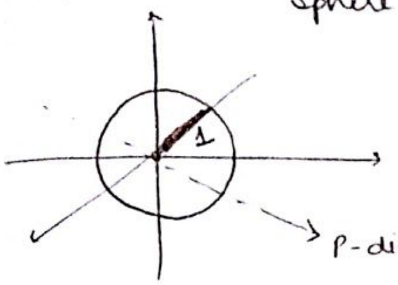
ESL WS 17/18 by Prof. Dr. Dr. THOMAS LENGAUER	SUBMITTED BY:
PROBLEM SET 3 DUE DATE : NOV 29, 2017 10 AM TA: MICHAEL SCHERER	HARSHITA JHAVAR 2566267 s8hajhav@stud.uni-saarland.de

PROBLEM 1: (T, 8 POINTS)

Problem 1. Given N data points uniformly distributed in a p-dimensional unit ball centered at origin. So, radius of sphere is 1.

To find: Median of distance from origin to closest data point.



Solution: Let 'd' be median distance which needs to be calculated. Since, 'd' is the median distance, so, ∵ given in question that N points are uniformly distributed, $\left(\frac{N}{2}\right)$ points will lie in region volume between sphere of radius = 1 and radius = d and other $\left(\frac{N}{2}\right)$ points will lie in region volume of sphere 'd'. So,

Volume of sphere of radius 1 =

$$V(1, p) = G(p)1^p = G(p).$$

Volume of sphere of radius 'd' :

$$V(d, p) = G(p)d^p.$$

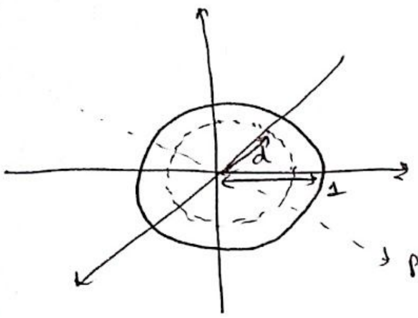
Volume between sphere of radius 1 and radius d is

$$V(1, p) - V(d, p).$$

$$\Rightarrow \quad G(p) - G(p)d^p \text{ ——— } (1)$$

But as I mentioned above, due to uniform distribution $\frac{N}{2}$ points will be in region between sphere of radius = 1 and radius = d. So, probability of being between 2 spheres is $\frac{1}{2}$. ——— (2).

So, since probability that a point falls into a sphere of radius r is proportional to the sphere's volume (given in question), so;



$$\Rightarrow \frac{1}{2} = \left(\frac{G(p) - G(p)\,d^{p}}{G(p)} \right)^{N}$$

since, we are talking of all N-data points.

$\Rightarrow$ Since $G(p)$ is a dimension dependent constant, so,

$$\frac{1}{2} = \left(\frac{1 - d^{p}}{1} \right)^{N}.$$

$$\Rightarrow \left(\frac{1}{2}\right)^{1/N} = 1 - d^{p}.$$

$$\Rightarrow \boxed{d = \left(1 - \left(\frac{1}{2}\right)^{1/N}\right)^{1/p}}.$$

Proved.

For K-nearest neighbour algorithm; this means that when there are data points distributed in a high dimensional space of features; there is a very high difficulty of finding a simple structure in the high dimensionally distributed data. So, K-means with its concept based on 'nearest-neighbour' classification will not function well for high-dimensional feature space as most points are far away from each other, so, distance metric of K-means fails here.

PROBLEM 2: (T, 12 POINTS)

Part a: Logistic regression is applicable for a setting in which the response variable or the outcome is a categorical variable and not a continuous value. This is basically binary form or with coded class labels for classification where the response is categorical and not continuous. For example: Predicting if a person has cancer/ does not have cancer based on the age will be logistic regression as the output variable is binary.
Linear regression (Y = f(X) + Constant) is applicable for a setting only when the output variable is dependent on its input predictor values and is continuous instead of being categorical. For example: Predicting the stock price rate trained on past records. The output here is a numerical value.

Part b:

Odds in favour of the event is the ratio of total chances in favour for an event to total chances against the event. The formula for Odds in favour for an event X with Probability P(X) is:

$$\text{Odds} = P(X) / (1-P(X))$$

However, odds against the event is reciprocal of the odds in favour of the event calculated above. This ratio intuitively gives us a measure of the likelihood of occurrence of an event (Odds in Favour) and non-occurrence of an event(Odds in Against).

Part c:

(Part c.) To prove:

$$p(x) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{(\beta_0 + \beta_1 X)}} \quad \text{— (1)}$$

is equivalent to

$$\frac{p(x)}{1-p(x)} = e^{\beta_0 + \beta_1 X} \quad \text{— (2)}$$

Proof:- Taking L.H.S. of (2) and substituting (1) in it; we get;

$$= \frac{p(x)}{1-p(x)}$$

= Using (1);

$$\frac{\dfrac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}}{1 - \dfrac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}}$$

$$= \frac{\dfrac{e^{\beta_0 + \beta_1 X}}{\cancel{1 + e^{\beta_0 + \beta_1 X}}}}{\dfrac{1 + \cancel{e^{\beta_0 + \beta_1 X}} - \cancel{e^{\beta_0 + \beta_1 X}}}{\cancel{1 + e^{\beta_0 + \beta_1 X}}}}$$

$$= e^{\beta_0 + \beta_1 X}.$$ which is RHS of (2).

Hence, proved.

Part d:

To show:
$$\frac{odd(x_i+\Delta)}{odd(x_i)} = e^{\beta_i \Delta}.$$

Proof →
$$\frac{odd(x_i+\Delta)}{odd(x_i)} = \frac{\dfrac{P(x_i+\Delta)}{1-P(x_i+\Delta)}}{\dfrac{P(x_i)}{1-P(x_i)}}$$

But $p(x_i) = \dfrac{e^{\beta_0+\beta_i x_i}}{1+e^{\beta_0+\beta_i x_i}}$ (from Part C).

Substituting it;

we get;
$$= \frac{\left(\dfrac{e^{\beta_0+\beta_i(x_i+\Delta)}}{1+e^{\beta_0+\beta_i(x_i+\Delta)}}\right) \times \dfrac{1}{1-\dfrac{e^{\beta_0+\beta_i(x_i+\Delta)}}{1+e^{\beta_0+\beta_i(x_i+\Delta)}}}}{\dfrac{e^{\beta_0+\beta_i x_i}}{1+e^{\beta_0+\beta_i x_i}} \times \dfrac{1}{1-\dfrac{e^{\beta_0+\beta_i x_i}}{1+e^{\beta_0+\beta_i x_i}}}}$$

$$= \frac{e^{\beta_0+\beta_i(x_i+\Delta)}}{1 \times \dfrac{e^{\beta_0+\beta_i x_i}}{1+e^{\beta_0+\beta_i x_i}} \times \dfrac{1+e^{\beta_0+\beta_i x_i}}{1}}$$

$$= \frac{e^{\cancel{\beta_0+\beta_i x_i}} \times e^{\Delta\beta_i}}{e^{\cancel{\beta_0+\beta_i x_i}}}$$

$$= e^{\Delta\beta_i}$$ → Hence, proved.

Part e:

(Part e.) Given; No. of features $p = 1$.

So, $Pr(Y=1|X) = p(X) = \dfrac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$

If $p(X) = 0.5$, then, substituting;

$$\Rightarrow \frac{1}{2} = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

$$\Rightarrow 1 + e^{\beta_0 + \beta_1 X} = 2 e^{\beta_0 + \beta_1 X}$$

$$\Rightarrow e^{\beta_0 + \beta_1 X} = 1$$

$\Rightarrow$ Taking log on both sides;

$$\beta_0 + \beta_1 X = 0$$

$$\Rightarrow \boxed{X = \frac{-\beta_0}{\beta_1}}$$

So, X has to ~~a~~ negative of ratio of β_0 and β_1 which are coefficient values in the model.

Probability $p(X) = 0.5$ tells us that predicting with only one feature value makes $P(Y=1|X) = p(X)$ and thus, makes 50% chances of Input ~~Test~~ variable to belong to 'class Y = 1' and ~~not~~ 50% chance of belonging to class 'not class Y=1'. Thus, each input is equally likely to belong to any of the two classes.

Part f:

Part f:- Logistic regression for k response classes by extending the given 2-way logistic regression model:-

$$P(Y=1|X) = \frac{e^{\beta_1 X}}{1+\sum_{j=1}^{K-1} e^{\beta_j X}}$$

$$P(Y=2|X) = \frac{e^{\beta_2 X}}{1+\sum_{j=1}^{K-1} e^{\beta_j X}}$$

$$\vdots$$

$$P(Y=K-1|X) = \frac{e^{\beta_{K-1} X}}{1+\sum_{j=1}^{K-1} e^{\beta_j X}}$$

$$P(Y=K|X) = \frac{1}{1+\sum_{j=1}^{K-1} e^{\beta_j X}}$$

for K-classes

Probability values for any input X.

Basically $1-\left(P(Y=K-1|X)+\cdots P(Y=1|X)\right)$.

So; from the above given probability values, we can then estimate the coefficient values as; (after taking log);

For K=1;

$$\log P(Y=1|X) = \beta_1 X - \log\left(1+\sum_{j=1}^{K-1} e^{\beta_j X}\right).$$

$$\Rightarrow \log(Y=1|X) = \beta_1 X + \log P(Y=K|X)$$

$$\Rightarrow \boxed{\log\frac{P(Y=1|X)}{P(Y=K|X)} = \beta_1 X}$$

coefficient β_1.

Similarly;

$$\vdots$$

$$\boxed{\log\frac{P(Y=K-1|X)}{P(Y=K|X)} = \beta_{K-1} X.}$$

To estimate the coefficient values; we have to use maximum likelihood estimation using

$$\text{Likelihood}(\beta) = \prod_{i:y_i=1} p(x_i) \cdot \prod_{i':y_{i'}\neq 1} (1-p(x_{i'}))$$

↳ zero

↳ inspired from Eq. 4.5 from book.

$$= \prod_{i=1}^{n} p(x_i)^{y_i} (1-p(x_i))^{1-y_i}$$

So, Maximizing the above likelihood function to get the corresponding coefficient values.

To maximize it; we can maximize its log because log is monotonic increasing. So; maximizing log likelihood;

$$L(\beta) = \prod_{i=1}^{n} p(x_i)^{y_i} (1-p(x_i))^{1-y_i}.$$

Taking log on both sides;

$$\Rightarrow \log L(\beta) = \sum_{i=1}^{n} y_i \log p(x_i) + (1-y_i)\log(1-p(x_i)).$$

$$= \sum_{i=1}^{n} y_i \log p(x_i) + \log(1-p(x_i)) - y_i \log(1-p(x_i))$$

$$= \sum_{i=1}^{n} y_i \log p(x_i) + \log(1-p(x_i)).$$

$$= \sum_{i=1}^{n} y_i \underbrace{\log\left(\frac{p(x_i)}{1-p(x_i)}\right)}_{\beta_i x} + \log(1-p(x_i))$$

↳ $p(x_i) = \underbrace{\dfrac{e^{\beta x}}{1+e^{\beta x}}}_{\text{Logistic function}}$.

$$= \sum_{i=1}^{n} y_i \beta_i x - \log(1+e^{\beta x})$$

↳ Now, β could be estimated using optimization algorithm as the function here is not in closed form.

So, taking its derivative will not give closure.

Again; for making predictions for any input X_i we can use the probabilities mentioned in beginning of Part f and then compute the maximum out of those probabilities to predict the class i.e.

$$\underbrace{\hat{y} = \underset{j=1\ldots K}{\text{argmax}}\ \hat{p}_j(x).}$$

PROBLEM 3: (T, 10 POINTS)
Part A: Linear Discriminant Analysis(LDA)

Part a.) We know that

$$P(Y=K \mid X=x) = P_K(x) = \frac{\pi_K f_K(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

— eq 4.10 from Book ISLR.

So, we are interested to maximize $P_K(x)$ and thus, interested to find 'K' or that class label which maximizes $P_K(x)$.

Since; $f_K(x) = \frac{1}{\sqrt{2\pi}\,\sigma_K} e^{\left(-\frac{(x-\mu_K)^2}{2\sigma_K^2}\right)}$ (Assumed Gaussian) eq 4.11 from Book ISLR.

So, we are interested in

$$= \underset{K}{\operatorname{argmax}} \; P_K(x)$$

$$= \underset{K}{\operatorname{argmax}} \; \pi_K f_K(x)$$

(Since, $\sum_{l=1}^{K} \pi_l f_l(x)$ is constant for all different numerator value / for all classes)

$$= \underset{K}{\operatorname{argmax}} \; \pi_K \times \frac{e^{\frac{-(x-\mu_K)^2}{2\sigma_K^2}}}{\sqrt{2\pi}\,\sigma_K}$$

$\longrightarrow$ numerator of eq^n 4.12 from Book ISLR.

Taking log on both sides; (as log is monotonic, so maximization can be performed on log as well).

$$= \underset{K}{\operatorname{argmax}} \left((\log \pi_K) + -\frac{(x-\mu_K)^2}{2\sigma_K^2} - \log\left(\sqrt{2\pi}\,\sigma_K\right) \right).$$

Since; shared variance is assumed, so, $\sigma_1^2 = \dots = \sigma_K^2 = \sigma^2$.

$$= \underset{K}{\operatorname{argmax}} \left((\log \pi_K) - \frac{(x-\mu_K)^2}{2\sigma^2} - \underbrace{\log\left(\sqrt{2\pi}\,\sigma\right)}_{\text{constant and not dependent on } K} \right)$$

$$= \underset{K}{\operatorname{argmax}} \left(\log \pi_K - \left(\frac{x^2 + \mu_K^2 - 2x\mu_K}{2\sigma^2} \right) \right).$$

$$= \underset{K}{\operatorname{argmax}} \left(\log(\pi_K) \underbrace{- \frac{x^2}{2\sigma^2}}_{\text{Not dependent on } K} - \frac{\mu_K^2}{2\sigma^2} + \frac{x\mu_K}{\sigma^2} \right).$$

$$= \underset{K}{\operatorname{argmax}} \left(\log(\pi_K) - \frac{\mu_K^2}{2\sigma^2} + \frac{x\mu_K}{\sigma^2} \right)$$

Proved

Part B: Quadratic Discriminant Analysis(QDA)

(Part B.) There is only one feature i.e. $p=1$.

$X \sim N(\mu_K, \sigma_K^2)$

Density function for one-dimensional normal distribution;

$$f_K(x) = \frac{1}{\sqrt{2\pi}\,\sigma_K} e^{\left(\frac{-1}{2\sigma_K^2}(x-\mu_K)^2\right)}.$$

Since,
$$p_K(x) = \frac{\pi_K f_K(x)}{\sum_{l=1}^{K} \pi_l f_l(x)} = \frac{\pi_K \times e^{\left(-1/2\sigma_K^2 (x-\mu_K)^2\right)} \times \left(1/\sqrt{2\pi}\,\sigma_K\right)}{\sum_{l=1}^{K} \pi_l \left(1/\sqrt{2\pi}\,\sigma\right) e^{-(1/2\sigma^2)(x-\mu_l)^2}}$$

$$= \frac{\pi_K e^{-(1/2\sigma_K^2)(x-\mu_K)^2}}{\sum_{l=1}^{K} \pi_l e^{-(1/2\sigma^2)(x-\mu_l)^2}}.$$

(As proved in Part A).

Again taking Log and ignoring the denominator's log as it is independent of K in order to maximize $p_K(x)$, we maximize numerator but this time; all σ^2 are not equal unlike Part A;

$$\Rightarrow \log p_K(x) = \log \pi_K - (1/2\sigma_K^2)(x-\mu_K)^2 - \underbrace{\log\left(\sum_{l=1}^{K} \pi_l e^{\frac{-(x-\mu_l)^2}{2\sigma^2}}\right)}_{\text{Ignored}}$$

$$= \log \pi_K - \left(1/2\sigma_K^2\right)(x-\mu_K)^2.$$

$$= \log \pi_K - \left(1/2\sigma_K^2\right)\left(x^2 + \mu_K^2 - 2\mu_K x\right).$$

$$= \log \pi_K - \underbrace{\frac{x^2}{2\sigma_K^2}}_{} - \frac{\mu_K^2}{2\sigma_K^2} + \frac{\mu_K x}{\sigma_K^2}.$$

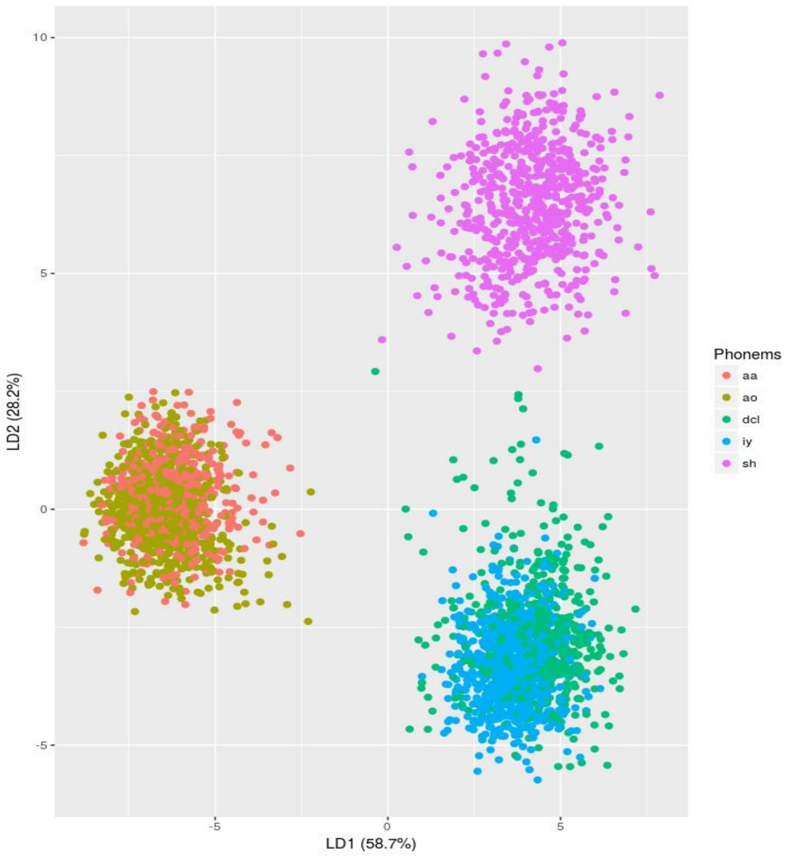
Not Linear as x has power 2, Infact quadratic as this is the highest power.

Proved

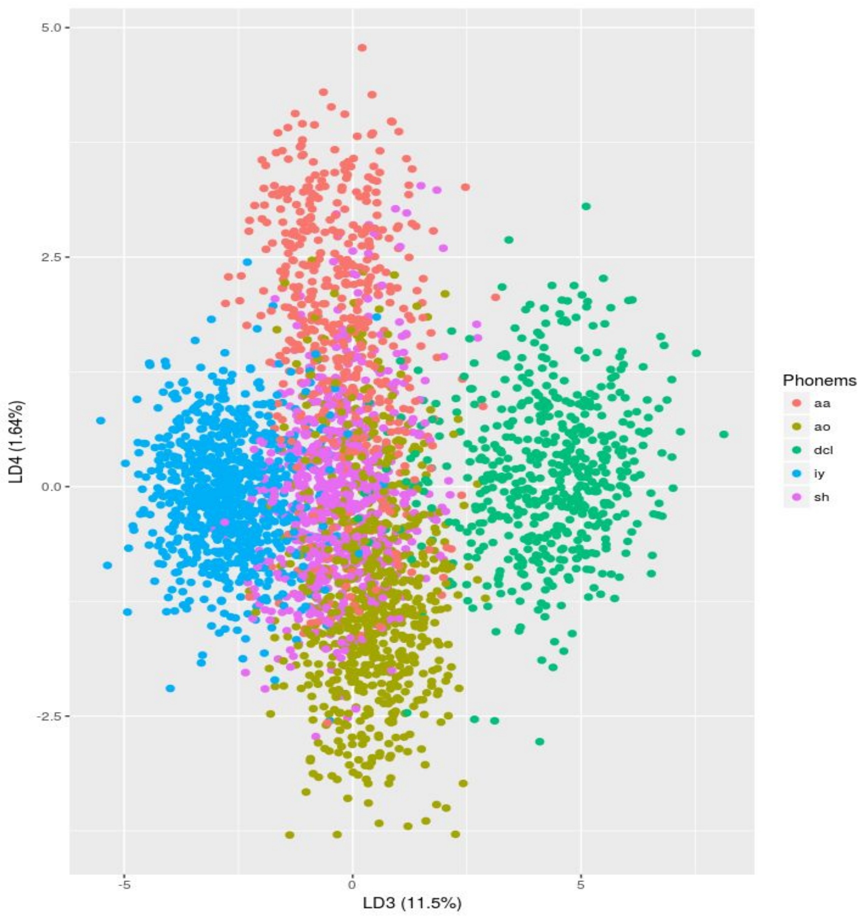
PROBLEM 4: (P, 20 POINTS)
Part a: In the attached R-Script file.

Part b:
The train error is 0.05598802 and the test error is 0.08041061. The test error is more than the train error because LDA is less flexible which results in less variance in training data during modelling and giving more bias for the test data.

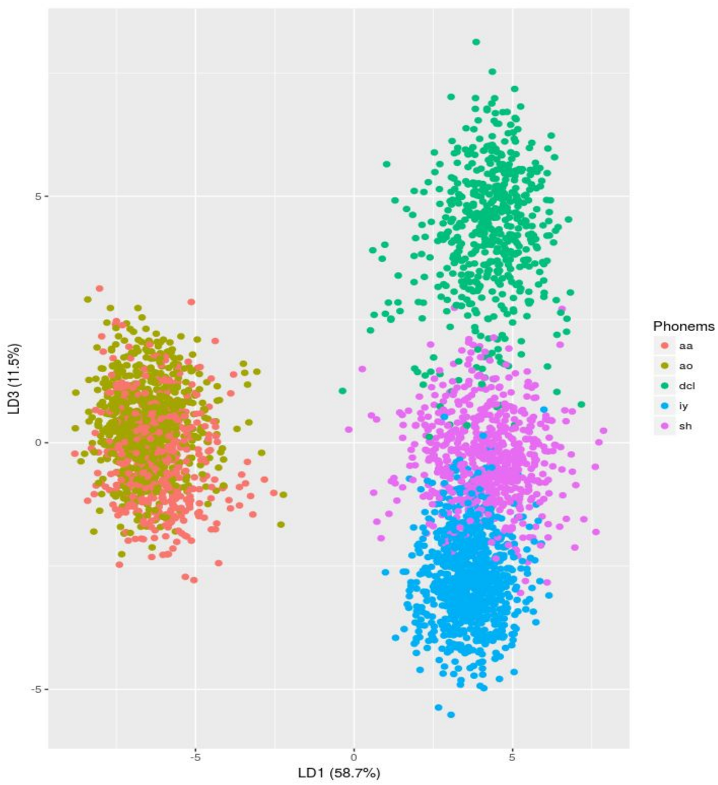
Part c:
Plotting LD1 and LD2: The clusters are not clearly separated. There is a lot of overlap.



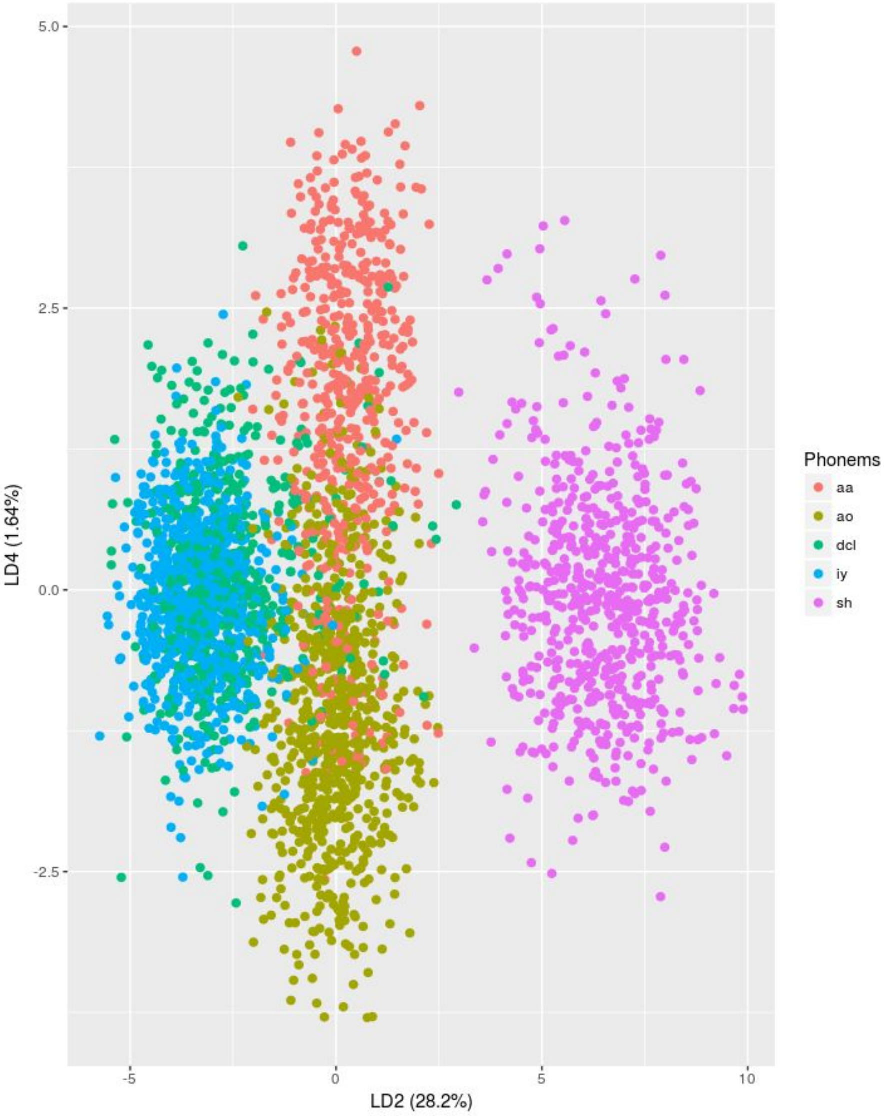
Plotting LD3 and LD4: Again, the boundary line for separation is not linear.



Below are other plots with different dimensions,
Between LD1 and LD3: Better representation than the above two plots due to clear visibility of the different points from different classes without much overlap.



Between LD2 and LD4: Best among all plots with least overlap.



Part d:
After subsetting the dataset for class "aa" and "ao" as the only possible labels, the training error is 0.1064163 and the test error is 0.214123.

Part e:
For QDA on full training data, the training error is 0 and the test error is 0.1582549.
For QDA on training data only from classes “aa” and “ao”, the training error is 0 while the test error is 0.3394077.

On comparing the results for the test error above, I will prefer to choose LDA over QDA as the test error is lesser in LDA. However, it also depends on the decision boundary as LDA performs better for linear decision boundaries while QDA performs better for non-linear decision boundaries.

Part f:
The confusion matrix for LDA is

	aa	ao
aa	121	39
ao	55	224

While the confusion matrix for QDA is

	aa	ao
aa	29	2
ao	147	261

Observation: LDA predicts more accurately “ao”[224] and “aa”[121] class labels with lesser false prediction for ‘ao’ but larger false prediction for ‘aa’ as compared to QDA model. There is more bias towards ‘aa’ in QDA as it incorrectly predicts many ‘ao’ as ‘aa’ with very less false predictions for ‘aa’.