COMP90055

Conventional Research Project

25 Credits

Machine Learning for Fertility Prediction

Author: Yiming Zhang

Student ID: 889262

Supervisor: Dr. Patrick Pang

University of Melbourne

School of Computing and Information Systems

This thesis is submitted for the degree of

Master of Information Technology (Computing)

Abstract

Fertility is a natural capability to produce offspring. However, if a patient has received particular cancer treatment, the patient may have infertility problems. Prediction of fertility status can help the cancer patients and the hospital. In this project, six machine learning algorithms K-Nearest Neighbour (KNN), Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF) and Artificial Neural Network (ANN) were used to predict the fertility status of the patient. I used Chi-squared based feature selection method to evaluate the impact of the feature selection to the evaluation results. And the results showed that feature selection only improved the performance of a few algorithms such as K-Nearest Neighbour and Artifical Neural Network, but not have an impact on other algorithms. A 10-fold cross-validation process had been applied to simulate the practical usage of prediction. Moreover, I evaluated the performance of the algorithms by accuracy, precision, recall, F1 score, and AUC. Learning curves were also plotted to analyze the performance of the algorithms. From the plots, they showed that K-Nearest Neighbour algorithm, Logistic Regression algorithm, and Support Vector Machine algorithm were all good-fit. Naïve Bayes algorithm was underfitted. Artificial Neural Network algorithm and Random Forest algorithm were overfitted. The best performance algorithm among the six machine learning algorithms was K-Nearest Neighbour, which achieved the highest accuracy of 0.7662, highest precision of 0.7752, highest recall of 0.7662, and highest F1 score of 0.7557.

Keywords: fertility prediction, machine learning

Declaration

I certify that

- this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university, and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.
- where necessary I have received clearance for this research from the University's Ethics Committee and have submitted all required data to the School
- the thesis is 6407 words in length (excluding text in images, table, bibliographies, and appendices).

Yiming Zhang

June 10th, 2019

Acknowledgments

I want to thank my supervisor, Dr. Patrick Pang, for the patient guidance, encouragement, and advice he has provided throughout my project. I have been fortunate to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

Table of Contents

List of Tables

List of Figures

1. Introduction

In the Introduction section, I discuss the background of the fertility problem first. Then I state the research problem as well as the aim and the scope of the project. Finally, I briefly illustrate the overview of the project.

1.1. Background

Fertility is the ability to have children, and infertility is the inability of a person to reproduce by natural means. Many common factors may affect fertility, such as age, weight, smoking, and other health issues. In particular, cancer and its treatment may cause fertility problems (Jenny, 2016). For instance, during the oncological treatment, the radiation to the brain and pelvis may cause high levels of damage to germ cells and affect reproductive hormone production (Yifan et al., 2018).

1.2. Problem Statement

The research problem of this project is to predict fertility probability for cancer patients. Moreover, in this research project, machine learning algorithms should be applied to predict the fertility probability. After using the machine learning algorithms, the performance of the machine learning algorithms should be compared, and the best performance machine learning algorithm should be selected.

1.3. Aim and Scope

This research project aims to use the cancer patient's dataset to train machine learning models, and then use the models to predict the fertility probability for the cancer patient. More specifically, six machine learning algorithms were applied to train and build the model, namely, K-Nearest Neighbour (KNN), Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF) and Artificial Neural Network (ANN). Moreover, the performance of these algorithms was analyzed and compared to find the most effective algorithm.

1.4. Overview of the Project

The experiment of the project consists of five parts, which are data preprocessing, feature selection, machine learning model training, machine learning model evaluation, and plotting learning curves. In the data preprocessing component, I pre-processed the raw original dataset. The preprocessing procedures include dropping empty data, converting categorical data to numerical data, and feature

scaling. In the feature selection part, the Chi-square test was used to select top K features. In the machine learning part, six machine learning algorithms were applied to train the classification model. In the evaluation part, evaluation metrics such as accuracy, precision, recall, F1 score, AUC were used to evaluate the performance of the model. Finally, learning curves were plotted to assist analysis of the model.

2. Literature Review

I have read and analyzed some journal articles and conference papers in the public health area, which is related to health informatics and machine learning. In this section, I illustrate some associated works on the public health area first and then describe some commonly used machine learning algorithms.

2.1 Related Work in Public Health Area

In the field of public health area, there are many applications of applying machine learning techniques for disease prediction. Some recent related journals and conference papers are mentioned below.

Jahidur Rahman Khan, Srizan Chowdhury, Humayera Islam, and Enayetur Raheem wrote the journal article "Machine Learning Algorithms to Predict the Childhood Anemia in Bangladesh" (Jahidur et al., 2019). In this article, six machine learning algorithms such as K-Nearest Neighbour (k-NN), Logistic Regression (LR), Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), Support Vector Machines (SVM) and Random Forest (RF) were used to predict the childhood Anemia status. The article then evaluated these algorithms by using evaluation metrics such as accuracy, sensitivity, specificity, and area under the curve (AUC). The best prediction algorithm among these six algorithms was random forest, which achieved the best classification accuracy of 68.53% with a sensitivity of 70.53%, specificity of 66.41% and AUC of 0.6857.

The research article "Application of Machine Learning Techniques for Clinical Predictive Modelling: A cross-sectional study on Non-alcoholic Fatty Liver Disease in China" was written by the Han Ma, Cheng-fu Xu, Zhe Shen, Chao-hui Yu, and You-ming Li (Ma et al., 2018). In this article, 11 state-of-the-art machine learning techniques were used includes Logistic Regression (LR), K-Nearest Neighbour (k-NN), Support Vector Machine (SVM), Naïve Bayes (NB), Hidden Naïve Bayes (HNB), Bayesian Network (BN), Decision Trees (C4.5), AdaBoosting, Bagging, Random Forest (RF) and Aggregating One-dependence Estimator (AODE). The authors computed accuracy, precision, recall,

specificity, and F-measure to evaluate the performance of the algorithms. Among all 11 algorithms, Logistic Regression (LR) demonstrated the best accuracy performance which achieved the highest accuracy 0.8341, SVM achieved the best classification precision of 0.725, Aggregating One-dependence Estimator (AODE) had the best recall of 0.680 and Bayesian Network (BN) reached the best F-measure value of 0.655.

Natalia Khuri and Shantanu Deshmukh wrote the conference paper "Machine Learning for Classification of Inhibitors of Hepatic Drug Transporters" (Natalia & Shantanu, 2018). In this article, the authors applied five machine learning algorithms, namely, K-Nearest Neighbour (K-NN), Partial Least Square (PLS), Support Vector Machine (SVM), Random Forest (RF) and Recursive Neural Network (RNN). The authors measured performance of these algorithms by accuracy, sensitivity, specificity, AUC, precision, and F1 score. The random forest method outperformed other methods and achieved the best performance.

2.2 Machine Learning Algorithms

Machine learning algorithms are model-free methods that provide efficient solutions to the classification problem. Many popular machine learning algorithms are widely used in the public health area such as K-Nearest Neighbour, Logistic Regression and Random Forest. In the following, I briefly describe each algorithm considered in this project.

2.2.1 K-Nearest Neighbour

K-Nearest Neighbour is an algorithm that builds the model only consists of storing the training data, and the algorithm finds the closest data points in the training set to be the prediction for the new data (Andreas & Sarah, 2016). K-NN is a non-parametric algorithm because it makes no explicit assumptions about the data distribution. In K-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbours, with the object being assigned to the class most common among its K-Nearest Neighbours.

2.2.2 Logistic Regression

Logistic Regression is the statistical method in classification problems. It measures the relationship between the predictor variables and the outcome variable (Jahidur et al., 2019). Also, it is the most widely used statistical method in classification problems in the public health area. Logistic Regression provides the probability for predicting the classes of categorical outcome variable by using given the set of predictors.

2.2.3 Naïve Bayes

Naïve Bayes classifier is based on applying Bayes's theorem with naive independence assumption. Naïve Bayes is attractive as it has an explicit and sound theoretical basis which guarantees optimal induction given a set of explicit assumption (Sona et al., 2013). Abstractly, Naïve Bayes is a conditional probability model, and it assumes that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities. In a classification problem, it is sufficient to predict the most probable class given a test observation (Sona et al., 2013).

2.2.4 Support Vector Machine

Support Vector Machine is a kernel based supervised machine learning technique and was originally proposed by Boser, Guyon, and Vapnik in 1992 (Jahidur et al., 2019). It assumes the data is linearly separable and aims to find a linear hyperplane (decision boundary) that will separate the data. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a linear hyperplane. New examples are then mapped into that same space and predicted to belong to a category based on which side of the hyperplane it is located. For non-linear SVM, a kernel function is needed. A kernel function in SVM is a function that equivalent to an inner product in some feature space, it implicitly maps data to a high-dimensional space.

2.2.5 Random Forest

Random Forest is an ensemble learning method for classification by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. To classify a new individual, features from this individual are used for classification using each classification tree in the forest (Jahidur et al., 2019). Each tree is built on a random subset of features of data, and the decision is made via the majority votes over all the trees in the forest.

2.2.6 Artificial Neural Network

Artificial Neural Network is inspired by the biological neural structure of the brain. Typically, each ANN consist of hundreds of simple processing units which are wired together in a complex communication network then collectively learn complex functions. Each unit or node is a simplified model of real neuron which sends off a new signal or fires if it receives a sufficiently strong Input signal from the other nodes to which it is connected (Sonali et al., 2014). Given sufficient training data, then an ANN can approximate very complex functions mapping raw data to output decisions. A typical ANN contains three layers: the input layer, hidden layers, and the output layer.

3. Experiment Design

Experiments were completed on Intel Core i7-8850H processor (2.60GHZ) running macOS with 16.0G memory. The following software and packages were installed: Python 3.7.2 (Van et al., 1995), anaconda-client 1.7.2 (Anaconda Software Distribution, 2017), Jupyter Notebook 1.0.0 (Fernando & Brian, 2007), Numpy 1.15.1 (Chris et al., 2011), Pandas 0.23.4 (Wes et al., 2010), Scikit-learn 0.19.2 (Fabian et al., 2011).

The experiment pipeline of the project includes five processes, which are data preprocessing, feature selection, model training, model evaluation, and plotting learning curve. The overall structure of the experiment was shown in Figure 1 below.

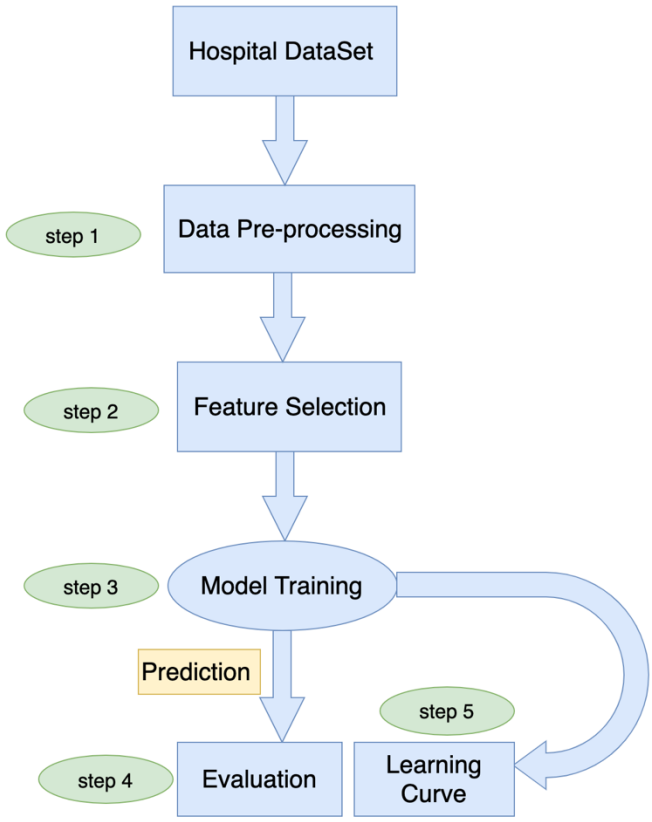

Figure 1: Overall pipeline for Experiment

3.1. Dataset Description

The dataset of this project sources from the hospital, it contains 7476 instances and 134 features such as age, smoking status, alcohol status, etc. The target feature was whether the patient still experiencing amenorrhea after 3/4/5 years of the start of chemo. However, many instances did not have the target feature and should be removed from the dataset as they are not labelled. After removing the instances without the target feature, the remaining dataset only contains 992 valid instances. The value for the

target feature is either 1 or 0, and if the value equals 1, it indicates that the patient experienced amenorrhea after 3/4/5 years of the start of chemo; otherwise, it equals 0. The mean value for the target feature was 0.738911, which means among the 992 valid instances there were 733 instances' target value equals 1 and 259 instances' target value equals 0.

3.2. Data Preprocessing

As there were some missing values in the dataset and some features were categorical data. Hence, data preprocessing is needed to resolve these issues. In this project, I used the Python Pandas library, which can transform the CSV file to data frame format, and it is easier and faster to apply data preprocessing on data frame format.

The first step of the data preprocessing is to divide the dataset into two parts. One part is the target feature dataset that contains labels (ground true) for each instance. Another part is the training and testing dataset. Also, as some features are related to the amenorrhea and the feature amenorrhea is strongly associated with target feature, hence these features need to be removed.

Then, by observing the dataset, it indicated that there were many missing data. To resolve missing data issue, two methods were found, which are deletion method and imputation method (Jiang et al., 2015). The deletion method is to delete the attribute if there are many empty data in this attribute, but this will lead to loss of information. The second method is to calculate the mean value of the feature and replace the missing value with the mean value. In this project, I decided to use both methods. I set a threshold of the number of non-empty values for each feature to 3, and if it is smaller than 3, then I just dropped this feature. Besides, for other features that have missing values, if the feature data type is float type, I used the second method which is to calculate the mean value and then used the mean value of the feature to replace the missing values.

To resolve the categorical data issue, I encoded the categorical data from categorical value to the numerical value. There are some methods and libraries in python that can achieve it. For instance, the "LabelEncoder" and "OneHotEncoder" methods in Scikit Learn library can accomplish the task (Amitabha, 2018). Also, Pandas library "get_dummies" method can convert the categorical data. In this project, I used "get_dummies" method to convert all the categorical data to float data. However, as the dataset was not formatted well, some float number was stored as a string type, and some space was stored in the Excel. To resolve these problems, I picked three features that have this issue manually and converted them to the float data type.

The final step of data preprocessing is data scaling. Feature scaling (also known as data normalization) is the method used to standardize the range of features of data. The reason for scaling is to transform feature values according to a defined rule so that all scaled features have the same degree of influence (Jiang et al., 2015). As the range of values of data may vary widely, it becomes a necessary step in data preprocessing when using machine learning algorithms. The method used in this project was "StandScaler" function from Sklearn library. It standardized features by removing the mean and scaling to unit variance.

3.3. Feature Selection

In machine learning, feature selection is the process of selecting a subset of features from a large feature dataset for use, without compromising the quality of the output (DeySarakar et al., 2013). There are many benefits to applying feature selection. For instance, it can reduce overfitting, as there are fewer noisy data compared to the full dataset. Besides, feature selection can improve accuracy and reduce training time. Four feature selection methods are widely used for machine learning in Python, which are univariate selection, recursive feature elimination, principal component analysis, and feature importance. In this project, I applied univariate selection; more specifically, Scikit Learn library "SelectKBest" class was used in this project. Univariate selection can be used to select those features that have the strongest relationship with the output variable. Hence, I applied "SelectKBest" class and Chi-square statistical test to find the features that have the highest Chi-square score.

The Chi-squared statistical test method is a statistical test of independence of two variables. I calculated the Chi-squared statistic value between every feature in training dataset and the target feature. If the target variable is independent of the feature variable, then the value of the Chi-square value is low. Otherwise, the Chi-square value is high, and it indicates they are dependent. In the experiment, I first calculated all the Chi-square value for all the features and then selected the top 50 features with the highest Chi-square value to use as the training dataset for the model training. Next, I picked the top 100 features and the top 200 features to evaluate whether feature selection affects the final evaluation result of the machine learning models.

3.4. Model Training

The project trained six machine learning models, and the experiment was done by using Python language, and Scikit Learn python library. The pipeline of training a model in Scikit Learn library consists of three steps. The first step is to create a model and specify the hyperparameters of the model. Next, fit the model in the first step with training data and learn the parameters. In the final step, apply

the trained model and use the test data to predict the labels (Hao, J. and Ho, T.K.,2019). Besides training the model, I also applied a method in Scikit Learn library called "GridSearchCV." This method was used to exhaustive search over specified parameter values for a model and returned the best parameter value.

The first machine learning model is K-Nearest Neighbour, and it was implemented by using the Scikit Learn python library. The model was trained by following the pipeline as stated above, and multiple parameter values of this model were tried by using "GridSearchCV" function. K-Nearest Neighbour is an algorithm that classifies an instance by a plurality vote of its neighbours. Hence the value of K is an important parameter. In the experiment, I experimented different K value ranges from 3 to 100, and the result showed that the model demonstrated the best performance when K equals 15.

The second machine learning model is Logistic Regression. The Logistic Regression model was also implemented in the Scikit Learn library, so I just followed the same pipeline as the K-Nearest Neighbour and trained Logistic Regression model. For the Logistic Regression model, I tried different parameter values for the parameter "solver." The parameter "solver" is the algorithm to use in the optimization problem, and there are five options which are 'newton-cg', 'lbfgs', 'liblinear', 'sag' and 'saga'. After applying 'GridSearchCV' function, 'lbfgs' solver was found to have the best performance among all the solvers.

The third machine learning model is Naïve Bayes. In Scikit Learn library, there are four different Naïve Bayes models which are Gaussian Naïve Bayes, Multinominal Naïve Bayes, complement Naïve Bayes and Bernoulli Naïve Bayes. After applying several experiments on these four models, the outcome indicated that the Gaussian Naïve Bayes demonstrated the best performance. The Gaussian Naïve Bayes implements the Gaussian Naïve Bayes algorithm for classification, and the likelihood of the feature is assumed to be Gaussian.

The fourth machine learning model is the Support Vector Machine. It was implemented in the Sklearn.svm.SVC class in the Scikit Learn library. Support Vector Machine is a kernel-based machine learning algorithm. In this model, I tried different kernel parameter by using 'GridSearchCV' function. The result showed that 'linear' kernel showed a better performance than the 'rbf' kernel as well as the 'sigmoid' kernel.

The next machine learning model is Random Forest, and the model is from Sklearn.ensemble class. A random forest is a meta estimator that fits many decision tree classifiers on various sub-samples of

the dataset and uses averaging to improve the predictive accuracy and control overfitting. The parameter 'n_estimator' is the number of decision trees in the random forest. I experimented different models with different 'n_estimator' value. Then, the result showed that the Random Forest model had the best performance when the 'n_estimator' equals 200.

The last machine learning model is Artifical Neural Network, and it is from Sklearn. neural_network class. It is a multi-layer perception classifier and optimizes the log-loss function using stochastic gradient descent. This model is more complicated than other models, and it has more parameters that need to be tuning. After experimenting with different sets of the parameters of the model, the result of the grid search indicated that the best performance Artifical Neural Network model had the parameters as follows: the value of parameter 'activation function' was 'tanh', the value of parameter 'solver' was 'sgd', the value of parameter 'early_stopping' was 'False', the value of parameter 'the max_iter' was "10000", the value of parameter 'learning_rate' was 'constant' and the value of parameter 'hidden_layer_sizes' was (100, 100, 100, 100, 100). The 'activation function' was a function that is applied in the hidden layer. Solver 'sgd' refers to stochastic gradient descent, and 'early stopping' was used to terminate training when the validation score was not improving. 'Max_iter' parameter in the model refers to the maximum number of iterations. 'Learning rate' is related to the weight updates and by setting the parameter learning rate 'constant', it means to apply the initial learning rate on the model. Moreover, the parameter 'hidden_layer_sizes' is a parameter that represents the number of neurons in the hidden layers.

3.5. Model Evaluation

Model evaluation is essential as it helps quantify a model's performance. In the model evaluation section, a 10-fold cross validation methodology was used to evaluate the performance of six algorithms. 10-fold cross-validation is the methodology K-fold cross-validation with the parameter K is set to be 10. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. K-fold cross-validation is very popular as it results in a less biased or less optimistic of the model than other mythologies. The procedure of the 10-fold cross validation is simple. Firstly, the dataset was equally partitioned into ten equal folds. On these partitioned folds, I iterated ten times, and each time I selected one-fold for testing and the remaining nine folds for training. Finally, the average evaluation metric value of the ten iterations was obtained (Yadav et al., 2016). Next, the confusion matrix and some evaluation metrics were described in the following. These evaluation metrics were used to evaluate the performance of the models in this experiment. As shown in Table 1, a confusion matrix is a performance measurement for the classification problem. It is a

table with four different combinations of predicted and actual values, which are TR = true positives, TN = true negatives, FP = false positives, and FN = false negatives.

	Actual Positive Value	Actual Negative Value
Predicted Positive Value	TP	FP
Predicted Negative Value	FN	TN

Table 1: Confusion Matrix

The performance of the algorithms was evaluated using the criteria: accuracy (Equation 1), precision (Equation 2), recall (Equation 3), F1 score (Equation 4) and AUC. AUC is the area under the Receiver Operating Characteristic (ROC) curve from prediction scores. It tells how much model is capable of distinguishing between classes.

$$Accuracy = \frac{\text{TP+ TN}}{TP+FP+TN+FN} \quad (1)$$

$$Precision = \frac{\text{TP}}{TP+FP} \quad (2)$$

$$Recall = \frac{\text{TP}}{TP+FN} \quad (3)$$

$$F1\ Score = 2 * \frac{\text{Precision}*\text{Recall}}{Precision + Recall} \quad (4)$$

3.6. Learning Curve

Learning curves are deemed effective tools for monitoring the performance of workers exposed to a new task and provide a mathematical representation of the learning process (Anzanello et al., 2011). Learning curves are widely used in machine learning, and it is common to create dual learning curves for a machine learning model during training. In this project, I used the learning curve class from Scikit Learn library, and the plot contains two curves, which are a training learning curve and validation learning curve (Jason, 2019). The learning curve is a tool to find out how much we can benefit from adding more data and whether the model is overfitting, underfitting, or good fitting. The training curve was calculated from the k-fold cross validation training dataset and it indicated how well the model was learning. The validation learning curve was calculated from the k-fold cross validation testing dataset, and it indicated how well the model was generalizing.

4. Experiment Results

4.1. Data Preprocessing Results

The original hospital dataset contains 7476 instances and 134 features. After removing the instances without target feature, the remaining dataset only contains 992 valid instances, and among the 992 valid instances, there were 733 instances target value is 1 and 259 instances target value is 0. For the training and testing dataset, after deleting the features that are associated with the target feature, there were 127 features. In the experiment, I set a threshold of the number of non-empty values for each feature to 3, if it is smaller than 3, then I drop this feature. After dropping the empty features by following this rule, the feature set decreased from 134 to 63. By observation, there were many categorical data in the dataset. As the categorical data cannot fit the machine learning algorithm, they must convert to the numerical type, which means they have to convert from string datatype to float datatype. After I applied the "get_dummies" function to convert all the categorical data to float data, the number of features increased to 260. Finally, the processed training dataset after the preprocessing step contains 992 instances and 260 features.

4.2. Feature Selection Results

10-fold cross-validation methodology was used in the model evaluation phase, and the evaluation metrics used to evaluate the performance were accuracy, precision, recall, F1 score, and AUC. The evaluation results of all six machine learning algorithms with tuned parameters for top 200 Chi-squared features, top 100 Chi-squared features, and top 50 Chi-squared features were shown in Table 3, Table 4 and Table 5 respectively. Moreover, to compare the effect of the feature selection to the evaluation results, I plotted six bar charts for all six machine learning algorithms.

In each bar chart, it compared the performance among the training dataset with all features, training dataset with the top 100 Chi-squared value features and training dataset with top 50 Chi-squared value features. As it was shown in Figure 2, the K-Nearest Neighbour model achieved the best performance when using the top 100 features. However, the Logistic Regression model had a better performance if it was applying the training dataset with all features. Naïve Bayes model, Support Vector Machine model, and Random Forest model also performed better when applying the training dataset without feature selection. Artificial Neural Network model was different, and it performed better when applying the top 50 features. Overall, the feature selection did not improve the performance for all the models; most of the models had similar performance after having the feature selection. Chi-squared based feature selection only enhanced the performance of a few models such as the K-Nearest Neighbour model and Artifical Neural Network model.

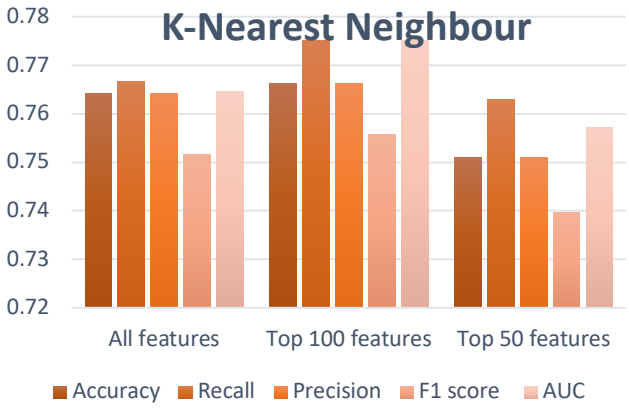

Figure 2: KNN Evaluation Result

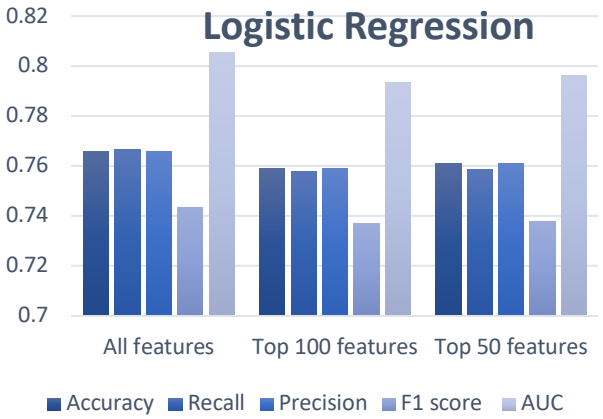

Figure 3: LR Evaluation Result

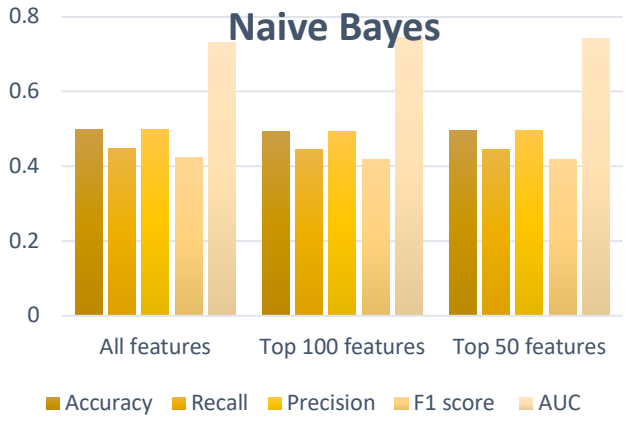

Figure 4: NB Evaluation Result

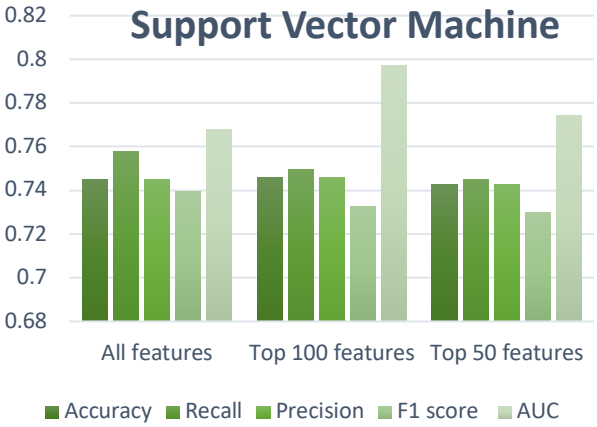

Figure 5: SVM Evaluation Result

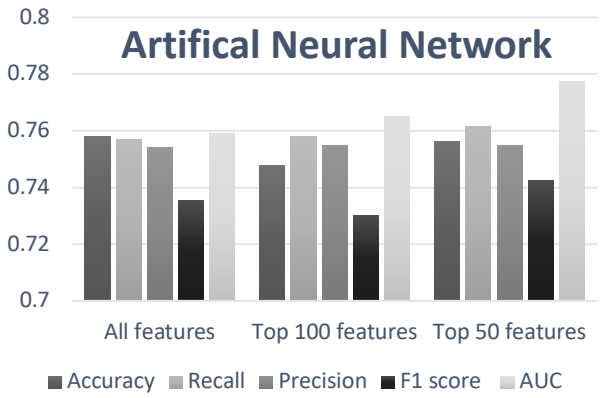

Figure 6: ANN Evaluation Result

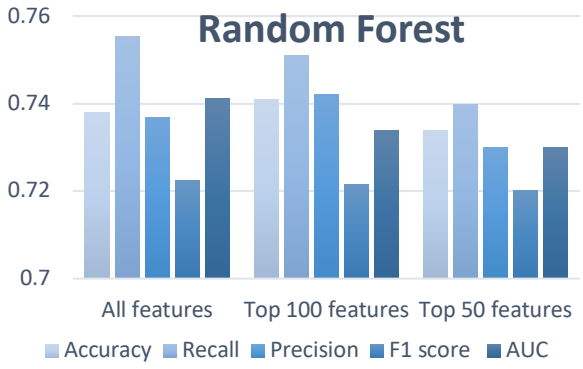

Figure 7: RF Evaluation Result

4.3. Selecting the Best Prediction Model among 6 Algorithms

The experiment evaluation results were shown in the table below for all six algorithms. Table 2 showed the evaluation results of six algorithms with all feature dataset (without feature selection). In addition, Table 3, Table 4, Table 5 showed the evaluation results with top 200 Chi-square score features, top 100 Chi-square score features, and top 50 Chi-square score features respectively.

Among all the experiment results, K-Nearest Neighbour model achieved the highest accuracy 0.7662, highest precision value of 0.7752, highest recall value of 0.7662, highest F1 score of 0.755 when applying the top 100 Chi-square score features. For AUC evaluation metric, Logistic Regression model had the highest AUC value, which was 0.8054 when using the training dataset without feature selection. Hence the best prediction model is the K-Nearest Model with parameter K equals 15.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**AUC**
K-Nearest Neighbour (k = 15)	**0.7642**	**0.7666**	**0.7642**	**0.7516**	0.7647
Logistic Regression (solver: 'lbfgs')	0.7661	**0.7666**	0.7661	0.7436	**0.8054**
Naïve Bayes (BernoulliNB）	0.5000	0.4499	0.5000	0.4241	0.7328
Support Vector Machine (SVC linear kernel)	0.7450	0.7579	0.7450	0.7395	0.7679
Random Forest (n_estimators= 200)	0.7381	0.7553	0.7370	0.7225	0.7412
Artificial Neural Network (5 hidden layers)	0.7582	0.7571	0.7541	0.7356	0.7593

Table 2: Experiment Result with All Feature Dataset

Model	Accuracy	Precision	Recall	F1-score	AUC
K-Nearest Neighbour (k = 15)	**0.7662**	**0.7730**	**0.7662**	**0.7556**	0.7746
Logistic Regression (solver: 'lbfgs')	0.7591	0.7581	0.7591	0.7372	**0.7955**
Naïve Bayes (BernoulliNB）	0.4939	0.5137	0.4939	0.4196	0.7436
Support Vector Machine (SVC linear kernel)	0.7451	0.7475	0.7451	0.7321	0.7774
Random Forest (n_estimators= 200)	0.7391	0.7548	0.7391	0.7325	0.7374
Artificial Neural Network (5 hidden layers)	0.7551	0.7542	0.7612	0.7276	0.7758

Table 3: Experiment Result with Top 200 Chi-Square Score Feature Dataset

Model	Accuracy	Precision	Recall	F1-score	AUC
K-Nearest Neighbour (k = 15)	**0.7662**	**0.7752**	**0.7662**	**0.7557**	0.7751
Logistic Regression (solver: 'lbfgs')	0.7591	0.7581	0.7591	0.7372	0.7933
Naïve Bayes (BernoulliNB）	0.4949	0.4451	0.4949	0.4183	0.7440
Support Vector Machine (SVC linear kernel)	0.7461	0.7495	0.7461	0.7330	**0.7972**
Random Forest (n_estimators= 200)	0.7410	0.7510	0.7421	0.7215	0.7340
Artificial Neural Network (5 hidden layers)	0.7481	0.7581	0.7551	0.7304	0.7652

Table 4: Experiment Result with Top 100 Chi-Square Score Feature Dataset

Model	Accuracy	Precision	Recall	F1-score	AUC
K-Nearest Neighbour (k = 15)	0.7510	**0.7629**	0.7510	0.7397	0.7573
Logistic Regression (solver: 'lbfgs')	**0.7611**	0.7588	**0.7611**	0.7381	**0.7964**
Naïve Bayes (BernoulliNB）	0.4959	0.4461	0.4959	0.4193	0.7432
Support Vector Machine (SVC linear kernel)	0.7430	0.7453	0.7430	0.7303	0.7745
Random Forest (n_estimators= 200)	0.7340	0.7399	0.7299	0.7203	0.7300
Artificial Neural Network (5 hidden layers)	0.7562	0.7618	0.7551	**0.7425**	0.7775

Table 5: Experiment Result with Top 50 Chi-Square Score Feature Dataset

4.4. Learning Curves Results

In this section, the learning curve of the best-predicted model K-Nearest Neighbour, as well as the learning curve of the Random Forest were analysed. The K-Nearest Neighbour learning curve and Random Forest learning curve were shown below in Figure 8, Figure 9, respectively. Other learning curves for the remaining models were shown in Figure 10, Figure 11, Figure 12, Figure 13, respectively in the Appendix section.

The y-axis of the plot is the accuracy score, and the x-axis of the plot is the training examples. The red line in the plot indicates the training score, and the green line in the plot indicates the validation score. As it was shown in Figure 8, in KNN learning curve, with the increase of the training data, both the training score and validation score increased which means the model had learned from the data and performed better. Moreover, the training curve and validation curve roughly converged to a high score between 0.75 to 0.80. The variance between the training score and the validation score was small. To conclude, the KNN model was proper fitting.

Compared to the KNN learning curve, RF learning curve, which was shown Figure 12, had a different shape. With the increase of the training data, the training curve of the RF model decreased while the validation curve increased. Moreover, the learning curve and validation curve did not converge, and the validation curve had a much lower value than the training curve. The significant variance between the learning curve and the validation curve indicated the RF model suffered from overfitting problem.

Similar to K-NN and RF models, other models can all be analysed in this way. For LR and SVM model, the training curve, and the validation curve converged to a high score and the variance was small. Hence the LR model and SVM model were both good-fit. For the NB model, the training curve and the validation curve converged to a low score. Hence the NB model was underfitted. For the ANN model, the variance was a little bit big. Hence the ANN model was a little bit overfitted.

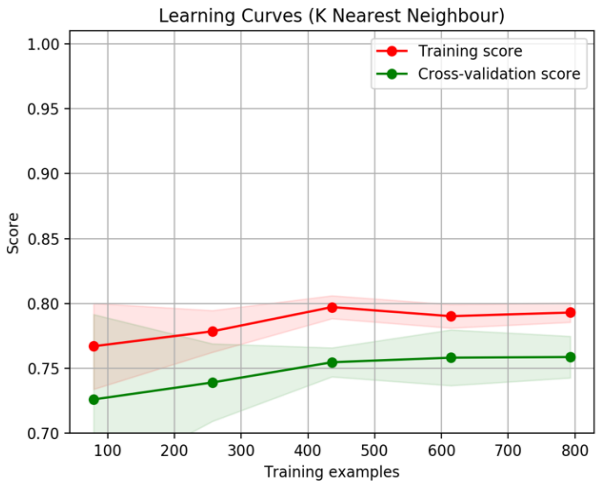

Figure 8: K-NN Learning Curve

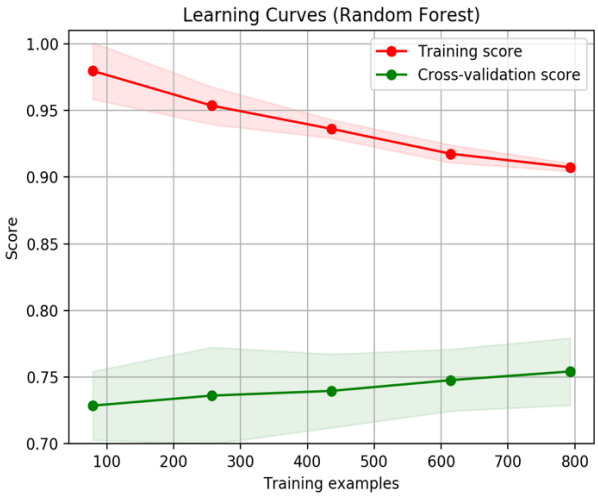

Figure 9: RF Learning Curve

5. Discussion

Integrating machine learning models and disease status to predict patient survival has become increasingly popular in the public health area. Recent machine learning algorithms have provided us powerful and promising tools for the study of prediction of disease (Ma et al., 2018). In this project, I trained six machine learning models to predict the fertility status of cancer patients. Also, I applied the Chi-squared based feature selection method and 10-fold cross-validation methodology. The results showed that feature selection did improve performance for some models. For instance, it improved the accuracy of the K-Nearest Neighbour model from 0.7642 to 0.7662. In addition, the F1 score of the K-Nearest Neighbour model also increased from 0.7516 to 0.7557. Besides the K-Nearest Neighbour model, Artifical Neural Network model also showed improvement when applying feature selection. The F1 score of Artifical Neural Network without feature selection was 0.7356, and the F1 score of Artifical Neural Network model with feature selection increased to 0.7425. However, the feature selection did not improve the performance of other models. As the Chi-squared based feature selection method only selected a subset of the dataset, and it caused the loss of data which may affect the performance of the model. The feature selection aims to remove the noisy data from the dataset, and the results of feature selection indicated that the processed hospital dataset did not contain too much noisy data.

Among the six machine learning models, the best performing model was K-Nearest Neighbour model with the highest accuracy (0.7662), highest precision (0.7752), highest recall (0.7662) and highest F1 score (0.7557). Other models also had a close performance, for example, the F1 score of the Logistic Regression model was 0.7436, the F1 score of the Support Vector Machine model was 0.7395, the F1 score of the Random Forest model was 0.7325. The only exception was the Naive Bayes model, and it had the worst performance among all the models with a 0.4241 F1 score. By analysing the learning curves, the plots of the learning curves also showed valued information. The learning curves of K-Nearest Neighbour, Logistic Regression, and Support Vector Machine indicated they were all appropriated fitted. However, for the Random Forest model and Artifical Neural Network model, the learning curves showed these models were overfitted, and more training data was needed. For Naïve Bayes, the learning curve indicated it was underfitted.

There are also some works that could be done in the future to improve the performance of the model. First of all, the model would perform better if there is more data from the hospital, especially for Random Forest model and Artifical Neural Network model as they were overfitted. Next, it would be better if the hospital could improve the quality of the dataset. For instance, in the given dataset, some

feature had a value such as '<=64', '>=64' and this kind of categorical data was not accurate and not helpful. If the hospital could provide a more well-formatted dataset, it would save time for the data preprocessing step and feature engineering step. Finally, if more time is given, I could tune more parameters of the machine learning algorithms, and it may result in a better performance.

6. Conclusion and Future Work

I compared six machine learning models to predict the fertility status of cancer patients. Among the six models, K-Nearest Neighbour model performed best with the highest accuracy of 0.7662, highest precision of 0.7752, highest recall of 0.7662, and highest F1 score of 0.7557. Applying these models may assist the hospital to predict on the fertility status of the cancer patients. Some future work may improve the performance of the model includes more data from the hospital and tuning more parameters of the algorithms. Moreover, if the hospital could provide a more well-formatted dataset that has less confusing categorical feature and more useful features, it would also improve the performance of the machine learning model.

Data Availability

Due to the confidentiality agreement I am unable to enclose the dataset with this submission, please contact Dr. Patrick Pang (pat.pang@unimelb.edu.au) if you need to access the dataset.

Code

The code implementation of the project is available in the following GitHub repository:
https://github.com/yiming95/COMP90055

If you have accessed the dataset from Dr. Patrick Pang, then you can run the code by following instructions below:

- The code of the python version is called 'ml_project.py'. Put the dataset CSV file in the same directory with the python script and use command line 'python ml_project.py' to run the code.
- The code of the Jupyter notebook version is called 'COMP90055.ipynb'. Put the dataset CSV file in the same directory with the notebook script and click 'run All' button.

Reference

[1] Jenny, M. (2016). *Fertility and Cancer.* Available from: https://www.cancer.org.au/content/about_cancer/ebooks/livingwithcancer/Fertility_and_cancer_booklet_May_2016.pdf.

[2] Yifan, W. Antoinette, A. & Shanna, L. (2018). *Systematic review of fertility preservation patient decision aids for cancer patients*. *Psycho-Oncology*. 28 (3), 459–462. Available from: https://onlinelibrary-wiley-com.ezp.lib.unimelb.edu.au/doi/full/10.1002/pon.4961.

[3] Andreas, M. & Sarah, G. (2016). *Introduction to Machine Learning with Python*. Sebastopol, O'Reilly Media Publishing.

[4] Jahidur, k., Srizan, C., Humayera, I. & Enayetur, R. (2019) *Machine Learning Algorithms to Predict the Childhood Anemia in Bangladesh*. Journal of Data Science,17(1). P195 – 218. Available from: https://eds-a-ebscohost-com.ezp.lib.unimelb.edu.au/eds/detail/detail?vid=2&sid=f08138ea-b249-4cfd-b4bc-00f36eb3595e%40sdc-v-sessmgr06&bdata=JnNpdGU9ZWRzLWxpdmUmc2NvcGU9c2l0ZQ%3d%3d#AN=edsarl.16838602.201901.201901290019.201901290019.195.217&db=edsarl.

[5] Jianglin, H., Yan-Fu, L. & Min, X. (2015). *An empirical analysis of data preprocessing for machine learning-based software cost estimation*. Information and Software Technology 67 (2015), 108–127. Available from: https://www-sciencedirect-com.ezp.lib.unimelb.edu.au/science/article/pii/S0950584915001275.

[6] Amitabha, D. (2018). *Data Preprocessing for Machine Learning.* Available from: https://medium.com/datadriveninvestor/data-preprocessing-for-machine-learning- 188e9eef1d2c.

[7] Sona, T. & Musa, M. (2013). *Learning the Naïve Bayes Classifier with Optimization Models* Vol. 23, No. 4, 787–795, Int. J. Appl. Math. Comput. Sci. Sonali, B. (2014) *Research Paper on Basic of Artificial Neural Network* Vol.2, Issue:1, ISSN: 2321-8269, International Journal on Recent and Innovation Trends in Computing and Communication.

[8] Hao, J. and Ho, T.K. (2019). *Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language*. Journal of Educational and Behavioral Statistics, 44(3), pp.348–361.

[9] Yadav, Sanjay, and Sanyam Shukla. (2016). *"Analysis of K-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification."* 2016 IEEE 6th International Conference on Advanced Computing *(IACC)*, Feb. 2016, 10.1109/iacc.2016.25.

[10] Anzanello, Michel Jose, and Flavio Sanson Fogliatto. (2011). *"Learning Curve Models and Applications: Literature Review and Research Directions."* International Journal of Industrial Ergonomics, vol. 41, no. 5, Sept. 2011, pp. 573–583, www.sciencedirect.com/science/article/abs/pii/S016981411100062X, 10.1016/j.ergon.2011.05.001.

[11] Jason, B. (2019). *"A Gentle Introduction to Learning Curves for Diagnosing Machine Learning Model Performance."* Machine Learning Mastery, 3 Apr. 2019, machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/.

[12] Ma, Han, et al. (2019). *"Application of Machine Learning Techniques for Clinical Predictive Modelling: A Cross-Sectional Study on Non-alcoholic Fatty Liver Disease in China."* BioMed Research International, vol. 2018, 3 Oct. 2018, pp. 1–9, 10.1155/2018/4304376.

[13] Natalia, K. & Shantanu, D. (2018). *Machine Learning for Classification of Inhibitors of Hepatic Drug Transporters*. 2018 17th IEEE International Conference on Machine Learning and Applications.

[14] DeySarakar, Subhajit, and Saptarsi Goswami (2013). *"Empirical Study on Filter Based Feature Selection Methods for Text Classification."* International Journal of Computer Applications, vol. 81, no. 6, 15 Nov. 2013, pp. 38–43, 10.5120/14018-2173.

[15] Wes McKinney. (2010). *Data Structures for Statistical Computing in Python*, Proceedings of the 9th Python in Science Conference, 51-56.

[16] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesna. (2011). *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, 12**,** 2825-2830.

[17] Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. (2011). *The NumPy Array: A Structure for Efficient Numerical Computation*, Computing in Science & Engineering, 13, 22-30.

[18] Van Rossum, G., & Drake Jr, F. L. (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

[19] Anaconda Software Distribution. (2017). Computer software. Anaconda, Inc., Nov. 2017. Web. <https:www.anaconda.com>.

[20] Fernando Pérez, Brian E. Granger. (2007), *IPython: A System for Interactive Scientific Computing*, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007, doi:10.1109/MCSE.2007.53. URL: https://ipython.org

Appendix

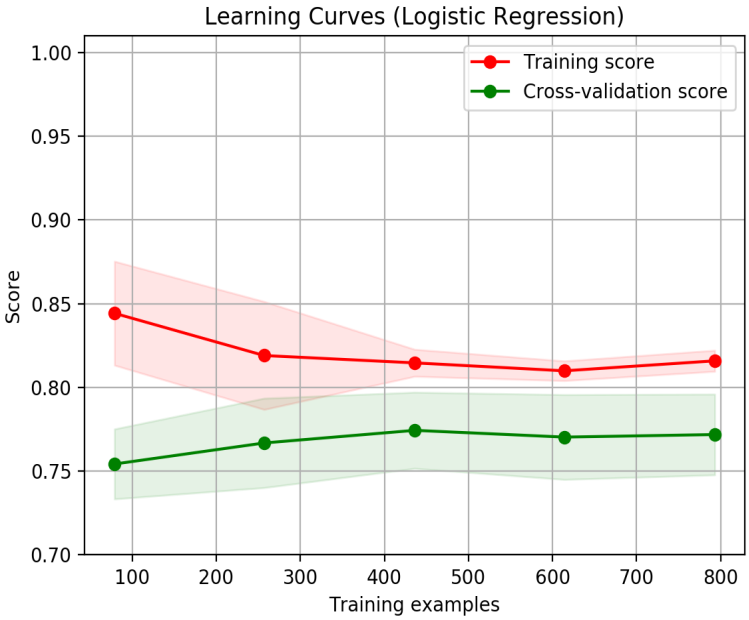

Figure 10: LR Learning Curve

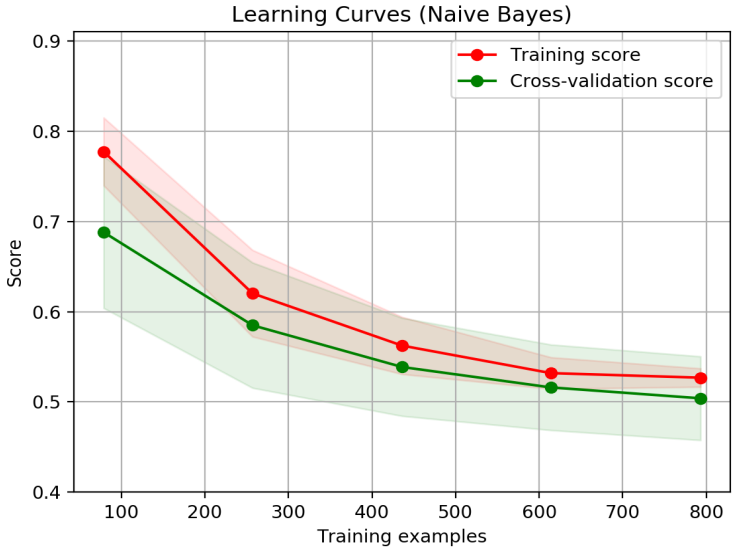

Figure 11: NB Learning Curve

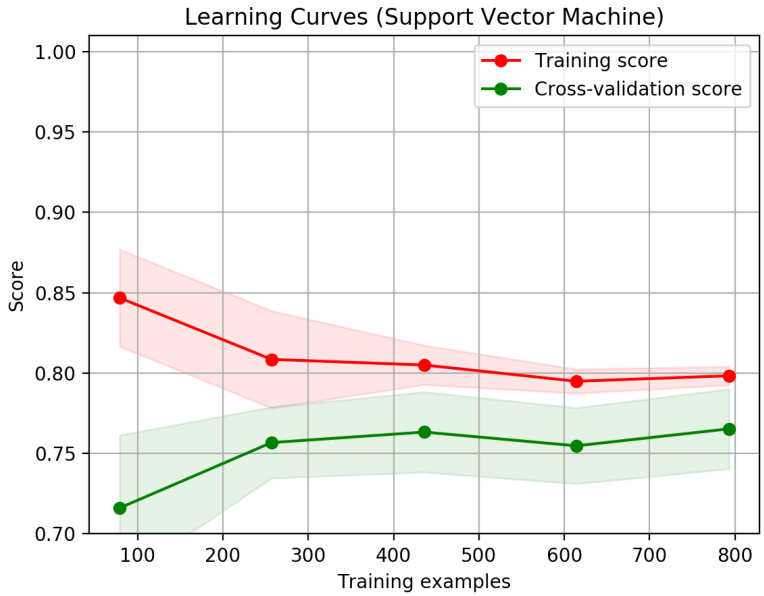

Figure 12: SVM Learning Curve

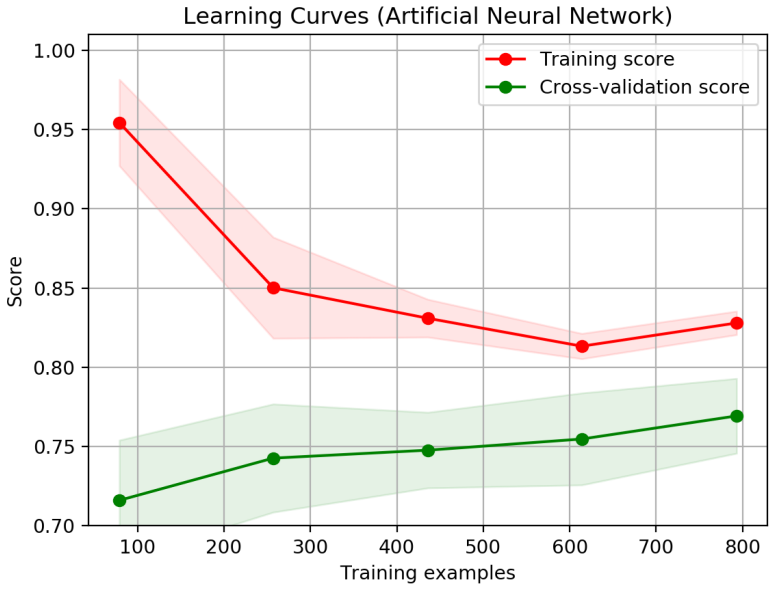

Figure 13: ANN Learning Curve