Feature_Request

Aviral Vijay

July 30, 2018

```
library(healthcareai)
```

```
## healthcareai version 2.1.1
## Please visit https://docs.healthcare.ai for full documentation and vignettes. Join the com
munity at https://healthcare-ai.slack.com
```

```
library(nnet)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
str(pima_diabetes)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    768 obs. of  10 variables:
##  $ patient_id    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ pregnancies   : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ plasma_glucose: int  148 85 183 89 137 116 78 115 197 125 ...
##  $ diastolic_bp  : int  72 66 64 66 40 74 50 NA 70 96 ...
##  $ skinfold      : int  35 29 NA 23 35 NA 32 NA 45 NA ...
##  $ insulin       : int  NA NA NA 94 168 NA 88 NA 543 NA ...
##  $ weight_class  : chr  "obese" "overweight" "normal" "overweight" ...
##  $ pedigree      : num  0.627 0.351 0.672 0.167 2.288 ...
##  $ age           : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ diabetes      : chr  "Y" "N" "Y" "N" ...
```

```
unique(pima_diabetes['weight_class'])
```

```
## # A tibble: 6 x 1
##   weight_class
##   <chr>
## 1 obese
## 2 overweight
## 3 normal
## 4 morbidly obese
## 5 <NA>
## 6 underweight
```

```
pima_diabetes
```

```
## # A tibble: 768 x 10
##    patient_id pregnancies plasma_glucose diastolic_bp skinfold insulin
##         <int>       <int>          <int>        <int>    <int>   <int>
##  1          1           6            148           72       35      NA
##  2          2           1             85           66       29      NA
##  3          3           8            183           64       NA      NA
##  4          4           1             89           66       23      94
##  5          5           0            137           40       35     168
##  6          6           5            116           74       NA      NA
##  7          7           3             78           50       32      88
##  8          8          10            115           NA       NA      NA
##  9          9           2            197           70       45     543
## 10         10           8            125           96       NA      NA
## # ... with 758 more rows, and 4 more variables: weight_class <chr>,
## #   pedigree <dbl>, age <int>, diabetes <chr>
```

** Problem Statment

1: Is there any relation between patient weight and having diabetes?

Solution first

Split pima_diabetes dataset into train and test

```
d_train <- pima_diabetes[1:700, ]
d_test <- pima_diabetes[701:768, ]
```

Prepare the data for model training where patient_id, pedigree are ignored variable i.e these will not be used in ### model training.

```
d_train_prepped <- prep_data(d = d_train, patient_id, pedigree, outcome = diabetes, impute = TRUE)
```

```
## Training new data prep recipe...
```

```
d_train_prepped
```

```
## healthcareai-prepped data. Recipe used to prepare data:
```

```
## Data Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          7
##
## Training data contained 700 data points and 340 incomplete rows.
##
## Operations:
##
## Sparse, unbalanced variable filter removed no terms [trained]
## Mean Imputation for pregnancies, plasma_glucose, ... [trained]
## Filling NA with missing for weight_class [trained]
## Adding levels to: other, missing [trained]
## Collapsing factor levels for weight_class [trained]
## Adding levels to: other, missing [trained]
## Dummy variables from weight_class [trained]
```

```
## Current data:
```

```
## # A tibble: 700 x 14
##    patient_id pedigree pregnancies plasma_glucose diastolic_bp skinfold
##         <int>    <dbl>       <int>          <dbl>        <dbl>    <dbl>
##  1          1    0.627           6            148         72       35
##  2          2    0.351           1             85         66       29
##  3          3    0.672           8            183         64       29.1
##  4          4    0.167           1             89         66       23
##  5          5    2.29            0            137         40       35
##  6          6    0.201           5            116         74       29.1
##  7          7    0.248           3             78         50       32
##  8          8    0.134          10            115         72.3     29.1
##  9          9    0.158           2            197         70       45
## 10         10    0.232           8            125         96       29.1
## # ... with 690 more rows, and 8 more variables: insulin <dbl>, age <int>,
## #   diabetes <fct>, weight_class_normal <dbl>, weight_class_obese <dbl>,
## #   weight_class_overweight <dbl>, weight_class_other <dbl>,
## #   weight_class_missing <dbl>
```

It can be seen above that ignored variables(patient_id, pedigree) are still present in the dataset therefore we are ### going to use the above receipe to prepare the training data again

```
d_train_prepped_by_recipe  <- prep_data(d = d_train, recipe = d_train_prepped)
```

```
## Prepping data based on provided recipe
```

```
d_train_prepped_by_recipe
```

```
## healthcareai-prepped data. Recipe used to prepare data:
```

```
## Data Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          7
##
## Training data contained 700 data points and 340 incomplete rows.
##
## Operations:
##
## Sparse, unbalanced variable filter removed no terms [trained]
## Mean Imputation for pregnancies, plasma_glucose, ... [trained]
## Filling NA with missing for weight_class [trained]
## Adding levels to: other, missing [trained]
## Collapsing factor levels for weight_class [trained]
## Adding levels to: other, missing [trained]
## Dummy variables from weight_class [trained]
```

```
## Current data:
```

```
## # A tibble: 700 x 12
##    pregnancies plasma_glucose diastolic_bp skinfold insulin   age diabetes
##          <int>          <dbl>        <dbl>    <dbl>   <dbl> <int> <fct>
##  1           6            148           72       35      154.    50 Y
##  2           1             85           66       29      154.    31 N
##  3           8            183           64       29.1    154.    32 Y
##  4           1             89           66       23       94     21 N
##  5           0            137           40       35      168     33 Y
##  6           5            116           74       29.1    154.    30 N
##  7           3             78           50       32       88     26 Y
##  8          10            115           72.3     29.1    154.    29 N
##  9           2            197           70       45      543     53 Y
## 10           8            125           96       29.1    154.    54 Y
## # ... with 690 more rows, and 5 more variables: weight_class_normal <dbl>,
## #   weight_class_obese <dbl>, weight_class_overweight <dbl>,
## #   weight_class_other <dbl>, weight_class_missing <dbl>
```

It can be seen above that ignored variables(patient_id, pedigree) are not present in the dataset therefore we can ### use the above preparred dataset for training purpose & will prepare the test data in the same way

```
d_test_prepped_by_recipe <- prep_data(d = d_test, recipe = d_train_prepped)
```

```
## Prepping data based on provided recipe
```

```
d_test_prepped_by_recipe
```

```
## healthcareai-prepped data. Recipe used to prepare data:
```

```
## Data Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          7
##
## Training data contained 700 data points and 340 incomplete rows.
##
## Operations:
##
## Sparse, unbalanced variable filter removed no terms [trained]
## Mean Imputation for pregnancies, plasma_glucose, ... [trained]
## Filling NA with missing for weight_class [trained]
## Adding levels to: other, missing [trained]
## Collapsing factor levels for weight_class [trained]
## Adding levels to: other, missing [trained]
## Dummy variables from weight_class [trained]
```

```
## Current data:
```

```
## # A tibble: 68 x 12
##    pregnancies plasma_glucose diastolic_bp skinfold insulin   age diabetes
##          <int>          <int>        <dbl>    <dbl>   <dbl> <int> <fct>
##  1           2            122         76       27      200     26 N
##  2           6            125         78       31      154.    49 Y
##  3           1            168         88       29      154.    52 Y
##  4           2            129         72.3     29.1    154.    41 N
##  5           4            110         76       20      100     27 N
##  6           6             80         80       36      154.    28 N
##  7          10            115         72.3     29.1    154.    30 Y
##  8           2            127         46       21      335     22 N
##  9           9            164         78       29.1    154.    45 Y
## 10           2             93         64       32      160     23 Y
## # ... with 58 more rows, and 5 more variables: weight_class_normal <dbl>,
## #   weight_class_obese <dbl>, weight_class_overweight <dbl>,
## #   weight_class_other <dbl>, weight_class_missing <dbl>
```

It can be seen above that ignored variables are not present in the dataset therefore we can use the above preparred ### dataset for test dataset.

Model training

```
m <- tune_models(d_train_prepped_by_recipe, outcome = diabetes)
```

```
##
## diabetes looks categorical, so training classification algorithms.
```

```
##
## After data processing, models are being trained on 11 features with 700 observations.
## Based on n_folds = 5 and hyperparameter settings, the following number of models will be t
rained: 50 rf's, 50 xgb's, and 100 glm's
```

```
## Training with cross validation: Random Forest
```

```
## Training with cross validation: eXtreme Gradient Boosting
```

```
## Training with cross validation: glmnet
```

```
##
## *** Models successfully trained. The model object contains the training data minus ignored
ID columns. ***
## *** If there was PHI in training data, normal PHI protocols apply to the model object. ***
```

```
summary(m)
```

```
## Models trained: 2018-07-30 13:14:05
##
## Models tuned via 5-fold cross validation over 10 combinations of hyperparameter values.
## Best performance: AUPR = 0.71, AUROC = 0.84
## By glmnet with hyperparameters:
##   alpha = 1
##   lambda = 0.0087
##
## Out-of-fold performance of all trained models:
##
## $`Random Forest`
## # A tibble: 10 x 9
##     mtry splitrule  min.node.size AUROC  Sens   Spec  ROCSD  SensSD SpecSD
##  * <int> <chr>              <int> <dbl> <dbl>  <dbl>  <dbl>   <dbl>  <dbl>
##  1     5 extratrees            16 0.833 0.865 0.585  0.0336 0.0272  0.0643
##  2     9 extratrees            19 0.829 0.863 0.585  0.0357 0.0261  0.0609
##  3     2 gini                  11 0.825 0.874 0.569  0.0375 0.0225  0.0706
##  4     2 gini                   1 0.824 0.863 0.573  0.0385 0.0251  0.0814
##  5     4 gini                  16 0.818 0.854 0.581  0.0373 0.0250  0.0610
##  6     4 gini                  12 0.814 0.843 0.577  0.0368 0.0293  0.0658
##  7     1 extratrees             3 0.813 0.991 0.0580 0.0334 0.0142  0.0495
##  8     1 extratrees            19 0.812 0.993 0.0497 0.0300 0.00972 0.0312
##  9     1 extratrees            18 0.809 0.996 0.0372 0.0351 0.00972 0.0264
## 10     1 extratrees            15 0.809 0.996 0.0455 0.0299 0.00972 0.0338
##
## $`eXtreme Gradient Boosting`
## # A tibble: 10 x 13
##       eta max_depth gamma colsample_bytree min_child_weight subsample
##  *  <dbl>     <int> <dbl>            <dbl>            <dbl>     <dbl>
##  1 0.0991         3  4.71            0.501           0.0160     0.915
##  2 0.215         10  9.26            0.716           0.951      0.715
##  3 0.258          7  9.74            0.679           1.68       0.731
##  4 0.199         10  4.89            0.772           2.43       0.711
##  5 0.152          9  1.97            0.707           6.64       0.533
##  6 0.238          3  1.73            0.752           3.17       0.974
##  7 0.279         10  6.44            0.835           0.515      0.741
##  8 0.227          6  2.64            0.572           3.34       0.697
##  9 0.294          7  3.22            0.711           5.90       0.873
## 10 0.194          5  6.41            0.537          21.2        0.528
## # ... with 7 more variables: nrounds <int>, AUROC <dbl>, Sens <dbl>,
## #   Spec <dbl>, ROCSD <dbl>, SensSD <dbl>, SpecSD <dbl>
##
## $glmnet
## # A tibble: 20 x 8
##    alpha  lambda AUROC  Sens   Spec  ROCSD  SensSD SpecSD
##  * <dbl>   <dbl> <dbl> <dbl>  <dbl>  <dbl>   <dbl>  <dbl>
##  1     1 0.00868 0.837 0.872 0.536  0.0287 0.0666  0.105
##  2     1 0.00487 0.836 0.867 0.540  0.0262 0.0684  0.0983
##  3     1 0.0253  0.834 0.891 0.511  0.0348 0.0531  0.0953
##  4     0 0.00868 0.833 0.874 0.531  0.0256 0.0582  0.104
##  5     0 0.00487 0.833 0.874 0.531  0.0256 0.0582  0.104
##  6     0 0.0253  0.833 0.876 0.523  0.0261 0.0561  0.105
##  7     0 0.0380  0.832 0.885 0.515  0.0264 0.0546  0.0974
##  8     0 0.0532  0.832 0.889 0.502  0.0272 0.0505  0.0877
##  9     0 0.126   0.828 0.902 0.469  0.0299 0.0491  0.106
## 10     1 0.0380  0.828 0.909 0.473  0.0382 0.0439  0.0915
## 11     0 0.168   0.827 0.915 0.444  0.0312 0.0505  0.117
```

```
## 12     0 0.170   0.827 0.915 0.444  0.0313 0.0505  0.117
## 13     0 0.795   0.819 0.985 0.141  0.0318 0.00976 0.0962
## 14     1 0.0532  0.819 0.915 0.444  0.0423 0.0482  0.0801
## 15     0 2.62    0.816 1     0      0.0324 0       0
## 16     1 0.170   0.792 0.993 0.0375 0.0439 0.00982 0.0559
## 17     1 0.168   0.792 0.993 0.0542 0.0439 0.00982 0.0543
## 18     1 0.126   0.792 0.972 0.270  0.0439 0.0305  0.0541
## 19     1 2.62    0.5   1     0      0      0       0
## 20     1 0.795   0.5   1     0      0      0       0
```

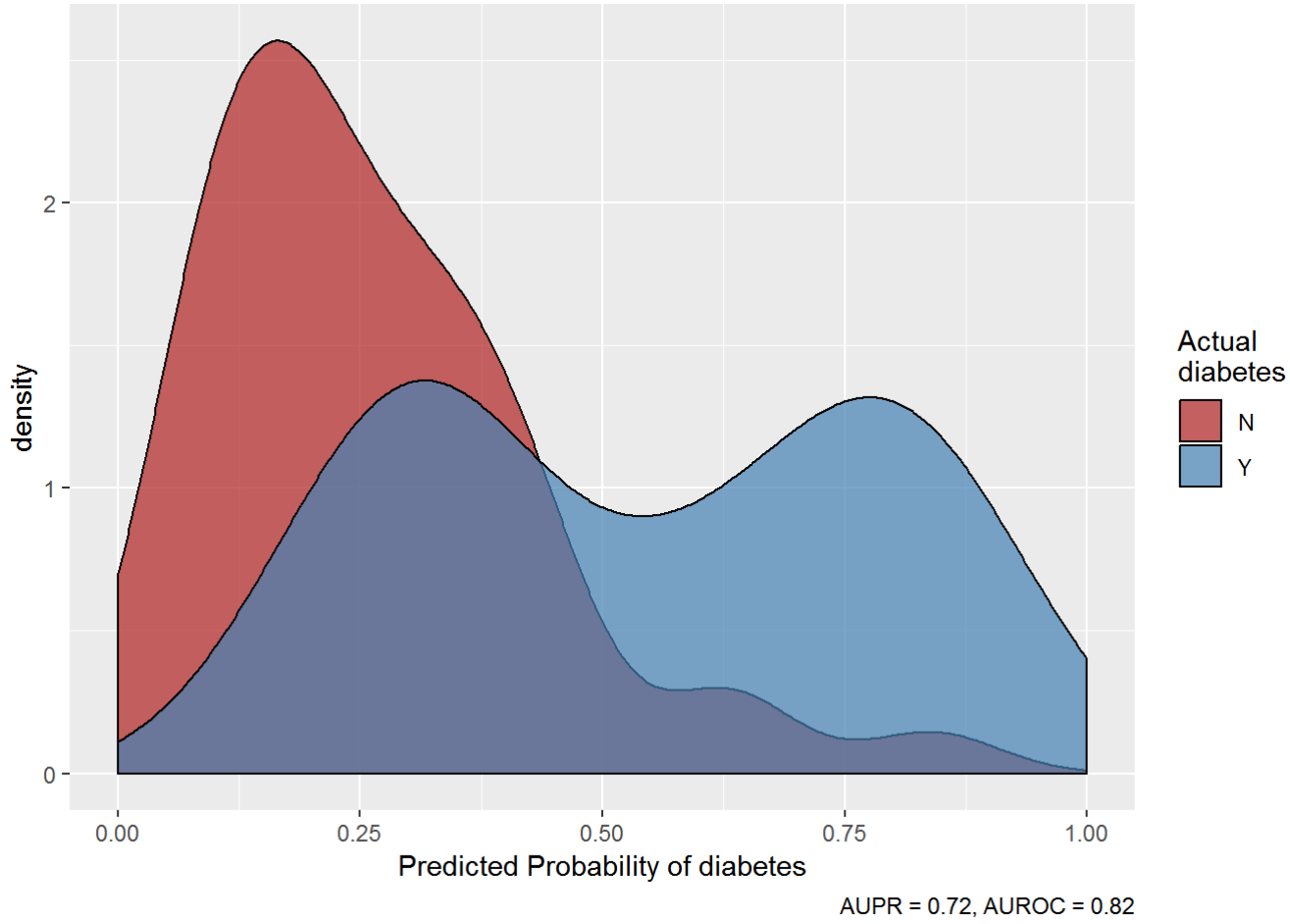
Prediction on test dataset

```
Prediction_Test <- predict(m, d_test_prepped_by_recipe)
Prediction_Test
```

```
## "predicted_diabetes" predicted by glmnet last trained: 2018-07-30 13:14:05
## Performance in training: AUROC = 0.84
```

```
## # A tibble: 68 x 13
##    diabetes predicted_diabe~ pregnancies plasma_glucose diastolic_bp
##  * <fct>               <dbl>       <int>          <int>        <dbl>
##  1 N                   0.341           2            122         76
##  2 Y                   0.302           6            125         78
##  3 Y                   0.712           1            168         88
##  4 N                   0.414           2            129         72.3
##  5 N                   0.150           4            110         76
##  6 N                   0.172           6             80         80
##  7 Y                   0.299          10            115         72.3
##  8 N                   0.363           2            127         46
##  9 Y                   0.828           9            164         78
## 10 Y                   0.168           2             93         64
## # ... with 58 more rows, and 8 more variables: skinfold <dbl>,
## #   insulin <dbl>, age <int>, weight_class_normal <dbl>,
## #   weight_class_obese <dbl>, weight_class_overweight <dbl>,
## #   weight_class_other <dbl>, weight_class_missing <dbl>
```

```
plot(Prediction_Test)
```



Conclusion: it can be seen above that ignored variable are not included in the model training and prediction, # therefore, it may be scenario where ignored variable also have the vital information which can be included in the # model training so that the final prediction based on the overall knowledge of trained model including the vital # information also in the ignored variable based on certain context.