Practica Final

Ahisahar Pretel Rodriguez 75937643S
Diego José Granados Álvarez 77150626R

Base de datos usada: Poker Hand
Enlace: https://archive.ics.uci.edu/ml/datasets/Poker+Hand

Índice

Descripción de la base de datos usada

La base de datos Poker Hand se compone de 1025010 observaciones con 11 atributos de tipo entero cada una. Cada observación representa una mano de poker junto con la clase a la que pertenece. Los atributos son:

- S1: Tipo de la carta #1
  Número (1-4) que representa {Corazones, Picas, Diamantes, Tréboles}

- C1: Rango de la carta #1
  Número (1-13) que representa (Ace,2,3,4,...,Joker,Queen,King)

- S2: Tipo de la carta #2
  Número (1-4) que representa {Corazones, Picas, Diamantes, Tréboles}

- C2: Rango de la carta #2
  Número (1-13) que representa (Ace,2,3,4,...,Joker,Queen,King)

- S3: Tipo de la carta #3
  Número (1-4) que representa {Corazones, Picas, Diamantes, Tréboles}

- C3: Rango de la carta #3
  Número (1-13) que representa (Ace,2,3,4,...,Joker,Queen,King)

- S4: Tipo de la carta #4
  Número (1-4) que representa {Corazones, Picas, Diamantes, Tréboles}

- C4: Rango de la carta #4
  Número (1-13) que representa (Ace,2,3,4,...,Joker,Queen,King)

- S5: Tipo de la carta #5
  Número (1-4) que representa {Corazones, Picas, Diamantes, Tréboles}

- C5: Rango de la carta #5
  Número (1-13) que representa (Ace,2,3,4,...,Joker,Queen,King)

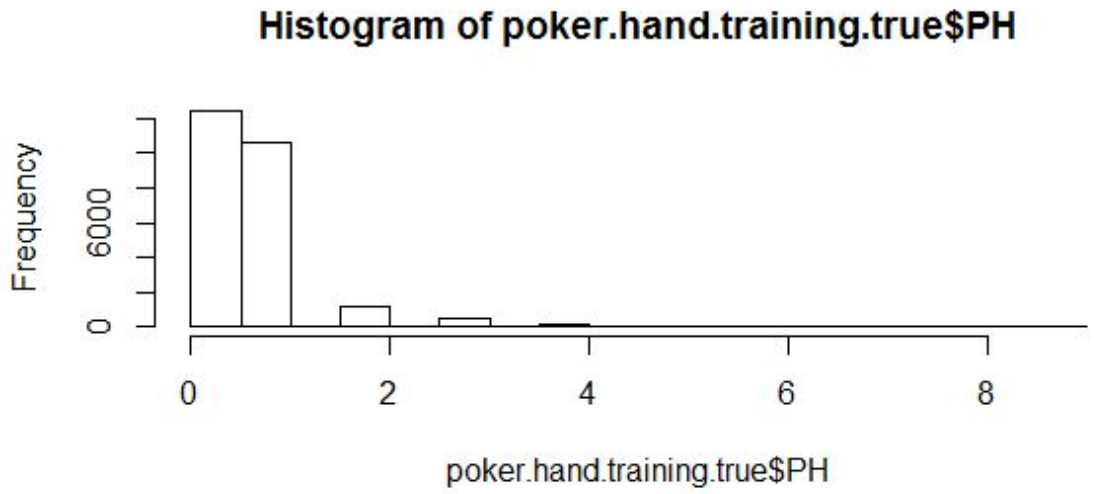
- Clase “Poker Hand”
  Número (0-9) que representa el tipo de mano.

  0.Nada
  1.Pareja
  2.Doble pareja
  3.Trio
  4.Escalera

5.Color
6.Full
7.Poker
8.Escalera de color
9.Escalera real

Descripción del problema

El objetivo de esta práctica es la de predecir sobre los datos de test a qué tipo de mano pertenece cada muestra por lo que nuestra variable a clasificar es PokerHand. Debemos hacer estas predicciones sin introducir información adicional sobre el juego del poker, el modelo es el que debe aprender por sí solo.



Este histograma representa la cantidad de muestras de cada tipo de mano en nuestro dataset de training. Podemos observar que la clase predominante es la clase 0 (con más del 50% de las muestras) seguida de la clase 1. Esto supone un desbalance de los datos que habrá que tratar más adelante.

```
> cor(poker.hand.training.true)
             S1           C1           S2           C2           S3           C3           S4           C4           S5           C5           PH
S1  1.000000000 -0.0103193120 -0.021262952  0.0082370764 -0.0196182529 -0.009946670 -0.017300508  0.003824018 -0.0241669794  0.006625273  0.008243775
C1 -0.010319312  1.0000000000  0.004460432 -0.0109927058 -0.0003824444 -0.026285111  0.004060986 -0.014438817  0.0048533450 -0.016967104  0.002311827
S2 -0.021262952  0.0044604319  1.000000000 -0.0024384141 -0.0293063440 -0.005303443 -0.020959992  0.012026048 -0.0118416433  0.005781572 -0.001134624
C2  0.008237076 -0.0109927058 -0.002438414  1.0000000000 -0.0053504858 -0.024701931 -0.008389535 -0.012933193  0.0006399903 -0.016382907 -0.005190634
S3 -0.019618253 -0.0003824444 -0.029306344 -0.0053504858  1.0000000000  0.017942232 -0.012998101  0.001873863 -0.0303054528 -0.002021573  0.003950721
C3 -0.009946670 -0.0262851107 -0.005303443 -0.0247019315  0.0179422324  1.000000000 -0.003191218 -0.016323189  0.0037311955 -0.010943182 -0.006255560
S4 -0.017300508  0.0040609855 -0.020959992 -0.0083895346 -0.0129981006 -0.003191218  1.000000000 -0.008695117 -0.0181998225  0.005345278 -0.001551697
C4  0.003824018 -0.0144388169  0.012026048 -0.0129331927  0.0018738628 -0.016323189 -0.008695117  1.000000000  0.0028104495 -0.014680974  0.010163039
S5 -0.024166979  0.0048533450 -0.011841643  0.0006399903 -0.0303054528  0.003731196 -0.018199823  0.002810449  1.0000000000 -0.003157110 -0.005616472
C5  0.006625273 -0.0169671041  0.005781572 -0.0163829074 -0.0020215725 -0.010943182  0.005345278 -0.014680974 -0.0031571104  1.000000000 -0.002414101
PH  0.008243775  0.0023118272 -0.001134624 -0.0051906338  0.0039507211 -0.006255560 -0.001551697  0.010163039 -0.0056164723 -0.002414101  1.000000000
```

Esta es la matriz de correlaciones de los datos de training. Podemos observar que ninguno de los coeficientes de correlación tiene un valor significativamente alto, lo que nos indica que no existe relación entre las variables. En realidad, este hecho tiene sentido ya que un

palo de una carta no depende del palo de otra carta ni del número de esta, de la misma forma que la clase de una muestra no depende únicamente de uno de los palos de las cartas que la componen.

En la siguiente tabla se muestran el número de instancias para cada clase en ambos conjuntos de training y test.

Clase/ Dataset	0	1	2	3	4	5	6	7	8	9
Trainig	12493	10599	1206	513	93	54	36	6	5	5
Test	501209	422498	47622	21121	3885	1996	1424	230	12	3

En total tenemos 25010 muestras en el conjunto de training y un millón en el conjunto de test.

Resumen del análisis de datos exploratorio realizado para determinar las técnicas seleccionadas

En primer lugar ha sido necesario ordenar los datos de cada muestra para eliminar redundancia en los datos y para que aprenda el clasificador. Dadas dos manos con las mismas cartas y en distinto orden, para el clasificador serían diferentes si no ordenamos las cartas.

Después hemos tenido que balancear la base de datos ya que, como se vió en el histograma del apartado anterior existe un desbalance de los datos.

Resumen general de la metodología llevada a cabo en el ajuste de los modelos empleados

Hemos creado dos modelos para cada uno de los tres usados (KNN, Bayes, SVM), uno con la base de datos original y otro con la base de datos ordenada para poder comprobar que al ordenar los datos se aumenta el porcentaje de predicciones correctas.

Detalles de la metodología de entrenamiento para cada modelo

NAIVE BAYES

Usando un método **naive bayes** entrenado sobre los datos de training sin ordenar prediciendo sobre los datos de test produce la siguiente matriz de confusión:

	truth									
predict	0	1	2	3	4	5	6	7	8	9
0	501209	422498	47622	21121	3885	1996	1424	230	12	3
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

Podemos observar que clasifica todas las muestras con clase 0 lo que es incorrecto. Al predecir cualquier muestra como clase 0 consigue un 50,1209% de error pero este modelo no es válido.

Al ordenar los datos obtenemos la siguiente matriz de confusión:

	truth									
predict	0	1	2	3	4	5	6	7	8	9
0	12265	7260	673	149	0	49	12	0	0	0
1	130774	105278	11140	4625	915	504	264	46	3	0
2	52843	48166	5777	2733	552	208	210	30	0	1
3	101029	86557	9688	4526	849	412	271	45	1	1
4	144226	119565	13510	6136	1212	579	449	63	7	0
5	0	0	0	0	0	0	0	0	0	0
6	20298	15807	1743	575	40	61	38	5	0	0
7	2672	2410	305	121	19	5	7	0	0	0
8	22848	24979	3268	1637	122	124	122	30	1	0
9	14254	12476	1518	619	176	54	51	11	0	1

Con la base de datos ordenada obtenemos un 12,9098%. Observamos que Naive Bayes no es un buen método para clasificar nuestra base de datos. El problema puede ser que este método no funciona bien clasificando clases con pocas ocurrencias, lo que se podría solucionar balanceando la base de datos.

KNN

Base de datos sin ordenar

	k = 1	k = 5	k = 10	k = 15

Tasa de acierto	65,6186%	68,8984%	69,9919%	70,3138%

KNN

Base de datos ordenada

	k = 1	k = 5	k = 10
Tasa de acierto	21,3331%	27,0815%	31,3869%

SVM

Realizando validación cruzada para comprobar cuales son los mejores valores para el modelo svm obtenemos los siguientes modelos:

Para la base de datos sin ordenar:

```
> summary(tune.out$best.model)

Call:
best.tune(method = svm, train.x = PH ~ ., data = poker.hand.training.true[1:1500,
    ], ranges = list(gamma = 2^(-1:1), cost = 2^(2:4)), kernel = "radial")


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  4
      gamma:  0.5
    epsilon:  0.1


Number of Support Vectors:  1426
```

Para la base de datos ordenada:

```
> summary(tune.out$best.model)

Call:
best.tune(method = svm, train.x = PH ~ ., data = train[1:1500, ], ranges = list(gamma = 2^(-1:1),
    cost = 2^(2:4)), kernel = "radial")


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  8
      gamma:  0.5
    epsilon:  0.1


Number of Support Vectors:  1319
```

Para la base de datos sin ordenar produce un porcentaje de acierto del: 46,6659%.

Para la base de datos ordenada produce un porcentaje de acierto del: 42.3054%. Vemos que ha empeorado el porcentaje de acierto. Esto se puede deber a que no hemos cogido la base de datos completa en la validación cruzada y no hemos obtenido los valores óptimos para gamma y el coste. Hemos cogido una porción de la base de datos porque los tiempos de ejecución eran demasiado altos y no podíamos comprobar los resultados correctamente.

Comparación entre las distintas técnicas

	Naive Bayes	SVM	KNN
Error de test	12,9098%	42.3054%	31,3869%

Valoración final de los resultados obtenidos

En este caso SVM obtiene mejores predicciones en nuestro problema.

Conclusiones

Nuestra base de datos tenía una gran cantidad de muestras ya que son necesarias para predecir nuestro problema ya que hay muchísimas combinaciones de cartas para una mano.
Esto nos ha afectado en los tiempos de ejecución de las operaciones (sobre todo en la validación cruzada) porque teníamos que esperar mucho tiempo para obtener resultados y comprobar que eran correctos.

También ha sido necesario ordenar los datos para que el clasificador no tomara como diferentes dos manos con las mismas cartas pero en diferente orden.

Nos ha faltado balancear los datos, lo que hubiera mejorado las predicciones de los modelos.