PRÁCTICA 2:

COMPARACIÓN DE ALGORITMOS DE APRENDIZAJE

MINERÍA DE DATOS

MIGUEL CHAVEINTE GARCÍA

1. RESUMEN EJECUTIVO DE LOS CONJUNTOS DE DATOS

- Soybean.arff:
  683 instancias, 35 atributos + 1 de clase, todos ellos nominales. La clase puede tomar 19 valores: {diaporthe-stem-canker, charcoal-rot, rhizoctonia-root-rot, phytophthora-rot, brown-stem-rot, powdery-mildew,downy-mildew, brown-spot, bacterial-blight,bacterial-pustule, purple-seed-stain, anthracnose,phyllosticta-leaf-spot, alternarialeaf-spot,frog-eye-leaf-spot, diaporthe-pod-&-stem-blight, cyst-nematode, 2-4-d-injury, herbicide-injury.}

  La distribución de la clase es de un 2,92% para diaporthe-stem-canker, charcoal-rot, rhizoctonia-root-rot, powdery-mildew, downy-mildew, bacterial-blight, bacterial-pustule, purple-seed-stain y phyllosticta-leaf-spot. Un 6,44% para brown-stem-rot, anthracnose. Un 12,88% para phytophthora-rot. Un 13,46% para brown-spot, y alternarialeaf-spot, frog-eye-leaf-spot un 13,32%; 1,17% para herbicide-injury, 2,04% para cyst-nematode, 2,19% y 2,34% para diaporthe-pod-&-stem-blight y 2-4-d-injury respectivamente.

- Vote.arff:
  435 instancias, 16 atributos + 1 de clase, todos ellos nominales. Los 16 atributos pueden tomar dos valores: 'y' (yes) o 'n' (no). La clase puede tomar dos valores: democrat y republican.

  La distribución de la clase es 54,25% para democrat, y 38,62% para republican.

- Labor.arff:
  57 instancias, 16 atributos + 1 de clase.Atributos numéricos: duration,wage-increase-first-year,wage-increase-third-year,working-hours,standby-pay,shift-differential,statutory-holidays. Atributos Nominales: cost-of-living-adjustment, pension,education-allowance, vacation, longterm-disability-assistance,contribution-to-dental-plan,bereavement-assistance,contribution-to-health-plan . La clase de tipo Nominal que puede tomar 2 posibles valores: bad y good.

  La distribución de clase es 35,09% bad, y 64,91% good.

- Ionosphere.arff:
  351 instancias, 34 atributos +1 de clase, todos ellos de tipo numérico. La clase es de tipo nominal y puede tomar dos valores: b y g.
  Distribución de 35,9% para b y 64,1% para g
- Diabetes.arff:
  768 instancias y 8 atributos+ 1 de clase, todos ellos de atributo numérico. La clase es de tipo nominal y puede tomar 2 valores: tested_negative y tested_positive.
  La distribución es 65,1% para tested_negative y 34,9% para tested_positive.
- Glass.arff:
  214 instancias, 9 atributos +1 de clase. Todos ellos de tipo numérico. La clase es nominal y puede tomar siete valores {build wind float, build wind non-float, vehic wind float, vehic wind non-float, containers, tableware, headlamps}.

La distribución es build wind float (32,7 %), build wind non-float (35,5 %), vehic wind float (7,9 %), vehic wind non-float (0 %), containers (6,1 %), tableware (4,2 %) y headlamps (13,6 %)

- Segment-test.arff:
  810 instancias y 19 atributos+1 de clase, todos ellos de tipo numéricos. La clase es de tipo nominal con 7 posibles valores: {brickface, sky, foliage, cement, window, path, grass}.

  La distribución es brickface (15,4 %), sky (13,6 %), foliage (15,1 %), cement t (13,6 %), window (15,5 %), path (11,6 %) y grass (15,2 %).

- Breast Cancer.arff:
  286 instancias y 9 atributos + 1 de clase, todos ellos de tipo nominal. La clase también de tipo nominal y puede tomar dos valores: no-recurrence-events y recurrence-events.
  La distribución es un 70,3% de no-recurrence-events y 29,7% de recurrence-events.
- Credit-g.arff:
  1000 instancias y 20 atributos + 1 de clase. 12 de ellos de tipo nominal y 7 de tipo numérico. La clase es de tipo nominal y puede tomar dos valores: good y bad.
  La distribución es 70% good y 30% bad.
- Iris.arff:
  150 instancias y 4 atributos + 1 de clase, todos ellos de tipo numérico. La clase es de tip nominal y puede tomar tres valores : Iris-setosa, Iris-versicolor e Iris-virginica.
  La distribución es del 33,33% para cada uno de los valores de la clase.
- Tmusic.arff:
  211 instancias, 18 atributos + 1 de clase, todos ellos de tipo numérico. La clase es de tipo nominal y puede tomar 4 valores : {R1,R2,R3, y R4}.
  La distribución es 42,6 para R1, 18,5% para R2 , 18% para R3 y 20,9% para R4.
- Thoracic-Surgery.arff:
  470 instancias y 16 atributos + 1 de clase, 3 numéricos y 12 de tipo nominal. La clase es de tipo nominal y puede tomar dos valores: T y F.
  En cuanto a la distribución es 14,9% para T y 85,1% para F.

  En todos estos conjuntos de datos aplicamos en los que fueran necesario el filtro de ReplaceMissingValues, con el que eliminamos las instancias de valores ausentes. Además escalamos los datos mediante la normalización en los atributos numéricos, convirtiéndolos en el rango [0,1].

2. COMPARACIÓN 2 MÉTODOS MISMO CONJUNTO DE DATOS: TEST DE MCNEMAR

Con el filtro no supervisado resample, como venía en el guión de la práctica, separamos el conjunto es test y training. Obteniendo 228 instancias de test y 455 de training.

Una vez sacado en un csv los resultados de clasificación de las instancias para OneR y J4.8, ejecutamos el script de Python creado para calcular los valores de McNemar (Consideramos A=J4.8 y B=OneR)

mcnemar.pdf

(Ver script al final del informe)

Tabla de contingencia resultante:

McNemar	Mal clasificados por h_B		Bien clasificados por h_B	
Mal clasificados por h_A	12	$\mathbf{n_{00}}$	2	$\mathbf{n_{01}}$
Bien clasificados por h_A	147	$\mathbf{n_{10}}$	67	$\mathbf{n_{11}}$

Bajo la hipótesis nula, los dos algoritmos deben tener la misma tasa de error $n_{01} = n_{10}$, que no se cumple en este caso.
Aplicamos el test de McNemar, que es aplicable si $n_{01} + n_{10} > 25$ que se cumple en nuestro caso.
El estadístico $\frac{(|n01-n10|-1)\text{^}2}{n01+n10}$ =139.17, distribuyéndose como una chi cuadra con un grado de libertad y confianza del 95%.
Como el estadístico es mayor que 3,841459 se rechaza la hipótesis nula con una confianza del 95%

3. COMPARACIÓN 2 MÉTODOS MISMO CONJUNTO DE DATOS: TEST DE STUDENT CON VALIDACIÓN CRUZADA(CORREGIDO)

○ Test de Student Pareado

- Algoritmo base NB:

Tasa de acierto %			
NB	J48	IB1	SVM
92.08	92.39	91.64	93.85

- Algoritmo base SVM :

Tasa de acierto %			
NB	J48	IB1	SVM
93,85	92.39	91.64	92,08

○ Test de Student Pareado Corregido

- Algoritmo base NB:

Tasa de acierto %			
NB	J48	IB1	SVM

92.08	92.39	91.64	93.85

- Algoritmo base SVM :

Tasa de acierto %			
NB	J48	IB1	SVM
93,85	92.39	91.64	92,08

4. COMPARACIÓN 2 MÉTODOS MISMO CONJUNTO DE DATOS: TEST DE STUDENT CON REPETICIÓN EN VALIDACIÓN CRUZADA(CORREGIDO)

- Test de Student Pareado

- Algoritmo base NB:

Tasa de acierto %			
NB	J48	IB1	SVM
92.20	92.63	91.35	93.10

- Algoritmo base SVM :

Tasa de acierto %			
NB	J48	IB1	SVM
93,10	92.63	91.35	92,20

- Test de Student Pareado Corregido

- Algoritmo base NB:

Tasa de acierto %			
NB	J48	IB1	SVM
92.20	92.63	91.35	93.10

- Algoritmo base SVM :

Tasa de acierto %			
NB	J48	IB1	SVM
93.10	92.63	91.35	92,20

Resultados similares que con la validación cruzada sin repetición. En el anterior funcionaban muy bien J4.8 y SVM; con reptición destaca el SVM.

5. DOS MÉTODOS, VARIOS CONJUNTOS DE DATOS: TEST DE SIGNOS

Error (alfa=0.05)		
	OneR	**J48**
Soybean	66.47	7.61
Vote	4.36	3.67
Labor	28.00	18.33
Ionosphere	19.08	8.54

Diabetes	28.52	26.03
Glass	41.99	33.25
Segment-test	36.17	6.54
Breast-cancer	34.26	24.46
Credit-g	33.90	29.20
Iris	8.00	4.00
Thoracic-Surgery	16.60	15.53
Tmusic	26.13	18.51

Victorias:	0	12

Tasa de error % (α = 0.05)		
	J48	**NB**
Soybean	7.61	7.92
Ionosphere	3.67	9.86 *
Vote	18.33	12.00
Diabetes	8.54	17.38 *
Labor	26.03	23.69
Glass	33.25	50.48 *
Segment-test	6.54	13.21 *
Breast-cancer	24.46	27.94
Credit-g	29.20	24.40
Iris	4.00	5.33
Thoracic-Surgery	15.53	22.13
Segment-challenge	18.51	42.23 *

Victorias:	9	3

$$N/2 + 1.96 * \sqrt{\frac{N}{2}}$$

Aplicando la formula sobre los N=12, el resultado aplicando la fórmula es 10.80.

- En el primer caso OneR y J48, obtiene 12 victorias > 10.8. Por ello se rechaza la hipótesis nula y por lo que podemos asegurar que J48 es mejor.
- En el segundo caso 9<10.8, por lo que se acepta la hipótesis nula; por lo que no se puede asegurar que un algoritmo sea superior a otro.

6. DOS MÉTODOS, VARIOS CONJUNTOS DE DATOS: RANKINGS

	Tasa de error %					Rankings				
	J48	**OneR**	**3NN**	**NB**	**SVM**	**J48**	**OneR**	**3NN**	**NB**	**SVM**
Soybean	7,64	66,35	8,46	8,02	6,94	2	5	4	3	1
Vote	3,54	4,37	6,07	9,79	4,28	1	3	4	5	2
Labor	17,85	23,79	11,55	8,73	9,12	4	5	3	1	2
Ionosphere	10,94	18,80	13,67	17,67	11,91	1	5	3	4	2
Diabetes	27,39	27,81	26,28	24,58	22,99	4	5	3	2	1
Glass	33,07	44,95	32,05	50,58	42,89	2	4	1	5	3

Segment-test	6,12	39,06	7,85	13,56	7,78	1	5	3	4	2
Breast-cancer	25,39	32,59	27,00	26,71	30,70	1	5	3	2	4
Credit-g	27,98	33,50	28,04	24,90	24,50	3	5	4	2	1
Iris	5,33	6,93	5,20	4,93	3,60	4	5	3	2	1
Thoracic-Surgery	15,32	16,30	17,32	25,28	15,49	1	3	4	5	2
Tmusic	20,00	30,34	41,87	41,98	36,87	1	2	4	5	3
Ranking:						2,08	4,33	3,25	3,22	2

- Test de Friedman
  N=12 / k=5
  $X^2_F = 14,43$
  Valor critíco X^2, con 3 grados de libertad, alfa=0.05 -> 7.82
  $X^2_F > X^2$, se rechaza la hipótesis nula- > rankings significativamente distintos.

- Test de Iman Davenport
  $F_F = 4.43$
  Valor crítico F con 4 y 44 grados de libertad, alfa =0.05 -> 2,584

  Como $F_F > F$ rechazamos la hipótesis -> rankings significativamente distintos.

7. DOS MÉTODOS, VARIOS CONJUNTOS DE DATOS: TEST POST-HOC

- Test de Nemenyi
  $q_{0.05} = 2,728$, 5 clasificadores.
  N=12
  K=5
  CD = 1,438

  - OneR y SVM: 4,33 – 2=2,33 > 1,438

  - OneR y J48: 4,33 – 2,08=2,25> 1,438

- Test de Bonferroni-Dunn
  - OneR frente al resto (peor frente al resto)

    CD=1.22

    4.33 – 1.22 = 3.11

  - SVM frente al resto (mejor frente al resto)

    CD=1.22

    2-1.22 =0.78

Anexo Script

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
j48_data=pd.read_csv('../OneDrive/Escritorio/MINERIA/PRACTICA 2/j48-ej1.csv')
```

In [3]:
```
j48_data.head()
```

Out[3]:

	inst#	actual	predicted	error	prediction
0	1	1:diaporthe-stem-canker	1:diaporthe-stem-canker	NaN	1.000
1	2	15:frog-eye-leaf-spot	14:alternarialeaf-spot	+	0.853
2	3	1:diaporthe-stem-canker	1:diaporthe-stem-canker	NaN	1.000
3	4	1:diaporthe-stem-canker	1:diaporthe-stem-canker	NaN	1.000
4	5	14:alternarialeaf-spot	14:alternarialeaf-spot	NaN	0.955

In [4]:
```
oner_data=pd.read_csv('../OneDrive/Escritorio/MINERIA/PRACTICA 2/oner-eje1.csv')
```

In [5]:
```
oner_data.head()
```

Out[5]:

	inst#	actual	predicted	error	prediction
0	1	1:diaporthe-stem-canker	8:brown-spot	+	1
1	2	15:frog-eye-leaf-spot	8:brown-spot	+	1
2	3	1:diaporthe-stem-canker	14:alternarialeaf-spot	+	1
3	4	1:diaporthe-stem-canker	14:alternarialeaf-spot	+	1
4	5	14:alternarialeaf-spot	14:alternarialeaf-spot	NaN	1

In [6]:
```
j48_error=j48_data[j48_data['error']=='+']
j48_error
```

Out[6]:

	inst#	actual	predicted	error	prediction
1	2	15:frog-eye-leaf-spot	14:alternarialeaf-spot	+	0.853
26	27	14:alternarialeaf-spot	15:frog-eye-leaf-spot	+	0.667
58	59	15:frog-eye-leaf-spot	14:alternarialeaf-spot	+	0.737
74	75	15:frog-eye-leaf-spot	14:alternarialeaf-spot	+	0.737
97	98	19:herbicide-injury	4:phytophthora-rot	+	0.943
108	109	15:frog-eye-leaf-spot	14:alternarialeaf-spot	+	0.737
133	134	15:frog-eye-leaf-spot	8:brown-spot	+	1.000
156	157	15:frog-eye-leaf-spot	14:alternarialeaf-spot	+	0.737
163	164	10:bacterial-pustule	9:bacterial-blight	+	0.867
173	174	15:frog-eye-leaf-spot	14:alternarialeaf-spot	+	0.737
182	183	3:rhizoctonia-root-rot	1:diaporthe-stem-canker	+	1.000
187	188	12:anthracnose	4:phytophthora-rot	+	0.963
210	211	8:brown-spot	15:frog-eye-leaf-spot	+	0.976
211	212	15:frog-eye-leaf-spot	13:phyllosticta-leaf-spot	+	0.667

In [7]:
```
oner_error=oner_data[oner_data['error']=='+']
oner_error
```

Out[7]:

	inst#	actual	predicted	error	prediction
0	1	1:diaporthe-stem-canker	8:brown-spot	+	1
1	2	15:frog-eye-leaf-spot	8:brown-spot	+	1
2	3	1:diaporthe-stem-canker	14:alternarialeaf-spot	+	1
3	4	1:diaporthe-stem-canker	14:alternarialeaf-spot	+	1
5	6	1:diaporthe-stem-canker	8:brown-spot	+	1
...	...	...	...	...	...
220	221	1:diaporthe-stem-canker	8:brown-spot	+	1
221	222	15:frog-eye-leaf-spot	4:phytophthora-rot	+	1
224	225	15:frog-eye-leaf-spot	4:phytophthora-rot	+	1
225	226	15:frog-eye-leaf-spot	4:phytophthora-rot	+	1
226	227	5:brown-stem-rot	2:charcoal-rot	+	1

159 rows × 5 columns

In [8]:
```
instancias_j48=set(j48_error['inst#'])
instancias_j48
```

Out[8]:
```
{2, 27, 59, 75, 98, 109, 134, 157, 164, 174, 183, 188, 211, 212}
```

In [9]:
```
instancias_oner=set(oner_error['inst#'])
print(instancias_oner)
```

```
{1, 2, 3, 4, 6, 7, 9, 10, 14, 15, 17, 18, 19, 21, 23, 24, 25, 26, 28, 29, 30, 31,
33, 34, 37, 39, 40, 41, 45, 49, 50, 52, 54, 59, 61, 62, 65, 66, 69, 71, 72, 73, 7
4, 75, 76, 77, 79, 80, 81, 83, 84, 85, 86, 87, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99, 102, 103, 105, 106, 108, 109, 110, 111, 113, 114, 115, 116, 117, 118, 119,
120, 123, 126, 127, 129, 130, 132, 133, 134, 135, 136, 138, 139, 140, 141, 142, 14
3, 145, 146, 149, 150, 151, 152, 153, 155, 156, 157, 159, 160, 161, 162, 163, 164,
165, 166, 167, 169, 170, 172, 173, 174, 175, 177, 179, 181, 182, 183, 184, 185, 18
7, 188, 189, 191, 193, 194, 195, 196, 197, 198, 199, 200, 201, 203, 204, 205, 209,
210, 212, 213, 214, 215, 216, 219, 220, 221, 222, 225, 226, 227}
```

In [10]:
```
#Intersección-> ejemplos mal clasificados por ha y hb
print(instancias_j48 & instancias_oner)
len(instancias_j48 & instancias_oner)
```

```
{2, 98, 164, 134, 75, 109, 174, 212, 183, 59, 188, 157}
```

Out[10]:
```
12
```

In [11]:
```
# Diferencia -> ejemplos mal clasificados por ha pero no por hb
print(instancias_j48 - instancias_oner)
```

```
{27, 211}
```

In [12]:
```
# Diferencia -> ejemplos mal clasificados por hb pero no por ha
print(instancias_oner -instancias_j48)
len(instancias_oner -instancias_j48)
```

```
{1, 3, 4, 6, 7, 9, 10, 14, 15, 17, 18, 19, 21, 23, 24, 25, 26, 28, 29, 30, 31, 33,
34, 37, 39, 40, 41, 45, 49, 50, 52, 54, 61, 62, 65, 66, 69, 71, 72, 73, 74, 76, 7
7, 79, 80, 81, 83, 84, 85, 86, 87, 89, 90, 91, 92, 93, 94, 95, 96, 97, 99, 102, 10
3, 105, 106, 108, 110, 111, 113, 114, 115, 116, 117, 118, 119, 120, 123, 126, 127,
129, 130, 132, 133, 135, 136, 138, 139, 140, 141, 142, 143, 145, 146, 149, 150, 15
1, 152, 153, 155, 156, 159, 160, 161, 162, 163, 165, 166, 167, 169, 170, 172, 173,
175, 177, 179, 181, 182, 184, 185, 187, 189, 191, 193, 194, 195, 196, 197, 198, 19
9, 200, 201, 203, 204, 205, 209, 210, 213, 214, 215, 216, 219, 220, 221, 222, 225,
226, 227}
```

Out[12]:
```
147
```

In [13]:
```
total_instancias=set(range(len(oner_data)))
print(total_instancias)
```

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 4
3, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 8
4, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103,
104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 12
0, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136,
137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 15
3, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169,
170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 18
6, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202,
203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 21
9, 220, 221, 222, 223, 224, 225, 226, 227}
```

In [14]:
```
#Ejemplos bien clasificados por ambos
print(total_instancias-(instancias_j48 | instancias_oner))
len(total_instancias-(instancias_j48 | instancias_oner))
```

```
{0, 128, 131, 5, 8, 137, 11, 12, 13, 16, 144, 147, 20, 148, 22, 154, 158, 32, 35,
36, 38, 168, 42, 43, 44, 171, 46, 47, 48, 176, 178, 51, 180, 53, 55, 56, 57, 58, 1
86, 60, 190, 63, 64, 192, 67, 68, 70, 202, 78, 206, 207, 208, 82, 88, 217, 218, 22
3, 224, 100, 101, 104, 107, 112, 121, 122, 124, 125}
```

Out[14]:
```
67
```

In [ ]: