Stroke Data Write-Up

Tom Matcham

22/03/2020

```
knitr::opts_chunk$set(echo = T)

# Libraries used

{
  library(DOT)
  library(DMwR)
  library(ROCR)
  library(sbrl)
  library(knitr)
  library(mclust)
  library(rcausal)
  library(cutpointr)
  library(tidyverse)
  library(kableExtra)

  options(scipen = 999)
}
```

In this write-up we explore the Kaggle stroke dataset. We start with some basic data exploration before building a predictive stroke model using Scalable Bayesian Rule Lists (SBRL) and finally using a causal discovery algorithm to explore causal relationships amongst variables.

Data Exploration

Let's load the data and convert all the integer columns to factors and they're really encoding binary data.

```
data = read_csv("train_2v.csv", col_types = cols()) %>%
    mutate(hypertension = as.factor(hypertension),
         heart_disease = as.factor(heart_disease),
         stroke = as.factor(stroke))
```

Missing values

```
 data %>%
  summarise_all(funs(100 * sum(is.na(.)) / n())) %>%
  gather() %>%
  filter(value > 0.0001) %>%
  kable %>%
  kable_styling
```

key	value
bmi	3.368664
smoking_status	30.626728

We see that only 3% of BMI data is missing so it may be reasonable just to remove these subjects from the analysis. Nearly a third of the smoking data is missing so we'll have to be a bit smarter about that.

Let's start by binning the numerical data. This will enable us to discover correlations between all features of the dataset and hopefully illuminate why so much of the smoking data is missing. Let's classify BMI as follows:

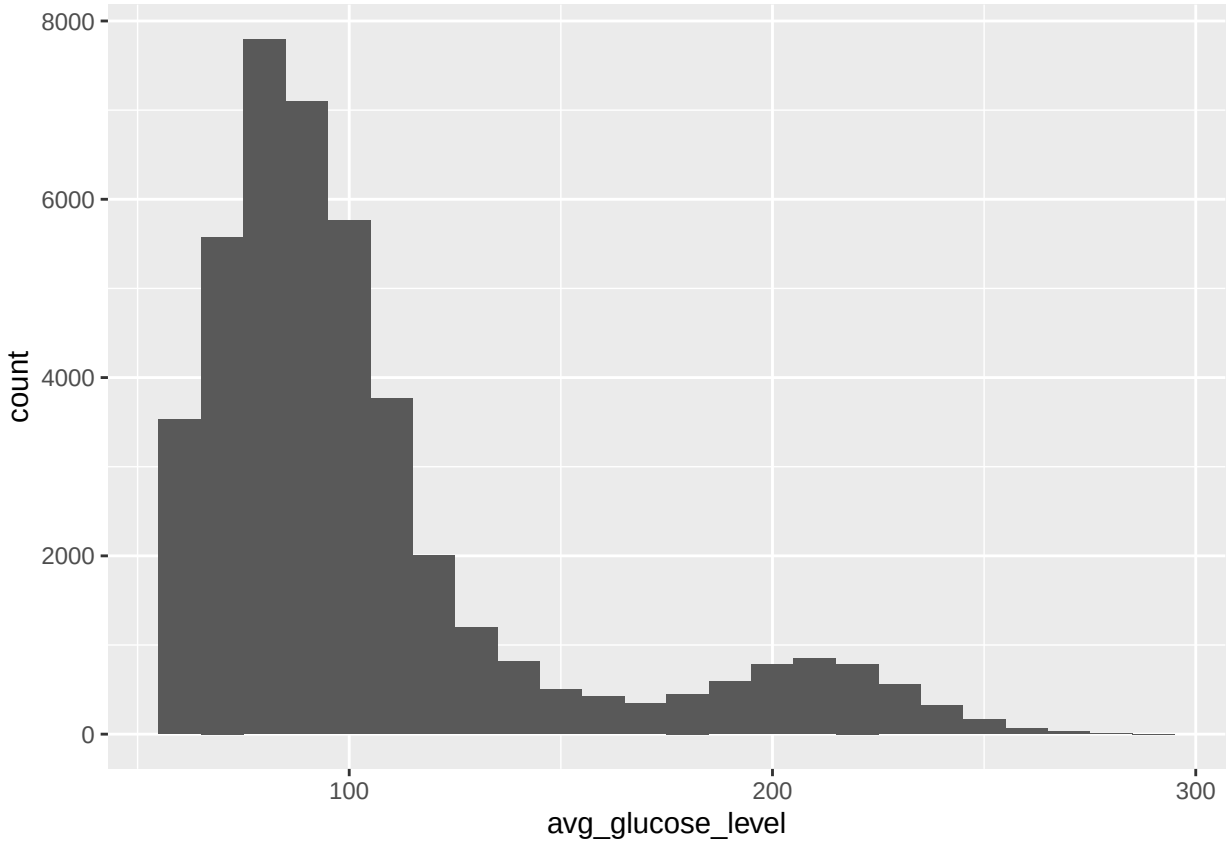
- Underweight: less than 18.5
- Normal weight: between 18.5 and 25
- Overweight: between 25–30
- Obese: greater than 30

We'll group age as follows. These bins were chosen partially for convenience and partially to avoid strata with few subjects:

- Child: 0 - 12 years old
- Teenager: 12 - 19 years old
- Young Adult: 19 - 35 years old
- Middle Aged: 35 - 55 years old
- Older adult: 55 - 65 years old
- Elderly: 65 years plus

Let's look for a reasonable way to group average blood glucose level.

```
ggplot(data, aes(avg_glucose_level)) +
  geom_histogram(binwidth = 10)
```



The distribution is bimodal and it seems a reasonable assumption that the higher mode is abnormal (possibly related to obesity?). I'll call the lower mode the 'normal' component and the upper mode the 'abnormal' component. For simplicity let's fit a non-parametric mixture model with two components, cluster the data based on those components then split by quartiles.

```
gmm = Mclust(data$avg_glucose_level, G = 2)

glucose_data = data.frame(classification = gmm$classification,
                          avg_glucose_level = data$avg_glucose_level)

quartiles = c(0.25, 0.5, 0.75)

quartile_funs = purrr::map(quartiles, ~partial(quantile, probs = .x, na.rm = TRUE)) %>%
  set_names(nm = map_chr(quartiles, ~paste0(.x*100, "%")))

glucose_quartiles_df = glucose_data %>%
  group_by(classification) %>%
  summarize_at(vars(avg_glucose_level), funs(!!!quartile_funs))

glucose_quartiles = glucose_quartiles_df %>%
  group_by(classification) %>%
  transpose %>%
  set_names(nm = .$classification)

glucose_quartiles_df %>%
  kable %>%
  kable_styling()
```

classification	25%	50%	75%
1	75.01	86.81	100.3975
2	158.45	196.12	216.8800

So let's use the following classifications for avg_glucose_level:

- If in component 1 and < lower quartile: normal_low
- If in component 1 and >= lower quartile and < upper qartile: normal_medium
- If in component 1 and >= upper qartile: normal_high
- If in component 1 and < lower quartile: abnormal_low
- If in component 1 and >= lower quartile and < upper qartile: abnormal_medium
- If in component 1 and >= upper qartile: abnormal_high

```
bmi_classes = c("underweight", "normal", "overweight", "obese")

age_classes = c("child", "teenager", "young_adult", "middle_aged", "older_adult", "elderly")

gl_classes = c("normal_low", "normal_medium", "normal_high", "abnormal_low", "abnormal_medium", "abnorma

data_as_factors = data %>%
  mutate(bmi_classification = case_when(
    bmi < 18.5 ~ bmi_classes[1],
    bmi >= 18.5 & bmi < 25 ~ bmi_classes[2],
    bmi >= 25 & bmi < 30 ~ bmi_classes[3],
    bmi >= 30 ~ bmi_classes[4]
  )) %>%
  mutate(bmi_classification = factor(bmi_classification, levels = bmi_classes)) %>%
  mutate(age_classification = case_when(
    age < 12 ~ age_classes[1],
    age >= 12 & age < 19 ~ age_classes[2],
    age >= 19 & age < 35 ~ age_classes[3],
    age >= 35 & age < 55 ~ age_classes[4],
    age >= 55 & age < 65 ~ age_classes[5],
    age >= 65 ~ age_classes[6]
  )) %>%
  mutate(age_classification = factor(age_classification, levels = age_classes)) %>%
  mutate(avg_glucose_class = gmm$classification) %>%
  mutate(gl_classification = case_when(
    avg_glucose_class == 1 &
      avg_glucose_level < glucose_quartiles[[1]]$`25%` ~ gl_classes[1],
    avg_glucose_class == 1 &
      avg_glucose_level >= glucose_quartiles[[1]]$`25%` &
      avg_glucose_level < glucose_quartiles[[1]]$`75%` ~ gl_classes[2],
    avg_glucose_class == 1 &
      avg_glucose_level >= glucose_quartiles[[1]]$`75%` ~ gl_classes[3],
    avg_glucose_class == 2 &
      avg_glucose_level < glucose_quartiles[[2]]$`25%` ~ gl_classes[4],
    avg_glucose_class == 2 &
      avg_glucose_level >= glucose_quartiles[[2]]$`25%` &
      avg_glucose_level < glucose_quartiles[[2]]$`75%` ~ gl_classes[5],
    avg_glucose_class == 2 &
      avg_glucose_level >= glucose_quartiles[[2]]$`75%` ~ gl_classes[6]
```

```
  )) %>%
  mutate(gl_classification = factor(gl_classification, levels = gl_classes))
```

Now that all the data is discretised I used the Goodman Kruskal Tau to explore two-way correlations between features (not included here for brevity). The tau values revealed a weak correlation between levels of smoking and age, so let's explore that relationship with a table.

```
table(data$smoking_status, data_as_factors$age_classification, useNA = "ifany") %>%
  kable %>%
  kable_styling()
```

	child	teenager	young_adult	middle_aged	older_adult	elderly
formerly smoked	33	198	1060	2181	1512	2509
never smoked	157	1221	3820	5338	2487	3030
smokes	2	117	1720	2620	1090	1013
NA	4736	1575	1766	2472	1136	1607

Since a large proportion of these NAs come from children, it seems reasonable to replace these NAs with 'never smoked'. The other NAs we will replace with a new level 'unknown'.

```
data_as_factors = data_as_factors %>%
  select(everything()) %>%
  mutate(smoking_status = case_when(
    is.na(smoking_status) & age_classification == "child" ~ "never smoked",
    is.na(smoking_status) ~ "unknown",
    TRUE ~ smoking_status))
```

After a brief look, no tau correlations provide a reasonable explanation for the missing BMI data so let's just remove it from the dataset.

```
data_as_factors = data_as_factors %>%
  filter(!is.na(bmi_classification))
```

Predictive Modelling

We now build a classification model of stroke incidence. When choosing an algorithm for this task I considered several options such as logistic regression, gradient boosting or decision trees but opted for SBRL for a number of reasons. Firstly, past experience has taught me that when dealing with health data, model interpretability is of utmost importance for clinicians and medical professionals. Secondly, I chose SBRL over more traditional tree algorithms like CART or C5.0 due to the fact the greedy construction method of these algorithms heavily affects the predictive quality of such algorithms. SBRL strikes a good balance between interpretability and quality.

Positive data imbalance.

In the dataset there is a severe imbalance between positive and negative stroke cases.

```
prop.table(table(data_as_factors$stroke))
```

```
##
##          0          1
## 0.98466784 0.01533216
```

This is in part related to the fact that the dataset includes a large number of young subjects.

This imbalance would severely affect the quality of most predictive models, so we use the SMOTE oversampling method to generate synthetic data. Note that we only use SMOTE on the training dataset, not the test dataset.

Modelling

```
# preprocessing
data_as_factors = data_as_factors %<>%
  mutate(id = as.factor(id)) %>%
  select_if(funs(is.character(.) || is.factor(.))) %>%
  mutate_all(as.factor)

set.seed(1)

#Create training set
train = data_as_factors %>% sample_frac(.70)
#Create test set
test = anti_join(data_as_factors, train, by = 'id')

# SMOTE

train = SMOTE(stroke ~., as.data.frame(train), perc.over=600, perc.under=100) %>%
  select(-id) %>%
  rename(label = stroke)

sbrl_model = sbrl(train, neg_sign = "0", pos_sign = "1", rule_minlen = 1, rule_maxlen = 2,
                         minsupport_pos=0.10, minsupport_neg=0.10,
                         lambda=10.0, eta=1.0, nchain=25)

test_pred = test %>%
  mutate(pred_prob = predict(sbrl_model, test)$V2)
```

Model evaluation

When evaluating our model it's important to consider if a particular metric is relevant to our problem. As previously noted, our dataset contains very few positive stroke victims (approximately 2%). As such, the null model (i.e. predict no stroke) would have a very high accuracy! Hence, we use the area under the ROC curve to evaluate the probability that our model would rank a randomly selected positive subject over a randomly selected negative subject.

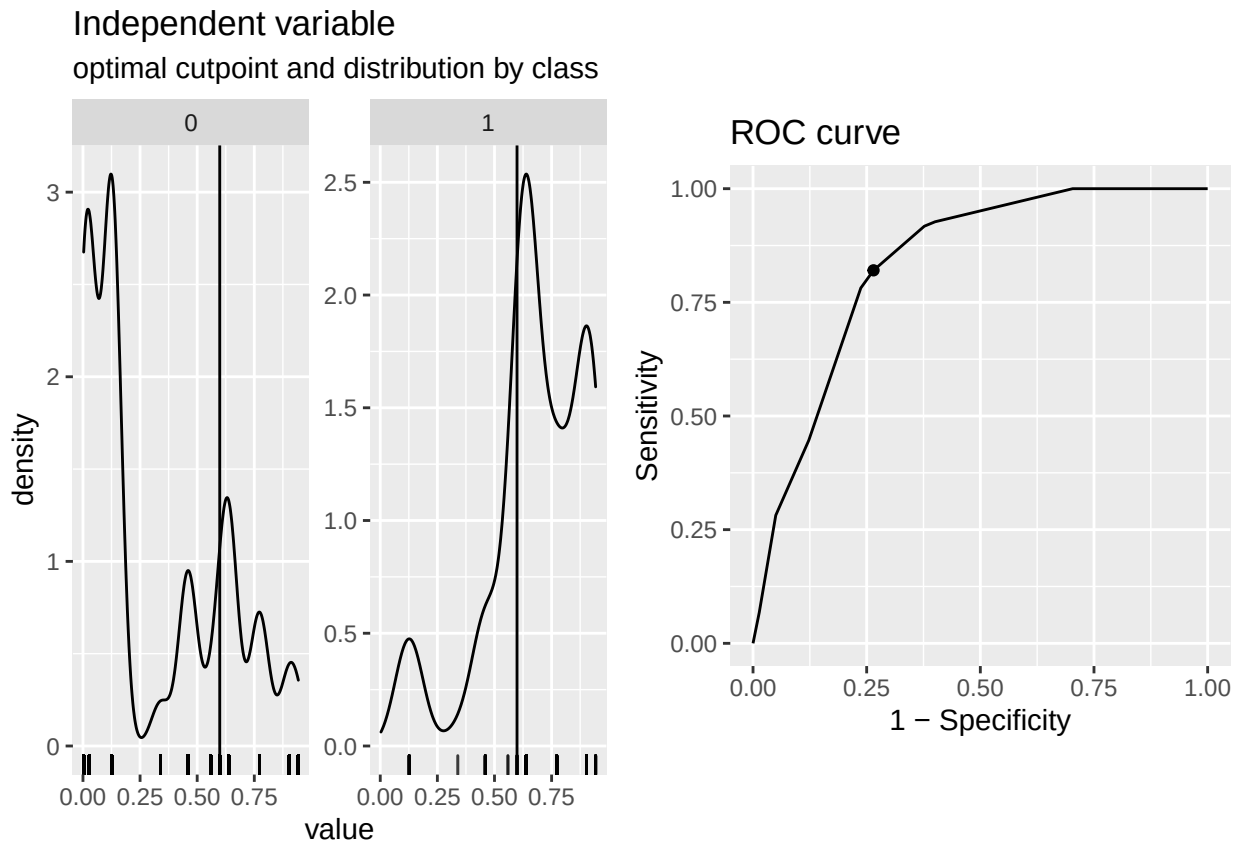
```
cp = cutpointr(test_pred, pred_prob, stroke,
                method = maximize_metric, metric = sum_sens_spec)

summary(cp)
```

```
## Method: maximize_metric
## Predictor: pred_prob
## Outcome: stroke
## Direction: >=
##
##     AUC     n n_pos n_neg
##  0.8333 12581   206 12375
##
##  optimal_cutpoint sum_sens_spec    acc sensitivity specificity  tp fn   fp   tn
```

```
##            0.5987        1.5553 0.7363      0.8204      0.7349 169 37 3280 9095
##
## Predictor summary:
##           Min.    5% 1st Qu. Median  Mean 3rd Qu.   95%  Max.    SD NAs
## overall 0.003 0.003   0.027  0.127 0.303   0.599 0.903 0.943 0.298   0
## 0       0.003 0.003   0.027  0.127 0.296   0.599 0.812 0.943 0.295   0
## 1       0.127 0.127   0.639  0.639 0.680   0.903 0.943 0.943 0.213   0
```

```
plot(cp)
```



We see that our model has a reasonable AUC of 0.83. The cutpointr method also provides us with the 'optimal' cut-off point of our prediction probability based on the sum of the model sensitivity and specificity.

What Features are Predictive of Stroke?

Let's take a look at the rule list our SBRL model produces.

```
print(sbrl_model)
```

```
## The rules list is :
## If      {ever_married=No,age_classification=elderly} (rule[24]) then positive probability = 0.942796
## else if {heart_disease=1} (rule[139]) then positive probability = 0.90293225
## else if {hypertension=1} (rule[180]) then positive probability = 0.77252252
## else if {age_classification=elderly} (rule[6]) then positive probability = 0.63872491
## else if {ever_married=Yes,gl_classification=abnormal_medium} (rule[40]) then positive probability = 
## else if {gl_classification=abnormal_high} (rule[99]) then positive probability = 0.55882353
## else if {age_classification=older_adult} (rule[9]) then positive probability = 0.45971564
## else if {age_classification=child} (rule[1]) then positive probability = 0.00306748
```

```
## else if {work_type=Self-employed,Residence_type=Urban} (rule[252]) then positive probability = 0.339
## else if {age_classification=young_adult} (rule[10]) then positive probability = 0.02692308
## else  (default rule)  then positive probability = 0.12671233
```

The most prominant features are age, glucose level, heart disease and hypertension. It's interesting to note that marital status makes two appearances in the rules and features in the first rule in the list.

Using Causal Discovery to Identifying Causes of Stroke

We now use a causal bayesian model to explore the causes of strokes. Since there may be latent variables not included in the dataset we are constrained to using causal discovery algorithms like FCI that produce DAG-like data structures called Partial Ancestral Graphs (PAGs). PAGs encode a weaker notion of causality than DAGs. AS with DAGs, the absence of an edge in a PAG denotes conditional independence between features. Edges can be of the form o-o, o–, o->, –>, <-> and —. Loosely speaking , bidirectional edges come from hidden variables (i.e. an unobserved confounded could have a causal relationship between both features).

We start by encoding our prior knowledge that the gender and age of the subject cannot be causally effected by any other features

```
forbid = list(
  c("hypertension","gender"),
  c("heart_disease","gender"),
  c("ever_married","gender"),
  c("work_type","gender"),
  c("Residence_type","gender"),
  c("smoking_status","gender"),
  c("label","gender"),
  c("age_classification","gender"),
  c("gl_classification","gender"),
  c("hypertension","age_classification"),
  c("heart_disease","age_classification"),
  c("ever_married","age_classification"),
  c("work_type","age_classification"),
  c("Residence_type","age_classification"),
  c("smoking_status","age_classification"),
  c("label","age_classification"),
  c("age_classification","age_classification"),
  c("gl_classification","age_classification"))

prior = priorKnowledge(forbiddirect = forbid)
```

We then use the FCI algorithm to create a PAG for the dataset. Unfortunately a bug in the FCI algorithm meant I have to remove BMI.

```
pag = tetradrunner(algoId = "fci", df = as.data.frame(train %>% select(-bmi_classification)),
                   scoreId = "cg-bic-score", dataType = "discrete",
                   alpha=0.1,faithfulnessAssumed=TRUE,maxDegree=-1,
                   verbose=F, priorKnowledge = prior)
```

Let's examine the edges which include strokes (note that label is the encoding from stroke)

```
pag$edges[str_detect(pag$edges, "label")]
```

```
## [1] "heart_disease o-> label"      "gl_classification o-> label"
## [3] "hypertension o-> label"       "work_type o-> label"
## [5] "gender o-> label"             "ever_married o-> label"
## [7] "age_classification o-> label" "smoking_status o-> label"
```

It's interesting to note that according to this model every variable could be actually be acting causally on stroke incidence through a confounder.

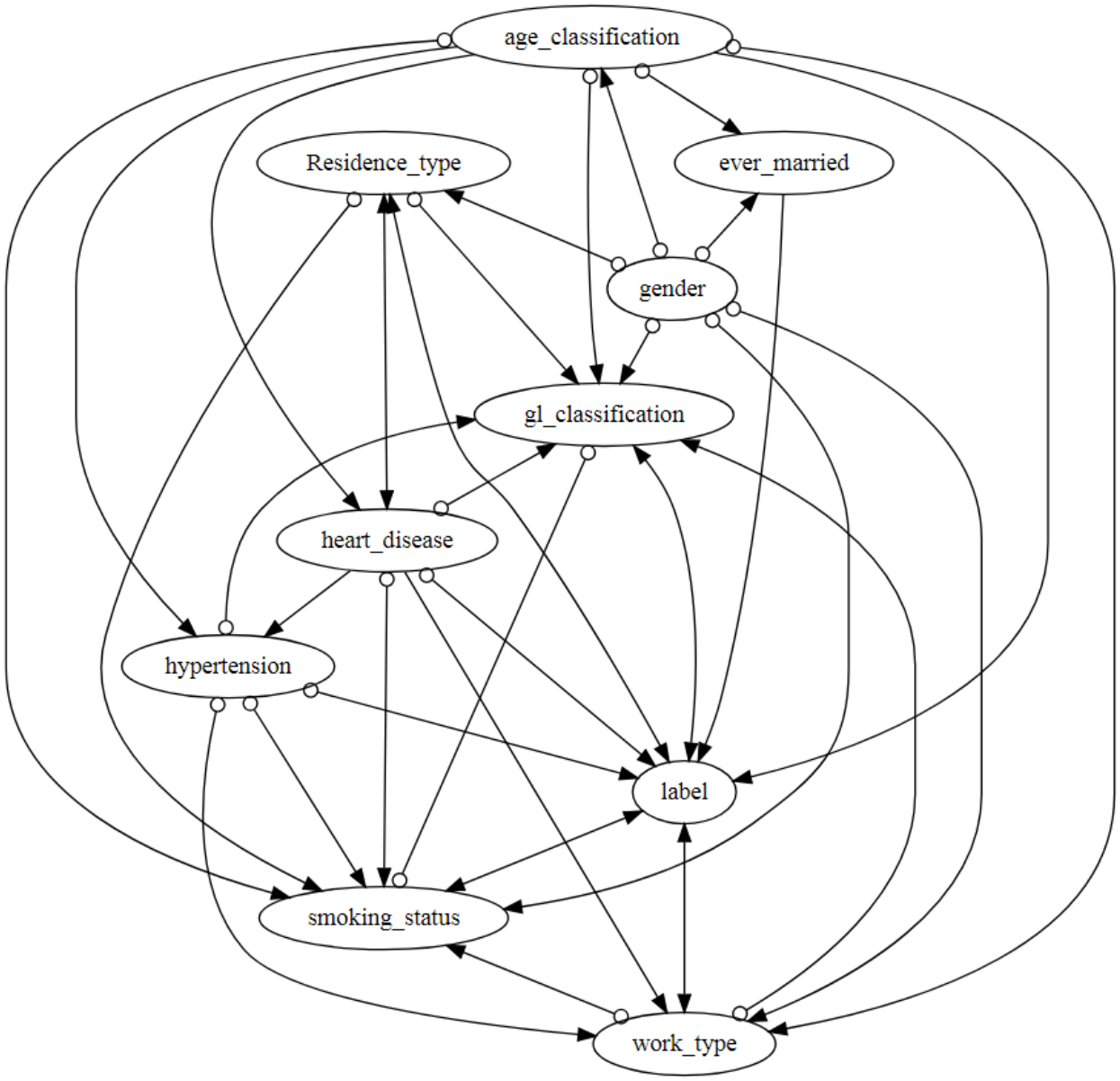


Figure 1: Visualisation of the PAG