IRIS FLOWER CLASSIFICATION PROJECT

PART 4 — Model Building, Prediction and Evaluation

```
import pandas as pd

from sklearn.datasets import load_iris

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

```
iris = load_iris()

df = pd.DataFrame(
iris.data,
columns=iris.feature_names
)

df["species"]=iris.target

df["species"]=df["species"].map({
0:"setosa",
1:"versicolor",
2:"virginica"
})

df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Next steps: Generate code with df | New interactive sheet

The Iris dataset is loaded using scikit-learn library. The dataset contains flower measurements used to train classification models.

```
X=df.drop("species",axis=1)

y=df["species"]
```

```
X_train,X_test,y_train,y_test = train_test_split(
X,
y,
test_size=0.3,
random_state=42
)
```

The dataset is divided into 70% training data and 30% testing data to evaluate model performance on unseen data.

```
lr = LogisticRegression(max_iter=200)

lr.fit(X_train,y_train)

pred_lr = lr.predict(X_test)

accuracy_lr = accuracy_score(y_test,pred_lr)

accuracy_lr
```

```
1.0
```

Logistic Regression learns linear relationships between flower measurements and species classes.

```
knn = KNeighborsClassifier(n_neighbors=5)

knn.fit(X_train,y_train)

pred_knn = knn.predict(X_test)

accuracy_knn = accuracy_score(y_test,pred_knn)

accuracy_knn
```

```
1.0
```

KNN classifies flowers based on similarity with nearest neighboring samples.

```
dt = DecisionTreeClassifier()

dt.fit(X_train,y_train)

pred_dt = dt.predict(X_test)

accuracy_dt = accuracy_score(y_test,pred_dt)

accuracy_dt
```

```
1.0
```

```
Decision Tree creates rule-based splits to classify species.
```

```
confusion_matrix(y_test,pred_lr)
```

```
array([[19,  0,  0],
       [ 0, 13,  0],
       [ 0,  0, 13]])
```

Confusion matrix shows correct and incorrect predictions for each species.

```
print("Logistic Regression Accuracy:",accuracy_lr)

print("KNN Accuracy:",accuracy_knn)

print("Decision Tree Accuracy:",accuracy_dt)
```

```
Logistic Regression Accuracy: 1.0
KNN Accuracy: 1.0
Decision Tree Accuracy: 1.0
```

Model Comparison:

Logistic Regression provides high accuracy with simple interpretation.

KNN performs well because Iris species are separated by distance-based measurements.

Decision Tree is easy to understand but may sometimes overfit data.

KNN or Logistic Regression generally perform best for this dataset.