Menentukan pengklasifikasi terbaik untuk memprediksi nilai bidang Boolean pada database donor darah

Ritabrata Maiti[1]

[1]Universitas Teknologi Delhi

Abstrak

Motivasi: Berkat digitalisasi, kita sering memiliki akses ke database besar, yang terdiri dari berbagai bidang informasi, mulai dari angka hingga teks dan bahkan nilai boolean.
Basis data semacam itu sangat cocok untuk pembelajaran mesin, klasifikasi, dan tugas analisis data besar. Kami dapat melatih pengklasifikasi, menggunakan data yang sudah ada dan menggunakannya untuk memprediksi nilai bidang tertentu, mengingat kami memiliki informasi mengenai bidang lain.

Secara khusus, dalam penelitian ini, kami melihat Electronic Health Records (EHRs) yang disusun oleh rumah sakit. EHR ini adalah cara yang mudah untuk mengakses data pasien individu, tetapi pemrosesan di sana secara keseluruhan masih tetap menjadi tugas. Namun, EHR yang terdiri dari struktur yang koheren dan ditabulasi dengan baik cocok untuk aplikasi bahasa mesin, melalui penggunaan pengklasifikasi. Dalam penelitian ini, kami melihat Kumpulan Data Pusat Layanan Transfusi Darah (Data diambil dari Pusat Layanan Transfusi Darah di Kota Hsin-Chu di Taiwan). Kami menggunakan pembelajaran mesin scikit-learn dengan python. Dari Support Vector Machines (SVM), kami menggunakan Support Vector Classification (SVC), dari model linier kami mengimpor Perceptron. Kami juga menggunakan K.neighborsclassifier dan pengklasifikasi pohon keputusan. Lebih-lebih lagi, kami menggunakan perpustakaan TPOT untuk menemukan saluran yang dioptimalkan menggunakan algoritme genetika. Dengan menggunakan pengklasifikasi di atas, kami menilai masing-masing dari mereka menggunakan validasi silang k kali lipat.

Hasil: Program pengujian bergantung pada pengujian individual dari pengklasifikasi. Ini menghitung jumlah prediksi yang banyak nilai sebenarnya dan menampilkan hitungan tersebut. Dengan menggunakan penghitungan, kami dapat memutuskan pengklasifikasi terbaik untuk database donor darah yang diberikan. Dengan menggunakan model yang paling akurat, atau kumpulan model ini, kami akan dapat menentukan prediksi yang paling akurat untuk setiap pasien. Di sini, kami ingin menentukan apakah seorang pasien telah mendonorkan darah pada Maret 2017. Prediksi ini adalah nilai boolean (1 atau 0), menunjukkan bahwa pasien telah mendonorkan darah dan 0 menunjukkan sebaliknya.

Kontak: ritabratamaiti@hiretrex.comhttps://github.com/

GitHub: ritabratamaiti/Blooddonorprediction

1. Perkenalan

Dalam 20 tahun terakhir, kemampuan penyimpanan media elektronik telah meningkat secara eksponensial. Akibatnya, volume data medis yang disimpan di media elektronik meningkat secara eksponensial. Berbagai data medis yang tersedia bagi kami berkisar dari gambar dan teks hingga video dan audio. Ini adalah salah satu dari sedikit tipe data yang tersedia dan dimanipulasi di pusat kesehatan. Seringkali data yang diperlukan dieksploitasi dan dianalisis pada tingkat individu. Misalnya, Magnetic Resonance Imaging (MRI) dan catatan kesehatan tekstual akan dianalisis untuk menetapkan diagnosis atau evolusi penyakit pasien.

Di antara semua tipe data yang berbeda, hanya data terstruktur yang dapat langsung digunakan untuk melatih algoritme pembelajaran mesin. Ini karena data yang ditabulasi cocok untuk pelatihan pengklasifikasi bahasa mesin. Melalui pengklasifikasi, kami dapat bekerja pada kumpulan data, dan memprediksi nilai yang ada di satu bidang kumpulan yang diberikan, asalkan kami memiliki informasi mengenai bidang kumpulan data lainnya. Algoritme pembelajaran mesin ini berfungsi dengan merasakan pola yang ada dalam kumpulan data yang ada dan menggunakan pola tersebut untuk memprediksi nilai bidang yang hilang.

Namun, masalah yang muncul adalah sebagian besar catatan dalam database medis adalah teks bebas yang tidak terstruktur, seperti grafik kesehatan pasien dan laporan pasien elektronik. Ini menggunakan bahasa alami, yaitu bahasa mudah dipahami dan diproses oleh manusia tetapi kurang begitu oleh mesin. Contohnya termasuk catatan tentang diagnosis pasien, resep atau bahkan catatan mengenai sampel jaringan. Sementara teks bebas dapat dibuat mesin dimengerti menggunakan algoritma NLP, rekonstruksi pola yang paling efektif dan prediksi dalam kedokteran dibuat menggunakan database terstruktur dan algoritma pengklasifikasi.

Inti teknis dari program ini bergantung pada fakta bahwa algoritma pelatihan yang berbeda memiliki tingkat keberhasilan yang berbeda ketika dilatih dengan dataset yang sama. Akibatnya, kita akan dapat menentukan pengklasifikasi mana yang bekerja paling baik dengan algoritma yang diberikan.

Sementara diagnosis dokter tetap sangat berharga dalam kedokteran, dengan ilmu data, diagnosis dan prediksi penyakit menjadi jauh lebih andal dan efisien. Menggunakan algoritma sebagai penolong dalam prediksi penyakit, deteksi dini dan analisis tidak hanya membantu dokter dalam analisis penyakit tetapi juga membantu dalam menurunkan angka kematian pasien.

Dalam masalah ini kita akan mengklasifikasikan apakah seorang pendonor telah mendonorkan darahnya pada waktu tertentu (dalam hal ini bulan Maret 2007), berdasarkan parameter:

- Kekinian donor darah sebelumnya (jumlah bulan sejak donor terakhir)
- Frekuensi donor darah
- Jumlah darah yang disumbangkan dalam sentimeter kubik (cc)
- Waktu dalam bulan sejak donasi pertama)

Kami juga akan menggunakan algoritme genetika untuk melakukan penyetelan hyperparameter pada pengklasifikasi, dan membandingkan skor akurasi pengklasifikasi ini dengan pengklasifikasi default.

2. Metode

Klasifikasi dapat dianggap sebagai dua masalah yang terpisah – klasifikasi biner dan klasifikasi multikelas. Dalam klasifikasi biner, tugas yang lebih mudah dipahami, hanya dua kelas yang terlibat, sedangkan klasifikasi multikelas melibatkan penetapan objek ke salah satu dari beberapa kelas.

Dalam klasifikasi biner, kita mengelompokkan objek ke dalam salah satu dari dua kelas, sedangkan dalam klasifikasi multiclass, kita mengelompokkan objek ke dalam salah satu dari banyak kelas. Masalah khusus kami mengharuskan kami untuk memutuskan apakah pasien tertentu telah menyumbangkan darah pada tanggal sebelumnya, dan sebagai hasilnya, pengklasifikasi kami memberi kami output boolean baik 1 atau 0, di mana 1 menunjukkan bahwa pasien telah menyumbangkan darah dan 0 menunjukkan sebaliknya.

Jadi, kita harus menyelesaikan masalah klasifikasi biner. Untuk mengetahui classifier yang paling efisien, kita harus melatih setiap classifier dengan dataset yang sama. Kemudian, kita harus memprediksi nilai 1 atau 0, untuk kumpulan data pasien baru, dan menentukan apakah prediksi tersebut cocok.

Dengan mengetahui jumlah prediksi yang cocok, kita dapat menghitung persentase kecocokan dari setiap database dan selanjutnya menentukan pengklasifikasi yang paling efisien.

Dalam pembelajaran mesin dan statistik, klasifikasi adalah masalah mengidentifikasi yang mana dari satu set kategori (sub-populasi) pengamatan baru milik, atas dasar satu set data pelatihan yang berisi pengamatan (atau contoh) yang keanggotaan kategorinya diketahui. . Contohnya adalah menetapkan email yang diberikan ke kelas "spam" atau "non-spam" atau menetapkan diagnosis kepada pasien tertentu seperti yang dijelaskan oleh karakteristik pasien yang diamati (jenis kelamin, tekanan darah, ada atau tidak adanya gejala tertentu, dll.). Klasifikasi adalah contoh pengenalan pola.

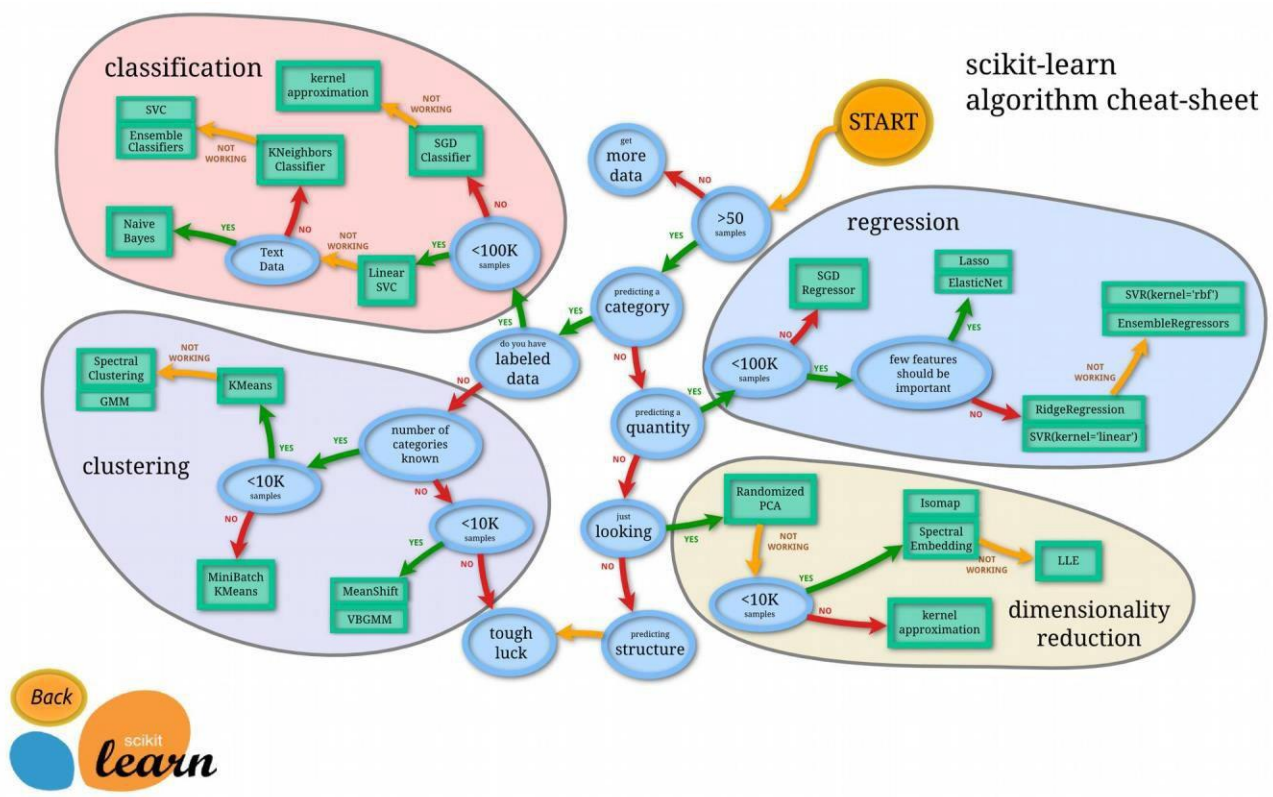
Dalam terminologi pembelajaran mesin,[1] klasifikasi dianggap sebagai contoh pembelajaran terawasi, yaitu pembelajaran di mana satu set pelatihan pengamatan yang diidentifikasi dengan benar tersedia. Prosedur tanpa pengawasan yang sesuai dikenal sebagai pengelompokan dan melibatkan pengelompokan data ke dalam kategori berdasarkan beberapa ukuran kesamaan atau jarak yang melekat.

Seringkali, pengamatan individu dianalisis ke dalam satu set sifat terukur, yang dikenal sebagai variabel atau fitur penjelas. Sifat-sifat ini bisa bermacam-macam kategori (misalnya "A", "B", "AB" atau "O", untuk golongan darah), ordinal (misalnya "besar", "sedang" atau "kecil"), bernilai integer (misalnya jumlah kemunculan kata tertentu dalam email) atau bernilai nyata (misalnya pengukuran tekanan darah). Pengklasifikasi lain bekerja dengan membandingkan pengamatan dengan pengamatan sebelumnya melalui fungsi kesamaan atau jarak.

Sebuah algoritma yang mengimplementasikan klasifikasi, terutama dalam implementasi konkret, dikenal sebagai classifier. Istilah "pengklasifikasi" kadang-kadang juga mengacu pada fungsi matematika, yang diterapkan oleh algoritma klasifikasi, yang memetakan data masukan ke suatu kategori.

Terminologi lintas bidang cukup bervariasi. Dalam statistik, di mana klasifikasi sering dilakukan dengan regresi logistik atau prosedur serupa, sifat-sifat pengamatan disebut variabel penjelas (atau variabel bebas, regresi, dll.), dan kategori yang diprediksi dikenal sebagai hasil, yang dianggap menjadi nilai-nilai yang mungkin dari variabel terikat. Dalam pembelajaran mesin, pengamatan sering dikenal sebagai instance, variabel penjelas disebut fitur (dikelompokkan ke dalam vektor fitur), dan kategori yang mungkin untuk diprediksi adalah kelas. Bidang lain mungkin menggunakan terminologi yang berbeda: misalnya dalam ekologi komunitas, istilah "klasifikasi" biasanya mengacu pada analisis klaster, yaitu jenis pembelajaran tanpa pengawasan, daripada pembelajaran terawasi yang dijelaskan dalam artikel ini.

2.1 Seleksi Pengklasifikasi



Melalui penggunaan diagram pemilihan classifier yang diberikan di sini: Kami kemudian mengikuti langkah-langkahnya:

- > 50 sampel : Ya
- Memprediksi kategori: Ya
- Data Berlabel : Ya
- Oleh karena itu masalah klasifikasi
- <100k sampel: Ya

Jadi kami telah memilih pengklasifikasi berikut:

1. SVC
2. Perceptron
3. KneighborsClassifier
4. Pengklasifikasi Pohon Keputusan

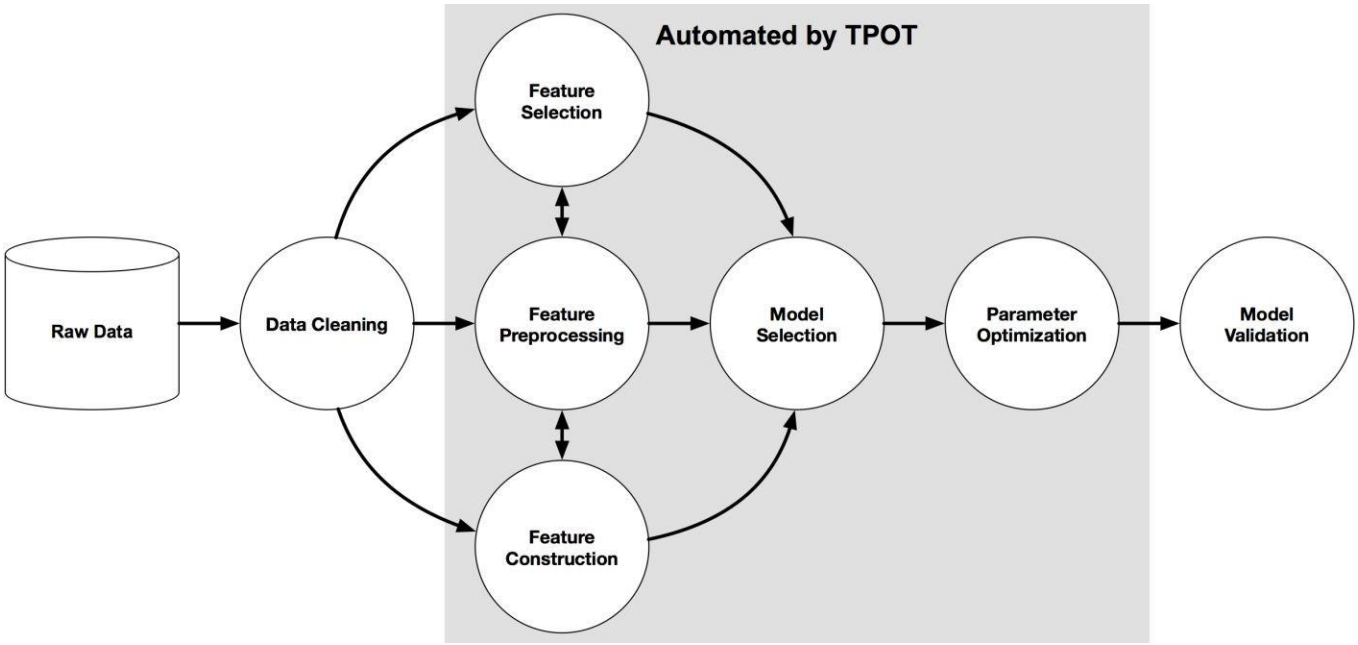
Selanjutnya, kami menyertakan Pengklasifikasi Nave Bayes dalam pengujian kami, karena akurasinya yang tinggi dalam klasifikasi biner. Kami juga akan melatih pengklasifikasi dari perpustakaan TPOT untuk memilih pengklasifikasi terbaik, sehubungan dengan akurasi, dan juga melakukan penyetelan hyperparameter pada pengklasifikasi tersebut, dan menemukan pipa terbaik. Kami kemudian akan mengevaluasi masing-masing pengklasifikasi melalui validasi silang k-fold dan menentukan pengklasifikasi terbaik berdasarkan skor rata-rata terbaik.

Dengan demikian, daftar terakhir dari classifier yang digunakan adalah:

- SVC
- Perceptron
- KneighborsClassifier
- Pengklasifikasi Pohon Keputusan
- Pengklasifikasi Naïve Bayes
- Pengklasifikasi TPOT

Kami akan membatasi diskusi kami pada cara kerja pengklasifikasi TPOT, karena pengklasifikasi lainnya adalah algoritma standar, dan implementasinya telah didokumentasikan dan dibahas secara ketat di Scikit-learn Paper (lihat referensi).

Ini adalah aliran otomatisasi perpustakaan TPOT. Itu dibangun di atas Scikit-learn dan melakukan penyetelan hyperparameter pada regressor dan classifier Scikit-learn.



Pengklasifikasi TPOT akan bekerja dengan ribuan saluran, dan kemudian merekomendasikan saluran yang bekerja paling baik untuk data yang diberikan. Ini menerapkan algoritme genetik pada populasi awal saluran pipa untuk memilih saluran pipa yang paling sesuai, mereproduksi saluran pipa generasi baru dan mengulangi proses ini selama beberapa generasi. Proses ini biasanya konvergen ke jalur pipa terbaik, kecuali jika secara eksplisit dihentikan oleh seorang programmer.

3. Hasil

Kami memperoleh skor akurasi berikut untuk pengklasifikasi (perhatikan bahwa nilai output ini diambil langsung dari konsol python).

1
0.7859531772575251
0,6923076923076923
0,6337792642140468
0,7178631051752922
0.6961602671118531
Rata-rata = 0.7052127012132818

DecisionTreeClassifier(class_weight=Tidak ada, kriteria='gini',
max_depth=Tidak ada,
max_features=Tidak ada, max_leaf_nodes=Tidak ada,
min_impurity_decrease=0,0, min_impurity_split=Tidak ada,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0,0, presort=Salah,
random_state=Tidak ada,
pemisah = 'terbaik')

2
0.81438127090301
0,7558528428093646
0.7274247491638796
0.7328881469115192
0.7245409015025042
Rata-rata = 0,7510175822580555

SVC(C=1.0, cache_size=200, class_weight=Tidak ada, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf', max_iter=-1,
probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

3
C:\Users\Ritabrata Maiti\Anaconda3\lib\sitepackages\sklearn\linear_model\stochastic_gradient.py:128: Mata Depan Peringatan:
parameter max_iter dan tol telah ditambahkan di <class 'sklearn.linear_model.perceptron.Perceptron'> di 0.19. Jika keduanya tidak disetel, defaultnya adalah max_iter=5 dan tol=None. Jika tol bukan None, max_iter default ke max_iter=1000. Dari 0.21, max_iter default akan menjadi 1000, dan tol default adalah 1e-3.
"dan tol default adalah 1e-3." % type(self), FutureWarning)
0.81438127090301
0.24414715719063546

0.7290969899665551
0.21535893155258765
0.7262103505843072
Rata-rata = 0,5458389400394191

Perceptron(alpha=0,0001, class_weight=None, eta0=1,0, fit_intercept=Benar,
max_iter=Tidak ada, n_iter=Tidak ada, n_jobs=1, penalti=Tidak ada, random_state=0,
shuffle=Benar, tol=Tidak ada, verbose=0, warm_start=Salah)

4
0,7474916387959866
0.7408026755852842
0,725752508361204
0.657762938230384
0.7262103505843072
Rata-rata = 0,7196040223114333

KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
metric_params=Tidak ada, n_jobs=1, n_neighbors=5, p=2,
weights='uniform')

5
0.81438127090301
0,7558528428093646
0.7290969899665551
0,7813021702838063
0.7262103505843072
Rata-rata = 0,7613687249094087

BernoulliNB(alpha=1.0, binarize=0.0, class_prior=Tidak ada, fit_prior=True)

6
0.8294314381270903
0,7675585284280937
0.7591973244147158
0.7929883138564274
0.7262103505843072
Rata-rata = 0,7750771910821269

Pipa(memori=Tidak ada,
langkah=[('linearsvc', LinearSVC(C=5.0, class_weight=Tidak ada, dual=False,
fit_intercept=Benar,
intersep_scaling=1, loss='squared_hinge', max_iter=1000, multi_class='ovr',
penalti='l2', random_state=None, tol=0.001, verbose=0))])

Kami mengamati bahwa pipeline yang dioptimalkan TPOT memiliki skor rata-rata terbaik diikuti oleh Naïve Bayes Classifier. Dalam hal ini pipa yang dioptimalkan TPOT kebetulan adalah pengklasifikasi LinearSVC dengan hyperparameter yang disetel.
(**Catatan**: Kami telah menjalankan metode optimasi fit classifier TPOT hanya untuk 5 generasi. Skor akurasi yang lebih baik dapat dicapai dengan meningkatkan jumlah generasi)

Sebagai kesimpulan, kami dapat mencatat bahwa jalur pipa yang dioptimalkan secara genetik bekerja paling baik dalam tugas klasifikasi ini. Namun, pengklasifikasi Naïve Bayes memberikan kinerja yang serupa, pada uji validasi silang k-fold.

4. Ucapan Terima Kasih

Sumber Kumpulan Data:
(Pemilik Asli dan
Donor) Prof. I-Cheng
Departemen Yeh
Informasi
Manajemen Chung-Hua
Universitas, Hsin Chu, Taiwan
30067, ROC
email: es '@'
chu.edu.tw TEL:886-3-
5186511
Tanggal Disumbangkan: 3 Oktober 2008

5. Referensi

1. Har-Peled, S., Roth, D., Zimak, D. (2003) "Klasifikasi Kendala untuk Klasifikasi dan Pemeringkatan Multiklas." Dalam: Becker, B., Thrun, S., Obermayer, K. (Eds) Kemajuan dalam Sistem Pemrosesan Informasi Saraf 15: Prosiding Konferensi 2002, MIT Press. ISBN 0-262-02550-7)

2. Scikit-belajar: Pembelajaran Mesin dengan Python, Pedregosa et al., JMLR 12, hlm. 2825-2830, 2011.

3. Yeh, I-Cheng, Yang, King-Jang, and Ting, Tao-Ming, "Penemuan pengetahuan pada model RFM menggunakan deret Bernoulli, "Sistem Pakar dengan Aplikasi, 2008

4. Rosenblatt, Frank. x. Prinsip Neurodinamika: Perceptrons dan Teori Mekanisme Otak. Spartan Books, Washington DC, 1961

5. Rao, CR (1952) Metode Statistik Lanjutan dalam Analisis Multivariat, Wiley. (Bagian 9c)

6. Anderson, TW (1958) Sebuah Pengantar Analisis Statistik Multivariat, Wiley.

7. Binder, DA (1978) "analisis cluster Bayesian", Biometrika, 65, 31-38.

8. Binder, DA (1981) "Perkiraan untuk aturan pengelompokan Bayesian", Biometrika, 68, 275–285.

9. Har-Peled, S., Roth, D., Zimak, D. (2003) "Klasifikasi Kendala untuk Klasifikasi dan Pemeringkatan Multiklas."

10. Evaluasi Alat Pengoptimalan Pipeline Berbasis Pohon untuk Mengotomatiskan Ilmu Data Randal S. Olson University of Pennsylvania olsonran@upenn.edu Nathan Bartley University of Chicago bartleyn@uchicago.edu Ryan J. Urbanowicz University of Pennsylvania ryanurb@upenn.edu Jason H. Moore University of Pennsylvania jhmoore@upenn.edu