AI-Based Medical Diagnosis: Bayesian Networks and Gaussian Processes

line 1: 1st Given Name Surname   
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 4th Given Name Surname  
line 2: *dept. name of organization*  
*(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 2nd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 5th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 3rd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 6th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

Abstract—This assessment focuses on implementing a software application for Medical Diagnosis using Bayesian Networks and Gaussian Processes. The datasets for Cardiovascular Disease and Diabetes include various features such as age, weight, glucose, and more. The task involves implementing Bayesian Networks to perform probabilistic reasoning, answering queries like disease prediction based on given parameters. The solution, implemented in Python, utilizes either discrete or Gaussian Bayesian Networks. The process includes reading datasets, defining Bayesian network structures, learning parameters, and answering probabilistic queries. Performance evaluation involves metrics like disease classification accuracy, AUC score, statistical distances, and training/test times. Task 2 compares the performance against Gaussian Processes, emphasizing consistent metric usage.

# Introduction

This task involves leveraging Artificial Intelligence (AI) techniques to create a software application for medical diagnosis. By utilizing Bayesian Networks and Gaussian Processes, the aim is to perform probabilistic reasoning on dataset Diabetes. This involves implementing algorithms to predict outcomes based on various factors like age, glucose levels, and Pregnancies and more. The target for our data is Outcome which has two possible values. The other features that were in continuous format, we have discretized them.

# DAtaset Information

## Diabetes Data

**Reason for using Diabetes Data:** Cardiovascular Disease datasets often contain a broader range of health factors, making it more complex. By focusing on diabetes, which has a more specific set of relevant variables, model development and interpretation might be more straightforward.

This dataset aims to explore how these variables interrelate and their collective influence on predicting diabetes outcomes. The information contained in these variables helps construct predictive models to assist in diagnosing or predicting the likelihood of diabetes based on an individual's health profile. The name of the dataset is **DiabetesPrediction** and it contain the following features: BP(BloodPressure),P(Pregnancies),G(Glucose),ST(SkinThickness),I(Insulin),BMI(BMI),DPF(DiabetesPedigreeFunction),A(Age),O(Outcome).

## Using Discretized Data over continuous data

Converting continuous features to discrete values in Bayesian Networks or Naive Bayes aids in simplifying calculations and handling assumptions of independence among variables. This transformation reduces computational complexity, making the model more robust to noisy data and improving interpretability.

# **Task 1**

# **Bayesian Networks**

In this task we are going to imeplement Bayesian Networks to answer probabilistic queries such as:

P(outcome=0|glucose=109, bmi=25.4, age=25)

P(outcome=1|glucose=183, bmi=23.3, age=58)

## **Bayesian Networks**

## Bayesian networks are a type of probabilistic graphical model comprised of nodes and directed edges.[1]

## Bayesian network models capture both conditionally dependent and conditionally independent relationships between random variables.

## Models can be prepared by experts or learned from data, then used for inference to estimate the probabilities for causal or subsequent events.

## **Choices with Proper Reasons to Solve task 1**

* We used the provided code during workshops to read our csv files to get the better and deep understanding of the code and concepts about reading files manually instead of using libraries.
* As described in task we use the **alarm-config.txt** file as sample to create the configuration of Diabetes data and bayes net algorithm. And we have used two different structures each provided in separate file. The files names are d-c.txt and d-cc.txt.
* We learnt the parameters of the Bayesian networks using Maximum Likelihood Estimation and write them to configuration files.
* Then we used probabilistic queries to test and answer.

# **The Performance metrics for Bayes Nets**

1. **Balanced Accuracy:** It measures the accuracy of a classifier on imbalanced datasets. It considers both true positives and true negatives, giving a balanced view of the classifier's performance across all classes.
2. **F1 Score:** It's the harmonic mean of precision and recall, providing a balance between them. It's a good measure when the class distribution is imbalanced.
3. **Area Under Curve (AUC)**: This metric assesses the model's ability to distinguish between classes. The AUC value closer to 1 indicates a better model performance.
4. Brier Score: It quantifies the accuracy of probabilistic predictions. Lower scores indicate better predictions; it's commonly used for probabilistic classifiers.
5. **KL Divergence (Kullback-Leibler Divergence):** It measures how one probability distribution diverges from another. In this context, it indicates the difference between predicted and true distributions.
6. **Training Time:** It represents the time taken to train the Bayesian network, which, in this case, should ideally come from the CPT\_Generator that constructs the network.
7. **Inference Time**: It measures the time taken to perform inference or make predictions using the trained Bayesian network on new data, here indicating a quick inference process.
8. Each metric provides specific insights into different aspects of the model's performance, such as accuracy, predictive power, and computational efficiency. Evaluating these metrics together gives a comprehensive view of how well the Bayesian network performs.

**Metrics Values of Bayes Net**

* Balanced Accuracy=0.7310039725532684
* F1 Score=0.6506024096385542
* Area Under Curve=0.8216383772721801
* Brier Score=0.1681462012072536
* KL Divergence=150.03238547663432
* Inference Time=0.03534889221191406 secs.

**Task 2-Guassian Process**

# **Guassian Process**

 Gaussian processes provide a mechanism for directly reasoning about the high-level properties of functions that could fit our data [2]. For example, we may have a sense of whether these functions are quickly varying, periodic, involve conditional independencies, or translation invariance. Gaussian processes enable us to easily incorporate these properties into our model, by directly specifying a Gaussian distribution over the function values that could fit our data.

# **The Performance metrics for Gaussian Process**

* Balanced Accuracy=0.7545454545454545
* F1 Score=0.673076923076923
* Area Under Curve=0.8772727272727273
* Brier Score=0.13580481561357483
* KL Divergence=40.83645060434881

# **The Performance COMPARISON of Bayes Nets and Gaussian Process**

1. performance comparison

| Table Head | Models | | |
| --- | --- | --- | --- |
| Bayes Nets | Gaussian Process |  |
| Accuracy | 0.7310039725532684 | 0.754545454545454 |  |
| F1 Score | 0.650602409638554 | 0.6730769230769 |  |
| Area Under Curve | 0.8216383772721801 | 0.877272727272727 |  |
| Brier Score | 0.16814620120725 | 0.135804815613574 |  |
| KL Divergence | 150.0323854766343 | 40.836450604348 |  |

# **The Examples of inferencing queries using Bayes Nets and Gaussian Process**

## Bayes Nets

**Query. 1**

alg\_name = 'InferenceByEnumeration'

file\_name = 'BayesNet/d-c.txt'

prob\_query = 'P(BloodPressure|Pregnancies=3,Outcome=1)'

**Outcome**

**unnormalised P(BloodPressure**)={'3': 0.008408864949253816, '2': 0.0037219566168828367, '1': 0.00020677536760460203, '0': 0.0008960265929532754, '4': 0.0015163526957670816, '5': 0.0003446256126743367}

**normalised P(BloodPressure)**={'3': 0.5570776255707762, '2': 0.2465753424657534, '1': 0.0136986301369863, '0': 0.0593607305936073, '4': 0.1004566210045662, '5': 0.0228310502283105}

**Execution Time:** 0.027605056762695312

**Query. 2**

alg\_name = 'InferenceByEnumeration'

file\_name = 'BayesNet/d-c.txt'

prob\_query = 'P(BloodPressure|Age=3,Outcome=0)'

**Outcome**

**unnormalised P(BloodPressure)**={'3': 0.007900926596681715, '2': 0.006548896376928695, '1': 0.00042250944367281903, '0': 0.0006337641655092286, '4': 0.0010985245535493297, '5': 0.00012675283310184572}

**normalised P(BloodPressure)**={'3': 0.47222222222222227, '2': 0.3914141414141415, '1': 0.02525252525252526, '0': 0.03787878787878789, '4': 0.06565656565656568, '5': 0.007575757575757579}

**Execution Time**: 0.03949689865112305

## Gassian Process

**Query. 1**

alg\_name = 'InferenceByEnumeration'

file\_name = 'BayesNet/d-c.txt'

prob\_query = 'P(BloodPressure|Pregnancies=3,Outcome=1)'

**Outcome**

**unnormalised P(BloodPressure)**={'3': 0.008408864949253816, '2': 0.0037219566168828367, '1': 0.00020677536760460203, '0': 0.0008960265929532754, '4': 0.0015163526957670816, '5': 0.0003446256126743367}

**normalised P(BloodPressure)**={'3': 0.5570776255707762, '2': 0.2465753424657534, '1': 0.0136986301369863, '0': 0.0593607305936073, '4': 0.1004566210045662, '5': 0.0228310502283105}

**Execution Time**: 0.019362211227416992

**Query. 2**

alg\_name = 'InferenceByEnumeration'

file\_name = 'BayesNet/d-c.txt'

prob\_query = 'P(BloodPressure|Age=3,Outcome=0)'

**Outcome**

**unnormalised P(BloodPressure)**={'3': 0.007900926596681715, '2': 0.006548896376928695, '1': 0.00042250944367281903, '0': 0.0006337641655092286, '4': 0.0010985245535493297, '5': 0.00012675283310184572}

**normalised P(BloodPressure)**={'3': 0.47222222222222227, '2': 0.3914141414141415, '1': 0.02525252525252526, '0': 0.03787878787878789, '4': 0.06565656565656568, '5': 0.007575757575757579}

**Execution Time:** 0.03513360023498535

# **Conclusions**

* Gaussian Processes show slightly higher balanced accuracy, indicating better overall classification performance.
* GP performs better in terms of precision and recall balance.
* GP demonstrates superior performance in distinguishing between classes based on the ROC curve.
* GP has a lower Brier Score, indicating better accuracy of probabilistic predictions.
* GP shows significantly lower divergence, suggesting a closer match between predicted and true distributions.

Overall, Gaussian Process model tends to outperform Bayesian Networks across most metrics, particularly in classification accuracy, predictive power, and matching distributions.

# **References**

[1] **Brownlee, Jason. 2019.** introduction-to-bayesian-belief-networks. *machinelearningmastery.com.* [Online] September 25, 2019. https://machinelearningmastery.com/introduction-to-bayesian-belief-networks/.

[2] chapter\_gaussian-processes/gp-intro. *d2l.ai.* [Online] https://d2l.ai/chapter\_gaussian-processes/gp-intro.html.