# TUGAS LAB IS388 Data Analysis

# Week 13: Machine Learning - Evaluation and Deployment

#### Semester Ganjil 2024/2025

```
import datetime
import uuid

studentName = "Brightly Virya"
studentNIM = "00000068227"
studentClass = "F"

myDate = datetime.datetime.now()
myDevice = str(uuid.uuid1())

print("Name: \t\t{}".format(studentName))
print("NIM: \t\t{}".format(studentNIM))
print("NIM: \t\t{}".format(studentClass))
print("Start: \t\t{}".format(myDate))
print("Device ID: \t{}".format(myDevice))
```

```
Name:            Brightly Virya
NIM:       00000068227
NIM:       F
Start:           2025-12-04 10:38:31.232172
Device ID:       b41a0c04-d0c2-11f0-96da-ba1afe477a3c
```

---

## Guided Lab

Enter Guided Lab Code Here:

## A simple model using the Iris dataset

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

iris = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-
databases/iris/iris.data', header=None)
iris.columns = ['sepal_length', 'sepal_width', 'petal_length',
'petal_width', 'class']

iris.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  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   class         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
iris.head()
```


```
{"columns":[{"name":"index","rawType":"int64","type":"integer"},
{"name":"sepal_length","rawType":"float64","type":"float"},
{"name":"sepal_width","rawType":"float64","type":"float"},
{"name":"petal_length","rawType":"float64","type":"float"},
{"name":"petal_width","rawType":"float64","type":"float"},
{"name":"class","rawType":"object","type":"string"}],"ref":"36b0f72c-
8e67-421b-b095-db4a09061879","rows":
[["0","5.1","3.5","1.4","0.2","Iris-setosa"],
["1","4.9","3.0","1.4","0.2","Iris-setosa"],
["2","4.7","3.2","1.3","0.2","Iris-setosa"],
["3","4.6","3.1","1.5","0.2","Iris-setosa"],
["4","5.0","3.6","1.4","0.2","Iris-setosa"]],"shape":
{"columns":5,"rows":5}}
```


## Split the data into features and labels

```
X = iris.drop('class', axis=1)
y = iris['class']
```

```
X.head()
```


```
{"columns":[{"name":"index","rawType":"int64","type":"integer"},
{"name":"sepal_length","rawType":"float64","type":"float"},
{"name":"sepal_width","rawType":"float64","type":"float"},
{"name":"petal_length","rawType":"float64","type":"float"},
{"name":"petal_width","rawType":"float64","type":"float"}],"ref":"e397
fad7-bc71-4731-91ed-0578c3436158","rows":
[["0","5.1","3.5","1.4","0.2"],["1","4.9","3.0","1.4","0.2"],
["2","4.7","3.2","1.3","0.2"],["3","4.6","3.1","1.5","0.2"],
["4","5.0","3.6","1.4","0.2"]],"shape":{"columns":4,"rows":5}}
```


```
y.tail()
```


```
{"columns":[{"name":"index","rawType":"int64","type":"integer"},
{"name":"class","rawType":"object","type":"string"}],"ref":"64e95caa-
```

d08d-4f74-8efe-ee2ab68081ee","rows":[["145","Iris-virginica"],
["146","Iris-virginica"],["147","Iris-virginica"],["148","Iris-
virginica"],["149","Iris-virginica"]],"shape":{"columns":1,"rows":5}}

## Split into training and testing sets

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.4, random_state=42)
```

## Create and train the model

```
model = KNeighborsClassifier()
model.fit(X_train, y_train)
```

```
KNeighborsClassifier()
```

## Make predictions

```
predictions = model.predict(X_test)
```

## Evaluate the model

```
accuracy = accuracy_score(y_test, predictions)
print(f'Model Accuracy: {accuracy}')
```

```
Model Accuracy: 0.9833333333333333
```

```
import pickle
```

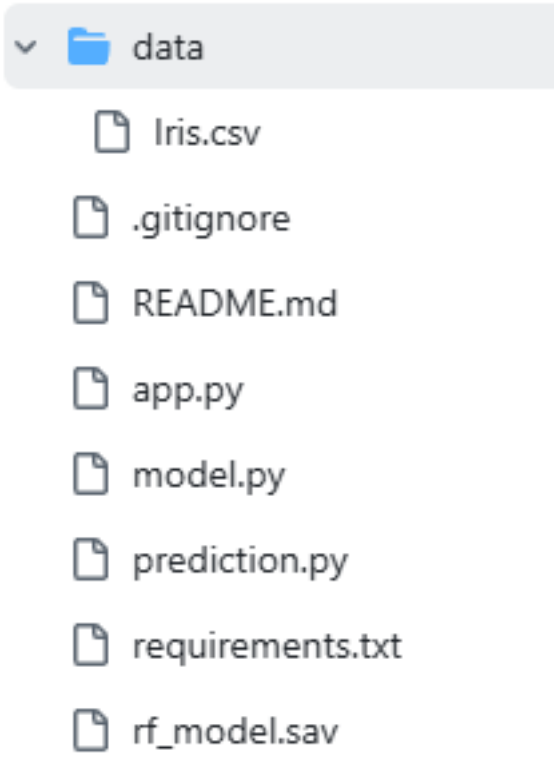
Bisa menyimpan dengan .pkl atau .sav

```
# Simpan model ke file
with open('modelKNN1.pkl', 'wb') as file:
    pickle.dump(model, file)
```

```
import joblib
# save the model to disk
joblib.dump(model, "knn_model.sav")
```

```
['knn_model.sav']
```

Lakukan deploy dari model KNN tersebut diatas (knn_model.sav/modelKNN1.pkl) menggunakan streamlit atau flask. Contoh deploy dengan streamlit seperti dibawah ini.



# Model Deployment - using Streamlit

## model.py

```python
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import joblib

# random seed
seed = 42

# Read original dataset
iris_df = pd.read_csv("data/iris.csv")
iris_df.sample(frac=1, random_state=seed)

# selecting features and target data
X = iris_df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
y = iris_df[['Species']]

# split data into train and test sets
# 70% training and 30% test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=seed, stratify=y)

# create an instance of the random forest classifier
clf = RandomForestClassifier(n_estimators=100)

# train the classifier on the training data
clf.fit(X_train, y_train)

# predict on the test set
y_pred = clf.predict(X_test)

# calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")  # Accuracy: 0.91

# save the model to disk
joblib.dump(clf, "rf_model.sav")
```

## Creating the Streamlit UI

Install Streamlit

pip install streamlit

Go to the "app.py" file and import the Streamlit library

```
import streamlit as st
import pandas as pd
import numpy as np
from prediction import predict


st.title('Classifying Iris Flowers')
st.markdown('Toy model to play to classify iris flowers into \
     (setosa, versicolor, virginica) based on their sepal/petal \
    and length/width.')

st.header("Plant Features")
col1, col2 = st.columns(2)

with col1:
    st.text("Sepal characteristics")
    sepal_l = st.slider('Sepal lenght (cm)', 1.0, 8.0, 0.5)
    sepal_w = st.slider('Sepal width (cm)', 2.0, 4.4, 0.5)

with col2:
    st.text("Pepal characteristics")
    petal_l = st.slider('Petal lenght (cm)', 1.0, 7.0, 0.5)
    petal_w = st.slider('Petal width (cm)', 0.1, 2.5, 0.5)

st.text('')
if st.button("Predict type of Iris"):
    result = predict(
        np.array([[sepal_l, sepal_w, petal_l, petal_w]]))
    st.text(result[0])


st.text('')
st.text('')
st.markdown(
```

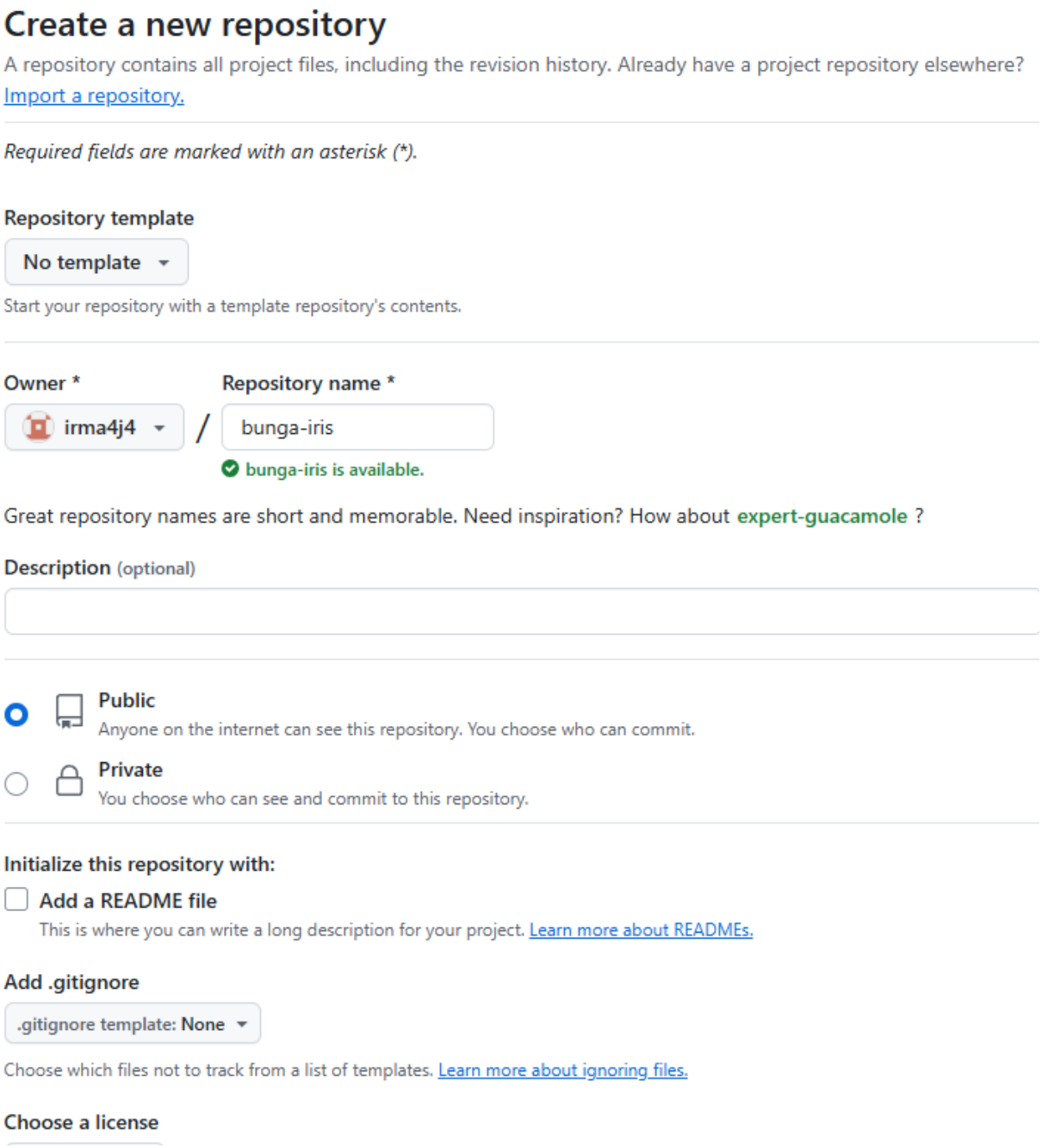
prediction.py

```python
import joblib


def predict(data):
    clf = joblib.load("rf_model.sav")
    return clf.predict(data)

```

Step 1: Create a requeriments.txt file at the root of your folder with the libraries that we used

joblib==0.14.1 streamlit==1.7.0 scikit-learn==0.23.1 pandas==1.0.5

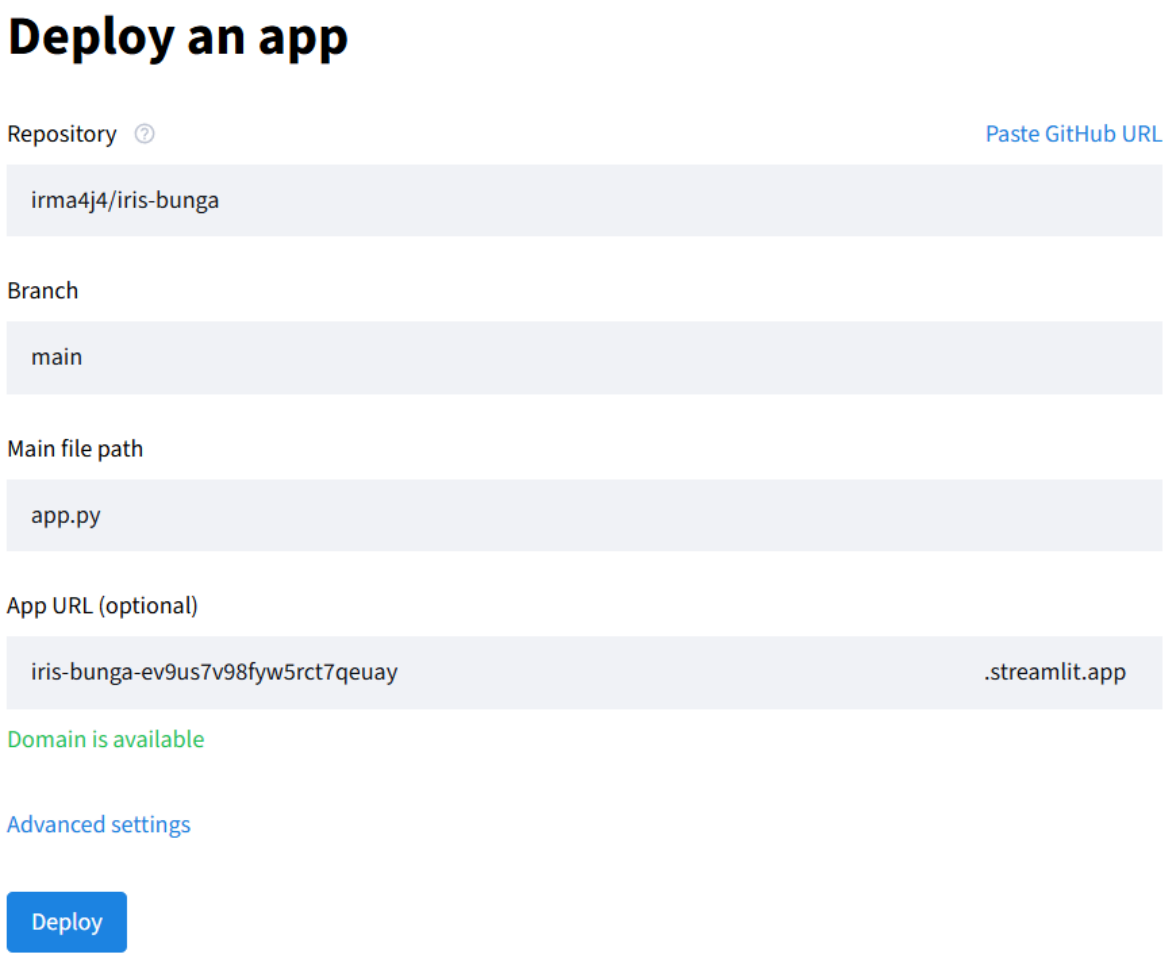
Step 2: Create a GitHub repository and push your code



Step 3: Create a Streamlit account and connect your GitHub profile to it

Step 4: Now, on the Streamlit dashboard click the “New app” button

Step 5: Link your Streamlit app with your GitHub repository:




Step 6: Click “Deploy”

# Conclusion

## Enter Your Conclusion Here:

praktikum minggu ke-13 ini fokus belajar evaluasi dan deployment model machine learning pakai dataset iris. intinya kita latihan bikin model k-nearest neighbors atau knn, membagi data training dan testing, sampai dapat akurasi model sekitar 98 persen. setelah modelnya jadi, kita simpan ke format pkl atau sav biar bisa dipakai lagi tanpa training ulang. terakhir kita belajar

deploy model itu jadi aplikasi web sederhana menggunakan streamlit dan mengunggah projectnya ke github sebagai portofolio.

---

```
myDate = datetime.datetime.now()
myDevice = str(uuid.uuid1())

print("Name: \t\t{}".format(studentName))
print("NIM: \t\t{}".format(studentNIM))
print("NIM: \t\t{}".format(studentClass))
print("End: \t\t{}".format(myDate))
print("Device ID: \t{}".format(myDevice))
```

```
Name:            Brightly Virya
NIM:        00000068227
NIM:        F
End:        2025-12-04 10:38:36.481824
Device ID:       b73b1176-d0c2-11f0-96da-ba1afe477a3c
```

---

# Reference

## Input Your Reference Here (Jika ada):

---