**MSc Project Report**

16th September 2019

John H Booth (13133420)

BBK\_BUCI058D7\_1819 - MSc Data Science Project

MSc in Data Science

Department of Computer Science and Information Systems, Birkbeck College, University of London, 2019

Identifying Feature Importance in Pediatric Post-Mortem Outcome with Machine Learning Models

Academic Declaration

I have read and understood the sections of plagiarism in the College Policy on assessment offences and confirm that the work is my own, with the work of others clearly acknowledged. I give my permission to submit my report to the plagiarism testing database that the College is using and test it using plagiarism detection software, search engines or meta-searching software.

The report may be freely copied and distributed provided the source is explicitly acknowledged.

Acknowledgements

Contents

[1 Abstract 4](#_Toc16663059)

[2 Introduction 5](#_Toc16663060)

[2.1 Background 5](#_Toc16663061)

[2.2 Literary Review 5](#_Toc16663062)

[2.3 Report outline 5](#_Toc16663063)

[2.4 Aim and Objectives 5](#_Toc16663064)

[3 Methods 7](#_Toc16663065)

[4 Data 8](#_Toc16663066)

[4.1 Source 8](#_Toc16663067)

[4.2 Data Cleaning 8](#_Toc16663068)

[4.3 Missing data 8](#_Toc16663069)

[4.4 Normalisation vs Standardisation 8](#_Toc16663070)

[5 Analysis 9](#_Toc16663071)

[5.1 Create models 9](#_Toc16663072)

[5.2 Tune models 9](#_Toc16663073)

[6 Results 10](#_Toc16663074)

[7 Conclusion 11](#_Toc16663075)

[7.1 Project Summary 11](#_Toc16663076)

[7.2 Project Evaluation 11](#_Toc16663077)

[7.3 Recommendations for Future Work 11](#_Toc16663078)

[References 12](#_Toc16663079)

[Appendix A – Example Project Code 13](#_Toc16663080)

[Appendix B – Example RDV structure 13](#_Toc16663081)

[Appendix C – Detailed output from and analytic process 13](#_Toc16663082)

[Appendix D – Deliverables on Attached CD ROM 13](#_Toc16663083)

[Random Thoughts 14](#_Toc16663084)

# Abstract

Post mortems are complex procedures that utilise a significant amount of hospital resources, yet despite this, cause of death is only determined in 45% of cases. The event itself can be very traumatic for the parents of the child, yet is essential for providing further clinical understanding of the patient’s cause of death. Given this, there is an imperative to extract the greatest possible value from the data. Here, we investigated whether machine learning could be used to derive novel insights from the prediction of post mortem outcomes.

A post mortem database containing 7000 records across 300 variables was analysed and categorised into stage of examination (external and internal). The outcome of the examination was summarised as either ‘cause of death determined’ or ‘not determined’. From these summarised data, cases were filtered by children aged <= 2 years, resulting in a dataset of 3,100 post mortems.

Following this, decision tree, random forest, and gradient boosting machine models were iteratively built for each stage of the post-mortem examination and compared using their accuracy metrics.

The naïve decision tree model using external examination data had a predictive performance of 67%. Model performance notably increased when trained on internal examination data. At each stage of the examination, a core set of data items, of which the final set included age, BMI, and heart weight were highlighted using model feature importance as key variables for determining post mortem outcome. The use of increasingly complex modelling techniques was able to boost the predictive performance of the model by as much as 10%.

This project clearly shows the value of collecting clinical procedural data which can then be modelled using machine learning techniques to inform clinical practice. With more time, further modelling, including unsupervised clustering could be undertaken to derive further insights.

Supervisor:

Nigel Martin (nigel@dcs.bbk.ac.uk)

Department of Computer Science and Information Systems, Birkbeck College, University of London, London, UK.

# Introduction

## Background

Great Ormond Street Hospital for Children NHS Trust (GOSH) is the country’s leading centre for treating sick children. With the UCL Great Ormond Street Institute of Child Health, GOSH is the largest centre for paediatric research outside the US.

Specialist Paediatric Pathologists perform perinatal, infant and childhood post-mortems including hospital referrals, forensic cases and those on behalf of Her Majesty’s Coroner.

The Pathology Department has established a research database containing details of all post-mortems performed between 1996 and 2017.  The database was originally used specifically for research into Sudden Unexpected Death in Infancy (SUDI). Since then it has been utilised for a number of other projects investigating SUDI, stillbirths and various aspects of paediatric autopsy procedure.

Currently the database holds 7000 records, each record representing an individual post-mortem. Up to 300 items of data can be defined for each post-mortem.

The purpose of this project is to use data science analytic techniques to develop operational strategies that can be applied to paediatric post-mortems to prioritise which data is required to achieve the target of specifying the cause of death.

## Literary Review

Synopsis of project proposal but NOT a copy. Concentrate on parts that are directly relevant to the project.

## Report outline

Highlight some key aspects of the report, include some interesting graphics clearly annotated.

NB Glossary of terms that cover the basic healthcare concepts covered in this document.

## Aim and Objectives

Take from project proposal; amended based on experience undertaking project?

The aims of this project are:

1. To develop a routine to extract data from the existing Post-mortem Research Database into an entity attribute value schema that will make the data more readily available for data analytics.
2. To apply the Decision Tree Analytical method to the extracted data to develop operational strategies that can be applied to paediatric post-mortems to prioritise which data is required to achieve the target of specifying the cause of death.
3. To investigate ensemble strategies, specifically Random Forests and Gradient Boosting to see how these techniques can improve on the basic Decision Tree method.
4. If time permits a final aim would be to consider if the approach of Neural Networks could be employed to enhance the results.

# Methods

## Project Pipeline

This overall project has been divided into a number of sections the output of each section provides the input for the following section to form the project pipeline. The process for each section will be coded in an appropriate environment described below but with the overall aim of creating a fully reproducible set of procedures that lead to a set of results.

A GIT hub repository has been created for the project and all project code, documents and images are stored and versioned in this repository. A link to the GIT repository is given in Appendix D - Deliverables.

The sections have been divided into two major sections; Data Engineering and Analytics.

## Data Engineering

The data engineering aspects of this project will be undertaken using the Python programing language, a general purpose programing language that used extensively in the world of data science. The development of python procedures will be carried out using PyCharm an integrated development environment (IDE).

The data manipulation carried out during this stage will be using structured query language (SQL) and will be instigated using the Python package PyODBC which allows the connection to external databases using ODBC connections and the production and return of SQL queries.

Where appropriate the data engineering code will be broken down into functions that will be unit tested prior to implementation. All processes will also be developed with integral profiling so that any bottlenecks can be identified and if possible their effect reduced so that the overall processing of the data can be as efficient as possible.

### Extract, Transform, Load (ETL) Process

The fundamental section of this stage is the ETL process on the Post Mortem Research database into the Health Analytics Schema (HAS) model using the EAV schema.

The basic structure will be:

* Create HAS Tables
* Create Concepts
* Create Patients and Staff
* Create Events
* Create Event Attributes

A detailed breakdown of the python code developed for the ETL process is given in Appendix B – ETL Process.

The output of this section will be the HAS database created and populated from the originating system.

### Creation of Summary and Reporting Attributes

Having created the base research data from the original data then a number of summary event attributes will be created for reporting and analytic purposes:

* Number of Attributes(ATTRIBUTES)
  + The number of event attributes each event has.
* Cause of Death Summary (COD2\_SUMM)
  + A summary of the COD2 attributes into:
    - Not Determined
    - Determined
    - Unknown
    - Not available
  + More details in the following Data section of the report.
* Macro and Histological Body System Attributes
  + Individual organ internal macro and histological examination results are summarised at the body system level for ease of analytics.
* External and Internal Examinations
  + A simple flag to indicate whether an individual post mortem event had had an external and or internal examination.

The output of this stage will be the addition of a number of event attributes added to the existing set of events and their originating attributes.

### Identifying Data to be included in this study

This process will be split into 2 stages:

* Include or exclude data
  + COD2\_summ
    - Only include events where the COD2\_summ is either Not Determined or determined.
  + Age category
    - Only include events for the following age categories:
      * Early Neonatal
      * Neonatal
      * Infant
      * Child - under the age of 2 years
  + Measurement Outliers
    - Any numerical values that fall outside what is physically possible.
* Identify for the 4 stages of the post mortem being considered in this study which features should be included:
  + External
  + Internal Stage 1 (Organ weights)
  + Internal Stage 2 (Macro examination)
  + Internal Stage 3 (Histological examination)

At this stage the issue of missing data for any chosen event will not be addressed.

The output of this stage will be four research data views (RDVs), one for each stage of the post mortem, in the form of CSV files.

### Data Wrangling

The final section of the data engineering stage will be to produce the data in the format most appropriate for analytics. Two forms of data wrangling will be used:

* One-hot encoding – Categorical features
  + Rather than each categorical feature having a single column of data with the appropriate category; each category has its own column with either a 1 or 0 depending on whether each event has that feature value.
* Numerical normalisation – Numeric features
  + Each numeric value will be normalised based on their predicted value for the age of the patient described by each event. This routine means that each numeric value will be in the range 0 – 1 with only outliers having larger values.

It should be notes that Z-Score standardisation of the numeric data was considered but not pursued as it didn’t take into account the age of the patient in each event.

The output of this section will be four adjusted RDVs, one for each stage of the post mortem, in the form of CSV files.

The detailed structure of the RDVs in both formats is shown in Appendix C – RDV Structure.

## Analytics

The analytic aspects of this project will be undertaken using the R programming language a language specifically developed for statistical computing. The development of R scripts will be carried out in R Studio an IDE for the R language.

In this section I will describe the key packages that I will be using and the specific parameters that have to be tuned to obtain an optimised model.

A basic tuning procedure will be adopted for all three modelling packages:

* Create model using default parameters for each post mortem stage
* Define a range for each parameter to be tuned.
* Change each parameter one by one and obtain an optimal value based on predictive accuracy.
* Repeat last step to see whether any changes in the parameters significantly affects each parameter.
* Finalise a set of parameters for each post mortem stage for each model.

The output for each of the modelling stages is an R function that can be called for that model with a training/test split for each post mortem stage. The function will save the resulting confusion matrices and relative feature importance in CSV files as well as plots specific to each model as PNG files.

### Visualisation: ggplot2

Ggplot2 is the principal graphics package used within R and is part of the tidyverse, a collection of packages aiming to bring some semblance of order in the slightly anarchic world of R programing.

This section has three main aims:

* Visualisation of the complete post mortem data set
* Visualisation of the sub set of data to be used for this study
* Develop a basic graphical framework that can be used for all images produced by the various further analytic sections.
  + Colour scheme – viridis a colour blind friendly colour palette
  + Theme.classic – a very basic no frills plotting theme.
  + PNG file naming convention for saving plots

The output of this section are two frames of visualisations saved as PNG files.

### Decision Tree: rpart

The rpart package uses recursive partitioning on trees, both classification and regression to achieve an optimum level of complexity for a given set of data.

The main hyper-parameters that can be tuned are:

* minsplit - the minimum number of observations that must exist in a node in order for a split to be attempted
* minbucket - the minimum number of observations in any terminal node. Use minsplit / 3
* cp – complexity parameter, used to define further pruning after the initial tree is produced

### Random Forests: randomForest

Classification and regression based on a forest of trees using random inputs, based on Breiman (2001)

The main hyper-parameters that can be tuned are:

* Mtry - Number of candidates draw to feed the algorithm. By default, it is the square of the number of columns.
* Maxnodes - Set the maximum amount of terminal nodes in the forest
* ntree - number of trees in the forest

### Gradient Boosted Decision Tree: xgboost

xtreme Gradient Boosting, which is an efficient implementation of the gradient boosting framework from Chen & Guestrin (2016)

The main hyper-parameters that can be tuned are:

* Eta – controls how much information from a new tree is used in boosting.
* max\_depth – controls the maximum depth of a tree.
* gamma - Controls the minimum reduction in the loss function required to grow a new node in a tree.
* min\_child\_weight - Controls the minimum number of observations (instances) in a terminal node.
* Subsample - This parameter determines if we are estimating a Boosting or a Stochastic Boosting.
* colsample\_bytree – Number of features to sample in each new tree.

### Combined Results

Using the functions developed for each model then each model can be run for each post mortem stage for a number of random seeds deriving the training/test data split. The CSV files from each model run can then be combined to produce:

* A comparison of model predictive accuracy for changing random seeds
* A comparison of the change in predictive accuracy of each model at each stage of the post mortem.
* A comparison of relative feature importance changes for different random seeds for each stage of the post mortem
* A final predictive accuracy of cause of death determined or not for each model at each stage of the post mortem. The predictive of accuracy of both not determined and determined cause of death can also be identified by model by stage of post mortem.
* A final set of relative feature importance by model by stage of post mortem.

# Data

Explain why the data matches the objectives.

## ETL Process

What are the main points that I want to get over:

* The original database structure is optimised for data recording.
  + Tables are divided up into subject groups and the overall structure is defined by primary and foreign keys.
  + Lots of blank fields
  + Difficult to comprehend how much data there really is.
  + 134 look-up tables no easy way to understand what links to where.
* Create EAV structure
  + Many fewer tables
  + Concepts table
    - Stores the hierarchical structure of the look-ups
    - Also defines to where they are linked.
  + Extract patient details into patient and patient attribute tables
  + Every post mortem is represented as a single event
  + Every field of data is represented as an event attribute with the event attribute type linked back to the concepts table. In the case of look-up values then the value is linked back to concepts table also. Every event attribute has a value; there are no NULL values in the EAV model.

Development process

* Create HAS\_Tables.py
* Created a separate analytic database to house the HAS schema with linked tables back to the PM Research database. This structure proved to be very inefficient and the bottleneck of the linked tables was identified.
* Changed procedure so that HAS tables still housed in separate database but the PM research database was connected to directly this provided a 10 fold improvement in processing time.
* The details of the EAV process are given in Appendix B.

## Creation of Summary and Reporting Attributes

What are the points that I want to get over:

* How easy it is to create additional attributes from existing ones
  + No structural changes
  + Clearly defined structure

Development process

* Modify\_events.py
* Process is creating the COD2\_SUMM attribute value type concept
  + Create COD2\_SUMM Concepts
  + Extract all events with event attribute of type COD2\_COD2ID
  + For each event
    - Apply mapping to COD2\_SUMM
    - Create new event attribute of type COD2\_SUMM
* Process for creating ATTRIBUTES
  + Create ATTRIBUTES concept value type numeric
  + Select all event attributes group by event count(attributes)
  + For each event
    - Create new event attribute of type ATTRIBUTES with a value of count(attributes)
* Similar processes were developed for:
  + system\_name + macro\_SyFiID and system\_name + histo\_SyHiID
    - Individual organ internal macro and histological examination results are summarised at the body system level.
    - For each system the results for the individual organs were noted and the maximum result, 1-3, for any organ was assigned to the system.
      * 001 - Normal
      * 002 - Abnormal but not COD
      * 003 - Abnormal COD
      * 999 - Other
  + ExternalExam and InternalExam
    - A simple flag to indicate whether an individual post mortem event had had an external and or internal examination.
      * True/False

## Identifying Data to be included in this study

What are the points that I want to get over:

* How easy it is to identify which events to include based on their event attributes

Development process

## Data Wrangling

* Z score
* Normalisation for Age
* One-hot encoding

# Analysis

Explain what analysis was undertaken and why.

## Missing data

* Gestation in days at birth.
* Balanced data

## Create models

## Tune models

# Results

Describe the results of the analysis.

# Conclusion

## Project Summary

Provide an overview/summary of your work and findings.

## Project Evaluation

Identify areas for improvement; discuss what you could have done better (particularly important if you failed some of your targets or your results as not as expected)

## Recommendations for Future Work

# References

Archer, K.J., 2010. rpartOrdinal: an R package for deriving a classification tree for predicting an ordinal response. *Journal of Statistical Software*, *34*, p.7.

Luellen, J.K., Shadish, W.R. and Clark, M.H., 2005. Propensity scores: An introduction and experimental test. *Evaluation Review*, *29*(6), pp.530-558.

(Example of using RPART package)

Therneau, T.M. and Atkinson, E.J., 1997. An introduction to recursive partitioning using the RPART routines.

List of reference to be included:

Python

Pycharm

pyODBC

R

R studio

ggplot2

The viridis color palettes:

<https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html>

Rpart

<https://cran.r-project.org/web/packages/rpart/index.html>

Randomforest

<https://cran.r-project.org/web/packages/randomForest/>

<https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>

XGBoost

<https://cran.r-project.org/web/packages/xgboost/>

<https://cran.r-project.org/web/packages/xgboost/xgboost.pdf>

Tuning xgboost in R: Part I

<https://insightr.wordpress.com/2018/05/17/tuning-xgboost-in-r-part-i/>

# Glossary of Terms

From the Royal College of Pathologists

<https://www.rcpath.org/discover-pathology/what-is-pathology/glossary-of-terms.html>

|  |  |
| --- | --- |
| Term | Description |
| Data Wrangling | the process of transforming and mapping data from one "raw" data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. |
| ETL | The process of pulling data out of one database and place it into another database. Within this process the data is cleans and transformed to be in a more appropriate schema for analytics. |
| Histopathology | The branch of pathology that involves looking at tissue under the microscope to diagnose disease. If you have a mole or a breast lump removed, the histopathologist will examine it to work out what it is. |
|  |  |
| Metabolic | A group of overlapping areas of clinical practice with a common dependence on the detailed understanding of basic biochemistry and medicine. These areas fall within the territory of both physicians and chemical pathologists. They include clinical nutrition, lipid abnormalities, diabetes, metabolic bone disease, porphyria and adult inherited metabolic disorders. |
| Microbiology | The diagnosis of infection caused by bacteria, fungi, parasites and viruses; identification of the best treatment options for infection; and the monitoring of antibiotic resistance. It also includes testing for how well a patient is responding to treatment of infection. |
| SQL |  |
|  |  |

# Appendix A – Example Project Code

Python Functions

Python ODBC

R Model

# Appendix B – ETL Process

# Appendix C –RDV structure

# Appendix C – Detailed output from an analytic process

RDM file output

# Appendix D – Deliverables on Attached CD ROM

# Random Thoughts

* Python vs R
  + Why used when
* Reproducible results
* R code; evolved through three stages
  + Individual model
  + Model run for all PM stages
  + Model as a function called with multiple random keys.
* Balanced data
* Methods before data?
  + Include Software architecture in methods?
    - Tools and programing language
      * Why use R over Python?
    - Project pipeline
      * PM Research Database => EAV Schema => Summary and Study based Attributes => RDV production => RDV Adjustments => Run models => review results
  + Testing and Profiling
* Agile approach
  + Reviewed progress from last week
  + Set objectives for the coming week