**Linear Regression (Language Used: R)**

- **To perform linear regression**

model=lm(cnt~.,data=hour)

summary(model)

-**To compare the performance of two models in order to distinguish significant features**

first=lm(cnt~.,data=hour)

second=lm(cnt~season+yr+hr+holiday+weekday+weathersit+temp+atemp+hum+windspeed,data=hour)

anova(second,first)//to check if removing some features has significant effect on the performance of the model

**Principal Component Regression(Language Used: R)**

-**To install the package and load the 'pls' library**

install.packages("pls")

library(pls)

-**To perform 10 fold cross-validation**

result1 <- pcr(cnt~.,ncomp = 10, data = train, validation = "CV")

plot(RMSEP(result1))//plots the RMSE vs number of components

RMSEP(result1,newdata=test)//uses the trained model on the test data

plot(result1, ncomp = 2, asp = 1, line = TRUE)//shows how the data is spread with respect to the regression line by taking 2 components

**Support Vector Regression(Language Used: R)**

-**To install package and load the 'e1071' library**

install.packages("e1071")

library(e1071)

-**To use different kernels (apart from the ones enlisted, other parameters can also be tuned for the respective kernels)**

model<-svm(formula = cnt ~ ., data = hour1, type = "eps", kernel = "linear", cross = 10)

model<-svm(cnt~.,data=hour,type='eps',kernel='polynomial',degree=2,cost=2,cross=10)

model<-svm(formula = cnt ~ ., data = hour, type = "eps", kernel = "radial", cost = 15, gamma = 0.005, cross = 10)

model<-svm(formula = cnt ~ ., data = hour1, type = "eps", kernel = "sigmoid", cost = 20, gamma = 0.5, epsilon = 0.01, cross = 10)

summary(model)

-**For fine tuning the parameters**

model<- tune.svm(cnt~.,data=hour,type='eps',kernel='radial',gamma = 2^(-1:1), cost = 2^(2:4),cross=10)

**Multinomial Logistic Regression(Language Used: Matlab)**

%extracting predictors from the csv file for training

X\_training = csvread('day.csv',1,2,[1,2,548,12]);

%extracting responses from the csv file for training

Y\_training = csvread('day.csv',1,13,[1,13,548,13]);

%quantile(Y,[0,.5,1])

%Y\_new = ordinal(Y\_training,{'2-99' '100-199' '200-299' '300-399' '400-499' '500-599' '600-699' '700-799' '800-899' '900-999' '1000-1099' '1100-1199' '1200-1299' '1300-1399' '1400-1499' '1500-1599' '1600-1699' '1700-1799' '1800-1899' '1900-1999' '2000-2099' '2100-2199' '2200-2299' '2300-2399' '2400-2499' '2500-2599' '2600-2699' '2700-2799' '2800-2899' '2900-2999' '3000-3099' '3100-3199' '3200-3299' '3300-3410'},[],[2 99 199 299 399 499 599 699 799 899 999 1099 1199 1299 1399 1499 1599 1699 1799 1899 1999 2099 2199 2299 2399 2499 2599 2699 2799 2899 2999 3099 3199 3299 3410]);

%discretization into bins

Y\_new = ordinal(Y\_training,{'2-39' '40-59' '60-79' '80-99' '100-199' '200-299' '300-399' '400-499' '500-599' '600-699' '700-799' '800-899' '900-999' '1000-1099' '1100-1199' '1200-1299' '1300-1399' '1400-1499' '1500-1599' '1600-1699' '1700-1799' '1800-1899' '1900-1999' '2000-2299' '2300-2999' '3000-3200'},[],[2 39 59 79 99 199 299 399 499 599 699 799 899 999 1099 1199 1299 1399 1499 1599 1699 1799 1899 1999 2299 2999 3200]);

%disp(Y\_new);

%disp(length(Y\_new));

%finding coefficients and p-values

[B,dev,stats]=mnrfit(X\_training,Y\_new,'model','ordinal');

P\_val=stats.p;

disp('P\_val'),disp(P\_val);

disp('Coefficients'),disp(B);

% % disp('length B'),disp(length(B));

disp('proportional odds'),disp([B(1:33)';repmat(B(34:end),1,33)]);

disp('stats.p'),disp(stats.p);

%extracting predictors from the csv file for testing

X\_test = csvread('day.csv',549,2,[549,2,731,12]);

%extracting responses from the csv file for testing

Y\_test = csvread('day.csv',549,13,[549,13,731,13]);

%Y\_test = csvread('day.csv',549,13,[549,13,549,13]);

count\_correct=0;

[pihat,dlow,hi]=mnrval(B,X\_test,stats,'model','ordinal');

%[pihat,dlow,hi]=mnrval(B,X\_test,stats,'model','ordinal','conditional');

%%disp('probablilities'),disp(pihat);

pred\_label=' ';

test\_label=' ';

%ROC\_Score=zeros(183,1);

ROC\_TestLabel={};

plotroc\_TestLabel=zeros(183,26);

%accuracy computed using max of the probabilities

for i = 1:length(pihat)

[row Max\_index]=max(pihat(i,:));

%disp(max(pihat(i,:)));

%disp(size(max(pihat(i,:))));

[X\_max Y\_max]=ind2sub(size(pihat(i,:)),Max\_index);

 if Y\_max==1.0

pred\_label='2-39';

 elseif Y\_max==2.0

  pred\_label='40-59';

 elseif Y\_max==3.0

  pred\_label='60-79';

 elseif Y\_max==4.0

  pred\_label='80-99';

 elseif Y\_max==5.0

  pred\_label='100-199';

 elseif Y\_max==6.0

  pred\_label='200-299';

 elseif Y\_max==7.0

  pred\_label='300-399';

 elseif Y\_max==8.0

  pred\_label='400-499';

 elseif Y\_max==9.0

  pred\_label='500-599';

 elseif Y\_max==10.0

  pred\_label='600-699';

 elseif Y\_max==11.0

  pred\_label='700-799';

 elseif Y\_max==12.0

  pred\_label='800-899';

 elseif Y\_max==13.0

  pred\_label='900-999';

 elseif Y\_max==14.0

  pred\_label='1000-1099';

 elseif Y\_max==15.0

  pred\_label='1100-1199';

 elseif Y\_max==16.0

  pred\_label='1200-1299';

 elseif Y\_max==17.0

  pred\_label='1300-1399';

 elseif Y\_max==18.0

  pred\_label='1400-1499';

 elseif Y\_max==19.0

  pred\_label='1500-1599';

 elseif Y\_max==20.0

  pred\_label='1600-1699';

elseif Y\_max==21.0

  pred\_label='1700-1799';

elseif Y\_max==22.0

  pred\_label='1800-1899';

elseif Y\_max==23.0

  pred\_label='1900-1999';

elseif Y\_max==24.0

  pred\_label='2000-2299';

elseif Y\_max==25.0

  pred\_label='2300-2999';

elseif Y\_max==26.0

  pred\_label='3000-3200';

 end

%   disp('pred\_label'),disp(pred\_label);

%   disp('y\_max'),disp(Y\_max);

%   disp('Y\_test(i)'),disp(Y\_test(i));

 if Y\_test(i)>=2.0 && Y\_test(i)<=39.0

  test\_label='2-39';

  plotroc\_TestLabel(i,1)=1;

 elseif Y\_test(i)>=40.0 && Y\_test(i)<=59.0

  test\_label='40-59';

  plotroc\_TestLabel(i,2)=1;

 elseif Y\_test(i)>=60.0 && Y\_test(i)<=79.0

  test\_label='60-79';

  plotroc\_TestLabel(i,3)=1;

 elseif Y\_test(i)>=80.0 && Y\_test(i)<=99.0

  test\_label='80-99';

  plotroc\_TestLabel(i,4)=1;

 elseif Y\_test(i)>=100.0 && Y\_test(i)<=199.0

  test\_label='100-199';

  plotroc\_TestLabel(i,5)=1;

 elseif Y\_test(i)>=200.0 && Y\_test(i)<=299.0

  test\_label='200-299';

  plotroc\_TestLabel(i,6)=1;

 elseif Y\_test(i)>=300.0 && Y\_test(i)<=399.0

  test\_label='300-399';

  plotroc\_TestLabel(i,7)=1;

 elseif Y\_test(i)>=400.0 && Y\_test(i)<=499.0

  test\_label='400-499';

  plotroc\_TestLabel(i,8)=1;

 elseif Y\_test(i)>=500.0 && Y\_test(i)<=599.0

  test\_label='500-599';

  plotroc\_TestLabel(i,9)=1;

 elseif Y\_test(i)>=600.0 && Y\_test(i)<=699.0

  test\_label='600-699';

  plotroc\_TestLabel(i,10)=1;

 elseif Y\_test(i)>=700.0 && Y\_test(i)<=799.0

  test\_label='700-799';

  plotroc\_TestLabel(i,11)=1;

  elseif Y\_test(i)>=800.0 && Y\_test(i)<=899.0

  test\_label='800-899';

  plotroc\_TestLabel(i,12)=1;

 elseif Y\_test(i)>=900.0 && Y\_test(i)<=999.0

  test\_label='900-999';

  plotroc\_TestLabel(i,13)=1;

 elseif Y\_test(i)>=1000.0 && Y\_test(i)<=1099.0

  test\_label='1000-1099';

  plotroc\_TestLabel(i,14)=1;

 elseif Y\_test(i)>=1100.0 && Y\_test(i)<=1199.0

  test\_label='1100-1199';

  plotroc\_TestLabel(i,15)=1;

 elseif Y\_test(i)>=1200.0 && Y\_test(i)<=1299.0

test\_label='1200-1299';

plotroc\_TestLabel(i,16)=1;

 elseif Y\_test(i)>=1300.0 && Y\_test(i)<=1399.0

  test\_label='1300-1399';

  plotroc\_TestLabel(i,17)=1;

 elseif Y\_test(i)>=1400.0 && Y\_test(i)<=1499.0

  test\_label='1400-1499';

  plotroc\_TestLabel(i,18)=1;

 elseif Y\_test(i)>=1500.0 && Y\_test(i)<=1599.0

  test\_label='1500-1599';

  plotroc\_TestLabel(i,19)=1;

 elseif Y\_test(i)>=1600.0 && Y\_test(i)<=1699.0

  test\_label='1600-1699';

  plotroc\_TestLabel(i,20)=1;

 elseif Y\_test(i)>=1700.0 && Y\_test(i)<=1799.0

  test\_label='1700-1799';

  plotroc\_TestLabel(i,21)=1;

 elseif Y\_test(i)>=1800.0 && Y\_test(i)<=1899.0

  test\_label='1800-1899';

  plotroc\_TestLabel(i,22)=1;

elseif Y\_test(i)>=1900.0 && Y\_test(i)<=1999.0

  test\_label='1900-1999';

  plotroc\_TestLabel(i,23)=1;

elseif Y\_test(i)>=2000.0 && Y\_test(i)<=2299.0

  test\_label='2000-2299';

  plotroc\_TestLabel(i,24)=1;

elseif Y\_test(i)>=2300.0 && Y\_test(i)<=2999.0

  test\_label='2300-2999';

  plotroc\_TestLabel(i,25)=1;

elseif Y\_test(i)>=3000.0 && Y\_test(i)<=3200.0

  test\_label='3000-3200';

  plotroc\_TestLabel(i,26)=1;

end

%   disp('test label'),disp(test\_label);

%disp('ROC\_Score'),disp(ROC\_Score);

 if strcmp(pred\_label,test\_label)==1

    count\_correct=count\_correct+1;

 end

%disp(Y\_max)

%disp(X\_max)

 ROC\_TestLabel{i}=test\_label;

end

% disp('ROC\_TestLabel'),disp(ROC\_TestLabel);

% disp('count\_correct'),disp(count\_correct);

%disp('predictions'),disp(pihat);

% disp('length pihat'),disp(length(pihat));

%accuracy computed using max of the probabilities

Accuracy=(count\_correct/length(Y\_test))\*100;

% disp('Accuracy'),disp(Accuracy)

%confidence bounds for the category

LL=pihat-dlow;

UL=pihat+hi;

%disp('confidence bounds for the category'),disp([LL;UL]);

%generating the ROC curves

plotroc(transpose(plotroc\_TestLabel),transpose(pihat))

[X1,Y1,T1,AUC1]=perfcurve(ROC\_TestLabel,pihat(:,1),'2-39');

% plot(X1,Y1)

%  hold on

[X4,Y4,T4,AUC4]=perfcurve(ROC\_TestLabel,pihat(:,4),'80-99');

% plot(X4,Y4)

[X5,Y5,T5,AUC5]=perfcurve(ROC\_TestLabel,pihat(:,5),'100-199');

% plot(X5,Y5)

[X6,Y6,T6,AUC6]=perfcurve(ROC\_TestLabel,pihat(:,6),'200-299');

% plot(X6,Y6)

[X7,Y7,T7,AUC7]=perfcurve(ROC\_TestLabel,pihat(:,7),'300-399');

% plot(X7,Y7)

[X8,Y8,T8,AUC8]=perfcurve(ROC\_TestLabel,pihat(:,8),'400-499');

% plot(X8,Y8)

[X9,Y9,T9,AUC9]=perfcurve(ROC\_TestLabel,pihat(:,9),'500-599');

% plot(X9,Y9)

[X10,Y10,T10,AUC10]=perfcurve(ROC\_TestLabel,pihat(:,10),'600-699');

% plot(X10,Y10)

[X11,Y11,T11,AUC11]=perfcurve(ROC\_TestLabel,pihat(:,11),'700-799');

% plot(X11,Y11)

[X12,Y12,T12,AUC12]=perfcurve(ROC\_TestLabel,pihat(:,12),'800-899');

% plot(X12,Y12)

[X13,Y13,T13,AUC13]=perfcurve(ROC\_TestLabel,pihat(:,13),'900-999');

% plot(X13,Y13)

[X14,Y14,T14,AUC14]=perfcurve(ROC\_TestLabel,pihat(:,14),'1000-1099');

% plot(X14,Y14)

[X15,Y15,T15,AUC15]=perfcurve(ROC\_TestLabel,pihat(:,15),'1100-1199');

% plot(X15,Y15)

[X16,Y16,T16,AUC16]=perfcurve(ROC\_TestLabel,pihat(:,16),'1200-1299');

% plot(X16,Y16)

[X17,Y17,T17,AUC17]=perfcurve(ROC\_TestLabel,pihat(:,17),'1300-1399');

% plot(X17,Y17)

[X18,Y18,T18,AUC18]=perfcurve(ROC\_TestLabel,pihat(:,18),'1400-1499');

% plot(X18,Y18)

[X19,Y19,T19,AUC19]=perfcurve(ROC\_TestLabel,pihat(:,19),'1500-1599');

% plot(X19,Y19)

 [X20,Y20,T20,AUC20]=perfcurve(ROC\_TestLabel,pihat(:,20),'1600-1699');

%  plot(X20,Y20)

[X21,Y21,T21,AUC21]=perfcurve(ROC\_TestLabel,pihat(:,21),'1700-1799');

% plot(X21,Y21)

[X22,Y22,T22,AUC22]=perfcurve(ROC\_TestLabel,pihat(:,22),'1800-1899');

% plot(X22,Y22)

[X23,Y23,T23,AUC23]=perfcurve(ROC\_TestLabel,pihat(:,23),'1900-1999');

% plot(X23,Y23)

[X24,Y24,T24,AUC24]=perfcurve(ROC\_TestLabel,pihat(:,24),'2000-2299');

% plot(X24,Y24)

[X25,Y25,T25,AUC25]=perfcurve(ROC\_TestLabel,pihat(:,25),'2300-2999');

% plot(X25,Y25)

[X26,Y26,T26,AUC26]=perfcurve(ROC\_TestLabel,pihat(:,26),'3000-3200');

% plot(X26,Y26)

%  hold off

% legend('2-39','1600-1699','Location','Best')

% legend('2-39','80-99','100-199','200-299','300-399','400-499','500-599','600-699','700-799','800-899','900-999','1000-1099','1100-1199','1200-1299','1300-1399','1400-1499','1500-1599','1600-1699','1700-1799','1800-1899','1900-1999','2000-2299','2300-2999','3000-3200','Location','Best')

% xlabel('False Positive Rate');

% ylabel('True Positive Rate');

% title('ROC for Classification by Multinomial Logistic Regression')

%

% disp('AUC1'),disp(AUC1);

% disp('AUC4'),disp(AUC4);

% disp('AUC5'),disp(AUC5);

% disp('AUC6'),disp(AUC6);

% disp('AUC7'),disp(AUC7);

% disp('AUC8'),disp(AUC8);

% disp('AUC9'),disp(AUC9);

% disp('AUC10'),disp(AUC10);

% disp('AUC11'),disp(AUC11);

% disp('AUC12'),disp(AUC12);

% disp('AUC13'),disp(AUC13);

% disp('AUC14'),disp(AUC14);

% disp('AUC15'),disp(AUC15);

% disp('AUC16'),disp(AUC16);

% disp('AUC17'),disp(AUC17);

% disp('AUC18'),disp(AUC18);

% disp('AUC19'),disp(AUC19);

%  disp('AUC20'),disp(AUC20);

% disp('AUC21'),disp(AUC21);

% disp('AUC22'),disp(AUC22);

% disp('AUC23'),disp(AUC23);

% disp('AUC24'),disp(AUC24);

% disp('AUC25'),disp(AUC25);

% disp('AUC26'),disp(AUC26);

ROC\_AUC=[AUC1 AUC4 AUC5 AUC6 AUC7 AUC8 AUC9 AUC10 AUC11 AUC12 AUC13 AUC14 AUC15 AUC16 AUC17 AUC18 AUC19 AUC20 AUC21 AUC22 AUC23 AUC24 AUC25 AUC26];

% % disp('SORTED ROC\_AUC'),disp(sort(ROC\_AUC));

% bar\_Xlabel=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24];

% BAR\_disp=bar(ROC\_AUC,0.4,'Facecolor',[0 0.5 0.5]);

% ylabel('AUC');

% title('area under the curve')

%generating the bar chart for comparison of the AUC

fHand=figure;

aHand=axes('parent',fHand);

hold(aHand,'on');

bar(1,ROC\_AUC(1),'parent',aHand,'Facecolor','g');

bar(2,ROC\_AUC(2),'parent',aHand,'Facecolor','c');

bar(5,ROC\_AUC(3),'parent',aHand,'Facecolor','c');

bar(7,ROC\_AUC(5),'parent',aHand,'Facecolor','c');

bar(4,ROC\_AUC(20),'parent',aHand,'Facecolor','c');

bar(3,ROC\_AUC(23),'parent',aHand,'Facecolor','c');

bar(6,ROC\_AUC(24),'parent',aHand,'Facecolor','c');

bar(8,ROC\_AUC(22),'parent',aHand,'Facecolor','y');

bar(9,ROC\_AUC(6),'parent',aHand,'Facecolor','y');

bar(10,ROC\_AUC(4),'parent',aHand,'Facecolor','y');

bar(11,ROC\_AUC(15),'parent',aHand,'Facecolor','y');

bar(12,ROC\_AUC(7),'parent',aHand,'Facecolor','y');

bar(13,ROC\_AUC(10),'parent',aHand,'Facecolor','y');

bar(14,ROC\_AUC(17),'parent',aHand,'Facecolor','y');

bar(15,ROC\_AUC(11),'parent',aHand,'Facecolor','y');

bar(16,ROC\_AUC(21),'parent',aHand,'Facecolor','y');

bar(17,ROC\_AUC(9),'parent',aHand,'Facecolor','y');

bar(18,ROC\_AUC(13),'parent',aHand,'Facecolor','y');

bar(19,ROC\_AUC(12),'parent',aHand,'Facecolor','y');

bar(20,ROC\_AUC(14),'parent',aHand,'Facecolor','y');

bar(21,ROC\_AUC(8),'parent',aHand,'Facecolor','y');

bar(22,ROC\_AUC(16),'parent',aHand,'Facecolor','y');

bar(23,ROC\_AUC(19),'parent',aHand,'Facecolor','y');

bar(24,ROC\_AUC(18),'parent',aHand,'Facecolor','r');

ylabel('AUC');

title('Barchart for Comparison of AUC')