DSC 424: Advanced Data Analysis
FINAL REPORT: CYLINDER BANDS IN ROTOGRAVURE PRINTING
Group 5: Theophile Dushime, Swathi Babu, Avyay Suri

INTRODUCTION

The rotogravure process is a direct transfer method for printing onto wood-pulp fiber based, synthetic, or laminated substrates. (1) Unfortunately, grooves or bands sometimes develop on the cylinder during the printing process and appear on the printed pages. The print run must then be halted and in some cases the cylinder replaced, at a substantial cost. The reasons for banding are largely unknown, and experts cannot reliably predict when it will occur. (2)

To predict that to some extent, the dataset "Cylinder Bands" with 540 observations and 41 variables was used. Three different models were used to examine these causes. Regularized regression, Linear Discriminant analysis and Decision trees were used.

EXPLORATORY DATA ANALYSIS

Deleting rows/variables timestamp as it doesn't help with the analysis, cylinderNo and jobNo as it had too many classes in its category, and they also just seemed like IDs rather than information that can be used for any analysis.

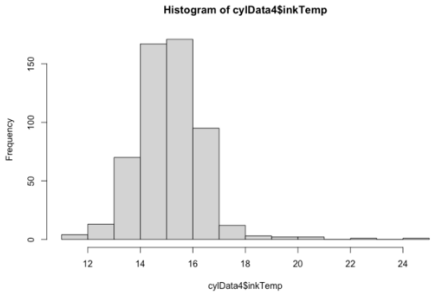
All the column values are in their respective units but range of some variables seem to be higher than others like press speed varies from 0 to 1000 and caliper varies from 0 to 1.0 and this could bring up scaling issues.

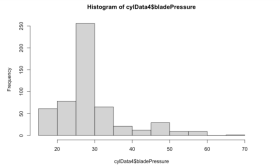
R did not read all of the variables with their respective datatype. It read everything as a char. So, converting it manually to either as a factor or a numeric variable, we get the proper data frame.

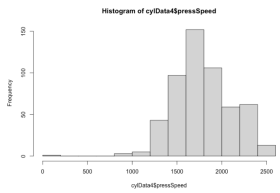
Observing the results after using summary statement, we could see that there were some columns which had values that don't belong. For e.g., ctdInk takes either YES or NO but it has a value 17. Going back to the dataset, we see that one row where elements are misplaced(row 574). We also saw that some columns have same values in different cases. So, we correct that as well by changing them all into uppercase.After doing that, we could see some variables do not vary at all. So, we remove those variables -- inkColor(all the values are KEY, there is no TYPE) and cylDiv (only GALLATIN and no WARSAW or MATTOON)

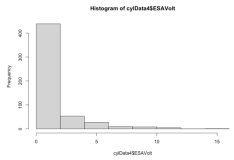
Looking at the histograms, almost all variables are skewed .Variable inkTemp might have some outliers. Variable bladePressure has some outliers around 70. Variable pressSpeed seems to have some outliers. Variable ESAVolt might have some outliers around 15. Variable ESAAmperage seems to have 2 outliers with values between 3-4 and 5-6. Variable rollerDurometer also seemed to have an outlier between 55 and 60.Variable currentDensity might have some outliers. Variable chromeContent also seems to have outliers at 90-92 and 94-96. It is important to note that since most of the variables are skewed (although normally distributed), we did try log transformation for the models mentioned further below and that did not help with the model.

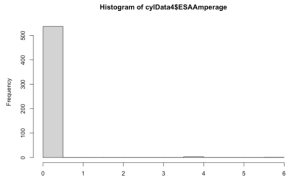
Since we do know that all these variables are necessary and that we do not have many observations, we are not removing these values and we are going to keep them for now. We will delete them if they seem to cause some problems later

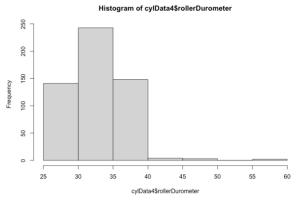


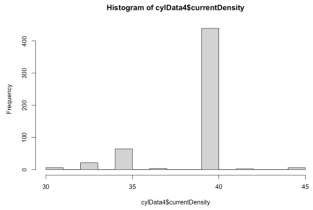


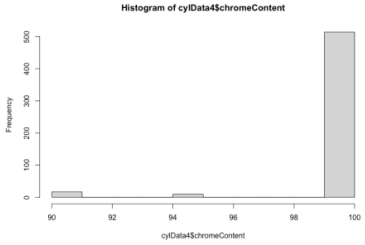






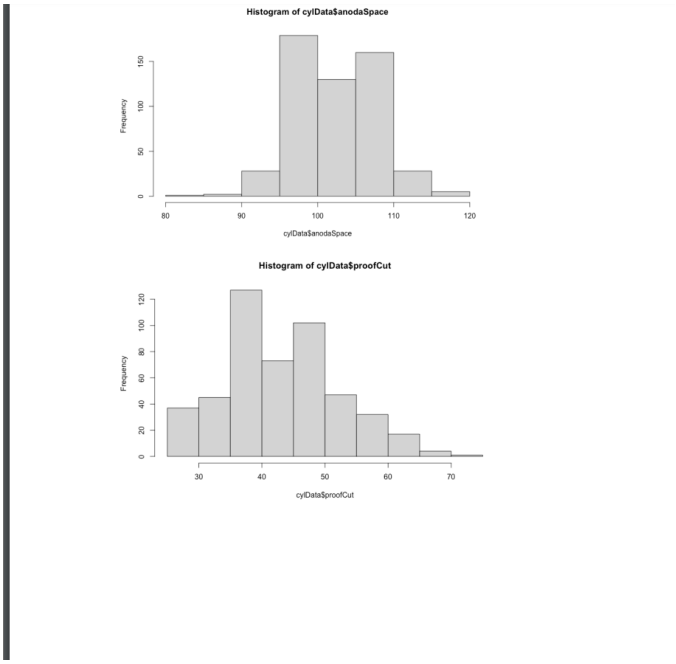


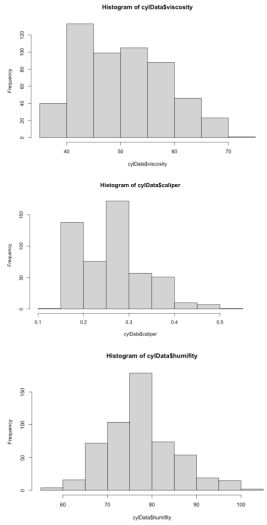


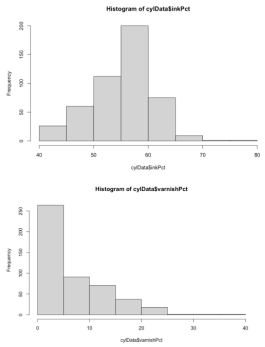


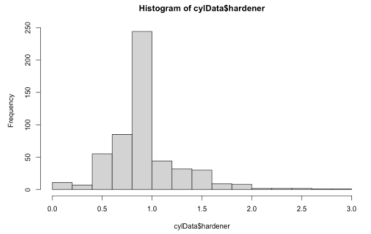
From the other histograms, we saw that some of the variables are bimodal like anodaSpace and caliper and others are unimodal like hardener, wax, solventPct, inkPct, varnishPct, humidity, viscosity, proofCut,

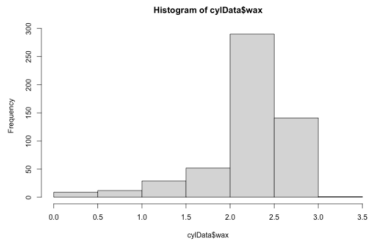
chromeContent, currentDensity,rollerDurometer, ESAAmperage, ESAVolt, pressSpeed, bladePressure and inkTemp.

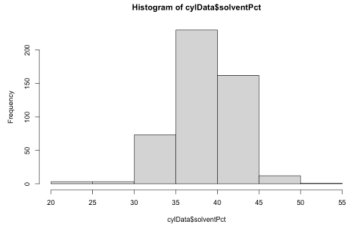


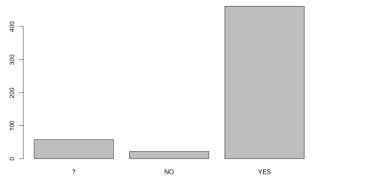




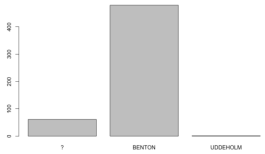




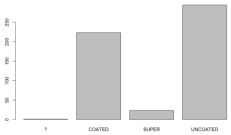




(ctdInk)

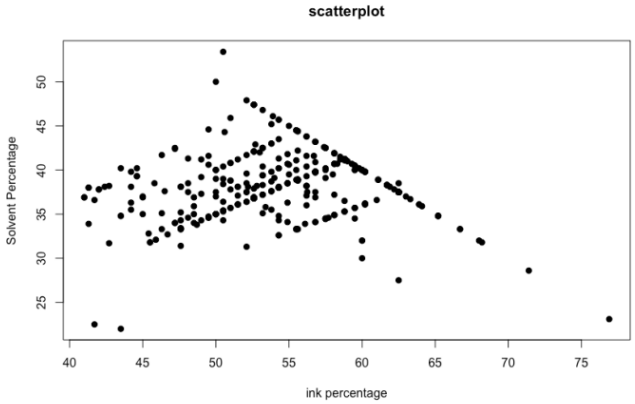


(bladeMfg)

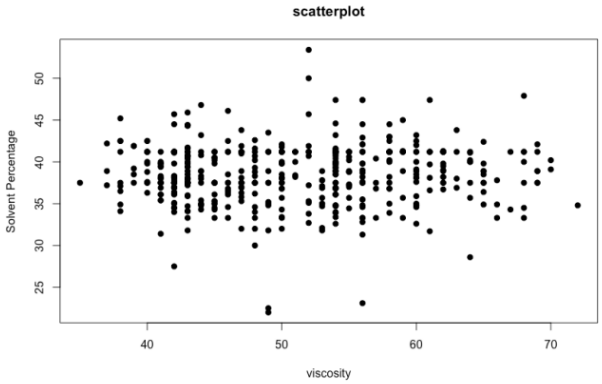


(paperType)

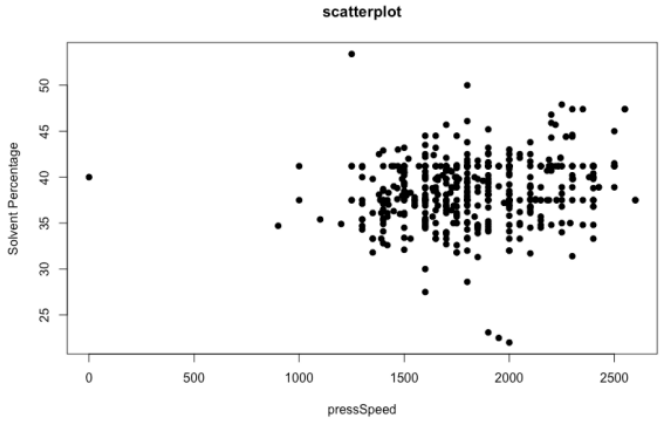
We used barplots to look at the distribution of categorical variables. As we can see, most of the variables are imbalanced especially variables like ctdInk, bladeMfg, paperType(low number of rows with value “super”), inkType(low number of rows with value “cover”), directStream, solventType, cylType,unitNo, cylSize, millLoc.



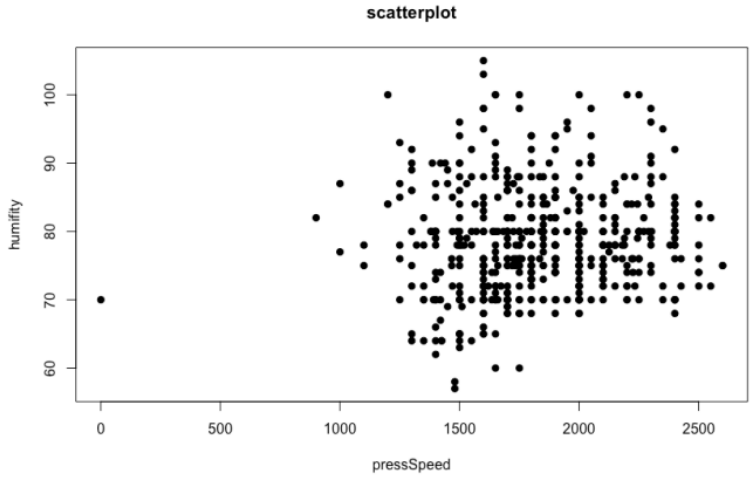
We can see that as ink percentage increases, solvent percentage also increases until 50%, then solvent percentage decreases as ink percentage increases.



There isn't any clear relationship between ink percentage and solvent percentage which is interesting as viscosity's definition was related to how much solvent is lost.



There is a slight linear relationship between pressSpeed and Solvent percentage.



There is a very vague linear relationship here but not much.

We saw that most of the variables don't have much of linear relationship which is confirmed further when we look at the correlation table for this data.

	proofCut	viscosity	caliper	inkTemp	humifity	roughness	bladePressure	varnishPct	pressSpeed	inkPct	solventPct	ESAVolt	ESAAmperage
proofCut	1.000000000	0.024952624	0.0733856615	-0.005335322	0.0068365493	-0.065619074	0.0943229550	0.361091175	-0.21891760	-0.331997548	-0.17325996	-0.100266922	0.0285262028
viscosity	0.024952624	1.000000000	0.1896751009	-0.060472611	0.2593958189	-0.002263445	0.0910220387	-0.002401596	0.02007017	-0.037040615	0.03051285	-0.014170964	0.0019583045
caliper	0.073385662	0.189675101	1.0000000000	-0.074649866	0.1005465884	0.062998459	0.0207726103	0.072613916	-0.07587464	-0.061462118	-0.06069763	0.042940306	0.1165378851
inkTemp	-0.005335322	-0.060472611	-0.0746498662	1.000000000	0.0555272652	-0.049915493	-0.0786818290	-0.019384966	0.06029164	-0.009962989	0.03118096	0.132644360	-0.0243036611
humifity	0.006836549	0.259395819	0.1005465884	0.055527265	1.0000000000	0.011659911	0.0123102857	-0.065583547	0.07690515	0.008749987	0.07623300	0.025051164	0.0237096071
roughness	-0.065619074	-0.002263445	0.0629984588	-0.049915493	0.0116599109	1.000000000	-0.0398099728	-0.006973624	0.03142815	0.047394772	-0.05419412	-0.005948418	0.0988387392
bladePressure	0.094322955	0.091022039	0.0207726103	-0.078681829	0.0123102857	-0.039809973	1.0000000000	-0.043274172	-0.18679186	0.016948150	0.06383588	-0.059443883	-0.0009177862
varnishPct	0.361091175	-0.002401596	0.0726139159	-0.019384966	-0.0655835475	-0.006973624	-0.0432741724	1.000000000	-0.16207147	-0.863191166	-0.53940492	-0.064631425	0.0686187347
pressSpeed	-0.218917604	0.020070173	-0.0758746363	0.060291637	0.0769051535	0.031428151	-0.1867918577	-0.162071468	1.00000000	0.095141930	0.15872142	0.228731964	-0.0756425464
inkPct	-0.331997548	-0.037040615	-0.0614621185	-0.009962989	0.0087499865	0.047394772	0.0169481504	-0.863191166	0.09514193	1.000000000	0.09106564	0.038960496	-0.0697132554
solventPct	-0.173259963	0.030512850	-0.0606976339	0.031180959	0.0762330009	-0.054194123	0.0638358822	-0.539404923	0.15872142	0.091065642	1.00000000	0.080251414	-0.0191415869
ESAVolt	-0.100266922	-0.014170964	0.0429403062	0.132644360	0.0250511641	-0.005948418	-0.0594438835	-0.064631425	0.22873196	0.038960496	0.08025141	1.000000000	-0.0434133089
ESAAmperage	0.028526203	0.001958304	0.1165378851	-0.024303661	0.0237096071	0.098838739	-0.0009177862	0.068618735	-0.07564255	-0.069713255	-0.01914159	-0.043413309	1.0000000000
wax	0.161868853	0.087982986	0.0554061346	-0.016772192	0.1114103111	-0.063773834	0.2276903056	0.166882783	-0.18767231	-0.152516592	-0.06446872	-0.060029203	0.0059778657
hardener	-0.047541147	-0.050176207	-0.0101398964	0.036057381	-0.0378446079	0.069253777	-0.0171943649	-0.137000822	-0.06051212	0.114965507	0.06098422	-0.031344826	-0.0053371035
rollerDurometer	0.344944520	-0.045877237	0.0838803475	-0.055522462	-0.0006023099	0.046581176	0.1718399434	0.252936686	-0.43343637	-0.242378105	-0.10772726	-0.126686407	0.1081070105
currentDensity	-0.059588045	-0.080647339	-0.1584666277	0.072038369	-0.1290320583	-0.040624098	-0.0427517360	-0.089622836	0.03099316	0.154069696	-0.07357446	-0.061629874	-0.0661291832
anodaSpace	-0.034264133	-0.054017924	-0.0002227224	0.031656947	-0.0800238930	-0.007298363	-0.0077978611	0.023066349	-0.00189459	-0.053570129	0.02384543	-0.070861308	-0.0723100017
chromeContent	-0.013735545	0.017327825	0.0180100185	-0.026186666	0.0103906303	0.003454255	0.0418778188	-0.012319480	-0.05478099	-0.105100515	0.20164297	0.059821210	-0.0823393031

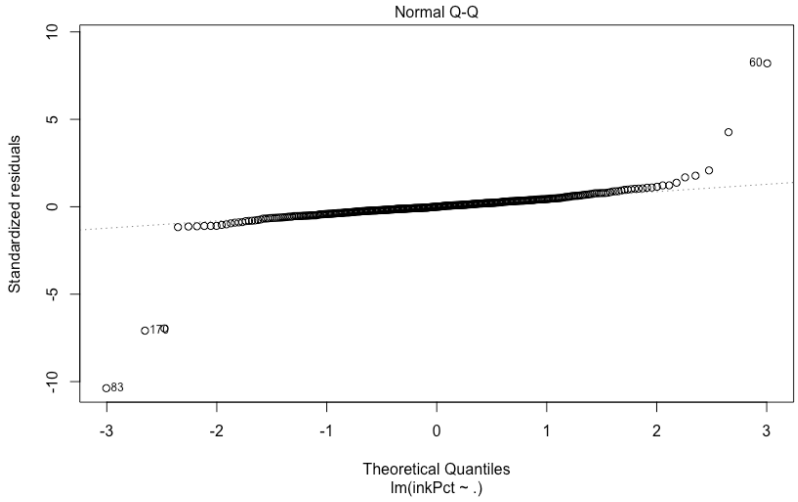
ANALYSIS TECHNIQUES

MODEL 1: Regularized Regression

We used OLS regression to predict the variable *inkPct* – ink percentage.

```
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)      1.033e+02  5.176e+00  19.951   <2e-16 ***
proofCut        -5.471e-03  8.644e-03  -0.633   0.5272
viscosity       -1.362e-02  8.666e-03  -1.571   0.1171
caliper         -9.417e-01  9.651e-01  -0.976   0.3299
inkTemp         -5.826e-02  5.317e-02  -1.096   0.2740
humifity        -1.207e-02  9.279e-03  -1.301   0.1942
roughness       -1.233e-03  4.148e-01  -0.003   0.9976
bladePressure    8.300e-03  1.400e-02   0.593   0.5537
varnishPct      -9.349e-01  1.365e-02 -68.485   <2e-16 ***
pressSpeed      -6.656e-05  2.993e-04  -0.222   0.8241
solventPct      -8.707e-01  2.388e-02 -36.460   <2e-16 ***
ESAVolt          4.555e-02  3.138e-02   1.452   0.1476
ESAAmperage      4.151e-02  1.560e-01   0.266   0.7904
wax              1.272e-02  1.572e-01   0.081   0.9355
hardener        -6.376e-02  1.968e-01  -0.324   0.7462
rollerDurometer -3.132e-02  2.140e-02  -1.463   0.1443
currentDensity   4.152e-02  2.892e-02   1.436   0.1520
anodaSpace      -4.888e-03  1.333e-02  -0.367   0.7142
chromeContent   -5.236e-02  3.842e-02  -1.363   0.1738
dgrainScreened  -3.656e-03  2.643e-01  -0.014   0.9890
dctdInk          8.462e-01  3.434e-01   2.464   0.0142 *
dbladeMfg       -1.303e-02  1.219e+00  -0.011   0.9915
dpaperCoated     1.262e-01  2.316e-01   0.545   0.5862
dpaperSuper     -5.717e-01  4.679e-01  -1.222   0.2227
dinkCoated      -3.690e-02  2.543e-01  -0.145   0.8847
dinkCover        3.579e-01  3.586e-01   0.998   0.3190
ddirectStream    3.502e-01  8.891e-01   0.394   0.6939
dsolventLine    -2.699e-01  3.735e-01  -0.723   0.4703
dsolventNaphta          NA         NA      NA       NA
dcylType        -2.583e-01  1.541e-01  -1.676   0.0947 .
dpressAlbert70  -7.603e-01  4.848e-01  -1.568   0.1178
dpressMotter70  -2.774e-01  1.306e+00  -0.212   0.8319
dpressAMotter94 -1.234e+00  1.239e+00  -0.996   0.3200
dpress802       -4.225e-02  1.376e+00  -0.031   0.9755
dpress813               NA         NA      NA       NA
dpress815       -9.464e-01  1.280e+00  -0.740   0.4601
dpress816       -8.765e-01  1.254e+00  -0.699   0.4849
dpress816       -8.765e-01  1.254e+00  -0.699   0.4849
dpress821        3.051e-01  4.017e-01   0.760   0.4480
dpress824       -1.963e-01  3.382e-01  -0.581   0.5619
dpress827        3.051e-01  3.440e-01   0.887   0.3758
dsizeCatalog    -1.804e-01  2.603e-01  -0.693   0.4887
dsizeSPIEGEL    -1.399e-01  2.451e-01  -0.571   0.5686
dcanadian       -1.862e-01  3.772e-01  -0.494   0.6219
dmidEuropean    -1.199e+00  5.113e-01  -2.346   0.0196 *
dnorthUs        -3.024e-01  3.574e-01  -0.846   0.3980
dscandinavian           NA         NA      NA       NA
dplatIngTank            NA         NA      NA       NA
unit2            3.102e-01  9.535e-01   0.325   0.7451
unit9            2.986e-01  9.419e-01   0.317   0.7514
unit7            1.362e-01  9.872e-01   0.138   0.8903
unit1            6.887e-01  9.346e-01   0.737   0.4617
unit5           -2.766e-01  9.875e-01  -0.280   0.7795
unit10           2.103e-01  1.041e+00   0.202   0.8401
dbandType       -1.009e-01  1.571e-01  -0.642   0.5212
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.17 on 328 degrees of freedom
Multiple R-squared:  0.9574,    Adjusted R-squared:  0.951
F-statistic: 150.3 on 49 and 328 DF,  p-value: < 2.2e-16

>
> rmseOlsTrain3 = sqrt(mean(olsfit3$residuals^2))
> rmseOlsTrain3
[1] 1.08948


  > rmseOlsTest3
  [1] 7.428988
```

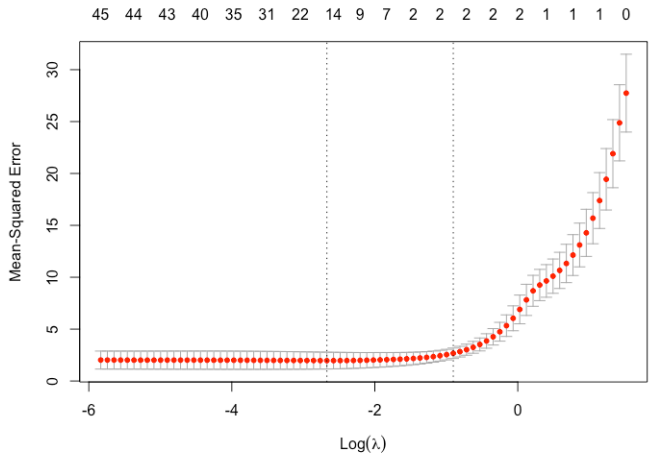
This model definitely has a R^2 and Adj-R^2 value of 0.9574 and 0.951 respectively. This could be indicative of overfitting. We would know as we look further. Especially the RMSE values, although are good individually, relatively there is a huge difference between them (1.08948 and 7.428988). This also indicates the overfitting problem. Furthermore, the model as a whole is statistically significant but all the variables in the model aren't. We might have to look at regularized regression for this model. There is no multicollinearity and looks like the normality assumption is also satisfied (apart from some outlier like points).



We tried Lasso and Elastic Net regression to solve the overfitting issue. We also chose these methods as they perform model selection as well.

Model Validation:
The data was randomly split into Training set and Test set and 7- fold cross validation was used.
The model was built using the training set and predictions are performed on the testing set to evaluate the model's performance on data it has not trained on.

Results:
Lasso- The Lasso results were pretty good. Using lambda.min (0.06081) ,the adjusted R^2 value was 94.3% and the analysis picked 17 independent variables.



Coefficients:

	s1
(Intercept)	**9.084655e+01**
proofCut	**-5.745540e-04**
viscosity	**-3.078243e-03**
humifity	**-8.811084e-03**
varnishPct	**-8.845308e-01**

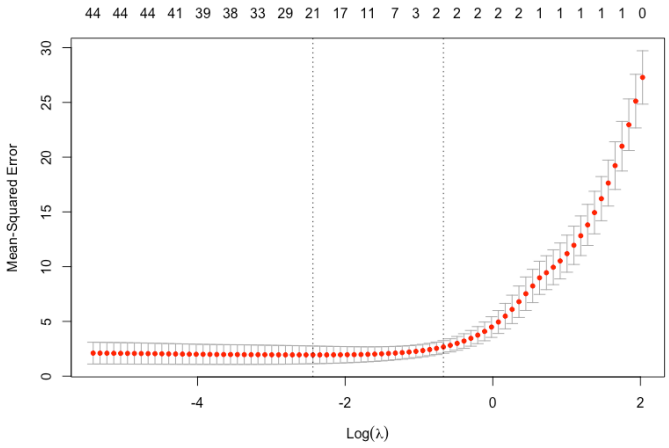
solventPct -7.587415e-01
rollerDurometer -7.076709e-03
currentDensity 2.635751e-02
anodaSpace -5.137868e-05
chromeContent -8.882454e-03
dpaperSuper -5.465738e-01
dsolventLine 4.126485e-03
dpressMotter70 6.225333e-02
dpress813 1.771637e-03
dpress827 3.399415e-02
dcanadian 1.459966e-01
dmidEuropean -3.695947e-02
unit1 4.547378e-01

The regression equation is $inkPct = 90.8465 - 0.00057proofCut - 0.00308viscosity - 0.00881humidity - 0.88453varnishPct - 0.758741solventPct - 0.00707rollerDurometer + 0.026357currentDensity - 0.000051anodaSpace - 0.008882chromeContent - 0.546573dpaperSuper + 0.004126dsolventLine + 0.062253dpressMotter70 + 0.001771dpress813 + 0.0339941dpress827 + 0.145997dcanadian - 0.036959dmidEuropean + 0.454737unit1$

The RMSE value for Lasso was 1.232354% which is way closer to 1.08948% than OLS's RMSE value of 7.494621%. This resolved the issue of overfitting as well with only 0.6% loss in R^2 which we think is an acceptable amount of loss for resolving the overfitting issue.

```
> rmseLasso   # Lasso test RMSE
[1] 1.225859

Call:  glmnet(x = xTrain, y = yTrain, alpha = 1, lambda = 0.06319358)

  Df %Dev  Lambda
1 21 94.3 0.06319
```

<u>Elastic Net-</u> The Elastic Net results were also good. Using lambda.min(0.0874407) ,the adjusted R^2 value was 94.36% and the analysis picked 21 independent variables.



Coefficients:

s1

(Intercept) 89.268001980
proofCut -0.002475735
viscosity -0.004044678
inkTemp -0.008015858
humifity -0.009476605
varnishPct -0.878654111
pressSpeed -0.000196399
solventPct -0.736618644
ESAAmperage -0.209057240
wax 0.065054905
rollerDurometer -0.009513926
currentDensity 0.052080522
anodaSpace -0.003835897
dpaperSuper -0.600638379
dpressMotter70 0.230467173
dpress813 0.098587955
dpress816 -0.115749889
dpress824 -0.089598236
dmidEuropean -0.838202967
dnorthUs -0.199970972
unit7 -0.150303733
unit1 0.158065424

The regression equation is $inkPct = 89.268 - 0.0025proofCut - 0.0040viscosity - 0.0080inkTemp - 0.0095humidity - 0.8786varnishPct - 0.0002pressSpeed - 0.7366solventPct - 0.2091ESAAmperage + 0.06505wax - 0.0095rollerDurometer + 0.05208currentDensity - 0.00383anodaSpace - 0.60063dpaperSuper + 0.23046dpressMotter70 + 0.0985dpress813 - 0.1157dpress816 - 0.0895dpress824 - 0.8382dmidEuropean - 0.1999dnorthUS - 0.1503unit7 + 0.1580unit1$

The RMSE value for Elastic Net was 1.250883 % which is also way closer to 1.08948% than OLS's RMSE value of 7.494621%. This also resolved the issue of overfitting with only 0.6% loss in R^2 which we think is an acceptable amount of loss for resolving the overfitting issue.

```
> rmseNet   # Lasso test RMSE
[1] 1.250883
```

```
Call:  glmnet(x = xTrain, y = yTrain, alpha = 0.6, lambda = 0.0874407)

  Df  %Dev  Lambda
1 21 94.36 0.08744
```

Analysis of the results:
Both the models seem to be working really well with capturing the variance of the dataset, with their individual RMSE values and w.r.t to each other (between testing and training). Overfitting seems to be resolved in both the methods. The only problem is Lasso seems to have excluded some variables that could be important for the analysis of inkPct like inkTemp etc… which were included in the Elastic Net. So, we choose Elastic Net model for this purpose. Now, this model can be used to predict the ink percentage to be used and hence a lot of resources can be used efficiently (like using exact amount of ink could also help with using less paper etc…)

MODEL 2: Linear Discriminant Analysis

The objective of this analysis was to build a Linear Discriminant Analysis model that can use different variables of our dataset to classify the data into two categories of the target variable dbandType which indicate the occurrence of BANDS (dbandType=1) or NOBAND (dbandType=0).
From Milestone2 we had 64 variables, of which 51 were numeric. All categorical variables were dropped since they have equivalent dummy variables. Variables press, dpressMotter70, dpress802, and dplatIngTank that are almost collinear with other variables were removed. Because LDA involves computation of matrix inverse, variables that are linear combinations of each other would affect matrix invertibility.

Model validation

The data was randomly split into Training set and Test set, with 80% of the data being assigned to the training and the rest of the data to testing.
The model was built using the training set and predictions are performed on the testing set to evaluate the model's performance on data it has not trained on.

Results:

Loadings:

The following are the original independent variables sorted by their loadings to indicate how important they are in influencing the classification of the data.

```
> print(fitlda$scaling[order(fitlda$scaling[, 1]), ])
      dcanadian    dmidEuropean   dscandanavian         caliper
   -2.407450334    -1.899825671    -1.872029716    -1.752489303
       dnorthUs  dsolventNaphta    dsizeCatalog             wax
   -1.646973189    -0.793262180    -0.718490946    -0.479163966
     dinkCoated        dcylType      varnishPct         ESAVolt
   -0.433133322    -0.370813611    -0.097496904    -0.073587535
  chromeContent      solventPct          inkPct          unitNo
   -0.073210777    -0.069708414    -0.037457272    -0.016712604
     pressSpeed  dgrainScreened      anodaSpace        humifity
   -0.001358380     0.005630830     0.006600666     0.008167215
       proofCut       viscosity rollerDurometer   bladePressure
    0.011411102     0.038048907     0.044764444     0.066407848
 currentDensity         inkTemp        hardener     platingTank
    0.072371498     0.089178973     0.135817090     0.160054211
   dsolventLine       dpress824       dpress827     ESAAmperage
    0.200070866     0.201464656     0.356421774     0.416551551
     roughness    dsizeSPIEGEL    dpaperCoated  dpressAlbert70
    0.446908344     0.453899981     0.491381051     0.537713977
     dpress813         dctdInk       dinkCover     dpaperSuper
    0.831391757     0.865182152     0.969915032     1.052931796
     dpress821   ddirectStream dpressAMotter94       dbladeMfg
    1.165909195     1.239548297     1.480326256     2.057369801
     dpress816       dpress815
    2.643018120     2.918663433
```

Performance of the model was evaluated numerically and graphically using both the training and the testing set as follows:

Accuracy:

Accuracy of the model on the train set: 79.17% / Accuracy of the model on the test set: 78.70%

Misclassification:

Confusion matrices show that on the test set 82.54% data of class 0 (NO BANDS) was correctly classified and 17.46% was misclassified while 73.33% data of class 1 (BANDS) was correctly classified while 26.67% was misclassified.
On the other hand, on the testing set, 86.75% of class 0 (NOBANDS) was correctly classified and 13.25% misclassified while 68.85% of class 1 (BANDS) was correctly classified and 31.15% misclassified

Results on the train set:

```
> table(cylTrain$dbandType, fitlda.train$class)

     0   1
  0 216  33
  1  57 126
> confusion(cylTrain$dbandType, fitlda.train$class)
         Accuracy Prior Frequency.0 Prior Frequency.1
           0.7917            0.5764            0.4236

Confusion Matrix
      Predicted (cv)
Actual      0      1
     0 0.8675 0.1325
     1 0.3115 0.6885
```

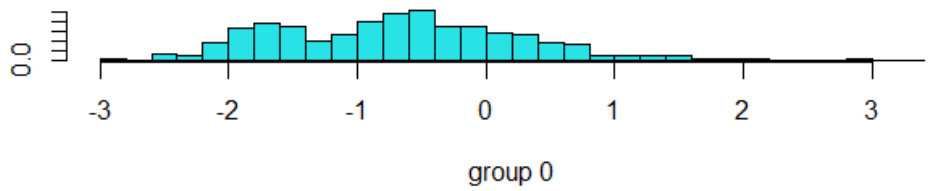
Results on the test set:

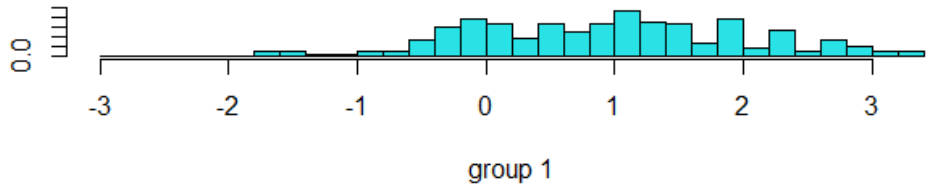
```
> table(cylTest$dbandType, fitlda.test$class)

     0  1
  0 52 11
  1 12 33
> confusion(cylTest$dbandType, fitlda.test$class)
         Accuracy Prior Frequency.0 Prior Frequency.1
           0.7870            0.5833            0.4167

Confusion Matrix
      Predicted (cv)
Actual      0      1
     0 0.8254 0.1746
     1 0.2667 0.7333
```

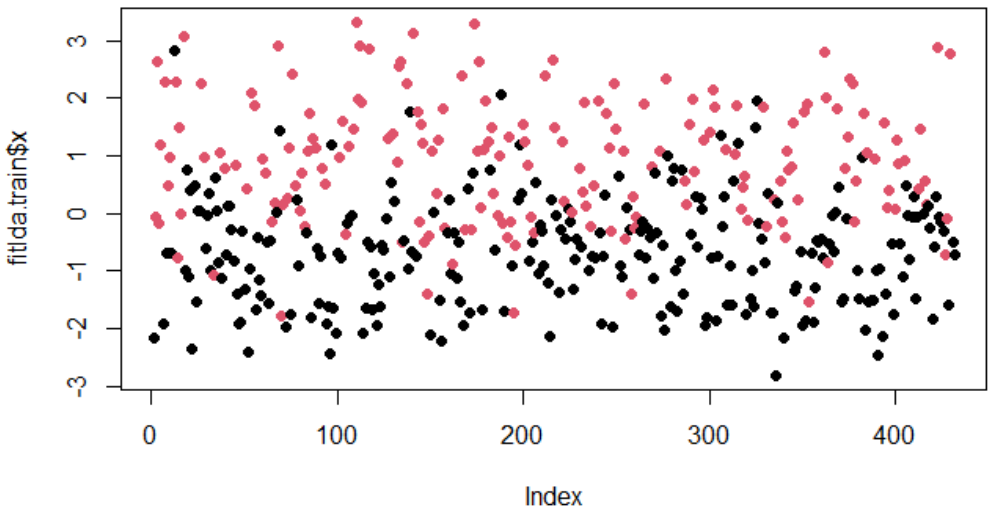
Classification histograms and scatterplot:

It can be seen from both the histograms and the scatterplots that the model separates a relatively large amount of datapoints. However, separation was not perfect. There are still a few points that make the two classes overlap in the middle.

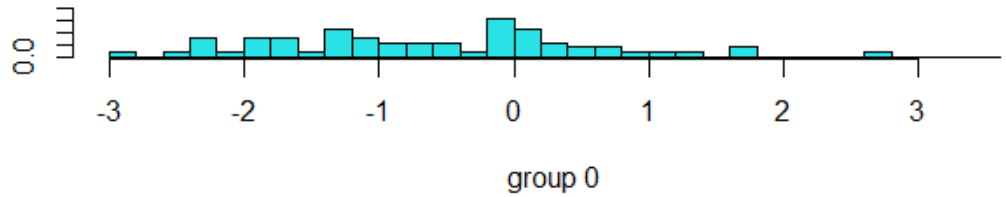
Histogram and scatter plot for the train set:

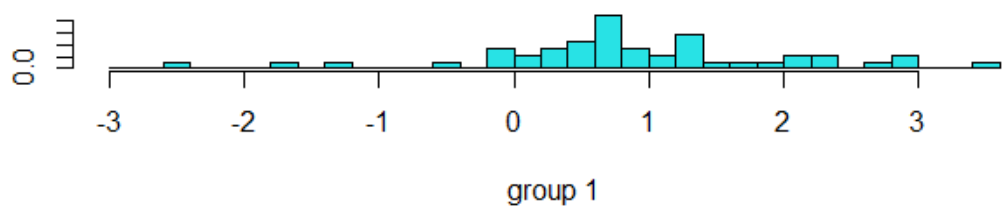


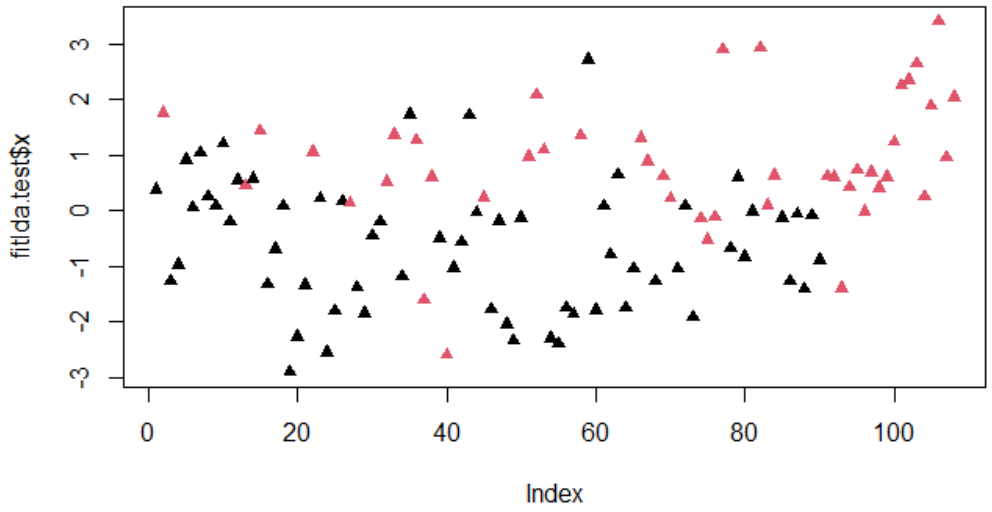




Histogram and scatter plot for the test set:





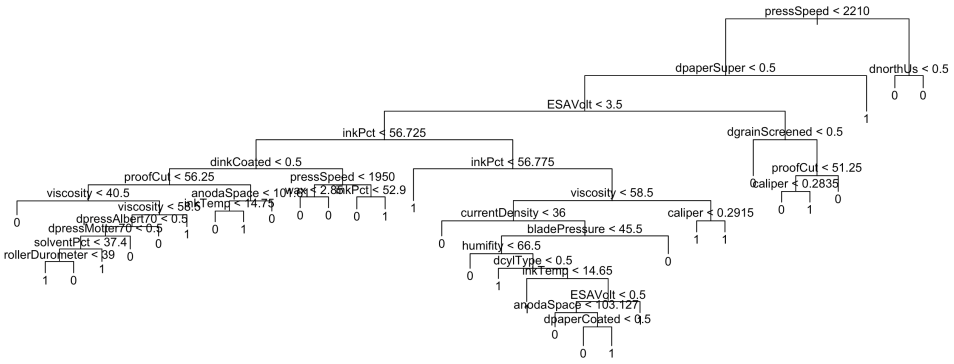


From all these numbers we can conclude that the model does fairy well on classifying the data. It had almost the same performance on both the training and on the testing sets. The model can make 79% accurate predictions of whether a particular combination of settings will potentially result in presence of bands in the cylinder during printing. These predictions are very important in the printing process since they can help to control the parameters that mostly influence banding in order to mitigate their effects including process delays and waste of resources.

MODEL 3: Decision Trees

We built a Decision Tree model that uses different variables of our dataset to classify the data into two categories of the target variable dbandType which indicate the occurrence of BANDS (dbandType=1) or NOBAND (dbandType=0). We used the tree function in r and the scatter criteria used was gini index.

Results:
The decision tree and its rules:



The following are the rules of the decision tree from above:
node), split, n, deviance, yval, (yprob)
* Denotes terminal node

1) root 378 516.800 1 (0.56878 0.43122)
2) pressSpeed < 2210 331 458.500 1 (0.51662 0.48338)
4) dpaperSuper < 0.5 316 435.900 1 (0.54114 0.45886)

```
        8) ESAVolt < 3.5 268 371.500 2 ( 0.49254 0.50746 )
         16) inkPct < 56.725 129 170.300 1 ( 0.62791 0.37209 )
           32) dinkCoated < 0.5 86 118.800 1 ( 0.53488 0.46512 )
             64) proofCut < 56.25 64  87.720 2 ( 0.43750 0.56250 )
              128) viscosity < 40.5 5   0.000 1 ( 1.00000 0.00000 ) *
              129) viscosity > 40.5 59  78.900 2 ( 0.38983 0.61017 )
                258) viscosity < 58.5 52  71.390 2 ( 0.44231 0.55769 )
                  516) dpressAlbert70 < 0.5 45  59.670 2 ( 0.37778 0.62222 )
                   1032) dpressMotter70 < 0.5 38  45.730 2 ( 0.28947 0.71053 )
                     2064) solventPct < 37.4 22  30.320 2 ( 0.45455 0.54545 )
                       4128) rollerDurometer < 39 15  17.400 2 ( 0.26667 0.73333 ) *
                       4129) rollerDurometer > 39 7   5.742 1 ( 0.85714 0.14286 ) *
                     2065) solventPct > 37.4 16   7.481 2 ( 0.06250 0.93750 ) *
                   1033) dpressMotter70 > 0.5 7   5.742 1 ( 0.85714 0.14286 ) *
                  517) dpressAlbert70 > 0.5 7   5.742 1 ( 0.85714 0.14286 ) *
                259) viscosity > 58.5 7   0.000 2 ( 0.00000 1.00000 ) *
             65) proofCut > 56.25 22  20.860 1 ( 0.81818 0.18182 )
              130) anodaSpace < 101.61 11  14.420 1 ( 0.63636 0.36364 )
                260) inkTemp < 14.75 5   0.000 1 ( 1.00000 0.00000 ) *
                261) inkTemp > 14.75 6   7.638 2 ( 0.33333 0.66667 ) *
              131) anodaSpace > 101.61 11   0.000 1 ( 1.00000 0.00000 ) *
           33) dinkCoated > 0.5 43  41.320 1 ( 0.81395 0.18605 )
             66) pressSpeed < 1950 29  14.560 1 ( 0.93103 0.06897 )
              132) wax < 2.85 23   0.000 1 ( 1.00000 0.00000 ) *
              133) wax > 2.85 6   7.638 1 ( 0.66667 0.33333 ) *
             67) pressSpeed > 1950 14  19.120 1 ( 0.57143 0.42857 )
              134) inkPct < 52.9 5   0.000 1 ( 1.00000 0.00000 ) *
              135) inkPct > 52.9 9  11.460 2 ( 0.33333 0.66667 ) *
         17) inkPct > 56.725 139 182.700 2 ( 0.36691 0.63309 )
           34) inkPct < 56.775 27   8.554 2 ( 0.03704 0.96296 ) *
           35) inkPct > 56.775 112 154.000 2 ( 0.44643 0.55357 )
             70) viscosity < 58.5 86 118.800 1 ( 0.53488 0.46512 )
              140) currentDensity < 36 12   6.884 1 ( 0.91667 0.08333 ) *
              141) currentDensity > 36 74 102.400 2 ( 0.47297 0.52703 )
                282) bladePressure < 45.5 67  91.070 2 ( 0.41791 0.58209 )
                  564) humifity < 66.5 5   0.000 1 ( 1.00000 0.00000 ) *
                  565) humifity > 66.5 62  81.770 2 ( 0.37097 0.62903 )
                   1130) dcylType < 0.5 23  21.250 2 ( 0.17391 0.82609 ) *
                   1131) dcylType > 0.5 39  54.040 2 ( 0.48718 0.51282 )
                     2262) inkTemp < 14.65 9   0.000 2 ( 0.00000 1.00000 ) *
                     2263) inkTemp > 14.65 30  39.430 1 ( 0.63333 0.36667 )
                       4526) ESAVolt < 0.5 24  26.990 1 ( 0.75000 0.25000 )
                         9052) anodaSpace < 103.127 11   0.000 1 ( 1.00000 0.00000 ) *
                         9053) anodaSpace > 103.127 13  17.940 1 ( 0.53846 0.46154 )
                           18106) dpaperCoated < 0.5 8   6.028 1 ( 0.87500 0.12500 ) *
                           18107) dpaperCoated > 0.5 5   0.000 2 ( 0.00000 1.00000 ) *
                       4527) ESAVolt > 0.5 6   5.407 2 ( 0.16667 0.83333 ) *
                283) bladePressure > 45.5 7   0.000 1 ( 1.00000 0.00000 ) *
             71) viscosity > 58.5 26  22.320 2 ( 0.15385 0.84615 )
              142) caliper < 0.2915 14   0.000 2 ( 0.00000 1.00000 ) *
              143) caliper > 0.2915 12  15.280 2 ( 0.33333 0.66667 ) *
        9) ESAVolt > 3.5 48  46.330 1 ( 0.81250 0.18750 )
         18) dgrainScreened < 0.5 31   0.000 1 ( 1.00000 0.00000 ) *
         19) dgrainScreened > 0.5 17  23.510 2 ( 0.47059 0.52941 )
           38) proofCut < 51.25 12  13.500 2 ( 0.25000 0.75000 )
             76) caliper < 0.2835 5   6.730 1 ( 0.60000 0.40000 ) *
             77) caliper > 0.2835 7   0.000 2 ( 0.00000 1.00000 ) *
           39) proofCut > 51.25 5   0.000 1 ( 1.00000 0.00000 ) *
      5) dpaperSuper > 0.5 15   0.000 2 ( 0.00000 1.00000 ) *
    3) pressSpeed > 2210 47  22.310 1 ( 0.93617 0.06383 )
      6) dnorthUs < 0.5 33   0.000 1 ( 1.00000 0.00000 ) *
7) dnorthUs > 0.5 14  14.550 1 ( 0.78571 0.21429 )
```

Performance of the model was evaluated numerically and graphically using both the training and the testing set as follows:

Accuracy:

Accuracy of the model on the train set: 91.53% / Accuracy of the model on the test set: 72.39%

```
tree.cyl   0   1
       0 195  12
       1  20 151
> err = (mistab[1,1] + mistab[2,2])/sum(mistab)
> err
[1] 0.9153439
```

```
> mistab

test.tree.cyl  0  1
            0 78 25
            1 20 40
> err = (mistab[1,1] + mistab[2,2])/sum(mistab)
> err
[1] 0.7239264
```

Misclassification:

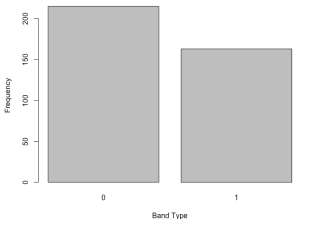
Confusion matrices show that on the training set 94.2% data of class 0 (NO BANDS) was correctly classified and 5.8% was misclassified while 88.3% data of class 1 (BANDS) was correctly classified while 11.7% was misclassified.

On the other hand, on the testing set, 75.73% of class 0 (NOBANDS) was correctly classified and 24.27% misclassified while 66.67% of class 1 (BANDS) was correctly classified and 33.33% misclassified.

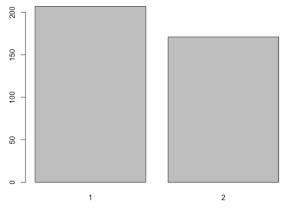
Classification bar charts:

It can be seen from both the bar charts that the model predicts the proportion of the Bands and No bands w.r.t each other in the training set well but the proportion in testing is not accurate. There is a greater difference between the number of observations with and without band.

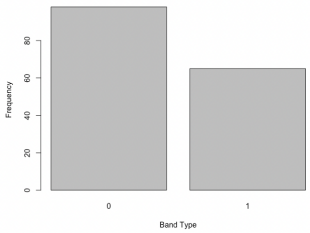
Bar chart of train dataset:



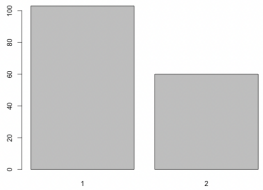
Bar chart of train dataset (predicted):



Bar chart of test dataset:



Bar chart of test dataset(predicted):



Analysis of the results:

From all these numbers we can conclude that the model does fairy well on classifying the data. The model performed better than testing set on training set with almost a 20% difference in accuracy. This could be an issue of overfitting but the accuracy of the model on testing set is in the acceptable range. Almost all classification models we tried also had similar accuracy.

Also, from the bar charts we can see that dBandType = 1 (BANDS) isn't predicted as well as dBandType = 0 (NO BANDS). As mentioned in the previous milestone, less type 2 error is preferred as predicting that there won't be any bands after printing when there would be bands isn't ideal as lot of resources would be wasted if banding appears on a printed sheet. Type 2 error here is 33.33% (more than type 1 which is 24.27%).

CONCLUSION

All models we tried were successful in predicting and classification. Regularized regression using lasso, ridge and Elastic Net performed well and Elastic Net were chosen as the best model to produce the best predictions. Both Linear Discriminant Analysis and Decision Tree models were able to classify the data into presence of bands and no bands with very close accuracy rates. It is recommended then to printing management to pay attention to variables that are the most important each model in order to maximize the quality of printing.

Works Cited

(n.d.). Retrieved from http://brestar8.mx.tripod.com/lec06.htm
(n.d.). Retrieved from https://www.bobst.com/usen/products/gravure-printing/process/