Preprocess data

Import raw data

In[1]:=
```
rawdata = SemanticImport["D:\\Google Drive\\Rich
     Internship 2020\\New data set\\data\\data (not normalized).xlsx"];
```

Synthesize missing values:

In[36]:=
```
data1 =
  Table[Values[Transpose[Normal[SynthesizeMissingValues[Normal[rawdata[All, {2, 3, i}]],
        Method → <|"LearningMethod" → "KernelDensityEstimation", "EvaluationStrategy" →
           "ModeFinding"|>, PerformanceGoal → "Quality"]]][[3]], {i, 4, 41}];
```

In[37]:=
```
Export["data (synthesized missing values).csv", Transpose[data1]];
```

In[46]:=
```
rawdata2 = Transpose[data1];
rawdata2 = Transpose[Table[N[Standardize[rawdata2[[All, k]]]], {k, 1, 38}]];
Export["data standardized.csv", rawdata2]
```

Out[48]= data standardized.csv

Classification

Import data

In[78]:=
```
importdata = SemanticImport[
   "D:\\Google Drive\\Rich Internship 2020\\New data set\\data\\data.xlsx"];
data = Values[Normal[importdata]][[All, 2 ;; 42]];
labels = Import["D:\\Google Drive\\Rich
       Internship 2020\\New data set\\data\\data.xlsx"][[1]][[1]];
```

column labels

In[68]:=
```
Table[{i, labels[[i]]}, {i, 1, 42}] // TableForm
```

Out[68]//TableForm=

1	patient_id
2	age
3	gender
4	white blood cell
5	Eosinophil
6	lactate dehydrogenase
7	C-reactive protein
8	Monocyte
9	Alanine transaminase
10	Aspartate transaminase
11	Glomerular filtration rate
12	Partial Pressure Of Oxygen
13	Hemoglobin
14	red blood cell
15	Hematocrit
16	Platelet
17	Albumin
18	Total bilirubin
19	Potassium
20	Sodium
21	Chlorine
22	Anion gap
23	PH
24	PaCO2
25	Blood oxygen saturation
26	Respiratory index
27	B-type Natriuretic Peptide
28	Myoglobin
29	Troponin I
30	Urine specific gravity
31	Urine leukocyte
32	Urinary Erythrocytes
33	Urea
34	Uric acid
35	AST-ALT ratio
36	Total protein
37	Globulin
38	Plasma prothrombin time
39	Thrombin time
40	Plasma fibrinogen
41	Creatine kinase isoenzyme
42	Severity

classification model

randomize data for training/test split

In[81]:=
```
SeedRandom[123];
rs = RandomSample[data, 285];
```

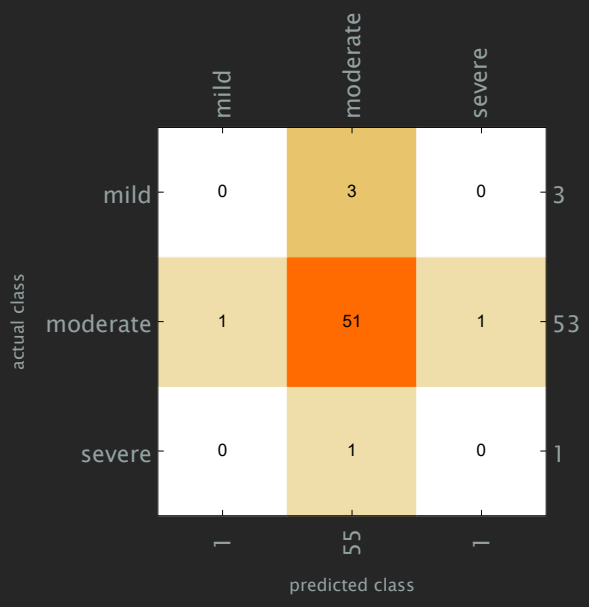
Split data into 80% training and 20% test

```
In[91]:= trainingset = rs[[1 ;; 0.80 * 285]];
         testset = rs[[0.80 * 285 + 1 ;; 285]];

In[96]:= c = Classify[trainingset → 41, Method → "NeuralNetwork", PerformanceGoal → "Quality"]

Out[96]= ClassifierFunction[ Input type: Mixed (number: 40)
                             Classes: mild, moderate, severe ]
```

Accuracy and Confusion Matrix

```
In[97]:= cm = ClassifierMeasurements[c, testset → 41]

Out[97]= ClassifierMeasurementsObject[ Classifier: NeuralNetwork
                                       Number of test examples: 57 ]

In[98]:= cm["Accuracy"]

Out[98]= 0.894737

In[99]:= cm["ConfusionMatrixPlot"]
```

Out[99]=



deployed model

```
In[100]:= SeedRandom[321];
          rs = RandomSample[data, 285];

In[102]:= trainingset = rs[[1 ;; 285]];
```

In[103]:=
```
c = Classify[trainingset → 41, Method → "NeuralNetwork", PerformanceGoal → "Quality"]
```

Out[103]=
```
ClassifierFunction[  Input type: Mixed (number: 40)
Classes: mild, moderate, severe ]
```

In[104]:=
```
Information[c]
```

Out[104]=

Classifier information

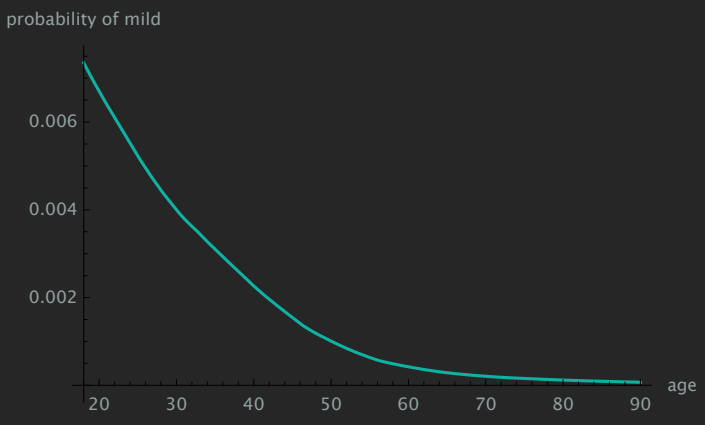
Data type	Mixed (number: 40)
Classes	mild, moderate, severe
Accuracy	(85.2 ± 3.1)%
Method	NeuralNetwork
Single evaluation time	3. ms/example
Batch evaluation speed	60.6 examples/ms
Loss	0.718 ± 0.16
Model memory	535. kB
Training examples used	285 examples
Training time	13. s

severity functions based on age

Plots

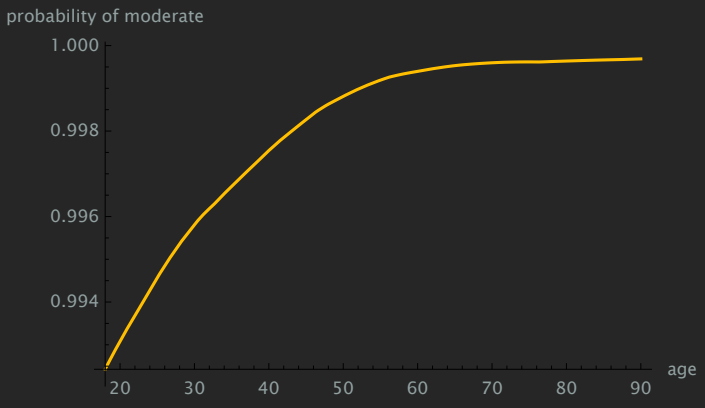
In[152]:=
```
Plot[f[age], {age, Min[data[[All, 1]]], Max[data[[All, 1]]]},
 PlotStyle → , AxesLabel → {"age", "probability of mild"}]
```

Out[152]=



In[153]:=
```
Plot[g[age], {age, Min[data[[All, 1]]], Max[data[[All, 1]]]},
 PlotStyle → , AxesLabel → {"age", "probability of moderate"}]
```

Out[153]=



In[155]:=
```
Plot[h[age], {age, Min[data[[All, 1]]], Max[data[[All, 1]]]},
 AxesLabel → {"age", "probability of severe"}, PlotStyle → ]
```

Out[155]=