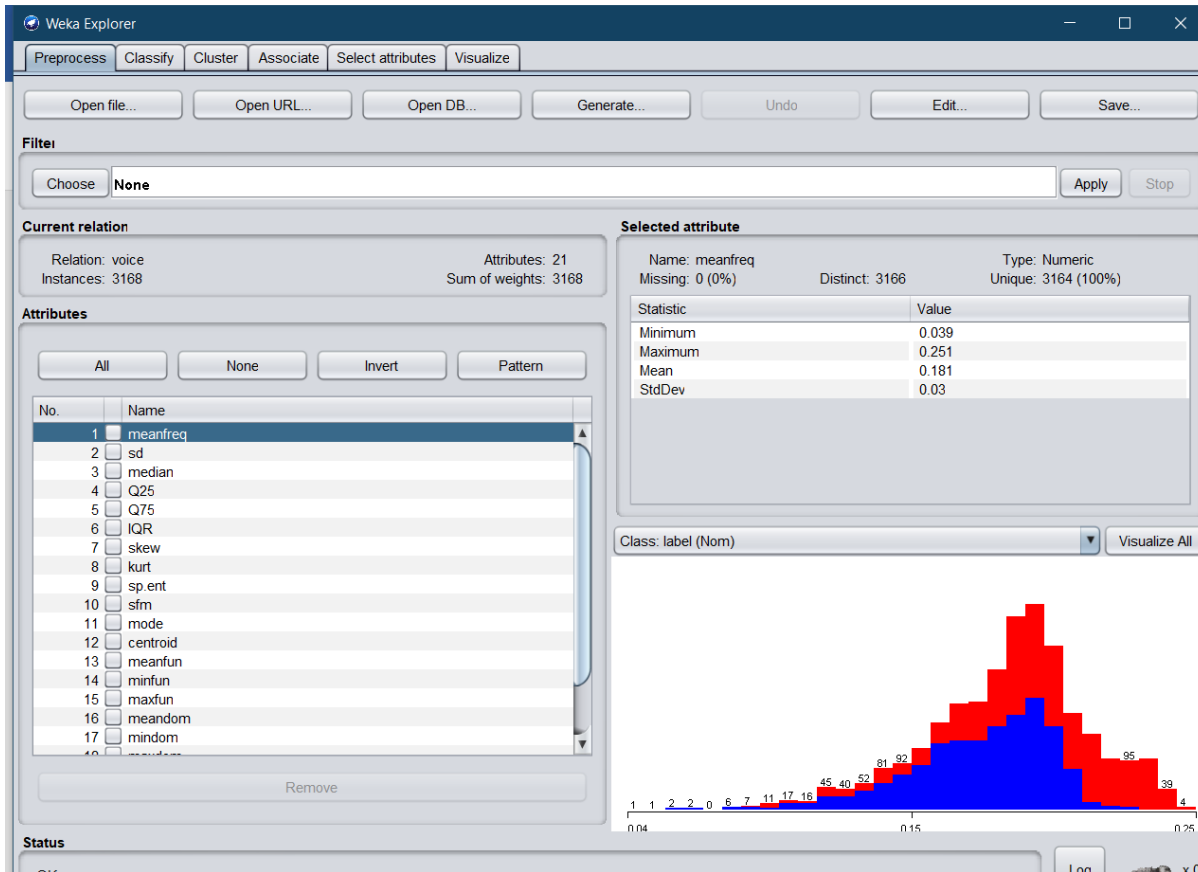


1. Abriendo el voice.csv seleccionado

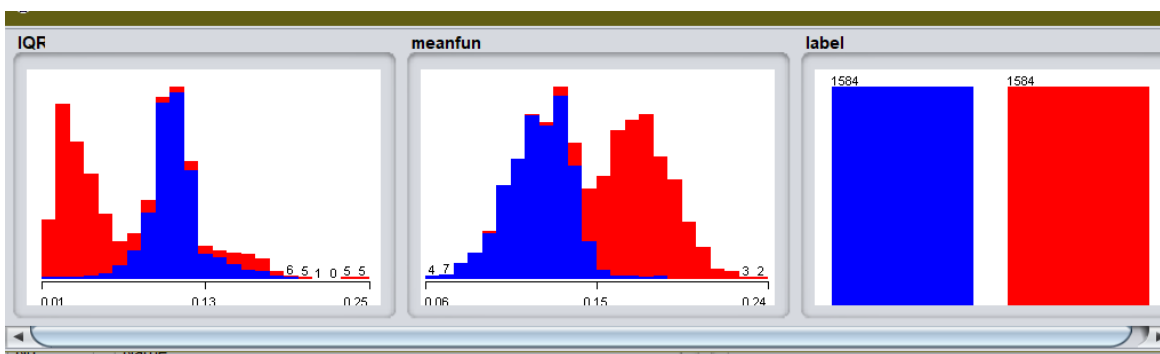


2. Algoritmos de Preprocesamiento (3 Supervisado)

Attribute Selection

Entre los subconjuntos de atributos calcula el grado de redundancia y capacidad predictiva individual, (escalada codiciosa aumentada con una función de retroceso.) backtracking

Aplicado el filtro tenemos los atributos con mas importancia (perdida de precisión) del dataset:



Siendo el IQR y meanfun los mas relevantes a la hora de revisión de algún clasificador. Comparando con los atributos eliminados

ClassConditionalValues

Simplemente separa los atributos por el tipo de clase es decir en este dataset se usa de output target 'Masculino' y 'Femenino'

1	<input type="checkbox"/>	pr_meanfreq male
2	<input type="checkbox"/>	pr_meanfreq female
3	<input type="checkbox"/>	pr_sd male
4	<input type="checkbox"/>	pr_sd female
5	<input type="checkbox"/>	pr_median male
6	<input type="checkbox"/>	pr_median female
7	<input type="checkbox"/>	pr_Q25 male
8	<input type="checkbox"/>	pr_Q25 female
9	<input type="checkbox"/>	pr_Q75 male
10	<input type="checkbox"/>	pr_Q75 female
11	<input type="checkbox"/>	pr_IQR male
12	<input type="checkbox"/>	pr_IQR female
13	<input type="checkbox"/>	pr_skew male
14	<input type="checkbox"/>	pr_skew female
15	<input type="checkbox"/>	pr_kurt male
16	<input type="checkbox"/>	pr_kurt female
17	<input type="checkbox"/>	pr_sp.ent male
18	<input type="checkbox"/>	pr_sp.ent female

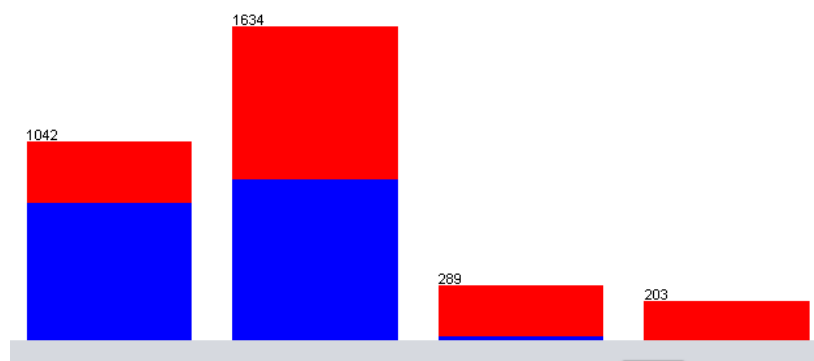
Creando valores nominales de los numéricos usando el target output o la clase.

Discretize

Convertir los datos numéricos en nominales categóricos, en este dataset aplico división por intervalo del mismo tamaño como se ve la imagen

Selected attribute			
Name: meanfreq		Type: Nominal	
Missing: 0 (0%)		Distinct: 4	
		Unique: 0 (0%)	
No.	Label	Count	Weight
1	'(-inf-0.171857]'	1042	1042.0
2	'(0.171857-0.208754]'	1634	1634.0
3	'(0.208754-0.225625]'	289	289.0
4	'(0.225625-inf)'	203	203.0

Class: label (Nom) Visualize All



Sin supervisión (3)

Por ser Weka una herramienta de data mining, los algoritmos para preprocesamiento son más amplios

Normalize

Este es uno de los mas usados según el algoritmo de clasificación que se use, el único que conozco que actúa mejor con atributos normalizados es el kNN, simplemente le dará un rango a todas las variables de forma que no tengan mas relevancia si los valores son mayores

No.	1: meanfreq	2: sd	3: median	4: Q25	5: Q75	6: IQR	7: skew	8: kurt	9: sp.ent	10: sfm	11: mode	12: centroid	13: meanfun	14: minfun	15: maxfun	16: meandom	17: nui
1	0.0964185...	0.47...	0.08412...	0.06...	0.20...	0.25...	0.367...	0.20...	0.635...	0.56...	0.0	0.096418...	0.1577063...	0.03050...	0.981525...	0.0	0.006...
2	0.1258280...	0.50...	0.11689...	0.07...	0.21...	0.24...	0.644...	0.48...	0.630...	0.59...	0.0	0.125828...	0.2876415...	0.03114...	0.834599...	4.07449535...	0.006...
3	0.1792221...	0.67...	0.10287...	0.03...	0.38...	0.45...	0.895...	0.78...	0.442...	0.54...	0.0	0.179222...	0.2369454...	0.03026...	0.954962...	6.01914086...	0.006...
4	0.5282609...	0.55...	0.58755...	0.38...	0.71...	0.40...	0.031...	0.00...	0.923...	0.85...	0.2995...	0.528260...	0.1834421...	0.04128...	0.834599...	0.06565879...	0.006...
5	0.4521952...	0.62...	0.45427...	0.31...	0.70...	0.47...	0.027...	0.00...	0.958...	0.92...	0.3723...	0.452195...	0.2791904...	0.03682...	0.929285...	0.23899359...	0.006...
6	0.4411734...	0.63...	0.43202...	0.27...	0.72...	0.53...	0.051...	0.00...	0.922...	0.87...	0.4019...	0.441173...	0.2996993...	0.03776...	0.857144...	0.09844823...	0.006...
7	0.5260614...	0.57...	0.59593...	0.37...	0.70...	0.41...	0.040...	0.00...	0.940...	0.90...	0.3078...	0.526061...	0.2767006...	0.08468...	0.929285...	0.15994165...	0.006...
8	0.5721134...	0.60...	0.53291...	0.44...	0.81...	0.44...	0.036...	0.00...	0.906...	0.84...	0.4583...	0.572113...	0.2058926...	0.04108...	0.233218...	0.09950499...	0.006...
9	0.4858135...	0.61...	0.50994...	0.35...	0.71...	0.44...	0.027...	0.00...	0.953...	0.91...	0.7825...	0.485813...	0.2260853...	0.04210...	0.834599...	0.11141637...	0.006...
10	0.4484569...	0.63...	0.44146...	0.30...	0.68...	0.47...	0.030...	0.00...	0.972...	0.95...	0.0417...	0.448456...	0.2763513...	0.04902...	0.904449...	0.11273449...	0.023...
11	0.5556145...	0.55...	0.62811...	0.40...	0.75...	0.42...	0.024...	0.00...	0.931...	0.86...	0.3441...	0.555614...	0.1830529...	0.06327...	0.082684...	0.15336770...	0.006...
12	0.4683933...	0.60...	0.46574...	0.35...	0.69...	0.42...	0.042...	0.00...	0.934...	0.88...	0.0432...	0.468393...	0.2671153...	0.04818...	0.904449...	0.08077687...	0.006...
13	0.4626898...	0.64...	0.45269...	0.33...	0.72...	0.46...	0.035...	0.00...	0.924...	0.86...	0.3872...	0.462689...	0.2036494...	0.03609...	0.626291...	0.16063697...	0.023...
14	0.6699180...	0.43...	0.71919...	0.52...	0.80...	0.36...	0.035...	0.00...	0.816...	0.62...	0.7850...	0.669918...	0.4170810...	0.07835...	0.981525...	0.43029067...	0.006...
15	0.6788423...	0.50...	0.72031...	0.52...	0.85...	0.40...	0.099...	0.02...	0.828...	0.66...	0.1785...	0.678842...	0.2594226...	0.05691...	0.981525...	0.41965450...	0.436...
16	0.6370820...	0.52...	0.71887...	0.46...	0.80...	0.41...	0.125...	0.04...	0.872...	0.74...	0.1786...	0.637082...	0.2552884...	0.04401...	0.812749...	0.54896883...	0.006...
17	0.7153506...	0.48...	0.78711...	0.53...	0.87...	0.41...	0.041...	0.00...	0.821...	0.62...	0.1790...	0.715350...	0.3172248...	0.03998...	0.981525...	0.48351357...	0.006...
18	0.6227964...	0.58...	0.56676...	0.49...	0.87...	0.44...	0.088...	0.01...	0.814...	0.68...	0.2141...	0.622796...	0.1326568...	0.03024...	0.904449...	0.02328011...	0.006...
19	0.6090968...	0.57...	0.53803...	0.46...	0.85...	0.46...	0.074...	0.01...	0.804...	0.64...	0.2144...	0.609096...	0.1533394...	0.03...	0.981525...	0.06279323...	0.006...
20	0.6340563...	0.56...	0.56980...	0.49...	0.87...	0.44...	0.077...	0.01...	0.790...	0.59...	0.2143...	0.634056...	0.1898390...	0.03050...	0.610344...	0.06279323...	0.006...
21	0.6299144...	0.60...	0.66638...	0.48...	0.87...	0.46...	0.081...	0.01...	0.767...	0.60...	0.2141...	0.629914...	0.2087545...	0.03081...	0.550543...	0.07731354...	0.006...
22	0.6689260...	0.57...	0.63266...	0.51...	0.91...	0.46...	0.070...	0.00...	0.725...	0.54...	0.2141...	0.668926...	0.2366002...	0.03278...	0.981525...	0.06848819...	0.006...
23	0.5863820...	0.55...	0.53773...	0.46...	0.80...	0.41...	0.099...	0.02...	0.774...	0.62...	0.2140...	0.586382...	0.0383163...	0.03042...	0.536515...	0.01756321...	0.006...
24	0.6179127...	0.58...	0.53977...	0.50...	0.89...	0.46...	0.077...	0.00...	0.719...	0.55...	0.2140...	0.617912...	0.1215628...	0.03050...	0.509474...	0.03178106...	0.006...
25	0.5716797...	0.60...	0.53486...	0.48...	0.84...	0.42...	0.176...	0.06...	0.798...	0.65...	0.2145...	0.571679...	0.2382520...	0.03253...	0.981525...	0.06744150...	0.006...
26	0.5918787...	0.58...	0.54363...	0.47...	0.85...	0.45...	0.117...	0.03...	0.830...	0.70...	0.2141...	0.591878...	0.1504756...	0.03018...	0.857144...	0.04594801...	0.006...
27	0.6149187...	0.59...	0.70127...	0.47...	0.84...	0.45...	0.119...	0.03...	0.784...	0.62...	0.2141...	0.614918...	0.1476674...	0.03312...	0.954962...	0.04767159...	0.006...
28	0.6122846...	0.55...	0.52824...	0.50...	0.89...	0.45...	0.084...	0.00...	0.672...	0.54...	0.4576...	0.612284...	0.4121082...	0.03122...	0.694571...	0.11102184...	0.006...
29	0.6043475...	0.56...	0.52253...	0.49...	0.95...	0.43...	0.059...	0.00...	0.719...	0.62...	0.4813...	0.604347...	0.3541850...	0.03329...	0.904449...	0.09860278...	0.006...

NumericCleaner

Este hará que los valores demasiado pequeños sean ignorados, eliminándolos de los examples. Como se ve en la imagen eliminará valores demasiado pequeños que no ‘afectan’ en gran medida la predicción

PKIDiscretize:

Esta forma de discretización es una ‘mejora’ del método que mostré más arriba, ahora con este se puede hacer binning de atributos con la misma frecuencia en sus examples como lo dice ‘obliga a que el número de bins sea igual a la raíz cuadrada del número de valores del atributo numérico.’ Su comportamiento como se espera será entonces

