Analisis Pengendalian Mutu pada Produk Susu Menggunakan Algoritma *Decision Tree* dan *Logistic Regression*

Disusun Oleh:

Nama : Mohammad Arif Mustofa

Asal PT : Universitas Sebelas Maret

ID : 149290040101-289

Apa pertanyaan Bisnisnya?

1. Mengapa pengendalian mutu suatu produk menjadi penting?
2. Bagaimana korelasi antar variable/*feature* terhadap tingkat mutu produk susu?
3. Bagaimana sebaran data pada dataset produk susu?
4. Bagaimana performa model dari *Decision Tree dan Logistic Regression* terhadap hasil prediksi?
5. Bagaimana pengaruh rasio pelatihan (*traing size)* data terhadap performa model?

Apa saja yang akan dibahas?

Data Understanding

Data Preprocessing

Visualisasi Data

Pemodelan dan Evaluasi

Prediksi

Selanjutnya

Apa itu Quality Control?

Quality Control adalah pengendalian mutu suatu produk. Peran *Quality Control* sangat diperlukan dalam berbagai sektor industri, mulai dari manufaktur hingga produksi tangan. Tugas umum dari QC adalah memeriksa dan menguji produk. Setiap perusahaan pasti ingin menghasilkan produk yang baik dan berkualitas, disini peranan seorang *quality control* sangat diperlukan.

Tujuan *Quality Control* adalah memastikan bahwa produk yang akan dipasarkan bebas dari cacat dan dapat diterima sesuai dengan persyaratan kualitas yang ditentukan. Jika ditemukan produk yang cacat maka diperlukan tindakan perbaikan yang sesuai.

Gambar: Google.com

Dataset
ini dikumpulkan secara manual dari observasi. Ini membantu kami membuat model pembelajaran mesin untuk memprediksi kualitas susu.

Dataset ini terdiri dari 7 variabel bebas yaitu **pH, Suhu, Rasa, Bau, Lemak, Kekeruhan, dan Warna**. Dan kelas target berupa **Grade** atau Kualitas susu. Parameter ini memainkan peran penting dalam analisis prediksi susu.

Bisa Target Rendah (Buruk) Sedang (Sedang) Tinggi (Baik) Jika Rasa, Bau, Lemak, dan Kekeruhan terpenuhi dengan kondisi optimal dengan angka 1 dan jika tidak dengan angka 0. Suhu dan pH diberikan nilai sebenarnya dalam dataset. Kita harus melakukan preprocessing data, dan teknik augmentasi data untuk membangun model statistik dan prediktif untuk memprediksi kualitas susu.

Gambar: Kaggle.com

Dataset URL:
https://www.kaggle.com/datasets/cpluzshrijayan/milkquality

- Melakukan *import* library yang dibutuhkan yaitu pandas, numpy, matplotlib dan seaborn untuk visualisasi dan sklearn untuk *built model*

```
In [100]: #import library yang diperlukan
          import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt
          import seaborn as sns
          from sklearn.tree import DecisionTreeClassifier
```

- Pembacaan dataset menggunakan *head()* untuk menampilkan 5 baris data teratas dan *tail()* untuk menampilkan 5 baris data terbawah

```
In [101]: data = pd.read_csv('milknew.csv')
          data.head()
```

Out[101]:

	pH	Temprature	Taste	Odor	Fat	Turbidity	Colour	Grade
0	6.6	35	1	0	1	0	254	high
1	6.6	36	0	1	0	1	253	high
2	8.5	70	1	1	1	1	246	low
3	9.5	34	1	1	0	1	255	low
4	6.6	37	0	0	0	0	255	medium

- Melihat dimensi baris dan kolom dari dataset menggunakan data.*shape()* diperoleh jumlah baris dan kolom adalah 1059 baris, 8 kolom

- Melakukan pengecekan *summary* menggunakan fungsi data.*describe() kemudian transpose untuk mentranspose*

```
In [102]: print("\n Data set ini memiliki jumlah baris dan kolom (baris, kolom) yaitu:", data.shape)
```

```
 Data set ini memiliki jumlah baris dan kolom (baris, kolom) yaitu: (1059, 8)
```

```
In [103]: print(f"Summary of Milk Dataset :\n")
          data.describe().T
```

```
Summary of Milk Dataset :
```

Out[103]:

	count	mean	std	min	25%	50%	75%	max
pH	1059.0	6.630123	1.399679	3.0	6.5	6.7	6.8	9.5
Temprature	1059.0	44.226629	10.098364	34.0	38.0	41.0	45.0	90.0
Taste	1059.0	0.546742	0.498046	0.0	0.0	1.0	1.0	1.0
Odor	1059.0	0.432483	0.495655	0.0	0.0	0.0	1.0	1.0
Fat	1059.0	0.671388	0.469930	0.0	0.0	1.0	1.0	1.0
Turbidity	1059.0	0.491029	0.500156	0.0	0.0	0.0	1.0	1.0
Colour	1059.0	251.840415	4.307424	240.0	250.0	255.0	255.0	255.0

```
In [104]: print(f"Informations of Milk Dataset :\n")
          print(data.info())
```

```
Informations of Milk Dataset :

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1059 entries, 0 to 1058
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   pH           1059 non-null   float64
 1   Temprature   1059 non-null   int64
 2   Taste        1059 non-null   int64
 3   Odor         1059 non-null   int64
 4   Fat          1059 non-null   int64
 5   Turbidity    1059 non-null   int64
 6   Colour       1059 non-null   int64
 7   Grade        1059 non-null   object
dtypes: float64(1), int64(6), object(1)
memory usage: 66.3+ KB
None
```

- Melihat informasi berupa **tipe data, kolom *feature* apa saja yang Null dan kolom data serta jumlah baris.**
- Dari tiap kolom *feature* didapatkan pH berupa data bertipe *float, s*edangkan suhu, rasa, bau, lemak, kekeruhan, warna berupa data tipe *integer* sedangkan kelas target bertipe *object*

Kolom feature “Warna”

```
In [105]: print(data["Colour"].shape)
          print(f'\n There are {len(data["Colour"].unique())} Colour enlisted here.\n')
          data["Colour"].unique()

          (1059,)

           There are 9 Colour enlisted here.

Out[105]: array([254, 253, 246, 255, 250, 247, 245, 240, 248], dtype=int64)
```

Kolom target

```
In [106]: print(data["Grade"].shape)
          print(f'\n There are {len(data["Grade"].unique())} Grade enlisted here.\n')
          data["Grade"].unique()

          (1059,)

           There are 3 Grade enlisted here.

Out[106]: array(['high', 'low', 'medium'], dtype=object)
```

- Dari kolom warna (Colour) dan Grade bisa dilihat unique /kategorik nya berapa banyak.
- Misalnya pada kolom “Colour” didapatkan ada sebanyak 9 jenis dengan kode warna RGB yang secara teori menuju warna putih.
- Kelas target memiliki 3 kategori tingkatan yaitu “high”, “low”, dan “medium”.

Mengecek data yang missing value dengan isnull() atau isna()

```
In [107]: data.isnull().sum()

Out[107]: pH             0
          Temprature     0
          Taste          0
          Odor           0
          Fat            0
          Turbidity      0
          Colour         0
          Grade          0
          dtype: int64
```

No data is Null/NaN

- Dari hasil yang ditampilkan tidak ditemukan data yang kosong artinya semua data telah terisi, sehingga tidak perlu dilakukan pengisian data menggunakan seperti mean, median atau modus.
- Misalnya terdapat data yang kosong pada kolom temperature, kita dapat mengisi dengan sintaks sebagai berikut:

Mean:

```
data['Temprature'].fillna(data['Temprature'].mean(),inplace=True)
```

Median:

```
data['Temprature'].fillna(data['Temprature'].median(),inplace=True)
```

Modus:

```
data['Temprature'].fillna(data['Temprature'].mode(),inplace=True)
```

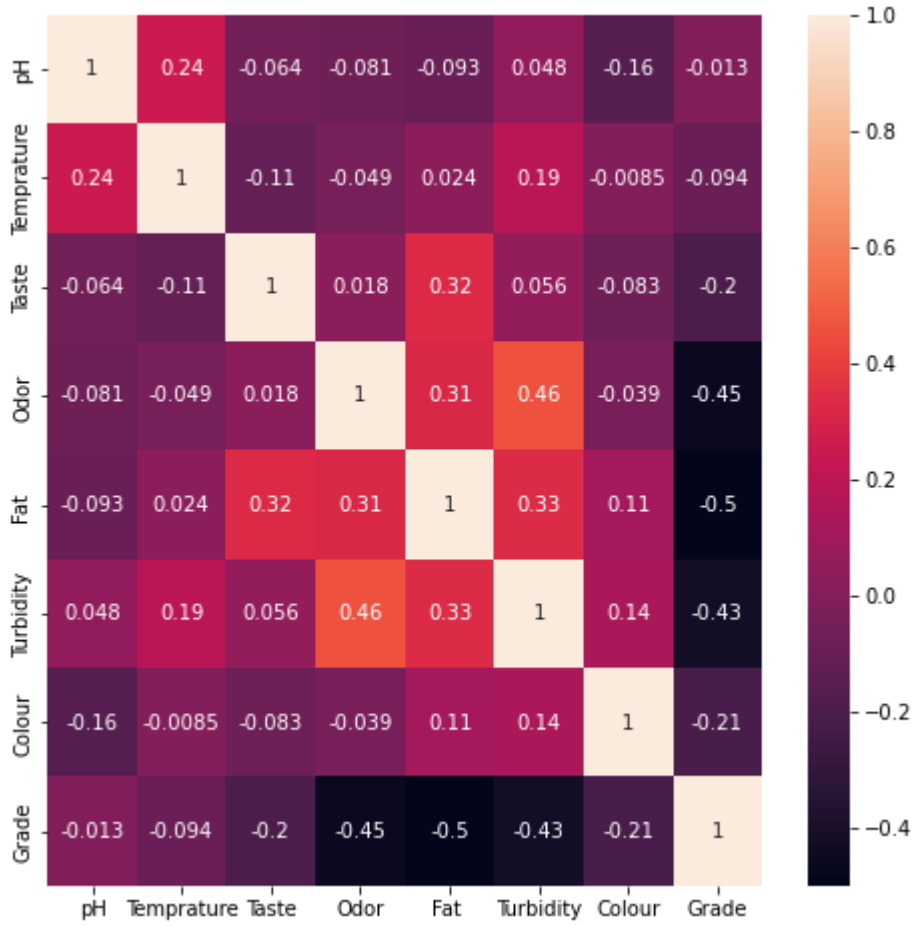
Encoding data untuk mengubah kelas object menjadi numerik. Agar ditentukan korelasi

```
In [280]: from sklearn import preprocessing

le = preprocessing.LabelEncoder()
le.fit(data["Grade"])
data["Grade"]=le.transform(data["Grade"])
Grade_labels = dict(zip(le.classes_, le.transform(le.classes_)))
print(Grade_labels)
```

```
{'high': 0, 'low': 1, 'medium': 2}
```

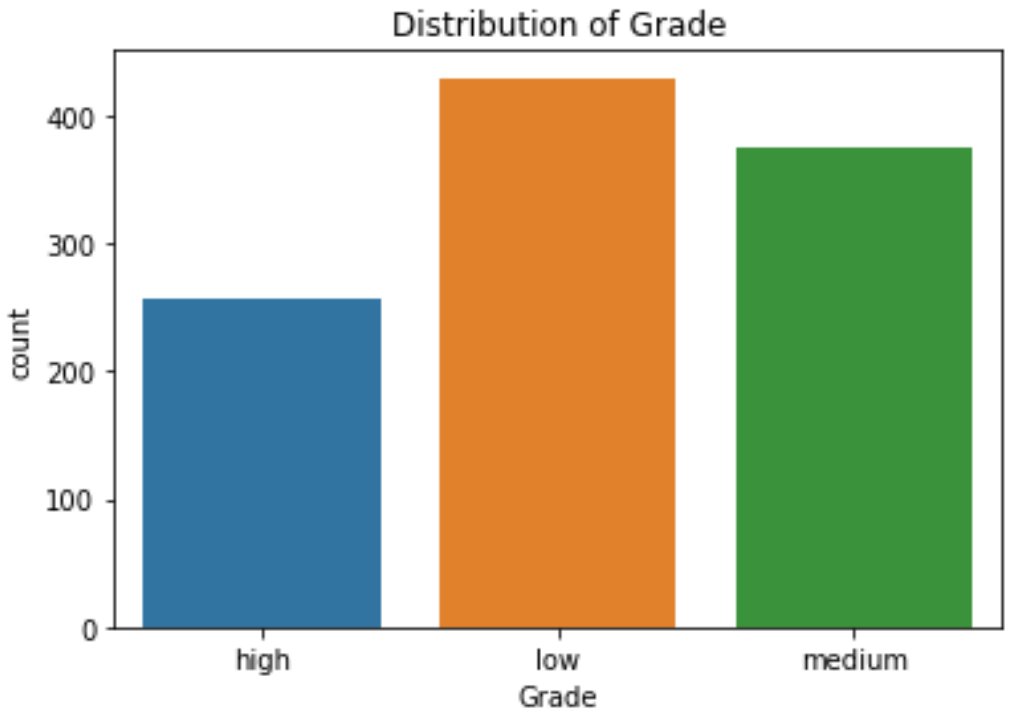
- Dari hasil yang ditampilkan menunjukkan bahwa kode 0= kualitas tinggi, 1=kualitas rendah, dan 2=kualitas medium

- Tabel heatmap didapatkan yang memiliki korelasi tinggi adalah 0.45, 0.5, dan 0.21

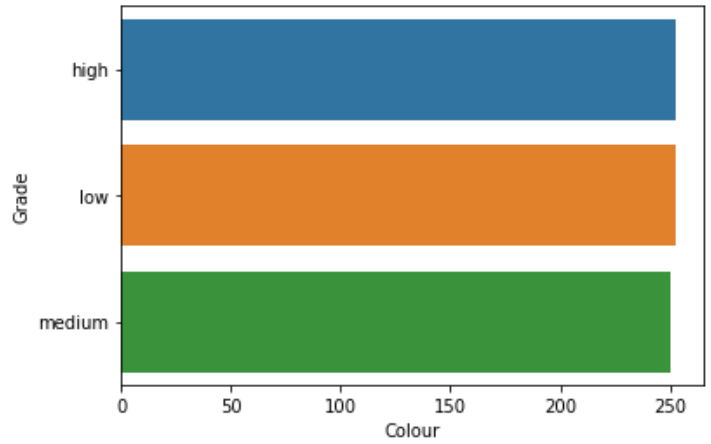


Tabel heatmap

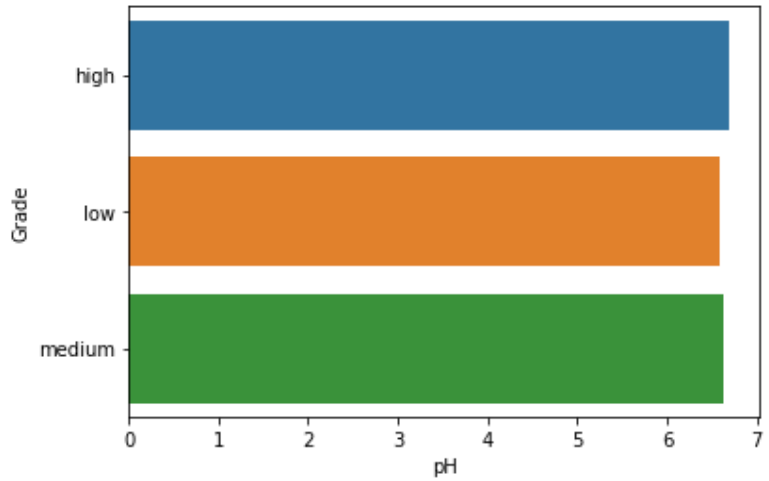
Sebaran Grade




```
<AxesSubplot:xlabel='Colour', ylabel='Grade'>
```




```
<AxesSubplot:xlabel='pH', ylabel='Grade'>
```



Test Train Split

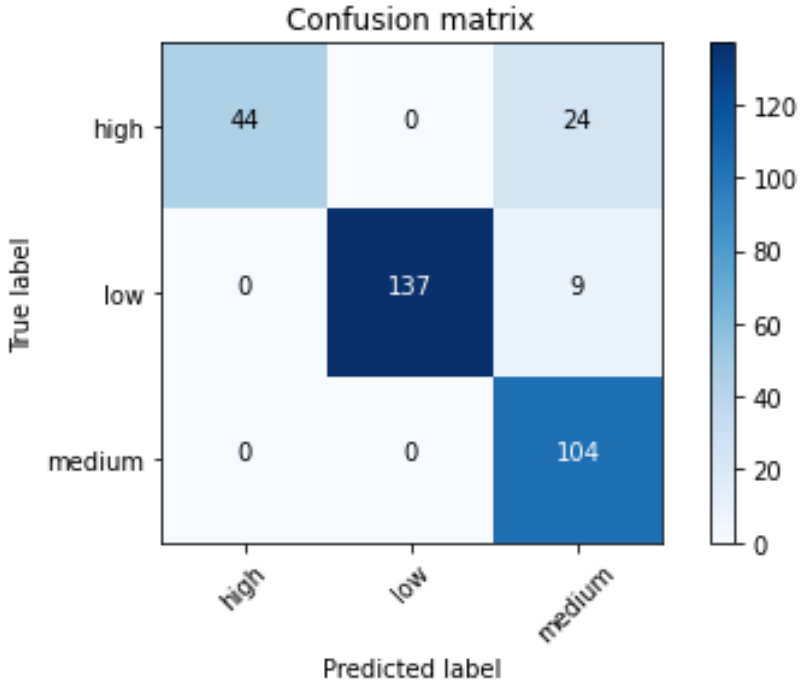
Model	Decision Tree	Logistic Regression
1	Train Size=0,7	Train Size=0,7
2	Train Size=0,8	Train Size=0,8

Data X ,y train, data X,y test diambil dari:

No	X	Y
	Data Feature seluruhnya `X = data.iloc[0:,:7]`	Data target 'Grade' `y = data["Grade"]`

```
In [166]: from sklearn.model_selection import train_test_split

          X_train, X_test, y_train, y_test = train_test_split(X,y, train_size = 0.7)
          X2_train, X2_test, y2_train, y2_test = train_test_split(X,y, train_size = 0.8)
          print(X_train.shape, X_test.shape)
          print(X2_train.shape, X2_test.shape)

          (741, 7) (318, 7)
          (847, 7) (212, 7)
```

Evaluasi model Decision Tree



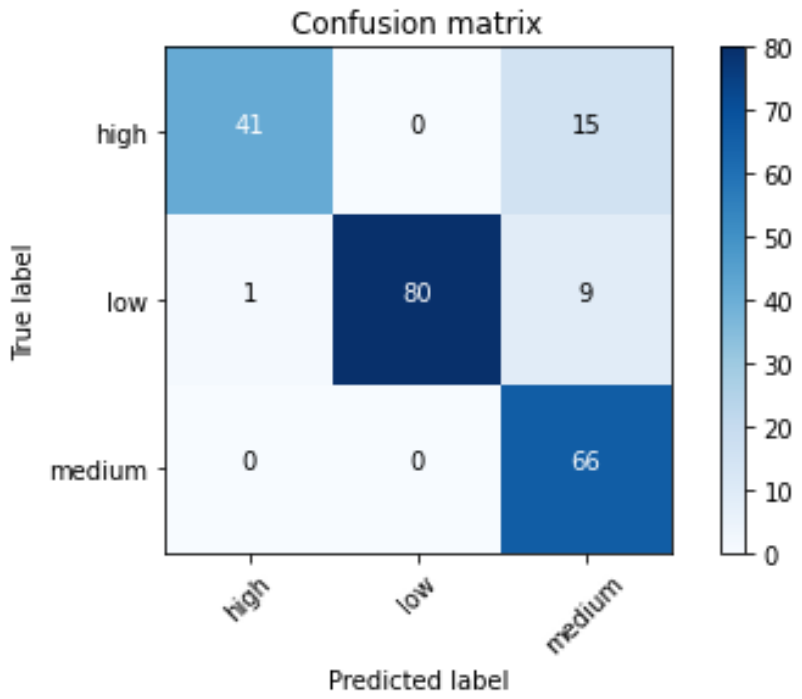

```
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.65      0.79        68
           1       1.00      0.94      0.97       146
           2       0.76      1.00      0.86       104

    accuracy                           0.90       318
   macro avg       0.92      0.86      0.87       318
weighted avg       0.92      0.90      0.89       318

[[ 44   0  24]
 [  0 137   9]
 [  0   0 104]]
```

- Terdapat 24 data yang hasil prediksinya memiliki kualitas medium namun sebenarnya memiliki kualitas yang tinggi (high), 9 data yang hasil prediksi menunjukkan kualitas medium namun sebenarnya memiliki kualitas rendah. Sisanya sesuai dengan hasil prediksi dan sesungguhnya.

$$Akurasi = \frac{TP}{total\ data} = \frac{44 + 137 + 104}{318} \times 100\% = \frac{285}{318} \times 100\% = 90\%$$

Evaluasi model Decision Tree




```
In [174]: print(classification_report(y2_test,y2_pred))
          print(confusion_matrix(y2_test,y2_pred))
```

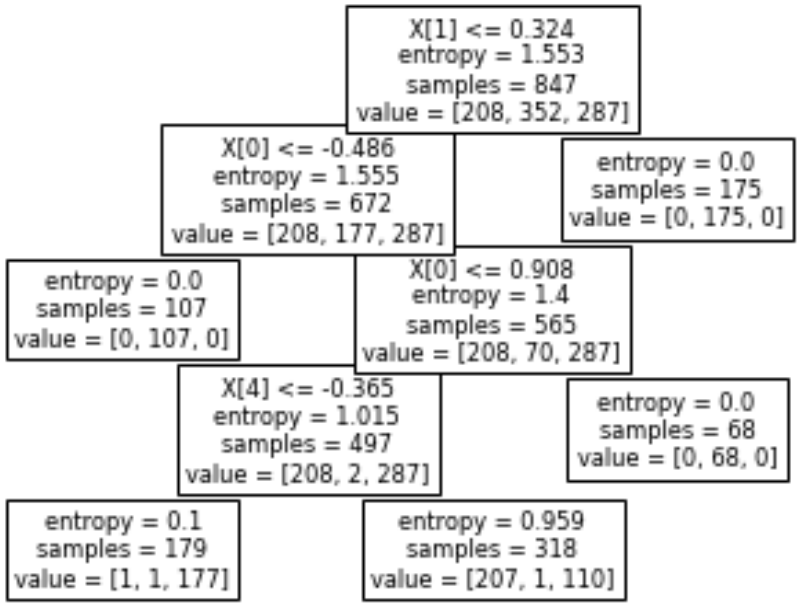
```
              precision    recall  f1-score   support

           0       0.98      0.73      0.84        56
           1       1.00      0.89      0.94        90
           2       0.73      1.00      0.85        66

    accuracy                           0.88       212
   macro avg       0.90      0.87      0.87       212
weighted avg       0.91      0.88      0.88       212

[[41  0 15]
 [ 1 80  9]
 [ 0  0 66]]
```

- Terdapat 15 data yang hasil prediksinya memiliki kualitas medium namun sebenarnya memiliki kualitas yang tinggi (high), 9 data yang hasil prediksi menunjukkan kualitas medium namun sebenarnya memiliki kualitas rendah, serta 1 data terprediksi berkualitas tinggi namun sebenarnya rendah. Sisanya sesuai dengan hasil prediksi dan sesungguhnya.

$$Akurasi = \frac{TP}{total\ data} = \frac{41 + 80 + 66}{212} \times 100\% = \frac{187}{212} \times 100\% = 88{,}2\%$$

Membangun Tree

```
In [51]: from sklearn import tree

         tree.plot_tree(gradeTree);
```



Evaluasi model Logistic Regression

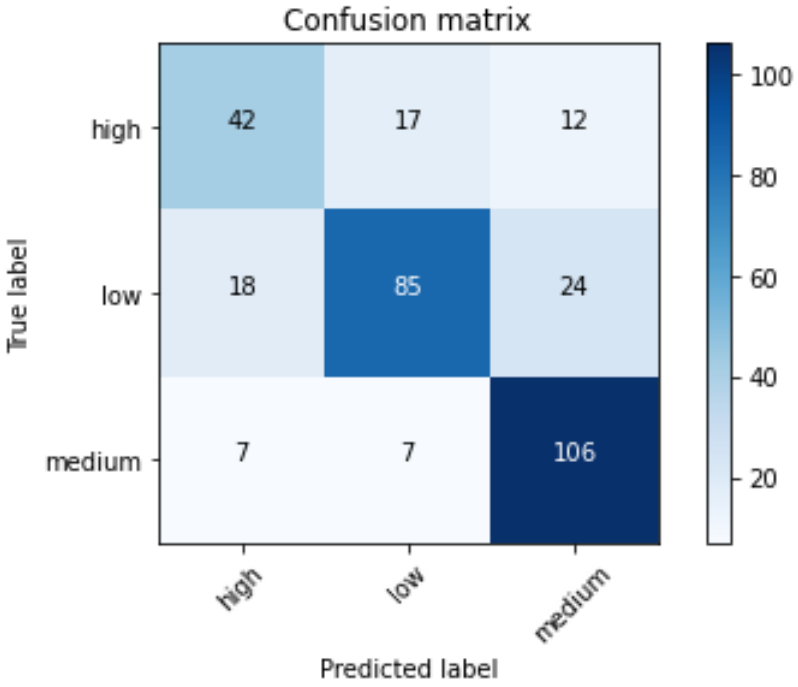
In [221]:
```
from sklearn.linear_model import LogisticRegression

LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_train,y_train)
LR
```

Out[221]:
```
LogisticRegression(C=0.01, solver='liblinear')
```

In [252]:
```
#menghitung akurasi
from sklearn import metrics

akurasi = metrics.accuracy_score(y_test, y_predLR)
print(akurasi)
```

```
0.7327044025157232
```

Hasilnya didapatkan bahwa model Decision Tree lebih baik daripada model Logistic Regression.



Decision Tree untuk pengujian prediksi

Prediksi dengan inputan

- pH = 6.1
- Suhu dalam derajat = 35
- Memiliki rasa
- Berbau
- Tidak memiliki kandungan lemak
- Keruh
- Warna kode = 250

Output: “Low quality”

```
In [275]: pH = float(input("Masukkan nilai pH = "))
          suhu = int(input("Masukkan nilai Suhu ="))
          rasa = int(input("Memiliki rasa atau Tidak? Yes=1, No=0 = "))
          bau = int(input("Berbau atau Tidak? Yes=1, No=0 = "))
          lemak = int(input("Memiliki kandungan lemak atau Tidak? Yes=1, No=0 = "))
          kekeruhan = int(input("Keruh atau Tidak? Yes=1, No=0 = "))
          warna = int(input("Masukkan kode warna susu dalam RGB (0-255)? = "))

          Masukkan nilai pH = 6.1
          Masukkan nilai Suhu =35
          Memiliki rasa atau Tidak? Yes=1, No=0 = 1
          Berbau atau Tidak? Yes=1, No=0 = 1
          Memiliki kandungan lemak atau Tidak? Yes=1, No=0 = 0
          Keruh atau Tidak? Yes=1, No=0 = 1
          Masukkan kode warna susu dalam RGB (0-255)? = 250

In [276]: databaru = [[pH,suhu,rasa,bau,lemak,kekeruhan,warna]]
          predBaru = gradeTree.predict(databaru)
          predBaru

Out[276]: array([1])
```

Dimana : 0 = Kualitas susu High (tinggi), 1 = Kualitas susu Low (rendah), 2 = Kualitas susu Medium (sedang)

TERIMA KASIH

Mohammad Arif

moh.arif11022000@gmail.com