Prediksi Kualitas Wine Menggunakan Metode *Decision Tree*

Putra Dwi Prasetyo (G64190037), Falih Alwana Yasril (G64190067), Laudza Muhammad Afin Tachtiar (G64190052), Antonius Anre Sianturi (G64190053), Abdul Hakim (G64190078), Muhammad Reyhan (G64190083)

Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor

Abstraksi

Wine merupakan minuman sari anggur merah yang difermentasi [12]. Setidaknya 234 juta hektoliter *wine* dikonsumsi di seluruh dunia pada tahun 2020 [13]. Karena banyaknya konsumsi dari *wine*, sertifikasi kualitas *wine* menjadi penting. Pada saat sertifikasi biasanya dilakukan analisis kandungan yang terdapat didalam *wine*. Cara yang dapat membantu dalam menganalisis kandungan yang terdapat didalam wine untuk menentukan kualitas adalah dengan menggunakan implementasi pada sistem cerdas. Pada tugas akhir ini, sistem cerdas dibangun menggunakan algoritma decision tree. Dari hasil penelitian kami diperoleh nilai akurasi terhadap data tes sebesar 0.83 dan akurasi terhadap data latih sebesar 0.93298.

1. Pendahuluan

Wine merupakan minuman sari anggur merah yang difermentasi [12]. *Wine* sudah menjadi minuman sehari hari untuk masyarakat pada daerah tertentu. Setidaknya 234 juta hektoliter *wine* dikonsumsi di seluruh dunia pada tahun 2020 [13].

Karena banyaknya konsumsi dari *wine*, sertifikasi kualitas wine menjadi penting. Hal tersebut diperlukan untuk mencegah adanya pemalsuan yang dapat berakibat pada kesehatan dari peminum *wine* tersebut. Terlepas kontroversinya, dampak buruk dari *wine* dapat diminimalisir dengan mengidentifikasi kualitas dari *wine*. Hal ini dapat dilakukan dengan menganalisis kandungan yang terdapat didalam *wine*.

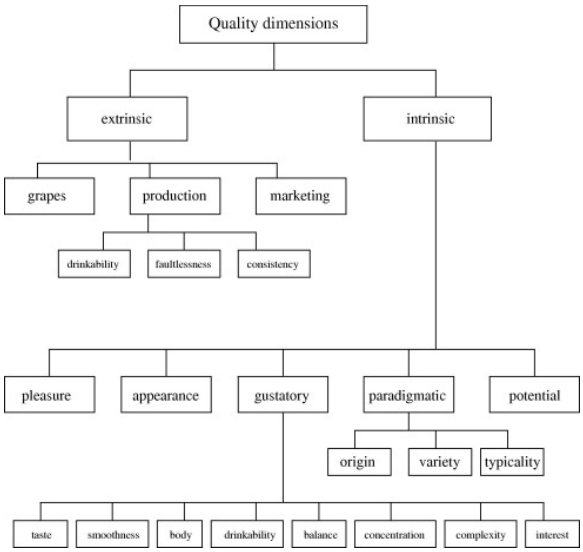
Analisis kandungan yang terdapat didalam wine biasanya telah dilakukan saat *wine* akan disertifikasi. Sertifikasi *wine* biasanya dilakukan dengan tes fisikokimia dan sensorik [4]. Tes fisikokimia digunakan untuk mengidentifikasi karakter dari wine seperti kepadatan, kandungan nilai alkohol dan pH, sedangkan tes sensorik dilakukan oleh ahli yang memahami karakteristik tertentu dari *wine*.

Cara lain yang dapat membantu dalam menganalisis kandungan yang terdapat didalam *wine* untuk menentukan kualitas adalah dengan menggunakan implementasi pada sistem cerdas. Pada tugas akhir ini, sistem cerdas dibangun menggunakan algoritma *Decision Tree*. Dengan adanya prediksi menggunakan *Decision Tree* diharapkan dapat membantu masyarakat peminum *wine* dalam mengidentifikasi *wine* yang akan diminum.

2. Tinjauan Pustaka

Wine adalah minuman tradisional beralkohol dengan kepentingan komersial tinggi diperoleh dari fermentasi keseluruhan bagian anggur yang dihancurkan menjadi sebuah jus. Berdasarkan definisi tersebut, kualitas wine terkait dengan komposisi dan juga variasi anggur [1]. *Wine* bisa diklasifikasikan berdasarkan merah, putih, *rose* berdasarkan tingkat kemanisan, kadar alkohol, kadar karbon dioksida, warna, jenis anggur, fermentasi, dan proses fermentasinya [7]. *Red wines* didapatkan dari fermentasi alkohol dari bagian keras anggur (kulit dan biji), *white wine* diproduksi dari fermentasi jus anggur [9]. Pernah dianggap sebagai komoditi mewah, saat ini *wine* sudah dapat dikonsumsi secara lebih luas [3]. Dalam rangka meningkatkan pertumbuhan, industri *wine* investasi ke teknologi *wine* dalam proses membuat dan menjual. Sertifikasi *wine* dan penilaian kualitas adalah kunci dalam konteks ini. Sertifikasi mencegah pemalsuan ilegal *wine* (untuk menjaga kesehatan manusia) dan menjamin kualitas untuk pasar *wine*. Evaluasi dari kualitas merupakan bagian dari proses sertifikasi dan dapat digunakan untuk meningkatkan produksi *wine* (dengan mengidentifikasi faktor paling berpengaruh) dan mengelompokkan anggur seperti brand premium. Sertifikasi *wine* umumnya dinilai dengan tes fisikokimia dan tes sensorik [4].

Selain itu, para konsumen *wine* memiliki pertimbangan terhadap kualitas *wine* dengan berbagai komponen [2].



Klasifikasi merupakan metode supervised learning. Prediksi dan klasifikasi dalam data mining adalah dua bentuk analisis data yang digunakan untuk mengekstrak model yang menggambarkan kelas data atau untuk memprediksi tren data di masa depan. Proses klasifikasi memiliki dua fase; yang pertama adalah proses pembelajaran dimana dataset training dianalisis dengan algoritma klasifikasi. Model atau pengklasifikasi yang dipelajari disajikan dalam bentuk aturan atau pola klasifikasi. Tahap kedua adalah penggunaan model untuk klasifikasi, dan kumpulan data uji digunakan untuk memperkirakan keakuratan aturan klasifikasi [10].

Decision tree adalah teknik untuk representasi data secara hierarkis. *Decision tree* menggunakan metode percabangan untuk menggambarkan setiap kemungkinan hasil dari suatu masalah. Struktur *tree* menunjukkan bagaimana satu pilihan mengarah ke yang berikutnya, dan penggunaan cabang menunjukkan bahwa setiap pilihan saling eksklusif. Sebuah pohon keputusan dapat digunakan untuk mengklasifikasi dan menemukan jawaban masalah yang kompleks. *Decision tree* dapat bekerja pada semua jenis dataset dan dapat menangani informasi kondisional dengan membagi dataset menjadi subkelompok. Subkelompok ini selanjutnya dianggap sebagai kumpulan data individu untuk pemrosesan dalam metodologi *decision tree* [5].

Information Gain didasarkan pada *Entropy*. *Information Gain* adalah perbedaan antara *entropy* kelas dan *entropy* bersyarat dari kelas dan fitur yang dipilih. Ia mengukur kegunaan fitur dalam klasifikasi. Dengan kata lain, mengukur pengurangan ketidakpastian setelah pemisahan set pada fitur. Jika nilai *information gain* meningkat, berarti fitur tersebut lebih berguna untuk klasifikasi. Fitur dengan *information gain* tertinggi adalah fitur terbaik yang dipilih untuk dipisah [11].

$$Entropy(S) = -\sum_{i=1}^{n} p_i * \log_2 p_i$$

Keterangan:
S : Himpunan kasus
n : Jumlah partisi S
pi : Proporsi Si terhadap S

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

Keterangan:
S : Himpunan kasus
A : Atribut
n : Jumlah partisi atribut (A)
|Si| : Jumlah kasus pada partisi ke-i
|S| : Jumlah kasus dalam S

Indeks GINI menentukan kemurnian kelas tertentu setelah memisahkan atribut tertentu. Pemisahan terbaik meningkatkan kemurnian set yang dihasilkan dari pemisahan [11].

$$Gini(S) = 1 - \sum_{i=1}^{n} P_i^2$$

Keterangan:
pi : Frekuensi kelas i relatif di S

3. Metode

Metode yang digunakan dalam penelitian ini adalah *decision Tree* untuk memprediksi kualitas *wine* dari faktor-faktornya menggunakan data yang diambil dari situs Kaggle. Terdapat sebelas faktor yang dimiliki oleh data *wine* yaitu, *fixed acidity*, *volatile acidity*, *citric acid*, *residual sugar*, *chloride*, *free sulfur dioxide*, *total sulfur dioxide*, *density*, *pH*, *sulphates*, dan *alcohol*. Kelas dari *quality* hanya ada enam, yaitu satu hingga delapan dari skala nol hingga sepuluh. Perhitungan dan konstruksi *decision tree* akan dilakukan menggunakan Python3.

4. Pembahasan

Pada bagian ini akan dibahas penggunaan Decision Tree untuk memprediksi kualitas dari Red Wine

menggunakan beberapa atribut yang telah disebutkan pada bagian sebelumnya. Data set yang digunakan merupakan data publik yang didapatkan dari Kaggle dengan alamat https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009.

Tujuan dari pengolahan data ini adalah memprediksi kualitas dari Red Wine berdasarkan hasil uji fisikokimia dari wine tersebut. Dataset yang kami gunakan berisi sebanyak 1599 record dengan atribut sebanyak 12 dengan kondisi data tidak ada yang kosong dan bertipe numerik. Data yang ada akan diolah menggunakan bahasa pemrograman Python dengan lingkungan google collab.

Berikut dilampirkan 5 data pertama dari dataset yang digunakan.

```
[ ] wine.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Pertama, akan dilakukan import beberapa library yang akan digunakan. Terdapat 4 library yang akan digunakan, yaitu numpy, pandas, sklearn, dan matplotlib.pyplot. Lalu akan dipisahkan fitur-fitur yang akan menjadi peubah tak bebas dan peubah bebas. Data yang memiliki label quality akan dipisahkan dari data yang lain dengan fungsi drop() untuk digunakan sebagai peubah tak bebas. Sisa data dengan label lainnya akan digunakan sebagai peubah bebas.

Kemudian data yang ada akan dibagi menjadi data training dan data testing dengan menggunakan train_test_split dari library sklearn. Pada kasus ini kami akan menggunakan 30% dari data yang ada sebagai data testing dan sebanyak 70% sisanya dari data yang ada akan menjadi data training.

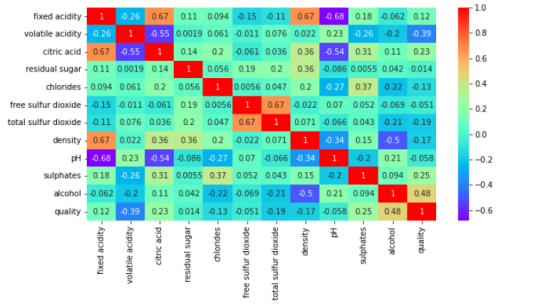
```
[ ] x = wine.drop(columns='quality')
    y = wine['quality']

    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=1)
```

Dari data training dan data testing tadi akan dibuat model klasifikasinya. Pada tahap ini “gini” akan ditetapkan sebagai parameter kriteria classifier. Hasil prediksi dari model yang didapatkan memiliki tingkat akurasi sebesar 0.59791667. Kami merasa bahwa tingkat akurasi yang dihasilkan oleh model kurang memuaskan, sehingga akan dilakukan optimalisasi berupa feature selection dan parameter tuning agar dapat meningkatkan tingkat akurasi model.

```
[ ] clf = DecisionTreeClassifier(criterion="gini")
    clf = clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)

    print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

    Accuracy: 0.5979166666666667
```

Proses feature selection yang digunakan pada kasus ini ada dua. Correlation dan SelectKBest. Pada proses feature selection berdasarkan nilai korelasinya, pertama-tama nilai korelasi antar fitur akan diperiksa terlebih dahulu.

```
[ ] plt.figure(figsize = (10,5))
    sns.heatmap(wine.corr(), annot = True, cmap="rainbow")
    plt.show()
```



Dari sebaran korelasi yang didapatkan, dengan menggunakan nilai threshold dari rentang -0.5 hingga 0.5 kami memutuskan untuk mengeliminasi fitur citric acid, total sulfur dioxide, dan density karena ketiga fitur tersebut memiliki nilai korelasi melebihi batas toleransi threshold. Setelah itu akan dilakukan pemodelan ulang dengan Classifier menggunakan gini sebagai parameter kriteria. Hasil akurasi yang didapatkan dari model baru setelah dilakukan feature selection berdasarkan korelasi meningkat menjadi 0.63125.

```
feature_corr = ['fixed acidity', 'volatile acidity', 'residual sugar', 'chlorides',
                'free sulfur dioxide', 'pH', 'sulphates', 'alcohol']

x = wine[feature_corr]
y = wine['quality']

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=1)

clf = DecisionTreeClassifier(criterion="gini")
clf = clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

Accuracy: 0.63125
```

Setelah menggunakan feature selection berdasarkan korelasi, akan dicoba metode lain yaitu SelectKBest. Fungsi SelectKBest merupakan sebuah metode untuk memilih fitur terbaik sebanyak K fitur dari seluruh fitur. Pada proses yang kami jalankan fitur yang akan dipilih ini berdasarkan parameter score function f_classif. Dari semua kombinasi fitur yang ada, kombinasi dari lima fitur yakni volatile acidity, citric acid, total sulfur dioxide, sulphates, dan alcohol memiliki nilai akurasi paling tinggi jika digunakan sebagai peubah bebas, yakni 0.6125.

```
[ ] from sklearn.datasets import make_classification
    from sklearn.feature_selection import SelectKBest
    from sklearn.feature_selection import f_classif

    X = wine[feature]
    y = wine['quality']

    for i in range(1, 12):
      # define feature selection
      fs = SelectKBest(score_func=f_classif, k=i)

      # apply feature selection
      X_selected = fs.fit_transform(X, y)
      print(fs.get_support(indices=True))

      X_train, X_test, y_train, y_test = train_test_split(X_selected, y, test_size=0.30, random_state=1)

      clf = DecisionTreeClassifier(criterion="entropy")
      clf = clf.fit(X_train,y_train)
      y_pred = clf.predict(X_test)

      print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Dari potongan kode SelectKBest di atas didapat *output* berikut.

```
[10]
Accuracy: 0.56875
[ 1 10]
Accuracy: 0.5583333333333333
[ 1  6 10]
Accuracy: 0.58125
[ 1  6  9 10]
Accuracy: 0.59375
[ 1  2  6  9 10]
Accuracy: 0.6125
[ 1  2  6  7  9 10]
Accuracy: 0.60625
[ 0  1  2  6  7  9 10]
Accuracy: 0.5958333333333333
[ 0  1  2  4  6  7  9 10]
Accuracy: 0.5791666666666667
[ 0  1  2  4  5  6  7  9 10]
Accuracy: 0.5708333333333333
[ 0  1  2  4  5  6  7  8  9 10]
Accuracy: 0.5791666666666667
[ 0  1  2  3  4  5  6  7  8  9 10]
Accuracy: 0.5895833333333333
```

Jika hasil optimalisasi selection feature berdasarkan nilai korelasi dibandingkan dengan fungsi SelectKBest, selection feature berdasarkan korelasi memiliki nilai akurasi yang lebih tinggi. Sehingga model yang selanjutnya akan digunakan adalah model setelah dilakukan optimalisasi berdasarkan nilai korelasi. Namun ketika dilakukan pemeriksaan terhadap skor akurasi train, terdapat indikasi overfitting pada model karena skor bernilai 1.0. Sehingga parameter tuning akan dilakukan pada langkah selanjutnya.

```
[96] print("Test Accuracy :",metrics.accuracy_score(y_test, y_pred))
     print("Train Accuracy:",metrics.accuracy_score(y_train, clf.predict(X_train)))

     Test Accuracy : 0.63125
     Train Accuracy: 1.0
```

Parameter tuning berguna untuk mencari parameter-parameter yang diperlukan untuk dimasukan ke dalam decision tree agar mendapatkan model yang optimal. Fungsi GridSearchCV yang merupakan fungsi untuk menentukan parameter tadi menghasilkan parameter ccp_alpha = 0.01, criterion = “entropy”, max_depth = 10, max_features = “auto”, dan random_state = 1.

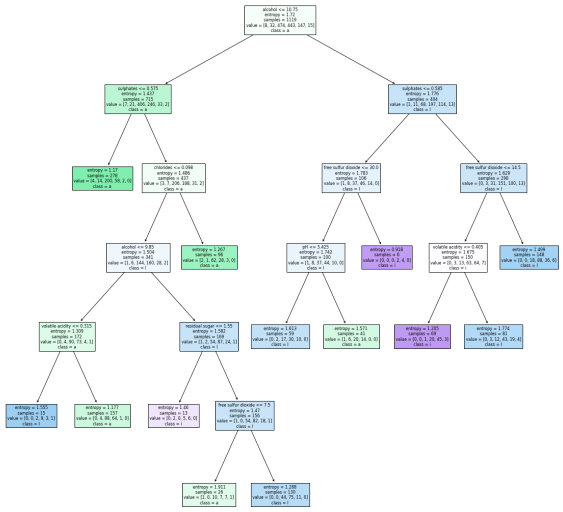
```
[ ] from sklearn.model_selection import GridSearchCV

    x = wine[feature_corr]
    y = wine['quality']

    X_train, X_test, y_train, y_test = train_test_split(x, y,
                                                        test_size=0.30,
                                                        random_state=1)

    param_grid = {'max_features': ['auto', 'sqrt', 'log2'],
                  'ccp_alpha': [0.1, .01, .001],
                  'max_depth' : [2, 3, 4, 5, 6, 7, 8, 9, 10],
                  'splitter' : ['best', 'random'],
                  'criterion' :['gini', 'entropy']
                 }
    tree_clas = DecisionTreeClassifier(random_state=1)
    grid_search = GridSearchCV(estimator=tree_clas, param_grid=param_grid, cv=5, verbose=True)
    grid_search.fit(X_train, y_train)

    final_model = grid_search.best_estimator_
    final_model

    Fitting 5 folds for each of 324 candidates, totalling 1620 fits
    DecisionTreeClassifier(ccp_alpha=0.01, criterion='entropy', max_depth=10,
                           max_features='auto', random_state=1)
```

Dengan menggunakan parameter baru yang didapatkan dari fungsi GridSearchCV, diperoleh model baru dengan tingkat akurasi dataset tes sebesar 0.55833 dan data set train sebesar 0.60768. Indikasi overfitting menghilang setelah parameter untuk decision tree diubah.

```
[26] x = wine[feature_corr]
     y = wine['quality']

     X_train, X_test, y_train, y_test = train_test_split(x, y,
                                                         test_size=0.30,
                                                         random_state=1)

     clf = DecisionTreeClassifier(ccp_alpha=0.01, criterion='entropy', max_depth=10,
                                  max_features='auto', random_state=1)
     clf = clf.fit(X_train,y_train)
     y_pred = clf.predict(X_test)
```

```
[27] print("Test Accuracy:",metrics.accuracy_score(y_test, y_pred))
     print("Train Accuracy:",metrics.accuracy_score(y_train, clf.predict(X_train)))

     Test Accuracy: 0.5583333333333333
     Train Accuracy: 0.6076854334226989
```

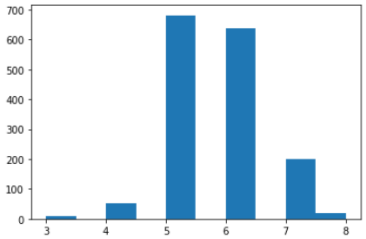
Setelah usaha untuk mengoptimalkan model dilakukan, diperoleh sebuah tree seperti pada gambar berikut.



Berdasarkan hasil akurasi yang diperoleh dengan langkah-langkah sebelumnya, kami berinisiatif untuk menyederhanakan kategori kualitas wine dari yang semula terdiri dari 10 kategori dengan skala 0 hingga 10 menjadi hanya 3 kategori, yakni ‘buruk’, ‘sedang’, dan ‘baik’ yang direpresentasikan dengan skala 0 hingga 2. Transformasi kategori yang dilakukan adalah mengubah nilai [1, 2, 3, 4] menjadi [0], [5, 6] menjadi [1], dan [7, 8, 9, 10] menjadi [2]. Hal tersebut didasarkan pada penilaian dasar orang awam terhadap kualitas *wine*.

```
[ ] y.value_counts().sort_index()

    3     10
    4     53
    5    681
    6    638
    7    199
    8     18
    Name: quality, dtype: int64
```

```
[ ] plt.hist(y)

    (array([ 10.,   0.,  53.,   0., 681.,   0., 638.,   0., 199.,  18.]),
     array([3. , 3.5, 4. , 4.5, 5. , 5.5, 6. , 6.5, 7. , 7.5, 8. ]),
     <a list of 10 Patch objects>)
```



```
[ ] quality_target = np.unique(wine.quality)
    quality_change = np.array([0, 0, 1, 1, 2, 2])
```

Pada blok terakhir potongan kode di atas, terlihat perubahan kategori kualitas wine.

Langkah selanjutnya adalah melakukan parameter tuning dengan menggunakan GridSearchCV seperti yang telah dilakukan pada dataset yang fitur kualitas wine-nya belum ditransformasi.

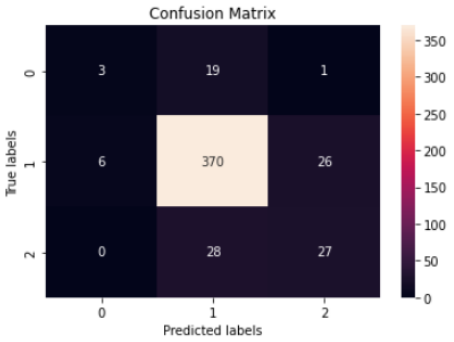
```
[ ] X_train, X_test, y_train, y_test = train_test_split(x, y,
                                                        test_size=0.30,
                                                        random_state=1)

    param_grid = {'max_features': ['auto', 'sqrt', 'log2'],
                  'ccp_alpha': [0.1, .01, .001],
                  'max_depth' : [5, 6, 7, 8, 9, 10],
                  'splitter' : ['best', 'random'],
                  'criterion' :['gini', 'entropy']
                 }
    tree_clas = DecisionTreeClassifier(random_state=1)
    grid_search = GridSearchCV(estimator=tree_clas, param_grid=param_grid, cv=5, verbose=True)
    grid_search.fit(X_train, y_train)

    final_model = grid_search.best_estimator_
    final_model

    Fitting 5 folds for each of 216 candidates, totalling 1080 fits
    DecisionTreeClassifier(ccp_alpha=0.001, max_depth=8, max_features='log2',
                           random_state=1)
```

Terlihat bahwa hasil parameter tuning menggunakan GridSearchCV adalah nilai-nilai parameter fungsi Decision Tree yang dianggap akan mengoptimalkan model, yaitu ccp_alpha=0.001, max_depth=8, max_features=’log2’, random_state=1, dan selebihnya menggunakan nilai default. Setelah dilakukan parameter tuning pada model yang telah dilakukan penyederhanaan kategori kualitas wine, diperoleh nilai akurasi terhadap data tes sebesar 0.83, dan akurasi terhadap data latih sebesar 0.93298. Hasil tersebut ditampakkan pada gambar di bawah ini.

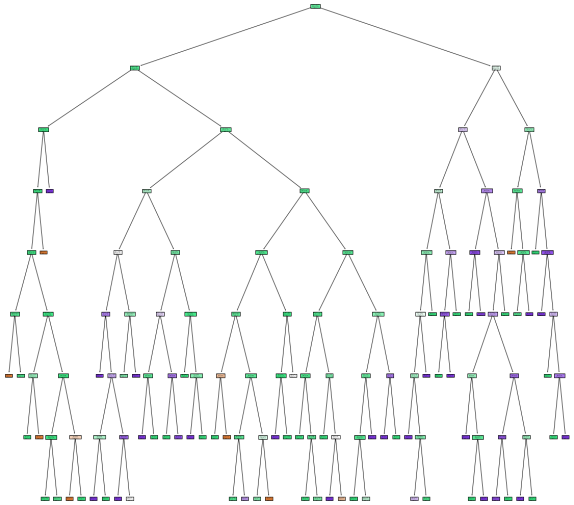
```
[ ] clf = DecisionTreeClassifier(ccp_alpha=0.001, max_depth=8, max_features='log2',
                                 random_state=1)
    clf = clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
```

```
[ ] print("Test Accuracy :",metrics.accuracy_score(y_test, y_pred))
    print("Train Accuracy:",metrics.accuracy_score(y_train, clf.predict(X_train)))

    Test Accuracy : 0.8333333333333334
    Train Accuracy: 0.9329758713136729
```

Lebih jelasnya lagi, berikut adalah tampilan Confusion Matrix dari model di atas.



Terakhir, berikut ini merupakan tampilan dari hasil Decision Tree yang bersesuaian dengan model yang telah dibangun.



Perbandingan akurasi hasil akurasi model Decision Tree.

No	Perubahan	Akurasi Test	Akurasi Train
1	Tidak diubah	0.59791667	1.0
2	Feature selection	0.63125.	1.0
3	GridSearch CV	0.55833	0.60768
4	Pembagian quality menjadi 3	0.83	0.93298

5. Kesimpulan

Berdasarkan hasil pemodelan yang dilakukan untuk memprediksi kualitas dari suatu jenis wine menggunakan

data yang didapat dari kaggle diperoleh kesimpulan sebagai berikut :

1. Pemodelan dilakukan dengan menggunakan 1599 record data dengan 12 fitur (*fixed acidity*, *volatile acidity*, *citric acid*, *residual sugar*, *chloride*, *free sulfur dioxide*, *total sulfur dioxide*, *density*, *pH*, *sulphates*, dan *alcohol*) yang kemudian dilakukan beberapa pemrosesan.
2. 1599 record data dibagi menjadi 2, 70% untuk training dan 30% untuk testing dengan akurasi untuk model awal sebesar 0.59791667. Kemudian dilakukan pemilihan fitur agar meningkatkan akurasi dari model. Dalam hal ini dilakukan 2 metode, dengan membandingkan korelasi dan juga metode SelectKBest. Dari 2 metode ini didapatkan akurasi yang lebih baik dengan penyeleksian fitur berdasarkan nilai korelasi nya. Dengan model tersebut masih memiliki indikasi overfitting dimana nilai akurasi untuk data train nya adalah 1.0, maka perlu dilakukan parameter tuning yang kemudian menghasilkan model dengan akurasi untuk data train dan testing nya sebesar 0.60768 dan 0.55833.
3. Dari 11 kategori kualitas wine tadi diubah menjadi 3 kategori yang dapat dipahami lebih mudah oleh orang awam yaitu ‘buruk’,’sedang’, dan ‘baik’. Setelah mengkategorikan menjadi 3 dilakukan kembali tuning parameter dan didapatkan nilai akurasi yang lebih tinggi yaitu 0.83 untuk data testing dan 0.93298 untuk data training.
4. Didapat root untuk decision tree tersebut adalah alcohol.

6. Saran

Pada deskripsi data tersebut disebutkan bahwa skala quality berkisar dari 0 sampai 10 akan tetapi pada dataset yang tersedia nilai quality yang muncul adalah 3 sampai 8. Untuk itu sekiranya perlu menambahkan record data agar mencakup data yang memiliki quality yang masuk ke dalam skala 0 sampai 10. Selain itu untuk kasus ini penting untuk menggali informasi lebih dalam mengenai kualitas wine agar selama pemodelan kita dapat menentukan hal-hal yang menjadi penentu bagi variabel respon yang akan diprediksi.

7. Daftar Pustaka

[1]Artero Ana, Artero Arturo, Tarín JJ, Cano A. 2015. The impact of moderate wine consumption on health. *Maturitas*. 80(1):3–13.

[2]Charters S, Pettigrew S. 2007. The dimensions of wine quality. *Food Quality and Preference*. 18(7):997–1007. doi:https://doi.org/10.1016/j.foodqual.2007.04.003.

[3]Cherkassky V, Ma Y. 2004. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural networks*. 17(1):113–126.

[4]E. Ebeler S. Flavor Chemistry : Thirty Years of Progress. Di dalam: *Chapter Linking flavour chemistry to sensory analysis of wine*. Kluwer Academic Publishers. hlm 409–422.

[5]Gulati P, Sharma A, Gupta M. 2016. Theoretical Study of Decision Tree Algorithms to Identify Pivotal Factors for Performance Improvement: A Review. *International Journal of Computer Applications*. 141:19–25. doi:10.5120/ijca2016909926.

[6]Hopfer H, Nelson J, Ebeler SE, Heymann H. 2015. Correlating Wine Quality Indicators to Chemical and Sensory Measurements. *Molecules*. 20(5):8453–8483. doi:10.3390/molecules20058453.

[7]Jackson RS. 2000. *Wine science: principle, practice, perception*. Academic Press.

[8]Markoski MM, Garavaglia J, Oliveira A, Olivaes J, Marcadenti A. 2016. Molecular Properties of Red Wine Compounds and Cardiometabolic Benefits. *Nutr Metab Insights*. 9:NMI.S32909. doi:10.4137/NMI.S32909.

[9]Ribéreau-Gayon P, Dubourdieu D, Donèche B, Lonvaud A. 2006. *Handbook of enology, Volume 1: The microbiology of wine and vinifications*. Volume ke-1. John Wiley & Sons.

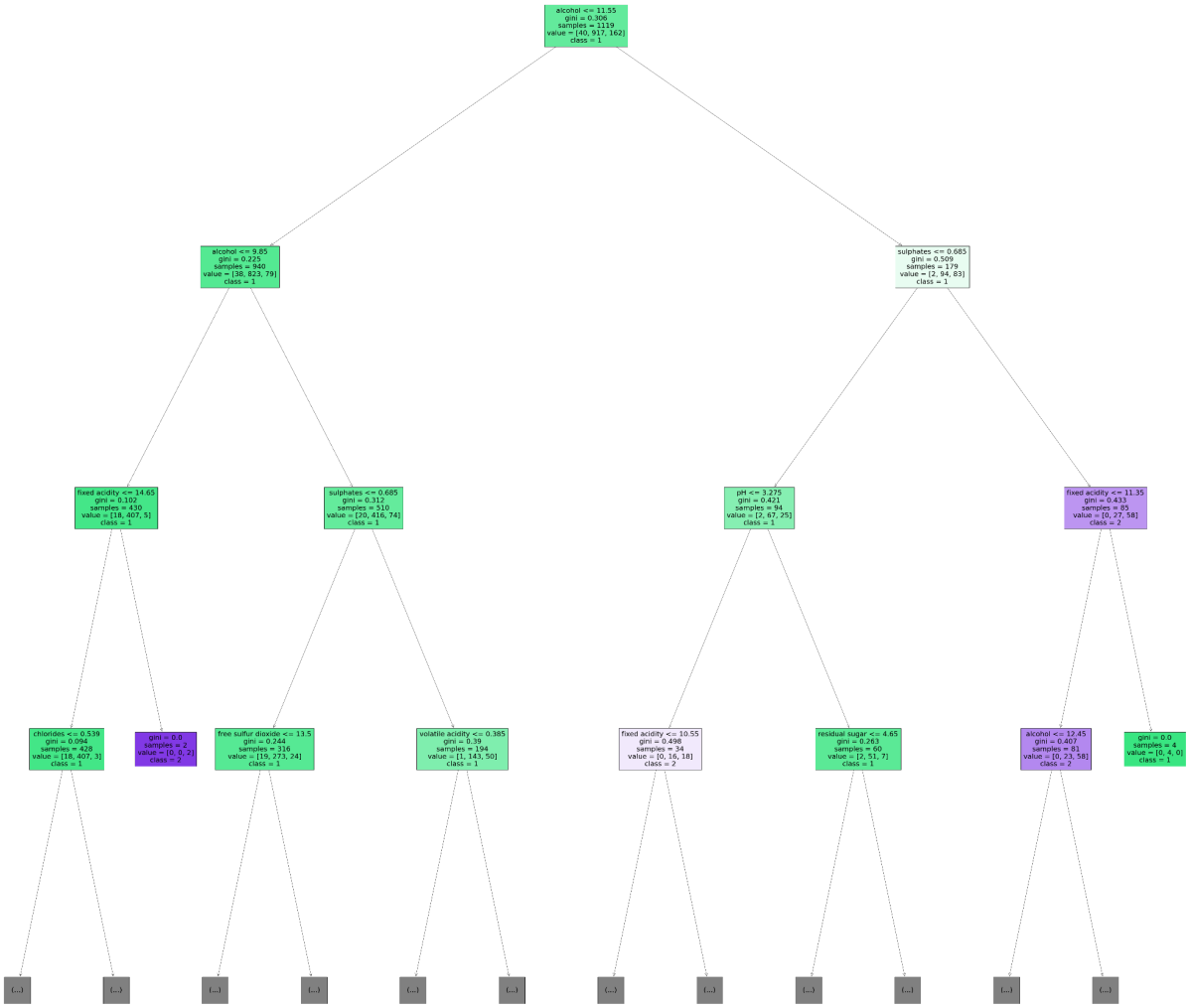
[10]Singh D, Choudhary N, Samota J. 2013. Analysis of Data Mining Classification with Decision treeTechnique.

[11]Suryakanthi T. 2020. Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm*. *International Journal of Advanced Computer Science and Applications*. 11. doi:10.14569/IJACSA.2020.0110277.

[12]wine | Definition, History, Varieties, & Facts | Britannica. [diakses 2021 Des 21]. https://www.britannica.com/topic/wine.

[13]Wine consumption worldwide, 2020. *Statista*., siap terbit. [diakses 2021 Des 21]. https://www.statista.com/statistics/232937/volume-of-global-wine-consumption/.

Lampiran


alcohol <= 11.55
gini = 0.308
samples = 1119
value = [40, 917, 162]
class = 1
alcohol <= 9.85
gini = 0.225
samples = 940
value = [38, 823, 79]
class = 1
sulphates <= 0.685
gini = 0.509
samples = 179
value = [2, 94, 83]
class = 1
fixed acidity <= 14.65
gini = 0.102
samples = 430
value = [18, 407, 5]
class = 1
sulphates <= 0.685
gini = 0.312
samples = 510
value = [20, 416, 74]
class = 1
pH <= 3.275
gini = 0.421
samples = 94
value = [2, 67, 25]
class = 1
fixed acidity <= 11.35
gini = 0.433
samples = 85
value = [0, 27, 58]
class = 2
chlorides <= 0.539
gini = 0.094
samples = 426
value = [18, 407, 3]
class = 1
gini = 0.0
samples = 2
value = [0, 0, 2]
class = 2
free sulfur dioxide <= 13.5
gini = 0.244
samples = 316
value = [19, 273, 24]
class = 1
volatile acidity <= 0.385
gini = 0.39
samples = 194
value = [1, 143, 50]
class = 1
fixed acidity <= 10.55
gini = 0.498
samples = 34
value = [0, 16, 18]
class = 2
residual sugar <= 4.65
gini = 0.263
samples = 60
value = [2, 51, 7]
class = 1
alcohol <= 12.45
gini = 0.407
samples = 81
value = [0, 23, 58]
class = 2
gini = 0.0
samples = 4
value = [0, 4, 0]
class = 1
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)