

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import bnlearn as bn

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

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

```
In [4]: train = pd.read_csv('train.csv').set_index('PassengerId')
test = pd.read_csv('test.csv').set_index('PassengerId')
```

```
In [5]: train.head(3)
```

```
Out[5]:
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
PassengerId											
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

```
In [6]: train.describe()
```

```
Out[6]:
```

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
In [7]: drop_list = ['Name', 'Age', 'Cabin', 'Ticket', 'Fare']
train = train.drop(columns=drop_list)
test = test.drop(columns=drop_list)
```

```
In [8]: train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 891 entries, 1 to 891
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Survived    891 non-null    int64
1   Pclass      891 non-null    int64
2   Sex         891 non-null    object
3   SibSp       891 non-null    int64
4   Parch       891 non-null    int64
5   Embarked    889 non-null    object
dtypes: int64(4), object(2)
memory usage: 48.7+ KB
```

In [9]: test.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 418 entries, 892 to 1309
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0    Pclass      418 non-null    int64
1    Sex         418 non-null    object
2    SibSp       418 non-null    int64
3    Parch      418 non-null    int64
4    Embarked    418 non-null    object
dtypes: int64(3), object(2)
memory usage: 19.6+ KB
```

In [10]: dfhot\_train, dfnum\_train = bn.df2onehot(train)  
dfhot\_test, dfnum\_test = bn.df2onehot(test)

```
[df2onehot] >Auto detecting dtypes.
100%|██████████| 6/6 [00:00<00:00, 173.85it/s]

[df2onehot] >Set dtypes in dataframe..
[df2onehot]: 100%|██████████| 6/6 [00:00<00:00, 345.25it/s]

[df2onehot] >Total onehot features: 18

[df2onehot] >Auto detecting dtypes.
100%|██████████| 5/5 [00:00<00:00, 799.95it/s]

[df2onehot] >Set dtypes in dataframe..
[df2onehot]: 100%|██████████| 5/5 [00:00<00:00, 420.84it/s]

[df2onehot] >Total onehot features: 14
```

In [11]: dfnum\_train

Out[11]:

	Survived	Pclass	Sex	SibSp	Parch	Embarked
0	0	3	1	2	1	3
1	1	1	0	2	1	1
2	1	3	0	1	1	3
3	1	1	0	2	1	3
4	0	3	1	1	1	3
...	...	...	...	...	...	...
886	0	2	1	1	1	3
887	1	1	0	1	1	3
888	0	3	0	2	3	3
889	1	1	1	1	1	1
890	0	3	1	1	1	2

891 rows × 6 columns

In [12]: dfnum\_target = dfnum\_train.pop('Survived')

```
In [13]: Xtrain, Xval, Ztrain, Zval = train_test_split(dfnum_train, dfnum_target, test_size=0.2, random_state=0)
valid = pd.concat([Xval, Zval], axis='columns')
dfnum = pd.concat([Xtrain, Ztrain], axis='columns')
dfnum
```

```
Out[13]:
```

	Pclass	Sex	SibSp	Parch	Embarked	Survived
140	3	0	1	3	1	0
439	2	1	1	1	3	0
817	2	1	2	2	1	0
378	3	1	1	1	1	0
491	3	1	1	1	3	0
...	...	...	...	...	...	...
835	1	0	2	2	1	1
192	3	0	2	1	3	1
629	3	1	1	1	2	0
559	3	0	2	1	3	1
684	2	1	2	2	3	0

712 rows × 6 columns

```
In [14]: def get_acc(model, df, col):
# Get accuracy score by the model for the validation dataset df with target col
pred = bn.predict(model, df, variables=[col])
print(pred)
acc = accuracy_score(df[col], pred[col])
print('Accuracy -', acc)
return acc
```

```
In [15]: %%time
# Structure Learning
DAG = bn.structure_learning.fit(dfnum, methodtype='hc', root_node='Survived', bw_list_method='nodes', verbose=3)

# Plot
G = bn.plot(DAG)

# Parameter Learning
model = bn.parameter_learning.fit(DAG, dfnum, verbose=3);

| SibSp(3) | 0.16223067173637515 | ... | 0.1409043112513144 |
+-----+-----+-----+-----+
| SibSp(4) | 0.15842839036755385 | ... | 0.1409043112513144 |
+-----+-----+-----+-----+
| SibSp(5) | 0.15842839036755385 | ... | 0.15667718191377497 |
+-----+-----+-----+-----+
[bnlearn] >CPD of Embarked:
+-----+-----+
| Embarked(0) | 0.147196 |
+-----+-----+
| Embarked(1) | 0.221379 |
+-----+-----+
| Embarked(2) | 0.183411 |
+-----+-----+
| Embarked(3) | 0.448014 |
+-----+-----+
[bnlearn] >Compute structure scores for model comparison (higher is better).
CPU times: total: 234 ms
Wall time: 6.19 s
```

```
In [16]: # About the model
model
```

```
Out[16]: {'model': <pgmpy.models.BayesianNetwork.BayesianNetwork at 0x253d40b7190>,
'adjmat': target   Pclass   Sex   SibSp   Parch   Embarked   Survived
source
Pclass      False   False   False   False      False      True
Sex          False   False   False   True       False      False
SibSp        False   False   False   False      False      False
Parch        False   False   True    False      False      False
Embarked     True    False   False   False      False      False
Survived     False   True    False   False      False      False,
'config': {'verbose': 3, 'method': 'bayes', 'n_jobs': -1},
'model_edges': [('Pclass', 'Survived'),
('Survived', 'Sex'),
('Sex', 'Parch'),
('Parch', 'SibSp'),
('Embarked', 'Pclass')],
'structure_scores': {'k2': -3174.9527427131247,
'bic': -3209.323591016362,
'bdeu': -3172.76499432183,
'bds': -3186.6279379330285},
'independence_test': None}
```

```
In [17]: # Get score of the model1
acc1 = get_acc(model, valid, 'Survived')
```

```
[bnlearn]> Remaining columns for inference: 5
```

```
100%|██████████| 59/59 [00:00<00:00, 1191.93it/s]
```

	Survived	p
0	0	0.725084
1	0	0.725084
2	0	0.725084
3	1	0.662098
4	0	0.507407
..	...	...
174	0	0.507407
175	0	0.725084
176	1	0.662098
177	0	0.725084
178	0	0.725084

```
[179 rows x 2 columns]
Accuracy - 0.8156424581005587
```