**An Elementary Unification of Machine Learning Mechanisms to Induct Febrile-Episode Survival Period for Acute Lymphoblastic Leukemia Patients**

The following codes have been used to produce this research article:

mat\_file = matfile('febrile.mat','Writable',true);

Gender = [1;1;0;1;1;0;0;0;0;1;0;1;1;1;0;0;0;0;0;1;0;0;1;0;1;1;0;0;1;1;1;0;1;0;0;1;0;1;0;0;0;0;1;0;0;0;1;1;1;0;0;0;1;1;0;1;0;1;0;0;1;0;1;0;0];

Immunophenotype={'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'T-Cell';'T-Cell';'Pre-B';'B-Cell';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'T-Cell';'T-Cell';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B';'Pre-B'; 'Pre-B'; 'Pre-B'; 'T-Cell'; 'Pre-B'; 'Pre-B'; 'Pre-B'};

Febrile = [22;22;21;23;25;33;15;47;46;49;27;24;22;38;47;40;23;31;13;25;25;23;10;23;25;37;20;12;25;16;49;15;6;49;49;30;25;11;15;28;42;39;24;36;35;25;17;39;42;37;37;15;31;15;29;23;33;23;9;29;36;39;57;14;35];

Censored=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0; 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];

mat\_file = matfile('febrile.mat','Writable',true);

save('febrile.mat', 'Gender', 'Immunophenotype', 'Febrile', 'Censored')

%mat\_file.myCell = myCell;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

(Two topics work for the model: Hazard and Survivor Functions for different groups

Cox proportional standard model for hazard)

mat\_file = matfile('febrile.mat','Writable',true);

Gender = [1;1;0;1;1;0;0;0;0;1;0;1;1;1;0;0;0;0;0;1;0;0;1;0;1;1;0;0;1;1;1;0;1;0;0;1;0;1;0;0;0;0;1;0;0;0;1;1;1;0;0;0;1;1;0;1;0;1;0;0;1;0;1;0;0];

Immunophenotype=[1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;0;0;1;2;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;0;0; 1;1;1;1;1;1;1;1;1;1;1;1; 1;1;1;1;1;1;1;1;0;1;1;1];

Febrile = [22;22;21;23;25;33;15;47;46;49;27;24;22;38;47;40;23;31;13;25;25;23;10;23;25;37;20;12;25;16;49;15;6;49;49;30;25;11;15;28;42;39;24;36;35;25;17;39;42;37;37;15;31;15;29;23;33;23;9;29;36;39;57;14;35];

Censored=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0; 0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];

mat\_file = matfile('febrile.mat','Writable',true);

save('febrile.mat', 'Gender', 'Immunophenotype', 'Febrile', 'Censored')

%mat\_file.myCell = myCell;

**1) Hazard and Survivor function for different groups**

**Step 1. Load and organize sample data.**

load febrile.mat

female = [Febrile(Gender==1),Censored(Gender==1)];

male = [Febrile(Gender==0),Censored(Gender==0)];

**Step 2. Estimate and plot cumulative distribution function for each gender.**

figure()

ecdf(gca,female(:,1),'Censoring',female(:,2));

hold on

[f,x] = ecdf(male(:,1),'Censoring',male(:,2));

stairs(x,f,'--r')

hold off

legend('female','male','Location','SouthEast')

**Step 3. Plot survivor functions.**

figure()

ax1 = gca;

ecdf(ax1,female(:,1),'Censoring',female(:,2),'function','survivor');

hold on

[f,x] = ecdf(male(:,1),'Censoring',male(:,2),'function','survivor');

stairs(x,f,'--r')

legend('female','male')

**Step 4. Fit Weibull survivor functions.**

Fit Weibull distributions to febrile times of female and male patients.

pd = fitdist(female(:,1),'wbl','Censoring',female(:,2))

pd2 = fitdist(male(:,1),'wbl','Censoring',male(:,2))

**Plot the Weibull survivor functions for female and male patients on estimated survivor functions.**

plot(0:1:60,1-cdf('wbl',0:1:60, 30.8941, 2.4192),'-.')

plot(0:1:60,1-cdf('wbl',0:1:60, 32.8792, 2.85247),':r')

hold off

legend('Festimated','Mestimated','FWeibull','MWeibull')

**Step 5. Estimate cumulative hazard and fit Weibull cumulative hazard functions.**

figure()

[f,x] = ecdf(female(:,1),'Censoring',female(:,2),...

'function','cumhazard');

plot(x,f)

hold on

plot(x,cumsum(pdf(pd,x)./(1-cdf(pd,x))),'-.')

[f,x] = ecdf(male(:,1),'Censoring',male(:,2),...

'function','cumhazard');

plot(x,f,'--r')

plot(x,cumsum(pdf(pd2,x)./(1-cdf(pd2,x))),':r')

legend('Festimated','FWeibull','Mestimated','MWeibull',...

'Location','North')

figure()

[f,x] = ecdf(female(:,1),'Censoring',female(:,2),...

'function','cumhazard');

plot(x,f)

hold on

[f,x] = ecdf(male(:,1),'Censoring',male(:,2),...

'function','cumhazard');

plot(x,f,'--r')

legend('Festimated','FWeibull' ,...

'Location','North')

**2)** **Cox proportional hazards Model for ALL Data**

Fit a Cox proportional hazard function with the variable Gender as the predictor variable, taking the censoring into account.

X = Gender;

[b,logl,H,stats] = coxphfit(X,Febrile, 'censoring',Censored);

**Add Age and Weight to the model.**

Fit a Cox proportional hazards model with the variables Gender and Immunophenotype.

X = [Gender Immunophenotype];

[b,logl,H,stats] = coxphfit(X,Febrile, 'censoring',Censored);

stats.beta

stats.p

### 3) Viewing a Classification/Regression Tree

mat\_file = matfile('febrile.mat','Writable',true);

Gender = {'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'male'; 'female'; 'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'female'; 'male'; 'female'; 'male'; 'male'};

%Immunophenotype=[1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;1;1];

Febrile = [23;25;33;27;24;23;31;25;25;23;23;25;25;30;25;28;24;35;25;31;29;23;33;23;29;35];

%Censored=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];

mat\_file = matfile('febrile.mat','Writable',true);

save('febrile.mat','Gender', 'Febrile')

%save('febrile.mat', 'Gender','Immunophenotype', 'Febrile', 'Censored')

%mat\_file.myCell = myCell;

ctree = fitctree(Febrile,Gender);

**view(ctree)**

**view(ctree,'mode','graph')**

### Predicting Responses with Classification Tree

Xnew = [24;25;33;27;24;24;31;25;25;24;24;25;25;30;25;28;24;35;25;31;29;24;33;24;29;35];

Ynew = predict(ctree,Xnew);

**Pruning**

mat\_file = matfile('febrile.mat','Writable',true);

Gender = {'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'male'; 'female'; 'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'female'; 'male'; 'female'; 'male'; 'male'};%Immunophenotype=[1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;1;1];

Febrile = [23;25;33;27;24;23;31;25;25;23;23;25;25;30;25;28;24;35;25;31;29;23;33;23;29;35];

%Censored=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];

mat\_file = matfile('febrile.mat','Writable',true);

save('febrile.mat','Gender', 'Febrile')

%save('febrile.mat', 'Gender','Immunophenotype', 'Febrile', 'Censored')

%mat\_file.myCell = myCell;

ctree = fitctree(Febrile,Gender);

#### Examining Resubstitution Error

resuberror = resubLoss(ctree)

resuberror= 4.7188

The resubstitution loss for a regression tree is the mean-squared error. The resulting value indicates that a typical predictive error for the tree is about the square root of 4.7, or a bit over 2.

**Estimate the cross-validation MSE (mean squared errors)**

rng 'default';

cvrtree = crossval(ctree);

cvloss = kfoldLoss(cvrtree)

cvloss = 23.8065

**Fit a Classification Tree (gives the same tree as above but with histogram as another figure)**

tc = fitctree(Febrile,Gender)

rng(1); % For reproducibility

MdlDefault = fitctree(Febrile,Gender,'CrossVal','on');

numBranches = @(x)sum(x.IsBranch);

mdlDefaultNumSplits = cellfun(numBranches, MdlDefault.Trained);

figure;

histogram(mdlDefaultNumSplits)

view(MdlDefault.Trained{1},'Mode','graph')

**Compute ROC curve using SVM**

**a) Load the data (for female)**

mat\_file = matfile('febrile.mat','Writable',true);

Gender = {'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'male'; 'female'; 'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'female'; 'male'; 'female'; 'male'; 'male'};%Immunophenotype=[1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;1;1];

Febrile = [23;25;33;27;24;23;31;25;25;23;23;25;25;30;25;28;24;35;25;31;29;23;33;23;29;35];

%Censored=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];

mat\_file = matfile('febrile.mat','Writable',true);

save('febrile.mat','Gender', 'Febrile')

resp1=strcmp(Gender,'female'); %resp1 is 1 if Gender=’female’ and is 0 if Gender =’male’

**Train an SVM classifier on the same sample data. Standardize the data.**

mdlSVM = fitcsvm(Febrile,resp1,'Standardize',true);

**Compute the posterior probabilities (scores).**

mdlSVM = fitPosterior(mdlSVM);

[~,score\_svm] = resubPredict(mdlSVM);

The second column of score\_svm contains the posterior probabilities of female (Febrile).

**Compute the standard ROC curve using the scores from the SVM model.**

[Xsvm,Ysvm,Tsvm,AUCsvm] = perfcurve(resp1,score\_svm(:,mdlSVM.ClassNames),'true');

**Plot the ROC curve.**

plot(Xsvm,Ysvm)

legend('Support Vector Machines','Location','Best')

xlabel('False positive rate');

ylabel('True positive rate');

title('ROC Curve for SVM for an Instance of Positive Class (Female)')

hold off

**b) Load the data (for male)**

mat\_file = matfile('febrile.mat','Writable',true);

Gender = {'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'male'; 'female'; 'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'female'; 'male'; 'female'; 'male'; 'male'};%Immunophenotype=[1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;1;1];

Febrile = [23;25;33;27;24;23;31;25;25;23;23;25;25;30;25;28;24;35;25;31;29;23;33;23;29;35];

%Censored=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];

mat\_file = matfile('febrile.mat','Writable',true);

save('febrile.mat','Gender', 'Febrile')

resp1=strcmp(Gender,'female'); %resp1 is 1 if Gender=’female’ and is 0 if Gender =’male’

**Train an SVM classifier on the same sample data. Standardize the data.**

mdlSVM = fitcsvm(Febrile,resp1,'Standardize',false);

**Compute the posterior probabilities (scores).**

mdlSVM = fitPosterior(mdlSVM);

[~,score\_svm] = resubPredict(mdlSVM);

The second column of score\_svm contains the posterior probabilities of female (Febrile).

**Compute the standard ROC curve using the scores from the SVM model.**

[Xsvm,Ysvm,Tsvm,AUCsvm] = perfcurve(resp1,score\_svm(:,mdlSVM.ClassNames),'false');

**Plot the ROC curve.**

plot(Xsvm,Ysvm)

legend('Support Vector Machines','Location','Best')

xlabel('False positive rate');

ylabel('True positive rate');

title('ROC Curve for SVM for an Instance of Negative Class (male)')

hold off

**Compute the standard ROC curve using the scores from the naive Bayes classification.**

[Xnb,Ynb,Tnb,AUCnb] = perfcurve(resp1,score\_nb(:,mdlNB.ClassNames),'true');

**Plot the ROC curves on the same graph.**

plot(Xlog,Ylog)

hold on

plot(Xsvm,Ysvm)

plot(Xnb,Ynb)

legend('Logistic Regression','Support Vector Machines','Naive Bayes','Location','Best')

xlabel('False positive rate');

ylabel('True positive rate');

title('ROC Curves for Logistic Regression, SVM, and Naive Bayes Classification')

hold off

**Compare the area under the curve for all three classifiers.**

AUClog

AUCsvm

AUCnb

All values are in my case 0.7030

Logistic regression has the highest AUC measure for classification and naive Bayes has the lowest. This result suggests that logistic regression has better in-sample average performance for this sample data.

**Plot ROC curve for classification tree**

mat\_file = matfile('febrile.mat','Writable',true);

Gender = {'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'male'; 'female'; 'female'; 'female'; 'male'; 'male'; 'female'; 'male'; 'male'; 'female'; 'male'; 'female'; 'male'; 'female'; 'male'; 'male'};%Immunophenotype=[1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;1;1];

Febrile = [23;25;33;27;24;23;31;25;25;23;23;25;25;30;25;28;24;35;25;31;29;23;33;23;29;35];

%Censored=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];

mat\_file = matfile('febrile.mat','Writable',true);

save('febrile.mat','Gender', 'Febrile') % predictor=Febrile, response=Gender

Model = fitctree(Febrile,Gender, 'ClassNames',{'female','male'});

**Predict the class labels and scores for the Gender based on the tree model**

[~,score] = resubPredict(Model);

[X,Y,T,~,OPTROCPT,suby,subnames] = perfcurve(Gender, score(:,1),'female');

**Plot the ROC curve and the optimal operating point on the ROC curve.**

plot(X,Y)

hold on

plot(OPTROCPT(1),OPTROCPT(2),'ro')

xlabel('False positive rate')

ylabel('True positive rate')

title('ROC Curve for Classification by Classification Tree')

hold off

**Find the threshold that corresponds to the optimal operating point.**

T((X==OPTROCPT(1))&(Y==OPTROCPT(2)))

**Specify male as the negative class and compute and plot the ROC curve for female.**

[X,Y,~,~,OPTROCPT] = perfcurve(Gender,score(:,1), 'female','negClass','male');

OPTROCPT

plot(X,Y)

hold on

plot(OPTROCPT(1),OPTROCPT(2),'ro')

xlabel('False positive rate')

ylabel('True positive rate')

title('ROC Curve for Classification by Classification Trees')

hold off

**Compute Pointwise Confidence Intervals for ROC curve**

mat\_file = matfile('febrile.mat','Writable',true);

Gender = [true,true,false,false,true,false,false,true,false,false,false,true,true,true,false,false,true,false, false,true,false,true,false,true,false,false]'; %1 for female, 0 for male

%Immunophenotype=[1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;0;1;1;1;1;1;1;1;1;1;1];

Febrile = [23;25;33;27;24;23;31;25;25;23;23;25;25;30;25;28;24;35;25;31;29;23;33;23;29;35];

%Censored=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];

mat\_file = matfile('febrile.mat','Writable',true);

save('febrile.mat','Gender', 'Febrile')

% Gender= Response, Febrile=Predictor

**Fit a logistic regression model.**

mdl = fitglm(Febrile,Gender,'Distribution','binomial','Link','logit');

**Plot the pointwise confidence intervals. (used this for my article)**

errorbar(X,Y(:,1),Y(:,1)-Y(:,2),Y(:,3)-Y(:,1));

xlim([-0.02,1.02]);

ylim([-0.8,1.8]);

xlabel('False positive rate')

ylabel('True positive rate')

title('ROC Curve with Pointwise Confidence Bounds')

legend('PCBwVA','Location','Best')

[X1,Y1,T1] = perfcurve(Gender,mdl.Fitted.Probability,'true','NBoot',1000);

figure()

errorbar(X1(:,1),Y1(:,1),Y1(:,1)-Y1(:,2),Y1(:,3)-Y1(:,1));

xlim([-0.02,1.02]);

ylim([-0.02,1.02]);

xlabel('False positive rate')

ylabel('True positive rate')

title('ROC Curve with Pointwise Confidence Bounds')

legend('PCBwTA','Location','Best')

**Specify the threshold values to fix and compute the ROC curve. Then plot the curve.**

[X1,Y1,T1] = perfcurve(Gender,mdl.Fitted.Probability,'true','NBoot',1000,'TVals',0:0.05:1);

figure()

errorbar(X1(:,1),Y1(:,1),Y1(:,1)-Y1(:,2),Y1(:,3)-Y1(:,1));

xlim([-0.02,1.02]);

ylim([-0.02,1.02]);

xlabel('False positive rate')

ylabel('True positive rate')

title('ROC Curve with Pointwise Confidence Bounds')

legend('PCBwTA','Location','Best')