Code ▾

Develop a Computational Phenotyping Algorithm to Identify Patients with Type II diabetes

Background on Diabetes

Type II diabetes is a type of diabetes that is caused by the body no longer recognizing and appropriately responding to insulin.

Diagnostic Criteria

- A fasting plasma glucose level of 126 mg/dL (7.0 mmol/L) or higher **OR**
- A 2-hour plasma glucose level of 200 mg/dL (11.1 mmol/L) or higher during a 75g OGTT **OR**
- A random plasma glucose of 200 mg/dL (11.1 mmol/L) or higher + symptoms **OR**
- HbA1c of 6.5% or higher

Treatments

- Metformin
- Sulfonylureas (e.g., glyburide, glimepiride…)
- Thiazolidinediones (e.g., pioglitazone…)
- DPP-4 inhibitors (e.g., sitagliptin…)
- SGLT2 inhibitors (e.g., canagliflozin…)
- GLP-1 receptor agonists (e.g., liraglutide…)

Laboratory Tests

- Plasma glucose (fasting / random / 2-hour OGTT)
- HbA1c

Set up connection to the Google BigQuery project

Load diababetes goldstandard

Hide

```
diabetes <- bq_project_query(
  my_project,
  "
  SELECT SUBJECT_ID, DIABETES
  FROM `course3_data.diabetes_goldstandard`
  "
) %>% bq_table_download()

knitr::kable(diabetes)
```

SUBJECT_ID	DIABETES
10011	0
10013	0
10026	0
10036	0
10038	0
10040	0
10044	0
10045	0
10046	0
10056	0
10065	0
10083	0
10098	0
10112	0
10117	0
10126	0
10127	0
40124	0
40277	0
40286	0
40601	0
40456	0
40595	0
40612	0
40687	0
41983	0
42231	0
42281	0

SUBJECT_ID	DIABETES
42321	0
43746	0
43827	0
43879	0
43909	0
44228	0
44212	0
10019	0
10029	0
10032	0
10035	0
10042	0
10043	0
10059	0
10064	0
10067	0
10074	0
10076	0
10088	0
10089	0
10090	0
10093	0
10101	0
10102	0
10119	0
10120	0
40310	0
40304	0
42075	0
42066	0
42135	0
42275	0
43748	0
42412	0
42458	0

SUBJECT_ID	DIABETES
43881	0
44154	0
10006	1
10017	1
10027	1
10033	1
10061	1
10069	1
10104	1
10111	1
10114	1
10124	1
10132	1
40503	1
40655	1
42033	1
42199	1
42302	1
42346	1
42367	1
43870	1
43927	1
10094	1
10106	1
10130	1
40177	1
40204	1
41795	1
41914	1
41976	1
42292	1
42430	1
43779	1
43798	1
44083	1

SUBJECT_ID	DIABETES
44222	1

In this table the DIABETES column is a 1 if the patient has a record of type II diabetes and a 0 if they did not have the condition.Of the 100 patients in the demo data set, 99 had notes that could be reviewed. Of those 99 records reviewed, 34 had type II diabetes.

Querying and Assessing ICD codes

There are many ICD-9 codes for diabetes:

ICD-9 Code	Label
250	Diabetes mellitus
250.0	Diabetes mellitus without mention of complication
250.00	Diabetes mellitus without mention of complication, type II or unspecified type, not stated as uncontrolled
250.01	Diabetes mellitus without mention of complication, type I (juvenile type), not stated as uncontrolled
250.02	Diabetes mellitus without mention of complication, type II or unspecified type, uncontrolled

Hide

```
#Load the tables
training <- tbl(con, "course3_data.diabetes_training")
diagnoses_icd <- tbl(con, "mimic3_demo.DIAGNOSES_ICD")

#Identify the patients with ICD_25000
icd_25000 <- diagnoses_icd %>%
  filter(ICD9_CODE == "25000") %>%
  distinct(SUBJECT_ID) %>%
  mutate(icd_25000 = 1)
knitr::kable(head(icd_25000,10))
```

SUBJECT_ID	icd_25000
10106	1
43779	1
40204	1
10006	1
43798	1
10017	1
10027	1
10033	1
40503	1

SUBJECT_ID	icd_25000
10045	1

Hide

```
# Join icd_25000 with the diabetes data frame
training_joined <- training %>%
  left_join(icd_25000, by = "SUBJECT_ID") %>%
  mutate(icd_25000 = coalesce(icd_25000, 0))

knitr::kable(head(training_joined,20))
```

SUBJECT_ID	DIABETES	icd_25000
10026	0	0
40310	0	0
10067	0	0
44228	0	0
10064	0	0
10126	0	0
10102	0	0
10045	0	1
42231	0	0
10065	0	0
40612	0	0
10043	0	0
40124	0	0
10032	0	0
42321	0	0
10076	0	0
10093	0	0
42275	0	0
40601	0	0
10040	0	0

Hide

```
# Function to Calculate Performance
library(caret)
getStats <- function(df, ...) {
  df %>%
    select(...) %>%
    mutate(across(everything(), ~ factor(.x, levels = c(1, 0)))) %>%
    table() %>%
    confusionMatrix()
}
```

Hide

```
# Calculate the performance of icd_25000
training %<>%
  left_join(icd_25000) %>%
  mutate(icd_25000 = coalesce(icd_25000, 0))
training %>%
  collect() %>%
  getStats(icd_25000, DIABETES)
```

```
## Confusion Matrix and Statistics
##
##          DIABETES
## icd_25000  1  0
##         1 19  1
##         0  8 52
##
##                Accuracy : 0.8875
##                  95% CI : (0.7972, 0.9472)
##     No Information Rate : 0.6625
##     P-Value [Acc > NIR] : 3.476e-06
##
##                   Kappa : 0.7313
##
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.7037
##             Specificity : 0.9811
##          Pos Pred Value : 0.9500
##          Neg Pred Value : 0.8667
##              Prevalence : 0.3375
##          Detection Rate : 0.2375
##    Detection Prevalence : 0.2500
##       Balanced Accuracy : 0.8424
##
##        'Positive' Class : 1
##
```

This code actually performs fairly well. ICD9 250.00 has a decent specificity of 98.11%. However the sensitivity is not great at only 70.37%.

Querying and Assessing laboratory data

As described in the introduction, there are a number of laboratory tests used to diagnose diabetes.We will take a look at just Hemoglobin A1C. MIMIC-III records lab tests with a variety of labels. We can search these labels in the D_LABITEMS table. The following SQL query was executed in BigQuery:

select * from mimic3_demo.D_LABITEMS where lower(LABEL) like “%a1c%”

The results were:

ITEMID	LABEL
50852	% Hemoglobin A1c
50854	Absolute A1c

Hide

```
# Load labevents table
labevents <- tbl(con, "mimic3_demo.LABEVENTS")

# Identify patients with hba1c
hba1c <- labevents %>%
  filter(ITEMID %in% c(50852,50854)) %>%
  distinct(SUBJECT_ID) %>%
  mutate(hba1c = 1)

# Merge hba1c indicator into training dataset
training %<>%
  left_join(hba1c) %>%
  mutate(hba1c = coalesce(hba1c, 0))

# Evaluate performance
training %>%
  collect() %>%
  getStats(hba1c, DIABETES)
```

```
## Confusion Matrix and Statistics
## 
##      DIABETES
## hba1c  1  0
##     1  7  3
##     0 20 50
## 
##                Accuracy : 0.7125
##                  95% CI : (0.6005, 0.8082)
##     No Information Rate : 0.6625
##     P-Value [Acc > NIR] : 0.2051844
## 
##                   Kappa : 0.2397
## 
##  Mcnemar's Test P-Value : 0.0008492
## 
##             Sensitivity : 0.2593
##             Specificity : 0.9434
##          Pos Pred Value : 0.7000
##          Neg Pred Value : 0.7143
##              Prevalence : 0.3375
##          Detection Rate : 0.0875
##    Detection Prevalence : 0.1250
##       Balanced Accuracy : 0.6013
## 
##        'Positive' Class : 1
## 
```

The combined HbA1c labs have a moderate specificity of 94.34%. However the sensitivity is very poor at only 25.93%.

Querying and Assessing Medication data

As described in the introduction, there are a number of medications used to treat diabetes. Let's try the first-line treatment metformin.

Hide

```
# Load PRESCRIPTIONS table
prescriptions <- tbl(con, "mimic3_demo.PRESCRIPTIONS")

# Identify patients with Metformin prescriptions
metformin <- prescriptions %>%
  filter(tolower(DRUG) %like% "%metformin%") %>%
  distinct(SUBJECT_ID) %>%
  mutate(metformin = 1)
knitr::kable(head(metformin,20))
```

SUBJECT_ID	metformin
10104	1
10106	1
43927	1

Hide

```
# Merge Metformin indicator into training dataset
training_metformin <- training %>%
    left_join(metformin, by = "SUBJECT_ID") %>%
    mutate(metformin = coalesce(metformin, 0))
knitr::kable(head(training_metformin,20))
```

SUBJECT_ID	DIABETES	icd_25000	hba1c	metformin
10026	0	0	0	0
40310	0	0	1	0
10067	0	0	0	0
44228	0	0	0	0
10064	0	0	0	0
10126	0	0	0	0
10102	0	0	0	0
10045	0	1	0	0
42231	0	0	0	0
10065	0	0	0	0
40612	0	0	0	0
10043	0	0	0	0
40124	0	0	0	0
10032	0	0	0	0
42321	0	0	0	0
10076	0	0	0	0
10093	0	0	0	0
42275	0	0	0	0
40601	0	0	1	0
10040	0	0	0	0

Hide

```
# Evaluate performance
training_metformin %>%
    collect() %>%
    getStats(metformin, DIABETES)
```

```
## Confusion Matrix and Statistics
## 
##          DIABETES
## metformin  1  0
##         1  2  0
##         0 25 53
## 
##                Accuracy : 0.6875
##                  95% CI : (0.5741, 0.7865)
##     No Information Rate : 0.6625
##     P-Value [Acc > NIR] : 0.3658
## 
##                   Kappa : 0.0958
## 
##  Mcnemar's Test P-Value : 1.587e-06
## 
##             Sensitivity : 0.07407
##             Specificity : 1.00000
##          Pos Pred Value : 1.00000
##          Neg Pred Value : 0.67949
##              Prevalence : 0.33750
##          Detection Rate : 0.02500
##    Detection Prevalence : 0.02500
##       Balanced Accuracy : 0.53704
## 
##        'Positive' Class : 1
## 
```

Metformin has a perfect specificity of 100%. However the sensitivity is exceptionally poor at only 7.41%. This is likely due to the fact that most hospitalized patients are transitioned to insulin during their hospital stay.

Querying and Assessing the mean value of blood glucose blood gas of at least 200 mg/dL

Hide

```
# 1) Create a binary feature: mean blood gas glucose >= 200
labevents <- tbl(con, "mimic3_demo.LABEVENTS")
d_labitems <- tbl(con, "mimic3_demo.D_LABITEMS")
mean_glucose_blood_bg_over200 <- labevents %>%
  inner_join(d_labitems, by = "ITEMID", suffix = c("_l","_d")) %>%
  filter(
    LABEL == "Glucose",
    FLUID == "Blood",
    CATEGORY == "Blood Gas"
  ) %>%
  group_by(SUBJECT_ID) %>%
  summarise(glucose_blood_bg_mean = mean(VALUENUM, na.rm = TRUE), .groups = "drop") %>%
  mutate(mean_glucose_blood_bg_over200 = if_else(glucose_blood_bg_mean >= 200, 1L, 0L)) %>%
  select(SUBJECT_ID, glucose_blood_bg_mean, mean_glucose_blood_bg_over200)
knitr::kable(head(mean_glucose_blood_bg_over200,20))
```

SUBJECT_ID	glucose_blood_bg_mean	mean_glucose_blood_bg_over200
40204	91.5000	0
10126	122.9600	0

SUBJECT_ID	glucose_blood_bg_mean	mean_glucose_blood_bg_over200
10027	126.6970	0
10093	61.0000	0
41976	206.0000	1
10059	129.5000	0
10006	77.0000	0
10019	163.3333	0
42135	106.6000	0
40595	161.3333	0
10045	136.0625	0
10042	143.1500	0
42075	113.0000	0
10120	289.6000	1
10111	113.8000	0
42292	92.0000	0
10061	144.5000	0
10065	135.2857	0
10127	116.0000	0
43927	130.1429	0

Hide

```
# 2) Join into training + fill missing with 0
training_glucose <- training %>%
  left_join(mean_glucose_blood_bg_over200, by = "SUBJECT_ID") %>%
  mutate(mean_glucose_blood_bg_over200 = coalesce(mean_glucose_blood_bg_over200, 0L))
knitr::kable(head(training_glucose,20))
```

SUBJECT_ID	DIABETES	icd_25000	hba1c	glucose_blood_bg_mean	mean_glucose_blood_bg_over200
10026	0	0	0	NA	0
40310	0	0	1	107.6667	0
10067	0	0	0	108.0000	0
44228	0	0	0	NA	0
10064	0	0	0	282.5000	1
10126	0	0	0	122.9600	0
10102	0	0	0	NA	0
10045	0	1	0	136.0625	0
42231	0	0	0	147.5000	0
10065	0	0	0	135.2857	0

SUBJECT_ID	DIABETES	icd_25000	hba1c	glucose_blood_bg_mean	mean_glucose_blood_bg_over200
40612	0	0	0	118.0000	0
10043	0	0	0	NA	0
40124	0	0	0	NA	0
10032	0	0	0	NA	0
42321	0	0	0	NA	0
10076	0	0	0	136.5000	0
10093	0	0	0	61.0000	0
42275	0	0	0	150.0000	0
40601	0	0	1	NA	0
10040	0	0	0	110.0000	0

Hide

```
# 3) Collect to local and run getStats
training_glucose_local <- training_glucose %>% collect()

cm <- training_glucose_local %>%
  getStats(mean_glucose_blood_bg_over200, DIABETES)

cm   # prints confusion matrix + stats
```

```
## Confusion Matrix and Statistics
## 
##                              DIABETES
## mean_glucose_blood_bg_over200  1  0
##                             1  2  2
##                             0 25 51
##                                           
##                Accuracy : 0.6625          
##                  95% CI : (0.5481, 0.7645)
##     No Information Rate : 0.6625          
##     P-Value [Acc > NIR] : 0.552           
##                                           
##                   Kappa : 0.0459          
##                                           
##  Mcnemar's Test P-Value : 2.297e-05       
##                                           
##             Sensitivity : 0.07407         
##             Specificity : 0.96226         
##          Pos Pred Value : 0.50000         
##          Neg Pred Value : 0.67105         
##              Prevalence : 0.33750         
##          Detection Rate : 0.02500         
##    Detection Prevalence : 0.05000         
##       Balanced Accuracy : 0.51817         
##                                           
##        'Positive' Class : 1               
## 
```

The mean value of blood glucose blood gas of at least 200 mg/dL has a good specificity of 96%. However the sensitivity is exceptionally poor at only 7.41%.

Querying and Assessing the comibination of metformin and insulin

Hide

```
#Load the prescriptions table
prescriptions <- tbl(con, "mimic3_demo.PRESCRIPTIONS")

#Identify the patients with metfomin and insulin prescriptions
metformin_and_insulin <- prescriptions %>%
  filter(lower(DRUG) %like% "metformin" |
           lower(DRUG) %like% "insulin") %>%
  mutate(metformin_counter = case_when(lower(DRUG) %like% "%metformin%" ~ 1,
                                       TRUE ~ 0),
         insulin_counter = case_when(lower(DRUG) %like% "%insulin%" ~ 1,
                                     TRUE ~ 0)) %>%
  group_by(SUBJECT_ID) %>%
  summarise(any_metformin = max(metformin_counter, na.rm = TRUE),
            any_insulin = max(insulin_counter, na.rm = TRUE)) %>%
  filter(any_metformin == 1,
         any_insulin == 1) %>%
  mutate(metformin_and_insulin = 1)
# Join with the training data
training %>%
  left_join(metformin_and_insulin) %>%
  mutate(metformin_and_insulin = coalesce(metformin_and_insulin, 0)) %>%
  collect() %>%
# Evaluate performance
  getStats(metformin_and_insulin, DIABETES)
```

```
## Confusion Matrix and Statistics
## 
##                      DIABETES
## metformin_and_insulin  1  0
##                     1  2  0
##                     0 25 53
## 
##                Accuracy : 0.6875
##                  95% CI : (0.5741, 0.7865)
##     No Information Rate : 0.6625
##     P-Value [Acc > NIR] : 0.3658
## 
##                   Kappa : 0.0958
## 
##  Mcnemar's Test P-Value : 1.587e-06
## 
##             Sensitivity : 0.07407
##             Specificity : 1.00000
##          Pos Pred Value : 1.00000
##          Neg Pred Value : 0.67949
##              Prevalence : 0.33750
##          Detection Rate : 0.02500
##    Detection Prevalence : 0.02500
##       Balanced Accuracy : 0.53704
## 
##        'Positive' Class : 1
## 
```

The comibination of metformin and insulin has a perfect specificity of 100%. However the sensitivity is exceptionally poor at only 7.41%.

Querying and Assessing the comibination of ICD9 OR Glucose>=200 OR Insulin

Hide

```
#Load tables
labevents <- tbl(con, "mimic3_demo.LABEVENTS")
d_labitems <- tbl(con, "mimic3_demo.D_LABITEMS")
#Identify the patients with ICD
any_t2d_icd <- diagnoses_icd %>%
  filter(ICD9_CODE %in% c("25000", "25002","25010", "25012","25020", "25022","25030",
                          "25032","25040", "25042","25050", "25052","25060", "25062",
                          "25070", "25072","25080", "25082","25090", "25092")) %>%
  distinct(SUBJECT_ID) %>%
  mutate(any_t2d_icd = 1)
#Identify the patients with Glucose>=200
any_glucose_blood_bg_over200 <- labevents %>%
  inner_join(d_labitems, by = c("ITEMID" = "ITEMID"), suffix = c("_l","_d")) %>%
  filter(LABEL == "Glucose",
         FLUID == "Blood",
         CATEGORY == "Blood Gas") %>%
  group_by(SUBJECT_ID) %>%
  mutate(glucose_blood_bg_over200_marker = case_when(VALUENUM >= 200 ~ 1,
                                                     TRUE ~0)) %>%
  summarise(any_glucose_blood_bg_over200 = max(glucose_blood_bg_over200_marker, na.rm = TRUE)) %>%
  select(SUBJECT_ID, any_glucose_blood_bg_over200)
#Identify the patients with Insulin
any_insulin <- prescriptions %>%
  filter(lower(DRUG) %like% "insulin") %>%
  distinct(SUBJECT_ID) %>%
  mutate(any_insulin = 1)
#Join with the training data
training %>%
  left_join(any_t2d_icd) %>%
  left_join(any_glucose_blood_bg_over200) %>%
  left_join(any_insulin) %>%
  mutate(any_t2d_icd = coalesce(any_t2d_icd, 0),
         any_glucose_blood_bg_over200 = coalesce(any_glucose_blood_bg_over200, 0),
         any_insulin = coalesce(any_insulin, 0)) %>%
  mutate(icd_or_glucose_or_insulin = case_when(any_t2d_icd == 1 |
                                                 any_glucose_blood_bg_over200 == 1 |
                                                 any_insulin == 1 ~ 1,
                                               TRUE ~ 0)) %>%
  collect() %>%
# Evaluate performance
  getStats(icd_or_glucose_or_insulin, DIABETES)
```

```
## Confusion Matrix and Statistics
## 
##                           DIABETES
## icd_or_glucose_or_insulin  1  0
##                          1 26 34
##                          0  1 19
##                                           
##                 Accuracy : 0.5625         
##                   95% CI : (0.447, 0.6732)
##      No Information Rate : 0.6625         
##      P-Value [Acc > NIR] : 0.9761         
##                                           
##                    Kappa : 0.2473         
##                                           
##   Mcnemar's Test P-Value : 6.338e-08      
##                                           
##              Sensitivity : 0.9630         
##              Specificity : 0.3585         
##           Pos Pred Value : 0.4333         
##           Neg Pred Value : 0.9500         
##               Prevalence : 0.3375         
##           Detection Rate : 0.3250         
##     Detection Prevalence : 0.7500         
##        Balanced Accuracy : 0.6607         
##                                           
##         'Positive' Class : 1              
## 
```

The comibination of ICD9 OR Glucose>=200 OR Insulin has a poor specificity of 35.9%. However the sensitivity is exceptionally high at only 96.3%.

Querying and Assessing the comibination of ICD9 AND Glucose>=200 AND Insulin

Hide

```
#Join with the traing data
training %>%
  left_join(any_t2d_icd) %>%
  left_join(any_glucose_blood_bg_over200) %>%
  left_join(any_insulin) %>%
  mutate(any_t2d_icd = coalesce(any_t2d_icd, 0),
         any_glucose_blood_bg_over200 = coalesce(any_glucose_blood_bg_over200, 0),
         any_insulin = coalesce(any_insulin, 0)) %>%
  mutate(icd_and_glucose_and_insulin = case_when(any_t2d_icd == 1 &&
                                                 any_glucose_blood_bg_over200 == 1 &&
                                                 any_insulin == 1 ~ 1,
                                               TRUE ~ 0)) %>%
  collect() %>%
# Evaluate performance
  getStats(icd_and_glucose_and_insulin, DIABETES)
```

```
## Confusion Matrix and Statistics
## 
##                           DIABETES
## icd_and_glucose_and_insulin  1  0
##                           1  3  0
##                           0 24 53
##                                           
##                Accuracy : 0.7             
##                  95% CI : (0.5872, 0.7974)
##     No Information Rate : 0.6625          
##     P-Value [Acc > NIR] : 0.2802          
##                                           
##                   Kappa : 0.1421          
##                                           
##  Mcnemar's Test P-Value : 2.668e-06       
##                                           
##             Sensitivity : 0.1111          
##             Specificity : 1.0000          
##          Pos Pred Value : 1.0000          
##          Neg Pred Value : 0.6883          
##              Prevalence : 0.3375          
##          Detection Rate : 0.0375          
##    Detection Prevalence : 0.0375          
##       Balanced Accuracy : 0.5556          
##                                           
##        'Positive' Class : 1               
## 
```

The comibination of ICD9 AND Glucose>=200 AND Insulin has a perfect specificity of 100%. However the sensitivity is exceptionally poor at only 11%.

Querying and Assessing the comibination of Insulin AND (ICD OR Glucose >=200)

Hide

```
#Join with the traing data
training %>%
  left_join(any_t2d_icd) %>%
  left_join(any_glucose_blood_bg_over200) %>%
  left_join(any_insulin) %>%
  mutate(any_t2d_icd = coalesce(any_t2d_icd, 0),
         any_glucose_blood_bg_over200 = coalesce(any_glucose_blood_bg_over200, 0),
         any_insulin = coalesce(any_insulin, 0)) %>%
  mutate(insulin_and_ICDorGlucose = case_when(any_insulin == 1 &&
                                                (any_glucose_blood_bg_over200 ==1 | any_t2d_icd == 1) ~
1,
                                              TRUE ~0)) %>%
  collect() %>%
# Evaluate performance
  getStats(insulin_and_ICDorGlucose, DIABETES)
```

```
## Confusion Matrix and Statistics
##
##                         DIABETES
## insulin_and_ICDorGlucose  1  0
##                        1 18  7
##                        0  9 46
##
##                Accuracy : 0.8
##                  95% CI : (0.6956, 0.8811)
##     No Information Rate : 0.6625
##     P-Value [Acc > NIR] : 0.005046
##
##                   Kappa : 0.5445
##
##  Mcnemar’s Test P-Value : 0.802587
##
##             Sensitivity : 0.6667
##             Specificity : 0.8679
##          Pos Pred Value : 0.7200
##          Neg Pred Value : 0.8364
##              Prevalence : 0.3375
##          Detection Rate : 0.2250
##    Detection Prevalence : 0.3125
##       Balanced Accuracy : 0.7673
##
##        ’Positive’ Class : 1
##
```

Using insulin alone had a specificity of 37.74%, and a sensitivity of 85.19%.By requiring that patients with insuline must also have a record of an ICD9 code or high glucose measurement we raised the specificity to 86.79%, and only dropped the sensitivity to 66.67%.

Rendered Report

The rendered HTML report is available here: - Download diabetes2.html (diabetes2.html)

Please download and open in a browser to view the full report.