Breast Cancer Data Prediction Using Daimensions

In this notebook, we'll be working with a dataset from the University of California Irvine's Machine Learning Repository. It has nine attribute columns to describe various aspects of cells and one classification column that classifies each cell as benign or malignant cancer. More information about the data can be found at: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic).

We have two goals: one is to build a model predicting whether a cell is benign or malignant on future cell data and the other is to use attribute rank to learn about which attributes of the cell are most important for predicting cancer in cells. Daimensions' attribute rank option is useful for a lot of biomedical data like cancer cells because most of the time we are not only looking to predict which cells are cancerous but also what caused the cancer. Attribute rank helps us learn about this aspect of the data by telling us which attributes most closely correlate with a cell's classification. This greatly contributes to our understanding of the data and helps guide us toward probable cause.

Here is a look at our training data and the attributes we're using. For the target column, 2 is benign and 4 is malignant.

In [1]:

```
! head cancer_train.csv
# For Windows command prompt:
# type cancer_train.csv | more
```

```
Clump_Thickness,Uniformity_of_Cell_Size,Uniformity_of_Cell_Shape,Marginal_Adhesion,Single
_Epithelial_Cell_Size,Bare_Nuclei,Bland_Chromatin,Normal_Nucleoli,Mitoses,Class
5,1,1,1,2,1,3,1,1,2
5,4,4,5,7,10,3,2,1,2
3,1,1,1,2,2,3,1,1,2
6,8,8,1,3,4,3,7,1,2
4,1,1,3,2,1,3,1,1,2
8,10,10,8,7,10,9,7,1,4
1,1,1,1,2,10,3,1,1,2
2,1,2,1,2,1,3,1,1,2
2,1,1,1,2,1,1,1,5,2
```

1. Get Measurements

We always want to measure our data before building our predictor in order to ensure we are building the right model. For more information about how to use Daimensions and why we want to measure our data beforehand, check out the Titanic notebook.

In [2]:

```
! btc -measureonly cancer_train.csv
```

```
WARNING: Could not detect a GPU. Neural Network generation will be slow.

Brainome Daimensions(tm) 0.99 Copyright (c) 2019 - 2021 by Brainome, Inc. All Rights Rese
rved.
Licensed to:              Alexander Makhratchev  (Evaluation)
Expiration Date:          2021-04-30   57 days left
Number of Threads:        1
Maximum File Size:        30 GB
Maximum Instances:        unlimited
Maximum Attributes:       unlimited
Maximum Classes:          unlimited
Connected to:             daimensions.brainome.ai  (local execution)



Command:
    btc -measureonly cancer_train.csv
```

```
Start Time:                 03/04/2021, 04:17


Data:
    Input:                      cancer_train.csv
    Target Column:              Class
    Number of instances:        559
    Number of attributes:       9
    Number of classes:          2
    Class Balance:              0: 63.15%, 1: 36.85%

Learnability:
    Best guess accuracy:          63.15%
    Data Sufficiency:            Maybe enough data to generalize. [yellow]

Capacity Progression:            at [ 5%, 10%, 20%, 40%, 80%, 100% ]
    Optimal Machine Learner:           3,   4,   4,   4,   5,   5


Estimated Memory Equivalent Capacity for...
    Decision Tree:                    12 parameters
    Neural Networks:                   6 parameters
    Random Forest:                    13 parameters

Risk that model needs to overfit for 100% accuracy using...
    Decision Tree:                  4.70%
    Neural Networks:               11.13%
    Random Forest:                 11.50%

Expected Generalization using...
    Decision Tree:                  43.36 bits/bit
    Neural Network:                 45.17 bits/bit
    Random Forest:                  43.00 bits/bit


Recommendations:




Time to Build Estimates:
    Decision Tree:              a few seconds
    Neural Network:               2 minutes

Messages:
Warning: Remapped class labels to be contiguous. Use -cm if DET/ROC-based accuracy measur
ements are wrong.
```

2. Build the Predictor

Based on our measurements, Daimensions recommends we use a decision tree, which has lower risk of overfit and higher generalization for this dataset. We are also using -rank to prioritize certain attributes from our data, and we'll look at which attributes Daimensions decides are important later.

In [2]:

```
! btc -v -v -f DT cancer_train.csv -o cancer_predict.py -e 10 -rank --yes
```

```
WARNING: Could not detect a GPU. Neural Network generation will be slow.

Brainome Daimensions(tm) 0.99 Copyright (c) 2019 - 2021 by Brainome, Inc. All Rights Rese
rved.
Licensed to:              Alexander Makhratchev  (Evaluation)
Expiration Date:          2021-04-30   56 days left
Number of Threads:        1
Maximum File Size:        30 GB
Maximum Instances:        unlimited
Maximum Attributes:       unlimited
```

```
Maximum Attributes:       unlimited
Maximum Classes:          unlimited
Connected to:             daimensions.brainome.ai  (local execution)


Command:
    btc -v -v -f DT cancer_train.csv -o cancer_predict.py -e 10 -rank --yes

Start Time:                 03/05/2021, 17:39



Attribute Ranking:
    Important columns:          Uniformity_of_Cell_Shape, Bare_Nuclei, Clump_Thickness, N
ormal_Nucleoli, Uniformity_of_Cell_Size
    Overfit risk:                  0.0%
    Ignoring columns:           Marginal_Adhesion, Single_Epithelial_Cell_Size, Bland_Chr
omatin, Mitoses
    Test Accuracy Progression:     Uniformity_of_Cell_Shape :   91.59%
                                   Bare_Nuclei :   95.17%  change   +3.58%
                                   Clump_Thickness :   96.24%  change   +1.07%
                                   Normal_Nucleoli :   97.14%  change   +0.89%
                                   Uniformity_of_Cell_Size :   97.32%  change   +0.18%



Data:
    Input:                      cancer_train.csv
    Target Column:              Class
    Number of instances:        559
    Number of attributes:       5
    Number of classes:          2
    Class Balance:              0: 63.15%, 1: 36.85%

Learnability:
    Best guess accuracy:          63.15%
    Data Sufficiency:            Not enough data to generalize. [red]

Capacity Progression:            at [ 5%, 10%, 20%, 40%, 80%, 100% ]
    Optimal Machine Learner:           2,   3,   4,   4,   4,   5


Estimated Memory Equivalent Capacity for...
    Decision Tree:                      5 parameters
    Neural Networks:                    6 parameters
    Random Forest:                     15 parameters

Risk that model needs to overfit for 100% accuracy using...
    Decision Tree:                 1.97%
    Neural Networks:             100.00%
    Random Forest:                13.04%

Expected Generalization using...
    Decision Tree:                103.31 bits/bit
    Neural Network:                44.67 bits/bit
    Random Forest:                 37.27 bits/bit


Recommendations:


    Note: Machine learner type DT given by user.


Time to Build Estimates:
    Decision Tree:             a few seconds
```

```
System Meter:                          cancer_predict.py
    Classifier Type:                   Decision Tree
    System Type:                       Binary classifier
    Training/Validation Split:           60% : 40%
    Accuracy:
        Best-guess accuracy:               63.14%
        Training accuracy:                 97.01% (325/335 correct)
        Validation Accuracy:               97.76% (219/224 correct)
        Overall Model Accuracy:            97.31% (544/559 correct)
        Improvement over best guess:       34.17% of possible  36.86%

    Model Capacity (MEC):                      5 bits
    Generalization Ratio:                  62.09 bits/bit
    Model Efficiency:                       6.83 /parameter
    Generalization Index:                  30.44
    Percent of Data Memorized:              3.29%

    Training Confusion Matrix (count):
                   2 |  202    7
                   4 |    3  123

    Validation Confusion Matrix (count):
                   2 |  139    5
                   4 |    0   80

    Full Confusion Matrix (count):
                   2 |  341   12
                   4 |    3  203

    Accuracy by Class:
               class |   TP   FP   TN   FN     TPR      TNR      PPV      NPV       F1
TS
                   2 |  341   12  203    3   99.13%   98.54%   96.60%   98.54%   97.85%
95.79%
                   4 |  203    3  341   12   94.42%   96.60%   98.54%   96.60%   96.44%
93.12%

End Time:
Runtime Duration:

Output: cancer_predict.py
READY.
Messages:
Note: Class labels required remapping onto contiguous integers. Mapped as follows: {'2':
0, '4': 1}
Warning: Remapped class labels to be contiguous. Use -cm if DET/ROC-based accuracy measur
ements are wrong.
```

3. Validate the Model

Now we can validate our model on a separate set of data that wasn't used for training.

In [4]:

```
! python3 cancer_predict.py -validate cancer_valid.csv
```

```
Classifier Type:                   Decision Tree
System Type:                       Binary classifier
Best-guess accuracy:               75.00%
Model accuracy:                    99.28% (139/140 correct)
Improvement over best guess:       24.28% (of possible 25.0%)
Model capacity (MEC):              5 bits
Generalization ratio:              22.55 bits/bit
Model efficiency:                  4.85%/parameter
System behavior
```

```
True Negatives:                     74.29% (104/140)
True Positives:                     25.00% (35/140)
False Negatives:                    0.00% (0/140)
False Positives:                    0.71% (1/140)
True Pos. Rate/Sensitivity/Recall:  1.00
True Neg. Rate/Specificity:         0.99
Precision:                          0.97
F-1 Measure:                        0.99
False Negative Rate/Miss Rate:      0.00
Critical Success Index:             0.97
Confusion Matrix:
 [74.29% 0.71%]
 [0.00% 25.00%]
```

4. Learn From Attribute Rank

From validating the data, we can see that the predictor has 98.57% accuracy. This is great for making predictions on future data. However, what might be of greater interest is looking at the output from building our predictor, specifically the attributes that Daimensions decided to use. Under the section of output called "Attribute Rank," Daimensions has listed the attributes used: Uniformity_of_Cell_Size, Bare_Nuclei, Clump_Thickness, Marginal_Adhesion, Mitoses, and Uniformity_of_Cell_Shape. This information about what attributes were the best predictors of malignant cancer cells is valuable to scientists looking for the causes of this cancer.

Citation

This breast cancer databases was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

Sources:

- **Dr. WIlliam H. Wolberg (physician), University of Wisconsin Hospitals, Madison, Wisconsin, USA**
- **Donor: Olvi Mangasarian (mangasarian@cs.wisc.edu), received by David W. Aha (aha@cs.jhu.edu)**
- **Date: 15 July 1992**