

 README

Tarea 4: Softmax

Este código base es prácticamente idéntico al de la tarea 3, pero con archivos base para la implementación de softmax. La idea es que usted importe rápidamente los avances realizados en la tarea 3 acá, y que continúe con la implementación de los métodos de optimización faltantes (adam y rmsprop), así como la hipótesis y pérdida de softmax.

Deberá en esta tarea trabajar en técnicas de visualización del espacio de características, aunque en los ejemplos de las lecciones ya encuentra bastante información al respecto.

Dependencias

Este código utiliza los paquetes statistics y automatic-differentiation. Desde la terminal de GNU/Octave los instala con:

```
pkg install -forge statistics  
pkg install "https://github.com/StevenWaldrip/Automatic-Differentiation/ar
```

Ejecución de la solución

- Ubíquese en la carpeta `root` del repositorio.

Softmax

- Ejecute el siguiente código en la terminal de GNU/Octave:

```
octave softmax_main.m
```

- Corresponde a la solución de la parte 1,2,3,4,5,6,7,8,9,10 de la tarea.

Nota: Al principio del código se encuentran flags para ejecutar partes específicas de la tarea por defecto se ejecutan todas las partes. Si desea ejecutar solo una parte, ponga la flag en false .

Flag	Descripción
part_4_5	Corresponde a la solución 4_5 de la tarea.
part_6_7_8_9_10	Corresponde a la solución 7,8,9,10 de la tarea.

Documentación de la Solución

También se generó un `solution.pdf` de este mismo readme. Se ubica en la carpeta `docs` junto con todas las gráficas de la tarea.

Parte 1 y 2

- Se agrega la hipótesis de softmax.

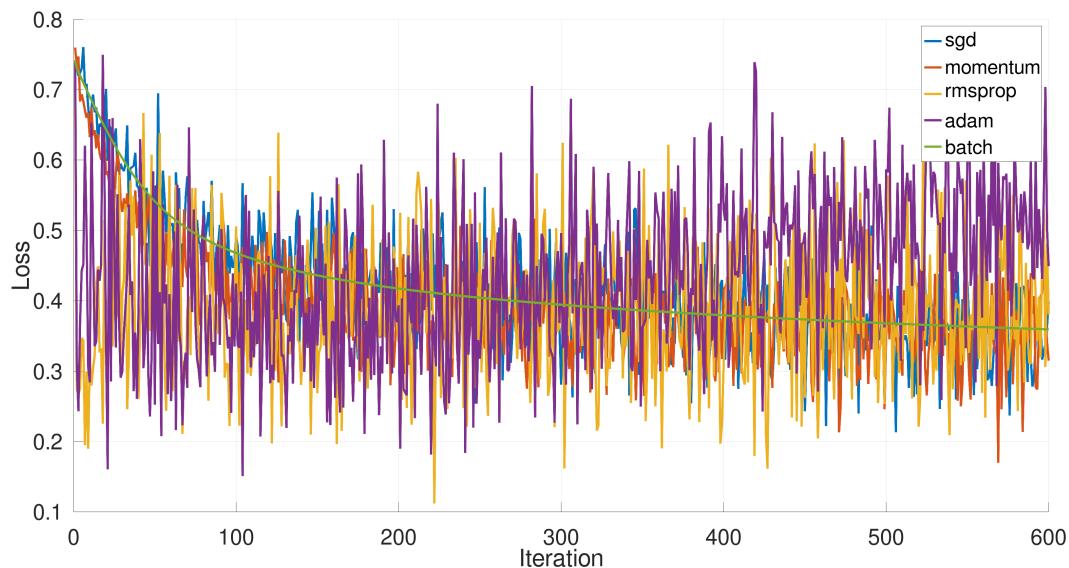
```
% Hypothesis function used in softmax
% Theta: matrix, its columns are each related to one
%         particular class.
% returns the hypothesis, which has only k-1 values for each sample
%         as the last one is computed as 1 minus the sum of all the rest.
function H=softmax_hyp(Theta,X)
    assert( rows(Theta) == columns(X) ); % check if Theta have the same number of features as X
    H = 1 ./ (1 + exp(-(X*Theta)));
endfunction
```

- Se agrega la perdida de la softmax.

```
% Loss function used in softmax
function err=softmax_loss(Theta,X,Y)
    assert(rows(Y)==rows(X));
    assert(columns(Y)==columns(Theta));
    ## residuals
    R = Y - softmax_hyp(Theta,X);
    ## Loss
    err = mean(sum(R.*R,2));      You, 6 hours ago • fix: better calculation loss
endfunction
```

Parte 3, 4, 5

- Se optimiza softmax con los métodos de "sgd" , "momentum" , "rmsprop" , "adam" y "batch" ,. Y se gráfica el error de la perdida.



- Se calcula el error empírico de softmax.

```
function [num_err, percentage_err] = softmax_empirical_error(Theta,X,Y)
    %% Softmax empirical error
    %% Returns the number of errors and the percentage of errors
    %%
    %% @param @var{Theta}: matrix, its columns are each related to one
    %%           particular class.
    %% @param @var{X}: matrix, its columns are each related to one
    %%           particular sample.
    %% @param @var{Y}: matrix, its columns are each related to one
    %%           particular sample.
    %% @returns @var{num_err}: number of errors
    %% @returns @var{percentage_err}: percentage of errors
    %% Usage:
    %%     [num_err, percentage_err] = softmax_empirical_error(Theta,X,Y)
    assert(columns(Theta)==columns(Y));
    H = softmax_hyp(Theta, X);
    %% max values in each row
    [~, sampler_maxH] = max(H, [], 2);      You, 6 hours ago • fix: compare index with
    [~, sampler_maxY] = max(Y, [], 2);

    % number of errors
    num_err = sum(sampler_maxH != sampler_maxY);
    percentage_err = num_err / rows(Y) * 100;

endfunction
```

- Se obtiene el resultado del error empírico de softmax, para cada de los métodos de optimización.

PART FOUR and PART FIVE

Training with method 'sgd'.

100% =====

-0.6632	-0.4589	1.0817
-2.8462	2.5257	1.1811
-0.9391	-0.4515	-1.7735
-1.8048	-2.4371	1.2604
2.5215	-2.2849	1.1267

Training error: 34 / 274 (12.4088 %)

Test error: 17 / 68 (25 %)

Training with method 'momentum'.

100% =====

-0.6770	-0.5027	1.1172
-3.0276	2.3107	1.1334
-0.8342	-0.4441	-1.6534
-1.6724	-2.3562	1.4006
2.6153	-2.0550	1.1337

Training error: 35 / 274 (12.7737 %)

Test error: 13 / 68 (19.1176 %)

Training with method 'rmsprop'.

100% =====

-18.2702	-20.8807	20.8136
-25.5875	50.2236	13.2716
-28.4690	-5.6364	-27.6916
-44.7543	-25.3019	19.2026
26.8087	-44.1661	19.5766

Training error: 31 / 274 (11.3139 %)

Test error: 8 / 68 (11.7647 %)

Training with method 'adam'.

100% =====

-30.2400	-23.6234	-36.6362
-29.2301	16.7847	40.4593
33.7466	28.4367	-0.3083
-29.1711	-34.7036	-11.3209
-6.1797	34.8213	33.1186

Training error: 29 / 274 (10.5839 %)

Test error: 3 / 68 (4.41176 %)

Training with method 'batch'.

100% =====

-0.6640	-0.4573	1.0935
-2.8542	2.5615	1.1731
-0.9682	-0.4513	-1.7820
-1.8165	-2.4250	1.2769
2.5211	-2.3011	1.1241

Training error: 33 / 274 (12.0438 %)

Test error: 15 / 68 (22.0588 %)

- Se observa que el de menor error empírico fue el "rmsprop" y el "adam".

Parte 6, 7, 8, 9

- Para encontrar de forma empírica la features más relevantes para softmax para conocer la especie de los pingüinos, se calcula el error empírico para dos pares de features. Utilizamos el método "adam" para softmax, para la minimización del error.

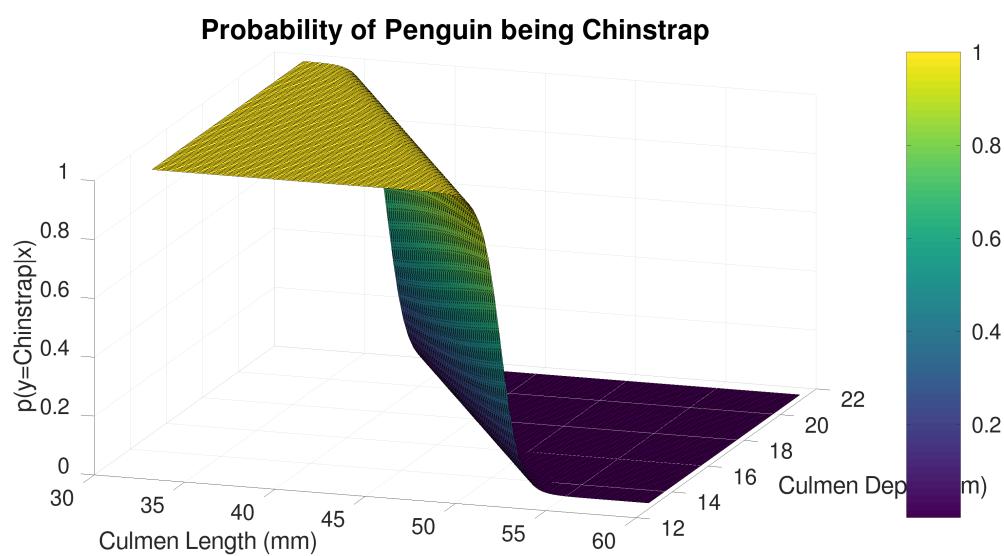
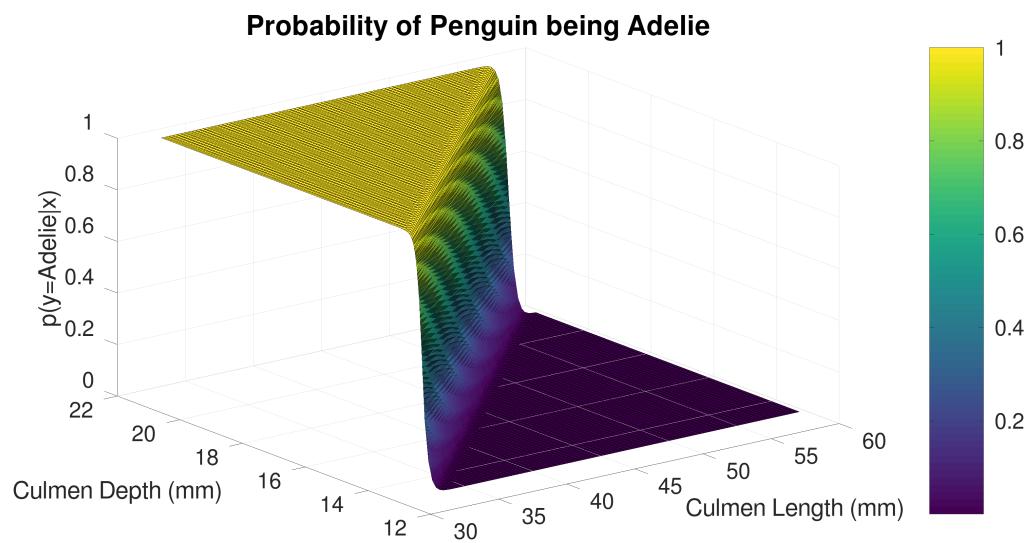
```
##### PART SEVEN - EIGHT - NINE - TEN #####
Combination: 1 2
100% =====
Test error: 13 / 68 (19.1176 %)
Combination: 1 3
100% =====
Test error: 40 / 68 (58.8235 %)
Combination: 1 4
100% =====
Test error: 16 / 68 (23.5294 %)
Combination: 1 5
100% =====
Test error: 13 / 68 (19.1176 %)
Combination: 2 3
100% =====
Test error: 10 / 68 (14.7059 %)
Combination: 2 4
100% =====
Test error: 20 / 68 (29.4118 %)
Combination: 2 5
100% =====
Test error: 10 / 68 (14.7059 %)
Combination: 3 4
100% =====
Test error: 35 / 68 (51.4706 %)
Combination: 3 5
100% =====
Test error: 10 / 68 (14.7059 %)
Combination: 4 5
100% =====
Test error: 10 / 68 (14.7059 %)

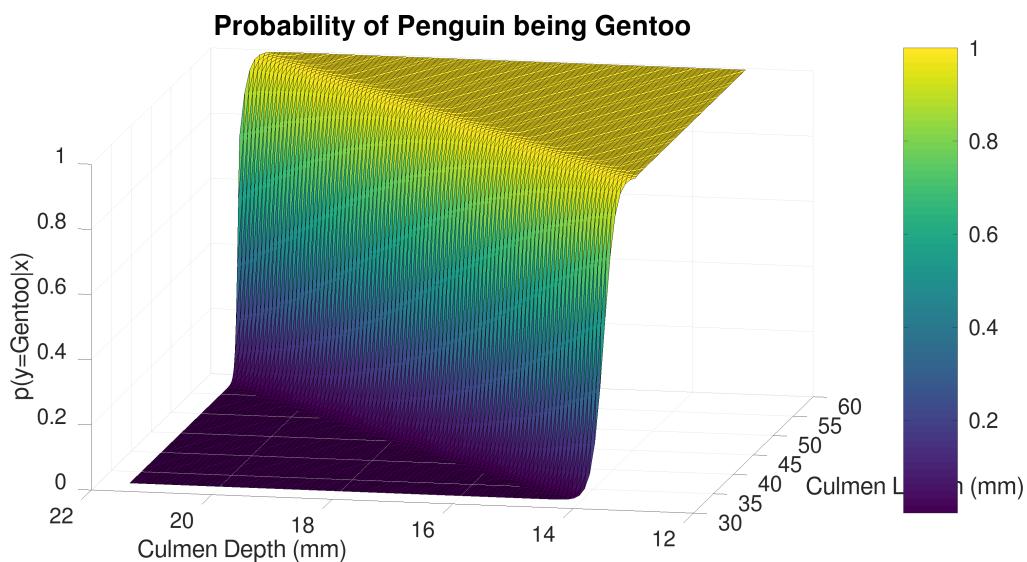
Best combination are:

Combination:
  2 3
[Culmen Length (mm), Culmen Depth (mm)]
percentage error: 14.7059 %
#####
```

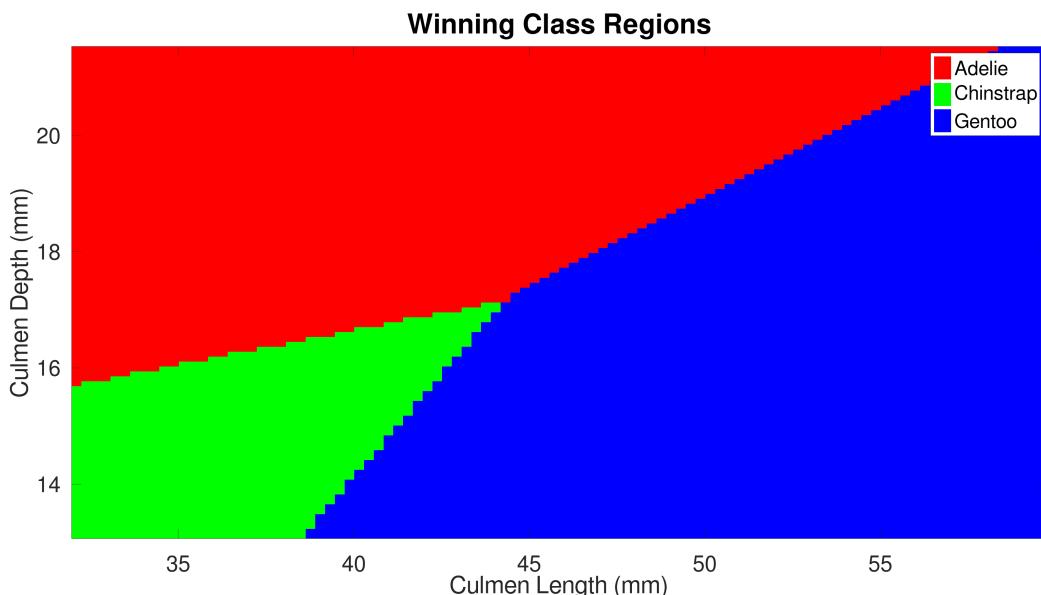
- Obtuvo que θ_3 “Culmen Depth (mm)” y θ_4 “Culmen Length (mm)” son las features más relevantes.

- Se dibuja la superficie generada por softmax de las features “Culmen Depth (mm)” y “Culmen Length (mm)”.

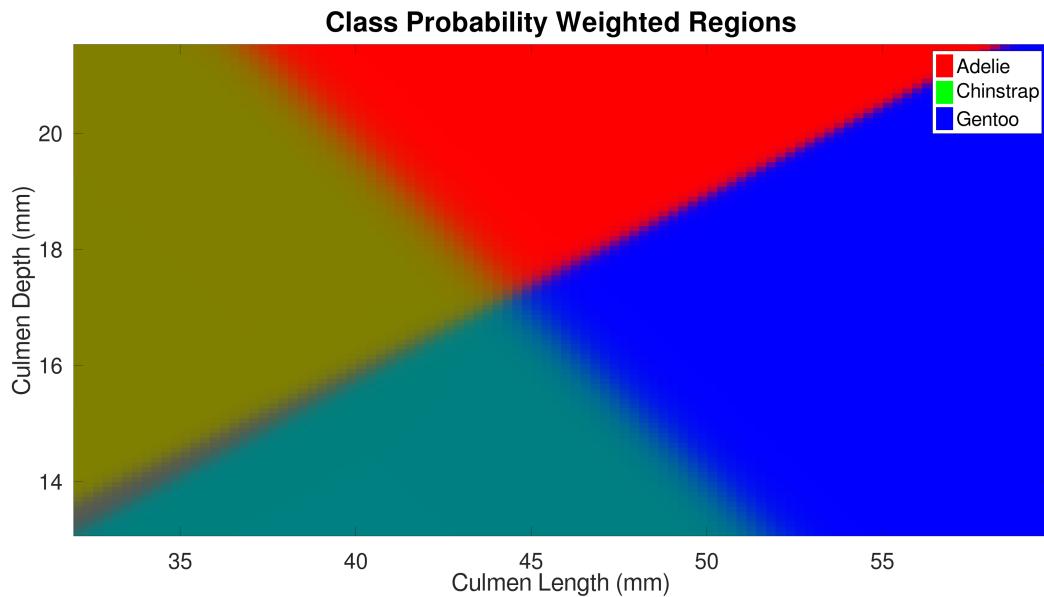




- Se gráfica la probabilidad del ganador de decisión de las features “Culmen Depth (mm)” y “Culmen Length (mm)”.

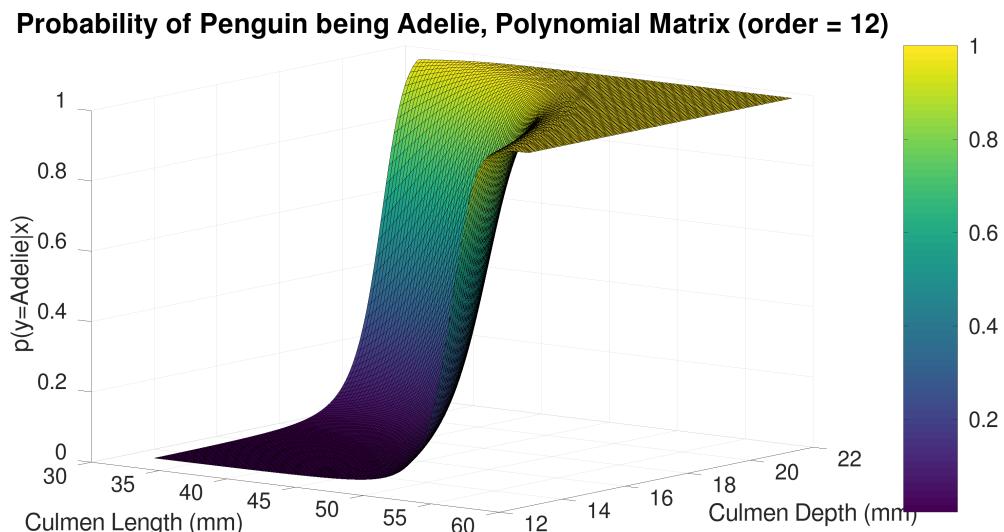


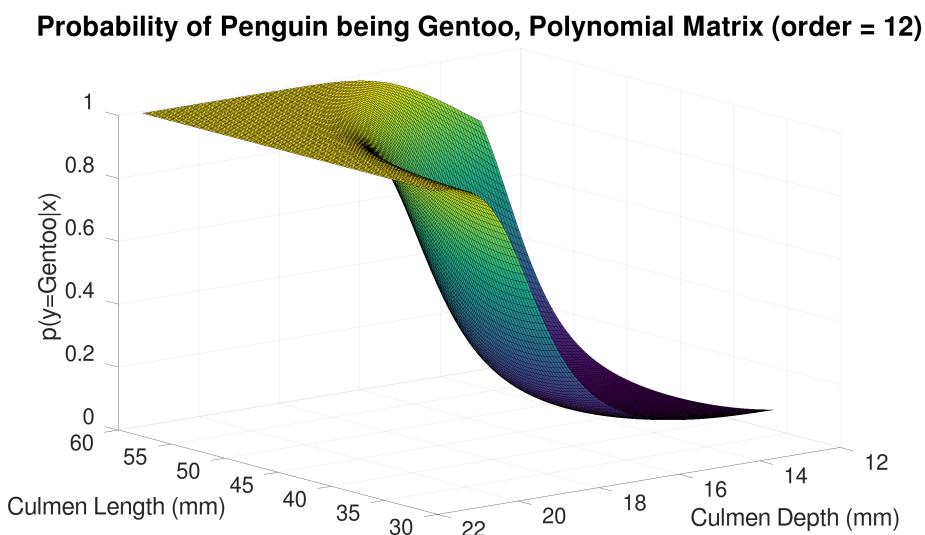
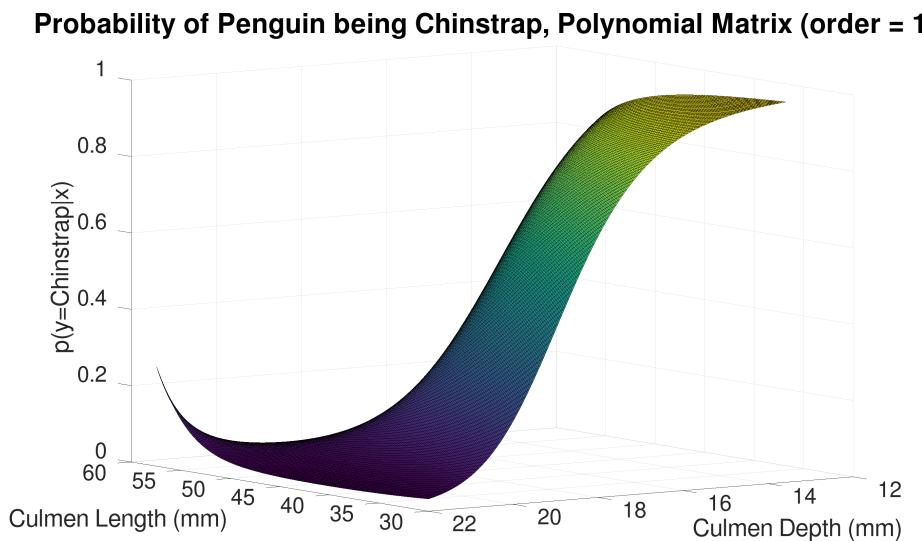
- Se gráfica la probabilidad del ganador con pesos de decisión de las features “Culmen Depth (mm)” y “Culmen Length (mm)”.



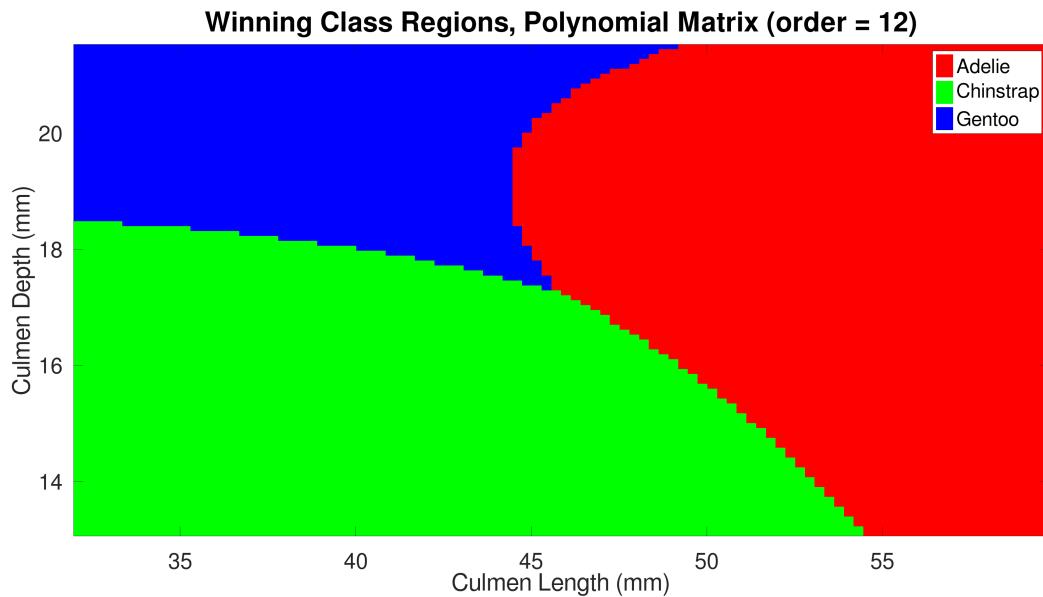
Parte 10

- Se dibuja la superficie generada por softmax de las features “Culmen Depth (mm)” y “Culmen Length (mm)”.

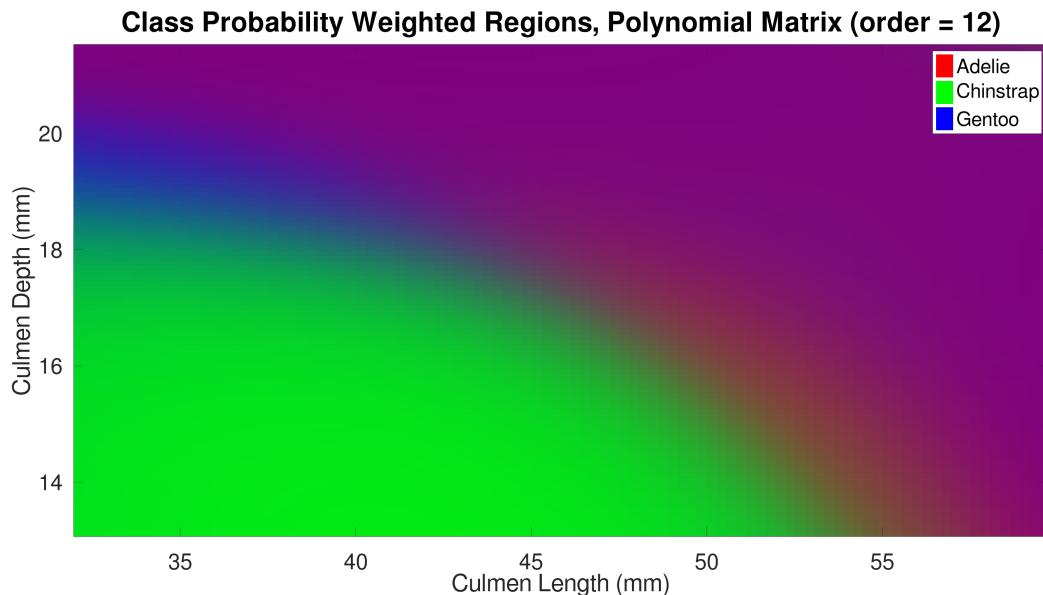




- Se gráfica la probabilidad del ganador de decisión de las features “Culmen Depth (mm)” y “Culmen Length (mm)”.



- Se gráfica la probabilidad del ganador con pesos de decisión de las features “Culmen Depth (mm)” y “Culmen Length (mm)”.



Integrantes

- Carlos Andrés Mata Calderón - 2019033834