

## segment-challenge

Al observar los datos podemos ver que todos son numéricos.

Viewer														
Relation: segment														
No.	1: region-centroid-col	2: region-centroid-row	3: region-pixel-count	4: short-line-density-5	5: short-line-density-2	6: vegde-mean	7: vegde-sd	8: hedge-mean	9: hedge-sd	10: intensity-mean	11: rawred-mean	12: rawblue-mean	13: rawgreen-mean	14: exred-r
	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	38.0	189.0	9.0	0.0	0.0	1.0	0.222222	6.22222	33.3185	29.0741	26.3333	35.2222	25.6667	-8.2
2	25.0	199.0	9.0	0.0	0.0	1.11111	0.607407	1.05556	0.462963	17.5185	13.1111	17.8889	21.5556	-13.
3	49.0	139.0	9.0	0.0	0.0	0.166667	0.077778	0.333333	0.080809	0.444444	0.0	1.33333	0.0	-1.3
4	63.0	220.0	9.0	0.0	0.0	3.05556	15.263	3.66667	6.08889	8.18519	6.55556	6.44444	11.5556	-4.8
5	161.0	135.0	9.0	0.0	0.0	0.055556	0.136083	0.111111	0.172133	1.25926	0.777778	3.0	0.0	-1.4
6	235.0	88.0	9.0	0.0	0.0	0.611111	0.240741	0.944445	0.32963	2.77778	0.444444	6.44444	1.44444	
7	67.0	32.0	9.0	0.0	0.0	0.944444	1.06284	1.77778	1.31092	126.222	115.111	142.222	121.333	-33.
8	188.0	182.0	9.0	0.0	0.0	1.61111	0.742868	4.16667	2.12655	58.0	51.8889	72.4444	49.6667	-18.
9	217.0	245.0	9.0	0.111111	0.111111	3.16667	3.01662	2.16667	1.24276	9.74074	7.44444	7.11111	14.6667	-6.8
10	9.0	171.0	9.0	0.0	0.0	1.5	1.00554	2.77778	1.64204	45.9259	41.0	57.2222	39.5556	-14.
11	149.0	117.0	9.0	0.222222	0.0	0.833333	0.255556	1.0	0.444445	21.2222	21.1111	26.3333	16.2222	-0.33
12	136.0	130.0	9.0	0.0	0.0	0.666667	0.177778	0.777778	0.307407	5.33333	6.55556	6.44444	3.0	3.6

Sería conveniente tenerlos como nominales

En los variables podemos observar

@atributo región-centroide-col numérico

@atributo región-centroide-fila numérico

@atributo región-píxel-recuento numérico

@attribute short-line-density-5 numérico

@attribute short-line-density-2 numérico

@attribute valor medio numérico

@attribute vegde-sd numérico

@attribute hedge-mean numérico

@attribute hedge-sd numérico

@attribute intensidad-media numérica

@attribute rawred-mean numérico

@attribute rawblue-mean numérico

@attribute rawgreen-mean numérico

@attribute exred-media numérica

@attribute exblue-mean numérico

@attribute exgreen-mean numérico

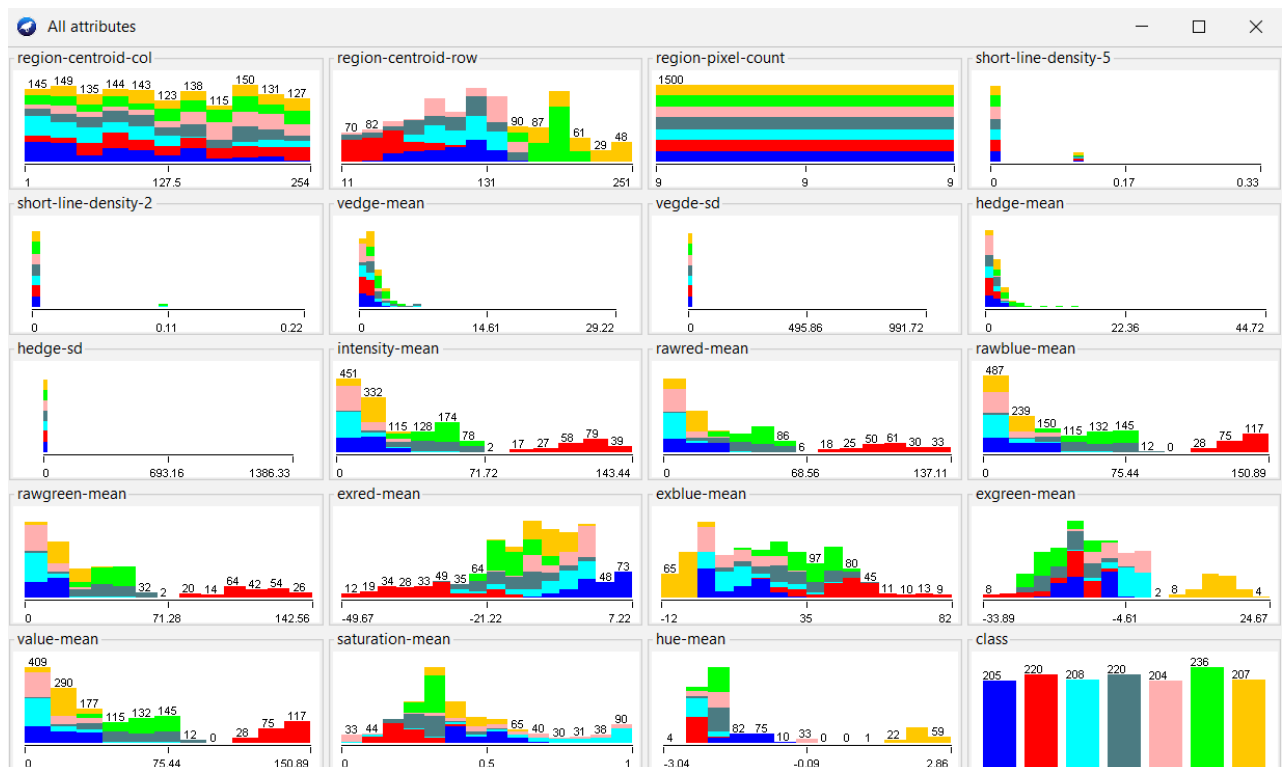
@attribute valor medio numérico

@attribute saturación media numérica

@attribute tonalidad media numérica

@attribute class {ladrillo,cielo,follaje,cemento,ventana,camino,hierba}

## Graficas de ayuda para la clasificación



### Modelo 1: J48

En este modelo podemos observar que tardo 0.06 segundo en clasificar.

Podemos observar que el modelo es bueno con un 95,73% instancias correctas, que clasifico como buenas 1436.

```
Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1436           95.7333 %
Incorrectly Classified Instances     64            4.2667 %
Kappa statistic                    0.9502
Mean absolute error                 0.0138
Root mean squared error             0.1057
Relative absolute error             5.6471 %
Root relative squared error         30.2115 %
Total Number of Instances          1500
```

Podemos observar las predicciones son muy buenas tienen tendencia a ser 1, podemos colaborar con la matriz de confusión que la diagonal está bien marcada, también su distribución a ambos lados de la diagonal son 27 que son las 64 que están menciona como clasificaciones incorrectas.

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,956	0,004	0,975	0,956	0,966	0,960	0,981	0,954	brickface
	1,000	0,001	0,995	1,000	0,998	0,997	1,000	0,995	sky
	0,942	0,018	0,895	0,942	0,918	0,905	0,975	0,889	foliage
	0,941	0,009	0,945	0,941	0,943	0,933	0,978	0,946	cement
	0,877	0,017	0,891	0,877	0,884	0,866	0,961	0,881	window
	0,987	0,001	0,996	0,987	0,991	0,990	0,997	0,992	path
	0,990	0,000	1,000	0,990	0,995	0,994	1,000	1,000	grass
Weighted Avg.	0,957	0,007	0,958	0,957	0,957	0,951	0,985	0,952	

=== Confusion Matrix ===

a	b	c	d	e	f	g	<-- classified as
196	0	3	1	5	0	0	a = brickface
0	220	0	0	0	0	0	b = sky
0	1	196	2	9	0	0	c = foliage
2	0	4	207	6	1	0	d = cement
3	0	16	6	179	0	0	e = window
0	0	0	3	0	233	0	f = path
0	0	0	0	2	0	205	g = grass

## Mejora del modelo J48

Si trabajamos con un porcentaje del 80% el modelo mejora mucho mas

Correctly Classified Instances	290	96.6667 %
Incorrectly Classified Instances	10	3.3333 %

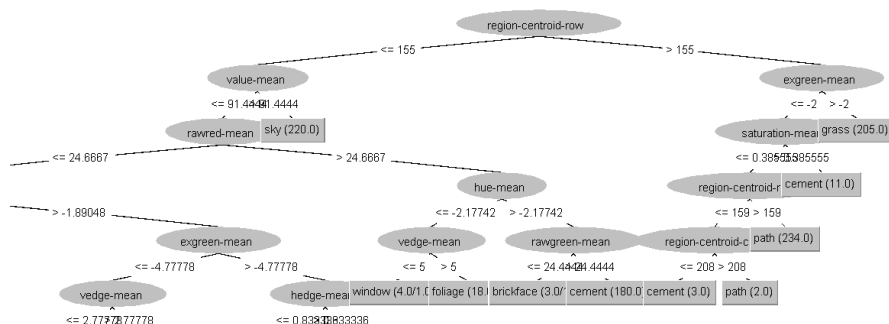
En la matriz de confusión, solo quedan 10 incorrectas

=== Confusion Matrix ===

a	b	c	d	e	f	g	<-- classified as
40	0	0	0	1	0	0	a = brickface
0	45	0	0	0	0	0	b = sky
0	0	38	0	4	0	0	c = foliage
0	0	3	51	0	1	0	d = cement
1	0	0	0	33	0	0	e = window
0	0	0	0	0	45	0	f = path
0	0	0	0	0	0	38	g = grass

En el árbol de decisiones, las variables que dan más peso al modelo

Podemos decir si tenemos un valor en la región centroeide col (region-centroid-col) y es menor o igual 155, se considera como valor medio (value\_mean) y si es mayor a 155 se considera que esta en el medio exverde (exgreen-mean),



## Modelo 2: MultilayerPerceptron

Este modelo a tardado 2.63 segundos en clasificar, es más lento al anterior (J48), las observaciones de este modelo son mejores a la anterior con un 97,067% instancias correctas, clasifico como buenas 1436.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1456           97.0667 %
Incorrectly Classified Instances    44           2.9333 %
Kappa statistic                    0.9658
Mean absolute error                0.0135
Root mean squared error            0.0841
Relative absolute error            5.5049 %
Root relative squared error        24.04 %
Total Number of Instances         1500
```

En el Recall podemos observar la cantidad de clases que podemos identificar y son muy buenas, si vemos la media de F1(F-measure) no hay discrepancias notorias.

=== Detailed Accuracy By Class ===

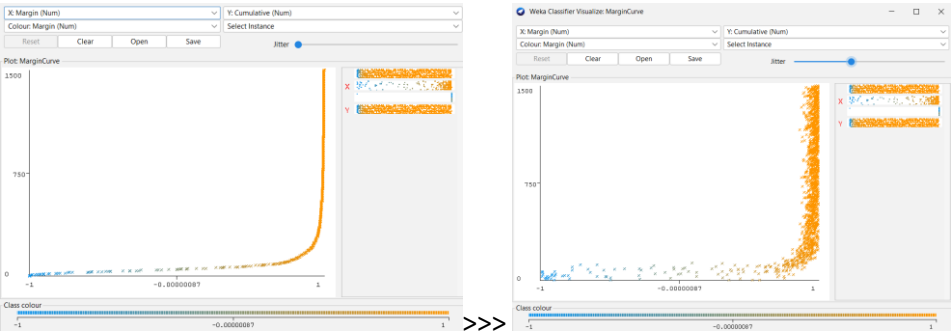
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,980	0,002	0,990	0,980	0,985	0,983	0,999	0,996	brickface
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	sky
	0,928	0,010	0,937	0,928	0,932	0,922	0,989	0,959	foliage
	0,977	0,011	0,939	0,977	0,958	0,951	0,994	0,980	cement
	0,902	0,011	0,929	0,902	0,915	0,902	0,991	0,955	window
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	path
	1,000	0,001	0,995	1,000	0,998	0,997	1,000	1,000	grass
Weighted Avg.	0,971	0,005	0,971	0,971	0,971	0,966	0,996	0,985	

La matriz de confusión, vemos que la diagonal está bien marcada mucho mejor que la J48, En la matriz de confusión, solo quedan 44 incorrectas

=== Confusion Matrix ===

	a	b	c	d	e	f	g	<-- classified as
201	0	0	3	1	0	0	0	a = brickface
0	220	0	0	0	0	0	0	b = sky
0	0	193	6	9	0	0	0	c = foliage
0	0	1	215	4	0	0	0	d = cement
2	0	12	5	184	0	1	1	e = window
0	0	0	0	0	236	0	0	f = path
0	0	0	0	0	0	0	207	g = grass

En la curva de márgenes, podemos visualizar que afecta poco el ruido aumentar, al aumentar los valores al modelo.



Mejora del modelo **MultilayerPerceptron**

Si trabajamos con un porcentaje del 80% el modelo mejora mucho mas

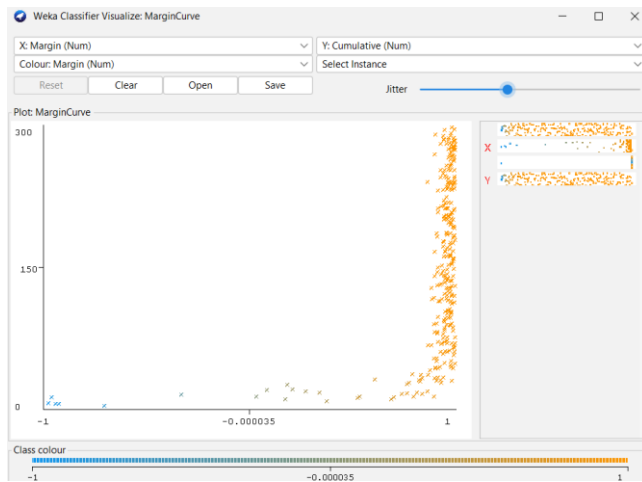
Correctly Classified Instances	294	98	%
Incorrectly Classified Instances	6	2	%

Ahora ya estamos trabajando con una precisión del 98%, obteniendo 294 clasificados como buenos.

En la matriz de confusión, solo quedan 6 incorrectas

```
=== Confusion Matrix ===
  a  b  c  d  e  f  g  <-- classified as
41  0  0  0  0  0  0 | a = brickface
 0 45  0  0  0  0  0 | b = sky
 0  0 37  0  5  0  0 | c = foliage
 0  0  0 54  1  0  0 | d = cement
 0  0  0  0 34  0  0 | e = window
 0  0  0  0  0 45  0 | f = path
 0  0  0  0  0  0 38 | g = grass
```

Pero aun al aumentar valores al modelo afecta el ruido.



### Modelo 3: RandomForest

El modelo es más rápido que el modelo MultilayerPerceptron porque tardo 0.29 segundos, pero no más rápido que J48, en cuanto a las predicciones inicialmente es el mejor con un 97.86 % y 1468 que clasificó como correctas,

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1468           97.8667 %
Incorrectly Classified Instances     32            2.1333 %
```

Tanto predicision y recall, están demasiado cerca de la media F1(F-measure) las discrepancias son mínimas.

```
=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,990	0,001	0,995	0,990	0,993	0,992	1,000	1,000	brickface
	1,000	0,001	0,995	1,000	0,998	0,997	1,000	1,000	sky
	0,966	0,009	0,948	0,966	0,957	0,950	0,998	0,988	foliage
	0,968	0,006	0,964	0,968	0,966	0,960	0,998	0,993	cement
	0,926	0,008	0,950	0,926	0,938	0,928	0,997	0,983	window
	0,996	0,001	0,996	0,996	0,996	0,995	1,000	1,000	path
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	grass
Weighted Avg.	0,979	0,003	0,979	0,979	0,979	0,975	0,999	0,995	

En matriz de confusión, vemos que la diagonal que está bien marcada mucho mejor que la J48 y MultilayerPerceptron, solo quedan 32 incorrectas

```

=== Confusion Matrix ===
      a  b  c  d  e  f  g  <-- classified as
203  0  0  0  2  0  0 | a = brickface
  0 220  0  0  0  0  0 | b = sky
  0  1 201  2  4  0  0 | c = foliage
  0  0  2 213  4  1  0 | d = cement
  1  0  9  5 189  0  0 | e = window
  0  0  0  1  0 235  0 | f = path
  0  0  0  0  0  0 207 | g = grass

```

## Mejora de modelo: RandomForest

Se puede decir que en un muestreo de un 80%, tenemos una clasificación correcta del 99.33% es decir 298 correctamente clasificados.

```

=== Summary ===

Correctly Classified Instances      298          99.3333 %
Incorrectly Classified Instances     2           0.6667 %
Kappa statistic                    0.9922
Mean absolute error                 0.019
Root mean squared error             0.0709
Relative absolute error             7.7509 %
Root relative squared error        20.246 %
Total Number of Instances          300

```

```

=== Detailed Accuracy By Class ===

```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	brickface
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	sky
	0,952	0,000	1,000	0,952	0,976	0,972	0,997	0,989	foliage
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	cement
	1,000	0,008	0,944	1,000	0,971	0,968	0,997	0,966	window
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	path
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	grass
Weighted Avg.	0,993	0,001	0,994	0,993	0,993	0,992	0,999	0,995	

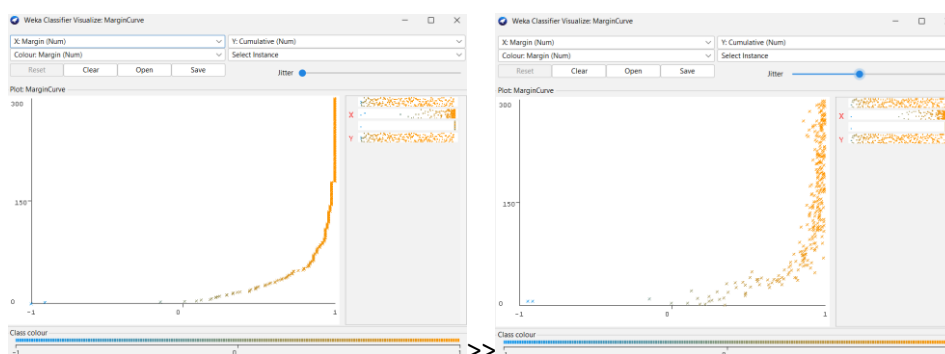
En matriz de confusión, tenemos una diagonal mejor que la J48 y MultilayerPerceptron y solo quedan 2 incorrectas

```

=== Confusion Matrix ===
      a  b  c  d  e  f  g  <-- classified as
41  0  0  0  0  0  0 | a = brickface
  0 45  0  0  0  0  0 | b = sky
  0  0 40  0  2  0  0 | c = foliage
  0  0  0 55  0  0  0 | d = cement
  0  0  0  0 34  0  0 | e = window
  0  0  0  0  0 45  0 | f = path
  0  0  0  0  0  0 38 | g = grass

```

Al aumentar datos al modelo aun afecta el ruido que afecta, todos los modelos quedan afectados por el ruido.



## Conclusiones

Modelo	Clasificaciones correctas antes de la mejor	Clasificaciones correctas después de la mejor	Mejor tiempo, al clasificación
J48	95,73%	96,67%	0.06 segundo
Multilayer Perceptron	97,07%	98,00%	2.63 segundos
Random Forest	97,867%	99,33%	0.29 segundos

El modelo que se acomoda a nuestros datos es el Random Forest, porque en la mejora obtuvo la mayor clasificación de datos correctos y los datos de la precisión y recall esta demasiado cerca de la media F1(F-measure).

En todos los modelos al aumentar dato son afectado por el ruido.