Análisis de técnicas Machine Learning para la estimación de medidas corporales

Javier M. Sánchez

Resumen—Actualmente, la mayoría de empresas de moda proporcionan una tienda virtual para realizar compras online. Según distintos estudios la tasa de retorno de ropa a causa de no acertar en la talla del producto elegido es muy elevada. Es por eso que en este documento nos centramos en mejorar la experiencia de usuario para la compra de camisetas, implementando un sistema para predecir la talla de un individuo varón en función de su edad, peso y altura. Esta implementación se realizará utilizando técnicas de Machine Learning y Data Mining como son Feature Selection para el análisis de las características más representativas del cuerpo humano o Deep Learning para la predicción de las tallas. Después de todo el análisis realizado, hemos llegado a la conclusión de que la edad, el peso y la altura son características que la mayoría de empresas usan, no gracias a su alto valor representativo dentro del cuerpo humano, sino a causa de su simplicidad. En cuanto al sistema implementado, podemos destacar que proporciona resultados gratamente buenos como para evitar el gasto que tienen las empresas en la devolución de ropa, casi en su totalidad.

Palabras clave—Mineria de datos, Clusterización, Red Neuronal Artificial, Deep Learning, K-means,Regresión Lineal, Máquina de Vectores de Soporte (MVS)

Abstract—Nowadays, a great number of companies provide a virtual shop to buy products online. Based on different analysis, the return ratio of clothes because of mistakes in chosing the right size is so high. As a consecuence of that, in this paper we will focus on improving the user experience for buying t-shirts by implementing a system to predict male t-shirt size through his age, weight and height. This implementation will use some Machine Learning and Data mining techniques such as Feature Selection, to analyze the most representative human measures and Neural Network to predict t-shirt sizes. After doing the whole analysis, we get to the conclusión that age, weight and height are features that most of the companies are using to predict, not because of its representative value in the human body, but because of its simplicity. About the implemented system, we can emphasize that the method is suitable for preventing almost all expenses that companies have because of clothes reimbursement.

Index Terms—Data Mining, Clustering, Artificial Neural Network, Deep Learning, K-means, Linear Regression, Support Vector Machines (SVM)

1 Introducción

COMPRAR online se ha vuelto una práctica muy común para la mayoría de internautas que prefieren realizar su pedido desde casa y pagar el producto mediante un terminal de punto de venta (TPV) online. Es por eso que, hoy en día, la mayoría de empresas ofrecen una solución a estos clientes implementando webs de compra online para sus productos, en concreto, para la ropa.

Analizando el mercado en el campo de las ventas online hemos observado que la tasa de retorno de la ropa es aproximadamente del 30% de media, casi el doble de la tasa de retorno de otros artículos, generando un 14% de pérdidas a la compañía. A causa de este factor, el 62% de los internautas no ha vuelto a hacer una compra online en el último mes[1]. Si además tenemos en cuenta los factores que dan valor a un eCommerce, podemos observar que la facilidad en el proceso de devolución y de reclamaciones están situados en la cuarta posición en el ranking de factores valorables de un eCommerce, con 6,9 puntos[2]. Otro de los puntos interesantes a tener en cuenta es que la tasa de internautas que repiten sus compras gracias a la buena política de devoluciones y cambios es del 34% y el porcentaje de la gente que abandona el mercado onlne a causa de este mismo factor es del 21%[3]. Tal y como podemos observar a menudo, las compañías carecen de un servicio de devoluciones online apropiado para que el usuario confie en la compra mediante Internet.

Para solventar este problema la mayoría de empresas han optado por añadir una guía de tallas o por la opción del calculo de tallas. Aún así, todas esas medidas son complicadas de calcular y no nos aseguran con total certeza que la talla sea la correcta.

- *E-mail de contacto: javi.ms10@gmail.com*
- *Mención realizada: Computación.*
- *Treball tutoritzat per: Jordi Gonzàlez (departament)*
- *Curs 2013/14*

Algunas empresas como My Shape, Size Me, Fits Me o Verisize, ya han análizado este problema y han realizado varias propuestas para solucionarlo. Todas y cada una de ellas pretenden ayudar al comprador a realizar una compra más cómoda y eficaz aunque niguna de ellas lo ha conseguido con gran fiabilidad. My shape[4] se centra en las mujeres y recomienda al cliente en función de la forma del cuerpo y las medidas personales que los usuarios les proporcionan. Size Me[5] pide al cliente la marca, el tamaño y la medición de su ropa favorita para analizarla y posteriormente recomendarsela. Fits Me[6] opta por diseñar un maniquí web que se puede ajustar con las medidas que el usuario introduce en la página. Y, finalmente, Verisize[7] una start-up nacida en el CVC que pretende realizar un sistema similar al explicado en este artículo , con la diferencia que ellos utilizan un cuarto parámetro, la forma o perfil del usuario, además de los tres que más adelante se comentarán.

La solución que nosotros proponemos es mucho más simple y automatizada. Esta consiste en analizar las medidas corporales para implementar la base de un sistema que, a partir de parámetros que el usuario considere fáciles de recordar como la edad, el peso y la altura, predecir la talla de camiseta correspondiente. El segundo servicio que ofreceríamos sería el de la conversión de piezas de ropa de diferentes empresas textiles a tallas. Es decir, las empresas de ropa nos proporcionarían las medidas que tienen establecidas en sus camisetas y nuestro sistema las procesaría todas conjuntamente generando un tallaje común a todas las marcas.

Es importante destacar que en este trabajo nos centraremos en el análisis y diseñaremos una arquitectura que nos permita realizar pruebas, pero no se implementará en ninguna tienda online.

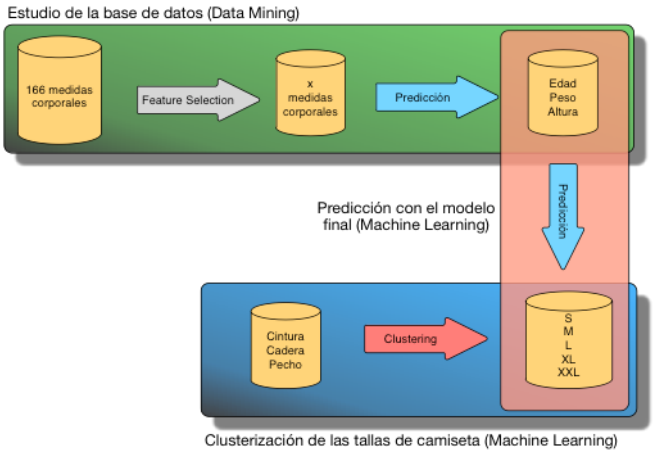


Fig.1 – Arquitectura del sistema

Dicho sistema (*ver Fig.1*) requiere la aplicación de algorítmos y técnicas relacionadas con campos de la inteligencia artifical enfocados al análisis e ingeniería de datos como Machine Learning o Data Mining. Así pues, de los dos servicios mencionados anteriormente, el primero utilizará técnicas de Machine Learning para hacer la predicción de las tallas y el segundo, para el análisis de los datos proporcionados por diferentes marcas. Utilizaremos Data Mining únicamente para el análisis de las medidas biométricas.

Tal y como podemos observar en la Figura 1, el proyecto está separado en tres fases relacionadas entre sí. La primera, que englobaremos dentro de Data Mining, contiene el proceso de análisis de todas las características iniciales para extraer aquellas de mayor relevancia, primero mediante "Feature Selection" aplicando varios filtros y segundo mediante el análisis de cuan buenas son las características restantes en cuanto a la predicción de todas y cada una de las demás. Para este proceso usaremos un Regresor Lineal[8] o un SVR (Suppor Vector Regressor)[9] en función del error mínimo obtenido por cada una de las técnicas.

A continuación, agruparemos en Machine Learning las dos siguientes fases. La primera consistirá en clasificar cada uno de los individuos dentro de una talla (S, M, L, XL, XXL) en función de la cintura, cadera y pecho. Para ello, utilizaremos KMeans[10], un algoritmo no supervisado basado en las distancias entre las muestras y el centro del cluster. La segunda fase consistirá en entrenar una red neuronal basada en Deep Learning[11] que nos permita generar un modelo robusto que prediga las tallas de un individio mediante las características extraidas en el proceso de data mining y el tallaje generado para cada uno de los individuos en la primera fase de Machine Learning.

Para finalizar el documento presentaremos los resultados obtenidos detallando las características usadas para la predicción, la clusterización que nuestro KMeans ha generado y los parámetros óptimos para implementar una red neuronal multicapa que genere el valor de la talla asociada a cada individuo.

2 ANÁLISIS DE MEDIDAS CORPORALES

Para realizar el análisis de medidas corporales se ha optado por una base de datos que proporcione un gran número de características del cuerpo humano en vez de una gran cantidad de muestras. Gracias a este dataset podemos obtener la altura de la pantorrilla o el arco del deltoide, en vez de la típica longitud del brazo o el ancho del pecho.

Además, nos hemos centrado únicamente en la antropometría de individuos varones. Esta elección viene dada debido a la hipótesis de que la predicción para hombres es más sencilla porque en el cuerpo de una mujer o un niño intervienen más parámetros para la determinación de una talla.

Por eso, nuestra base de datos *(http://mreed.umtri.umich.edu/mreed/downloads.html)* nos proporciona 1773 muestras de hombres y cada una con 166 características, algunas sin mucha importancia y otras

muy relevantes, tal y como se verá más adelante.

Nuestro dataset se dividió, inicialmente, en tres partes: training, valicación y test. Para realizar este proceso escogimos el fichero .csv inicial, alteramos el orden de las muestras dadas y se eligió un 60%, 20% y 20% de cada parte, respectivamente.

El conjunto de training nos permitirá enseñar a nuestro modelo, mientras que el conjunto de validación nos ayudará a retocarlo en función de los resultados obtenidos, y de esta manera predecir con el mínimo error en el test.

2.1 Metodología de evaluación

Al tratarse de un análisis de distintas técnicas usadas en el campo del Machine Learning es difícil marcar una meta cuantitativa a la cual llegar para considerar que el trabajo realizado ha sido satisfactorio.

En este proyecto se ha decidido analizar los tres elementos que componen nuestro sistema de predicción de tallas y observar si en cada uno de ellos se ha cumplido el objetivo marcado.

Para el primer bloque sobre la generación del input para nuestro problema y análisis de la base de datos hemos tenido en cuenta cuan buenas son las características elegidas para representar las medidas corporales, en función del RMSE.

En el segundo apartado formado por la clusterización de las tallas analizaremos las ventajas e inconvenientes de la técnica escogida.

Finalmente, compararemos el error de predicción que tenemos en la red neuronal generada en función del estudio de los parámetros realizados. Además, compararemos el resultado final obtenido con el análisis de mercado realizado al comienzo del documento y observaremos los beneficios obtenidos

Los modelos generados para la predicción de característicticas, tanto con el Regresor Lineal como con el SVR, se basaran el error llamado "Root Mean Square Error" (1).

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}{n}}. \qquad (1)$$

Donde $\hat{y}_n$ es un vector del mismo tamaño que y_n y n es el número total de muestras. Básicamente, lo que está calculando es el sumatorio de la diferencia entre la predicción que nuestro sistema ha dado y el valor real.

Cabe destacar que éste error se utilizará únicamente en el primer proceso de nuestro análisis, es decir, en el análisis de las características.

2.2 Selección de características biométricas

Una vez tenemos nuestro dataset dividido en training, validación y test, hemos fijado el error mediante el cual determinaremos la eficiencia de nuestros modelos, procederemos a analizar las características para determinar cuales nos aportan más o menos información a la hora de generar un modelo que determine las características más representativas para el cuerpo humano y fáciles de recordar.

Para ello, es necesario normalizar los datos, ya que cada una de las características de nuestra base de datos están definidas entre rangos de valores muy distintos. Podemos observar, por ejemplo, que el ancho del talón, medido en mílimetros, se mueve en un rango de entre 55 y 90, mientras que el ancho de la cintura entre 282 y 416.

Para mejorar la eficiencia de nuestros modelos, optaremos por normalizar (2) los datos entre 0 y 1 mediante un reescalado.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (2)$$

Teniendo los valores de todas las características en un mismo rango, permitiremos al modelo realizar cálculos más fácilmente, y sobre todo, que las características con rangos de valores más altos no dominen sobre los bajos.

Una vez normalizados los datos, calcularemos el histograma de cada una de las columnas (166) para observar que distribución siguen. Únicamente nos quedaremos con aquellas que sigan una distribución normal. Ésta elección aportará al modelo un mayor rendimiento debido a que la distribución normal representa el comportamiento de las leyes de las ciencias naturales.

Una vez hemos reducido el total en características a 130, eliminamos también información innecesaria como el identificador de la muestra o características con valores nulos, ya que no nos aporta ningún valor significativo.

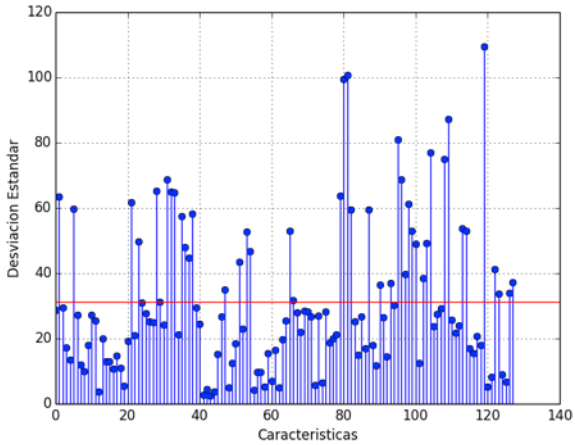


Fig. 2 – Filtraje por media de las desviaciones estándar

El siguiente paso es calcular la desviación estándar de las

características restantes. Mediante este filtro obtendremos las características más compactas y menos diversas. Para filtrar las características calcularemos la media de todas las desviaciones estándar de cada columna y la utilizaremos como umbral (*ver Fig.2*) para determinar aquellas que son válidas y aquellas que no.

En este punto nos encontramos con 84 características que quedan por analizar, ya que habremos eliminado aquellas superiores a la media, debido a que son muy dispersas, y mantendremos las que no superen este umbral (más compactas). Aplicando este filtro se descubrió que la edad ya no es una de las características candidatas a la predicción optima del resto de medidas corporales, como más adelante corroboraremos.

Seguidamente, intentaremos encontrar las características más representativas del conjunto final. Para ello aplicaremos un algoritmo de aprendizaje supervisado para predecicciones llamado Regresión Lineal. Éste algoritmo nos permite encontrar una función (3) que represente de la mejor forma posible el conjunto de datos proporcionados.

$$y = x_0 + \theta_1 x_1 + \cdots + \theta_n x_n \qquad (3)$$

La variable y nos indica la salida a nuestra entrada, x_0 es el bias, x_n es cada una de las características de la base de datos y θ_n es el valor que el regresor calcula en función del conjunto de entrenamiento proporcionado.

Aplicaremos, entonces, el regresor lineal a cada una de las características, cogiendo como entrada cada una de ellas y como salida todo el resto. Para cada una de las salidas comprobaremos el error obtenido, RMSE, y realizaremos una media para tener una idea de la calidad de esa característica para predecir todas las demás.

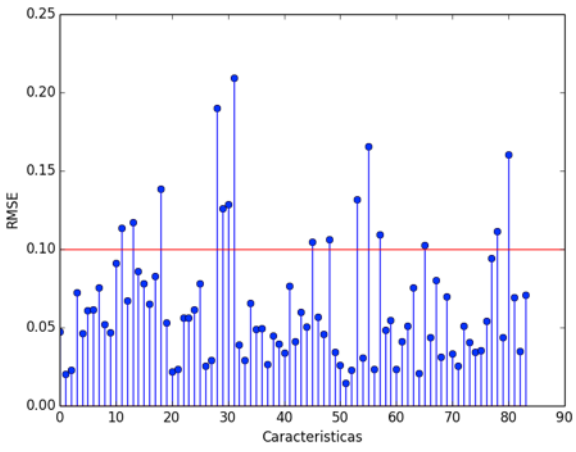


Fig. 3 – Filtraje por predicción de cada una de las características

A continuación, eliminaremos aquellas características en las que el error sea mayor a 0.1, es decir, un error mayor al 10%.(*ver Fig.3*)

Finalmente, obtendremos 69 características relevantes del conjunto de 166. En estos momentos empezamos a analizar cada una de las medidas del cuerpo humano y observamos que entre ellas no se encuentra ni la altura ni el peso. En cambio, las tres medidas con un menor error de predicción son: la distancia desde la planta del pie a la rodilla cuando el individuo está sentado (error del 0.01), junto con la distancia desdel hombro hasta la superficie apoyada en la silla medida en línea recta (error del 0.02) y la distancia entre el hombro y el codo formando el brazo un ángulo de 90 grados (error del 0.02).

2.3 Priorizando memorización a precisión

Aún así, las medidas más optimas encontradas por el regresor lineal tienen un inconveniente, y es que son difíciles de recordar. Es por eso que hemos optado por realizar el estudio con las tres características mencionadas al inicio del documento: la edad, el peso y la altura. A continuación nos aseguraremos de que éstas medidas tienen un buen comportamiento a la hora de describir al resto de ellas. Para ello, intentaremos predecir cada una de las medidas restantes (163) de todo el conjunto de test introduciendo la edad, el peso y la altura como parámetros de entrada, independientemente uno de los otros dos.

En este proceso utilizaremos dos algoritmos, el regresor Lineal ya mencionado anteriormente, y el SVR (Suppot Vector Regressor). El SVR es un algoritmo de aprendizaje supervisado que realiza un mapeo no lineal del espacio de características sobre un kernel para, posteriormente, encontrar un modelo lineal que se ajuste a los datos proporcionados. La elección de este kernel y la configuración de estos parámetros es lo que nos permitirá generar un modelo más o menos eficiente pero en este artículo, solamente trabajaremos con los modelos RBF (Radial Basis Function), Polinomial y Lineal.

Para valorar su efectividad, compararemos los resultados entre un Regresor Lineal y un SVR (Support Vector Regressor), como se ha realizado en la Tabla 1.

	Edad	**Altura**	**Peso**	**Media**
Regresion Lineal	0.235	0.047	0.046	0.109
SVR(lineal)	5.873	4.204	6.737	5.538
SVR (poly)	6.195	4.222	6.747	5.721
SVR (TBF)	6.887	5.536	8.002	6.8

Tabla 1- Error en la predicción de la Edad, Peso y Altura

Como podemos observar, la Regresión Lineal nos aporta el mejor rendimiento, prediciendo la edad con un error del 0.23, la altura con un error del 0.046 y el peso con un error del 0.046.

3 AGRUPACIÓN EN TALLAS

Cada empresa textil realiza un tallaje diferente de sus productos, por lo que no hay un estandar fijado para esta conversión. En esta sección, proponemos una conversión automática de prendas a un tallaje óptima, que nos permitirá relacionarlo posteriormente con las medidas corporales previamente consideradas.

Por lo tanto, la segunda fase del proyecto consistirá en agrupar cada uno de los individuos de nuestra base de datos en cada una de las cinco tallas con las que trabajaremos. Este proceso se puede realizar usando varios parámetros de entrada, y es lo que en un futuro se quiere ofrecer como servicio. En nuestro caso optaremos por utilizar la longitud de la cintura, de la cadera y del pecho, ya que son las características que la mayoría de guías de tallaje utilizan en sus tiendas online. Excepto la longitud de la cadera, el resto de medidas se encuentra explícitamente en nuestra base de datos. Para obtener la longiud de la cadera hemos tenido que multiplicar el diámetro de la cadera, que sí que se encontraba en nuestro dataset, por π.

3.1 K-means

K-means es un algoritmo para aprendizaje no supervisado basado en el cálculo de tantos centroides como clusters se quieran obtener, y la distancia euclidiana de las muestras a estos puntos. Éste algoritmo nos ofrece la posibilidad de clasificar todas y cada una de las muestras listadas en nuestra base de datos para obtener el tallaje de cada individuo.

Como entradas al algoritmo utilizaremos: la longitud del pecho, cadera y cintura, como ya hemos mencionado anteriormente. De esta forma generamos un espacio 3D (*ver Fig. 3*) en el que cada muestra quedará etiquetada por su tallaje.

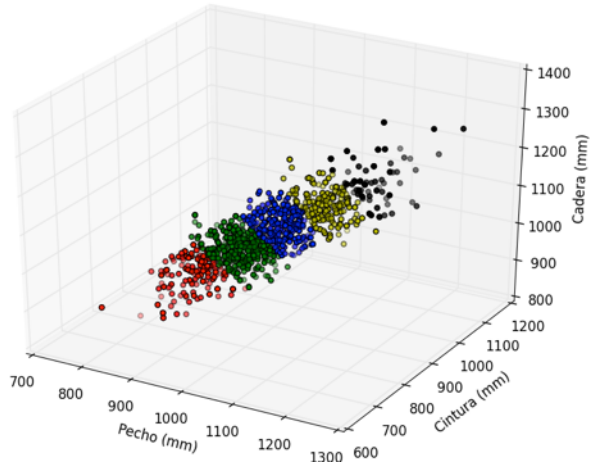


Fig. 3 – K-means de las tallas S, M, L, XL y XXL

En la figura se ven las muestras correspondiente a las tallas S (en rojo), M (en verde), L (en azul), XL (en amarillo) y XXL (en negro). Este mapeo se ha realizado partiendo del hecho de que las tallas pequeñas corresponden a valores pequeños de la cadera, la cintura y el pecho, justamente al contrario que con las tallas grandes.

Observando detenidamente la gráfica podemos observar que hay alguna muestra que podría corresponder a un sexto cluster (XS o XXXL) como es el caso de la muestra roja (por abajo) o las muestras negras (por encima).

Para comprobar la efectividad del KMeans se ha optado por analizar (*ver Tabla 2*) el número de muestras clasificadas, para así tener una visión global de las tallas que se encuentran en nuestro dataset, y la distancia media de cada punto de cada cluster respecto a su correspondiente centroide. Como podemos observar la distancia media de cada cluster es relativamente pequeña (no llega al 1%) comparado con el rango de valores de la longitud de la cadera, cintura y pecho, que llegan a valores de 1000 mm.

Tras el agrupamiento realizado dividiremos nuestro dataset general en training y test en función de la cantidad de tallas. El conjunto de training estrá formado por el 60% de cada una de las tallas para clasificar, mientras que nuestro test estará compuesto por el restante 40% de las muestras de cada talla.

Talla	# Muestras	Media distancia al centroide
S	261	47.18
M	563	48.39
L	444	49.90
XL	389	64.19
XXL	116	46.15

Tabla 2 – Clusterización en K-means

Se ha seguido este proceso ya que nos interesa tener un mínimo número de muestras de cada talla para que el aprendizaje sea eficiente. De lo contrario, el modelo generado por la red neuronal (punto 4) estaría muy bien entrenado para ciertas tallas, y para otras siempre daría errores en la predicción. De esta forma, conseguiremos que el entrenamiento de cada una de las tallas sea proporcional a la cantidad de muestras que existen de cada cluster.

4 DEEP LEARNING

Finalmente, para la predicción de las tallas de los individuos, utilizaremos Deep Learning, un concepto basado en redes neuronales compuestas por más de una capa intermedia u oculta. Nos permite generar un modelo no lineal y estadístico adaptado para soportar una gran cantidad de dimensiones como entrada, al contrario que ,por ejemplo, SVM o un clasificador multi-variante mediante una función logística.

Para nuestro problema configuraremos una red neuronal formada por 3 neuronas en la capa de entrada (edad, altura y peso) y 5 neuronas en la salida(*ver Fig.4*). La en-

trada estará compuesta por valores normalizados de cada uno de los individuos de training y la salida será un vecto binario que indicará qué talla es la correspondiente para la muestra de entrada. El número de las capas intermedias y la cantidad de neuronas que contendrá cada una se analizará en la primera sección de este apartado. A continuación, se explicará el algoritmo de actualización de pesos utilizado y por qué no se ha utilizado otro. Finalmente, tendremos en cuenta los parámetros más importantes que ayudan a que una red neuronal sea estable y eficaz.

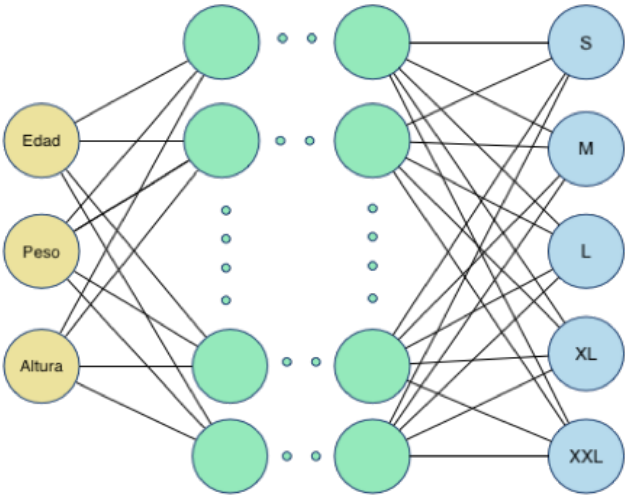


Fig. 4 – Ejemplo de arquitectura para una red neuronal multicapa

A partir de este estudio intentaremos obtener la configuración que nos aporte un menor error de predicción para nuestro conjunto de entrenamiento y de test, con la intención de que en un futuro se obtenga la red neuronal generada y se aplique a cualquier tienda virtual. Sin embargo, este proceso es realmente complicado ya que todos y cada uno de los parámetros pueden ser dependientes entre sí.

4.1 Arquitectura

En el estudio de la arquitectura optima se empezó realizando pruebas mediante el aumento del número de neuronas en una sola capa. Esto nos daría una idea del rango de neuronas que deberíamos indicar en cada una de las capas. El siguiente paso sería, aplicando ese número de neuronas a cada capa, ir incrementando el número de estas y observar su comportamiento.

Tal y como podemos observar en la Figura 5, cuando la red neuronal de una sola capa se compone por más de 200 neuronas, el error tanto de entrenamiento como el de test aumenta de forma considerable indicando la presencia de overfitting. A causa de esto, las pruebas posteriores se realizarán probando neuronas en un rango de 1 a 100. En la gráfica anterior podemos observar el comportamiento de training para hacernos una idea de cómo se comporta en comparación con el de test. En las futuras gráficas únicamente se mostrará el error de test, ya que el de training se ha conseguido reducir al 0.0%.

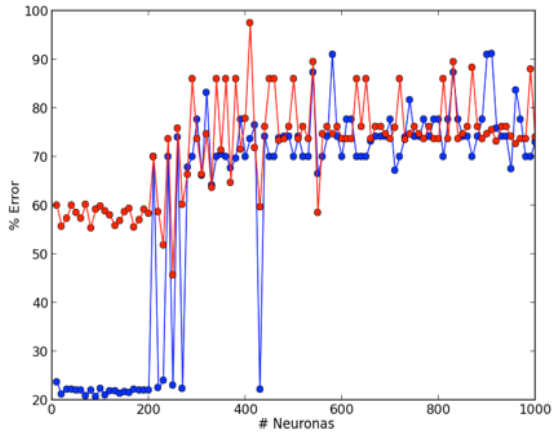


Fig.5 – Error en el conjunto de training (azul) y test (rojo) en function del número de neuronas.

En cuanto al número de capas ocultas podríamos decir que si es elevado, la red neuronal empezará a generar un error increíblemente elevado debido a la cantidad de parámetros que debe tener en cuenta. Intentaremos ajustar estos parámetros los mejor posible, añadiendo un número de capas y neuronas lo suficientemente alto de manera que no se produzca un underfiting, pero lo suficientemente bajo para generalizar las muestras de test y no producir un overfitting.

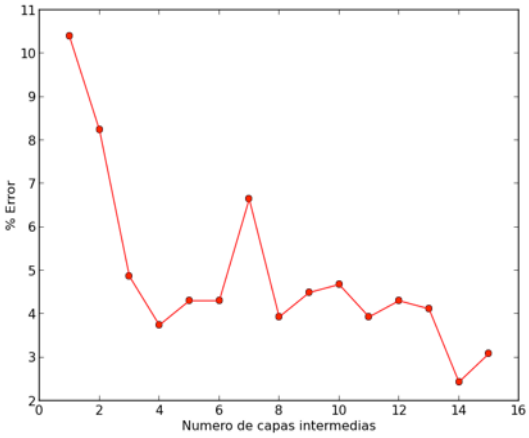


Fig.6 – Error en el conjunto de test para la variación del número de capas intermedias

En la Figura 6 podemos observar que el número de capas óptimo es de 14. Aún así, para el resto de pruebas utilizaremos 4 capas intermedias ya que las pruebas tardarán mucho menos y la variación del resultado de las pruebas es mínimo. Como dato interesante, cabe destacar que para redes neuronales de una y dos capas el error si que llega a ser el doble, posiblemente debido a un underfitting de la base de datos de test.

Así pues, para las pruebas sobre el análisis de los parámetros se ha fijado la red neuronal a 1 capa de entrada con 3 neuronas, 4 capas intermedias con 80 neuronas cada una y 1 capa de salida con 5 neuronas.

4.2 Backpropagation

Observando el conjunto de algoritmos implementados para la actualización de los pesos en las redes neuronales se optó por elegir Backpropagation [12] debido a su efectividad mediante un proceso relativamente simple.

Es una algoritmo de aprendizaje supervisado y está compuesto por dos fases. La primera consiste en generar una o varias salidas, en función del tipo de red neuronal, mediante la inicialización de pesos, ya sea aleatoria o no. Una vez generada la salida obtenemos el error total restando la salida generada menos el valor de la salida real. A partir de ahí, se generarán los errores parciales a cada una de las capas y neuronas, mediante los cuales obtendremos las derivadas parciales necesarias para aplicar un algoritmo, como por ejemplo el descenso de gradiente, que nos permita minimizar la función de coste.

Como función de activación de las neuronas utilizaremos la Sigmoidea[13] ya que es la más común y la más usada actualmente.

4.3 Estimación de parámetros

Los parámetros[14] que principalmente afectan al aprendizaje de la red neuronal son: el coeficiente de aprendizaje, el momentum y el regularizador. Partiendo de la base de los primeros resultados en la arquitectura de la red neuronal, generaremos una de las mismas características: 1 capa de entrada con 3 neuronas, 4 intermedias con 80 neuronas cada una y 1 capa de salida con 5 neuronas.

El análisis consistirá en variar cada uno de los parámetros des de 0 hasta 1, aumentando el valor en unidades de 0.01, excepto en el caso del regularizador que la variación de los valores será de 0.001, entre 0 y 0.1 ya que los valores superiores penalizaban el rendimiento de la red neuronal.

El coeficiente de aprendizaje es el parámetro que nos ayuda a minimizar nuestra función de coste con mayor o menor rapidez y con más o menos precisión. Un valor

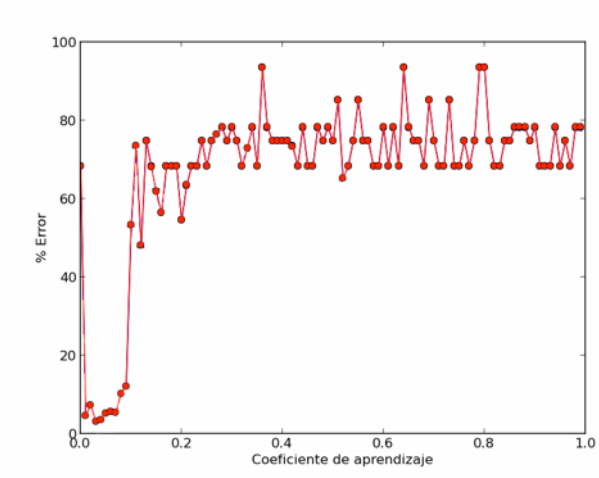


Fig.7 – Error en el rango de valores del coeficiente de aprendizaje

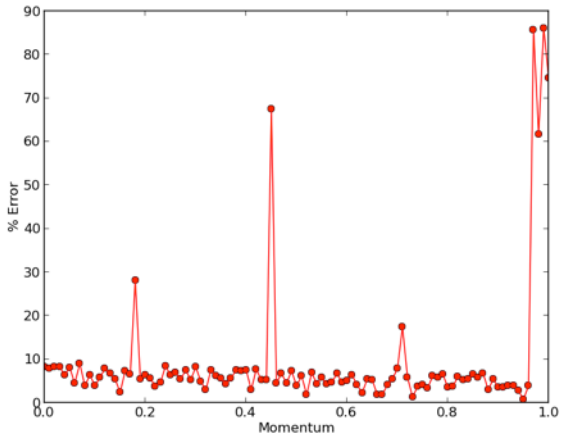


Fig.8 – Error en el rango de valores del momentum

muy alto generará un modelo en poco tiempo pero poco preciso, mientras que un valor demasiado bajo generará un modelo muy preciso a costa de un tiempo muy elevado.

Como podemos ver en la gráfica generada (*ver Fig.7*) los valores del coeficiente de aprendizaje bajos corresponden a errores relativamente bajos, a excepción del 0.0, valor con el cuál jamás aprende nuestro modelo. En cambio, a medida que vamos aumentando este parámetro, en concreto a partir del 0.1, observamos que el error de test se dispara. Esto se debe a que en el algoritmo de minimización de la función de coste, si el valor del coeficiente de aprendizaje es excesivamente alto, la variación del valor de la función de coste será muy elevado y por eso al algoritmo le será difícil converger en un mínimo global. Tambien puede pasar que si el valor es excesivamente bajo, nos quedemos atrapados en un mínimo local, pero como podemos observar en esta gráfica, para valores bajos la red neuronal obtiene valores relativamente buenos. A cause de esto, podemos decir que nuestro espacio de características no contiene muchos mínimos locales que impidan llegar al mínimo local. Eso sí, si al coeficiente de aprendizaje le damos un valor excesivamente bajo es probable que la solución la encuentre en un tiempo muy elevado.

Por otro lado tenemos el momentum. Este parámetro nos ayuda a salir de un mínimo local en el caso de que el algoritmo de minimización de la función de coste se quede estancado. Además, nos permite converger a una velocidad más elevada ya que es un parámetro que multiplica al peso.

La gráfica (*ver Fig.8*) nos permite observar que este parámetro no acaba siendo determinante para el aprendizaje de la red neuronal, aun así hay ciertos valores que causan un cierto retroceso en el proceso de aprendizaje, en concreto, valores superiores a 0,9. Esto significa que ,o hay pocos mínimos locales en nuestro espacio de características, o que estos no son lo suficientemente profundos como para que el algoritmo se quede estancado y necesite

de un valor muy elevado para salir.

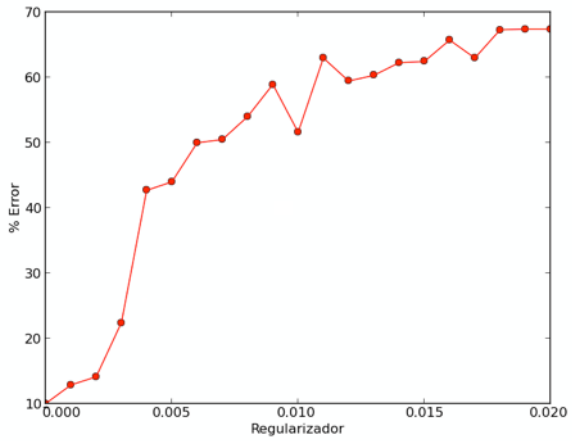


Fig.9 – Error en el rango de valores del regularizador

El último parámetro analizado es el regularizador. Este parámetro ayudar a reducir el overfitting causado por el modelo generado. En el análisis (*ver Fig.9*) realizado para este parámetro hemos observado que es preferible asignarle valores bajos entre 0 y 0.1, ya que a partir de ahí el error aumenta considerablemente. Si hubieramos considerado 14 capas neuronales, el valor del regularizador hubiera sido mas elevado debido al número tan alto de parámetros que hubiera tenido el modelo.

5 ANÁLISIS DE LOS RESULTADOS

Una vez realizado el análisis de las medidas corporales más destacables en el cuerpo de un varón, hemos detectado que las características con un menor error de predicción respecto a las demás y por lo tanto más representativas son: la distancia desde la planta del pie a la rodilla cuando el individuo está sentado, con un error del 0.01, la distancia desdel hombro hasta la superficie apoyada en la silla medida en línea recta, con un error del 0.02 y finalmente la distancia entre el hombro y el codo formando el brazo un ángulo de 90 grados, con un error aproximado al 0.02.

Sin embargo, las empresas hoy en día no las tienen en cuenta en ningúnas de sus tecnologías. Este análisis nos da a entender que hay que mantener un equilibrio entre una buena experiencia del usuario, facilitándole el uso de la herramienta incorporando las medidas de edad, peso y altura, y la optimización de los resultados. La mayoría de compañías estarian dispuestas a fallar en un 5% sus estimaciones, a cambio de que al usuario le sea amigable la utilización de su plataforma. Es por eso que nuestro análisis finalmente se ha decantado por estos parámetros básicos.

Hablando sobre la optimización de los resultados,y después del análisis realizado en la Tabla 1 podemos ver que la edad no es una característica representativa, almenos como el peso y la altura, que aun no siendo de las más relevantes se encuentran entre las 69 destacadas, debido a que el error de predicción, aún siendo bajo, cuatruplica el de las otras dos características. Así que es posible que en futuros experimentos este parámetro pueda no tenerse en cuenta y se añada alguno de los tres mencionados al inicio de la sección que proporcionaban un error de predicción muy bajo

En cuanto a la generación de la salida, podemos decir que el K-means implementado consigue separar las muestras de una forma eficiente, incluyendo en cada cluster una gran cantidad de muestras de cada talla y dejando ver claramente que hay algunas muestras que podrían clasificarse en tallas más pequeñas (XS) o más grandes (XXXL). Sí es cierto que la clasificación que se observa en la Figura 3 genera clusters con poca separación entre ellos, no dejando claro a qué talla pertenece más de una muestra. En cuanto a los tallajes extremos, no se ha optado por realizar dos clusters más con estas muestras ya que la cantidad de muestras de entrenamiento y de test no serían suficientes para crear un modelo robusto y fiable.

Para definir los parámetros que compondrán finalmente nuestra red neuronal hemos analizado todas y cada una de las gráficas y tablas realizadas y hemos escogido el parámetro que nos aportaba el menor error, tanto en el conjunto de entrenamiento como en el de test.

Después del estudio realizado, hemos optado por diseñar la red neuronal óptima para este problema con estos parámetros:

- 1 capa de entrada con 3 neuronas.
- 1 capa de salida con 5 neuronas.
- 4 capas intermedias de 80 neuronas.
- Coeficiente de aprendizaje del 0.03.
- Momentum del 0.73.
- Regularizador del 0.0.
- Backpropagation como algoritmo de actualización de pesos mediante la function sigmoidea de activación de neuronas.

Según los resultados (*ver Fig. 3*) , la red neuronal se mantiene estable y alrededor del 20% de error de entrenamiento y 60% de error en el test, cuando insertamos hasta 200 neuronas. En el momento que superamos este umbral, la red se desestabiliza y los errores, tanto de entrenamiento como de test crecen alarmantemente.

Gracias al análisis realizado para el número de capas intermedias, se ha decidio asignarle un valor de 4, con 80 neuronas por capa. El número de capas y de neuronas nos permite tener una red neuronal lo suficientemente robusta como para, no solo adaptarse al 100% al conjunto de train, sino para obtener un error de test bajo.

Como hemos analizado anteriormente, el valor del coeficiente de aprendizaje escogido es lo suficientemente elevado como para converger en un tiempo adecuado y lo

suficientemente bajo como para encontrar el mínimo global. En cuanto al momentum, debido a que los valores bajos del coeficiente de aprendizaje no nos han ocasionado errores elevados, podemos destacar que este parámetro no tiene gran relevancia y, como vemos en la Figura 6, la mayoría de los valores se adecuan a nuestro modelo. Finalmente, se ha podido observar que el regularizador es preferible mantenerlo con valor 0. La conclusión que podemos extraer de éste parámetro es que la red neuronal no necesita darle valor alguno debido a que es capaz de generalizar lo suficientemente bien como para no tener que reajustarse mediante este parámetro. También es posible que al tener un dataset con pocas muestras y el conjunto de test sea tan reducido, no sea necesario generalizar más ya que cubre todos los casos posibles. En el caso de que añadiéramos muchisimas más capas ocultas a nuestra red neuronal, es problable que tuviéramos que retocar este parámetro.

Gracias a esta configuración hemos conseguido obtener un error del 5% en el conjunto de test. Analizando el resultado desde un punto de vista de márqueting y del interés comercial, podemos deducir que la empresa se ahorraría un 95% de los gastos de devoluciones por el ratio de devoluciones (en le mejor de los casos ya que el sistema no es fiable al 100%) de los gastos en devoluciones de camisetas si utilitzaran nuestro sistema para predecir la talla que el usuario necesita. Como podemos observar , los beneficios son muy elevados comparados con el coste de su implementación y mantenimiento.

Para remarcar más las mejoras, a continuación se muestra la matriz de confusión[16] (*ver Fig.10*) que consta de cinco columnas y cinco filas, correspondientes a las cinco tallas (S, M, L, XL y XXL).

	S	M	L	XL	XXL
S	77	2	0	0	0
M	5	150	14	0	0
L	0	0	118	17	0
XL	0	7	0	113	0
XXL	0	0	0	0	31

Fig. 10 – Matriz de confusión de tallas

Uno de los puntos críticos para mejorar los resultados ha sido la normalización de los datos. Para hacernos una idea, introduciendo los parámetros anteriores en una red neuronal obteníamos un error del 20% en el entrenamiento y un 40% en el test teniendo los datos sin normalizar. Si anlizamos el resultado desde una perspectiva económica la empresa se ahorra un 60% del coste actual. Sin embargo, gracias a la normalización, la precisión actual es del 95%.

Aún habiendo obtenido unos resultados gratamente satisfactorios, hemos observado que en algunos experimentos la cantidad de muestras para el entrenamiento y test no eran suficientes. Esto se debe a que para que una red neuronal funcione eficazmente y podamos afirmar que el error de test es tal, deberíamos tener una base de datos de millones de muestras con la cual observar si realmente nuestro modelo generaliza lo suficiente como para implementar el sistema en una tienda online.

6 CONCLUSIÓN

Para concluir este documento se ha considerado aportar una valoración técnica del proyecto realizado y un resumen de lo que supone la implementación del mismo en una tienda online de venta de camisetas.

Centrandonos en el desarrollo e implementación del proyecto podemos decir que se ha conseguido realizar un análisis exhaustivo de distintas técnicas englobadas dentro del Machine Learning y Data mining con satisfactorios resultados. Se han perfeccionado técnicas como la Regresión Lineal y SVR o K-means y se han aprendido de nuevas, como el Deep Learning.

En cuanto a los resultados cuantitativos del proyecto cabe destacar que se ha conseguido desarrollar un sistema basado en redes neuronales capaz de predecir tallas de camisetas para hombre y mujer con un error mínimo (0% en entrenamiento y 5% en test). Tal y como se planteó en la introducción del documento, nuestro objetivo no era implementar el sistema dentro de una tienda online de venta de ropa, sino el análisis de las técnicas que nos aportarían un mejor resultado. Así pues, este análisis ha dado sus frutos, tanto sobre el conjunto de datos de entrada para seleccionar nuestro input y generar nuestro output, como sobre el conjunto de parámetros necesarios para optimizar una red neuronal multicapa, permitiendo, a base de exportar el modelo optimizado, incorporar nuestro sistema en empresas de venta online de ropa en futuros proyectos.

Como trabajo futuro se podría implementar este mismo sistema para mujeres y niños y además añadiendo otras piezas de ropa, permitiendo que una tienda de ropa que distribuya a todos los géneros y edades pueda incluir este aplicativo, sin restringir el uso a hombres adultos. Otra de las ramas de investigación en este campo sería llegar a predecir las 166 características que nuestra base de datos contempla a partir de las tres básicas: edad, altura y peso. Ésta última opción aportaría un gran avance en el sector textil ya que no sería necesario tomar medidas de ningún parámetro corporal para realizar cualquier pieza de ropa a medida, sino que introduciendo los tres parámetros mencionados sabriamos con gran exactitud la longitud de la cadera, de la muñeca o incluso el diámetro del cuello. Además, esto nos podría ayudar en la generación de un avatar virtual con el cual se realizarían las compras sin la necesidad de especificar tu talla u otros datos de interés en cada tienda de ropa online.

AGRADECIMIENTOS

En primer lugar me gustaría agradecer a mi tutor Jordi Gonzàlez por la atención y el interés mostrado a lo largo de todo el proceso, así como su dedicación resolviendo mis dudas lo antes posible. También quería darle las gracias al CVC por ofrecerme un ordenador de una potencia suficiente como para poder realizar las pruebas necesarias que requerían un alto nivel de cómputo, y que sin su ayuda no podría haber realizado pruebas tan exhaustivas.

BIBLIOGRAFIA

[1] Misoutlets.com, *Estudio sobre la compra de moda online,* Madrid, Febrero 2011

[2] The Cocktail Anaysis, *El comportamiento del comprador de moda online*, Junio 2011

[3] IAB, *Estudio anual eCommerce IAB Spain*, Junio 2013

[4] Myshape, *www.myshapestylist.com*

[5] Size Me, *www.sizeme.com*

[6] Fits Me, *www.fits.me*

[7] Verisize, *www.verisize.com*

[8] T.Hastie, R.Tibshirani, J.Friedman, *The Elements of Statistical Learning,* 2nd edition, Agosto 2008

[9] C. Ho, C. Lin, *Large-scale Linear Support Vector Regression,* en *Journal of Machine Learning Research,* Noviembre 2012

[10] P. Tan, M.Steinbach , V.Kumar, *Cluster Analysis: Basic Concepts and Algorithms,* en *Introduction to Data Mining*, 2005

[11] Y. Bengio, 4.*Neural Networks for Deep Architectures*, en *Learning Deep Architectures for AI*, 2009

[12] M.Cilimkovic, *Neural Networks and Back Propagation Algorithm,* Dublin (disponible en *http://www.dataminingmasters.com/uploads/studentProjects/NeuralNetworks.pdf*)

[13] R. Rojas, *7-The Backpropagation Algorithm*, en *Neural Networks – A Systematic Introduction,* New York, 1996

[14] Tom M. Mitchell, *Chapter 4: Artificial Neural Networks,* en *Machine Learning*, WCB-McGraw-Hill, 1997

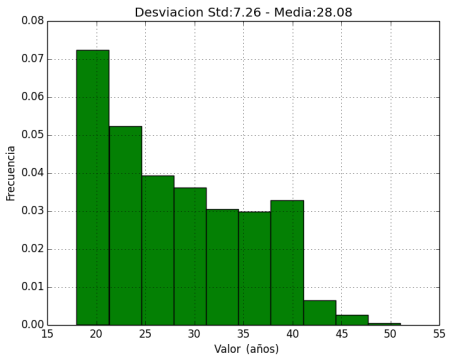
[15] Machine Learning by Andrew Mg, *www.coursera.com*, Standford University

[16] *Confusion Matrix, http://scikit-learn.org/stable/auto_examples/plot_confusion_matrix.html*

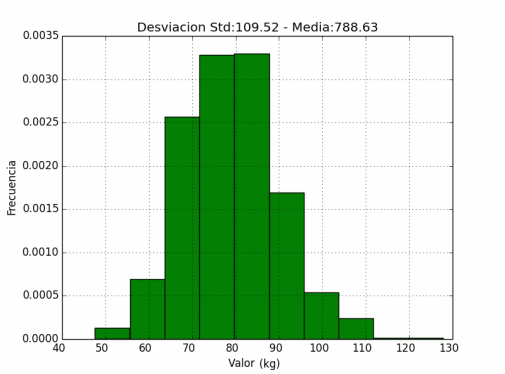
APÉNDICE

A1. CARACTERÍSTICAS MÁS USADAS

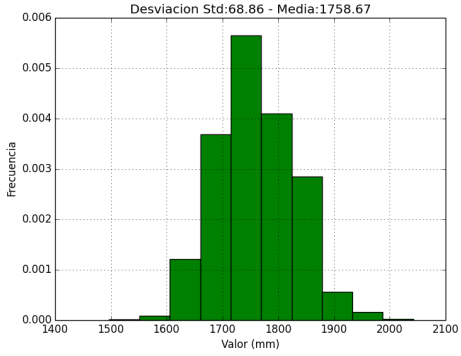
A continuación se adjuntarán los histogramas y una breve descripción correspondientes a la edad, peso y altura. Éstas son las características más comunes en la mayoría de compañías que trabajan con tallas de ropa.



1- Edad: Aunque este parámetro no siga una distribución normal, es importante para detectar a los niños con los cuales no funcionaría nuestro sistema.



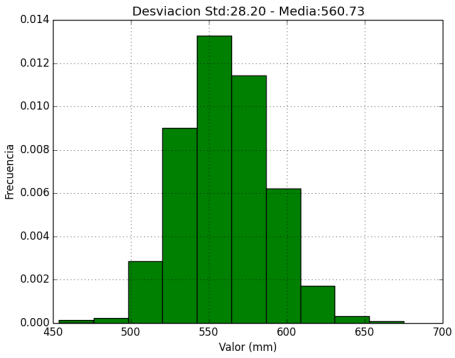
2- Peso: Podemos observar que sí que sigue una distribución normal y se considera el parámetro más importante de los tres ya que es el que consigue predecir al resto con el error más bajo.



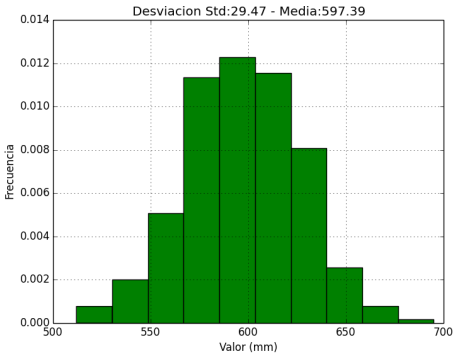
3- Altura: Esta última medida también sigue una distribución normal y aún no siendo el más relevante, es simple de recordar y aporta valor a la red neuronal.

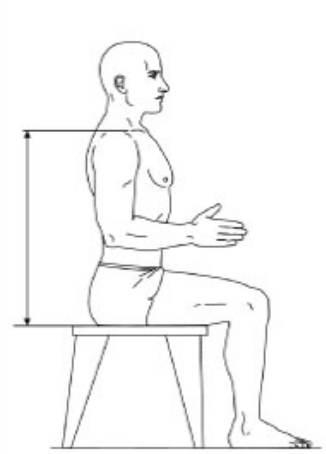
A1. CARACTERÍSTICAS MÁS PRECISAS

En esta sección se incluirán las gráficas de las 5 medidas corporales más descriptivas ordenadas por relevancia, junto con sus descripciones

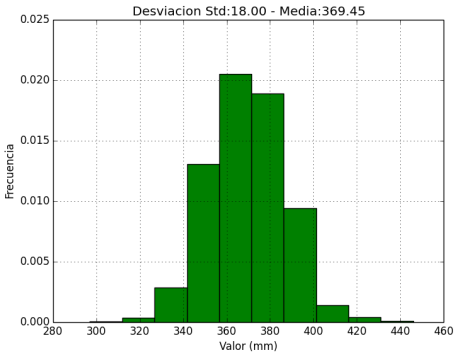


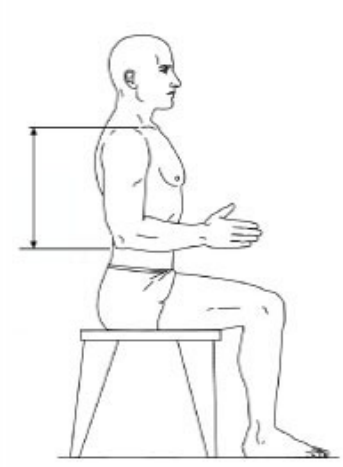
1- Knee Height Sitting: La distancia vertical que hay entre la planta del pie y la parte superior de la rodilla medida con un antropómetro. El sujeto se sienta con los muslos en paralelo, las rodillas flexionadas un 90º y los pies en línea con los muslos.



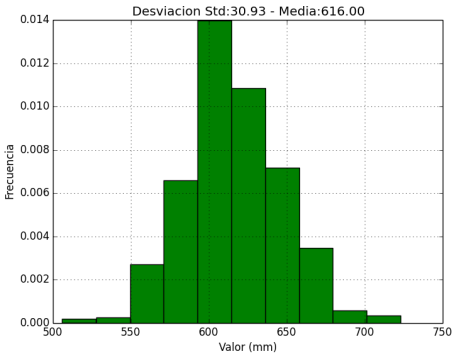


2-Acromion Height Sitting: La distancia vertical que hay entre la planta del pie y la parte superior de la rodilla medida con un antropómetro. El sujeto se sienta con los muslos en paralelo, las rodillas flexionadas un 90º y los pies en línea con los muslos.

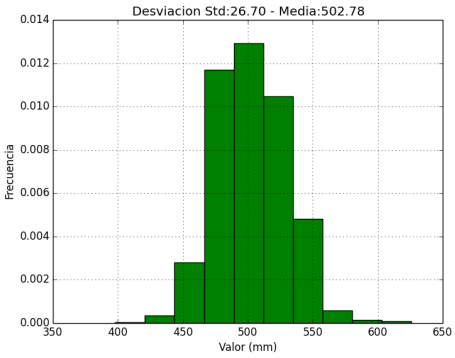


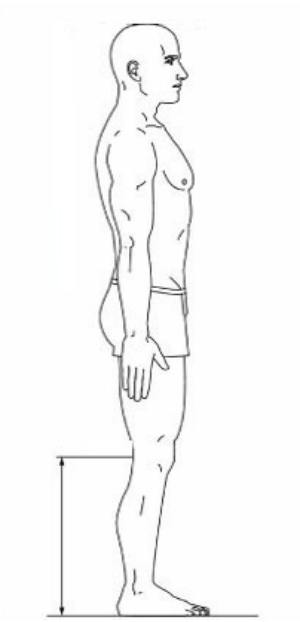


3-Shoulder-Elbow Lenght: La distancia entre el acromion situado en la parte superior del hombro y olecranon en la parte inferior del codo medida con una barra calibradora en paralelo al eje más largo del brazo. El sujeto se pone de pie con el brazo tendido de lado y el codo flexionado a 90º. La mano se mantiene recta y la palma de la mano hacia adentro.



4-Buttock-Knee Lenght: La distancia horizontal entre la parte superior de las nalgas y el punto anterior entre las rodillas medida con un antropómetro. El sujeto se sienta erguido. Los muslos en paralelo con las rodillas flexionadas a 90º con los pies en línea con los mismos.





5-Lateral Femoral Epicondyle Height: La distancia vertical entre una superficie de apoyo y el femoral epicóndilo lateral en el exterior de la rodilla derecha medida con un antropómetro. El sujeto se pone erguido con los talones juntos y el peso distribuido por igual en los pies.