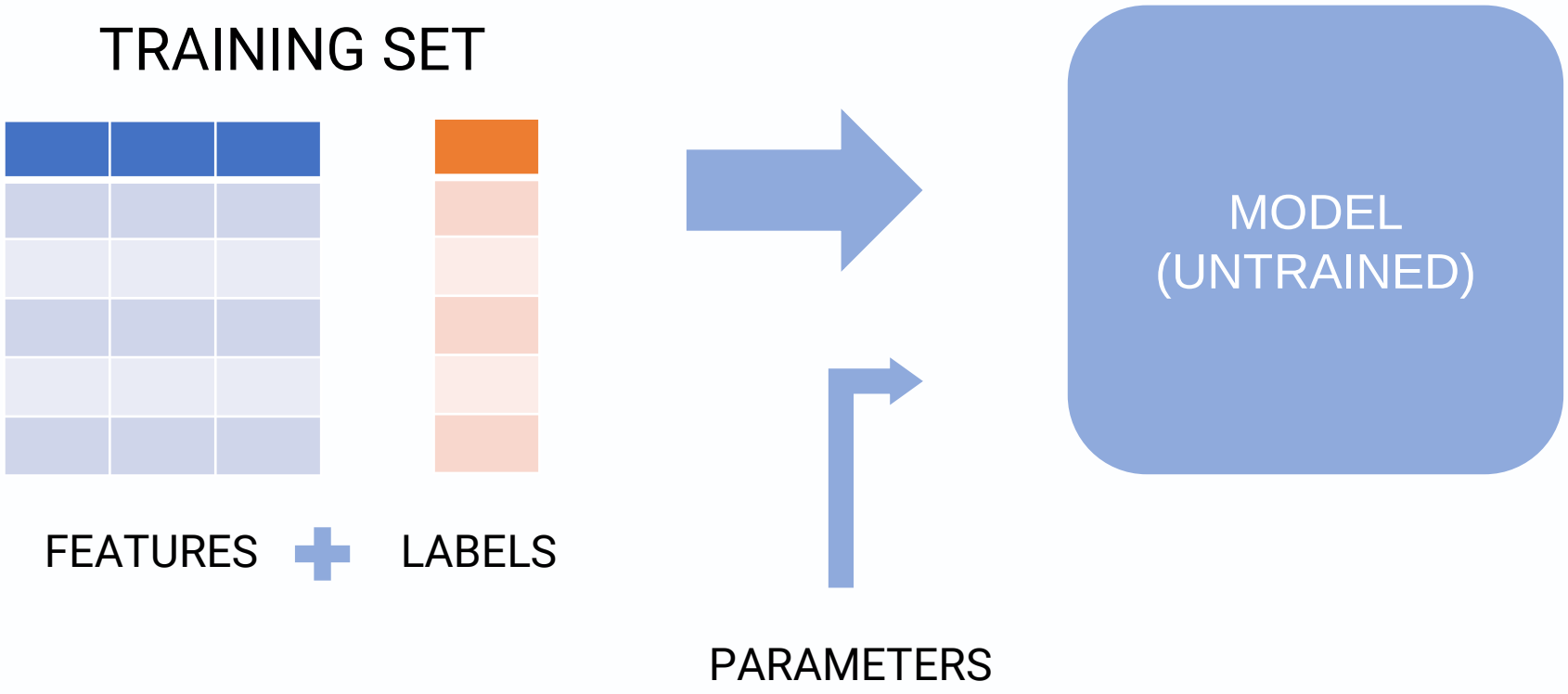
DECISION TREE

with Python Scikit-Learn

February 2023

Review

TRAINING THE MODEL



Training The Model (Decision Tree)

1. Import the model from a package (sklearn)
2. Create an instance of the model
3. Train the model (often using the `fit()` function)

```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier, export_text, plot_tree
```

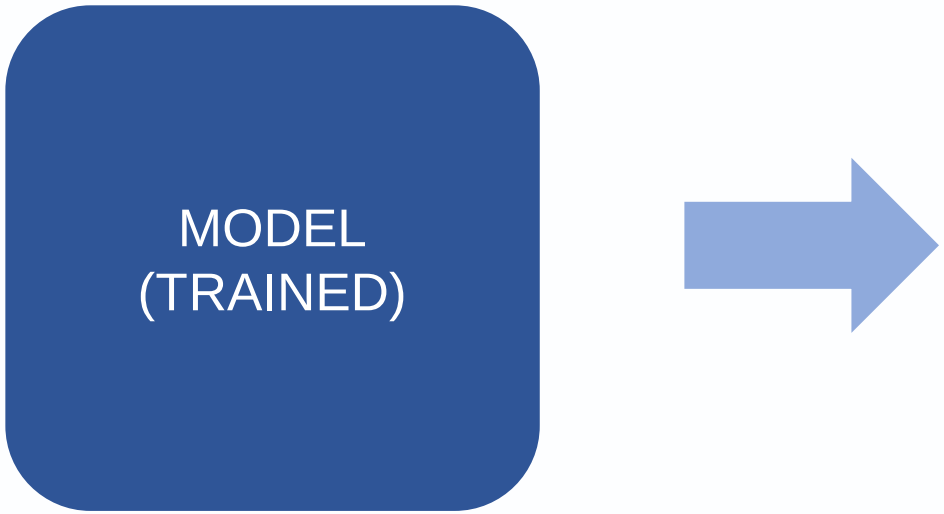
...

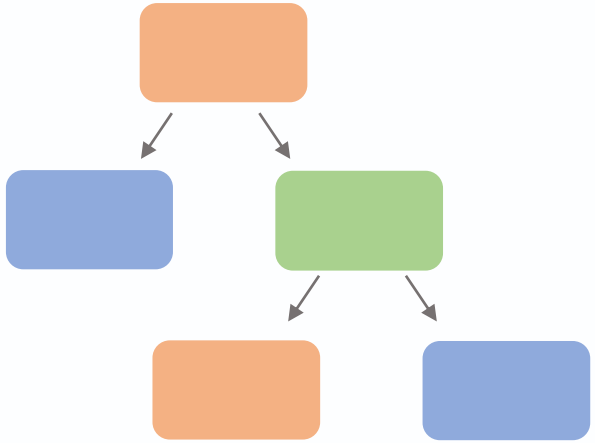
```
# Fit the decision tree model to the iris data
dt_classifier = tree.DecisionTreeClassifier()
dt_classifier = dt_classifier.fit(iris_train_df, iris_train_y)
```

DECISION TREE

GENERATE TREE

MODEL
(TRAINED)





Generate Tree

Decision Tree Visualization:

- Trees from trained Decision Tree models can be extracted
  - `export_text(model, feature_names=features_list)`
- Output: string representing the tree

```
# Visualize the tree as a text
tree_text = export_text(dt_classifier, feature_names=iris_features)
print(tree_text)
```

Generate Tree

Decision Tree Visualization:

```
|--- petal length (cm) <= 2.45
|   |--- class: 0
|--- petal length (cm) >  2.45
|   |--- petal width (cm) <= 1.65
|   |   |--- petal length (cm) <= 4.95
|   |   |   |--- class: 1
|   |   |--- petal length (cm) >  4.95
|   |   |   |--- sepal length (cm) <= 6.05
|   |   |   |   |--- sepal width (cm) <= 2.45
|   |   |   |   |   |--- class: 2
|   |   |   |   |--- sepal width (cm) >  2.45
|   |   |   |   |   |--- class: 1
|   |   |   |--- sepal length (cm) >  6.05
|   |   |   |   |--- class: 2
|   |--- petal width (cm) >  1.65
|   |   |--- class: 2
```

Generate Tree

Decision Tree Visualization:

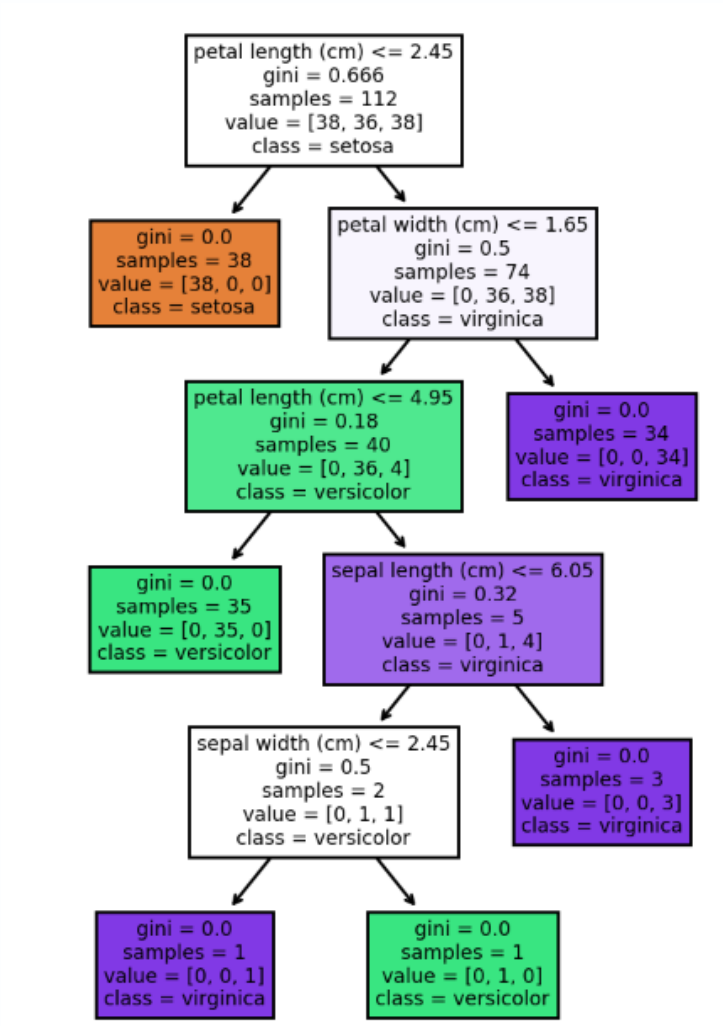
- Can be plotted as well using matplotlib

```
# Visualize the tree as a plot
fig, axes = plt.subplots(nrows=1, ncols=1, figsize=(4, 6), dpi=200)
plot_tree(dt_classifier, filled=True, feature_names=iris_features, class_names=iris_classes, ax=axes)
plt.show()
```

Generate Tree

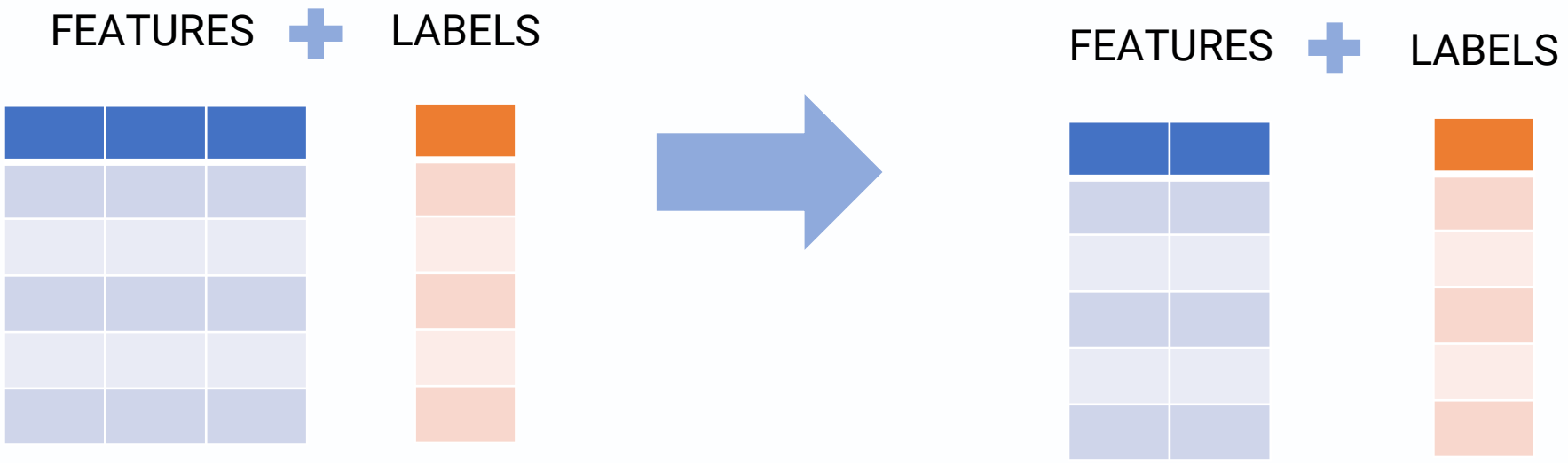
Decision Tree Visualization:

- Can be plotted as well using matplotlib



Overview

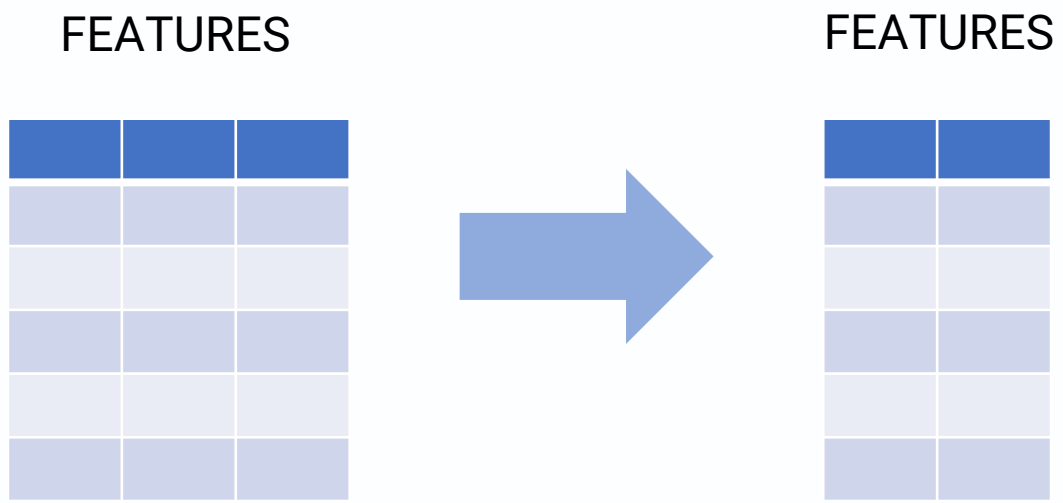
REDUCING FEATURES



Reducing Features

Remove Columns

- We can remove column/s and use the new dataframe to train a new decision tree model and compare trees they generate



Reducing Features

Remove Columns

- Insert list of feature names that will be included in the new dataframe as demonstrated below:

```
iris_df = iris_df[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)']]
```

Reducing Features

Original Iris Dataframe Features:

```
iris_df
```

3]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
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

Reducing Features

New Iris Dataframe Features:

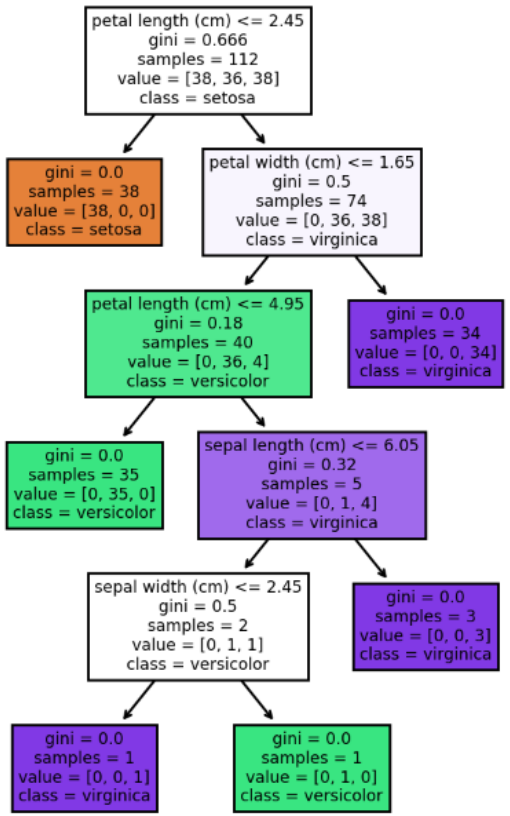
```
iris_df = iris_df[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)']]
iris_df
```

8]:

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

Reducing Features

ORIGINAL TREE



NEW TREE