DATA MINING ASSIGNMENT 3

UT EID: vv4734
Name: Vidhoon Viswanathan

Problem 1

a)
Table 1.1 Comparison of RMSE for different Learning rate

Learning rate	Train RMSE	Test RMSE
0.0065	0.9893458	0.6258678
0.00575	0.9893283	0.6253554
0.00025	0.988645	0.5745223
MLR	0.9862762	0.5971078

Table 1.2 Comparison of number of epochs for different learning rates

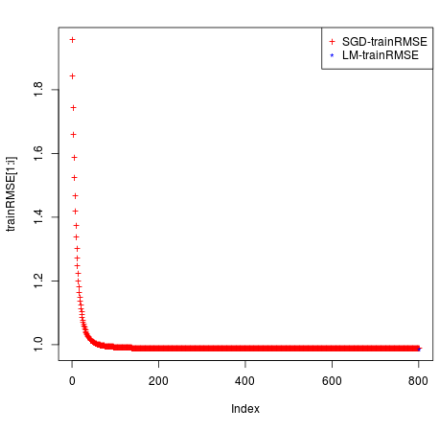
Learning rate	Epochs
0.0065	801
0.00575	884
0.00025	7931

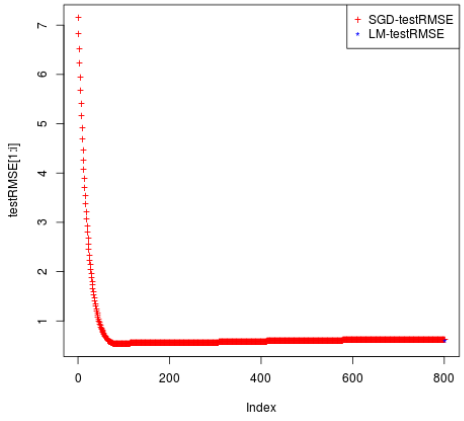
Terminating condition used:
When change in coefficients matrix is not greater than ε where $\varepsilon = 0.00001$

Please refer results below for RMSE vs epochs plots.

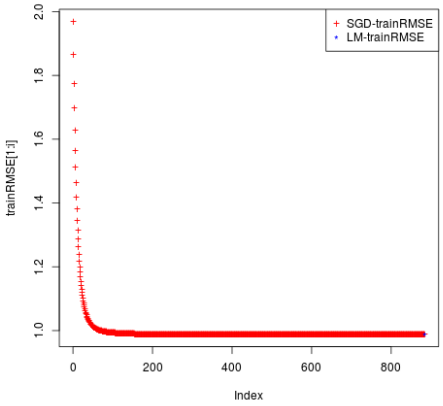
b) From Table 1.1, we can see that MLR (batch) has least train RMSE value while both MLR and SGD with rate=0.00025 have comparable RMSE with test data. It can be observed that, as learning rate is reduced the test RMSE value approaches MLR performance. But it must be also noted that it takes longer number of epochs to converge.

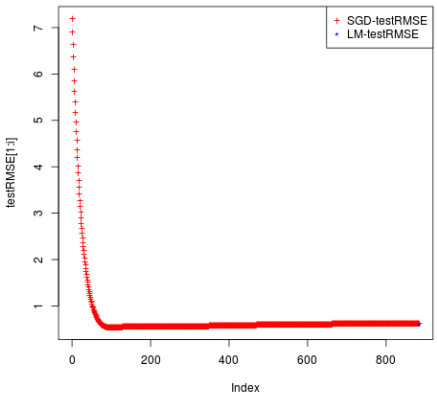
Train RMSE (left) and Test RMSE (right) vs epochs for learning rate = 0.0065



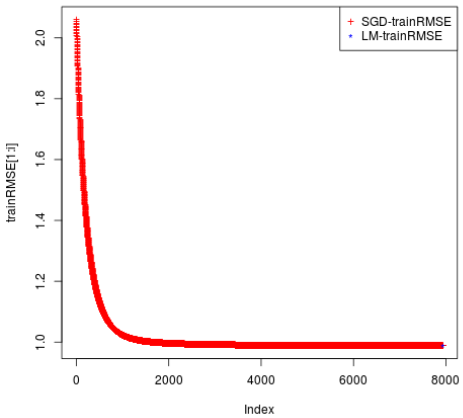


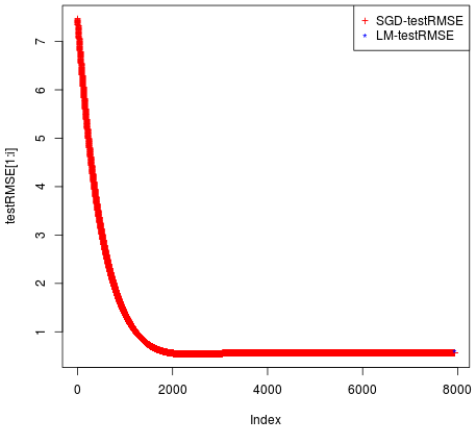
Train RMSE (left) and Test RMSE (right) vs epochs for learning rate = 0.00575





Train RMSE (left) and Test RMSE (right) vs epochs for learning rate = 0.00025





Problem 3

Table 3.1 Comparison of different classifiers for Machine1.csv data

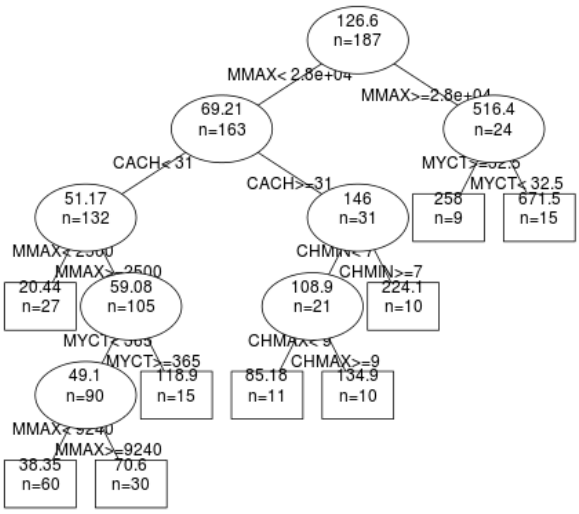
Classifier	Train MAE	Train MSE	Train R^2	Test MAE	Test MSE	Test R^2
RT (pruned)	39.4295	6741.605	0.7492286	58.78611	7451.228	0.5452306
SVR	32.6217	4640.045	0.8274016	32.95853	3938.685	0.759611
MLR	37.85944	3166.383	0.8822182	44.10398	5475.449	0.665818

a) Steps to avoid overfitting:

I used the method of pruning based on complexity parameter tuning with cross validation results to avoid overfitting. It involved the following steps:

1. Build a deep RT by giving a very small CP value (like CP=0.001)
2. Find the CP value corresponding to least cross validation error.
3. Prune the deep RT using the CP value from step 2.

If we set the complexity parameter to a very small value (CP=0.001), then a deep tree is generated that is both complex and is possibly overfit to the training data. Here is the deep tree that I got without tuning the complexity parameter:



To reduce depth and prevent overfitting, I examined the CP table of this deep tree to identify the

'nsplit' - number of splits for minimum cross validation error. Here is a sample CP table obtained during one of my runs:

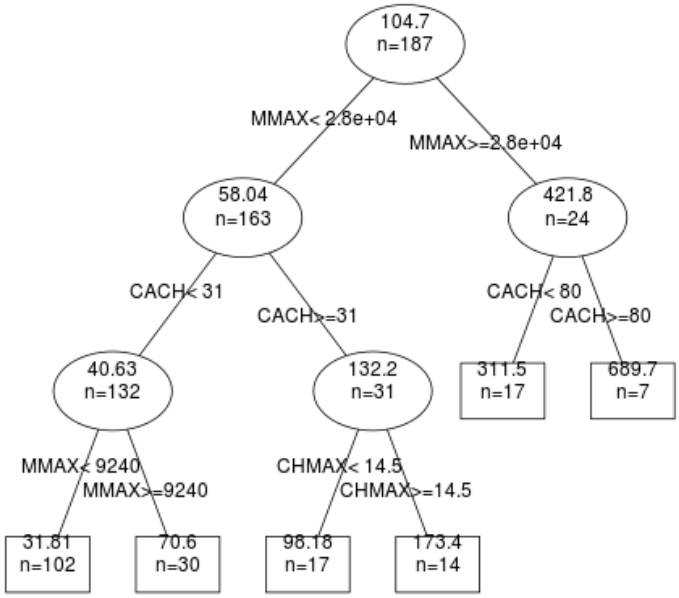
```
        CP nsplit rel error  xerror    xstd
1 0.5507410      0   1.00000 1.00893 0.33854
2 0.1410643      1   0.44926 0.46734 0.16396
3 0.0418381      2   0.30819 0.44013 0.16140
4 0.0086482      3   0.26636 0.40060 0.16128
5 0.0069371      4   0.25771 0.40167 0.16126
6 0.0012092      5   0.25077 0.39042 0.16134
7 0.0011957      6   0.24956 0.39143 0.16134
8 0.0010661      7   0.24837 0.39143 0.16134
9 0.0010000      8   0.24730 0.39119 0.16134
```

We can see that the CP value of .00121 has the least cross validation error. Hence I pruned the tree corresponding to this CP value.

The pruned RT model has:

Classifier	Train MAE	Train MSE	Train R^2	Test MAE	Test MSE	Test R^2
RT (pruned)	39.4295	6741.605	0.7492286	58.78611	7451.228	0.5452306

b) Here is the pruned version of my tree:

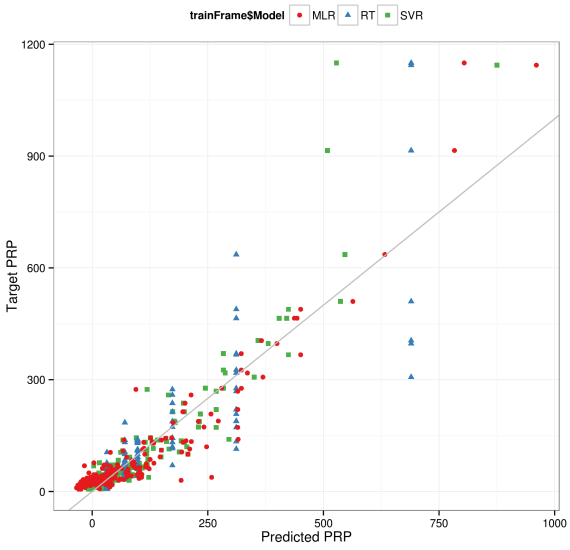


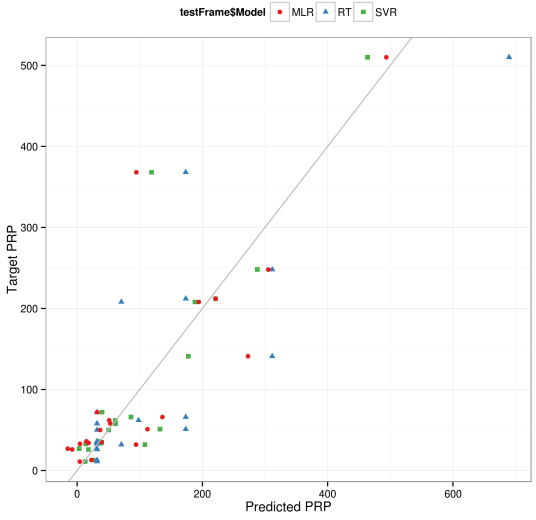
The pruned version of the tree has only **5 splits and 6 leaf nodes.**

c)

Classifier	Train MAE	Train MSE	Train R^2	Test MAE	Test MSE	Test R^2
SVR	32.6217	4640.045	0.8274016	32.95853	3938.685	0.759611

d)





Comments and observations:

1. RT model has large error for many cases whereas SVR prediction is close to accurate in all cases. This is because the number of values assigned by RT model is equal to the number of leaves the model has. This is in agreement with the results in [a] and [c]. We can find that RT(pruned) has higher MSE,MAE and hence its R^2 value is lower compared to SVR model.

2. Another key observation is that in RT model, the value assigned to a particular case is the mean of the values of training cases grouped in the leaf to which it belongs. We know that, mean is sensitive to the specific values seen in training. Hence, this largely increases the error in new/unseen training cases.

3. The high performance of MLR on training set is expected. This is because, SVR's cost function is epsilon resistant in the sense that it adds cost only to those predictions that vary beyond an epsilon limit (epsilon = 0.01). On the other hand, MLR is more fit to the training data. Hence, we find the R^2 value of MLR to be higher compared to other two models. This is proved by smaller R^2 value of MLR in comparison with SVR for test data. RT model cannot compete with MLR due to the limitations stated in [1] and [2]

e) With the distorted data set, a new pruned version of Regression Tree gets created:

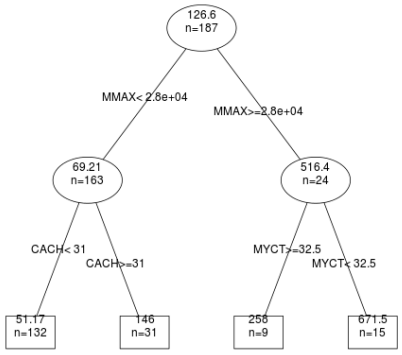
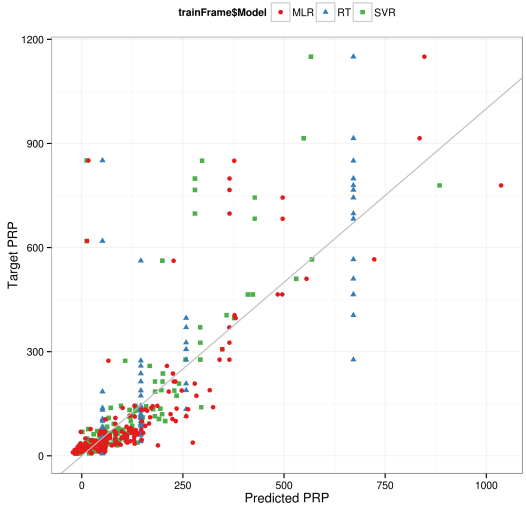
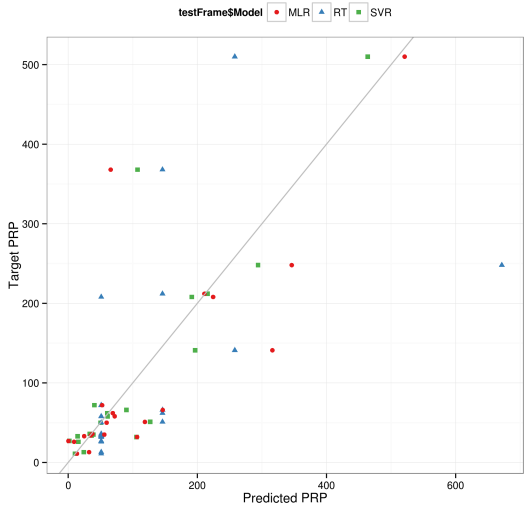


Table 3.2 Comparison of different classifiers for Machine2.csv

Classifier	Train MAE	Train MSE	Train R^2	Test MAE	Test MSE	Test R^2
RT	52.59111	11527.48	0.7136081	82.79612	17308.87	0.5640904
SVR	52.81092	16386.9	0.5928794	33.98009	4264.935	0.739699
MLR	60.33072	14177.2	0.6477779	46.39699	7203.853	0.5603286

Prediction results on Machine2.csv





Comments and Observations:

1) The performance of MLR degrades greatly (15.84% drop) and SVR degrades slightly while the performance of RT remains the same. The performance of MLR dips greatly because its cost function is not robust to outliers. Hence, in a dataset with many outliers, MLR model's fit would not perform well. In the case of SVR, the cost function is robust to outliers. Hence, there is only a slight dip in the performance.

2) RT model varies significantly (new tree) indicating that it is unstable. This is the reason why RT model is able to provide the same performance in the distorted dataset. Since the decision rules for each leaf change, the model naturally minimizes the effect of outliers. Hence, inspite of being a poor fit to data, the model is robust to outliers. In this case, the model is able to maintain its performance.

We can conclude that MLR cannot provide good fits for data with outliers while SVR and RT models are robust to outliers to a good extent.

f) R-squared of Random forest:

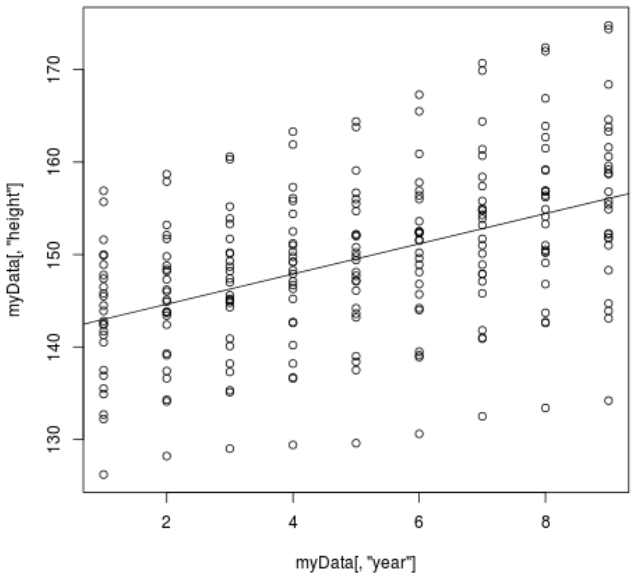
I created a random forest model using Machine1.csv data and another with Machine2.csv and obtained the train and test R-squared results:

Training Data	**Train R^2**	**Test R^2** (MachineTest.csv)
Machine1.csv	0.9320484	0.8231731
Machine2.csv	0.7049978	0.6492783

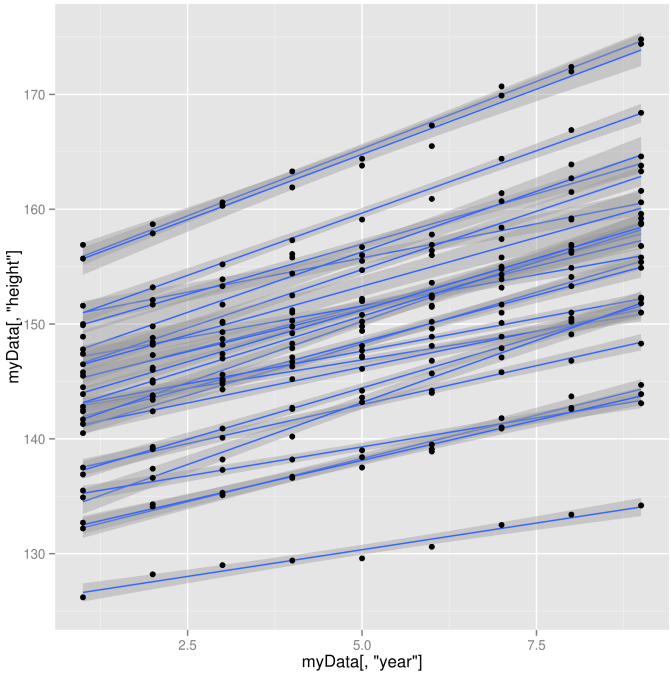
The ensemble solution produces better results as expected. It increases the Train R^2 values by 24% while Test R^2 increases by 19.07% for Machine1.csv. There is appreciable increase in Machine2.csv dataset also which validates the effectiveness of ensemble solution.

Problem 6

a) Relationship between height and year:



b)Relationship between height and year grouped by student id:



Comment on differences between (a) and (b)

It is clear from the two plots that the overall model does not fit the data. The R-squared value is 0.2123 for the overall model. But the plot for students grouped by ID fits each student well. This is because of the reasoning below.

Every student id has multiple observations over different years. **This violates the fundamental assumption of linear model fitting which is that data points are independent of one another.** This is not true anymore. The observations from the same student over different years are dependent. Hence the height data for each year can be grouped by student ids to look like they belong to different sub population. This can be seen in the grouped plot. The individual lines for each sub population seem to be fit better.

c)Table 6.1 Comparison of MSE for three model

Model	Group ID	MSE
Global Model	-	68.73458
Grouped Model (By student ID)	student ID 1	29.45143
	student ID 2	15.84143
	student ID 3	16.55714
	student ID 4	11.53143
	student ID 5	0.6052571
	student ID 6	3.224286
	student ID 7	0.9
	student ID 8	3.587729
	student ID 9	0.4642857
	student ID 10	29.23173
	student ID 11	14.50571
	student ID 12	0.3342857
	student ID 13	32.17
	student ID 14	6.702857
	student ID 15	0.3720143
	student ID 16	24.31714
	student ID 17	5.382857
	student ID 18	17.26429
	student ID 19	34.11857
	student ID 20	14.09429
	student ID 21	9.972129
	student ID 22	7.658571
	student ID 23	1.595714
	student ID 24	35.21714
	student ID 25	0.7428571
	student ID 26	12.35857

Multi Level Model	-	4.234867

d)Table 6.2 Comparison of MSE for three models

Model	Group ID	MSE
Global Model	-	80.06475
Grouped Model (By student ID)	student ID 1	0.1229778
	student ID 2	0.2713088
	student ID 3	1.643045
	student ID 4	0.8346104
	student ID 5	4.409226
	student ID 6	0.9661224
	student ID 7	0.9508844
	student ID 8	0.4189132
	student ID 9	3.104649
	student ID 10	1.2579
	student ID 11	3.118141
	student ID 12	0.6629279
	student ID 13	10.10042
	student ID 14	3.396239
	student ID 15	1.242047
	student ID 16	0.03240091
	student ID 17	2.679385
	student ID 18	0.1837415
	student ID 19	10.2312
	student ID 20	0.6678005
	student ID 21	5.235413
	student ID 22	2.25129
	student ID 23	1.599921
	student ID 24	10.94954
	student ID 25	1.823607

	student ID 26	0.5588227
Multi Level Model	-	2.720906

e) Comments on [c] and [d]

For the given dataset, irrespective of the amount of training data, we can see that Multilevel model performs better than the global model. This is because the height data from each student over consecutive years cannot be regarded as independent of each other. **This is proved by the large standard error in the intercept of the global model. This is because the dependence on group is treated as nuisance.**

```
> summary(globalModel)

Call:
lm(formula = height ~ year, data = trainData)

Residuals:
     Min       1Q   Median       3Q      Max
-16.9346  -5.5221   0.2154   4.3654  13.9154

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  141.485      3.189  44.363   <2e-16 ***
year           1.650      2.017   0.818    0.417
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.273 on 50 degrees of freedom
Multiple R-squared:  0.01321,   Adjusted R-squared:  -0.006529
F-statistic: 0.6692 on 1 and 50 DF,  p-value: 0.4172
```

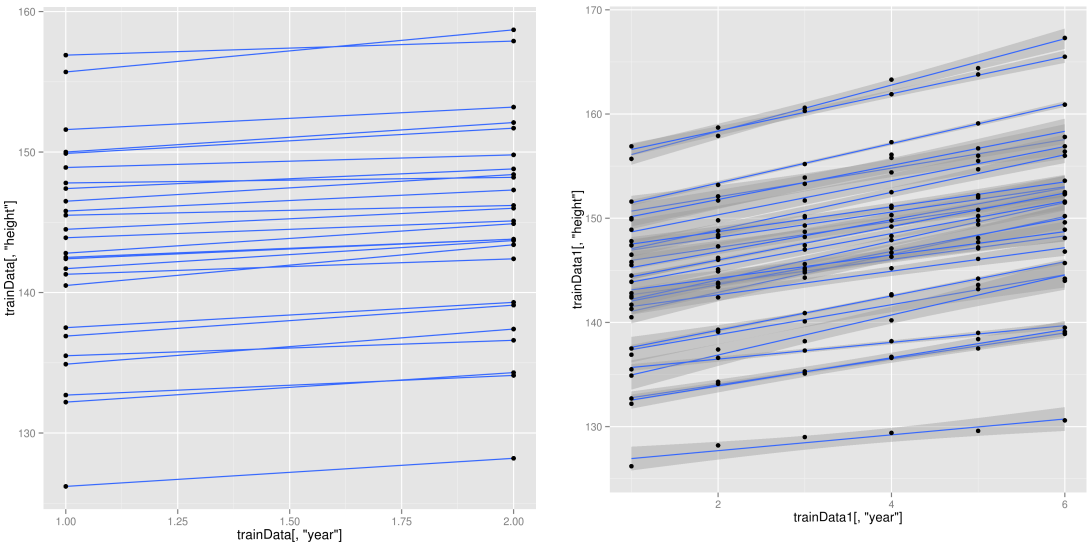
We can see that the multilevel model which takes into account of this dependence on group to have very low standard error on all predictors and intercepts compared to the global model.

```
> summary(multiLevelModel)
Linear mixed-effects model fit by REML
 Data: trainData
       AIC      BIC    logLik
  234.6233 244.1834 -112.3116

Random effects:
```

```
 Formula: ~year | id
 Structure: Diagonal
        (Intercept)          year  Residual
StdDev:    7.258798 0.0001057151 0.4468781

Fixed effects: height ~ year
               Value Std.Error DF  t-value p-value
(Intercept) 141.4846 1.4369928 25 98.45882       0
year          1.6500 0.1239417 25 13.31271       0
 Correlation:
     (Intr)
year -0.129

Standardized Within-Group Residuals:
        Min           Q1          Med           Q3          Max
-1.45443786 -0.48041316  0.03572478  0.46937991  1.56652040

Number of Observations: 52
Number of Groups: 26
```

This empirically shows that multilevel model fits the given data better than a global model.

With increase in training data, multilevel model's MSE decreases which is in agreement with our intuition. But, MSE of global model increases. This is because the effect of grouping is predominant when the training data increases. To give a visual hint on this aspect, compare the two figures below:



The left figure is the individual regression fit using 2 years as training data while the right figure is

the same using 6 years as training data. With less training data, the slope differences between most of the lines are comparable. We can see that the slopes of the individual lines have increased with more training data appearing less comparable now. This indicates that the grouping becomes stronger with increased training data. **This also explains the reason we have used random effects for the coefficient of 'year' predictor in the multilevel regression model. It accounts for the influence of grouped data on the slope of the linear fit.** This is also a reason for the boost in performance of multilevel model for latter case.

Problem-4

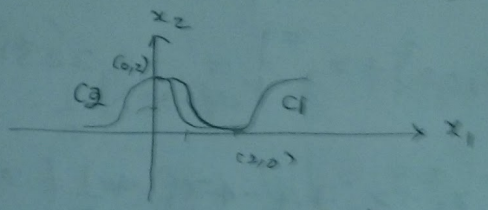
Given:

Points obtained from R^2

$X : (x_1, x_2) \in R^2$

$C_1 \rightarrow$ Normal with mean $(2,0) \rightarrow \mu_1$

$C_2 \rightarrow$ Normal with mean $(0,2) \rightarrow \mu_2$



$P(C_1) = 1/5$

$P(C_2) = 4/5$

$$P(X/C_1) = \frac{1}{2\pi |\Sigma|^{1/2}} \exp\left\{ \frac{-1}{2} (x-\mu_1)^T \Sigma^{-1} (x-\mu_1) \right\}$$

$$P(X/C_2) = \frac{1}{2\pi |\Sigma|^{1/2}} \exp\left\{ \frac{-1}{2} (x-\mu_2)^T \Sigma^{-1} (x-\mu_2) \right\}$$

a) To find decision boundary,

$$P(C_1/X) = P(C_2/X)$$

$$P(C_1) \cdot P(X/C_1) = P(C_2) \cdot P(X/C_2)$$

$$\frac{1}{5} \cdot \exp\left\{ \frac{-1}{2} (x-\mu_1)^T \Sigma^{-1} (x-\mu_1) \right\} = \frac{4}{5} \cdot \exp\left\{ \frac{-1}{2} (x-\mu_2)^T \Sigma^{-1} (x-\mu_2) \right\}$$

$$\log \frac{1}{5} - \frac{1}{2} (x-\mu_1)^T \Sigma^{-1} (x-\mu_1) = \log \frac{4}{5} - \frac{1}{2} (x-\mu_2)^T \Sigma^{-1} (x-\mu_2)$$

$$(X-\mu_1)^T \Sigma^{-1} (X-\mu_1) = [x_1-2 \quad x_2] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1-2 \\ x_2 \end{bmatrix}$$

$$= (x_1-2)^2 + x_2^2$$

Similarly

$$(X-\mu_2)^T \Sigma^{-1} (X-\mu_2) = (x_2-2)^2 + x_1^2$$

Now,

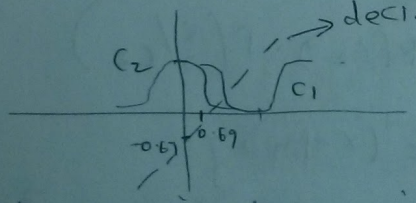
$$-1.6094 - \frac{1}{2}\left[(x_1-2)^2 + x_2^2\right] = -0.2231 - \frac{1}{2}\left[(x_2-2)^2 + x_1^2\right]$$

$$-1.3863 - \frac{1}{2}(x_1^2 + x_2^2 + 4 - 4x_1) + \frac{1}{2}(x_1^2 + x_2^2 + 4 - 4x_2) = 0$$

$$2x_1 - 2x_2 = 1.3863$$

$$\boxed{x_1 - x_2 = 0.6932}$$ is the required decision boundary.

(b) from (a) we see that



If C_1 has twice the misclassification penality as C_2, the decision boundary needs to move towards C_2 allowing more C_1 elements to classified correctly.

So, we find a decision boundary such that,

$$\text{Cost factor}(C_1) \cdot P(C_1/x) = \text{cost factor}(C_2) \cdot P(C_2/x)$$

2. $P(C_1/x) = P(C_2/x)$

$\Rightarrow$ From (a)

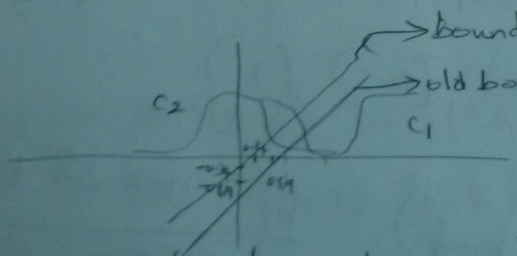
$$\log 2 + \log \frac{1}{5} - \frac{1}{2}\left[(x-\mu_1)^T \Sigma^{-1} (x-\mu_1)\right] = \log \frac{4}{5} - \frac{1}{2}\left[(x-\mu_2)^T \Sigma^{-1} (x-\mu_2)\right]$$

$$0.6932 - 1.6094 + 0.2231 - \frac{1}{2}(-4x_1) + \frac{1}{2}(-2x_2) = 0$$

$$2x_1 - 2x_2 = 0.6931$$

$$\boxed{x_1 - x_2 = 0.3466}$$

Now,



We can see the boundary with misclassificat penality has moved Towards C_2 allowing more items to be classified as C_1. accomodating the misclassification cost.

Problem-5

a) $P(B=good, F=empty, G=empty, S=yes)$

$= P(B=good) \cdot P(F=empty) \cdot P(G=empty / B=good, F=empty) \cdot P(S=yes / B=good, F=empty)$

$= (0.9) \cdot (0.2) \cdot (0.8) \cdot (0.2)$

$= \boxed{0.0288}$

b) $P(B=bad, F=empty, G=not\ empty, S=no)$

$= P(B=bad) \cdot P(F=empty) \cdot P(G=not\ empty / B=bad, F=empty) \cdot P(S=no / B=bad, F=empty)$

$= (0.1) \cdot (0.2) \cdot (0.1) \cdot (1)$

$= \boxed{0.002}$

c) $P(S=yes / B=bad) = \dfrac{P(S=yes \text{ and } B=bad)}{P(B=bad)}$

$P(S=yes, B=bad)$

$= \sum_{G,F} P(S=yes, B=bad)$

(3)

$= P(S=yes, B=bad, G=empty, F=empty)$

$+$

$P(S=yes, B=bad, G=empty, F=not\ empty)$

$+$

$P(S=yes, B=bad, G=not\ empty, F=empty)$

$+$

$P(S=yes, B=bad, G=not\ empty, F=not\ empty)$

$= P(S=yes / B=bad, F=empty) \cdot P(G=empty / B=bad, F=empty)$

$\cdot P(B=bad) \cdot P(F=empty)$

$+$

$P(S=yes / B=bad, F=empty) \cdot P(G=not\ empty / B=bad, F=empty)$

$\cdot P(B=bad) \cdot P(F=empty)$

$+$

$\ldots$

$= (0) + (0.1)(0.2) \cdot (0.1)(0.8) +$

$(0) + (0.1) \cdot (0.8) \cdot (0.1)(0.8)$

$= 0.0016 + 0.0064$

$= 0.008$

$$\therefore P(S = yes \mid B = bad) = \frac{0.008}{0.1}$$

$$= \boxed{0.08}$$

[Disclaimer: Used wikipedia sample problem as reference]

Problem-2

Given, points from R^n.

Assume the points from two sets S_1 & S_2 are

1) linearly seperable and
2) convex hulls intersect.

Let the points from S_1 be X_i

" the points from S_2 be Y_j

Now, from wikipedia:

A convex hull from points of S_1 is of the form $\left\{ \sum_{i=1}^{|S_1|} \alpha_i x_i \;\middle|\; (\forall i : \alpha_i \geq 0) \wedge \sum_{i=1}^{|S_1|} \alpha_i = 1 \right\}$

Similarly, convex hull from points of S_2

$$\left\{ \sum_{j=1}^{|S_2|} \beta_j y_j \;\middle|\; (\forall j : \beta_j \geq 0) \wedge \sum_{j=1}^{|S_2|} \beta_j = 1 \right\}$$

When the two hulls intersect, there must be atleast one point of intersection that is in both hulls. Let that point of intersection be 'k'.

Since the points are seperable, there is a hyperplane seperating x_i's and y_j's. Let it be such that,

$$P^T x_i + p_0 > 0 \quad \forall i$$
$$P^T y_j + p_0 < 0 \quad \forall j$$

→ Notation borrowed from Wolfram Math

be the conditions seperating the two sets of points.

Since, we have a point of intersection 'k', these two conditions should be satisfied for 'k'.

$$P^T k + p_0 = P^T \sum_{i=1}^{|S_1|} \alpha_i x_i + p_0$$

$$= P^T \sum_{j=1}^{|S_2|} \beta_j y_j + p_0$$

$$\therefore \sum_{i=1}^{|S_1|} \alpha_i (p^T x_i + p_0) = \sum_{j=1}^{|S_2|} \beta_j (p^T y_j + p_0)$$

$\Rightarrow \alpha_i = 0 \ \forall i$ and $\beta_i = 0 \ \forall j$ as $\alpha_i \geq 0$ and $\beta_i \geq 0$

But we have a <u>contradiction</u> since

$$\sum_{i=1}^{|S_1|} \alpha_i = 1 \quad \& \quad \sum_{j=1}^{|S_2|} \beta_j = 1$$

Hence, if convex hulls intersect, points are not seperable linearly.