from matplotlib import pyplot as plt

from sklearn import datasets

from sklearn.tree import DecisionTreeClassifier

from sklearn import tree

# Prepare the data data

iris = datasets.load\_iris()

X = iris.data

y = iris.target

# Fit the classifier with default hyper-parameters

clf = DecisionTreeClassifier(random\_state=1234)

model = clf.fit(X, y)

# 1

text\_representation = tree.export\_text(clf)

print(text\_representation)

# if you want to save the tree...

with open("decistion\_tree.log", "w") as fout:

fout.write(text\_representation)

# 2

fig = plt.figure(figsize=(25,20))

\_ = tree.plot\_tree(clf,

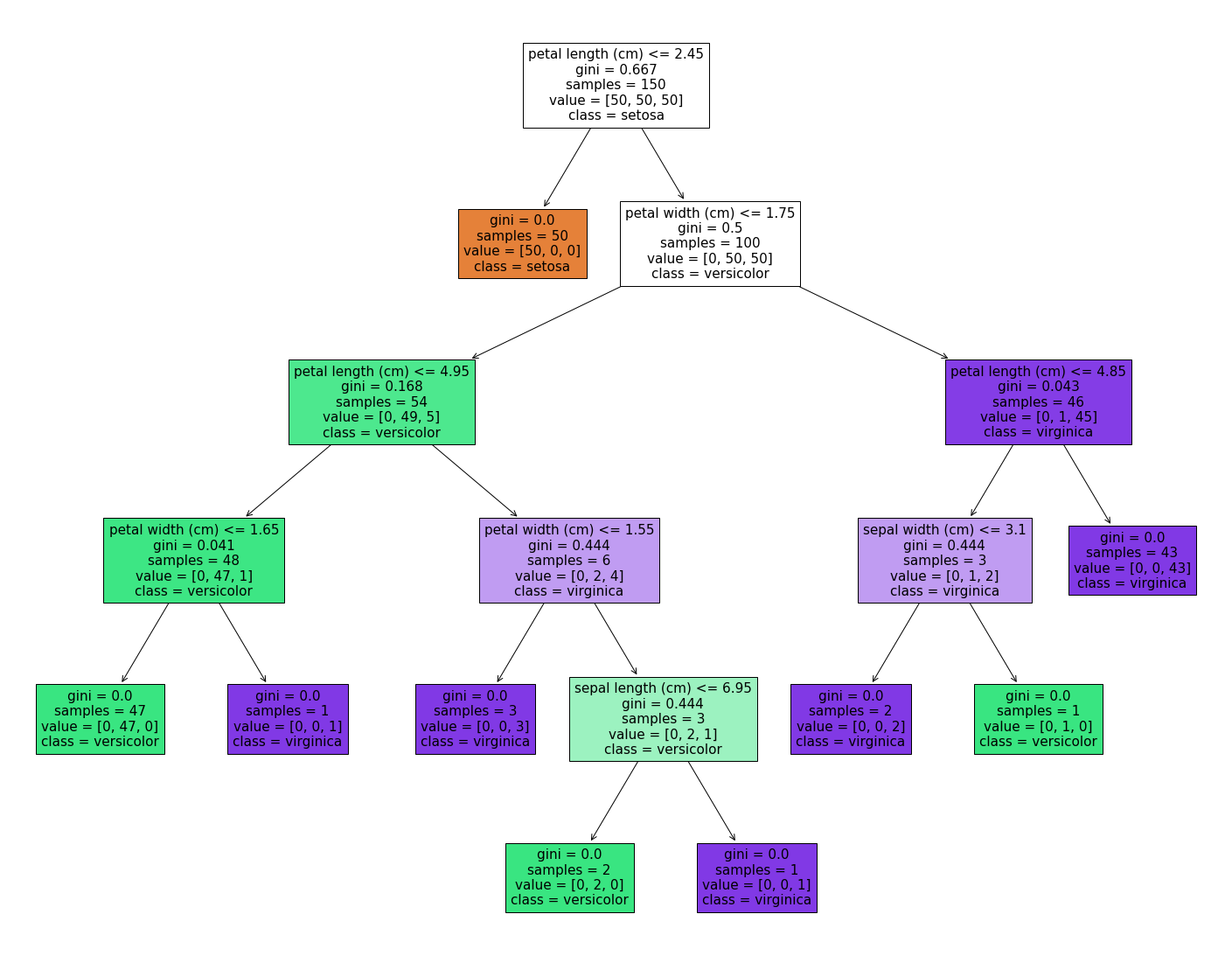
feature\_names=iris.feature\_names,

class\_names=iris.target\_names,

filled=True)

# if you want to save the tree...

fig.savefig("decistion\_tree.png")



# 3

import graphviz

# DOT data

dot\_data = tree.export\_graphviz(clf, out\_file='C:\\Users\\ryans\\OneDrive\\Desktop\\Briefcase\\PDFs\\1-ALL PYTHON & R CODE SAMPLES\\Decision Tree Visualized\\tree.dot',

feature\_names=iris.feature\_names,

class\_names=iris.target\_names,

filled=True)

from sklearn import datasets

from sklearn.tree import DecisionTreeRegressor

from sklearn import tree

# Prepare the data data

boston = datasets.load\_boston()

X = boston.data

y = boston.target

# To keep the size of the tree small, I set max\_depth = 3.

# Fit the regressor, set max\_depth = 3

regr = DecisionTreeRegressor(max\_depth=3, random\_state=1234)

model = regr.fit(X, y)

# 1

text\_representation = tree.export\_text(regr)

print(text\_representation)

fig = plt.figure(figsize=(25,20))

\_ = tree.plot\_tree(regr, feature\_names=boston.feature\_names, filled=True)

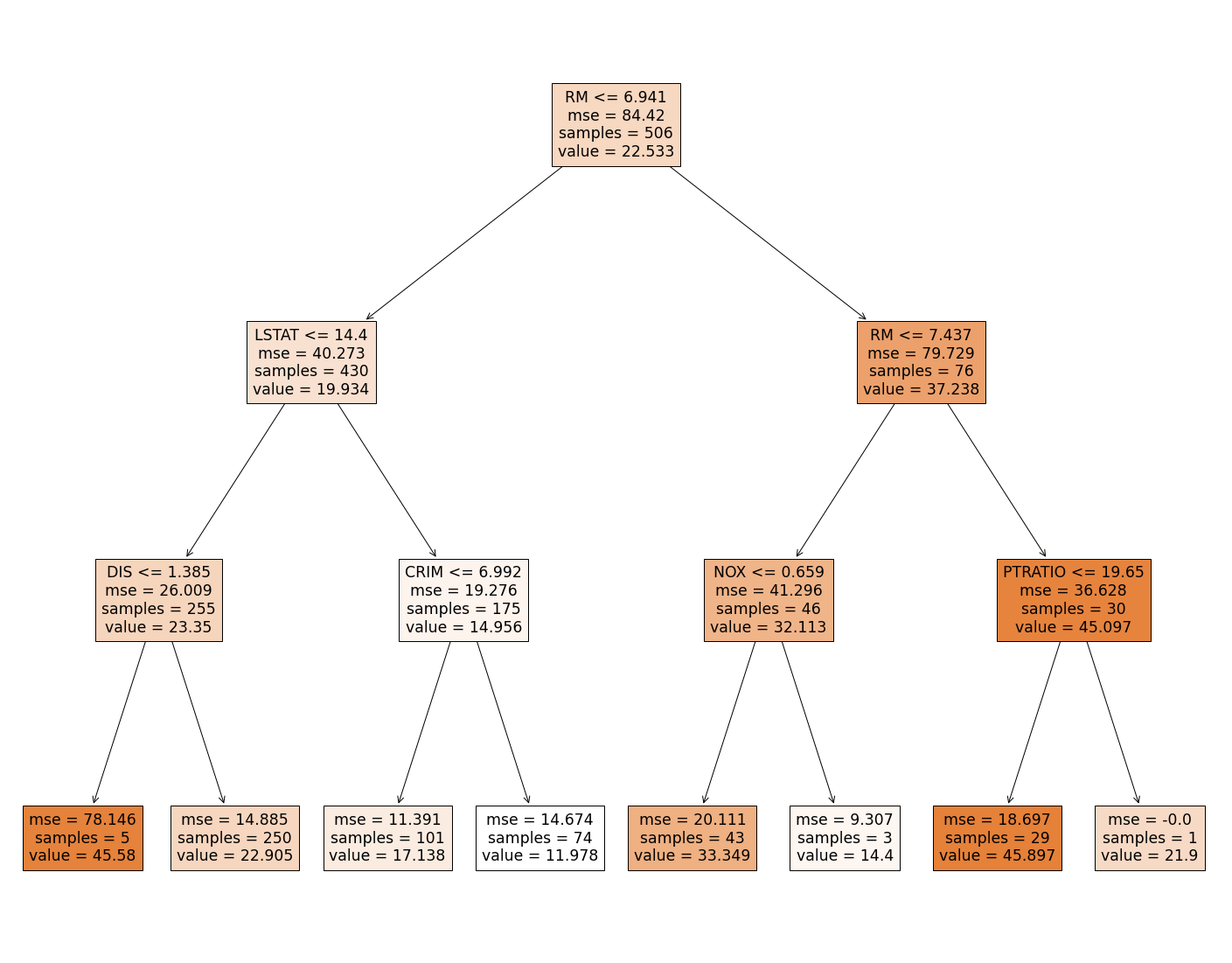
# 2

fig = plt.figure(figsize=(25,20))

\_ = tree.plot\_tree(regr,

feature\_names=boston.feature\_names,

filled=True)



# <https://mljar.com/blog/visualize-decision-tree/>