DATA SCIENCE: FACTOR ANALYSIS OF THE CAUSES OF ANEMIA

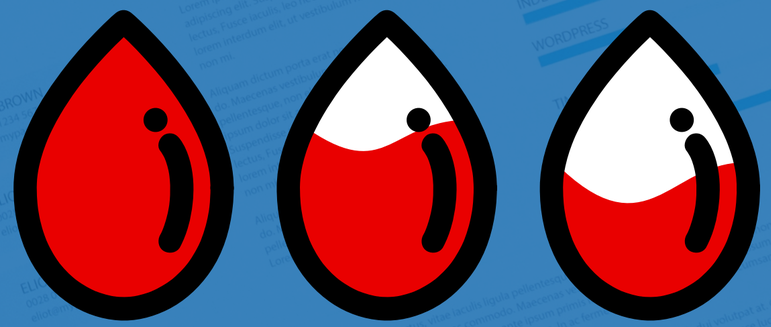


TABLE OF CONTENT

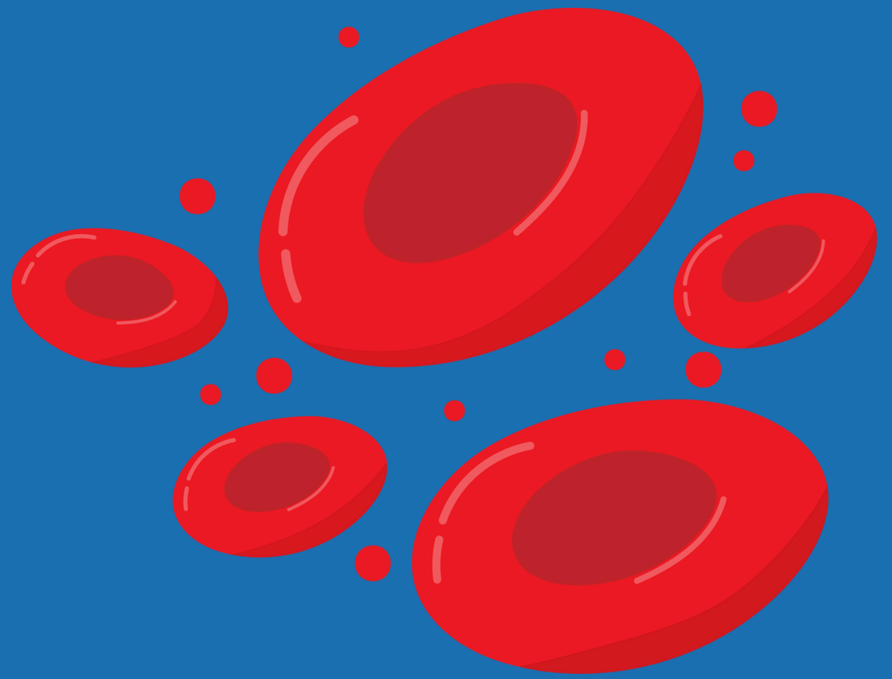
About the writer

Rifqi Arrayan

Hi! I am a third semester student at UPN Veteran Jakarta. I am a fast learner and have a strong interest in data analysis. My organizational experience has equipped me with excellent teamwork and communication skills. I enjoy being involved in various projects in the IT sector, and I continue to look for opportunities to learn and improve my skills through professional organizations, and am currently exploring the data field.

Background

Anemia is a significant health condition in Indonesia, with a high prevalence especially in children, adolescent girls and pregnant women. The main causes include iron deficiency, other nutritional deficiencies, chronic diseases, and genetic conditions. Anemia has a negative impact on maternal and child health, productivity, and child growth and development. Mitigation efforts involve iron supplementation, food fortification, improving diet, and controlling related diseases. A multi-sectoral approach involving various parties is needed to reduce the prevalence of anemia and improve the quality of life of Indonesian people.



PROBLEM AND GOALS

PROBLEM

Anemia is a serious health problem that not only causes significant morbidity and reduced quality of life, but also imposes a high cost burden on the healthcare system in Indonesia. Early detection and timely intervention are essential to reduce the severity of the condition and the resulting complications. However, early diagnosis of anemia is often difficult due to non-specific early symptoms and similarities with other diseases. Therefore, effective predictive tools and coping strategies are needed to help early diagnosis and more efficient treatment of anemia.

VS

GOALS

Anemia data processing aims to develop a prediction model that can assist in the early detection of anemia based on clinical and demographic data. By using machine learning techniques, this model is expected to:

1. Predict anemia with high accuracy.
2. Identify the main risk factors that contribute to the occurrence of anemia.
3. Improve efficiency in early detection of anemia cases.
4. Identify the best machine learning algorithms for data modeling and classification.
5. Provide prediction to the patient whether anemia or not based on the given factors.

TOOLS

Google colab is used as platform for analyzing data

Google Sheets is used to make data easier preprocessing is like removing duplicates

pyhton is used data for data analysis processes

ANALYSIS PROCESS

1. Data Preparation
2. Exploratory Data Analysis
3. Preprocessing Data
4. Processing Data
5. Modeling Data
6. Evaluation

Data Preparation

This data is taken from kaggle, where this data is data on factors that cause anemia taken from a sample of a hospital.
where anemia is very much suffered by women and adolescents

Rows

1420

Columns

6

Import library data

```
from google.colab import files
upload = files.upload()
```

```
Choose Files anemia.csv
• anemia.csv(text/csv) - 34627 bytes, last modified: 6/19/2024 - 100% done
Saving anemia.csv to anemia.csv
```

Displays summary information about the DataFrame, including the number of rows and columns, column names, the number of non-zero values in each column, the data type for each column, and memory usage.

Reading dataset structure

```
df.info ()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1421 entries, 0 to 1420
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Gender      1421 non-null   int64
 1   Hemoglobin  1421 non-null   float64
 2   MCH         1421 non-null   float64
 3   MCHC        1421 non-null   float64
 4   MCV         1421 non-null   float64
 5   Result      1421 non-null   int64
dtypes: float64(4), int64(2)
memory usage: 66.7 KB
```

Read the entire contents of the dataset

```
import pandas as pd
df=pd.read_csv("anemia.csv")
df
```

	Gender	Hemoglobin	MCH	MCHC	MCV	Result
0	1	14.9	22.7	29.1	83.7	0
1	0	15.9	25.4	28.3	72.0	0
2	0	9.0	21.5	29.6	71.2	1
3	0	14.9	16.0	31.4	87.5	0
4	1	14.7	22.0	28.2	99.5	0
...	...	...	...	...	...	...
1416	0	10.6	25.4	28.2	82.9	1
1417	1	12.1	28.3	30.4	86.9	1
1418	1	13.1	17.7	28.1	80.7	1
1419	0	14.3	16.2	29.5	95.2	0
1420	0	11.8	21.2	28.4	98.1	1

1421 rows × 6 columns

Exploraty Data Analysis

```
df.describe ()
```

	Gender	Hemoglobin	MCH	MCHC	MCV	Result
count	1421.000000	1421.000000	1421.000000	1421.000000	1421.000000	1421.000000
mean	0.520760	13.412738	22.905630	30.251232	85.523786	0.436312
std	0.499745	1.974546	3.969375	1.400898	9.636701	0.496102
min	0.000000	6.600000	16.000000	27.800000	69.400000	0.000000
25%	0.000000	11.700000	19.400000	29.000000	77.300000	0.000000
50%	1.000000	13.200000	22.700000	30.400000	85.300000	0.000000
75%	1.000000	15.000000	26.200000	31.400000	94.200000	1.000000
max	1.000000	16.900000	30.000000	32.500000	101.600000	1.000000

Processing Data

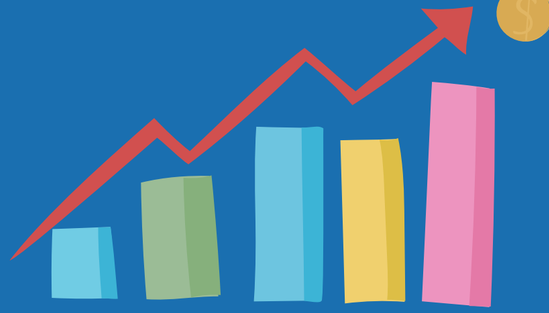
Missing value data

Looking at the dataset whether there is a missing value or not, it can be seen that in the anemia dataframe there is no missing value marked with 0

```
missing_data_anemia = df.isnull().sum()
print(missing_data_anemia)
```

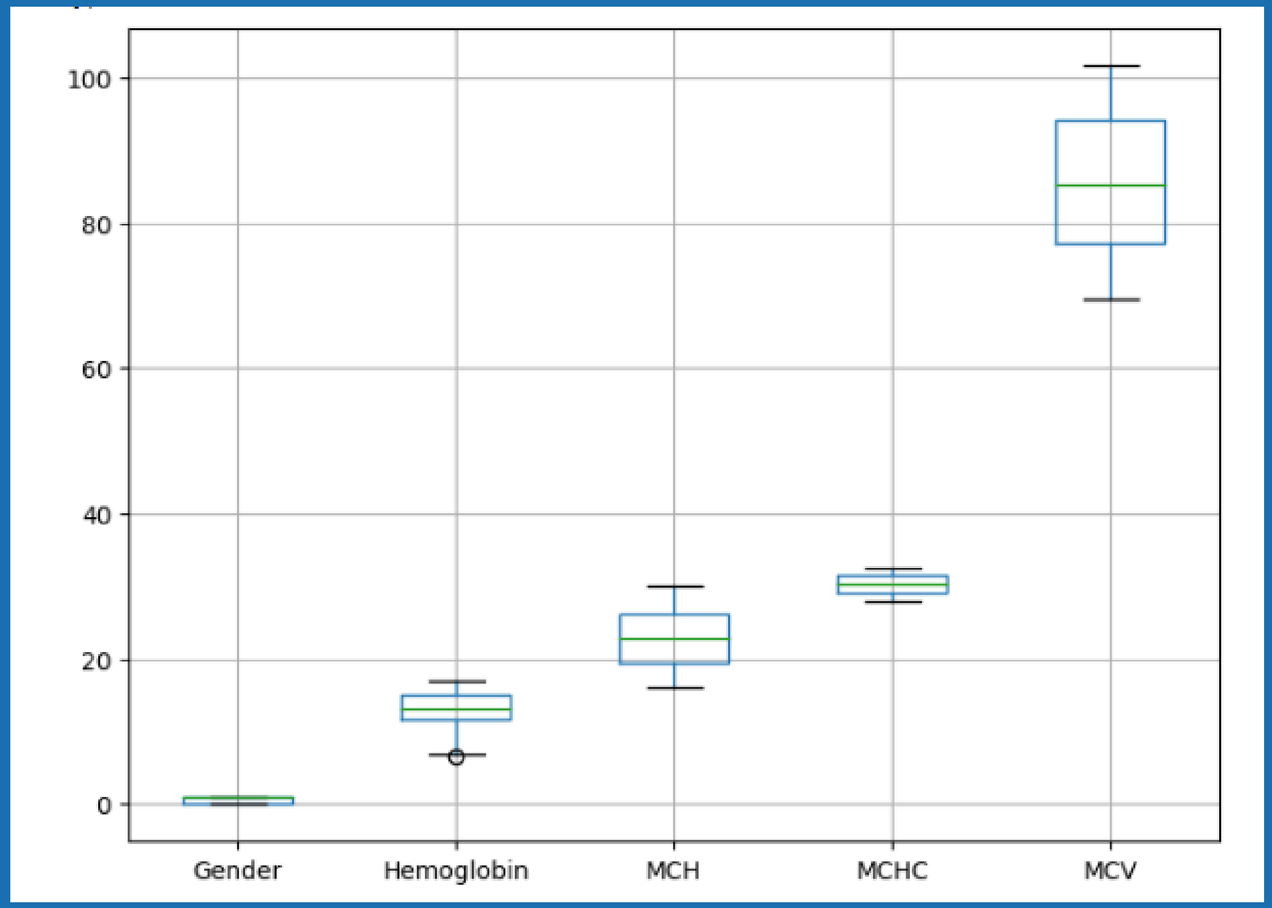
```
Gender          0
Hemoglobin      0
MCH             0
MCHC            0
MCV             0
Result          0
dtype: int64
```

Data Outlier



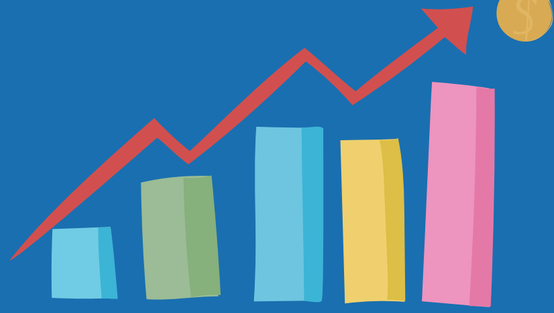
```
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')

plt.figure(figsize=(8,6))
df[['Gender','Hemoglobin','MCH','MCHC','MCV']].boxplot()

#menghitung Q1, Q3 dan IQR pada kolom numerik
numeric_col=['Hemoglobin','MCH','MCHC','MCV']
Q1=df[numeric_col].quantile(0.25)
Q3=df[numeric_col].quantile(0.75)
IQR=Q3-Q1

#menentukan outlier
outliers=((df[numeric_col] < (Q1 - 1.5 * IQR)) | (df[numeric_col] > (Q3 + 1.5 * IQR))).sum()

print('Data outlier:')
print(outliers)
```

Data Outlier




```
Data outlier:
Hemoglobin    1
MCH           0
MCHC          0
MCV           0
dtype: int64
```

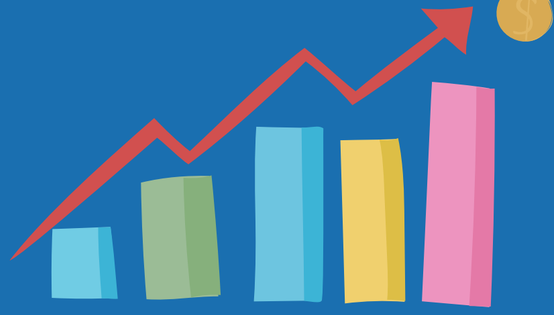
It can be seen that there is one outlier in hemoglobin data.



Removing Outlier

```
dataset_clean = df.copy()

# Kolom numerik yang digunakan
numeric_col = ['Hemoglobin','MCH','MCHC','MCV']

# Menghitung Q1, Q3 dan IQR pada kolom numerik
Q1 = dataset_clean[numeric_col].quantile(0.25)
Q3 = dataset_clean[numeric_col].quantile(0.75)
IQR = Q3 - Q1

# Menentukan outlier
outliers = (dataset_clean[numeric_col] < (Q1 - 1.5 * IQR)) | (dataset_clean[numeric_col] > (Q3 + 1.5 * IQR))

# Menghapus outlier
dataset_no_outlier = dataset_clean[~outliers.any(axis=1)]

print('Data frame tanpa outlier:')
print(dataset_no_outlier)
```

Removing Outlier



Data frame tanpa outlier:

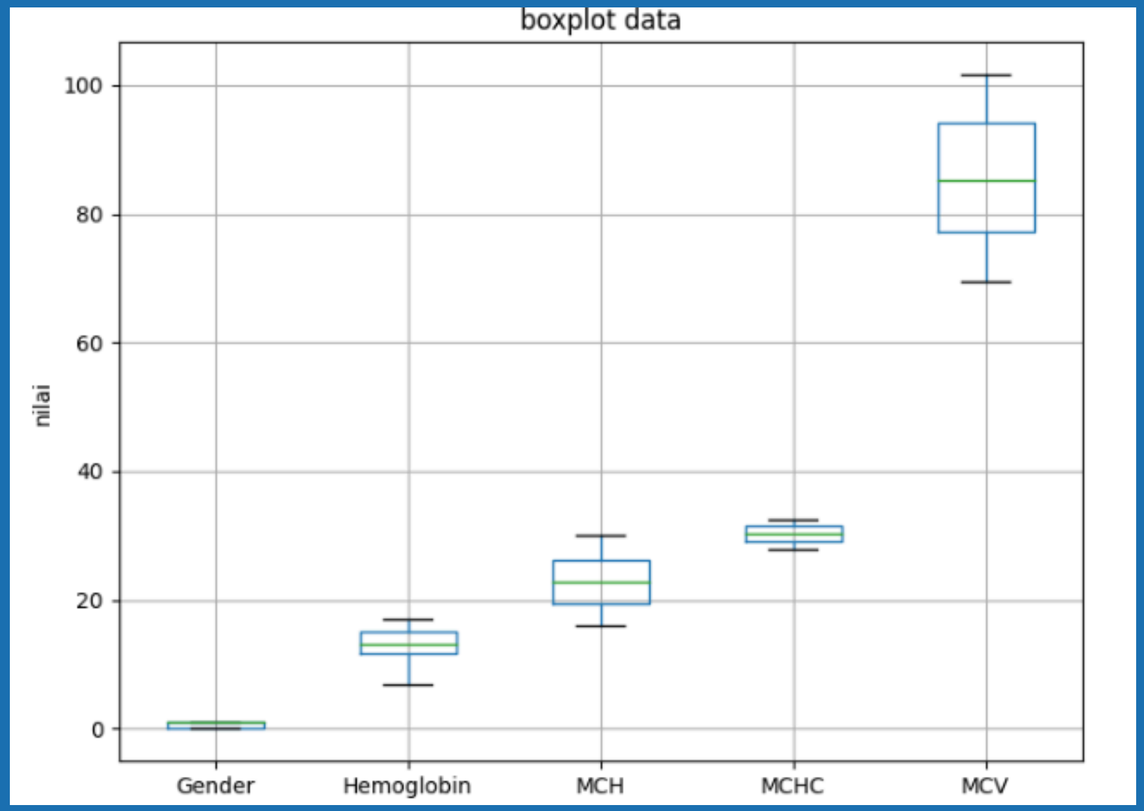
	Gender	Hemoglobin	MCH	MCHC	MCV	Result
0	1	14.9	22.7	29.1	83.7	0
1	0	15.9	25.4	28.3	72.0	0
2	0	9.0	21.5	29.6	71.2	1
3	0	14.9	16.0	31.4	87.5	0
4	1	14.7	22.0	28.2	99.5	0
...	...	...	...	...	...	...
1416	0	10.6	25.4	28.2	82.9	1
1417	1	12.1	28.3	30.4	86.9	1
1418	1	13.1	17.7	28.1	80.7	1
1419	0	14.3	16.2	29.5	95.2	0
1420	0	11.8	21.2	28.4	98.1	1

[1420 rows x 6 columns]

Removing Outlier

```
plt.figure(figsize=(8,6))
dataset_no_outlier[['Gender','Hemoglobin','MCH','MCHC','MCV']].boxplot()
plt.title('boxplot data')
plt.ylabel('nilai')
plt.show()
```

It can be seen that there is one outlier in the hemoglobin column, I deleted it because the number of outliers is not too significant.



Processing Data

Data Processing is a series of actions or processes is done to convert raw data into a form more useful, informative, and understandable.

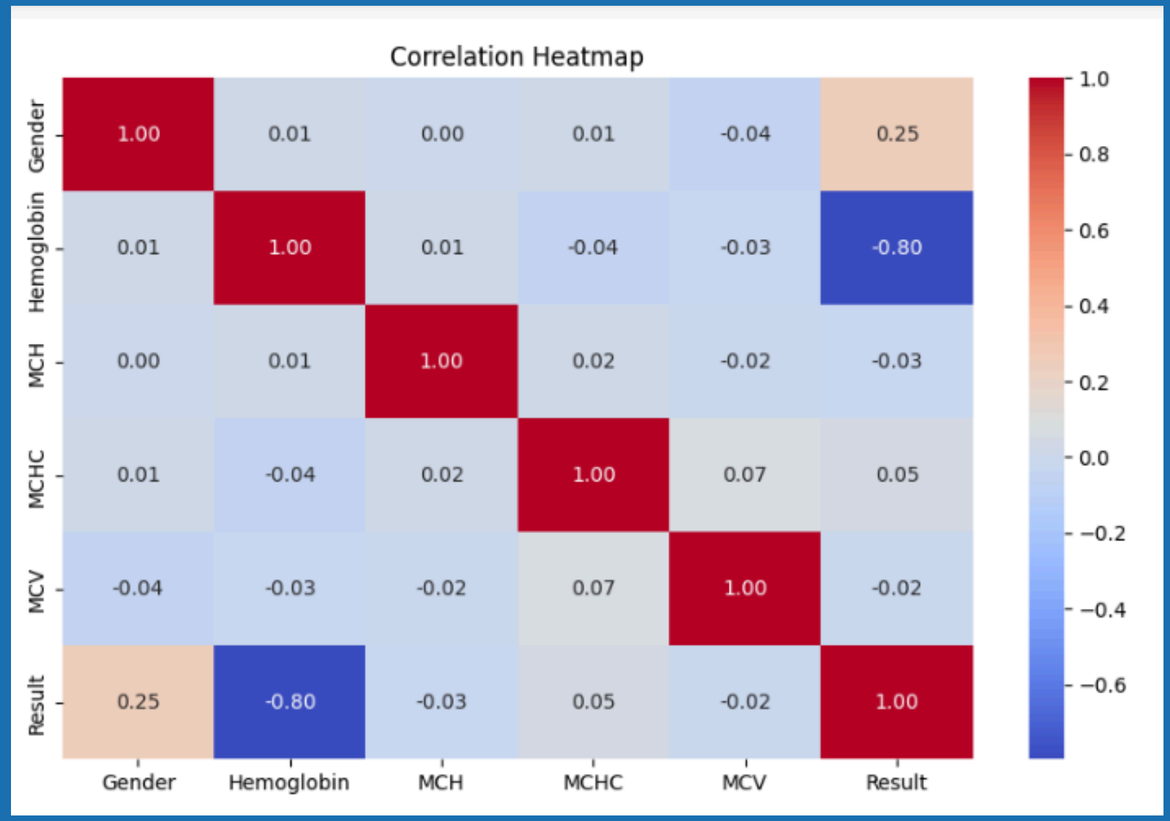
```
# memisahkan atribut pada dataset dan menyimpannya pada sebuah variabel
X = df[df.columns[:5]]

# memisahkan label pada dataset dan menyimpannya pada sebuah variabel
y = df['Result']
```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler() #disebut z score normalization
scaler.fit(X)
X = scaler.transform(X)
```

```
from sklearn.model_selection import train_test_split

# memisahkan data untuk training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

Check the correlation of each feature

```
import seaborn as sns

correlation_matrix = df.corr()
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



the correlation value of each feature with the Result:

Gender: 0.25
Hemoglobin:-0.80
MCH:-0.03
MCHC:0.05
MCV:-0.02
The feature that affects the result the most is Hemoglobin with a correlation value of -0.80. This high negative correlation value indicates that there is a strong relationship between hemoglobin and anemia outcome in this dataset. The negative correlation value means that as the hemoglobin level increases, the likelihood of the outcome (anemia) decreases, and vice versa.

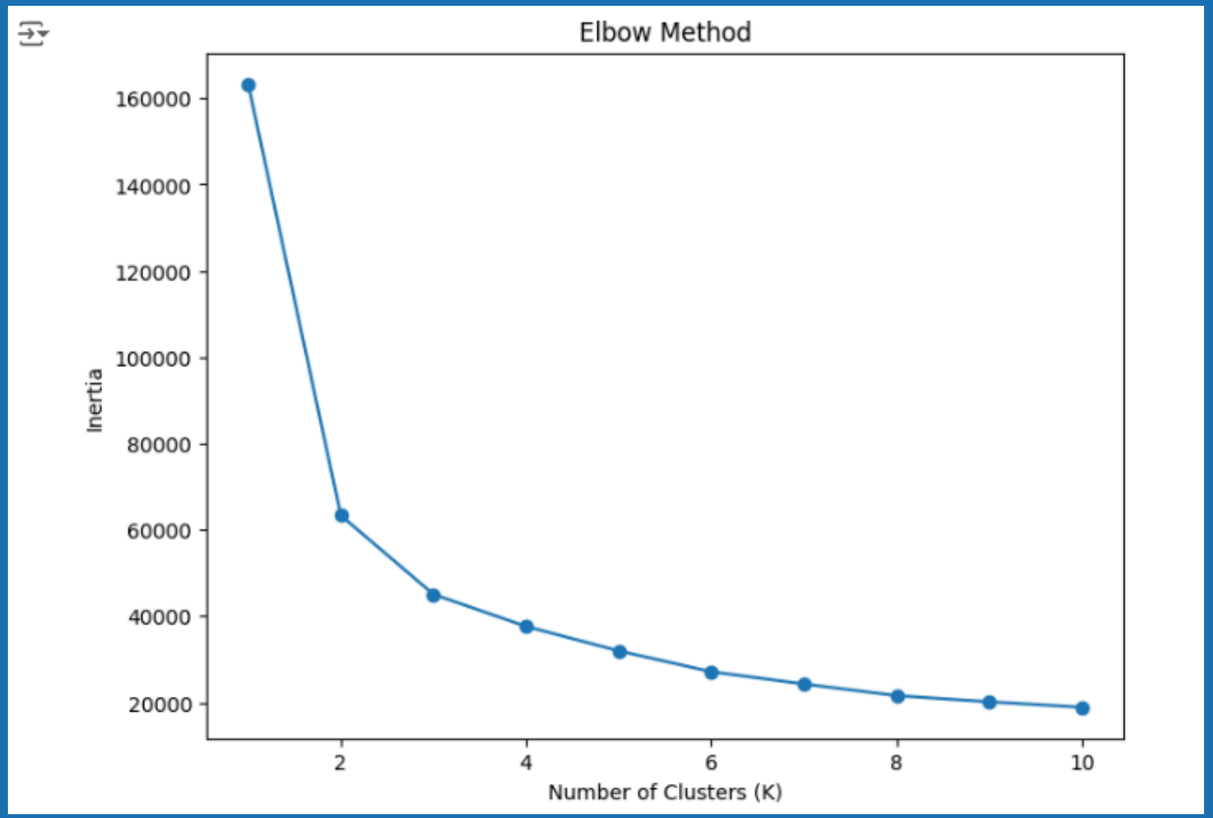
K-Means Modeling

K-Means is choosing a random samples to be used as centroids. Centroid is a sample of the data which is the center of a cluster

```
from sklearn.cluster import KMeans

inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df)
    inertia.append(kmeans.inertia_)
```

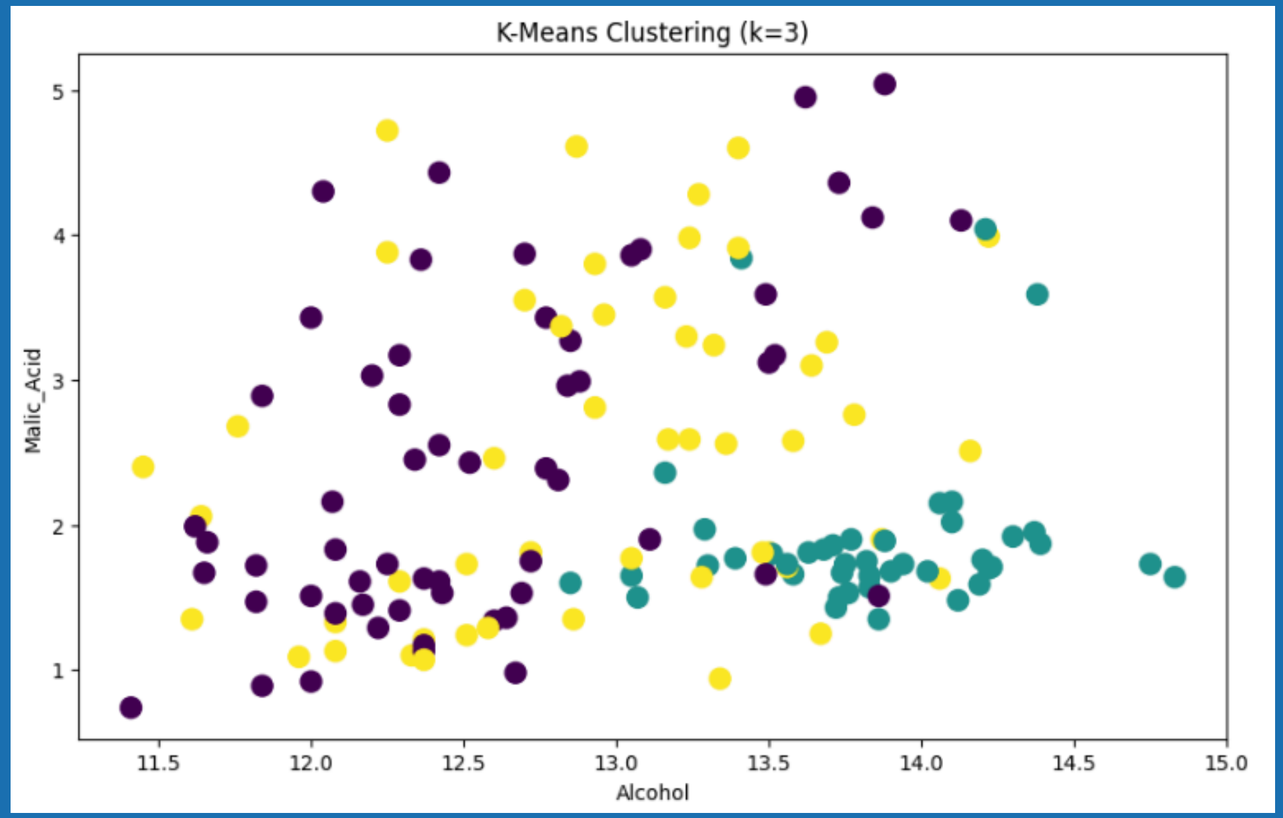
```
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Inertia')
plt.title('Elbow Method')
plt.show()
```

```
k = 3
kmeans = KMeans(n_clusters=k, random_state=42)
clusters = kmeans.fit_predict(df)
```

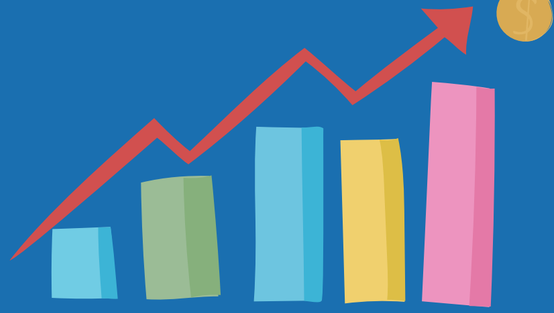


K-Means modelling

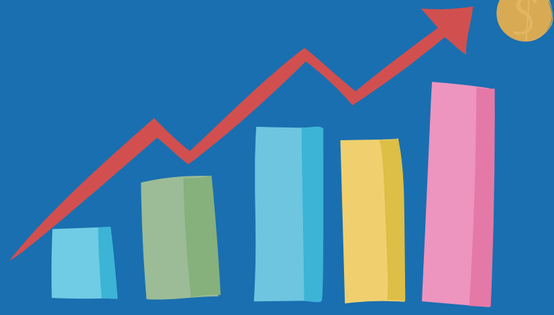
```
plt.figure(figsize=(10, 6))
plt.scatter(df['Hemoglobin'], df['MCH'], c=clusters, cmap='viridis', s=100)
plt.xlabel('Hemoglobin')
plt.ylabel('MCH')
plt.title('K-Means Clustering (k=3)')
plt.show()
```



Modeling data



-Machine Learning Models are commonly used computer programs to recognize patterns in data or make predictions.
-This machine learning model is created from a machine learning algorithm, which trained using labeled, unlabeled, or labeled data using mixed data.
Machine learning is based on algorithms with data labeled (Supervised Learning), unlabeled (Unsupervised Learning), or using mixed data (Reinforcement Learning).
-Model training is the process of running a machine algorithm learning to be able to process the dataset in it which has been divided into training data and optimize the algorithm to find certain patterns or outputs.
-This function will produce a rule and also a data structure called a trained machine learning model.



SVM (Support vector Machine)

used to complete classification, regression, and outlier detection.

```python
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split

# memisahkan antara x dan y
X = df[df.columns[:5]]
y = df['Result']

# memisahkan data untuk training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)

# membuat objek SVC dan memanggil fungsi fit untuk melatih model
clf = SVC()
clf.fit(X_train, y_train)
#fit berguna untuk pembuatan model dalam pembelajaran mesin

from sklearn.metrics import accuracy_score

# Lakukan prediksi pada data pengujian
y_pred_clf = clf.predict(X_test)

# Hitung akurasi model
accuracy = accuracy_score(y_test, y_pred_clf)
#Fungsi accuracy_score membandingkan label yang sebenarnya dari data pengujian (y_test) dengan label yang diprediksi oleh model (y_pred_clf)

print("Akurasi Model SVM:", accuracy)
```

```
Akurasi Model SVM: 0.9147121535181236
```

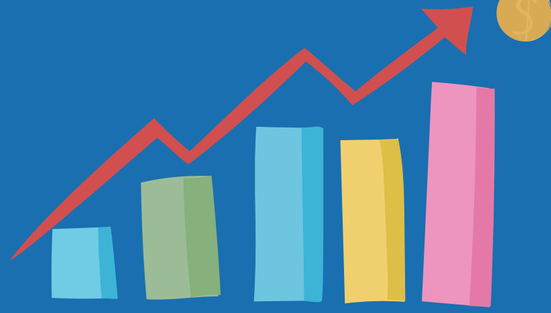
Decion Tree

Decision trees are powerful algorithms, meaning they can be used in complex problems.

```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)

# Lakukan prediksi pada data pengujian
y_pred_dtc = dtc.predict(X_test)

# Hitung akurasi model
accuracy = accuracy_score(y_test, y_pred_dtc)

print("Akurasi Model Tree:", accuracy)
```

```
Akurasi Model Tree: 1.0
```

Logistic Regression



One method commonly used for classification. In the case of classification, logistic regression works by calculating the class probability of a sample.

```
from sklearn import linear_model

logistic = linear_model.LogisticRegression()
logistic.fit(X_train, y_train)

# Lakukan prediksi pada data pengujian
y_pred_logistic = logistic.predict(X_test)

# Hitung akurasi model
accuracy = accuracy_score(y_test, y_pred_logistic)

print("Akurasi Model Logistic Regression:", accuracy)
```

```
Akurasi Model Logistic Regression: 0.9893390191897654
```

Evaluation

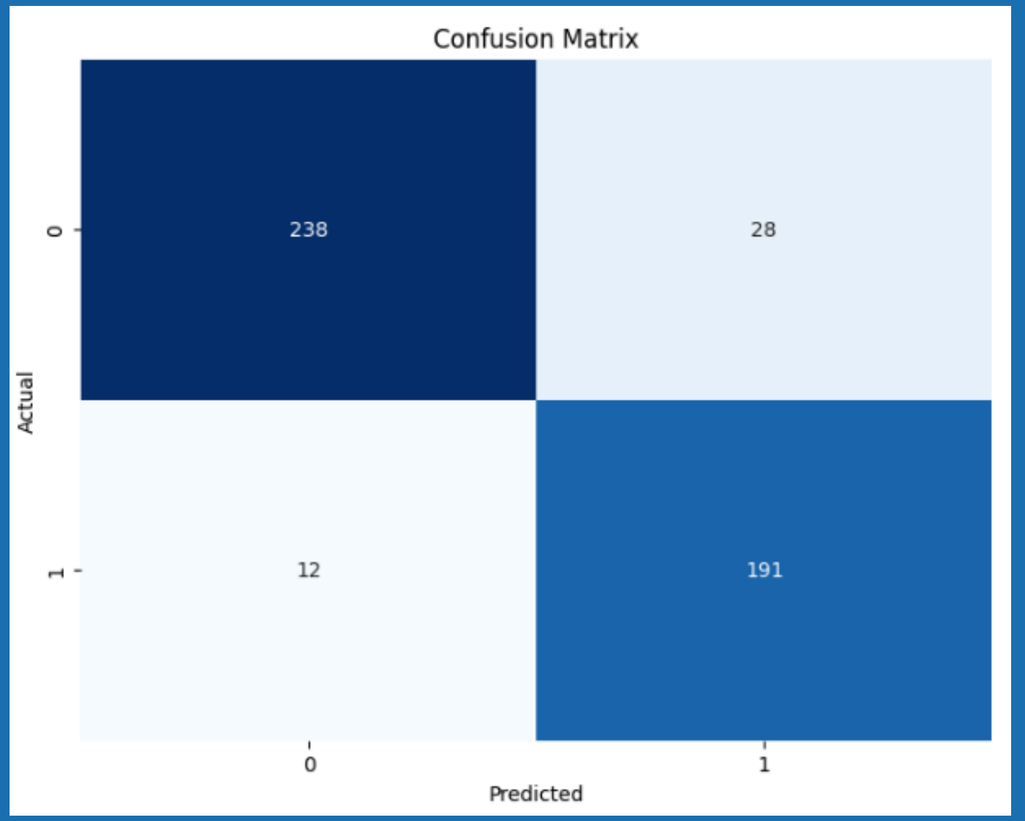
In the process of developing a machine learning model, no all the resulting models can be used immediately. Data Scientists need to carry out testing and evaluation of that model.

Testing and evaluating models is an important task for a Data Scientist to ensure model quality and reliability which was developed. With this task in mind, Data Scientists can confirm decisions and insightstaken from the data can become a solid basis inbusiness decision making or in developing effective solution

Confusion matrix

SVM (Support vector Machine)

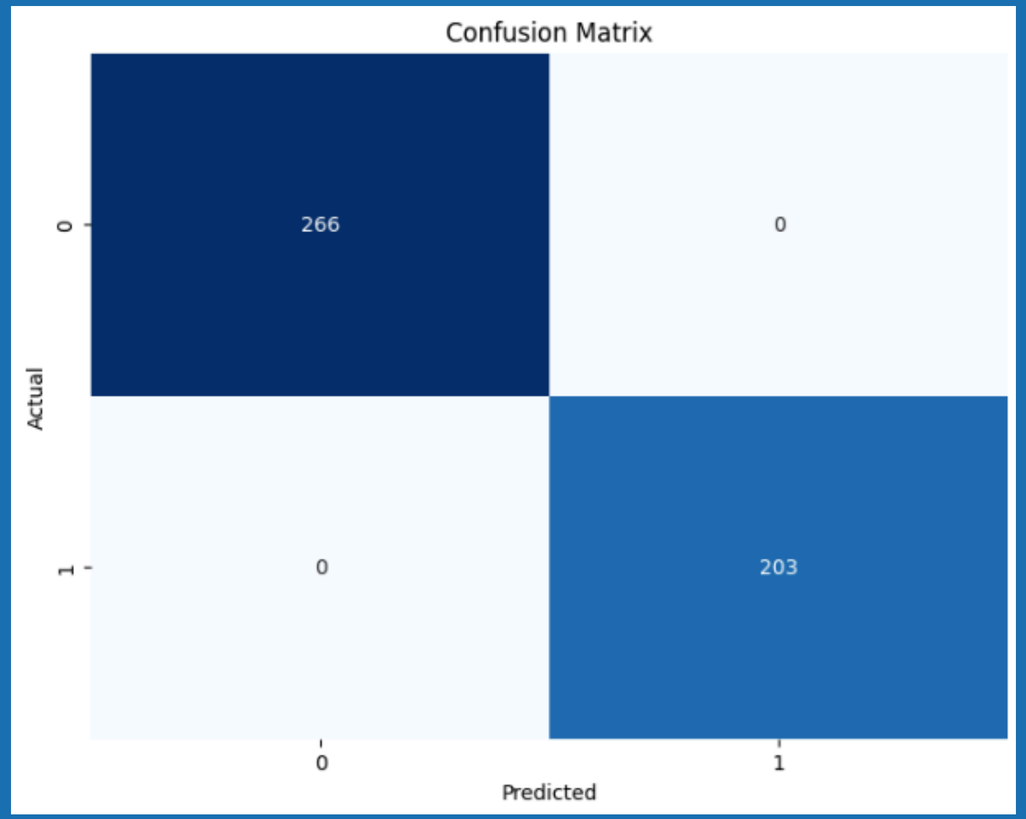
```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

conf_matrix = confusion_matrix(y_test, y_pred_clf)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



A total of 238 data were proven correct in the prediction for label 0
A total of 191 data were proven correct in predictions for label 1

Decion Tree

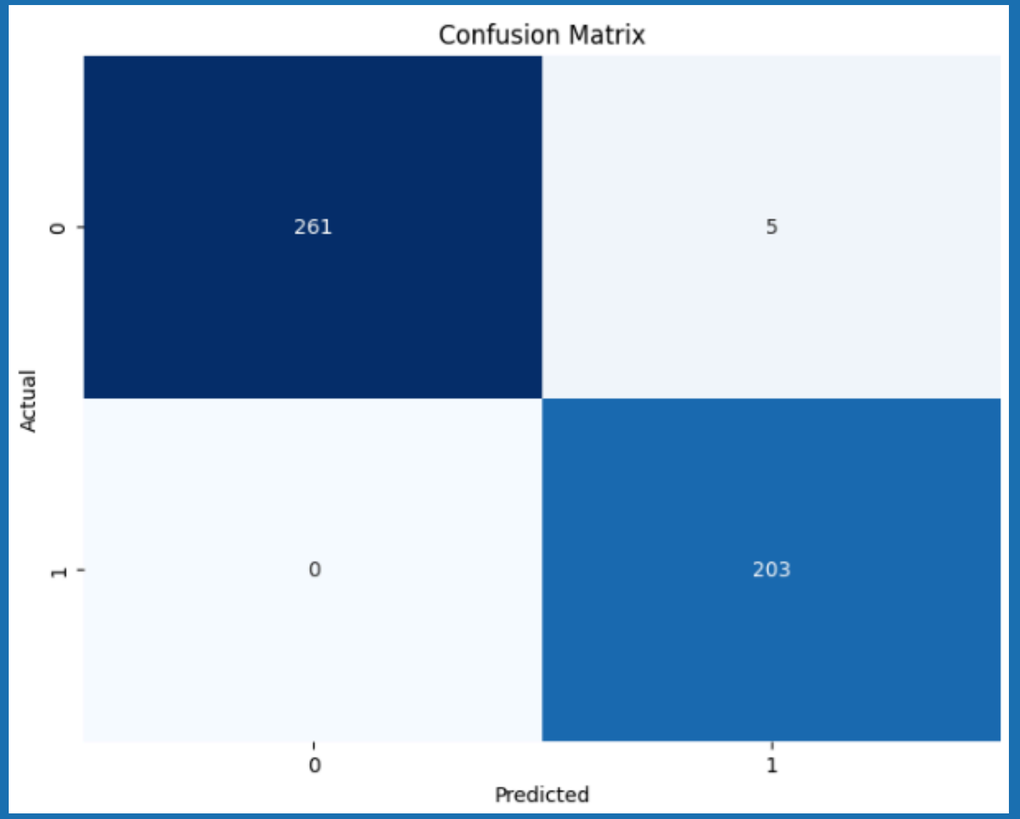
```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

conf_matrix = confusion_matrix(y_test, y_pred_dtc)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



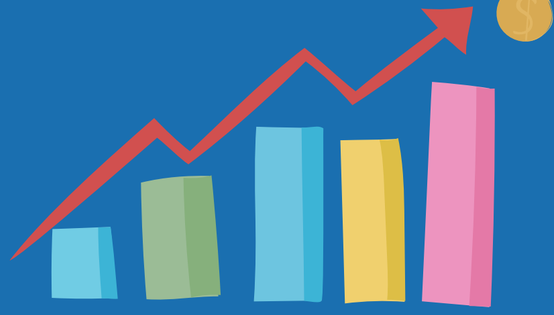
A total of 266 data proved correct in the prediction for label 0
A total of 203 data proved correct in prediction for label 1.
In addition, this machine modeling is the best because this modeling has no wrong guesses.

Logistic Regression

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

conf_matrix = confusion_matrix(y_test, y_pred_logistic)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



A total of 261 data were proven correct in the prediction for label 0
A total of 203 data were proven correct in predictions for label 1



Predicting Anemia Outcome Using Logistic Regression

```python
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder

# Step 1: Memuat dataset dan pra-pemrosesan data
df = pd.read_csv('anemia.csv')

# Memisahkan fitur (X) dan target (y)
X = df.drop('Result', axis=1)  # Fitur-fitur untuk melakukan prediksi
y = df['Result']               # Target atau label yang ingin diprediksi (0 atau 1)

# Step 2: Pelatihan model
model = LogisticRegression()
model.fit(X, y)
```

```python
# Step 3: Fungsi prediksi berdasarkan input
def predict_outcome(input_data):
    input_df = pd.DataFrame([input_data])
    # Prediksi Outcome
    prediction = model.predict(input_df)
    return prediction[0]

# Contoh penginputan data
input_data = {
    'Gender': 1,
    'Hemoglobin': 4.00,
    'MCH': 22.0,
    'MCHC': 30.0,
    'MCV': 3.0
}

predicted_outcome = predict_outcome(input_data)
print(f'Predicted Outcome: {predicted_outcome}')
```

```
Predicted Outcome: 1
```

CONCLUSION

1.By using heatmap correlation analysis, it can be concluded that the most significant factor in determining whether a person suffers from anemia is hemoglobin with a correlation result of -0.80, which means that if a person has a high negative correlation value, this indicates that there is a strong relationship between hemoglobin. and anemia results in this anemia dataset. A negative correlation value means that the higher the hemoglobin level, the possibility of the outcome (anemia) decreasing, and vice versa.

2.From the evaluation results of the three machine learning models used, namely SVM, Decision Tree, and Logistic Regression, it is known that Decision Tree achieved the highest accuracy, namely reaching 1.0. This indicates that the model is very suitable for modeling and predicting data related to anemia.

3. Evaluation results from 33% of the data tested and 67% of the data trained using SVM show that this model was successful in guessing correctly that someone was not infected with anemia (label 0) was 238 data, while those who were infected with anemia (label 1) were 191 data. If you use a decision tree, this model is successful in guessing correctly that someone is not infected with anemia (label 0) is 266 data, while those who are infected with anemia (label 1) are 203 data, and if you use logistic regression, this model is successful in correctly guessing that someone is not infected with anemia (label 0) is 261 data while those infected with anemia (label 1) are 203 data

4.By applying machine learning techniques such as logistic regression, we can utilize information such as gender, hemoglobin, MCH, MCHC, and also MCV to predict whether someone is classified as anemic or not based on the factors given


THANKS YOU