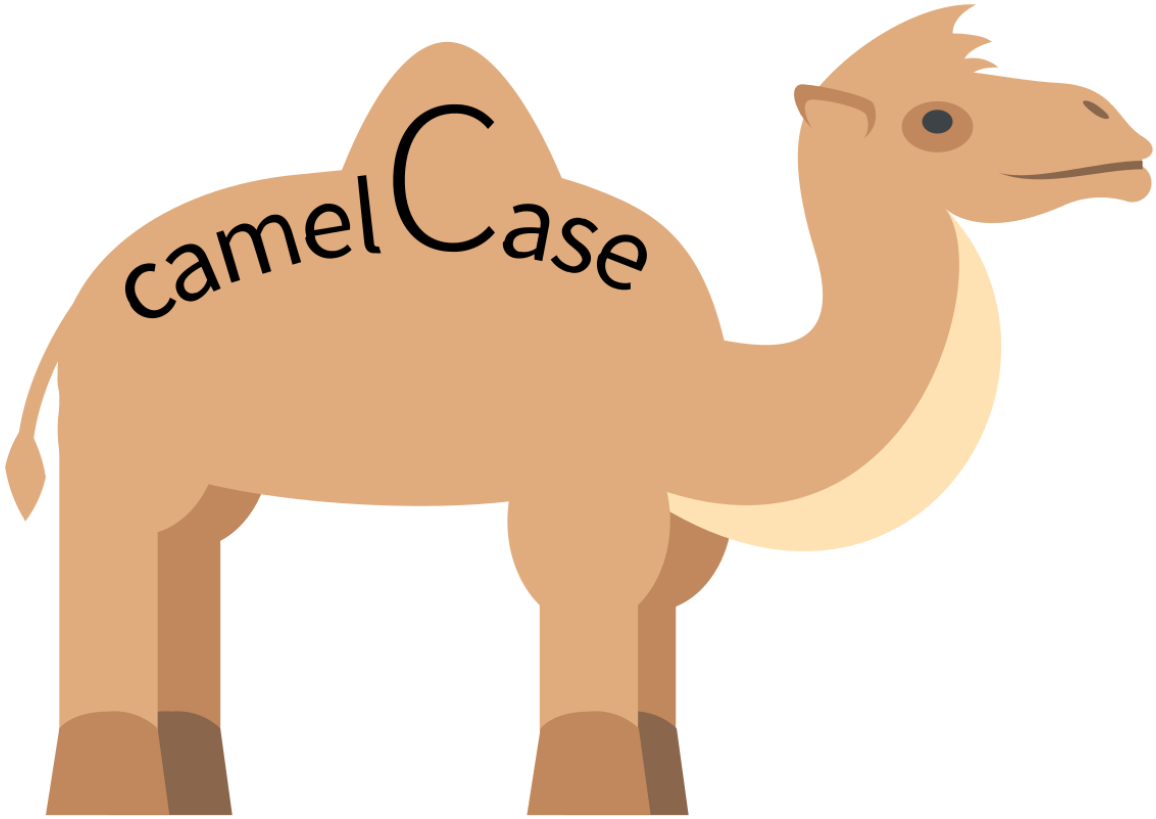
Decession tree classifier

Shivam

1. Shivam
2. pooja
3. preeti



In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
from sklearn.datasets import load_iris
```

In [3]:
```
df = load_iris()
```

In [4]:
```
x = df.data
y = df.target
```

In [5]:
```
x,y
```

Out[5]:
```
(array([[5.1, 3.5, 1.4, 0.2],
        [4.9, 3. , 1.4, 0.2],
        [4.7, 3.2, 1.3, 0.2],
        [4.6, 3.1, 1.5, 0.2],
        [5. , 3.6, 1.4, 0.2],
        [5.4, 3.9, 1.7, 0.4],
        [4.6, 3.4, 1.4, 0.3],
        [5. , 3.4, 1.5, 0.2],
        [4.4, 2.9, 1.4, 0.2],
        [4.9, 3.1, 1.5, 0.1],
        [5.4, 3.7, 1.5, 0.2],
        [4.8, 3.4, 1.6, 0.2],
        [4.8, 3. , 1.4, 0.1],
        [4.3, 3. , 1.1, 0.1],
        [5.8, 4. , 1.2, 0.2],
        [5.7, 4.4, 1.5, 0.4],
        [5.4, 3.9, 1.3, 0.4],
        [5.1, 3.5, 1.4, 0.3],
        [5.7, 3.8, 1.7, 0.3],
```

In [6]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_s
```

In [7]:
```
# check null values
np.isnan(x).sum()
```

Out[7]:
```
0
```

In [8]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [9]:
```
dt = DecisionTreeClassifier()
```

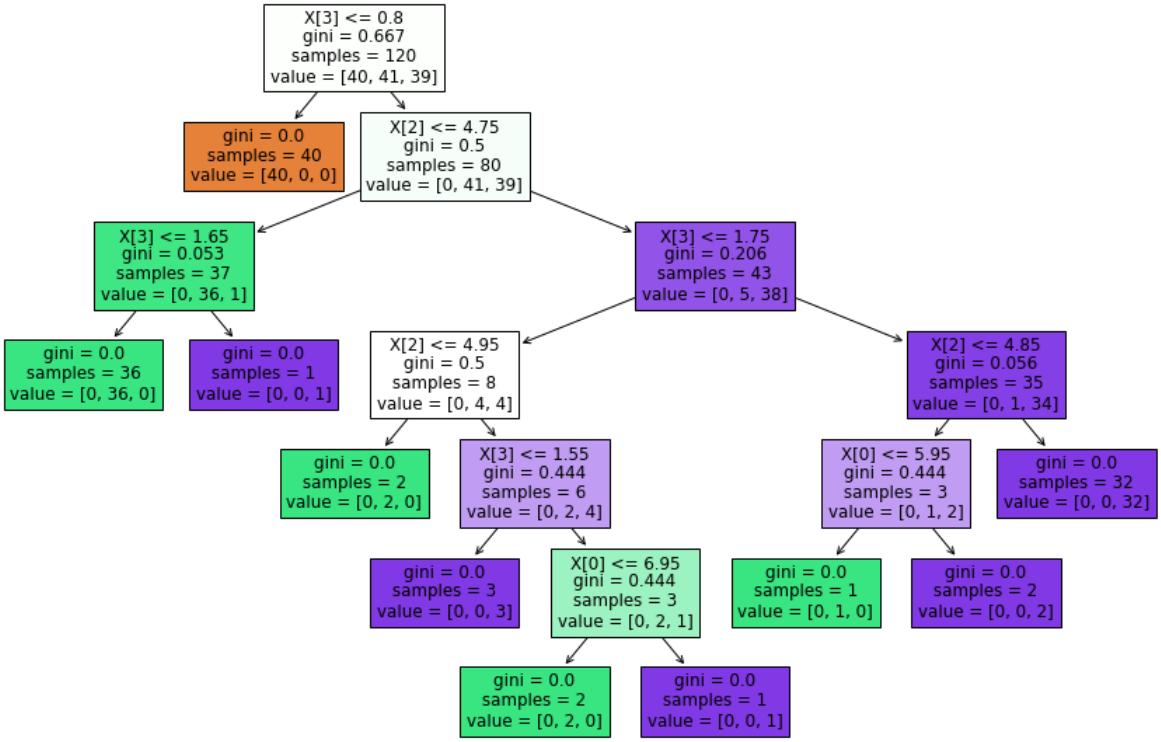
In [10]:
```
dt.fit(X_train, y_train)
```

Out[10]:
```
DecisionTreeClassifier()
```

In [11]:
```
from sklearn import tree
```

In [12]:
```
plt.figure(figsize=(15,10))
tree.plot_tree(dt,filled=True)
```

Out[12]:
```
[Text(0.3076923076923077, 0.9285714285714286, 'X[3] <= 0.8\ngini = 0.667\nsampl
es = 120\nvalue = [40, 41, 39]'),
 Text(0.23076923076923078, 0.7857142857142857, 'gini = 0.0\nsamples = 40\nvalue
= [40, 0, 0]'),
 Text(0.38461538461538464, 0.7857142857142857, 'X[2] <= 4.75\ngini = 0.5\nsampl
es = 80\nvalue = [0, 41, 39]'),
 Text(0.15384615384615385, 0.6428571428571429, 'X[3] <= 1.65\ngini = 0.053\nsam
ples = 37\nvalue = [0, 36, 1]'),
 Text(0.07692307692307693, 0.5, 'gini = 0.0\nsamples = 36\nvalue = [0, 36,
0]'),
 Text(0.23076923076923078, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(0.6153846153846154, 0.6428571428571429, 'X[3] <= 1.75\ngini = 0.206\nsamp
les = 43\nvalue = [0, 5, 38]'),
 Text(0.38461538461538464, 0.5, 'X[2] <= 4.95\ngini = 0.5\nsamples = 8\nvalue =
[0, 4, 4]'),
 Text(0.3076923076923077, 0.35714285714285715, 'gini = 0.0\nsamples = 2\nvalue
= [0, 2, 0]'),
 Text(0.46153846153846156, 0.35714285714285715, 'X[3] <= 1.55\ngini = 0.444\nsa
mples = 6\nvalue = [0, 2, 4]'),
 Text(0.38461538461538464, 0.21428571428571427, 'gini = 0.0\nsamples = 3\nvalue
= [0, 0, 3]'),
 Text(0.5384615384615384, 0.21428571428571427, 'X[0] <= 6.95\ngini = 0.444\nsam
ples = 3\nvalue = [0, 2, 1]'),
 Text(0.46153846153846156, 0.07142857142857142, 'gini = 0.0\nsamples = 2\nvalue
= [0, 2, 0]'),
 Text(0.6153846153846154, 0.07142857142857142, 'gini = 0.0\nsamples = 1\nvalue
= [0, 0, 1]'),
 Text(0.8461538461538461, 0.5, 'X[2] <= 4.85\ngini = 0.056\nsamples = 35\nvalue
= [0, 1, 34]'),
 Text(0.7692307692307693, 0.35714285714285715, 'X[0] <= 5.95\ngini = 0.444\nsam
ples = 3\nvalue = [0, 1, 2]'),
 Text(0.6923076923076923, 0.21428571428571427, 'gini = 0.0\nsamples = 1\nvalue
= [0, 1, 0]'),
 Text(0.8461538461538461, 0.21428571428571427, 'gini = 0.0\nsamples = 2\nvalue
= [0, 0, 2]'),
 Text(0.9230769230769231, 0.35714285714285715, 'gini = 0.0\nsamples = 32\nvalue
= [0, 0, 32]')]
```


X[3] <= 0.8
gini = 0.667
samples = 120
value = [40, 41, 39]
gini = 0.0
samples = 40
value = [40, 0, 0]
X[2] <= 4.75
gini = 0.5
samples = 80
value = [0, 41, 39]
X[3] <= 1.65
gini = 0.053
samples = 37
value = [0, 36, 1]
X[3] <= 1.75
gini = 0.206
samples = 43
value = [0, 5, 38]
gini = 0.0
samples = 36
value = [0, 36, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
X[2] <= 4.95
gini = 0.5
samples = 8
value = [0, 4, 4]
X[2] <= 4.85
gini = 0.056
samples = 35
value = [0, 1, 34]
gini = 0.0
samples = 2
value = [0, 2, 0]
X[3] <= 1.55
gini = 0.444
samples = 6
value = [0, 2, 4]
X[0] <= 5.95
gini = 0.444
samples = 3
value = [0, 1, 2]
gini = 0.0
samples = 32
value = [0, 0, 32]
gini = 0.0
samples = 3
value = [0, 0, 3]
X[0] <= 6.95
gini = 0.444
samples = 3
value = [0, 2, 1]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 2
value = [0, 0, 2]
gini = 0.0
samples = 2
value = [0, 2, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]

In [13]:
```
!pip install  dtreeviz
```

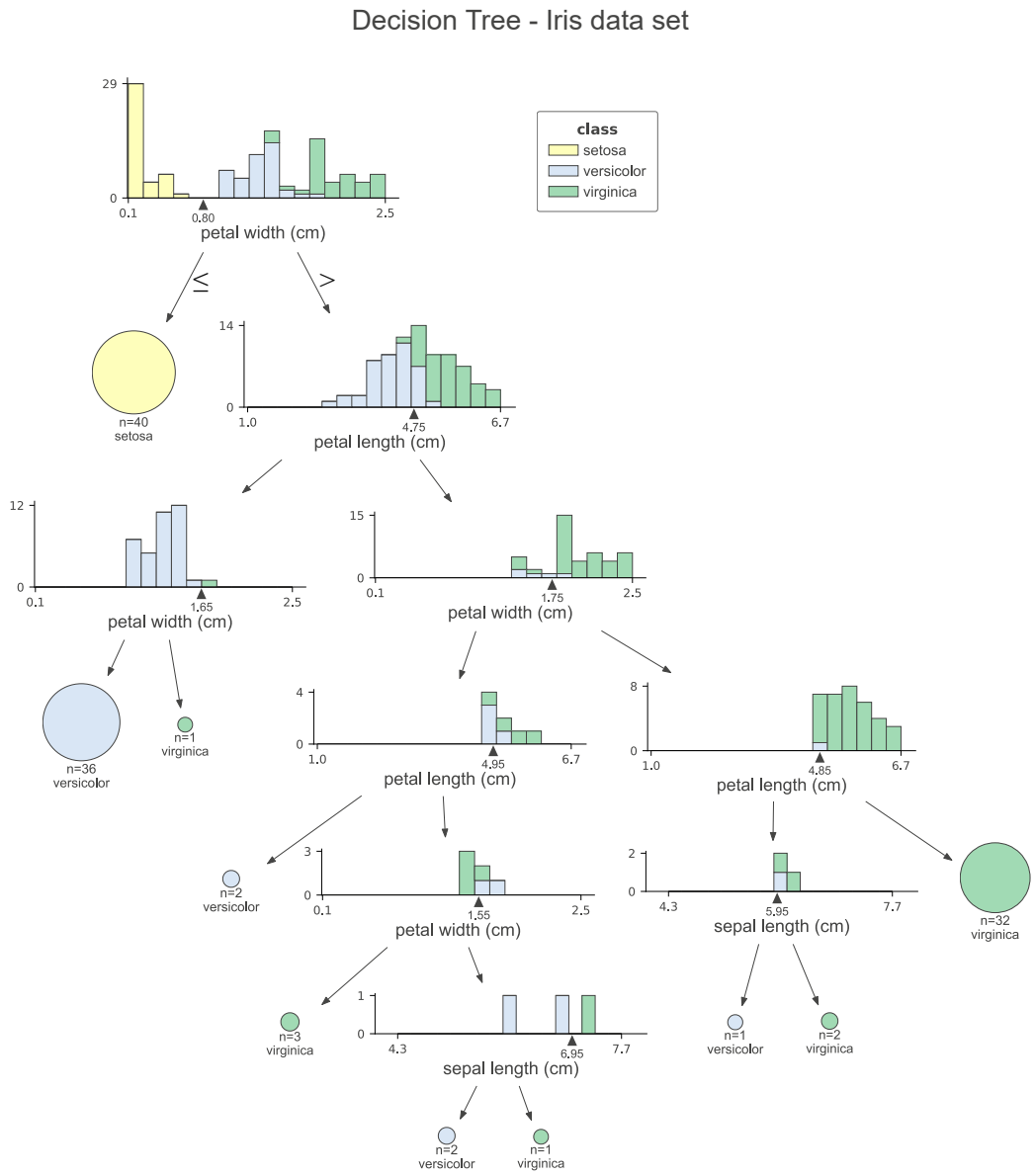
```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: dtreeviz in c:\users\user18\appdata\roaming\pyth
on\python39\site-packages (1.3.7)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packa
ges (from dtreeviz) (1.21.5)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-
packages (from dtreeviz) (3.5.1)
Requirement already satisfied: pytest in c:\programdata\anaconda3\lib\site-pack
ages (from dtreeviz) (7.1.1)
Requirement already satisfied: graphviz>=0.9 in c:\users\user18\appdata\roaming
\python\python39\site-packages (from dtreeviz) (0.20.1)
Requirement already satisfied: colour in c:\users\user18\appdata\roaming\python
\python39\site-packages (from dtreeviz) (0.1.5)
Requirement already satisfied: scikit-learn in c:\programdata\anaconda3\lib\sit
e-packages (from dtreeviz) (1.0.2)
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-pack
ages (from dtreeviz) (1.4.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\programdata\anaconda3\lib
\site-packages (from matplotlib->dtreeviz) (3.0.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\li
b\site-packages (from matplotlib->dtreeviz) (4.25.0)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\sit
e-packages (from matplotlib->dtreeviz) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\programdata\anaconda3\lib\si
te-packages (from matplotlib->dtreeviz) (9.0.1)
Requirement already satisfied: packaging>=20.0 in c:\programdata\anaconda3\lib
\site-packages (from matplotlib->dtreeviz) (21.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anaconda3\li
b\site-packages (from matplotlib->dtreeviz) (1.3.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\programdata\anaconda3
\lib\site-packages (from matplotlib->dtreeviz) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-pa
ckages (from python-dateutil>=2.7->matplotlib->dtreeviz) (1.16.0)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\sit
e-packages (from pandas->dtreeviz) (2021.3)
Requirement already satisfied: attrs>=19.2.0 in c:\programdata\anaconda3\lib\si
te-packages (from pytest->dtreeviz) (21.4.0)
Requirement already satisfied: iniconfig in c:\programdata\anaconda3\lib\site-p
ackages (from pytest->dtreeviz) (1.1.1)
Requirement already satisfied: pluggy<2.0,>=0.12 in c:\programdata\anaconda3\li
b\site-packages (from pytest->dtreeviz) (1.0.0)
Requirement already satisfied: py>=1.8.2 in c:\programdata\anaconda3\lib\site-p
ackages (from pytest->dtreeviz) (1.11.0)
Requirement already satisfied: tomli>=1.0.0 in c:\programdata\anaconda3\lib\sit
e-packages (from pytest->dtreeviz) (1.2.2)
Requirement already satisfied: atomicwrites>=1.0 in c:\programdata\anaconda3\li
b\site-packages (from pytest->dtreeviz) (1.4.0)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-pa
ckages (from pytest->dtreeviz) (0.4.4)
Requirement already satisfied: joblib>=0.11 in c:\programdata\anaconda3\lib\sit
e-packages (from scikit-learn->dtreeviz) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3
\lib\site-packages (from scikit-learn->dtreeviz) (2.2.0)
Requirement already satisfied: scipy>=1.1.0 in c:\programdata\anaconda3\lib\sit
e-packages (from scikit-learn->dtreeviz) (1.7.3)
```

In [14]:
```
from dtreeviz.trees import *
```

In [15]:
```
viz = dtreeviz(dt,
               x_data=X_train,
               y_data=y_train,
               target_name='class',
               feature_names=df.feature_names,
               class_names=list(df.target_names),
               title="Decision Tree - Iris data set")
```

In [16]:
```
viz
```

Out[16]:



Decession tree Regressor

In [18]:
```
from sklearn.datasets import load_boston
```

In [21]:
```
df = load_boston()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: Fut
ureWarning: Function load_boston is deprecated; `load_boston` is deprecated in
1.0 and will be removed in 1.2.

    The Boston housing prices dataset has an ethical problem. You can refer to
    the documentation of this function for further details.

    The scikit-learn maintainers therefore strongly discourage the use of this
    dataset unless the purpose of the code is to study and educate about
    ethical issues in data science and machine learning.

    In this special case, you can fetch the dataset from the original
    source::

        import pandas as pd
        import numpy as np


        data_url = "http://lib.stat.cmu.edu/datasets/boston"
        raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
        data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
        target = raw_df.values[1::2, 2]

    Alternative datasets include the California housing dataset (i.e.
    :func:`~sklearn.datasets.fetch_california_housing`) and the Ames housing
    dataset. You can load the datasets as follows::

        from sklearn.datasets import fetch_california_housing
        housing = fetch_california_housing()

    for the California housing dataset and::

        from sklearn.datasets import fetch_openml
        housing = fetch_openml(name="house_prices", as_frame=True)

    for the Ames housing dataset.

  warnings.warn(msg, category=FutureWarning)
```

In [22]:
```
x = df.data
y= df.target
```

In [23]:
```
x,y
```

Out[23]:
```
(array([[6.3200e-03, 1.8000e+01, 2.3100e+00, ..., 1.5300e+01, 3.9690e+02,
         4.9800e+00],
        [2.7310e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9690e+02,
         9.1400e+00],
        [2.7290e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9283e+02,
         4.0300e+00],
        ...,
        [6.0760e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
         5.6400e+00],
        [1.0959e-01, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9345e+02,
         6.4800e+00],
        [4.7410e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
         7.8800e+00]]),
 array([24. , 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15. ,
        18.9, 21.7, 20.4, 18.2, 19.9, 23.1, 17.5, 20.2, 18.2, 13.6, 19.6,
        15.2, 14.5, 15.6, 13.9, 16.6, 14.8, 18.4, 21. , 12.7, 14.5, 13.2,
        13.1, 13.5, 18.9, 20. , 21. , 24.7, 30.8, 34.9, 26.6, 25.3, 24.7,
        21.2, 19.3, 20. , 16.6, 14.4, 19.4, 19.7, 20.5, 25. , 23.4, 18.9,
        35.4, 24.7, 31.6, 23.3, 19.6, 18.7, 16. , 22.2, 25. , 33. , 23.5,
        19.4, 22. , 17.4, 20.9, 24.2, 21.7, 22.8, 23.4, 24.1, 21.4, 20. ,
        20.8, 21.2, 20.3, 28. , 23.9, 24.8, 22.9, 23.9, 26.6, 22.5, 22.2,
        23.6, 28.7, 22.6, 22. , 22.9, 25. , 20.6, 28.4, 21.4, 38.7, 43.8,
        33.2, 27.5, 26.5, 18.6, 19.3, 20.1, 19.5, 19.5, 20.4, 19.8, 19.4,
        21.7, 22.8, 18.8, 18.7, 18.5, 18.3, 21.2, 19.2, 20.4, 19.3, 22. ,
        20.3, 20.5, 17.3, 18.8, 21.4, 15.7, 16.2, 18. , 14.3, 19.2, 19.6,
        23. , 18.4, 15.6, 18.1, 17.4, 17.1, 13.3, 17.8, 14. , 14.4, 13.4,
        15.6, 11.8, 13.8, 15.6, 14.6, 17.8, 15.4, 21.5, 19.6, 15.3, 19.4,
        17. , 15.6, 13.1, 41.3, 24.3, 23.3, 27. , 50. , 50. , 50. , 22.7,
        25. , 50. , 23.8, 23.8, 22.3, 17.4, 19.1, 23.1, 23.6, 22.6, 29.4,
        23.2, 24.6, 29.9, 37.2, 39.8, 36.2, 37.9, 32.5, 26.4, 29.6, 50. ,
        32. , 29.8, 34.9, 37. , 30.5, 36.4, 31.1, 29.1, 50. , 33.3, 30.3,
        34.6, 34.9, 32.9, 24.1, 42.3, 48.5, 50. , 22.6, 24.4, 22.5, 24.4,
        20. , 21.7, 19.3, 22.4, 28.1, 23.7, 25. , 23.3, 28.7, 21.5, 23. ,
        26.7, 21.7, 27.5, 30.1, 44.8, 50. , 37.6, 31.6, 46.7, 31.5, 24.3,
        31.7, 41.7, 48.3, 29. , 24. , 25.1, 31.5, 23.7, 23.3, 22. , 20.1,
        22.2, 23.7, 17.6, 18.5, 24.3, 20.5, 24.5, 26.2, 24.4, 24.8, 29.6,
        42.8, 21.9, 20.9, 44. , 50. , 36. , 30.1, 33.8, 43.1, 48.8, 31. ,
        36.5, 22.8, 30.7, 50. , 43.5, 20.7, 21.1, 25.2, 24.4, 35.2, 32.4,
        32. , 33.2, 33.1, 29.1, 35.1, 45.4, 35.4, 46. , 50. , 32.2, 22. ,
        20.1, 23.2, 22.3, 24.8, 28.5, 37.3, 27.9, 23.9, 21.7, 28.6, 27.1,
        20.3, 22.5, 29. , 24.8, 22. , 26.4, 33.1, 36.1, 28.4, 33.4, 28.2,
        22.8, 20.3, 16.1, 22.1, 19.4, 21.6, 23.8, 16.2, 17.8, 19.8, 23.1,
        21. , 23.8, 23.1, 20.4, 18.5, 25. , 24.6, 23. , 22.2, 19.3, 22.6,
        19.8, 17.1, 19.4, 22.2, 20.7, 21.1, 19.5, 18.5, 20.6, 19. , 18.7,
        32.7, 16.5, 23.9, 31.2, 17.5, 17.2, 23.1, 24.5, 26.6, 22.9, 24.1,
        18.6, 30.1, 18.2, 20.6, 17.8, 21.7, 22.7, 22.6, 25. , 19.9, 20.8,
        16.8, 21.9, 27.5, 21.9, 23.1, 50. , 50. , 50. , 50. , 50. , 13.8,
        13.8, 15. , 13.9, 13.3, 13.1, 10.2, 10.4, 10.9, 11.3, 12.3,  8.8,
         7.2, 10.5,  7.4, 10.2, 11.5, 15.1, 23.2,  9.7, 13.8, 12.7, 13.1,
        12.5,  8.5,  5. ,  6.3,  5.6,  7.2, 12.1,  8.3,  8.5,  5. , 11.9,
        27.9, 17.2, 27.5, 15. , 17.2, 17.9, 16.3,  7. ,  7.2,  7.5, 10.4,
         8.8,  8.4, 16.7, 14.2, 20.8, 13.4, 11.7,  8.3, 10.2, 10.9, 11. ,
         9.5, 14.5, 14.1, 16.1, 14.3, 11.7, 13.4,  9.6,  8.7,  8.4, 12.8,
        10.5, 17.1, 18.4, 15.4, 10.8, 11.8, 14.9, 12.6, 14.1, 13. , 13.4,
        15.2, 16.1, 17.8, 14.9, 14.1, 12.7, 13.5, 14.9, 20. , 16.4, 17.7,
```

```
       19.5, 20.2, 21.4, 19.9, 19. , 19.1, 19.1, 20.1, 19.9, 19.6, 23.2,
       29.8, 13.8, 13.3, 16.7, 12. , 14.6, 21.4, 23. , 23.7, 25. , 21.8,
       20.6, 21.2, 19.1, 20.6, 15.2,  7. ,  8.1, 13.6, 20.1, 21.8, 24.5,
       23.1, 19.7, 18.3, 21.2, 17.5, 16.8, 22.4, 20.6, 23.9, 22. , 11.9]))
```

In [31]:
```
from sklearn.tree import DecisionTreeRegressor
```

In [35]:
```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_s
```

In [43]:
```
dtr = DecisionTreeRegressor(max_depth=2)
```

In [44]:
```
dtr.fit(X_train,y_train)
```

Out[44]:
```
DecisionTreeRegressor(max_depth=2)
```

In [45]:
```
# plot the tree
plt.figure(figsize=(25,20))
viz = dtreeviz(dtr,
               x_data=X_train,
               y_data=y_train,
               target_name='price',
               feature_names=df.feature_names,
               title="Decision Tree - Boston housing")
               #show_just_path=True)
```

```
<Figure size 1800x1440 with 0 Axes>
```

In [46]:
```
viz
```

Out[46]: