


COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS FOR CLASSIFYING THE IRIS DATASET



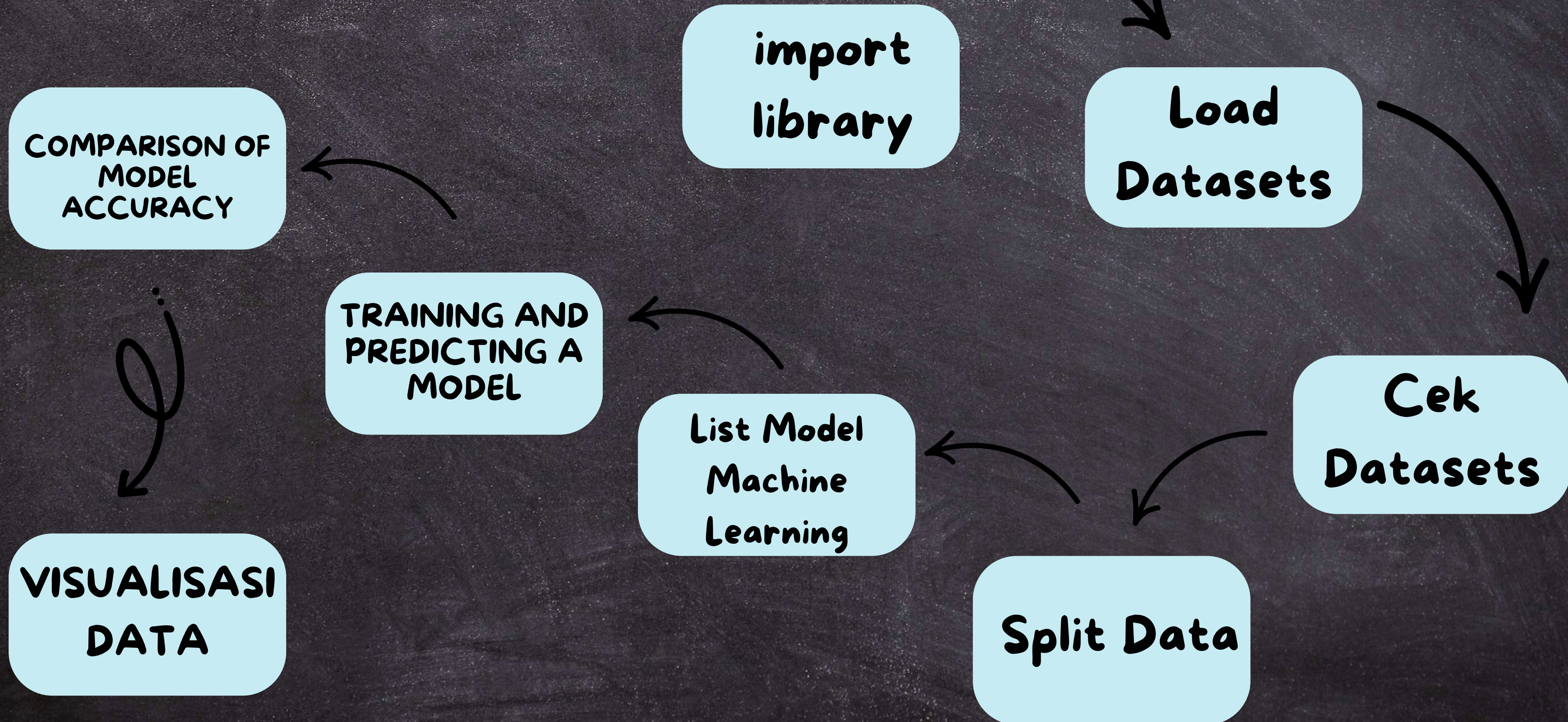
ABOUT ME



I AM A STUDENT IN THE PGSD PROGRAM AT WR SUPRATMAN UNIVERSITY IN SURABAYA. I AM A BEGINNER IN THE FIELD OF DATA SCIENCE AND AM EAGER TO DELVE DEEPER INTO IT, PARTICULARLY IN THE APPLICATION OF DATA ANALYSIS TECHNIQUES AND MACHINE LEARNING TO ENHANCE THE LEARNING PROCESS. WITH MY EDUCATIONAL BACKGROUND, I HOPE TO INTEGRATE DATA SCIENCE KNOWLEDGE INTO EDUCATIONAL PRACTICES AND CONTRIBUTE TO THE DEVELOPMENT OF INNOVATIVE SOLUTIONS THAT CAN SUPPORT MORE EFFECTIVE AND EFFICIENT LEARNING

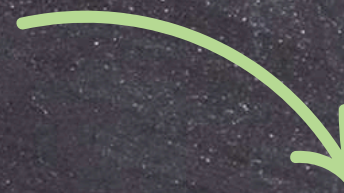


Flow Chart



IMPORT LIBRARY

```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
```



Import library pandas, numpy dan algoritma machine learning yang akan dibandingkan. dalam model ini ada 6 algoritma machine learning:

1. naives bayes
2. random forest
3. Decision tree
4. logistic Regression
5. KNN
6. SVM

hal ini dilakukan untuk melihat model klasifikasi yang paling sesuai udengan iris dataset

Load Datasets

Load dataset untuk mengimpor data yang digunakan untuk analisis. dalam percobaan ini menggunakan load datasets dari scikit-learn

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0
6	4.6	3.4	1.4	0.3	0
7	5.0	3.4	1.5	0.2	0
8	4.4	2.9	1.4	0.2	0
9	4.9	3.1	1.5	0.1	0
10	5.4	3.7	1.5	0.2	0
11	4.8	3.4	1.6	0.2	0
12	4.8	3.0	1.4	0.1	0

Load Datasets

```
# Membuat dataset Iris dari scikit-learn dan mengonversi ke DataFrame
from sklearn.datasets import load_iris
import pandas as pd

# Memuat dataset Iris
iris = load_iris()
x = iris.data    # Merupakan input untuk machine learning
y = iris.target  # Merupakan output yang diinginkan oleh machine learning

# Melakukan konversi data fitur dan target menjadi DataFrame
df_x = pd.DataFrame(x, columns=iris.feature_names)
df_y = pd.Series(y, name='target') # Menggunakan huruf besar pada 'Series'

# Menggabungkan fitur dan target dalam satu DataFrame
df = pd.concat([df_x, df_y], axis=1)

# Menampilkan 13 baris pertama dari dataset
df.head(13)
```


Cek Datasets

`df.info()` adalah metode yang digunakan dalam pustaka pandas di Python untuk memberikan ringkasan informasi tentang DataFrame, seperti jumlah total entri, jumlah kolom, nama kolom, tipe data, jumlah nilai non-null, dan memori yang digunakan



✓ CEK DATASETS

✓
0s

df.info()

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 150 entries, 0 to 149  
Data columns (total 5 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   sepal length (cm)      150 non-null   float64  
1   sepal width (cm)       150 non-null   float64  
2   petal length (cm)      150 non-null   float64  
3   petal width (cm)       150 non-null   float64  
4   target                 150 non-null   int64  
dtypes: float64(4), int64(1)  
memory usage: 6.0 KB
```


SPLIT DATA

membagi data dalam machine learning
biasanya menggunakan `train_test_split` dari
pustaka `scikit-learn`



✓ SPLIT DATA

```
✓ [14] [46] #split dataset menjadi training dan testing  
0s X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=20)
```


LIST MODEL MACHINE LEARNING

list model machine learning untuk menguji
beberapa algoritma sekaligus untuk dibanding mana
yang paling akurat



✓ LIST MODEL MACHINE LEARNING

```
✓ [18] #list model yang akan diuji  
0s models = {  
    "Logistic Regression": LogisticRegression(max_iter=200),  
    "Decision Tree": DecisionTreeClassifier(),  
    "Random Forest": RandomForestClassifier(),  
    "SVM": SVC(),  
    "KNN": KNeighborsClassifier(),  
    "Naive Bayes": GaussianNB()  
}
```


Training and Predicting a Model

✓ MELATIH DAN MEMPREDIKSI MODEL

```
results = []
for name, model in models.items():
    model.fit(X_train, y_train) # Melatih model
    y_pred = model.predict(X_test) # Memprediksi data uji
    acc = accuracy_score(y_test, y_pred)
    results.append((name, acc))
    print(f"Model: {name}")
    print(classification_report(y_test, y_pred, target_names=data.target_names))
    print("-" * 50)
```

Pre

- **Accuracy** : mengetahui seberapa banyak prediksi yang benar dari keseluruhan data
- **Precision**: untuk mengetahui seberapa banyak prediksi positif yang benar dari semua yang diprediksi positif
- **recall**: seberapa banyak data positif yang berhasil diprediksi dengan benar
- **F1-score**: rata-rata harmonik dari precision dan recall
- **support**: untuk mengetahui seberapa banyak data ada dalam setiap dataset

Model: Logistic Regression				
	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	8
versicolor	0.85	1.00	0.92	11
virginica	1.00	0.82	0.90	11
accuracy			0.93	30
macro avg	0.95	0.94	0.94	30
weighted avg	0.94	0.93	0.93	30

Model: Decision Tree				
	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	8
versicolor	0.85	1.00	0.92	11
virginica	1.00	0.82	0.90	11
accuracy			0.93	30
macro avg	0.95	0.94	0.94	30
weighted avg	0.94	0.93	0.93	30

Model: Random Forest				
	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	8
versicolor	0.85	1.00	0.92	11
virginica	1.00	0.82	0.90	11
accuracy			0.93	30
macro avg	0.95	0.94	0.94	30
weighted avg	0.94	0.93	0.93	30

Model: SVM				
	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	8
versicolor	1.00	1.00	1.00	11
virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Model: KNN				
	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	8
versicolor	1.00	0.91	0.95	11
virginica	0.92	1.00	0.96	11
accuracy			0.97	30
macro avg	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

COMPARISON OF MODEL ACCURACY

Untuk memastikan bahwa model yang kita pilih bukan cuma punya akurasi tinggi, tapi juga bisa diandalkan dan efektif sesuai dengan kebutuhan aplikasi yang kita inginkan

✓ PERBANDINGAN AKURASI

```
✓ [20] results_df = pd.DataFrame(results, columns=["M  
0s print("\nPerbandingan Akurasi Model:")  
print(results_df)
```



Perbandingan Akurasi Model:

	Model	Accuracy
0	Logistic Regression	0.933333
1	Decision Tree	0.933333
2	Random Forest	0.933333
3	SVM	1.000000
4	KNN	0.966667
5	Naive Bayes	0.933333

VISUALISASI DATA

untuk menyajikan data dalam bentuk grafik untuk memahami informasi yang terkandung dalam dataset, seperti mengungkap pola, tren, dan hubungan yang mungkin tidak terlihat dalam data mentah

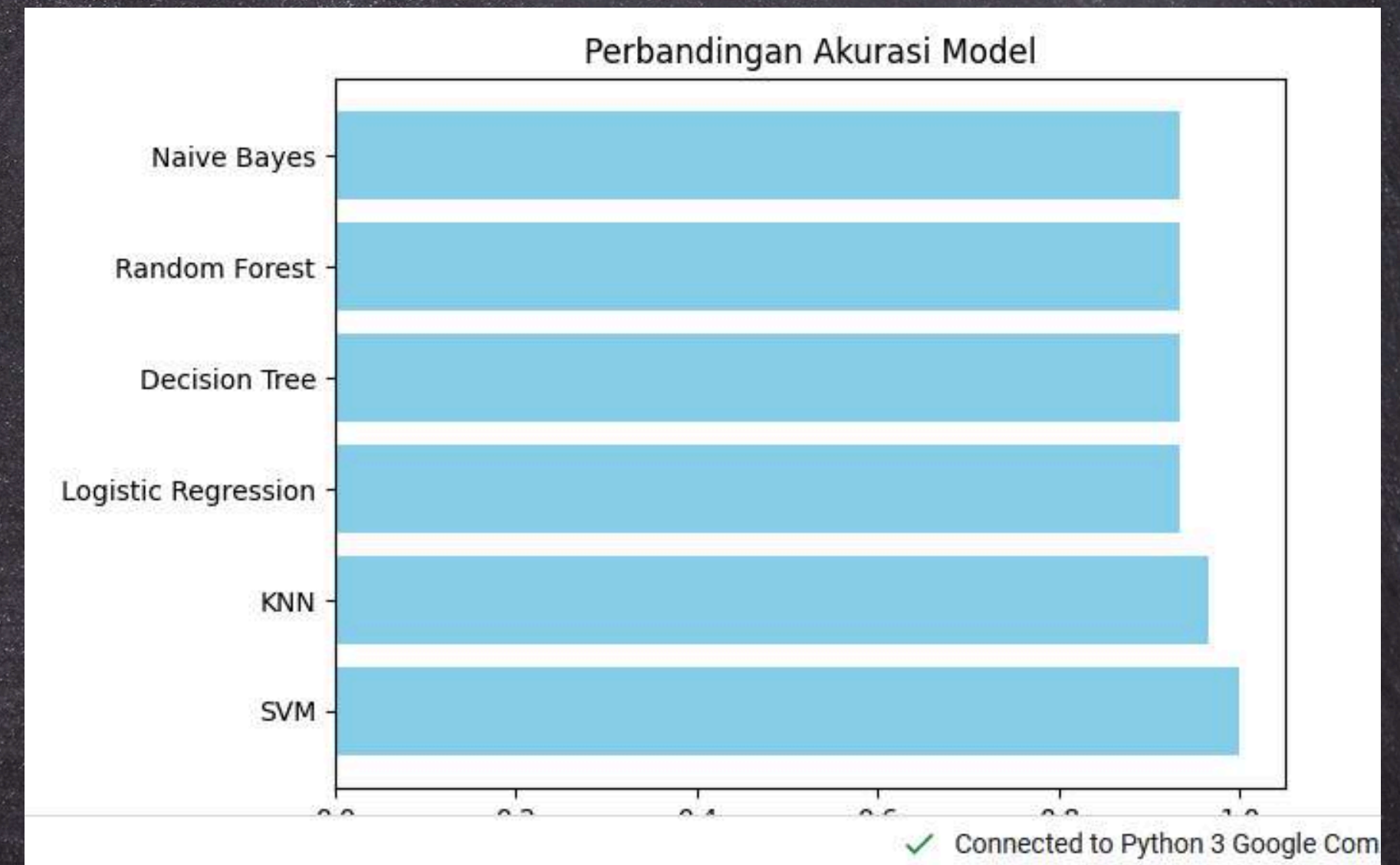


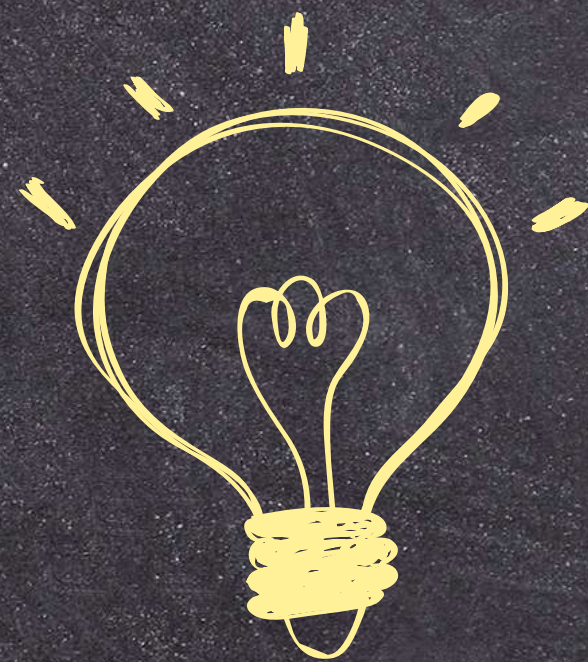
```
Q {x} ✓ 0s VISUALISASI DATA
import matplotlib.pyplot as plt
results_df.sort_values(by="Accuracy", ascending=False, inplace=True)
plt.barh(results_df["Model"], results_df["Accuracy"], color="skyblue")
plt.xlabel("Accuracy")
plt.title("Perbandingan Akurasi Model")
plt.show()
```


VISUALISASI DATA



Berdasarkan hasil perbandingan model machine learning untuk proses klasifikasi dataset iris, dapat disimpulkan bahwa SVM memiliki score paling tinggi diantara yanglain.





TERIMA KASIH