HarvardX PH125.9x Data Science: Choose Your Own Project Submission

Kai Yin Phyllis Lun

3/7/2022

Executive summary

The dataset for this Choose Your Own Project was taken from Kaggle (https://www.kaggle.com/kaushil268/disease-prediction-using-machine-learning), as shared by user KAUSHIL268. The data set was designed for machine learning practice to predict 42 prognoses from 132 symptoms. The training set contains 4920 observations while the testing set contains 42. The goal of this project is to design a machine learning algorithm that is able to predict prognoses accurately from a variety of symptoms.

Methods

I. Data import, cleaning, and exploration

Data structure

There are 4920 observations in the training set and 42 in the test set. Additionally there is a mismatch in terms of number of variables (column) in the test set (133) and in the training set (134). There are 134 variables in the training data set, among which 133 are the symptoms and 1 is the prognosis. For some reason there is also an empty variable (X). The variable X is not available in the testing data set. Therefore, I have removed the empty variable and make sure that the test set and training set have matching columns.

```
dim(train_disease)
```

```
## [1] 4920  134
```

```
dim(test_disease)
```

```
## [1]  42 133
```

```
which(names(train_disease)!=names(test_disease))
```

```
## [1] 134
```

```
names(train_disease)[134]
```

```
## [1] "X"
```

```
summary(train_disease$X)
```

```
##    Mode    NA's
## logical    4920
```

```
summary(test_disease$X)
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```
train_disease<-train_disease %>% select(-X)
```

Symptoms

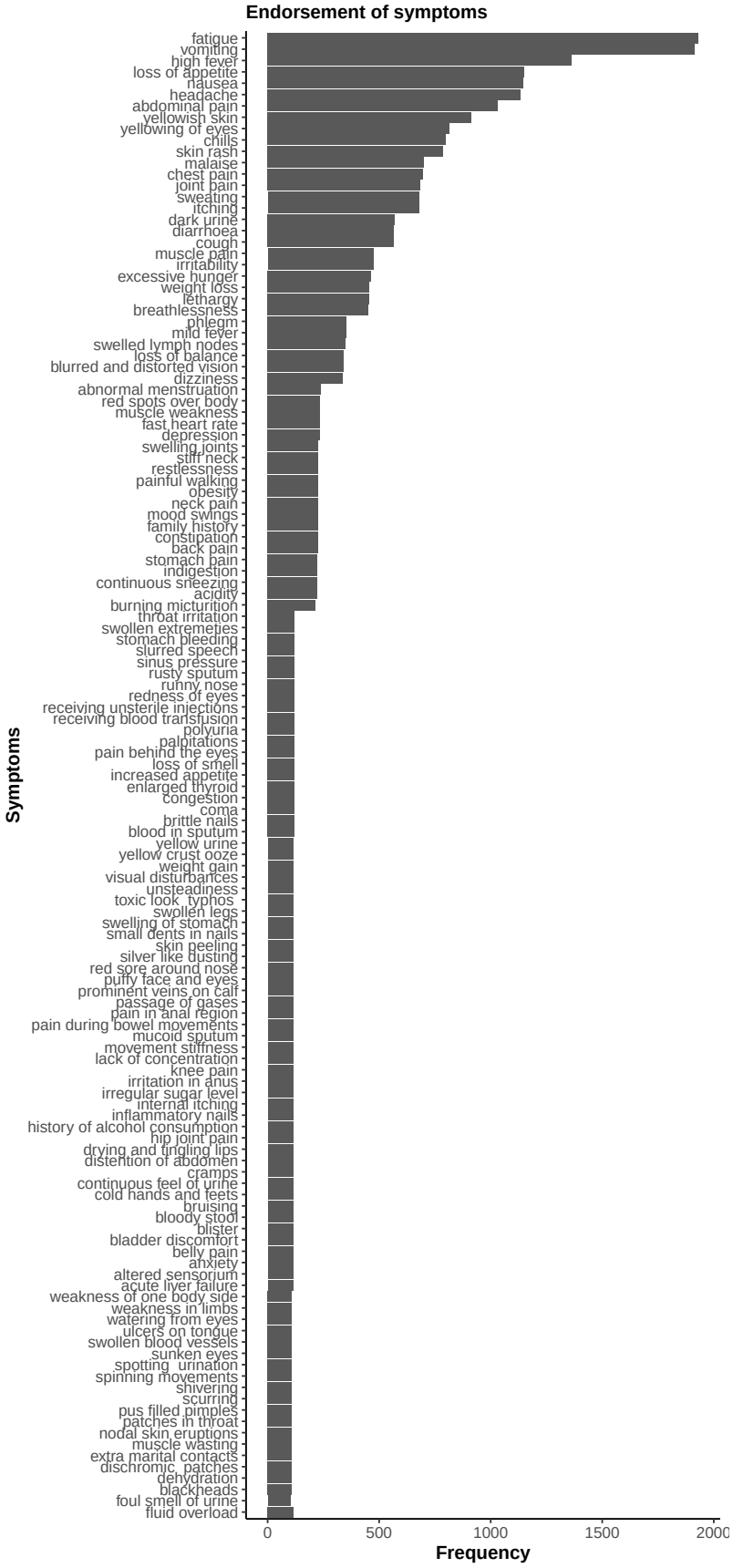
There are a total of 132 symptoms. Their names are extracted and processed from the variable names. The first 6 symptoms are displayed here.

```
symptoms<-train_disease %>% select(-prognosis) %>%
  names() %>% str_replace_all("[^[:alnum:]]", " ")
symptoms[symptoms=="fluid overload 1"]<-"fluid overload"
head(symptoms)
```

```
## [1] "itching"              "skin rash"            "nodal skin eruptions"
## [4] "continuous sneezing"  "shivering"            "chills"
```

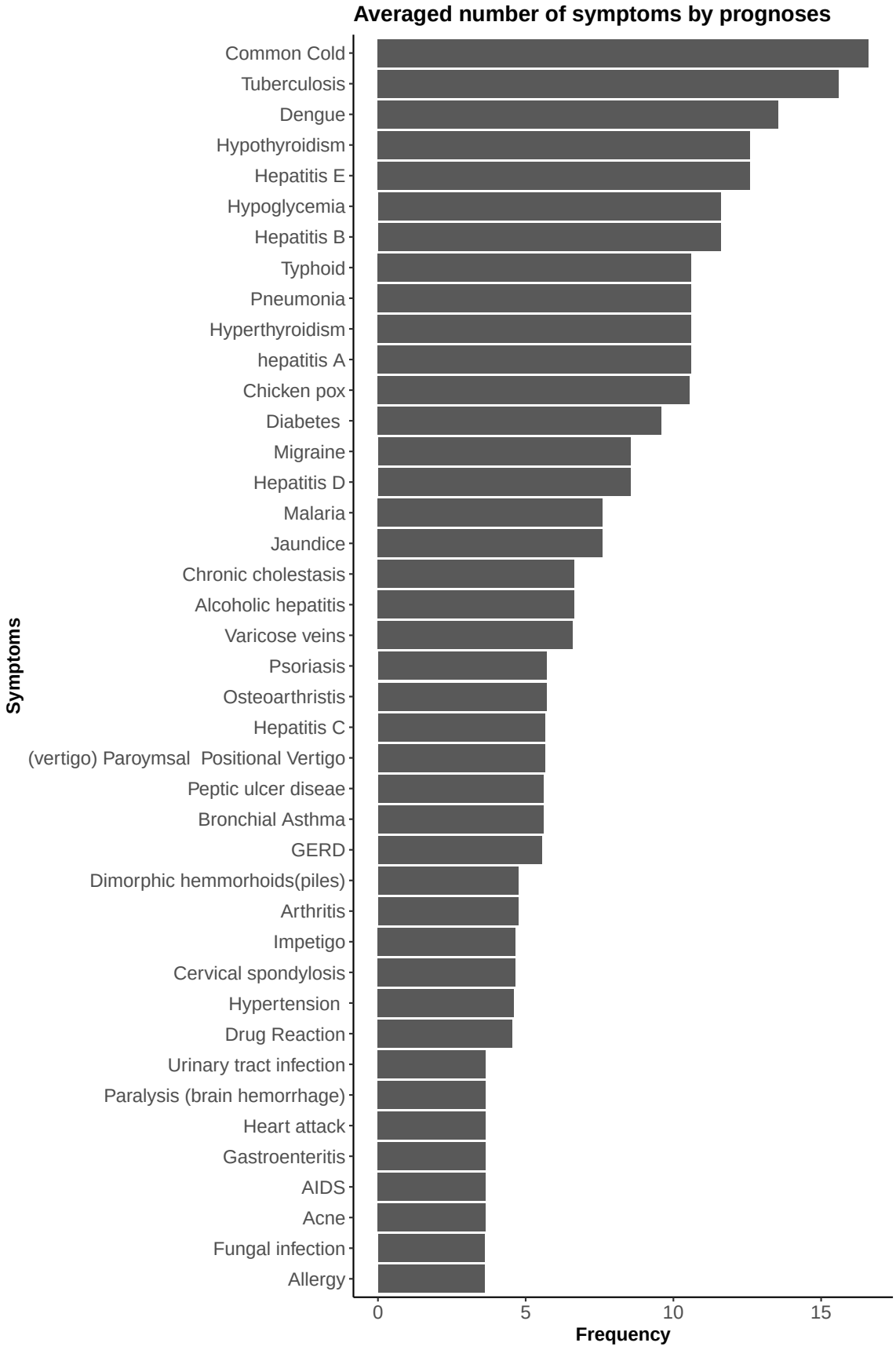
Data visualization on the frequency of symptoms shows that the most frequently reported symptoms are fatigue, vomit, high fever, loss of appetite and neusea.The least frequently reported symptoms are pus filled pimples, blackheads, scurring, foul smell of urine and fluid overload.

```
#Data visualisation:
symptom.df<-train_disease%>% select(-prognosis) %>%
  summarise_all(sum) %>% t() %>% as.data.frame() %>%
  rename("frequency"="V1")
symptom.df<-cbind(symptoms,symptom.df)
rownames(symptom.df)<-NULL
symptom.df %>% ggplot(aes(x = reorder(symptoms,frequency), y=frequency)) +
  geom_bar(stat = "identity", size=1) +
  labs(title="Endorsement of symptoms",x="Symptoms",
       y="Frequency", face="bold")+
  coord_flip () +
  theme_classic()+
  theme(axis.text=element_text(size=10),
        axis.title=element_text(size=12, face="bold"),
        plot.title = element_text(size=12, face="bold"),
        )
```


Endorsement of symptoms
Symptoms
fatigue
vomiting
high fever
loss of appetite
nausea
headache
abdominal pain
yellowish skin
yellowing of eyes
chills
skin rash
malaise
chest pain
joint pain
sweating
itching
dark urine
diarrhoea
cough
muscle pain
irritability
excessive hunger
weight loss
lethargy
breathlessness
phlegm
mild fever
swelled lymph nodes
loss of balance
blurred and distorted vision
dizziness
abnormal menstruation
red spots over body
muscle weakness
fast heart rate
depression
swelling joints
stiff neck
restlessness
painful walking
obesity
neck pain
mood swings
family history
constipation
back pain
stomach pain
indigestion
continuous sneezing
acidity
burning micturition
throat irritation
swollen extremeties
stomach bleeding
slurred speech
sinus pressure
rusty sputum
runny nose
redness of eyes
receiving unsterile injections
receiving blood transfusion
polyuria
palpitations
pain behind the eyes
loss of smell
increased appetite
enlarged thyroid
congestion
coma
brittle nails
blood in sputum
yellow urine
yellow crust ooze
weight gain
visual disturbances
unsteadiness
toxic look (typhos)
swollen legs
swelling of stomach
small dents in nails
skin peeling
silver like dusting
red spore around nose
puffy face and eyes
prominent veins on calf
passage of gases
pain in anal region
pain during bowel movements
mucoid sputum
movement stiffness
lack of concentration
knee pain
irritation in anus
irregular sugar level
internal itching
inflammatory nails
history of alcohol consumption
hip joint pain
drying and tingling lips
distention of abdomen
cramps
continuous feel of urine
cold hands and feets
bruising
bloody stool
blister
bladder discomfort
belly pain
anxiety
altered sensorium
acute liver failure
weakness of one body side
weakness in limbs
watering from eyes
ulcers on tongue
swollen blood vessels
sunken eyes
spotting urination
spinning movements
shivering
scurring
pus filled pimples
patches in throat
nodal skin eruptions
muscle wasting
extra marital contacts
dischromic patches
dehydration
blackheads
foul smell of urine
fluid overload
0
500
1000
1500
2000
Frequency

Data visualization on the averaged number of symptoms by prognosis reveals that people with common cold, tuberculosis, dengue, hypothyroidism and hepatitis E reported the most symptoms (more than 10), while those with allergy, fungal infection, acne, AIDS, and gastroenteritis reported the least number of symptoms (about 3).

```
symptom_no<-train_disease %>% select(-prognosis) %>%rowSums(.)
symptom.df.2<-as.data.frame(cbind(train_disease,symptom_no))
symptom.df.2 %>% group_by(prognosis) %>%
  summarise(n=mean(symptom_no)) %>% arrange(desc(n)) %>%
  ggplot(aes(x=reorder(prognosis,n),y=n))+
  geom_col() +
  coord_flip () +
  theme_classic()+
  labs(title="Averaged number of symptoms by prognoses",
       x="Symptoms", y="Frequency", face="bold")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=12, face="bold"),
        plot.title = element_text(size=14, face="bold"))
```


Averaged number of symptoms by prognoses
Common Cold
Tuberculosis
Dengue
Hypothyroidism
Hepatitis E
Hypoglycemia
Hepatitis B
Typhoid
Pneumonia
Hyperthyroidism
hepatitis A
Chicken pox
Diabetes
Migraine
Hepatitis D
Malaria
Jaundice
Chronic cholestasis
Alcoholic hepatitis
Varicose veins
Psoriasis
Osteoarthristis
Hepatitis C
(vertigo) Paroymsal Positional Vertigo
Peptic ulcer diseae
Bronchial Asthma
GERD
Dimorphic hemmorhoids(piles)
Arthritis
Impetigo
Cervical spondylosis
Hypertension
Drug Reaction
Urinary tract infection
Paralysis (brain hemorrhage)
Heart attack
Gastroenteritis
AIDS
Acne
Fungal infection
Allergy
Symptoms
0
5
10
15
Frequency

Prognoses

In both the training and testing sets, there are 41 unique prognoses. In the training set, each distinct prognosis has 120 observations. Here I verify that the training set and test set have the same prognoses.

```
train_disease %>%  summarize(n_prognsis = n_distinct(prognosis))
```

```
##   n_prognsis
## 1         41
```

```
train_disease %>% count(prognosis, sort=TRUE)
```

```
##                                   prognosis   n
## 1  (vertigo) Paroymsal  Positional Vertigo 120
## 2                                      Acne 120
## 3                                      AIDS 120
## 4                      Alcoholic hepatitis 120
## 5                                   Allergy 120
## 6                                 Arthritis 120
## 7                         Bronchial Asthma 120
## 8                       Cervical spondylosis 120
## 9                                Chicken pox 120
## 10                      Chronic cholestasis 120
## 11                               Common Cold 120
## 12                                    Dengue 120
## 13                                  Diabetes  120
## 14            Dimorphic hemmorhoids(piles) 120
## 15                              Drug Reaction 120
## 16                            Fungal infection 120
## 17                             Gastroenteritis 120
## 18                                       GERD 120
## 19                               Heart attack 120
## 20                                hepatitis A 120
## 21                                Hepatitis B 120
## 22                                Hepatitis C 120
## 23                                Hepatitis D 120
## 24                                Hepatitis E 120
## 25                               Hypertension  120
## 26                              Hyperthyroidism 120
## 27                                 Hypoglycemia 120
## 28                               Hypothyroidism 120
## 29                                     Impetigo 120
## 30                                     Jaundice 120
## 31                                      Malaria 120
## 32                                     Migraine 120
## 33                               Osteoarthristis 120
## 34            Paralysis (brain hemorrhage) 120
## 35                      Peptic ulcer diseae 120
## 36                                  Pneumonia 120
## 37                                  Psoriasis 120
## 38                               Tuberculosis 120
## 39                                    Typhoid 120
```

```
## 40                 Urinary tract infection 120
## 41                          Varicose veins 120
```

```
train_disease<-train_disease %>% mutate(prognosis=as.factor(prognosis))
test_disease<-test_disease %>% mutate(prognosis=as.factor(prognosis))
sum(levels(train_disease$prognosis)==levels(test_disease$prognosis))
```

```
## [1] 41
```

Analyses

Since the prognosis is a categorical variable, only a limited types of machine learning models (e.g., decision tree model and random forest model) are applicable. Accuracy is used as the criterion of model performance.

Here I set up the training set for training machine learning models.

```
train_disease_y<-train_disease$prognosis
train_disease_x <- train_disease %>% select(-prognosis)
```

1. Decision tree model

In the decision tree model, the CP value is fine-tuned to improve the accuracy of the model.The best CP value selected from boostrapping is 0.001005025. In the final model, it is found that mild fever, headache, sweating,yellowing of eyes, and muscle weakness were the most important features.

```
set.seed(3, sample.kind = "Rounding")
fit_rpart <- train(prognosis ~ .,
                   method = "rpart",
                   tuneGrid =
                    data.frame(cp = seq(0.0, 0.1, len = 200)),
                   data = train_disease)
fit_rpart$bestTune
```

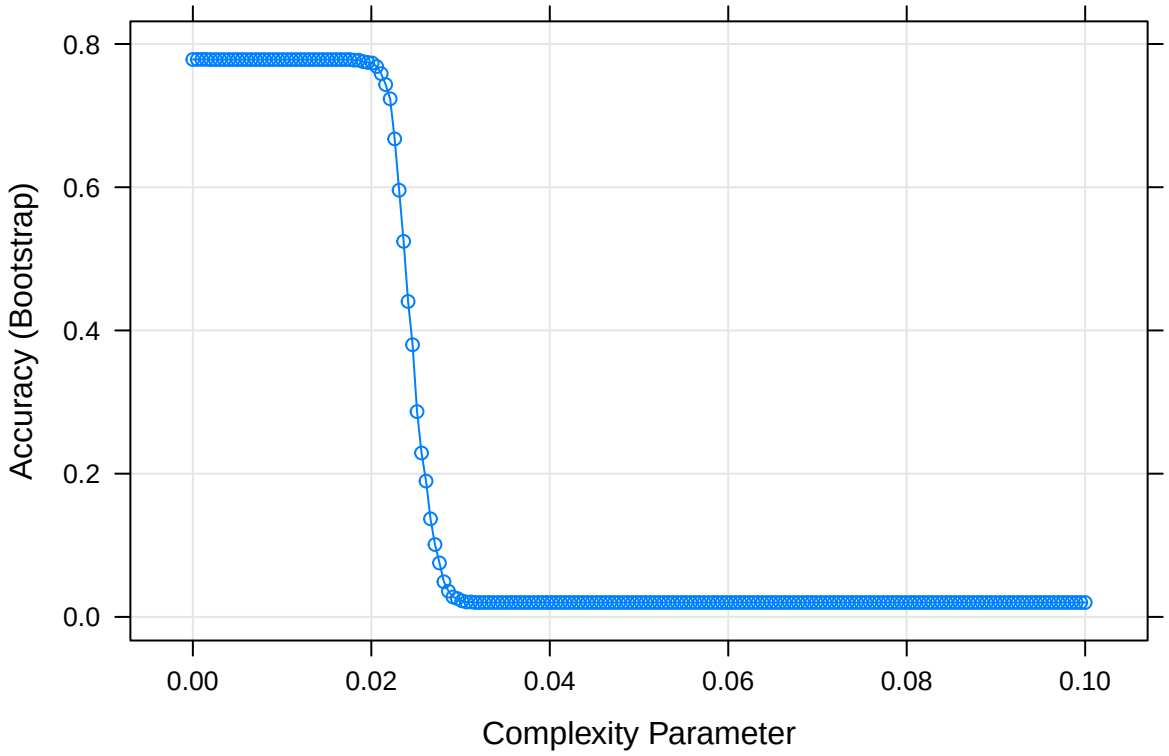
```
##            cp
## 3 0.001005025
```

```
fit_rpart$finalModel$variable.importance[1:5]
```

```
##        mild_fever          headache          sweating yellowing_of_eyes
##          318.0000          298.7502          242.6893          240.0000
##   muscle_weakness
##          233.6349
```

Plot of the CP parameter

```
#plotting the cp parameter tuning
plot(fit_rpart)
```



After fitting the model with the test data, the model reaches an accuracy of 0.88.

```
y_hat <- predict(fit_rpart, test_disease)
dt_accuracy<-confusionMatrix(
  y_hat,as.factor(test_disease$prognosis))$overall["Accuracy"]

disease_results <- tibble(method = "Decision tree",
                          accuracy = dt_accuracy)
```

2. Random forest model

The second machine learning model fitted is the random forest model. From the default setting, a total of 500 trees are planted and 11 features are selected as predictors each time.

```
set.seed(3, sample.kind = "Rounding")
rt_fit <- randomForest(prognosis ~ ., data = train_disease)
rt_fit$ntree #500
```

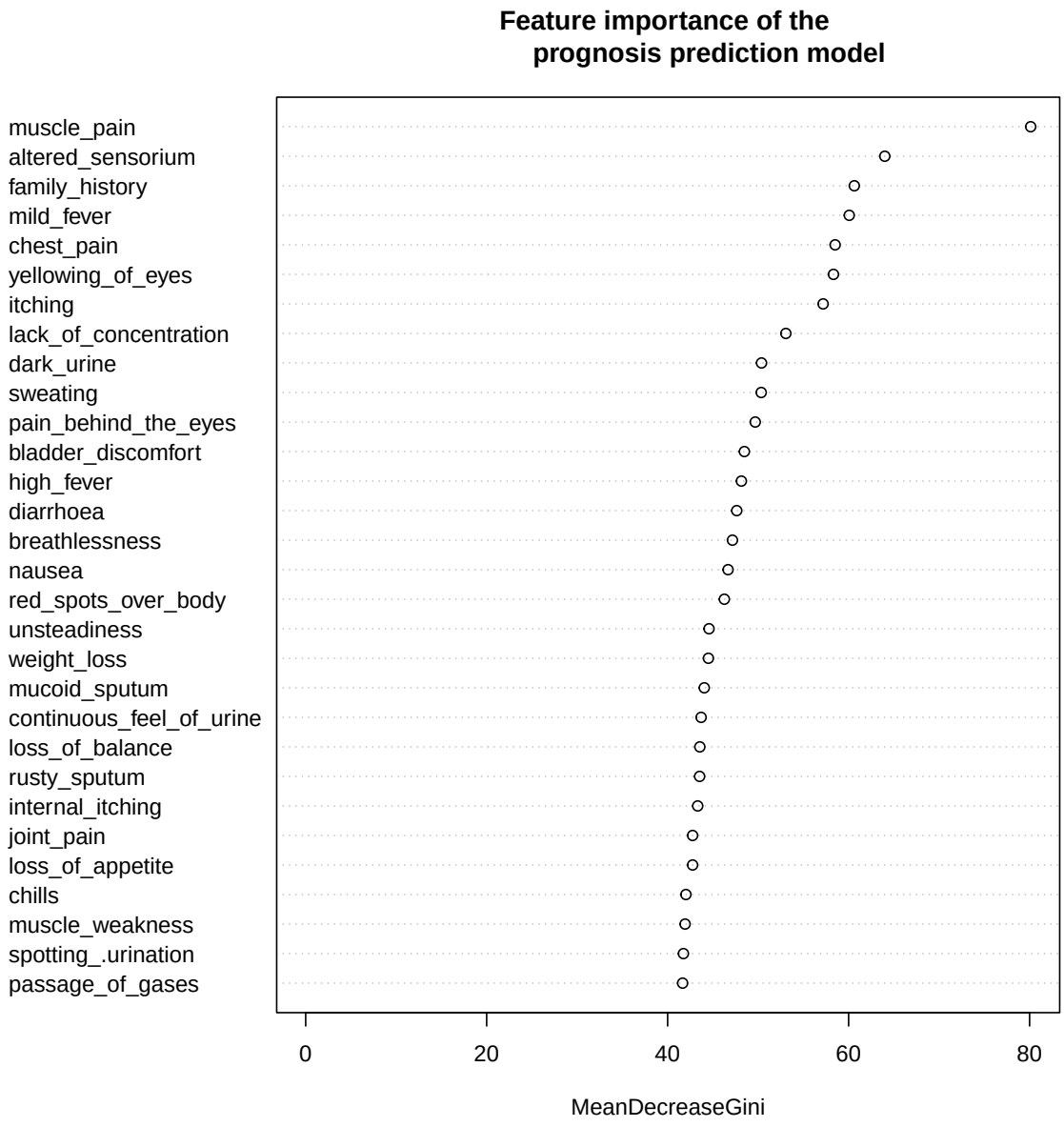
```
## [1] 500
```

```
rt_fit$mtry #11
```

```
## [1] 11
```

Feature importance plot of the random forest model

```
varImpPlot(rt_fit, main="Feature importance of the
           prognosis prediction model", cex=0.9)
```



After fitting the random forest model with the test data, the model reaches an accuracy of 0.98.

```
#accuracy in the test set
y_hat <- predict(rt_fit, test_disease)
rf_accuracy<-confusionMatrix(y_hat,
                             as.factor(test_disease$prognosis))$overall["Accuracy"]
rf_accuracy
```

```
##  Accuracy
## 0.9761905
```

```
disease_results <-bind_rows(disease_results,
                            data.frame(method="Random forest model 1",
                                       accuracy=rf_accuracy))
```

3. Fine-tuned random forest model

In this new version of the random forest model, I have used 5-fold cross-validation (sampling 20% of the training-set observations) to determine the best models from varying number of split nodes and predictors used.

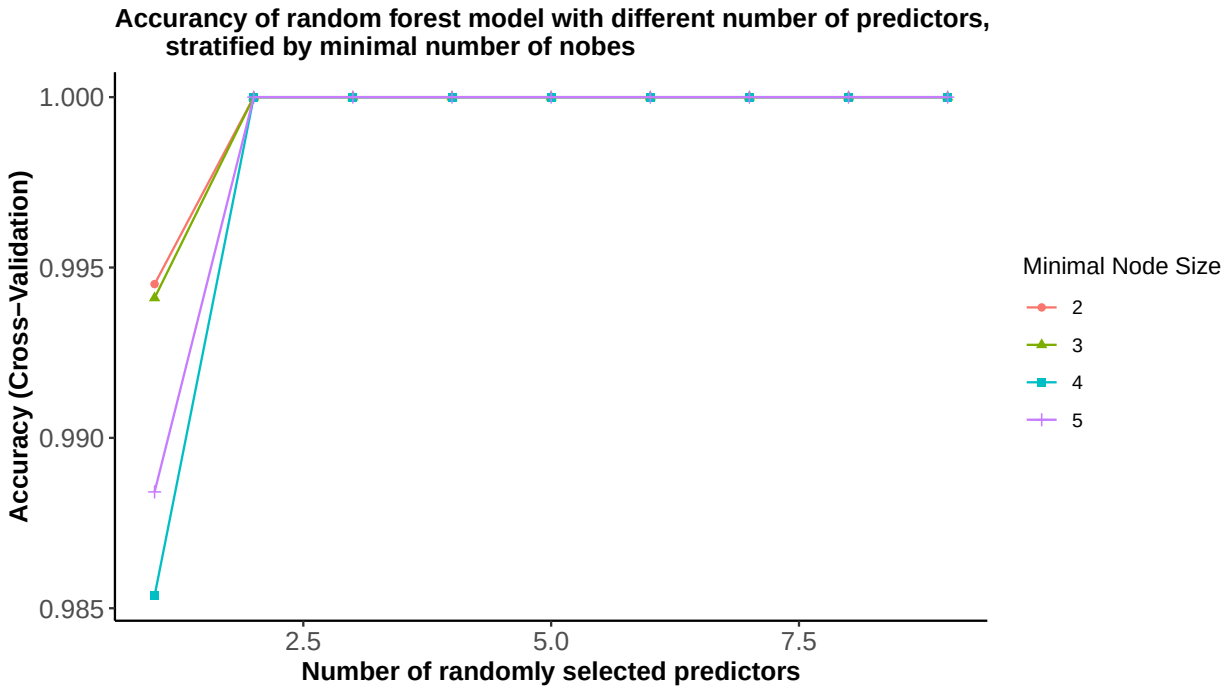
```
library(Rborist)
train_disease_x<-train_disease%>% select(-prognosis)
train_disease_y<-train_disease%>% select(prognosis) %>%
  pull(.)%>% as.factor()


set.seed(3, sample.kind = "Rounding")
control <- trainControl(method="cv", number = 5, p = 0.2)
grid <- expand.grid(minNode=c(2,3,4,5),  predFixed =seq(2:10))
train_rf <-  train(train_disease_x,
                   train_disease_y,
                   method = "Rborist",
                   nTree = 500,
                   trControl = control,
                   tuneGrid = grid)

train_rf
```

```
## Random Forest
##
## 4920 samples
##  132 predictor
##   41 classes: '(vertigo) Paroymsal  Positional Vertigo', 'Acne', 'AIDS', 'Alcoholic hepatitis', 'Allo
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 3936, 3936, 3936, 3936, 3936
## Resampling results across tuning parameters:
##
##   minNode  predFixed  Accuracy   Kappa
##   2        1          0.9945122  0.9943750
##   2        2          1.0000000  1.0000000
##   2        3          1.0000000  1.0000000
##   2        4          1.0000000  1.0000000
##   2        5          1.0000000  1.0000000
##   2        6          1.0000000  1.0000000
##   2        7          1.0000000  1.0000000
##   2        8          1.0000000  1.0000000
```

```
##   2        9          1.0000000  1.0000000
##   3        1          0.9941057  0.9939583
##   3        2          1.0000000  1.0000000
##   3        3          1.0000000  1.0000000
##   3        4          1.0000000  1.0000000
##   3        5          1.0000000  1.0000000
##   3        6          1.0000000  1.0000000
##   3        7          1.0000000  1.0000000
##   3        8          1.0000000  1.0000000
##   3        9          1.0000000  1.0000000
##   4        1          0.9853659  0.9850000
##   4        2          1.0000000  1.0000000
##   4        3          1.0000000  1.0000000
##   4        4          1.0000000  1.0000000
##   4        5          1.0000000  1.0000000
##   4        6          1.0000000  1.0000000
##   4        7          1.0000000  1.0000000
##   4        8          1.0000000  1.0000000
##   4        9          1.0000000  1.0000000
##   5        1          0.9884146  0.9881250
##   5        2          1.0000000  1.0000000
##   5        3          1.0000000  1.0000000
##   5        4          1.0000000  1.0000000
##   5        5          1.0000000  1.0000000
##   5        6          1.0000000  1.0000000
##   5        7          1.0000000  1.0000000
##   5        8          1.0000000  1.0000000
##   5        9          1.0000000  1.0000000
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were predFixed = 2 and minNode = 2.
```

Similar to the prior random forest model, 500 trees are planted in this version. According to the cross-validation results of the model, the most optimal model uses two predictors and 2 observations.

```
ggplot(train_rf)+
  labs(title = "Accurancy of random forest model with different number of predictors,
       stratified by minimal number of nobes")+
  theme_classic()+
  scale_x_continuous("Number of randomly selected predictors") +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=12, face="bold"),
        plot.title = element_text(size=12, face="bold"))
```




```
train_rf$finalModel$param
```

```
## $nTree
## [1] 500
```

The 5 most important variables are headache, vomiting, stiff neck, yellowing of eyes, and abdominal pain. The only important feature shared by the two random forest models is the yellowing of eyes.

```
varImp(train_rf)
```

```
## Rborist variable importance
##
##   only 20 most important variables shown (out of 132)
##
##                          Overall
## headache                  100.00
## vomiting                   74.52
## stiff_neck                 70.28
## yellowing_of_eyes          67.17
## abdominal_pain             65.79
## hip_joint_pain             64.39
## muscle_pain                63.56
## bloody_stool               63.21
## irritability               62.82
## chills                     62.71
## neck_pain                  61.82
## sunken_eyes                61.08
## high_fever                 60.72
## muscle_wasting             59.84
```

```
## yellowish_skin             59.56
## burning_micturition        57.55
## nausea                     56.54
## painful_walking            54.94
## continuous_feel_of_urine   54.93
## passage_of_gases           54.65
```

The fine-tuned random forest model reaches an accuracy of 0.98.

```
y_hat <- predict(train_rf, test_disease)
rf_tuned_accuracy<-confusionMatrix(y_hat,
                                   as.factor(test_disease$prognosis))$
  overall["Accuracy"] #0.9761905
rf_tuned_accuracy
```

```
##  Accuracy
## 0.9761905
```

```
disease_results <-bind_rows(disease_results,
                            data.frame(method="Random forest model 2",
                                       accuracy=rf_tuned_accuracy))
```

Conclusion

In this self-selected project, I trained several decision tree and random forest models to predict diseases prognoses from a variety of symptoms. The data set, as obtained from Kaggle, contained 132 symptoms and the 42 corresponding prognoses. Both the decision tree and random forest models achieved fairly high accuracy (range: 0.88-0.98). In comparison, both random forest models (default and fine-tuned) have the highest accuracy of 0.98. The fine-tuning did not improve the performance of the random forest model.

	method	accuracy
...1	Decision tree	0.8809524
Accuracy...2	Random forest model 1	0.9761905
Accuracy...3	Random forest model 2	0.9761905

Limitations

There are several limitations to this project. Firstly, there was only one observation of each observation (except for Fungal infection, which has 2 observations) in the test set. The lack of variations in features for each prognosis might have limited the generalizability of the accuracy calculated from the current test set. On a similar note, each prognosis has 120 observations in the training set. The training data set, as well as the machine learning models, would be benefited from some variability in the number of observations for each prognosis, such as prevalence of prognoses in a hospital or primary care setting. The inclusion of such data would have greatly improved the machine learning model and increased utility of the machine learning model. The small size of the training set have also limited the bootstrapping and cross-validation procedures. Also, the model performance would also be greatly enhanced if patient data such as sociodemographic variables and specific medical history (other than the family history that was vaguely stated as a symptom in the current data sets) are available.

Future work and implications

Despite the limitations in the data sets, the strong performance of the final model has given me confidence that the model could readily predict diseases based on the given set of symptoms. As a proof of concept, the current model would be a good first step to develop a machine learning model for medical diagnostics. It would be interesting to continue developing the model based on real-world hospital records (which would have addressed the limitations stated above), rendering the model more flexible and adaptive to patient-level characteristics and more powerful for handling a wider variety of symptoms and prognoses.