

UNIVERSIDAD COMPLUTENSE DE MADRID  
UNIVERSIDAD POLITÉCNICA DE MADRID

MÁSTER EN TRATAMIENTO ESTADÍSTICO-COMPUTACIONAL DE LA  
INFORMACIÓN



El poder predictivo del modelo de personalidad humana Big Five  
en el ámbito de la ciencia de datos

Daniel Gómez Aguado

Curso académico 2021 - 2022

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. El modelo de personalidad Big Five</b>	<b>4</b>
2.1. Tests de personalidad . . . . .	5
<b>3. Probema a resolver</b>	<b>6</b>
<b>4. Metodología</b>	<b>7</b>
4.1. Fuente de los datos . . . . .	7
4.2. Tratamiento de datos . . . . .	7
4.3. El algoritmo a desarrollar . . . . .	8
4.4. Primera etapa: máquina de soporte vectorial . . . . .	8
4.5. Segunda etapa: los árboles de decisión . . . . .	9
<b>5. Resultados (I)</b>	<b>10</b>
5.1. Comparación de rendimiento . . . . .	10
5.2. Ramas de los árboles . . . . .	12
<b>6. Discusión (I)</b>	<b>14</b>
<b>7. Lógica difusa</b>	<b>15</b>
7.1. Definición de número borroso . . . . .	15
7.2. Aplicación . . . . .	16
7.3. Grado de pertenencia . . . . .	18
<b>8. Resultados (II)</b>	<b>19</b>
8.1. Detalles previos . . . . .	20
8.2. Grado de pertenencia igual o mayor a 0.5 . . . . .	21
8.3. Grado de pertenencia igual o mayor a 0.65 . . . . .	22

8.4. Grado de pertenencia igual o mayor a 0.75 . . . . .	23
<b>9. Discusión (II)</b>	<b>24</b>
<b>10. Conclusión y vistas a futuro</b>	<b>24</b>

## 1. Introducción

El aprendizaje estadístico tiene cabida en múltiples áreas de estudio. El desarrollo de la ciencia de datos permite detectar nuevas relaciones no disponibles a simple vista, que precisaran del uso de un gran número de registros y su posterior tratamiento para detectarse.

En particular, el desarrollo de nuevas máquinas u algoritmos mejora en gran medida a los modelos previos de los que se disponían de clasificación de datos, que de una forma más coloquial, siempre ha sido el pan de cada día: todos classificamos y juzgamos productos, acciones, o eventos en categorías, o más pertenecientes a algunas de ellas que otras.

En el campo más aplicado de la psicología, se han desarrollado numerosos modelos de personalidad. Tienen el afán de hallar un mayor entendimiento del psique humano, y a efectos prácticos, aplicados en la práctica, tienden a desembocar en herramientas de clasificación.

Modelos de personalidad populares con la suficiente rigurosidad científica son:

- El *five factor model* o modelo Big Five (McCrae & John, 1992): estudia la personalidad humana como un modelado compuesto de cinco categorías (por cuyas iniciales el modelo también es conocido como OCEAN): *Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness* y *Neuroticism*. Este modelo se creó originalmente mediante estudios exhaustivos del lenguaje.
- El modelo HEXACO (Michael Ashton and Kibeom Lee, 1994): de seis categorías: *Honesty-Humility*, *Emotionality*, *eXtraversion*, *Agreeableness*, *Conscientiousness* y *Openness to experience*. Por lo que es similar al Big Five, pero intenta paliar defectos del mismo introduciendo *Honesty-Humility*, y asociando ciertas subcategorías (como la irritabilidad o el mal genio) no a *neuroticism*, sino a *agreeableness*.
- El modelo de psicología evolutiva, surgido tras el gran impacto de '*On the Origin of Species*' (Charles Darwin, 1859) en la comunidad científica. Atribuye un porcentaje (alrededor del 30 – 50 %) de la personalidad humana a la genética, y lo restante (50 – 70 %) al entorno de crecimiento de la persona: trato recibido, traumas de nacimiento, experiencias de vida... Este estudio busca responder, además del origen de la personalidad, por qué algunas características de una persona se heredan con mayor o menor frecuencia, o por qué hay más o menos determinado por nacimiento.

Existen otros estudios de personalidad con mayor rechazo por la comunidad científica, pero no por ello carentes de impacto, como el modelo de Myers-Briggs. Este modelo es un puro ejemplo de clasificador: propone 16 tipos de personalidad basándose en cuatro ejes: Extraversion-Introversion, iNTuition-Sensing, Thinking-Feeling y Perceiving-Judging, para asignar un código de 4 letras al individuo en cuestión. Por ejemplo: ESTP, INFP, ISTJ, ENFJ...

**Para nuestro proyecto**, y retornando al tópico de la ciencia de datos, nos interesaremos en el test de personalidad de internet con mayor relevancia para el primer modelo de los citados: el Big Five. Antes de comenzar con el planteamiento del problema en cuestión, es necesario revisar con mayor detalle las categorías fundamentales de este modelo.

## 2. El modelo de personalidad Big Five

Este modelo se fundamenta en una distinción de cinco áreas generales que clasificarían la personalidad de una persona. Explicadas de forma esquematizada:

- **Openness**, o Apertura a la experiencia en español. Esta faceta contiene características tales como la imaginación o perspicacia, curiosidad por el mundo y/o personas, o el interés por aprender y vivir experiencias nuevas. Quienes puntúan alto en ello suelen tener numerosos intereses, más aventureros y creativos, mientras que aquellos con puntuación más baja tienen más interés en lo convencional.
- **Conscientiousness**, o Escrupulosidad en español. Una persona escrupulosa (alta puntuación) tiene altos niveles de autoconsciencia, buen control de impulso, y es centrada en sus tareas. Por contraste, aquellos que no (baja puntuación) se encuentran en mayor comodidad sin considerar estos factores, más por libre y sin tanto control en detalles u organización.
- **Extraversion**, o Extroversión en español. Se entiende mejor considerando al eje extroversión-introversión como la polaridad de una persona para poder recargarla": mientras que los extrovertidos (con alta puntuación) se alimentan de interacción con el mundo exterior, no es así con los introvertidos (de baja puntuación) que precisan de un entorno más privado para "recargar las pilas".
- **Agreeableness**, o Amabilidad en español. Predisposiciones a confiar, al altruismo, al afecto, o más comportamientos pro-sociales se incluyen aquí. Las personas de alta puntuación tienden a ser más cooperativas que aquellas que no, que por su parte tienden más a la competición (que no se interprete como si estas fuesen son excluyentes!).
- **Neuroticism**, o Neuroticismo en español. La frecuencia de periodos de tristeza o melancolía, así como la inestabilidad emocional, son tenidas en cuenta por esta categoría. Aquellos que puntúan alto en esta categoría suelen experimentar ansiedad, irritabilidad, o cambios de ánimo. Sus opuestos son por contraste más resistentes y menos propensos a pasar por estas experiencias.

A partir de este punto, nos referiremos a las categorías por sus nombres en inglés (más cortos, y dan sentido al nombre del modelo OCEAN).

La evaluación se realiza mediante la obtención de una puntuación en cada categoría. Superando o encontrándose por debajo de cierto valor, determina si la persona siendo evaluada ha calificado como de alta o baja puntuación en cada uno de los ejes disponibles.

El test Big Five es, a día de hoy, uno de los modelos dominantes en estudios de psicología humana. Su capacidad de predicción puede entenderse mejor considerando que, además de estas 5 categorías principales, el estudio considera un buen número de subcategorías pertenecientes a cada una de las cinco anteriores. Esto es especialmente detectable en los tests de personalidad de internet, pudiéndose usar el oficial como ejemplo ilustrativo al ser el más visual.

## 2.1. Tests de personalidad

En la población hay un gran interés por el descubrimiento personal. Siendo el internet una poderosa herramienta, hoy en día disponible para un amplio porcentaje de la población y en el que puede buscarse toda clase de contenido, pueden encontrarse (sea por accidente o con la intención) muchísimos tests de identificación personal con determinadas categorías.

En el ámbito de la psicología humana, el Big Five, el MBTI o el eneagrama de personalidad (no mencionado en la introducción; a modo de resumen, se fundamenta en categorizar por motivaciones personales principalmente) son algunos modelos de los que más resultados se encuentran con una simple búsqueda. Del modelo Big Five, los siguientes dos tests son fuertemente relevantes:

- *Free open-source BigFive personality traits test - Big Five*[1]. Este es un proyecto web desarrollado bajo licencia MIT, primer resultado en páginas de búsqueda. Es tan utilizado por su simple pero efectiva interfaz y el detalle en los resultados de cada categoría. Por ejemplo, en la Figura 1 se revisa la puntuación de Agreeableness en relación a lo puntuado en otras seis subcategorías (*Trust, Morality, Altruism, Cooperation, Modesty y Sympathy*). En total, estamos hablando de preguntas que evalúan 30 categorías diferentes (que se agrupan en 5).
- El test IPIP-NEO (International Personality Item Pool Representation of the NEO PI-R™)[2]. Este test se compone de 300 preguntas en su versión larga y 120 en su versión corta. Es un test que hace uso de una escala de puntuación, revisada con los años, en la que se evalúa cómo pesa cada respuesta a una pregunta al cómputo total. Es un modelo bien valorado, con una de sus aplicaciones más modernas, el producto NEO PI-R® (Paul T. Costa, Jr. y Robert R. McCrae) siendo considerada la más eficaz para la aplicación del modelo Big Five.

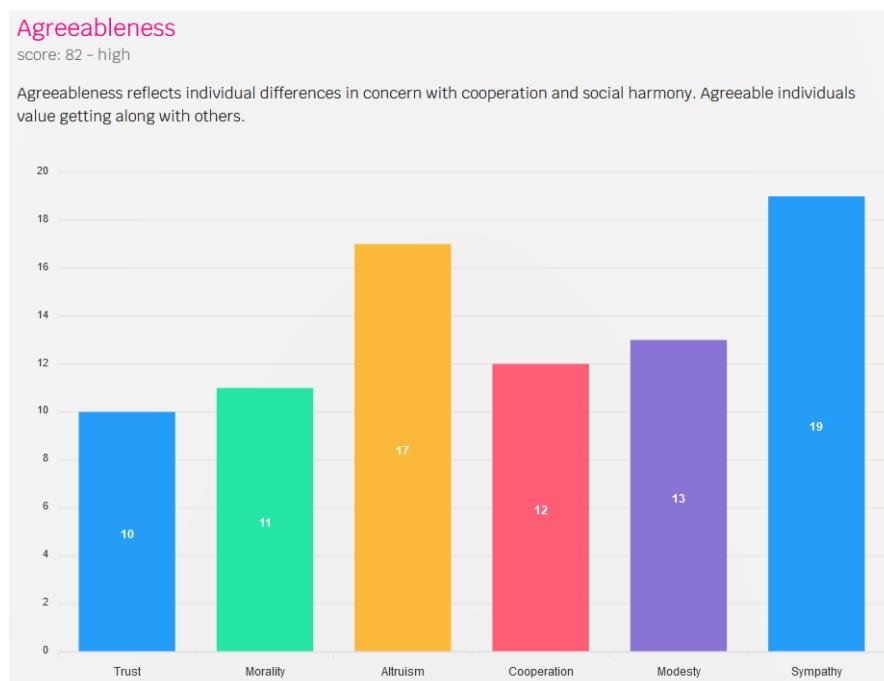


Figura 1: análisis del eje de *agreeableness* en detalle, en el test de [bigfive-test.com](http://bigfive-test.com).

### 3. Problema a resolver

Estos tests de personalidad que circulan por las distintas páginas web, y los de la plataforma IPIP sin excepción, tienden a menudo a ser rodeados por distintos tipos de sesgo. Hay dos a destacar:

- El sesgo en la metodología de **puntuación**. Al plantear las preguntas, muchas veces la puntuación aportada a según qué categorías puede ser inadecuada o mejor dicho, **limitada**.
- El sesgo en las respuestas dadas por el usuario. Es mucho más complicado de analizar que el anterior, puesto que normalmente se asume sinceridad y reflexión por parte del individuo al momento de realizar esta clase de evaluaciones.

Existen algunos tests que, tratando ambos problemas, tratan de predecir un “tipo falso” y un “tipo real”, a partir de analizar las preguntas más confusas de tests, o más propensas a distintas interpretaciones que puedan dar parte de una tipología errónea o sesgo por parte del usuario.

En el test que nos vamos a centrar, el IPIP-NEO, se tiene a disposición del usuario las tablas de puntuación empleadas. Estas únicamente puntúan una de las cinco categorías +1 o -1, sin posibilidad de que aporten valor a varias simultáneamente, o que unas tengan mayor peso que otras.

Disponemos de usuarios que, previamente y siguiendo la metodología dada por la página oficial de IPIP, han reconstruido la puntuación que correspondería a 300000 usuarios que completaron con éxito el test largo de IPIP-NEO (archivo .csv también disponible en internet, con las respuestas dadas por cada uno).

Trataremos de analizar todos estos datos. Tomando una muestra, hay varios aspectos a tener en cuenta.

- Primera etapa: encontrar si realmente podemos predecir estos datos eficientemente. Para ello, puesto que las puntuaciones se encuentran en estado continuo (porcentajes, dominio [0, 1]), se discretizarán y tratarán como clases que permita a una máquina de clasificación supervisada realizar su trabajo.
- Segunda etapa: interpretabilidad. Es crucial en un proyecto como este: queremos, a partir de los parámetros de entrenamiento (que son las respuestas a la pregunta) predecir un target ordinal (el tipo de personalidad asignado a partir de la discretización). Para ello, elaboraremos un segundo modelo que encuentre qué preguntas cobran la mayor relevancia en la evaluación de cada uno de los cinco rasgos.

Podría entenderse el proyecto como un intento de ingeniería inversa sobre un algoritmo de cálculos de puntuación ya existente, mediante el uso de máquinas de aprendizaje estadístico.

Encontrar nuevas relaciones entre usuarios y preguntas mediante el algoritmo adecuado arrojará luz en estos defectos de diseño de tests, dándose paso a nuevos diseños capaces de considerar sutilezas que, de lo hacerse, llevarán a clasificaciones inadecuadas. Si bien ni los modelos de personalidad ni sus tests pretenden establecer una serie de categorías capaces de predecir completamente el desarrollo psicológico de los individuos, son una herramienta útil para el descubrimiento personal de bastantes personas, y es esto lo que motiva la resolución de este problema.

## 4. Metodología

### 4.1. Fuente de los datos

Propiciada originalmente por el repositorio de IPIP<sup>1</sup>, se dispone de un dataset de muchos usuarios, realizando el test largo de IPIP-NEO (de 300 preguntas, que juntas determinan una puntuación en cada una de las cinco divisiones (*Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness* y *Neuroticism*).

Los archivos .csv utilizados son:

- **quiz\_questions.csv**: contiene alrededor de 200000 respuestas al test de usuarios identificados mediante la columna **id**.
- **big\_five\_scores.csv**: contiene alrededor de 300000 puntuaciones calculadas, para usuarios identificados mediante la columna **case\_id**

Debido al incidente previo con la dificultad para encontrar los datos, se descargaron una vez (finalmente) encontrados y se introducen así a un Jupyter Notebook (Python 3) de nombre **BigFiveProjectRNAE.ipynb**.

Al ser dos datasets y haber sido complicada su búsqueda, columnas como **age**, **sex** o **country** ayudaron a verificar que las puntuaciones de **big\_five\_scores.csv** correspondieran con usuarios de **quiz\_questions.csv**.

### 4.2. Tratamiento de datos

Columnas a tener en cuenta en el tratamiento de datos:

- **id** o **case\_id** (para ambos renombrada a **id**). Identifican a cada usuario.
- 01.1-01.10, 02.1-02.10... 06.1-06.10 (y similar para el resto de categorías C, E, A, N). Comprenden las respuestas a las preguntas, en una escala de 1 a 5, entera, de menor a mayor acuerdo del usuario con lo expuesto.
- **openness\_score**, **conscientiousness\_score**, **extraversion\_score**, **agreeable\_score** y **neuroticism\_score**. Son las columnas de puntuación de cada usuario por categoría.

Ya que esta etapa consiste en una sucesión de pasos, se enlistan para su seguimiento en el cuaderno:

- 1. Quitamos las columnas carentes de relevancia de cada dataset, quedándonos solo con la columna de identidad **id**, las columnas de puntuación de categorías y las de respuesta a cada pregunta.

---

<sup>1</sup>Actualmente, estos datos fueron eliminados del repositorio y requirieron de una búsqueda más exhaustiva. Se adjuntan junto con lo demás del proyecto.

- 2. Tenemos tantísimos datos que no podremos procesarlos todos. Por ahora, limpiaremos aquellos con missings. También filtramos aquellos usuarios presentes en el dataset de puntuaciones, pero no en el de respuestas a preguntas.
- 3. Se discretizan las variables de puntuación. Como encontrar un criterio para marcar el corte es complicado, optamos por hallar la media para el dataset actual (95450 usuarios) de cada puntuación. Utilizamos dicha media para marcar el corte, en categorías tales como **O+** y **O-** para *openness*. Similar para el resto de ejes.
- 4. Creamos una nueva columna **type**, que junte los cinco valores en las puntuaciones discretizadas para crear “tipos de personalidad” con el código: **O+C+E+A+N-** (el caso más común).
- 5. Finalmente, y como el dataset actual se encuentra muy desbalanceado para las 32 clases dadas por **type**, realizamos un bajomuestreo de xategorías a la misma cantidad de elementos que la más baja, **O+C-E-A-N-**.

Actualmente, disponemos de un dataset que contiene 32 tipos distintos de personalidad y bajomuestreado, mientras que decidimos conservar el previo con 95450 usuarios y las categorías discretizadas separadas entre sí. Ambos serán necesarios para las siguientes etapas.

### 4.3. El algoritmo a desarrollar

Es momento de decidir qué modelo emplearemos para hallar resultados de los datos. Para este proyecto, se optó por dos implementaciones supervisadas, de cierta sencillez para obtener potentes resultados y, en el caso de segunda etapa, hacerlo con **una elevada interpretabilidad**. El código se fundamenta en la librería **SciKit** de Python para Machine Learning [3].

Para todo entrenamiento de modelos, se utilizó la división 70 % train - 30 % test. El número de filas (usuarios) empleados ha sido por debajo o alrededor de 10000, de los 95450 que actualmente teníamos. Para el dataset de las 32 clases bajomuestreado, se trabajo con 10240 elementos; por contraste, para cada árbol desarrollado, se emplearon 5000.

### 4.4. Primera etapa: máquina de soporte vectorial

Para la primera modelización, se seguirá un modelo **SVM** o de máquina/clasificación de soporte vectorial no lineal de más de dos clases. En nuestra programación, la comparación entre clases se realiza *one vs one* (uno a uno).

Estas dos particularidades en un algoritmo de máquina de soporte vectorial lo hacen bastante complejo. No obstante, es el mejor molde sobre el que podemos trabajar al darnos libertad para escoger un **kernel**, además de diferenciar de forma más definida las fronteras entre clases.

Una vez realizado un Análisis de Componentes Principales (o PCA, abreviado) sobre el dataset de entrenamiento, se continúa operando mediante su introducción a la máquina de soporte vectorial. Para probarla, se ha implementado una *pipeline* o tubería, que en términos de programación es una cadena de procesos que conecta la salida de uno con la entrada del siguiente. Se utilizan diferentes parámetros de kernel (funciones de base radial), así como numerosas propuestas para el hiperparámetro C, y diferentes valores para el número de componentes obtenido del PCA, introducido a la máquina como una serie de parámetros. La tubería encontrará la combinación ganadora.

### **Parámetros introducidos:**

- **Número de componentes:** 23, 63, 129
- **Kernel:** rbf, sigmoid
- **C:** 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0

El resultado principal obtenido de elaborar este algoritmo es una matriz de confusión, que fácilmente permite vislumbrar los fallos o aciertos al categorizar por tipos de personalidad. En el cuaderno, está preparado para una fácil ejecución, ya introducidos los datasets.

Si bien en el proceso se comenta que tomamos muestras para el proceso general, una matriz de confusión adjunta resultante de este proceso, **CM\_TOTAL\_SVM\_95450.png**, se obtuvo para todos los 95450 datos.

### **4.5. Segunda etapa: los árboles de decisión**

Es un modelo de predicción (clasificación o regresión) basado en reglas. Los árboles de decisión son muy intuitivos, pues codifican una serie de elecciones similar a un *if/else* de cualquier lenguaje de programación. La gran ventaja de esta técnica es el **aprendizaje automático** de qué elecciones hacer a partir de los datos, identificándose las mejores elecciones y ramificaciones a lo largo del árbol generado.

#### **Ventajas:**

- Los modelos son fáciles de interpretar y los árboles pueden ser visualizados.
- Los datos de entrada requieren muy poco preprocesamiento.
- El costo computacional del uso del árbol para predecir la categoría de un ejemplo es mínimo comparado con otras técnicas.

#### **Desventajas:**

- Puede ser tan complejo que se memoriza el conjunto de datos (sobreajuste).
- Son muy sensibles al desbalance de clases (sesgo).

Atendiendo a las desventajas, se realizaron cinco iteraciones de validación cruzada por árbol, de modo que obtengamos los más óptimos disponibles para cada caso. Por su parte, el desbalance de clases fue tratado previamente.

En el cuaderno, se han juntado todos los pasos realizados para la construcción del árbol en una función ejecutable, que obtiene el modelo del árbol junto con representaciones gráficas de la matriz de confusión y árbol resultante.

De todos los experimentos con árboles realizados (que a continuación pasaremos a comentar), uno de ellos pudo hacerse para 32 clases del dataset. Puesto que ya disponemos de varias matrices de confusión de la máquina de soporte vectorial, puede ser interesante comparar **CM\_TOTAL\_SVM\_10240.png** con **CM\_TOTAL\_TREE**, obtenida por el árbol.

## 5. Resultados (I)

Dividiremos este apartado en dos etapas.

### 5.1. Comparación de rendimiento

Echemos un vistazo a las matrices de confusión obtenidas para ambos modelos.

Matriz de confusión de la máquina de soporte vectorial: Figura 2.

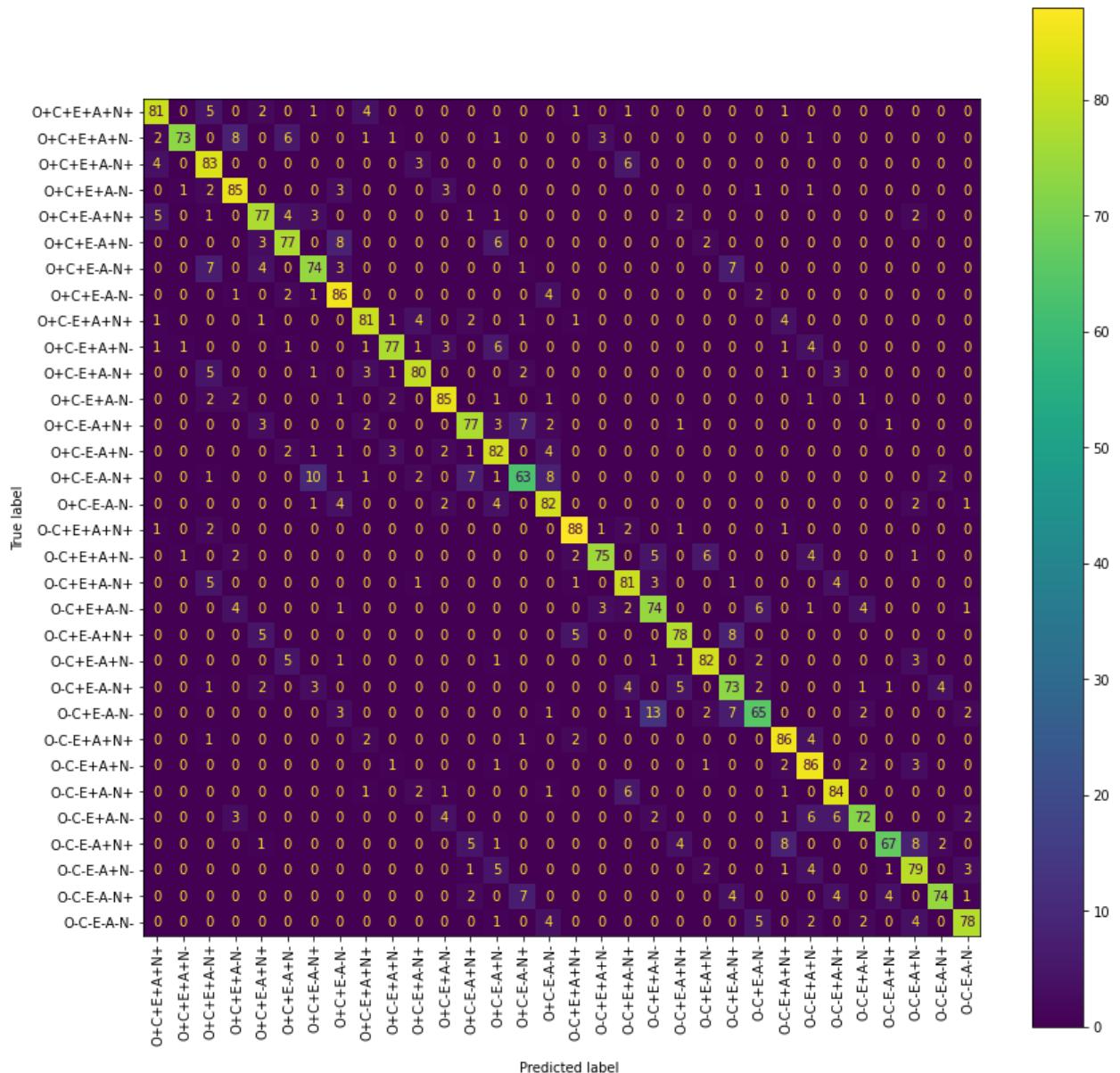


Figura 2: matrices de confusión de las 32 clases para la SVM.

Parece llevar a cabo una clasificación muy precisa del modelo.

Matriz de confusión de la máquina de soporte vectorial: Figura 3.

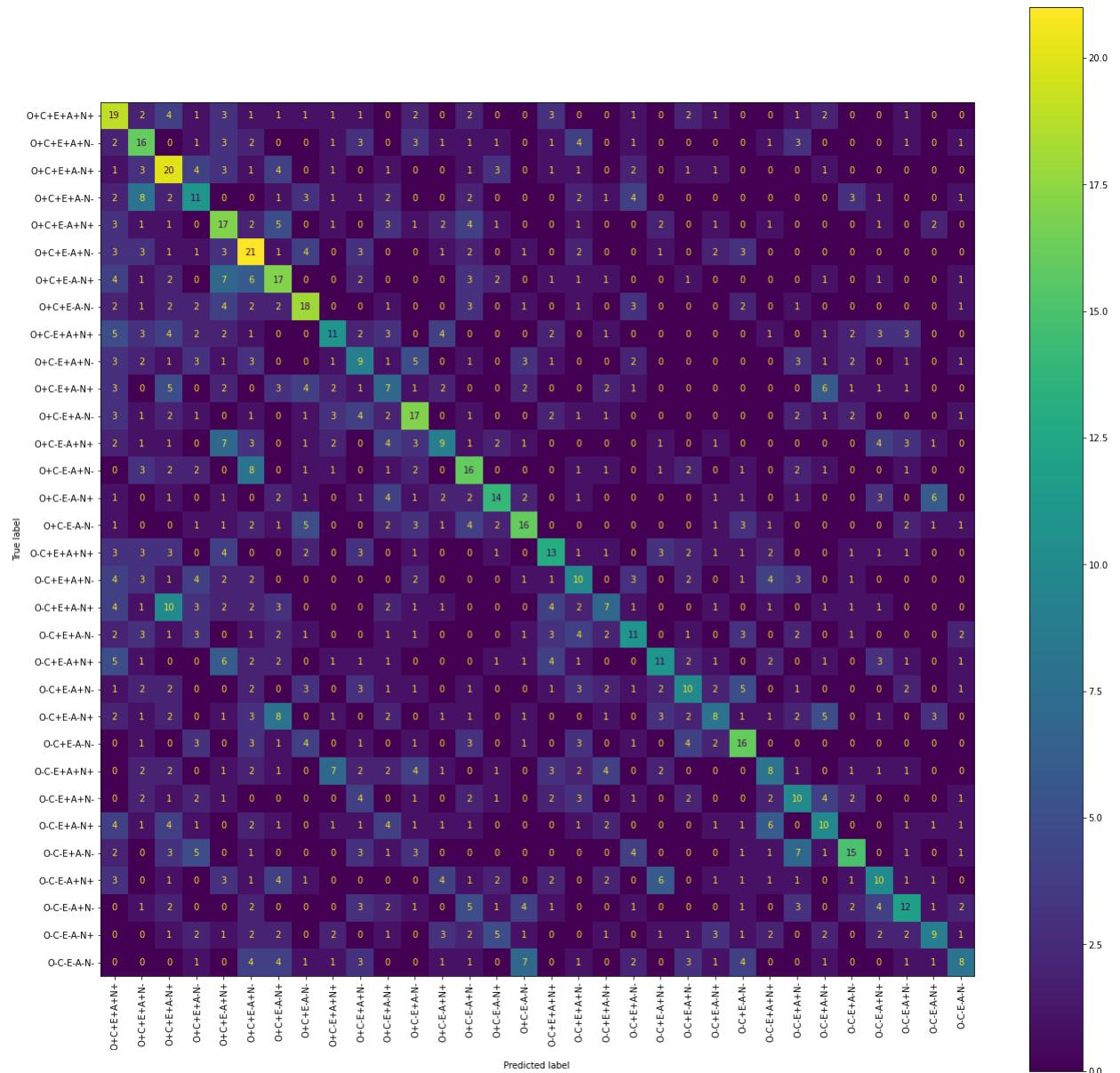


Figura 3: matrices de confusión de las 32 clases para el árbol.

Aunque no realiza mal la clasificación, presenta más problemas para categorizar ciertas clases. Por ejemplo, no distingue completamente a los usuarios de personalidad **O+C-E+A+N-**, **O+C-E-A-N+** o **O-C+E-A-N-**.

Curiosamente, el árbol, pese a su peor rendimiento como clasificador, encuentra entre sus mejores clasificaciones a la realizada a **O+C-E+A+N-**. Si pudiera mejorarse y ejecutarse para más datos (para lo que requeriría mayor rendimiento, ya que no se podía subir mucho más de 5000 filas), tal vez podría hacer una comparativa más sustanciosa con la máquina de soporte vectorial como clasificador.

## 5.2. Ramas de los árboles

A continuación, pasaremos a mostrar las ramas obtenidas para cada árbol elaborado por ejes por separado (O, C, E, A, N).

Realizamos tres iteraciones de cada árbol, para observar cómo es la toma de decisiones en cada caso. Es decir, las preguntas que se realiza en las primeras capas de profundidad.

Dado que se estableció en el planteamiento del problema la presencia de sesgo en este tipo de tests de personalidad (por la selección de puntuación por pregunta siendo muy pobre), nos fijaremos en **preguntas no pertenecientes originalmente al eje para el cual se realizó el árbol**, hasta la capa de profundidad 6. Los resultados para cada eje se presentan a continuación (en inglés), junto con su profundidad:

### *Openness:*

- Depth=3; Act wild and crazy
- Depth=3; Believe in an eye for an eye.
- Depth=4; Often feel uncomfortable around others.
- Depth=4; Like to act on a whim.
- Depth=5; Tend to dislike soft-hearted people.
- Depth=5; Have a sharp tongue.
- Depth=5; Take control of things.
- Depth=6; Value cooperation over competition.
- Depth=6; Do a lot in my spare time.

### *Conscientiousness:*

- Depth=3; Get back at others.
- Depth=4; Take advantage of others.
- Depth=4; Am comfortable in unfamiliar situations.
- Depth=5; Like to stand during the national anthem.
- Depth=5; Stick to the rules.
- Depth=5; Believe laws should be strictly enforced.
- Depth=5; Stumble over my words.
- Depth=5; Am afraid of many things.
- Depth=6; Can't make up my mind.
- Depth=6; Would never cheat on my taxes.

- Depth=6; Take advantage of others.

***Extraversion:***

- Depth=3; Have a low opinion of myself.
- Depth=4; Make people feel welcome.
- Depth=4; Do the opposite of what is asked.
- Depth=4; Am comfortable in unfamiliar situations.
- Depth=4; Feel comfortable with myself.
- Depth=5; Get caught up in my problems.
- Depth=5; Do crazy things.
- Depth=5; Like to begin new things.
- Depth=6; Believe that too much tax money goes to support artists.
- Depth=6; Like to act on a whim.
- Depth=6; Enjoy wild flights of fantasy.
- Depth=6; Am not bothered by messy people.
- Depth=6; Am comfortable in unfamiliar situations.
- Depth=6; Am not interested in theoretical discussions.
- Depth=6; Easily resist temptations.
- Depth=6; Stumble over my words.
- Depth=6; Have a sharp tongue.

***Agreeableness:***

- Depth=3; Rarely notice my emotional reactions.
- Depth=4; Demand quality.
- Depth=5; Have a lot of fun.
- Depth=5; Lose my temper.
- Depth=5; Can talk others into doing things.
- Depth=5; React quickly.
- Depth=5; Seek to influence others.
- Depth=6; Am passionate about causes.

- Depth=6; Have difficulty starting tasks.
- Depth=6; Like order.
- Depth=6; Get angry easily.
- Depth=6; Break rules.

***Neuroticism:***

- Depth=4; Waste my time.
- Depth=4; Carry out my plans.
- Depth=4; Don't understand things.
- Depth=4; Look at the bright side of life.
- Depth=5; Put little time and effort into my work.
- Depth=6; Enjoy being part of a loud crowd.
- Depth=6; Do a lot in my spare time.
- Depth=6; Would never go hang gliding or bungee jumping.

La estructura de árbol también permite observar a qué preguntas suele dar el algoritmo más peso, observándose tendencias en la raíz y primeras divisiones de cada eje.

Ha sido posible conocer a qué preguntas se respondía haciendo uso del fichero `IPIP-NEO-ItemKey.xlsx` para la traducción del código de ordenación de éstas en el dataset (las 300 preguntas tienen divisiones marcadas de 30 en 30, donde se recorren A1-A6 y similar para el resto. De ahí los nombres de las columnas).

## 6. Discusión (I)

Observando las preguntas nuevas señaladas en el apartado de Resultados para cada eje del modelo Big Five, hay varios detalles generales a tener en cuenta.

- *Extraversion* es la categoría con mayor cantidad de estas preguntas, entre los tres árboles. El motivo es que la introversión y extroversión son conceptos que abarcan más que los demás, y por tanto puede manifestarse en numerosas otras preguntas. Muchas de las preguntas tienen que ver con evadirse (introversión), o cuestiones sociales (extroversión).
- *Neuroticism* es por contraste la categoría más robusta o hermética de las cinco, no precisando de demasiada revisión. Uno de los tres árboles obtenidos, el más simple, no poseía ninguna pregunta utilizada en el test para evaluar otro eje. No obstante, este aislamiento puede ir vinculado con lo especial de la categoría: es la que tiene menor puntuación de media entre todos los usuarios, y la única cuyas preguntas pueden ser comúnmente asociadas a conceptos más específicos como la ansiedad.

- Hubo un modelo posterior al Big Five (el HEXACO) que ya decidió cambiar categorías por encontrar correlaciones (que las pudieran hacer redundantes) y aspectos no tan efectivos del Big Five. Una de ellas, aspectos de *neuroticism* enviados a *agreeableness*, pueden ser detectados (en agreeableness: React quickly, Have difficulty starting tasks, Get angry easily; todos aspectos que trata de cubrir el neuroticismo).

A partir de esta información, ¿podrían reconsiderarse las reglas para la realización de esta clase de tests o cuestionarios? Tal y como se vio, evaluar de forma tan limitada y estricta es un sistema muy limitado; aunque es comprensible que no se hiciera preliminarmente por la complicación de estimar aspectos como qué peso le corresponde a cada pregunta para cada eje, con la tecnología actual pueden elaborarse cuestionarios mejores, y que alcancen el mismo peso que este IPIP-NEO.

En cuanto al aspecto técnico de clasificación, gana la máquina de soporte vectorial sin duda. Tal y como se encuentra implementada, es un algoritmo más eficiente.

Por su parte, es gracias al árbol que podemos realizar la evaluación e influencia de preguntas sobre categorías.

## 7. Lógica difusa

Una vez realizado el proyecto original, se estudió la posibilidad de introducir fundamentos de lógica difusa en su algoritmo de categorización. El motivante principal para ello es la diferencia de criterios de evaluación existente entre individuos: las palabras “bajo”, “alto” o “medio” pueden significar magnitudes diferentes entre individuos.

El peso de estas categorizaciones sobre el individuo tambien es un factor a tener en cuenta. En ocasiones, un resultado puede no ser del todo significativo Dicho de otro modo, darse “por descarte” o no haber mejor opción disponible.

Introduciendo la **lógica difusa**, podemos solventar ambas limitaciones del modelado previo. Estructuraremos esta sección definiendo primero qué es un número borroso, que permitirá evaluar la “pertenencia” a cada posible valoración (alto, bajo, medio...) por categoría en un usuario.

Para incluir la lógica difusa a nuestro algoritmo en Python, empleamos la librería SciKit-Fuzzy [4].

### 7.1. Definición de número borroso

Se define como un subconjunto borroso  $A$  de los números reales  $\mathcal{R}$  tal que:

- Es convexo: para todo  $x, y \in \mathcal{R}$  y  $\lambda \in [0, 1]$  se cumple  $A(\lambda x + (1 - \lambda)y) \geq \min(A(x), A(y))$ .
- Tiene un único valor modal  $x \in \mathcal{R} | A(x) = 1$ .
- Soporte acotado  $sop(A) = \{x \in \mathcal{R} | A(x) > 0\}$ .
- Función de pertenencia  $\mu_A : \mathcal{R} \rightarrow [0, 1]$  continua.

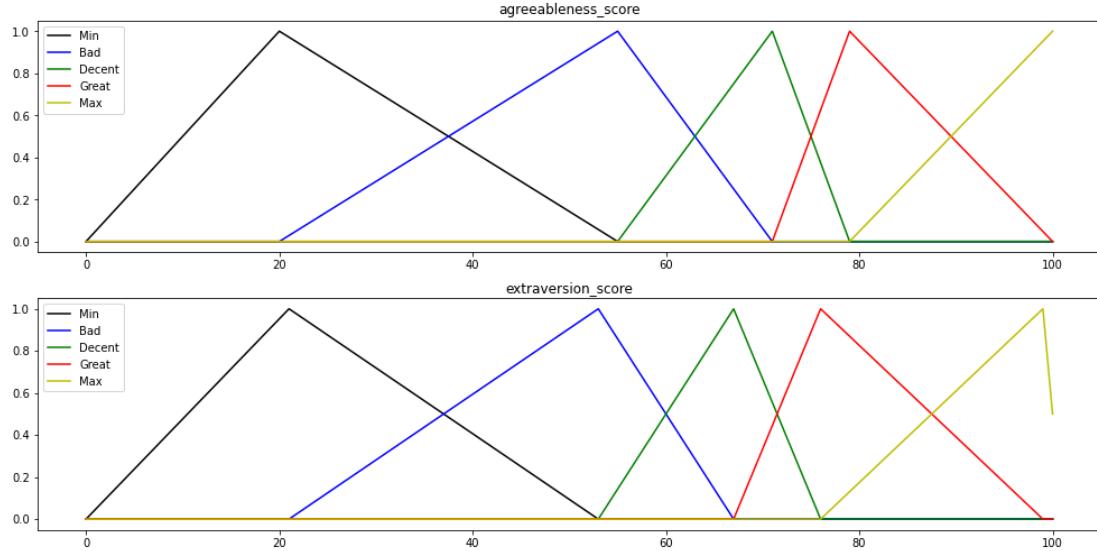


Figura 4: números borrosos triangular para dos de las cinco categorías.

Las nociones matemáticas empleadas en la definición acarrean resultados como los observados en la Figura 4.

## 7.2. Aplicación

Para el caso que nos ocupa, podemos escoger entre dos acercamientos:

- Dar una definición completamente manual de los números borrosos para cada categoría: por ejemplo, que el número borroso de “bajo *Openness*” tiene su valor modal en  $x = 0,25$ , el medio en  $x = 0,5$  y el alto en  $x = 0,75$ .
- Aprender a partir de los datos ya presentes en el DataBase: tenemos 95450 usuarios disponibles. Calculando los promedio, máximo y mínimo valores de puntuación por categoría tras tomar el test, podemos partir de valores más adecuados. Para introducir categorizaciones intermedias, haríamos interpolaciones entre estos promedio y máximo/mínimo, dando un peso a cada uno.

Hemos escogido la segunda opción. Propondremos cinco números borrosos correspondientes a “muy bajo”, “bajo”, “medio”, “alto” y “muy alto”, visibles en la anterior Figura 4 (funciones de pertenencia triangulares) o las Figuras 5 y 6 (funciones de pertenencia gaussianas y gaussianas-sigmoidales, respectivamente).

En el código, se da la opción de trabajar con el modelo triangular, o gauss-sigmoidal, al ser el primero el caso más simple de definición y el segundo aquel que dio mejores resultados en lo que respecta a ajuste en la matriz de confusión.

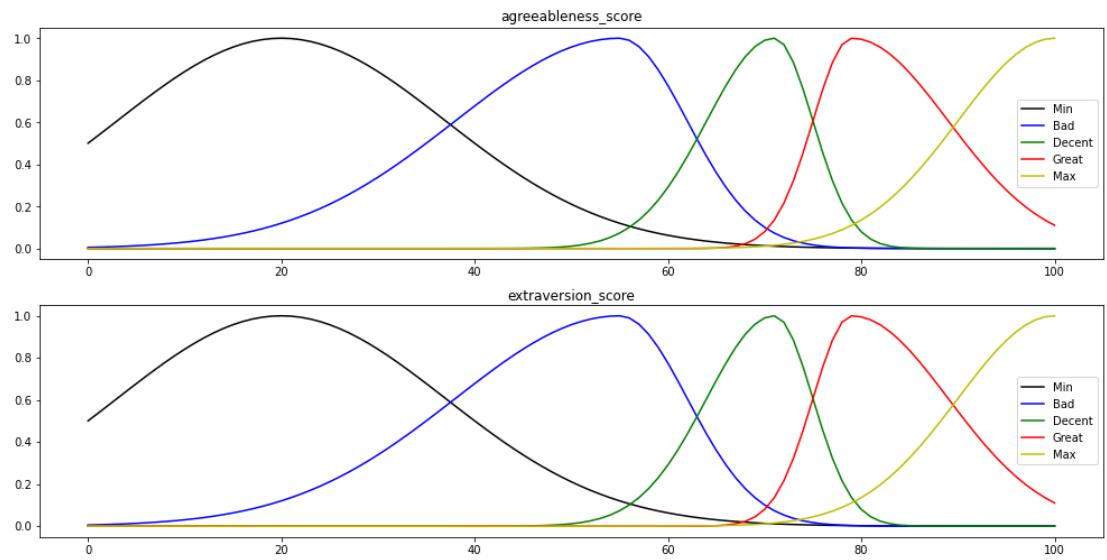


Figura 5: números borrosos de gauss para dos de las cinco categorías.

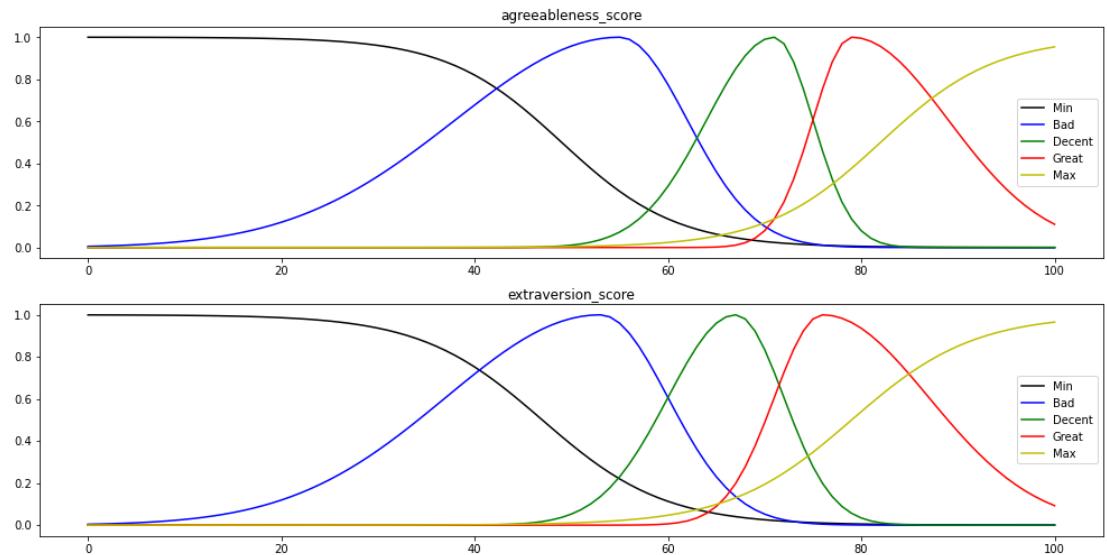


Figura 6: números borrosos de gauss + sigmoidal para dos de las cinco categorías.

### 7.3. Grado de pertenencia

Para cada valor de  $x$  referente a una categoría (o conjunto), un número borroso  $A$  da un valor denominado **grado de pertenencia**:  $A(x)$ . Es indicativo de la adecuación del valor de la categoría (*Neuroticism*, por ejemplo) a la valoración (“alto”, “bajo”, “medio”).

Así, a un usuario que realizó el test, con cinco puntuaciones para cada categoría obtenidas, se le podrán calcular grados de pertenencia a un número de posibles valoraciones dado (en nuestro caso, las cinco mencionadas previamente) por categoría. Es decir:  $5 * 5 = 25$  grados de pertenencia en total.

**Ejemplo:** en la categoría de *Agreeableness*, un usuario obtiene  $[0,17, 0,76, 0,45, 0,00, 0,00]$  grados de pertenencia. La lectura del string es [“muy bajo”, “bajo”, “medio”, “alto”, “muy alto”], por lo que en este caso la valoración con mayor grado de pertenencia para *Agreeableness* en el individuo es “bajo”.

Ya que una clasificación completa requiere de considerar las cinco categorías, y disponemos de cinco posibles valoraciones para cada una, como resultado habrá  $5^5 = 3125$  posibles combinaciones. Para no requerir de generarlas todas, recurramos a componer las clases presentes esporádicamente.

Denominando cada calificación individual de forma simplificada:

Categoría	“Muy bajo”	“Bajo”	“Medio”	“Alto”	“Muy alto”
<i>Openness</i>	O-	o-	--	o+	O+
<i>Conscientiousness</i>	C-	c-	--	c+	C+
<i>Extraversion</i>	E-	e-	--	e+	E+
<i>Agreeableness</i>	A-	a-	--	a+	A+
<i>Neuroticism</i>	N-	n-	--	n+	N+

Sean  $A_O$ ,  $A_C$ ,  $A_E$ ,  $A_A$ ,  $A_N$  los grados de pertenencia para cada categoría. La condición de pertenencia a una clase cualquiera es: “if  $A_O$  is low AND  $A_C$  is low AND  $A_E$  is very high AND  $A_A$  is high AND  $A_N$  is medium  $\rightarrow$  clase o-c-E+a+\_\_”.

En nuestro código, realizamos primero las valoraciones individuales con los números borrosos, obteniéndose así los grados de pertenencia por separado. Seguidamente, por las reglas de pertenencia se obtiene la clase completa (que no es más que juntar las cinco postuladas previamente). **Para el nuevo grado de pertenencia de la clase “total”, se halla buscando el mínimo entre las presentes (función propuesta por Zadeh).**

Disponer de clasificaciones obtenidas no solo a partir de consideraciones hechas a partir de números borrosos, sino con un parámetro que nos de la pertenencia a dicha clase, permite numerosas consideraciones que veremos en la toma de resultados. Cabe also mencionar que podrían haberse obtenido no solo la clase de mayor grado de pertenencia, sino un top 3, o top 5, de clases de mayor pertenencia, de modo que se disponga de información aun más concreta, personalizada de las preferencias cognitivas de cada usuario.

Tanto lo último mencionado como todo lo desarrollado en este apartado ha sido solo posible gracias a la introducción de la lógica difusa.

## 8. Resultados (II)

Disponiendo de los grados de pertenencia de cada usuario a su clase asignada, puede ser interesante filtrar según una cota mínima. La idea es pasar del conjunto de usuarios más global, a uno clasificado más marcadamente, y ver cómo reacciona la matriz de confusión a ello.

- Pueden aparecer un total de 3125 clases diferentes. No obstante, en el DataBase empleado se detecta que muchas de ellas solo poseen un usuario de casi 100000. Son clases de uno o muy pocos usuarios que pueden dar problemas al elaborar el modelo de entrenamiento. Por consiguiente, hacemos uso de un **método de filtrado de clases** tal que cada clase tenga que aparecer una cota mínima de veces para ser considerada.
  - Quitaremos las clases con resultados neutros para cada categoría “\_”. Son más complicados de visualizar y, para una comparación más directa con el resultado previo obtenido sin lógica difusa, merece más la pena considerar preferencias marcadas en todas las categorías.
  - Apoyando al punto previo, podría plantearse realizar este experimento para tres valoraciones en vez de cinco. De este modo, podría realizarse una comparación aun más directa. Sin embargo, al probarlo, las clasificaciones con un grado de pertenencia notorio no resultaban demasiadas. Teniendo esto en cuenta, queda expuesto que tres valoraciones de partida tal vez no sea lo más adecuado.

Con **cinco posibilidades**, no solo explotamos la facilidad de los números borrosos para permitirnos más posibilidades, sino que **logramos categorizar con mayor eficacia**. Dispondremos así de un número competente de filas para relizar modelos de aprendizaje estadístico.

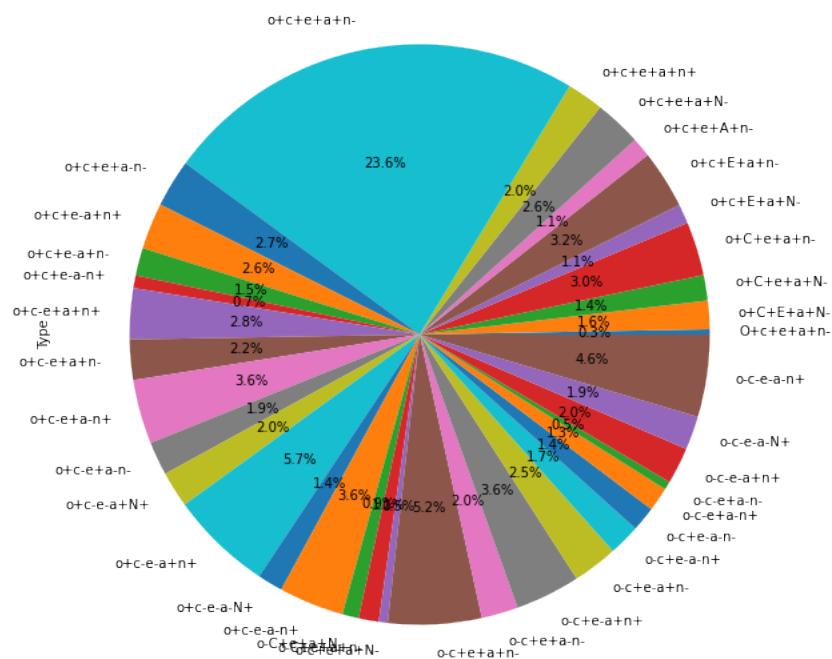


Figura 7: distribución de clases previo a bajomuestreo.

Parámetros tomados y algunas otras consideraciones:

- **Filtrado de clases:** se requerirá que tengan una ocupación mínima del 0,05 % ( $954500,0005 = 47$  mínimas apariciones).
- **Cotas mínima de grado de pertenencia:** se han computado resultados para 0,5 (baja), 0,65 (media) y 0,75 (alta).
- **Números borrosos:** mejores resultados para el caso gauss+sigmoidal.
- **División de conjuntos de entrenamiento y test para la SVM:** 70 % – 30 %.
- Realizamos **bajomuestreo**, ya que habrá clases o tipos que harán aparición en un gran número de ocasiones (ver Figura 7).

Una vez realizadas estas consideraciones, es momento de observar **cómo actúa ahora la máquina de soporte vectorial sobre el conjunto de datos**.

### 8.1. Detalles previos

En el caso de lógica difusa así como el previo, hemos hallado una representación del número de componentes de X (las respuestas a las 300 preguntas, es decir, 300 componentes disponibles) necesarias para la cobertura de determinadas cantidades de varianza del sistema (PCA):

Curva acumulativa de la varianza explicada VS n° de componentes principales

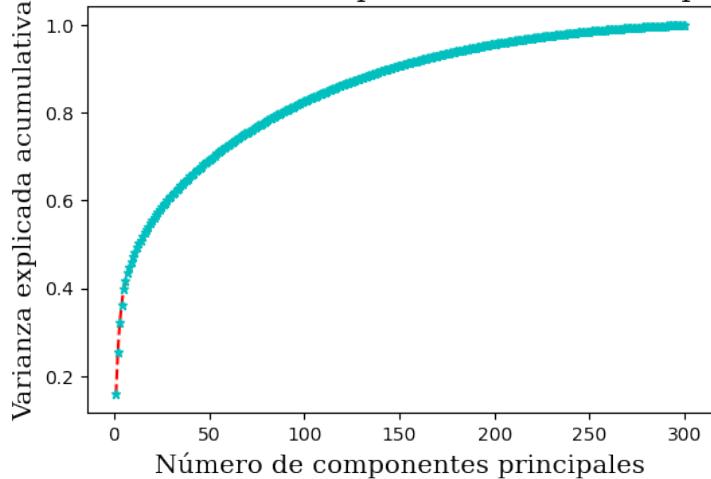


Figura 8: curva acumulativa de varianza explicada, obtenida mediante PCA (caso: 0.75 grado de pertenencia mínimo).

Pueden escogerse ciertos números de componentes principales para el PCA atendiendo a lo observado en la curva. Esta representación ayuda en la toma de decisiones. En este caso, para todo grado de pertenencia que se plantee. Para nuestro caso, es perteneciente a lo mostrado en esta Figura 8 la elección de número de componentes que realiza la máquina, entre varias dadas manualmente.

## 8.2. Grado de pertenencia igual o mayor a 0.5

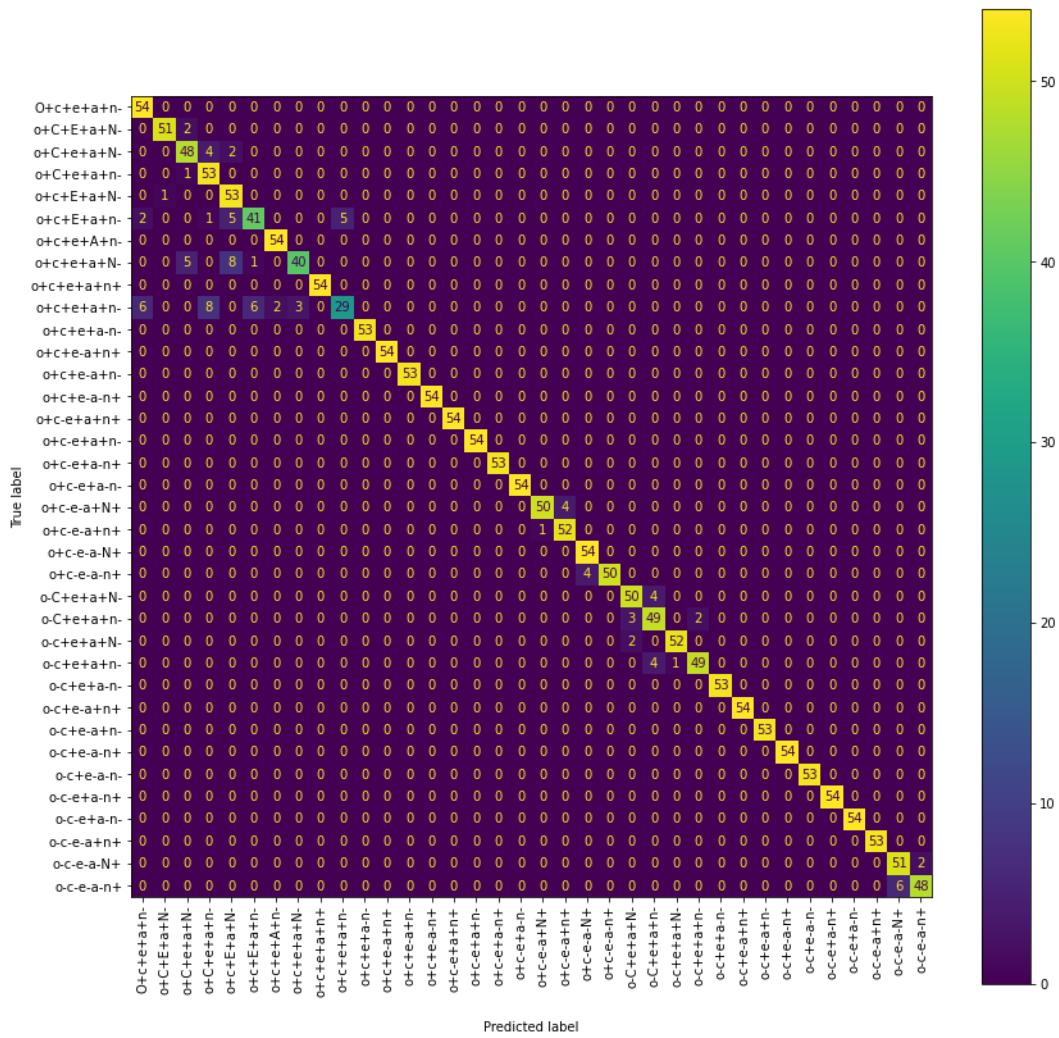


Figura 9: matriz de confusión, grado de pertenencia de **0.5** (bajo).

Dadas las condiciones, actualmente disponemos de 36 clases.  
Forma de los datasets:  
`X_train (4510, 300), y_train (4510,)`  
`X_train (1934, 300), y_train (1934,))`

```
Mejor configuración de parámetros:  
    pca_n_components: 110  
    svc_kernel: rbf  
Exactitud (val): 0.9379  
Exactitud (test): 0.9509
```

Figura 10: tamaño de los datasets y configuraciones del SVM, grado de pertenencia de 0.5 (**bajo**).

### 8.3. Grado de pertenencia igual o mayor a 0.65

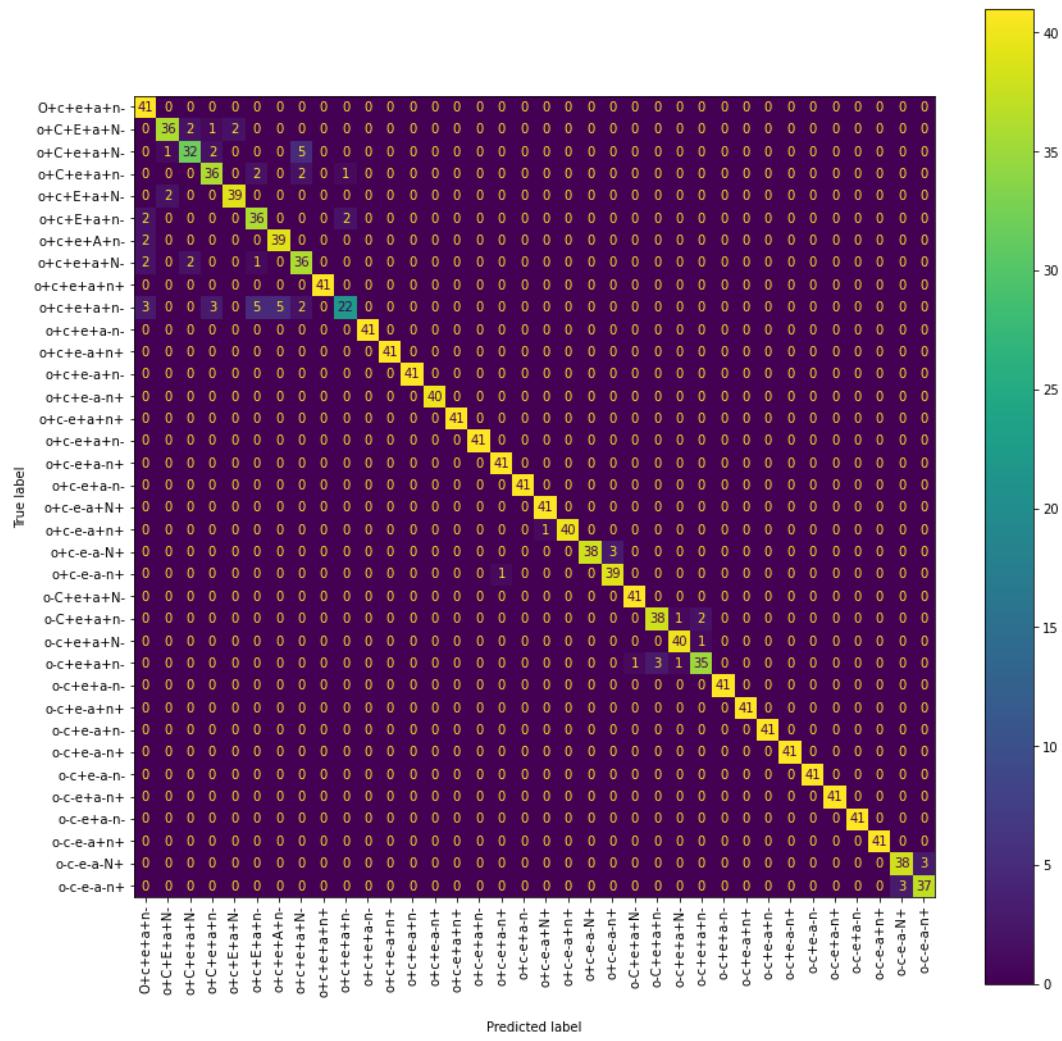


Figura 11: matriz de confusión, grado de pertenencia de **0.65 (medio)**.

Dadas las condiciones, actualmente disponemos de 36 clases  
Forma de los datasets:  
`X_train (3427, 300), y_train (3427,)`  
`X_train (1469, 300), y_train (1469,)`

```
Mejor configuración de parámetros:  
    pca_n_components: 110  
    svc_kernel: rbf  
Exactitud (val): 0.9300  
Exactitud (test): 0.9530
```

Figura 12: tamaño de los datasets y configuraciones del SVM, grado de pertenencia de **0.65** (medio).

#### 8.4. Grado de pertenencia igual o mayor a 0.75

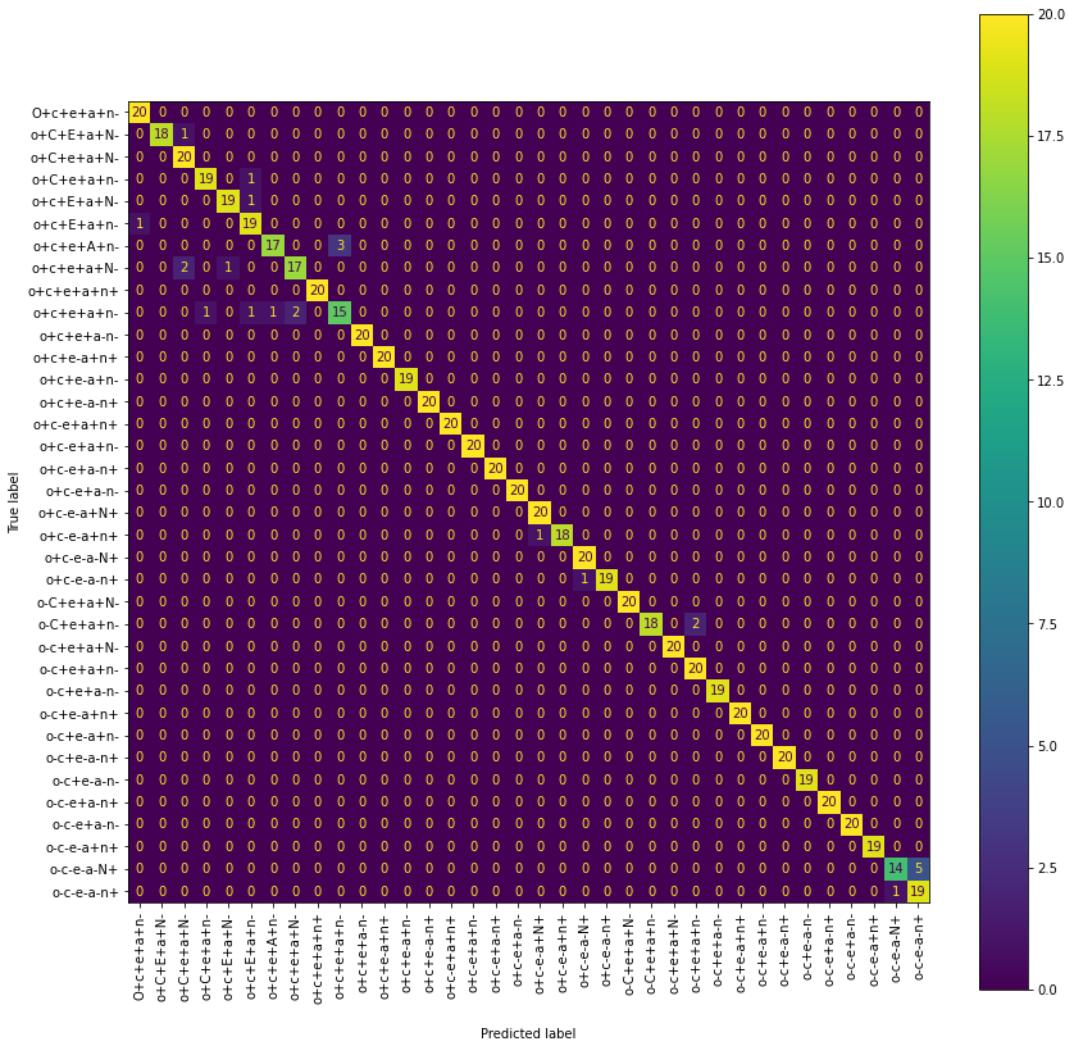


Figura 13: matriz de confusión, grado de pertenencia de **0.75 (alto)**.

Dadas las condiciones, actualmente disponemos de 36 clases  
Forma de los datasets:  
`X_train (1663, 300), y_train (1663,)`  
`X_train (713, 300), y_train (713,))`

```
Mejor configuración de parámetros:  
    pca_n_components: 11  
    svc_kernel: rbf  
Exactitud (val): 0.9609  
Exactitud (test): 0.9649
```

Figura 14: tamaño de los datasets y configuraciones del SVM, grado de pertenencia de **0.75 (alto)**.

## 9. Discusión (II)

La introducción de la lógica difusa ha supuesto numerosas modificaciones en la base de datos, que dio como resultado un modelo más moldeable de clasificación de personalidades humanas.

- El empleo de números borrosos ha mejorado la precisión del modelo de un 0.7426 en validación y 0.7869 en test, a **0.9379** y **0.9509** (grado de pertenencia mínimo de 0.5, el más general).
- Categorizar personalidades humanas es un problema muy complejo, y muchas veces **los resultados de un cuestionario no aportan información de valor (grado de pertenencia bajo)**. Este resultado es novedoso del acercamiento por lógica difusa.
- Se dispone de un menor número de limitaciones al definir las valoraciones según números borrosos. El usuario puede describir diversas formas de hacerlo, sin requerir de extensos ejercicios de imaginación al poner cotas (que sería el equivalente a añadir categorías al caso de lógica binaria, y aun así en este caso no habría grados de pertenencia disponibles).

## 10. Conclusión y vistas a futuro

Hemos llevado a cabo un análisis de datos de un gran número de usuarios respondiendo y obteniendo una puntuación tras la realización del test de personalidad IPIP-NEO. El protocolo ha consistido en desarrollar máquinas capaces de predecir la clasificación realizada, suponiendo al inicio que no conocemos la clave. Tras obtener resultados, utilizamos ésta para comparar cómo fue llevada a cabo la clasificación original, y qué lagunas puede tener tal método observando algunas de las conclusiones realizadas por nuestros árboles de decisión.

El desarrollo previo de una máquina de soporte vectorial nos asegura disponer de un buen poder predictivo para la clasificación correcta de tipos de personalidad, que complementa a la interpretabilidad del problema adquirida por el árbol. Así mismo, con la introducción de la lógica difusa a su diseño, es ahora una herramienta distinguida por diferenciar entre grados de pertenencia de cada usuario, según el criterio por números borrosos dado por el usuario.

En el futuro, podría mejorarse para situaciones tales como: valorar almacenar más de una posible clase por elemento de la base de datos (con sus respectivos grados de pertenencia), para afinar más en los tipos de usuario disponibles; la obtención de nuevos árboles de decisión con lógica difusa, que sirvan de contraposición a los ya ideados mediante lógica binaria, entre otras posibilidades.

## Referencias

- [1] Prueba de personalidad Big Five <https://bigfive-test.com/>
- [2] International Personality Item Pool: A Scientific Collaboratory for the Development of Advanced Measures of Personality and Other Individual Differences <https://ipip.ori.org/>
- [3] SciKit-Fuzzy - skfuzzy v0.2 docs <https://scikit-learn.org/stable/index.html>
- [4] scikit-learn: Machine Learning in Python <https://pythonhosted.org/scikit-fuzzy/overview.html>

## Anexo: matrices de confusión individuales y árboles de decisión

En total, se han realizado tres ejecuciones para la obtención de cada árbol, bajo diferentes semillas:

- Árboles 1: generados bajo `random_state=22`.
- Árboles 2: generados bajo `random_state=7`.
- Árboles 3: generados bajo `random_state=722`.

En caso de necesitarse consultar material adicional a la memoria, los nombres de las imágenes adjuntas (en la carpeta Resultados) son sencillos de interpretar.

### Matrices de confusión por categoría

A continuación, se muestran dos de las matrices de confusión obtenidas para los tres árboles de cada categoría.

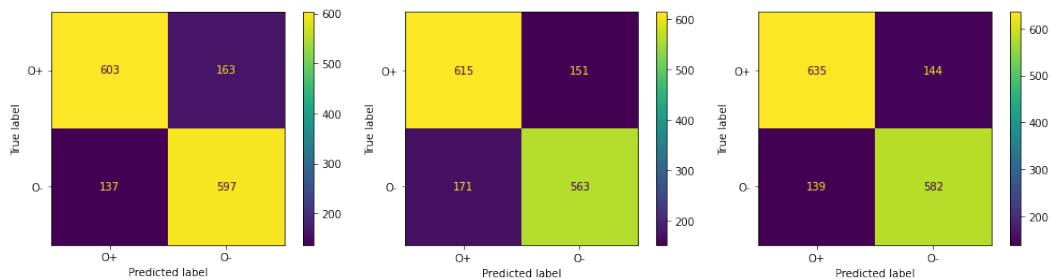


Figura 15: matrices de confusión encontradas para *openness*.

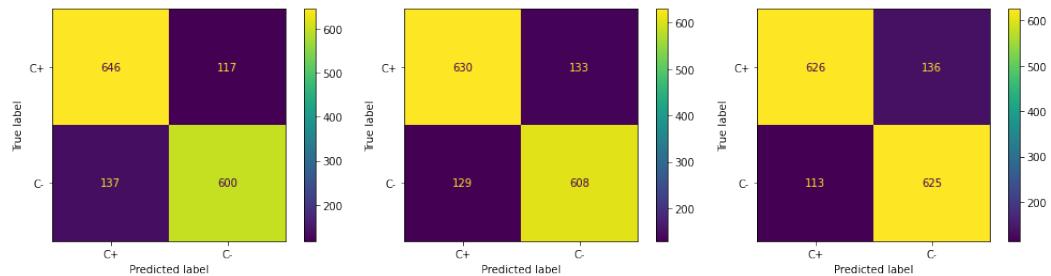


Figura 16: matrices de confusión encontradas para *conscientiousness*.

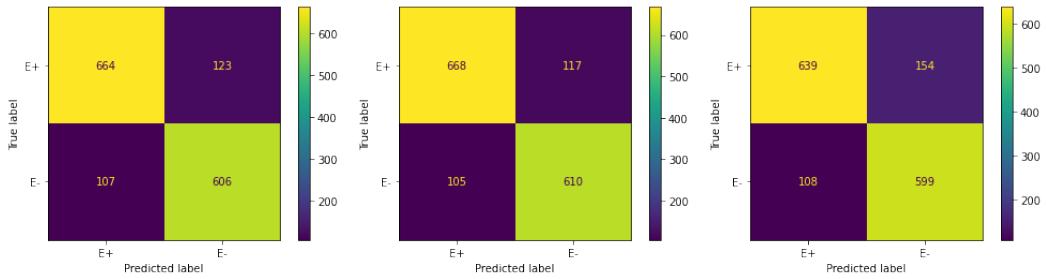


Figura 17: matrices de confusión encontradas para *extraversion*.

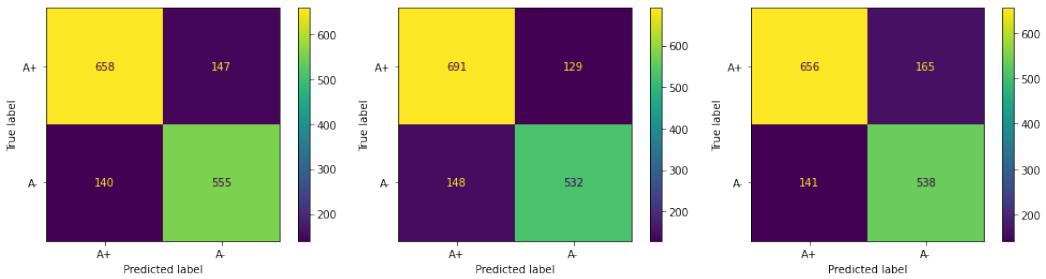


Figura 18: matrices de confusión encontradas para *agreeableness*.

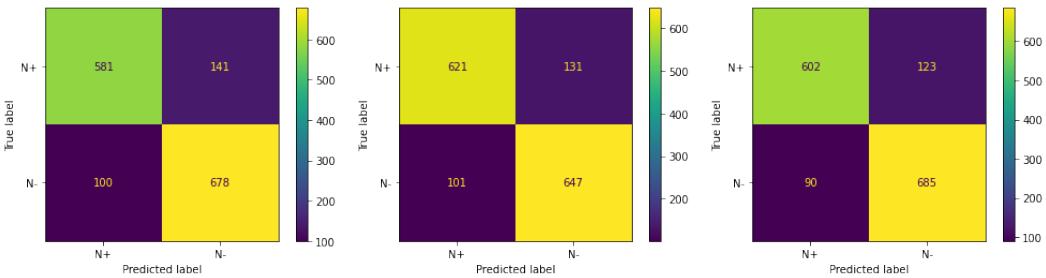


Figura 19: matrices de confusión encontradas para *neuroticism*.

## Árboles de decisión

La gran mayoría de árboles de decisión son en exceso grandes. Aunque se puede limitar su profundidad en la representación gráfica, siguen apareciendo con los nodos en exceso separados. Por tanto, se adjuntan al proyecto a parte de la memoria.

Mostraremos como ejemplo sólo aquellos que sean legibles en papel.

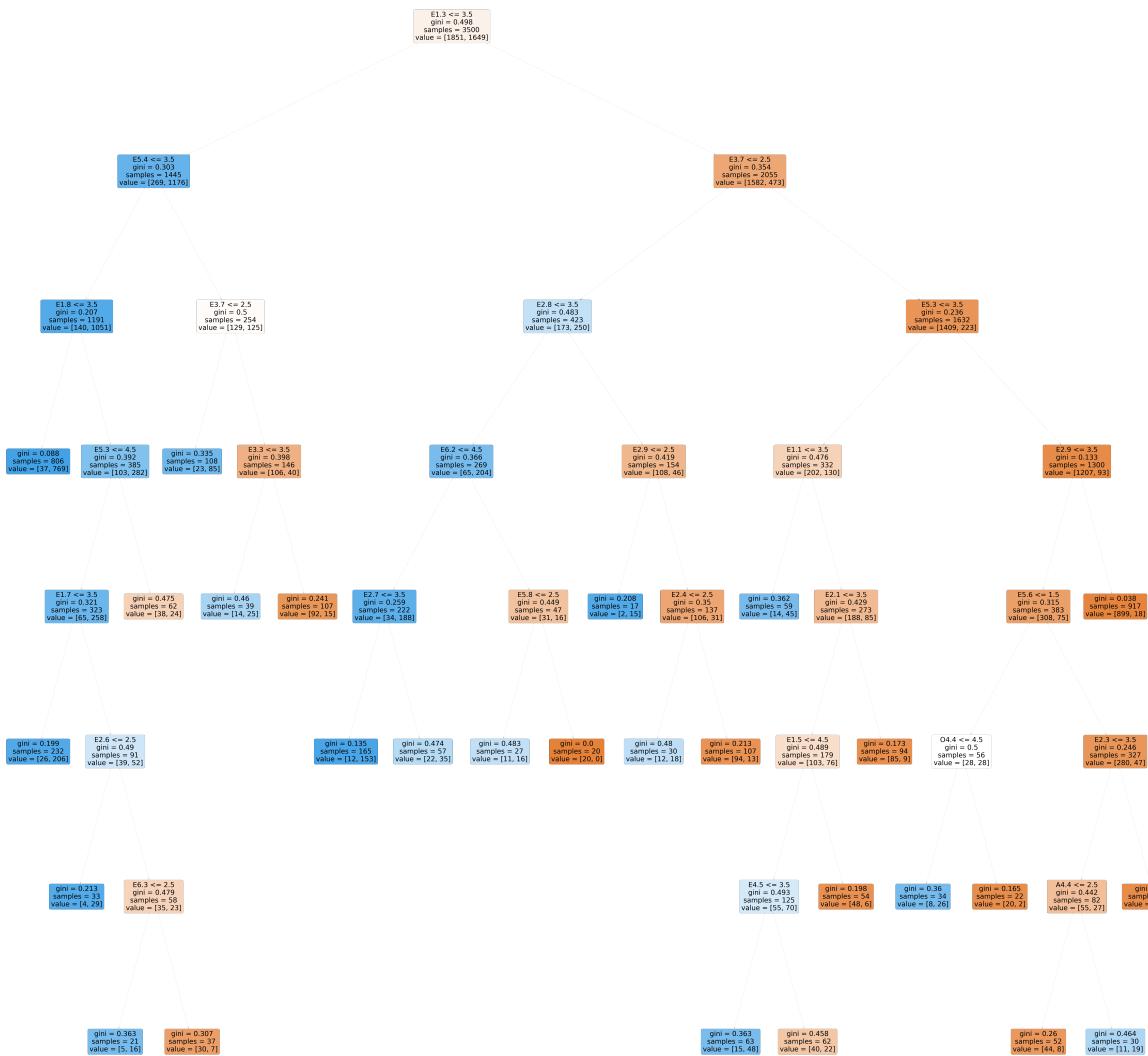


Figura 20: árbol de decisión más ligero encontrado para *extraversion*.

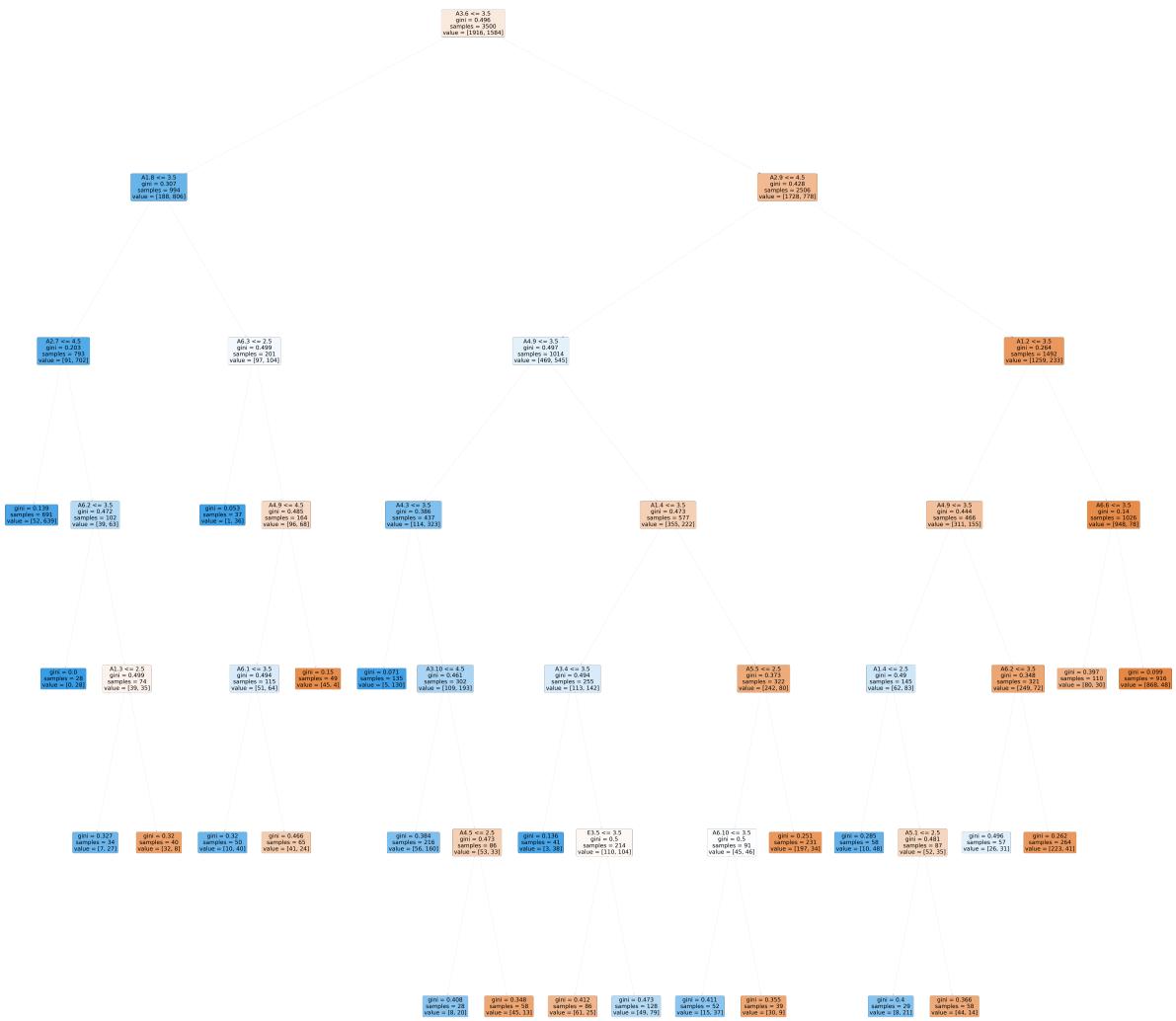


Figura 21: árbol de decisión más ligero encontrado para *agreeableness*.

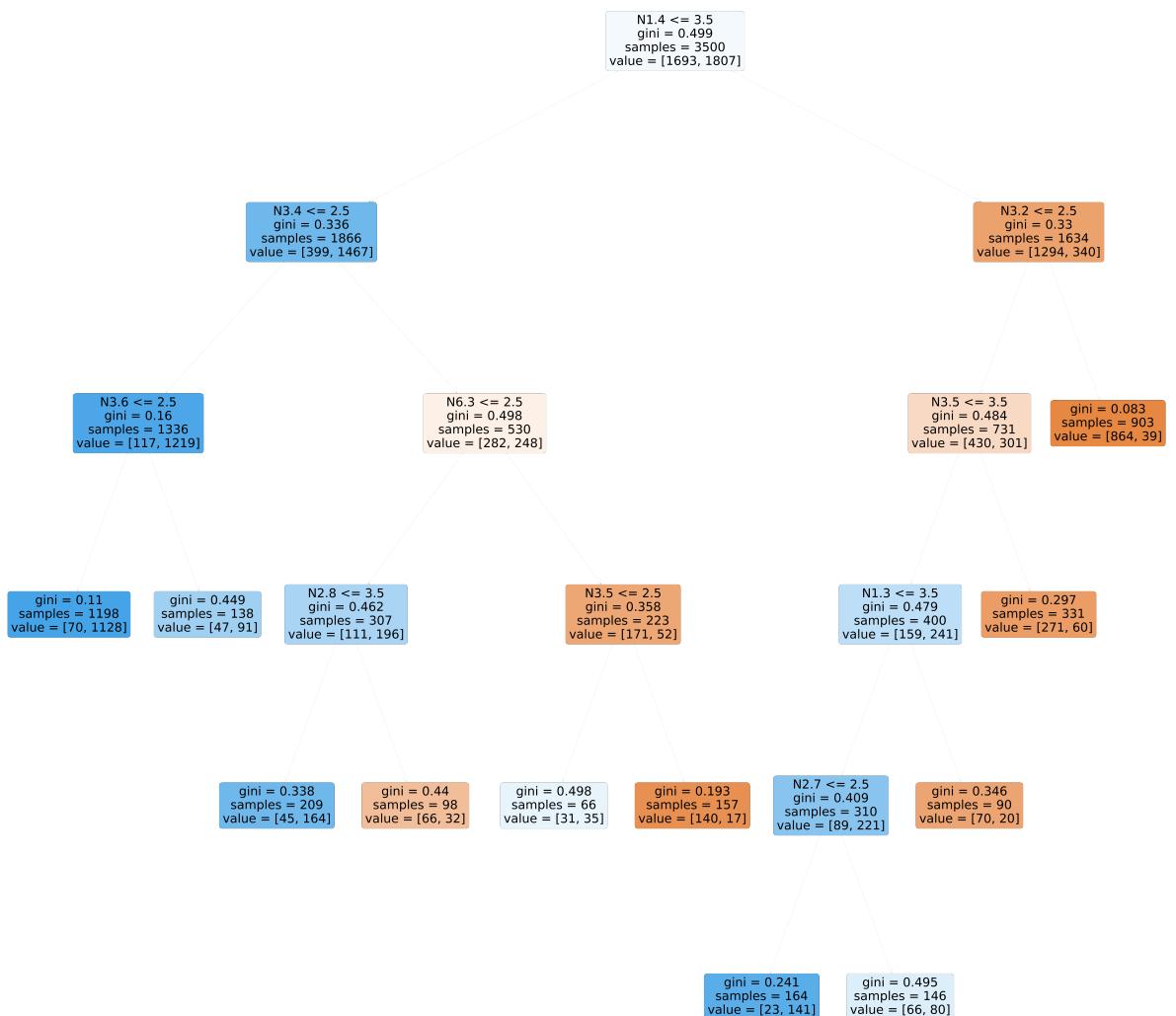


Figura 22: árbol de decisión más ligero encontrado para *neuroticism*.