behind this decision is that by having an offset of 0.5, weaker classifiers with accuracy closer to the offset value will have an exponentially lower impact on the final prediction of a particular sample. This hinges on the philosophy that models that perform exceptionally well(with enough training samples) during training time are much more trustworthy in their prediction during test time than models that are only correct some of the time, and thus, we would like to put a heavier reliance to arrive at a final prediction on models which perform well. Furthermore, an interesting -and unexplored-side effect of setting offset to 0.5 is that weaker classifiers that perform extremely poorly(less than 50% accuracy) would have their prediction vote inverted. This follows from an analogy that a student has learned the subjects of a class if either the student aces (gets 100% on) the exam(for the sake of argument assume the exam is true or false to mimic a binary classification problem) or the student gets 0% on the exam; both of these scenarios require the student to demonstrate an understanding of the material to achieve their respective scores. Similar to this, a weaker classifier that achieves only 10% accuracy could potentially be just as informative as a model with 90% accuracy come final prediction voting.

3.9 Feature Selection

One particular dataset, the Alzheimer's handwriting dataset, contains 400+ features that include features that are redundant and have minimal correlation with the class label. The dataset that contains a larger number of features compared to the size of the dataset is known to reduce ML model performance, namely the Logistic Regression technique that we will be conducting comparative analysis on. To reduce the dimension/feature size of the dataset, we implement feature selection using the filter method. Filter method calculates the correlation score of each feature and selects the top-k highest correlation feature in the dataset using: $p(x_k, y) = \frac{Cov(x_k,y)}{Var(x_k)Var(y)}$ where x_k represent the k^{th} feature and y is the class label. In our comparative analysis project, we have selected the top-k features where the $k = 0.2 * |datasets|$, i.e. 20% of the original dataset size, with a minimum of at least 10 features being selected. To analyze the effect of the feature selection technique, we have also decided to remove features with negative correlation.

3.10 Feature Reduction

Similarly with to the challenge as seen in subsection 3.9, feature selection. we utilize the feature reduction technique to reduce the feature dimension size while retaining the data information. Feature reduction is implemented with the principal component analysis (PCA) technique that is known to reduce the noise and find meaningful ways to flatten the data by focusing on the things that are different among features. PCA compure adjusts the original data X by the mean $X' = X - \overline{X}$, where $\overline{X}$ is the mean of the data X. Then compute the covariance matrix $S = \frac{1}{n-1} X'^T X'$. Calculate the eigenvectors and eigenvalues of the covariance matrix $S\alpha = \lambda\alpha$, where α denotes the eigenvectors and λ denotes the eigenvalues. pair the respective eigenvector and eigenvalue together, (α_j, λ_j) and sort the pairs by λ_j. lastly, select the top-k λ_j where k is the intended feature size to be reduced to, and transform the original data by doing for k iterations, $\forall x \in X'$, $x_{new} = \alpha_j * x$. Other than using the feature reduction technique to reduce the feature dimension size for model accuracy improvement, feature reduction is also utilized for data visualization to better understand of the data distribution in each dataset. Data visualization allows us to understand and explain the behavior and accuracy results from each model.

3.11 Hyperparameter Tuning

One challenge in the realm of machine learning often is using the optimal hyperparameter for ML models, choosing the optimal learning rate and iteration is important in achieving a balance in improving a model's accuracy with the lowest considerable training time in the actual dataset. Rather than brute force testing different learning rate with a different number of iteration that causes huge hyperparameter tuning time, the hyperparameter tuning function first tune the optimal learning rate, then tune the shortest number of iteration to achieve high accuracy. In hyperparameter tuning, we use a cross-validation technique, which splits the training set into 5 folds, wherein each split, we use one fold as the validation set, and the other 4 as the training set. for each learning rate and iterations hyperparameter, we train the models 5 times, each with a different fold as a validation set, with the other 4 as a training set, and average the 5 split's accuracy. At the end, we select the learning rate and iteration with the highest average 5 split validation accuracy score.

3.12 Data Visualization

As mentioned in subsection 3.10, we use a feature reduction technique for data visualization for all 4 datasets. To rationalize and explain the behavior and result from our SVM and LR models, visualize Alzheimer's handwriting, breast cancer, spam email, and water potability datasets in a 2D scatter plot and 3D scatter plot to see if these data are separable in the 2D plane of separable in a higher dimensional plane such as 3D plane.

4 Experimental Setups

In investigating the difference between SVM, BLR, and MLR in comparing their accuracy. We have trained SVM, BLR, and MLR on all 4 datasets with the same feature, same training set, and testing set, with hyperparameter learning rate = 0.001 and number of iterations = 100 as the base case. After the base case, we implement each technique to SVM, BLR, and MLR to see how each technique affects the accuracy of these models, specifically, we implement the techniques below before testing:

- normalization of dataset

- kernelization (Polynomial, Radial Basis, Sigmoid)
- feature selection (20% of original dataset size, minimum at 10 features)
- feature reduction (dimension reduce to 2)
- shuffling (add randomization to the order of data, resulting random distribution of class in training and testing set)
- Ensemble Learning (Bagging and boosting, both with and without offset, total of 4 implementation)
- Hyperparameter tuning
- Data reconstruction

With a total of 14 baseline accuracy scores, including base case. In the end, we analyze a different combination of the different techniques that have shown promising improvement (i.e. feature reduction, ensemble learning) and select the best-performing combination to compare with the baseline accuracy score and to be compared against the other models. To improve the reliability of the result, we added 2 constraints to the combination phase. first, we must shuffle the dataset 3 times, to ensure randomness, second, we test these model 10 times, each with 3 shuffling, and get the average accuracy score as well as their variance in accuracy.

the combination of techniques selected for SVM, BLR, and MLR models on each dataset are as below, note that all model's combination implements shuffling and normalization:

Support Vector Machine:

- Alzheimer's handwriting: Bagging with 0.5 offset and 1000 weak models, margin size = 0.1.
- Breast cancer: feature reduction ($k = 2$), Bagging with 0.5 offset, margin size = 0.1.
- Spam email: Data reconstruction, radial basis kernelization, feature reduction ($k = 2$), boosting with 0.5 offset, margin size = 0.1.
- Water potability: Feature reduction ($k = 2$), boosting with 0.5 offset, margin size = 0.1.

Binomial Logistic Regression:

- Alzheimer's handwriting: Feature reduction ($k = 2$), bagging with 0.5 offset.
- Breast cancer: Feature reduction ($k = 2$), hyperparameter tuning.
- Spam email: Polynomial kernelization.
- Water potability: Polynomial kernelization, feature reduction ($k = 2$), bagging with 0.5 offset.

Multinomial Logistic Regression:

- Alzheimer's handwriting: Feature selection ($k = 20\%$), feature reduction ($k = 2$), hyper parameter tuning.
- Breast cancer: Feature reduction ($k = 2$), hyperparameter tuning.
- Spam email: Data reconstruction, feature reduction ($k = 2$), bagging with 0.5 offset.
- Water potability: Polynomial kernelization, feature reduction ($k = 2$), bagging with 0.5 offset.

5 Experimental Results

The experiment result shows the performance of SVM, BLR, and MLR within the baseline constraint and when implemented with different combinations of techniques. The results show improvement in model accuracy performance when implemented with a combination of techniques, except interestingly the BLR on the breast cancer dataset and spam email dataset, which have shown a decreased accuracy of approximately 10% accuracy for the breast cancer dataset. SVM and MLR are on par in terms of accuracy performance. SVM shows a slight advantage in the Alzheimer's handwriting dataset by about 4% and the water potability dataset by about 1%. MLR showed a slight advantage in the breast cancer dataset by about 1% and spam email by about 2%. Subsection 5.1, 5.2.1, 5.2.2 show the results of the model's accuracy score in a bar chart plot.

5.1 Support Vector Machine (SVM)

The experimental results of the SVM model on Alzheimer's handwriting, breast cancer, spam email, water potability, and combination comparison respectively can be visualized as follows:

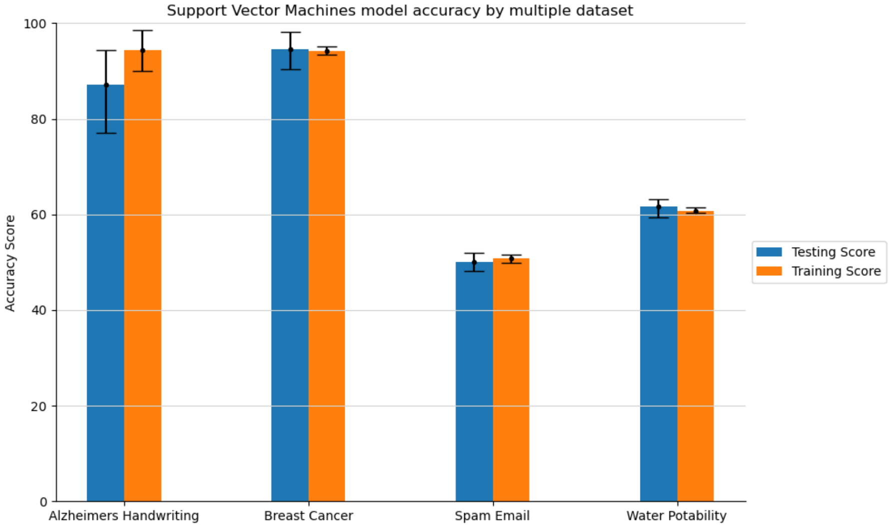
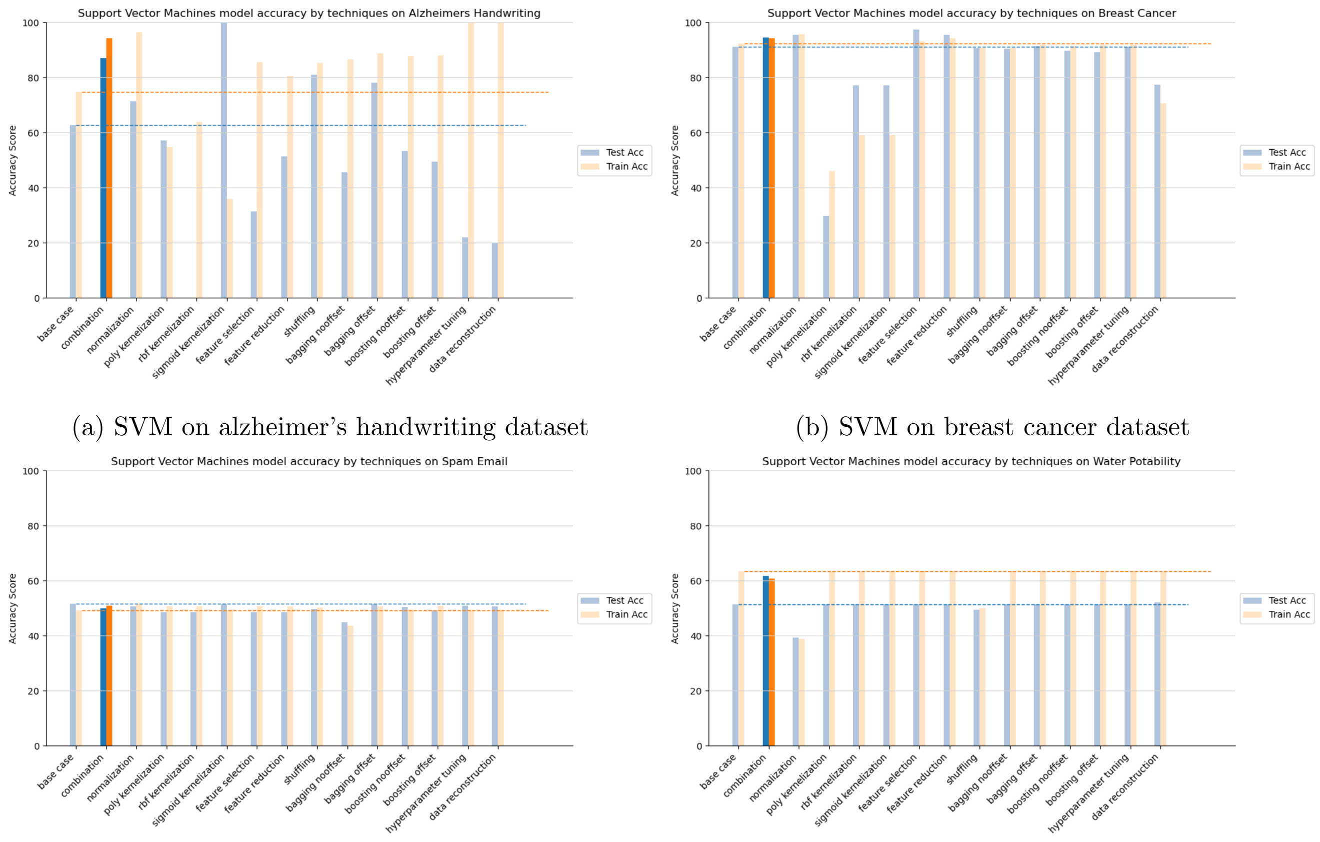


Figure 1: Barchart plot for SVM accuracy on combinations against basecase and all techniques



(a) SVM on alzheimer's handwriting dataset

(b) SVM on breast cancer dataset

(c) SVM on spam email dataset

(d) SVM on water potability dataset

Figure 2: Barchart plot for SVM accuracy on combinations against basecase and all techniques

Observing the above results, it is easy to see that the base case of the SVM model applied to the dataset generally performs sub-optimally. However, with a few minor adjustments in the form of hyperparameter tuning and combination techniques, we can generally increase our model performance with varying degrees of success. What is more interesting, however, is our experimental result for the water potability and spam email dataset. It can be observed our results for the water potability and spam email dataset -regardless of techniques utilized- do not get much better than random guessing(50% accuracy). While our initial intuition is that our selection of techniques or parameters were simply suboptimal, we will discuss later that the reason behind these results may be more nuanced.

5.2 Logistic Regression

5.2.1 Binomial Logistic Regression (BLR)

The experimental results of the BLR model on Alzheimer's handwriting, breast cancer, spam email, water potability, and combination comparison respectively can be visualized as follows:

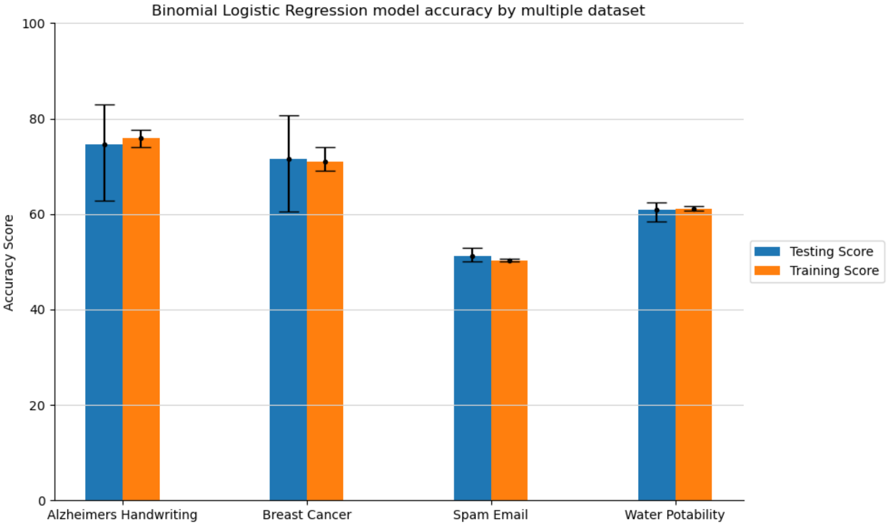
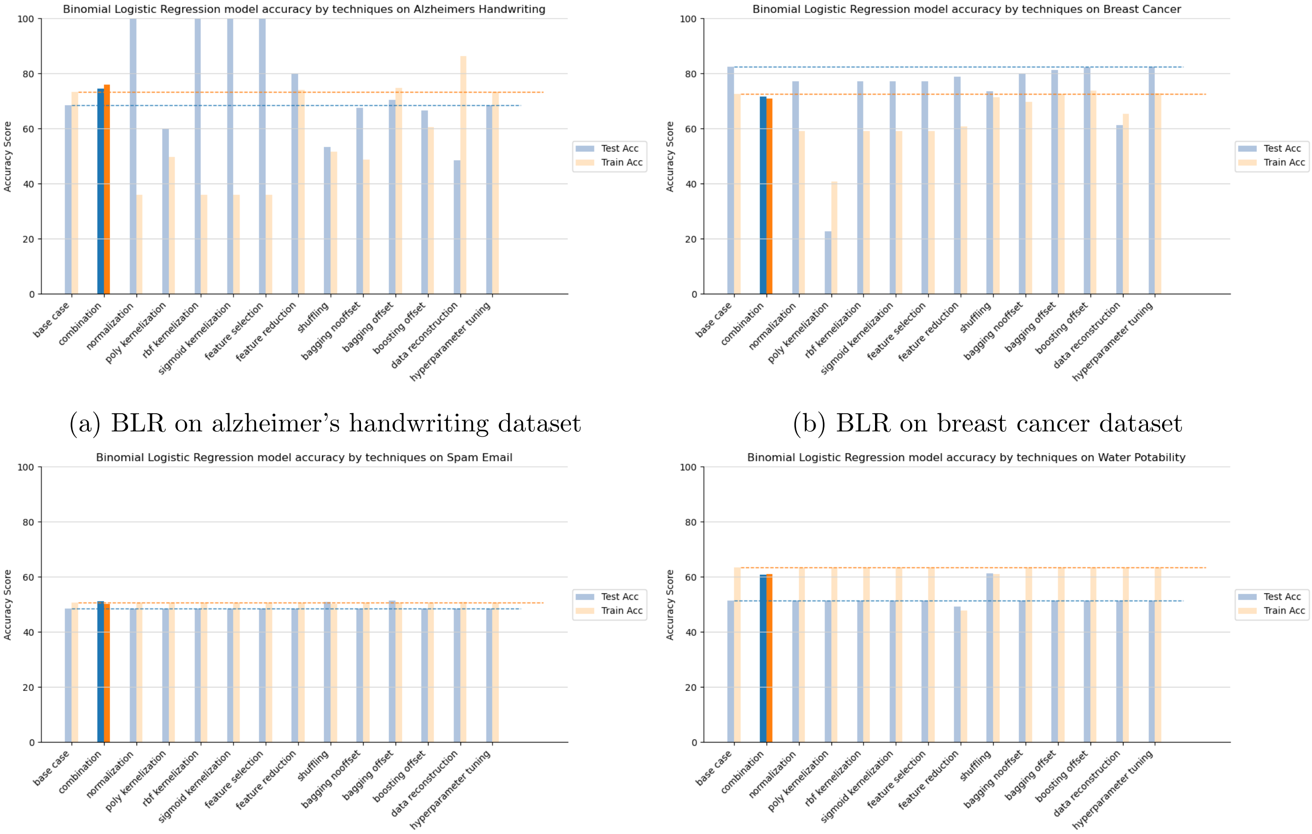


Figure 3: Barchart plot for BLR accuracy on combinations against basecase and all techniques



(a) BLR on alzheimer's handwriting dataset

(b) BLR on breast cancer dataset

(c) BLR on spam email dataset

(d) BLR on water potability dataset

Figure 4: Barchart plot for BLR accuracy on combinations against basecase and all techniques

Looking at the results shown by BLR, specifically on the Alzheimer's handwriting dataset, we can see a few techniques that produce 100% accuracy. However, this is not because the technique is outstanding, but because of the data distribution in the Alzheimer's handwriting dataset, causing the testing dataset to only contain -1 class label, and due to the huge variance in the value and a high number of feature, we have seen the probability prediction to be very low and causes value to nan value, causing the model to automatically predicts -1 due assuming the probability of 0% being class 1 label. Note that the result of shuffling is not as reliable because it only shuffles the data in the dataset. Noticed that interestingly BLR with a combination of techniques has shown reduced accuracy performance on breast cancer datasets.

5.2.2 Multinomial Logistic Regression (MLR)

The experimental results of the MLR model on Alzheimer's handwriting, breast cancer, spam email, water potability, and combination comparison respectively can be visualized as follows:

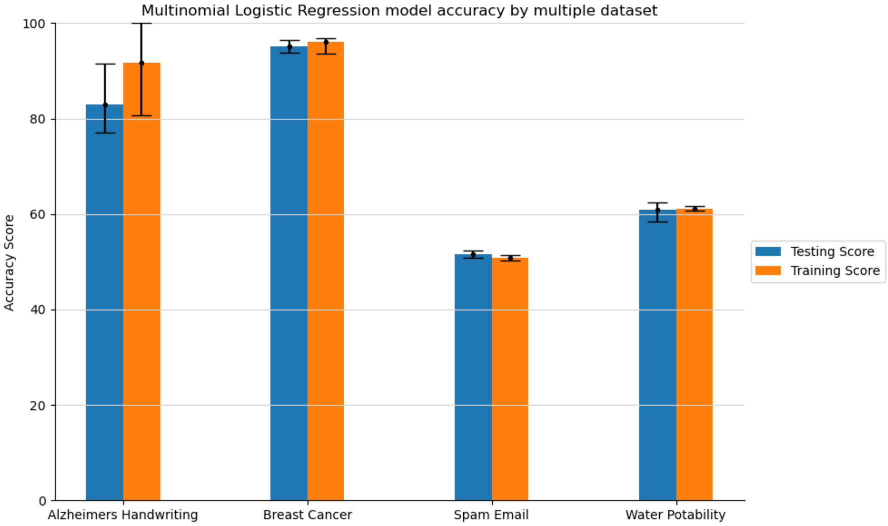
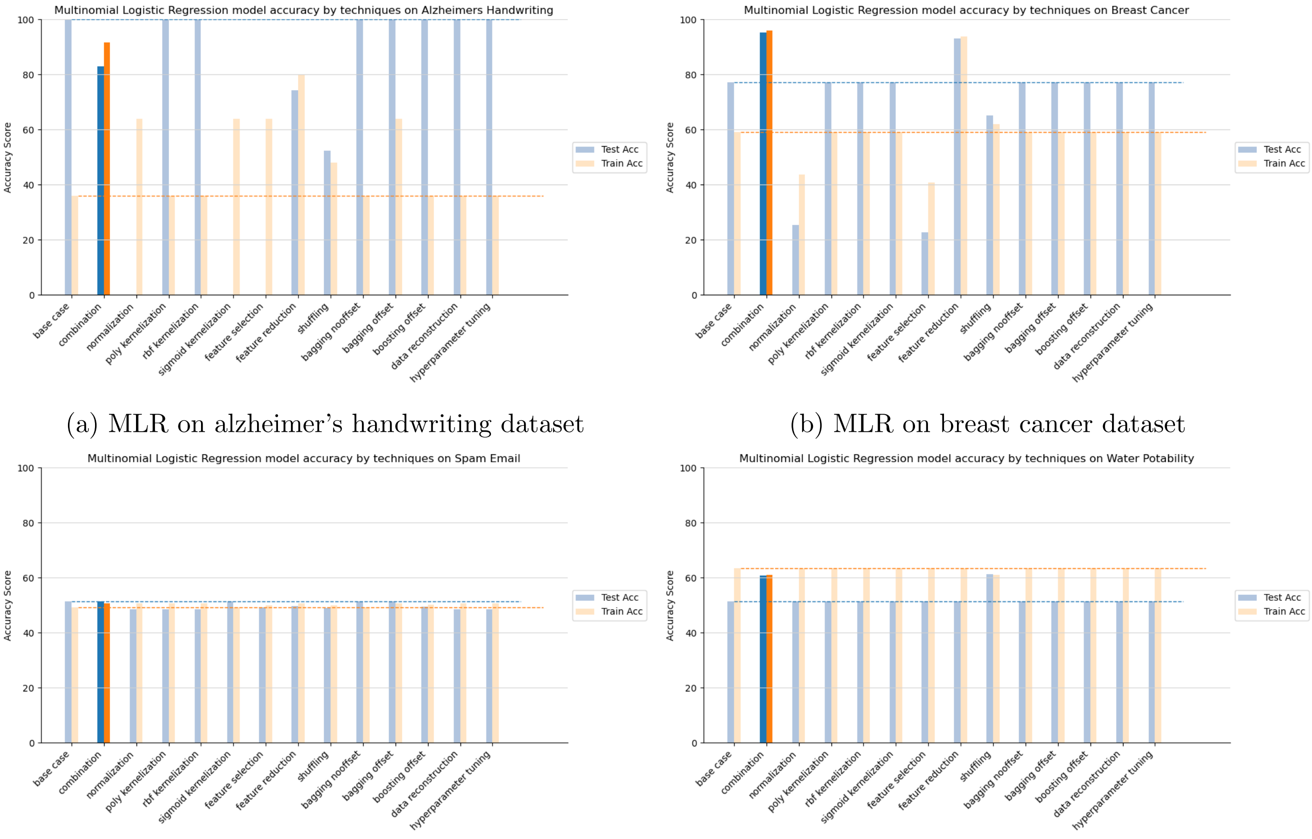


Figure 5: Barchart plot for MLR accuracy on combinations against basecase and all techniques



(a) MLR on alzheimer's handwriting dataset

(b) MLR on breast cancer dataset

(c) MLR on spam email dataset

(d) MLR on water potability dataset

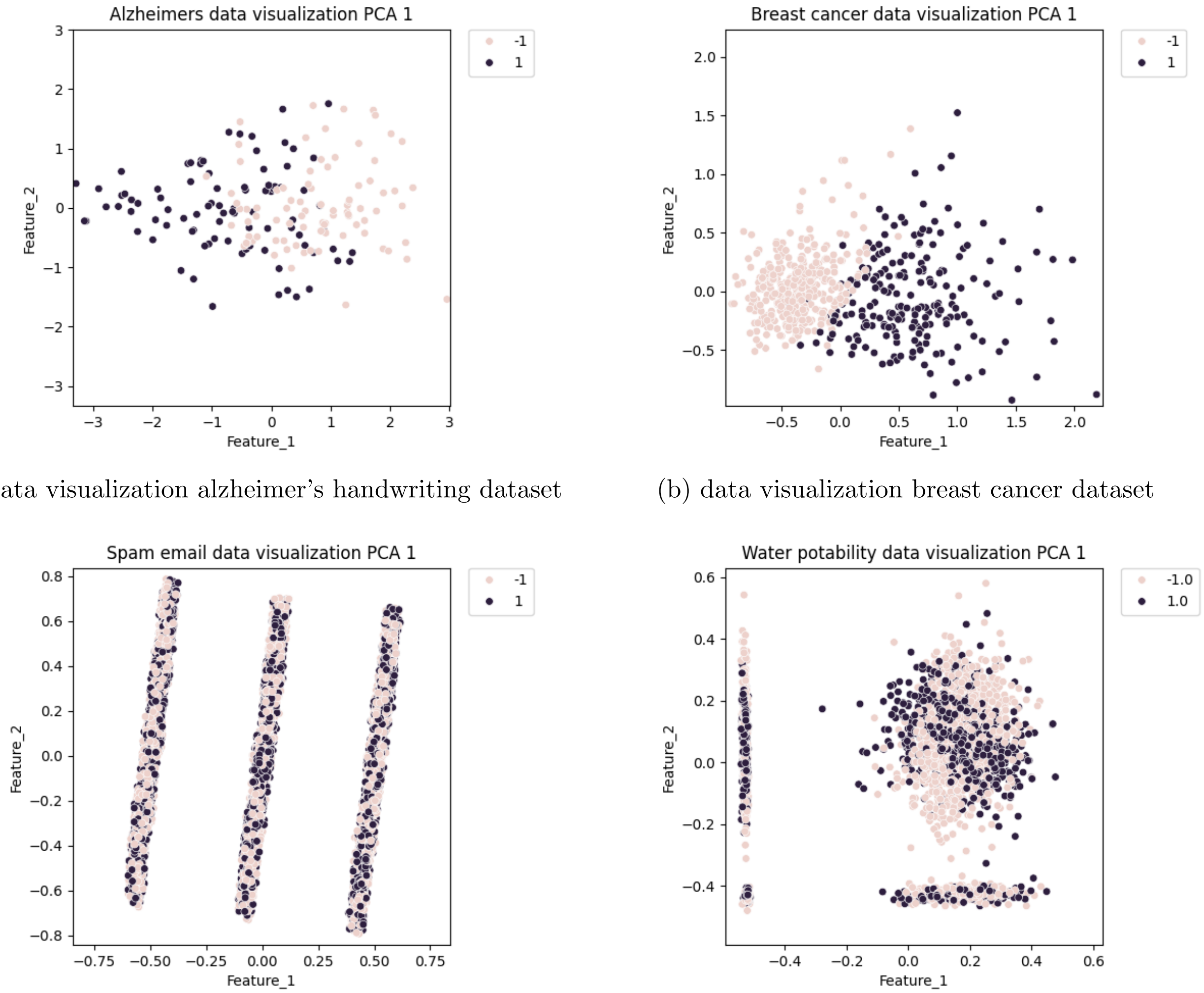
Figure 6: Barchart plot for MLR accuracy on combinations against basecase and all techniques

The result from MLR has shown much better prediction in comparison to BLR in all 4 datasets, our theory is because MLR behaves in terms of prediction of the class label by comparing the probability score for class 1 and -1 label, rather than in BLR, the prediction is class 1 if the probability is 0.5, else class -1 otherwise. Meaning if MLR predicts 0.4 on class 1 and 0.2 on class -1, MLR will select class 1 whilst if BLR predicts 0.4, it will predict class -1. Observing the Alzheimer's handwriting dataset, MLR contains the most 100% and even 0% prediction, which again due to the distribution of the dataset and MLR only predicts class 1 or class -1 label. a similar case is seen in the breast cancer dataset, where the prediction score using normalization and feature selection shows the inverse accuracy score from the others. It is interesting to note that just by feature reduction alone, MLR can achieve close prediction scores to combination techniques in the breast cancer dataset.

5.3 Data Visualization

In our experiment result as shown in subsection 5, we can see that there is minimal improvement in the spam email and water potability dataset. To explain why all 3 models are unable to cross the 70% accuracy score for the 2 datasets, we have visualized the 4 datasets in 2D and 3D scatter plots, to see the data points distribution, mainly in investigating if the data is separable or not, as well as linear or not.

The data visualization for Alzheimer's handwriting, breast cancer, spam email, and water potability is shown below:



(a) data visualization alzheimer's handwriting dataset

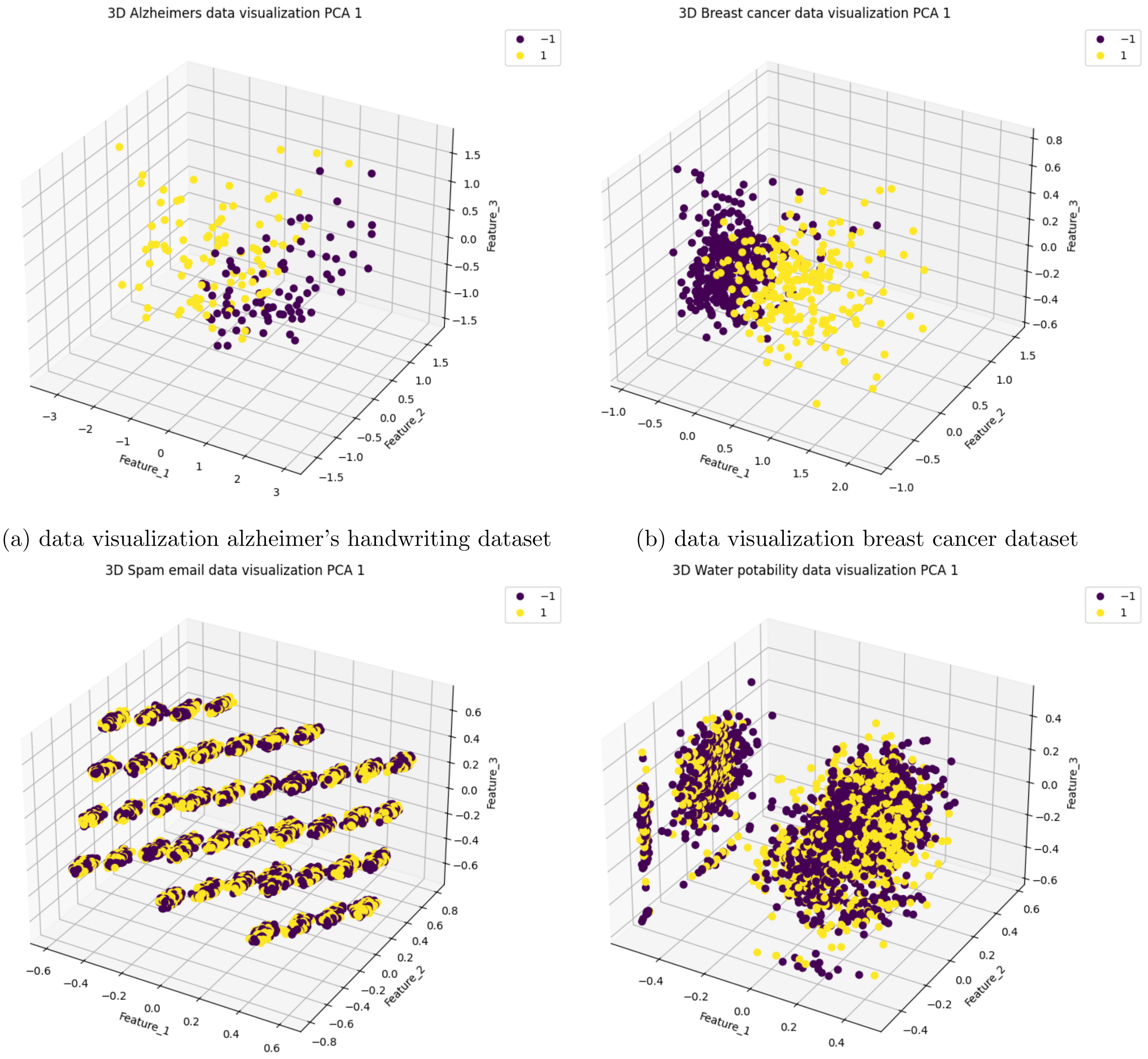
(b) data visualization breast cancer dataset

(c) data visualization spam email dataset

(d) data visualization water potability dataset

Figure 7: 2D data visualization for 4 dataset

As shown in the 2D data visualization, the Alzheimer's handwriting and breast cancer datasets are nonseparable linear datasets, spam email, and water potability datasets are nonseparable nonlinear datasets where classes are clustered together, hence the reasoning why all 3 models, SVM BLR, and MLR is unable to accurately predict spam email and water potability dataset. Similarly in a 3D data visualization scatter plot below:



(a) data visualization alzheimer's handwriting dataset

(b) data visualization breast cancer dataset

(c) data visualization spam email dataset

(d) data visualization water potability dataset

Figure 8: 2d data visualization for 4 dataset

6 Future plans

While we investigate many techniques in our experiments there are still many other areas we would like to explore. The first and most obvious is the experimentation with other models as well as multi-class classification problems. While we did investigate multinomial logistic regression, we had simply applied it to a binary case rather than a true multi-class context.

In addition, we would like to refine the data reconstruction method we introduced. The current implementation of the method utilizes a rather naive approach to reconstruct missing data, our current theory is that by utilizing probabilities to approximate the best feature to use for reconstruction, we would be able to achieve a more realistic reconstruction of the data.

Furthermore, due to the problems arising from experimentation with the handwriting dataset, we would also like to investigate data augmentation in numerical data. While there exists Gaussian generation for numerical data, we would like to pursue a less computationally expensive yet effective approach to data augmentation.

One area we did not focus on during our experiments is computational complexity analysis. We noticed that throughout experimentation and as we introduced more methods to apply to our model, the computational time to train a model and make predictions increased exponentially. By analyzing the complexity of our methods and making refinements, we would be able to streamline our experiments as well as make comparisons in the tradeoff of performance vs speed.

Purely based on speculatory grounds, we name another one of our interests ”partial-data models”. The idea behind this is for certain datasets we want to make multiple models with the dataset partitioned in such a way that each model only trains a specific value range of samples. For example, if feature 1 has a value range of 1 to 10, then we could potentially train 3 models; one model for a range of < 4 another model for $4 \geq f \geq 7$, then the last for > 7. The intuition behind this is that certain datasets may exhibit more erratic or anomalous behaviors in certain value ranges. By partitioning the dataset into multiple value ranges, we could potentially decrease the complexity of the model, and decrease computational cost. In addition, if certain value ranges do not have a boundary to separate the classes of a dataset, then the overall performance of the model is only negatively affected by a particular partition of the data, which in turn we could then focus purely on improving the model for that specific partition.

Lastly, we want to investigate something we call ”smart drop-out”. The idea is that certain samples may contain feature values that are not within their ”normal” range, and because of this, predictions on such data points are prone to error. We would like to investigate methods for detecting these anomalous data points and decide if it would be beneficial to drop certain feature values during prediction time. This idea stems from our experimental result that the performance of a model could potentially increase when applying random corruption to the dataset. While this is not explicitly documented in our paper, we noted this during our experiments with the breast cancer dataset on the SVM model.

7 Conclusion

From the experimental result, it is shown that SVM was able to perform the best in the base case while MLR is on par with SVM when implemented with a combination of different techniques. BLR is somehow unable to perform well in Alzheimer’s handwriting and breast cancer datasets. Multinomial Logistic Regression performs better than Binomial Logistic Regression compared to MLR and SVM. SVM was shown to have a slight advantage over BLR and MLR in Alzheimer’s handwriting and water potability, while MLR was shown a slight advantage over BLR and SVM in breast cancer and spam email datasets. In conclusion, different models perform best with different combinations of techniques where each model needs to implement different combinations of techniques to achieve the highest accuracy. In addition, we can show that feature reduction is the top technique that almost increases all model performance for all datasets as well as ensemble learning (bagging & boosting) that has also been shown to improve performance in combination of techniques to non-separable datasets.(source code for models and technique implemented is available at https://github.com/YeeChuen/Comparative-analysis-of-various-machine-learning-method)

References

Bansal, M., A. Goyal, and A. Choudhary (2022). A comparative analysis of k-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. *Decision Analytics Journal 3*, 100071.

Shou-Tung, C., H. Yi-Hsuan, H. Yu-Len, K. Shou-Jen, T. Hsin-Shun, and C. D.-R. Wu Hwa-Koon (2009). Comparative analysis of logistic regression, support vector machine and artificial neural network for the differential diagnosis of benign and malignant solid breast tumors by the use of three-dimensional power doppler imaging. *kjr 10*(5), 464–471.

Widyahastuti, Febrianti, and T. V. Utami (2017). Predicting students performance in final examination using linear regression and multilayer perceptron. *2017 10th International Conference on Human System Interactions (HSI)*, 188–192.