

Trabajo Fin de Asignatura: Reconocimiento de Trastornos Psicológicos mediante Red Neuronal

Profesor: Manuel Martín-Merino Acera

Alumno: Jorge de Andrés González, Exp: 78509

Curso: 4º

Facultad, Grado: Informática, Ingeniería Informática

Universidad: Universidad Pontificia de Salamanca

Índice

1. Objetivo del Trabajo	3
2. Datos del Trabajo	4
a. Etiquetas del Trabajo	6
3. Procedimiento	7
4. Fundamentos Teóricos	8
a. Informática	8
b. Psicología	10
c. Estadística.....	10
5. Ejecución	11
6. Resultados	24
7. Conclusiones	31
8. Bibliografía	33

1. Objetivo del Trabajo

El objetivo del trabajo es la visualización, tratamiento y clasificación de datos, tanto mediante una red neuronal como mediante KNN. Estos datos son variables que representan a una persona (explicadas posteriormente), mientras que las clases serán el tipo de trastorno psicológico que sufren. Estos han sido encuadrados dentro de grupos, ya que los trastornos son demasiado complejos y exactos como para poder ser predichos.

Este trabajo va a estudiar la posibilidad de hacer una predicción del trastorno psicológico que una persona puede tener a partir de ciertas variables de mucha importancia a la hora de hacer una evaluación psicológica.

Para conseguir esto, se hará un estudio sobre qué variables son necesarias para llegar a conclusiones fiables, se obtendrán datos reales, se creará un DataSet, se codificará una solución de inteligencia artificial y finalmente analizaremos estos resultados obtenidos.

Palabras Clave: Trastorno Psicológico, Inteligencia Artificial, R, DataSet, Predicción

Abstract

The goal of this work is the visualization, treatment and classification of data using both a neural network and KNN. This data are variables that represent a person (later described), while the classes are the type of psychological disorders that they suffer. This have been put into groups, due to the complexity of the disorders, and also due to the complexity to be forecasted exactly.

This work will study the possibility of forecasting the psychological disorder that a person may have by some very important variables in psychological evaluations.

To achieve this, a study will be done about which variables are necessary to get to reliable conclusions, real data will be collected, a dataset will be created, an AI solution will be coded and finally this results will be analyzed.

Key Words: Psychological Disorder, Artificial Intelligence, R, DataSet, Prediction

2. Datos del Trabajo

Los datos que se utilizan en esta investigación son datos reales, de pacientes reales. Para proteger su intimidad, no se ha proporcionado ni la foto ni los apellidos de los mismos, utilizando como variables del estudio las siguientes:

- **Edad:** La edad del paciente influye en los trastornos que puede tener, ya que ciertos trastornos prácticamente no se dan con algunas edades, mientras que en ciertas edades otros trastornos son mucho más frecuentes.
 - Tratamiento de los datos: Hablando de edad, tendremos un número que simplemente será la edad de la persona. Como este es una variable que comprende un rango mucho mayor que las otras, voy a tener que hacer una pequeña conversión, consistente en un centrado y escalado que explicaré posteriormente.
- **Sexo:** El sexo, al igual que la edad, es uno de los factores clave. El sexo, aunque de una manera muy inferior, determina también en cierto modo qué trastornos puede tener, ya que los hombres y las mujeres, en algunos trastornos, tienen diferente predisposición.
 - Tratamiento de los datos: Hablando del sexo, tenemos también una variable donde será 1 si es hombre y 0 si es mujer.
- **Relación con el contexto:** Nos referimos con esto a las relaciones familiares y/o de pareja. Si las relaciones son conflictivas, es un determinante importante de determinados trastornos psicológicos. Las relaciones que no son buenas se pueden deber a diversos motivos: Desde una mala relación hasta traumas como abusos, fallecimiento, separación de los padres... Por esto tendremos que hacer un pequeño procesamiento de esta variable para adecuarla a la red neuronal.
 - Tratamiento de los datos: Hablando de relación con el contexto, es importante distinguir entre si es buena o mala, y dentro de si es mala, tenemos que distinguir entre si se lleva mal con las personas o ha sufrido una situación traumática, ya que estas dos situaciones pueden llevar a diferentes trastornos. Por ello, tendremos 3 variables.
- **Estilo educativo:** El estilo educativo es una de las mejores variables de las que disponemos, sobre todo en la infancia y adolescencia. Este se va a dividir en tres estilos principales: Estricto, normal, permisivo. Muchas patologías provienen de la educación recibida, por lo que lo he considerado como un elemento clave en esta investigación.

- Tratamiento de los datos: Como ya he comentado anteriormente, el estilo educativo es un elemento fundamental para obtener resultados verídicos en este trabajo. Así, lo dividiremos en “Estricto”, “Correcto” y “Permisivo”. Como una red neuronal no puede interpretar esto, cada uno de estos elementos pasará a ser una variable binaria en sí, de tal manera que valdrá 0 si no es el caso y 1 si lo es. De este modo, para cada línea del DataSet (sujeto estudiado) podremos tener en estas tres variables sólo un 1 y dos 0. No he tenido en consideración el codificarlo mediante una sola variable con valor lineal (0, 1, 2), ya que esto no está representando distancias euclídeas, y por lo tanto esta notación podría llevar a error.
- **Resiliencia:** La resiliencia es la capacidad que tienen las personas para superar las adversidades que se les pueden presentar en la vida, y de esta manera seguir mirando hacia delante con perspectivas de futuro. Obviamente, este factor es muy importante a la hora de caer en trastornos como la depresión y la ansiedad, y haré un pequeño tratamiento de los datos que explicaré posteriormente.
 - Tratamiento de los datos: Para hacer el tratamiento, y como en los casos anteriores, la red neuronal no es capaz de interpretar datos así, de tal manera que dividiré la resiliencia en los siguientes parámetros: “Alta”, “Media”, “Baja”. Seguiré la misma notación que en el estilo educativo, con 3 variables booleanas excluyentes, donde sólo una de las tres pueda tener un 1.
- **Habilidades Sociales:** Las habilidades sociales de la persona son fundamentales, aparte de una de las bases de la inteligencia emocional. Mediante estas, la persona se pone en contacto con otras, y da lugar a las relaciones interpersonales, que son básicas para el bienestar psicológico (equilibrio y estabilidad emocional). De este modo, las habilidades sociales serán una variable de mi estudio. También haré un pequeño tratamiento de los datos, que será explicado próximamente.
 - Tratamiento de los datos: Haré lo mismo que con las variables anteriores. Las habilidades sociales se dividirán en “inhibido”, “asertivo” y “agresivo”, y convertiremos cada una en una variable en sí, obteniendo un 1 si es el caso y un 0 si no es el caso. Al igual que en los campos anteriores, por cada sujeto estudiado solo tendremos un 1 y dos ceros.
- **Distorsiones:** Las distorsiones cognitivas son las bases del pensamiento distorsionado. Las situaciones de la vida son interpretadas mediante pensamientos, que conllevan sentimientos y los sentimientos llevan a acciones.

A veces, los pensamientos los hacemos distorsionadamente, de tal manera que nuestros sentimientos y por lo tanto acciones no serán las adaptativas, y por lo tanto, la posibilidad de sufrir un trastorno aumenta.

Esta forma de pensar, sentir y actuar es conocida como la teoría de Aaron Beck. Estas distorsiones son llamadas distorsiones cognitivas, y son 10. Para una pequeña simplificación, usaremos las 9 más usuales.

- Tratamiento de los datos: Trataremos cada distorsión cognitiva como una variable única, binaria, donde será 0 si el paciente no presenta la distorsión y 1 si presenta la distorsión. Al contrario que en las variables anteriores, un paciente puede presentar varias distorsiones cognitivas a la vez, o incluso llegar a presentar todas, por lo que en este caso estas variables no son excluyentes. Pasemos a verlas:
 - “Pensamiento Dicotómico”: Ver las cosas “de una manera booleana”, o blanco o negro.
 - “Generalización Excesiva”: Elaborar una regla general ilógica a partir de algo.
 - “Etiquetar”: Consiste en culpabilizar de manera integral a la persona, en vez de centrarse en la conducta errónea.
 - “Filtro Mental”: Solo ver la parte negativa de las situaciones, sin centrarse en la parte positiva.
 - “Magnificación y Minimización”: Magnificar los errores y minimizar los éxitos, o exagerar la gravedad de una situación determinada.
 - “Conclusiones arbitrarias”: Sacar conclusiones o hacer interpretaciones negativas no justificadas por hechos.
 - “Pseudo-responsabilidad”: Verse a sí mismo como la causa de sucesos desafortunados, de los que uno no es responsable.
 - “Los Debería”: Intentar empujarse a sí mismo con “debería”, “tengo que” y frases similares de tal manera que eso acaba en culpa, ira y resentimiento.
 - “Razonamiento Emocional”: Consiste en actuar en base a la emoción, y es muy propio de numerosos problemas psicológicos.
- **Impulsividad:** La impulsividad es la realización de acciones sin el previo análisis de los mismos. Las personas se dividirán en impulsivas o no impulsivas.
 - Tratamiento de los datos: Hablando de impulsividad, tenemos que considerar si la persona es impulsiva o no. Para ello, usará una variable booleana, y si la persona es agresiva será un 1 y si no será un cero.

- **Conclusiones Arbitrarias:** También es conocido como “lectura de pensamiento”, ya que se basa en tomar como cierto lo que creo que los demás piensan de mí.
 - Tratamiento de los datos: Al igual que he hecho con el resto de distorsiones, tendré un 1 en caso de que esta distorsión esté presente en el paciente, y un 0 en caso de que no lo esté.

Etiquetas del Trabajo

Las etiquetas vendrán dadas por los trastornos psicológicos que sufren las personas. Estos trastornos psicológicos los divido en 4 grupos para su estudio:

- Trastorno Obsesivo-Compulsivo: Obtendrá el número 1.
- Trastorno de Ansiedad: Obtendrá el número 2.
- Trastorno relacionado con Depresión: Obtendrá el número 3.
- Trastorno de Personalidad: Obtendrá el número 4.

Estos números han sido dados de manera aleatoria, ya que no importa en este caso qué número represente qué trastorno. Simplemente se van a utilizar para identificar cada uno de los grupos, y ver si las predicciones en el test son correctas o no. Estos datos se introducirán en la última columna de los datos, como una variable, y a partir de ahora los podremos denominar “etiquetas del DataSet”.

3. Procedimiento

El procedimiento que voy a seguir a la hora de hacer este estudio es el siguiente:

- 1) Obtención de los datos de los pacientes reales: Para ello, me reuniré con psicóloga profesional donde me dará los datos deseados de los pacientes (remarcando que se respetará su intimidad), y así obtendré un DataSet para empezar a trabajar. También haré todo el preprocesamiento de los datos anteriormente descrito.
 - a. Es importante remarcar que el DataSet está hecho totalmente a mano, yendo paciente a paciente de una consulta real (5 horas de trabajo).
- 2) Creación de una red neuronal multicapa, donde iré probando diferentes combinaciones hasta obtener la deseada y más óptima. Análisis de los datos.
- 3) Creación de una solución por KNN, pruebas para determinar el K óptimo, análisis de los datos.
- 4) Obtención de conclusiones del trabajo.

4. Fundamentos Teóricos

A continuación, se presentarán los fundamentos teóricos que hay que conocer para poder interpretar correctamente el trabajo:

a) Informática:

La red neuronal será creada en lenguaje R, mediante la librería “nnet”, mientras que el preprocesamiento de los datos, mediante centrado y escalado será hecho mediante la librería “caret” de R. El tipo de red neuronal que utilizaré será un perceptrón multicapa, ya que es una red estándar dentro de las predicciones en inteligencia artificial y un buen primer paso para ver qué resultados me arroja. Al ser multicapa, me permite resolver problemas que no se separan linealmente, cosa que es la limitación del perceptrón monocapa. Numerosos autores califican este tipo de redes como “aproximadores universales”, lo cual me reafirma que es un buen primer paso.

El funcionamiento de las neuronas varía según estemos entrenando a la red o prediciendo con ella:

En caso de estar entrenando a la red, la neurona recibe el input que le llega, así como que almacena el output, y actualiza su estado interno para que el output deseado y el output real sean lo más próximos posible. Con esto, conseguimos lo que se denomina como “ajuste del modelo”.

En caso de estar prediciendo con la red, las neuronas ya tienen su estado interno supuestamente ajustado, por lo que al recibir un input, hacen una predicción del mismo y obtienen un output que sacan hacia la siguiente capa de neuronas.

Estas redes neuronales utilizan un algoritmo conocido como Backpropagation o Retropropagación, que sirve para minimizar el error cometido en una red neuronal. Esto ocurre debido a que un error pequeño en el input de una neurona, al pasar por las sucesivas neuronas se puede convertir en un error mucho más grande, y esto hay que solucionarlo. Así, al final de la red neuronal se hará una función llamada “función de pérdida”. De esta manera, basándonos en la capacidad que tienen las integrales de ser descomponibles, podemos ir de delante hacia atrás retropropagando el error, hasta llegar al principio y retocar los parámetros para que, en la siguiente pasada de la red neuronal, se cometa un error menor.

Además, como es un problema de alta dimensionalidad, una red neuronal es una buena solución para este problema.

Esta red neuronal tendrá tantas neuronas de entrada como variables (24 neuronas), que la propia función ya nos creará automáticamente y tendrá tantas neuronas en la capa intermedia como queramos. Este es el factor principal con el que voy a trabajar. En la librería nnet, y función homónima, este hiperparámetro tiene como nombre “size”. Finalmente, tendrá tantas neuronas en la capa de salida como dimensiones tengan las etiquetas. En nuestro caso, solo hay una dimensión en la capa de salida, por

lo que tendré simplemente 1 neurona. Esto también lo hace también automáticamente la función nnet.

Otros parámetros con los que trabajaré son:

- Decay: Es un parámetro de regularización para la simplificación de la red neuronal. Mediante él, intentamos evitar el overfitting. Así, cuando entrenamos redes neuronales, tras cada iteración (y su correspondiente actualización) multiplicamos los pesos por un valor que suele estar entre 0.001 y 0.1. De esta manera, se previene de que los pesos crezcan demasiado.
Para las pruebas que iré haciendo, jugaré tanto con decay como sin él, y compararé resultados.
- Softmax: La función softmax es otro hiperparámetro de la función nnet. Para activarlo, simplemente tenemos que poner "softmax = T". Softmax es una función de activación que trasforma los números de entrada en la neurona en probabilidades. Estas probabilidades, como ya conocemos, deberán de sumar un total de 1 y así, esta función hace muy buenos análisis de probabilidad.
Su funcionamiento en la red radica en añadir una nueva capa, en la que se aplican los elementos probabilísticos explicados anteriormente. Esta capa tendrá tantas neuronas como la capa de resultado.
La función softmax es muy utilizada en clasificaciones, por lo que será un elemento a tener muy en cuenta a la hora de hacer el entrenamiento de la red neuronal, pero por otra parte softmax presupone que un miembro pertenece solo a una clase. Esto, como podemos comprobar, choca contra los intereses de mi problema, por lo que usaré softmax únicamente con fines de investigación, pero desecharé estos resultados a la hora de obtener la conclusión final.

Hay dos problemas principales que nos podemos encontrar en el entrenamiento del perceptrón multicapa:

- El primero de ellos es caer en mínimos locales. Esto es un gran problema, ya que caer en ellos hace que el entrenamiento no mejore, aunque no se haya alcanzado la convergencia que queríamos.
- El segundo es el overfitting. El overfitting consiste en el sobreajuste del modelo, de tal manera que el modelo vaya elemento por elemento aprendiendo a qué grupo pertenece, en vez de hacer una generalización un poco mayor que abarque al resto. Esto, aunque a primera vista pueda parecer una cosa buena, a la hora de generalizar con datos de test es un problema, ya que en vez de haber hecho un buen trabajo y adaptar la mayoría a su grupo, puede cometer errores de clasificación por "habernos centrado demasiado en hacer perfecto" el entrenamiento.

Tras el entrenamiento, así como tras la predicción, obtendré la matriz de confusión. Esta se caracteriza por ser una matriz de N filas x M columnas, y en mi caso, se puede mover N entre 1 y 4. Cada una de estas filas representará un grupo en el que la red

neuronal ha predicho que un paciente pertenece. M también se podrá mover entre 1 y 4, pero lo normal es que sea 4, ya que representa cada columna al grupo al que realmente pertenece cada persona del grupo de test.

Así, puedo ver la precisión que ha tenido la red neuronal prediciendo, y ver donde ha puesto a cada paciente del grupo de test, con lo que puedo ver qué errores son más comunes.

Una predicción será correcta si el paciente ha sido puesto en la diagonal de esta matriz, donde $N = M$.

Otro aspecto técnico referente al preprocesamiento de los datos es que no voy a hacer una simplificación de las columnas, ya que, aunque con 24 dimensiones este problema se considera de alta dimensionalidad y por lo tanto, una simplificación podría ser útil, los datos han sido escogidos a propósito para que tengan mucha significación, y esta técnica no será necesaria. De todas maneras, estudiaré la correlación de las variables en la parte práctica para poder obtener conclusiones.

Finalmente, haré una pequeña clasificación mediante KNN, ya que es otro algoritmo bastante conocido dentro de las clasificaciones, y veré los resultados que obtengo. Este algoritmo se basa en las distancias euclídeas que hay entre un punto y el resto (para comprobar la similaridad), y así ir clasificando a ciertos puntos como de un grupo o de otro. Cuanto mayor sea el K, más puntos tendrá en cuenta de los de su alrededor para clasificar a uno, y por lo tanto hará una mayor generalización, mientras que si el K es pequeño, tendremos una gran posibilidad de caer en overfitting. Como regla general, para elegir K, debemos de hacer la siguiente fórmula matemática:

$$K = \sqrt{n.total.data}$$

b) Psicología:

El elemento que tendremos que tener en cuenta más importante de psicología es que un TOC siempre lleva un cuadro de ansiedad. Debido a ello, cada vez que en la matriz de confusión un cuadro de TOC lo introduzca dentro de un cuadro de ansiedad, tendremos que interpretarlo como algo normal. Hay que prestar atención también a que esto no es recíproco, por lo que un cuadro de ansiedad no tiene por qué llevar un TOC.

c) Estadística:

Definimos correlación como la explicación de una variable a otra. Consiste en la dirección y la proporción en la que dos variables tienen el mismo comportamiento, de

tal manera que si una varía en sus valores la otra se comportará de una manera relacionada (que puede ser inversa). Dos apuntes principales sobre la correlación:

- La correlación es un valor que oscila entre 1 y -1
 - 1 -> Explicación perfecta en la misma dirección.
 - 0 -> Las variables son totalmente independientes.
 - -1 -> Las variables se explican perfectamente de forma inversa.
- Hay 3 métodos principales para calcular la correlación:
 - Método de Pearson: La correlación de Pearson se utiliza cuando las variables del problema son cuantitativas y continuas.
 - Método de Kendall: Mide la correlación entre varios grupos de N entidades cada uno.
 - Método de Spearman: La correlación de Spearman se utiliza cuando las variables del problema pueden ser tanto continuas como discretas. Es decir, las variables son totalmente aleatorias. Como este método se ajusta un poco más que los otros dos hacia mi problema, usaré este método para calcular la correlación.

5. Ejecución:

Mi primer paso es la obtención de datos a mano entrevistando a una psicóloga profesional, mediante la cual obtengo numerosos pacientes para conformar el DataSet. De cada paciente, obtengo 25 variables, más el grupo de trastorno psicológico al que realmente pertenece.

Antes de comenzar la codificación, tengo que comprobar que el DataSet está correctamente creado. Para ello, compruebo primero todos los datos manualmente, y los introduzco dentro de un ".txt".

Lo primero que hago es importar el DataSet en el environment, mediante la función "read.csv". Utilizo esta y no read.csv2 porque en esta última el separador por defecto es el punto y coma, mientras que el DataSet que he creado está separado sólo por comas.

Esta diferencia se utiliza cuando se separan los decimales por coma, pero en mi caso no tengo decimales, y si los tuviera los separaría mediante un punto (notación americana).

Una vez que he importado el DataSet, tengo que hacer un preprocesado para preparar los datos para la red neuronal.

Este preprocesado se basa en lo siguiente:

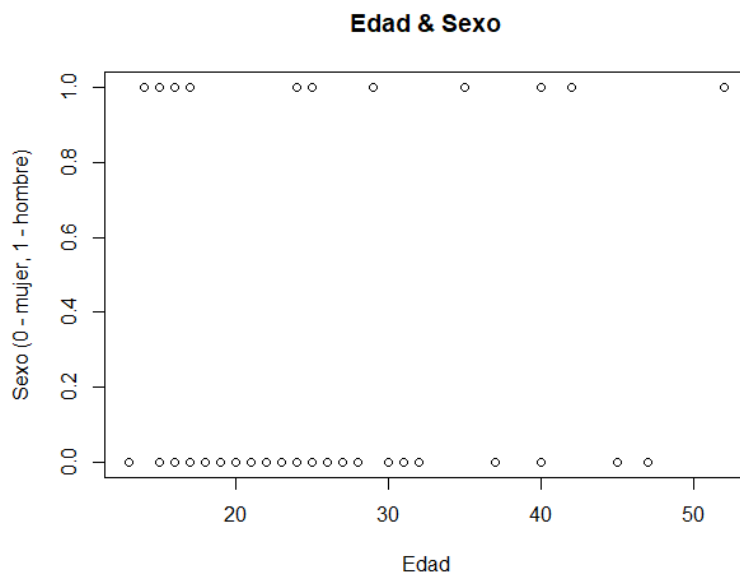
- Eliminación de columnas superfluas para la red neuronal:
 - o Columna con los nombres (no afectarán en la predicción)
 - o Columna con las etiquetas (por ahora no las necesitaremos, y las dejaré en otra matriz)
- Comprobación de los tipos de los datos, para evitar errores.
- Centrado y Escalado de los datos para evitar datos que sobresalgan sobre otros, y así tener todas las variables iguales ante la red neuronal.

Además, mediante la función summary veo las estadísticas principales de cada columna.

Junto con este preprocesado, ahora voy a hacer una visualización de los datos que tenemos.

Edad Y Sexo

Como podemos apreciar en esta gráfica, en este estudio con datos reales, las mujeres que acuden a consulta son jóvenes, estando alrededor de los 21 años, mientras que los hombres que acuden a consulta son menos, y su distribución es mucho más esparcida por el espectro de la edad.



Para la correcta interpretación de las siguientes gráficas tenemos que tener en cuenta los siguientes datos, ya que no hay la misma cantidad de pacientes con el mismo trastorno ni con las mismas características:

- Pacientes Impulsivos: 41
- Pacientes Agresivos: 14
- Pacientes Inhibidos: 42

Agresividad E Impulsividad

La siguiente gráfica que muestro es la que me obtiene la relación entre la agresividad y la impulsividad:

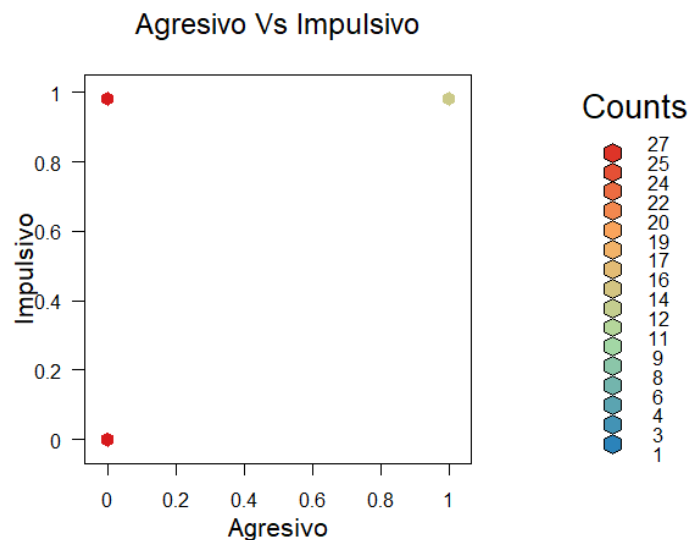
Lo que podemos apreciar en esta gráfica es que hay una gran cantidad de pacientes que no son agresivos (es decir, son inhibidos o asertivos), pero que son impulsivos.

También podemos ver que hay gran cantidad de pacientes que tampoco son agresivos pero que no son impulsivos.

Es decir, a partir de no ser agresivo no podemos inferir si un paciente es impulsivo o no.

Finalmente, vemos un conjunto de pacientes de tamaño medio que son agresivos e impulsivos, pero nos damos cuenta de que, en este estudio, una persona agresiva siempre va a ser impulsiva, no existen casos en los que no se da.

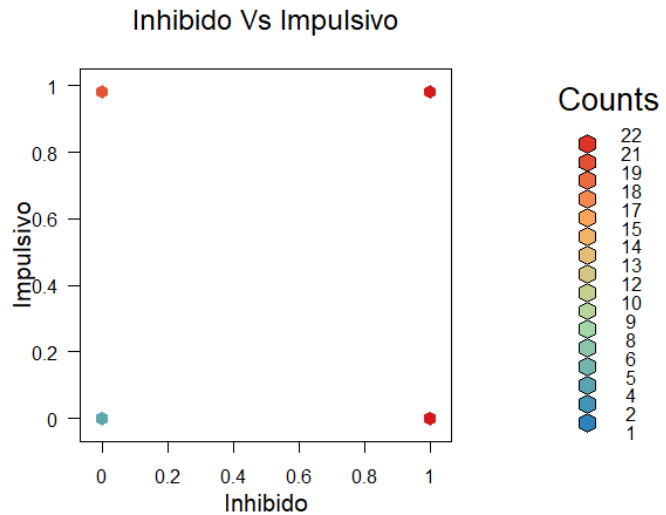
Obviamente, con esta gráfica y con las siguientes estamos obteniendo los datos de personas en la consulta, pero no son datos poblacionales. Con esto quiero remarcar que, aunque pacientes agresivos e impulsivos aparecen en la gráfica con una menor cantidad, esto no quiere decir que en la sociedad esta proporción sea así, sino que simplemente en la consulta las cantidades son así.



Inhibición e Impulsividad

En la siguiente gráfica muestro la relación entre inhibición e impulsividad:

Como podemos ver, aquí en la consulta hay pacientes de todo tipo. Principalmente podemos ver que hay una gran cantidad de pacientes impulsivos, ya que los dos puntos referentes a ello están en rojo (es decir, tanto de personas no inhibidas como de personas que lo son). Esto quiere decir que una persona inhibida puede ser impulsiva o no. Lo que es



también interesante destacar con esta gráfica es que hay muchas personas también en la consulta que no son impulsivas y son inhibidas.

Si nos damos cuenta, la gráfica de “Agresivo Vs Impulsivo” y la gráfica de “Inhibido vs Impulsivo” son prácticamente simétricas. Esto tiene sentido, ya que, si obviamos al pequeño reducto de pacientes asertivos que tenemos, el no ser agresivo conlleva prácticamente seguro ser inhibido, y el no ser inhibido conlleva con gran probabilidad el ser agresivo.

Así, por ejemplo, una persona en la primera tabla con agresivo = 0 e impulsivo = 1, podríamos a grosso modo generalizar que esa misma persona estaría en la segunda tabla con inhibido = 1 e impulsivo = 0, lo cual aproximadamente se cumple.

Al igual, vemos en la segunda tabla que inhibido = 0 e impulsivo = 1 lleva un color más anaranjado, lo cual significa que hay menos pacientes en esa condición. Si lo llevamos a la primera tabla, vemos que agresivo = 1 e impulsivo = 1 conlleva una menor cantidad de pacientes.

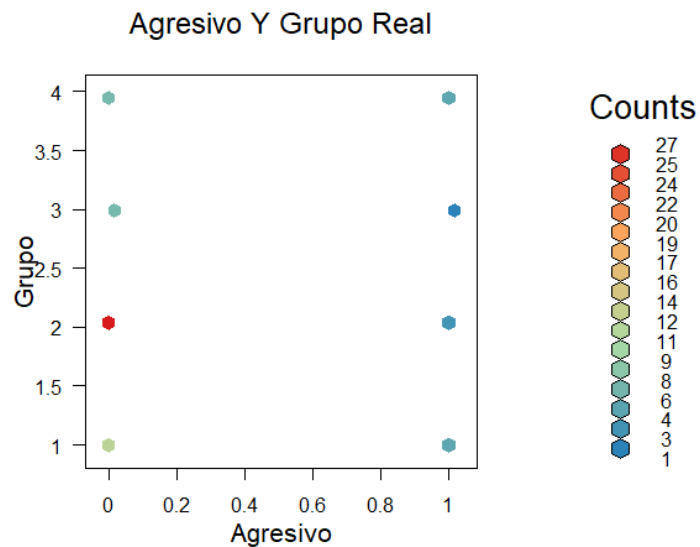
Finalmente, podemos observar en la primera tabla como agresivo = 1 e impulsivo = 0 no tiene representación, y si esto lo llevamos a la segunda tabla (siguiendo con la generalización) se traduciría en inhibido = 0 e impulsivo = 0, de lo que vemos que hay muy poca representación (unos 3 pacientes).

Agresividad y Grupo de Pertenencia

En la siguiente gráfica muestro la relación entre la agresividad y el grupo al que realmente pertenecen estas personas:

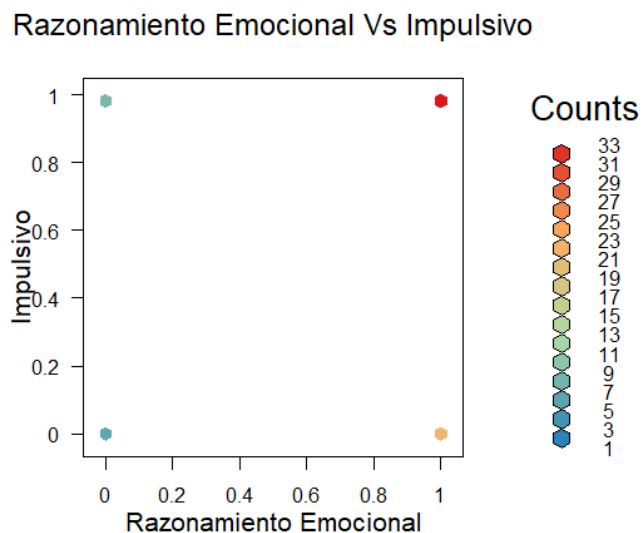
Esta gráfica es realmente interesante, ya que vemos mucha variabilidad en la relación entre agresividad y grupo de trastorno. Principalmente podemos obtener dos conclusiones:

- 1) Las personas no agresivas suelen estar encuadradas en algún trastorno psicológico que tiene que ver con la ansiedad.
- 2) Las personas no agresivas, aunque en menor medida, suelen estar relacionadas con TOCs.



Razonamiento Emocional e Impulsividad

Una gráfica también muy interesante es la comparación del razonamiento emocional con la impulsividad. Para recordar, el razonamiento emocional consiste en actuar según tus emociones del momento, y la impulsividad consiste en la realización de acciones sin un previo razonamiento de los mismos. Es decir, voy a comparar si en los pacientes de la consulta, el hecho de llevarse por un sentimiento hace que no piensen en cómo van a actuar.



Lo primero que podemos observar en este gráfico es que una amplísima mayoría de los pacientes en consulta tienen el razonamiento emocional como una de sus distorsiones cognitivas. Dentro de estos, una mayoría son impulsivos, por lo que en reglas generales se cumple.

Dicho esto, también es verdad que un cierto grupo de personas que poseen razonamiento emocional no son impulsivas, por lo que no podemos inferir que una persona, por tener razonamiento emocional, vaya a ser automáticamente impulsiva, aunque tenga más probabilidades.

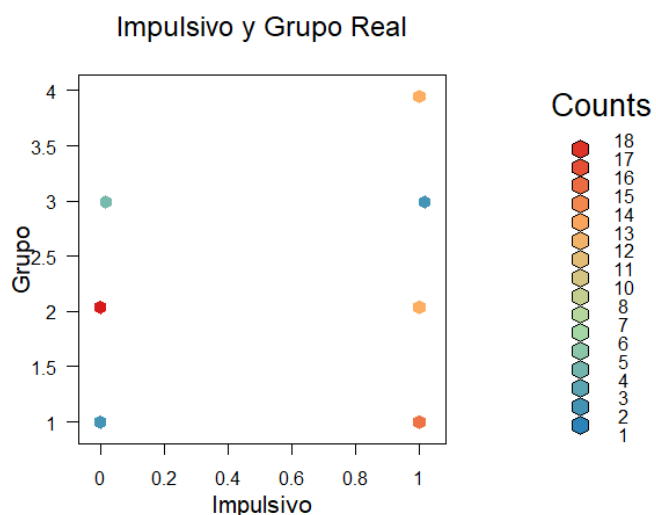
Impulsividad y Grupo Real

Finalmente, voy a sacar otro gráfico para comparar la impulsividad con el grupo real al que pertenecen los pacientes, para ver si me da datos más reveladores que la gráfica anterior:

Como podemos observar, aquí hay más variabilidad.

Conclusiones que podemos obtener del siguiente gráfico:

- 1) Las personas no impulsivas suelen estar encuadradas en trastornos relacionados con la ansiedad.
- 2) Las personas impulsivas suelen estar encuadradas, en mayor medida, en TOCs.
- 3) Las personas con trastorno de personalidad que están en la consulta son todos impulsivos.



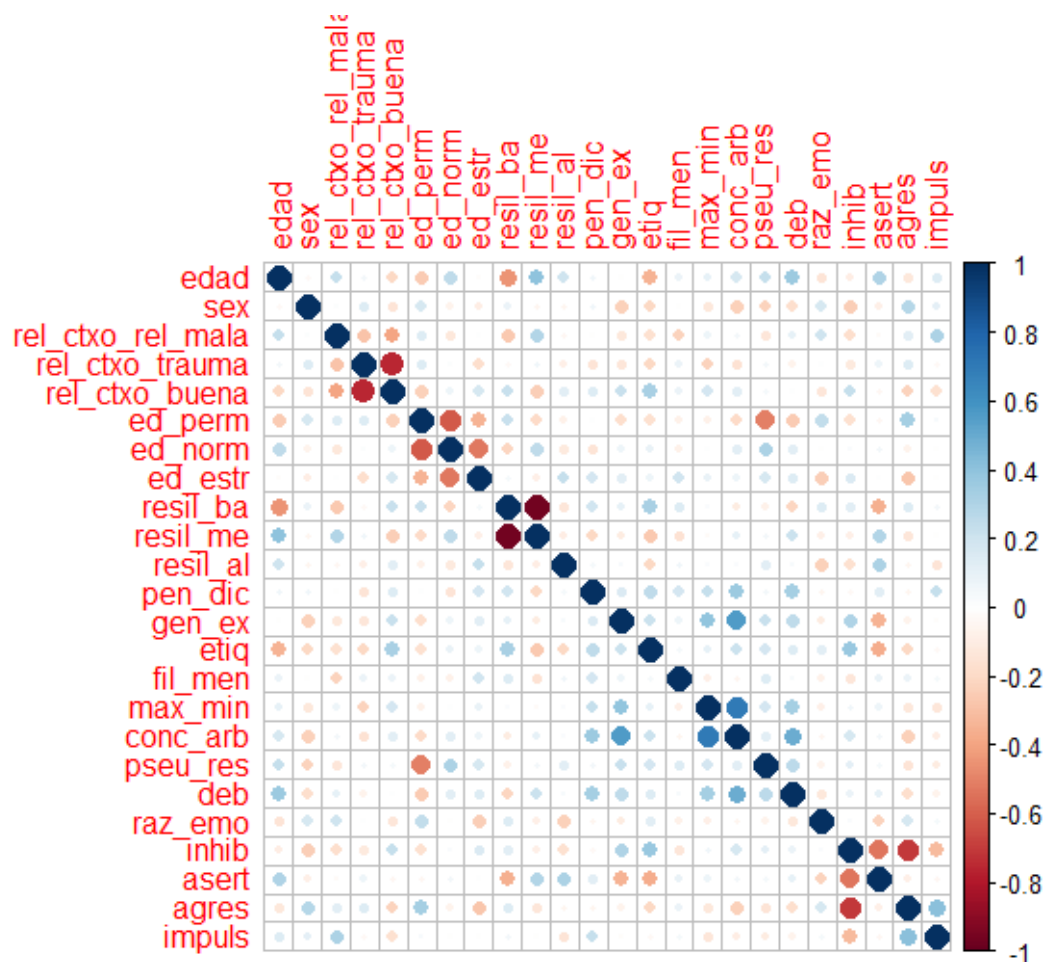
Como conclusiones conjuntas de los dos gráficos anteriores obtenemos:

- 1) Las personas que pertenecen al grupo de TOC son personas principalmente impulsivas, y con una ligera tendencia a no ser agresivas.
- 2) Las personas que pertenecen al grupo de la ansiedad no son agresivas en una amplia mayoría, al igual que por una mayoría no son impulsivas, habiendo aquí mayor variabilidad.

- 3) Las personas que pertenecen al grupo de la depresión son personas que en su mayoría no son agresivas, al igual que en su mayoría no son impulsivas.
- 4) Las personas que sufren trastornos de la personalidad son personas que en su mayoría no son agresivas, pero todas estas personas son muy impulsivas.

Ahora que he hecho un preprocesado de los datos, he comprobado que todo está correctamente y he hecho una primera visualización de los datos para ver relaciones que a primera vista no se ven, vamos a ver la correlación de todos los datos.

Esta correlación, como vimos anteriormente, la debía de hacer mediante Spearman para una mayor corrección, y el resultado obtenido es el siguiente:



Como se puede observar, podemos sacar algunas conclusiones de estas correlaciones. Estas correlaciones que he obtenido las puedo dividir en dos grupos:

- 1) Correlaciones Lógicas: Me refiero con correlaciones lógicas a aquellas que se tienen que dar a la fuerza, porque se refieren a variables excluyentes. Vemos por ejemplo:

- a. Relación Con El Contexto Trauma y Relación Con El Contexto Buena:
Obviamente si tenemos una relación con el contexto buena no tendremos un trauma, mientras que si tenemos un trauma no tendremos una relación con el contexto buena. Debido a ello, vemos la fuerte correlación negativa que tienen.
 - b. La resiliencia baja y la resiliencia media:
Como es lógico, una persona con resiliencia baja no tendrá una resiliencia media, y viceversa. Por ello, también tienen una correlación negativa muy fuerte (-1).
 - c. Agresividad e Inhibición:
Una persona agresiva no puede ser inhibida, y viceversa. Con esto obtendremos una correlación negativa de nuevo.
- 2) Correlaciones Interesantes: Con estas correlaciones me estoy refiriendo a aquellas con las que podemos obtener información que no es tan obvia.
- a. Educación Permisiva y Pseudo-Responsabilidad:
Tienen una correlación negativa con un valor de aproximadamente 0,45. Esto lo que nos está diciendo es que las personas con una educación que ha sido normal o estricta tienden a tener mayor pseudo-responsabilidad que aquellas personas que han sido educados muy permisivamente.
 - b. Conclusiones Arbitrarias y Generalización Excesiva:
Existe una fuerte correlación positiva entre estas dos distorsiones cognitivas. Es decir, las personas que cometen generalizaciones excesivas suelen sacar conclusiones arbitrarias y viceversa.
 - c. Debería y Conclusiones Arbitrarias:
Existe una correlación de aproximadamente 0,5 entre estas dos variables. Esto significa que muchos pacientes que cometen el error de obtener conclusiones arbitrarias, también se culpan por sistema con “debería de haber hecho...”.

Como podemos ver, las relaciones entre las variables no son demasiado altas, a excepción de las correlaciones lógicas que hemos obtenido anteriormente. Esto significa que, aunque hubiera querido hacer una simplificación de los datos, no hubiera sido posible simplificar demasiado debido a la gran variabilidad de los mismos.

Como apunte matemático, me gustaría destacar que esta matriz es simétrica, por lo tanto con la mitad superior o la inferior hubiera bastado. También que todos los valores de la diagonal son 1, ya que la correlación de una variable consigo misma es 1.

Una vez que hemos hecho todo esto, podemos pasar a la predicción. Para estas predicciones, voy a usar una red neuronal multicapa, ya que con la cantidad de variables que tenemos una red monocapa sería demasiado simple.

Para esta red multicapa, usaré la librería nnet de R.

El procedimiento que seguiré será el siguiente para cada prueba:

- 1) Entrenamiento de la red neuronal sin softmax, obtención de resultados de entrenamiento.
- 2) Acción de los datos de test y obtención de resultados.
- 3) Apuntar los mejores resultados en la tabla de resultados finales en el apartado 6.
- 4) Entrenamiento de la red con softmax, obtención de resultados de entrenamiento.
- 5) Acción de los datos de test y obtención de resultados.
- 6) Apuntar los mejores resultados en la tabla de resultados finales.
- 7) (Opcional) Repetir el proceso con otros parámetros.

Siguiendo esto, vamos a empezar con dos neuronas en la capa intermedia, y según lo que vayamos viendo iremos variando la cantidad.

- 2 Neuronas, Sin Softmax:

Obtengo un entrenamiento bastante bueno, pero un resultado de test del 50% nada más.

Esto me hace pensar en overfitting, pero con solo dos neuronas me parece algo un poco descabellado. Vamos a ver con las siguientes pruebas a ver que obtenemos.

- 2 Neuronas, Con Softmax:

Vuelvo a obtener un entrenamiento bastante bueno, pero vuelvo a obtener un test bastante pobre.

Para comprobar que no sea culpa del overfitting, voy a probar con solo 1 neurona en la capa intermedia, a ver como se comporta el sistema.

- 1 Neurona, Sin Softmax:

Obtengo un resultado interesante, ya que el entrenamiento ha caído hasta un máximo del 76%. En cambio, en el test obtengo un 50% de acierto, por lo que estamos generalizando igual de bien con menos neuronas.

- 1 Neurona, Con Softmax

El entrenamiento con softmax ha llegado a tener el mismo acierto que el entrenamiento sin softmax, de un 76%. Al hacer el test, obtenemos también un máximo del 50% de acierto.

Por ahora, ante la igualdad, podemos decir que nos quedaríamos con la red con solo 1 neurona, ya que obtenemos mejores resultados con una red más sencilla, donde encima el overfitting es menor, por lo que la generalización del entrenamiento es mejor.

Vamos a hacer unas pruebas con más neuronas, para ver como se comporta el sistema, ver si cae en overfitting y si podemos solucionar esto con el hiperparámetro “decay” para obtener mejores resultados.

- 5 Neuronas, Sin Softmax, Sin Decay

En entrenamiento suele rondar el 80% de acierto, pero he llegado a alcanzar un acierto del 94%. Hay serias posibilidades de haber caído en overfitting, lo comprobaremos con los datos de test:

Obtenemos un máximo de un 50% de acierto, pero solo en momentos en los que la red neuronal baja a un acierto de test del 67%, por lo que parece bastante fiable, y que generaliza bien.

- 5 Neuronas, Con Softmax, Sin Decay

Consigo entrenamientos todos superiores al 90% de acierto, mientras que he conseguido un máximo del 98% de acierto. Siguiendo la máxima de: “Desconfía de todo lo perfecto, sobre todo con datos reales”, creo que estamos en overfitting. De todas formas, vamos a comprobarlo con los datos de test...

En test obtengo datos similares a lo anterior, con un 58% de acierto máximo.

Está claro a la vista de estos datos que el aumento de neuronas, con el parámetro softmax, no le ha venido bien al sistema. Debido a esto, en las siguientes pruebas voy a probar con el parámetro Decay y con un número de neuronas intermedio, de tal forma que podamos ir ajustando.

Es importante remarcar que el parámetro decay es extremadamente sensible, por lo que voy a hacer numerosas pruebas, y los resultados que publicaré en la tabla final en el apartado 6 serán con el mejor decay hallado. Para hacer totalmente transparente el proceso, todos los decays que probaré los enseñaré a continuación.

- 5 Neuronas, Sin Softmax, Con Decay

Mejores Entrenamientos y Mejores Tests (% de acierto)

	Entrenamiento	Test
Decay = 0,001	98%	41%
Decay = 0,005	98%	41%
Decay = 0,01	98%	41%
Decay = 0,03	98%	41%
Decay = 0,05	96%	50%
Decay = 0,075	94%	41%
Decay = 0,1	90%	50%
Decay = 0,15	89%	41%

Como podemos ver, con un decay del 0.05 o del 0.1 obtenemos los mejores resultados. No es casualidad que sean valores altos que nos permitan evitar el overfitting, ya que provocan una mayor regularización y por lo tanto generalizan mejor.

Debido a esto, voy a seguir aumentando el valor del decay, para poder comprobar si un mayor aumento ayuda a mejorar el resultado:

	Entrenamiento	Test
Decay = 0,2	85%	41%
Decay = 0,3	80%	41%

Como se puede ver, simplificar aún más el problema con mayor decay no está ayudando.

- 5 Neuronas, Con Softmax, Con Decay

A la vista de los resultados anteriores, y tal como todo apuntaba, un decay demasiado pequeño no va a hacer nada. De esta manera, empezaré a hacer las pruebas con un decay más grande ahora.

Mejor Entrenamiento y Mejor Test (% de acierto)

	Entrenamiento	Test
Decay = 0,01	98%	41%
Decay = 0,05	98%	50%
Decay = 0,075	98%	41%
Decay = 0,1	98%	50%
Decay = 0,15	96%	41%
Decay = 0,2	92%	41%
Decay = 0,3	89%	41%
Decay = 0,4	87%	41%
Decay = 0,5	83%	41%
Decay = 0,6	81%	41%

En este caso, con el softmax activado, nos está pasando lo mismo. Una simplificación está viniendo bien, pero llega un momento donde esta simplificación está siendo contraproducente, porque no está ayudando a mejorar el problema, está dejándolo estancado en un acierto que no aumenta. Incluso estoy viendo que, en los sucesivos entrenamientos a los que someto la red neuronal, está obteniendo de media valores más bajos.

Como vemos, con 5 neuronas no estamos teniendo un gran resultado. Debido a ello, tenemos que buscar soluciones. Lo que se me ocurre es que, ya que con decay estamos obteniendo mejores resultados, vamos a bajar el número de neuronas de la capa intermedia a 3, a ver si obtenemos unos resultados más satisfactorios:

- 3 Neuronas, Sin Softmax, Sin Decay

Obtengo en entrenamiento un máximo de un 90% de acierto, lo cual es bastante interesante. En test obtengo un máximo del 50% de acierto.

- 3 Neuronas, Con Softmax, Sin Decay

Obtengo un máximo de 96% de acierto en entrenamiento, lo cual es mucho. En cambio, en test, obtengo también un máximo de un 50 % de acierto.

En este momento, podemos darnos cuenta de que la simplificación de la red neuronal, al bajarle neuronas, ha venido bien para tener una mejor predicción. Vamos a jugar un poco con el decay, a ver si obtenemos los mismos resultados que con 2 y 1 neurona, o incluso los mejoramos con este parámetro de regularización.

	Entrenamiento	Test
Decay = 0,001	90%	58%
Decay = 0,005	92%	50%
Decay = 0,01	94%	50%
Decay = 0,03	92%	58%
Decay = 0,05	94%	41%
Decay = 0,075	90%	50%
Decay = 0,1	89%	50%
Decay = 0,2	80%	41%

- 3 Neuronas, Sin Softmax, Con Decay

Al igual que he hecho anteriormente, voy a ir jugando con el decay con cuidado y voy a obtener resultados, que iré poniendo a continuación:
Mejor Entrenamiento y Test (% de acierto)

	Entrenamiento	Test
Decay = 0,001	89%	58%
Decay = 0,005	90%	50%
Decay = 0,01	92%	75%
Decay = 0,03	90%	58%
Decay = 0,05	90%	58%
Decay = 0,075	89%	50%
Decay = 0,1	85%	50%
Decay = 0,2	76%	58%

- 3 Neuronas, Con Softmax, Con Decay

Mejor Entrenamiento y Test (% de acierto)

	Entrenamiento	Test
Decay = 0,001	96%	41%
Decay = 0,005	92%	41%
Decay = 0,01	98%	50%
Decay = 0,03	96%	58%
Decay = 0,05	94%	41%
Decay = 0,075	92%	41%
Decay = 0,1	94%	41%
Decay = 0,2	90%	41%

Finalmente, para la visualización de los resultados, he creado un fichero llamado “resultados.txt”, en el que muestro los mejores resultados obtenidos con cada set-up

de la red neuronal (son los mismos datos que los que muestro en la segunda tabla del siguiente apartado).

Así, voy a obtener un gráfico que me permita tomar, de una manera más visual, una decisión de qué opción es la mejor. Este gráfico, en la ejecución de R, es interactivo. Para la visualización en la memoria, he obtenido una captura del mismo, que lo veremos al final del siguiente apartado.

A continuación, voy a hacer una pequeña clasificación mediante K Nearest Neighbours (KNN). Aquí lo que debo de hacer es elegir un K, y por estándar se suele elegir $K = \sqrt{totalDatos}$. Por no alargar demasiado el trabajo, podré directamente los resultados en el apartado de resultados para su visualización e interpretación.

6. Resultados:

Ahora voy a exponer los mejores resultados obtenidos de **las redes neuronales**. Estos resultados se basan en los resultados obtenidos en el test, y muestran lo siguiente:

La columna de entrenamiento muestra el porcentaje de acierto que se tenía en la pasada en la que se consiguió esa puntuación de test. Habiendo dicho esto, este resultado no tiene por qué coincidir con el que se haya mostrado anteriormente en las pruebas, ya que ese resultado simplemente mostraba “el mejor entrenamiento”, pero no “la mejor generalización”, que es en lo que nos estamos centrando principalmente ahora.

La columna de test muestra, en tanto por uno, el mejor resultado conseguido con esa configuración de la red neuronal.

	Entrenamiento	Test
2 Neuronas, Sin Softmax, Sin Decay	0,89	0,50
2 Neuronas, Con Softmax, Sin Decay	0,87	0,41
1 Neurona, Sin Softmax, Sin Decay	0,76	0,50
1 Neurona, Con Softmax, Sin Decay	0,76	0,50
5 Neuronas, Sin Softmax, Sin Decay	0,67	0,50
5 Neuronas, Con Softmax, Sin Decay	0,94	0,58
5 Neuronas, Sin Softmax, Con Decay (0.1)	0,90	0,41
5 Neuronas, Con Softmax, Con Decay (0.05)	0,96	0,50
3 Neuronas, Sin Softmax, Sin Decay	0,90	0,50
3 Neuronas, Con Softmax, Sin Decay	0,96	0,50
3 Neuronas, Sin Softmax, Con Decay (0,01)	0,80	0,75
3 Neuronas, Con Softmax, Con Decay (0,03)	0,96	0,58

Es importante tener en cuenta de que estos resultados son los de la inclusión exacta en un grupo. Con esto quiero hacer hincapié en que no se ha tenido en cuenta la matriz de confusión para ver los “errores” al incluir los casos de TOC dentro de los de ansiedad.

Debido a ello, y por no alargar demasiado el trabajo (ya que, para hacerlo de una forma óptima, en cada prueba habría que mirarlo), voy a ver las matrices de correlación de cada uno de los set-ups de la tabla de mejores resultados, de tal manera que pueda ver si algunos elementos del grupo 1 se han incluido dentro del grupo 2, y por lo tanto se deberían de contar como acierto para la red neuronal.

- 1 Neurona, Sin Softmax

```
> table( pacientes.prediccion.test.1neu.class , matriz.pacientes.etiquetas[~-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.1neu.class 1 2 3 4
1 2 0 0 1
2 2 4 1 2
> sum( diag( table( pacientes.prediccion.test.1neu.class, matriz.pacientes.etiquetas[~-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.5
```

Como podemos ver, ha colocado correctamente 2 pacientes en el grupo 1, y 4 pacientes en el grupo 2. El resto de pacientes ha fallado. Respecto a nuestra investigación, podemos ver como ha colocado a 2 pacientes en el grupo 2 cuando pertenecían realmente al 1.

Si seguimos lo expuesto en los elementos técnicos psicológicos, un cuadro de TOC (1) siempre lleva un cuadro de ansiedad (2). Es decir, un paciente del grupo 1 encuadrado en el grupo 2 sería correcto a efectos técnicos.

Esto ha ocurrido en dos pacientes, cuyo grupo correcto es el 1 pero han sido encuadrados en el grupo 2.

Debido a esto, el acierto de la red se aumenta de 6 de 12 (50% de acierto) a 8 de 12, y por lo tanto aumentamos este acierto al 66% de acierto.

Esto lo haré con las sucesivas pruebas, y publicaré únicamente ya el resultado final:

- 1 Neurona, Con Softmax

```
> table( pacientes.prediccion.test.1neu.class , matriz.pacientes.etiquetas[~-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.1neu.class 1 2 3 4
1 2 0 0 1
2 2 4 1 2
> sum( diag( table( pacientes.prediccion.test.1neu.class, matriz.pacientes.etiquetas[~-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.5
```

Como podemos ver, tenemos 6 pacientes correctamente clasificados, lo cual nos da un 50% de acierto. Añadimos 2 pacientes que son del grupo 1 y los ha clasificado en el grupo 2, haciendo un acierto final del 66%.

50% → 66%

- 2 Neuronas, Sin Softmax

```
> table( pacientes.prediccion.test.2neu.class , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.2neu.class 1 2 3 4
                                     1 2 0 0 1
                                     2 2 4 1 2
> sum( diag( table( pacientes.prediccion.test.2neu.class, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.5
```

Tenemos 2 pacientes correctamente clasificados en el grupo 1, y 4 pacientes correctamente clasificados en el grupo 2. Añadimos dos pacientes que los ha clasificado en el grupo 2, a pesar de pertenecer al 1. Por lo tanto, obtenemos un 66% de acierto final.

50% → 66%

- 2 Neuronas, Con Softmax

```
> table( pacientes.prediccion.test.2neu.class , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.2neu.class 1 2 3 4
                                     1 1 0 0 1
                                     2 2 4 1 2
                                     4 1 0 0 0
> sum( diag( table( pacientes.prediccion.test.2neu.class, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.4166667
```

Tenemos 1 paciente correctamente clasificado en el primer grupo, y 4 pacientes correctamente clasificados en el segundo grupo. Tenemos que añadir dos pacientes clasificados en el grupo dos a pesar de pertenecer al 1, por lo que obtenemos un total del 58% de acierto.

41% → 58%

- 3 Neuronas, Sin Softmax, Sin Decay

```
> table( pacientes.prediccion.test.3neu.class , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.3neu.class 1 2 3 4
                                     1 3 1 0 2
                                     2 1 3 1 1
> sum( diag( table( pacientes.prediccion.test.3neu.class, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.5
```

Tenemos 3 paciente correctamente clasificados en el grupo 1, y 3 pacientes correctamente clasificados en el grupo 2. Así, obtenemos un 50% de acierto. Añadimos 1 paciente que lo ha clasificado en el grupo 2 a pesar de pertenecer al 1, y obtenemos un total del 58% de acierto.

50% → 58%

- 3 Neuronas, Con Softmax, Sin Decay

```
> table( pacientes.prediccion.test.3neu.class , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.3neu.class 1 2 3 4
1 2 1 0 0
2 2 3 0 2
3 0 0 1 1
> sum( diag( table( pacientes.prediccion.test.3neu.class, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.5
```

Tenemos dos pacientes correctamente clasificados en el grupo 1, 3 pacientes correctamente clasificados en el grupo 2, y 1 paciente correctamente clasificado en el grupo 3. Esto nos hace un 50% de acierto, pero si añadimos 2 pacientes que están clasificados dentro de la ansiedad que deberían de estar en TOC, obtenemos un 66% de acierto.

50% → 66%

- 3 Neuronas, Sin Softmax, Con Decay (0,001)

```
> table( pacientes.prediccion.test.3neu.class.decay , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.3neu.class.decay 1 2 3 4
1 2 0 0 0
2 0 4 0 0
3 1 0 0 1
4 0 0 1 3
> sum( diag( table( pacientes.prediccion.test.3neu.class.decay, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.75
```

Aquí tenemos 2 pacientes correctamente clasificados en TOC, 4 pacientes correctamente clasificados en ansiedad, y 3 pacientes clasificados correctamente en trastornos de personalidad. Ningún paciente de TOC ha sido clasificado como ansiedad, por lo que nos quedamos con el 75% de acierto de la red.

75% → 75%

- 3 Neuronas, Con Softmax, Con Decay (0,03)

```
> table( pacientes.prediccion.test.3neu.class.decay , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.3neu.class.decay 1 2 3 4
1 2 0 0 0
2 2 3 0 1
3 0 1 1 1
4 0 0 0 1
> sum( diag( table( pacientes.prediccion.test.3neu.class.decay, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.5833333
```

Tenemos 2 pacientes correctamente clasificados en el grupo 1, 3 pacientes correctamente clasificados en el grupo 2, 1 paciente correctamente clasificado en el grupo 3 y finalmente 1 paciente correctamente clasificado en el grupo 4. Esto nos da un 58% de acierto, al que hay que añadir dos pacientes clasificados como ansiosos que pertenecen al grupo de TOC realmente, dando un 75% de acierto final.

58% → 75%

- 5 Neuronas, Sin Softmax, Sin Decay

```
> table( pacientes.prediccion.test.5neu.class , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.5neu.class 1 2 3 4
                                     1 2 0 0 1
                                     2 2 4 1 2
> sum( diag( table( pacientes.prediccion.test.5neu.class, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.5
```

Tenemos 2 pacientes correctamente clasificados en el grupo 1, y 4 pacientes correctamente clasificados en el grupo 2, lo que nos da un 50% de acierto. Además, añadimos a 2 pacientes del grupo de ansiedad que realmente pertenecen al grupo 1, aumentando el acierto al 66%.

50% → 66%

- 5 Neuronas, Con Softmax, Sin Decay

```
> table( pacientes.prediccion.test.5neu.class , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.5neu.class 1 2 3 4
                                     1 3 0 0 2
                                     2 1 4 1 1
> sum( diag( table( pacientes.prediccion.test.5neu.class, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.5833333
```

Tenemos aquí 3 pacientes correctamente clasificados en el grupo 1, y 4 pacientes correctamente clasificados en el grupo 2. Añadimos un paciente más de ansiedad que pertenece a TOC, obteniendo un 66% de acierto.

58% → 66%

- 5 Neuronas, Sin Softmax, Con Decay (0,1)

```
> table( pacientes.prediccion.test.decay.5neu.class , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.decay.5neu.class 1 2 3 4
                                           1 1 0 0 1
                                           2 2 4 1 2
                                           4 1 0 0 0
> sum( diag( table( pacientes.prediccion.test.decay.5neu.class, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.4166667
```

Tenemos un paciente correctamente clasificado en el grupo 1 y 4 pacientes correctamente clasificados en el grupo 2. Esto nos da un 41% de acierto, al que añadimos 2 pacientes clasificados como ansiosos que realmente pertenecen a TOC, por lo que aumentamos el acierto al 58%.

41% → 58%

- 5 Neuronas, Con Softmax, Con Decay (0,05)

```
> table( pacientes.prediccion.test.decay.5neu.class , matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] )
pacientes.prediccion.test.decay.5neu.class 1 2 3 4
                                           1 1 0 0 2
                                           2 2 4 0 1
                                           3 0 0 1 0
                                           4 1 0 0 0
> sum( diag( table( pacientes.prediccion.test.decay.5neu.class, matriz.pacientes.etiquetas[-conjuntoEntrenamiento, 25] ) ) )/12
[1] 0.5
```

Finalmente, obtenemos 1 paciente correctamente clasificado en el grupo 1, 4 pacientes correctamente clasificados en el grupo 2, y 1 paciente correctamente clasificado en el grupo 3. Añadimos 2 pacientes clasificados como del grupo de la ansiedad que realmente pertenecen a TOC, y obtenemos un total del 66% de acierto.

50% → 66%

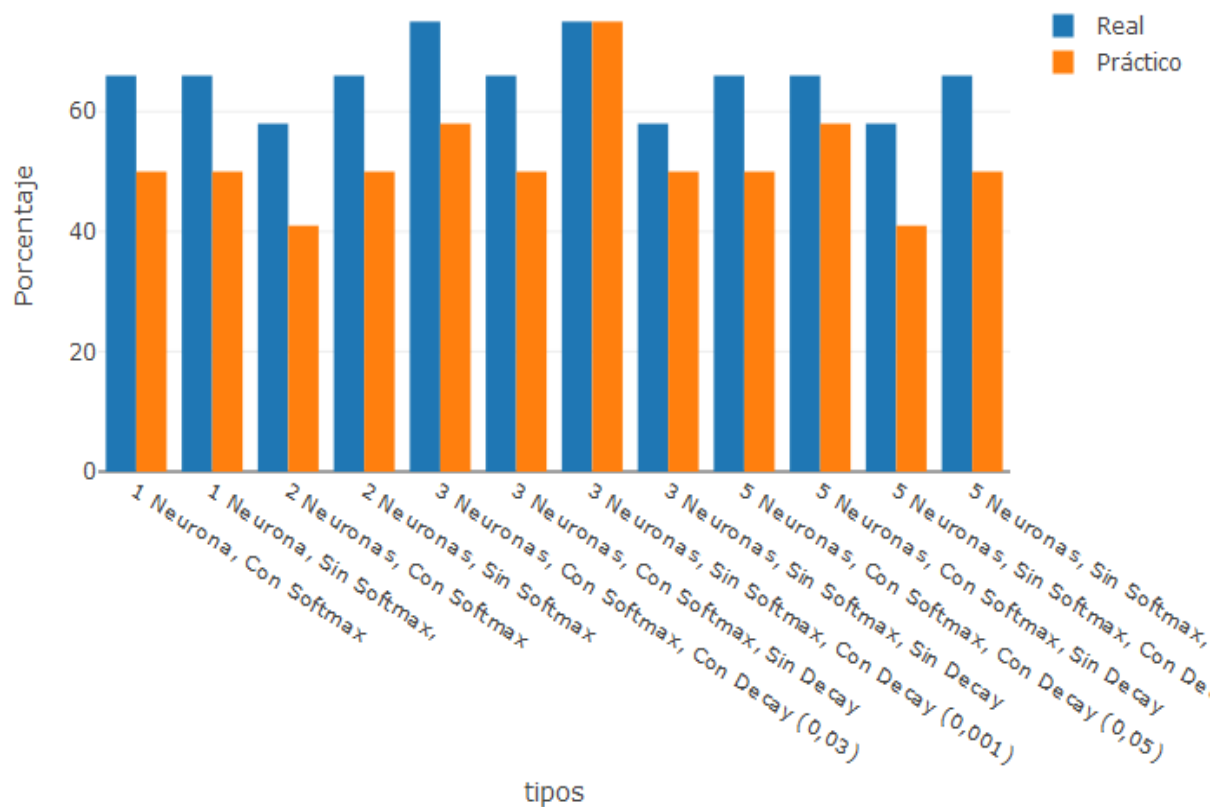
Ahora mostraré la tabla con los resultados finales de las pruebas con los perceptrones multicapa:

Tabla de aciertos reales (resultados finales) vs test práctico (resultados anteriores):

	Test Real (% Acierto)	Test Práctico (% Acierto)
1 Neurona, Sin Softmax	66%	50%
1 Neurona, Con Softmax	66%	50%
2 Neuronas, Sin Softmax	66%	50%
2 Neuronas, Con Softmax	58%	41%
3 Neuronas, Sin Softmax, Sin Decay	58%	50%
3 Neuronas, Con Softmax, Sin Decay	66%	50%
3 Neuronas, Sin Softmax, Con Decay (0,001)	75%	75%
3 Neuronas, Con Softmax, Con Decay (0,03)	75%	58%
5 Neuronas, Sin Softmax, Sin Decay	66%	50%
5 Neuronas, Con Softmax, Sin Decay	66%	58%
5 Neuronas, Sin Softmax, Con Decay (0,1)	58%	41%
5 Neuronas, Con Softmax, Con Decay (0,05)	66%	50%

De esta manera, obtengo 2 resultados con un 75% de acierto. Pero recordando lo comentado respecto al softmax, debo de desechar este resultado por corrección teórica, ya que está presuponiendo que un individuo sólo puede pertenecer a una clase. **Así, el resultado óptimo con el que me quedaré será el obtenido con la red neuronal multicapa de 3 neuronas en la capa intermedia, sin softmax y con un pequeño decay de 0,001.**

Esto lo podemos ver de una forma más clara en el gráfico que comenté en el punto 5, y que muestro a continuación:



Ahora expondré los datos obtenidos con **KNN**:

- K = 8

Con K = 8, lo cual es la raíz cuadrada más cercana al número de datos que tengo, obtengo la siguiente predicción:

Como podemos ver, tenemos un acierto del 0,083 + 0,25, al que hay que añadir los del grupo 2 (que también pertenecen al 1), por lo que un 0,25 más.

Total = 58% de acierto.

etiquetasTest	prediccion.knn.8		Row Total
	1	2	
1	1	3	4
	0.250	0.750	0.333
	0.250	0.375	
	0.083	0.250	
2	1	3	4
	0.250	0.750	0.333
	0.250	0.375	
	0.083	0.250	
3	0	2	2
	0.000	1.000	0.167
	0.000	0.250	
	0.000	0.167	
4	2	0	2
	1.000	0.000	0.167
	0.500	0.000	
	0.167	0.000	
column Total	4	8	12
	0.333	0.667	

- K = 6

Con K = 6, obtenemos una suma de aciertos de: 0,25 + 0,25 = 50% de acierto.

Como explicamos anteriormente en teoría, es posible que se haya cometido un poco de overfitting al hacer esta reducción de “neighbours”, ya que el resultado final de test ha disminuido.

- K = 10

Con K = 10, obtenemos la siguiente suma: $0,167 + 0,167 + 0,167 = 50\%$

Como podemos ver, el resultado ha bajado también, por lo que, siguiendo la teoría explicada en el punto 4, es posible que el modelo sea demasiado general y por lo tanto no esté clasificando de una forma óptima.

Por lo tanto, confirmamos lo que dice la teoría: La raíz cuadrada del total de los elementos que tengo es la mejor K que puedo poner en mi KNN, ya que hemos visto que una reducción de los individuos por grupo (y por lo tanto una mayor exactitud a la hora de agrupar) ha hecho perder generalización, y empezar a entrar en overfitting, y un aumento de los individuos por grupo ha hecho generalizar también peor debido a que ha sido demasiado general.

7. Conclusiones:

La primera conclusión que se puede sacar de este trabajo es que, la predicción de elementos relacionados con la mente humana y con sus malos funcionamientos es extremadamente compleja, ya que hay que tener numerosos parámetros en cuenta, y muchos casos tienen muchas características (y, por lo tanto, variables) en común.

La segunda conclusión que saco es que no hace falta una red neuronal extremadamente compleja para resolver este caso, ya que estamos viendo que con 3 neuronas y un ligero toque de decay podemos obtener resultados bastante aceptables con un 58% de encuadre exacto y un 75% de acierto en encuadre teórico, lo cual son resultados realmente buenos.

Junto con esto, me gustaría destacar un detalle que ha podido pasar por alto. Cuando la red ha obtenido unos coeficientes más bajos de acierto, ha tendido a clasificar todo el test en 2, o, como mucho, 3 grupos. Esto se ha dado cuando tenía demasiadas neuronas en la capa intermedia y no he usado decay, o cuando tenía muy pocas. En cambio, cuando he usado decay para simplificar cuando tenía demasiadas neuronas, o cuando he ajustado mejor el número de neuronas, la red ha tendido a tener en cuenta los 4 grupos a la hora de clasificar. Es decir, con esto tengo otra confirmación de que la red tendía a una sobresimplificación del problema, o a un overfitting que hacía que los grupos más minoritarios (Depresión y Personalidad) fueran sobreajustados y, con ello, probabilísticamente casi excluidos del test.

La tercera conclusión que se puede obtener de este trabajo, es que hay demasiadas pocas observaciones para afinar. Muchos porcentajes y resultados se repiten, y con un

test de solo 12 personas podemos obtener pocas diferencias de error, y por lo tanto podemos ver peor el encuadre que hace la red. Además, las redes neuronales se suelen entrenar con cientos, e incluso miles de datos, y el entrenamiento de una con 55 pacientes es una cantidad escasa, que puede hacer que haga un peor agrupamiento de cara a la predicción. Es verdad que este es un problema fácilmente solucionable mediante la recogida de mayor cantidad de datos, pero que no se ha solucionado por falta de tiempo (ya que los datos se han recogido totalmente a mano, para ganar totalmente en realismo).

Como cuarta conclusión, sobre KNN, puedo decir que ha conseguido acierto de test del 58%, lo cual es bastante, pero no ha llegado al nivel del perceptrón multicapa, que se quedó en el 75%. Por lo tanto, puedo decir que no ha generalizado lo suficientemente bien, quizás por los pocos datos que tengo, o quizás por la naturaleza del algoritmo.

Como quinta y última conclusión, me parece que este problema se puede probar a abordar con otro tipo de técnicas (como las redes RBF o las bayesianas), o incluso con Deep Learning, pero habiendo obtenido un 75% de acierto en este tipo de red con tan pocos datos y un problema tan complejo creo que es un buen resultado, por lo que es una confirmación de que he elegido de una forma correcta las variables que tenía que usar, así como que el centrado y escalado de los datos también ha sido un acierto.

Como conclusión hacia lo aprendido en la asignatura, me gustaría remarcar que las soluciones basadas en inteligencia artificial nos rodean en el día a día, y que actúan aunque muchas veces no las veamos. Hay numerosas soluciones ya solo en el campo de la predicción y clasificación, y por supuesto tener todas las opciones en cuenta, además de los detalles de cada una (como los parámetros), es vital para obtener resultados óptimos.

Al igual, no todo se centra en los algoritmos de predicción y clasificación, redes neuronales y clustering, sino que también las técnicas usadas en Big Data, como el tratamiento de datos y su limpieza, como técnicas matemáticas para la simplificación del DataSet, así como técnicas de visualización, se antojan esenciales para la resolución de los problemas reales de Inteligencia Artificial.

8. Bibliografía:

- 1) [Online] https://metacademy.org/graphs/concepts/weight_decay_neural_networks
- 2) [Online] <https://www.surveysystem.com/correlation.htm>
- 3) [Online] <https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax>
- 4) [Online] <https://www.quora.com/What-is-the-difference-between-read-csv-and-read-csv2-in-R-programming>
- 5) [Online] <https://topepo.github.io/caret/pre-processing.html>
- 6) [Online] <https://www.quora.com/Whats-the-best-way-to-visualize-high-dimensional-data>
- 7) [Online, PDF] <http://bibing.us.es/proyectos/abreproy/12166/fichero/Volumen+1+-+Memoria+descriptiva+del+proyecto%252F3+-+Perceptron+multicapa.pdf>
- 8) [Online] <http://radacad.com/prediction-via-knn-k-nearest-neighbours-concepts-part-2>
- 9) [Online] <https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7>
- 10) [Online] <https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e>
- 11) [Online] <https://sinfallas.wordpress.com/2017/11/14/el-perceptron-y-perceptron-multicapa-que-es-y-con-que-se-come/>
- 12) [Libro] Inteligencia Emocional - Daniel Goleman
- 13) [Libro] Sentirse Bien – Robin Burns. Ed Paidós, 1994.
- 14) [Libro] DSM IV. Manual Internacional de Diagnóstico.
- 15) [Ponencia] Inteligencia Emocional, Mediación y Solución de Conflictos – Mar González
- 16) Beck, A. Freeman, A. Terapia Cognitiva en los trastornos de la personalidad. Ed Paidós, 1995.