Install causalnex, pydotplus and required libraries

```
!pip install causalnex
```

```
Collecting causalnex
  Downloading https://files.pythonhosted.org/packages/ff/c2/20e24f332cb42b33030fef87
    |████████████████████████████████| 133kB 11.3MB/s
Requirement already satisfied: scipy<1.6,>=1.2.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: pandas<2.0,>=1.0 in /usr/local/lib/python3.7/dist-pac
Collecting prettytable<0.8,>=0.7.2
  Downloading https://files.pythonhosted.org/packages/ef/30/4b0746848746ed5941f05247
Requirement already satisfied: numpy<2.0,>=1.14.2 in /usr/local/lib/python3.7/dist-p
Collecting scikit-learn!=0.22.2.post1,<0.23.0,>=0.20.2
  Downloading https://files.pythonhosted.org/packages/71/b0/471bfdb7741523dfbddd038c
    |████████████████████████████████| 7.1MB 16.5MB/s
Collecting wrapt<1.12,>=1.11.0
  Downloading https://files.pythonhosted.org/packages/23/84/323c2415280bc4fc880ac505
Requirement already satisfied: networkx~=2.5 in /usr/local/lib/python3.7/dist-packag
Collecting pgmpy<0.2.0,>=0.1.12
  Downloading https://files.pythonhosted.org/packages/a3/0e/d9fadbfaa35e010c04d43acd
    |████████████████████████████████| 337kB 35.2MB/s
Requirement already satisfied: torch~=1.7 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/di
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: statsmodels in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pyparsing in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: patsy>=0.4.0 in /usr/local/lib/python3.7/dist-package
Building wheels for collected packages: prettytable, wrapt
  Building wheel for prettytable (setup.py) ... done
  Created wheel for prettytable: filename=prettytable-0.7.2-cp37-none-any.whl size=1
  Stored in directory: /root/.cache/pip/wheels/80/34/1c/3967380d9676d162cb59513bd9dc
  Building wheel for wrapt (setup.py) ... done
  Created wheel for wrapt: filename=wrapt-1.11.2-cp37-cp37m-linux_x86_64.whl size=68
  Stored in directory: /root/.cache/pip/wheels/d7/de/2e/efa132238792efb6459a96e85916
Successfully built prettytable wrapt
ERROR: tensorflow 2.4.1 has requirement wrapt~=1.12.1, but you'll have wrapt 1.11.2
Installing collected packages: prettytable, scikit-learn, wrapt, pgmpy, causalnex
  Found existing installation: prettytable 2.1.0
    Uninstalling prettytable-2.1.0:
      Successfully uninstalled prettytable-2.1.0
  Found existing installation: scikit-learn 0.22.2.post1
    Uninstalling scikit-learn-0.22.2.post1:
      Successfully uninstalled scikit-learn-0.22.2.post1
  Found existing installation: wrapt 1.12.1
    Uninstalling wrapt-1.12.1:
      Successfully uninstalled wrapt-1.12.1
Successfully installed causalnex-0.9.2 pgmpy-0.1.14 prettytable-0.7.2 scikit-learn-0
```

```
pip install "causalnex[all]"
```

```
Requirement already satisfied: causalnex[all] in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: scipy<1.6,>=1.2.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: pgmpy<0.2.0,>=0.1.12 in /usr/local/lib/python3.7/dist
Requirement already satisfied: networkx~=2.5 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: scikit-learn!=0.22.2.post1,<0.23.0,>=0.20.2 in /usr/l
Requirement already satisfied: pandas<2.0,>=1.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: torch~=1.7 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: wrapt<1.12,>=1.11.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: prettytable<0.8,>=0.7.2 in /usr/local/lib/python3.7/d
Requirement already satisfied: numpy<2.0,>=1.14.2 in /usr/local/lib/python3.7/dist-p
Collecting pygraphviz<2.0,>=1.5; extra == "all"
  Downloading https://files.pythonhosted.org/packages/3a/d6/2c56f09ee83dbebb62c40487
    |████████████████████████████████| 122kB 10.9MB/s
Requirement already satisfied: pyparsing in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: statsmodels in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/di
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: patsy>=0.4.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (f
Building wheels for collected packages: pygraphviz
  Building wheel for pygraphviz (setup.py) ... error
  ERROR: Failed building wheel for pygraphviz
  Running setup.py clean for pygraphviz
Failed to build pygraphviz
Installing collected packages: pygraphviz
    Running setup.py install for pygraphviz ... error
ERROR: Command errored out with exit status 1: /usr/bin/python3 -u -c 'import sys, s
```

```
!pip install pydotplus
```

```
Requirement already satisfied: pydotplus in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pyparsing>=2.0.1 in /usr/local/lib/python3.7/dist-pac
```

```
!pip install -q pydot
```

```
!apt install libgraphviz-dev
!pip install pygraphviz
```

```
0 upgraded, 8 newly installed, 0 to remove and 31 not upgraded.
Need to get 2,120 kB of archives.
After this operation, 7,128 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtk2.0-common all 2.24
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtk2.0-0 amd64 2.24.32
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgail18 amd64 2.24.32-1
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgail-common amd64 2.24
Get:5 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libxdot4 amd64 2.40.1
Get:6 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libgvc6-plugins-gtk a
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libgraphviz-dev amd64
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtk2.0-bin amd64 2.24.
Fetched 2,120 kB in 0s (17.2 MB/s)
Selecting previously unselected package libgtk2.0-common.
```

```
(Reading database ... 160983 files and directories currently installed.)
Preparing to unpack .../0-libgtk2.0-common_2.24.32-1ubuntu1_all.deb ...
Unpacking libgtk2.0-common (2.24.32-1ubuntu1) ...
Selecting previously unselected package libgtk2.0-0:amd64.
Preparing to unpack .../1-libgtk2.0-0_2.24.32-1ubuntu1_amd64.deb ...
Unpacking libgtk2.0-0:amd64 (2.24.32-1ubuntu1) ...
Selecting previously unselected package libgail18:amd64.
Preparing to unpack .../2-libgail18_2.24.32-1ubuntu1_amd64.deb ...
Unpacking libgail18:amd64 (2.24.32-1ubuntu1) ...
Selecting previously unselected package libgail-common:amd64.
Preparing to unpack .../3-libgail-common_2.24.32-1ubuntu1_amd64.deb ...
Unpacking libgail-common:amd64 (2.24.32-1ubuntu1) ...
Selecting previously unselected package libxdot4.
Preparing to unpack .../4-libxdot4_2.40.1-2_amd64.deb ...
Unpacking libxdot4 (2.40.1-2) ...
Selecting previously unselected package libgvc6-plugins-gtk.
Preparing to unpack .../5-libgvc6-plugins-gtk_2.40.1-2_amd64.deb ...
Unpacking libgvc6-plugins-gtk (2.40.1-2) ...
Selecting previously unselected package libgraphviz-dev.
Preparing to unpack .../6-libgraphviz-dev_2.40.1-2_amd64.deb ...
Unpacking libgraphviz-dev (2.40.1-2) ...
Selecting previously unselected package libgtk2.0-bin.
Preparing to unpack .../7-libgtk2.0-bin_2.24.32-1ubuntu1_amd64.deb ...
Unpacking libgtk2.0-bin (2.24.32-1ubuntu1) ...
Setting up libgtk2.0-common (2.24.32-1ubuntu1) ...
Setting up libxdot4 (2.40.1-2) ...
Setting up libgtk2.0-0:amd64 (2.24.32-1ubuntu1) ...
Setting up libgail18:amd64 (2.24.32-1ubuntu1) ...
Setting up libgail-common:amd64 (2.24.32-1ubuntu1) ...
Setting up libgvc6-plugins-gtk (2.40.1-2) ...
Setting up libgraphviz-dev (2.40.1-2) ...
Setting up libgtk2.0-bin (2.24.32-1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-packages/ideep4py/lib/libmkldnn

Collecting pygraphviz
  Using cached https://files.pythonhosted.org/packages/3a/d6/2c56f09ee83dbebb62c40
Building wheels for collected packages: pygraphviz
  Building wheel for pygraphviz (setup.py) ... done
  Created wheel for pygraphviz: filename=pygraphviz-1.7-cp37-cp37m-linux_x86_64.wh
  Stored in directory: /root/.cache/pip/wheels/32/59/00/14934a4292c4359eeabcdbf90f
Successfully built pygraphviz
Installing collected packages: pygraphviz
Successfully installed pygraphviz-1.7
```

```
!pip install pygraphviz
```

```
Collecting pygraphviz
  Using cached https://files.pythonhosted.org/packages/3a/d6/2c56f09ee83dbebb62c4048
Building wheels for collected packages: pygraphviz
  Building wheel for pygraphviz (setup.py) ... error
  ERROR: Failed building wheel for pygraphviz
  Running setup.py clean for pygraphviz
Failed to build pygraphviz
Installing collected packages: pygraphviz
    Running setup.py install for pygraphviz ... error
ERROR: Command errored out with exit status 1: /usr/bin/python3 -u -c 'import sys, s
```

```
sudo apt-get install python-pip python-virtualenv
```

```
  File "<ipython-input-18-7d5889a17d3c>", line 1
    sudo apt-get install python-pip python-virtualenv
           ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
!pip install graphviz !apt-get install graphviz
```

```
ERROR: Invalid requirement: '!apt-get'
```

Import data

```
import pandas as pd
```

```
data=pd.read_csv('breast-cancer_csv.csv')
data.head()
```

	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
0	40-49	premeno	15-19	0-2	yes	3	right	left_up	no	recurrence-events
1	50-59	ge40	15-19	0-2	no	1	right	central	no	no-recurrence-events
2	50-59	ge40	35-39	0-2	no	2	left	left_low	no	recurrence-events

```
data.dropna(inplace=True)
```

```
data.isna().sum()
```

```
age            0
menopause      0
tumor-size     0
inv-nodes      0
node-caps      0
deg-malig      0
breast         0
breast-quad    0
```

```
irradiat       0
Class          0
dtype: int64
```

```
data.Class.replace({'recurrence-events':1,'no-recurrence-events':0},inplace=True)
```

```
data.irradiat.replace({'no':0,'yes':1},inplace=True)
```

```
data['node-caps'].replace({'no':0,'yes':1},inplace=True)
```

```
data['breast'].replace({'right':0,'left':1},inplace=True)
```

```
data['menopause'].replace({'premeno':0,'ge40':1,'lt40':2},inplace=True)
```

```
data['breast-quad'].replace({'left_low':0,'left_up':1,'central':2,'right_low':3,'right_up'
```

```
data['menopause'].unique()
```

```
array([0, 1, 2])
```

```
data.head()
```

	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
0	40-49	0	15-19	0-2	1	3	0	1	0	1
1	50-59	1	15-19	0-2	0	1	0	2	0	0
2	50-59	1	35-39	0-2	0	2	1	0	0	1

```
data['age'].unique()
```

```
array(['40-49', '50-59', '60-69', '30-39', '70-79', '20-29'], dtype=object)
```

```
data['age'].replace({'20-29':0,'30-39':1,'40-49':1,'50-59':2,'60-69':2,'70-79':3},inplace=
```

```
data['inv-nodes'].replace({'0-2':0,'3-5':1,'6-8':2,'9-11':3,'12-14':4,'15-17':5,'24-26':6}
```

```
data['tumor-size'].replace({'0-4':0,'5-9':0,'10-14':1,'15-19':1,'20-24':2,'25-29':2,'30-34
```

```
data['breast-quad'].unique()
```

```
array([1, 2, 0, 4, 3])
```

```
data.menopause.unique()
```

```
     array([0, 1, 2])
```

```
data.head()
```

	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
0	1	0	1	0	1	3	0	1	0	1
1	2	1	1	0	0	1	0	2	0	0
2	2	1	3	0	0	2	1	0	0	1
3	1	0	3	0	1	3	0	0	1	0
4	1	0	3	1	1	2	1	4	0	1

Applying the NOTEARS algorithm to learn the structure.

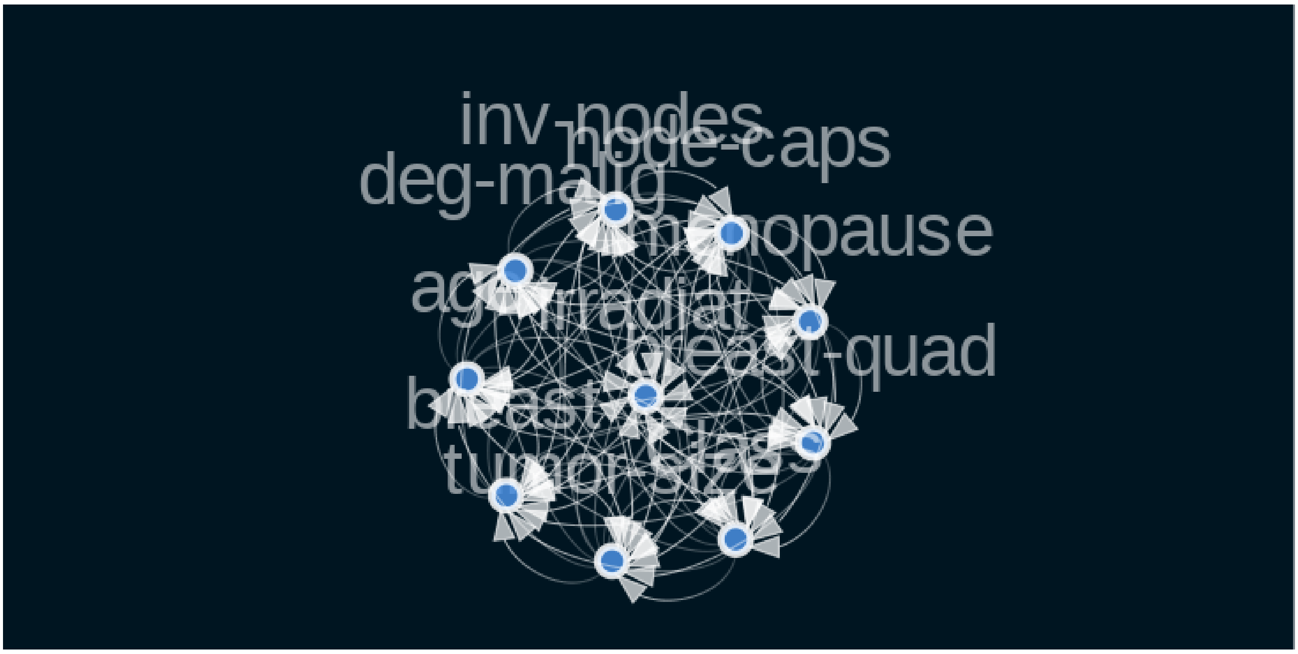
```
# silence warnings
import warnings
warnings.filterwarnings("ignore")
```

```
from causalnex.structure.notears import from_pandas
sm = from_pandas(data)
```

visualise the learned StructureModel using the plot function.

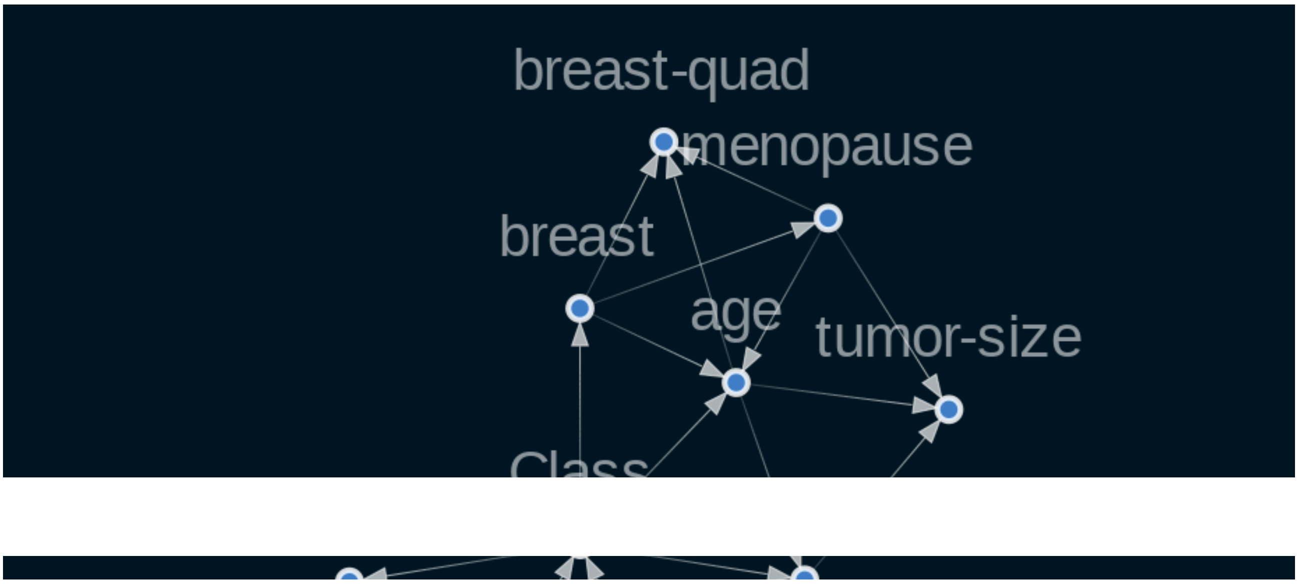
```
from IPython.display import Image
from causalnex.plots import plot_structure, NODE_STYLE, EDGE_STYLE

viz = plot_structure(
    sm,
    graph_attributes={"scale": "0.5"},
    all_node_attributes=NODE_STYLE.WEAK,
    all_edge_attributes=EDGE_STYLE.WEAK)
Image(viz.draw(format='png'))
```



The reason why we have a fully connected graph here is we haven’t applied thresholding to the weaker edges. Thresholding can be applied either by specifying the value for the parameter w_threshold in from_pandas, or we can remove the edges by calling the structure model function, remove_edges_below_threshold.

```
sm.remove_edges_below_threshold(0.3)
viz = plot_structure(
    sm,
    graph_attributes={"scale": "0.5"},
    all_node_attributes=NODE_STYLE.WEAK,
    all_edge_attributes=EDGE_STYLE.WEAK)
Image(viz.draw(format='png'))
```

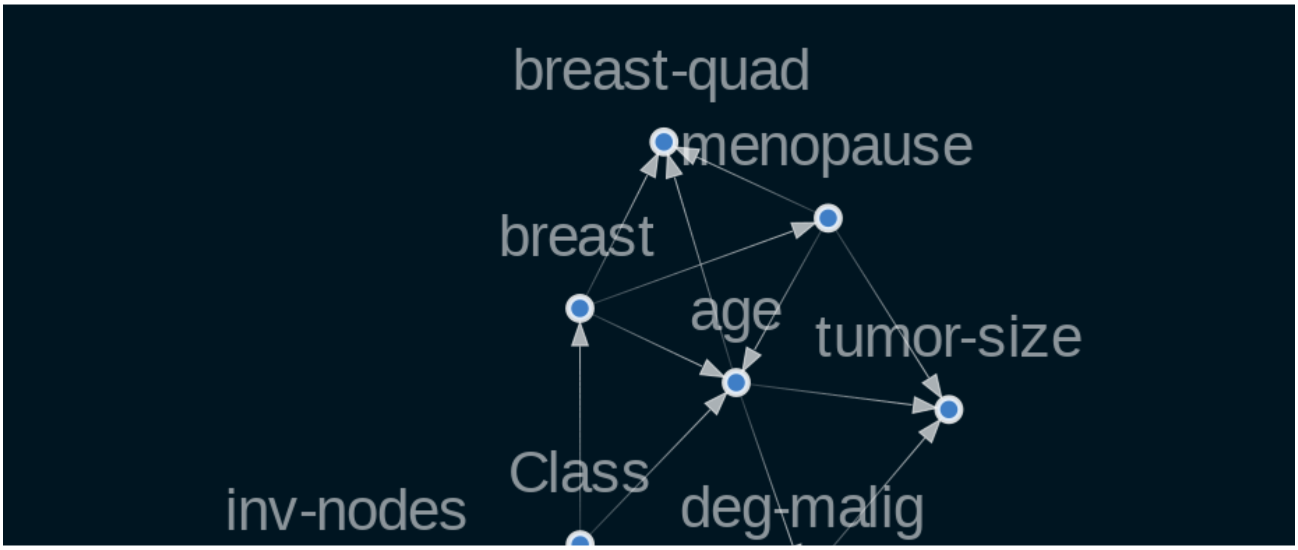


Modifying the Structure

To correct erroneous relationships, we can incorporate domain knowledge into the model after structure learning. We can modify the structure model through adding and deleting the edges. For example, we can add and remove edges as:

```
# sm.add_edge("breast-quad", "Class")
# sm.add_edge('menopause','Class')


viz = plot_structure(
    sm,
    graph_attributes={"scale": "0.5"},
    all_node_attributes=NODE_STYLE.WEAK,
    all_edge_attributes=EDGE_STYLE.WEAK)
Image(viz.draw(format='png'))
```



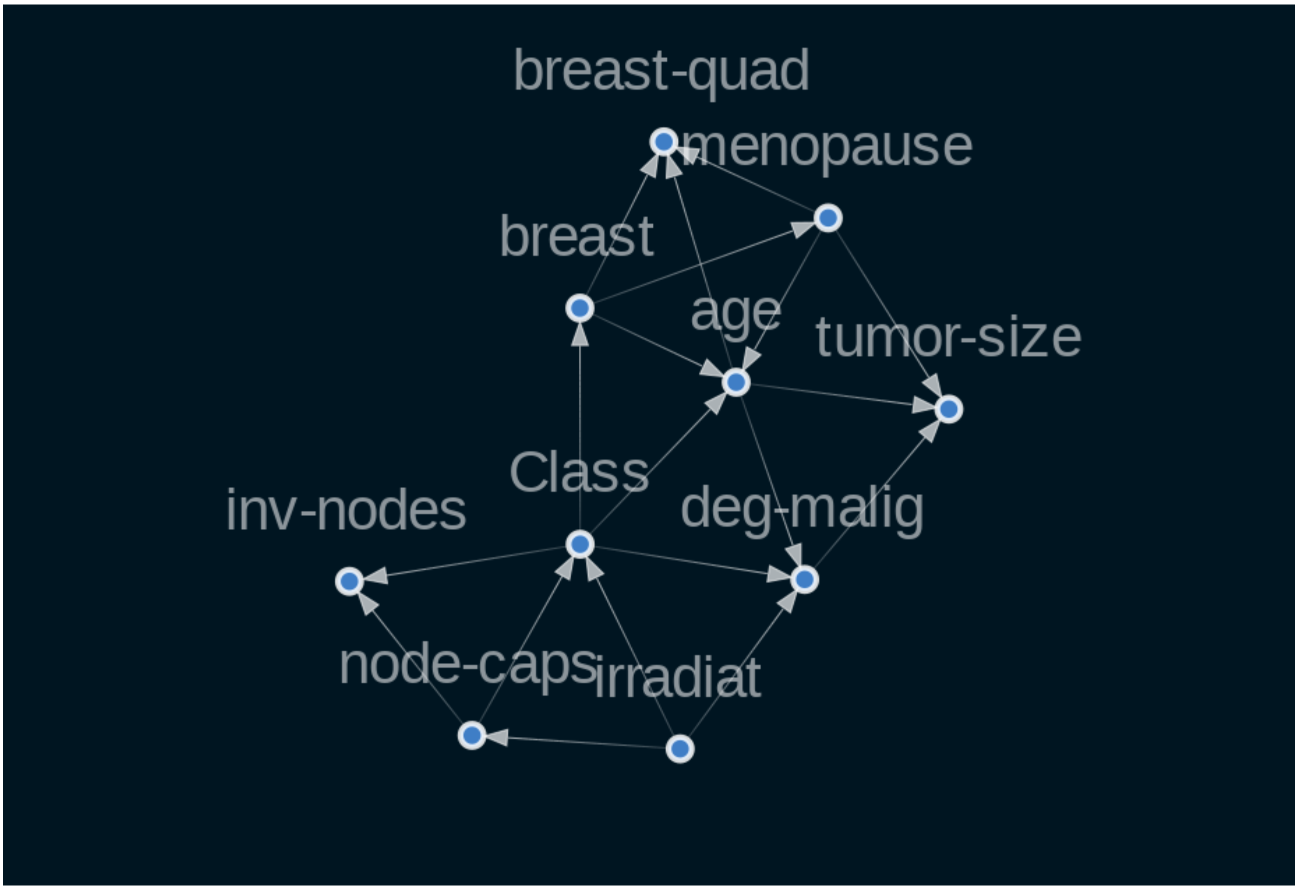
We can now visualise our updated structure to confirm it looks reasonable.

```
viz = plot_structure(
    sm,
    graph_attributes={"scale": "0.5"},
    all_node_attributes=NODE_STYLE.WEAK,
    all_edge_attributes=EDGE_STYLE.WEAK)
Image(viz.draw(format='png'))
```

We can see there are two separate subgraphs here in the visualisation plot: Dalc->Walc and the other big subgraph. We can retrieve the largest subgraph easily by calling the StructureModel function get_largest_subgraph().

```
sm = sm.get_largest_subgraph()

viz = plot_structure(
    sm,
    graph_attributes={"scale": "0.5"},
    all_node_attributes=NODE_STYLE.WEAK,
    all_edge_attributes=EDGE_STYLE.WEAK)
Image(viz.draw(format='png'))
```



After deciding on how the final structure model should look, we can instantiate a BayesianNetwork.

```
from causalnex.network import BayesianNetwork

bn = BayesianNetwork(sm)
```

We are now ready to move on to learning the conditional probability distribution of different features in the BayesianNetwork.

Fitting the Conditional Distribution of the Bayesian Network

Preparing the Discretised Data

Bayesian Networks in CausalNex support only discrete distributions. Any continuous features, or features with a large number of categories, should be discretised prior to fitting the Bayesian Network. Models containing variables with many possible values will typically be badly fit, and exhibit poor performance.

For example, consider P(G2 | G1), where G1 and G2 have possible values 0 to 20. The discrete conditional probability distribution is therefore specified using 21x21 (441) possible combinations - most of which we will be unlikely to observe.

CausalNex provides a few helper methods to make discretisation easier. Let’s start by reducing the number of categories in some of the categorical features by combining similar values. We will make numeric features categorical by discretisation, and then give the buckets meaningful labels.

Cardinality of Categorical Features

```
data.head()
```

	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
0	1	0	1	0	1	3	0	1	0	1
1	2	1	1	0	0	1	0	2	0	0
2	2	1	3	0	0	2	1	0	0	1
3	1	0	3	0	1	3	0	0	1	0
4	1	0	3	1	1	2	1	4	0	1

```
data.irradiat.replace({0:'no',1:'yes'},inplace=True)
```

```
data['node-caps'].replace({0:'no',1:'yes'},inplace=True)
```

```
data.head()
```

	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
0	1	0	1	0	yes	3	0	1	no	1
1	2	1	1	0	no	1	0	2	no	0
2	2	1	3	0	no	2	1	0	no	1
3	1	0	3	0	yes	3	0	0	yes	0
4	1	0	3	1	yes	2	1	4	no	1

```
Class_map = {0: "healthy", 1: "breast cancer"}

data["Class"] = data["Class"].map(Class_map)
data.head()
```

	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
0	1	0	1	0	yes	3	0	1	no	breast cancer
1	2	1	1	0	no	1	0	2	no	healthy
2	2	1	3	0	no	2	1	0	no	breast cancer
3	1	0	3	0	yes	3	0	0	yes	healthy

Train-Test split

```
from sklearn.model_selection import train_test_split

train, test = train_test_split(data, train_size=0.9, test_size=0.1, random_state=7)
```

Model Probability

With the learnt structure model from earlier and the discretised data, we can now fit the probability distrbution of the Bayesian Network. The first step in this is specifying all of the states that each node can take. This can be done either from data, or providing a dictionary of node values. We use the full dataset here to avoid cases where states in our test set do not exist in the training set. For real-world applications, these states may need to be provided using the dictionary method.

```
bn = bn.fit_node_states(data)
```

Fit Conditional Probability Distributions

The fit_cpds method of BayesianNetwork accepts a dataset to learn the conditional probablilty distributions (CPDs) of each node, along with a method of how to do this fit.

```
bn = bn.fit_cpds(train, method="BayesianEstimator", bayes_prior="K2")
```

once we have the CPDs, we can inspect them through the cpds property, which is a dictionary of node->cpd.

```
bn.cpds["Class"]
```

```
    irradiat      no                yes


                Class
```

The CPD dictionaries are multi-indexed, and so the loc function can be a useful way to interact with them:

Predict the State given the Input Data

The predict method of BayesianNetwork allows us to make predictions based on the data using the learnt Bayesian Network.

```
data.loc[16, data.columns != 'Class']
```

```
age              2
menopause        0
tumor-size       1
inv-nodes        1
node-caps       no
deg-malig        1
breast           0
breast-quad      1
irradiat        no
Name: 16, dtype: object
```

```
predictions = bn.predict(data, "Class")
```

```
print('The prediction is \'{prediction}\''.format(prediction=predictions.loc[18, 'Class_pr
```

```
The prediction is 'healthy'
```

The prediction by the Bayesian Network turns out to be a Fail. Let’s compare this to the ground truth:

```
print('The ground truth is \'{truth}\''.format(truth=data.loc[18, 'Class']))
```

```
The ground truth is 'healthy'
```

which turns out to be the same.

Model Quality

To evaluate the quality of the model that has been learned, CausalNex supports two main approaches: Classification Report and Reciever Operating Characteristics (ROC) / Area Under the ROC Curve (AUC). In this section each will be discussed.

Classification Report

To obtain a classification report using a BN, we need to provide a test set, and the node we are trying to classify. The report will predict the target node for all rows in the test set, and evaluate how well those predictions are made.

```
from causalnex.evaluation import classification_report
classification_report(bn, test, "Class")
```

```
{'Class_breast cancer': {'f1-score': 0.5,
  'precision': 0.75,
  'recall': 0.375,
  'support': 8},
 'Class_healthy': {'f1-score': 0.8636363636363635,
  'precision': 0.7916666666666666,
  'recall': 0.95,
  'support': 20},
 'accuracy': 0.7857142857142857,
 'macro avg': {'f1-score': 0.6818181818181818,
  'precision': 0.7708333333333333,
  'recall': 0.6625,
  'support': 28},
 'weighted avg': {'f1-score': 0.7597402597402596,
  'precision': 0.7797619047619048,
  'recall': 0.7857142857142857,
  'support': 28}}
```

This report shows that the model we have defined is able to classify whether a person having breast cancer or not.

ROC / AUC

Reciever Operating Characteristics (ROC), and the Area Under the ROC Curve (AUC) can be obtained using the roc_auc method within the CausalNex metrics module. Again, a test set and target node must be provided. The ROC curve is computed by micro-averaging predictions made across all states (classes) of the target node.

```
from causalnex.evaluation import roc_auc
roc, auc = roc_auc(bn, test, "Class")
print(auc)

     0.7691326530612245
```

The AUC value for our model is good enough, giving us confidence in the performance.

Querying Marginals

After iterating over our model structure, CPDs, and validating our model quality, we can query our model under defferent observation to gain insights.

Baseline Marginals

To query the model for baseline marginals that reflect the population as a whole, a query method can be used. First let’s update our model using the complete dataset, since the one we currently have was only built from training data.

```
bn = bn.fit_cpds(data, method="BayesianEstimator", bayes_prior="K2")

     WARNING:root:Replacing existing CPD for age
     WARNING:root:Replacing existing CPD for tumor-size
     WARNING:root:Replacing existing CPD for deg-malig
     WARNING:root:Replacing existing CPD for breast-quad
     WARNING:root:Replacing existing CPD for menopause
     WARNING:root:Replacing existing CPD for node-caps
     WARNING:root:Replacing existing CPD for inv-nodes
     WARNING:root:Replacing existing CPD for Class
     WARNING:root:Replacing existing CPD for breast
     WARNING:root:Replacing existing CPD for irradiat
```

0s completed at 12:27 PM