Problem 1

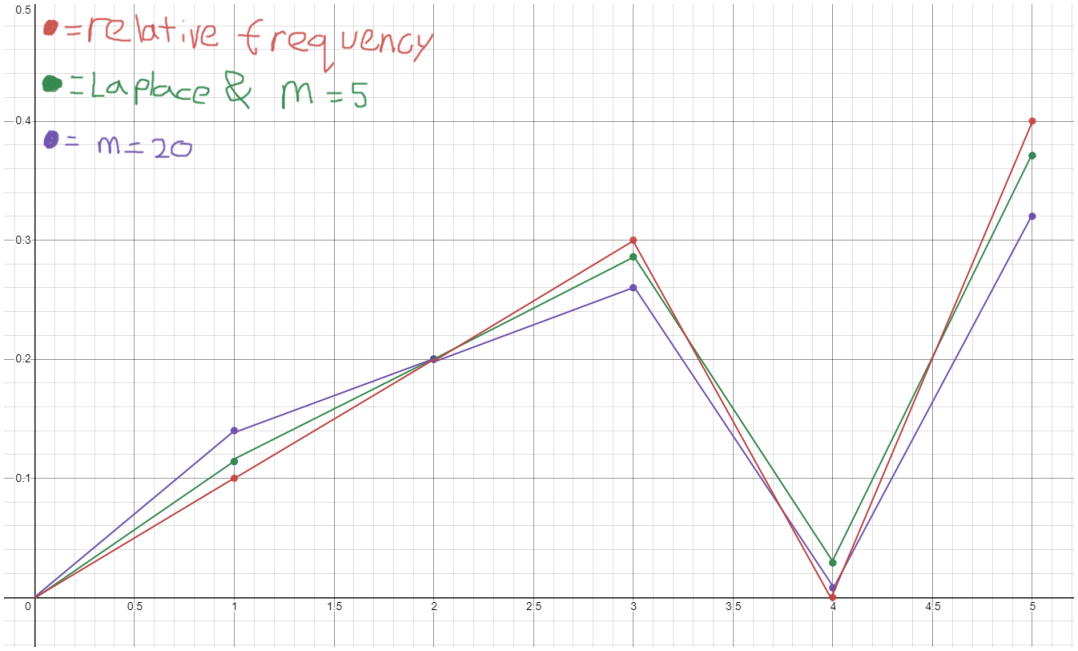
a. 3 x 2 x 2 x 3 x 4 = 144 –> 2^{144} nanoseconds
b. 4 x 3 x 3 x 4 x 5 = 720 nanoseconds
c. $(2^3 - 1)(2^2 - 1)(2^2 - 1)(2^3 - 1)(2^4 - 1)$
7 x 3 x 3 x 7 x 15 = 6615 nanoseconds

Problem 2

Point: [2, 2, 3, '+'] **a)** 0.743 **b)** 0 **c)** 0.257 **d)** 0.066
Point: [3, 3, 2, '+'] **a)** 0.928 **b)** 0 **c)** 0.072 **d)** 0.005
Point: [1, 2, 3, '+'] **a)** 0.186 **b)** 0 **c)** 0.814 **d)** 0.663
Point: [1, 4, 1, '+'] **a)** -0.557 **b)** 1 **c)** 1.557 **d)** 2.425
Point: [4, 4, 4, '+'] **a)** 3.343 **b)** 0 **c)** 0 **d)** 0
Point: [2, 2, 2, '+'] **a)** 0.0 **b)** 1 **c)** 1.0 **d)** 1.0
Point: [3, 3, 1, '-'] **a)** -0.186 **b)** 1 **c)** 1.186 **d)** 1.406
Point: [1, 1, 1, '-'] **a)** 1.671 **b)** 0 **c)** 0 **d)** 0
Point: [3, 3, 2, '-'] **a)** -0.928 **b)** 1 **c)** 1.928 **d)** 3.719
Point: [0, 4, 2, '-'] **a)** 0.371 **b)** 0 **c)** 0.629 **d)** 0.395
Point: [4, 0, 0, '-'] **a)** 1.114 **b)** 0 **c)** 0 **d)** 0
Point: [0, 0, 3, '-'] **a)** 1.114 **b)** 0 **c)** 0 **d)** 0

Problem 3

1	0.1	0.114	0.114	0.14
2	0.2	0.2	0.2	0.2
3	0.3	0.286	0.286	0.26
4	0	0.029	0.029	0.08
5	0.4	0.371	0.371	0.32



As the number of the pseudo counts increases, the probabilities of the 4 classes grows closer to the overall average which would have been 1/5

Problem 4

a. 4 x 4 x 3 x 3 = 144 –> 2^{144} conjunctive hypotheses
b. (Budget=[Low,High], Genre=Drama, Director=Great)
c. (Genre = drama)
d. (Budget=Low, Genre=Drama, Famous Actors=No, Director=Great)

Problem 5

	p	weight	impurity	Feature impurity
Budget = low	1/3	6/16	0.918	0.873
Budget = medium	3/5	5/16	0.971	
Budget = high	4/5	5/16	0.722	
Genre = Documentary	2/5	5/16	0.971	0.85
Genre = Drama	5/6	3/8	0.65	
Genre = Comedy	2/5	5/16	0.971	
FamousActor = Yes	5/8	1/2	0.954	0.977
FamourActor = No	1/2	1/2	1	
Director = Great	5/7	7/16	0.863	0.935
Director = Unknown	4/9	9/16	0.991	

This means that our root node will be the "Genre" feature

Genre=Documentary				
	p	weight	impurity	Feature impurity
Budget = low	0	2/5	0	0.55
Budget = medium	2/3	3/5	0.918	
Budget = high	0	0	N/A	
FamousActor = Yes	1/3	3/5	0.918	0.95
FamourActor = No	1/2	2/5	1	
Director = Great	0	1/5	0	0.8
Director = Unknown	1/2	4/5	1	

Next node will be Budget

Genre=Comedy				
	p	weight	impurity	Feature impurity
Budget = low	1/2	2/5	1	0.8
Budget = medium	0	1/5	0	
Budget = high	1/2	2/5	1	
FamousActor = Yes	1	1/5	0	0.649
FamourActor = No	1/4	4/5	0.811	
Director = Great	2/3	3/5	0.918	0.589
Director = Unknown	0	2/5	0	

Next node will be director

Genre=Drama				
	p	weight	impurity	Feature impurity
Budget = low	1/2	2/6	1	0.333
Budget = medium	1	1/6	0	
Budget = high	1	3/6	0	
FamousActor = Yes	3/4	4/6	0.811	0.541
FamourActor = No	1	2/6	0	
Director = Great	1	1/2	0	0.459
Director = Unknown	2/3	1/2	0.918	

Next node will be budget

Genre=Documentary and Budget=Low				
	p	weight	impurity	Feature impurity
FamousActor = Yes	0	1	0	0
FamourActor = No	0	0	N/A	
Director = Great	0	1/2	0	0
Director = Unknown	0	1/2	0	

next node is FamousActor

Genre=Documentary and Budget=Medium				
	p	weight	impurity	Feature impurity
FamousActor = Yes	1	1/3	0	0.667
FamourActor = No	1/2	2/3	1	
Director = Great	0	0	N/A	0.918
Director = Unknown	2/3	1	0.918	

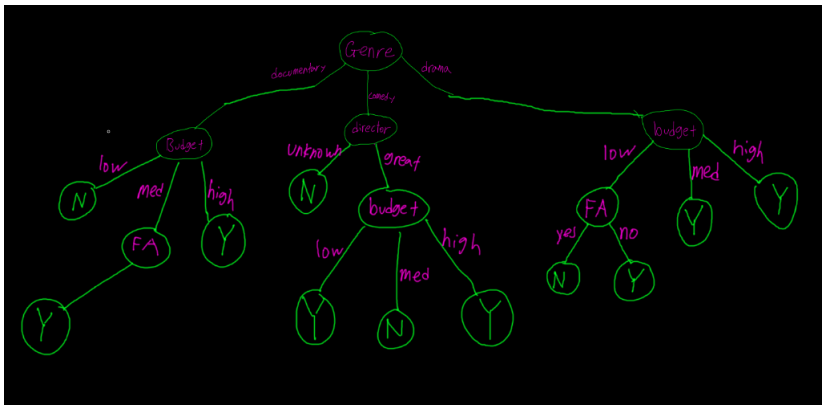
next node is FamousActor

Genre=Comedy and Director=Great				
	p	weight	impurity	Feature impurity
Budget = Low	1	1/3	0	
Budget = Medium	0	1/3	N/A	0
Budget = High	1	1/3	0	

Since budget's impurity value is 0 then we default to that as the next node. Same case for Director=unknown

Genre=Drama and Budget=Low				
	p	weight	impurity	Feature impurity
FamousActors = Yes	0	1/2	N/A	0.5
FamousActors = No	1	1/2	1	
Director = Great	1	1/2	1	
Director = Unknown	0	1/2	N/A	0.5

Next node is FamousActors



Training Data
Labelled Yes: 1 2 3 4 5 6 7 9 10 111 12 13 14 15 16
Labelled No: 8
Error Rate: 15/16

Test Data 1
Labelled Yes: 5 6 9 11 12 13 14 15 17 18 19
Labelled No: 1 2 3 4 7 8 10 16 20
Error Rate: 9/20

Test Data 2
Labelled Yes: 1 3 4 5 6 8 9 11 12 14 16 17 19 20
Labelled No: 2 7 10 13 15 18
Error Rate: 6/20

Test Data 3
Labelled Yes: 1 3 4 5 6 8 9 11 12 13 14 15 16
Labelled No: 2 7 10 13 15
Error Rate: 5/20

Problem 6

For my program I took the following steps
1. Sort the training data into 3 seperate arrays, 1 per class
2. Find the centroids of each of the classes
3. Create a normalized vector between two centroids and find the "d" value by inputting a midpoint
4. Once the equations are created I check for each point for what side of the plane it is on.
5. Checking with the 01 discriminant function if the value is positive then it chooses class 0 and compares with the 20 discriminant function to choose between class 2 and 0.
6. Step 5 is repeated for each point.

Performance
Test 1 accuracy: 9 Errors out of 75 points
Test 2 accuracy: 33 Errors out of 300 points

CodaLab Performance
Accuracy: 0.93
Precision: 0.89
F measure: 0.89
Recall: 0.89
F1 Score: 0.89