

## PERBANDINGAN AKURASI KLASIFIKASI DARI ALGORITMA NAIVE BAYES, C4.5, DAN ONER (1R)

M Zainal Arifin

**Abstrak** : Artikel ini menjabarkan secara ringkas mengenai beberapa jenis algoritma klasifikasi seperti Bayes classifiers, Decision Tree classifiers, dan Rule-Based classifiers. Dari beberapa jenis algoritma klasifikasi tersebut diperkenalkan algoritma Naïve Bayes, C4.5, dan OneR yang merepresentasikan ketiga jenis algoritma tersebut. Dengan menggunakan WEKA Experimenter dan dataset yang bersumber dari UCI, dilakukan perbandingan terhadap ketiga algoritma tersebut. Dari hasil perbandingan didapatkan bahwa secara umum algoritma C4.5 memiliki tingkat akurasi yang tinggi pada dataset yang relatif cukup independen. Sedangkan algoritma OneR mempunyai tingkat akurasi yang tinggi yang sebanding dengan algoritma C4.5 pada dataset yang memiliki jumlah nilai kelas yang sedikit, dan memiliki tingkat akurasi yang sangat rendah ketika diterapkan pada dataset yang memiliki jumlah nilai kelas yang besar.

**Keywords** : Naïve Bayes, C4.5, OneR.

Algoritma yang bertujuan untuk melakukan klasifikasi pada bidang *machine learning* dapat diukur berdasarkan dua kriteria utama: tingkat akurasi pada suatu test set independen, dan tingkat kompleksitas dari algoritma itu sendiri. Hubungan antara kedua kriteria itu seringkali berbanding terbalik, ketika tingkat akurasi tinggi didapati algoritma tersebut merupakan algoritma yang kompleks. Dan ketika terdapat suatu algoritma yang cukup sederhana dan mudah dimengerti, algoritma itu hanya optimal pada beberapa kasus saja, dengan tingkat akurasi yang tidak independen (bergantung pada dataset yang digunakan) [1].

Artikel ini berisi laporan hasil eksperimen pada algoritma yang cukup banyak digunakan dalam bidang *machine learning*. Eksperimen bertujuan untuk mengukur performansi algoritma pada dataset yang secara umum digunakan pada penelitian-penelitian *machine learning*. Faktor performansi utama yang ingin diperbandingkan pada

eksperimen ini adalah tingkat akurasi dari algoritma independen terhadap dataset yang digunakan. Algoritma yang diperbandingkan beserta lingkungan eksperimen akan dijelaskan lebih detail pada bab-bab berikutnya.

### METODE

Eksperimen ini menggunakan tiga jenis algoritma klasifikasi dan ingin membandingkan algoritma dari ketiga jenis tersebut dalam proses klasifikasi terhadap suatu dataset. Ketiga jenis algoritma klasifikasi yang digunakan adalah *Bayes Classifiers*, *Decision Tree Classifiers*, dan *Rule-Based Classifiers*.

### Bayes Classifiers

Bayes classifiers menggunakan pendekatan probabilistik dalam melakukan proses inferensi. Pada machine learning hal ini menjadi penting karena bayes classifiers menggunakan pendekatan kuantitatif dalam pembobotan atribut-atribut pendukung hipotesis.

Beberapa algoritma yang tergolong *Bayes classifiers* menerapkan pendekatan-pendekatan yang paling praktis pada suatu permasalahan tertentu. Michie menunjukkan pada artikelnya [2] bahwa *Bayes classifiers* juga kompetitif dengan algoritma lain pada banyak kasus dan bahkan pada beberapa kasus algoritma ini lebih unggul.

Klasifikasi yang menggunakan *decision tree* memiliki karakteristik sebagai berikut [3]:

- Setiap *training example* dapat meningkatkan atau menurunkan nilai probabilitas untuk kebenaran suatu hipotesa secara incremental. Hal ini memudahkan adanya pendekatan yang fleksibel dibanding algoritma yang langsung menggugurkan suatu hipotesis ketika ditemukan bahwa hipotesis tersebut tidak konsisten dengan satu *training example*.
- Pengetahuan yang didapat sebelumnya dapat dikombinasikan dengan data yang diamati untuk menentukan nilai probabilitas akhir pada suatu hipotesis. Hal ini dapat dilakukan dengan menambahkan nilai probabilitas sebelumnya pada setiap kandidat hipotesis, atau dengan mendistribusikan suatu nilai probabilitas berdasar data yang diamati pada setiap hipotesis yang mungkin.
- Dapat mengakomodasi hipotesis yang menggunakan prediksi probabilistik.
- Instans baru dapat diklasifikasikan dengan menggabungkan prediksi dari beberapa hipotesis yang berbobot sesuai probabilitas tiap hipotesis.

- Dapat menghasilkan suatu *decision making* yang optimal pada kasus dimana secara perhitungan biasa sulit untuk dilakukan.

### Decision Tree Classifiers

*Decision tree* merepresentasikan suatu klasifikasi berdasar pendekatan yang supervised. Sebuah *decision tree* merupakan sebuah struktur pohon sederhana dimana simpul non-terminal mencerminkan pengujian terhadap satu atau lebih atribut dari dataset, cabang dari pohon mencerminkan nilai dari atribut dan simpul terminal mencerminkan hasil akhir klasifikasi yang diambil.

Klasifikasi yang menggunakan *decision tree* memiliki beberapa karakteristik sebagai berikut:

- Instans dideskripsikan dalam bentuk pasangan *attribute-value*.
- Hipotesis bersifat *disjunctive*.
- Memungkinkan adanya *noise* pada data.
- Pohon yang dibentuk oleh *decision tree* mudah dimengerti
- Mudah dikonversi menjadi sejumlah aturan produksi
- Dapat mengklasifikasikan baik data numerik maupun kategorikal, namun atribut akhir klasifikasi harus bersifat kategorikal.
- Tidak dibutuhkan asumsi terlebih dahulu pada data yang terlibat.
- Atribut hasil klasifikasi bernilai diskrit, dan tidak mendukung *multiple-attribute* pada hasil akhir klasifikasi.

- Algoritma *decision tree* bersifat *unstable*, dalam artian adanya sedikit variasi pada training data dapat menyebabkan berbedanya seleksi atribut pada setiap simpul pada pohon yang dibangun. Efek ini bisa menjadi signifikan karena pemilihan atribut berpengaruh pada subpohon dari simpul tersebut.
- Pohon yang dibangun dari dataset bertipe numerik bisa menjadi cukup kompleks karena pemilahan atribut pada tipe data numerik dilakukan melalui perbandingan biner.

### Rule-based Classifiers

*Rule-based classifier* melakukan klasifikasi terhadap dataset dengan membangun suatu aturan yang berlaku. Aturan tersebut direpresentasikan dalam bentuk klausa IF-THEN. Pendekatan yang dilakukan dalam membentuk klausa ini adalah berawal dari spesifik kemudian menggeneralisasi hipotesa tersebut hingga suatu aturan yang cukup akurat ditemukan. Proses pembentukan aturan dapat dilakukan berdasar statistik dari training data yang sesuai dengan *target function* [3].

Aturan yang dibentuk disadari jarang mencakup keseluruhan data yang sesuai dengan *target function*, sehingga pada iterasi berikutnya dibentuk lagi aturan baru yang terhubung dengan aturan lama secara *disjunctive*. Iterasi ini dilakukan terus hingga keseluruhan data yang sesuai dengan *target function* telah diwakili oleh aturan-aturan yang didapat.

Eksperimen ini menggunakan tiga algoritma yang masing-masing me-

wakili tiga jenis algoritma klasifikasi yang berbeda. Algoritma yang digunakan adalah Naive Bayes yang termasuk pada kategori *Bayes Classifiers*, C4.5 yang termasuk pada kategori *Decision Tree Classifiers*, dan OneR yang termasuk pada kategori *Rule-Based Classifiers*.

### Naive Bayes

Naive Bayes classifier bekerja berdasarkan aturan probabilitas pada Bayes classifier dan menerapkan konsep yang intuitif namun tetap dapat diperbandingkan. Bahkan pada beberapa kasus, dapat dilihat bahwa Naive Bayes lebih unggul dibanding algoritma lain yang lebih kompleks. Algoritma ini menggunakan keseluruhan atribut pada dataset yang diperbandingkan satu dengan yang lain tanpa adanya ketergantungan antar atributnya.

Sebagai contoh, pada training data terdapat beragam hewan, seperti gajah, kera, dan jerapah. Classifier harus mengklasifikasikan suatu instans baru yang ditemui. Secara umum telah diketahui bahwa gajah memiliki atribut seperti memiliki belalai, gading, berekor pendek, berukuran besar, dsb. Kera berukuran kecil, sering berlompatan, dan dapat memanjat pohon; sedangkan jerapah mempunyai leher yang tinggi, bertelinga pendek, dan memiliki badan yang tinggi.

Naive Bayes classifier akan mempertimbangkan setiap atribut itu secara terpisah ketika mengklasifikasikan suatu instans baru. Jadi, ketika memeriksa apakah suatu instans adalah seekor gajah, Naive Bayes classifier tidak akan memeriksa apakah hewan tersebut me-

memiliki belalai, dan memiliki gading, dan berukuran besar. Namun, classifier akan memisah secara terpisah apakah hewan tersebut memiliki belalai, apakah memiliki gading, apakah berukuran besar, dst. Hal ini berjalan sesuai asumsi bahwa suatu atribut berfungsi secara independen terhadap atribut lain yang terdapat pada *sample*.

Probabilitas pada Naïve Bayes classifier dapat dihitung menggunakan persamaan [5]:

$$Pr[H|E] = \frac{Pr[E_1|H] \cdot Pr[E_2|H] \cdot \dots \cdot Pr[E_n|H]}{Pr[E]}$$

dimana  $Pr[A]$  merupakan probabilitas terjadinya event  $A$ ,  $Pr[A|B]$  merupakan probabilitas terjadinya event  $A$  berdasar event  $B$ ,  $E_n$  adalah atribut ke  $n$  dari instans,  $H$  adalah output yang diharapkan, dan  $E$  adalah kombinasi dari keseluruhan atribut yang ada.

Beberapa keunggulan dan kelemahan dari Naïve Bayes classifier adalah sebagai berikut [5]:

- Menerapkan teknik yang sederhana namun menghasilkan output dengan tingkat akurasi yang cukup tinggi, apalagi ketika digabungkan dengan pendekatan lain.
- Memperlakukan setiap atribut secara independen dan sama penting, yang dapat menyebabkan hasil yang bergeser dari yang seharusnya, terutama ketika terdapat banyak atribut yang saling berkait, dimana kaitan tersebut akan mempunyai efek yang lebih pada output yang didapat, baik lebih baik maupun lebih buruk.
- Naïve Bayes tidak memfasilitasi output yang bersifat kategorikal.

## C4.5

Algoritma C4.5 merupakan perkembangan dari algoritma *decision tree original* (ID3) dengan adanya tambahan *pruning*. Algoritma ini dapat dilakukan dengan langkah sebagai berikut:

1. Pilih suatu atribut yang paling membedakan nilai dari atribut output.
2. Buat sebuah cabang pohon untuk setiap nilai yang mungkin dari atribut yang dipilih.
3. Pilah instans-instans kedalam beberapa kelompok sehingga merefleksikan nilai dari atribut pada atribut terpilih.
4. Untuk setiap kelompok, hentikan pemilihan atribut jika:
  - Setiap anggota kelompok mempunyai nilai yang sama dengan atribut output, hentikan pemilihan atribut pada path tersebut dan labeli cabang di path tersebut dengan nilai atribut output.
  - Kelompok tersebut mengandung sebuah simpul atau tidak dapat dilakukan pemilihan atribut pembeda pada langkah berikutnya. Labeli cabang tersebut dengan output yang paling dominan pada sisa instans yang ada.
5. Untuk setiap kelompok, yang belum dilabeli sebagai simpul terminal, ulangi langkah di atas.

Pada WEKA, algoritma ini diimplementasikan dengan nama yang berbeda yaitu J48.

## OneR (1R)

Algoritma OneR yang merupakan singkatan dari "One Rule" merupa-

kan algoritma klasifikasi sederhana yang menghasilkan tepat satu aturan saja. Karena tepat satu aturan maka algoritma ini juga bisa dikategorikan sebagai *decision tree* yang memiliki tepat satu level *decision tree*. Studi secara komprehensif terhadap performansi OneR menunjukkan bahwa algoritma ini dapat menghasilkan aturan yang sedikit kurang akurat jika dibandingkan algoritma - algoritma lain yang jauh lebih kompleks dan susah dimengerti oleh manusia. Algoritma OneR juga dapat menangani adanya *missing value* dan atribut bertipe numerik, sehingga selain sederhana juga mempunyai tingkat adaptasi yang tinggi.

Algoritma OneR berjalan dengan membuat satu rule untuk setiap atribut pada training data, kemudian memilih rule yang mempunyai error rate terkecil. Untuk membuat sebuah rule pada sebuah atribut, kelas dengan frekuensi terbesar pada tiap atribut harus ditentukan. Kelas dengan frekuensi terbesar inilah yang dijadikan acuan bahwa kelas tersebut bersesuaian dengan suatu nilai atribut tertentu. Sebuah rule adalah suatu himpunan nilai atribut yang dikaitkan dengan kelas yang mayoritas.

*Error rate* dari sebuah rule adalah banyaknya training data yang mempunyai kelas yang tidak sesuai dengan mayoritas. OneR memilih aturan yang memiliki nilai *error rate* terendah. Pada suatu saat ketika didapati dua atau lebih aturan yang memiliki nilai *error rate* terendah, maka aturan dipilih secara acak. Pada WEKA, pemilihan aturan tidak berdasarkan *error rate* terendah tapi berdasarkan tingkat akurasi yang tertinggi. Begitu juga

ketika didapati dua aturan atau lebih yang memiliki nilai *error rate* sama, WEKA tidak memilih secara acak tapi memilih aturan yang pertama kali ditemukan [6].

### Experiment Tools

Eksperimen ini menggunakan WEKA versi 3.4.13 sebagai alat bantu eksperimen. WEKA merupakan software yang dikhususkan pada bidang machine learning, diimplementasikan dalam bahasa Java, dikembangkan oleh Universitas Waikato, dan bersifat *free software* berdasar GNU GPL. Saat ini, WEKA merupakan tools yang paling populer pada bidang *machine learning* karena mempunyai keunggulan pada beberapa faktor [6]:

- Tersediasecara bebas berdasar GNU GPL.
- Bersifat portable karena diimplementasikan dalam bahasa Java sehingga dapat berjalan di hampir seluruh sistem operasi.
- Menyediakan koleksi algoritma *preprocessing, clustering, classification, regression, dan feature selection* secara komprehensif.
- Adanya GUI sehingga mudah digunakan bagi pengguna yang belum pernah menggunakan sebelumnya.

Keseluruhan fitur yang disediakan WEKA dapat diakses pada *WEKA Explorer* yang berkaitan dengan pemrosesan dataset tunggal dengan menggunakan satu algoritma tertentu pada suatu waktu. Selain itu, terdapat pula fasilitas pemrosesan sejumlah dataset dengan menggunakan beberapa algoritma

secara bersamaan yang dapat diakses melalui *WEKA Experimenter*. Dengan menggunakan *WEKA Experimenter*, performansi dari beberapa algoritma pada sekelompok dataset dapat dibandingkan dengan lebih mudah dan lebih sistematis. Penulis melakukan eksperimen menggunakan bantuan *WEKA Experimenter* ini.

## HASIL

### Dataset

Lima dataset yang bersumber dari UCI Machine Learning Repository dipilih berdasarkan lima karakteristik: jumlah instans pada dataset, jumlah atribut yang digunakan, banyaknya kelas yang digunakan pada proses klasifikasi, ada tidaknya *missing value* pada dataset tersebut, dan pola distribusi instans pada kelas-kelas yang dituju. Dari karakteristik tersebut, lima dataset yang dipilih adalah "Automobile Data Set", "Annealing Data Set", "Letter Recognition Data Set", "Vowel Recognition Data Set", dan "Congressional Voting Records Data Set".

**Autos** dataset merupakan dataset yang instansnya menggambarkan keamanan suatu mobil untuk dikendarai. Dataset ini mempunyai jumlah instans yang sedikit (205), jumlah atribut yang cukup banyak (26), mempunyai atribut dengan *missing value*, dan beragamnya atribut yang dapat digunakan sebagai kelas untuk proses klasifikasi. Terdapat beberapa atribut pada dataset ini yang dapat digunakan sebagai kelas dalam proses klasifikasi seperti *normalized-loss* (kemungkinan kehilangan) dan *symboling* (tingkat keamanan ber-

kendara). Dari beberapa atribut tersebut, klasifikasi berdasar keamanan dipilih karena distribusi kelas yang agak sedikit tidak merata (terdapat dua kelas terbesar yang ketika dijumlahkan jumlah instansnya melebihi 50%, yaitu kelas dengan *symboling* bernilai '0' dan '1' yang berarti tidak terlalu aman namun tidak pula terlalu beresiko).

**Anneal** dataset merupakan dataset yang instansnya menggambarkan proses penempaan baja. Dataset ini mempunyai jumlah instans yang cukup banyak (798), jumlah atribut yang banyak (38), mempunyai atribut dengan *missing value*, dan mempunyai banyak kelas (6). Pemilihan dataset ini lebih didasari karena tidak meratanya distribusi instans pada dataset ini (76% tergolong ke suatu kelas).

**Letter** dataset merupakan dataset yang instansnya digunakan untuk mengidentifikasi salah satu dari 26 huruf dalam alfabet. Huruf yang digunakan berupa huruf kapital yang di-*scan* dan diekstrak *feature*-nya. Dataset ini mempunyai jumlah instans sangat banyak (20000), jumlah atribut yang merepresentasikan *numeric feature* tiap huruf (16), tidak memiliki atribut yang *missing value* (ini dikarenakan atribut merupakan hasil ekstraksi *feature*), dan nilai kelas yang besar (26, sebanyak jumlah huruf pada alfabet). Pada dataset ini distribusi instans sangat merata dengan jumlah instans terbesar pada suatu kelas sebanyak 813 dan jumlah instans terkecil pada suatu kelas sebanyak 734. Pemilihan dataset ini didasari atas meratanya distribusi instans pada kelas walaupun jumlah instans cukup banyak.

**Vowel** dataset merupakan dataset yang instansnya menggambarkan pengenalan suatu huruf vokal dari suatu suku kata yang diucapkan oleh seseorang. Instans dibuat berdasarkan ekstraksi *feature* dari suara pembicara. Untuk menambah akurasi, setiap orang mengucapkan enam kali pada setiap suku kata. Dataset ini mempunyai jumlah instans yang sedang (528), jumlah atribut yang merupakan hasil ekstraksi *feature* dan jenis kelamin dari pembicara (10), tidak memiliki *missing value*, dan distribusi kelas yang sangat merata. Meratanya distribusi ini dikarenakan setiap orang mengucapkan suatu suku kata dengan jumlah yang sama. Dataset ini dipilih karena jumlah instans dalam kelas yang sama sehingga ingin diketahui sejauh mana pengaruhnya ketika proses training dan proses klasifikasi dijalankan.

**Vote** dataset merupakan dataset yang instansnya menggambarkan pemilihan partai pada pemilu yang diselenggarakan di Amerika Serikat. Partai yang dipilih hanya dua buah yaitu Partai Republik atau Demokrat. Atribut-atribut yang digunakan pada dataset ini kesemuanya ber-

sifat boolean yang hanya bernilai ya atau tidak saja. Pada beberapa instans, didapati atribut yang bernilai '?' yang berarti pada atribut tersebut tidak bernilai 'ya' dan tidak pula bernilai 'tidak'. Nilai seperti ini dianggap sebagai *missing value*.

Vote dataset mempunyai jumlah instans yang cukup sebesar 435 responden, dengan 267 termasuk Partai Demokrat dan 168 tergolong Partai Republik. Jumlah atribut pada dataset ini tidak terlalu banyak, hanya 16 atribut yang kesemuanya bertipe boolean. Kelas yang digunakan sebagai tujuan pengelompokan hanya berjumlah dua buah yang dapat dipandang sebagai boolean pula. Pada dataset ini dimungkinkan adanya *missing value*, dan pola distribusi kelas yang tidak berbeda jauh yaitu 45,2% terklasifikasi dalam Partai Demokrat dan 54,8% masuk pada Partai Republik. Dataset ini dipilih karena tipenya yang unik yaitu keseluruhan atributnya bahkan kelas klasifikasinya bertipe boolean.

Rangkuman keseluruhan informasi dari dataset yang digunakan pada eksperimen ini dapat dilihat pada Tabel 1.

**Tabel 1. Informasi dataset dari lima dataset yang dipilih**

	Number of Instances	Number of Attributes	Number of Class Values	Class Distribution	Missing Values
Autos	205	26	7	Agak Tidak Merata	Ya
Anneal	798	38	6	Tidak Merata	Ya
Letter	20000	16	26	Merata	Tidak
Vowel	528	10	11	Merata	Tidak
Vote	435	16	2	Merata	Ya

## PEMBAHASAN

Ketiga algoritma yang ditentukan dijalankan pada kelima dataset yang dipilih menggunakan *WEKA*

*Experimenter* sebagai alat bantu. Eksperimen dijalankan dengan menggunakan 66,6% dataset sebagai *training data* dan sisanya sebagai *data test*. Perbandingan

antara tiap algoritma menggunakan *two-tail* test dengan confidence level sebesar 95% yang telah terimplementasi *WEKA Experimenter*.

Pada Tabel 2 dapat dilihat bahwa tingkat akurasi yang didapat oleh algoritma C4.5 yang diimplementasikan dalam bentuk J48 oleh WEKA mempunyai tingkat akurasi yang cukup signifikan pada kelima dataset yang digunakan, kemudian algoritma Naive Bayes yang unggul sedikit jika dibandingkan algoritma OneR. Untuk dataset Vote, algoritma OneR memiliki tingkat akurasi yang tinggi, yang sebanding dengan algoritma C4.5. Hal ini didapat karena pada dataset Vote, keseluruhan atribut bertipe boolean sehingga proses klasifikasi dataset dengan berdasar satu aturan cukup untuk kasus tersebut.

Berbeda dengan tingkat akurasi pada dataset Letter, dimana didapati bahwa algoritma OneR memiliki akurasi yang sangat rendah (17,19%). Hal ini dikarenakan jumlah kelas tujuan klasifikasi sangat banyak (26) sehingga besar kemungkinan bahwa penggunaan satu aturan saja tidaklah cukup.

**Tabel 2 Persentase Akurasi Klasifikasi**

	bayes. NaiveBayes	trees. J48	rules. OneR
Autos	55.92	72.94	53.60
Anneal	87.57	98.33	83.46
Letter	64.07	86.32	17.19
Vowel	61.36	76.12	32.99
Vote	89.04	95.13	95.47

Selain akurasi dari algoritma, ingin dibandingkan pula lama proses training yang dibutuhkan pada tiap algoritma untuk setiap dataset. Pada Tabel 3 dapat dilihat bahwa proses training terlama adalah untuk dataset Letter yang melibat-

kan 20.000 training data. Hal ini berarti waktu yang diperlukan untuk proses training sebanding dengan banyaknya training data yang digunakan.

Selain lama proses training, dari Tabel 3 dapat dilihat pula bahwa secara rata-rata algoritma C4.5 membutuhkan waktu yang jauh lebih lama dibandingkan kedua algoritma yang lain. Hal ini dikarenakan proses yang terlibat dalam algoritma C4.5 cukup kompleks jika dibandingkan dengan algoritma lain. Meski demikian jika dikombinasikan dengan hasil pada Tabel 2 dapat dilihat bahwa waktu training yang lama sebanding dengan tingkat akurasi klasifikasi.

**Tabel 3. Lama Proses Training Data**

	bayes. NaiveBayes	trees. J48	rules. OneR
Autos	0.00	0.01	0.00
Anneal	0.01	0.04	0.00
Letter	0.36	7.71	0.33
Vowel	0.00	0.12	0.00
Vote	0.00	0.00	0.00

## KESIMPULAN

Dari ketiga algoritma yang dibandingkan, diketahui bahwa algoritma C4.5 merupakan algoritma yang memiliki tingkat akurasi yang tinggi independen terhadap dataset yang digunakan. Algoritma OneR akan mempunyai tingkat akurasi yang tinggi jika banyaknya nilai dari kelas klasifikasi cukup rendah. Hal ini tercermin dari hasil eksperimen pada dataset Letter dan Vote, dimana Letter memiliki jumlah nilai kelas terbanyak (26) dan Vote memiliki jumlah nilai kelas paling sedikit (2).

**DAFTAR RUJUKAN**

Robert C. Holte. 1993. *Very Simple Classification Rules Perform Well on Most Commonly Used Datasets*. Ottawa, Canada.

Michie, D. et al. (1994). *Machine learning, neural and statistical classification*. New York: Ellis Horwood.

Mitchell, Tom. (1997). *Machine Learning*. McGraw-Hill.

Wikipedia, Inc.

[http://en.wikipedia.org/wiki/Weka\\_\(machine\\_learning\)](http://en.wikipedia.org/wiki/Weka_(machine_learning)).

Algorithm for Unsupervised Learning: Naïve Bayes Rule Generator.

[http://grb.mnsu.edu/grbts/doc/manual/Naive\\_Bayes.html](http://grb.mnsu.edu/grbts/doc/manual/Naive_Bayes.html)

Gaya Buddhinath, Damien Derry. 2003. *A Simple Enhancement to One Rule Classification*. Melbourne, Australia.