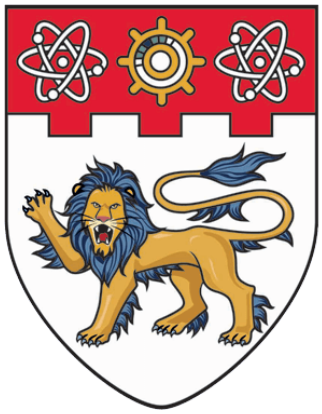


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CE/CZ4032 Data Analytics & Mining

Project Report

Group 18 - Contribution:

Jeffrey (U1620172K) - 16.67%

Gantari Evanda Raufani (U1620108F) - 16.67%

Michelle Vanessa (U1620071L) - 16.67%

Natassa Karinka (U1620138L) - 16.67%

Stella Marcella (U1720356F) - 16.67%

Wilbert (U1721406E) - 16.67%

School of Computer Science and Engineering

Academic Year 2019/2020 Semester 1

Table of Contents

Abstract

In this project, Knowledge Discovery in Databases (KDD) is implemented to perform breast tumour prediction using Breast Cancer Wisconsin dataset from Kaggle, which consists of 569 data samples of digitized image of a fine needle aspirate (FNA) of a breast mass. Several data mining techniques such as K-Nearest Neighbour Classifier, Naive Bayes Classifier, Decision Tree Classifier, Artificial Neural Network, and K-means Clustering are used to do the prediction. Recursive Feature Elimination based on Support Vector Regression model is also implemented to find the optimal number of features to achieve better performance. The result of our project shows that different techniques have different optimal number of features to achieve the best performance possible. Full-resolution graphs and source code in Python can be found in the GitHub repository.

1. Problem Description

1.1. Motivation

In Singapore's society, there has been an increase in the number of cancer cases over the past decade. According to National Cancer Centre Singapore (NCSS), one in every 4 to 5 people in Singapore may develop cancer in their lifetime, and the number of people living with cancer will continue to increase in the future. The most common cancer affecting men is colorectal cancer (17.2%), while breast cancer is the most common cancer amongst women in Singapore, accounting for almost 30% of the cases. Additionally, according to Singapore Cancer Society, 1 in 14 women will develop breast cancer by 75 years old.

In light of the prevalence of breast cancer among women in Singapore and the high risk associated with it, it is important to classify breast tumour cases (as benign or malignant) as correctly and timely as possible so that health institutions can administer appropriate treatment to patients as early as possible. With recent advancement in Information Technology and the abundant amount of data available today, Knowledge Discovery from Data (KDD) can be performed to predict the class of breast tumour cases. Insightful information such as trends and patterns can be obtained from analysing the data and applying data mining techniques.

1.2. Problem Definition

In this experiment, data mining is performed on the dataset extracted from digitised image of a fine needle aspirate (FNA) of a breast mass. The model trained will be able to predict whether the tumor is benign or malignant. The aim of doing this experiment is to reduce the risk of misdiagnosis given certain features from a digitized image of a breast mass.

1.3. Related Work

With the increasing occurrence of women diagnosed with breast cancer, there are a large number of studies using data mining techniques to identify and analyse the tumour, to assist planning of clinical trials and treatment to administer to each patient. One such study is the breast cancer risk prediction study (2015) by Obafemi Awolowo University, Nigeria and McPHERSON University, Nigeria. The breast cancer data used for this study is collected from LASUTH and the data is classified using two algorithms, J48 decision trees and Naive Bayes algorithms. Evaluating the performance of the two algorithms, J48 decision trees showed a

higher accuracy; a more effective and efficient classification for the prediction of breast cancer risks among the Nigerian women who participated in the study.

Shajahan et al (2013) worked on the application of data mining techniques to model breast cancer data using decision trees to predict the presence of cancer. Data collected contained 699 instances (patient records) with 10 attributes and the output class as either benign or malignant. Input used contained sample code number, clump thickness, cell size and shape uniformity, cell growth and other results physical examination. The results of the supervised learning algorithm applied showed that the random tree algorithm had the highest accuracy of 100% and error rate of 0 while CART had the lowest accuracy with a value of 92.99% but naïve bayes' had the accuracy of 97.42% with an error rate of 0.0258.

Rajesh et al (2012) used SEER dataset for the diagnosis of breast cancer using the C4.5 classification algorithm. The algorithm was used to classify patients into either pre-cancer stage or potential breast cancer cases. Random tests were performed on the dataset which contained information for 1183 patients including the age of diagnosis, regional lymph nodes measures, and sequence number of tumours, dimension of primary tumour and contiguous growth of the primary tumour. The analysis involved the use of three random 500 records form the pre-processed data of 1183 and was used as training data and the lowest error rate achieved was 0.599. During the testing phase, the C4.5 classification rules were applied to a test sample and the algorithm showed had an accuracy of 92.2%.

Delen et al (2005) compared ANN, decision tree and logistic regression techniques for breast cancer prediction analysis. They used the SEER data of twenty variables in the prediction models. From the experiment the author found that the decision tree with 93.6% accuracy and ANN with 91.2% are more superior to logistic regression with 89.2% accuracy. The study is based on multiple prediction models for breast cancer survivability using large datasets along with 10 fold cross validation method. It provides a relative prediction ability of different data mining methods. In future this work is extended by collecting real dataset in the clinical laboratory

2. Approach

2.1. Methodology

2.1.1. Data selection

The dataset used in this experiment was taken from Kaggle. It contains 569 tuples of 30 features and one categorical data, classifying whether the set is identified as benign or malignant.

2.1.2. Data cleaning and pre-processing

The dataset is checked for any missing values, some irrelevant features are removed, display the correlation matrix for all features and normalisation of the data. The normalised data is then split into 2 sets, training and testing samples.

2.1.3. Data transformation

During the implementation of some methods, the categorical one-dimensional target data is transformed into one-hot matrix.

2.1.4. Data mining

In this project, 5 different approaches on data mining are implemented, 4 of which are unsupervised learning using classifying methods. From the given data, the algorithm is used to predict whether the output is benign or malignant and calculate its accuracy.

2.2. Algorithms

2.2.1. Supervised Learning

Supervised learning is a type of machine learning where the algorithm is given the input and target output. The system will learn the sample input and map a function so that it can predict an output of unseen data as accurately as possible.

2.2.1.1. k-Nearest Neighbour (k-NN) Classifier

k-Nearest Neighbour Classifier uses k numbers of “nearest” points to perform classification. The test data is classified based on the majority vote of the class labels of k numbers of “nearest” neighbours, in which the “nearest” neighbours are given weights based on the distance to the test point. The closer the neighbour is to the test point, the greater the weight.

2.2.1.2. Naive Bayes Classifier

Naive Bayes Classifier is a probabilistic classifier that makes classifications using Bayes Theorem.

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

With Bayes Theorem, the probability of C occurs given that A also occurs can be calculated. It is called naive because it assumes that the features are independent of each other and does not affect one another. It also assumes that the features have equal contribution to the outcome.

2.2.1.3. Decision Tree Classifier

A decision tree is a tree in which each non-leaf node is labeled with an input feature. The training data is then split into subsets (children of the particular node) based on a set of splitting rules, in this case based on the entropy. This process is repeated on each subset until it produce leaf nodes in which all the remaining data has the same class (Malignant or Benign).

2.2.1.4. Artificial Neural Network

Artificial Neural Network is an approach on machine learning that is inspired by the human brain system. The network consists of several layers of neurons that learn a set of data to find its pattern and predict the result of unseen data based on the learned pattern. The network learns the data using optimisation methods so that it can predict output with optimal accuracy.

2.2.2. Unsupervised Learning

In unsupervised learning, a machine learning model is only given a set of input data and no target output. The model is aimed to learn the underlying structure or distribution of the data.

2.2.2.1. K-means Clustering

K-means clustering is a partitional clustering approach in which each point is assigned to the cluster with the closest centroid (centre point). Initially, the centroids will be chosen randomly. Then, an iterative computation is done to optimize the centroids. The iterations will be considered as done when either the amount of iterations specified has been reached or the position of the centroids has stabilized.

3. Implementations

3.1. Understanding the dataset

This file data.csv contains the dataset used for training and testing. It contains 569 records, each describing the characteristics of the cell nuclei present in the digitised image of a fine needle aspirate (FNA). The features are described below:

- **id**: ID number
- **diagnosis**: The diagnosis of breast tissues (63% M = malignant, 37% B = benign)
- **radius_mean**: mean of distances from center to points on the perimeter (range: 6.98 - 28.1)
- **texture_mean**: standard deviation of gray-scale values (range: 9.71 - 39.3)
- **perimeter_mean**: mean size (perimeter) of the core tumor (range: 43.8 - 189)
- **area_mean**: mean size (area) of the core tumor (range: 144 - 2.5k)
- **smoothness_mean**: mean of local variation in radius lengths (range: 0.05 - 0.16)
- **compactness_mean**: mean of perimeter^2 / area - 1.0 (range: 0.02 - 0.35)
- **concavity_mean**: mean of severity of concave portions of the contour (range: 0 - 0.43)
- **concave points_mean**: mean for number of concave portions of the contour (range: 0 - 0.2)
- **symmetry_mean** (range: 0.11 - 0.3)
- **fractal_dimension_mean**: mean for "coastline approximation" - 1 (range: 0.05 - 0.1)
- **radius_se**: standard error for the mean of distances from center to points on the perimeter (range: 0.11 - 2.87)
- **texture_se**: standard error for standard deviation of gray-scale values (range: 0.36 - 4.88)
- **perimeter_se**: standard error for the mean size (perimeter) of the core tumor (range: 0.76 - 22)
- **area_se**: standard error for the mean size (area) of the core tumor (range: 6.8 - 542)
- **smoothness_se**: standard error for local variation in radius lengths (range: 0 - 0.03)
- **compactness_se**: standard error for perimeter^2 / area - 1.0 (range: 0 - 0.14)
- **concavity_se**: standard error for severity of concave portions of the contour (range: 0 - 0.4)
- **concave points_se**: standard error for number of concave portions of the contour (range: 0 - 0.05)
- **symmetry_se** (range: 0.01 - 0.08)
- **fractal_dimension_sestandard** error for "coastline approximation" - 1 (range: 0 - 0.03)
- **radius_worst**: "worst" or largest mean value for mean of distances from center to points on the perimeter (range: 7.93 - 36)
- **texture_worst**: "worst" or largest mean value for standard deviation of gray-scale values (range: 12 - 49.5)
- **perimeter_worst**: “worst” or largest mean value for mean size (perimeter) of the core tumor (range: 50.4 - 251)
- **area_worst**: “worst” or largest mean value for mean size (area) of the core tumor (range: 185 - 4.25k)
- **smoothness_worst**: "worst" or largest mean value for local variation in radius lengths (range: 0.07 - 0.22)
- **compactness_worst**: "worst" or largest mean value for perimeter^2 / area - 1.0 (range: 0.03 - 1.06)
- **concavity_worst**: "worst" or largest mean value for severity of concave portions of the contour (range: 0 - 1.25)

- **concave points_worst**: "worst" or largest mean value for number of concave portions of the contour (range: 0 - 0.29)
- **Symmetry_worst** (range: 0.16 - 0.66)
- **fractal_dimension_worst**: "worst" or largest mean value for "coastline approximation" - 1 (range: 0.06 - 0.21)

3.2. Pre-processing the dataset

3.2.1. Reading the data

Using Pandas library, data.csv is read and saved as a dataframe of 569 tuples with 33 features.

3.2.2. Removing unnecessary columns

The data contains some information that is unnecessary in the data mining process, such as id and empty columns. These columns are removed from the dataframe using Pandas built-in function. At the end, there are 30 relevant features used for data mining process.

3.2.3. Standardising the data

In this step, the data is normalised using scikit-learn library StandardScaler.

3.2.4. Splitting the data

From the cleaned dataset, the 569 data is divided into two datasets, the training dataset and testing dataset with ratio of 70:30 respectively.

3.3. Training the model

3.3.1. k-Nearest Neighbour

The k-Nearest Neighbour classifier is implemented by using the function KNeighborsClassifier from the Scikit Learn library. In the experiment, different models are built with different k values, k = [3, 5, 7, 11, 13, 17, 19]. Recursive Feature Elimination (RFE) is implemented using scikit-learn library RFE in this project. RFE is performed based on the result of the linear Support Vector Regression on the training set. The number of features tested are [10, 15, 20, 25, 30]. Odd k values are used to prevent the two classes to have the same count. Each training data is weighted based on its distance to the test data by the parameter *weights = 'distance'*.

For each k values and models built, the method fit fits the model using the training data and the respective target values. The test data is then predicted using the method predict. The accuracy of the classifier is calculated using the accuracy_score function from the Scikit Learn library.

3.3.2. Naive Bayes

Similar to the k-Nearest Neighbour classifier, the Naive Bayes classifier is implemented by using the function GaussianNB from the Scikit Learn library. The models are built with different number of features from [10, 15, 20, 25, 30] by features selection using the RFE with SVR. Using the fit method, the model fits Gaussian Naive Bayes according to the training data and respective target values. The classification of the test data is performed using the method predict. The accuracy of the classifier is calculated using the accuracy_score function from the Scikit Learn library.

3.3.3. Decision Tree

The decision tree classifier is implemented by using the function DecisionTreeClassifier from the Scikit Learn library. Different decision trees are built with different number of features from [10, 15, 20, 25, 30]. The features selection is done using the RFE with SVR. Using the fit method, the model fits the training data and respective target values into the decision tree model. The classification of the test data is performed using the method predict. The accuracy of the classifier is calculated using the accuracy_score function from the Scikit Learn library.

3.3.4. Artificial Neural Network

Before training the network, the target output needs to be converted into one-hot matrix since neural network produces one-hot matrix as an output. In this algorithm, Keras library is used to implement the network, and it requires the data to be a NumPy array. Hence, the data from pre-processing is converted into NumPy array.

In this project, we create a 3-layer neural network consisting of input layer, hidden layer with 10 neurons, and an output softmax layer with 2 neurons for the two classes, benign and malignant. This network is represented by Keras library Sequential and Dense. Then, the network is trained by the training set using Stochastic Gradient Descent with learning rate 0.001 and decay parameter 10^{-6}.

3.3.5. K-Means Clustering

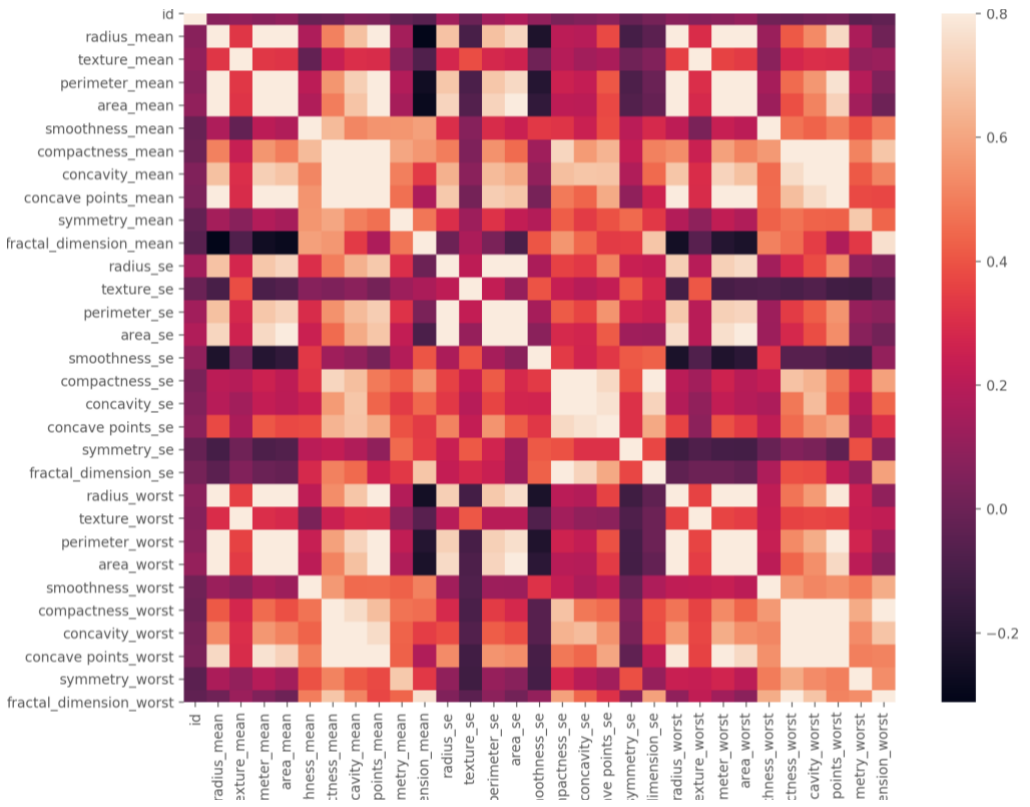
Scikit Learn library is used to implement the clustering algorithm using the KMeans function and TSNE (T-distributed Stochastic Neighbor Embedding) function to plot the clusters. In this experiment, the training dataset is combined with the testing dataset. Then, KMeans function is used to group them into two clusters. After the centroids have stabilized, the finalized model is obtained. This model is then compared with the actual dataset to obtain the accuracy.

Both the centroids and the dataset have 30 features. The obtained centroids are appended into the combined dataset to be passed into a function called TSNE. This function will create a 2D visualization to visualize the clusters.

4. Experimental Results and Analysis

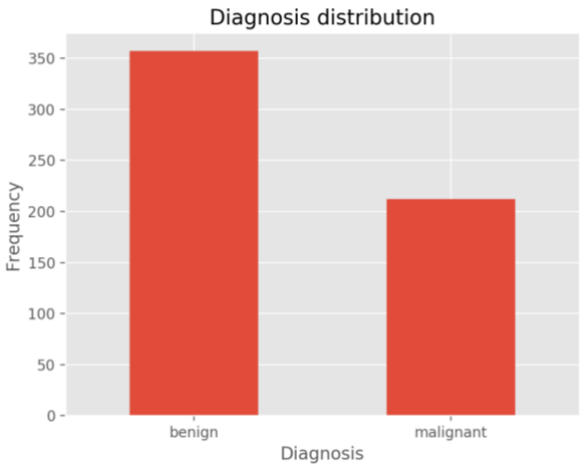
4.1. Experimental Setup

After pre-processing, the data is visualised in a correlation matrix. Correlation matrix shows the correlation of two variables. The brighter the colour, the more correlated the two variables are.



As can be seen from the correlation matrix, there are a few features which are highly correlated with one another. For example, radius_mean, perimeter_mean, and area_mean. This is due to the fact that perimeter and area are derived from radius. Similarly, id is not correlated to any other features since it is a unique identification number that does not affect the classification process.

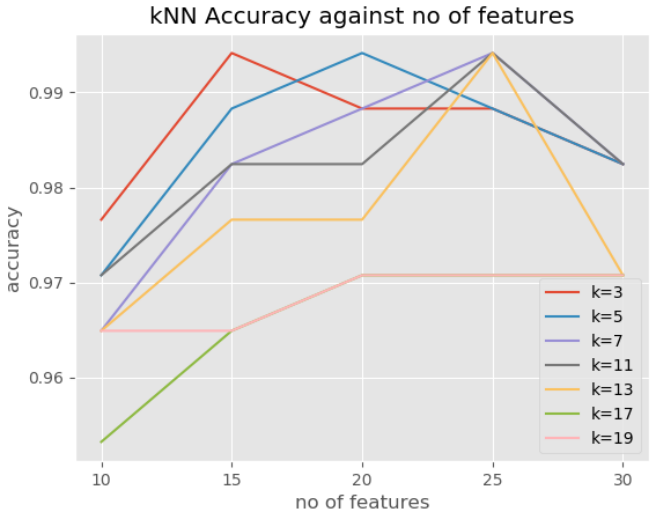
Furthermore, the distribution of the classes is also visualised in a graph.



4.2. Results and Analysis

4.2.1. K-Nearest Neighbours

Using different values of k and RFE to remove insignificant features, a total of 35 separate trainings are performed on the model using k values of 3, 5, 7, 11, 13, 17 and 19, each with 10, 15, 20, 25, and 30 number of features. The test accuracy of each model is evaluated and compared.



From the graph, the lower k values generally have higher accuracies than the higher k values for the same number of features. For example, for number of features = 15, the accuracies are:

k = 3, no of features 15, accuracy 0.994152

k = 5, no of features 15, accuracy 0.988304

k = 7, no of features 15, accuracy 0.982456

k = 11, no of features 15, accuracy 0.982456

k = 13, no of features 15, accuracy 0.976608

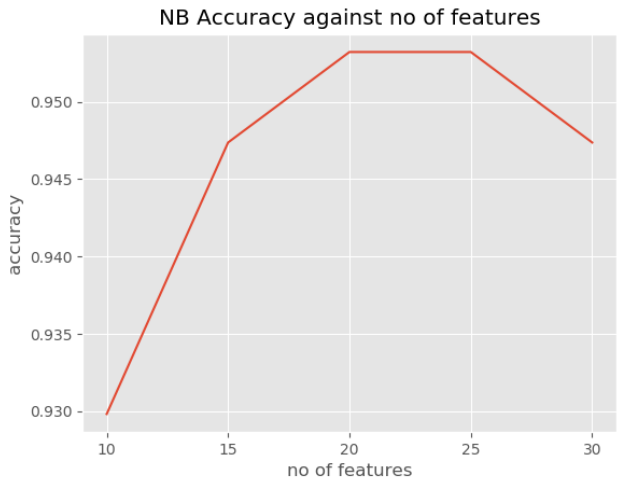
k = 17, no of features 15, accuracy 0.964912

k = 19, no of features 15, accuracy 0.964912

Hence, as we compare the accuracies of the models, we conclude that the optimal k value is 3 and the optimal number of features is 15 with the accuracy 0.994152, and the time taken to build and test the model is 0.86251 second.

4.2.2. Naive Bayes

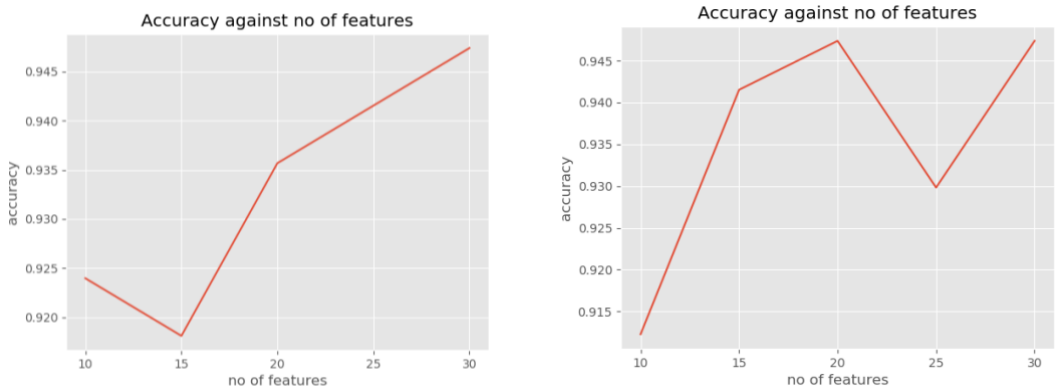
Using RFE, the insignificant features are removed, and 5 separate trainings are performed on the model using 10, 15, 20, 25, and 30 number of features. The test accuracy of each model is evaluated and compared.



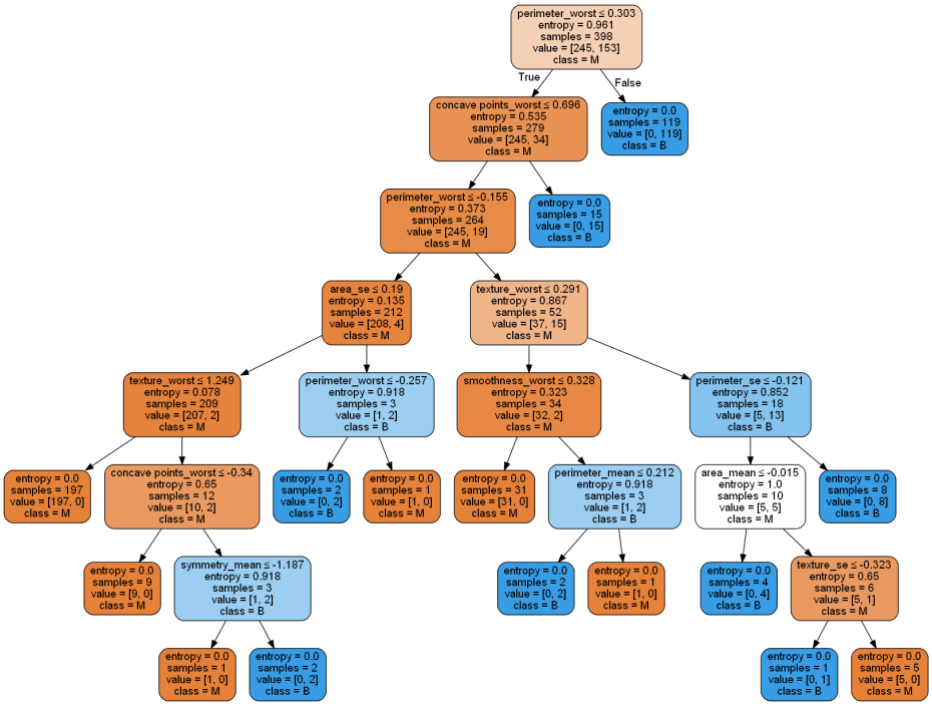
From the graph, the optimal number of features are tied at 20 and 25, with accuracy 0.953216. Thus, 20 features are selected as the optimal number of features to minimise the curse of dimensionality. The runtime to train and test the data using this optimal number of features is 0.67818 second

4.2.3. Decision Tree

Using RFE, the insignificant features are removed, and 5 separate trees are built using 10, 15, 20, 25, and 30 number of features. The test accuracy of each tree is evaluated and compared.



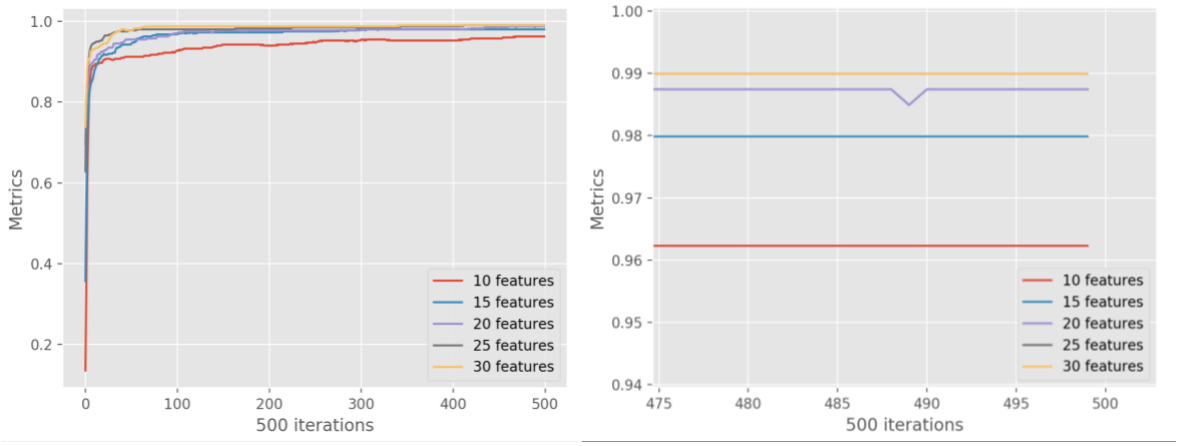
Since there can be different trees built even with the same features, the accuracy of the trees differ every time the program is run. The two graphs above shows that the optimal number of features is using all 30 features gives the highest accuracies. For the graph on the left, the accuracy for 30 features is 0.947368 and the runtime for training and test data is 0.0799 second.



The figure above shows one of the possible trees built with 30 features, and the nodes split based on the entropy values. However, not all 30 features are seen on the decision tree because the decision tree will stop splitting once the entropy = 0, meaning that the remaining data are classified as in the same class (Malignant or Benign).

4.2.4. Artificial Neural Network

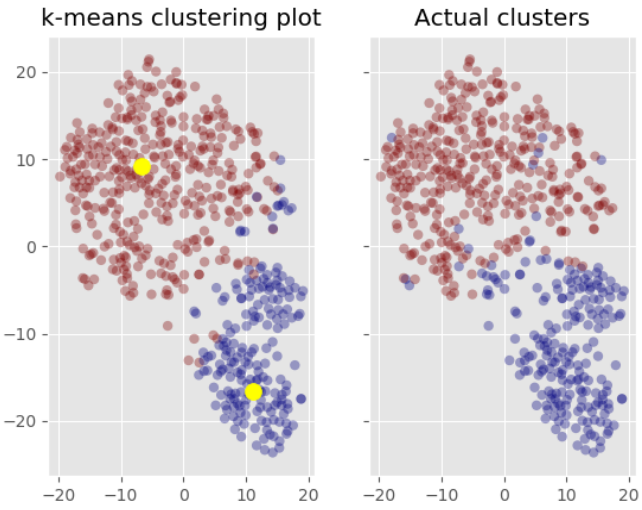
The neural network model is trained with the provided training set using 500 iterations. Using RFE, the insignificant features are removed, and 5 separate trainings are performed on the model using 10, 15, 20, 25, and 30 number of features. The test accuracy of each model is evaluated and compared.



Based on the graphs above, the accuracy of the model trained using 25 features is actually the same as that of 30 features. Hence, 25 features is the optimal number of features to reduce the curse of dimensionality. The accuracy of this approach using 25 features is 0.98111, and the time taken to build and train the model is 0.97991 seconds.

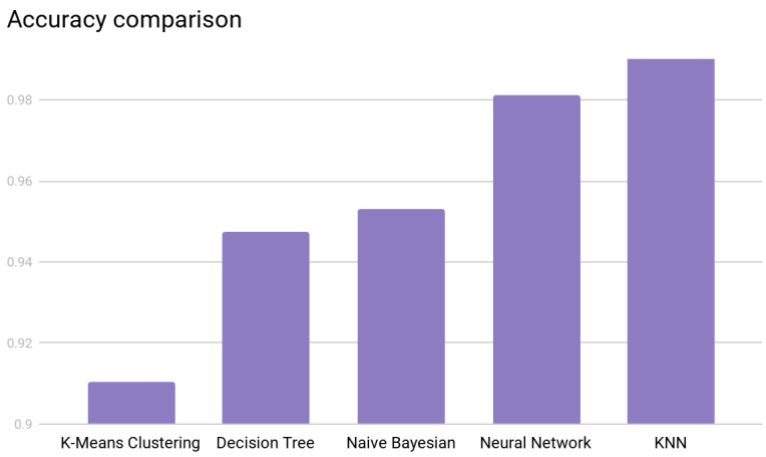
4.2.5. K-Means Clustering

After the dataset has been clustered, a 2-dimensional plot was created from all the 30 features. The k-means clustering plot shows the predicted clusters of 539 samples and 2 center points of both clusters.

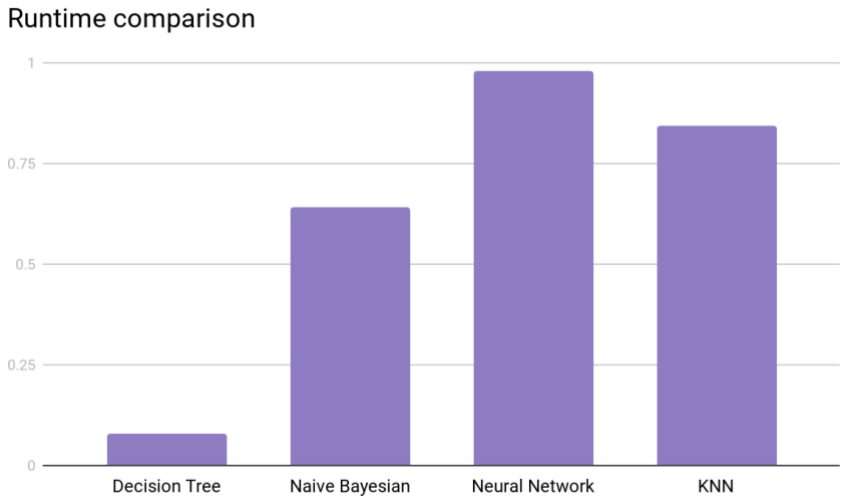


As shown in the figures above, K-Means clustering algorithm is able to produce a quite accurate model representing the dataset, with accuracy around 91%. This implies that the dataset is well separated between the benign and malignant data.

4.3. Comparison Schemes



By comparing the accuracy of each data mining approach, it can be seen that KNN has the highest accuracy. This might be due to a small dataset size and a large number of features, and as can be seen from the figure in section 4.2.5, the dataset is already clustered well.



The runtime is calculated from building the model until the training process is completed. Decision tree has the lowest runtime since it is the least complex model, whereas neural network, being the most complex model, has the highest runtime.

5. Discussion of Pros and Cons

Based on the accuracy and runtime graphs, the runtime of the decision tree is the lowest of all approaches, but it also has the lowest accuracy rate. The KNN classifier has the best accuracy, but also has a high runtime compared to the rest of the approaches. It can be concluded that there is a trade-off between accuracy and runtime.

6. Conclusions

6.1. Summary of project achievements

The KDD approach used in this project has enabled us to create several models implemented with certain algorithm that have a very high accuracy in prediction. We also managed to compare the performance (runtime and accuracy) of all the models produced and discovered that all models compensate one aspect with the other.

6.2. Directions for improvements

The current dataset only has 569 sets of data, and it is not enough to train complex models well since more complex models have more weights to train and hence, they require more data. Therefore, data augmentation should be performed to obtain a larger dataset. It is a technique to increase the diversity of data based on an existing dataset without collecting new data.

References

Cancer Statistics. (n.d.). Retrieved from National Cancer Center Singapore: https://www.nccs.com.sg/patient-care/cancer-types/cancer-statistics

Breast Cancer Campaign. (n.d.). Retrieved from Singapore Cancer Society: https://www.singaporecancersociety.org.sg/events/campaigns/breast-cancer-campaign.html

Breast Cancer Wisconsin (Diagnostic) Data Set. (n.d.). Retrieved from Kaggle: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data

What is an artificial neural network? Here's everything you need to know. (n.d.). Retrieved from Digital Trends: https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

Stochastic gradient descent. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Stochastic_gradient_descent

Supervised and Unsupervised Machine Learning Algorithms. (n.d.). Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

1000x Faster Data Augmentation. (n.d.). Retrieved from Berkeley Artificial Intelligence Research: https://bair.berkeley.edu/blog/2019/06/07/data_aug/

BREAST CANCER RISK PREDICTION USING DATA MINING CLASSIFICATION TECHNIQUES. (n.d.). Retrieved from Research Gate: https://www.researchgate.net/publication/276480208_BREAST_CANCER_RISK_PREDICTION_USING_DATA_MINING_CLASSIFICATION_TECHNIQUES

Appendix

Datasets

All datasets used in this project can be downloaded from Breast Cancer Wisconsin in Kaggle database https://www.kaggle.com/uciml/breast-cancer-wisconsin-data.

Scripts/Source Codes (if you implement your own codes)

https://github.com/Jeffrey-Huang98/CZ4032_Group18.git

Implementation Guidelines (instructions for using any tools)

Implementation guidelines can be found in the README file in the git repository.