# MDD - Laboratorio I: Tareas 1-7

**Silvia Arenales Muñoz**

## Tarea 1: Video -- Documentos TV: "Big data, conviviendo con el algoritmo"

1. **De todas las aplicaciones que se mencionan, ¿cuál es la que más te ha llamado la atención? ¿por qué? ¿la ves "factible"-"creíble"?**

La mayoría de las aplicaciones mencionadas, son actualmente usadas. Aunque la que más me ha llamado la atención podría decir que sería la utilización de algoritmos para predecir el cambio climático o enfermedades para la medicina personalizada. Esta última me llama mucho la atención el cómo puede ver el algoritmo la influencia de los cambios de los genes en las enfermedades para prevenir y curar. Creo que en un futuro gracias a estos algoritmos se podrá avanzar tanto en el ámbito de la medicina como en cualquier ciencia, dando avances asombrosos.

2. **¿Qué persona entrevistada es la que más te ha llamado la atención? ¿por qué?**

Gemma Galdón Directora y fundadora de Éticas Research & Consulting, ya que además de que no la conocía, por la información que decía sobre las super empresas y la información., como Axión.

3. **¿Cuál es la idea-reflexión que más te ha llamado la atención? ¿por qué?**

Una aplicación que comenta Chema Alonso me ha llamado la atención, la cual es DataTransparnt y Apps el cual contabiliza el dinero que gana la red social en la que se está por los clicks dados en los anuncios.

4. **De entre los riesgos y dudas-peligros éticos que plantean los entrevistados, ¿cuál es el que te ha llamado la atención?**

Carlos Castillo comenta un algoritmo llamado *compass* el cual determina el porcentaje que una persona vaya a prisión. Estos algoritmos no son del todo éticos ya que si comete el mismo número de errores para gente de raza blanca o negra, los seguirá cometiendo. Es decir, los algoritmos tienden a reproducir las desigualdades que hay en los datos que observan. Estos son alimentados por datos, por tanto si no se utilizan técnicas de algoritmos no discriminatorios, se reproducirían los estereotipos.

5. **Coméntame al menos un párrafo, una reflexión más tuya, propia, más de "tu cosecha"...**

La *datalización* del mundo es necesarios para que la IA aprenda y así pueda ayudar a otros casos. Por ejemplo una persona enferma da acceso a que se usen sus datos para que se pueda investigar acerca de su enfermedad, para así cuando haya otra persona que padezca con síntomas parecidos, se pueda detectar con más facilidad. Sin embargo, no todo el mundo quiere esa sobreexposición de sus datos. La batalla sobre la posesión, el acceso, el uso y creación de conocimiento a partir de los datos es uno de los problemas de la IA.

6. **Échame una mano y busca en la web videos introductorios similares a éste, para futuros alumnos de la asignatura, que te parezcan apropiados para ello. Resúmelo. Este subapartado no tienes porqué responderlo en las primeras semanas de la asignatura, sino unas semanas más tarde, con más experiencia, previo a la entrega del laboratorio.**

Este vídeo también de la misma casa, habla del potencial de la IA que tiene para las empresas. Luego estos dos TED talks introductorios para entender qué es la IA y su potencial, aunque están en ingles : 1 y 2.

# Tarea 2: Clasificador del vecino más próximo - K-NN

1. **Escogemos la base de datos "CoverTypePrediction-Kaggle.arff" de nuestro directorio. Entiende el problema a clasificar ("variable clase a predecir").**

Nos encontramos ante un problema de clasificación llamado *Forest Cover Type Prediction* el cual trata de predecir qué tipos de bosque crecen en una región en función de las características del medio. Tenemos 4 diferentes áreas (wilderness area), 56 atributos de los cuales nos basamos (en clasificador entrena) para clasificarlos en 7 diferentes tipos de bosques.

2. **Explica los siguientes parámetros del clasificadro "IBk": "KNN", "distanceWeighting".**

K-NN: clasificador que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. Se le tiene que indicar el número de influencias que va a tener en cuenta.

DistanceWeighting: la ponderación permite que los vecinos más cercanos al punto de datos tengan más influencia en el valor predicho.

3. **Basándote en los apuntes de la teoría y tu intuición, explica los siguientes valores del parámetro "distanceWeighting": "NoDistanceWeighting", "Weight by 1/distance".**

NoDistanceWeighting: No tiene en cuenta lo cercano que está un dato a la hora de asignarle un valor de la clase.

Weight by 1/distance: Este caso sí que tiene en cuenta la distancia a la que está su dato vecino a la hora de darle un valor de la clase.

4. **Haz pruebas de manera informal, cambiando los valores de los parámetros "KNN" y "distanceWeighting": y viendo cómo cambia la tasa de acierto, estimando ésta con el método "Hold-out" (%66 training, %33 test) (en WEKA, "Percentage split"). Haz algunos comentarios sobre los cambios en la tasa de acierto en base a las variaciones en los valores de estos parámetros, y comentar cómo ha afectado su variación a la tasa de acierto. Muestra una gráfica, con un formato atractivo que tú elijas, para mostrar los resultados.**

Cuando probamos con %66 training y %33 test además con un clasificador KNN con K=1, obtenemos la siguiente matriz de confusión:

Con la siguiente tasa de validación:

Ahora cambiamos a 80% el porcentaje de entrenamiento y 20% de test y obtenemos lo siguiente:

A continuación, con 90% de entrenamiento y 10% de test.

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 493 163   1   0  31   1  60 |   a = 1
 123 439  19   1  76  28  13 |   b = 2
   0   8 512  55  10 120   0 |   c = 3
   0   0  34 708   0  26   0 |   d = 4
   5  36  17   0 648  11   1 |   e = 5
   1  14  98  38  16 599   0 |   f = 6
  19   3   0   0   0   0 714 |   g = 7
```

```
=== Summary ===

Correctly Classified Instances        4113               80.0039 %
Incorrectly Classified Instances      1028               19.9961 %
Kappa statistic                          0.7667
Mean absolute error                      0.0571
Root mean squared error                  0.239
Relative absolute error                 23.3272 %
Root relative squared error             68.2965 %
Total Number of Instances             5141
```

Por tanto, observamos que a medida que aumentamos el dataset de entrenamiento, el algoritmo aprende mejor y por lo tanto, a la hora de realizar el proceso de testo comete menos fallos.

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 292  93   1   0  18   1  35 |   a = 1
  69 267  10   1  39  15  10 |   b = 2
   0   8 319  33   6  75   0 |   c = 3
   0   0  23 429   0  16   0 |   d = 4
   2  15  10   0 363   7   1 |   e = 5
   1   9  49  21  10 357   0 |   f = 6
  12   1   0   0   0   0 406 |   g = 7
```

```
=== Summary ===

Correctly Classified Instances        2433               80.4563 %
Incorrectly Classified Instances       591               19.5437 %
Kappa statistic                          0.7719
Mean absolute error                      0.0558
Root mean squared error                  0.2363
Relative absolute error                 22.7992 %
Root relative squared error             67.5204 %
Total Number of Instances             3024
```

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 147  41   0   0  12   1  18 |   a = 1
  32 135   5   0  19   7   6 |   b = 2
   0   6 167  12   5  38   0 |   c = 3
   0   0  12 207   0   8   0 |   d = 4
   1   9   5   0 178   3   0 |   e = 5
   0   7  20   6   8 178   0 |   f = 6
   5   0   0   0   0   0 214 |   g = 7
```

```
=== Summary ===

Correctly Classified Instances        1226               81.0847 %
Incorrectly Classified Instances       286               18.9153 %
Kappa statistic                          0.7793
Mean absolute error                      0.054
Root mean squared error                  0.2325
Relative absolute error                 22.0675 %
Root relative squared error             66.4311 %
Total Number of Instances             1512
```

5. **¿Por qué crees que WEKA agrupa en la familia "lazy" a este tipo de clasificador? Ten en cuenta lo que quiere decir en inglés... pista: ¿cuándo ha empezado a "trabajar" el clasificador? ¿a qué me refiero por "trabajar"? Es verdad que tendrás más argumentos para responder a esta pregunta después de haber visto más tipos de clasificadores. Si no lo ves claro, lo podrás responder dentro de unas semanas...**

Se dice que es "lazy", vago, por que es vago de aprender. Es decir, K-NN es un aprendiz perezoso porque no aprende una función discriminatoria de los datos de entrenamiento, sino que "memoriza" el conjunto de datos de entrenamiento.

# Tarea 3: Evaluación de clasificadores: método Hold-out; TPR, FPR, scores desde la matriz de confusión

1. **Describe el problema que recoge el dataset:**

En esta tercera tarea vamos a trabajar con el dataset "Spotify-songs_data-predictGenre.arff" . Básicamente se puede enunciar como la predicción del género de una canción mediante unas características. Este predice 750 canciones en 5 diferentes géneros de música seleccionados escogidas desde la API de Spotify. Se han recogido 50 canciones de 15 grupos/artistas diferentes y tenemos 13 características (14 atributos con la variable a predecir).

Por tanto, las variables predictoras son las 13 características *(acousticness, danceability, duration:ms, energy, instrumentalness, key, liveness, loudness, mode, speechiness, timepo, time_signature, valence)* , la variable a predecir es el género y sus valores son los 750 canciones.

2. **Estima y muestra el porcentaje de bien clasificados mediante el método H, hold-out, entrenando con 66%, testando en el 34% restante.**

Primero especifico que he escogido 5 vecinos a considerar para la clasificación y el pesado de cada vecino de su voto Weigh by 1/distance.

3. **Analiza la matriz de confusión resultado de la validación. Analiza y muestra los resultados de : TPR, FPR y Precisión**

Obtenemos los siguientes datos:

Observamos por ejemplo, en el caso a) , rock, en la primera columna de la primera final, 166 han sido correctamente clasificados (True Positive) , y el resto de la fila (11 , 3 , 4 y 53) han sido clasificados como falsos negativos, es decir, se han clasificado como rock cuando no lo son realmente. Los de la primera columna mientras tanto, (15, 20, 3 y 45) han sido sido clasificados como falsos positivos. Es decir, se han clasificado de otro género cuando son rock realmente.

```
=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 166  11  33   4  53 |   a = rock
  15 166  14  50   8 |   b = jazz
  20  11 168   2  46 |   c = reggae
   3  40   5 214   0 |   d = classical
  45   3  27   0 171 |   e = dance
```

En esta tabla podemos observar los diferentes valores de precisión que toma dependiendo de la clase.

4. **Define brevemente cada uno de estos términos, en una frase.**

True Positive Rate, TPR (Sensibilidad o Recall): La sensibilidad de una prueba representa la probabilidad de que un sujeto enfermo tenga un resultado positivo en la prueba.

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

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,622    0,082    0,667      0,622    0,643      0,554    0,864     0,708     rock
               0,656    0,064    0,719      0,656    0,686      0,614    0,917     0,749     jazz
               0,680    0,077    0,680      0,680    0,680      0,603    0,913     0,754     reggae
               0,817    0,055    0,793      0,817    0,805      0,753    0,957     0,866     classical
               0,695    0,104    0,615      0,695    0,653      0,565    0,900     0,716     dance
Weighted Avg.  0,694    0,076    0,696      0,694    0,694      0,618    0,910     0,759
```

False Positive Rate, FPR (1-Specifity): La especificidad de una prueba representa la probabilidad de que un sujeto sano tenga un resultado negativo en la prueba.

Precision: La precisión es el ratio entre el número de documentos relevantes recuperados entre el número de documentos recuperados.

5. **Escoge el valor de clase "jazz" como "positivo", y muestra y explica los valores de TPR, FPR, Precision. Ya que cada uno de estos valores decimales es fruto de una división de dos valores, muestra qué división concreta ha formado cada valor-score (TPR, FPR, Precision).**

TPR se trata de los valores correctamente clasificados positivamente entre los reales positivos.

FPR se trata de los valores incorrectamente clasificados positivamente entre los reales negativos.

Precision se trata de los correctamente clasificados como positivos entre los predichos como positivos.

Por ejemplo, si tenemos en cuenta la matriz de confusión, el jazz tiene los siguientes valores, los cuales se calculan de la siguiente manera:

TPR = 166/(15+166+14+50+8) = 0,656126

FPR = (11+11+40+3)/(166+11+33+4+53+20+11+168+2+46+3+40+5+214+45+3+27+171) = 0,063600

Precision = 166 / (11+166+11+40+3) = 0,718614

Sin embargo, en la tabla anterior se muestran con detalle los valores de cada clase.

## Tarea 4: Problemas "benchamark" de clasificación

1. **¿Cuál es el problema de clasificación supervisada que reflejan? ¿Cuál es la variable clase ("problem class", "class to be predicted"), y esta qué distintos valores toma? ¿Está el problema de clasificación "desbalanceado": son la mayoría de casos de una clase y unos pocos del resto?**

He seleccionado la base de datos de "mobile-price-classification".

El problema de clasificación que podemos observar, es que KNN (1-NN el caso escogido) no es un buen clafisicador para este problema ya que solamente ha acertado el 39,5588% de los casos.

```
=== Summary ===

Correctly Classified Instances         269               39.5588 %
Incorrectly Classified Instances       411               60.4412 %
Kappa statistic                          0.1935
Mean absolute error                      0.3311
Root mean squared error                  0.4354
Relative absolute error                 88.2828 %
Root relative squared error            100.5311 %
Total Number of Instances              680
```

Como podemos observar en la matriz de confusión, mostrada a continuación, el problema está bastante desbalanceado. En el caso de low-cost, se ve que hay una mayoría de casos de acierto pero aun así sí que hay bastantes casos que se han clasificado low-cost los cuales no deberían.

```
=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 70 59 22 12 |  a = low-cost
 38 55 53 22 |  b = medium-cost
 17 52 54 53 |  c = high-cost
  8 26 49 90 |  d = very-high-cost
```

2. **¿Cuántas variables predictoras hay en total?**

Tenemos 20 variables predictoras, que son básicamente las características del teléfono móvil: powe_numeric, blueetooth, clock_speed ,dual_sim, fc ,four_g, int_memory ,m_dep , mobile_wt , n_cores , pc , px_height , px_width , ram , sc_h , sc_w, talk_time, three_g, touch_screen y wiki.

3. **trata de entender, y luego de explicar en la respuesta, al menos 3-4 variables predictoras que pudieran ser relevantes en la clasificación (o más, si así lo consideras).**

- blueetooth: es una variable predictora binaria el cual indica si tiene o no el dispositivo conexión blueetooth.

- dual_sim: es una variable predictora binaria el cual indica si tiene o no el dispositivo tiene la posibilidad de tener dos sims.

- n_cores: es una variable predictora numérica la cual indica le número de cores (núcleos) que tiene el dispositivo.

4. **comenta cualquier otro punto o ángulo del problema de clasificación que haya llamado tu atención: ¿por qué han llamado tu atención, por qué te "gustan"? ¿por qué has escogido éstas y no otras? ¿qué otros datasets has considerado y ojeado y finalmente no has incluido, cuáles?**

La verdad que este me llamaba ya que sentía curiosidad cómo se clasificaban los móviles y también me parecía más fácil de comprender. Aunque también me llamaba la atención el dataset de diabetes ya que tengo una amiga que la padece y es muy cercana a mí.

# Tarea 5 - Evaluación de clasificadores: métodos de evaluación

1. **Trabajando con el dataset "Pokemon_rarity.arff":**

Estamos trabajando con un dataset que trata de clasificar un número (12365 exactamente) de muestras, cartas de pokemons, para clasificarlos según su

rareza. Para ello, se han tenido en cuenta 26 variables predictoras. Por tanto tenemos 2 variables identificadoras (id y name) más 26 predictoras y la variable a predecir (rarity). Puede clasificarse en 20 tipos de rareza la carta.

Se ha clasificado con los siguientes datos:

Clasificador K-nearest neighbour con K=20. Asimismo, la distanceWighting elegimos la opción Weight by 1/distance para que se tenga en cuenta los vecinos más cercanos, exactamente los 5 más cercanos.

2. **Estima y muestra el porcentaje de bien clasificados ("Correctly classified instances, %") de las siguientes formas :**
   1. **Método de "resustitución - error aparente":**

      | | | |
      |---|---|---|
      | Correctly Classified Instances | 12365 | 100 % |
      | Incorrectly Classified Instances | 0 | 0 % |

   2. **Método H, hold-out, entrenando con 66% :**

      | | | |
      |---|---|---|
      | Correctly Classified Instances | 2692 | 64.0343 % |
      | Incorrectly Classified Instances | 1512 | 35.9657 % |

   3. **Método H 5 veces, haciendo finalmente un "average" de los resultados de validación de esas 5 ejecuciones:**

      Random seed: 2

      | | | |
      |---|---|---|
      | Correctly Classified Instances | 2652 | 63.0828 % |
      | Incorrectly Classified Instances | 1552 | 36.9172 % |

      Random seed: 3

      | | | |
      |---|---|---|
      | Correctly Classified Instances | 2676 | 63.6537 % |
      | Incorrectly Classified Instances | 1528 | 36.3463 % |

      Random seed 4:

      | | | |
      |---|---|---|
      | Correctly Classified Instances | 2705 | 64.3435 % |
      | Incorrectly Classified Instances | 1499 | 35.6565 % |

      Random seed: 5

      | | | |
      |---|---|---|
      | Correctly Classified Instances | 2677 | 63.6775 % |
      | Incorrectly Classified Instances | 1527 | 36.3225 % |

4. **Método 10-fold cross-validation :**

| | | |
|---|---|---|
| Correctly Classified Instances | 8019 | 64.8524 % |
| Incorrectly Classified Instances | 4346 | 35.1476 % |

5. **Método 5-fold cross-validation :**

| | | |
|---|---|---|
| Correctly Classified Instances | 7964 | 64.4076 % |
| Incorrectly Classified Instances | 4401 | 35.5924 % |

6. **Leave-one-out:**

| | | |
|---|---|---|
| Correctly Classified Instances | 8033 | 64.9656 % |
| Incorrectly Classified Instances | 4332 | 35.0344 % |

Para que sea leave-one-out cross validation, K debe ser igual que N, es decir, en folds debemos de poner 20, ya que es el número K, que hemos escogido al principio del laboratorio.

Es verdad que a la hora de ejecutarlo, comparando con las anteriores métodos tarda un tiempo más, ya que tiene 20 iteraciones a realizar.

7. **(WEKA no tiene implementado el 0,632 bootstrapping)**

**3. ¿Devuelven todos el mismo porcentaje de error estimado? ¿por qué?**

No, no devuelven el mismo error esperado. Cada técnica de validación calcula de diferente manera su error esperado.

El método de resustitución calcula de la siguiente manera, sobre los N casos en el conjunto de datos:

$$\hat{p}_M = \frac{1}{N}\sum_{j=1}^{N}\delta(c_j, c_{Mj})$$

El método de Holdout calcula el error de la siguiente manera siendo h el número de casos en el conjunto de test:

$$\hat{p}_M = \frac{1}{h}\sum_{j=1}^{h}\delta(c_j, c_{Mj})$$

El método de validación cruzada, k-fold cross validation de la siguiente manera:

$$\hat{p}_M = \frac{1}{k} \sum_{i=1}^{k} \hat{p}_i$$

4. **¿Cuál de los métodos anteriores devuelve el mejor porcentaje de error estimado? ¿era esperable? ¿por qué?**

El método de resustitución, error aparente, ya que devuelve el menor número de error aparente. Sin embargo, este método no es honesto, ya que estamos calculando sobre las muestras aprendidas, si el modelo las memoriza, no habrá error.

5. **¿Cuál te parece la estimación más fiable de las 6 anteriores?**

El método de validación cruzada (Leave-one-out cross-validation) ya que mejora el modelo mediante la validación de los datos.

6. **¿Cuál de los 6 métodos anteriores de estimación crees que le ha exigido más trabajo de cómputo a WEKA?**

Al método de validación cruzada, cuando realizamos Leave-one.out cross-validation, ya que divide el data set en k subconjuntos (20 en nuestro caso) y utiliza un subconjunto como test y el resto como training, y realiza permutaciones de modo que todas las hojas se usen como test. Es decir, el proceso se repite k veces cada vez con una capa de testeo distinto.

7. **cuando le pedimos a WEKA que estime el error mediante 10-fold cross-validation, ¿cuántos modelos aprende WEKA en total? ¿Por qué no son 10? ¿Con qué porcentaje de casos del total aprende cada uno de esos modelos?**

Lo que creará weka será 10 folds(capas) para entrenar, es decir serán 10 entrenamientos independientes y al final estas 10 accuracies que deberían de ser parecidos entre sí, serán la del modelo final. Por tanto, aprende solo un modelo a partir de 10 diferentes entrenamientos.

8. **¿Puede darse el caso que para una "seed-semilla" y método de validación concreto, en una ejecución el clasificadorX "sea mejor" que el "clasificadorY", y que para otra "seed-semilla" ocurra justamente lo contrario? ¿Por qué crees que se puede dar esta situación?**

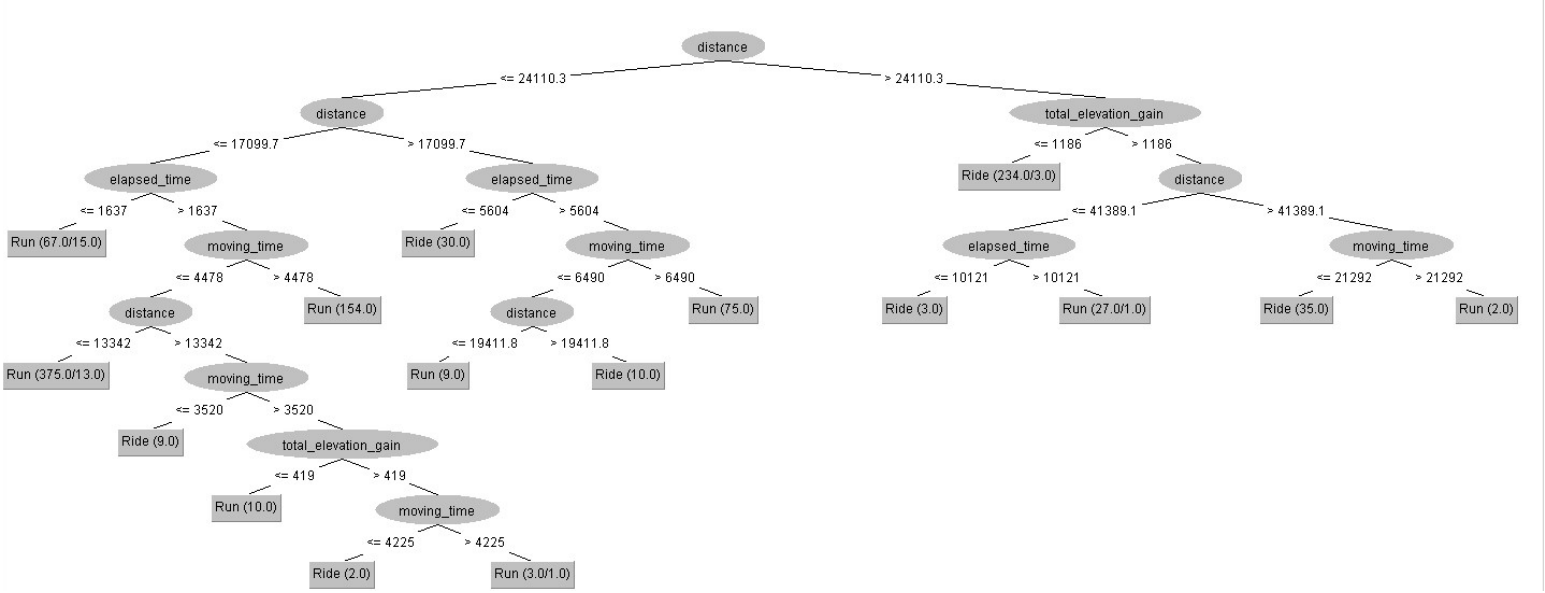
Por que al ser aleatorio, puede que en algún caso se ajuste mejor que otro.

# Tema 6 - Árboles de clasificación & Decisión trees

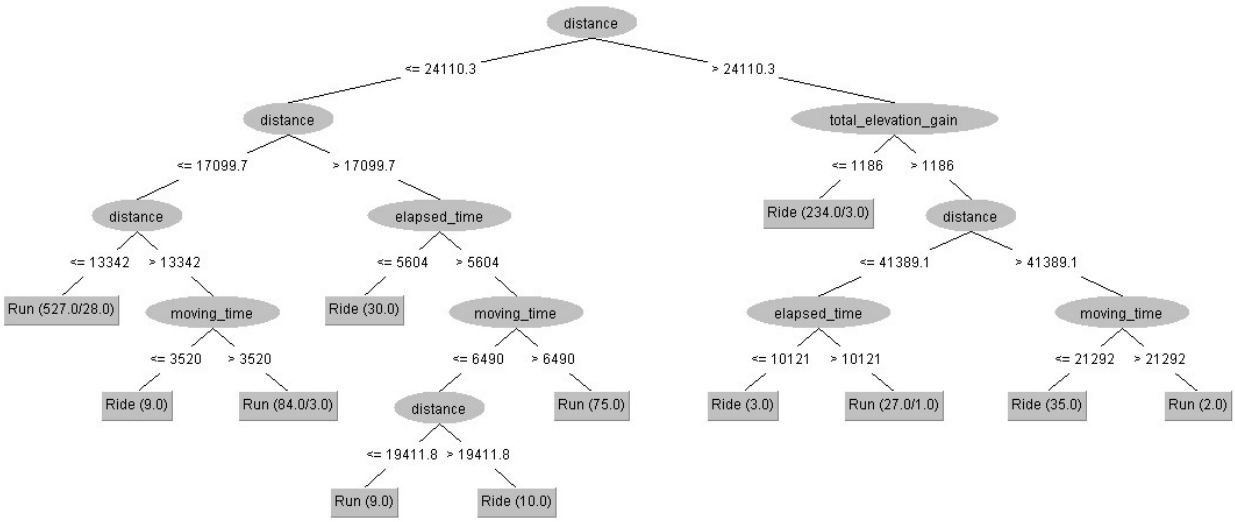
1. **Tabajamos con la base de datos "strava_actividad"**

Este dataset contiene datos relativos a actividades deportivas obtenidas de la plataforma Strava.

2. **Método C4.5 de construcción de árboles de clasificación**
    1. **Sin podar**



## 2. Podado




Ambos casos son en entrenamiento. Ahora no nos estamos fijando en los casos de error y de acierto.

3. **¿Son los dos árboles distintos? ¿hasta qué punto? ¿en el número de variables intermedias? ¿en el número de hojas?**

Observamos que lo árboles son distintos. En el árbol sin podar, tenemos 16 hojas y su nivel es de 31, mientras que en el árbol podado tenemos 12 hojas y 23 niveles.

4. **¿Aparecen todas las variables originales en los árboles finales? ¿tienen que aparecer obligatoriamente (recuerda cómo es el método de construcción de árboles de clasificación)?**

Si aparecen todas las variables, aunque no es obligatorio que tangan que aparecer todas, ya que al podar no afecta al resultado final, ya que estamos eligiendo el mejor modelo que se ajusta a nuestro problema.

5. **Fíjate en la versión del árbol podado. En cada hoja final, aparecen entre paréntesis uno (o dos) números. Acerca de estos, ¿qué crees que pueden ser los números que aparecen en la izquierda? Para pensar sobre ello, olvídate de los decimales, y suma los números que aparecen**

**en la parte izquierda del paréntesis. Recuerda cuántos casos en total hay en la base de datos...**

Esos números representan las clases. Si un nodo hoja tiene dos números significa que unos cuantos ha clasificado mal. Es decir, el número a la derecha son aquellos de la clase correctamente clasificados, mientras que los de la izquierda no.

6. **Y siguiendo la pregunta anterior, en la versión del árbol podado: ¿qué pueden ser (olvídate de los decimales) los números de la parte derecha en los paréntesis de cada hoja? En algunas hojas no hay un segundo número en dicho paréntesis.**

Los números de la derecha son aquellos que están mal clasificados. Si este solamente tiene un número, es buena señal ya que significa que ha clasificado correctamente.

7. **Fíjate que hasta ahora no nos ha preocupado la "validación" del árbol, sino su estructura. Vamos a validarlo. Para uno de los árboles que elijas (el podado o no), y un valor de clase que tú elijas ("tú decides cuál es la clase positiva"), muestra los valores estimados de "true positive rate (TPR)" y "false positive rate (FPR)" (estimando el porcentaje de acierto con una 10-fold cross-validation) ¿qué son el TPR y FPR? Defínelos brevemente, con una frase "tuya", precisa. Nada de copiar de la red, etc.**

TPR, True positive rate, son aquellos correctamente clasificados positivos entre los reales positivos. Mientras que FPR, False Positive rate son los incorrectamente clasificados como positivos entre los negativos reales.

8. **Tienes que crear un segundo dataset en formato *.arff con 5 instancias . ¿Entiendes lo que estamos haciendo? Esto es, estamos entrenando el modelo con nuestros 1045 casos etiquetados, y utilizándolo para predecir la clase en 5 instancias de las cuales desconocemos la etiqueta.**

Archivo utilizado:

@relation strava-activities

@attribute type {Run, Ride}

@attribute distance real

@attribute moving_time real

@attribute elapsed_time real

@attribute total_elevation_gain real

@data

"?",9600.7,2511,2511,80.0

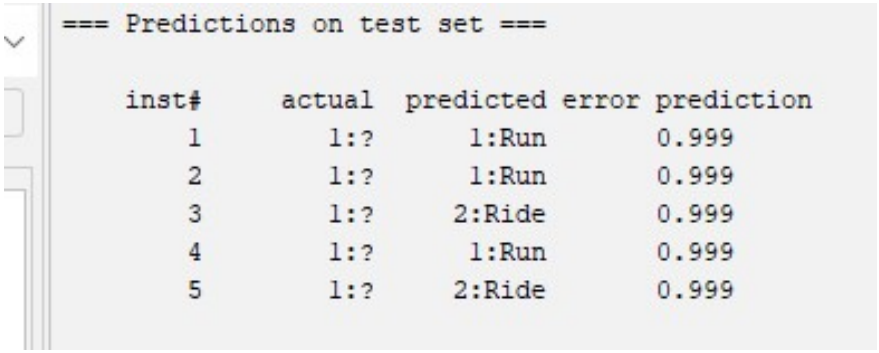
"?",19896.3,6495,6538,528.0

"?",29606.5,4428,4428,391.0

"?",14944.7,4883,4837,485.0

"?",33228.5,4606,4661,621.0

Se está intentando clasificar estos 5 nuevas instancias con nuestro clasificador. Este es el resultado:



```
=== Predictions on test set ===

    inst#     actual  predicted error prediction
        1        1:?      1:Run       0.999
        2        1:?      1:Run       0.999
        3        1:?     2:Ride       0.999
        4        1:?      1:Run       0.999
        5        1:?     2:Ride       0.999
```

Podemos ver que para la primera instancia se le asocia de tipo *Run*, al igual que el segundo y el cuarto. Mientras que el tercero y el quinto de tipo *Ride*.

# Tarea 7 - Clasificadores Bayesianos, Naïves-Bayes

El dataset está compuesto por 286 registros de pacientes que presentaron o no recurrencia de cáncer de mama después de cinco años de una cirugía. Los atributos no son suficientes para discriminar con mucha precisión entre las dos clases, pero son un muy buen ejemplo para aprender a procesar datos.

**1. ¿Qué nos está mostrando la salida de WEKA?**

Podemos observar una tabla la cual es un conjunto de datos donde muestra las probabilidades de padecer o no recurrencia de cáncer en función de las variables seleccionadas.

```
Naive Bayes Classifier

                          Class
Attribute      no-recurrence-events    recurrence-events
                          (0.7)                (0.3)
=============================================================
menopause
  lt40                     6.0                  3.0
  ge40                    95.0                 36.0
  premeno                103.0                 49.0
  [total]                204.0                 88.0

node-caps
  yes                     26.0                 32.0
  no                     172.0                 52.0
  [total]                198.0                 84.0

breast
  left                   104.0                 50.0
  right                   99.0                 37.0
  [total]                203.0                 87.0

irradiat
  yes                     38.0                 32.0
  no                     165.0                 55.0
  [total]                203.0                 87.0
```

**2. Indica y marca explícitamente cada uno de los "p(Xi=xi|C=c)", "p(C=c)" que aparecen: para todas las variables (Xi) y valores de la clase (C).**

Menopause:

P(menopause = lt40 | C=no-recurrence-events) = 6/204

P(menopause = lt40 | C=recurrence-events)= 3/88

P(menopause = ge40 | C=no-recurrence-events) = 95/204

P(menopause = ge40 | C=recurrence-events)= 36/88

P(menopause = premeno | C=no-recurrence-events)=103/204

P(menopause = premeno | C=recurrence-events)=49/88

Node-caps:

P(node-caps = yes | C=no-recurrence-events) = 26/198

P(node-caps = yes | C=recurrence-events)=32/84

P(node-caps = no | C=no-recurrence-events)= 172/198

P(node-caps = no | C=recurrence-events)=52/84

Breast:

P(Breast = left | C=no-recurrence-events)=104/203

P(Breast = left | C=recurrence-events)=50/87

P(Breast = right | C=no-recurrence-events)=99/203

P(Breast = right | C=recurrence-events)=37/87

Irradiat:

P(Irradiat = yes | C=no-recurrence-events)=38/203

P(Irradiat = yes | C=recurrence-events)=32/87

P(Irradiat = no | C=no-recurrence-events)=165/203

P(Irradiat = no | C=recurrence-events)=55/87

P(no-recurrence-events) = 0.7

P(recurrence-events) = 0.3

3. **Supón que llega un nuevo caso (el 289ª, ¿no?), donde el médico desconoce o duda sobre su pronóstico, y quiere saber cuál es el pronóstico que propone el clasificador "NaiveBayes" aprendido. Supongamos que el nuevo caso es: nuevo_caso = {Menopause = premeno; node-caps=yes, breast=left, irradiat=yes} Realiza "a mano", "tú mismo", para este caso, el trabajo del clasificador naive Bayes con los estadísticos "p(Xi=xi|C=c)" y "p(C=c)" antes expuestos, y calcula**

**para cada valor de la clase "p(C=c|nuevo_caso)". ¿Por qué valor de la clase apuesta el clasificador naive Bayes aprendido? Ésta es la lógica de funcionamiento de naive Bayes, trata de entenderlo.**

P(C=no-recurrence-events | nuevo_caso) $\propto$ P(C=no-recurrence-events)*P(menopause = premeno | C=no-recurrence-events)*P(node-caps = yes | C=no-recurrence-events) *P(Breast = left | C=no-recurrence-events)*P(Irradiat = yes | C=no-recurrence-events) = 0.7*(103/204)*(172/198)*(104/203)*(38/203)=0.0294437

P(C=recurrence-events | nuevo_caso) $\propto$ P(C=recurrence-events)*P(menopause = premeno | C=recurrence-events)*P(node-caps = yes | C=recurrence-events) *P(Breast = left | C=ecurrence-events)*P(Irradiat = yes | C=recurrence-events) = 0.3*(49/88)*(32/84)*(50/87)*(32/87)=0.01345199

Apostará para este nuevo caso que se tratará de la clase no-recurrence-events, es decir no necesitará tratamiento, ya que P(no-recurrence-events | nuevo_caso) >P(recurrence-events | nuevo_caso).

4. **Haz lo mismo para el siguiente caso, calculando su "p(C=no-recurrence-events|caso_nuevo)" y "p(C=recurrence-events|caso_nuevo)": nuevo_caso1={Menopause = lt40; node-caps=no, breast=right, irradiat=no}**

P(C=no-recurrence-events | nuevo_caso1) $\propto$ P(C=no-recurrence-events)*P(menopause = lt40 | C=no-recurrence-events)*P(node-caps = no |

C=no-recurrence-events) *P(Breast = right | C=no-recurrence-events)*P(Irradiat = no | C=no-recurrence-events) = 0.7*(6/204)*(172/198)*(99/203)*(167/203) = 0.00717533

P(C=recurrence-events | nuevo_caso1) $\propto$ P(C=recurrence-events)*P(menopause = lt40 | C=recurrence-events)*P(node-caps = no | C=recurrence-events) *P(Breast = right | C=recurrence-events)*P(Irradiat = no | C=recurrence-events) = 0.3*(3/88)*(52/84)*(37/87)*(55/87)=0.00170219

Pasa lo mismo en el caso anterior, apostará para este nuevo caso que se tratará de la clase no-recurrence-events, es decir no necesitará tratamiento, ya que P(no-recurrence-events | nuevo_caso) >P(recurrence-events | nuevo_caso).

5. **--> Según tu opinión, responde, ¿cómo deberían ser las probabilidades "p(Xi=xi|C=no-recurrence-events)" y "p(Xi=xi|C=recurrence-events)" para una variable Xi que ayudase a discriminar-diferenciar entre las clases del problema? ¿Parecidas entre sí? ¿Diferentes? ¿Lo ves para alguna de las predictoras que tenemos, cuál, por qué?**

No deberían de ser proporcionales ya que no estamos hablando de probabilidades. Es decir, no tienen que sumar 1 ambas.

6. **Por otro lado, responde, ¿cómo serían "p(Xi=xi|C=no-recurrence-events)" y "p(Xi_=xi|C=recurrence-events)" para una variable Xi que NO ayude a diferenciar-discriminar entre las distintas clases del problema? Da un ejemplo de entre las predictoras que tenemos.**

Serían dos probabilidades bastante parecidas.

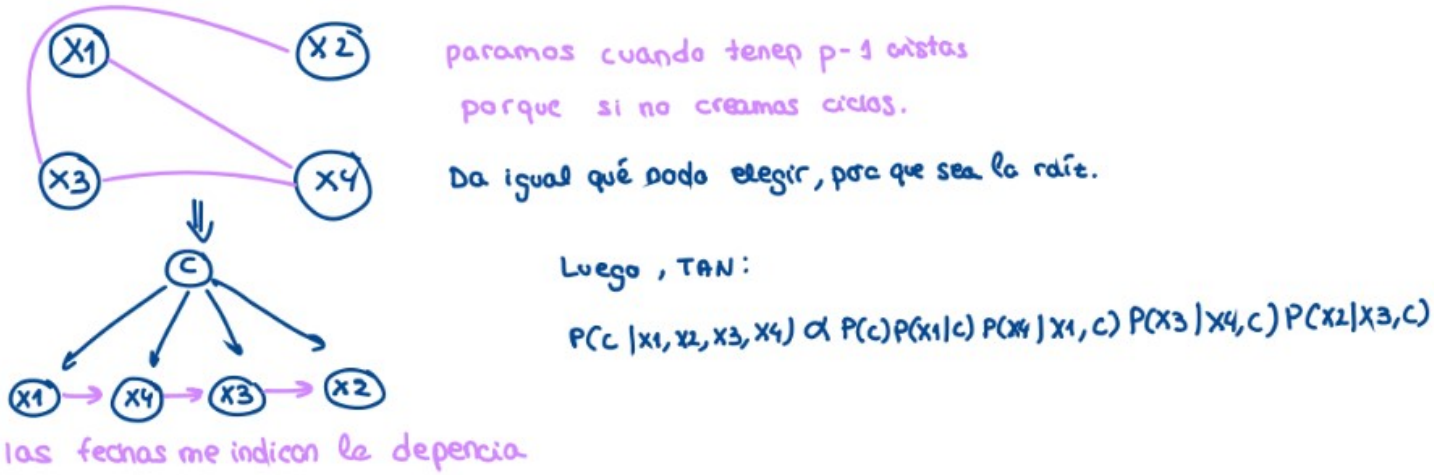
7. **--> Tarea para ti. Basándonos en las transparencias 10,11 y 12 del tema: "Naive Bayes aumentado a árbol (TAN)". Invéntate y muestra unos valores de información mutua entre cada par posible de variables condicionado a la clase (I(X_i,X_j|C)". Aplica el pseudocódigo de la transparencia 11 para construir la estructura TAN final correspondiente (transparencia 12). Dibújala. Explica el proceso.**

Modelo de clasificación que da el algoritmo TAN:

Las estimaciones de las infromaciones mutuas de pares de variables son:

I(x1,x2C|)=0.4464, I(x1,x3|C)= 0.1610, I(x1,x4|C)=0.5219, I(X2,X3|C)=0.1610, I(x2,X4|C)=0.5219, I(X3,X4|C)=0.0365

I(X1,X4|C)>I(X3,X4|C)>I(X4,X3|C)>I(X2,X3|C)>I(X2,X4|C)>I(X1,X2|C)




8. **--> Tarea para ti, refléjalo en tu documentación: busca en la web con tu intuición, y describe una aplicación que hayas encontrado y donde se haya utilizado esta técnica de naive Bayes para resolverlo. Describe el problema de clasificación que recoge.**

Por ejemplo, el gmail, marca si un mensaje es o no spam. Naive Bayes se aplica para este ejemplo, el cual se puede encontrar como filtrado bayesiano de

spam. Se suelen utilizar un conjunto de palabras características para identificar los correos spam.

Unas palabras particulares tienen determinadas probabilidades de ocurrir en el correo electrónico spam y en el corre electrónico legítimo.