APLIKASI PENENTUKURAN KEBARANGKALIAN TERHADAP MODEL PEMBELAJARAN GABUNGAN BAGI MERAMAL PENYAKIT BUAH PINGGANG KRONIK

AMALIA HUDA BINTI SELAMAT

UNIVERSITI KEBANGSAAN MALAYSIA

APLIKASI PENENTUKURAN KEBARANGKALIAN TERHADAP MODEL PEMBELAJARAN GABUNGAN BAGI MERAMAL PENYAKIT BUAH PINGGANG KRONIK

AMALIA HUDA BINTI SELAMAT

PROJEK YANG DIKEMUKAKAN UNTUK MEMENUHI
SEBAHAGIAN DARIPADA SYARAT MEMPEROLEHI
IJAZAH SARJANA SAINS DATA

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2022

PENGAKUAN

Saya akui karya ini adalah hasil kerja saya sendiri kecuali nukilan dan ringkasan yang tiap-tiap satunya telah saya jelaskan sumbernya.

23 Februari 2022

AMALIA HUDA BINTI SELAMAT
P104359

PENGHARGAAN

Dengan nama Allah yang Maha Pemurah lagi Maha Penyayang. Syukur kepada Allah S.W.T kerana dengan izin dan rahmat-Nya dapat saya menyiapkan kajian ini. Saya ingin merakamkan setinggi-tinggi penghargaan kepada semua yang telah membantu saya sama ada secara terus atau tidak terus. Pertama sekali, jutaan terima kasih yang tidak terhingga saya ucapkan kepada penyelia utama saya, Dr. Shahnorbanun Sahran yang sentiasa memberikan bantuan, bimbingan, ilmu, teguran dan nasihat yang berguna sepanjang saya menyiapkan kajian ini. Jutaan terima kasih juga saya kalungkan kepada kedua ibu bapa saya, Encik Selamat bin Othman dan Puan Juriah binti Jaafar dan adik-beradik saya, Fatin Najihah binti Selamat dan Iffah Liyana binti Selamat atas semangat dan doa yang tidak putus buat saya.

Saya juga ingin merakamkan setinggi-tinggi penghargaan buat rakan-rakan saya, Nur Farah Suhada binti Mohd Rosely, Fadhlin Nadhirah binti Mohd Fauzi, Nurul Nashuda binti Hambali, Puteri Nur Farahin binti Wan Abdul Rashid dan lain-lain atas segala dorongan sepanjang saya menyiapkan kajian ini. Akhir sekali, terima kasih yang tidak terhingga saya ucapkan kepada para pensyarah Fakulti Teknologi dan Sains Maklumat atas segala bantuan yang diberikan. Semoga Tuhan sentiasa merahmati kalian di mana sahaja kalian berada.

ABSTRAK

Penyakit buah pinggang kronik merupakan salah satu penyakit yang menyumbang kepada kadar kematian yang tinggi di seluruh dunia. Penyakit buah pinggang terjadi apabila berlaku kegagalan fungsi buah pinggang. Terdapat lima peringkat dalam penyakit buah pinggang kronik. Peringkat 5 memerlukan pesakit untuk menjalani rawatan dialisis atau pemindahan buah pinggang. Kadar rawatan yang tinggi bagi dialiasi dan pemindahan buah pinggang membebankan para pesakit yang sebahagiannya boleh mendapatkan rawatan pencegahan yang lebih awal sekiranya penyakit buah pinggang dapat dikesan dengan lebih awal. Simptom bagi penyakit buah pinggang kronik kebiasaannya tidak begitu spesifik. Kaedah pembelajaran mesin dalam bidang perubatan boleh membantu membuat pengesanan lebih awal bagi pesakit yang menghidap penyakit buah pinggang kronik. Kajian ini memfokuskan terhadap pembinaan model pengelasan menggunakan teknik pembelajaran mesin bagi meramal penyakit buah pinggang kronik. Data diperoleh daripada *Kaggle dataset*. Data melalui fasa pra-pemprosesan untuk meningkatkan kualiti dan memudahkan pengaplikasian teknik pembelajaran mesin. Lima algoritma model pengelasan iaitu regresi logistik, Bayes naif, mesin sokongan vektor, hutan rawak dan *multilayer perceptron* dipilih sebagai model pengelasan asas. Kaedah *recursive feature elimination* dipilih sebagai kaedah pemilihan atribut. Bagi mencapai kepelbagaian dalam model pembelajaran gabungan heterogen, kaedah penentukuran kebarangkalian diaplikasi terhadap model pengelasan asas. Ini kerana, terdapat beberapa model yang tidak boleh menjana anggaran kebarangkalian yang tepat. Kemudian, model pembelajaran gabungan dibina berdasarkan model pengelasan asas untuk meramal penyakit buah pinggang kronik. Model pembelajaran gabungan yang dipilih ialah *simple averaging*, *weighted averaging*, *stacking A* dan *stacking B*. Prestasi model pengelasan, penentukuran kebarangkalian dan model pembelajaran gabungan dinilai berdasarkan nilai ketepatan, ukuran F, AUROC, kehilangan log, skor Brier dan *expected calibration error*. Berdasarkan keputusan, kaedah pemilihan atribut membantu meningkatkan prestasi model pengelasan asas. Teknik pembelajaran gabungan *stacking A* dengan aplikasi penentukuran kebarangkalian terhadap model regresi logistik dan Bayes naif menunjukkan prestasi terbaik dengan nilai kehilangan log dan skor Brier yang paling rendah iaitu 0.0004 dan 0.0001.

APPLYING PROBABILITY CALIBRATION TO ENSEMBLE LEARNING MODEL TO PREDICT CHRONIC KIDNEY DISEASE

ABSTRACT

Chronic kidney disease is one of the leading causes of death in world. Chronic kidney disease is determined by the failure of kidney to function fully to the body. There are five stages in chronic kidney disease. For stage 5 chronic kidney disease, patients usually need to undergo dialysis treatment or kidney transplantation. The costs for these treatments are usually expensive thus becoming a burden to some of the patients who can actually get an early treatment and proper medical handling if the disease manage to be detected in its early stage. The symptoms for chronic kidney disease are usually not specific which makes it harder to detect the disease in its early stage. Machine learning technique in medical field can help to make early detection for chronic kidney disease. This study developed a classification model using machine learning technique to predict chronic kidney disease. The dataset used undergo the pre-processing stage in which the data is cleaned, transformed and reduced to increase the quality of the data and to facilitate the application of machine learning technique. Five classification algorithms used as base classifiers in this study are logistic regression, Naïve Bayes, Support Vector Machines, Random Forest and multilayer perceptron. Recursive feature elimination method is chosen as the attribute selection technique. To achieve good diversity for heterogeneous ensemble learning, probability calibration method known as Platt calibration is used to calibrate all the base classifiers. This is because, some of the classifiers are not able to produce the correct probability estimates. After that, the ensemble learning models are built by using the combination of the base classifiers to predict chronic kidney disease. Ensemble methods involved are simple averaging, weighted averaging and stacking A and stacking B. The performance of base classifiers, Platt scaling and ensemble methods are measured by using accuracy, F-measure, AUROC, log loss, Brier score and expected calibration error. The result shows that attribute selection method helped to improve the performance of base classifiers. Stacking A with calibrated logistic regression and Naïve Bayes gives the best log loss and Brier score value which are 0.0004 and 0.0001 respectively.

KANDUNGAN

Halaman

SENARAI JADUAL

SENARAI ILUSTRASI

SENARAI SINGKATAN

ANN	Artificial Neural Network
AUROC	Area Under Receiver Operating Characteristics
ALO	Ant Lion Optimization
API	Application Programming Interface
AUC	Area Under Curve
CAD	Computer Aided Diagnosis
CBOE	Chicago Board Options Exchange
CFS	Correlation-based Feature Selection
CHAID	Chi-square Automatic Interaction Detection
CKD	Chronic Kidney Disease
CNN	Convolutional Neural Network
CSV	Comma Separated Values
DLBCL	Diffuse Large B-Cell Lymphoma
DNN	Deep Neural Network
ED	Emergency Department
ECE	Expected Calibration Error
ECOC	Error Correcting Output Code
ESRD	End Stage Renal Disease
FNN	Feedforward Neural Network
FPR	False Positive Rate
FT-IR	Fourier Transform Infrared Spectroscopy
GTSRB	German Traffic Sign Recognition Benchmark
K/DOQI	Kidney Disease Quality Outcome Initiative
kNN	k-Nearest Neighbors
LS	Least Squares

LS-STM	Least Square Support Tensor Machine
LSVM	Linear Support Vector Machine
MAE	Mean Absolute Error
MIRA	Margin Infused Relaxed Algorithm
MLP	Multilayer Perceptron
MNIST	Modified National Institute Of Standards And Technology
MSE	Mean Squared Error
NHISS	National Health Insurance Sharing Services
NHMS	National Health Morbidity Survey
NPV	Negative Predicted Value
RMSE	Root Mean Squared Error
RFE	Recursive Feature Elimination
RFECV	Recursive Feature Elimination With Cross Validation
ROC	Receiver Operating Characteristics
SMOTE	Synthetic Minority Oversampling Technique
WEKA	Waikato Environment for Knowledge Analysis
P-CRA	Probabilistic Caries Risk Assessment
PLSR	Partial Least Square Regression
PNN	Probabilistic Neural Network
PPV	Positive Predicted Value
ReLU	Rectified Linear Unit
RFE	Recursive Feature Elimination
STM	Support Tensor Machine
TPR	True Positive Rate
USPS	United States Postal Service

BAB I

PENGENALAN

1.1 PENDAHULUAN

Buah pinggang merupakan salah satu organ yang penting dalam tubuh badan manusia. Di dalam tubuh badan manusia, terdapat sepasang buah pinggang yang terletak pada bahagian atas dan belakang abdomen dan dilindungi oleh tulang rusuk. Setiap buah pinggang mempunyai struktur dan fungsinya yang tersendiri. Buah pinggang berfungsi untuk membersihkan darah, membuang sisa toksik dan menghasilkan urin. Setiap hari, buah pinggang akan menapis kira-kira 120 hingga 150 kuart darah untuk menghasilkan 1 hingga 2 kuart urin (Nandhini & Aravinth 2021). Bagi orang dewasa, panjang buah pinggang adalah kira-kira 10 cm dan 6cm lebar (Lerma & Pandya 2015). Menurut *Kidney Disease Quality Outcome Initiative* (K/DOQI), penyakit buah pinggang kronik atau *chronic kidney disease* (CKD) didefinisikan sebagai kegagalan buah pinggang atau kadar penapisan gromerular kurang daripada 60 ml/min/1.73 m^2 selama tiga bulan atau lebih tanpa mengira puncanya.

Antara faktor yang menyumbang kepada CKD adalah penyakit kencing manis dan tekanan darah tinggi. CKD juga dipengaruhi oleh gaya hidup seseorang. 10% daripada populasi penduduk di seluruh dunia menghidap CKD dan setiap tahun jutaan penduduk mati kerana tiada akses kepada rawatan yang berpatutan. Purata bilangan CKD yang direkodkan telah mencapai lebih dari 200 kes bagi 1 juta populasi di dunia setiap tahun bagi kebanyakan negara. Negara seperti Amerika Syarikat, Taiwan dan sesetengah kawasan di Mexico mencatatkan lebih 400 kes bagi 1 juta penduduk setiap tahun (Wibawa et al. 2017). Terdapat 5 peringkat atau tahap dalam CKD di mana peringkat 1 adalah peringkat awal dan peringkat 5 adalah peringkat kronik. Jadual 1.1 menunjukkan peringkat CKD dan penerangan.

Jadual 1.1 Peringkat CKD dan penerangan

Peringkat CKD	**Penerangan**	**Kadar penapisan glomerular (ml/min/1.73 m$^{2)}$**
Peringat 1	Kerosakan buah pinggang tanpa pengurangan kadar penapisan gromerular dan kebiasaannya tanpa sindrom	Melebihi 90
Peringkat 2	Kerosakan buah pinggang dengan tekanan darah tinggi dan sedikit pengurangan kadar penapisan gromerular serta kemungkinan disfungsi organ-organ lain	60-89
Peringkat 3	Pengurangan yang signifikan dalam kadar penapisan gromerular, peningkatan aras urin dan creatinine dalam darah	45-89
Peringkat 4	Kegagalan fungsi buah pinggang yang serius, peningkatan yang tinggi bagi urin dan creatinine dalam darah, kegagalan fungsi organ-organ lain	15-59
Peringkat 5	Buah pinggang tidak lagi berfungsi mengikut kesesuaian tubuh badan, kegagalan dalam organ-organ badan yang lain	Kurang dari 15

Peringkat 5 juga dikenali sebagai penyakit buah pinggang tahap akhir atau *end-stage renal disease* (ESRD) yang memerlukan pesakit untuk membuat rawatan dialisis atau pemindahan buah pinggang (Lerma & Pandya 2015). Sehingga hari ini, masih tiada ubat spesifik untuk merawat CKD, akan tetapi CKD boleh dicegah dan dirawat pada peringkat awal dan progres bagi penyakit ini boleh dihentikan (Ministry of Health Malaysia 2018).

Pembelajaran mesin merupakan salah satu bidang dalam kecerdasan buatan yang memfokuskan kepada pengembangan program komputer untuk mengenal pasti pola yang terdapat dalam sesuatu data. Terdapat pelbagai jenis kaedah pembelajaran mesin iaitu pembelajaran berselia, pembelajaran tidak terselia, dan pembelajaran pengukuhan. Salah satu cara untuk merawat penyakit adalah dengan mengenal pasti penyakit di tahap paling awal. Dalam sektor kesihatan, pembelajaran mesin merupakan cabang yang semakin berkembang dan dapat diaplikasi untuk mengenal pasti penyakit di tahap awal. Cabang ini meliputi usaha untuk membantu meningkatkan taraf kesihatan dengan menganalisis dan merawat pelbagai penyakit (Tazin et al. 2017).

1.2 LATAR BELAKANG KAJIAN

Penyakit buah pinggang kronik meninggalkan kesan terhadap para pesakit. Perubahan yang ketara dapat dilihat dari segi emosi pesakit apabila disahkan menghidap penyakit kronik. Dari segi fizikal, kemerosotan berat badan secara mendadak berlaku kepada

pesakit yang dipengaruhi oleh ketiadaan selera makan dan kawalan pengambilan air. Pesakit juga perlu menjalani rawatan berkala yang mengambil masa yang panjang. Dari segi emosi, pergerakan pesakit terbatas dan menyebabkan aktiviti sosial dengan masyarakat setempat juga terbatas (Ibrahim et al. 2011).

Jumlah kos bagi rawatan dialisis pada tahun 2015 dianggarkan sebanyak RM1.5 bilion untuk 38,000 pesakit dan dianggarkan mencapai sehingga RM4 bilion pada tahun 2040 tanpa mengambil kira kadar inflasi. Walaupun dengan perkembangan yang pesat dalam peruntukan rawatan dialisis, jumlah sebenar beban ekonomi bagi sektor kerajaan masih tidak dapat dikenal pasti.

Berdasarkan kajian yang dijalankan oleh Ismail et al. (2019), jumlah perbelanjaan sektor kerajaan bagi ESRD sepanjang tempoh tujuh tahun (2010-2016) adalah sebanyak RM5.76 bilion di mana Kementerian Kesihatan Malaysia menyumbang sebanyak 55% diikuti oleh Pertubuhan Keselamatan Sosial (16%), Jabatan Perkhidmatan Awam (11%), organisasi zakat (11%), Kementerian Pertahanan (5%) dan Kementerian Pendidikan (2%). Kadar perbelanjaan bagi rawatan ESRD meningkat dengan lebih pantas jika dibandingkan dengan jumlah perbelanjaan kesihatan bagi sektor kerajaan.

Trend peningkatan perbelanjaan bagi ESRD adalah membimbangkan. Ini kerana, peruntukan bagi rawatan ESRD per kapita telah mencapai 25 kali ganda lebih tinggi berbanding jumlah perbelanjaan kesihatan per kapita yang menunjukkan semakin tinggi sumber kesihatan yang diperuntukkan untuk merawat pesakit yang menghidap ESRD yang menyumbang hanya sebahagian kecil daripada populasi penduduk Malaysia. Kadar perbelanjaan ini boleh dikurangkan dan kaedah yang lebih jimat dan berkesan boleh diaplikasi.

Penggunaan informasi teknologi di hospital dapat membantu dalam pemprosesan dan memudahkan penyimpanan maklumat para pesakit yang semakin meningkat. Secara tidak langsung, penggunaan informasi teknologi dapat menjimatkan masa dan mewujudkan satu sistem yang cekap dan sistematik bagi pengurusan rekod pesakit. Selain itu, penggunaan teknologi pembelajaran mesin dalam bidang perubatan semakin meningkat sejak akhir-akhir ini. Teknologi seperti *computer-aided diagnosis*

(CAD) dapat membantu pakar perubatan untuk mendiagnos CKD (Wibawa et al. 2017). Pelbagai kaedah berkaitan CAD telah dibina untuk membantu mengenal pasti simptom penyakit buah pinggang seperti rangkaian neuron buatan, *fuzzy logic* dan *neuro fuzzy*. Kehadiran teknologi pembelajaran mesin membantu untuk menambah baik sistem perubatan khususnya di Malaysia dan seluruh dunia secara umumnya.

1.3 PERMASALAHAN KAJIAN

Berdasarkan data yang diperoleh Kementerian Kesihatan Malaysia, CKD merupakan salah satu penyakit yang serius dan semakin meningkat dalam kalangan rakyat Malaysia. Dalam satu tinjauan yang dilaksanakan oleh Tinjauan Kebangsaan Kesihatan dan Morbiditi (NHMS) pada tahun 2011 menunjukkan sebanyak 9.07% kes CKD di kawasan semenanjung Malaysia dan meningkat kepada 15.5% pada tahun 2018. Bilangan penduduk Malaysia yang memerlukan dialisis atau pemindahan buah pinggang juga meningkat. Tahap penyakit kencing manis juga semakin melonjak dan merupakan faktor utama bagi ESRD yang menyumbang sebanyak 65% pada tahun 2016 (Ministry of Health Malaysia 2018). Negara Malaysia merupakan antara negara yang mencatatkan kes ESRD yang paling tinggi kesan daripada penyakit kencing manis (Saran et al. 2018).

Bilangan penduduk Malaysia yang menghidap CKD dijangka semakin meningkat pada masa hadapan disebabkan oleh penyakit kencing manis dan darah tinggi. Berdasarkan data *Malaysian Dialysis and Transplant Registry* (MDTR) pada tahun 2015, sebanyak 58% pesakit yang menjalani dialisis adalah yang berumur 55 tahun dan ke atas (Ministry of Health Malaysia 2018). Pada tahun 2016, sebanyak 39,711 pesakit menjalani dialisis dan 1,814 menjalani rawatan pemindahan buah pinggang (Ismail et al. 2019).

CKD pada peringkat awal kebiasaannya berlaku tanpa sebarang simptom dan kekurangan pakar kesihatan nefrologi mengakibatkan sebilangan pesakit yang menghidap CKD tidak dapat menjalani diagnosis yang tepat. Diagnosis bagi peringkat awal penyakit ini adalah penting kerana dapat mengelakkan kadar kegagalan buah pinggang sebanyak 50% (Ministry of Health Malaysia 2018). CKD masih boleh dielakkan dengan langkah pencegahan awal. Disebabkan tahap CKD adalah berbeza

dan sukar dijangka pada peringkat awal, ramalan yang tepat bagi CKD menggunakan kaedah pembelajaran mesin dapat membantu pakar kesihatan khususnya doktor untuk membantu pesakit mengambil langkah pencegahan awal (Wibawa et al. 2017). Sehubungan dengan itu, penggunaan teknologi seperti model peramalan dengan menggunakan teknik pembelajaran mesin dipercayai dapat membantu pakar kesihatan seperti doktor untuk membuat keputusan secara tepat dan mampu memberi rawatan yang sepatutnya kepada para pesakit dalam tempoh waktu yang singkat. Kaedah penentukuran kebarangkalian dalam data perubatan adalah penting kerana ketepatan model ramalan kadangkala mengelirukan. Model pengelasan seperti mesin sokongan vektor dan Bayes naif tidak memberikan kebarangkalian yang tepat. Lebih tepat kebarangkalian kepada nilai kelas sebenar, lebih baik sesuatu model itu. Oleh itu, kaedah penentukuran kebarangkalian seperti kaedah Platt dapat membantu meningkatkan keyakinan dan kualiti dalam ramalan CKD.

1.4 OBJEKTIF KAJIAN

Objektif bagi kajian ini adalah:

1. Mengkaji model pengelasan bagi meramal pesakit-pesakit yang menghidap CKD dan tidak menghidap CKD.
2. Meningkatkan prestasi model pengelasan yang digunakan dengan mengaplikasi kaedah penentukuran kebarangkalian dan menggunakan pembelajaran gabungan.
3. Membandingkan prestasi model pengelasan dan model pembelajaran gabungan sebelum dan selepas mengaplikasi kaedah penentukuran kebarangkalian dan memilih model peramalan yang terbaik.

1.5 SKOP KAJIAN

Skop kajian ditetapkan untuk memberi fokus pada kajian supaya tidak tersasar dari tujuan kajian dan mencapai objektif yang ditetapkan. Data kajian yang digunakan diperoleh dari *Kaggle* dataset dalam bentuk *comma-separated values* (CSV) yang terdiri daripada data angka dan data nominal yang mana data ini meliputi pesakit-

pesakit yang menghidap penyakit buah pinggang di sebuah hospital di India. Tempoh masa data diambil adalah selama dua bulan.

1.6 KEPENTINGAN KAJIAN

Kajian ini dijangka memberi pengetahuan mengenai kepentingan penentukuran kebarangkalian bagi model pengelasan untuk data perubatan.

1.7 STRUKTUR TESIS

Laporan ini memerihalkan keseluruhan proses pembangunan projek iaitu dari proses perancangan sehingga ke proses pelaksanaan. Setiap proses akan dibincangkan dan dilaporkan secara berasingan. Laporan ini terbahagi kepada lima bab. Ringkasan bab seterusnya adalah seperti di bawah:

Bab II menjelaskan mengenai kajian-kajian yang telah dilakukan oleh para penyelidik yang lain berkaitan pelbagai teknik klasifikasi CKD dan penentukuran kebarangkalian.

Bab III menerangkan keseluruhan proses dalam kajian ini. Metodologi yang digunakan untuk membangun model klasifikasi CKD akan dibincangkan, di mana ia melibatkan perbandingan dan pengujian antara teknik-teknik klasifikasi. Bab ini juga akan membincangkan tentang aplikasi penentukuran kebarangkalian secara terperinci.

Bab IV membentangkan hasil kajian dan penilaian terhadap prestasi teknik-teknik klasifikasi yang digunakan mengklasifikasi CKD.

Bab V merupakan bab terakhir. Bab ini akan membincangkan rumusan tentang keseluruhan kajian dan cadangan penambahbaikan untuk kajian masa hadapan.

BAB II

KAJIAN LITERASI

2.1 PENDAHULUAN

Bab ini akan menyentuh mengenai kajian lepas yang membincangkan mengenai teknik-teknik klasifikasi yang digunakan ke atas data diagnosis penyakit buah pinggang kronik (CKD). Selain itu, bab ini juga akan menjelaskan tentang penentukuran kebarangkalian dan evolusinya berserta kajian-kajian lampau mengenai penentukuran kebarangkalian.

2.2 KAJIAN LAMPAU MENGENAI PEMBELAJARAN MESIN MENGGUNAKAN DATA DIAGNOSIS CKD

Tazin et al. (2017) mengaplikasikan model seperti Bayes naif, pokok keputusan, mesin sokongan vektor dan *k-Nearest Neighbors* (kNN) dalam mengklasifikasi CKD. Data CKD diperoleh daripada UCI *Machine Learning Repository*. Parameter-parameter yang terlibat untuk menilai prestasi model tersebut adalah *Kappa Statistics* (K), min ralat mutlak (MAE), punca min ralat kuasa dua (RMSE), luas bagi lengkung *Receiver Operating Characteristics* (AUROC) dan ketepatan. Metode pemilihan fitur yang digunakan adalah berdasarkan algorithma *ranking* dan 15 daripada 25 atribut telah dipilih. Perisian yang digunakan dalam kajian ini ialah *Waikato Environment for Knowledge Analysis* (WEKA). *10-fold cross validation* digunakan untuk membahagi data latihan dan data pengujian. Kajian ini dijalankan dengan dua kaedah. Kaedah pertama adalah sebelum implementasi algoritma *ranking* dan kaedah kedua adalah selepas implementasi algoritma *ranking*. Bagi kaedah pertama, algoritma pokok keputusan mencatatkan nilai K, ROC dan ketepatan yang tertinggi iaitu masing-masing dengan nilai 0.979, 0.999 dan 99%. Bagi algoritma kNN, jika nilai k=1, nilai ketepatan adalah 95.75%, jika k=2, nilai ketepatan meningkat kepada 96.25% dan berkurang kepada 94.75% jika nilai k=3. Bagi kaedah kedua, selepas implementasi algoritma *ranking*, prestasi algoritma-algoritma yang terlibat meningkat kecuali pokok keputusan. Nilai K bagi algoritma Bayes naif meningkat dari 0.896 kepada 0.916, mesin sokongan

vektor meningkat dari 0.953 kepada 0.968 dan kNN meningkat kepada 0.947 dari 0.911. Akhir sekali, nilai ketepatan bagi model tersebut dicatatkan bagi bilangan atribut yang berbeza iaitu 25, 20, 15 dan 10. Berdasarkan keputusan perbandingan prestasi mengikut bilangan atribut, bilangan atribut yang optimal adalah 15 yang mencatatkan nilai ketepatan yang paling tinggi berbanding lain-lain bilangan. Sebagai kesimpulan, pemilihan atribut membantu meningkatkan prestasi model pengelasan.

Selain itu, menurut kajian yang dilakukan oleh Wibawa et al. (2017) kaedah pemilihan fitur dan pembelajaran gabungan digunakan untuk meningkatkan prestasi model pengelasan CKD. Data CKD diperoleh daripada UCI *Machine Learning Repository*. 24 atribut (tidak termasuk atribut '*classification*') telah dikurangkan kepada 17 atribut dengan menggunakan metode *Correlation-based Feature Selection* (CFS). Algoritma-algoritma pengelasan yang terlibat sebagai pengelasan asas adalah kNN, mesin sokongan vektor dan Bayes naif. Kemudian, algoritma *AdaBoost* digunakan untuk mengklasifikasi pesakit-pesakit CKD. *10-fold cross validation* digunakan untuk membahagi antara data latihan dan data pengujian. Ketepatan, kejituan, *recall* dan ukuran F merupakan parameter-parameter yang diambil kira untuk mengukur prestasi algoritma-algoritma tersebut. Terdapat tiga kaedah pengelasan yang berbeza dijalankan iaitu pengelasan dengan penggunaan algoritma pengelasan asas sahaja tanpa pemilihan fitur dan pembelajaran gabungan, penggunaan pengelasan asas dengan pemilihan fitur dan kaedah terakhir adalah penggunaan kaedah pemilihan fitur dan pembelajaran gabungan. Bagi kaedah pertama, kesemua algoritma pengelasan asas memperlihatkan lebih dari 0.94 bagi ketepatan, kejituan, *recall* dan ukuran F. Kaedah kedua mencatatkan peningkatan bagi kesemua parameter algoritma. Kadar ketepatan bagi Bayes naif dan mesin sokongan vektor meningkat sebanyak 0.005 manakala kNN meningkat sebanyak 0.020. Bagi kaedah terakhir, kadar ketepatan juga meningkat bagi ketiga-tiga algoritma. Ketepatan bagi Bayes naif adalah 0.980, 0.981 bagi kNN dan 0.975 bagi mesin sokongan vektor. Pemilihan fitur dan pembelajaran gabungan berjaya meningkatkan kadar kejituan algoritma pengelasan asas. Walau bagaimanapun, kadar *recall* tidak meningkat dalam kesemua algoritma. Berdasarkan kajian ini, CFS dan *AdaBoost* menambah baik diagnosis CKD kerana terdapat peningkatan prestasi bagi kesemua algoritma pengelasan yang terlibat.

Dalam kajian yang telah dilaksanakan oleh Shankar et al. (2018), model rangkaian neuron dalam (DNN) telah dicadangkan bagi mengelaskan CKD. Data CKD diperoleh daripada UCI *Machine Learning Repository*. *Ant Lion Optimization* (ALO) digunakan sebagai kaedah pemilihan fitur. Rangkaian neuron buatan (ANN) dilengkapi beberapa lapisan tersembunyi dan output dinamakan sebagai DNN. Parameter-parameter yang digunakan untuk mengukur prestasi DNN adalah ketepatan, kejituan, sensitiviti dan kespesifikan. Prestasi bagi model DNN yang dicadangkan dibandingkan dengan model NN, *Convolutional Neural Network* (CNN), *Back spread* dan kNN. Berdasarkan prestasi model tersebut, model DNN mencatatkan keputusan yang bermanfaat dengan nilai ketepatan 96.63%, kejituan 90.45%, kespesifikan 91.22% dan sensitiviti 98.22%. Kesimpulannya, model DNN yang dicadangkan menunjukkan prestasi yang terbaik.

Algoritma *Probabilistic Neural Network* (PNN), *Multilayer Perceptron* (MLP), mesin sokongan vektor dan fungsi radial basis telah digunakan dalam kajian oleh Rady dan Anwar (2019). Data latihan dan data pengujian dibahagi mengikut teknik *k-fold cross validation*. Prestasi setiap algoritma dianalisis dinilai menggunakan sensitiviti, kespesifikan, ketepatan, kejituan dan ukuran F. Kehilangan data pula diisi dengan nilai median. Dalam kajian ini, klasifikasi CKD dijalankan bagi setiap peringkat penyakit bermula dari peringkat pertama hingga peringkat kelima. Bagi peringkat pertama, PNN mencatatkan nilai ketepatan, kejituan dan ukuran F yang tertinggi iaitu 99.7%, 98.7% dan 99.37%. Sebaliknya, MLP mencatatkan nilai ketepatan, kejituan dan ukuran F yang paling rendah iaitu 77.29%, 44% dan 21.15%. Bagi peringkat kedua penyakit, PNN juga mencatatkan nilai ketepatan, kejituan dan ukuran F tertinggi iaitu 98.9%, 98.7% dan 97.5%. Nilai-nilai yang paling rendah dicatatkan oleh MLP dengan nilai 71.47% bagi ketepatan, 42.03% bagi kejituan dan 52.97% bagi ukuran F. Peringkat ketiga penyakit masih menunjukkan PNN memberikan nilai-nilai tertinggi bagi ketepatan, kejituan dan ukuran F iaitu 96.96%, 89% dan 93.6%. Mesin sokongan vektor mencatatkan nilai-nilai terendah iaitu 73.68% bagi ketepatan, 44.04% bagi kejituan dan 50.26% bagi ukuran F. PNN masih menunjukkan prestasi yang baik bagi peringkat keempat penyakit dengan nilai 99.7% bagi ketepatan, 100% bagi kejituan dan 99.1% bagi ukuran F. Mesin sokongan vektor masih menunjukkan nilai-nilai yang paling rendah iaitu 80.33% bagi ketepatan, 34.09% bagi kejituan dan 29.7% bagi ukuran F.

Untuk peringkat kelima, PNN masih mencatatkan nilai-nilai yang paling tinggi bagi ketepatan, kejituan dan ukuran F iaitu 98%, 100% dan 94%. Mesin sokongan vektor mencatatkan nilai 90.58% ketepatan, 71.88% kejituan dan 73.02% ukuran F sekaligus mencatatkan nilai paling rendah. Bagi peringkat ketiga hingga kelima, mesin sokongan vektor mencatatkan nilai-nilai paling rendah walaupun ketiga-tiga nilai mencatatkan 100% semasa latihan. Dari segi masa pemprosesan, fungsi radial basis menunjukkan masa yang paling lama iaitu 149 saat berbanding PNN yang mencatatkan hanya 12 saat. Kesimpulannya, PNN adalah algoritma terbaik bagi kajian ini.

Almasoud dan Ward (2019) menjalankan kajian pembelajaran mesin bagi pengelasan CKD menggunakan bilangan atribut yang paling kecil. Empat model pengelasan telah digunakan dalam kajian ini iaitu regresi logistik, mesin sokongan vektor, hutan rawak dan *gradient boosting*. Kaedah pemilihan fitur dijalankan dengan mencari korelasi antara atribut. Bilangan atribut yang digunakan dalam pengelasan CKD ialah tiga dan atribut yang terlibat ialah albumin, hemoglobin dan *sepcific gravity*. Set data latihan dan pengujian dibahagikan mengikut 10-*fold cross validation.* Prestasi bagi setiap model dinilai menggunakan ketepatan, ukuran F, kejituan, sensitiviti, kespesifikan dan AUC. Berdasarkan keputusan, ketepatan bagi kesemua model pengelasan melebihi 97%. Keputusan paling baik dicatatkan oleh model gradient boosting dengan nilai 99.1% bagi ukuran F, 98.8% bagi sensitiviti dan 99.33% bagi kespesifikan.

Dalam kajian lain yang dijalankan oleh Pasadana et al. (2019), tiada pemilihan fitur dilaksanakan. Data CKD diperoleh daripada UCI *Machine Learning Repository*. Perisian yang digunakan adalah WEKA dan *10-fold cross validation* digunakan untuk membahagi data latihan dan data pengujian. 11 teknik algoritma pokok keputusan telah digunakan iaitu *DecisionStump, HoeffdingTree, J48, CTC, J48graft, LMT, NBTree, RandomForest, RandomTree, RepTree,* dan *SimpleCart.* Prestasi setiap algoritma dianalisis dan dinilai menggunakan ketepatan, kejituan, *recall,* MAE, ukuran F, K dan masa. Berdasarkan keputusan, algoritma *RandomForest* mempunyai ketepatan yang tertinggi iaitu 100%. Selain itu, *RandomForest* juga mencatatkan nilai kejituan, *recall,* ukuran F dan K yang tertinggi dengai nilai 1 diikuti dengan algoritma *J48, J48graft, dan NBTree* yang mana masing-masing mencatatkan kadar ketepatan 99%, 98,75% dan

98,5%. Algoritma yang mencatakan kadar ketepatan yang paling rendah adalah *DecisionStump* dengan nilai 92%. Berdasarkan masa, algoritma *RandomTree, DecisionStump* dan *J48* adalah lebih pantas dan mengklasifikasi CKD. *RandomTree* mencatatkan masa 0 manakala bagi *DecisionStump* dan *J48* mencatatkan masa 0.01. Algoritma *NBTree* mencatatkan masa paling lama iaitu 2.99. Berdasarkan kajian ini, algoritma *RandomForest* mencatatkan keputusan yang paling bermanfaat.

Tambahan lagi, kajian yang telah dijalankan oleh Jongbo et al. (2020) menggunakan pembelajaran gabungan untuk mengklasifikasi CKD. Data CKD diperoleh daripada UCI *Machine Learning Repository*. Algoritma pengelasan asas yang digunakan dalam kajian ini adalah Bayes naif, kNN dan mesin sokongan vektor. Tiada kaedah pemilihan fitur dilakukan. Set data dibahagi kepada 70% data latihan dan 30% data pengujian. Teknik *backfill* telah diaplikasikan bagi mengisi ketiadaan data di mana nilai selepas ketiadaan data digunakan untuk mengisi nilai ketiadaan data tersebut. Kaedah pembelajaran gabungan *bagging* dan *random subspace* adalah dua teknik pembelajaran gabungan yang digunakan. Algoritma pembelajaran gabungan *Random subspace* memilih subset fitur secara rawak daripada set data yang original dengan penggantian dan kemudian algoritma ini mempelajari model pengelasan asas yang digunakan hanya berdasarkan pada subset fitur tersebut. Kaedah ini membolehkan model individu tidak terlalu fokus kepada atribut yang tinggi kadar ramalannya dalam data latihan. Kaedah ini juga meliputi secara umum apabila terdapat fitur yang bertindih di dalam set data. K, sensitiviti, kespesifikan, ketepatan dan ROC merupakan parameter-parameter yang digunakan untuk menilai prestasi model tersebut. Selepas kajian dilakukan dan menurut prestasi model pengelasan asas, kNN menunjukkan prestasi yang bermanfaat dengan nilai ketepatan 0.950, K dengan nilai 0.894 dan ROC dengan nilai 0.960. Nilai ketepatan bagi Bayes naif adalah 0.942 dan pokok keputusan adalah 0.892. Teknik pembelajaran gabungan *bagging* berhasil bagi meningkatkan ketepatan bagi kesemua model pengelasan asas dan kNN masih mencatatkan nilai ketepatan yang tertinggi iaitu 0.983. Bayes naif dan pokok keputusan mencatatkan nilai ketepatan 0.950. Bagi teknik pembelajaran gabungan *random subspace*, ketepatan bagi kNN dan pokok keputusan meningkat manakala tiada perubahan bagi Bayes naif. kNN mencatatkan nilai 1.000 bagi kesemua parameter. Berdasarkan kajian ini, pembelajaran gabungan berhasil dalam meningkatkan prestasi model pengelasan asas.

Dalam kajian yang dijalankan oleh Nandhini dan Aravinth (2021), sebanyak 11 model pengelasan yang digunakan untuk mengklasifikasi CKD iaitu regresi logistik, kNN, *Support Vector Classifier* dengan kernel linear, *Support Vector Classifier* dengan kernel fungsi radial basis, Bayes naif, pokok keputusan, hutan rawak, *XGBoost*, *Extra Tree*, *AdaBoost* dan rangkaian neuron (NN). Data CKD diperoleh daripada UCI *Machine Learning Repository*. Set data dibahagikan kepada 70% data latihan, 15% data pengesahan bersilang dan 15% data pengujian. Dalam kajian ini, atribut-atribut yang mengandungi lebih daripada 20% kehilangan data telah dikecualikan. Bagi atribut-atribut lain yang mengandungi ketiadaan data, algoritma *k-Nearest Neighbor Imputer* telah digunakan untuk mengisi ketiadaan data tersebut. Kemudian, kaedah pemilihan fitur menggunakan korelasi atribut diaplikasikan. Selepas mengambil kira pengagihan atribut dan perspektif perubatan, tujuh daripada 24 atribut (tidak termasuk attribute '*classification*') telah dipilih untuk meramalkan diagnosis CKD. Selepas latihan, pengesahan bersilang dan ujian, kesemua model pengelasan yang digunakan mencatatkan ketepatan lebih daripada 90% dan enam daripada 11 model tersebut menunjukkan prestasi yang terbaik. Model pengelasan tersebut adalah pokok keputusan, hutan rawak, *XGBoost*, *Extra Tree*, *AdaBoost* dan kNN. Bagi pengesahan bersilang dan ujian, model ini menunjukkan ketepatan 100% kecuali kNN. Sebagai kesimpulan, pemilihan fitur dan pengetahuan domain membantu meningkatkan prestasi model pengelasan.

Chittora et al. (2021) menggunakan tiga kaedah pemilihan fitur untuk pengelasan CKD iaitu CFS, kaedah *forward feature selection* dan kaedah *Least Absolute Shrinkage and Selection Operator* (LASSO). Perisian yang digunakan untuk pengelasan CKD adalah IBM SPSS. Algoritma-algoritma pengelasan yang digunakan dalam kajian ini adalah ANN dengan tiga lapisan tersembunyi, C5.0, regresi logistik, mesin sokongan vektor linear (LSVM), kNN, *Chi-square Automatic Interaction Detection* (CHAID) dan *Random Tree*. Kaedah yang digunakan untuk mengatasi masalah keseimbangan data adalah *Synthetic Minority Oversampling Technique* (SMOTE). Parameter-parameter yang digunakan untuk mengukur prestasi model pengelasan adalah matriks kekeliruan, ketepatan, kejituan, ralat klasifikasi, *recall*, ukuran F, ROC dan luas di bawah lengkung (AUC) dan pekali Gini. Set data dibahagi kepada 50% data latihan dan 50% data pengujian. Beberapa perbandingan prestasi

dilakukan. Antaranya, prestasi model pengelasan tanpa kaedah pemilihan fitur dan SMOTE. Model C5.0 mencapai nilai ketepatan tertinggi dengan nilai 96.10%, 92.40% bagi kejituan dan 97.30% bagi *recall*, 94.80% bagi ukuran F, AUC dengan nilai 97.80%, pekali Gini dengan nilai 0.96. Sebaliknya, kNN dengan k=5 mencatatkan nilai ketepatan yang paling rendah iaitu 64.39%, kejituan dengan nilai 59.01% dan *recall* dengan nilai 96%. Kemudian, prestasi model pengelasan dengan pemilihan fitur tanpa kaedah SMOTE dinilai. Kaedah CFS mengandungi enam fitur daripada 24. Model LSVM mencatatkan ketepatan yang paling tinggi iaitu 95.12%, kejituan dan *recall* dengan nilai 93.34%. Regresi logistik mencatatkan nilai ketepatan yang paling rendah iaitu 51.22%, kejituan dengan nilai 96.87% dan *recall* dengan nilai 92.54%. Bagi kaedah *forward feature selection*, enam fitur telah dipilih dan C5.0 mencatatkan nilai ketepatan yang paling tinggi iaitu 96.10%, 98.55% bagi kejituan, 90.67% bagi *recall*. Model kNN dengan k=5 mencatatkan nilai ketepatan yang terendah iaitu 76.10%, 95.58% bagi kejituan dan 95.58% bagi *recall*. Bagi kaedah LASSO enam fitur juga telah dipilih. Model LSVM dan CHAID menunjukkan nilai ketepatan yang tertinggi iaitu 97.07%. LSVM menunjukkan nilai 98.59% bagi kejituan dan 93.33% bagi *recall*. CHAID menunjukkan nilai kejituan 100% dan *recall* 92%. Model yang menujukkan ketepatan paling rendah dengan nilai 56.59% adalah kNN dengan k=5. Nilai kejituan adalah 92% dan *recall* adalah 100%. Bagi kaedah menggunakan SMOTE dan LASSO, model seperti ANN, CHAID, LSVM dan *Random Tree* dipilih kerana menunjukkan prestasi yang baik dalam kaedah-kaedah sebelum ini. Kaedah ini menunjukkan prestasi yang lebih baik berbanding kaedah SMOTE tanpa LASSO. LSVM mencatatkan nilai ketepatan yang tertinggi iaitu 98.46%, kejituan dengan nilai 98.59% dan *recall* dengan nilai 97.22%. Keempat-empat model menunjukkan prestasi yang lebih baik dengan SMOTE. Bagi kaedah SMOTE tanpa pemilihan fitur, LSVM mencatatkan nilai ketepatan 98.86%, 96.67% bagi kejituan dan 100% bagi *recall*. Sebagai kesimpulan, kaedah SMOTE adalah kaedah yang terbaik bagi mengatasi ketidakseimbangan data dalam kajian ini.

Senan et al. (2021) mengaplikasikan mesin sokongan vektor, kNN, pokok keputusan dan hutan rawak sebagai model pengelasan pesakit CKD dan bukan CKD. Terdapat 400 orang pesakit dan 24 fitur dalam data CKD yang diperoleh daripada *University of California, Irvine Machine Learning Repository*. Kehilangan data untuk

atribut nominal diisi dengan mod manakala kehilangan data untuk atribut numeric diisi dengan min. Bagi pemilihan fitur, teknik *Recursive Feature Elimination* (RFE) diaplikasikan untuk memilih atribut-atribut yang paling penting. Kemudian, korelasi antara atribut untuk mencari korelasi positif dan korelasi negative dilakukan. Parameter-parameter yang terlibat untuk menilai prestasi model pengelasan adalah ketepatan, kejituan, *recall* dan ukuran F. Set data dibahagi kepada 75% data latihan dan 25% data pengujian. Berdasarkan keputusan, model hutan rawak mencatatkan prestasi yang paling baik dengan 100% ketepatan, kejituan, *recall* dan ukuran F. Model terbaik selepas hutan rawak adalah pokok keputusan yang mencatatkan 99.17% ketepatan. Algoritma kNN mencatatkan 98.33% ketepatan dan mesin sokongan vektor mencatatkan ketepatan 96.67%. Prestasi algoritma yang digunakan dalam kajian ini dibandingkan dengan prestasi algoritma dalam kajian-kajian sebelum ini yang menggunakan set data yang sama. Model yang dicadangkan memberikan keputusan yang bermanfaat.

Kajian yang telah dijalankan oleh Wang et al. (2021) menggunakan CKD set data yang mempunyai 1 juta pesakit dan 24 atribut. Set data diperoleh daripada *National Health Insurance Sharing Service* (NHISS). Tiga atribut baru telah ditambah untuk tujuan kajian ini. Dua jenis eksperimen dijalankan. Eksperimen pertama adalah berkenaan regresi menggunakan 23 atribut dan *creatinine* sebagai atribut sasaran. Eksperimen yang kedua berkenaan dengan pengelasan CKD menggunakan prediktor *creatinine*. Model regresi yang terlibat termasuk hutan rawak, *XGBoost* dan *ResNet*. Kaedah penilaian model adalah menggunakan MSE dan MAE. Kaedah undersampling digunakan untuk mengatasi masalah ketidakseimbangan data. Bagi eksperimen pertama, algoritma pembelajaran gabungan *bagging* diaplikasikan menggunakan set-set *undersampled data* yang berbeza untuk melatih prediktor-prediktor. Nilai R-kuasa dua (R^2) digunakan untuk tujuan menilai prestasi model regresi. Lebih besar R^2, lebih bermanfaat model tersebut. *XGBoost* menunjukkan nilai R^2 yang paling besar iaitu 0.5523 berbanding hutan rawak dengan nilai R^2 0.5343 dan *ResNet* dengan nilai R2 0.5297. Kaedah *cost-sensitive loss function* diaplikasikan untuk meningkatkan prestasi model *XGBoost*. R^2 meningkat kepada 0.5546 selepas *cost-sensitive lost function* diaplikasikan. Bagi eksperimen kedua, kaedah pembelajaran gabungan digunakan

untuk mengklasifikasi CKD. Nilai R^2 meningkat kepada 0.5590 bagi kaedah pembelajaran gabungan.

2.3 PENENTUKURAN KEBARANGKALIAN

Platt (1999) mencadangkan teknik bagi penyesuaian fungsi sigmoid kepada output model pengelasan mesin sokongan vektor. Output bagi mesin sokongan vektor merupakan nilai yang tidak ditentukur dan nilai tersebut bukan nilai kebarangkalian. Penentukuran kebarangkalian merupakan kaedah untuk mengekstrak kebarangkalian yang berguna dalam fasa pasca pemprosesan daripada output mesin sokongan vektor. Persamaan sigmoid adalah seperti Persamaan 2.1 berikut:

$$P(y = 1|f) = p_i = \frac{1}{1 + e^{(Af_i + B)}} \tag{2.1}$$

Di mana $f(x)$ merupakan skor pengelasan, A dan B merupakan parameter yang ditemui nilainya dengan *maximum likelihood estimation* bagi data latihan:

$$min - \sum_i t_i \log(p_i) + (1 - t_i)\log(1 - p_i) \tag{2.2}$$

Untuk mengelakkan *overfitting*, *cross-validation* boleh digunakan. Tetapi, Platt mencadangkan supaya mentransformasi label y kepada kebarangkalian sasaran:

$$t_+ = \frac{N_+ + 1}{N_+ + 2} \text{ untuk sampel positif } (y = 1) \tag{2.3}$$

$$t_- = \frac{1}{N_- + 2} \text{ untuk sampel negatif } (y = -1) \tag{2.4}$$

Pada awalnya, penentukuran kebarangkalian digunakan pada model mesin sokongan vektor, tetapi lama-kelamaan penggunaan kaedah ini semakin berkembang dan diuji pada model pengelasan yang lain. Berikut merupakan evolusi penentukuran kebarangkalian.

Bennett (2000) mentaksir penentukuran model pengelasan Bayes naif. Penentukuran kebarangkalian diaplikasikan kepada output model Bayes naif. Set data yang digunakan ialah set data Reuters 21578 untuk pengelasan teks dan mengandungi 90 kelas. Tetapi di dalam makalah ini, hanya dua kelas difokuskan iaitu kelas *Corn* dan *Earn.* Bagi menilai metode yang diaplikasi, MSE bagi penganggaran *posterior* dan prestasi model pengelasan diperiksa menggunakan ukuran F. Rajah *reliability* bagi kedua-dua kelas diplot dan rajah *reliability* bagi kelas Corn menunjukkan penentukuran kebarangkalian menambah baik nilai MSE dan menunjukkan konsistensi bagi kelas *Earn*. Apabila penentukuran kebarangkalian diaplikasi pada kesemua 90 kelas, 83 daripadanya menunjukkan penambahbaikan dari nilai MSE. Berdasarkan keputusan akhir, ukuran F berkurang dengan signifikan selepas kaedah penentukuran kebarangkalian. Sebagai konklusi, penggunaan fungsi sigmoid yang lain perlu diteroka untuk menentukur model Bayes naif.

Niculescu-Mizil dan Caruana (2005) menggunakan dua kaedah penentukuran kebarangkalian iaitu kaedah Platt dan regresi isotonik pada 10 model pengelasan berselia iaitu mesin sokongan vektor, NN, pokok keputusan, *Memory-based Learning, Bagged Trees,* hutan rawak, *Boosted Trees, Boosted Stumps,* Bayes naif dan regresi logistik. Set data pengelasan binari digunakan iaitu *ADULT, COV_TYPE, LETTER, HS IndianPine92* dan *SLAC*. Histogram kebarangkalian sebelum dan selepas penentukuran kebarangkalian menunjukkan histogram-histogram adalah lebih serupa antara satu sama lain selepas kaedah Platt diaplikasi. Penentukuran mengurangkan perbezaan antara kebarangkalian bagi model tersebut secara signifikan. Penentukuran tidak mampu untuk membetulkan ramalan bagi model pokok keputusan dan Bayes naif. Penentukuran kebarangkalian bertindak secara paling efektif sekiranya set data adalah kecil. Selepas penentukuran, model yang memberikan kebarangkalian terbaik adalah *Boosted Trees,* hutan rawak, mesin sokongan vektor, *Bagged Trees* yang tidak ditentukur dan NN yang tidak ditentukur.

Lin et al. (2007) mencadangkan kaedah penambahbaikan untuk menyelesaikan masalah *regularized maximum likelihood.* Kaedah yang digunakan oleh John C. Platt adalah algoritma Levenberg-Marquardt yang mempunyai satu kelebihan iaitu keringkasan. Kaedah yang dicadangkan dalam kajia ini adalah menggunakan kaedah

Newton dengan pematahbalikan (*Newton's method with backtracking*). Implementasinya adalah seperti Persamaan 2.5 di bawah:

$$-(t_i \log(p_i) + (1 - t_i)\log(1 - p_i)) \tag{2.5}$$

$$= (t_i - 1)(Af_i + B) + \log(1 + e^{(Af_i + B)}) \tag{2.6}$$

$$= t_i(Af_i + B) + \log(1 + e^{(-Af_i - B)}) \tag{2.7}$$

Jika nilai $Af_i + B \geq 0$, maka Persamaan 2.7 digunakan. Jika tidak, maka Persamaan 2.6 digunakan.

Dua set data digunakan untuk membandingkan prestasi algoritma Platt yang asal dan yang telah ditambah baik iaitu set data *sonar* dan *shuttle*. Berdasarkan keputusan akhir, algoritma Platt mencatatkan prestasi yang baik pada set data *sonar* dan menunjukkan beberapa ralat dalam set data *shuttle*. Manakala algoritma yang dicadangkan mencatatkan prestasi yang baik bagi kedua-dua set data.

2.4 KAJIAN LAMPAU MENGENAI PENENTUKURAN KEBARANGKALIAN

Smith dan Windeatt (2015) menjalankan kajian pengelasan berbilang kelas bagi pengecaman wajah. Model pengelasan berbilang wajah yang digunakan di dalam kajian ini adalah pembelajaran gabungan *Error-Correcting Output Code* (ECOC). Penentukuran kebarangkalian digunakan untuk mentafsir skor pengelasan kepada kebarangkalian. Pangkalan data *Cohn-Kanade* dijadikan sebagai sumber gambar-gambar wajah manusia. Sejumlah 456 gambar digunakan dan dikelaskan kepada 12 kelas. Setiap 640 x 480 piksel gambar ditukarkan kepada skala kelabu dan tetingkap gambar di bahagian mata diputarkan dan diskalakan kepada 150 x 75 piksel. Pemilihan fitur diaplikasikan sebelum pengelasan dibuat. Satu lapisan tersebunyi dalam algoritma MLP digunakan untuk melatih algoritma Levenberg-Marquardt. Sebelum penentukuran kebarangkalian diaplikasi, skor tentukur berada kurang daripada 0.5. Selepas penentukuran kebarangkalian diaplikasi, skor menjadi lebih daripada 0.5 dan

pengelasan menjadi lebih tepat. Sebagai konklusi, penentukuran kebarangkalian menambah baik kadar kesilapan dan tidak mempengaruhi ROC.

Kajian lain yang telah dijalankan oleh Baumann et al. (2015) bertujuan untuk meningkatkan prestasi algoritma hutan rawak yang asal dengan memperkenalkan nod kebarangkalian menggunakan penentukuran kebarangkalian dan membandingkan prestasi algoritma hutan rawak yang asal dengan algoritma yang telah ditentukur menggunakan tiga jenis set data. Metode yang dicadangkan menyatukan sepenuhnya penentukuran kebarangkalian di dalam setiap nod proses membuat keputusan di dalam algoritma hutan rawak di mana fungsi sigmoid dipetakan kepada ruang fitur nod yang sepadan. Set data yang digunakan adalah *German Traffic Sign Recognition Benchmark* (GTSRB), pengiktirafan digit tulisan tangan *Modified National Institute of Standards and Technology* (MNIST), *United States Postal Service* (USPS) dan *Letter*. Kesemua eksperimen diulang sebanyak lima kali. Bagi set data GTSRB nilai ketepatan bagi hutan rawak yang telah ditentukur dengan penentukuran kebarangkalian mencapai 89% manakala bagi algoritma yang asal, nilai ketepatan mencapai 88%. Bagi set data MNIST, kadar kesilapan yang paling rendah dicatatkan adalah 2.55% dan bagi set data USPS dan Letter, hutan rawak yang telah ditentukur dengan penentukuran kebarangkalian mencatatkan prestasi yang lebih baik daripada algoritma hutan rawak yang asal untuk bilangan pokok yang kecil dengan penambahbaikan sehingga 6%. Bagi kesemua nod, nilai kebarangkalian digandakan dengan kekerapan kelas relatif yang sepadan dengan nod. Ini menghasilkan taburan kebarangkalian akhir sekali gus memberikan nilai anggaran yang lebih baik dan tidak hanya bergantung kepada klasifikasi.

Williams dan Dagli (2017) menjalankan kajian untuk mengenal pasti label *identification* (ID) bahasa di Twitter bagi pengenalan bahasa automatik, membandingkan prestasi dua model pengelasan dengan menggunakan jenis-jenis set data Twitter yang berbeza dan mengaplikasikan penentukuran kebarangkalian untuk menentukur nilai-nilai output bagi model pengelasan *Margin Infused Relaxed Algorithm* (MIRA). Di dalam kajian ini, dua algoritma pengelasan iaitu MIRA dan *langid.py* digunakan dalam empat jenis eksperimen. Eksperimen pertama adalah menggunakan label-label Twitter *Application Programming Interface* (API) sebagai

asas kesahihan bagi klasifikasi bahasa. Bahasa-bahasa yang terlibat adalah Bahasa Melayu, Bahasa Inggeris, Bahasa Indonesia, Bahasa Portugis dan Bahasa Sepanyol. Eksperimen kedua adalah untuk menapis set data Twitter berdasarkan negara-negara di mana bahasa-bahasa dalam eksperimen satu merupakan bahasa yang paling prominen. Sebagai contoh, ciapan dalam Bahasa Melayu diwakili oleh negara Malaysia. Eksperimen ketiga adalah untuk mengklasifikasi sempadan geografi berdasarkan label-label Twitter. Eksperimen keempat adalah untuk mengesahkan bahasa sasaran bagi ciapan menggunakan *Amazon Mechanical Turk* (MTurk) *Human Intelligence Tasks* (HITs). Eksperimen ini menggunakan set data yang sama dari eksperimen ketiga. Bagi eksperimen ini, tiga orang pekerja yang mempunyai kelulusan MTurk lebih daripada 95% untuk melengkapkan HIT dan mengelaskan bahasa yang terdapat di dalam satu-satu ciapan. Pekerja-pekerja tersebut perlu memilih satu daripada tiga pernyataan yang berkait dengan ciapan iaitu teks itu merangkumi hanya satu bahasa X, teks itu merangkumi bahasa X dan sekurang-kurangnya satu bahasa X dan teks tersebut tidak merangkumi bahasa X. Label yang disahkan kemudian dikelaskan menggunakan kedua-dua model pengelasan. Model pengelasan MIRA ditentukur menggunakan penentukuran kebarangkalian kerana output model pengelasan MIRA hampir serupa dengan output dari mesin sokongan vektor. penentukuran kebarangkalian digunakan untuk mentafsir keputusan dari tiga eksperimen iaitu ekperimen kedua, ketiga dan keempat. Berdasarkan hasil akhir, model pengelasan MIRA menunjukkan prestasi yang lebih baik berbanding langid.py bagi kesemua eksperimen. Bagi setiap set data, penentukuran kebarangkalian lebih cenderung untuk mempengaruhi sifat tentukur untuk ciapan dalam Bahasa Indonesia berbanding ciapan dalam Bahasa Melayu. Output kebarangkalian yang dihasilkan oleh penentukuran kebarangkalian adalah benar dan metode ini tidak mengganggu kesahihan model pengelasan MIRA yang asal.

Dalam kajian yang dilakukan oleh Walsh et al. (2017) adalah untuk mengenal pasti kaedah penentukuran yang memberikan prestasi yang boleh diterima menggunakan data pengesahan yang paling sedikit kuantitinya dan mengaplikasikan penggunaan klinikal untuk menilai kos andaian bagi sebarang kesilapan tentukur. Penggunaan klinikal merujuk kepada utiliti-utiliti, kos dan kerosakan dalam penggunaan model ramalan dalam latihan. Model ramalan yang digunakan dalam kajian ini adalah LASSO untuk menerbitkan satu kohort pemerhatian berdasarkan data

kesihatan elektronik retrospektif pesakit-pesakit luar di *Columbia University Medical Center* bermula tahun 2005 hingga 2009. Ramalan risiko kemasukan semula dibangunkan, disahkan dan kemudian ditentukur meliputi data dari tahun-tahun subsekuen. Selepas itu, penggunaan klinikal diaplikasikan terhadap model paduan yang belum dan telah ditentukur dengan memberi perhatian kepada kos dan utiliti. Data yang digunakan sebagai data latihan adalah data kemasukan dan kemasukan semula pesakit dari tahun 2005 hingga 2008. Data yang digunakan sebagai data pengujian adalah data kemasukan dan kemasukan semula pada tahun 2009. Tiada data pengujian digunakan untuk model yang telah ditentukur. Data melalui fasa pra-pemprosesan menggunakan Python. Analisis-analisis statistik dijalankan menggunakan perisian R. Beberapa kaedah tentukur digunakan iaitu kaedah Platt, *logistic calibration* dan *prevalence adjustment*. Prestasi model ramalan kemudian dibandingkan menggunakan ROC. Bagi mengukur prestasi kaedah tentukur, *statistic Z Spiegelhalter*, RMSE bagi ramalan *binned*, skor Brier, kecerunan dan pintasan tentukur digunakan. Berdasarkan keputusan akhir, nilai ROC berada di antara 0.7 hingga 0.86. Prestasi *logistic calibration* dan kaedah Platt adalah lebih baik daripada prestasi *prevalence adjustment*.

Dalam kajian yang telah dijalankan oleh Calvi et al. (2019), sebuah formulasi model *Support Tensor Machine* (STM) menggunakan *least squares* (LS) dibangunkan untuk meramal kewangan dinamakan sebagai *Least Square Support Tensor Machine* (LS-STM). Model ini digunakan untuk meramal gerakan harian harga bagi indeks kewangan S&P 500. Output bagi model LS-STM diterjemahkan oleh penentukuran kebarangkalian untuk menilai kebarangkalian output tersebut. Data bagi kajian ini merangkumi tempoh masa dari Januari 2006 sehingga Januari 2017 dari *Chicago Board Options Exchange's* (CBOE) *Volatility Index* (VIX). Set data kemudian dibahagi kepada tensor urutan ketiga dan analisis dijalankan dengan 250 hari *sliding windows*. Prestasi LS-STM dibandingkan dengan mesin sokongan vektor menggunakan nilai parameter C = {0.01, 0.1, 1, 10, 30, 50, 100}. Nilai ketepatan ditafsirkan sebagai peratusan kejayaan LS-STM meramal gerakan harga S&P 500 berdasarkan data sejarah. Output bagi LS-STM bertumpu secara konsisten kepada nilai sebenar disebabkan oleh penentukuran kebarangkalian. Berdasarkan keputusan tersebut, LS-STM menunjukkan prestasi yang lebih baik berbanding mesin sokongan vektor. Interpretasi kebarangkalian

keputusan melalui penentukuran kebarangkalian menunjukkan kestabilan terhadap perubahan-perubahan parameter.

Kajian oleh Sanderson et al. (2020) dijalankan untuk membandingkan prestasi model regresi logistik dengan *XGBoost* bagi mengkuantitikan risiko kematian bunuh diri dalam 90 hari bagi lawatan *parasuicide* di jabatan kecemasan (ED). *Parasuicide* merujuk kepada lawatan ED bagi pencederaan diri sendiri yang tidak menyebabkan kematian. Data diambil dari lima pusat kesihatan di Alberta, Canada mengenai pesakit-pesakit dengan lawatan ED bagi *parasuicide* dari tahun 2010 sehingga 2017. Terdapat sejumlah 268 orang mati akibat bunuh diri dalam masa 90 hari dan 33, 436 orang yang masih hidup. Bagi mengatasi masalah ketidakseimbangan data, model pengelasan merangkumi berat kelas sebanyak 124/12 bagi pesakit yang mati akibat bunuh diri dan 1/125 bagi pesakit yang masih hidup. Data latihan dan data pengujian bagi kedua-dua model pengelasan adalah menggunakan *10-fold cross validation.* Parameter yang digunakan bagi menilai prestasi model pengelasan adalah AUC kerana AUC ialah parameter yang berkait rapat dengan sensitiviti, kespesifikan, nilai ramalan positif (PPV) dan nilai ramalan negatif (NPV). Penentukuran model pengelasan dibandingkan menggunakan kebarangkalian yang diramalkan dengan kebarangkalian yang sebenar. Penentukuran kebarangkalian digunakan untuk menentukur model pembelajaran mesin. Sebagai alternatif, model ramalan *XGBoost* yang kedua dibangunkan untuk meramal hasil kebarangkalian sebenar menggunakan hasil kebarangkalian yang diramal. Berdasarkan keputusan akhir, diskriminasi yang ditunjukkan oleh model *XGBoost* adalah lebih baik berbanding model regresi logistik. Model *XGBoost* alternatif juga adalah lebih baik berbanding model yang ditentukur menggunakan kaedah Platt.

Tambahan lagi, Trottini et al. (2020) menjalankan kajian tentang kepentingan menentukur model ramalan dalam mentaksir risiko karies gigi. Kajian ini menyediakan alatan dan garis panduan bagi pentaksiran wajar ke atas pentukuran model *Probabilistic Caries Risk Assessment* (P-CRA). Data yang digunakan merupakan kajian susulan yang terdiri daripada 957 pelajar sekolah berumur antara 7 hingga 9 tahun yang tinggal di Sassari, sebuah daerah yang terletak di Sardinia, Itali. Data dianalisis menggunakan perisian R. Plot tentukur licinan dihasilkan menggunakan kebarangkalian ramalan karies gigi yang baru melawan kebarangkalian pemerhatian licinan. Kaedah tentukur

yang digunakan adalah kaedah Platt dan regresi isotonik. Lima parameter bagi mengukur prestasi tentukur diadaptasikan iaitu *calibration-in-the-large* (α), kecerunan tentukur (β), indeks tentukur (ICI) bersama sukatan pendamping E50 dan E90. Penutukuran sempurna merujuk kepada α = 0, β = 1, ICI = E50 = E90 = 0. ICI bagi model yang tidak ditentukur adalah 0.1059, model yang ditentukur menggunakan kaedah Platt adalah 0.0210 dan model yang dientukur menggunakan regresi isotonik adalah 0.0157. Berdasarkan keputusan akhir, kedua-dua kaedah Platt dan regresi isotonik menambah baik penentukuran model secara signifikan.

Tsakanikas et al. (2020) mengusulkan aliran kerja pembelajaran mesin bagi klasifikasi spektroskopi bahan makanan mentah bagi industri masa hadapan. Pembangunan aliran kerja bagi pengenalan bahan makanan mentah dalam makalah ini menggunakan spektrum *Fourier-transform infrared spectroscopy* (FT-IR) yang diimplementasi di dalam *Python*. Data mentah sensor akan melalui fasa pra-pemprosesan dan normalisasi. Kemudian, pengurangan dimensi berselia dilakukan berdasarkan *Partial Least Square Regression* (PLSR). Dalam langkah akhir, model pengelasan dibina berdasarkan tujuh jenis bahan makanan mentah iaitu lembu, khinzir, ikan, nanas, sayur arugula, ayam dan bayam. Model yang dibina adalah mesin sokongan vektor. Bagi klasifikasi binari, penentukuran kebarangkalian digunakan sebagai kaedah tentukur bagi mengelakkan kesilapan dalam klasifikasi dan untuk interprestasi keputusan akhir. Antara parameter yang digunakan bagi mengukur prestasi model mesin sokongan vektor adalah ketepatan, ukuran F, sensitiviti, kejituan dan kespesifikan. Keputusan akhir menunjukkan kesemua nilai parameter mencapai nilai 100%.

Feng (2021) menjalankan kajian mengusulkan kaedah untuk menentukur kebarangkalian bagi menambah baik analisis ketidakpastian bagi klasifikasi litofasies. Analisis ketidakpastian adalah satu indikator penting dalam memberikan keyakinan dalam ramalan. Kaedah pembelajaran mesin yang digunakan dalam kajian ini adalah hutan rawak. Kaedah tentukur yang diaplikasi dalam kajian ini adalah penentukuran kebarangkalian kerana set data yang kecil. Set data pertama bagi menentukur ketidakpastian dalam klasifikasi litofasies adalah data pengelogan yang dikumpul dari Hugoton dan Panama Fields, Amerika Utara. Data pengelogan ini mengandungi lima

atribut iaitu sinar gamma, kerintangan, perbezaan keporosan ketumpatan neutron, purata keporosan ketumpatan neutron dan kesan fotoelektrik. Sembilan litofasies digabungkan kepada tiga kelas iaitu Kelas 1, Kelas 2 dan Kelas 3 bergantung kepada respons pengelogan, komposisi mineral dan persekitaran pengendapan. Enam litofasies digunakan sebagai data latihan, satu litofasies digunakan sebagai pengesahihan data dan satu sebagai data pengujian. Data latihan digunakan dalam model hutan rawak, pengesahihan data digunakan untuk menentukur kebarangkalian menggunakan penentukuran kebarangkalian. Selepas latihan dan pengesahihan, data pengujian digunakan untuk mengukur prestasi model pengelasan dan kaedah tentukur. Bagi Kelas 1 dan Kelas 3, model yang tidak dan telah ditentukur menunjukkan prestasi yang serupa. Penambahbaikan bagi model yang telah ditentukur ditunjukkan melalui nilai pekali korelasi Matthews (MCC) yang lebih besar iaitu 0.6156 berbanding 0.6057 bagi model yang tidak ditentukur. Set data kedua yang diaplikasi dikumpul dari Volve Field, yang terletak di bahagian Laut Utara. Terdapat enam fitur pengelogan iaitu angkup, keporosan neutron, kerintangan dalam, kerintangan sederhana, faktor fotoelektrik dan ketumpatan. Kesemua data dibahagikan kepada 60% data latihan, 20% data pengesahihan dan 20% data pengujian. Skor Brier menunjukkan nilai yang lebih kecil bagi model yang telah ditentukur iaitu 0.2681 berbanding 0.2717 bagi model yang tidak ditentukur. Kesimpulannya, penentukuran kebarangkalian membantu mengelakkan ramalan yang terlalu yakin.

Kajian yang dilakukan oleh Fan et al. (2021) mengaplikasi penentukuran kebarangkalian dalam pelbagai algoritma pembelajaran mesin. Kajian ini bertujuan untuk membandingkan algoritma pembelajaran mesin yang asal dengan algoritma yang telah ditentukur. Selain itu, kajian ini juga bertujuan untuk membangunkan model ramalan untuk meramal bahaya kematian bagi pesakit *diffuse large B-cell lymphoma* (DLBCL) yang menerima rawatan dalam dua tahun. Set data yang digunakan dalam kajian ini adalah dari Shanxi Cancer Hospital, China. Sebanyak 406 pesakit yang di diagnos dengan DLBCL antara bulan April 2010 sehingga bulan Mei 2017 bersama 17 jenis atribut telah di analisis. Bagi memilih atribut ramalan, tiga kaedah digunakan iaitu model *Cox*, model logistik dan algoritma hutan rawak. Bahagian pertama dalam kajian ini adalah untuk membina model ramalan asas menggunakan algoritma-algoritma regresi logistik, mesin sokongan vektor, Bayes naif, hutan rawak dan rangkaian neural

suap maju (FNN). Bahagian kedua adalah untuk mengaplikasi penentukuran kebarangkalian terhadap model asas tersebut. Kaedah penentukuran yang digunakan adalah *shape-restricted polynomial regression* (RPR), kaedah Platt dan regresi isotonik. Bahagian ketiga bagi kajian ini adalah penggabungan model asas kepada pembelajaran gabungan. Terdapat tiga kaedah pembelajaran gabungan yang dijalankan iaitu *simple averaging, weighted averaging* dan *stacking*. Kelima-lima model pengelasan yang tidak ditentukur digabungkan, kemudian model tersebut ditentukur dan digabungkan untuk pembelajaran gabungan. Untuk menyempurnakan pembinaan model dan proses penilaian, ujian *hold-out* dan 3-*fold cross validation* digabungkan. Eksperimen tersebut diulang sebanyak 300 kali. Berdasarkan keputusan untuk bahagian pertama kajian ini, kelima-lima model mencatatkan nilai AUC melebihi 0.75. Selepas proses penentukuran, ralat bagi model Bayes naif, hutan rawak dan mesin sokongan vektor menurun secara signifikan. Kaedah penentukuran menggunakan RPR mencapai prestasi terbaik dengan model mesin sokongan vektor berbanding kaedah Platt dan regresi isotonik. Walaupun diskriminasi bagi kelima-lima model adalah serupa, perbezaan dalam penentukuran adalah ketara. Diskriminasi merujuk kepada keupayaan model untuk mendiagnos penyakit dengan tepat. Dalam kajian ini, kaedah penentukuran RPR mencatatkan prestasi yang lebih baik berbanding kaedah Platt dan regresi isotonik.

2.5 MODEL DISKRIMINATIF DAN GENERATIF

Model diskriminatif merujuk kepada model pembelajaran yang digunakan dalam pengelasan statistik (Goyal 2021). Objektif model diskriminatif ialah untuk membuat pengelasan dengan menemukan batasan untuk memisahkan kelas. Model diskriminatif tidak berupaya untuk menjana data yang baru akan tetapi, ianya lebih teguh terhadap *outliers.* Contoh bagi model diskriminatif ialah regresi logistik, mesin sokongan vektor dan hutan rawak.

Model generatif merujuk kepada model yang mempelajari taburan kebarangkalian bagi setiap kelas dalam satu set data (Goyal 2021). Model generatif menggunakan konsep kebarangkalian bersama dan mencipta contoh baharu di mana input dan output wujud pada masa yang sama. Model generatif lebih sensitif terhadap *outliers*. Contoh bagi model pembelajaran generatif ialah Bayes naif dan MLP.

Berdasarkan kajian yang dilakukan oleh Fan et al. (2021), bagi mencapai prestasi yang memuaskan, model pengelasan asas yang digunakan dalam model pembelajaran gabungan seharusnya memiliki ketepatan dan kepelbagaian yang baik. Terdapat tiga cara untuk mencapai kepelbagaian yang baik iaitu kepelbagaian set data, kepelbagaian parameter dan kepelbagaian struktur. Kaedah kepelbagaian set data menggunakan gandaan set data yang dijana daripada set data yang asal untuk melatih pelbagai model pengelasan asas. Gandaan set data yang berbeza membolehkan model pengelasan asas menjana output yang berbeza. Kaedah kepelbagaian parameter melatih model pengelasan asas yang berbeza menggunakan parameter yang berbeza. Kaedah kepelbagaian struktur menggunakan algoritma berbeza untuk menjana model pengelasan asas yang berbeza dan model pembelajaran gabungan yang terhasil dikenali sebagai pembelajaran gabungan heterogen.

Dalam kajian ini, lima algoritma berbeza digunakan bagi mencapai pembelajaran gabungan heterogen antaranya regresi logistik, Bayes naif, mesin sokongan vektor, hutan rawak dan MLP. Fan et al. (2021) dan Niculescu-Mizil dan Caruana (2005) membuktikan bahawa model seperti Bayes naif, hutan rawak dan mesin sokongan vektor tidak boleh menjana anggaran kebarangkalian yang tepat walaupun memiliki prestasi pengelasan yang baik. Oleh itu, kajian ini mencadangkan struktur model yang berbeza untuk mencapai ketepatan dan kepelbagaian yang tinggi. Sebelum mengaplikasi pembelajaran gabungan, penentukuran kebarangkalian diaplikasi pada model yang tidak ditentukur dengan baik untuk meningkatkan prestasi model pembelajaran gabungan.

2.6 KESIMPULAN

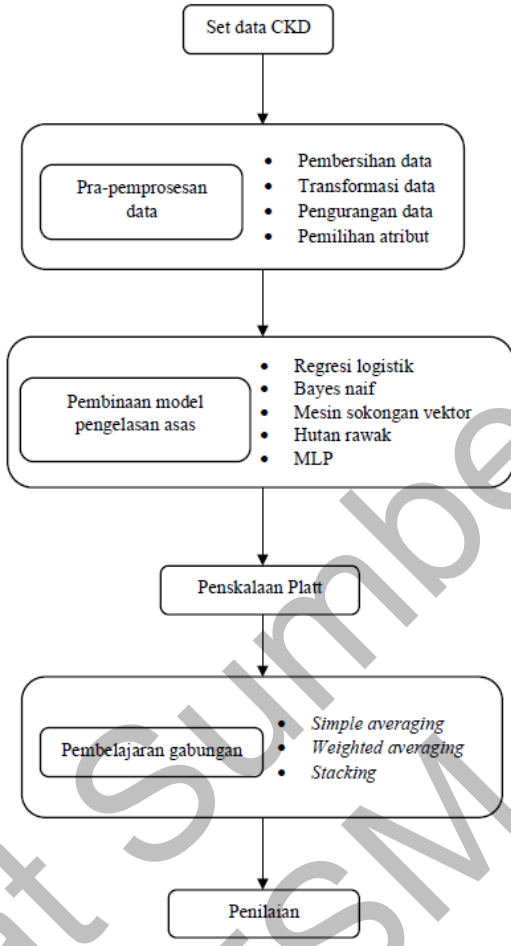
Keseluruhan bab ini menceritakan tentang kajian lepas mengenai kesemua topik yang berkaitan dengan kajian ini iaitu pembelajaran aplikasi penentukuran kebarangkalian kepada pembelajaran gabungan bagi set data CKD. Pertama sekali, subtopik kajian lampau berkaitan data diagnosis CKD menggunakan teknik pembelajaran mesin seperti pokok keputusan, Bayes naif dan regresi logistik. Kemudian, subtopik seterusnya mengenai penentukuran kebarangkalian dan evolusi algoritma tersebut. Subtopik akhir menceritakan tentang kajian lampau yang menggunakan penentukuran kebarangkalian dalam pembelajaran mesin.

BAB III

KAEDAH KAJIAN

3.1 PENDAHULUAN

Bab ini akan membincangkan mengenai kaedah kajian dan juga rangka kerja yang terlibat dalam setiap proses yang berlaku dalam kajian ini. Oleh hal yang demikian, rangka kerja untuk kajian ini telah dibahagikan kepada empat peringkat. Peringkat pertama adalah pemahaman set data yang digunakan dalam kajian ini iaitu set data CKD. Peringkat kedua merupakan fasa pra-pemprosesan data. Peringkat ini menceritakan tentang bagaimana proses pembersihan dan transformasi data serta pemilihan fitur dilakukan sebelum fasa pembinaan model. Peringkat ketiga ialah mengenai pembinaan model pengelasan bagi meramal CKD menggunakan kaedah pembelajaran mesin. Terdapat tiga tahap dalam pembinaan model pengelasan. Tahap pertama adalah menggunakan lima jenis model pengelasan untuk meramal CKD. Tahap kedua adalah kaedah penentukuran kelima-lima model pengelasan menggunakan penentukuran kebarangkalian. Tahap ketiga ialah mengaplikasi model pembelajaran gabungan dengan menggunakan model pengelasan yang tidak dan telah ditentukur. Peringkat keempat dan terakhir ialah penilaian model pengelasan, kaedah penentukuran dan model pembelajaran gabungan yang digunakan dalam kajian ini. Beberapa parameter digunakan bagi mengukur prestasi model pengelasan dalam kajian ini. Langkah-langkah bagi membina model pembelajaran gabungan dengan penentukuran kebarangkalian dalam kajian ini ditunjukkan dalam Rajah 3.1.



Rajah 3.1 Metodologi kajian

3.2 SEKITARAN PERISIAN

Python digunakan sebagai bahasa pengaturcaraan sepanjang kajian ini dijalankan. *Python* merupakan bahasa pengaturcaraan tahap tinggi yang mudah dan efektif untuk digunakan. *Python* dicipta oleh Guido van Rossum pada tahun 1991. *Jupyter Notebook* digunakan sebagai ekosistem bagi pembelajaran mesin. Dalam kajian ini, beberapa perpustakaan *Python* digunakan untuk menjalankan tugas-tugas tertentu Jadual 3.1 menunjukkan perpustakaan *Python* yang digunakan berserta fungsi.

Jadual 3.1 Perpustakaan *Python* dan fungsi

Perpustakaan *Python*	**Fungsi**
Pandas	Digunakan untuk analisis dan pengolahan data dalam fasa pra-pemprosesan data
Numpy	Mewakili perkataan *Numerical Python* dan mengandungi objek tatasusunan berbilang matra. Operasi matematik dilakukan menggunakan perpustakaan ini.

Matplotlib	Digunakan untuk perwakilan dan visualisasi data di mana rajah dibentuk dan diadaptasi dalam bentuk 2 dimensi.
Scikit-learn	Perpustakaan sumber terbuka yang digunakan bersama Numpy dan Matplotlib. Mengandungi algoritma pembelajaran mesin seperti pengelasan dan *clustering.*
Seaborn	Digunakan sebagai visualisasi data yang dibina berdasarkan Matplotlib.

3.3 PEMAHAMAN DATA

Fasa pemahaman data bermula dari pengumpulan data untuk perlombongan dan mengenal pasti masalah kualiti data. Akses kepada data yang berkualiti akan membantu untuk mengenal pasti sesuatu masalah dari awal dan mengorak langkah untuk menyelesaikan masalah tersebut. Data yang berkualiti juga akan membekalkan justifikasi dan bukti bagi membantu membuat sesuatu keputusan. Pengumpulan data yang sistematik akan membantu menjimatkan masa untuk mengakses data.

Set data yang digunakan untuk pengelasan CKD dalam kajian ini diperoleh dari *Kaggle dataset*, https://www.kaggle.com/mansoordaku/ckdisease. Set data ini mengandungi 26 atribut dan 400 orang pesakit buah pinggang. Set data mengandungi informasi tentang bacaan hemoglobin, sel darah putih dan simptom-simptom penyakit buah pinggang seperti hipertensi dan diabetes mellitus. Pesakit dalam data ini dilabel sama ada di diagnos dengan CKD atau tidak. Data terdiri daripada data angka dan nominal. Jadual 3.2 menunjukkan senarai atribut, jenis dan penerangan bagi atribut berkenaan manakala Rajah 3.2 menunjukkan sebahagian data yang digunakan dalam kajian ini.

Jadual 3.2 Jenis-jenis dan penerangan atribut bagi set data

Bil.	Atribut	Jenis	Penerangan
1.	id	Angka	Id pesakit dalam angka 0 hingga 399
2.	age	Angka	Umur pesakit (tahun)
3.	bp	Angka	Bacaan tekanan darah (mm/HG)
4.	sg	Angka	Bacaan graviti spesifik
5.	al	Angka	Bacaan albumin
6.	su	Angka	Bacaan gula dalam darah
7.	rbc	Nominal	Bacaan sel darah merah (normal atau abnnormal)
8.	pc	Nominal	Bacaan sel nanah (normal atau abnormal)
9.	pcc	Nominal	Gumpalan sel nanah (present atau not present)
10.	ba	Nominal	Bakteria (present atau not present)
11.	bgr	Angka	Bacaan *blood glucose random* (mg/dl)
12.	bu	Angka	Bacaan urea darah (mg/dl)
13.	sc	Angka	Bacaan serum creatinine (mg/dl)
14.	sod	Angka	Bacaan natrium (mEq/L)
15.	pot	Angka	Bacaan kalium (mEq/L)

16.	hemo	Angka	Bacaan hemoglobin (gm)
17.	pcv	Angka	Bacaan packed cell volume
18.	wc	Angka	Kiraan sel darah putih per microliter
19.	rc	Angka	Kiraan sel darah merah per microliter
20.	htn	Nominal	Hipertensi (yes atau no)
21.	dm	Nominal	Diabetes mellitus (yes atau no)
22.	cad	Nominal	Penyakit koronari arteri (yes atau no)
23.	appet	Nominal	Selera pesakit (good atau poor)
24.	pe	Nominal	Pedal edema (yes atau no)
25.	ane	Nominal	Anemia (yes atau no)
26.	classification	Nominal	Atribut sasaran (ckd atau not ckd)

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no	no	no	good	no	no	ckd

5 rows × 26 columns

Rajah 3.2 Petikan set data yang digunakan dalam kajian

3.4 PRA-PEMPROSESAN DATA

Fasa pra-pemprosesan data amat penting untuk meningkatkan kualiti data sekali gus meningkatkan kualiti pengetahuan yang boleh diekstrak daripada data. Fasa pra-pemprosesan data melibatkan beberapa langkah iaitu proses pembersihan data, integrasi data, transformasi data dan pengurangan data sebelum teknik perlombongan data dapat diaplikasikan. Fasa ini dilaksanakan sehingga mencapai output yang dikehendaki.

Dalam kajian ini, nama-nama atribut seperti dalam Jadual 3.2 di atas adalah dalam singkatan. Proses menggantikan singkatan tersebut dengan istilah yang betul bagi setiap atribut dilaksanakan. Jadual 3.3 menunjukkan penggantian tersebut manakala Rajah 3.3 menunjukkan petikan penggantian istilah dalam *Jupyter Notebook.*

Jadual 3.3 Penggantian istilah bagi atribut dalam set data

Bil.	Atribut	Penggantian Istilah
1.	bp	blood_pressure
2.	sg	specific_gravity
3.	al	albumin
4.	su	sugar
5.	rbc	red_blood_cell
6.	pc	pus_cell
7.	pcc	pus_cell_clumps

8.	ba	bacteria
9.	bgr	blood_glucose_random
10.	bu	blood_urea
11.	sc	serum_creatinine
12.	sod	sodium
13.	pot	potassium
14.	hemo	hemoglobin
15.	pcv	packed_cell_volume
16.	wc	white_blood_cell_count
17.	rc	red_blood_cell_count
18.	htn	hypertension
19.	dm	diabetes_mellitus
20.	cad	coronary_artery_disease
21.	appet	appetite
22.	pe	pedal_edema
23.	ane	anemia

```
col = {'id':'id',
       'age':'age',
       'bp':'blood_pressure',
       'sg':'specific_gravity',
       'al':'albumin',
       'su':'sugar',
       'rbc':'red_blood_cell',
       'pc':'pus_cell',
       'pcc':'pus_cell_clumps',
       'ba':'bacteria',
       'bgr':'blood_glucose_random',
       'bu':'blood_urea',
       'sc':'serum_creatinine',
       'sod':'sodium',
       'pot':'potassium',
       'hemo':'hemoglobin',
       'pcv':'packed_cell_volume',
       'wc':'white_blood_cell_count',
       'rc':'red_blood_cell_count',
       'htn':'hypertension',
       'dm':'diabetes_mellitus',
       'cad':'coronary_artery_disease',
       'appet':'appetite',
       'pe':'pedal_edema',
       'ane':'anemia',
       'classification':'classification'}

data.rename(columns=col, inplace=True)
```

Rajah 3.3 Petikan penggantian istilah di dalam *Jupyter Notebook*

3.4.1 Pembersihan Data

Data yang dikumpul lazimnya tidak terlepas daripada pelbagai kesilapan dan ralat. Sebagai contoh, data tidak konsisten, kehilangan data, pertindihan rekod dan kehadiran *outliers*. Proses pembersihan data membantu untuk mengesan dan membetulkan kesilapan dan ralat yang ada pada data.

Dalam kajian ini, terdapat beberapa data yang tidak konsisten. Sebagai contoh, bagi perkataan *yes, no* dan *ckd*, terdapat kesilapan ejaan seperti *\tyes, \tno* dan *ckd\t*. Rajah 3.4 dan Rajah 3.5 menunjukkan ralat bagi perkataan tersebut dalam beberapa atribut. Rajah 3.6 menunjukkan pembetulan bagi ralat ini dalam *Jupyter Notebook*.

```
hypertension has unique values ['yes' 'no' nan]
diabetes_mellitus has unique values ['yes' 'no' ' yes' '\tno' '\tyes' nan]
coronary_artery_disease has unique values ['no' 'yes' '\tno' nan]
appetite has unique values ['good' 'poor' nan]
pedal_edema has unique values ['no' 'yes' nan]
anemia has unique values ['no' 'yes' nan]
classification has unique values ['ckd' 'ckd\t' 'notckd']
```

Rajah 3.4 Ralat perkataan yang terdapat dalam set data

```
hypertension has unique values ['yes' 'no' nan]
diabetes_mellitus has unique values ['yes' 'no' nan]
coronary_artery_disease has unique values ['no' 'yes' nan]
appetite has unique values ['good' 'poor' nan]
pedal_edema has unique values ['no' 'yes' nan]
anemia has unique values ['no' 'yes' nan]
classification has unique values ['ckd' 'notckd']
```

Rajah 3.5 Pembetulan bagi ralat perkataan dalam set data

```
#Replace incorrect values
data['diabetes_mellitus']=data['diabetes_mellitus'].replace(to_replace={' yes':'yes','\tyes':'yes','\tno':'no'})
data['coronary_artery_disease']=data['coronary_artery_disease'].replace(to_replace={'\tno':'no'})
data['packed_cell_volume']=data['packed_cell_volume'].replace(to_replace={'\t?':'','\t43':'43'})
data['white_blood_cell_count']=data['white_blood_cell_count'].replace(to_replace={'\t6200':'6200','\t8400':'8400','\t?':''})
data['red_blood_cell_count']=data['red_blood_cell_count'].replace(to_replace={'\t?':''})
data['classification']=data['classification'].replace(to_replace={'ckd\t':'ckd'})
```

Rajah 3.6 Pembetulan bagi ralat perkataan di dalam *Jupyter Notebook*

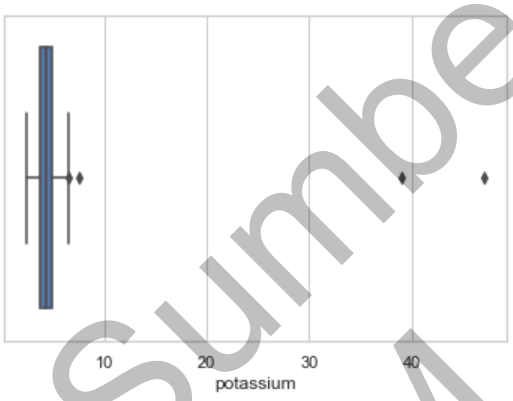
Set data ini tidak mengandungi sebarang pertindihan rekod. Rajah 3.7 menunjukkan cebisan kod bagi mengenal pasti sebarang pertindihan rekod.

```
In [83]: print('Are there any duplicated rows?')
         any(data.duplicated())

         Are there any duplicated rows?

Out[83]: False
```

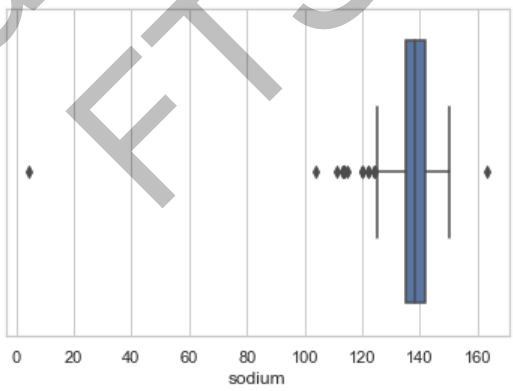
Rajah 3.7 Pertindihan rekod dalam set data

Selepas mengenal pasti ketiadaan pertindihan rekod, proses mengenal pasti *outliers* dilakukan. *Outliers* merupakan nilai ekstrem yang sangat berbeza dan menunjukkan penyimpangan yang melampau dari kebanyakan nilai dalam set data. Proses mengendalikan *outliers* amat penting kerana *outliers* lazimnya ralat dan boleh mempengaruhi hasil akhir dalam satu-satu analisis (Deneshkumar et al. 2014). Dalam data perubatan, nilai yang ekstrem tidak boleh disamakan dengan *outliers*. Ini kerana,

nilai yang ekstrem ini adalah nilai yang sahih. Dalam kajian ini, *outliers* terdapat di dalam atribut *sodium* dan *potassium*. Nilai *sodium* yang sepatutnya berada dalam julat 135 mEq/L hingga 145 mEq/L (Almasoud & Ward 2019) manakala nilai *potassium* yang paling tinggi adalah 7.6 mEq/L (Gheno et al. 2003). Nilai yang tidak berada dalam julat tersebut dan lebih daripada 7.6 mEq/L dicatatkan sebagai data yang hilang. Rajah 3.8 dan Rajah 3.9 menunjukkan *boxplot* bagi atribut *sodium* dan *potassium*.



Rajah 3.8 *Boxplot* bagi atribut *potassium*



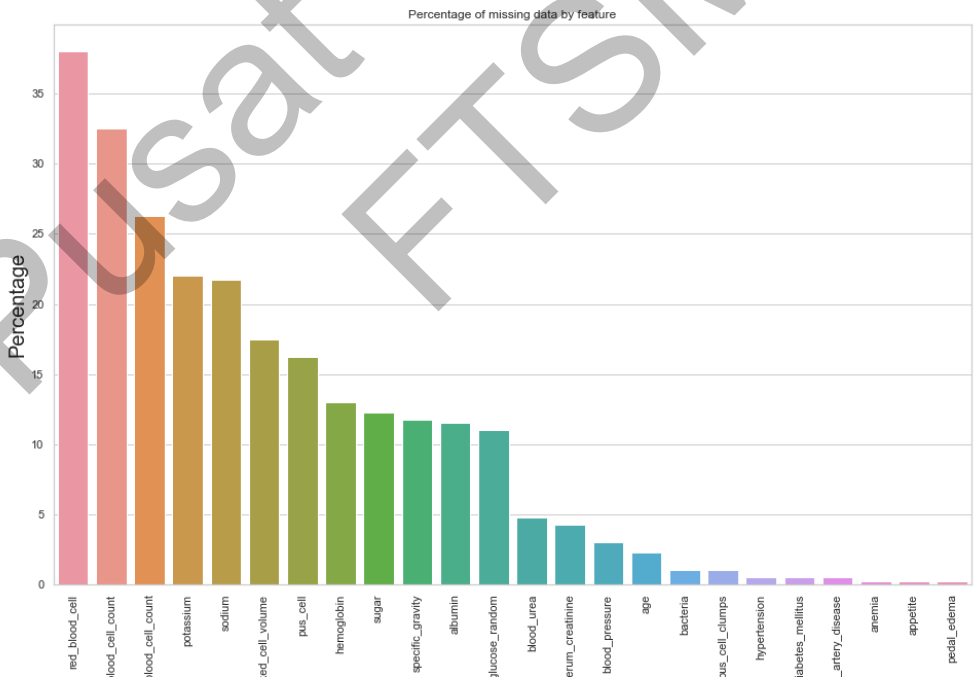
Rajah 3.9 *Boxplot* bagi atribut *sodium*

Selepas itu, proses menggantikan data yang hilang dilaksanakan. Berdasarkan set data CKD, 24 daripada 26 atribut mengalami masalah kehilangan data. Kehilangan data paling banyak direkodkan oleh atribut *red_blood_cell* dengan 152 data yang hilang. Mengikut nilai peratusan, kehilangan data bagi atribut *red_blood_cell* meliputi 38% daripada keseluruhan set data. Rajah 3.10 menunjukkan jumlah data yang hilang

dalam setiap atribut dan Rajah 3.11 menunjukkan peratusan data yang hilang dalam setiap atribut.

```
In [85]: data.isnull().sum()

Out[85]: id                          0
         age                         9
         blood_pressure             12
         specific_gravity           47
         albumin                    46
         sugar                      49
         red_blood_cell            152
         pus_cell                   65
         pus_cell_clumps             4
         bacteria                    4
         blood_glucose_random       44
         blood_urea                 19
         serum_creatinine           17
         sodium                     87
         potassium                  88
         hemoglobin                 52
         packed_cell_volume         70
         white_blood_cell_count    105
         red_blood_cell_count      130
         hypertension                2
         diabetes_mellitus           2
         coronary_artery_disease     2
         appetite                    1
         pedal_edema                 1
         anemia                      1
         classification              0
         dtype: int64
```

Rajah 3.10 Jumlah data yang hilang bagi setiap atribut



Rajah 3.11 Peratusan data yang hilang bagi setiap atribut

Disebabkan peratusan data yang hilang adalah tinggi bagi kebanyakan atribut, data tersebut tidak boleh dibuang begitu sahaja kerana akan menjejaskan kualiti set data ini dan akan mempengaruhi model pengelasan yang diaplikasikan. Oleh sebab yang

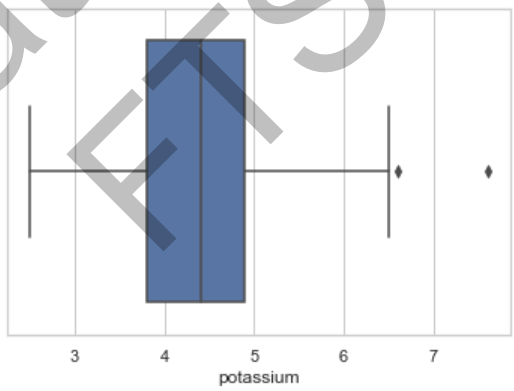
demikian, data yang hilang diganti dengan nilai-nilai tertentu. Bagi atribut angka, data yang hilang diganti dengan nilai median kerana set data yang kecil. Bagi atribut nominal, data yang hilang diganti dengan mod bagi atribut tersebut. Pelaksanaan penggantian nilai data-data yang hilang bagi atribut angka ditunjukkan dalam Rajah 3.12 manakala penggantian data-data yang hilang bagi atribut nominal ditunjukkan dalam Rajah 3.13. Rajah 3.14 menunjukkan *boxplot* bagi atribut potassium selepas penggantian data manakala Rajah 3.15 menunjukkan *boxplot* bagi atribut sodium selepas penggantian data.

```
Imputing for age with 55.0
Imputing for blood_pressure with 80.0
Imputing for specific_gravity with 1.02
Imputing for albumin with 0.0
Imputing for sugar with 0.0
Imputing for blood_glucose_random with 121.0
Imputing for blood_urea with 42.0
Imputing for serum_creatinine with 1.3
Imputing for sodium with 138.0
Imputing for potassium with 4.4
Imputing for hemoglobin with 12.649999999999999
Imputing for packed_cell_volume with 40.0
Imputing for white_blood_cell_count with 8000.0
Imputing for red_blood_cell_count with 4.8
```

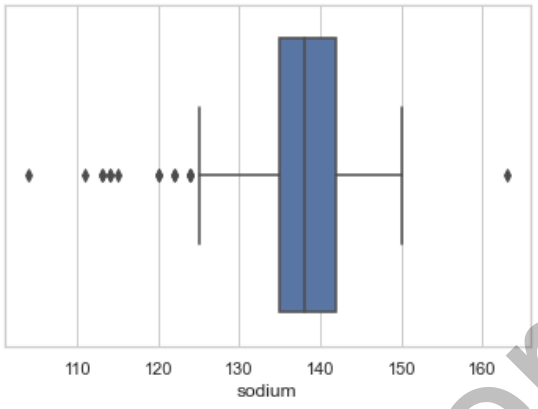
Rajah 3.12 Penggantian nilai kehilangan data bagi atribut angka

```
Imputing for red_blood_cell with normal
Imputing for pus_cell with normal
Imputing for pus_cell_clumps with notpresent
Imputing for bacteria with notpresent
Imputing for hypertension with no
Imputing for diabetes_mellitus with no
Imputing for coronary_artery_disease with no
Imputing for appetite with good
Imputing for pedal_edema with no
Imputing for anemia with no
```

Rajah 3.13 Penggantian nilai kehilangan data bagi atribut nominal



Rajah 3.14 *Boxplot* bagi atribut *potassium* selepas penggantian data



Rajah 3.15 *Boxplot* bagi atribut *sodium* selepas penggantian data

3.4.2 Transformasi Data

Selepas fasa pembersihan data, fasa transformasi data dijalankan. Dalam fasa ini, data ditransformasi agar lebih mudah diurus bagi tujuan pemodelan dan untuk meningkatkan kecekapan mesin bagi membina model yang terbaik. Pola yang ditemui juga menjadi lebih mudah untuk difahami (Han et al. 2012).

Dalam kajian ini, semua atribut nominal diubah menjadi angka. Sebagai contoh, label normal dan abnormal yang mewakili atribut *red_blood_cell* masing-masing diubah kepada nilai 1 dan 0. Proses ini diulang bagi *atribut pus_cell, pus_cell_clumps, bacteria, hypertension, diabetes_mellitus, coronary_artery_disease, appetite, pedal_edema, anemia* dan *classification*. Jadual 3.4 dan Rajah 3.16 menunjukkan proses transformasi ini.

Jadual 3.4 Data asal dan transformasi data

Atribut	Data asal	Transformasi data
red_blood_cell	Normal	1 = normal
	Abnormal	0 = abnormal
pus_cell	Normal	1 = normal
	Abnormal	0 = abnormal
pus_cell_clumps	Present	1 = present
	Notpresent	0 = notpresent
bacteria	Present	1 = present
	Notpresent	0 = notpresent
hypertension	Yes	1 = yes
	No	0 = no
diabetes_mellitus	Yes	1 = yes
	No	0 = no
coronary_artery_disease	Yes	1 = yes

	No	0 = no
appetite	Good	1 = poor
	Poor	0 = good
pedal_edema	Yes	1 = yes
	No	0 = no
anemia	Yes	1 = yes
	No	0 = no
classification	Ckd	0 = ckd
	Not ckd	1 = notckd

hypertension	diabetes_mellitus	coronary_artery_disease	appetite	pedal_edema	anemia	classification
1	1	0	0	0	0	0
0	0	0	0	0	0	0
0	1	0	1	0	1	0
1	0	0	1	1	1	0
0	0	0	0	0	0	0

Rajah 3.16 Petikan data yang telah ditransformasi

Proses normalisasi juga dijalankan ke atas set data. Set data menunjukkan pelbagai unit ukuran yang digunakan sebagai contoh, mg/dl dan mm/HG. Penggunaan unit ukuran boleh mempengaruhi proses analisis data (Han et al. 2012). Untuk mengelakkan kebergantungan data pada unit ukuran, proses normalisasi dilaksanakan. Proses normalisasi adalah proses mengubah nilai data supaya berada dalam julat 0.0 sehingga 1.0. Proses ini membantu untuk memberikan berat yang sama kepada semua atribut. Dalam kajian ini, *MinMaxScaler* dari *Scikit-learn* digunakan untuk proses normalisasi data. Normalisasi min-max merupakan transformasi linear yang dilakukan pada set data yang asal. Normalisasi ini memetakan satu nilai, v_i dalam A kepada nilai v'_i dalam julat $[new\_min_A, new\_max_A]$. Persamaan bagi proses normalisasi ini adalah seperti Persamaan 3.1.

$$v'_i = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \tag{3.1}$$

Rajah 3.17 menunjukkan sebahagian hasil normalisasi data dalam beberapa atribut menggunakan *MinMaxScaler*.

]:

	age	blood_pressure	specific_gravity	albumin	sugar	red_blood_cell	pus_cell	pus_cell_clumps	bacteria	blood_glucose_random	...	packed_cell_volu
0	0.522727	0.230769	0.75	0.2	0.0	1.0	1.0	0.0	0.0	0.211538	...	0.7777
1	0.056818	0.000000	0.75	0.8	0.0	1.0	1.0	0.0	0.0	0.211538	...	0.6444
2	0.681818	0.230769	0.25	0.4	0.6	1.0	1.0	0.0	0.0	0.856838	...	0.4888
3	0.522727	0.153846	0.00	0.8	0.0	1.0	0.0	1.0	0.0	0.202991	...	0.5111
4	0.556818	0.230769	0.25	0.4	0.0	1.0	1.0	0.0	0.0	0.179487	...	0.5777

5 rows × 25 columns

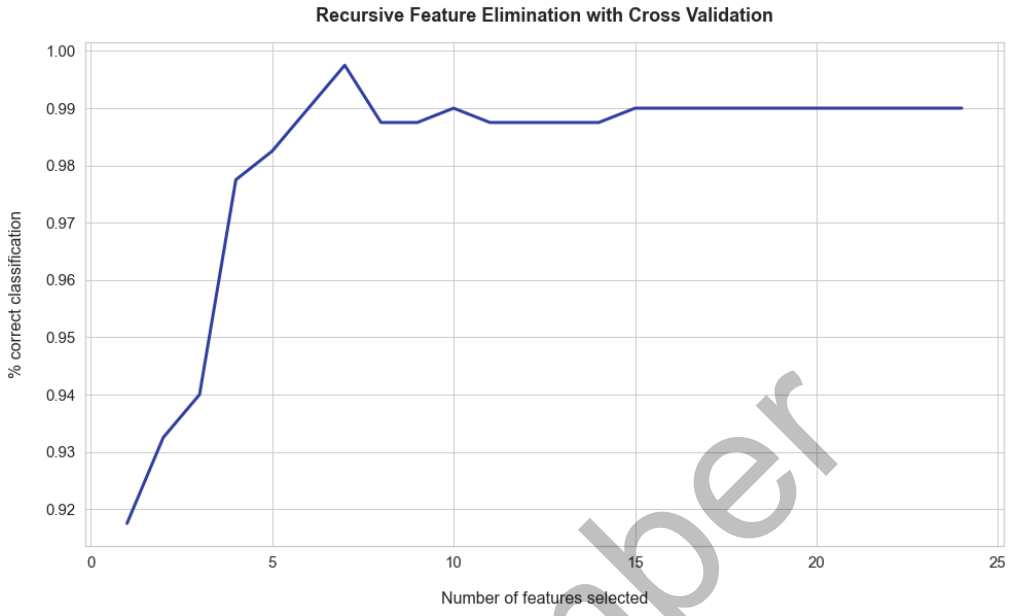
Rajah 3.17 Data yang telah dinormalisasi

3.4.3 Pemilihan Fitur

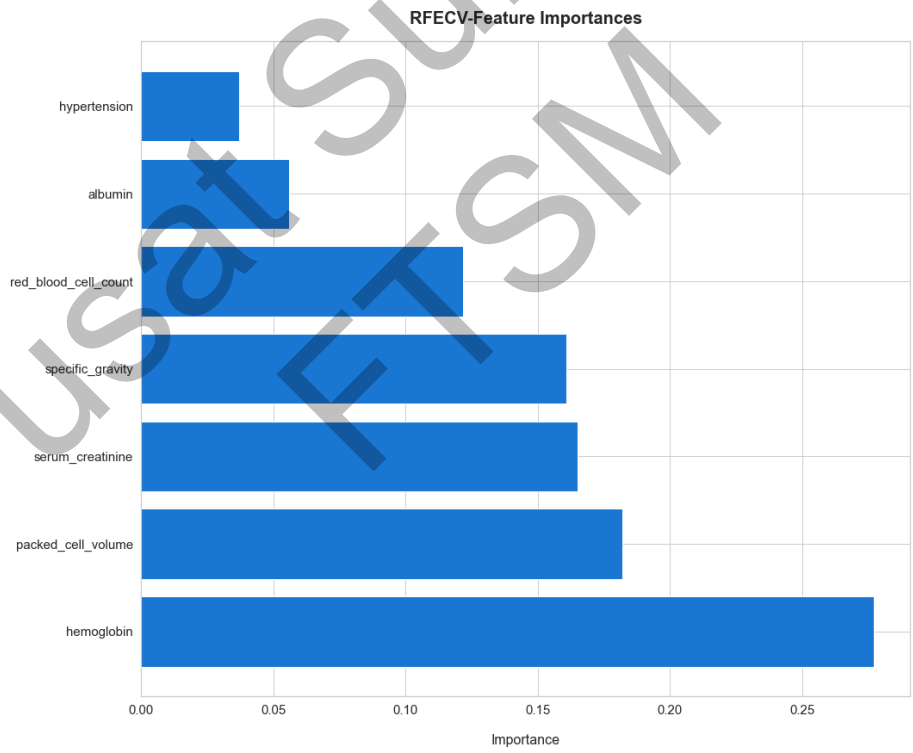
Pengurangan data bertujuan untuk mengecilkan perwakilan set data dari segi *volume* dan pada masa yang sama memelihara integriti data yang asal. Ini kerana, atribut yang tidak relevan boleh mempengaruhi prestasi sebahagian model pengelasan. Proses perlombongan pada set data yang kecil seharusnya menjadi lebih cekap kerana masa perlombongan menjadi lebih kecil dan seharusnya memberikan hasil analisis yang sama.

Pengurangan dimensi merupakan salah satu teknik dalam proses mengurangkan data (Han et al. 2012). Pemilihan subset atribut merupakan teknik di dalam pengurangan dimensi yang bertujuan untuk mengecilkan data dengan mengecualikan atribut yang tidak relevan atau bertindih. Dalam kajian ini, kaedah *Recursive Feature Elimination with Cross-Validation* (RFECV) digunakan sebagai kaedah pemilihan fitur. RFECV merupakan teknik pemilihan fitur *wrapper* yang berfungsi dengan memilih subset atribut yang optimum berdasarkan ketepatan klasifikasi. Teknik *wrapper* menilai subset atribut menggunakan algoritma pembelajaran mesin berdasarkan kualiti prestasi subset terhadap algoritma tersebut.

Atribut *classification* merupakan atribut sasaran dalam kajian ini. RFECV dari *Scikit-learn* digunakan bagi mengaplikasi teknik RFECV. Algoritma hutan rawak dengan parameter *random_state*=101 bersama *stratified* 10-*fold cross-validation* digunakan sebagai algoritma pembelajaran mesin dalam teknik RFECV. Rajah 3.18 menunjukkan graf garis bagi bilangan atribut yang dipilih dengan peratusan pengelasan yang tepat. Rajah 3.19 menunjukkan graf palang bagi tujuh atribut yang optimum. Berdasarkan graf, bilangan optimum bagi atribut yang digunakan dalam pemodelan ialah tujuh.



Rajah 3.18 Graf bagi bilangan atribut yang optimum menggunakan RFECV



Rajah 3.19 Graf palang bagi tujuh atribut yang optimum

Graf palang dalam Rajah 3.19 menunjukkan tujuh daripada 25 atribut yang terdiri daripada atribut *hypertension, albumin, red_blood_cell_count, specific_gravity, serum_creatinine, packed_cell_volume* dan *hemoglobin* telah dipilih bagi tujuan pengelasan.

3.5 PEMBAHAGIAN SET DATA

Pembahagian set data merupakan langkah penting bagi menilai prestasi model pembelajaran mesin. Set data dibahagikan kepada data latihan dan data pengujian. Data latihan digunakan bagi melatih model pengelasan. Data pengujian digunakan untuk membuat pengelasan dan prestasi model pengelasan akan dibandingkan berdasarkan data-data ini.

Dalam kajian ini, set data dibahagikan kepada 80% data latihan dan 20% data pengujian.

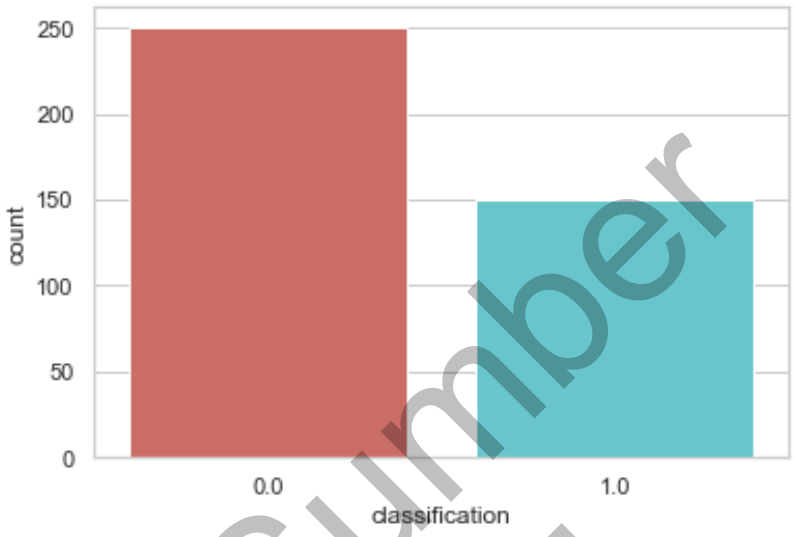
3.6 PEMBINAAN MODEL PENGELASAN

Selepas fasa pra-pemprosesan data, pembinaan model pengelasan dilaksanakan dengan menggunakan teknik pembelajaran mesin. Teknik pembelajaran mesin digunakan untuk mengenal pasti dan mengekstrak maklumat penting daripada data dan membuat keputusan dengan hanya sedikit intervensi manusia. Terdapat pelbagai teknik pembelajaran mesin yang boleh digunakan untuk mengekstrak maklumat daripada data antaranya pembelajaran berselia dan pembelajaran tidak terselia.

Pembelajaran berselia melibatkan set data berlabel untuk melatih algoritma model pembelajaran mesin membuat pengelasan dan meramalkan output. Sebagai contoh, algoritma Bayes naif digunakan bagi klasifikasi cendawan yang selamat dan tidak selamat dimakan. Pembelajaran tidak terselia menggunakan set data yang tidak berlabel. Algoritma ini menemui pola-pola yang membantu dalam menyelesaikan masalah pengelompokan dan sekutuan.

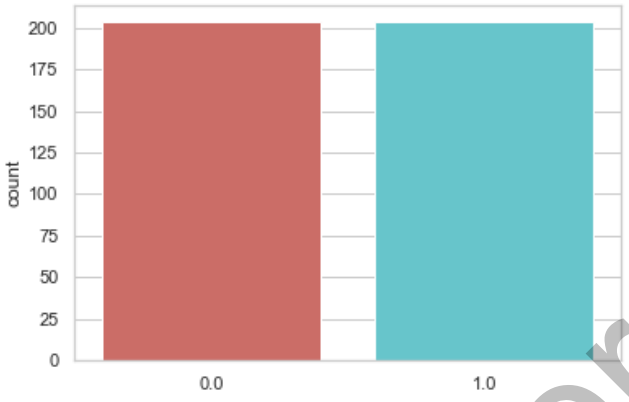
Kajian ini memfokuskan kepada pembelajaran berselia bagi pengelasan CKD. Selepas data dibahagikan kepada data latihan dan data pengujian, algoritma *Synthetic Minority Oversampling Technique* (SMOTE) diaplikasi kepada data latihan untuk mengatasi masalah data tidak seimbang. Data tidak seimbang merupakan satu keadaan di mana jumlah kelas di dalam atribut yang satu lebih sedikit atau lebih banyak berbanding jumlah kelas yang lainnya (Siringoringo 2018). Dalam kajian ini, berdasarkan atribut *classification*, jumlah kelas 0 atau yang menghidap CKD ialah 250 manakala jumlah kelas 1 atau yang tidak menghidap CKD ialah 150. Ini bermakna,

jumlah kelas 0 melebihi kelas 1 sebanyak 100. Nilai peratusan bagi kelas 0 ialah 62.5% dan nilai peratusan bagi kelas 1 ialah 37.5%. Rajah 3.20 menunjukkan carta palang bagi distribusi atribut *classification*.



Rajah 3.20 Jumlah data kelas 0 dan kelas 1 bagi atribut *classification*

Algoritma SMOTE diperkenalkan oleh Chawla et al. (2002). Algoritma SMOTE merupakan teknik *oversampling* di mana data kelas minoriti diduplikasi dengan data sintetik dengan menggunakan algoritma kNN. SMOTE bermula dengan memilih data minoriti secara rawak. Kemudian, berdasarkan jumlah data sintetik yang diperlukan, nilai k ditentukan. Dalam kajian ini, perpustakaan *Python* iaitu *imbalanced-learn* digunakan bagi aplikasi SMOTE kepada data latihan dengan parameter *sampling_strategy=auto, random_state=None, k_neighbors*=5 dan *n_jobs=None*. Rajah 3.21 menunjukkan distribusi atribut *classification* bagi data latihan selepas aplikasi SMOTE.



Rajah 3.21 Jumlah data bagi atribut *classification* selepas aplikasi SMOTE

Selepas itu, model pengelasan diaplikasikan kepada data. Model pengelasan asas yang terlibat adalah regresi logistik, Bayes naif, mesin sokongan vektor, hutan rawak dan MLP.

3.6.1 Regresi Logistik

Regresi logistik merupakan algoritma pembelajaran mesin dari bidang statistik. Regresi logistik dinamakan sempena fungsi yang digunakan dalam kaedah ini, iaitu fungsi logistik atau lebih dikenali sebagai fungsi sigmoid. Fungsi sigmoid ditunjukkan seperti Persamaan 3.2 di bawah.

$$\frac{1}{1 + e^{-value}} \tag{3.2}$$

Di mana e ialah nilai asas logaritma asli dan $value$ ialah nilai yang hendak ditransformasi. Dalam regresi logistik, nilai input (x) digabungkan secara linear menggunakan nilai pemberat atau pekali untuk meramal nilai output (y). Nilai output adalah dalam nilai binari (0 atau 1). Persamaan di bawah menunjukkan persamaan bagi regresi logistik.

$$y = \frac{e^{B_0 + B_1 \times x}}{1 + e^{B_0 + B_1 \times x}} \tag{3.3}$$

Di mana y ialah output yang diramalkan, B_0 ialah nilai pintasan, B_1 ialah pekali bagi nilai input tunggal (x). Setiap lajur dalam input data mempunyai nilai pekali B yang diperoleh daripada data latihan. Persamaan 3.3 ditulis dalam bentuk kebarangkalian input (X) berada dalam kelas (Y=1).

$$P(X) = P(Y = 1|X) \tag{3.4}$$

$$p(X) = \frac{e^{B_0+B_1\times x}}{1 + e^{B_0+B_1\times x}} \tag{3.5}$$

Persamaan 3.5 di atas ditulis seperti Persamaan 3.6 selepas penghapusan nilai *e* di sebelah kanan persamaan:

$$\ln\left(\frac{p(X)}{1-p(X)}\right) = B_0 + B_1 \times x \tag{3.6}$$

Nisbah di sebelah kiri Persamaan 3.6 dikenali sebagai nisbah *log of odds*. Ini menunjukkan model pengelasan regresi logistik adalah kombinasi linear daripada input, tetapi kombinasi ini berkaitan dengan *log of odds* bagi kelas lalai.

$$\ln(odds) = B_0 + B_1 \times x \tag{3.7}$$

$$odds = e^{B_0+B_1\times x} \tag{3.8}$$

Dalam kajian ini, model pengelasan Regresi logistik dibina menggunakan perpustakaan *Python*, *Scikit-learn*. Parameter yang digunakan dalam algoritma ini ialah *max_iter*=1000, *C*=1 dan *solver=lbfgs* dan *random_state=None.*

3.6.2 Bayes Naif

Algoritma Bayes naif merupakan algoritma model pengelasan yang mudah difahami dan banyak digunakan dalam kajian-kajian lain. Algoritma ini menggunakan teorem Bayes,

$$P(h|d) = \frac{P(d|h) \times P(h)}{P(d)} \quad (3.9)$$

Di mana

1. $P(h|d) =$ kebarangkalian hipotesis (h) berdasarkan data (d) dan dikenali sebagai kebarangkalian posterior.
2. $P(d|h) =$ kebarangkalian data (d) jika hipotesis (h) adalah benar.
3. $P(h) =$ kebarangkalian hipotesis (h) adalah benar.
4. $P(d) =$ kebarangkalian data (d).

Selepas kebarangkalian posterior dihitung, hipotesis dengan nilai kebarangkalian yang paling tinggi akan dipilih. Kebarangkalian ini secara formal dikenali sebagai maksimum posterior. Persamaan ini boleh ditulis seperti berikut,

$$MAP(h) = \max\big(P(h|d)\big) \quad (3.10)$$

$$MAP(h) = max\left(\frac{P(d|h) \times P(h)}{P(d)}\right) \quad (3.11)$$

Nilai $P(d)$ merupakan nilai pemalar,

$$MAP(h) = \max\,(d|h) \times P(h)) \quad (3.12)$$

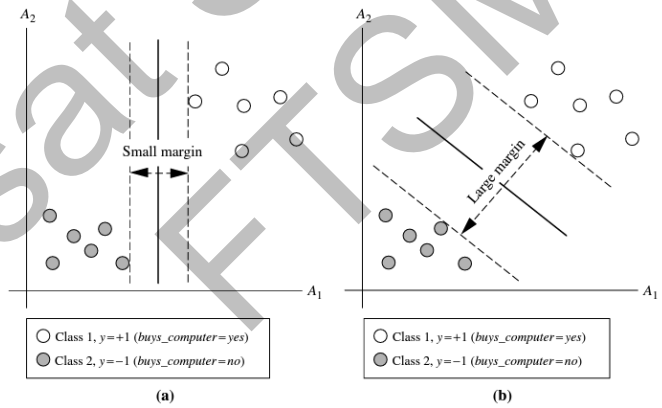
Masa pemprosesan algoritma Bayes naif adalah pendek kerana tiada pekali yang perlu dihitung semasa proses latihan.

Dalam kajian ini, model pengelasan Bayes naif dibina menggunakan fungsi *Scikit-learn*. Parameter yang terlibat dalam algoritma ini ialah *var_smoothing*=0.00187381742286.

3.6.3 Mesin Sokongan Vektor

Model pengelasan mesin sokongan vektor pertama kali diperkenalkan pada era 1970-an dan telah mendapat perhatian sejak 15 tahun ke belakang (Bergin & Reilly 2006). Mesin sokongan vektor sesuai digunakan bagi data linear dan data tidak linear. Mesin sokongan vektor berfungsi dengan memetakan data latihan yang diberi kepada satu ruang ciri berdimensi tinggi dengan menempatkan hipersatah pemisah yang optimum dan boleh mengasingkan kelas dalam satu-satu atribut (Han et al. 2012). Persoalan yang timbul adalah, bagaimana algoritma ini mencari hipersatah pemisah yang optimum?

Algoritma mesin sokongan vektor menyelesaikan masalah ini dengan mencari hipersatah dengan maksimum marginal. Rajah 3.22 menunjukkan margin bagi mesin sokongan vektor. Berdasarkan Rajah 3.22, kedua-dua hipersatah (a) dan (b) boleh mengelaskan data dengan betul. Akan tetapi, hipersatah dengan margin yang besar adalah lebih tepat.



Rajah 3.22 Margin bagi mesin sokongan vektor

Sumber: Han et al. (2012)

Jarak antara hipersatah dengan titik data terdekat dikenali sebagai margin. Margin dihitung sebagai jarak serenjang dari garis hipersatah kepada titik data terdekat. Hanya titik data terdekat yang terlibat dalam pembinaan model mesin sokongan vektor. Titik terdekat ini dikenali sebagai vektor sokongan.

Mesin sokongan vektor mempunyai satu parameter, *C* yang digunakan bagi kes-kes di mana titik-titik data tidak dapat diasingkan secara linear. Parameter ini digunakan

sebagai penalti dan mempengaruhi titik data yang dibenarkan untuk jatuh di sempadan kelas yang salah. Semakin besar nilai *C*, semakin kurang sensitif algoritma mesin sokongan vektor kepada data latihan yang bermakna margin hipersatah lebih kecil dan lebih banyak titik data dibenarkan untuk jatuh di sempadan kelas yang salah. Sebaliknya, semakin kecil nilai parameter *C*, semakin sensitif algoritma mesin sokongan vektor kepada data latihan yang bermakna lebih besar margin hipersatah, sekali gus lebih sedikir titik data dibenarkan untuk jatuh di sempadan kelas yang salah (Brownlee 2016).

Pembelajaran hipersatah dalam algoritma mesin sokongan vektor dilaksanakan dengan transformasi masalah menjadi algebra linear dengan peranan kernel. Ukuran kesamaan atau jarak antara titik data baru dengan vektor sokongan ditentukan oleh kernel. Terdapat beberapa kernel dalam mesin sokongan vektor iaitu kernel linear, polinomial dan fungsi radial basis.

Bagi kernel linear, persamaan bagi meramalkan input baru menggunakan hasil darab bintik antara input data,x dan vektor sokongan, xi. Persamaan bagi kernel linear ditulis seperti Persamaan 3.13.

$$K(x, x_i) = \sum (x \times x_i) \tag{3.13}$$

Kernel polynomial membenarkan garis melengkung di dalam ruang ciri mesin sokongan vektor. Persamaan bagi kernel polinomial adalah seperti Persamaan 3.14 di mana d adalah darjah bagi polinomial:

$$K(x, x_i) = 1 + \sum (x \times x_i)^d \tag{3.14}$$

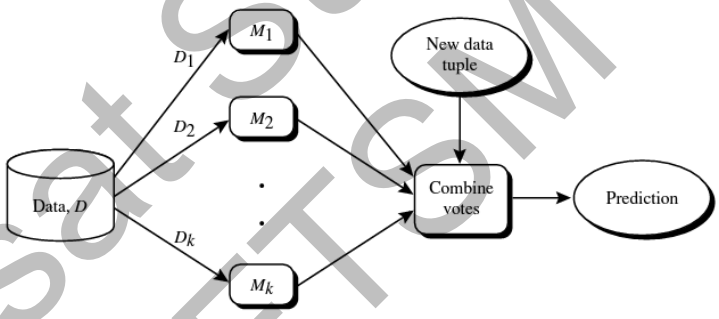
Persamaan bagi kernel fungsi radial basis adalah seperti Persamaan 3.15 di bawah di mana *gamma* ialah parameter yang mesti dispesifikasikan kepada algoritma pembelajaran. Nilai *gamma* biasanya terletak di dalam julat $(0,1)$.

$$K(x, x_i) = e^{(-gamma \times \sum(x - x_i{}^2))} \quad (3.15)$$

Dalam kajian ini, mesin sokongan vektor dengan kernel fungsi radial basis diaplikasi menggunakan fungsi *Scikit-learn.* Parameter yang terlibat dalam algoritma ini ialah *gamma=auto, probability=True, C*=10 dan *kernel=linear*.

3.6.4 Hutan Rawak

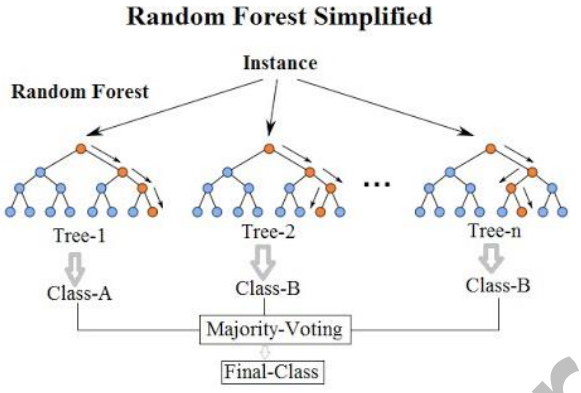
Pembelajaran gabungan berfungsi dengan menggabungkan beberapa siri model pengelasan $M_1, M_2, \ldots, M_k$ sebagai model pengelasan asas. Pembelajaran gabungan bertujuan untuk membina dan menambah baik ketepatan model sedia ada. Model pengelasan gabungan membuat peramalan dengan kaedah pengundian daripada model asas. Rajah 3.23 menunjukkan ilustrasi secara ringkas model pengelasan gabungan.



Rajah 3.23 Pembelajaran gabungan

Sumber: Han et al. (2012)

Terdapat beberapa jenis model pengelasan gabungan antaranya *bagging*, *boosting* dan *stacking*. Hutan rawak merupakan lanjutan daripada kaedah *bagging* bagi pokok keputusan. Hutan rawak membina beberapa set algoritma pokok keputusan dan membuat pengelasan berdasarkan undian daripada algoritma pokok keputusan tersebut. Semakin tinggi bilangan pokok yang tiada korelasi, semakin tinggi ketepatan algoritma hutan rawak. Rajah 3.24 menunjukkan ilustrasi bagi algoritma hutan rawak.



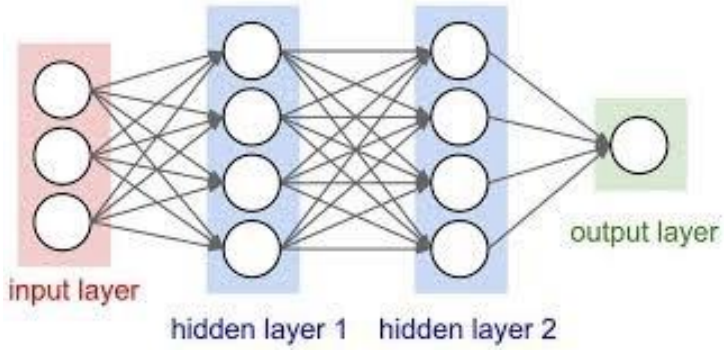
Rajah 3.24 Ilustrasi bagi algoritma hutan rawak

Sumber: Koehrsen (2017)

Algoritma hutan rawak dalam kajian ini dibina dengan menggunakan fungsi *Scikit-learn*. Parameter yang terlibat dalam algoritma ini ialah *bootstrap=True*, *max_features=sqrt*, dan *n_estimators*=100.

3.6.5 Multilayer Perceptron (MLP)

Rangkaian neuron buatan berfungsi seperti jaringan saraf otak manusia, di mana neuron saling terhubung satu dengan lainnya untuk memproses informasi. (Yunus 2020). Algoritma MLP merupakan sejenis rangkaian perceptron untuk membentuk model pengelasan. Algoritma MLP memiliki lapisan input, lapisan tersembunyi dan lapisan output. Lapisan input merupakan lapisan yang menerima data untuk diproses pada lapisan seterusnya. Lapisan tersembunyi memiliki prosedur dan pemberat untuk menghasilkan output melalui fungsi pengaktifan. Lapisan output merupakan lapisan terakhir yang menghasilkan sistem output (Yunus 2020). Bilangan lapisan tersembunyi boleh ditambah untuk meningkatkan kerumitan model pengelasan mengikut keperluan tugasan. Rajah 3.25 menunjukkan ilustrasi bagi model pengelasan MLP dengan dua lapisan tersembunyi.



Rajah 3.25 Ilustrasi bagi algoritma MLP

Sumber: Pandey (2021)

Fungsi pengaktifan memetakan input dengan pemberat kepada output neuron. Fungsi pengaktifan ini mempengaruhi nilai ambang bagi mengaktifkan neuron. Terdapat beberapa fungsi pengaktifan bagi algoritma MLP antaranya fungsi sigmoid, fungsi *rectified linear unit* (ReLU) dan fungsi tangen hiperbolik (tanh).

Kajian ini memfokuskan penggunaan algoritma MLP dengan tiga lapisan tersembunyi dengan lapan neuron, dan fungsi tangen hiperbolik sebagai fungsi pengaktifan. Model MLP dibina menggunakan fungsi *Scikit-learn*. Parameter yang terlibat dalam algoritma ini ialah *solver=adam, max_iter*=1000, *activation=tanh* dan *hidden_layer_sizes*=(8,8,8).

3.7 PENENTUKURAN MODEL PENGELASAN

Sebagai alternatif kepada meramalkan label kelas secara terus, model pengelasan seharusnya boleh meramalkan kebarangkalian bagi setiap data berada dalam satu-satu kelas. Meramalkan kebarangkalian bagi model pengelasan memberikan impak dari segi interpretasi ketepatan model pengelasan. Sebagai contoh, terdapat dua jenis model pengelasan, Model A dan Model B. Model A memberikan 86% ketepatan dan 0.86 keyakinan dalam setiap ramalan yang dibuat. Model B memberikan 86% ketepatan dan 0.98 keyakinan dalam setiap ramalan yang dibuat. Dalam contoh ini, model A adalah lebih baik berbanding model B kerana model A sentiasa memberikan ketepatan 86%. Sebaliknya, model B terlalu yakin dalam setiap ramalan yang dibuat. Sekiranya sesuatu model pengelasan memberikan kebarangkalian 0.86 sesuatu data itu terletak dalam kelas positif, seharusnya jangkaan terhadap ketepatan data tersebut berada dalam kelas positif adalah 86%.

Penentukuran model pengelasan merujuk kepada proses transformasi skor pengelasan kepada nilai kebarangkalian (Sarkar 2019). Penentukuran model penting sekiranya kebarangkalian bagi pengelasan adalah penting. Dalam kajian ini, set data perubatan digunakan. Model pengelasan yang dibina seharusnya berupaya memberikan kebarangkalian bagi label kelas. Sebagai contoh, kebarangkalian seorang pesakit dijangkiti penyakit kronik pada peringkat awal adalah penting bagi seorang doktor untuk memberikan preskripsi yang tepat kepada pesakit tersebut. Terdapat dua kaedah untuk menentukur model pengelasan iaitu kaedah Platt dan regresi isotonik. Kaedah Platt sesuai digunakan jika set data adalah kecil iaitu kurang 1000 (Niculescu-Mizil & Caruana 2005). Kajian ini memfokuskan kaedah Platt dalam penentukuran model pengelasan asas.

Penentukuran kebarangkalian menggunakan fungsi sigmoid untuk mengekstrak kebarangkalian daripada model pengelasan seperti dalam Persamaan 2.1. Parameter A dan B dianggarkan melalui kaedah *maximum likelihood estimation*. Rajah 3.26 menunjukkan pengiraan parameter A dan B.

Log Loss(L) = ***-Yi * log(Pi) - (1-Yi) * log(1-Pi)*** (for Binary Classification)

where , $Pi = 1/(1 + e^{(}Af(x) + B))$

Let, Pi=a;
then,
Log Loss(L) = -Yi * log(a) - (1-Yi) * log(1-a) ———(1)

Now , we will use **Gradient Descent** to find optimal values for parameters **A and B**
$\therefore A' = A - r[\partial L/\partial A]$ **and** $B' = B - r[\partial L/\partial B]$

Equation : $\partial L/\partial A = \partial L/\partial a * \partial a/\partial A$ ———(2)

From Equation———(1):
$\partial L/\partial a = (-Yi/a) - ((1 - Yi)/(1 - a))(-1)$
$\partial L/\partial a = (a - Yi)/(a/(1 - a))$

$\partial a/\partial A = -a(1 - a)(f)$

From Equation ———(2):
$\partial L/\partial A = -(a - Yi)(f) = (Yi - a)(f) = (Yi - Pi)(f)$

Similarly,
$\partial L/\partial B = (Yi - a) = (Yi - Pi)$

Rajah 3.26 Pengiraan parameter A dan B

Sumber: Sarkar (2019)

Dalam kajian ini, penentukuran kebarangkalian diaplikasi dengan menggunakan fungsi *Scikit-learn*. Penentukuran kebarangkalian diaplikasi kepada model pengelasan

asas. Model pengelasan asas yang telah ditentukur diwakili sebagai regresi logistik-Platt, Bayes naif-Platt, Mesin sokongan vektor-Platt, hutan rawak-Platt dan MLP-Platt.

3.8 PEMBELAJARAN GABUNGAN

Kaedah pembelajaran gabungan digunakan dalam dua peringkat. Peringkat pertama ialah pembelajaran gabungan tanpa aplikasi penentukuran kebarangkalian terhadap model pengelasan asas dan peringkat kedua ialah pembelajaran gabungan dengan aplikasi penentukuran kebarangkalian terhadap model pengelasan asas. Output bagi setiap model pengelasan asas adalah dalam bentuk kebarangkalian. Nilai kebarangkalian ini menjadi nilai input bagi setiap model pembelajaran gabungan. Jadual 3.5, Jadual 3.6, Jadual 3.7, Jadual 3.8 dan Jadual 3.9 menunjukkan contoh output bagi setiap model pengelasan asas yang digunakan dalam kajian ini iaitu regresi logistik, Bayes naif, mesin sokongan vektor, hutan rawak dan MLP. Terdapat empat kaedah pembelajaran gabungan yang digunakan dalam kajian ini iaitu *simple averaging*, *weighted averaging*, *stacking A* dan *stacking B*.

Jadual 3.5 Output bagi model regresi logistik

```
array([7.08213200e-03, 8.78333416e-01, 8.61602633e-01, 2.03559997e-04,
       5.40218454e-04, 5.74284899e-02, 1.56520976e-01, 8.48739754e-01,
       3.97335104e-04, 5.31978279e-01, 1.22958292e-02, 5.97488183e-02,
       3.93118352e-01, 8.20425434e-01, 1.13428098e-03, 2.33633897e-02,
       8.60762059e-01, 1.89456403e-01, 1.09758583e-02, 3.61922884e-03,
       2.35847610e-03, 9.13476677e-01, 5.24233045e-03, 7.53391627e-01,
       9.31672452e-03, 9.24430844e-01, 2.83814987e-02, 4.21364016e-04,
       8.41951578e-01, 4.10531342e-01, 6.96915783e-03, 9.33956500e-01,
       9.15659878e-01, 1.34537111e-03, 2.89309155e-01, 3.17962949e-03,
       4.23893823e-03, 2.99675617e-01, 5.76478575e-02, 9.72899646e-01,
       6.90272841e-02, 8.35232349e-01, 1.23819330e-01, 1.15259367e-01,
       9.42979233e-01, 4.23239087e-03, 9.48631674e-01, 9.54316584e-01,
       7.47012447e-03, 1.79188308e-01, 7.43473135e-01, 2.72574884e-02,
       7.78550786e-01, 1.98772242e-03, 8.39145403e-01, 9.75518540e-01,
       1.73892176e-01, 9.42415367e-04, 1.67807018e-02, 8.93579878e-03,
       1.74163210e-03, 9.20229079e-01, 3.99489031e-02, 9.02814056e-01,
       3.00980800e-04, 9.73062277e-01, 1.74749682e-04, 5.65251199e-03,
       4.41477525e-05, 7.59714445e-01, 2.21371243e-01, 8.02879294e-01,
       4.01172477e-02, 8.64513877e-04, 9.27835866e-03, 5.45822749e-05,
       9.61697521e-01, 8.77725172e-01, 8.85430914e-01, 9.28959477e-01])
```

Jadual 3.6 Output bagi model Bayes naif

```
array([0.00000000e+000, 9.99999214e-001, 9.99980350e-001, 0.00000000e+000,
       0.00000000e+000, 0.00000000e+000, 4.84510008e-201, 9.99997432e-001,
       0.00000000e+000, 9.99421035e-001, 0.00000000e+000, 9.43457171e-031,
       9.75316058e-001, 9.99998549e-001, 0.00000000e+000, 0.00000000e+000,
       9.99999621e-001, 3.15357556e-002, 0.00000000e+000, 0.00000000e+000,
       0.00000000e+000, 9.99999770e-001, 0.00000000e+000, 9.99986490e-001,
       0.00000000e+000, 9.99999779e-001, 0.00000000e+000, 0.00000000e+000,
       9.99999460e-001, 7.72217413e-001, 0.00000000e+000, 9.99999966e-001,
       9.99998484e-001, 0.00000000e+000, 2.95366881e-022, 0.00000000e+000,
       0.00000000e+000, 4.53606474e-025, 6.14652619e-093, 9.99999988e-001,
       0.00000000e+000, 9.99998417e-001, 7.91170835e-053, 0.00000000e+000,
       9.99999869e-001, 0.00000000e+000, 9.99999845e-001, 9.99999901e-001,
       0.00000000e+000, 7.57075543e-089, 2.99552532e-019, 4.29156791e-096,
       9.99994611e-001, 0.00000000e+000, 9.99997499e-001, 9.99999979e-001,
       1.08737208e-003, 0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
       0.00000000e+000, 9.99999479e-001, 2.93809067e-094, 9.99998103e-001,
       0.00000000e+000, 9.99999961e-001, 0.00000000e+000, 0.00000000e+000,
       0.00000000e+000, 9.99991385e-001, 4.15434947e-088, 9.99997882e-001,
       8.99275710e-095, 0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
       9.99999891e-001, 9.99988587e-001, 9.99999436e-001, 9.99999625e-001])
```

Jadual 3.7 Output bagi model mesin sokongan vektor

```
array([2.45045429e-06, 9.80669228e-01, 9.58566831e-01, 1.00000010e-07,
       1.00000010e-07, 4.18532127e-03, 6.85660202e-04, 9.64719129e-01,
       1.00000010e-07, 3.17390794e-01, 3.36225548e-05, 3.64928437e-05,
       7.44269010e-02, 9.41175049e-01, 1.00000010e-07, 2.43949369e-04,
       9.78882683e-01, 6.89708417e-03, 7.95618002e-05, 1.50378365e-06,
       1.00000010e-07, 9.96506106e-01, 4.50788790e-06, 8.36208042e-01,
       6.86594192e-05, 9.99983540e-01, 3.74685087e-04, 1.00000010e-07,
       9.61826113e-01, 2.01522062e-01, 9.72042190e-06, 9.99989560e-01,
       9.92054244e-01, 1.14012793e-07, 4.19867097e-02, 1.00000010e-07,
       4.50928186e-06, 1.35904274e-02, 9.28069573e-05, 9.99999901e-01,
       8.65694840e-05, 9.58851866e-01, 2.24761093e-04, 2.50233967e-04,
       9.99985056e-01, 4.41880992e-06, 9.99985425e-01, 9.99997358e-01,
       5.68222790e-06, 2.50797480e-03, 6.68815258e-01, 7.87273896e-06,
       8.67148114e-01, 1.00246684e-07, 9.71867968e-01, 9.99999952e-01,
       2.40679266e-03, 1.00000010e-07, 1.80238860e-04, 9.80175154e-06,
       1.03826364e-07, 9.92575709e-01, 2.79534104e-05, 9.86833182e-01,
       1.00000010e-07, 9.99999914e-01, 1.00000010e-07, 6.13516626e-06,
       1.00000010e-07, 8.53036830e-01, 6.52725933e-03, 9.25706802e-01,
       2.60402828e-05, 1.00000010e-07, 2.46706548e-05, 1.00000010e-07,
       9.99998691e-01, 9.77178316e-01, 9.91331119e-01, 9.96159065e-01])
```

Jadual 3.8 Output bagi model hutan rawak

```
array([0.  , 0.96, 0.95, 0.  , 0.  , 0.  , 0.  , 1.  , 0.  , 0.27, 0.  ,
       0.  , 0.  , 1.  , 0.  , 0.  , 1.  , 0.01, 0.02, 0.  , 0.  , 1.  ,
       0.  , 0.96, 0.  , 0.99, 0.  , 0.  , 0.97, 0.22, 0.  , 1.  , 0.83,
       0.  , 0.22, 0.  , 0.  , 0.04, 0.  , 0.99, 0.03, 1.  , 0.  , 0.03,
       1.  , 0.  , 0.97, 0.99, 0.  , 0.  , 0.18, 0.  , 0.86, 0.  , 0.99,
       1.  , 0.  , 0.  , 0.  , 0.  , 0.  , 1.  , 0.  , 1.  , 0.  , 1.  ,
       0.  , 0.03, 0.  , 1.  , 0.11, 0.99, 0.  , 0.  , 0.  , 0.  , 1.  ,
       0.96, 1.  , 0.99])
```

Jadual 3.9 Output bagi model MLP

```
array([0.00162667, 0.99303291, 0.98964967, 0.00159235, 0.00159297,
       0.00177107, 0.00179204, 0.99119893, 0.00159631, 0.10244152,
       0.00167511, 0.0018196 , 0.01709586, 0.96832249, 0.00159869,
       0.00167379, 0.987851  , 0.00387789, 0.00168884, 0.00163063,
       0.00160144, 0.99487892, 0.00163928, 0.94803566, 0.00171163,
       0.99582017, 0.001733  , 0.00159279, 0.9834827 , 0.01893637,
       0.00163122, 0.99585851, 0.99561989, 0.00160597, 0.00423099,
       0.0016074 , 0.00163307, 0.00279471, 0.00177102, 0.99729875,
       0.00168784, 0.98882969, 0.00190575, 0.0016995 , 0.99660147,
       0.00163281, 0.99664056, 0.99631391, 0.00161384, 0.00205073,
       0.29806796, 0.00169406, 0.9548145 , 0.00159927, 0.97604939,
       0.99751018, 0.00335341, 0.00160181, 0.00171731, 0.00160982,
       0.00160161, 0.99533468, 0.00173206, 0.99475345, 0.00159431,
       0.99749615, 0.00159072, 0.00161913, 0.00159078, 0.9140426 ,
       0.00216353, 0.96713359, 0.00171313, 0.00159754, 0.00164616,
       0.00159049, 0.99731185, 0.9914722 , 0.98902666, 0.99561846])
```

3.8.1 *Simple Averaging*

Kaedah pembelajaran gabungan *simple averaging* menggunakan purata sebagai pemberat daripada semua model pengelasan untuk membuat ramalan. Dalam kajian ini terdapat lima model pengelasan dan setiap model diberikan pemberat sebanyak 0.2. Nilai 0.2 dipilih kerana terdapat lima model pengelasan asas yang terlibat dan setiap model diberikan pemberat yang sama nilai dimana jumlah nilai pemberat ialah 1. Model *simple averaging* dengan aplikasi penentukuran kebarangkalian terhadap model pengelasan asas diwakili sebagai *simple averaging-Platt.*

3.8.2 *Weighted Averaging*

Kaedah ini adalah lanjutan daripada kaedah *simple averaging*. Bagi kaedah ini, nilai 1-*expected calibration error* (ECE) bagi setiap model pengelasan asas dijadikan sebagai pemberat untuk membuat ramalan akhir. Semakin kecil nilai ECE, semakin rendah nilai ralat tentukur. Oleh itu, bagi memberi nilai pemberat yang terbaik, nilai 1-ECE digunakan. Model *weighted averaging* dengan aplikasi penentukuran kebarangkalian terhadap model pengelasan asas diwakili sebagai *weighted averaging-Platt.*

3.8.3 *Stacking A*

Pembelajaran gabungan yang diaplikasi pada peringkat ini ialah *stacking. Stacking* berfungsi dengan menggabungkan ramalan yang dibuat berdasarkan sekurang-kurangnya dua model pengelasan asas kemudian menggunakan *meta-learner* untuk membuat ramalan yang baru.

Sebagai contoh, untuk menyelesaikan tugasan pengelasan, model kNN, regresi logistik dan Bayes naif dipilih sebagai model pengelasan asas dan menggunakan model NN sebagai meta-learner untuk membuat ramalan akhir berdasarkan input daripada ketiga-tiga model pengelasan asas. Dalam kajian ini, model regresi logistik, Bayes naif, mesin sokongan vektor, hutan rawak dan MLP digunakan sebagai model pengelasan asas dan *meta-learner* yang digunakan dalam kajian ini ialah hutan rawak. Algoritma hutan rawak dipilih sebagai *meta-learner* kerana algoritma hutan rawak memberikan prestasi yang paling baik dari segi diskriminasi berbanding model pengelasan asas yang lain. Jadual 3.10 menunjukkan contoh input bagi *meta-learner*. Model *stacking A* dengan aplikasi penentukuran kebarangkalian terhadap model pengelasan asas diwakili sebagai *stacking A-Platt.*

Jadual 3.10 Input bagi model *stacking A*

	LR + Platt	GNB + Platt	SVM	RF	MLP
0	0.827423	0.897762	8.908118e-01	0.98	0.974740
1	0.035712	0.018055	9.280612e-04	0.00	0.002357
2	0.000282	0.018051	1.000000e-07	0.00	0.001615
3	0.000404	0.018051	7.489518e-07	0.00	0.001625
4	0.881479	0.897778	9.556323e-01	1.00	0.983988
...	...	...	...	...	...
391	0.979158	0.897781	9.999953e-01	1.00	0.996760
392	0.658296	0.897289	5.484046e-01	0.88	0.600153
393	0.930192	0.897780	9.887242e-01	1.00	0.988300
394	0.698138	0.897536	6.654967e-01	0.94	0.776014
395	0.908503	0.897779	9.735418e-01	1.00	0.990638

3.8.4 *Stacking B*

Model pembelajaran gabungan *stacking B* dibina dengan menghitung median bagi output daripada kelima-lima model pengelasan asas. Nilai-nilai median ini kemudian digabungkan dengan nilai output bagi setiap model pengelasan. *Meta-learner* yang digunakan dalam model ini ialah hutan rawak. Jadual 3.11 menunjukkan contoh input bagi *meta-learner*. Model *stacking B* dengan aplikasi penentukuran kebarangkalian terhadap model pengelasan asas diwakili sebagai *stacking B-Platt.*

Jadual 3.11 Input bagi model *stacking B*

	LR + Platt	GNB + Platt	SVM	RF	MLP	Median
0	0.827423	0.897762	8.908118e-01	0.98	0.974740	0.897762
1	0.035712	0.018055	9.280612e-04	0.00	0.002357	0.002357
2	0.000282	0.018051	1.000000e-07	0.00	0.001615	0.000282
3	0.000404	0.018051	7.489518e-07	0.00	0.001625	0.000404
4	0.881479	0.897778	9.556323e-01	1.00	0.983988	0.955632
...	...	...	...	...	...	...
391	0.979158	0.897781	9.999953e-01	1.00	0.996760	0.996760
392	0.658296	0.897289	5.484046e-01	0.88	0.600153	0.658296
393	0.930192	0.897780	9.887242e-01	1.00	0.988300	0.988300
394	0.698138	0.897536	6.654967e-01	0.94	0.776014	0.776014
395	0.908503	0.897779	9.735418e-01	1.00	0.990638	0.973542

3.9 PENILAIAN BAGI MODEL PENGELASAN ASAS, PENENTUKURAN KEBARANGKALIAN DAN MODEL PEMBELAJARAN GABUNGAN

Ukuran penilaian penting bagi menjelaskan prestasi sesuatu model pengelasan. Ukuran penilaian digunakan untuk mengenal pasti keberkesanan model yang digunakan terhadap set data. Dalam kajian ini, diskriminasi dan penentukuran dinilai. Diskriminasi ialah kebolehan untuk meramal sama ada seseorang pesakit menghidap CKD atau tidak manakala penentukuran mengukur konsistensi antara kebarangkalian ramalan dengan kebarangkalian sebenar (Fan et al. 2021). Ketepatan, ukuran F dan AUROC adalah contoh-contoh ukuran penilaian yang sering digunakan bagi membandingkan prestasi model pengelasan. Dalam kajian ini, penilaian bagi model pengelasan yang digunakan juga diukur dengan menggunakan kehilangan log (*log loss*), skor Brier, dan ECE. Terdapat beberapa terma yang perlu dikenal pasti dalam ukuran penilaian iaitu:

1. Positif benar (TP): Kes di mana ramalan dan nilai sebenar adalah 1 (benar).
2. Negatif benar (TN): Kes di mana ramalan dan nilai sebenar adalah 0 (salah).
3. Positif palsu (FP): Kes di mana ramalan adalah 1 dan nilai sebenar adalah 0. Juga dikenali sebagai kesalahan jenis I.
4. Negatif palsu (FN): Kes di mana ramalan adalah 0 dan nilai sebenar adalah 1. Juga dikenali sebagai kesalahan jenis II.