p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

Model *Decision Tree* untuk Prediksi Jadwal Kerja menggunakan *Scikit-Learn*

Retnani Latifah*[1], Emi Setia Wulandari[1] dan Priadhana Edi Kreshna[1]
[1]Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jakarta,
Kampus B Jl Cempaka Putih Tengah 27, 10510
**Corresponding Author* : retnani.latifah@ftumj.ac.id

Abstrak

Data yang memiliki tipe kategorikal dan numerikal seperti data jadwal kerja memiliki tantangan tersendiri untuk dapat dilakukan prediksi karena data tipe kategorikal perlu perhatian khusus. Decision tree merupakan salah satu algoritma yang dapat digunakan untuk prediksi dan dapat menggunakan data kategorikal maupun data numerikal. Salah satu *library* yang dapat digunakan untuk menerapkan decision tree adalah scikit-learn, yang berjalan di python. Scikit-learn menerapkan optimasi dari algoritma CART dan meskipun hanya dapat mengolah data numerikal, *library* ini menyediakan fitur untuk menangani data kategorikal. Penelitian ini membuat model *decision tree* menggunakan scikit-learn untuk membuat model prediksi jadwal kerja. Data yang digunakan berjumlah 54 data dengan 3 variabel kategorikal dan 1 variabel numerikal. Dari hasil penerapan diperoleh sebuah model *decision tree* dengan kedalaman pohon adalah 6. Hasil evaluasinya menunjukkan hasil yang baik yaitu nilai akurasi mencapai di atas 0,7 dan presisi mencapai di atas 0,9. Persentase pemisahan data terbaik adalah dengan menggunakan 30% data uji dan 70% data latih. Saat dibandingkan, model *decision tree* memiliki akurasi yang lebih baik dibandingkan KNN, di mana akurasi *decision tree* dapat mencapai angka di atas 0,8 sedangkan KNN dibawah itu.

Kata kunci: decision tree, klasifikasi, scikit-learn, python

Abstract

Predicting data with categorical and numerical type, such as working schedule data, is quite a challenge since it need certain process. Decision tree is one of many algorithms to use as classification and can handle categorical and numerical data. One library that can be used for decision tree is scikit learn, which runs in python. Scikit-learn implemented an optimized CART algorithm and although it could only handle numerical data, but it facilitates some features to deal with categorical. This study built a decision tree model in scikit-learn to predict working schedule. There are 54 data with 3 categorical variables and 1 numerical variable. From the implementation, a 6-depth decision tree model has been built. The evaluation showed a good result, with accuracy up to above 0.7 and precision up to above 0.9. The best splitted data is 30% validation set and 70% training set. The decision tree model has better accuracy compare to KNN, where decision tree accuracy up to above 0.8 while KNN is under.

Keywords : *decision tree, classification, scikit-learn, python*

PENDAHULUAN

Salah satu metode *supervised machine learning* non parametrik yang digunakan untuk klasifikasi atau regresi adalah metode *decision tree*. Algoritma ini menghasilkan suatu model yang dapat memprediksi kategori data dengan cara mempelajari aturan penentuan kategori berdasarkan fitur-fitur yang dimiliki oleh data (Ceballos, 2019)(Ochiai, Masuma, & Tomii, 2019). Berdasarkan tipe kategori datanya, decision tree dibedakan menjadi dua jenis yaitu *classification tree* dan *regression tree*. *Classification tree* memiliki kategori berupa data diskrit berhingga, sedangkan *regression tree* memiliki kategori berupa data diskrit berhingga atau data kontinu (Topîrceanu & Grosseck, 2017).

Metode *decision tree*, baik *classification tree* maupun *regression tree*, memiliki banyak variasi yang telah dikembangkan oleh peneliti-

p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

peneliti sebelumnya. Beberapa diantaranya adalah algoritma C45, CART, CHAID, CRUISE, GUIDE, QUEST, dan M5 (Loh, 2011). Selain pengembangan dari algoritmanya itu sendiri, saat ini juga telah dikembangkan *library* atau *software package* untuk melakukan penerapan *decision tree* secara lebih mudah. Diantaranya adalah *library machine learning scikit-learn* (Pedregosa et al., 2011) di python.

Algoritma CART (Classification and Regression Trees) membangun pohon binary menggunakan fitur dan *threshold* dengan nilai information gain terbesar di setiap *node*. Algoritma ini mirip dengan algoritma C4.5 akan tetapi algoritma ini dapat melakukan regresi dan tidak melakukan komputasi aturan-aturan. Algoritma CART yang digunakan di scikit-learn merupakan versi optimasi dan belum dapat menggunakan data kategorikal sehingga data kategorikal perlu dirubah menjadi numerikal terlebih dahulu (Pedregosa et al., 2019).

Studi yang dilakukan pada penelitian ini adalah melakukan penerapan *decision tree* menggunakan *library scikit-learn*. Data yang digunakan terdiri dari data numerik dan data kategorikal. Dari model *decision tree* yang dihasilkan, akan didapatkan sebuah *decision tree* yang dapat digunakan untuk memprediksi jadwal kerja. Penelitian mengenai penentuan jadwal kerja telah dilakukan sebelumnya oleh Achmad dan Slamat (2012).

Penelitian tersebut (Achamad & Slamat, 2012) menggunakan algoritma *decision tree* C.45 untuk membangun model prediksi jadwal kerja karyawan. Variabel yang digunakan yaitu umur, jenis kelamin, tingkat pendidikan, agama, level, lama kerja dan keadaan kesehatan. Sedangkan jenis data dari masing-masing variabel adalah data numerikal atau kategorikal. Data yang digunakan sebagai label kelas adalah data jadwal, yang mana memiliki 2 jenis jadwal yaitu jadwal A dan jadwal B. Penelitian tersebut menghasilkan sebuah aplikasi yang dapat menentukan jadwal kerja karyawan secara otomatis dengan menginput nilai dari fitur-fitur yang dibutuhkan. Akurasi dari metode *decision tree* C4.5 yang diterapkan adalah sebesar 87%.

METODE

Data yang digunakan pada penelitian ini memiliki 4 variabel berupa data numerik dan kategorikal. Variabel-variabel yang digunakan yaitu jenis kelamin, umur, golongan pekerjaan, dan status kepegawaian. Sedangkan label kelasnya adalah berdasarkan jadwal kerja, yaitu pagi dan malam. Data yang digunakan berjumlah 54 data yang diambil dari salah satu perusahaan yang ada di Jakarta.

Terdapat 3 variabel kategorikal yaitu jenis kelamin, golongan pekerjaan dan status kepegawaian. *Decision tree* yang ada di scikit-learn hanya mampu memproses data numerik sehingga data-data kategorikal perlu dilakukan preprocessing terlebih dahulu. Variabel-variabel tersebut perlu dilakukan *preprocessing* sebelum dapat digunakan untuk membuat model *decision tree*. *Preprocessing* yang digunakan adalah menggunakan modul LabelEncoder di *scikit-learn*. Modul ini memberi label kelas pada data kategorikal, yaitu dari 0 sampai n kelas. Hasil pelabelan menggunakan LabelEncoder (Vaish, 2017) dapat dilihat pada tabel 1.

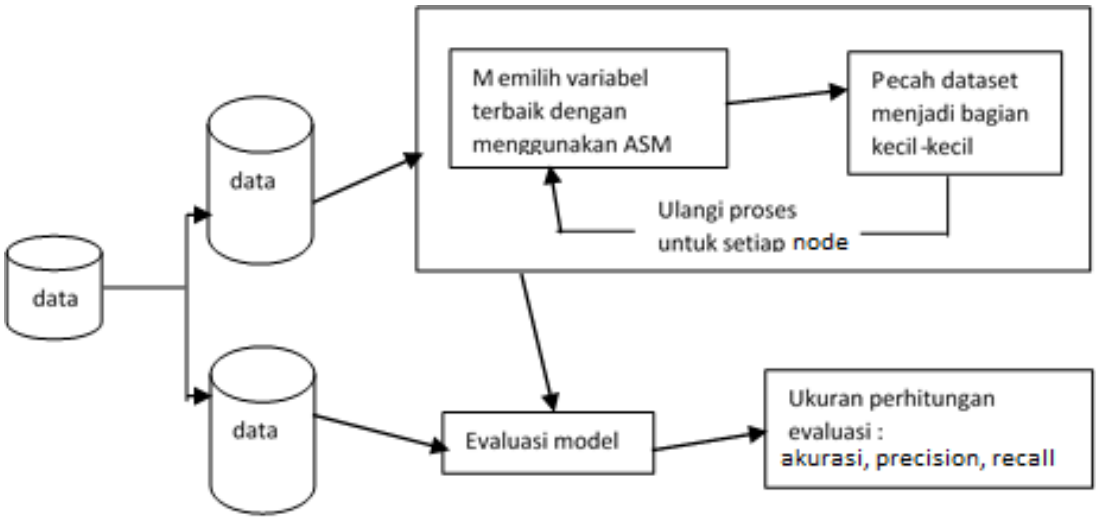
Tabel 1. Hasil *Preprocessing* Variabel Kategorikal

Variabel	**Sebelum**	**Setelah**
Jenis Kelamin	Pria	0
	Wanita	1
Golongan Pekerjaan	1	0
	2	1
	4	2
	5	3
Status Kepegawaian	Kontrak	0
	Tetap	1

Setelah semua variabel berbentuk data numerik, maka data akan diolah dengan *decision tree* untuk mendapatkan model prediksi jadwal kerja. Sesuai dengan namanya, *decision tree* membangun model klasifikasi dengan bentuk struktur pohon yang mana setiap *node* dari pohon diasosikan dengan atribut dari data, dalam hal ini adalah variabel data (Ochiai et al., 2019). Bagian bawah dari *node* yang ada di *decision tree* atau yang biasa disebut sebagai daun (*leaves*), pada kasus optimal, memiliki nilai yang merupakan karakteristik dari label kelas (Aguilar-chinea et al., 2019). *Decision tree* merupakan metode yang mudah dipahami dan diinterpretasikan karena visualisasinya seperti sebuah *flowchart* yang mana dengan mudah ditiru oleh kemampuan berpikir manusia

p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

(Navlani, 2018). Langkah-langkah *decision tree* dapat dilihat pada gambar 1.



Gambar 1. Langkah-Langkah Decision Tree (Navlani, 2018)

Pada gambar 1, terdapat istilah ASM yang merupakan singkatan dari Atribute Selection Measure yaitu salah satu cara untuk memilih kriteria pemisah untuk mengelompokkan data sebaik mungkin (Navlani, 2018). ASM yang digunakan di algoritma CART, yang merupakan algoritma yang diterapkan di scikit-learn *decision tree*, adalah Gini Index. Alat ukur ini mengukur *purity* dari setiap *node*, dimana jika nilainya lebih dari nol maka menunjukkan bahwa ada sampel yang termasuk di kelas yang lain (Ceballos, 2019). Rumus umum gini index dapat dilihat pada *equation* 1 (Navlani, 2018).

$$Gini\ (D) = 1 - \sum_{i=1}^{m} p_i^2 \qquad (1)$$

p_i adalah peluang suatu pasangan data di D berada pada kelas Ci

Setelah model *decision tree* didapatkan, maka selanjutnya adalah melakukan evaluasi model. Evaluasi yang pertama dilakukan adalah pengujian dengan berbagai persentase data latih dan data uji untuk mengetahui berapa persentase yang paling baik untuk membangun model *decision tree*. Dari 54 data yang tersedia, dilakukan pemecahan data menjadi data latih dan data uji secara acak.

Untuk setiap persentase data latih dan data uji, dilakukan perhitungan akurasi, *precision*, *recall* dan F1 measure. Akurasi adalah perbandingan hasil klasifikasi yang sesuai dengan semua hasil klasifikasi. *Recall* adalah seberapa berhasil algoritma mengenali suatu kelas, sedangkan *precision* adalah seberapa tepat hasil klasifikasi dari keseluruhan data dan F1-measure merupakan gabungan *recall* dan *precision* yang mana mewakili keseluruhan kinerja metode (Ridok & Latifah, 2015). Nilai precision, recall dan F1 measure yang akan digunakan pada pengujian ini adalah nilai *macroaverage*.

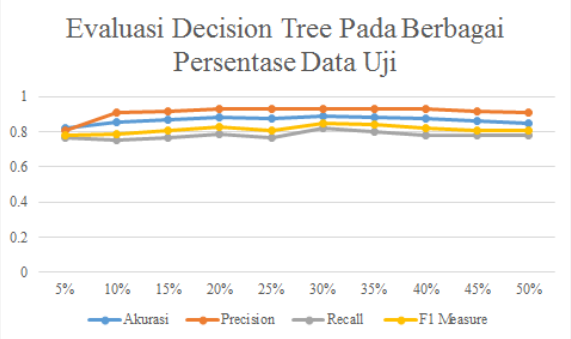
Evaluasi kedua adalah membandingkan *decision tree* dengan KNN, yang merupakan salah satu metode klasifikasi dengan kinerja yang baik. Evaluasi yang dibandingkan adalah nilai akurasi. Pada evaluasi ini tidak dilakukan setting parameter KNN sehingga setting yang digunakan adalah *default setting* dari modul KNN pada scikit-learn library.

HASIL DAN PEMBAHASAN

Pengujian dengan berbagai persentase data latih dan data uji dilakukan dengan 10 variasi yaitu menggunakan persentase data uji 5%, 10%, 15% dan seterusnya sampai 50%. Pada pengujian ini tidak dilakukan setting nilai max depth dan nilai minimum sample leafHasil pengujian dapat dilihat pada gambar 2. Dari gambar tersebut terlihat bahwa semua nilai evaluasi di atas 0,7 yang menunjukkan bahwa kinerja metode decision tree dengan persentase data uji terbaik adalah 30% data uji dan 70% data latih. Meskipun perbedaan nilai evaluasi untuk setiap persentase data uji tidak berbeda terlalu jauh. Nilai akurasi, precision, recall dan

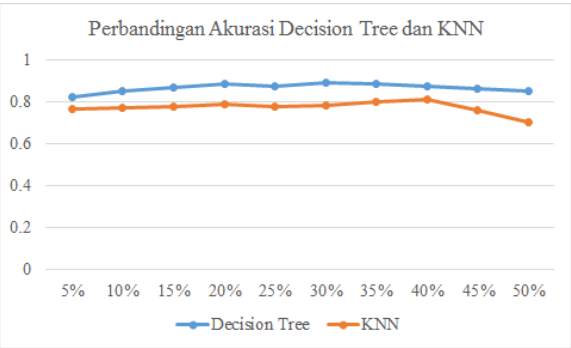
p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

F1-measure dengan data uji 30% adalah 0,892, 0,93, 0,82 dan 0,85. Nilai-nilai evaluasi tersebut konsisten dan menunjukkan bahwa decision tree sudah bekerja dengan baik.



Gambar 2. Evaluasi Decision Tree Pada Berbagai Persentase Data Uji

Perbandingan akurasi *decision tree* dengan KNN dapat dilihat pada gambar 3, di mana terlihat bahwa *decision tree* memiliki kinerja yang lebih baik dibandingkan dengan KNN. Decision tree memiliki nilai akurasi di atas 0,8 sedangkan KNN memiliki nilai akurasi di bawah 0,8. Hal ini bisa dijelaskan karena *decision tree* dapat menemukan hubungan dan interaksi non-linear pada data (Floares, Calin, & Manolache, 2016). Oleh karena itu *decision tree* memiliki kinerja yang lebih baik karena KNN cenderung mencari kemiripan antar data.



Gambar 3. Perbandingan Akurasi Decision Tree dan KNN

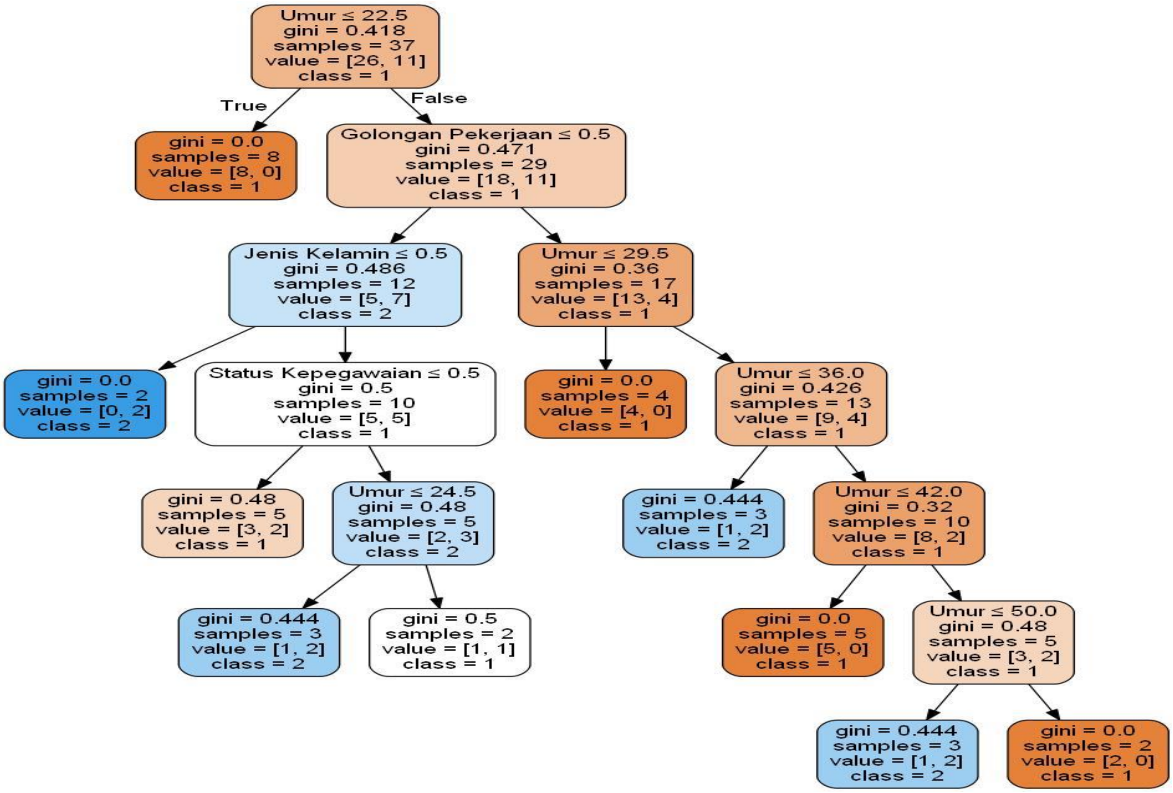
Model prediksi *decision tree* tanpa *setting* khusus dan data uji sebanyak 30% dapat dilihat pada gambar 4. Gambar tersebut adalah hasil *generate* dari *tools* graphviz (Ellson, Gansner, Koutsofios, North, & Woodhull, 2001). Kedalaman (*depth*) pohon adalah 6, di mana *root* berada pada *depth* ke-0 dan dua *leaves* paling bawah berada pada *depth* ke-6. Minimum sampel yang digunakan pada setiap *node* adalah 2.

Pada *root* terlihat bahwa kelas yang dipilih adalah kelas 1 yaitu jadwal pagi. Gini index *root* = 0.418, menunjukkan bahwa ada sampel yang termasuk ke kelas 2 yaitu jadwal malam. Jumlah sampel yang digunakan di *root* adalah 37 dengan 26 sampel termasuk kelas 1 dan 11 sampel termasuk kelas 2. Variabel umur ≤ 22,5 dipilih karena setelah dilakukan pemisahan, fitur tersebut memiliki nilai *purity* yang paling kecil.

Pada *depth* ke-1, di cabang *true*, terlihat bahwa nilai gini index adalah 0 yang menunjukkan bahwa delapan sampel memiliki kelas yang sama yaitu 1. Sisanya berada pada cabang *false*. Variabel yang digunakan untuk memisahkan node adalah variabel golongan pekerjaan. Pada *depth* ke -2, variabel golongan pekerjaan dipecah kembali menjadi dua cabang yaitu jenis kelamin dan umur ≤ 29,5. Pemecahan menjadi cabang dilakukan secara terus-menerus sampai semua data digunakan.

Pada *decision tree* dengan kategorikal, *node* biasanya ditulis dalam bentuk "apakah jenis kelamin adalah laki-laki". Sedangkan pada model *decision tree* yang diperoleh dengan menggunakan *scikit-learn*, kategorikal dianggap seperti numerikal sehingga node berbentuk "apakah jenis kelamin ≤ 0.5". Hal ini dikarenakan pada proses preprocessing menggunakan LabelEncoder, data kategorikal dirubah menjadi numerikal dan diproses seperti layaknya data kontinu di decision tree. Label Encoder merubah kategorikal menjadi angka terurut sehingga kurang tepat untuk digunakan terutama jika data bukan ordinal. Hasil evaluasi decision tree menunjukkan nilai akurasi dan presisi yang cukup tinggi yaitu di atas 0,7 dan di atas 0,9 karena data latih dan data uji di-*encoding* pada saat bersamaan. *Decision tree* akan mengalami kesulitan untuk memprediksi data yang diinput terpisah dan masih berupa data kategorikal.

p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek



Gambar 4. Model *Decision Tree* Untuk Prediksi Jadwal Kerja

SIMPULAN DAN SARAN

Dari 54 data jadwal kerja yang memiliki variabel numerikal dan kategorikal, diperoleh sebuah model decision tree untuk memprediksi jadwal kerja. Model dibuat menggunakan library scikit-learn sehingga data kategorikal perlu dilakukan preprocessing menggunakan modul Label Encoder pada scikit-learn. Model decision tree memiliki nilai evaluasi yang cukup tinggi yaitu akurasi di atas 0,7 dan presisi di atas 0,9. Hasil pengujian dengan berbagai variasi persentase data uji menunjukkan bahwa hasil evaluasi tidak berbeda jauh, namun data uji 30% memiliki nilai evaluasi tertinggi. Evaluasi decision tree lebih baik jika dibandingkan dengan evaluasi KNN yaitu decision tree memiliki akurasi di atas 0,8 sedangkan akurasi KNN dibawahnya. Pengubahan data kategorikal menjadi data numerikal yang dilakukan oleh scikit-learn kurang tepat dilakukan karena data kategorikal dianggap sebagai angka terurut.

UCAPAN TERIMAKASIH

Penelitian ini didukung dan dibiayai oleh Fakultas Teknik Universitas Muhammadiyah Jakarta.

DAFTAR PUSTAKA

Achamad, B. D. M., & Slamat, F. (2012). Klasifikasi Data Karyawan Untuk Menentukan Jadwal Kerja Menggunakan Metode Decision Tree. *Jurnal IPTEK*, *16*(1), 17–23.

Aguilar-chinea, R. M., Castilla, I., Expósito, C., Aguilar-chinea, R. M., Castilla, I., Moreno-vega, J. M., & Moreno-vega, J. M. (2019). Using a decision tree algorithm to predict the robustness of a transshipment schedule. *Procedia Computer Science*, *149*, 529–536. https://doi.org/10.1016/j.procs.2019.01.172

Ceballos, F. (2019). Scikit-Learn Decision Trees Explained - Training, Visualizing, and Making Predictions with Decision Trees. Retrieved from https://towardsdatascience.com/scikit-learn-decision-trees-explained-

803f3812290d

Ellson, J., Gansner, E., Koutsofios, L., North, S., & Woodhull, G. (2001). Graphviz - Open Source Graph Drawing Tools. In *Lecture Notes in Computer Science* (pp. 483–484). Springer-Verlag. Retrieved from https://graphviz.gitlab.io/

Floares, A. G., Calin, G. A., & Manolache, F. B. (2016). Bigger Data is Better for Molecular Diagnosis Tests Based on Decision Trees. In *In: Tan Y., Shi Y. (eds) Data Mining and Big Data. DMBD 2016. Lecture Notes in Computer Science, vol 9714* (pp. 288–295). Springer, Cham. https://doi.org/https://doi.org/10.1007/978-3-319-40973-3_29

Loh, W. (2011). Classification and Regression Trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, (January 2011). https://doi.org/10.1002/widm.8

Navlani, A. (2018). Decision Tree Classification in Python. Retrieved from https://www.datacamp.com/community/tutorials/decision-tree-classification-python

Ochiai, Y., Masuma, Y., & Tomii, N. (2019). Improvement of timetable robustness by analysis of drivers' operation based on decision trees. *Journal of Rail Transport Planning & Management*, *9*(March), 57–65. https://doi.org/10.1016/j.jrtpm.2019.03.001

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, *12*, 2825--2830.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2019). Decision Tree. Retrieved from https://scikit-learn.org/stable/modules/tree.html

Ridok, A., & Latifah, R. (2015). Klasifikasi Teks Bahasa Indonesia Pada Corpus Tak Seimbang Menggunakan NWKNN. In *Konferensi Nasional Sistem & Informatika 2015* (pp. 9–10).

Topîrceanu, A., & Grosseck, G. (2017). Decision tree learning used for the classification of student archetypes in online courses. *Procedia Computer Science*, *112*, 51–60. https://doi.org/10.1016/j.procs.2017.08.021

Vaish, P. (2017). Decision Trees in scikit-learn. Retrieved August 14, 2019, from https://adataanalyst.com/scikit-learn/decision-trees-scikit-learn/