uc3m | Universidad Carlos III de Madrid

Grado en Estadística y Empresa
(2016-2020)

Trabajo Fin de Grado

Procedimiento para estimar la intención de voto en España mediante métodos tipo ensemble

---

Juan Miguel Rodríguez Lago

Tutor

Andrés M. Alonso Fernández

Universidad Carlos III de Madrid

23/06/2020

RESUMEN

En una época de tenso clima político el uso de las encuestas electorales se ha convertido en una figura primordial tanto para partidos políticos como para los medios de comunicación. El tratamiento e imputación de los datos faltantes mediante técnicas estadísticas resulta estrictamente necesario para lograr obtener estimaciones de voto fiables. En este estudio se aborda el tema desde una perspectiva del aprendizaje automático. Haciendo uso de los datos del barómetro preelectoral 3263 que publicó el Centro de Investigaciones Sociológicas, se implementan modelos de tipo ensemble con el objeto de imputar los datos de aquellas personas entrevistadas que no respondieron a la pregunta objetivo del estudio, su intención de voto en las elecciones generales de noviembre de 2019.

Tras realizar predicciones con los modelos generados se obtienen estimaciones fiables que se comparan con los resultados de las elecciones y se observa que estos modelos proporcionan una alternativa interesante a las técnicas usadas tradicionalmente para abordar este tipo de problemas.

Palabras Clave: CIS, ensembles, imputación datos faltantes, intención de voto en España, *machine learning*.

DEDICATORIA

A mis queridos padres, Juan José y Carmen, por su amor incondicional, por el sacrificio y el esfuerzo que han dedicado para enseñarme el camino correcto, ayudándome a recorrerlo sin dudar de mí en ningún instante. A mi hermano Borja y mi cuñada Aroa, por confiar en mí y ser un apoyo constante en los momentos más complicados.
A Vanesa, por su alegría y vitalidad, por enseñarme a hacer frente a la adversidad.
A Azahara, por su cariño, por acompañarme en todo momento, por levantarme en cada tropiezo y por su paciencia infinita.

A mis familiares Rafael, Antonio y Juan Antonio, fallecidos durante el transcurso de la carrera. A Loli y Lauri, fallecidos recientemente víctimas de la pandemia ocasionada por el virus SARS-CoV-2.

A todos mis profesores, por haberme enseñado tanto, por su dedicación y ayuda a lo largo de estos cuatro años.

Muy en especial, quiero agradecer a mi profesor y tutor Andrés Modesto Alonso Fernández la inestimable ayuda que me ha brindado a lo largo de estos meses, tanto de manera presencial como sobre todo durante los meses de confinamiento a través del correo electrónico, resolviendo dudas a cualquier hora del día sin descanso.

ÍNDICE

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

1.- Introducción.

La política en España experimenta una época agitada, en un escenario donde la sociedad española se ha visto en la necesidad de acudir dos veces a las urnas en el año 2019. En medio de esta vorágine de enfrentamiento político una figura se ha alzado como determinante y de gran interés, tanto para los propios partidos políticos, como para los medios de comunicación y la sociedad en general. Se trata de las encuestas electorales. Existen en la actualidad una extensa red de casas demoscópicas, periódicos y organismos tanto privados como públicos, encargados de realizar encuestas electorales con el fin último de determinar la intención de voto del electorado en las próximas elecciones. Una de las mayores macroencuestas que se realizan en España es la del Centro de Investigaciones Sociológicas, más conocido por su acrónimo, CIS.

Es bastante común hoy en día ver como los medios de comunicación de todo tipo, radio, prensa y televisión se hacen eco de los resultados de las encuestas del CIS. La polémica suscitada a raíz del barómetro 3263 presentado antes de las elecciones generales de noviembre del año 2019, el cuál obtuvo unos resultados muy dispares con lo sucedido posteriormente en las urnas, llegó a ocasionar duras críticas (Okdiario, 2019) [1], tanto a la organización como a su director José Félix Tezanos, llegando a pedir la dimisión del director en algunos casos (El país, 2019) [2]. Las críticas se endurecieron más si cabe debido a que el resto de los sondeos pronosticaban unos resultados que se asemejaron bastante a lo sucedido *a posteriori* en las elecciones generales. La defensa del director del CIS fue atribuir los malos resultados a la sentencia del "procés" y a la exhumación de Franco (El país, 2019) [2], hechos acaecidos con posterioridad al barómetro del CIS y con anterioridad a las elecciones de noviembre.

Muchas voces críticas achacan al CIS tanto su metodología como el denominado uso de la "cocina" de las encuestas con el objetivo de influenciar a la sociedad española en pro de un determinado partido político y en detrimento del resto de partidos del lado opuesto del espectro político (El mundo, 2019) [3]. Esta "cocina" de la encuesta lo que intenta es realizar una corrección de los problemas que surgen en la estimación, tales como la considerable proporción de no respuesta o el sesgo hacia la izquierda producido por la propia composición de las muestras, más fáciles de conseguir en hogares de bajos ingresos (Díaz de Rada y Núñez, 2009) [4]. Por estos motivos se pueden producir desajustes, tales como resultados que reflejen la opinión de momentos anteriores o el sesgo producido por la técnica empleada en la encuesta. Este desajuste en los resultados es lo que intenta corregir la "cocina", usando correctores de estimación de voto, como usar el cociente entre el voto real de las últimas elecciones y el voto recordado de la encuesta anterior, que hace que se le dé más peso a unos entrevistados que a otros en función de su recuerdo de voto, o usar la suma de las variables intención de voto y simpatía por un partido político (Escobar, et al. 2014) [5].

Más allá de la polémica que susciten las técnicas empleadas o el tratamiento que se dé a los datos, existe un problema inherente a la encuesta del CIS y al resto de encuestas, que no es otro que el problema de los datos faltantes.

En el barómetro preelectoral presentado por el CIS antes de las elecciones de noviembre se puede observar como la variable de mayor interés, la intención de voto contiene unos resultados para valores como “no sabe” y “no contesta” de alrededor de un 32% del total. A eso hay que sumarle la cantidad de valores faltantes del resto de variables. Surge la necesidad de imputar esos valores en pos de obtener una mejor estimación de los pronósticos electorales.

Este trabajo se centra en este último aspecto. Para realizar la imputación de los datos de la variable de interés, intención de voto, se usarán técnicas de aprendizaje automático, más conocido por su nombre en inglés *machine learning.* Concretamente, se utilizarán técnicas de aprendizaje supervisado conocidas como ensembles. Para la implementación de los modelos y el tratamiento de los datos se usará el software R, cuyo código completo aparecerá en los anexos finales.

Con el uso de esas técnicas aplicadas al tratamiento de datos se intentará obtener predicciones fiables usando modelos generados a partir de la muestra de la encuesta del CIS para posteriormente, comparar a posteriori los resultados electorales de las elecciones generales de noviembre de 2019, así como con los resultados de las estimaciones del barómetro 3263 del CIS (CIS, 2019) [6].

2 - Pre-proceso de datos.

Con objeto de facilitar el tratamiento de los datos para su posterior análisis, resulta necesario realizar una serie de operaciones con los datos originales de la encuesta del CIS. Primero se realiza una selección de las variables más influyentes para cumplir con el objetivo del trabajo. Una vez realizada la selección, se procede a realizar una recodificación sobre dichas variables, con el fin de facilitar una mejor comprensión de los resultados obtenidos. Resultará necesario realizar transformaciones del tipo de variable a lo largo del trabajo con el mismo objetivo.

2.1 – Selección de variables.

Tras realizar la descarga de los datos originales de la encuesta del CIS, se observa que el número de atributos o variables que aparecen en el estudio es de 146, lo cual resulta demasiado elevado incluso para modelos flexibles como son los procedimientos de tipo ensemble que se utilizarán posteriormente para predecir la intención de voto. Por ello, resulta necesario realizar una selección de los atributos más importantes para ese objetivo.

Las preguntas del cuestionario del CIS se pueden clasificar en cuatro temáticas generales: preguntas electorales, ideológicas, evaluadoras y sociodemográficas (Escobar, et al., 2014) [5].

- Preguntas electorales: son las referidas al comportamiento del votante potencial tanto en pasadas como en futuras elecciones.
- Preguntas ideológicas: aquellas que se refieren a las creencias de las personas entrevistadas en relación con la política.
- Preguntas evaluadoras: aquellas que pretender conseguir la valoración de situaciones actuales por parte de las personas entrevistadas.
- Preguntas sociodemográficas: aquellas variables de tipo sociodemográfico que sirven para calibrar la muestra.

Dentro de cada una de estas temáticas se van a elegir las más relevantes para el objetivo del trabajo. Siguiendo las indicaciones de Escobar et al., (2014), se determinan que las siguientes variables son las más influyentes y son las finalmente escogidas (**Anexo 1**).

- P1: Grado de interés por la política.
- P3: Valoración de la situación política general de España.
- P4: Valoración de la situación económica general de España.
- P9 (Intención de voto): Intención de voto en las elecciones generales de noviembre de 2019.
- P9A: Partido político por el que se siente más simpatía en las elecciones generales de noviembre de 2019.

- P9B: Intención de voto alternativo en las elecciones generales de noviembre de 2019.
- P11: Escala de probabilidad (0-10) de votar en las elecciones generales de noviembre de 2019
- P13: Partido político que considera más cercano a sus ideas.
- P14: Partido político que se cree va a ganar las elecciones generales de noviembre de 2019.
- P15: Partido político que desearía ganador en las elecciones generales de noviembre de 2019.
- P16: Preferencia personal como presidente del Gobierno.
- P17: Participación electoral en las elecciones generales de abril de 2019.
- P17A: Recuerdo de voto en las elecciones generales de abril de 2019 de los votantes
- P18: Escala de probabilidad de clasificación ideológica 1 izquierda a 10 derecha.
- P21: Autodefinición de su ideología política.
- P23: Sexo de la persona entrevistada.
- P24: Edad de la persona entrevistada.
- P25: Escolarización de la persona entrevistada.
- P25A: Nivel de estudios alcanzado por la persona entrevistada.
- P26: Situación laboral de la persona entrevistada.
- P27: Religiosidad de la persona entrevistada.
- P28: Clase social subjetiva de la persona entrevistada.
- VOTSIMG: Voto+simpatía en las elecciones generales de noviembre de 2019.
- P17AR: Recuerdo de voto en las elecciones generales de abril de 2019 de los votantes.
- RECUERDO: Recuerdo de voto en las elecciones generales de abril de 2019.

Las variables P17AR y RECUERDO se diferencian en la inclusión del partido recientemente creado Más País como independiente de Podemos en la variable P17AR, mientras que en RECUERDO esos votos aparecen incluidos en Podemos.

2.2 – Recodificación.

Con la finalidad de simplificar los resultados y obtener una mejor comprensión del análisis se procede a realizar una recodificación de las variables seleccionadas en el apartado anterior. Se usa el software R para la codificación (**Anexo** 2).

Las variables pertenecientes a la categoría de preguntas electorales se recodifican en función de ocho clases, de las cuales las cinco primeras pertenecen a los partidos mayoritarios, 1 para PP, 2 para PSOE, 3 para UP (Unidas Podemos es la confluencia de Izquierda Unida IU, EQUO, En Comú Podem y Podemos), 4 para Ciudadanos y 5 para VOX, la clase 6 se deja para el resto de los otros partidos, la clase 7 para abstención y 99 para no sabe/no contesta.

En las variables pertenecientes a la categoría de preguntas evaluadoras como situación económica y política se procede a una recodificación de escala, que toma valores como "muy buena", "buena", "regular". "mala" y "muy mala", mientras que en las preguntas de tipo ideológico se procede a recodificar en determinadas clases de ideología dentro del espectro político, tales como "derechas", "izquierdas", "centro", "comunista", "nacionalista", "apolítico" y "otras".

Para las preguntas sociodemográficas se usa una recodificación de las respuestas que facilita la comprensión. La variable edad se recodifica en intervalos de edad de 18-25 años, 25-35años, 35-45 años, 45-55 años, 55-65 años, 65-75 años y mayores de 75 años. mientras que en la pregunta de situación laboral se toman los valores "trabaja", "pensionista", "parado", "trabajo doméstico sin remunerar" y "otra situación laboral". En la variable de estudios alcanzados se toman los valores "primarios", "secundarios/FP" y "superiores".

Toda la recodificación puede verse en el **Anexo 2**.

2.3 – Transformación de variables.

La mayoría de las variables seleccionadas son de tipo carácter a excepción de algunas numéricas. A lo largo del trabajo resulta necesario su transformación a otro tipo de variables. Para la construcción de las muestras se usa el tipo numérico, con el fin de implementar los patrones de valores faltantes en la muestra de entrenamiento y test (**Anexo 5**). Posteriormente, para la construcción de los modelos es necesaria su transformación a factor (**Anexo 5**). Para realizar dichas transformaciones se usan sentencias sencillas de R que pueden visualizarse en los anexos indicados.

3.- Estadística descriptiva.

A continuación, se procede a realizar un análisis descriptivo utilizando las variables seleccionadas, la recodificación y transformación de tipo de variables utilizada en los apartados anteriores. Dicho análisis consta de un análisis exploratorio de datos (EDA por sus siglas en inglés), posteriormente se realiza un análisis univariante de las variables seleccionadas y para finalizar, se procede con un análisis bivariante de las mismas.

3.1 - Análisis Exploratorio de datos. EDA (*Exploratory Data Analysis*).

Para realizar este análisis inicial de los datos seleccionados se carga la librería *DataExplorer* de R (**Anexo 1**). Con la función `create_report()` de dicho paquete se crea directamente un archivo .html que contiene un EDA de los datos. A continuación, se muestran los resultados más significativos del análisis.

Inicialmente, se puede observar en la ilustración que se muestra a continuación el número de filas, 17650, y el número de columnas, 25. Existen 14 columnas de tipo discreto y 11 de tipo continuo. Podemos observar la cantidad de datos faltantes en la muestra, que ascienden a un total de 52723 de un total de 441250 observaciones.

Basic Statistics

Raw Counts

Name	Value
Rows	17,650
Columns	25
Discrete columns	14
Continuous columns	11
All missing columns	0
Missing observations	52,723
Complete Rows	0
Total observations	441,250
Memory allocation	3.4 Mb

Ilustración 1 . Estadísticas básicas

A continuación, se puede observar el tipo de atributos que se han seleccionado. Se trata de un *data frame* con 25 atributos, todos ellos de tipo carácter o numérico.

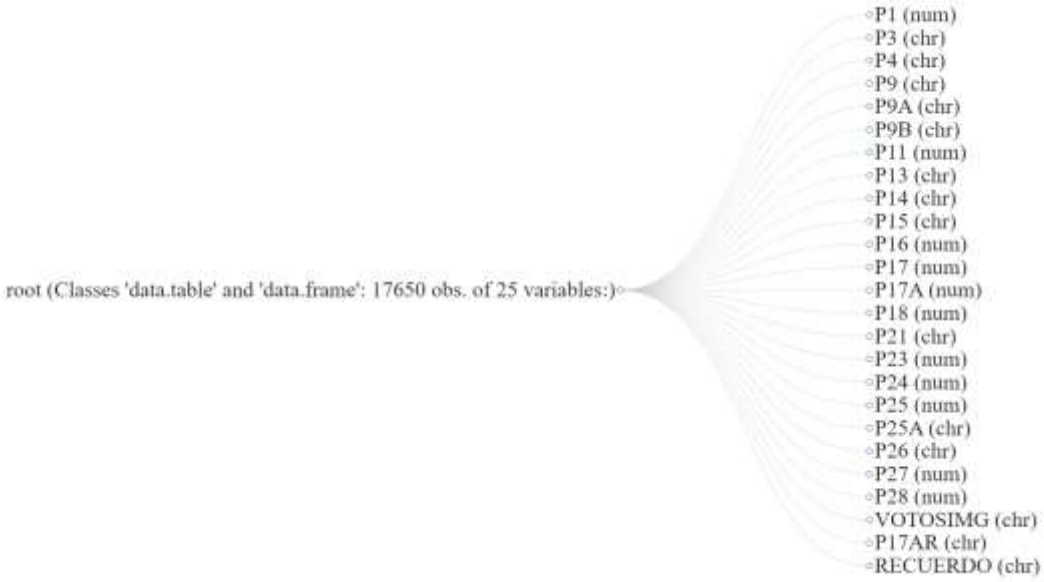


Ilustración 2. Tipo de variables escogidas

Debemos conocer el número de valores faltantes existentes en las variables seleccionadas. En el siguiente gráfico se pueden observar que las variables con mayor número de valores faltantes, que son P9A y P9B correspondientes a simpatía por partido político e intención de voto alternativo respectivamente. Lo más significativo es que la variable de interés del estudio P9 (Intención de voto) tiene un 32,59% de NA´s.

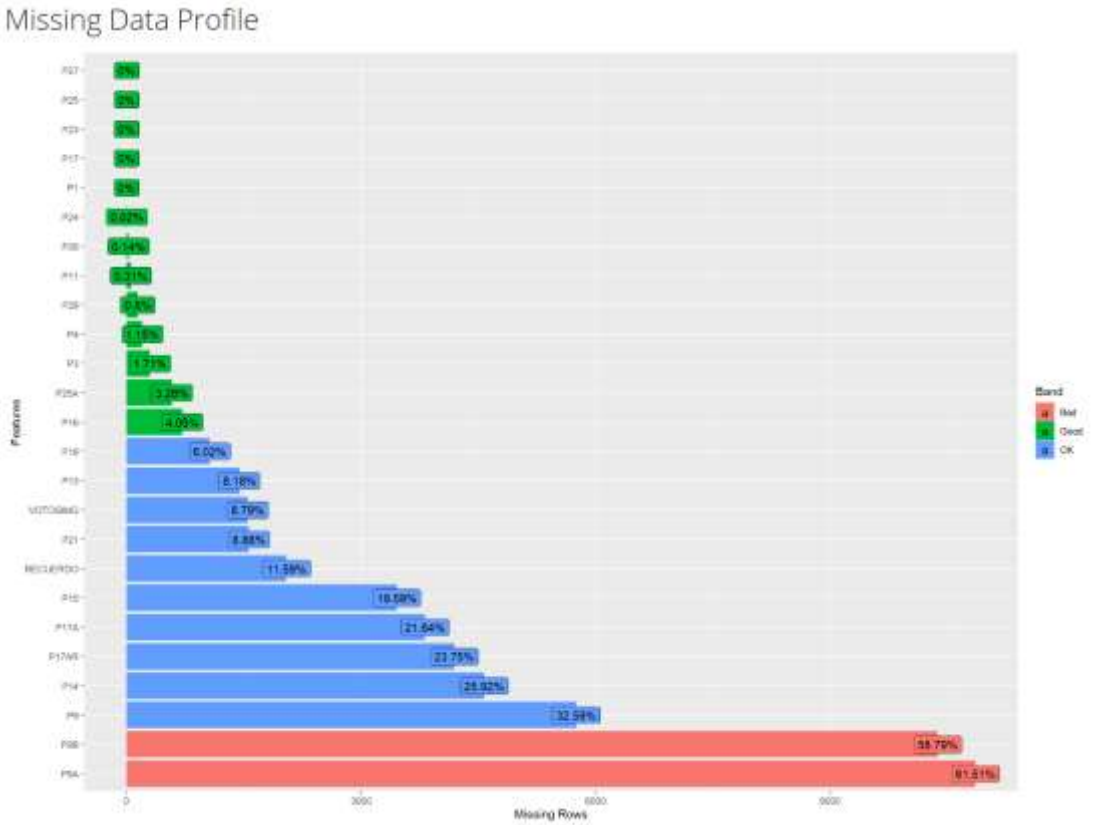


Ilustración 3 . Valores faltantes en los atributos.

3.2 - Análisis descriptivo univariante.

Se realiza un análisis descriptivo de las variables más influyentes en el estudio del CIS. Para ello, se crearán tablas con la frecuencia absoluta aparecida en la muestra y la proporción en tanto por ciento sobre el total observado. Para una mejor comprensión de los resultados obtenidos, se realizan gráficos de barras de cada variable.

- **Variable P9. Intención de voto para las elecciones de 2019.**

Esta pregunta es la variable de interés del estudio y por tanto la más importante. Con el objetivo de obtener una mayor claridad en el análisis se realiza una recodificación de las categorías de las respuestas (**Anexo 2**). En la recodificación se incluyen los partidos mayoritarios, tales como son PP, PSOE, Unidas Podemos (confluencia de partidos en las que incluyen IU, Podemos, EQUO y en Comú Podem), Ciudadanos y Vox como categorías independientes y en la categoría "Otros" se incluyen el resto de los partidos no mayoritarios. La categoría abstención incluye aquellos encuestados que no tienen intención de ir a votar y aquellos que han respondido como "voto en blanco" y "voto nulo" a la pregunta. Finalmente, en la categoría NS/NC se incluyen aquellos encuestados que han respondido a la pregunta como "no sabe", "no contesta" y "no lo tiene decidido aún".

Se realiza una tabla de frecuencias con el objeto de analizar los resultados de la pregunta en la encuesta (**Anexo 3**).

	PP	***PSOE***	***UP***	***Ciudadanos***	***VOX***	***Otros***	***Abstención***	***NS/NC***
Frecuencia absoluta	1841	3574	1201	716	726	1504	2336	5752
Proporción (%)	10.43	20.25	6.80	4.05	4.11	8.52	13.23	32.59

Tabla 1. Intención de voto (Variable P9).

Se observa en la Tabla 1 que el partido con mayor proporción en intención de voto para las elecciones generales del 2019 es el PSOE con un 20,25% con respecto al total, seguido del PP con un 10,43%, Unidas Podemos con un 6,80%, VOX con un 4,11% y Ciudadanos con un 4,05%. El resto de los partidos no mayoritarios obtienen un 8,52% y el porcentaje de abstención es del 13,23% en la encuesta. Cabe destacar el amplio porcentaje de encuestados que o bien no contestan a la pregunta o responden que no lo saben o que no lo tienen decidido. Éstos últimos se corresponden con un 32,59% del total, lo que quiere decir que más del 30% de las respuestas de la variable de interés del estudio son considerados como datos faltantes a imputar.

Este hecho supone un problema a la hora de estimar la intención de voto en España para las elecciones generales y es por ello que resulta necesario imputar dichos valores faltantes con ese objetivo.

Se realiza un diagrama de barras donde se pueden observar los resultados, con las proporciones en tanto por ciento de la tabla anterior, con el fin de facilitar la comprensión de las respuestas dadas a la pregunta. Se observa a simple vista como los datos faltantes de la categoría NS/NC obtienen la mayor proporción en esta pregunta.

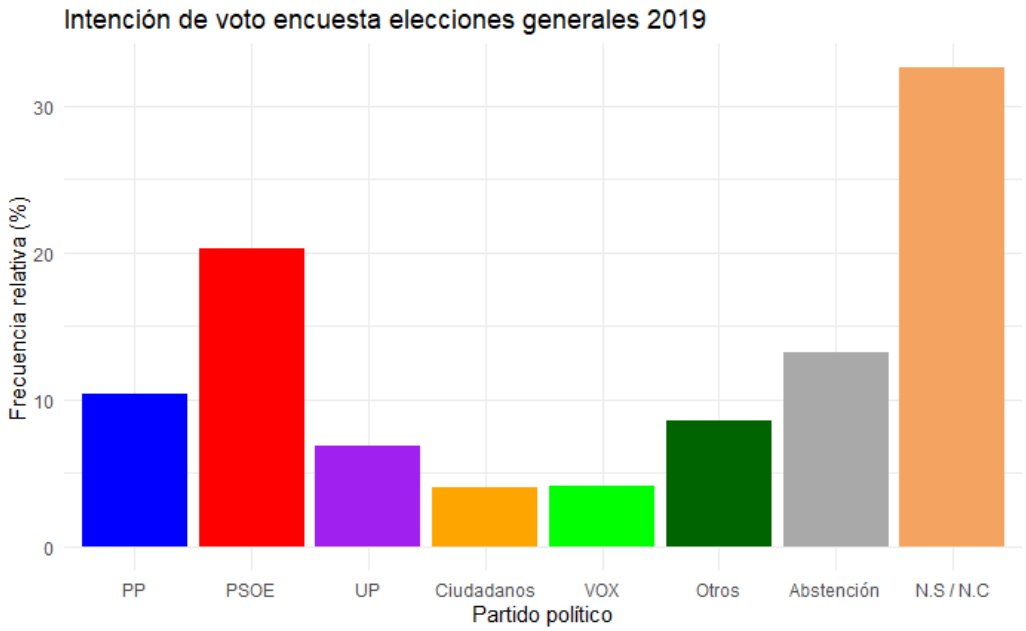


Ilustración 4. Intención de voto encuesta elecciones generales 2019

- **Variable P21. Clasificación ideológica.**

Esta pregunta tiene como objetivo determinar la ideología política de los encuestados. Se realiza una recodificación de las categorías originales (**Anexo 2**) con el fin de facilitar la comprensión de las respuestas. Se determinan siete clases de ideología, las cuales clasifican la ideología de los encuestados en derechas, centro, izquierdas, comunista, nacionalista, otras (entre las que se incluyen feminista, ecologista y otras) y apolítico, la cual no se lee en el momento de la entrevista con el fin de evitar sesgos.

Se realiza una tabla de frecuencias para representar los resultados de esta variable (**Anexo 3**).

	Derechas	***Centro***	***Izquierdas***	***Comunista***	***Nacionalista***	***Otras***	***Apolítico***
Frecuencia absoluta	3025	1868	5854	289	528	4518	1568
Proporción (%)	17.14	10.60	33.16	1.63	2.99	25.60	8.88

Tabla 2. Clasificación ideológica (Variable P21).

Se realiza un diagrama de barras para visualizar los resultados obtenidos tras la recodificación y cálculo de los porcentajes de las proporciones.

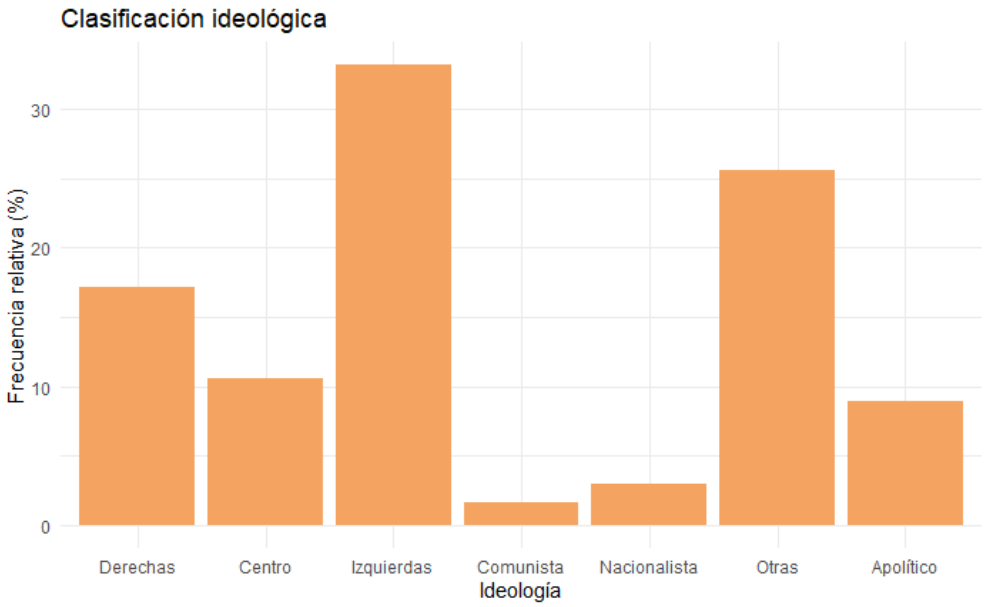


Ilustración 5. Clasificación ideológica.

Se observa que la ideología con mayor proporción entre los encuestados es izquierdas con el 33,11% del total, seguido de otras ideologías con un 25,6 %, derechas y centro con un 17,13% y 10,58% respectivamente. Los encuestados que se consideran apolíticos suponen un 8,88% del total y las ideologías con menor representación en la muestra son nacionalista con un 3% aproximadamente y comunista con un 1,63%.

- **Variable P3. Situación politica.**

Esta pregunta tiene como objetivo determinar la valoración del encuestado de la situación política actual en España. Se realiza una recodificación (**Anexo 2**) en la que se incluyen los niveles de valoración para la situación en una escala de muy buena, buena, regular, mala, muy mala y ns/nc para aquellos que no quieran o no sepan responder a la cuestión.

En la siguiente tabla de frecuencias absolutas se representan los resultados para esta variable (**Anexo 3**).

	Muy buena	***Buena***	***Regular***	***Mala***	***Muy mala***	***NS/NC***
Frecuencia absoluta	36	261	3151	6412	7488	302
Proporción (%)	0.20	1.47	17.85	36.32	42.42	1.71

Tabla 3. Situación política (Variable P3).

Como se puede apreciar, la valoración de los encuestados respecto a la situación política actual en España no es muy favorable ya que, la gran mayoría, un 42,42%, la califica como muy mala y el 36,32% como mala y un 17,85% como regular. Las calificaciones de buena y muy buena solamente obtienen un 1,47% y un 0,2%, respectivamente, mientras que los que no contestan o no saben suponen únicamente el 1,7% de los encuestados.

Se realiza un diagrama de barras donde se puede observar a simple vista la desfavorable valoración y se pone de manifiesto el descontento de los encuestados respecto a la situación política en España en el momento de la realización de la encuesta.

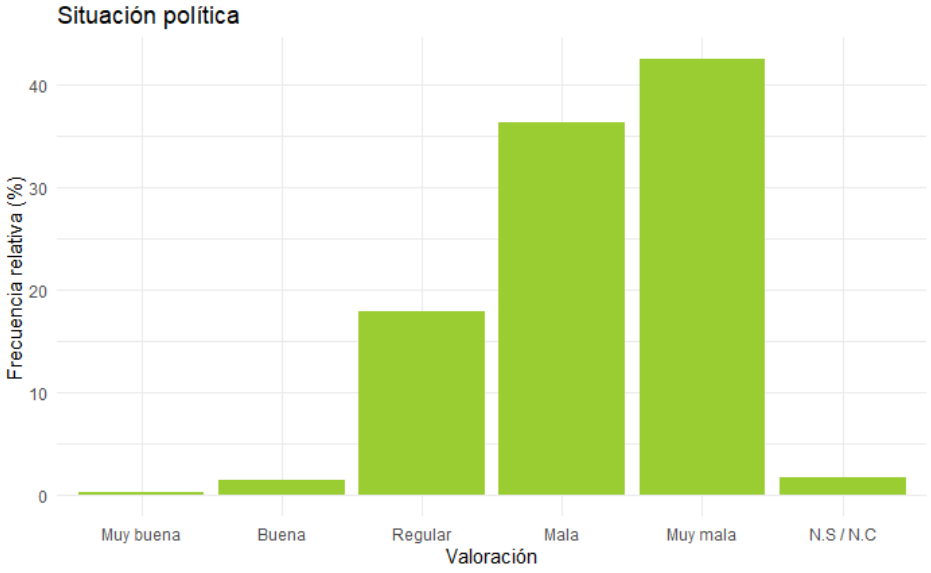


Ilustración 6. Situación política

- **Variable P4. Situación económica.**

Esta pregunta tiene como objeto determinar la valoración subjetiva de los encuestados con respecto a la situación económica del país en el momento de la realización de la encuesta. Se realiza una recodificación (**Anexo 2**) en la que se incluyen los niveles de valoración para la situación en una escala de muy buena, buena, regular, mala, muy mala y NS/NC para aquellos que no quieran o no sepan responder a la cuestión.

En la siguiente tabla de frecuencias se representan los resultados para esta variable (**Anexo 3**).

	Muy buena	***Buena***	***Regular***	***Mala***	***Muy mala***	***NS/NC***
Frecuencia absoluta	24	652	6188	6314	4269	203
Proporción (%)	0.13	3.70	35.06	35.77	24.19	1.15

Tabla 4. Situación económica (Variable P4).

En los resultados de la tabla anterior se pueden apreciar que la valoración de los encuestados con respecto a la situación económica en el momento de la realización de la encuesta es muy desfavorable. Un 35,77%, la califica como mala y el 24,19% como mala y un 35.05% como regular. Las calificaciones de buena y muy buena solamente obtienen un 3,7% y un 0,13% respectivamente, mientras que los que no contestan o no saben suponen únicamente el 1,15% de los encuestados.

Se realiza un diagrama de barras donde se puede observar a simple vista la desfavorable valoración y se pone de manifiesto el descontento de los encuestados respecto a la situación económica en España en el momento de la realización de la encuesta.

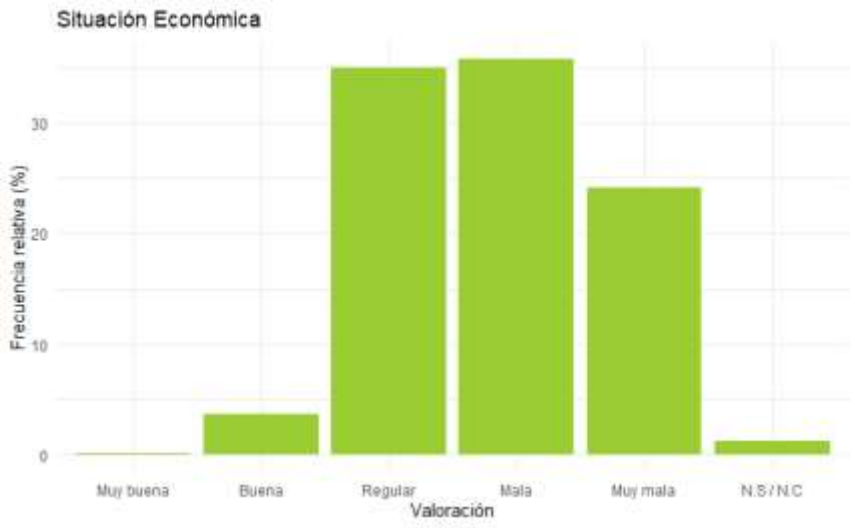


Ilustración 7. Situación económica

- **Variable P24. Edad.**

En esta variable se incluye la edad del encuestado. Se admiten personas mayores de 18 años, edad mínima requerida para poder votar en España, sin que haya límite de edad superior. Vemos la distribución de la variable edad en el siguiente histograma.

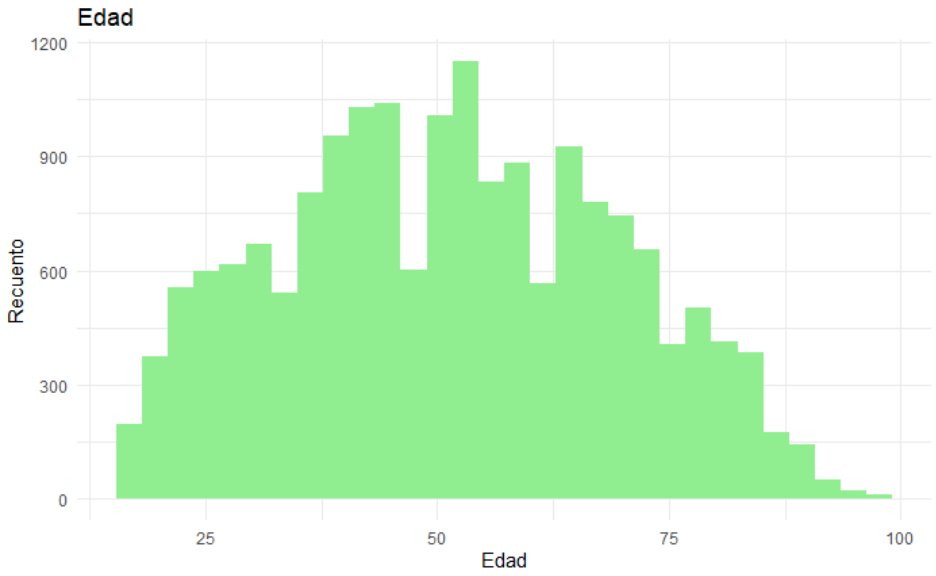


Ilustración 8. Edad .

Con objeto de facilitar la interpretación de la edad de los encuestados se realiza una recodificación (**Anexo** 2), en la que se divide la variable edad en intervalos comprendidos en 18-25 años, 25-35 años, 35- 45 años, 45-55 años, 55-65 años, 65-75 años y >75 años. Vemos en la siguiente tabla las frecuencias absolutas y la proporción en tanto por ciento del total (**Anexo 3**).

	18-25	***25-35***	***35-45***	***45-55***	***55-65***	***65-75***	***>75***
Frecuencia absoluta	1541	2306	3247	3363	2906	2405	1882
Proporción (%)	8.73	13.06	18.40	19.05	16.46	13.62	10.66

Tabla 5. Edad por intervalos (Variable P24).

Se puede observar que la muestra se encuentra bien repartida en todos los intervalos, señal de ser una buena muestra representativa de la población con derecho a voto.

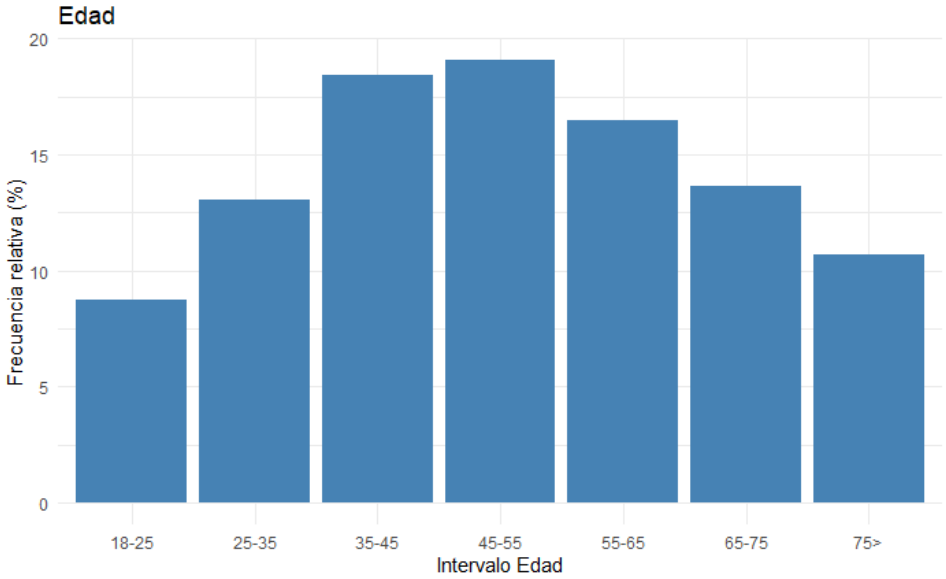


Ilustración 9. Edad por intervalos.

- **Variable P26. Situación laboral.**

En esta pregunta se pretende conocer la situación laboral actual de la persona entrevistada. Es una variable importante a la hora de votar ya que, dependiendo de su situación, el voto de la persona puede verse influenciado. Se realiza una recodificación de las respuestas originales (**Anexo 2**) en el que las categorías a clasificar son trabajador, pensionista, estudiante, trabajo doméstico sin remunerar, otra situación laboral y no sabe/no contesta.

En la siguiente tabla de frecuencias se representan los resultados para esta variable (**Anexo 3**).

	Trabaja	***Pensionista***	***Parado***	***Estudiante***	***Trabajo doméstico***	***Otra situación***	***NS/NC***
Frecuencia absoluta	8264	5082	2350	777	1055	97	25
Proporción (%)	46.82	28.80	13.31	4.40	5.98	0.55	0.14

Tabla 6. Situación laboral (Variable P26).

Se observa que la gran mayoría de las personas entrevistadas se encuentran en la categoría “Trabaja” con un 46,82%, seguido del grupo de los pensionistas con un 28,79%. Los parados constituyen un 13,31 % del total. En menor medida se encuentran los estudiantes con un 4,40%, las personas con trabajo doméstico no remunerado con un 5.97% y personas con otra situación laboral y que no responden con un 0,5% y 0,14% respectivamente. En el siguiente gráfico se muestran los resultados anteriores.

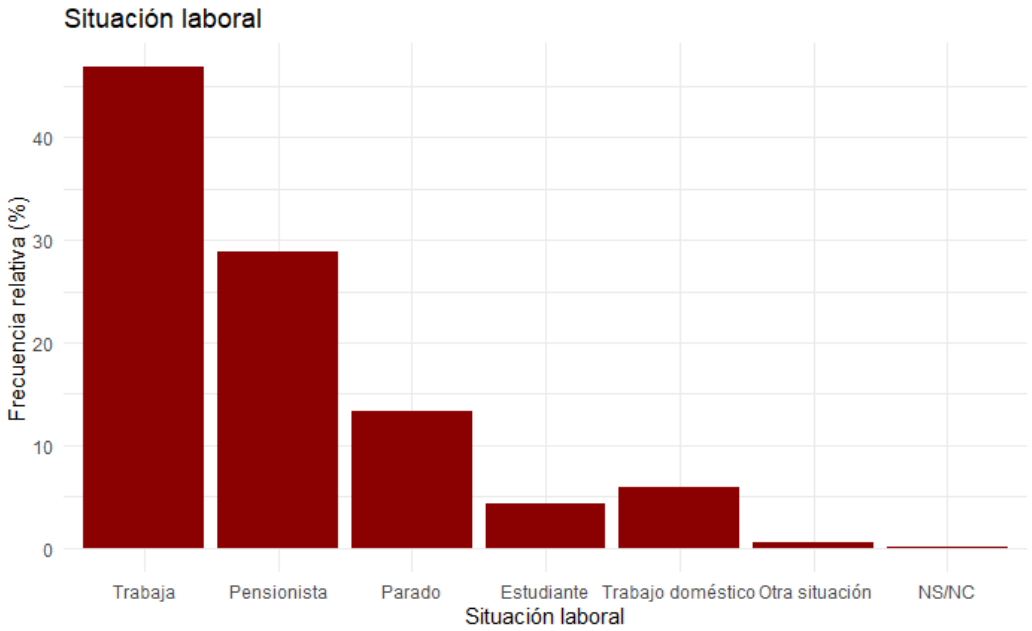


Ilustración 10. Situación laboral.

- **Variable P25A. Nivel de estudios.**

Esta pregunta se interesa por el nivel de estudios alcanzado por la persona entrevistada a lo largo de su vida. Es una variable influyente en el voto ya que, dependiendo del nivel de estudios alcanzado el voto se puede inclinar por una parte del espectro político. Se realiza una recodificación para una mayor claridad del análisis (**Anexo 2**). Los niveles educativos recodificados se clasifican en sin estudios/ primarios, secundarios/ formación profesional, superiores, no sabe. En la siguiente tabla se muestran las frecuencias absolutas de cada nivel de estudios y la proporción en tanto por ciento de las respuestas obtenidas (**Anexo 3**).

	Sin estudios / Primarios	***Secundarios/ FP***	***Superiores***	***No sabe***
Frecuencia absoluta	3401	9717	3953	29
Proporción (%)	19,88	56,82	23,11	0,17

Tabla 7. Nivel de estudios (Variable P25A).

A la vista de los resultados obtenidos se puede apreciar claramente que más de la mitad de las personas encuestadas tienen un nivel educativo de estudios secundarios/FP con un 56,82%, mientras que niveles superiores de estudios son alcanzados por el 23,11% de la muestra y sin estudios/primarios con un 19,88% del total. Las personas que contestaron "no sabe" a la pregunta representan el 0,17% del total. En el siguiente diagrama de barras se muestran los resultados obtenidos.

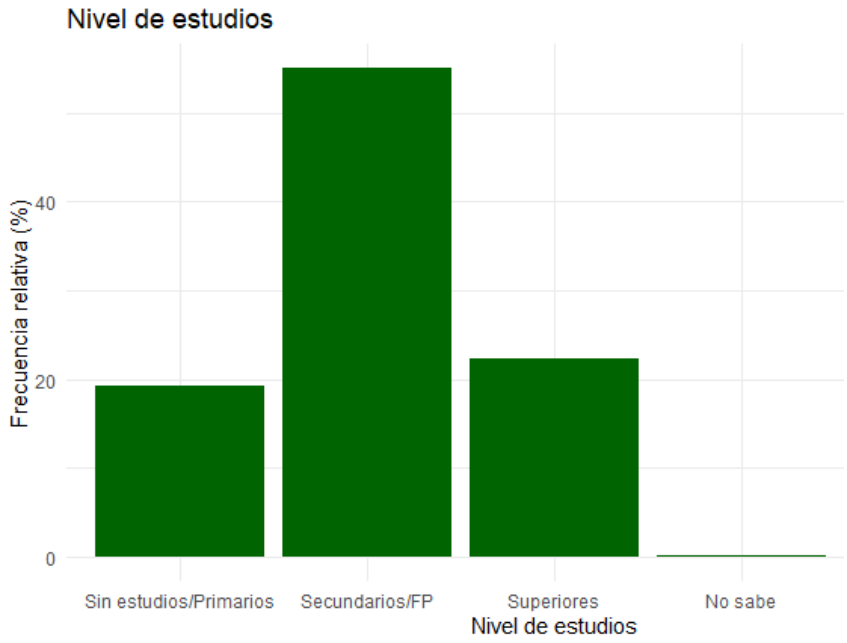


Ilustración 11. Nivel de estudios.

3.3- Análisis descriptivo bivariante.

Se realiza un análisis descriptivo bivariante con el objetivo de conocer las diferencias entre las respuestas en cuanto a clasificación ideológica e intención de voto en función de otras variables como son nivel de estudios, situación laboral y la edad agrupada en intervalos. Se obtiene así una visión más amplia del grado de influencia que tienen estas variables sobre las variables de interés en el estudio. Se realizan diagramas de barras con las frecuencias relativas, desglosados en función de las variables auxiliares. Las tablas de frecuencias y de propociones en tanto por ciento pueden consultarse en el **Anexo 4**.

- **Clasificación ideológica por nivel de estudios.**

Se quiere conocer si existen diferencias significativas en cuanto a la respuesta de los entrevistados a su clasificación ideológica en función del nivel de estudios alcanzado. Se realiza un contraste de independencia chi cuadrado $\chi 2$ para ver si las dos variables son dependientes (**Anexo 4**). Por tanto, se formulan las hipótesis nula (H_0) y alternativa (H_1).

H_0: las dos variables son independientes

H_1: las dos variables son dependentes

Se realiza el contraste en R usando la instrucción `chisq.test()` sobre la tabla de frecuencias absolutas conjunta. El p-valor, entendiéndose éste como la probabilidad de encontrar una mayor discrepancia que la observada en la muestra, tiene un valor de 2.2e-16, que al ser menor que $\alpha = 0.05$, tenemos evidencia estadística suficiente como para rechazar la hipótesis nula H_0. Por tanto, implícitamente se puede inferir que las variables clasificación ideológica y nivel de estudios son dependientes.

Se realiza un diagrama de barras con las coordenadas invertidas, desglosadas las respuestas por nivel de estudios, que se muestra a continuación.

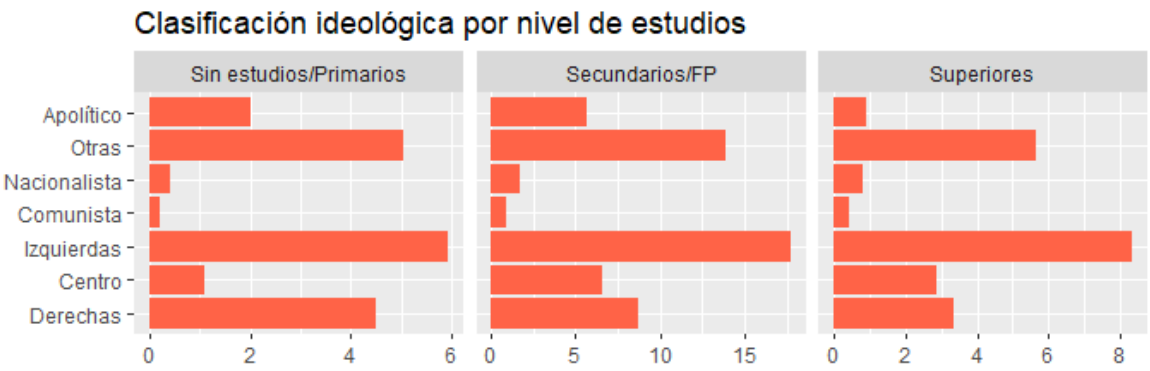


Ilustración 12. Clasificación ideológica por nivel de estudios.

Cabe destacar que a medida que aumenta el nivel de estudios se observa un descenso en las respuestas de clase apólitica, lo cual nos sugiere a simple vista que puede haber una relación inversa en este sentido. Mientras que los resultados de la clase izquierda se mantienen constantes, independientemente del nivel de estudios, los resultados de derechas sufren un ligero descenso a medida que se aumenta el nivel de estudios entre los encuestados. El resto de clases tienen un comportamiento similar en todos los niveles de estudios alcanzados.

- **Clasificación ideológica por intervalos de edad.**

Para comprobar si existen diferencias en la determinación de la ideología politíca en función de la edad por intervalos, se realiza un contraste de independencia chi cuadrado (**Anexo 4**) y se observa que el p-valor es bastante menor que $\alpha = 0.05$, por lo que se determina que ambas variables son dependientes. Seguidamente, se realizan los diagramas que se muestran a contiuación. Se puede observar el grupo de edad y sus respuestas a la identificación dentro del espectro de ideología política.

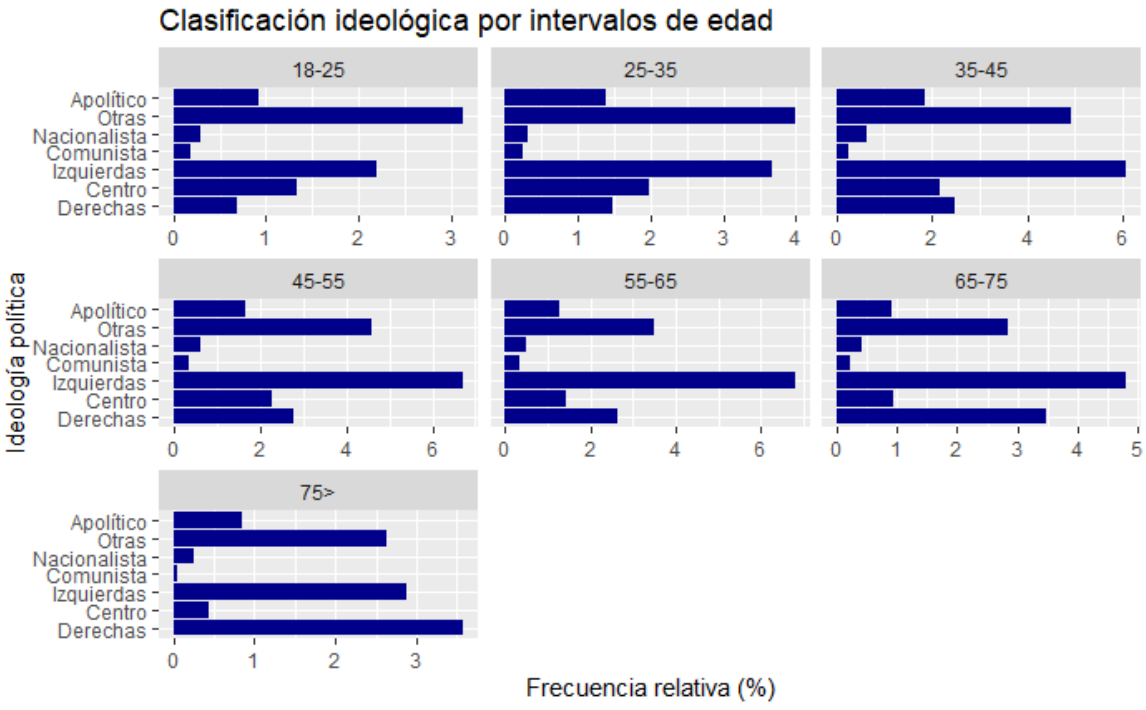


Ilustración 13. Clasificación ideológica por intervalos de edad.

Lo primero que llama la atención es el aumento de la clase de derechas a medida que aumenta la edad, llegando a ser la respuesta elegida mayoritariamente por personas mayores de 75 años. Para las personas que se consideran de centro se observa un descenso en las respuestas a medida que aumenta el intervalo de edad. Las clases de izquierda, comunista y nacionalista tienen un comportamiento similar en cuanto a los intervalos de edad, apenas existen variaciones en las respuestas.

- **Clasificación ideológica por situación laboral.**

Para conocer la clasificación ideológica de los encuestados en función de su situación laboral, para determinar si existen diferencias significativas en las respuestas. Se realiza un contraste de independencia chi cuadrado (**Anexo 4**) y se observa que el p-valor es bastante menor que $\alpha = 0.05$, por lo que se determina que ambas variables son dependientes. Se realiza el diagrama de barras desglosado por la situación laboral de la persona entrevistada.

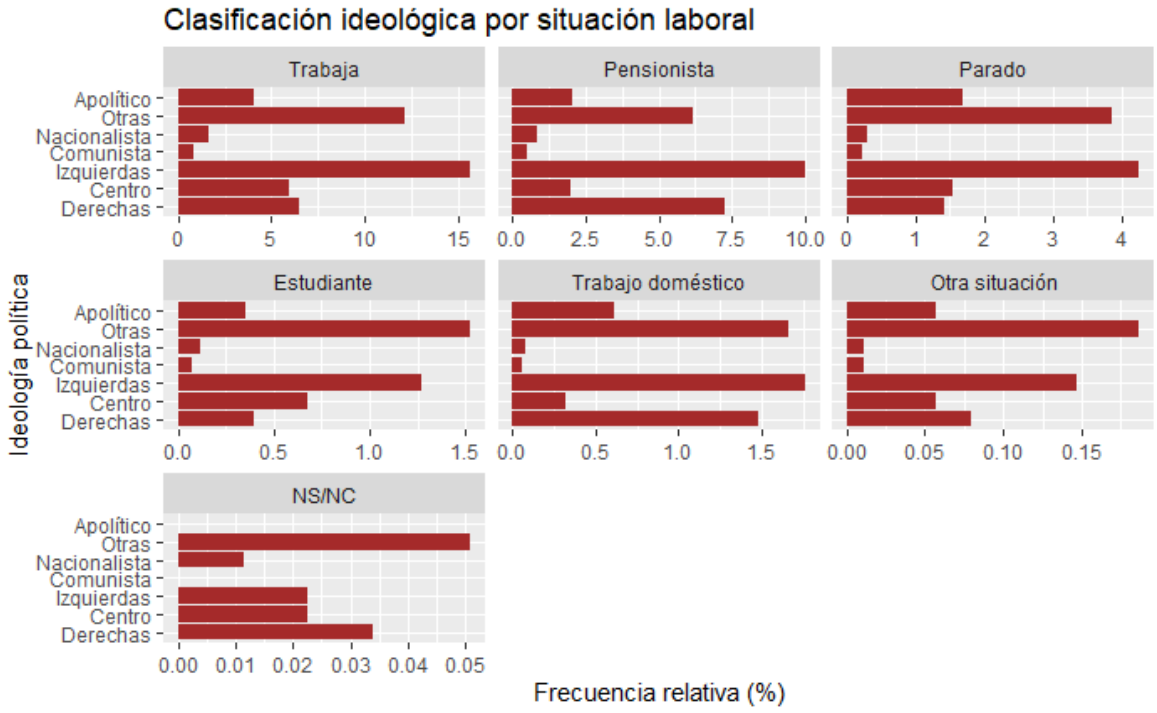


Ilustración 14. Clasificación ideológica por situación laboral.

Se observa que en las personas que trabajan la gran mayoría se identifica con una ideología de izquierdas, seguido de otras ideologías, mientras que centro y derechas alcanzan niveles similares dentro de este grupo. Se observa un comportamiento de respuesta similar en el grupo de parados. En cuanto a los pensionistas se observa un aumento considerable de ideología de derechas, a pesar que la ideología de izquierdas sigue siendo la opción preferida en este grupo. En las personas que realizan trabajo doméstico sin remunerar se puede ver un aumento en la ideología de derechas que llega casi a la par con izquierdas, siendo superada por ésta levemente.

- **Intención de voto por nivel de estudios.**

Se quiere conocer si en la variable de interés del estudio, intención de voto, existen diferencias significativas en las respuestas en función del nivel de estudios alcanzado por la persona entrevistada. Se realiza un contraste de independencia chi cuadrado (**Anexo 4**) y se observa que el p-valor es bastante menor que $\alpha = 0.05$, por lo que se determina que ambas variables son dependientes. Seguidamente, se realiza el diagrama de barras desglosado por nivel de estudios que se muestra a continuación.

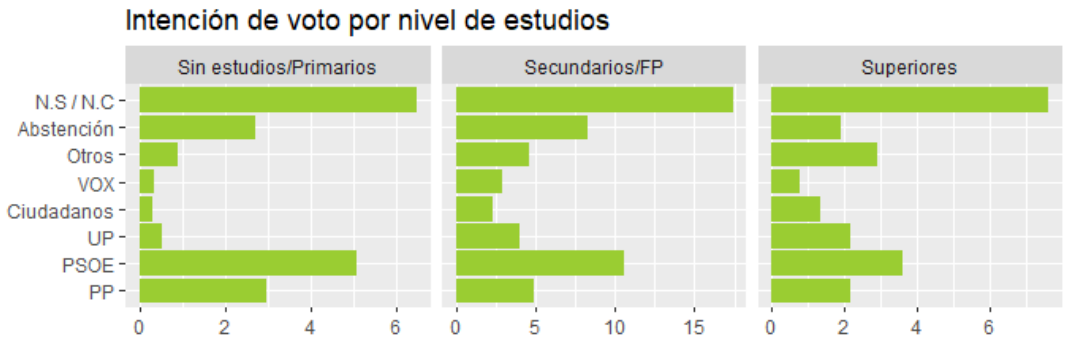


Ilustración 15. Intención de voto por nivel de estudios.

Se observa un nivel de respuestas de no sabe/no contesta similar en todos lo niveles de estudios. En cuanto a la abstención, hay un ligero descenso a media que aumenta el nivel de estudios. El PSOE, a pesar de ser la elección mayoritaria en todos los niveles, experimenta un descenso gradual a medida que se aumenta el nivel de estudios, lo mismo sucede con el PP, que siendo el segundo más elegido por los encuestados experimenta un descenso similar. Este resultado sugiere que a simple vista el bipartidismo sufre una caída de respuestas a medida que aumenta el nivel de estudios. Los partidos como VOX, Ciudadanos y UP experimentan un aumento en las respuestas de las personas entrevistadas a media que aumenta el nivel de estudios alcanzado, por lo que obtienen sus mejores resultados en personas con nivel de estudios superiores.

- **Intención de voto por intervalos de edad.**

Se comprueba si en la intención de voto existen diferencias significativas en las respuestas por intervalos de edad. Se realiza un contraste de independencia chi cuadrado $\chi 2$ para ver si las dos variables son dependientes (**Anexo 4**). Se realiza el contraste en R usando la instrucción `chisq.test()` sobre la tabla de frecuencias absolutas conjunta. El p-valor tiene un valor de 2.2e-16, que al ser menor que $\alpha = 0.05$, tenemos evidencia estadística suficiente como para rechazar la hipótesis nula. De forma implícita, se puede inferir que las variables intención de voto y edad por intervalos son dependientes.

Se muestran los siguientes diagramas de barras desglosados por intervalos de edad.

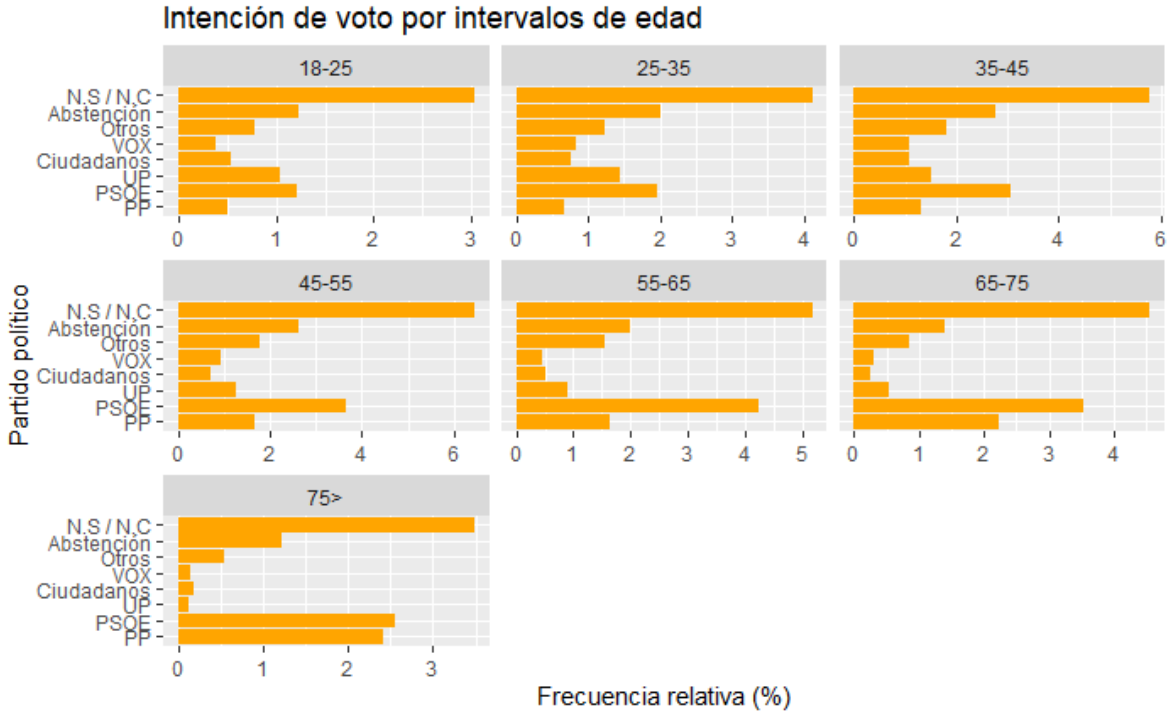


Ilustración 16. Intención de voto por intervalos de edad.

A simple vista se puede observar como el bipartidismo PP-PSOE va experimentando un aumento gradual a medida que aumenta la edad, siendo muy significativo en personas de 65 años en adelante. Ocurre lo contrario con UP que obtiene más respuestas en intervalos de edad correspondientes a jóvenes y va sufriendo un descenso a medida que aumenta la edad. Un comportamiento similiar puede ser observado en las respuestas de intención de voto a VOX y Ciudadanos, los cuales obtienen sus mejores resultados en intervalos de edad intermedios, hasta los 45 años, a partir del cual empiezan a descender. Los niveles de abstención experimentan una ligera caída en personas con edad avanzada.

- **Intención de voto por situación laboral.**

Se comprueba si en la intención de voto existen diferencias significativas en las respuestas dependiendo de la situación laboral de la persona entrevistada. Se realiza un contraste de independencia chi cuadrado (**Anexo 4**) y se observa que el p-valor es bastante menor que $\alpha = 0.05$, por lo que se determina que ambas variables son dependientes. Se muestran los siguientes diagramas de barras desglosados por situación laboral.

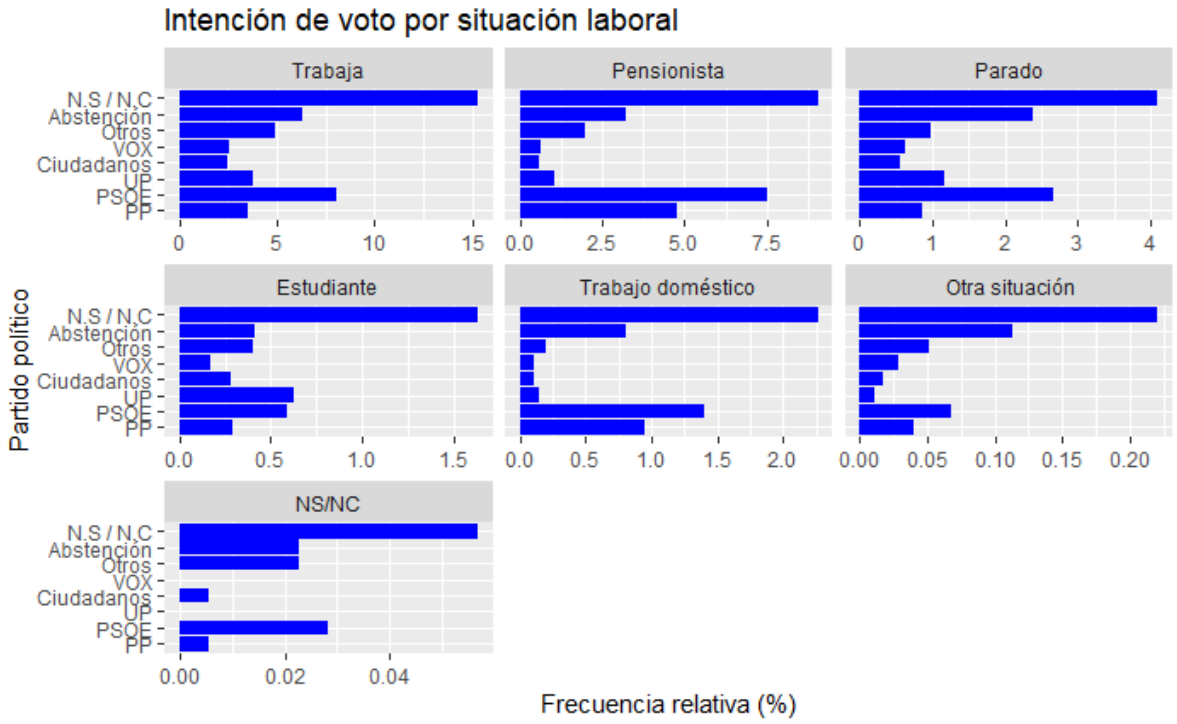


Ilustración 17. Intención de voto por situación laboral.

Se observa que el PP obtiene sus mejores resultados en pensionistas y personas en situación de trabajo doméstico sin remunerar, mientras que obtiene resultados inferiores similares entre sí en personas en situación de trabajar, parados o estudiantes. Un comportamiento similar puede observarse en el PSOE, que obtiene sus mejores resultados entre pensionistas y trabajadores domésticos. No obstante, obtiene buenos resultados entre los trabajodores y parados, sufriendo un descenso entre los estudiantes. En el caso de UP, obtiene los mejores resultados entre estudiantes, donde es la opción más elegida por delante del PSOE. Entre los parados y trabajadores obtiene buenos resultados, donde es la segunda opción por delante del PP. Existe un descenso considerable en los resultados entre pensionistas y trabajadores domésticos. En el caso de VOX obtiene sus mejores resultados entre los trabajadores y parados, en estudiantes sufre un ligero descenso que se ve acentuado entre pensionistas y trabajadores domésticos. Ciudadanos obtiene sus mejores resultados entre trabajadores y estudiantes, sufriendo un ligero descenso en los parados y un claro descenso entre los pensionistas y trabajadores domésticos al igual que ocurre con VOX.

- **Intención de voto por clasificación ideológica.**

Resulta interesante ver los resultados de la intención de voto en función de la clasificación ideológica de las personas entrevistadas, ya que sirve para medir la calidad de las respuestas de intención de voto. Es decir, cabe esperar que personas que se identifiquen con ideología de izquierdas tengan intención de voto a partidos como PSOE o UP. De igual manera ocurre con personas que se identifiquen con ideología de derechas que serán más propensos a tener intención de voto por partidos como PP o VOX. De esta forma podemos calibrar la veracidad de las respuestas proporcionadas por las personas entrevistadas en la encuesta.

Se realiza un contraste de independencia chi cuadrado $\chi 2$ (**Anexo 4**) para ver si las dos variables son dependientes . Por tanto, se formulan las hipótesis nula (H_0) y alternativa (H_1).

H_0: las dos variables son independientes

H_1: las dos variables son dependentes

Se realiza el contraste en R usando la instrucción `chisq.test()` sobre la tabla de frecuencias absolutas conjunta. El p-valor tiene un valor de 2.2e-16, que al ser menor que $\alpha = 0{,}05$, tenemos evidencia estadística suficiente como para rechazar la hipótesis nula H_0. De forma implícita, se puede inferir que las variables intención de voto y clasificación ideológica son dependientes.

Se pueden ver los resultados en los siguientes diagramas de barras desglosados por cada ideología política.

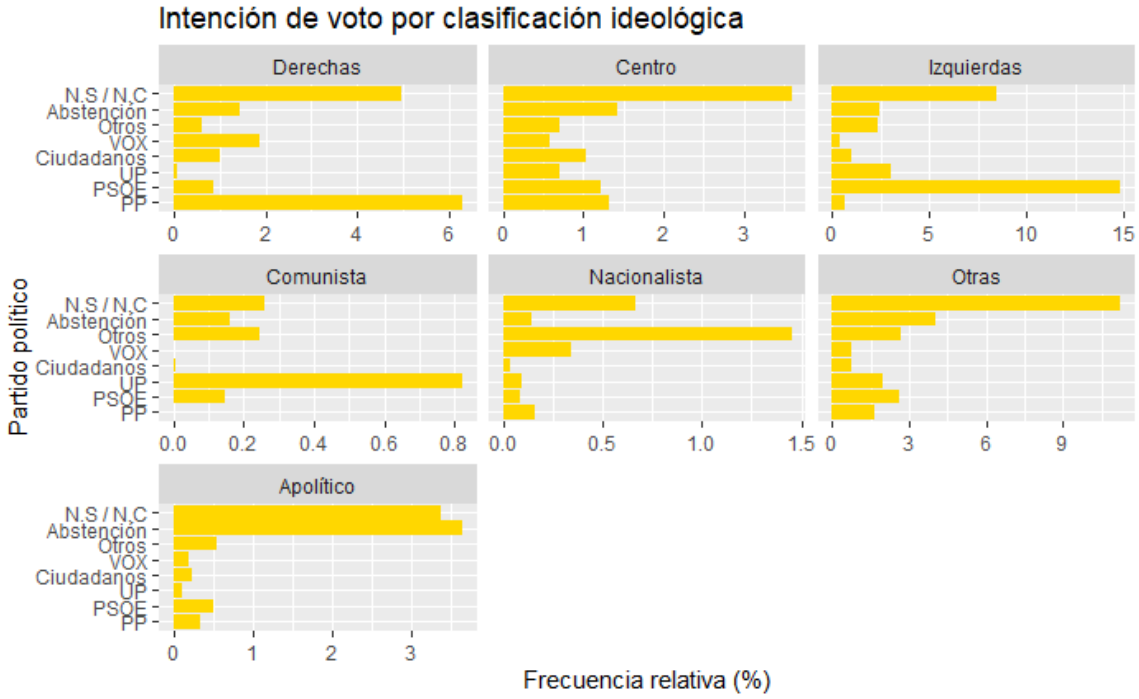


Ilustración 18 . Intención de voto por clasifiación ideológica.

Se observa a simple vista que los resultados cumplen con lo que cabría esperar en función de cada ideología. En personas que se identifican de derechas los mejores resultados los obtiene el PP con aplastante mayoría, seguido de VOX y Ciudadanos. En personas que se identifican de izquierdas los mejores resultados por amplia mayoría los obtiene el PSOE seguido de UP. En personas con ideología comunista se observa que la inmensa mayoría tiene intención de votar a UP, seguido del PSOE y se observa que partidos como VOX o PP no reciben ninguna respuesta en intención de voto, lo cual es de esperar. En los nacionalistas se ve claramente como la gran mayoría prefiere votar a otros partidos, lo cual es bastante lógico, ya que en la recodificación se incluyó a todos los partidos nacionalistas dentro de ese grupo. En personas que se identifican como de centro existe una dispersión en la intención de voto a todo el espectro político.

Por tanto, se observa que los resultados concuerdan con lo que cabría esperar y podemos inferir que las respuestas son bastante veraces y no existe ocultación de voto real o intento de engaño en la muestra.

4- Metodología.

Con objeto de realizar una implementación de modelos que nos permitan obtener predicciones fiables de la intención de voto en España a través de la muestra seleccionada del barómetro preelectoral 3263 del CIS, se hace uso del análisis de segmentación, concretamente de técnicas de árboles de clasificación.

Se exponen a continuación aquellas más relevantes (Escobar, 2007) [7].

- **Segmentación AID (*Automatic Interaction Detection*)**:

  Algoritmo creado en 1963 por Morgan y Sonquist, diseñado para variables independientes cuantitativas que segmenta mediante la descomposición de sumas cuadráticas intergrupales o externas. Es decir, realiza particiones binarias de la muestra, generando subconjuntos más homogéneos.

- **Segmentación CHAID (*Chi Automatic Interaction Detection*)**:

  Algoritmo similar al procedimiento anterior que utiliza el estadístico chi cuadrado $\chi 2$. Sirve para variables no necesariamente cuantitativas, pueden ser nominales u ordinales. Se diferencia del AID en que éste solo permite segmentaciones en dos trozos mientras que CHAID permite subdivisiones de tres o más fragmentos. En este procedimiento si las categorías no son distintas, se produce un proceso de sucesivas fusiones de categorías.

- **Árboles de clasificación y regresión CART (*Classification and Regression Trees)*:**

  Serie de algoritmos propuestos por Breiman, Friedman, Olshen y Stone en 1984. Se amplían las medidas para la selección de las mejores variables dependiendo de si las variables predictoras son cualitativas o cuantitativas. Para variables cualitativas se utiliza una medida denominada de Gini y para variables cuantitativas mínimos cuadrados ordinarios y/o residuos absolutos. Introducen una técnica denominada poda del árbol. No se fusionan las categorías como ocurre con CHAID, más bien actúa de forma similar a AID en la que siempre se divide la muestra en dos mitades en cada paso, tratando de encontrar la división que ofrezca una mejor medida de Gini para cada clasificador. En estos procedimientos se puede discernir la importancia de las variables que son reflejadas en el árbol ayudando así al análisis de la variable dependiente.

- **Algoritmo QUEST (*Quick, Unbiased, Efficient Statistical Tree*):**

  Este algoritmo propuesto por Loh y Shih en 1997 es también de carácter binario. Usa análisis clúster o conglomerados para variables cuantitativas y análisis discriminante cuadrático para variables cualitativas. Selecciona la variable que mejor segmenta los datos para ramificar cada nodo en dos etapas. Primero elimina el mejor predictor para luego buscar la mejor partición con la variable seleccionada.

Se ha tomado la decisión que la mejor implementación para realizar el objetivo de este trabajo es partir de los algoritmos CART con el fin de realizar los denominados métodos de conjunto o de ensemble. Estos métodos se desarrollaron siguiendo la línea de los CART, en la que se aplican combinaciones de manera sucesiva a las submuestras. Es por ello por lo que inicialmente, se realiza un algoritmo de tipo CART denominado *rpart*, a modo de ilustración, con la finalidad de mejorar la comprensión posterior de los métodos de ensemble.

Los métodos de ensemble construyen muchos modelos para posteriormente usarlos de forma conjunta y suelen ser más precisos que los modelos base. Forman parte de los mejores modelos de *machine learning* que se pueden usar para el tratamiento y análisis de datos, siendo usualmente ganadores en competiciones en páginas especializadas como *Kaggle*.

Al ser métodos muy flexibles tienen una desventaja con respecto a modelos menos flexibles y es la pérdida de interpretabilidad del modelo y la posibilidad de que el modelo siga el ruido muy de cerca y sobreajuste los nuevos datos al realizar las predicciones (James, et al. 2017) [8].

Existen dos tipos de ensembles principales (Kuhn y Johnson, 2016) [9]:

- ***Bagging***: Construyen varios modelos base en paralelo utilizando el mismo método. Para la construcción de los modelos se generan distintas muestras en cada modelo para posteriormente calcular la media de los modelos en el caso de regresión y clase mayoritaria en el caso de clasificación. Se consigue de esta forma eliminar los sobre ajustes producidos por los sesgos aleatorios que pudieran existir en el conjunto de entramiento. Uno de los modelos de *bagging* más utilizado es el denominado *Random Forest* (Breiman, 2001) [10] o bosque aleatorio, en el que se generan muestras aleatorias con reemplazamiento en la creación de las muestras. Posteriormente se utiliza un proceso denominado aleatorización, en el que para elegir un atributo para un nodo se elige al mejor de un subconjunto de atributos elegido aleatoriamente y no al mejor de todos ellos.

- ***Boosting:*** Término que viene del inglés *to boost* (mejorar). Se trata de una técnica en la que se parte inicialmente de modelos base denominados *weak learners* (métodos débiles), que no distan mucho de conseguir un error al azar, para posteriormente ir añadiendo de forma secuencial modelos base al ensemble de modo que cada modelo añadido corrija los errores del modelo anterior. Ya que cada modelo se centra en los fallos del anterior, si los fallos son debidos al ruido, *boosting* seguirá de cerca el ruido pudiendo ocasionar sobre ajuste en las predicciones con datos nuevos. Los modelos *boosting* más conocidos son *Adaboost* para clasificación binaria, en el que se van asignando pesos, que se irán adaptando, de manera que los datos difíciles pasarán a tener más peso y los más fáciles, menos. *Gradient Boosting Machine* para regresión, aunque se puede usar para clasificación, en el que se ajusta modelos a nuevos datos cuyas salidas son los errores del modelo anterior y *Extreme Gradient Boosting.*

A continuación, se implementan los algoritmos *rpart*, *random forest*, *gradient boosting*, *extreme gradient boosting* y C.5 con una variante de *boosting* para clasificación.

4.1- Partición de la muestra e implementación de patrones de datos faltantes.

Resulta necesario para la implementación de los modelos de ensemble realizar una transformación de la muestra seleccionada en apartados anteriores. Se realiza una reorganización de la muestra de la siguiente manera (**Anexo 5**).

La variable de interés que se quiere predecir es P9 (Intención de voto), a la cual vamos a denominar Y. El resto de predictores se van a denominar X.

Se reorganiza la matriz de datos [X,Y] = [(X1,Y1);(X2,Y2)] donde:

- (X1,Y1) es la muestra donde los Y1 (P9, intención de voto) tienen valores distintos de NA's (valores faltantes). De aquí se realiza una partición de la muestra en dos submuestras, una de entrenamiento a la cual se llamará *x1y1_train* y otra para datos de prueba que se llamará *x1y1_test.* Se realiza este procedimiento ya que, no tiene sentido tener valores faltantes en P9 en la muestra de entrenamiento porque no nos serviría para ajustar el modelo. Tampoco tiene sentido tener valores faltantes en P9 en la muestra de prueba porque no podemos verificar el desempeño de los procedimientos.

- (X2, Y2) es la muestra donde los Y2 (P9, intención de voto) tienen NA´s. Estos son los individuos cuya intención de voto es desconocida y queremos predecir.

A continuación, se modifican los patrones de datos faltantes en la muestra generada. El motivo de usar esta modificación es que cuando el patrón de las personas entrevistadas que no responden es distinto de los que sí responden, es posible que las predicciones se concentren en la categoría que más se le parezca. La idea es entrenar el modelo con datos que tengan patrones lo más parecidos a los que luego tendríamos que predecir. En las muestras de entrenamiento se crean patrones de faltantes similares a los patrones de los faltantes en P9.

Para poder generar patrones de faltantes transformamos los valores tipo 99 (No sabe/no contesta) de todas las variables a NA. Posteriormente, debemos transformar todas las variables a tipo numérico, ya que se realiza una multiplicación de los predictores de la muestra X1 por el cociente entre la muestra X2 divida por ella misma. Es decir, X1= X1 * X2 / X2 convenientemente conformadas. Hay que tener cuidado de no modificar la variable P9, por lo que la operación solo se realiza a los predictores. Esta operación es interesante porque vale 1 si hay dato, pero vale NA si no lo hay. De esta forma se consigue implementar los patrones en la muestra X1Y1. Una vez realizado este proceso se realiza una transformación de los datos faltantes al valor 99 como estaban inicialmente y se obtiene una partición de la muestra X1Y1 en entrenamiento y prueba. Para la muestra de entrenamiento, denominada *x1y1_train,* se realiza un muestreo aleatorio de la muestra X1Y1, en el que se escogen el 70% de los datos asegurando la reproducibilidad del modelo fijando la semilla aleatoria. El 30% de los datos restantes no escogidos aleatoriamente en la parte de entrenamiento son los denominados *x1y1_test*, que serán usados para prueba. Para la implementación de los modelos de ensemble se transforman de nuevo las variables a tipo factor.

4.2- R-part (Recursive Partitioning and Regression Trees).

Se implementa un modelo de árbol de decisión CART denominado particionamiento recursivo o R-part. Esta técnica de aprendizaje supervisado nos va a servir de inicio para una mejor compresión de los modelos de ensemble, sobre todo de *Random Forest*. Tenemos una variable dependiente, P9 (Intención de voto), la cual es nuestro objetivo y queremos obtener una función que tenga como premisa predecir el valor de la intención de voto para valores desconocidos, en función de una variables independientes o predictores. Se usa el paquete *rpart* y *rpart.plot* de R para implementar el modelo (**Anexo 6**). El procedimiento de este algoritmo es encontrar los predictores que mejor separa los datos en grupos, que se corresponden con cada una de las categorías de nuestro objetivo, que en este caso es un variable de tipo factor con 7 categorías. La separación se realiza por medio de reglas y a cada regla le corresponde un nodo. Esto quiere decir que los datos para los que la regla es verdadera tienen más probabilidad de pertenecer a una categoría que a otra.

De modo que, una vez realizado este proceso, los datos son particionados en categorías a partir de las reglas obtenidas. Se repite el proceso de forma recursiva hasta que resulta imposible obtener una mejor separación y el algoritmo se detiene.

Tras realizar la implementación en R (**Anexo 6**), se entrena el modelo con los datos de entrenamiento *x1y1_train* y se realiza una predicción usando como datos nuevos los datos de prueba *x1y1_test.* Seguidamente, se obtiene la matriz de confusión, en la que podemos observar cómo de preciso se han clasificado los datos de prueba.

Se puede observar que la *accuracy* (precisión) de los datos de prueba es del 89,38%, lo cual es una medida bastante buena.

	Categoría						
Predicción	1	2	3	4	5	6	7
1	525	0	0	2	5	2	14
2	50	1039	39	18	15	72	37
3	0	2	298	0	0	19	1
4	1	0	0	158	1	1	4
5	0	0	0	1	194	0	1
6	4	0	18	2	0	332	3
7	11	6	7	8	7	28	643

Tabla 8. Matriz de confusión R-part.

Una vez que tenemos el modelo evaluado y entrenado realizamos las predicciones para los datos de X2Y2, en los cuales aparecen todos los datos faltantes de la variable de interés P9 (Intención de voto).

Se realiza un gráfico, a modo de ilustración, para visualizar el comportamiento del árbol generado con el modelo creado.

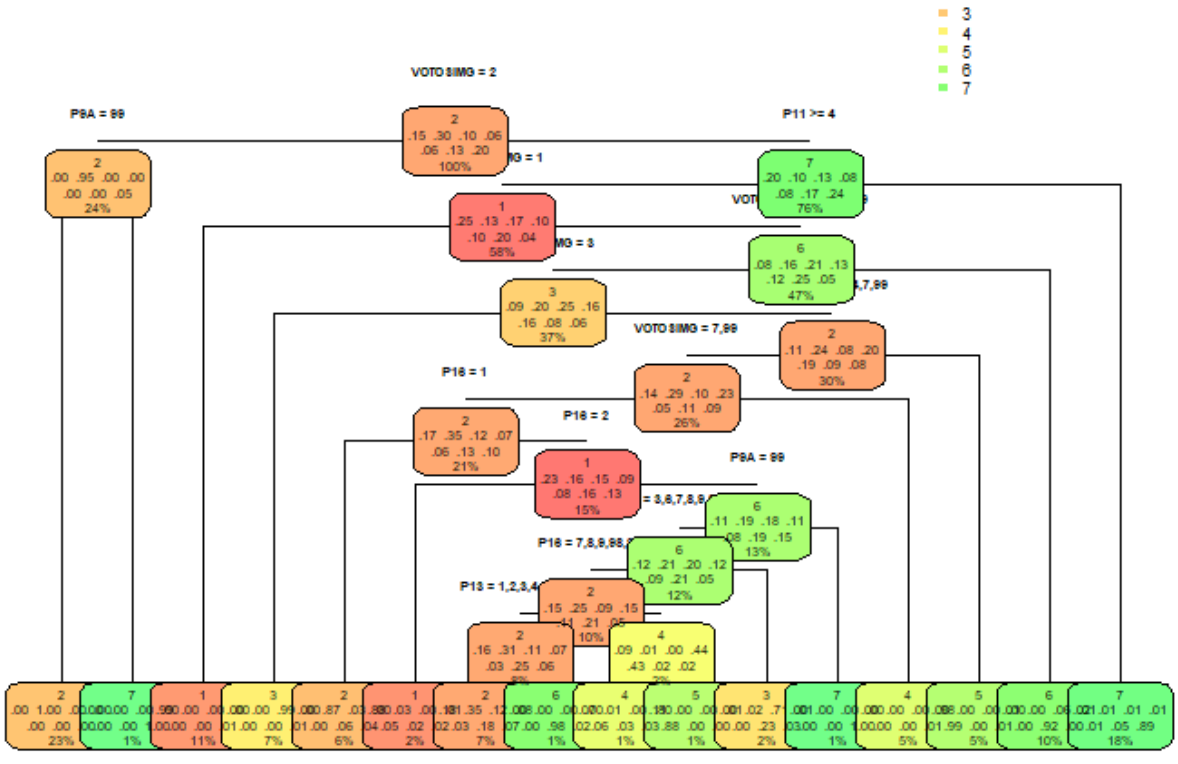


Ilustración 19. Árbol R-part.

Cada uno de los rectángulos representan los nodos del árbol y en encima de éstos se puede observar la regla de clasificación. Los colores de los rectángulos representan una categoría, la cual viene incluida en la parte superior del interior del rectángulo con su número correspondiente. El porcentaje de la parte inferior de cada rectángulo indica la proporción de casos que se han incluido en ese nodo, mientras que los números internos indican la proporción de casos que pertenecen a cada categoría.

Por ejemplo, si nos fijamos en los nodos terminales (aquellos donde ya no se divide más el árbol, es decir los últimos) el primero de la parte izquierda indica que todos los datos son clasificados como clase 2 (PSOE según la recodificación) ya que se indica con 1 en la posición 2 y que éstos representan un 23% del total de todos los datos. De forma análoga se puede interpretar el resto de los nodos.

Se puede observar que de manera sencilla y sin necesidad de gran potencia de computación, se pueden obtener modelos con buenos resultados de predicción. Además, este tipo de árboles proporcionan una buena interpretabilidad de las reglas que se están usando y son el punto de partida del siguiente modelo, que son los bosques aleatorios, donde se pierde la interpretabilidad como se verá en el siguiente apartado.

Pero este algoritmo tiene una desventaja y es que los resultados pueden variar ostensiblemente en función de los datos que se usen para entrenar el modelo. Se puede sobre ajustar los nuevos datos que queremos predecir por este motivo.

4.3- Random Forest.

Tras comprender cómo funcionan los arboles tipo CART, resulta más sencillo entender cómo funcionan los métodos de *bagging (bootstrap aggregating)*, los cuales son utilizados para reducir la varianza promediando un conjunto de modelos base. Si los modelos que se van agregando en este proceso están correlacionados, el proceso de *bagging* no logrará reducir en exceso la varianza, con lo que no se consigue mejorar el modelo.

Random Forest (Breiman, 2001) [10] es una modificación del algoritmo de *bagging*, el cual puede esquivar el problema de la correlación al realizar una selección aleatoria de los predictores antes de evaluar cada división y elegir el mejor de esa selección. El procedimiento consiste en realizar un muestreo aleatorio con reemplazo. Posteriormente, se realiza un proceso de aleatorización que consiste en lo siguiente: al elegir un atributo para un nodo, no se elige al mejor de todos ellos, sino que se elige el mejor de un subconjunto con m atributos. Por ejemplo, en nuestro caso hay M=25 atributos y m=5, entonces 5 se cogen aleatoriamente, y el mejor de esos 5 es seleccionado. Normalmente, para clasificación se usa la raíz cuadrada de M y para regresión se usa m = M/3. Una vez finalizado el proceso de aleatorización, se usan los vectores aleatorios para construir árboles de decisión similares al visto en el apartado anterior. Por lo tanto, árboles diferentes usan ahora atributos diferentes y posteriormente, se combinan todos.

Para la implementación del modelo en R (**Anexo 7**) se usa la librería *randomForest*. Se indica que la variable objetivo o dependiente es P9 (Intención de voto) y el resto son de predictores o variables independientes del modelo. Al híper-parámetro m se lo suele denominar *mtry*, que en nuestro caso vale 5. Por defecto, la función realiza 500 árboles, se puede modificar cambiando el valor del hiper- parámetro *ntree.* Se entrena el modelo con los datos de entrenamiento *x1y1_train.* Se observa en el siguiente gráfico como los errores se van reduciendo a medida que aumenta el número de árboles, llegando a estabilizarse alrededor de 250.

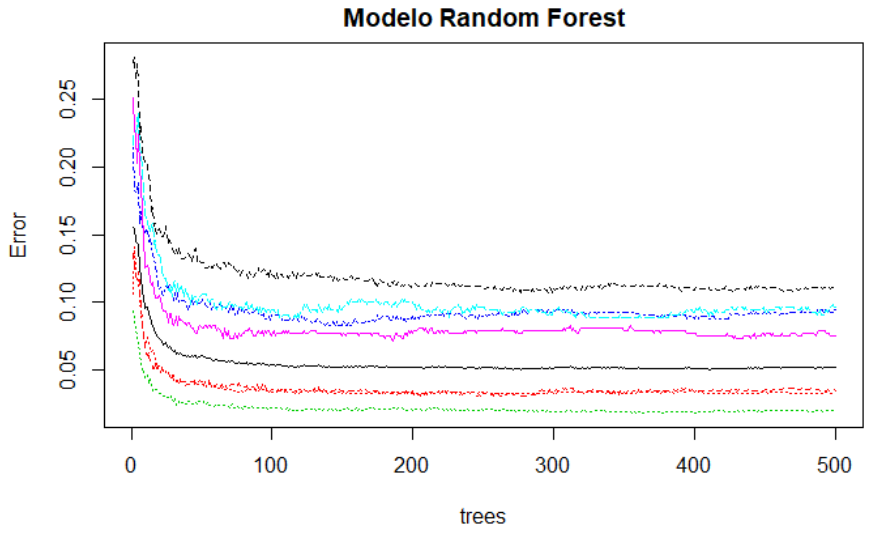


Ilustración 20 . Errores modelo Random Forest.

Vemos la matriz de confusión del modelo con los datos de entrenamiento para comprobar cómo se comporta al clasificar las categorías.

	Categoría							
Predicción	1	2	3	4	5	6	7	class.error
1	1206	18	0	5	7	7	7	0.03520000
2	11	2475	12	5	0	11	12	0.02019002
3	0	33	760	3	0	34	9	0.09415971
4	12	15	3	476	2	14	5	0.09677419
5	16	6	1	5	466	3	7	0.07539683
6	19	37	32	7	1	936	18	0.10857143
7	6	21	6	6	1	14	1579	0.03306797

Tabla 9. Matriz de confusión Random Forest.

Se puede observar en la matriz como los errores de entrenamiento para cada clase son bastante bajos y se obtiene una *accuracy* (precisión) del 94,37% y un error *out-of-bag* del 5,17%, lo cual son resultados bastante satisfactorios. Se procede a explicar qué clase de error es éste último y se realiza el cálculo de la importancia de las variables para terminar realizando las predicciones con los datos de prueba.

- *Out of bag error:*

  Para evaluar un modelo es conveniente realizar la prueba usando datos distintos a los de entrenamiento. Los modelos Random Forest son capaces de evaluar el modelo sin separar una parte de los datos (la partición en entrenamiento y prueba), ya que utilizan el hecho de que no todos los datos se encuentran en las muestras y usan esos que no están incluidos para evaluar el árbol. El error de cada dato se calcula usando únicamente aquellos árboles en los que no se utilizó dicho dato para construirlo.

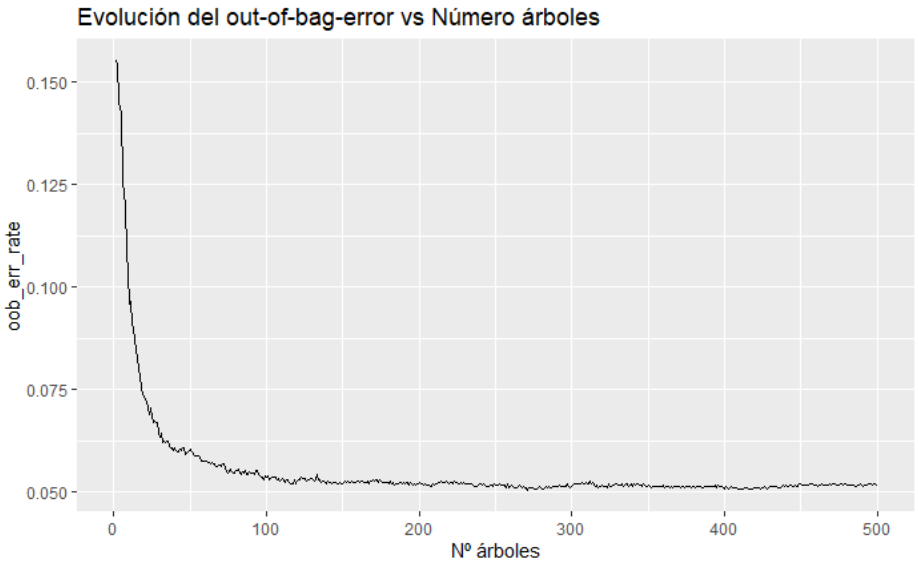


Ilustración 21. Evolución del OOB vs Número de árboles en Random Forest.

Se observa en el gráfico anterior como el error *out of bag* va disminuyendo a medida que aumentamos el número de árboles, hasta llegar a un punto, alrededor de 300 donde estabiliza. En las salidas del modelo que propociona R se puede ver como este error es del 5,17%, lo cual es un buen indicador que indica que el modelo ha clasificado bastante bien en entrenamiento.

- Importancia de los predictores. Reducción de *Accuracy*.

  Al tratarse de un método muy flexible que aumenta la capacidad del modelo para realizar predicciones tiene una contrapartida que es la pérdida de interpretabilidad del modelo. Al no poder representar de forma gráfica, como se puede hacer con el modelo de *rpart* visto en el apartado anterior, ya que se trata de combinaciones de muchos árboles, perdemos interpretabilidad.

  Para suplir este inconveniente se realiza el cálculo de la importancia de los predictores, que nos permite obtener una idea de cuáles son las variables más influyentes en el modelo y cuáles tienen más peso a la hora de realizar las predicciones. Se calcula la importancia de las variables con la instrucción `importance()` y se realiza un gráfico usando como medida la reducción de *accuracy.*

Se observa en la siguiente ilustración que las variables más influyentes son VOTOSIMG (voto +simpatía partido) y P11 (escala de probabilidad de votar) en términos de reducción de la precisión, en menor medida P16 (preferencia personal como presidente del gobierno), P9A (simpatía partido político) y P13 (partido que considera más cercano a sus ideas) son también relevantes.

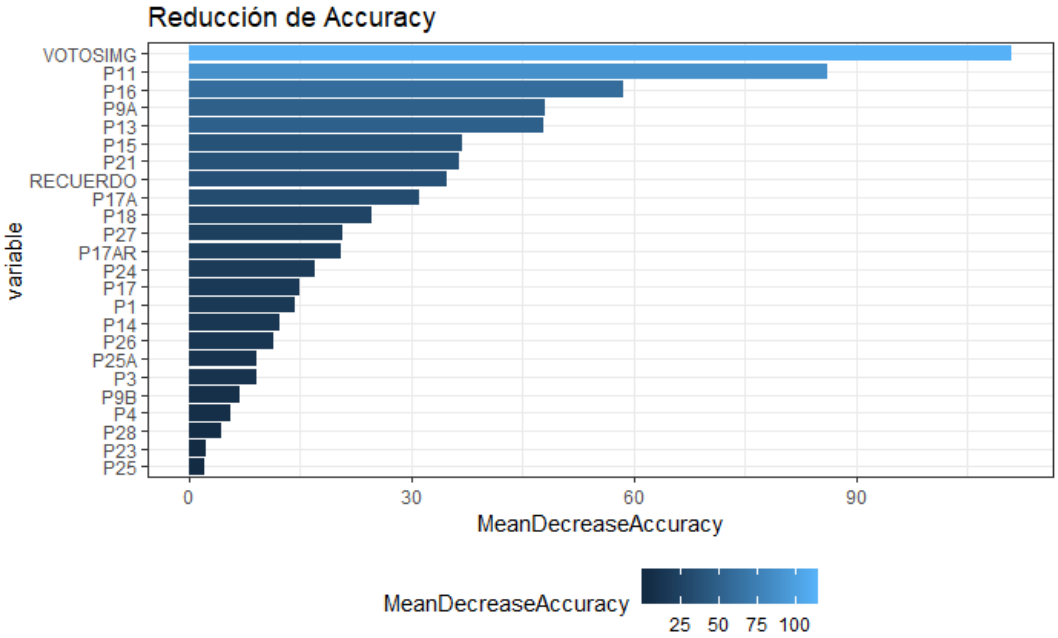


Ilustración 22. Importancia variables en Random Forest. Reducción de Accuracy.

- Importancia de los predictores. Impureza de Gini.

  Otra medida para evaluar la importancia de los predictores en la instrucción `importance()` es la reducción de la impureza o medida de Gini. Es una métrica que mide con qué frecuencia se elige aleatoriamente un elemento del conjunto, el cual es clasificado de forma incorrecta si es clasificado de forma aleatoria en base a la distribución de las categorías en el subconjunto. Se calcula realizando un sumatorio de las probabilidades de cada elemento de ser elegido multiplicado por la probabilidad de error de ser incluido en un categoría. Cuando todos los elementos de un nodo se corresponden con una sola categoría el índice alcanza un valor de cero y por tanto, su mínimo.

  Se puede observar en la siguiente ilusración que, de nuevo VOTOSIMG es la variable más importante, seguida de P16, P13, P11y P9A, tal y como ocurría con la métrica anterior. En menor medida las variables RECUERDO (recuerdo de voto en las elecciones anteriores) y P15 (partido que desearía ganador) son también relevantes.

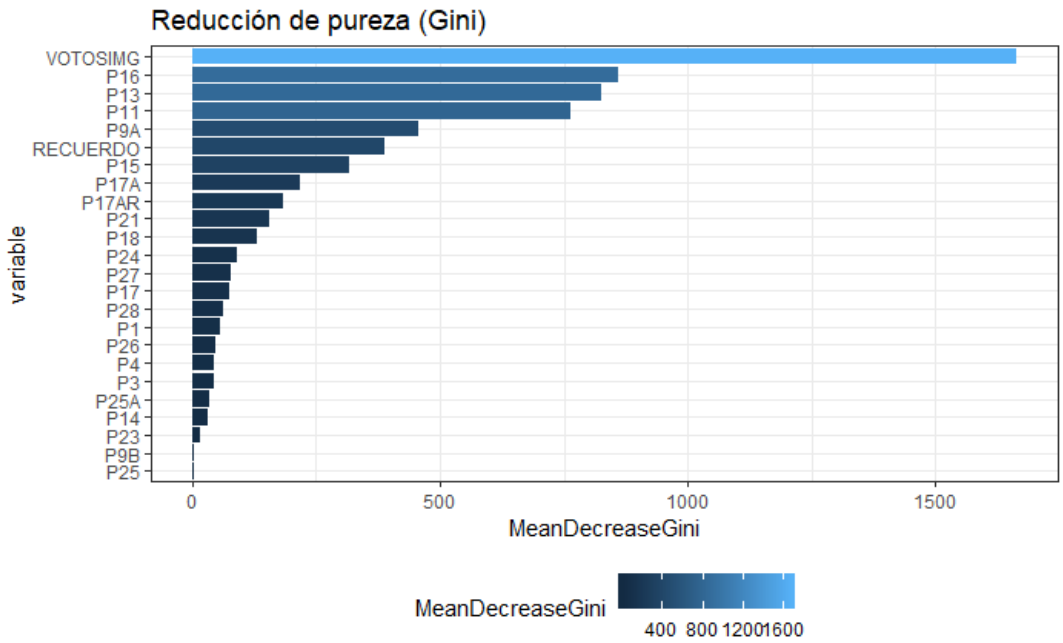


Ilustración 23. Importancia variables en Random Forest. Impureza de Gini.

4.3- C5.0 Boosting.

Este algoritmo es una particularización de *Adaboost* para clasificación multiclase. En nuestro caso la variable de interés intención de voto no es dicotómica, sino que son siete clases, se procede por tanto a la realización de un caso particular del algoritmo C5.0, el cual incluye *boosting* en el paquete C50 de R. Este algoritmo lo propuso Quinlan (Quinlan, 1999) [11], al igual que su predecesor C4.5 para crear árboles de clasificación. Tiene un comportamiento similar a *Adaboost*, en el que asigna pesos a los errores para que el siguiente árbol añadido de forma secuencial se centre en solucionar, con la diferencia que permite fijar distintos pesos a cada error. Al realizar el ensemble, la métrica que usa para dividir el árbol no es la impureza de Gini sino la entropía. Si se añade otro árbol que no mejora lo suficiente al anterior, el procedimiento se detiene (Amat, 2017) [12].

Para la creación del modelo en R (**Anexo 8**), usamos como predictor P9 (Intención de voto) y el resto de las variables como predictores. Entrenamos el modelo con los datos de entrenamiento *x1y1_train*, usamos 100 árboles indicando en el híper-parámetro *trials* que queremos dicha cantidad.

Una vez entrenado el modelo realizamos las predicciones con los datos de prueba *x1y1_test* y obtenemos la matriz de confusión para ver la precisión de las clasificaciones.

	Categoría						
Predicción	1	2	3	4	5	6	7
1	569	6	3	2	7	4	5
2	10	1017	13	5	5	19	8
3	1	4	326	1	0	16	3
4	2	1	2	174	2	4	2
5	0	0	0	0	202	0	2
6	4	12	13	4	3	396	9
7	5	7	5	3	3	15	674

Tabla 10. Matriz de confusión C5.0

Se puede observar en la matriz de confusión que el algoritmo clasifica bastante bien los datos de prueba, obteniendo una precisión (*accuracy*) del 94,11% con un error de 0.0588565 en las predicciones al clasificar los datos de prueba. Posteriormente se realizan las predicciones de nuestro interés con los datos de la muestra X2Y2 en donde tenemos valores faltantes en la variable intención de voto, las cuales se mostrarán en los siguientes apartados.

Con objeto de identificar la importancia de las variables más influyentes en el modelo se usan dos tipos de métricas que incorpora el algoritmo para los ensembles.

- Importancia según el uso del predictor:

  Esta métrica mide el porcentaje de las observaciones de la muestra de entrenamiento que han sido elegidas para aquellos nodos en los que el predictor que se está midiendo ha participado. En el siguiente gráfico vemos las más influyentes.

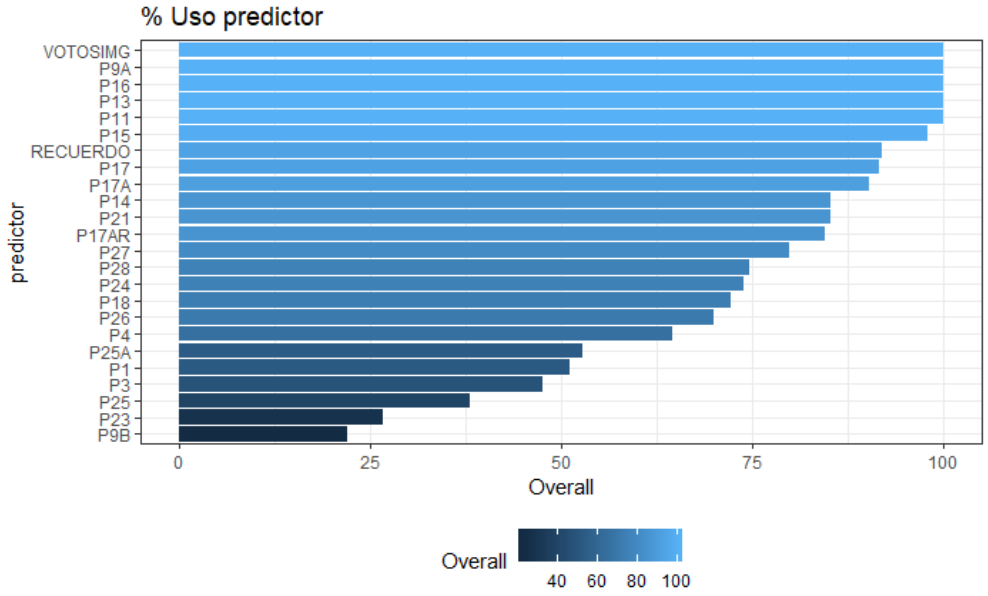


Ilustración 24 . Importancia variables en C5. Uso predictor.

- Importancia de los predictores según divisiones:

  Esta métrica mide el porcentaje de divisiones en las que ha participado cada predictor. Se puede apreciar en el gráfico como la importancia de las variables según esta medida difiere de la anterior, siendo P11, P13 y P16 las más importantes.

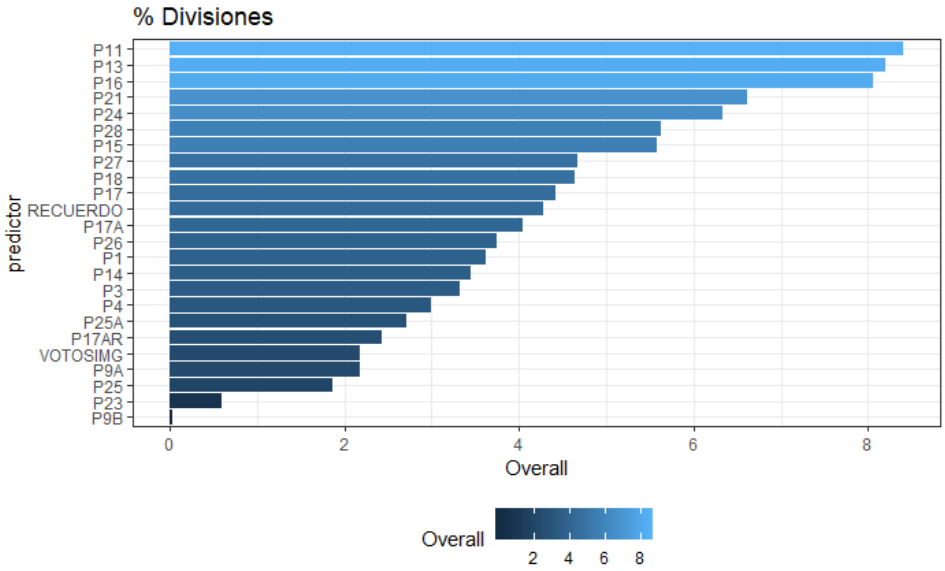


Ilustración 25. Importancia variables en C5. Porcentaje de disviones.

4.4- Gradient Boosting.

Este algoritmo es una adaptación de la idea de *boosting* comentada anteriormente para problemas de regresión, aunque también se puede utilizar para clasificación de varias clases dado su alto nivel de flexibilidad. Tiene una función de coste, que se usa como medida de error a minimizar, por ejemplo, el error cuadrático medio. Basándose en la función de coste se implementan los *weak learners*, que en nuestro caso son árboles. En cada iteración, se ajusta un modelo nuevo cuyos datos son los errores del modelo anterior y se minimiza la función de coste. El proceso se repite M veces hasta conseguir reducir los errores todo lo posible (Amat, 2017) [12]. Tiene una desventaja y es que al ser un algoritmo de tipo secuencial le puede afectar el ruido, ya que el modelo recién añadido se centra en los errores del anterior y si estos errores son debidos a la existencia de ruido *gradient boosting* los seguirá de cerca. Esto puede ocasionar que se sobre ajuste el modelo con los datos de entrenamiento, sobre todo si el número de modelos base es muy grande, llegando a tener muy poco error de entrenamiento, pero al introducir los datos nuevos que se quieren predecir, las predicciones no sean buenas y se produzca un alto error de prueba.

Para su implementación en R se utiliza la librería *gbm* (**Anexo 9**). Se crea el modelo usando como variable objetivo la variable de interés P9 (Intención de voto) y el resto de las variables se usan como predictores o variables independientes en el modelo. Se especifica la distribución multinomial, ya que nuestro problema de clasificación consta de varias categorías y se incluyen los parámetros *n.trees* con 500 árboles y se usa validación cruzada con 5 *folds* con el objetivo de obtener el error en función del número de árboles indicando el valor en el argumento *cv.folds*, el parámetro *shrinkage* se deja con el valor por defecto de 0,001 ya que controla la influencia que tiene cada modelo sobre todo el ensemble. Se entrena el modelo con los datos de entrenamiento *xy1_train* y se procede a calcular la mejor iteración usando la función `gbm.perf()` que incorpora el paquete *gbm.* En el gráfico siguiente se puede observar cómo va descendiendo el error a medida que vamos aumentando las iteraciones.

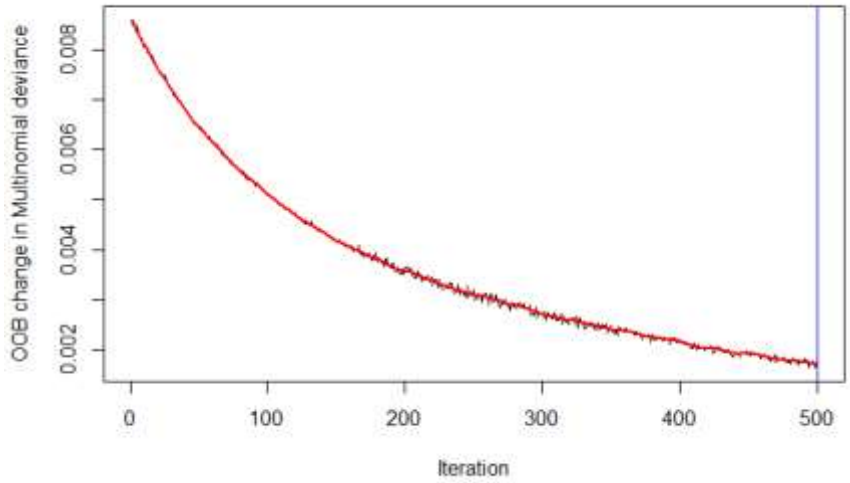


Ilustración 26 . Descenso del OOB por iteración en Gradient Boosting.

Se procede a calcular las predicciones con los datos de prueba *x1y1_test* y obtenemos la matriz de confusión. Podemos ver como la precisión (*accuracy*) del modelo es del 86,69% , lo cual no es un mal dato de la clasificación de las distintas categorías.

	Categoria						
Predicción	1	2	3	4	5	6	7
1	530	0	0	2	5	2	27
2	52	1037	50	46	41	101	61
3	0	2	291	0	0	5	10
4	0	0	0	139	0	0	5
5	0	0	0	0	174	0	6
6	3	0	18	1	0	330	2
7	6	8	3	1	2	16	592

Tabla 11. Matriz de confusión Gradient Boosting

- Importancia de los predictores según la reducción del MSE:

  Se calcula la importancia de las variables usando como métrica la reducción del MSE (*Mean Square Error*) o error cuadrático medio. Se observa que la variable más importante es VOTOSIMG seguida de P11, P16, P13 y finalmente P9A. Muy parecido a la reducción de la medida de Gini en *Random Forest*.

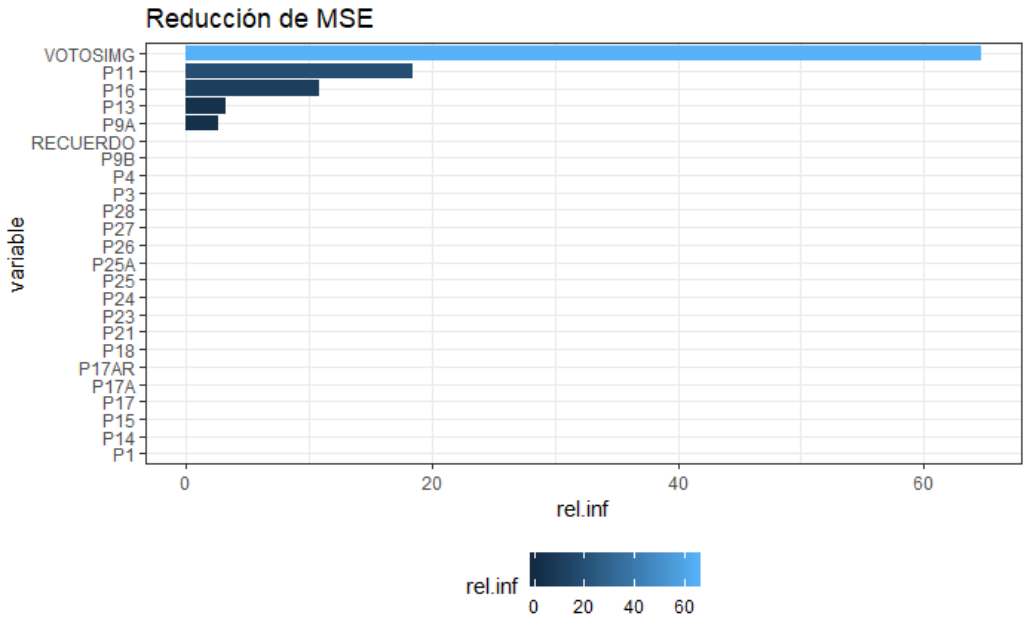


Ilustración 27. Importancia variables en Gradient Boosting.Reducción del MSE.

4.5- Extreme Gradient Boosting.

Este algoritmo es una implementación de los árboles de decisión diseñado para aumentar la velocidad y la eficiencia en el tiempo de cómputo. Es una implementación del árbol de decisión que usa el aumento del gradiente de una forma muy rápida en comparación con otras implementaciones del mismo estilo. El termino aumento de gradiente viene de un "enfoque donde se crean nuevos modelos que predicen los residuos o errores de modelos anteriores y luego se suman para hacer la predicción final. Se llama aumento de gradiente porque utiliza un algoritmo de descenso de gradiente para minimizar la pérdida al agregar nuevos modelos" (Brownlee, 2016) [13]. Este algoritmo es una mejora con respecto a *Gradient Boosting* con el fin de optimizar sus recursos. Sus características más reseñables son que se puede utilizar tanto para regresión como para clasificación, se puede paralelizar la construcción de los modelos, y tiene implementado un manejo automático de los valores faltantes. Este algoritmo tiene mucha fama por ser ganador de muchas competiciones de análisis de datos en páginas como *Kaggle*.

Para su implementación en R se utiliza la librería *xgboost* (**Anexo 10**). Se crea el modelo usando como variable objetivo la variable de interés P9 (Intención de voto) y el resto de las variables se usan como predictores o variables independientes. Se usa la librería *caret* para entrenar el modelo con los datos de entrenamiento *x1y1_train* y se utiliza validación cruzada de 5 *folds* para calibrar los parámetros del modelo.

En el siguiente gráfico se puede ver la representación del nivel de precisión (*accuracy*) para los distintos parámetros mediante validación cruzada. Para una mejor visualización se adjunta gráfico ampliado en el **Anexo 10**.

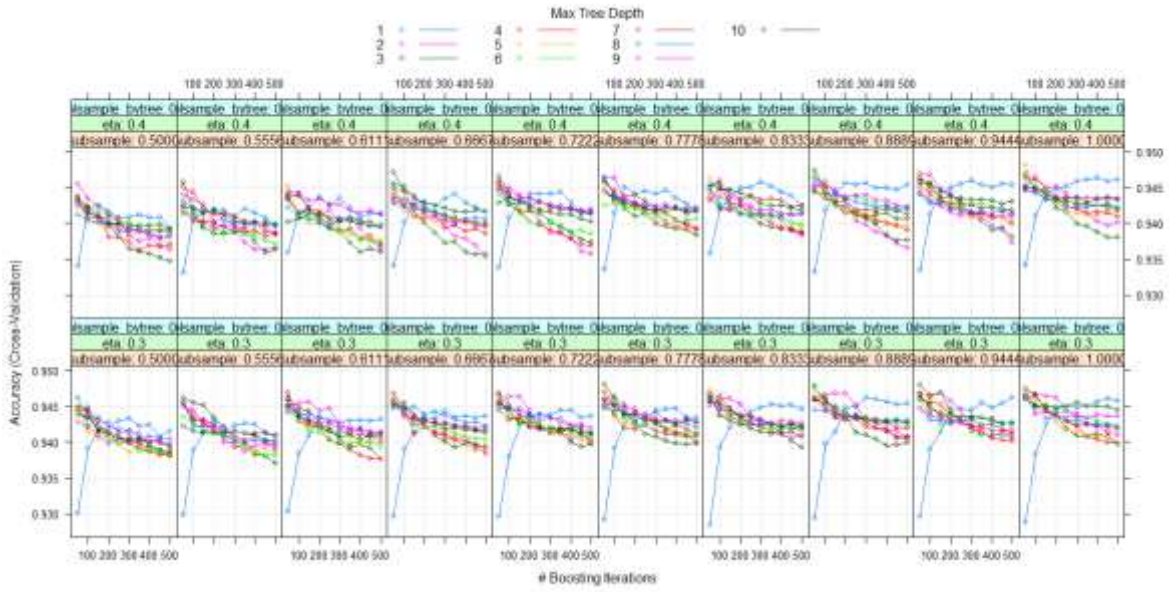


Ilustración 28. Representación de accuracy en Extreme Gradient Boosting

Una vez entrenado el modelo con los datos de entrenamiento se realizan las predicciones para los datos de prueba *x1y1_test* y se obtiene la matriz de confusión. Se puede ver que la precisión de la clasificación ha sido de un 94,45% lo cual es muy buen resultado.

	Categoría						
Predicción	1	2	3	4	5	6	7
1	568	4	2	7	7	6	9
2	11	1028	15	7	5	27	7
3	0	2	332	0	0	13	3
4	2	0	2	170	0	3	2
5	0	0	0	0	205	0	1
6	4	5	8	4	3	393	7
7	6	8	3	1	2	12	674

Tabla 12. Matriz de confusión Extreme Gradient Boosting

5- Predicciones intención de voto.

Una vez que se tiene todos los modelos creados y entrenados con los datos de entrenamiento *x1y1_train* y evaluados con los datos de prueba *x1y1_test*, se procede a realizar el objetivo principal del trabajo que es realizar predicciones de la variable de interés, P9 (Intención de voto), en la submuestra denominada X2Y2. Una vez realizadas dichas predicciones se suman a los valores de la variable P9 en la muestra X1Y1 para obtener los valores totales con sus frecuencias absolutas y se calculan las proporciones en tanto por ciento del total.

A continuación, se realizan estas predicciones con cada modelo creado en los apartados anteriores para posteriormente realizar una comparación tanto con los resultados obtenidos por el CIS como con los resultados de las elecciones generales de noviembre de 2019.

5.1- Predicciones R-part.

Para la realización de las predicciones con el modelo *rpart* se utiliza la función `predict()` y se especifica que los nuevos datos son los datos de la muestra X2Y2 (**Anexo 11**). Se crea una tabla de frecuencias absolutas con la suma de estas predicciones junto con los valores de P9 intención de voto de la muestra X1Y1 y se obtienen las proporciones totales en tanto por ciento.

	PP	***PSOE***	***UP***	***Ciudadanos***	***VOX***	***Otros***	***Abstención***
Predicción X2Y2	562	1407	294	358	91	277	2763
Respuestas X1Y1	1841	3574	1201	716	726	1504	2336
Suma	2403	4981	1495	1074	817	1781	5099
Proporción (%)	13.61	28.22	8.47	6.08	4.62	10.10	28.90

Tabla 13. Predicciones R-part.

Se puede observar en la tabla que, para este modelo, la abstención obtiene una estimación de casi el 29%, el partido más votado sería el PSOE con una estimación del 28,22%, seguido del PP con una estimación del 13,61%, UP obtiene una estimación del 8,5% aproximadamente, Ciudadanos el 6,08% y VOX obtiene una estimación del 4,62%. El resto de los partidos obtienen una estimación de voto del 10,10%.

5.2- Predicciones Random Forest.

Para la realización de las predicciones del modelo *random forest* se utiliza la función `predict()` y se especifica que los nuevos datos son los datos de la muestra X2Y2 (**Anexo 11**). Se crea una tabla de frecuencias absolutas con la suma de estas predicciones junto con los valores de P9 intención de voto de la muestra X1Y1 y se obtienen las proporciones totales en tanto por ciento.

	PP	***PSOE***	***UP***	***Ciudadanos***	***VOX***	***Otros***	***Abstención***
Predicción X2Y2	295	486	102	99	57	564	4149
Respuestas X1Y1	1841	3574	1201	716	726	1504	2336
Suma	2136	4060	1303	815	783	2068	6485
Proporción (%)	12.10	23.00	7.39	4.62	4.43	11.71	36.75

Tabla 14. Predicciones Random Forest.

Se observa en la tabla que, para este modelo, la abstención obtiene una estimación del 36,75%, el partido más votado sería el PSOE con una estimación del 23%, seguido del PP con una estimación del 12,1%. UP obtiene una estimación del 7,4% aproximadamente, Ciudadanos el 4,62% y VOX obtiene una estimación del 4,43%. El resto de los partidos obtienen una estimación de voto del 11,71%.

5.3- Predicciones C.50.

Para la realización de las predicciones del modelo C5.0 se utiliza la función `predict()` y se especifica que los nuevos datos son los datos de la muestra X2Y2 (**Anexo 11**). Se crea una tabla de frecuencias absolutas con la suma de estas predicciones junto con los valores de P9 intención de voto de la muestra X1Y1 y se obtienen las proporciones totales en tanto por ciento.

	PP	***PSOE***	***UP***	***Ciudadanos***	***VOX***	***Otros***	***Abstención***
Predicción X2Y2	290	391	92	111	39	575	4254
Respuestas X1Y1	1841	3574	1201	716	726	1504	2336
Suma	2131	3965	1293	827	765	2079	6590
Proporción (%)	12.07	22.46	7.32	4.70	4.33	11.78	37.34

Tabla 15. Predicciones C.50

Se puede observar en la tabla que, para este modelo, la abstención obtiene una estimación del 37,34%. El partido más votado sería el PSOE con una estimación del 22,46%, seguido del PP con una estimación del 12,1% aproximadamente, UP obtiene una estimación del 7,32%, Ciudadanos el 4,7% y VOX obtiene una estimación del 4,33%. El resto de los partidos obtienen una estimación de voto del 11,8% aproximadamente.

5.4- Predicciones Gradient Boosting.

Para la realización de las predicciones del modelo *gradient boosting* se utiliza la función `predict()` y se especifica que los nuevos datos son los datos de la muestra X2Y2 (**Anexo 11**). Se crea una tabla de frecuencias absolutas con la suma de estas predicciones junto con los valores de P9 intención de voto de la muestra X1Y1 y se obtienen las proporciones totales en tanto por ciento.

	PP	***PSOE***	***UP***	***Ciudadanos***	***VOX***	***Otros***	***Abstención***
Predicción X2Y2	588	2274	353	314	77	283	1863
Respuestas X1Y1	1841	3574	1201	716	726	1504	2336
Suma	2429	5848	1554	1030	803	1787	4199
Proporción (%)	13.77	33.13	8.80	5.83	4.55	10.12	23.80

Tabla 16. Predicciones Gradient Boosting

Observando los resultados en la tabla vemos que, para este modelo, la abstención obtiene una estimación del 23,8%. El partido más votado sería el PSOE con una estimación del 33,13%, seguido del PP con una estimación del 13,77% aproximadamente, UP obtiene una estimación del 8,8%, Ciudadanos el 5,83% y VOX obtiene una estimación del 4,55%. El resto de los partidos obtienen una estimación de voto del 10,12%.

5.5- Predicciones Extreme Gradient Boosting.

Para la realización de las predicciones se utiliza la función `predict()` y se especifica que los nuevos datos son los datos de la muestra X2Y2 (**Anexo 11**). Se crea una tabla de frecuencias absolutas con la suma de estas predicciones junto con los valores de P9 intención de voto de la muestra X1Y1 y se obtienen las proporciones totales en tanto por ciento.

	PP	***PSOE***	***UP***	***Ciudadanos***	***VOX***	***Otros***	***Abstención***
Predicción X2Y2	392	495	84	114	62	468	4137
Respuestas X1Y1	1841	3574	1201	716	726	1504	2336
Suma	2233	4069	1285	830	788	1972	6473
Proporción (%)	12.65	23.05	7.30	4.70	4.46	11.17	36.67

Tabla 17. Predicciones Extreme Gradient Boosting

Se puede observar en la tabla que, para este modelo, la abstención obtiene una estimación del 36,67%. El partido más votado sería el PSOE con una estimación del 23,05%, seguido del PP con una estimación del 12,65% aproximadamente, UP obtiene una estimación del 7,3%, Ciudadanos el 4,7% y VOX obtiene una estimación del 4,46%. El resto de los partidos obtienen una estimación de voto del 11,17%.

6- Comparativa con las estimaciones del CIS y los resultados de las elecciones generales 2019.

Se realiza una comparativa entre los resultados de cada modelo con el fin de analizar los resultados obtenidos en los apartados anteriores. Posteriormente, se realiza una comparativa de los resultados obtenidos por todos los modelos con los obtenidos por el CIS, así como con los resultados de las elecciones generales de noviembre de 2019.

- Comparativa de modelos:

  Se realiza una tabla comparativa en la que figuran los errores y precisión de cada modelo (**Anexo 12**).

Modelo	Error	Accuracy
R-part	0.10622197	0.8938
Random Forest	0.05633408	0.9437
C.50	0.05885650	0.9411
Gradient Boosting	0.13312780	0.8669
Extreme Gradient Boosting	0.05549327	0.9445

Tabla 18. Errores y Accuracy modelos.

Se puede observar en la tabla comparativa que el modelo con el menor error y la mayor precisión es *Extreme Gradient Boosting* aunque Random Forest y C5.0 tienen resultados muy similares.

- Comparativa de resultados de abstención:

  Se comparan los resultados de abstención obtenidos en los modelos creados en apartados anteriores con los resultados de elecciones generales y una estimación de la abstención a partir de las estimaciones del porcentaje de abstención provinciales (CIS, 2019) [6] y de las cifras del censo electoral cerrado a 1 de noviembre de 2019 publicadas por el INE. Se adjunta tabla con los cálculos realizados en el **Anexo 12**.

  Se puede observar en la tabla mostrada a continuación como las predicciones realizadas por los modelos dan un valor de abstención de entre el 29% y el 37% aproximadamente, lo cual concuerda con los resultados obtenidos en las elecciones generales con más del 30% de abstención. El modelo *Gradient Boosting* predice una abstención del 23,8% que concuerda con la estimación del 23,32% obtenida en el **Anexo 12** usando las estimaciones provinciales del CIS.

	Abstención
Rpart	28.90
Random Forest	36.75
C.50	37.34
Gradient Boosting	23.80
Extreme GB	36.67
Resultados Elecciones 19	30.10
Estimación CIS	23.32

Tabla 19. Comparativas modelos con resultados de abstención.

- Comparativa de resultados de los modelos con los resultados del CIS:

  Con objeto de comparar los resultados obtenidos en cada modelo con los resultados del barómetro del CIS, concretamente con los resultados de la estimación de voto CIS en tanto por ciento sobre voto válido (CIS, 2019) [6], se realiza un cálculo (**Anexo 12**) de las predicciones de las categorías de los partidos y se restan las abstenciones.

  El cálculo para la estimación en tanto por ciento sobre voto válido es el siguiente: a las predicciones de las primeras seis categorías se le suman las respuestas de la variable P9 (Intención de voto) para esas categorías y se multiplica por cien. A ese resultado se le divide entre el total de encuestados menos la resta de las predicciones para la abstención y las respuestas de abstención en P9.

  Se realiza una tabla comparativa en donde aparecen las estimaciones de voto de cada partido para cada modelo y los resultados obtenidos por el CIS.

	PP	***PSOE***	***UP***	***Ciudadanos***	***VOX***	***Otros***
Rpart	19.14	39.7	11.91	8.55	6.51	14.19
Random Forest	19.13	36.36	11.67	7.30	7.01	18.52
C.50	19.26	35.85	11.70	7.47	6.91	18.80
Gradient Boosting	18.05	43.47	11.55	7.65	5.97	13.28
Extreme GB	19.97	36.40	11.50	7.42	7.05	17.64
Resultados CIS	18.10	32.20	11.10	10.60	7.90	20.10
Resultados Elecciones 19	20.82	28.00	12.84	6.79	15.09	16,39

Tabla 20. Comparativas modelos con resultados del CIS y resultados elecciones.

Se puede observar en la tabla como los resultados obtenidos para el PP son similares tanto para cada modelo, con valores entre el 18% y 20% aproximadamente, como para los resultados obtenidos por el CIS, con un valor del 18,1%. Concuerdan con los resultados obtenidos por el PP en las elecciones generales, con un 20,82%.
En el caso del PSOE, casi todos los modelos obtienen unas estimaciones de voto válido entre el 35,85% y 36,4% similares al CIS, que obtiene un valor de 32,2%, a excepción de *Gradient Boosting y Rpart* que proporcionan valores más altos para las estimaciones, entre el 39,7% y 43,47%. Los resultados de las elecciones para este partido, el más votado son del 28%, resultados inferiores a los proporcionados por los modelos.
El caso de UP es el que más se acerca entre los modelos y los resultados electorales y del CIS, ya que todos los modelos obtienen unas estimaciones alrededor del 11%, cifra que se aproxima a la obtenida por el CIS con un valor del 11,1% y que se encuentra muy próxima al resultado de las elecciones que fue del 12,84%. Para Ciudadanos el CIS obtiene una estimación del 10,6% cifra más elevada que las estimaciones obtenidas por los modelos que oscilan alrededor del 7%, en este caso las predicciones de los modelos concuerdan mucho con los resultados electorales que fueron del 6,79%. En el caso de VOX, los modelos obtienen unas estimaciones de entre el 6,5% y 7%, mientras que el CIS proporciona un valor del 7.9%, bastante similar a los modelos. En este caso los resultados electorales para este partido fueron más elevados con un valor de 15,09%. En el caso de otros partidos el CIS estima con un 20,1% el voto mientras que en los modelos oscilan entre el 13,3% y el 18,8% aproximadamente. Los resultados de las elecciones para el resto de los partidos fueron del 16,39%, cifra que se aproxima a la proporcionada por algunos modelos.

7- Conclusión.

El objetivo de las encuestas electorales estriba tanto en realizar sondeos para analizar el clima político como para realizar estimaciones de posibles resultados de las elecciones. Se ha podido comprobar como el trabajo de realizar inferencia sobre una muestra con el fin de aplicar conclusiones sobre la población no es sencillo. Ante la imposibilidad de poder realizar una encuesta a toda la población resulta necesario aplicar técnicas estadísticas para resolver el problema. Uno de los problemas fundamentales a los que se enfrentan las técnicas de inferencia estadística es el tratamiento e imputación de los valores faltantes, ¿cómo podemos imputar los valores de aquellas personas encuestadas que no han sabido o no han querido responder a la pregunta objetivo del estudio? La respuesta a la pregunta anterior no es trivial y el CIS lleva años trabajando al respecto con el objetivo de mejorar correcciones de estimación de voto que mejoren sus resultados.

En este trabajo se ha implementado un procedimiento con el objetivo de estimar la intención de voto en España. Utilizando algoritmos de aprendizaje supervisado, denominados ensembles, se han obtenido predicciones que a posteriori han resultado efectivas al compararlas con los resultados de las elecciones generales de noviembre del 2019. Exceptuando la estimación para el partido VOX, la cual se predecía con un valor inferior según los modelos creados, las predicciones para el resto de los partidos y la abstención han proporcionado unos valores cercanos a dichos resultados.

La principal desventaja de estos modelos tan flexibles radica en la pérdida de interpretabilidad, ya que muchos de estos modelos actúan como una caja negra en la que no se sabe con certeza lo que está ocurriendo en el interior, a pesar de dar muy buenos resultados en las predicciones. Sin embargo, este inconveniente se compensa con el cálculo de la importancia de las variables en cada modelo. Se ha podido comprobar que variables como voto + simpatía, partido político que considera más cercano a sus ideas, partido político por el que siente más simpatía y preferencia personal como presidente del gobierno son variables que han tenido una mayor relevancia para los modelos a la hora de realizar las predicciones. A pesar de carecer de una interpretabilidad avanzada como pueden dar otros modelos menos flexibles, la compensación en cuanto a la mejora de las predicciones hacen que sean métodos atractivos para resolver este tipo de problemas de clasificación multiclase con imputación de datos faltantes.

Se ha podido mostrar que el empleo de estos métodos de ensemble resultan efectivos a la hora de imputar las respuestas de los indecisos y de los que no quieren responder a la pregunta determinante del estudio, obteniendo unas predicciones que no distan mucho de la realidad observada en los resultados electorales. Por ello, puede resultar interesante en el futuro profundizar en el empleo de estos métodos con fines parecidos.

8- Referencias

[1] Sela. L., 2019. *OkDiario* [En línea]
Disponible en:
https://okdiario.com/espana/masterchef-pinche-cocina-tezanos-no-dio-ni-encuesta-del-10-n-4799290
[Último acceso: 15 de junio 2020].

[2] Ordaz. P., 2019. *El país*. [En línea]
Disponible en:
https://elpais.com/politica/2019/11/11/actualidad/1573464350_919895.html?rel=str_articulo#1592354742891
[Último acceso: 15 de junio 2020].

[3] Garrido. H., 2019. *El mundo*. [En línea]
Disponible en:
https://www.elmundo.es/espana/2019/07/04/5d1d1baafdddff650a8b4570.html
[Último acceso: 16 de junio 2020].

[4] Díaz de Rada. V., Núñez. A., 2009. *Estudio de las incidencias en la investigación con encuesta. El caso de los barómetros del CIS.* Madrid: Centro de Investigaciones Sociológicas.

[5] Escobar. M., *et al*. 2014. *Los pronósticos electorales con encuestas*. Madrid: Centro de Investigaciones Sociológicas.

[6] CIS, 2019. Centro de Investigaciones Sociológicas. [En línea]
Disponible en:
http://www.cis.es/cis/opencm/ES/11_barometros/listadoestimacionintencionvoto.jsp
[Último acceso: 18 de junio 2020].

[7] Escobar. M., 2007. *El análisis de segmentación: técnicas y aplicaciones de los árboles de clasificación.* Madrid: Centro de Investigaciones Sociológicas.

[8] James. G., *et al*. 2017. *An Introduction to Statistical Learning with Applications in R.* New York: Springer.

[9] Kuhn. M., Johnson. K., 2016. Chapter 8: Regression Trees and Rule-Based Models. En: *Applied Predictive Modeling*. New York: Springer

[10] Breiman. L., 2001. *Random Forest*. Berkeley: University of California.

[11] Quinlan. R., Kohavi. R., 1999. *C5.1.3 Decision Tree Discovery.* Sidney: University of New South Wales.

[12] Amat. J., 2017, *Árboles de predicción: bagging, random forest, boosting y C5.0.* [En línea]
Disponible en: https://rpubs.com/Joaquin_AR/255596
[Último acceso: 11 de junio 2020].

[13] Brownlee. J., 2016. *A Gentle Introduction to XGBoost for Applied Machine Learning.* [En línea]
Disponible en : https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
[Último acceso: 17 de junio 2020].

[14] Breiman. L., 1994. *Bagging Predictors.* Berkeley: University of California.

[15] 2019. *El país*. [En línea]
Disponible en: https://resultados.elpais.com/elecciones/2019/generales/congreso/
[Último acceso: 19 de junio 2020].

9- Anexos

- **Anexo 1. Carga de datos y selección de variables.**

```
library(foreign)
cis<-read.spss("3263.sav",use.value.labels=TRUE,
max.value.labels=TRUE, to.data.frame=TRUE)

dim(cis)

[1] 17650   146

library(dplyr)

cis<-cis%>%
select(P1,P3,P4,P9,P9A,P9B,P11,P13,P14,P15,P16,P17,P17A,P18,P21,P23,P2
4,P25,P25A,P26,P27,P28,VOTOSIMG,P17AR,RECUERDO)

# EDA

library(DataExplorer)

create_report(cis)
```

- **Anexo 2. Recodificación.**

```
# CÓDIGO RECODIFICACIÓN

# Variable P9 Intención de voto
cis<-cis%>%
  mutate(P9=
if_else(cis$P9==1,"1",if_else(cis$P9==2,"2",if_else(cis$P9==3,"3",if_e
lse(cis$P9==4,"4",if_else(cis$P9==18,"5",if_else(cis$P9==96,"6",if_els
e(cis$P9==77,"7",if_else(cis$P9==97,"7",if_else(cis$P9==98,"99",if_els
e(cis$P9==94,"99",if_else(cis$P9==99,"99",if_else(cis$P9==5,"3",if_els
e(cis$P9==22,"3",if_else(cis$P9==21,"3",if_else(cis$P9==6,"3","6")))))
))))))))),missing=NULL)

# variable P9A Partido político por el que siente más simpatía.
cis<-cis%>%
 mutate(P9A=
if_else(cis$P9A==1,"1",if_else(cis$P9A==2,"2",if_else(cis$P9A==3,"3",i
f_else(cis$P9A==4,"4",if_else(cis$P9A==18,"5",if_else(cis$P9A==96,"6",
if_else(cis$P9A==77,"7",if_else(cis$P9A==97,"7",if_else(cis$P9A==98,"9
9",if_else(cis$P9A==94,"99",if_else(cis$P9A==99,"99",if_else(cis$P9A==
5,"3",if_else(cis$P9A==22,"3",if_else(cis$P9A==21,"3",if_else(cis$P9A=
=6,"3","6")))))))))))))),missing=NULL)
```

```
# variable P9B Intención de voto alternativo
cis<-cis%>%
 mutate(P9B=
if_else(cis$P9B==1,"1",if_else(cis$P9B==2,"2",if_else(cis$P9B==3,"3",i
f_else(cis$P9B==4,"4",if_else(cis$P9B==18,"5",if_else(cis$P9B==96,"6",
if_else(cis$P9B==77,"7",if_else(cis$P9B==97,"7",if_else(cis$P9B==98,"9
9",if_else(cis$P9B==94,"99",if_else(cis$P9B==99,"99",if_else(cis$P9B==
5,"3",if_else(cis$P9B==22,"3",if_else(cis$P9B==21,"3",if_else(cis$P9B=
=6,"3","6")))))))))))))),missing=NULL)

# variable P13 Partido político que considera más cercano a sus ideas
cis<-cis%>%
 mutate(P13=
if_else(cis$P13==1,"1",if_else(cis$P13==2,"2",if_else(cis$P13==3,"3",i
f_else(cis$P13==4,"4",if_else(cis$P13==18,"5",if_else(cis$P13==96,"6",
if_else(cis$P13==77,"7",if_else(cis$P13==97,"7",if_else(cis$P13==98,"9
9",if_else(cis$P13==94,"99",if_else(cis$P13==99,"99",if_else(cis$P13==
5,"3",if_else(cis$P13==22,"3",if_else(cis$P13==21,"3",if_else(cis$P13=
=6,"3","6")))))))))))))),missing=NULL)

# Variable P14 Partido político que se cree va a ganar las elecciones
cis<-cis%>%
 mutate(P14=
if_else(cis$P14==1,"1",if_else(cis$P14==2,"2",if_else(cis$P14==3,"3",i
f_else(cis$P14==4,"4",if_else(cis$P14==18,"5",if_else(cis$P14==96,"6",
if_else(cis$P14==77,"7",if_else(cis$P14==97,"7",if_else(cis$P14==98,"9
9",if_else(cis$P14==94,"99",if_else(cis$P14==99,"99",if_else(cis$P14==
5,"3",if_else(cis$P14==22,"3",if_else(cis$P14==21,"3",if_else(cis$P14=
=6,"3","6")))))))))))))),missing=NULL)

# Variable P15 Partido político que desearía ganador en las elecciones
cis<-cis%>%
 mutate(P15=
if_else(cis$P15==1,"1",if_else(cis$P15==2,"2",if_else(cis$P15==3,"3",i
f_else(cis$P15==4,"4",if_else(cis$P15==18,"5",if_else(cis$P15==96,"6",
if_else(cis$P15==77,"7",if_else(cis$P15==97,"7",if_else(cis$P15==98,"9
9",if_else(cis$P15==94,"99",if_else(cis$P15==99,"99",if_else(cis$P15==
5,"3",if_else(cis$P15==22,"3",if_else(cis$P15==21,"3",if_else(cis$P15=
=6,"3","6")))))))))))))),missing=NULL)

# Variable P17A Recuerdo de voto en las elecciones generales de abril
de 2019
cis<-cis%>%
 mutate(P17A=
if_else(cis$P17A==1,"1",if_else(cis$P17A==2,"2",if_else(cis$P17A==3,"3
",if_else(cis$P17A==4,"4",if_else(cis$P17A==18,"5",if_else(cis$P17A==9
6,"6",if_else(cis$P17A==77,"7",if_else(cis$P17A==97,"7",if_else(cis$P1
7A==98,"99",if_else(cis$P17A==94,"99",if_else(cis$P17A==99,"99",if_els
e(cis$P17AR==5,"3",if_else(cis$P17AR==22,"3",if_else(cis$P17AR==21,"3"
,if_else(cis$P17AR==6,"3","6")))))))))))))),missing=NULL)
```

```
# Variable P25A Nivel de estudios alcanzado por la persona
entrevistada
cis<-cis%>%
  mutate(P25A=
if_else(cis$P25A==1,"1",if_else(cis$P25A==2,"1",if_else(cis$P25A==3,"2
",if_else(cis$P25A==4,"2",if_else(cis$P25A==5,"2",if_else(cis$P25A==6,
"2",if_else(cis$P25A==7,"2",if_else(cis$P25A==8,"3",if_else(cis$P25A==
9,"3",if_else(cis$P25A==99,"99","2")))))))))),missing=NULL)


# Variable P26  Situación laboral de la persona entrevistada
cis<-cis%>%
  mutate(P26=
if_else(cis$P26==1,"1",if_else(cis$P26==2,"2",if_else(cis$P26==3,"2",i
f_else(cis$P26==4,"3",if_else(cis$P26==5,"3",if_else(cis$P26==6,"4",if
_else(cis$P26==7,"5",if_else(cis$P26==8,"6",if_else(cis$P26==9,"99","6
")))))))),missing=NULL))

# Variables P3 y P4 Valoración de la situación política y económica
general de España
cis<-cis%>%
  mutate(P3=
if_else(cis$P3==1,"1",if_else(cis$P3==2,"2",if_else(cis$P3==3,"3",if_e
lse(cis$P3==4,"4",if_else(cis$P3==5,"5",if_else(cis$P3==8,"99","99")))
)),missing=NULL)

cis<-cis%>%
  mutate(P4=
if_else(cis$P4==1,"1",if_else(cis$P4==2,"2",if_else(cis$P4==3,"3",if_e
lse(cis$P4==4,"4",if_else(cis$P4==5,"5",if_else(cis$P4==8,"99","99")))
)),missing=NULL)

# P21 Clasificación ideológica
cis<-cis%>%
  mutate(P21=
if_else(cis$P21==1,"1",if_else(cis$P21==2,"1",if_else(cis$P21==3,"2",i
f_else(cis$P21==4,"3",if_else(cis$P21==5,"3",if_else(cis$P21==6,"3",if
_else(cis$P21==7,"4",if_else(cis$P21==8,"5",if_else(cis$P21==97,"99","
6")))))))),missing=NULL)

# Variable P17AR Recuerdo de voto en las elecciones generales de abril
de 2019 de los votantes
cis<-cis%>%
 mutate(P17AR=
if_else(cis$P17AR==1,"1",if_else(cis$P17AR==2,"2",if_else(cis$P17AR==3
,"3",if_else(cis$P17AR==4,"4",if_else(cis$P17AR==18,"5",if_else(cis$P1
7AR==96,"6",if_else(cis$P17AR==77,"7",if_else(cis$P17AR==97,"7",if_els
e(cis$P17AR==98,"99",if_else(cis$P17AR==94,"99",if_else(cis$P17AR==99,
"99",if_else(cis$P17AR==5,"3",if_else(cis$P17AR==22,"3",if_else(cis$P1
7AR==21,"3",if_else(cis$P17AR==6,"3","6")))))))))))))),missing=NULL)
```

```r
# Variable RECUERDO  de voto en las elecciones generales de abril de 2019
cis<-cis%>%
 mutate(RECUERDO=
if_else(cis$RECUERDO==1,"1",if_else(cis$RECUERDO==2,"2",if_else(cis$RECUERDO==3,"3",if_else(cis$RECUERDO==4,"4",if_else(cis$RECUERDO==18,"5",if_else(cis$RECUERDO==96,"6",if_else(cis$RECUERDO==77,"7",if_else(cis$RECUERDO==97,"7",if_else(cis$RECUERDO==98,"99",if_else(cis$RECUERDO==94,"99",if_else(cis$RECUERDO==99,"99",if_else(cis$RECUERDO==5,"3",if_else(cis$RECUERDO==22,"3",if_else(cis$RECUERDO==21,"3",if_else(cis$RECUERDO==6,"3","6")))))))))))))),missing=NULL)
```

```r
# Variale VOTSIMG Voto+simpatía en las elecciones generales de noviembre de 2019
cis<-cis%>%
 mutate(VOTOSIMG=
if_else(cis$VOTOSIMG==1,"1",if_else(cis$VOTOSIMG==2,"2",if_else(cis$VOTOSIMG==3,"3",if_else(cis$VOTOSIMG==4,"4",if_else(cis$VOTOSIMG==18,"5",if_else(cis$VOTOSIMG==96,"6",if_else(cis$VOTOSIMG==77,"7",if_else(cis$VOTOSIMG==97,"7",if_else(cis$VOTOSIMG==98,"99",if_else(cis$VOTOSIMG==94,"99",if_else(cis$VOTOSIMG==99,"99",if_else(cis$VOTOSIMG==5,"3",if_else(cis$VOTOSIMG==22,"3",if_else(cis$VOTOSIMG==21,"3",if_else(cis$VOTOSIMG==6,"3","6")))))))))))))),missing=NULL)
```

P3 y P4. Situación económica y política		Recodificación situación económica y política	
Muy buena	1	Muy buena	1
Buena	2	Buena	2
Regular	3	Regular	3
Mala	4	Mala	4
Muy mala	5	Muy mala	5
N.S.	8	N.S/N.C	99
N.C.	9		

P26. Situación laboral		Recodificación Situación laboral	
Trabaja	1	Trabaja	1
Jubilado/a o pensionista (Anteriormente ha trabajado)	2	Pensionista	2
Pensionista (Anteriormente no ha trabajado, sus labores, etc.)	3	Parado	3
Parado/a y ha trabajado antes	4	Estudiante	4
Parado/a y busca si primer empleo	5	Trabajo doméstico no remunerado	5
Estudiante	6	Otra situación	6
Trabajo doméstico no remunerado	7	No sabe /No contesta	99
Otra situación	8		
N.C.	9		

P9. Intención de voto		Recodificación intención de voto	
PP	1	PP	1
PSOE	2	PSOE	2
Podemos	3	Unidas Podemos*	3
Ciudadanos	4	Ciudadanos	4
IU	5	Vox	5
VOX	18	Otros Partidos	6
PACMA	17	Abstención	7
En Comú Podem	6	No sabe / No contesta	99
Compromís	7		
ERC	8		
PDeCAT (JxCat)	9		
En Marea	10		
EAJ-PNV	11		
EH Bildu	12		
CC	13		
UPN	14		
FAC (Foro Asturias)	15		
Nueva Canarias	16		
CUP	19		
Los Verdes	20		
Unidos Podemos	21		
EQUO	22		
Voto blanco	96		
Voto nulo	77		
No votará	97		
No lo tiene decidido aún	94		
N.S.	98		
N.C.	99		

* Unidas Podemos está formado por la coalición de Podemos, EQUO, IU y En Comú Podem.

P21. Clasificación política		Recodificación clasificación política	
Conservador/a	1	Derechas	1
Demócrata cristiano/a	2	Centro	2
Liberal	3	Izquierdas	3
Progresista	4	Comunista	4
Socialdemócrata	5	Nacionalista	5
Socialista	6	Apolítico	99
Comunista	7	Otras	6
Nacionalista	8		
Feminista	9		
Ecologista	10		
Otras respuestas	11		
(NO LEER) Apolítico	97		
N.S.	98		
N.C.	99		

P9A. Partido político por el que siente más simpatía		Recodificación P9A	
PP	1	PP	1
PSOE	2	PSOE	2
Podemos	3	Unidas Podemos	3
Ciudadanos	4	Ciudadanos	4
IU	5	Vox	5
VOX	18	Otros Partidos	6
PACMA	17	Abstención	7
En Comú Podem	6	No sabe / No contesta	99
Compromís	7		
ERC	8		
PDeCAT (JxCat)	9		
En Marea	10		
EAJ-PNV	11		
EH Bildu	12		
CC	13		
UPN	14		
FAC (Foro Asturias)	15		
Nueva Canarias	16		
CUP	19		
Los Verdes	20		
Unidos Podemos	21		
EQUO	22		
Voto blanco	96		
Voto nulo	77		
No votará	97		
No lo tiene decidido aún	94		
N.S.	98		
N.C.	99		

P9B. Intención de voto alternativo		Recodificación P9B	
PP	1	PP	1
PSOE	2	PSOE	2
Podemos	3	Unidas Podemos	3
Ciudadanos	4	Ciudadanos	4
IU	5	Vox	5
VOX	18	Otros Partidos	6
PACMA	17	Abstención	7
En Comú Podem	6	No sabe / No contesta	99
Compromís	7		
ERC	8		
PDeCAT (JxCat)	9		
En Marea	10		
EAJ-PNV	11		
EH Bildu	12		
CC	13		
UPN	14		
FAC (Foro Asturias)	15		
Nueva Canarias	16		
CUP	19		
Los Verdes	20		
Unidos Podemos	21		
EQUO	22		
Voto blanco	96		
Voto nulo	77		
No votará	97		
No lo tiene decidido aún	94		
N.S.	98		
N.C.	99		

P13. Partido político que considera más cercano a sus ideas		Recodificación P13	
PP	1	PP	1
PSOE	2	PSOE	2
Podemos	3	Unidas Podemos	3
Ciudadanos	4	Ciudadanos	4
IU	5	Vox	5
VOX	18	Otros Partidos	6
PACMA	17	Abstención	7
En Comú Podem	6	No sabe / No contesta	99
Compromís	7		
ERC	8		
PDeCAT (JxCat)	9		
En Marea	10		
EAJ-PNV	11		
EH Bildu	12		
CC	13		
UPN	14		
FAC (Foro Asturias)	15		
Nueva Canarias	16		
CUP	19		
Los Verdes	20		
Unidos Podemos	21		
EQUO	22		
Voto blanco	96		
Voto nulo	77		
No votará	97		
No lo tiene decidido aún	94		
N.S.	98		
N.C.	99		

P14. Partido político que se cree va a ganar las elecciones		Recodificación P14	
PP	1	PP	1
PSOE	2	PSOE	2
Podemos	3	Unidas Podemos	3
Ciudadanos	4	Ciudadanos	4
IU	5	Vox	5
VOX	18	Otros Partidos	6
PACMA	17	Abstención	7
En Comú Podem	6	No sabe / No contesta	99
Compromís	7		
ERC	8		
PDeCAT (JxCat)	9		
En Marea	10		
EAJ-PNV	11		
EH Bildu	12		
CC	13		
UPN	14		
FAC (Foro Asturias)	15		
Nueva Canarias	16		
CUP	19		
Los Verdes	20		
Unidos Podemos	21		
EQUO	22		
Voto blanco	96		
Voto nulo	77		
No votará	97		
No lo tiene decidido aún	94		
N.S.	98		
N.C.	99		

P15. Partido político que desearía ganador en las elecciones		Recodificación P15	
PP	1	PP	1
PSOE	2	PSOE	2
Podemos	3	Unidas Podemos	3
Ciudadanos	4	Ciudadanos	4
IU	5	Vox	5
VOX	18	Otros Partidos	6
PACMA	17	Abstención	7
En Comú Podem	6	No sabe / No contesta	99
Compromís	7		
ERC	8		
PDeCAT (JxCat)	9		
En Marea	10		
EAJ-PNV	11		
EH Bildu	12		
CC	13		
UPN	14		
FAC (Foro Asturias)	15		
Nueva Canarias	16		
CUP	19		
Los Verdes	20		
Unidos Podemos	21		
EQUO	22		
Voto blanco	96		
Voto nulo	77		
No votará	97		
No lo tiene decidido aún	94		
N.S.	98		
N.C.	99		

P17A. Recuerdo de voto en las elecciones generales		Recodificación P17A	
PP	1	PP	1
PSOE	2	PSOE	2
Podemos	3	Unidas Podemos	3
Ciudadanos	4	Ciudadanos	4
IU	5	Vox	5
VOX	18	Otros Partidos	6
PACMA	17	Abstención	7
En Comú Podem	6	No sabe / No contesta	99
Compromís	7		
ERC	8		
PDeCAT (JxCat)	9		
En Marea	10		
EAJ-PNV	11		
EH Bildu	12		
CC	13		
UPN	14		
FAC (Foro Asturias)	15		
Nueva Canarias	16		
CUP	19		
Los Verdes	20		
Unidos Podemos	21		
EQUO	22		
Voto blanco	96		
Voto nulo	77		
No votará	97		
No lo tiene decidido aún	94		
N.S.	98		
N.C.	99		

VOTOSIMG. Voto+simpatía		Recodificación VOTOSIMG	
PP	1	PP	1
PSOE	2	PSOE	2
Podemos	3	Unidas Podemos	3
Ciudadanos	4	Ciudadanos	4
IU	5	Vox	5
VOX	18	Otros Partidos	6
PACMA	17	Abstención	7
En Comú Podem	6	No sabe / No contesta	99
Compromís	7		
ERC	8		
PDeCAT (JxCat)	9		
En Marea	10		
EAJ-PNV	11		
EH Bildu	12		
CC	13		
UPN	14		
FAC (Foro Asturias)	15		
Nueva Canarias	16		
CUP	19		
Los Verdes	20		
Unidos Podemos	21		
EQUO	22		
Voto blanco	96		
Voto nulo	77		
No votará	97		
No lo tiene decidido aún	94		
N.S.	98		
N.C.	99		

P.25A Nivel de estudios alcanzado		Recodificación Nivel de estudios	
N.P.	0	Sin estudios/ Primarios*	1
Menos de 5 años de escolarización	1	Secundarios/ Formación Profesional**	2
Primaria (enseñanza primaria o hasta 5º de EGB)	2	Superiores***	3
Formación Profesional inicial	3	No sabe /No contesta	99
Secundaria (Bachillerato Elemental, EGB completa, ESO completa)	4		
Formación Profesional de grado medio (FP I)	5		
Bachillerato/ COU/ PREU (Bachillerato superior, BUP, bachillerato)	6		
Formación Profesional de grado superior (FP II)	7		
Universitarios medios (diplomatura, arquitectura o ingeniería técnica)	8		
Universitarios superiores (licenciatura, grado, máster oficial, doctorado)	9		
Otros estudios no reglados	10		
N.C.	99		

* Formada por respuestas N.P., Menos de 5 años de escolarización y Primaria (enseñanza primaria o hasta 5º de EGB).

** Formada por respuestas Formación Profesional inicial, Secundaria (Bachillerato Elemental, EGB completa, ESO completa), Formación Profesional de grado medio (FP I), Bachillerato/ COU/ PREU (Bachillerato superior, BUP, bachillerato) y Formación Profesional de grado superior (FP II).

***Formada por respuestas Universitarios medios (diplomatura, arquitectura o ingeniería técnica) y Universitarios superiores (licenciatura, grado, máster oficial, doctorado).

- **Anexo 3. Análisis univariante.**

```
# P9. intención de voto

# frecuencia absoluta

(t_P9<-table(cis$P9))
   1    2    3    4    5    6    7   99
1841 3574 1201  716  726 1504 2336 5752

(p_P9<-prop.table(t_P9)*100)
        1         2         3         4         5         6         7
10.430595 20.249292  6.804533  4.056657  4.113314  8.521246 13.235127
       99
32.589235

# gráfico intención de voto
library(ggplot2)

ggplot(cis)+
  geom_bar(aes(x=cis$P9,y=..count../length(cis$P9)*100),fill=
c("Blue", "Red",
"Purple","Orange","Green","Darkgreen","DarkGrey","sandybrown")) +
 scale_x_discrete(labels=c("PP","PSOE","UP","Ciudadanos",
"VOX","Otros","Abstención","N.S / N.C"))+
 labs(x="Partido político", y= "Frecuencia relativa (%)") +
 ggtitle ("Intención de voto encuesta elecciones generales 2019")+
 theme_minimal()
```

```
# P21 clasificación ideológica

# frecuencia absoluta

(t_P21<-table(cis$P21))
   1    2    3    4    5    6   99
3025 1868 5854  289  528 4518 1568

(p_P21<-prop.table(t_P21)*100)
        1         2         3         4         5         6        99
17.138810 10.583569 33.167139  1.637394  2.991501 25.597734  8.883853

# gráfico clasificación ideológica
ggplot(cis)+

geom_bar(aes(x=as.factor(P21),y=..count../length(cis$P21)*100),fill="s
andybrown") +

scale_x_discrete(labels=c("Derechas","Centro","Izquierdas","Comunista"
, "Nacionalista","Otras", "Apolítico"))+
  labs(x="Ideología", y= "Frecuencia relativa (%)" ) +
  ggtitle("Clasificación ideológica")+
  theme_minimal()
```

```r
# nivel de estudios P25A

# frecuencias absolutas
cis$P25A = factor(cis$P25A, labels = c("Sin estudios/Primarios",
"Secundarios/FP", "Superiores","No sabe"))
(t_P25A<-table(cis$P25A))
```

```
Sin estudios/Primarios         Secundarios/FP              Superiores
                  3401                   9717                    3953
               No sabe
                    29
```

```r
(p_P25A<-prop.table(t_P25A)*100)
```

```
Sin estudios/Primarios         Secundarios/FP              Superiores
            19.8888889             56.8245614             23.1169591
               No sabe
             0.1695906
```

```r
# gráfico nivel de estudios
ggplot(cis)+
geom_bar(aes(x=cis$P25A,y=..count../length(cis$P25A)*100),fill="darkgr
een") +
  labs(x="Nivel de estudios", y= "Frecuencia relativa (%)" ) +
  ggtitle("Nivel de estudios")+
  theme_minimal()
```

```r
# P26 Situación laboral


# frecuencias absolutas
cis$P26 = factor(cis$P26, labels = c("Trabaja", "Pensionista",
"Parado","Estudiante","Trabajo doméstico"," Otra situación","NS/NC"))
(t_P26<-table(cis$P26))
```

```
          Trabaja       Pensionista            Parado
Estudiante
             8264              5082              2350
777
Trabajo doméstico    Otra situación             NS/NC
             1055                97                25
```

```r
(p_P26<-prop.table(t_P26)*100)
```

```
          Trabaja       Pensionista            Parado
Estudiante
       46.8215297        28.7932011        13.3144476
4.4022663
Trabajo doméstico    Otra situación             NS/NC
        5.9773371         0.5495751         0.1416431
```

```
# gráfico Situación laboral
ggplot(cis)+
geom_bar(aes(x=cis$P26,y=..count../length(cis$P26)*100),fill="darkred"
) +
  labs(x="Situación laboral", y= "Frecuencia relativa (%)" ) +
  ggtitle("Situación laboral")+
  theme_minimal()
```

```
# P3 y P4. Situación política y económica


# frecuencias relativas
(t_P3<-table(cis$P3))


   1    2    3    4    5   99
  36  261 3151 6412 7488  302

(p_P3<-prop.table(t_P3)*100)


        1         2         3         4         5        99
 0.203966  1.478754 17.852691 36.328612 42.424929  1.711048

(t_P4<-table(cis$P4))


   1    2    3    4    5   99
  24  652 6188 6314 4269  203

(p21_25A<-prop.table(t_P4)*100)


         1          2          3          4          5          99
 0.1359773  3.6940510 35.0594901 35.7733711 24.1869688  1.1501416

# Gráficos situación política y económica

ggplot(cis)+
  geom_bar(aes(x= factor(P3),y =
..count../length(cis$P3)*100),fill="Yellowgreen")+
  scale_x_discrete(labels=c("Muy buena","Buena","Regular","Mala", "Muy
mala","N.S / N.C"))+
  labs(x="Valoración", y= " Frecuencia relativa (%)")+
  ggtitle ("Situación política")+
  theme_minimal()



ggplot(cis)+
  geom_bar(aes(x= factor(P4),y =
..count../length(cis$P21)*100),fill="Yellowgreen")+
  scale_x_discrete(labels=c("Muy buena","Buena","Regular","Mala", "Muy
mala","N.S / N.C"))+
  labs(x="Valoración", y= " Frecuencia relativa (%)")+
  ggtitle ("Situación Económica")+
  theme_minimal()
```

```
# Edad P24

# gráfico histograma edad
ggplot(cis)+
  geom_histogram(aes(x=P24),fill="lightgreen") +
  labs(x="Edad", y= "Recuento" ) +
  ggtitle("Edad")+
  theme_minimal()

# intervalos de edad

cis<-cis%>%
  mutate(P24= if_else(cis$P24>=18&cis$P24<=25,"18-
25",if_else(cis$P24>=26&cis$P24<=35,"25-
35",if_else(cis$P24>=36&cis$P24<=45,"35-
45",if_else(cis$P24>=46&cis$P24<=55,   "45-
55",if_else(cis$P24>=56&cis$P24<=65,"55-
65",if_else(cis$P24>=66&cis$P24<=75,"65-75", "75>")))))),missing=NULL)

(t_P24<-table(cis$P24))
18-25 25-35 35-45 45-55 55-65 65-75   75>
 1541  2306  3247  3363  2906  2405  1882

(p_P24<-prop.table(t_P24)*100)
    18-25     25-35     35-45     45-55     55-65     65-75       75>
 8.730878 13.065156 18.396601 19.053824 16.464589 13.626062 10.662890

# gráfico edad intervalos
ggplot(cis)+
geom_histogram(aes(x=P24,y=..count../length(cis$P24)*100),fill="steelb
lue",stat =   "count") +
  labs(x="Intervalo Edad", y= "Frecuencia relativa (%)" ) +
  ggtitle("Edad")+
  theme_minimal()
```

- **Anexo 4. Análisis bivariante.**

```
# Clasificación ideológica por nivel de estudios

# frecuencia absoluta

(t21_25A<-table(cis$P21,cis$P25A))
     Sin estudios/Primarios Secundarios/FP Superiores No sabe
  1                     794           1530        596       4
  2                     196           1155        504       2
  3                    1050           3133       1472      12
  4                      40            162         81       0
  5                      72            297        144       2
  6                     893           2448        994       7
  99                    356            992        162       2

(p21_25A<-prop.table(t21_25A))
```

```
    Sin estudios/Primarios Secundarios/FP   Superiores      No sabe
  1           0.0464327485   0.0894736842 0.0348538012 0.0002339181
  2           0.0114619883   0.0675438596 0.0294736842 0.0001169591
  3           0.0614035088   0.1832163743 0.0860818713 0.0007017544
  4           0.0023391813   0.0094736842 0.0047368421 0.0000000000
  5           0.0042105263   0.0173684211 0.0084210526 0.0001169591
  6           0.0522222222   0.1431578947 0.0581286550 0.0004093567
  99          0.0208187135   0.0580116959 0.0094736842 0.0001169591
```

```r
# gráfico Clasificación ideológica por nivel de estudios

cis$P25A = factor(cis$P25A, labels = c("Sin estudios/Primarios",
"Secundarios/FP", "Superiores","No sabe"))

ggplot(data = cis) +
  geom_bar(aes(x = as.factor(P21), y =
..count../length(cis$P21)*100),fill="tomato1",na.rm = TRUE)+
ggtitle("Clasificación ideológica por nivel de estudios") +
  labs(x = "Ideología política", y = "Frecuencia relativa (%)") +
  facet_wrap(.~ P25A,scales = "free_x") +


scale_x_discrete(labels =
c("Derechas","Centro","Izquierdas","Comunista",
"Nacionalista","Otras", "Apolítico")) +
  coord_flip()+
  theme_grey()

# Contraste de independencia chi^2
chisq.test(t21_25A)
```

```
    Pearson's Chi-squared test

data:  t21_25A
X-squared = 387.52, df = 18, p-value < 2.2e-16
```

```r
# Clasificación ideológica por edad

# frecuencia absoluta

(t_P21_24<-table(cis$P21,cis$P24))
```

```
    18-25 25-35 35-45 45-55 55-65 65-75  75>
  1   120   261   437   494   466   614  633
  2   235   348   384   404   254   165   78
  3   389   648  1073  1188  1200   849  507
  4    31    42    47    60    58    42    9
  5    50    57   111   110    85    72   43
  6   554   704   867   810   619   501  463
  99  162   246   328   297   224   162  149
```

```
(p_P21_24<-prop.table(t_P21_24)*100)


         18-25     25-35     35-45     45-55     55-65     65-75
75>
  1  0.6798867 1.4787535 2.4759207 2.7988669 2.6402266 3.4787535
3.5864023
  2  1.3314448 1.9716714 2.1756374 2.2889518 1.4390935 0.9348442
0.4419263
  3  2.2039660 3.6713881 6.0793201 6.7308782 6.7988669 4.8101983
2.8725212
  4  0.1756374 0.2379603 0.2662890 0.3399433 0.3286119 0.2379603
0.0509915
  5  0.2832861 0.3229462 0.6288952 0.6232295 0.4815864 0.4079320
0.2436261
  6  3.1388102 3.9886686 4.9121813 4.5892351 3.5070822 2.8385269
2.6232295
  99 0.9178470 1.3937677 1.8583569 1.6827195 1.2691218 0.9178470
0.8441926

ggplot(data = cis) +
  geom_bar(aes(x =
P21,y=..count../length(cis$P21)*100),fill="darkblue",na.rm = TRUE)+
ggtitle("Clasificación ideológica por intervalos de edad") +
  labs(x = "Ideología política", y = "Frecuencia relativa (%)") +
  facet_wrap(.~ P24,scales = "free_x") +
  scale_x_discrete(labels =
c("Derechas","Centro","Izquierdas","Comunista",
"Nacionalista","Otras", "Apolítico")) +
  coord_flip()+
  theme_grey()
```

```
# intención de voto por edad

# frecuencia absoluta

(t_P9_24<-table(cis$P9,cis$P24))


    18-25 25-35 35-45 45-55 55-65 65-75  75>
  1    88   119   232   293   290   394  425
  2   214   347   542   647   749   623  452
  3   185   257   267   220   156    93   23
  4    96   136   190   126    92    45   31
  5    69   148   191   163    78    52   25
  6   138   217   317   312   274   149   97
  7   217   355   487   464   354   245  214
  99  534   727  1021  1138   913   804  615
```

```
(p_P9_24<-prop.table(t_P21_24)*100)
```

```
        18-25     25-35     35-45     45-55     55-65     65-75
75>
  1  0.6798867 1.4787535 2.4759207 2.7988669 2.6402266 3.4787535
3.5864023
  2  1.3314448 1.9716714 2.1756374 2.2889518 1.4390935 0.9348442
0.4419263
  3  2.2039660 3.6713881 6.0793201 6.7308782 6.7988669 4.8101983
2.8725212
  4  0.1756374 0.2379603 0.2662890 0.3399433 0.3286119 0.2379603
0.0509915
  5  0.2832861 0.3229462 0.6288952 0.6232295 0.4815864 0.4079320
0.2436261
  6  3.1388102 3.9886686 4.9121813 4.5892351 3.5070822 2.8385269
2.6232295
  99 0.9178470 1.3937677 1.8583569 1.6827195 1.2691218 0.9178470
0.8441926
```

```
ggplot(data = cis) +
  geom_bar(aes(x = P9,y=..count../length(cis$P9)*100),fill=
"orange",na.rm = TRUE)+   ggtitle("Intención de voto por intervalos de
edad") +
  labs(x = "Partido político", y = "Frecuencia relativa (%)") +
  facet_wrap(.~ P24,shrink = TRUE,scales = "free_x") +
  scale_x_discrete(labels = c("PP","PSOE","UP","Ciudadanos",
"VOX","Otros","Abstención","N.S / N.C")) +
  coord_flip()+
  theme_grey()
```

```
# contraste de independencia chi^2
chisq.test(t_P9_24)
```

```
    Pearson's Chi-squared test

data:  t_P9_24
X-squared = 1284.4, df = 42, p-value < 2.2e-16
```

```
# intención de voto por nivel de estudios
# frecuencia absoluta
(t_P9_25A<-table(cis$P9,cis$P25A))
```

```
     Sin estudios/Primarios Secundarios/FP Superiores No sabe
  1                     521            868        381       2
  2                     893           1868        635       3
  3                      91            715        379       2
  4                      54            413        234       2
  5                      62            519        138       2
  6                     158            808        511       2
  7                     480           1455        331       2
  99                   1142           3071       1344      14
```

```
(p_P9_25A<-prop.table(t_P9_25A)*100)
```

```
    Sin estudios/Primarios Secundarios/FP  Superiores     No sabe
  1             3.04678363     5.07602339  2.22807018  0.01169591
  2             5.22222222    10.92397661  3.71345029  0.01754386
  3             0.53216374     4.18128655  2.21637427  0.01169591
  4             0.31578947     2.41520468  1.36842105  0.01169591
  5             0.36257310     3.03508772  0.80701754  0.01169591
  6             0.92397661     4.72514620  2.98830409  0.01169591
  7             2.80701754     8.50877193  1.93567251  0.01169591
  99            6.67836257    17.95906433  7.85964912  0.08187135
```

```
ggplot(data = cis) +
  geom_bar(aes(x = P9,y=..count../length(cis$P9)*100),fill=
"yellowgreen",na.rm = TRUE)+   ggtitle("Intención de voto por nivel de
estudios") +
  labs(x = "Partido político", y = "Frecuencia relativa (%)") +
  facet_wrap(.~ P25A,shrink = TRUE,scales = "free_x") +
  scale_x_discrete(labels = c("PP","PSOE","UP","Ciudadanos",
"VOX","Otros","Abstención","N.S / N.C")) +
  coord_flip()+
  theme_grey()

# contraste de independencia chi^2
chisq.test(t_P9_25A)
```

```
    Pearson's Chi-squared test

data:  t_P9_25A
X-squared = 753.45, df = 21, p-value < 2.2e-16
```

```
# clasificación ideológica por situación laboral
# frecuencia absoluta
(t_P21_26<-table(cis$P21,cis$P26))
```

```
    Trabaja Pensionista Parado Estudiante Trabajo doméstico
  1    1141        1284    249         70               261
  2    1053         353    272        119                57
  3    2771        1765    753        224               311
  4     139          84     41         12                11
  5     292         145     54         20                13
  6    2143        1087    682        270               294
  99    725         364    299         62               108

     Otra situación NS/NC
  1              14     6
  2              10     4
  3              26     4
  4               2     0
  5               2     2
  6              33     9
  99             10     0
```

```
(p_P21_26<-prop.table(t_P21_26)*100)
```

```
        Trabaja Pensionista      Parado  Estudiante Trabajo doméstico
  1   6.46458924  7.27478754  1.41076487  0.39660057        1.47875354
  2   5.96600567  2.00000000  1.54107649  0.67422096        0.32294618
  3  15.69971671 10.00000000  4.26628895  1.26912181        1.76203966
  4   0.78753541  0.47592068  0.23229462  0.06798867        0.06232295
  5   1.65439093  0.82152975  0.30594901  0.11331445        0.07365439
  6  12.14164306  6.15864023  3.86402266  1.52974504        1.66572238
  99  4.10764873  2.06232295  1.69405099  0.35127479        0.61189802

     Otra situación       NS/NC
  1      0.07932011  0.03399433
  2      0.05665722  0.02266289
  3      0.14730878  0.02266289
  4      0.01133144  0.00000000
  5      0.01133144  0.01133144
  6      0.18696884  0.05099150
  99     0.05665722  0.00000000
```

```
ggplot(data = cis) +
  geom_bar(aes(x =
P21,y=..count../length(cis$P21)*100),fill="brown",na.rm = TRUE)+
ggtitle("Clasificación ideológica por situación laboral") +
  labs(x = "Ideología política", y = "Frecuencia relativa (%)") +
  facet_wrap(.~ P26,scales = "free_x") +
  scale_x_discrete(labels =
c("Derechas","Centro","Izquierdas","Comunista",
"Nacionalista","Otras", "Apolítico")) +
  coord_flip()+
  theme_grey()
```

```
# contraste de independencia chi^2
chisq.test(t_P21_26)
```

```
    Pearson's Chi-squared test

data:  t_P21_26
X-squared = 705.83, df = 36, p-value < 2.2e-16
```

```
# intención de voto por situación laboral
```

```
(t_P9_26<-table(cis$P9,cis$P26))
```

```
     Trabaja Pensionista Parado Estudiante Trabajo doméstico
  1      624         840    151         51               167
  2     1414        1322    469        104               248
  3      669         190    205        110                25
  4      439         106     99         50                18
  5      450         111    110         30                20
```

```
  6      862          351    172          71                 35
  7     1109          567    420          73                143
  99    2697         1595    724         288                399

      Otra situación NS/NC
  1                7     1
  2               12     5
  3                2     0
  4                3     1
  5                5     0
  6                9     4
  7               20     4
  99              39    10
```

```
(p_P9_26<-prop.table(t_P9_26)*100)
```

```
          Trabaja  Pensionista       Parado   Estudiante Trabajo
doméstico
  1   3.535410765  4.759206799  0.855524079  0.288951841
0.946175637
  2   8.011331445  7.490084986  2.657223796  0.589235127
1.405099150
  3   3.790368272  1.076487252  1.161473088  0.623229462
0.141643059
  4   2.487252125  0.600566572  0.560906516  0.283286119
0.101983003
  5   2.549575071  0.628895184  0.623229462  0.169971671
0.113314448
  6   4.883852691  1.988668555  0.974504249  0.402266289
0.198300283
  7   6.283286119  3.212464589  2.379603399  0.413597734
0.810198300
  99 15.280453258  9.036827195  4.101983003  1.631728045
2.260623229

      Otra situación        NS/NC
  1      0.039660057  0.005665722
  2      0.067988669  0.028328612
  3      0.011331445  0.000000000
  4      0.016997167  0.005665722
  5      0.028328612  0.000000000
  6      0.050991501  0.022662890
  7      0.113314448  0.022662890
  99     0.220963173  0.056657224
```

```
ggplot(data = cis) +
  geom_bar(aes(x = P9,y=..count../length(cis$P9)*100),fill=
"blue",na.rm = TRUE)+   ggtitle("Intención de voto por situación
laboral") +
  labs(x = "Partido político", y = "Frecuencia relativa (%)") +
  facet_wrap(.~ P26,shrink = TRUE,scales = "free_x") +
  scale_x_discrete(labels = c("PP","PSOE","UP","Ciudadanos",
"VOX","Otros","Abstención","N.S / N.C")) +
  coord_flip()+
  theme_grey()
```

```r
# contraste de independencia chi^2
chisq.test(t_P9_26)
```

```
    Pearson's Chi-squared test

data:  t_P9_26
X-squared = 1058.7, df = 42, p-value < 2.2e-16
```

```r
# intención de voto por clasificación ideológica

(t_P9_21<-table(cis$P9,cis$P21))
```

```
       1    2    3    4    5    6   99
  1 1108  233  120    0   29  293   58
  2  151  216 2624   26   15  455   87
  3   17  126  534  145   17  344   18
  4  180  182  175    1    7  129   42
  5  331  103   63    0   60  136   33
  6  107  123  413   43  256  467   95
  7  254  250  428   28   26  709  641
  99 877  635 1497   46  118 1985  594
```

```r
(p_P9_21<-prop.table(t_P9_21)*100)
```

```
                1           2           3           4           5
  1   6.277620397 1.320113314 0.679886686 0.000000000 0.164305949
  2   0.855524079 1.223796034 14.866855524 0.147308782 0.084985836
  3   0.096317280 0.713881020 3.025495751 0.821529745 0.096317280
  4   1.019830028 1.031161473 0.991501416 0.005665722 0.039660057
  5   1.875354108 0.583569405 0.356940510 0.000000000 0.339943343
  6   0.606232295 0.696883853 2.339943343 0.243626062 1.450424929
  7   1.439093484 1.416430595 2.424929178 0.158640227 0.147308782
  99  4.968838527 3.597733711 8.481586402 0.260623229 0.668555241

                6           99
  1   1.660056657 0.328611898
  2   2.577903683 0.492917847
  3   1.949008499 0.101983003
  4   0.730878187 0.237960340
  5   0.770538244 0.186968839
  6   2.645892351 0.538243626
  7   4.016997167 3.631728045
  99 11.246458924 3.365439093
```

```r
cis$P21<-factor(cis$P21, labels =
c("Derechas","Centro","Izquierdas","Comunista",
"Nacionalista","Otras", "Apolítico"))
```

```
ggplot(data = cis) +
  geom_bar(aes(x = P9,y=..count../length(cis$P9)*100),fill=
"gold",na.rm = TRUE)+   ggtitle("Intención de voto por clasificación
ideológica") +
  labs(x = "Partido político", y = "Frecuencia relativa (%)") +
  facet_wrap(.~ P21,shrink = TRUE,scales = "free_x") +
  scale_x_discrete(labels = c("PP","PSOE","UP","Ciudadanos",
"VOX","Otros","Abstención","N.S / N.C")) +
  coord_flip()+
  theme_grey()

# contraste de independencia chi^2
chisq.test(t_P9_21)


    Pearson's Chi-squared test

data:  t_P9_21
X-squared = 9804.9, df = 42, p-value < 2.2e-16
```

- **Anexo 5. Partición muestra y patrones.**

```
# Ponemos NA`s donde haya 99.
cis<-cis%>%
  mutate(P1= na_if(cis$P1,"99"))
cis<-cis%>%
  mutate(P3= na_if(cis$P3,"99"))
cis<-cis%>%
  mutate(P4= na_if(cis$P4,"99"))
cis<-cis%>%
  mutate(P9= na_if(cis$P9,"99"))
cis<-cis%>%
  mutate(P9A= na_if(cis$P9A,"99"))
cis<-cis%>%
  mutate(P9B= na_if(cis$P9B,"99"))
cis<-cis%>%
  mutate(P11= na_if(cis$P11,"99"))
cis<-cis%>%
  mutate(P13= na_if(cis$P13,"99"))
cis<-cis%>%
  mutate(P14= na_if(cis$P14,"99"))
cis<-cis%>%
  mutate(P15= na_if(cis$P15,"99"))
cis<-cis%>%
  mutate(P16= na_if(cis$P16,"99"))
cis<-cis%>%
  mutate(P17= na_if(cis$P17,"99"))
cis<-cis%>%
  mutate(P17A= na_if(cis$P17A,"99"))
cis<-cis%>%
  mutate(P17AR= na_if(cis$P17AR,"99"))
cis<-cis%>%
  mutate(P18= na_if(cis$P18,"99"))
cis<-cis%>%
```

```
  mutate(P21= na_if(cis$P21,"99"))
cis<-cis%>%
  mutate(P23= na_if(cis$P23,"99"))
cis<-cis%>%
  mutate(P24= na_if(cis$P24,"99"))
cis<-cis%>%
  mutate(P25= na_if(cis$P25,"99"))
cis<-cis%>%
  mutate(P25A= na_if(cis$P25A,"99"))
cis<-cis%>%
  mutate(P26= na_if(cis$P26,"99"))
cis<-cis%>%
  mutate(P27= na_if(cis$P27,"99"))
cis<-cis%>%
  mutate(P28= na_if(cis$P28,"99"))
cis<-cis%>%
  mutate(VOTOSIMG= na_if(cis$VOTOSIMG,"99"))
cis<-cis%>%
  mutate(RECUERDO= na_if(cis$RECUERDO,"99"))

# muestra nueva
x<-cis%>%

select(P1,P3,P4,P9A,P9B,P11,P13,P14,P15,P16,P17,P17A,P18,P21,P23,P24,P
25,P25A,P26,P27,P28,VOTOSIMG,P17AR,RECUERDO)

y<-cis%>%
  select(P9)

XY<-data.frame(x,y)

x1y1<- subset(XY,!is.na(P9))
x2y2<- subset(XY,is.na(P9))

# se transforman todos los predictores a numérico para que la división
tenga sentido

x1y1$P1<-as.numeric(x1y1$P1)
x1y1$P3<-as.numeric(x1y1$P3)
x1y1$P4<-as.numeric(x1y1$P4)
x1y1$P9<-as.numeric(x1y1$P9)
x1y1$P9A<-as.numeric(x1y1$P9A)
x1y1$P9B<-as.numeric(x1y1$P9B)
x1y1$P13<-as.numeric(x1y1$P13)
x1y1$P14<-as.numeric(x1y1$P14)
x1y1$P15<-as.numeric(x1y1$P15)
x1y1$P16<-as.numeric(x1y1$P16)
x1y1$P17<-as.numeric(x1y1$P17)
x1y1$P17A<-as.numeric(x1y1$P17A)
x1y1$P21<-as.numeric(x1y1$P21)
x1y1$P23<-as.numeric(x1y1$P23)
x1y1$P25<-as.numeric(x1y1$P25)
x1y1$P25A<-as.numeric(x1y1$P25A)
x1y1$P26<-as.numeric(x1y1$P26)
x1y1$P27<-as.numeric(x1y1$P27)
x1y1$P28<-as.numeric(x1y1$P28)
```

```
x1y1$VOTOSIMG<-as.numeric(x1y1$VOTOSIMG)
x1y1$P17AR<-as.numeric(x1y1$P17AR)
x1y1$RECUERDO<-as.numeric(x1y1$RECUERDO)

x2y2$P1<-as.numeric(x2y2$P1)
x2y2$P3<-as.numeric(x2y2$P3)
x2y2$P4<-as.numeric(x2y2$P4)
x2y2$P9<-as.numeric(x2y2$P9)
x2y2$P9A<-as.numeric(x2y2$P9A)
x2y2$P9B<-as.numeric(x2y2$P9B)
x2y2$P13<-as.numeric(x2y2$P13)
x2y2$P14<-as.numeric(x2y2$P14)
x2y2$P15<-as.numeric(x2y2$P15)
x2y2$P16<-as.numeric(x2y2$P16)
x2y2$P17<-as.numeric(x2y2$P17)
x2y2$P17A<-as.numeric(x2y2$P17A)
x2y2$P21<-as.numeric(x2y2$P21)
x2y2$P23<-as.numeric(x2y2$P23)
x2y2$P25<-as.numeric(x2y2$P25)
x2y2$P25A<-as.numeric(x2y2$P25A)
x2y2$P26<-as.numeric(x2y2$P26)
x2y2$P27<-as.numeric(x2y2$P27)
x2y2$P28<-as.numeric(x2y2$P28)
x2y2$VOTOSIMG<-as.numeric(x2y2$VOTOSIMG)
x2y2$P17AR<-as.numeric(x2y2$P17AR)
x2y2$RECUERDO<-as.numeric(x2y2$RECUERDO)



#  se transforma en NA en las mismas posiciones que hay en las
muestras a predecir para que el patrón sea el mismo en la muestra
x1y1. Las observaciones de x1y1 tendrán patrones de NA similares a
x2y2.

dim(x1y1)

[1] 11898    25

dim(x2y2)

[1] 5752   25
```

```
x1y1[1:5752,-25] <- x1y1[1:5752,-25]*(x2y2[1:5752,-25]/x2y2[1:5752,-
25])
x1y1[5753:(2*5752),-25] <- x1y1[5753:(2*5752),-25]*(x2y2[1:5752,-
25]/x2y2[1:5752,-25])
```

```
# Ponemos 99 en lugar de NA.
x1y1<-x1y1%>%
  mutate(P1= if_else(is.na(x1y1$P1), 99, x1y1$P1))
x1y1<-x1y1%>%
  mutate(P3= if_else(is.na(x1y1$P3), 99, x1y1$P3))
x1y1<-x1y1%>%
  mutate(P4= if_else(is.na(x1y1$P4), 99, x1y1$P4))
x1y1<-x1y1%>%
```

```
  mutate(P9= if_else(is.na(x1y1$P9), 99, x1y1$P9))
x1y1<-x1y1%>%
  mutate(P9A= if_else(is.na(x1y1$P9A), 99, x1y1$P9A))
x1y1<-x1y1%>%
  mutate(P9B= if_else(is.na(x1y1$P9B), 99, x1y1$P9B))
x1y1<-x1y1%>%
  mutate(P11= if_else(is.na(x1y1$P11), 99, x1y1$P11))
x1y1<-x1y1%>%
  mutate(P13= if_else(is.na(x1y1$P13), 99, x1y1$P13))
x1y1<-x1y1%>%
  mutate(P14= if_else(is.na(x1y1$P14), 99, x1y1$P14))
x1y1<-x1y1%>%
  mutate(P15= if_else(is.na(x1y1$P15), 99, x1y1$P15))
x1y1<-x1y1%>%
  mutate(P16= if_else(is.na(x1y1$P16), 99, x1y1$P16))
x1y1<-x1y1%>%
  mutate(P17= if_else(is.na(x1y1$P17), 99, x1y1$P17))
x1y1<-x1y1%>%
  mutate(P17A= if_else(is.na(x1y1$P17A), 99, x1y1$P17A))
x1y1<-x1y1%>%
  mutate(P17AR= if_else(is.na(x1y1$P17AR), 99, x1y1$P17AR))
x1y1<-x1y1%>%
  mutate(P18= if_else(is.na(x1y1$P18), 99, x1y1$P18))
x1y1<-x1y1%>%
  mutate(P21= if_else(is.na(x1y1$P21), 99, x1y1$P21))
x1y1<-x1y1%>%
  mutate(P23= if_else(is.na(x1y1$P23), 99, x1y1$P23))
x1y1<-x1y1%>%
  mutate(P24= if_else(is.na(x1y1$P24), 99, x1y1$P24))
x1y1<-x1y1%>%
  mutate(P25= if_else(is.na(x1y1$P25), 99, x1y1$P25))
x1y1<-x1y1%>%
  mutate(P25A= if_else(is.na(x1y1$P25A), 99, x1y1$P25A))
x1y1<-x1y1%>%
  mutate(P26= if_else(is.na(x1y1$P26), 99, x1y1$P26))
x1y1<-x1y1%>%
  mutate(P27= if_else(is.na(x1y1$P27), 99, x1y1$P27))
x1y1<-x1y1%>%
  mutate(P28= if_else(is.na(x1y1$P28), 99, x1y1$P28))
x1y1<-x1y1%>%
  mutate(VOTOSIMG= if_else(is.na(x1y1$VOTOSIMG), 99, x1y1$VOTOSIMG))
x1y1<-x1y1%>%
  mutate(RECUERDO= if_else(is.na(x1y1$RECUERDO), 99, x1y1$RECUERDO))



x2y2<-x2y2%>%
  mutate(P1= if_else(is.na(x2y2$P1), 99, x2y2$P1))
x2y2<-x2y2%>%
  mutate(P3= if_else(is.na(x2y2$P3), 99, x2y2$P3))
x2y2<-x2y2%>%
  mutate(P4= if_else(is.na(x2y2$P4), 99, x2y2$P4))
x2y2<-x2y2%>%
  mutate(P9= if_else(is.na(x2y2$P9), 99, x2y2$P9))
```

```
x2y2<-x2y2%>%
  mutate(P9A= if_else(is.na(x2y2$P9A), 99, x2y2$P9A))
x2y2<-x2y2%>%
  mutate(P9B= if_else(is.na(x2y2$P9B), 99, x2y2$P9B))
x2y2<-x2y2%>%
  mutate(P11= if_else(is.na(x2y2$P11), 99, x2y2$P11))
x2y2<-x2y2%>%
  mutate(P13= if_else(is.na(x2y2$P13), 99, x2y2$P13))
x2y2<-x2y2%>%
  mutate(P14= if_else(is.na(x2y2$P14), 99, x2y2$P14))
x2y2<-x2y2%>%
  mutate(P15= if_else(is.na(x2y2$P15), 99, x2y2$P15))
x2y2<-x2y2%>%
  mutate(P16= if_else(is.na(x2y2$P16), 99, x2y2$P16))
x2y2<-x2y2%>%
  mutate(P17= if_else(is.na(x2y2$P17), 99, x2y2$P17))
x2y2<-x2y2%>%
  mutate(P17A= if_else(is.na(x2y2$P17A), 99, x2y2$P17A))
x2y2<-x2y2%>%
  mutate(P17AR= if_else(is.na(x2y2$P17AR), 99, x2y2$P17AR))
x2y2<-x2y2%>%
  mutate(P18= if_else(is.na(x2y2$P18), 99, x2y2$P18))
x2y2<-x2y2%>%
  mutate(P21= if_else(is.na(x2y2$P21), 99, x2y2$P21))
x2y2<-x2y2%>%
  mutate(P23= if_else(is.na(x2y2$P23), 99, x2y2$P23))
x2y2<-x2y2%>%
  mutate(P24= if_else(is.na(x2y2$P24), 99, x2y2$P24))
x2y2<-x2y2%>%
  mutate(P25= if_else(is.na(x2y2$P25), 99, x2y2$P25))
x2y2<-x2y2%>%
  mutate(P25A= if_else(is.na(x2y2$P25A), 99, x2y2$P25A))
x2y2<-x2y2%>%
  mutate(P26= if_else(is.na(x2y2$P26), 99, x2y2$P26))
x2y2<-x2y2%>%
  mutate(P27= if_else(is.na(x2y2$P27), 99, x2y2$P27))
x2y2<-x2y2%>%
  mutate(P28= if_else(is.na(x2y2$P28), 99, x2y2$P28))
x2y2<-x2y2%>%
  mutate(VOTOSIMG= if_else(is.na(x2y2$VOTOSIMG), 99, x2y2$VOTOSIMG))
x2y2<-x2y2%>%
  mutate(RECUERDO= if_else(is.na(x2y2$RECUERDO), 99, x2y2$RECUERDO))
```

```
# partimos x1y1 en train y test
set.seed(8)
x1y1_train <- sample_frac(x1y1, 0.7)
x1y1_test <- setdiff(x1y1,x1y1_train)

# pasamos a factor las variables de train, test y x2y2

x1y1_train$P1<-as.factor(x1y1_train$P1)
x1y1_train$P3<-as.factor(x1y1_train$P3)
x1y1_train$P4<-as.factor(x1y1_train$P4)
x1y1_train$P9<-as.factor(x1y1_train$P9)
x1y1_train$P9A<-as.factor(x1y1_train$P9A)
```

```
x1y1_train$P9B<-as.factor(x1y1_train$P9B)
x1y1_train$P13<-as.factor(x1y1_train$P13)
x1y1_train$P14<-as.factor(x1y1_train$P14)
x1y1_train$P15<-as.factor(x1y1_train$P15)
x1y1_train$P16<-as.factor(x1y1_train$P16)
x1y1_train$P17<-as.factor(x1y1_train$P17)
x1y1_train$P17A<-as.factor(x1y1_train$P17A)
x1y1_train$P21<-as.factor(x1y1_train$P21)
x1y1_train$P23<-as.factor(x1y1_train$P23)
x1y1_train$P25<-as.factor(x1y1_train$P25)
x1y1_train$P25A<-as.factor(x1y1_train$P25A)
x1y1_train$P26<-as.factor(x1y1_train$P26)
x1y1_train$P27<-as.factor(x1y1_train$P27)
x1y1_train$P28<-as.factor(x1y1_train$P28)
x1y1_train$VOTOSIMG<-as.factor(x1y1_train$VOTOSIMG)
x1y1_train$P17AR<-as.factor(x1y1_train$P17AR)
x1y1_train$RECUERDO<-as.factor(x1y1_train$RECUERDO)

x1y1_test$P1<-as.factor(x1y1_test$P1)
x1y1_test$P3<-as.factor(x1y1_test$P3)
x1y1_test$P4<-as.factor(x1y1_test$P4)
x1y1_test$P9<-as.factor(x1y1_test$P9)
x1y1_test$P9A<-as.factor(x1y1_test$P9A)
x1y1_test$P9B<-as.factor(x1y1_test$P9B)
x1y1_test$P13<-as.factor(x1y1_test$P13)
x1y1_test$P14<-as.factor(x1y1_test$P14)
x1y1_test$P15<-as.factor(x1y1_test$P15)
x1y1_test$P16<-as.factor(x1y1_test$P16)
x1y1_test$P17<-as.factor(x1y1_test$P17)
x1y1_test$P17A<-as.factor(x1y1_test$P17A)
x1y1_test$P21<-as.factor(x1y1_test$P21)
x1y1_test$P23<-as.factor(x1y1_test$P23)
x1y1_test$P25<-as.factor(x1y1_test$P25)
x1y1_test$P25A<-as.factor(x1y1_test$P25A)
x1y1_test$P26<-as.factor(x1y1_test$P26)
x1y1_test$P27<-as.factor(x1y1_test$P27)
x1y1_test$P28<-as.factor(x1y1_test$P28)
x1y1_test$VOTOSIMG<-as.factor(x1y1_test$VOTOSIMG)
x1y1_test$P17AR<-as.factor(x1y1_test$P17AR)
x1y1_test$RECUERDO<-as.factor(x1y1_test$RECUERDO)




x1y1$P1<-as.factor(x1y1$P1)
x1y1$P3<-as.factor(x1y1$P3)
x1y1$P4<-as.factor(x1y1$P4)
x1y1$P9<-as.factor(x1y1$P9)
x1y1$P9A<-as.factor(x1y1$P9A)
x1y1$P9B<-as.factor(x1y1$P9B)
x1y1$P13<-as.factor(x1y1$P13)
x1y1$P14<-as.factor(x1y1$P14)
x1y1$P15<-as.factor(x1y1$P15)
x1y1$P16<-as.factor(x1y1$P16)
x1y1$P17<-as.factor(x1y1$P17)
x1y1$P17A<-as.factor(x1y1$P17A)
```

```
x1y1$P21<-as.factor(x1y1$P21)
x1y1$P23<-as.factor(x1y1$P23)
x1y1$P25<-as.factor(x1y1$P25)
x1y1$P25A<-as.factor(x1y1$P25A)
x1y1$P26<-as.factor(x1y1$P26)
x1y1$P27<-as.factor(x1y1$P27)
x1y1$P28<-as.factor(x1y1$P28)
x1y1$VOTOSIMG<-as.factor(x1y1$VOTOSIMG)
x1y1$P17AR<-as.factor(x1y1$P17AR)
x1y1$RECUERDO<-as.factor(x1y1$RECUERDO)

x2y2$P1<-as.factor(x2y2$P1)
x2y2$P3<-as.factor(x2y2$P3)
x2y2$P4<-as.factor(x2y2$P4)
x2y2$P9<-as.factor(x2y2$P9)
x2y2$P9A<-as.factor(x2y2$P9A)
x2y2$P9B<-as.factor(x2y2$P9B)
x2y2$P13<-as.factor(x2y2$P13)
x2y2$P14<-as.factor(x2y2$P14)
x2y2$P15<-as.factor(x2y2$P15)
x2y2$P16<-as.factor(x2y2$P16)
x2y2$P17<-as.factor(x2y2$P17)
x2y2$P17A<-as.factor(x2y2$P17A)
x2y2$P21<-as.factor(x2y2$P21)
x2y2$P23<-as.factor(x2y2$P23)
x2y2$P25<-as.factor(x2y2$P25)
x2y2$P25A<-as.factor(x2y2$P25A)
x2y2$P26<-as.factor(x2y2$P26)
x2y2$P27<-as.factor(x2y2$P27)
x2y2$P28<-as.factor(x2y2$P28)
x2y2$VOTOSIMG<-as.factor(x2y2$VOTOSIMG)
x2y2$P17AR<-as.factor(x2y2$P17AR)
x2y2$RECUERDO<-as.factor(x2y2$RECUERDO)
```

- **Anexo 6. R-part.**

```
# rpart
library(rpart)

# modelo con datos de train
cis_arbolx1y1<-rpart(P9 ~. ,data=x1y1_train)
summary(cis_arbolx1y1)
```

```
Call:
rpart(formula = P9 ~ ., data = x1y1_train)
  n= 8329

           CP nsplit rel error    xerror        xstd
1  0.16646562      0 1.0000000 1.0000000 0.007229262
2  0.11752542      2 0.6670688 0.6670688 0.007843910
3  0.09357229      3 0.5495433 0.5495433 0.007644689
4  0.07013614      4 0.4559710 0.4559710 0.007322085
5  0.06703429      5 0.3858349 0.3858349 0.006972471
6  0.06548337      6 0.3188006 0.3119076 0.006486057
7  0.01774944      7 0.2533172 0.2533172 0.005995702
```

```
8  0.01706014      8 0.2355678 0.2426331 0.005894361
9  0.01206273     12 0.1673272 0.1721523 0.005109586
10 0.01085645     13 0.1552645 0.1562985 0.004899087
11 0.01033948     14 0.1444081 0.1511287 0.004827110
12 0.01000000     15 0.1340686 0.1409616 0.004680331

Variable importance
VOTOSIMG      P13      P16 RECUERDO     P17A      P15      P11
P9A
      25       15       12       12       10        7        7
5
   P17AR      P17
       5        2
```

```
# predicción con datos de test
predic_cis_arbolx1y1<-predict(cis_arbolx1y1, newdata = x1y1_test, type
= "class")
head(predic_cis_arbolx1y1,50)
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25
 7  6  3  3  2  3  3  7  2  2  1  1  6  7  2  2  2  1  3  6  2  6  7
7  3
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50
 1  7  7  6  7  6  2  7  6  6  6  3  2  6  7  2  6  2  2  4  4  7  1
2  2
Levels: 1 2 3 4 5 6 7
```

```
# matriz de confusión
library(caret)

confusionMatrix(as.factor(predic_cis_arbolx1y1),as.factor(x1y1_test[["
P9"]]))
```

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4    5    6    7
         1  525    0    0    2    5    2   14
         2   50 1039   39   18   15   72   37
         3    0    2  298    0    0   19    1
         4    1    0    0  158    1    1    4
         5    0    0    0    1  194    0    1
         6    4    0   18    2    0  332    3
         7   11    6    7    8    7   28  643

Overall Statistics

               Accuracy : 0.8938
                 95% CI : (0.8832, 0.9037)
    No Information Rate : 0.2934
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8677
```

```
 Mcnemar's Test P-Value : NA
Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Class: 6
Sensitivity            0.8883   0.9924  0.82320  0.83598  0.87387
0.73128
Specificity            0.9923   0.9084  0.99314  0.99793  0.99940
0.99133
Pos Pred Value         0.9580   0.8181  0.93125  0.95758  0.98980
0.92479
Neg Pred Value         0.9781   0.9965  0.98030  0.99089  0.99170
0.96198
Prevalence             0.1656   0.2934  0.10146  0.05297  0.06222
0.12724
Detection Rate         0.1471   0.2912  0.08352  0.04428  0.05437
0.09305
Detection Prevalence   0.1536   0.3559  0.08969  0.04624  0.05493
0.10062
Balanced Accuracy      0.9403   0.9504  0.90817  0.91695  0.93664
0.86130
                     Class: 7
Sensitivity            0.9147
Specificity            0.9766
Pos Pred Value         0.9056
Neg Pred Value         0.9790
Prevalence             0.1970
Detection Rate         0.1802
Detection Prevalence   0.1990
Balanced Accuracy      0.9456
```

```r
# plot arbol
library(rpart.plot)
rpart.plot(cis_arbolx1y1,type = 1 ,box.palette = "RdYlGn",extra =
104,roundint = TRUE,cex = 0.4,tweak = 1.1,fallen.leaves = TRUE,snip =
TRUE)
```

- **Anexo 7. Random Forest.**

```r
# Random Forest
library(randomForest)

set.seed(8)
modelo_randomforestx1y1 <- randomForest(P9 ~ ., data = x1y1_train,
                                        mtry = 5 , ntree = 500, nodesize =
3,
                                        importance = TRUE)

modelo_randomforestx1y1
```

```
Call:
 randomForest(formula = P9 ~ ., data = x1y1_train, mtry = 5, ntree =
500,      nodesize = 3, importance = TRUE)
               Type of random forest: classification
```

```
                    Number of trees: 500
No. of variables tried at each split: 5

        OOB estimate of  error rate: 5.01%
Confusion matrix:
     1    2   3   4   5   6    7 class.error
1 1204   16   0   6   7  10    7  0.03680000
2   10 2481   9   4   0  10   12  0.01781473
3    1   31 767   2   0  29    9  0.08581645
4   13    9   3 476   2  14   10  0.09677419
5   16    5   0   4 468   2    9  0.07142857
6   21   35  33   9   1 937   14  0.10761905
7    5   21   8   5   0  15 1579  0.03306797
```

```r
# predicciones RF x1y1 test
levels(x1y1_test$P28) <- c(levels(x1y1_test$P28), "7")
predicciones_RFx1y1<-predict(modelo_randomforestx1y1, newdata =
x1y1_test,type = "class")
head(predicciones_RFx1y1,50)
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25
 7  6  3  3  2  6  3  7  2  2  1  1  6  7  2  2  6  7  3  6  2  6  7
7  3
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50
 1  7  7  6  7  6  2  7  6  6  6  3  2  6  7  2  6  1  2  4  4  7  1
2  2
Levels: 1 2 3 4 5 6 7
```

```r
plot(modelo_randomforestx1y1, main = "Modelo Random Forest")
```

```r
confusionMatrix(as.factor(predicciones_RFx1y1),as.factor(x1y1_test[["P
9"]]))
```

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4    5    6    7
         1  574    0    2    5    7    4   10
         2    8 1029   17    7    6   25    8
         3    0    2  327    1    0   18    5
         4    1    0    2  169    1    2    2
         5    0    0    1    0  201    0    2
         6    3   10    9    4    3  394    7
         7    5    6    4    3    4   11  669

Overall Statistics

               Accuracy : 0.9425
                 95% CI : (0.9344, 0.95)
    No Information Rate : 0.2934
    P-Value [Acc > NIR] : < 2.2e-16
```

```
                 Kappa : 0.9292

 Mcnemar's Test P-Value : 3.225e-06

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Class: 6
Sensitivity            0.9712   0.9828  0.90331  0.89418  0.90541
0.8678
Specificity            0.9906   0.9718  0.99189  0.99763  0.99910
0.9884
Pos Pred Value         0.9535   0.9355  0.92635  0.95480  0.98529
0.9163
Neg Pred Value         0.9943   0.9927  0.98911  0.99410  0.99376
0.9809
Prevalence             0.1656   0.2934  0.10146  0.05297  0.06222
0.1272
Detection Rate         0.1609   0.2884  0.09165  0.04737  0.05633
0.1104
Detection Prevalence   0.1687   0.3083  0.09893  0.04961  0.05717
0.1205
Balanced Accuracy      0.9809   0.9773  0.94760  0.94591  0.95225
0.9281
                     Class: 7
Sensitivity            0.9516
Specificity            0.9885
Pos Pred Value         0.9530
Neg Pred Value         0.9881
Prevalence             0.1970
Detection Rate         0.1875
Detection Prevalence   0.1967
Balanced Accuracy      0.9701
```

```r
# importancia de variables para random forest
library(tidyverse)
library(ggpubr)

importancia_pred_RF <-
as.data.frame(importance(modelo_randomforestx1y1, scale = TRUE))
importancia_pred_RF <- rownames_to_column(importancia_pred_RF, var =
"variable")

# Accuracy
RF_accuracy<- ggplot(data=importancia_pred_RF, aes(x=reorder(variable,
MeanDecreaseAccuracy),
                                          y = MeanDecreaseAccuracy,
                                          fill = MeanDecreaseAccuracy))
+
    labs(x = "variable", title = "Reducción de Accuracy") +
    geom_col() +
    coord_flip() +
    theme_bw() +
    theme(legend.position = "bottom")
```

```
# Gini
RF_gini <- ggplot(data = importancia_pred_RF, aes(x =
reorder(variable, MeanDecreaseGini),
                                                  y = MeanDecreaseGini,
                                                  fill = MeanDecreaseGini)) +
    labs(x = "variable", title = "Reducción de pureza (Gini)") +
    geom_col() +
    coord_flip() +
    theme_bw() +
    theme(legend.position = "bottom")

RF_accuracy

RF_gini

# OOB

oob_err_rate <- data.frame(oob_err_rate =
modelo_randomforestx1y1$err.rate[, 1],
                           arboles =
seq_along(modelo_randomforestx1y1$err.rate[, 1]))
ggplot(data = oob_err_rate, aes(x = arboles, y = oob_err_rate)) +
  geom_line() +
  labs(title = "Evolución del out-of-bag-error vs Número árboles",
       x = "Nº árboles")+
  theme_grey()
```

- **Anexo 8. C.50**

```
# C5.0 Clasificación

library(C50)
modelo_C5_x1y1 <- C5.0(as.factor(P9) ~ ., data =x1y1_train,
                  trials = 100,trControl = ctrl,
                  tuneGrid = grid)
modelo_C5_x1y1


Call:
C5.0.formula(formula = as.factor(P9) ~ ., data = x1y1_train, trials =
 100, trControl = ctrl, tuneGrid = grid)

Classification Tree
Number of samples: 8329
Number of predictors: 24

Number of boosting iterations: 100
Average tree size: 184.9

Non-standard options: attempt to group attributes

predicciones_C5_x1y1 <- predict(modelo_C5_x1y1, newdata = x1y1_test,
type = "class")
head(predicciones_C5_x1y1,50)
```

```
 [1] 7 6 3 3 6 6 3 7 2 2 1 1 6 7 2 4 6 7 3 6 2 6 7 7 3 1 7 7 6 7 6 2 7
6 6
[36] 6 3 2 6 7 2 6 1 2 4 4 7 1 2 2
Levels: 1 2 3 4 5 6 7
```

```
confusionMatrix(predicciones_C5_x1y1,as.factor(x1y1_test$P9))
```

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4    5    6    7
         1  571    6    3    4    6    4    6
         2    9 1015   11    6    4   20    9
         3    0    5  329    1    0   12    3
         4    2    1    2  172    3    4    2
         5    0    0    0    0  203    0    2
         6    4   13   12    3    3  399   10
         7    5    7    5    3    3   15  671

Overall Statistics

               Accuracy : 0.9417
                 95% CI : (0.9335, 0.9492)
    No Information Rate : 0.2934
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9283

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Class: 6
Sensitivity            0.9662   0.9694  0.90884  0.91005  0.91441
0.8789
Specificity            0.9903   0.9766  0.99345  0.99586  0.99940
0.9855
Pos Pred Value         0.9517   0.9451  0.94000  0.92473  0.99024
0.8986
Neg Pred Value         0.9933   0.9872  0.98975  0.99497  0.99435
0.9824
Prevalence             0.1656   0.2934  0.10146  0.05297  0.06222
0.1272
Detection Rate         0.1600   0.2845  0.09221  0.04821  0.05689
0.1118
Detection Prevalence   0.1682   0.3010  0.09809  0.05213  0.05746
0.1244
Balanced Accuracy      0.9782   0.9730  0.95114  0.95295  0.95691
0.9322
                     Class: 7
Sensitivity            0.9545
Specificity            0.9867
Pos Pred Value         0.9464
Neg Pred Value         0.9888
Prevalence             0.1970
```

```
Detection Rate         0.1881
Detection Prevalence   0.1987
Balanced Accuracy      0.9706
```

```
(error_c5 <- mean(predicciones_C5_x1y1 != x1y1_test$P9))
```

```
[1] 0.05829596
```

```
# importancia variables c5
library(ggpubr)
library(tidyverse)

importancia_c5_usage <- C5imp(modelo_C5_x1y1, metric = "usage")
importancia_c5_usage <- importancia_c5_usage %>%
                        rownames_to_column(var = "predictor")
importancia_c5_usage
```

```
   predictor Overall
1         P9A  100.00
2         P11  100.00
3         P13  100.00
4         P16  100.00
5    VOTOSIMG  100.00
6         P15   94.24
7    RECUERDO   86.10
8         P21   85.83
9         P28   85.06
10      P17A    82.82
11        P27   75.12
12        P14   74.35
13        P17   71.10
14        P26   66.54
15         P1   59.79
16        P24   59.02
17         P4   55.44
18         P3   54.34
19        P18   53.46
20      P25A    51.77
21        P25   39.20
22        P9B   20.49
23        P23   17.40
24     P17AR    14.67
```

```
importancia_c5_splits <- C5imp(modelo_C5_x1y1, metric = "splits")
importancia_c5_splits <- importancia_c5_splits %>%
                         rownames_to_column(var = "predictor")
importancia_c5_splits
```

```
   predictor    Overall
1         P11 8.70124594
2         P16 8.43039003
3         P13 8.26787649
4         P21 7.13028169
5         P24 6.37188516
6         P15 5.89788732
7         P28 5.44420368
8         P27 4.64517876
```

```
9         P18 4.57746479
10  RECUERDO 4.46235103
11        P17 4.38109426
12       P17A 4.13055255
13        P26 3.95449621
14         P1 3.69041170
15        P14 3.50081257
16         P3 3.27058505
17         P4 3.12161430
18       P25A 2.49187432
19        P9A 2.44447454
20  VOTOSIMG 2.41061755
21        P25 1.90276273
22        P23 0.65682557
23      P17AR 0.08802817
24        P9B 0.02708559
```

```
C5_uso <- ggplot(data = importancia_c5_usage, aes(x =
reorder(predictor, Overall),
                                                  y = Overall, fill =
Overall)) +
    labs(x = "predictor", title = "% Uso predictor") +
    geom_col() +
    coord_flip() +
    theme_bw() +
    theme(legend.position = "bottom")

c5_division <- ggplot(data = importancia_c5_splits, aes(x =
reorder(predictor, Overall),
                                                   y = Overall, fill =
Overall)) +
    labs(x = "predictor", title = "% Divisiones") +
    geom_col() +
    coord_flip() +
    theme_bw() +
    theme(legend.position = "bottom")


C5_uso

c5_division
```

- **Anexo 9 Gradient Boosting.**

```
library(gbm)

Loaded gbm 2.1.5

# Gradient boosting
grad_boost<-gbm(as.factor(P9)~., data = x1y1_train,distribution =
"multinomial",n.trees = 500, shrinkage = 0.001,cv.folds = 5)
summary(grad_boost)
```

```
               var   rel.inf
VOTOSIMG VOTOSIMG 64.772629
P11           P11 18.722765
P16           P16 10.900561
P13           P13  3.211803
P9A           P9A  2.392242
P1             P1  0.000000
P3             P3  0.000000
P4             P4  0.000000
P9B           P9B  0.000000
P14           P14  0.000000
P15           P15  0.000000
P17           P17  0.000000
P17A         P17A  0.000000
P18           P18  0.000000
P21           P21  0.000000
P23           P23  0.000000
P24           P24  0.000000
P25           P25  0.000000
P25A         P25A  0.000000
P26           P26  0.000000
P27           P27  0.000000
P28           P28  0.000000
P17AR       P17AR  0.000000
RECUERDO RECUERDO  0.000000
```

```
grad_boost
```

```
gbm(formula = as.factor(P9) ~ ., distribution = "multinomial",
    data = x1y1_train, n.trees = 500, shrinkage = 0.001, cv.folds = 5)
A gradient boosted model with multinomial loss function.
500 iterations were performed.
The best cross-validation iteration was 500.
There were 24 predictors of which 5 had non-zero influence.
```

```
# mejor iteración con out of bag
best_iter<-gbm.perf(grad_boost,method = "OOB",plot.it =
TRUE,oobag.curve = TRUE)
```

```
best_iter
```

```
[1] 500
attr(,"smoother")
Call:
loess(formula = object$oobag.improve ~ x, enp.target = min(max(4,
    length(x)/10), 50))

Number of Observations: 500
Equivalent Number of Parameters: 39.85
Residual Standard Error: 4.82e-05
```

```
# predicciones x1y1_test
predicciones_gradientboosting<-predict.gbm(grad_boost, newdata =
x1y1_test, n.trees=best_iter,type = "response")
library(Rfast)
```

```
Membership = as.matrix(predicciones_gradientboosting[,,1])
Membership <- rowMaxs(Membership)
table(Membership)

Membership
   1    2    3    4    5    6    7
 566 1388  308  147  180  354  625

library(caret)
confusionMatrix(as.factor(Membership), as.factor(x1y1_test$P9))

Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4    5    6    7
         1  530    0    0    2    5    2   27
         2   52 1037   50   46   41  101   61
         3    0    2  291    0    0    5   10
         4    0    0    0  139    0    0    8
         5    0    0    0    0  174    0    6
         6    3    0   18    1    0  330    2
         7    6    8    3    1    2   16  589

Overall Statistics

               Accuracy : 0.866
                 95% CI : (0.8544, 0.877)
    No Information Rate : 0.2934
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8322

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Class: 6
Sensitivity            0.8968   0.9904  0.80387  0.73545  0.78378
0.72687
Specificity            0.9879   0.8608  0.99470  0.99763  0.99821
0.99229
Pos Pred Value         0.9364   0.7471  0.94481  0.94558  0.96667
0.93220
Neg Pred Value         0.9797   0.9954  0.97822  0.98538  0.98583
0.96142
Prevalence             0.1656   0.2934  0.10146  0.05297  0.06222
0.12724
Detection Rate         0.1485   0.2906  0.08156  0.03896  0.04877
0.09249
Detection Prevalence   0.1586   0.3890  0.08632  0.04120  0.05045
0.09922
Balanced Accuracy      0.9423   0.9256  0.89928  0.86654  0.89100
0.85958
                     Class: 7
Sensitivity            0.8378
Specificity            0.9874
```

```
Pos Pred Value         0.9424
Neg Pred Value         0.9613
Prevalence             0.1970
Detection Rate         0.1651
Detection Prevalence   0.1752
Balanced Accuracy      0.9126
```

```
# importancia variables para gradient boosting

importancia_pred_gb <- summary(grad_boost, plotit = TRUE)

ggplot(data = importancia_pred_gb, aes(x = reorder(var, rel.inf), y =
rel.inf,
                                       fill = rel.inf)) +
  labs(x = "variable", title = "Reducción de MSE") +
  geom_col() +
  coord_flip() +
  theme_bw() +
  theme(legend.position = "bottom")
```

- **Anexo 10. Extreme Gradient Boosting**

```
# extreme gradient boosting

library(gbm)

library(xgboost)

cv_5 = trainControl(method = "cv", number = 5)

set.seed(8)
modelo_xgboost = train(P9 ~ ., data = x1y1_train,
                 method = "xgbTree",
                 trControl = cv_5,
                 verbose = FALSE,
                 tuneLength = 10,na.action = na.pass)

modelo_xgboost
```

```
eXtreme Gradient Boosting

8329 samples
  24 predictor
   7 classes: '1', '2', '3', '4', '5', '6', '7'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 6661, 6664, 6662, 6663, 6666
Resampling results across tuning parameters:
```

```
  eta  max_depth  colsample_bytree  subsample  nrounds  Accuracy
  0.3  1          0.6               0.5000000   50      0.9294030
  0.3  1          0.6               0.5000000  100      0.9380478
  0.3  1          0.6               0.5000000  150      0.9416508
  0.3  1          0.6               0.5000000  200      0.9410497
  0.3  1          0.6               0.5000000  250      0.9422504
  0.3  1          0.6               0.5000000  300      0.9429724
  0.3  1          0.6               0.5000000  350      0.9416524
  0.3  1          0.6               0.5000000  400      0.9430936
  0.3  1          0.6               0.5000000  450      0.9430934
  0.3  1          0.6               0.5000000  500      0.9410523
  0.3  1          0.6               0.5555556   50      0.9295247
  0.3  1          0.6               0.5555556  100      0.9388883
  0.3  1          0.6               0.5555556  150      0.9411703
  0.3  1          0.6               0.5555556  200      0.9420107
  0.3  1          0.6               0.5555556  250      0.9426104
  0.3  1          0.6               0.5555556  300      0.9434523
  0.3  1          0.6               0.5555556  350      0.9432123
  0.3  1          0.6               0.5555556  400      0.9426119
  0.3  1          0.6               0.5555556  450      0.9427319
  0.3  1          0.6               0.5555556  500      0.9421320
  0.3  1          0.6               0.6111111   50      0.9301234
```

...

```
# predicciones x1y1_test
predicciones_xgboost<-predict(modelo_xgboost, newdata = x1y1_test)
summary(predicciones_xgboost)
```

```
   1    2    3    4    5    6    7
 597 1093  356  183  210  422  707
```

```
confusionMatrix(predicciones_xgboost,as.factor(x1y1_test$P9))
```

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4    5    6    7
         1  569    2    2    5    6    5    8
         2   10 1027   15    6    5   26    4
         3    0    3  335    0    1   14    3
         4    2    1    2  173    1    1    3
         5    0    0    0    0  206    1    3
         6    4    6    6    3    2  394    7
         7    6    8    2    2    1   13  675

Overall Statistics

               Accuracy : 0.947
                 95% CI : (0.9392, 0.9541)
    No Information Rate : 0.2934
    P-Value [Acc > NIR] : < 2.2e-16
```

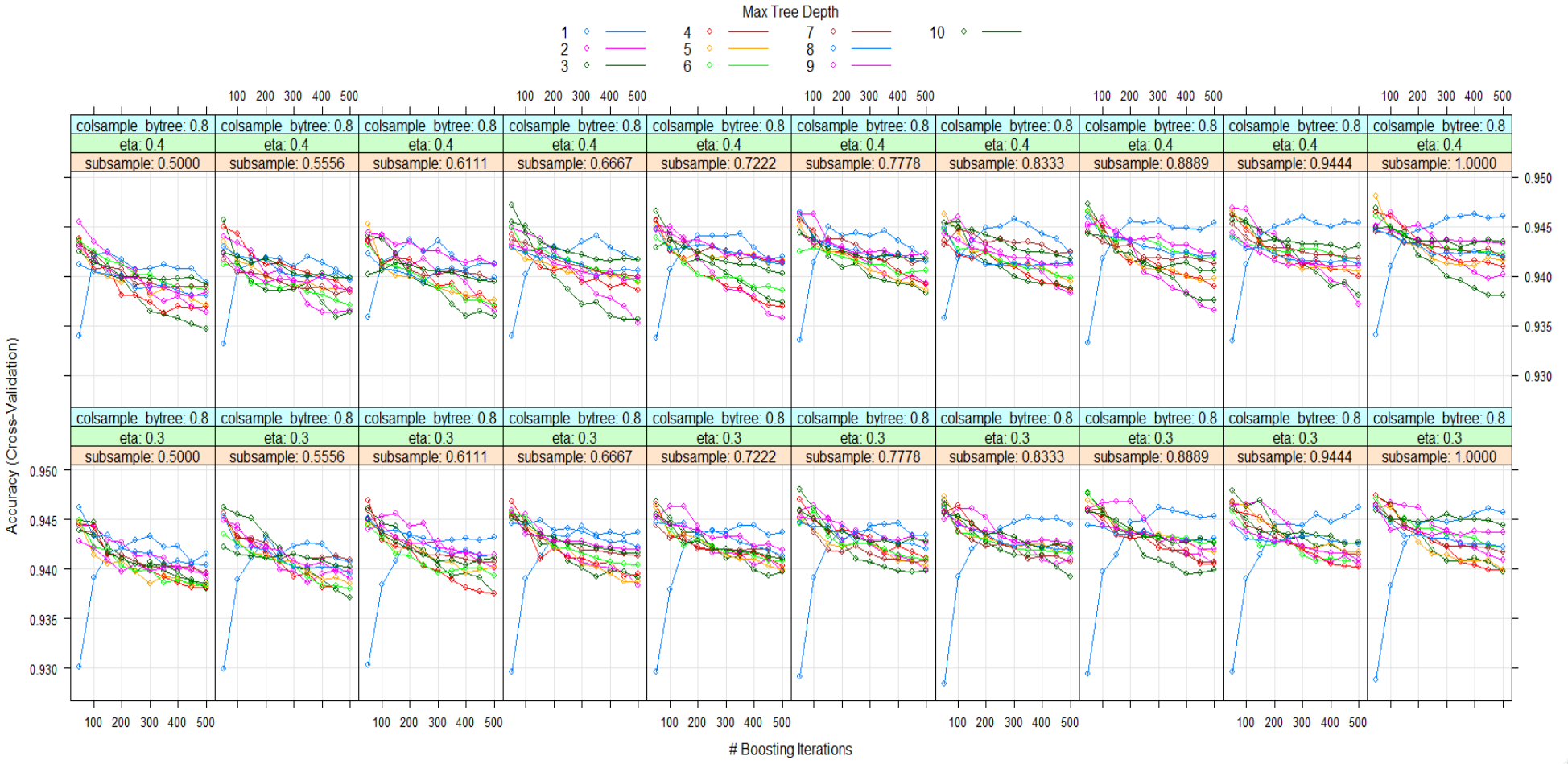
```
                 Kappa : 0.9347

 Mcnemar's Test P-Value : 3.409e-05

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Class: 6
Sensitivity            0.9628   0.9809  0.92541  0.91534  0.92793
0.8678
Specificity            0.9906   0.9738  0.99345  0.99704  0.99880
0.9910
Pos Pred Value         0.9531   0.9396  0.94101  0.94536  0.98095
0.9336
Neg Pred Value         0.9926   0.9919  0.99159  0.99527  0.99524
0.9809
Prevalence             0.1656   0.2934  0.10146  0.05297  0.06222
0.1272
Detection Rate         0.1595   0.2878  0.09389  0.04849  0.05774
0.1104
Detection Prevalence   0.1673   0.3063  0.09978  0.05129  0.05886
0.1183
Balanced Accuracy      0.9767   0.9774  0.95943  0.95619  0.96337
0.9294
                     Class: 7
Sensitivity            0.9602
Specificity            0.9888
Pos Pred Value         0.9547
Neg Pred Value         0.9902
Prevalence             0.1970
Detection Rate         0.1892
Detection Prevalence   0.1982
Balanced Accuracy      0.9745
```

```
plot(modelo_xgboost)
```


Max Tree Depth
1
2
3
4
5
6
7
8
9
10
colsample_bytree: 0.8
eta: 0.4
subsample: 0.5000
subsample: 0.5556
subsample: 0.6111
subsample: 0.6667
subsample: 0.7222
subsample: 0.7778
subsample: 0.8333
subsample: 0.8889
subsample: 0.9444
subsample: 1.0000
eta: 0.3
Accuracy (Cross-Validation)
Boosting Iterations

- **Anexo 11. Predicciones**

```
# predicción Rpart con x2y2


cis_arbolx1y1ALL<-rpart(P9 ~. ,data=x1y1)
predic_cis_arbolx2y2<-predict(cis_arbolx1y1ALL, newdata = x2y2, type =
"class")
head(predic_cis_arbolx2y2,50)

 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25
 7  2  3  6  4  7  3  6  2  6  2  7  7  2  2  2  7  2  7  7  7  2  7
2  7
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50
 2  6  5  2  7  2  2  2  2  2  2  7  7  7  7  1  7  7  7  7  2  7  2
2  7
Levels: 1 2 3 4 5 6 7

summary(predic_cis_arbolx2y2)

   1    2    3    4    5    6    7
 562 1407  294  358   91  277 2763

summary(x1y1$P9)

   1    2    3    4    5    6    7
1841 3574 1201  716  726 1504 2336

(tabla_rpart_predic_total<-
summary(predic_cis_arbolx2y2)+summary(x1y1$P9))

   1    2    3    4    5    6    7
2403 4981 1495 1074  817 1781 5099

(prop_rpart_predic_total<-prop.table(tabla_rpart_predic_total)*100)

        1         2         3         4         5         6         7
13.614731 28.220963  8.470255  6.084986  4.628895 10.090652 28.889518
```

```
# predicciones RF x2y2
levels(x2y2$P9B) <- levels(x1y1$P9B)
modelo_randomforestx1y1ALL <- randomForest(P9 ~ ., data = x1y1,
                                    mtry = 5 , ntree = 500, nodesize =
1,
                                    importance = TRUE)
predicciones_RFx2y2<-predict(modelo_randomforestx1y1ALL, newdata =
x2y2,type = "class")
head(predicciones_RFx2y2,50)

 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25
 7  6  3  6  7  7  7  6  2  6  1  7  7  6  1  2  7  6  7  7  7  6  7
2  7
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50
```

```
 7  6  7  7  7  2  6  2  7  6  2  7  7  7  7  7  7  7  7  7  7  7  7
7  7
Levels: 1 2 3 4 5 6 7

summary(predicciones_RFx2y2)

   1    2    3    4    5    6    7
 295  490   89  103   48  555 4172

summary(x1y1$P9)

   1    2    3    4    5    6    7
1841 3574 1201  716  726 1504 2336

(tabla_RF_predic_total<-summary(predicciones_RFx2y2)+summary(x1y1$P9))

   1    2    3    4    5    6    7
2136 4064 1290  819  774 2059 6508

(prop_RF_predic_total<-prop.table(tabla_RF_predic_total)*100)

        1         2         3         4         5         6         7
12.101983 23.025496  7.308782  4.640227  4.385269 11.665722 36.872521
```

```
# Predicciones C5 x2y2
predicciones_C5_x2y2 <- predict(modelo_C5_x1y1, newdata = x2y2, type =
"class")
head(predicciones_C5_x2y2,50)

 [1] 7 6 6 6 7 7 7 6 2 6 1 7 7 6 1 2 7 6 7 7 7 6 7 2 7 1 6 7 6 7 2 2 2
7 2
[36] 2 7 7 7 7 7 7 7 7 7 7 7 7 7 7
Levels: 1 2 3 4 5 6 7

summary(predicciones_C5_x2y2)

   1    2    3    4    5    6    7
 300  404   90  114   40  558 4246

summary(x1y1$P9)

   1    2    3    4    5    6    7
1841 3574 1201  716  726 1504 2336

(tabla_C5_predic_total<-
summary(predicciones_C5_x2y2)+summary(x1y1$P9))

   1    2    3    4    5    6    7
2141 3978 1291  830  766 2062 6582

(prop_C5_predic_total<-prop.table(tabla_C5_predic_total)*100)

        1         2         3         4         5         6         7
12.130312 22.538244  7.314448  4.702550  4.339943 11.682720 37.291785
```

```r
# predicciones GB x2y2
predicciones_gradientboosting_x2y2<-predict.gbm(grad_boost, newdata =
x2y2, n.trees=best_iter,type = "response")

library(Rfast)
Membership_x2y2 = as.matrix(predicciones_gradientboosting_x2y2[,,1])
Membership_x2y2 <- rowMaxs(Membership_x2y2)

table(Membership_x2y2)

Membership_x2y2
   1    2    3    4    5    6    7
 588 2274  353  319   77  283 1858

summary(x1y1$P9)

   1    2    3    4    5    6    7
1841 3574 1201  716  726 1504 2336

(tabla_GB_predic_total<-table(Membership_x2y2)+summary(x1y1$P9))

Membership_x2y2
   1    2    3    4    5    6    7
2429 5848 1554 1035  803 1787 4194

(prop_GB_predic_total<-prop.table(tabla_GB_predic_total)*100)

Membership_x2y2
        1         2         3         4         5         6         7
13.762040 33.133144  8.804533  5.864023  4.549575 10.124646 23.762040
```

```r
# predicciones XGB x2y2

predicciones_xgboost_x2y2<-predict(modelo_xgboost, newdata = x2y2)

summary(predicciones_xgboost_x2y2)

   1    2    3    4    5    6    7
 383  500   86  122   55  486 4120

summary(x1y1$P9)

   1    2    3    4    5    6    7
1841 3574 1201  716  726 1504 2336

(tabla_XGB_predic_total<-
summary(predicciones_xgboost_x2y2)+summary(x1y1$P9))

   1    2    3    4    5    6    7
2224 4074 1287  838  781 1990 6456

(prop_XGB_predic_total<-prop.table(tabla_XGB_predic_total)*100)

        1         2         3         4         5         6         7
12.600567 23.082153  7.291785  4.747875  4.424929 11.274788 36.577904
```

```
# predicciones estimación de voto sobre voto válido.

# predicciones [1:6] + P9[1:6])*100/(Total de encuestados -
Predicciones[7] - P9[7]).


# Rpart

(tabla_rpart_predic_total_2<-tabla_rpart_predic_total[1:6])

   1    2    3    4    5    6
2403 4981 1495 1074  817 1781

(abstenciones_rpart<-tabla_rpart_predic_total[7])

   7
5099

(prop_total_pred_rpart<-
tabla_rpart_predic_total_2*100/(length(cis$P9)-abstenciones_rpart))

        1         2         3         4         5         6
19.145885 39.686081 11.911401  8.557087  6.509441 14.190104

# Random Forest

(tabla_RF_predic_total_2<-tabla_RF_predic_total[1:6])

   1    2    3    4    5    6
2136 4064 1290  819  774 2059

(abstenciones_RF<-tabla_RF_predic_total[7])

   7
6508

(prop_total_pred_RF<-tabla_RF_predic_total_2*100/(length(cis$P9)-
abstenciones_RF))

        1         2         3         4         5         6
19.170705 36.474601 11.577814  7.350565  6.946688 18.479627

# C5.0

(tabla_C5_predic_total_2<-tabla_C5_predic_total[1:6])

   1    2    3    4    5    6
2141 3978 1291  830  766 2062

(abstenciones_C5<-tabla_C5_predic_total[7])

   7
6582

(prop_total_pred_C5<-tabla_C5_predic_total_2*100/(length(cis$P9)-
abstenciones_C5))

        1         2         3         4         5         6
19.344055 35.941453 11.664257  7.499096  6.920853 18.630286
```

```
# Gradient Boosting

(tabla_GB_predic_total_2<-tabla_GB_predic_total[1:6])

Membership_x2y2
   1    2    3    4    5    6
2429 5848 1554 1035  803 1787

(abstenciones_GB<-tabla_GB_predic_total[7])

   7
4194

(prop_total_pred_GB<-tabla_GB_predic_total_2*100/(length(cis$P9)-
abstenciones_GB))

Membership_x2y2
        1         2         3         4         5         6
18.051427 43.460166 11.548751  7.691736  5.967598 13.280321

# Extreme Gradient Boosting

(tabla_XGB_predic_total_2<-tabla_XGB_predic_total[1:6])

   1    2    3    4    5    6
2224 4074 1287  838  781 1990

(abstenciones_XGB<-tabla_XGB_predic_total[7])

   7
6456

(prop_total_pred_XGB<-tabla_XGB_predic_total_2*100/(length(cis$P9)-
abstenciones_XGB))

        1         2         3         4         5         6
19.867786 36.394497 11.497231  7.486153  6.976952 17.777381
```

- **Anexo 12. Comparativas.**

Oficina del Censo Electoral
Censo electoral de españoles residentes en España (CER)
Número de electores por provincia de inscripción

ESTIMACIÓN DE LA ABSTENCIÓN

ESTIMACIONES DE ABSTENCIÓN DEL CIS 23,32

Provincia de inscripción		Censo cerrado a 1-nov-2019	Provincia	Total Abstenciones	8112543
Total general		**34.785.813**			
02	ALBACETE	301.325	Albacete	0,142	42788
03	ALICANTE/ALACANT	1.233.262	Alicante/Alacant	0,281	346547
04	ALMERÍA	459.782	Almería	0,188	86439
01	ARABA/ÁLAVA	250.872	Araba/Álava	0,127	31861
33	ASTURIAS	850.575	Asturias	0,235	199885
05	ÁVILA	128.059	Ávila	0,223	28557
06	BADAJOZ	541.342	Badajoz	0,173	93652
07	BALEARS, ILLES	779.156	Illes Balears	0,255	198685
08	BARCELONA	3.992.791	Barcelona	0,301	1201830
48	BIZKAIA	907.709	Bizkaia	0,337	305898
09	BURGOS	280.116	Burgos	0,242	67788
10	CÁCERES	326.288	Cáceres	0,179	58406
11	CÁDIZ	971.114	Cádiz	0,16	155378
39	CANTABRIA	461.569	Cantabria	0,339	156472
12	CASTELLÓN/CASTELLÓ	410.266	Castellón/Castelló	0,1	41027
13	CIUDAD REAL	387.435	Ciudad Real	0,124	48042
14	CÓRDOBA	627.470	Córdoba	0,212	133024
15	CORUÑA, A	926.191	Coruña (A)	0,276	255629
16	CUENCA	150.501	Cuenca	0,07	10535
20	GIPUZKOA	554.573	Gipuzkoa	0,302	167481
17	GIRONA	503.937	Girona	0,239	120441
18	GRANADA	703.984	Granada	0,213	149949
19	GUADALAJARA	182.909	Guadalajara	0,32	58531
21	HUELVA	390.160	Huelva	0,397	154894
22	HUESCA	164.654	Huesca	0,104	17124
23	JAÉN	512.088	Jaén	0,343	175646
24	LEÓN	383.071	León	0,265	101514
25	LLEIDA	297.173	Lugo	0,258	76671
27	LUGO	276.130	Lleida	0,187	51636
28	MADRID	4.770.112	Madrid	0,144	686896
29	MÁLAGA	1.141.636	Málaga	0,348	397289
30	MURCIA	1.027.401	Murcia	0,185	190069
31	NAVARRA	483.288	Navarra	0,231	111640
32	OURENSE	258.563	Ourense	0,123	31803
34	PALENCIA	133.398	Palencia	0,205	27347
35	PALMAS, LAS	822.552	Palmas (Las)	0,248	203993
36	PONTEVEDRA	771.780	Pontevedra	0,167	128887
26	RIOJA, LA	233.347	Rioja (La)	0,134	31268
37	SALAMANCA	273.575	Salamanca	0,427	116817
38	SANTA CRUZ DE TENERIFE	756.008	Tenerife	0,221	167078
40	SEGOVIA	115.657	Segovia	0,042	4858
41	SEVILLA	1.510.428	Sevilla	0,302	456149
42	SORIA	69.648	Soria	0,173	12049
43	TARRAGONA	555.663	Tarragona	0,207	115022
44	TERUEL	102.048	Teruel	0,154	15715

45	TOLEDO	513.008	Toledo	0,247	126713
46	VALENCIA/VALÈNCIA	1.902.188	Valencia/València	0,269	511689
47	VALLADOLID	417.845	Valladolid	0,189	78973
49	ZAMORA	147.072	Zamora	0,178	26179
50	ZARAGOZA	712.557	Zaragoza	0,152	108309
51	CEUTA	59.296	Ceuta	0,155	9191
52	MELILLA	54.241	Melilla	0,337	18279

```
# comparativa modelos

(error_rpart <- mean(predic_cis_arbolx1y1 != x1y1_test$P9))

[1] 0.106222

(error_RF <- mean(predicciones_RFx1y1 != x1y1_test$P9))

[1] 0.05745516

(error_c5 <- mean(predicciones_C5_x1y1 != x1y1_test$P9))

[1] 0.05829596

(error_GB <- mean( Membership != x1y1_test$P9))

[1] 0.1339686

(error_XGB <- mean(predicciones_xgboost != x1y1_test$P9))

[1] 0.05297085



accuracy_rpart<- 0.8938
accuracy_RF<-0.9437
accuracy_C5<-0.9411
accuracy_GB<-0.8669
accuracy_XGB<-0.9445


(comparacion_modelos<- data.frame(Errores=
c(error_rpart,error_RF,error_c5,error_GB,error_XGB), Accuracy=
c(accuracy_rpart,accuracy_RF,accuracy_C5,accuracy_GB,accuracy_XGB),row
.names = c("R-part","Random Forest", "C.50","Gradient
Boosting","Extreme Gradient Boosting")))

                           Errores Accuracy
R-part                  0.10622197   0.8938
Random Forest           0.05745516   0.9437
C.50                    0.05829596   0.9411
Gradient Boosting       0.13396861   0.8669
Extreme Gradient Boosting 0.05297085   0.9445

(min_error<-which.min(comparacion_modelos$Errores))

[1] 5

(max_accuracy<-which.max(comparacion_modelos$Accuracy))

[1] 5
```