



DEPARTAMENTO DE
INGENIERIA COMERCIAL
UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

Departamento de Ingeniería Comercial

MBA, Magíster en Gestión Empresarial

ESTUDIO DE PREFACTIBILIDAD BASADO EN UN MODELO DE NEGOCIO PARA EL CUIDADO DE PLANTAS VIVAS

Tesina de Grado presentada por

Nicolás Andrés Rozas Lukaschewsky

Como requisito para optar al grado de

MBA, Magíster en Gestión Empresarial

Guía de Tesina Dr. Zócimo Campos Jaque

Agosto de 2023

TITULO DE TESINA: **“ESTUDIO DE PREFACTIBILIDAD BASADO EN UN
MODELO DE NEGOCIO PARA EL CUIDADO DE
PLANTAS VIVAS”**

AUTOR: **NICOLÁS ANDRÉS ROZAS LUKASCHEWSKY**

TRABAJO DE TESINA, presentando en cumplimiento parcial de los requisitos para el Grado de MBA, Magíster en Gestión Empresarial de la Universidad Técnica Federico Santa María.

OBSERVACIONES: _____

COMISIÓN DE TESINA:

- **Profesor Guía:** Zócimo Campos
- **Correferente Externo:** Lorena Paredes
- **Correferente Interno:** Lionel Valenzuela
- **Ayudante Sala Virtual:** José López

Santiago, agosto 2023

Todo el contenido, análisis, conclusiones y opiniones vertidas en este estudio son de mi exclusiva responsabilidad.

Nombre: NICOLÁS ANDRÉS
ROZAS LUKASCHEWSKY.

Fecha: 03/08/2023

Índice de contenidos

1.	Introducción.....	5
2.	Estado del arte	6
2.1.	Antecedentes Cualitativos y Cuantitativos	6
2.2.	Definición Específica del Tema de Investigación y/o Problemática.....	10
2.3.	Justificación para el Desarrollo de esta Tesis	11
3.	Marco Teórico	12
4.	Modelo de Negocio	27
4.1.	Segmento de clientes.....	27
4.2.	Propuesta de valor	28
4.3.	Canales	29
4.4.	Relación con el cliente	29
4.5.	Flujos de Ingreso.....	30
4.6.	Recursos Clave	30
4.7.	Actividades Clave.....	30
4.8.	Red de Partners	31
4.9.	Estructura de Costos.....	31
5.	Metodología de Investigación.....	32
5.1.	Diseño de Metodología y tipo de investigación	32
5.1.1.	Fuente de recolección de datos.....	34
5.1.2.	Construcción del árbol de decisión	41
5.2.	Análisis de Resultados	52
5.2.1.	Resultados sin ajustes.....	52
5.2.2.	Resultados con ajustes.....	54
6.	Testeo del Modelo.....	62
6.1.	Criterios para la Creación de la Muestra	62
6.2.	Creación de la muestra	64
6.3.	Predicción de la Variable “Cuidados”	65
7.	Conclusiones y Recomendaciones	67
8.	Bibliografía	69
9.	Anexos	71
9.1.	Anexo N°1: Instalación de Librerías Python	71

9.2. Anexo N°2: Configuración para análisis gráfico	72
9.3. Anexo N°3: Parametrización de las respuestas	72
9.4. Anexo N°4: Generación de estadística gráfica.....	78
9.5. Anexo N°5: Evaluación del nivel de exactitud en función de la variable de mantenimiento	79
9.6. Anexo N°6: generación de representación gráfica del árbol de decisión .	80
9.7. Anexo N°7: Generación de Matriz de confusión.....	81
9.8. Anexo N°8: Código de Penalización	82
9.9. Anexo N°9: evaluación de la exactitud en función de la variable Cuidados	83
9.10. Anexo N°10: Código para la Búsqueda Randomizada	84
9.11. Anexo N°11: Generación de la muestra utilizando el muestreo estratificado y jerarquizado sobre los conjuntos de datos	86
9.12. Anexo N°12: código para predecir la variable de pago por el servicio de mantenimiento	90

Índice de Tablas

Tabla 1: Ranking de importaciones de plantas vivas a nivel mundial.....	7
Tabla 2: Distribución de viveros en Chile	8
Tabla 3: Análisis estadístico pregunta de cantidad de plantas en el hogar	45
Tabla 4: Análisis estadístico pregunta por el pago diario de un servicio de cuidado de plantas.....	45
Tabla 5: Análisis estadístico pregunta por el pago diario de un servicio de mantención de plantas	45
Tabla 6: Estadística descriptiva de cantidad de plantas en el hogar	46
Tabla 7: Estadística descriptiva de precio del servicio de cuidados por día.....	46
Tabla 8: Estadística descriptiva de precio por el servicio de mantenimiento de plantas.....	47
Tabla 9: Exactitud del modelo	51
Tabla 10: Exactitud del modelo preajustado.....	55
Tabla 11: Exactitud del modelo ajustado.....	59
Tabla 12: Valores óptimos del modelo ajustado.....	60
Tabla 13: Población Objetivo por edad y sexo	62
Tabla 14: Distribución Porcentual por sexo de la población chilena el 2017	63

Índice de Ilustraciones

Ilustración 1: Clasificación de viviendas según altitud	14
Ilustración 2: Representación de la elevación y el precio por metro cuadrado de las viviendas de Nueva York y San Francisco	15
Ilustración 3: Representación de límites en la elevación y el precio por metro cuadrado de las viviendas de Nueva York y San Francisco.....	16
Ilustración 4: Ordenamiento en función de distintos atributos de las viviendas de Nueva York y San Francisco	17
Ilustración 5: Número de viviendas de Nueva York y San Francisco según elevación	18
Ilustración 6: Comparación en la predicción del número de viviendas de Nueva York o San Francisco según altitud	19
Ilustración 7: Comparación en la predicción del número de viviendas de Nueva York o San Francisco según altitud en otro punto del histograma (0 msnm).....	20
Ilustración 8: Comparación en la predicción del número de viviendas de Nueva York o San Francisco según altitud utilizando el coeficiente de Gini	21
Ilustración 9: Representación gráfica de varios puntos de división para visualizar la distribución de las viviendas de Nueva York y San Francisco	22
Ilustración 10: Mejor punto de división según el algoritmo en un momento dado de las viviendas de Nueva York y San Francisco.....	23
Ilustración 11: Árbol de decisión para clasificar si las viviendas son de Nueva York o San Francisco	24
Ilustración 12: Representación de cómo los datos de las viviendas de Nueva York y San Francisco fluyen a través del árbol de decisión	24
Ilustración 13: Representación de cómo los datos de prueba fluyen a través del árbol de decisión en comparación con los datos de entrenamiento para clasificar las viviendas de Nueva York y San Francisco	25
Ilustración 14: Modelo Canvas	27
Ilustración 15: Cantidad de personas por rango etéreo.....	28
Ilustración 16: Fórmula para calcular el tamaño de la muestra	34
Ilustración 17: Resultados encuesta pregunta Sexo	36
Ilustración 18: Resultados encuesta pregunta Edad	37
Ilustración 19: Resultados encuesta pregunta Estado Civil.....	37
Ilustración 20: Resultados encuesta pregunta Tipo de Vivienda	38
Ilustración 21: Resultados encuesta pregunta empleabilidad	38
Ilustración 22: Resultados encuesta pregunta Tipo de Trabajo.....	39
Ilustración 23: Resultados encuesta pregunta gusto por las plantas.....	39
Ilustración 24: Resultados encuesta pregunta plantas de interior o exterior	39
Ilustración 25: Resultados encuesta pregunta conocimiento sobre plantas	40
Ilustración 26: Resultados encuesta pregunta cantidad de plantas en el hogar	40
Ilustración 27: Etapas para la construcción del árbol de decisión	41

Ilustración 28: Histograma de correlación multivariable	49
Ilustración 29: Árbol de decisión sin ajustes.....	52
Ilustración 30: Matriz de confusión del modelo.....	57
Ilustración 31: Matriz de confusión penalizada.....	58
Ilustración 32: Árbol de decisión del modelo ajustado.....	61
Ilustración 33: Histograma de predicciones hechas por el modelo.....	66

1. Introducción

Existen estudios científicos que evidencian la presencia de más de 740 especies exóticas (que no son originarias del territorio nacional) en Chile, las cuales son, entre otras, el “espinillo o chacai” (*Ulex europaeus*), utilizado como un cerco vivo y/o para el control de las dunas en sectores costeros o la zarza (*Rubus ulmifolius*) y la retamilla (*Teline monspessulana*) las cuales están entre las especies cuya colonización de territorio ha tenido el mayor impacto, desplazando a otras especies. Se ha constatado que 743 especies vegetales exóticas se han establecido en el país según el Laboratorio de Invasiones Biológicas (LIB) de la Universidad de Concepción.

De acuerdo con el primer mapa sobre estas especies alóctonas, publicado en la revista Nature, en el mundo se han encontrado 13.168 variedades que se han asentado en un lugar distinto al de su origen. Es decir, el 3,9% de toda la flora vascular ha colonizado territorios, en la mayoría de los casos, por acción humana, siendo Europa la principal fuente de origen.

Esta introducción de plantas exóticas ha generado un mercado en nuestro país relacionado a la venta de plantas decorativas declaradas como “poco comunes”, las cuales pueden llegar a adquirir valores por sobre los quinientos mil pesos y se comercializan a través de diversos portales WEB o tiendas de plantas especializadas, pudiendo encontrar especies tales como la *Venus Marston* (planta carnívora) o *Stenocarpus* (arbusto decorativo), entre otras. Es justamente en este nicho desde donde se plantea la idea a trabajar en el presente documento, la cual se basa en el planteamiento de un modelo de negocios basado en el cuidado de plantas vivas.

A continuación, se procede con la investigación para alcanzar los objetivos planteados a través del método científico, entregando así conclusiones y/o recomendaciones para el desarrollo del modelo planteado.

2. Estado del arte

2.1. Antecedentes Cualitativos y Cuantitativos

Durante los últimos tres años el comercio mundial de plantas vivas, esquejes e injertos ha fluctuado entre 8.000 y 8.200 millones de dólares según datos del Ministerio de Agricultura a través de ODEPA¹. Las exportaciones de este tipo de productos en el año 2020 alcanzaron 8.189 millones de dólares, lo que significó un alza de 2,4% respecto al año 2019, cuando se registró una leve baja, después del peak alcanzado en el año 2018 (8.223 millones de dólares). La tasa media de crecimiento anual registrada en el gráfico es de 3,2% en estos últimos cuatro años. Si se analizan los últimos diez años de exportación de este tipo de plantas, se concluye que también ha habido una tendencia de alza, aumentando de manera considerable los valores de exportación, como también la diversificación de mercados de destino.

En cuanto a las importaciones, el país que registra la mayor demanda de plantas vivas, esquejes e injertos, es Alemania, con una importación total por un valor de 1.511 millones de dólares durante el año 2020 (-3,2% respecto al año 2019), lo que representa 20% del total importado a nivel mundial de plantas vivas, esquejes e injertos durante el año 2020. Aunque el total importado por Alemania durante el año 2020 fue menor que los valores registrados en el año anterior, las importaciones de este tipo de productos a nivel mundial aumentaron en 4,7% respecto al año 2019:

¹ Oficina de Estudios y Políticas Agrarias

Tabla 1: Ranking de importaciones de plantas vivas a nivel mundial

Nº	País	2016	2017	2018	2019	2020	Variación 2020/2019 (%)	Participación 2020 (%)
1	Alemania	1.373.122	1.467.308	1.705.361	1.561.230	1.511.487	-3,2	20
2	Francia	802.569	797.429	725.617	674.615	692.319	2,6	9,2
3	Países Bajos	533.149	529.481	586.404	586.017	599.754	2,3	7,9
4	Reino Unido	434.442	447.504	488.340	385.787	473.073	22,6	6,3
5	Estados Unidos	338.752	377.402	390.086	378.292	407.062	7,6	5,4
6	Bélgica	301.174	270.498	283.829	262.970	312.753	18,9	4,1
7	Italia	268.562	322.757	345.050	313.452	300.570	-4,1	4
8	Suiza	224.984	234.030	269.964	274.174	280.330	2,2	3,7
9	Austria	229.202	254.894	294.919	283.967	271.431	-4,4	3,6
10	Suecia	170.782	176.461	194.524	187.182	203.325	8,6	2,7
11	Canadá	161.713	165.957	164.296	174.241	185.634	6,5	2,5
80	Chile	1.528	1.341	2.062	2.334	5.919	153,6	0,1
	Los demás	1.847.886	1.921.677	2.150.486	2.136.784	2.317.865	8,5	30,7
	Total	6.687.865	6.966.739	7.600.938	7.221.045	7.561.522	4,7	100

Fuente: elaborado por Odepa a partir de información de Trade Map.

La información disponible muestra que la participación de mercado se ha diversificado y ha aumentado el número de países importadores. Dentro del ranking mundial, los diez primeros países detallados en la tabla anterior registraron un total de 5.052 millones de dólares, que representan 67% del total importado durante el año 2020. Con estas cifras se puede decir que, al igual que en las exportaciones mundiales, las importaciones de plantas vivas, esquejes e injertos presentan una alta concentración de mercado, predominando también la participación de países europeos.

Desde el año 2018 Chile ha avanzado siete puestos dentro de este ordenamiento, posicionándose en el lugar número 80 dentro del ranking mundial de importaciones de plantas. Chile pasó de un total importado de 1,5 millones de dólares (2016) a casi 6 millones de dólares durante el año 2020, con una tasa de crecimiento anual de 40%. Estas cifras dan señales de que Chile está interesado en el ingreso de nuevo material genético, y está atento al comercio internacional de nuevas variedades que se han ido validando en el extranjero, debido a la creciente demanda de material de primera calidad a nivel mundial. Dentro de este escenario, Chile ha asumido el dinamismo que tiene este tipo de industria, y se encuentra trabajando para agilizar el ingreso de nuevo material genético, manteniendo por sobre todas las cosas su excelente condición fito y zoonosanitaria.

En términos generales, las plantas son comercializadas en Chile a través de los viveros, las estadísticas de viveros registrados en el país son manejadas por el Servicio Agrícola y Ganadero (SAG). Durante el año 2020 el SAG registró un total de 2.972 viveros inscritos, lo que refleja una pequeña alza (1,2%) respecto al

número de viveros que existían en el año 2019. Como se detalla en la siguiente tabla:

Tabla 2: Distribución de viveros en Chile

Región	Cantidad	Superficie (ha) ²	Participación (%)
Arica y Parinacota	16	10,9	0,2
Tarapacá	13	4,9	0,1
Antofagasta	12	0,9	0
Atacama	17	s/i	s/i
Coquimbo	132	35,2	0,7
Valparaíso	528	592	11,3
Metropolitana	630	2.000,10	38,1
O'Higgins	351	671	12,8
Maule	558	538	10,2
Bío Bío	295	1.103,00	21
La Araucanía	248	164,6	3,1
Los Ríos	70	21,9	0,4
Los Lagos	75	60,8	1,2
Aysén	13	37,6	0,7
Magallanes	14	13,1	0,2
Total	2.972	5.253,10	100

Fuente: Odepa con información del Servicio Agrícola y Ganadero

La región que concentra un mayor número de viveros inscritos es la Región Metropolitana, con un total de 630 viveros y una superficie total estimada de 2.000 hectáreas (38% de la superficie nacional). En segundo lugar, se encuentra la Región del Maule (558 viveros), seguida por las regiones de Valparaíso (528) y O'Higgins (351). Esta lista de participación varía si se compara el número de hectáreas dedicadas a viveros por región, ya que el tamaño de éstos se diferencia mucho entre una región y otra. En este caso en primer lugar se encuentra siempre la Región Metropolitana, pero la siguen la Región del Bío Bío (con 1.103 hectáreas y 21% de la superficie nacional registrada en 2013), la Región de O'Higgins (con 671 hectáreas y 12,8%) y la Región de Valparaíso (con 592 hectáreas y 11,3%). El dato de superficie es estimado, ya que en algunos casos la inscripción de algunos viveros no especifica con exactitud las hectáreas destinadas a esta actividad.

En términos cualitativos, el presente estudio se centrará principalmente en el análisis de las especies "tropicales" existentes y disponibles en los viveros nacionales, a continuación, se detalla el listado de especies cuyos cuidados se consideran difíciles en función de la necesidad de un ambiente adecuado dadas las condiciones meteorológicas y de suelo de nuestro país:

- **Alocasia:** es un género de plantas con rizoma o bulbo perenne de la familia Araceae. Son nativas de las áreas tropicales y subtropicales de Asia y Australia, aunque hoy se encuentran cultivadas por todo el mundo. En Chile

se conocen comúnmente como manto de Eva o amazónica, no obstante, existen alrededor de 80 especies de Alocasia en el mundo.

- **Colocasia:** Es un género de más de 25 especies de angiospermas de la familia Araceae. Es originario de las regiones tropicales y subtropicales de Asia. Son plantas perennes herbáceas que poseen un gran rizoma sobre o ligeramente bajo la superficie. El tamaño de sus hojas es de entre 20 a 150 cm. de largo, con forma de punta de flecha. Debido a su forma y a su tamaño suelen ser conocidas como orejas de elefante.
- **Strelitziaceae:** son una familia de plantas herbáceas rizomatosas de las regiones tropicales y subtropicales de América y África. La familia fue reconocida por sistemas de clasificación modernos como el sistema de clasificación APG III (20094) y el APWeb, en los que posee 3 géneros y 7 especies. Los miembros de esta familia pueden ser reconocidos por sus hojas, dísticas y de largo pecíolo, y sus inflorescencias rodeadas de una bráctea verde y dura con forma de bote.
- **Plantas variegata:** Es la apariencia de zonas diferentemente coloreadas en las hojas y a veces en el tallo de las plantas, lo que puede ser debido a un número de causas. Una cierta variegación es atractiva y ornamental y los jardineros tienden a preservar estos casos.
- **Calathea:** Es un género de plantas de la familia Marantaceae, nativo de América tropical, principalmente de Brasil y Perú, muchas de las especies son populares como plantas hogareñas o de ornato, algunas conocidas como de la plegaria o cebra.

2.2. Definición Específica del Tema de Investigación y/o Problemática

El tema de investigación a tratar en el presente documento se basa en el supuesto o tesis de la existencia de una necesidad por parte de un grupo de la población que requiere o estaría dispuesta a pagar por un servicio de cuidado de plantas. Para abordar este supuesto, se plantean los siguientes objetivos:

- Como objetivo general se establece el siguiente:
 - Generar un estudio que permita entregar información de una cierta población respecto a su decisión de pago para tomar la decisión de si es factible emprender en el negocio del cuidado de plantas exóticas de interior en el mercado chileno.
- Como objetivos específicos se establecen los siguientes:
 - Determinar a grandes rasgos el modelo de negocio para el servicio ofertado.
 - Establecer una encuesta que permita conocer patrones de comportamiento de una cierta población en torno a las plantas.
 - Elaborar un modelo basado en árboles de decisión para conocer si existe gente o no que invertiría en un servicio determinado.
 - Crear una muestra en base a parámetros recolectados con el fin de testear el modelo y predecir la intención de compra.

2.3. Justificación para el Desarrollo de esta Tesis

A modo de justificación, es pertinente mencionar el alto costo de las nuevas especies de plantas que se han visto en el mercado chileno, los valores fluctúan entre los \$30.000 y \$50.000 pesos chilenos para las plantas de tamaño pequeño y para las plantas ya maduras (más de 30 cm de altura) los valores superan los \$100.000.

También, podemos apreciar que existe una baja competencia en el mercado del cuidado de plantas, los conceptos de plantas exóticas propiamente tal y “Plantsitter” son conceptos relativamente nuevos en el rubro. Se han encontrado alrededor de tres proveedores del servicio, dos de ellos de carácter “amateur” y sólo uno profesional que cubren principalmente la zona central de nuestro país y ofreciendo los servicios de cuidado, mantención y venta de plantas de interior.

Otro punto que refuerza el abordar este mercado, es el aumento en las importaciones de plantas que ha evidenciado nuestro país posterior a la pandemia, los espacios comunes de las casas, departamentos y oficinas cada vez muestran más plantas vivas por sobre los adornos plásticos. Sin mencionar el hecho de la masificación del teletrabajo, el cual ha provocado que las personas pasen más tiempo en su casa y aumentando el consumo de plantas exóticas.

3. Marco Teórico

El presente documento se basa en la teoría de decisión con el objetivo de poder predecir un acto en un contexto determinado por ciertas variables que influyen en él, esto se remonta a los tiempos del diseño de experimentos, Según Stanley (1966), “debemos ensanchar nuestra perspectiva temporal, y reconocer que la experimentación continua y múltiple es más propia de la actividad científica que los experimentos únicos y definitivos”. Además, mencionan en la misma publicación que “Las pruebas que realizamos hoy si llegan a tener éxito, exigirán repetición y validaciones cruzadas en otros momentos y en otras condiciones antes de convertirse en adquisición estable para el acervo científico y ser susceptibles de segura interpretación teórica”. Hoy se sabe la gran importancia que pueden asumir diversas contingencias, correspondientes de la “acción” conjunta de dos variables experimentales.

Los experimentos pueden incluir algunas variables en cualquiera de los dos sentidos o en ambos a la vez. Por ejemplo, incorporando al diseño más de una variable “independiente” (sexo, grado escolar, método con que se enseña aritmética, estilo y tamaño de los tipos de imprenta, etc.) y/o empleando más de una variable “dependiente” (número de errores, velocidad, diversas pruebas, etc.). Los procedimientos de Fisher son multivariados en el primer sentido y univariados en el segundo.

Como fundamento señalamos una concepción evolutiva del conocimiento, según la cual la aplicación práctica y el conocimiento científico son el resultado de la acumulación de ciertas tentativas seleccionadas y remanentes del caudal de observaciones recogidas por la experiencia. Esta concepción inspira gran respeto por la tradición en la práctica pedagógica.

Por ejemplo, cuando Juan Pérez, director de escuela, tiene que decidir entre adoptar un libro de texto modificado o continuar con la versión anterior, es probable que haga su elección fundándose en datos insuficientes. A parte de la eficiencia misma para la enseñanza y el aprendizaje, son muchas las consideraciones que habrá de tomar en cuenta. El director hará lo correcto en una de estas dos formas posibles: Reteniendo el libro antiguo cuando sea tan bueno o mejor que el revisado o adoptando este último cuando sea superior al primero. Pero puede equivocarse también de dos maneras: reteniendo el libro antiguo cuando el nuevo es mejor que el revisado o adoptando este último cuando sea superior al primero. En cada una de las dos elecciones erróneas es de suponer que se producirán inconvenientes diversos, como, por ejemplo:

- Mayor costo financiero y de gasto de energías;
- Costo para el director, en forma de quejas de los maestros, padres y miembros del consejo escolar;
- Costo para los maestros, los alumnos y la sociedad a causa de una peor instrucción.

Estos costos, evaluados en términos de dinero, energía, confusión, menor aprendizaje y mayor riesgo personal, se deben sopesar frente a la probabilidad de que se produzca cada una de dichas alternativas, así como la que se detecte el error mismo.

Otro método similar, es el que plantea Chernoff y Moses (1959) a través de la Teoría de la decisión, en la cual se detallan los factores que atentan contra la validez interna y externa en diseños experimentales:

- Validez Interna: mínima imprescindible, sin la cual es imposible interpretar el modelo, ¿Introducían, en realidad, una diferencia los tratamientos empíricos en este caso experimental concreto?
- Validez externa: plantea la interrogante de la posibilidad de generalización ¿A qué poblaciones, situaciones, variables de tratamiento y variables de medición puede generalizarse este efecto?

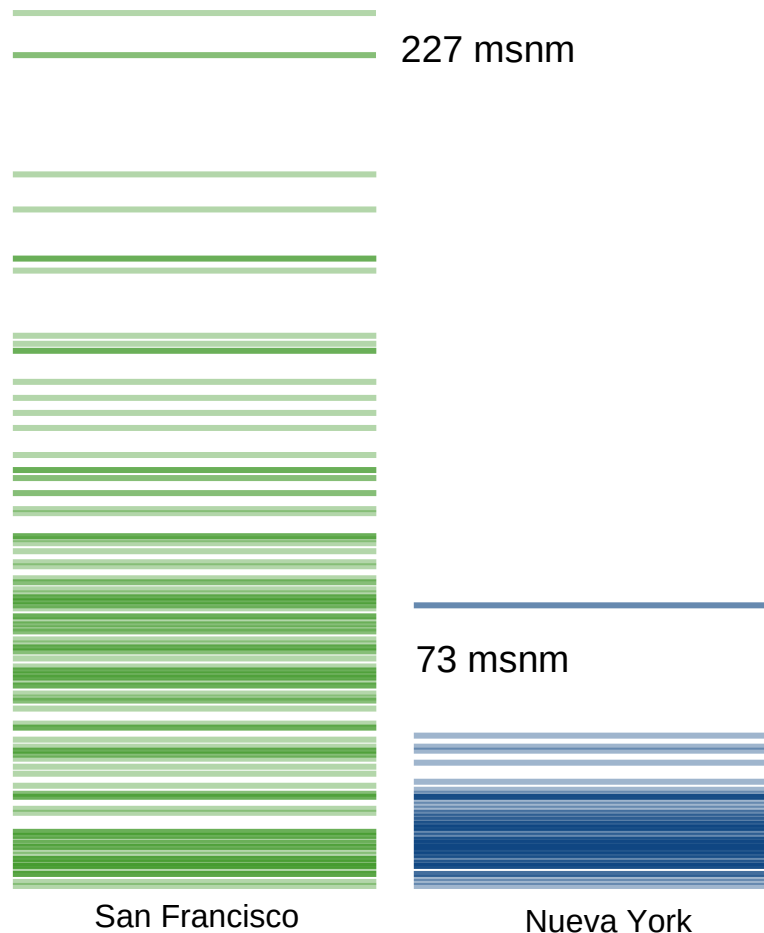
Para ejemplificar de mejor manera la “validez interna” de un experimento, existen distintas variables externas que, de no controlarlas en el diseño experimental, podrían generar efectos que se confundirían con el del estímulo experimental, estas constituyen los efectos de:

- Historia
- Maduración
- Administración de test
- Regresión estadística
- Sesgos resultantes en una selección diferencial de participantes para los grupos de comparación.
- Mortalidad experimental, o diferencia en la pérdida de participantes para los grupos de comparación.
- Interacción entre la selección y la maduración
- Efecto retroactivo o de interacción de las pruebas
- Efectos de interacción de los sesgos de selección y la variable experimental
- Efectos reactivos de los dispositivos experimentales
- Interferencias de los tratamientos múltiples.

Dichos problemas se han ido trabajando en base a las nuevas tecnologías disponibles, estas, por ejemplo, aplicadas a través del *Machine Learning* o aprendizaje de máquina, en el cual los computadores aplican técnicas de aprendizaje estadístico para identificar patrones en los datos de manera automática.

Estas técnicas pueden ser utilizadas para hacer predicciones con una alta precisión. Para ejemplificar esto, utilizaremos la referencia de Stephanie Yee y Tony Chu, quienes en su portal R2D3, publican un ejemplo de cómo el Machine Learning puede ayudarnos a distinguir si una vivienda pertenece a Nueva York o San Francisco, utilizando datos de viviendas como input para crear dicho modelo. Para esto, comienzan con suposiciones, por ejemplo, para el caso en cuestión, en términos del machine learning, categorizar los datos de viviendas de ambas ciudades corresponde a una tarea de clasificación. Si asumimos que la localidad de San Francisco es relativamente montañosa, la elevación de una vivienda podría ser una buena manera de distinguir ambas ciudades. Basado en los datos de elevación de las viviendas, se podría afirmar que una vivienda localizada a más 73 [msnm] debería ser clasificada como una que pertenece a San Francisco.

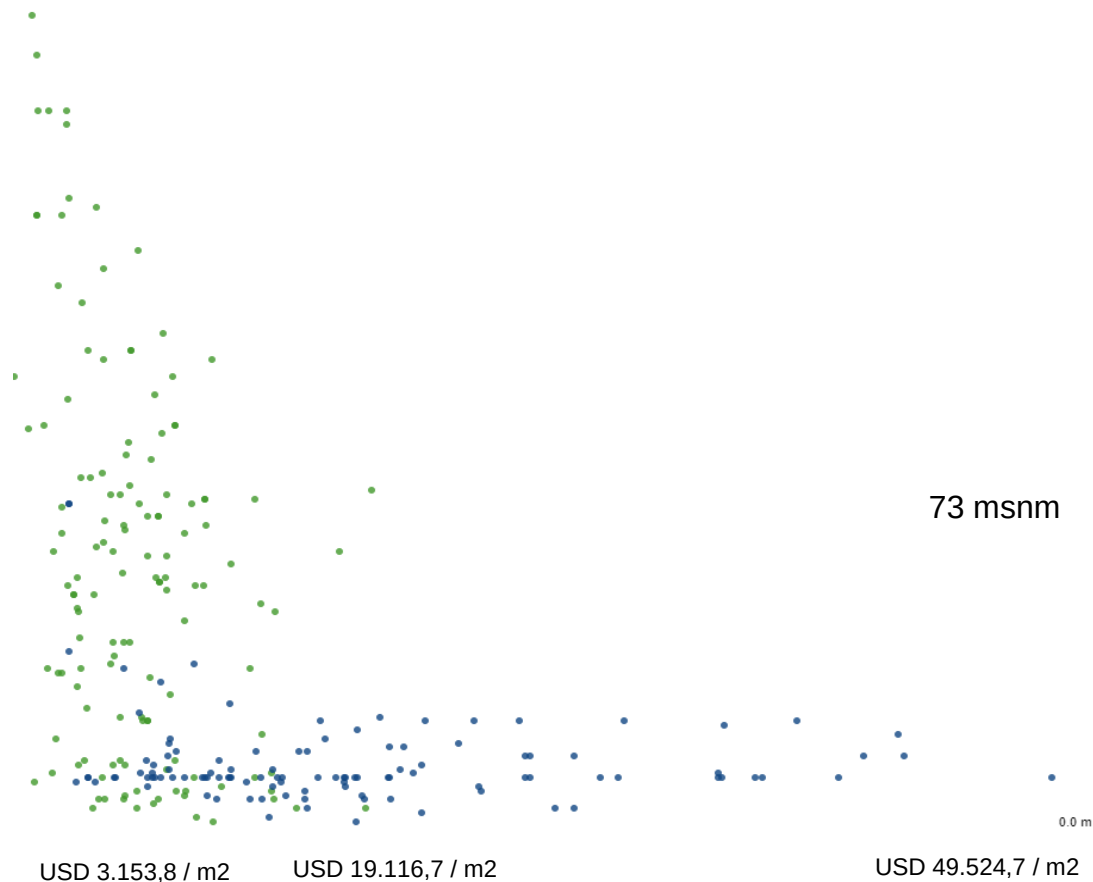
Ilustración 1: Clasificación de viviendas según altitud



Luego, al mismo conjunto de datos, es posible añadirle “dimensiones”, las dimensiones en un conjunto de datos se llaman características, predictores o variables. Por ejemplo, los apartamentos en Nueva York pueden ser

extremadamente caros de acuerdo con su superficie cuadrada. Así que visualizar la elevación y el precio por metro cuadrado en un gráfico de dispersión nos permite distinguir las viviendas con menor elevación:

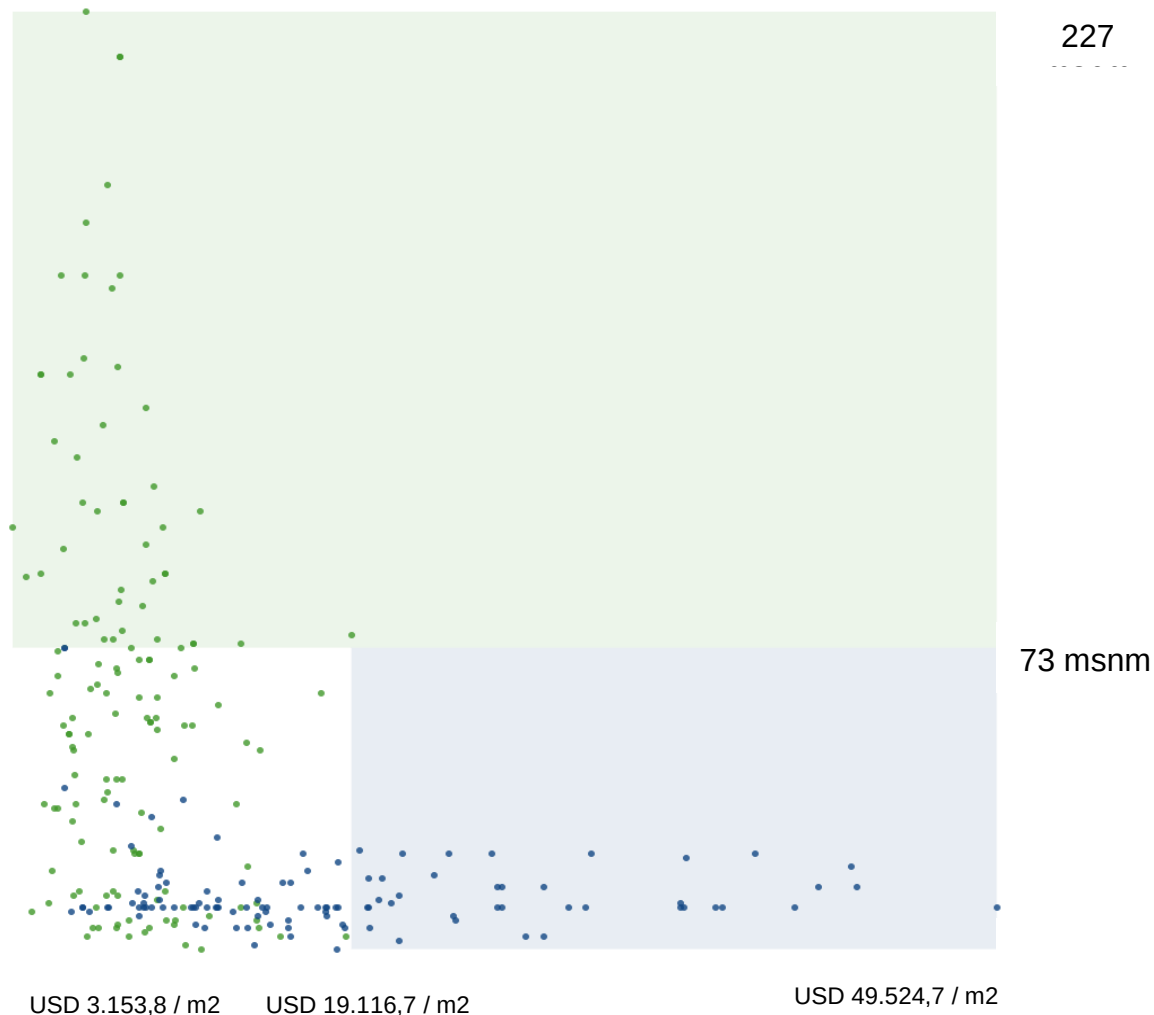
Ilustración 2: Representación de la elevación y el precio por metro cuadrado de las viviendas de Nueva York y San Francisco



Los datos sugieren que entre las viviendas que están a 73 [msnm] o menos de altura y aquellas que cuestan más que USD 19.116,7 por metro cuadrado, estarán en la ciudad de Nueva York.

Ahora bien, es necesario establecer límites para poder visualizar con mayor claridad los datos en el gráfico de dispersión, la identificación de los límites en los datos a través del uso de las matemáticas es la esencia del aprendizaje estadístico. Por supuesto que a su vez se necesitará información adicional para distinguir las viviendas con menores elevaciones y menores precios por metro cuadrado, pero para el caso del establecimiento de límites, se puede observar lo siguiente:

Ilustración 3: Representación de límites en la elevación y el precio por metro cuadrado de las viviendas de Nueva York y San Francisco

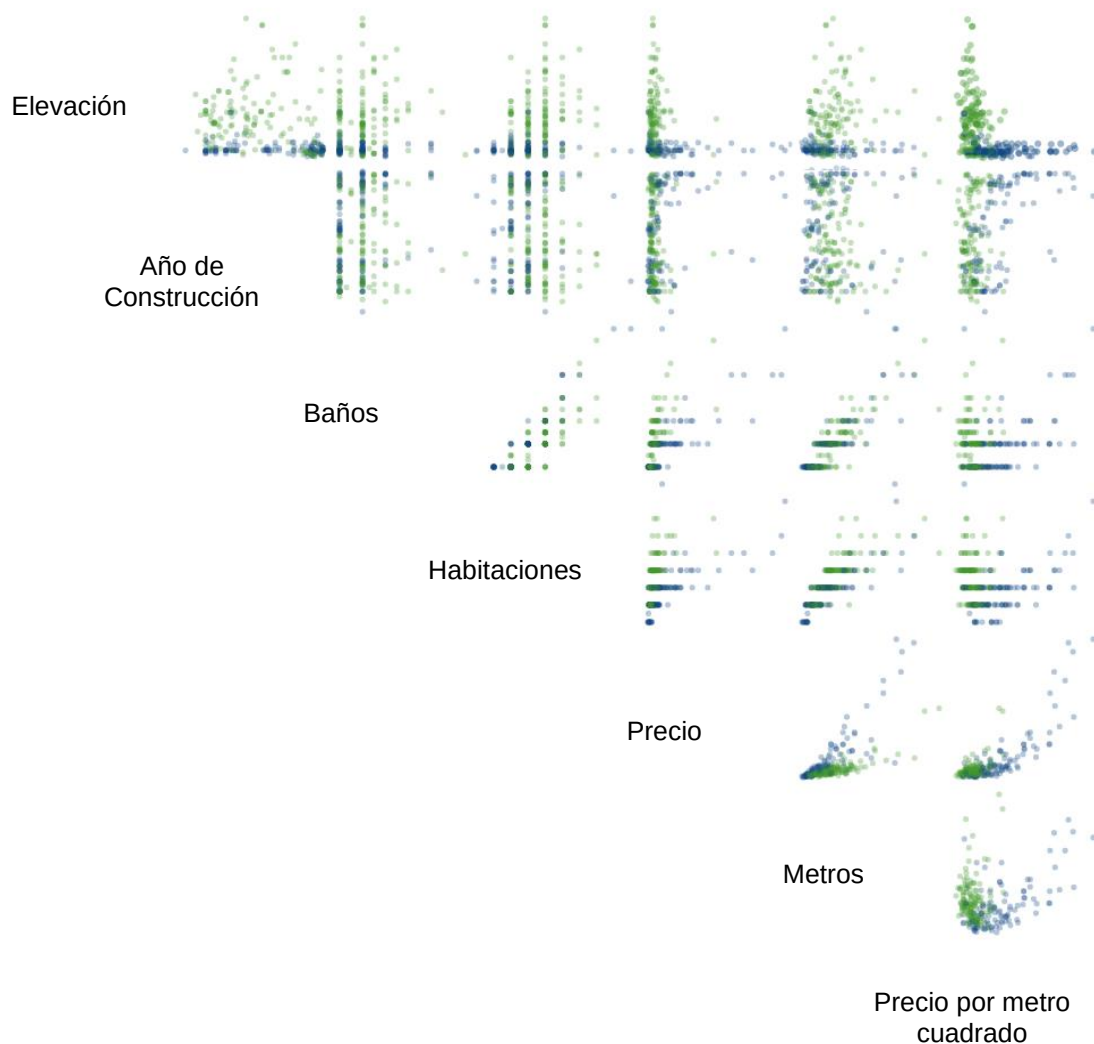


Utilizando las observaciones de elevación mayores a 73 msnm y el precio por metro cuadrado mayor que USD 19.116,7 como los límites de las regiones en el gráfico de dispersión, las viviendas dibujadas en verde y en azul, estarían en San Francisco y Nueva York, respectivamente.

Para crear el modelo del ejemplo citado con el fin de reflejar el comportamiento de los datos y algunos conceptos básicos del machine learning, se utilizó una base de datos que resultó en un modelo de seis dimensiones diferentes. El proceso de crear un modelo también es conocido como “entrenar” un modelo.

Al representar las variables en una matriz de dispersión, es posible identificar las relaciones entre cada par de dimensiones:

Ilustración 4: Ordenamiento en función de distintos atributos de las viviendas de Nueva York y San Francisco

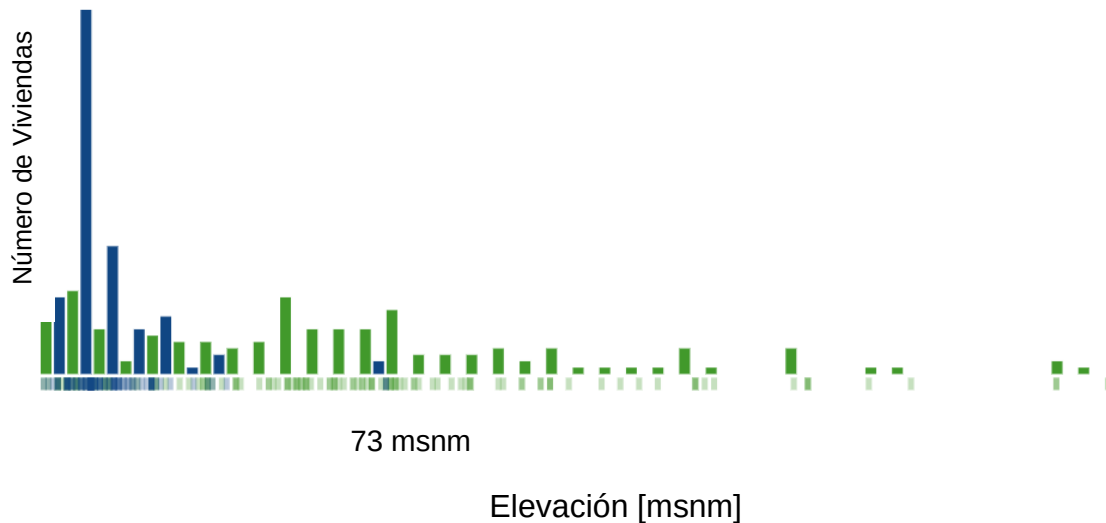


Hay claramente patrones comunes en los datos, pero ya no es tan fácil delinear los límites dado que ya no son tan “obvios”, es en estos casos donde aparece el machine learning, justamente en la búsqueda de patrones en los datos. Los métodos empleados por el machine learning para la identificación de dichos patrones, se basan en el aprendizaje estadístico para identificar los límites.

Un ejemplo de un método de machine learning es un árbol de decisión. Los árboles de decisión observan el comportamiento de una variable a la vez y se constituyen en un método relativamente simple (aunque rudimentario) de machine learning.

Para el mismo conjunto de datos utilizado por Stephanie Yee y Tony Chu, podemos volver a revisar el límite antes propuesto (“todos los departamentos por sobre los 73 msnm pertenecen a San Francisco”), pero utilizando las técnicas del machine learning, para ello, se representarán los datos a través de un histograma:

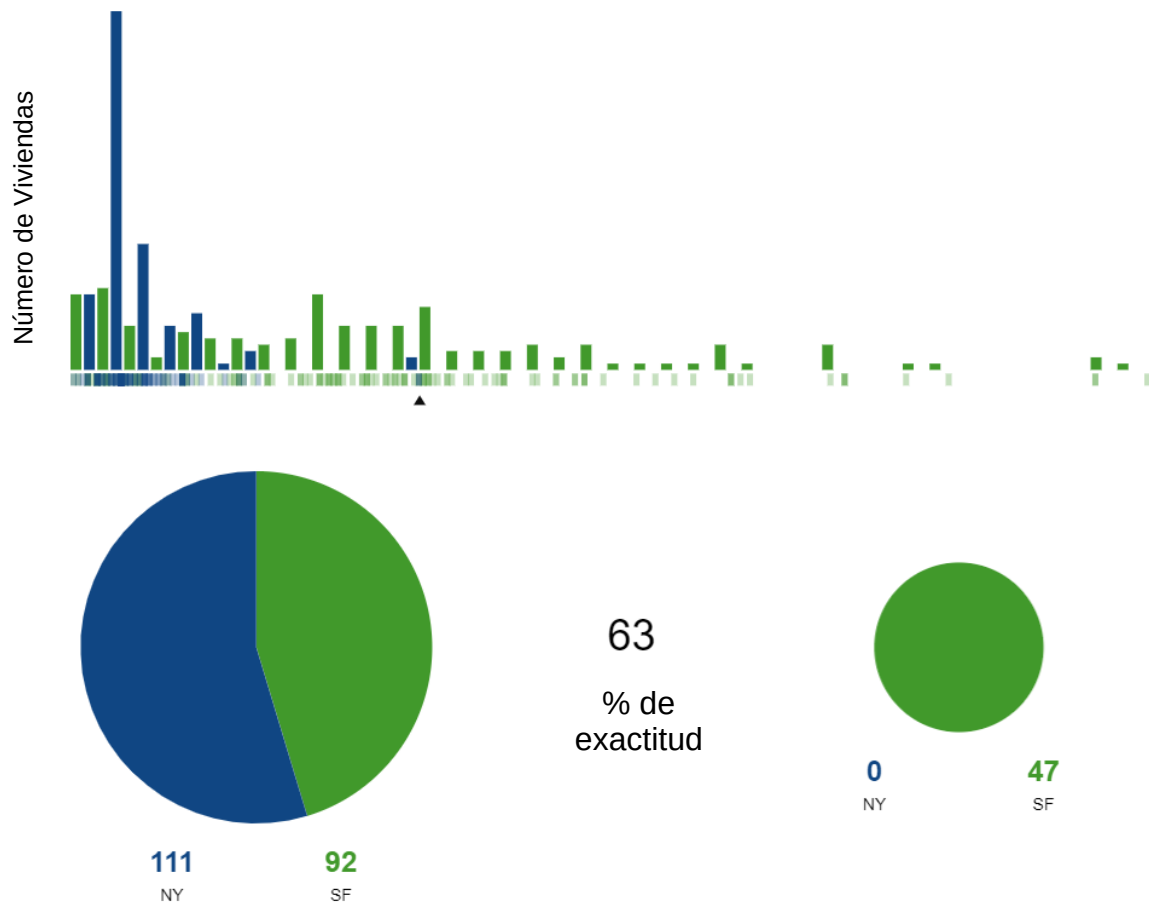
Ilustración 5: Número de viviendas de Nueva York y San Francisco según elevación



A través de esta representación, es posible observar mejor qué tan frecuentemente aparecen las viviendas en cada elevación. Mientras que la vivienda más alta en Nueva York está a 73 msnm, la mayoría de ellas parecen tener elevaciones mucho menores.

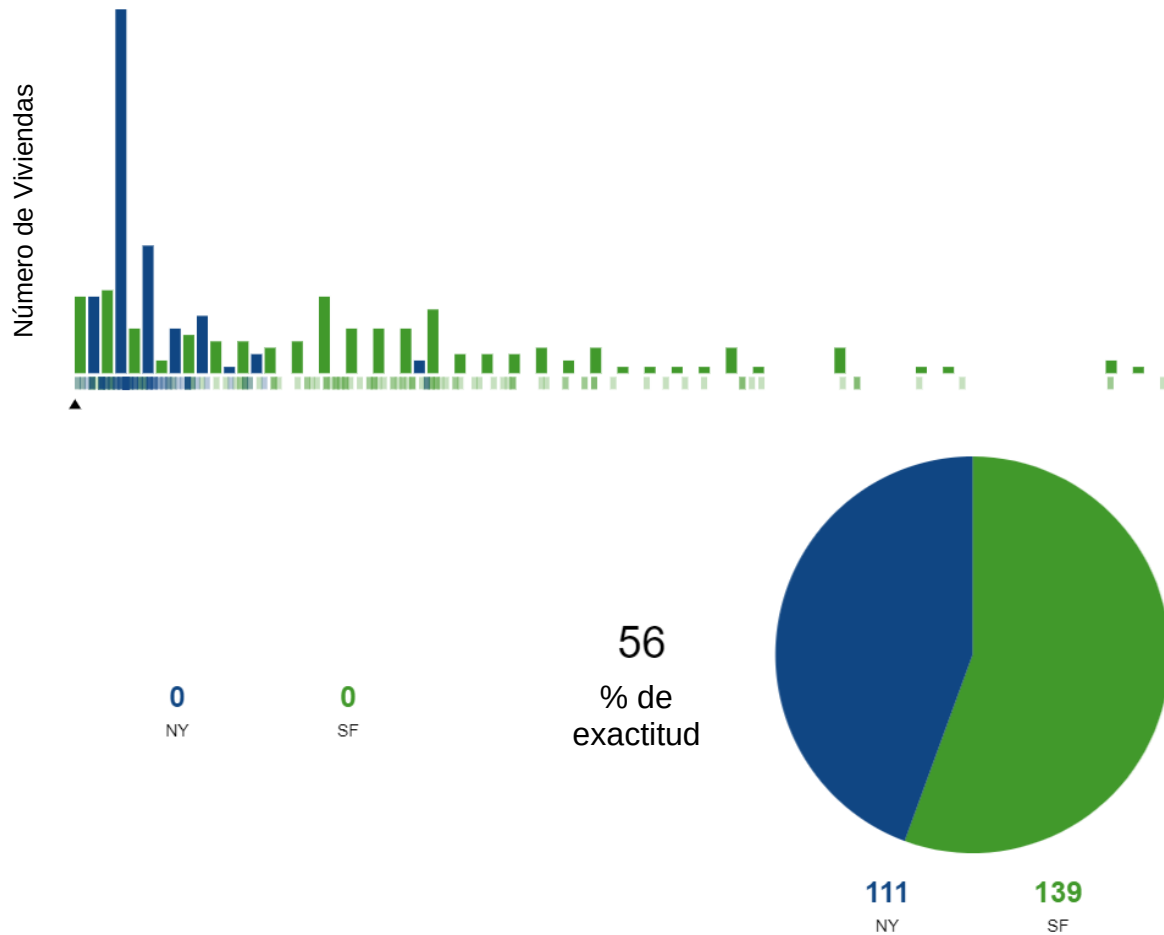
Un árbol de decisión usa secuencias si/entonces (if/then) para definir patrones en los datos. Por ejemplo, si la elevación de una vivienda está por encima de cierto número, entonces la vivienda posiblemente estará ubicada en San Francisco. En machine learning estas declaraciones son denominadas bifurcaciones, y dividen los datos en dos ramas que se basan en algún valor, ese valor es denominado un punto de división. Las viviendas a la izquierda de ese punto son categorizadas de una manera, mientras que las de la derecha se categorizan de una otra manera. Un punto de división es como un límite, pero aplicado a un árbol de decisión:

Ilustración 6: Comparación en la predicción del número de viviendas de Nueva York o San Francisco según altitud



Escoger un punto de división tiene ventajas y desventajas. Nuestra división inicial (~73msnm) clasifica incorrectamente algunas viviendas de San Francisco como si estuviesen en Nueva York. Al mirar esa gran sección verde en el gráfico circular a la izquierda, que corresponde a todas aquellas viviendas de San Francisco que fueron calificadas de manera errónea. Estas se denominan falsos negativos. Sin embargo, un punto de división que captura sólo viviendas de San Francisco va a incluir también muchas viviendas de Nueva York. Estas son denominadas falsos positivos:

Ilustración 7: Comparación en la predicción del número de viviendas de Nueva York o San Francisco según altitud en otro punto del histograma (0 msnm)

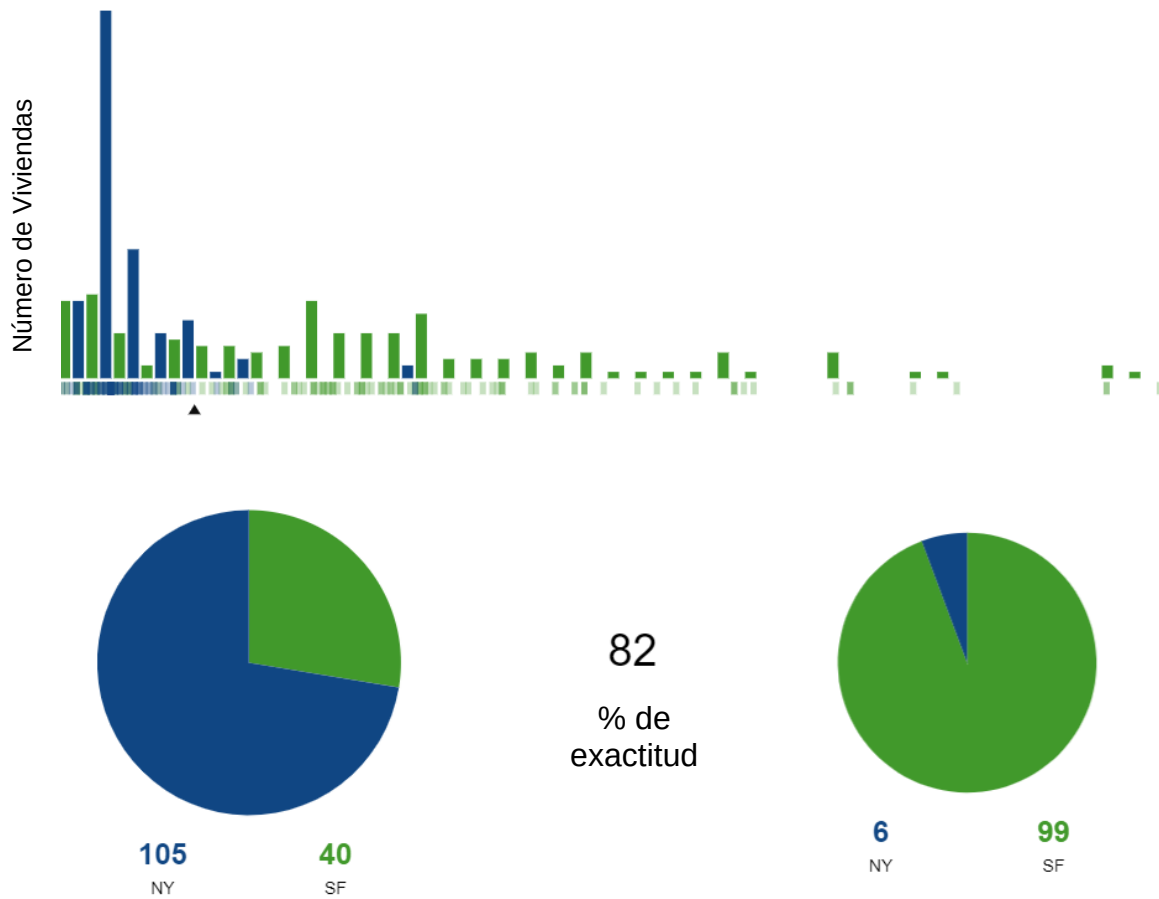


En la mejor división, los resultados de cada rama deberían ser tan homogéneos (o puros) como sea posible. Hay varios métodos matemáticos que puedes emplear para calcular la mejor división como, por ejemplo, el coeficiente de Gini² o la entropía cruzada³. Sin embargo, en el ejemplo de las viviendas, incluso la mejor división de una característica en particular no logra separar las viviendas de San Francisco de las de Nueva York:

² El coeficiente de Gini es una medida de la desigualdad ideada por el estadístico italiano Corrado Gini. Normalmente se utiliza para medir la desigualdad en los ingresos dentro de un país, pero puede utilizarse para medir cualquier forma de distribución desigual.

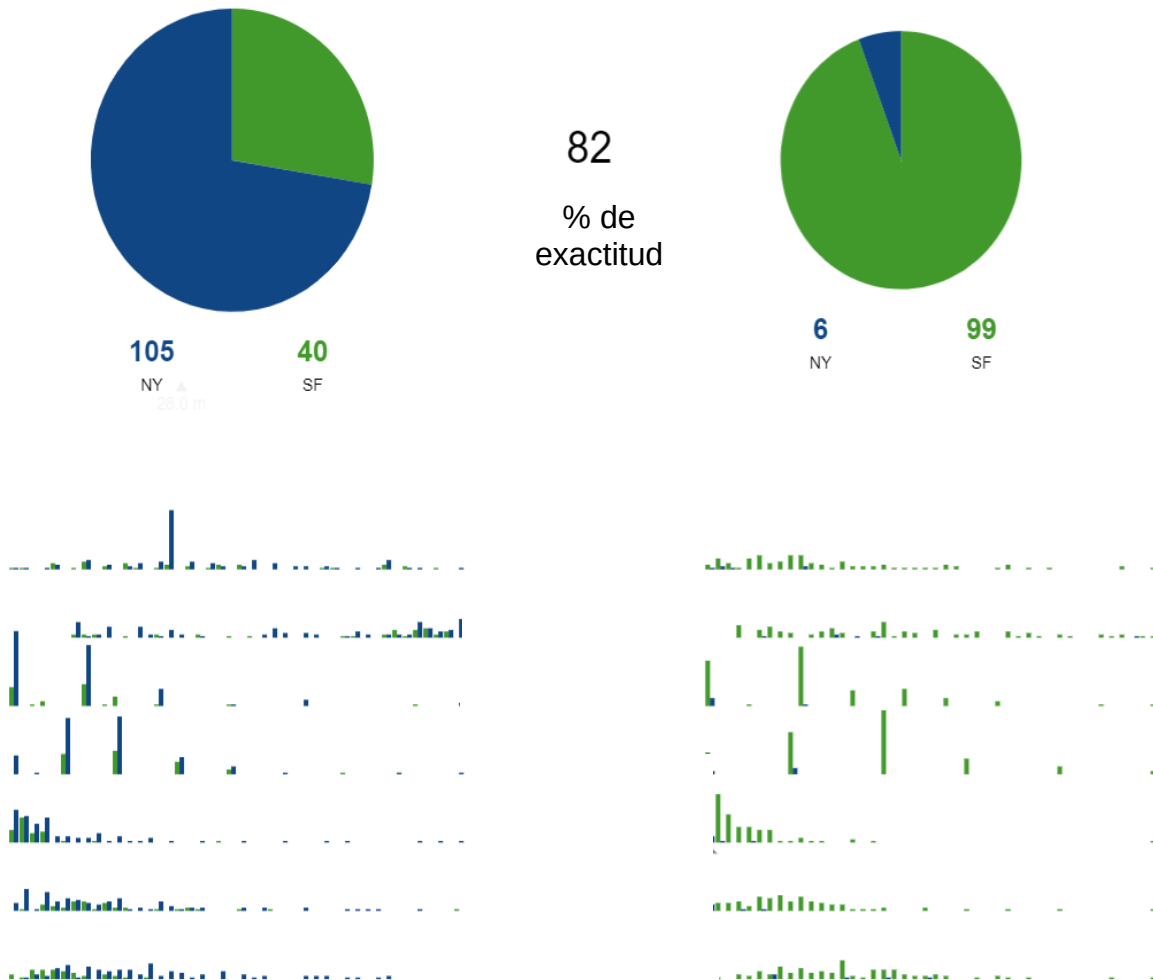
³ La entropía cruzada es una métrica que puede utilizarse para reflejar la precisión de los pronósticos probabilísticos y está estrechamente vinculada con la estimación por máxima verosimilitud.

Ilustración 8: Comparación en la predicción del número de viviendas de Nueva York o San Francisco según altitud utilizando el coeficiente de Gini



Para agregar otro punto de división, el algoritmo repite el proceso mencionado anteriormente en los subconjuntos de datos:

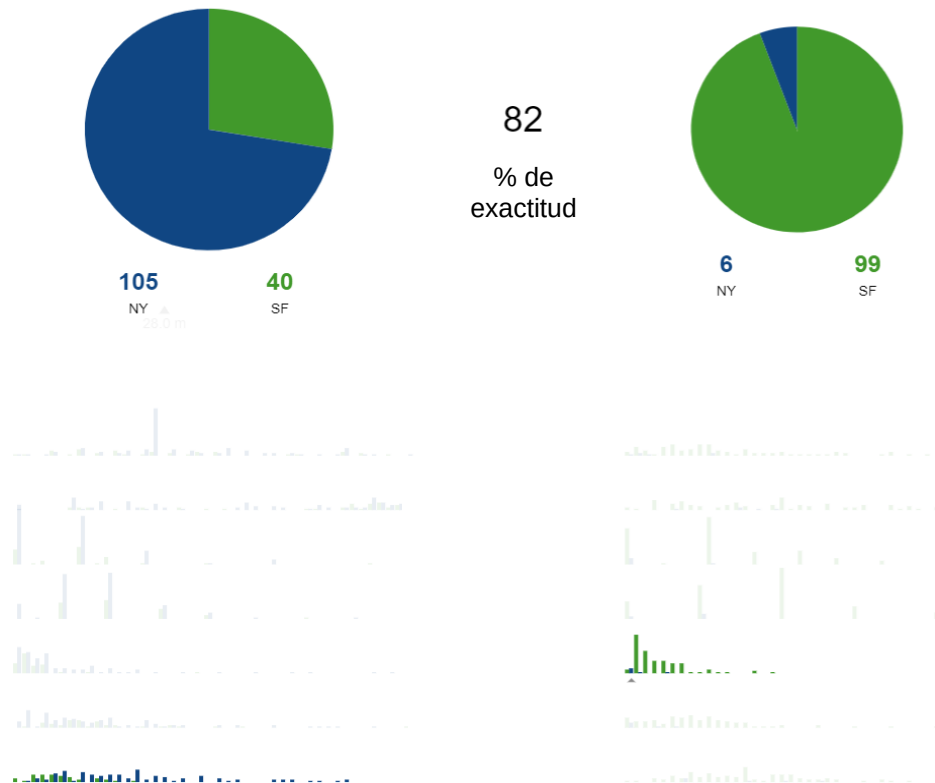
Ilustración 9: Representación gráfica de varios puntos de división para visualizar la distribución de las viviendas de Nueva York y San Francisco



Los histogramas muestran la distribución de cada subconjunto, que se repiten para cada variable. Esta repetición se denomina recursión, y es un concepto que aparece frecuentemente en los modelos de entrenamiento, razón por la cual, los computadores son tan buenos aplicando técnicas de aprendizaje estadístico ya que pueden hacer tareas repetitivas de manera rápida, y sin aburrirse.

La mejor división para el ejemplo de las viviendas variará con base en qué rama del árbol se esté observando. El algoritmo descrito en el ejemplo es codicioso, ya que realiza un enfoque de “arriba a abajo” para dividir los datos. En otras palabras, el algoritmo busca a la variable que haga cada subconjunto lo más homogéneo posible, en ese momento en particular:

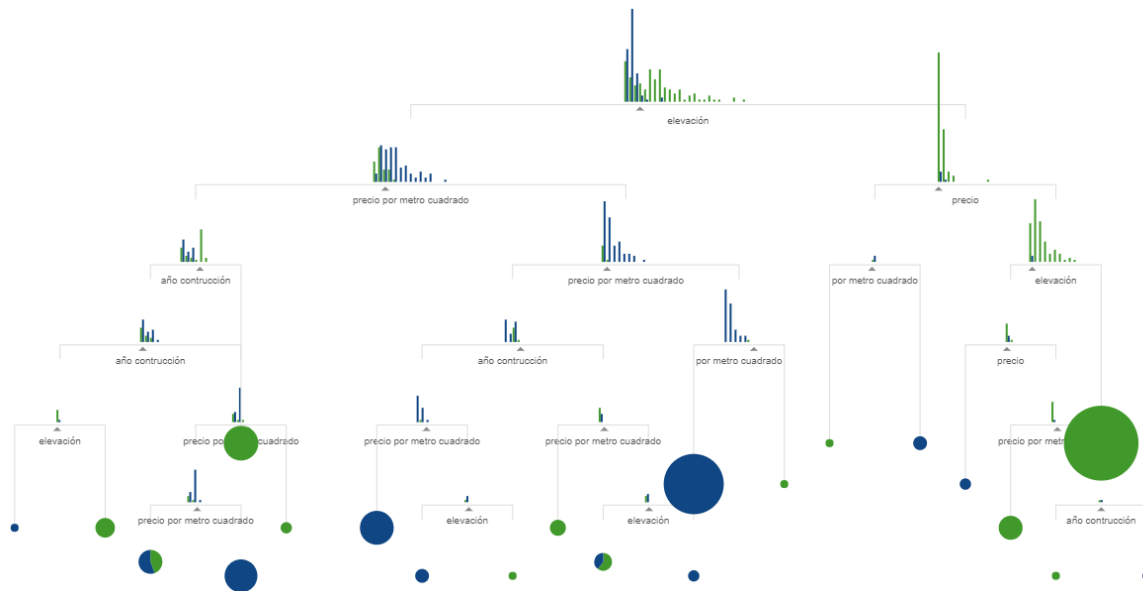
Ilustración 10: Mejor punto de división según el algoritmo en un momento dado de las viviendas de Nueva York y San Francisco



Para este caso, las viviendas de menores elevaciones, el precio por metro cuadrado es USD 1.061 por m², la mejor variable para la próxima declaración si/entonces es “para viviendas de mayor elevación, el precio es de USD 51.4500”.

El crecimiento del árbol de decisión va en función de la cantidad de declaraciones que se le vayan agregando al modelo. Cada nueva bifurcación, agrega nueva información que puede aumentar la precisión de la predicción del árbol. Por ejemplo, la división de una capa más profunda aumenta el porcentaje de exactitud del árbol en un 84%, si agregamos una más, podemos llegar al 96%, nuestro árbol de decisiones comienza a verse de la siguiente forma:

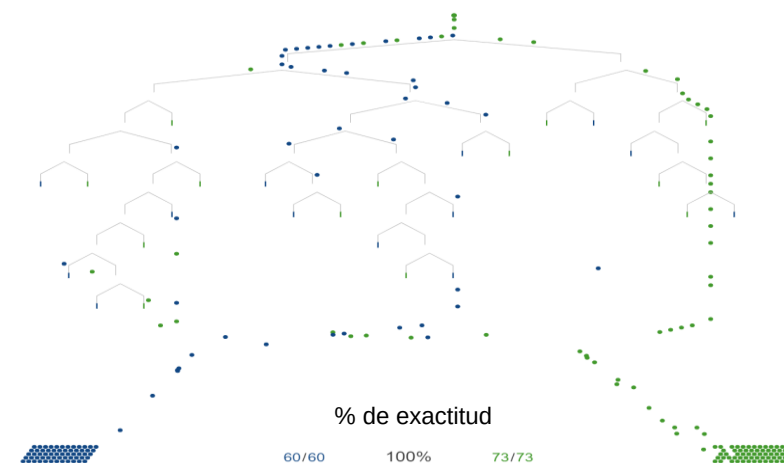
Ilustración 11: Árbol de decisión para clasificar si las viviendas son de Nueva York o San Francisco



Incluso podríamos seguir agregando ramas hasta que las predicciones del árbol sean precisas al 100%, asegurando que al final de cada rama todas las casas están San Francisco, o todas en Nueva York. Estas ramas finales del árbol se denominan nodos hoja, los modelos de árboles de decisión clasificarán las viviendas en cada hoja de acuerdo con qué clase de viviendas están en la mayoría.

El modelo de árbol de decisión, recién entrenado, determina si una vivienda está en San Francisco o en Nueva York, al pasar cada dato a través de las ramas. En la siguiente imagen, es posible observar que los datos que fueron utilizados para entrenar el árbol fluyen a través de él:

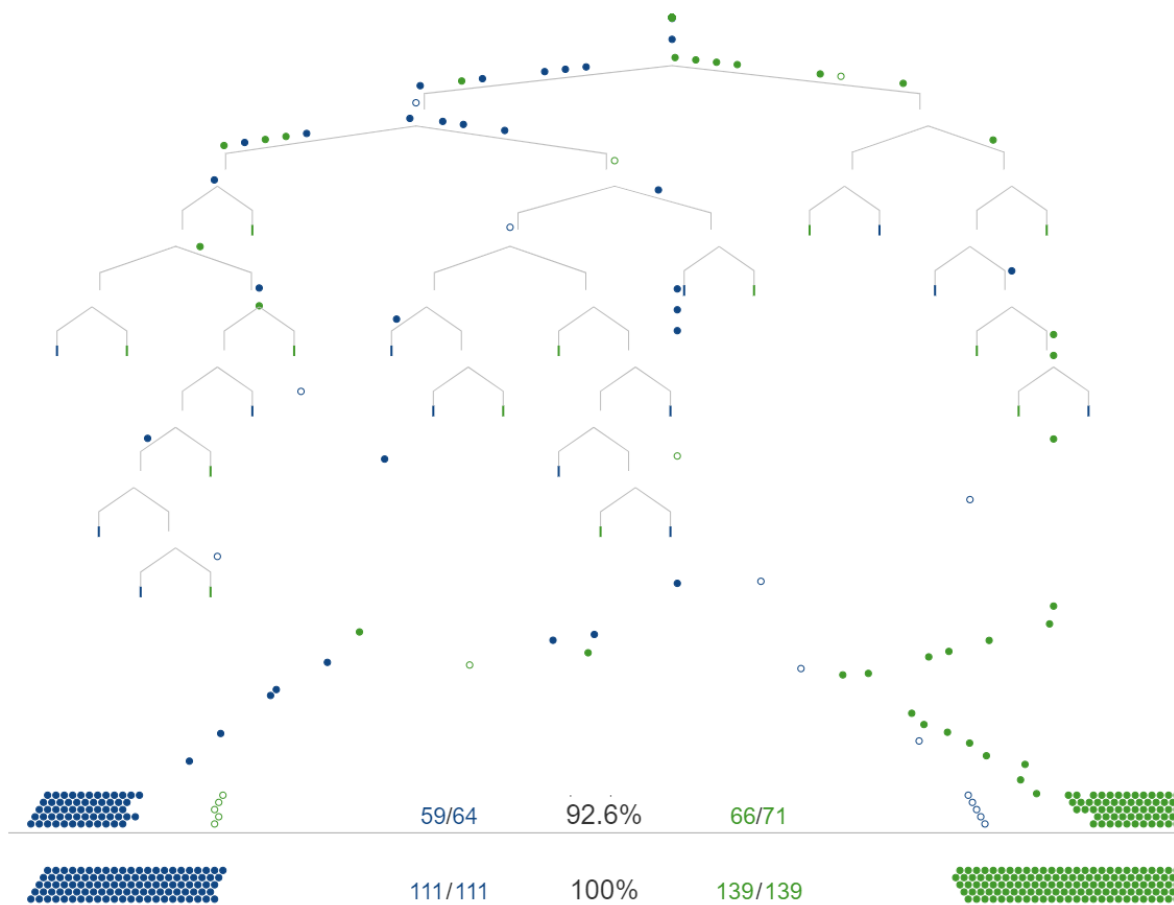
Ilustración 12: Representación de cómo los datos de las viviendas de Nueva York y San Francisco fluyen a través del árbol de decisión



Estos datos se denominan datos de entrenamiento porque fueron usados para entrenar el modelo. Debido a que se hizo crecer el árbol hasta que llegase a una precisión del 100%, este árbol asocia perfectamente cada punto de los datos de entrenamiento con su correspondiente ciudad de manera exacta.

Una vez entrenado el modelo, lo siguiente es verificar el comportamiento del árbol con datos no vistos con anterioridad, estos datos se denominan datos de prueba. Idealmente, el árbol debería comportarse de manera similar tanto en datos conocidos como desconocidos:

Ilustración 13: Representación de cómo los datos de prueba fluyen a través del árbol de decisión en comparación con los datos de entrenamiento para clasificar las viviendas de Nueva York y San Francisco



Los errores que pudiesen estar ocurriendo en el modelo, se producen por sobreajuste. El modelo ha aprendido a tratar a cada detalle en los datos de entrenamiento como importantes, incluso aquellos que resultaron ser irrelevantes,

el sobreajuste⁴ es parte de un concepto fundamental en machine learning la cual corresponde a la compensación entre el sesgo y la varianza.

Como podemos apreciar, los árboles de decisión nos permiten elaborar modelos que, una vez entrenados, nos permiten identificar patrones en los datos a través del uso de declaraciones si/entonces con el objetivo de generar límites y poder hacer predicciones de estos.

A través del uso de árboles de decisión, se desarrollará un modelo con el objetivo de predecir el comportamiento de una población en torno a la demanda que podría llegar a tener el negocio para el cuidado de plantas vivas a fin de tomar la decisión de desarrollar dicho negocio o no.

⁴ Una forma de abordar los errores de sobreajuste es imponer límites sobre cómo crece un árbol cambiando el umbral de tamaño mínimo de nodo. A medida que aumenta el umbral de tamaño mínimo de nodo, hay menos divisiones. Los árboles se vuelven menos tupidos

4. Modelo de Negocio

Para representar el modelo de negocios a través del cual se busca representar la manera en la cual la idea de negocio busca generar valor a los clientes, se utilizará el modelo Canvas de negocios, el que por medio de un diseño modular permite visualizar los elementos necesarios para construir una propuesta de valor. Este modelo contempla 9 módulos para el desarrollo completo de la propuesta:

Ilustración 14: Modelo Canvas



4.1. Segmento de clientes

Los clientes a los cuales está enfocado este negocio tienen características socioculturales pertenecientes a los estratos sociales AB (clase alta), C1a (clase media acomodada) ⁵, tanto hombres como mujeres entre los 30 y 60 años, cuyas características propias de las viviendas que habitan son departamentos,

⁵ Para mayor detalle de los GSE y su caracterización se puede visitar el sitio <https://aimchile.cl/gse-chile/>

townhouses y/o casas, ubicadas en la Región Metropolitana y que no permiten generar el ambiente adecuado para el desarrollo de las plantas de interior o exterior.

Según los datos obtenidos del CENSO 2017, el tamaño de la población objetivo a la cual está dirigido el modelo de negocios es el siguiente:

Ilustración 15: Cantidad de personas por rango etéreo

Rango Etéreo	Cantidad de Personas
20 - 24	1.354.863
25 - 30	1.482.634
30 - 34	1.316.984
35 - 39	1.239.853
40 - 44	1.180.390
45 - 49	1.169.394
50 - 54	1.182.076
55 - 59	1.090.201
> 60	3.045.822

Fuente: Resultados definitivos Censo 2017 - Instituto Nacional de Estadística

El universo de 13.062.217 personas a las cuales está dirigido el modelo de negocio, está definido por las variables antes mencionadas justificadas bajo el sesgo inicial de que las personas que trabajan y están entre los 30 a más de 60 años, son las que pagarían por el servicio. Esto se comprobará a través de la aplicación de una encuesta con el objetivo de aportar los datos que se utilizarán en el árbol de decisión.

4.2. Propuesta de valor

Como propuesta de valor para este negocio, se proponen dos líneas derivadas del concepto de “cuidado de plantas”, estas son:

- Cuidado de plantas: Centrado en clientes cuya necesidad principal es encargar su o sus plantas para el cuidado por un período de tiempo determinado. Pensado principalmente para las ausencias en períodos de vacaciones y/o viajes de negocios de los dueños de las plantas con el fin de que puedan dejarlas en buenas manos y que cuando vuelvan, se les devolverá en igual o mejores condiciones.

- **Mantenimiento integral:** Centrado en clientes cuya necesidad es el mantenimiento de sus plantas de forma mensual o semanal, según corresponda. Este servicio busca la generación de contratos de largo aliento con clientes empresas o personas naturales que busquen tener sus plantas en óptimas condiciones durante todo el año. Enfocado en la mantención in situ de todas las plantas que el cliente estime conveniente.

4.3. Canales

Los canales de venta serán distintos para cada una de las líneas de negocio antes mencionadas. Sin embargo, habrá una página web y perfiles en redes sociales que centralizará las solicitudes de ventas. Los canales de venta serán:

- **Redes sociales:** Canal de ventas principalmente centrado en el servicio de cuidado de plantas debido a la inmediatez y rápido contacto que existe con la necesidad del cuidado de plantas al momento de ausentarse del hogar.
- **Página web:** Principalmente centrado en el servicio de mantenimiento integral de plantas, puesto que esto requiere de una descripción más amplia y detallada de lo que se ofrece, con generación de cotizaciones y planes ad hoc a las necesidades de cada cliente interesado.
- **Presencial:** debido a la existencia de un lugar físico en donde se establecerá el ambiente para el cuidado de las plantas, se generará un puesto de ventas físico para la atención de los clientes que vayan a dejar sus plantas y/o estén interesados en conocer los servicios.

4.4. Relación con el cliente

La relación con el cliente estará centrada en un modelo de atención B2C⁶, debido a que los servicios están diseñados para la atención de clientes que necesiten el contacto con el negocio. Sin embargo, no se descarta la posibilidad de explorar en negocios de relación B2B⁷ a través del diseño de productos como capacitaciones y/o venta de productos, los cuales, para efectos de este estudio, no serán considerados.

⁶De negocio a cliente final o Business to Customer por sus siglas en inglés.

⁷ De negocio a negocio o Business to Business por sus siglas en inglés.

4.5. Flujos de Ingreso

Los flujos de ingreso provendrán principalmente de las ventas que existan de ambos servicios, el monto y volumen de los ingresos será estimado utilizando la encuesta a través de la cual se entregarán los datos para el árbol de decisión con el que se evaluará la factibilidad del presente modelo de negocio. Para efectos del presente modelo Canvas, será expresado simplemente como una declaración.

4.6. Recursos Clave

Como recursos clave del presente modelo de negocio, se establece el siguiente listado:

- Terreno que permita el establecimiento de microclimas controlados con el objetivo de albergar plantas exóticas.
- Automóvil que permita el traslado de plantas de un lugar a otro con el objetivo de facilitar el transporte a los clientes que contraten el servicio.
- Personal calificado para el cuidado de las plantas, esto se transforma en un recurso esencial dado que generará todo el valor que tiene el negocio y permitirá la entrega en mejores condiciones y para la mantención de estas, permitirá una notoria mejora con el servicio. Este personal puede variar desde ingenieros agrónomos o forestales hasta lugareños con experiencia en el cuidado de plantas que provengan de países tropicales.
- Fertilizantes, abonos, tierra y todo lo necesario para la mantención de las plantas. Esto incluye desde los maceteros hasta el invernadero.

4.7. Actividades Clave

Dentro de las actividades clave, podemos destacar las siguientes:

- Marketing digital: dado que este negocio utilizará como canal de venta las redes sociales, es de suma importancia posicionarse en los motores de búsqueda en los temas relacionados al cuidado de plantas en el hogar, conceptos como el de *Plantsitter* deben estar dentro de las palabras claves de las cuales el negocio pueda captar público objetivo y a su vez, lograr una buena segmentación de este.

- Cuidado de plantas: esta actividad resalta dado que acá se encuentra el valor agregado del negocio, es a través de este tipo de actividades que se desarrollará el modelo de negocio y se cultivará la relación con el cliente ya sea a través del cuidado de las plantas por ausencia de sus dueños o mantención de estas. Esta actividad incluye la generación de conocimiento vital para el cuidado específico de las especies tropicales y/o exóticas que existen en la actualidad, por lo que se convierte en un activo de valor para el negocio.
- Logística: Aunque no siempre será necesario, la planificación de las rutas, almacenaje y distribución de todas las plantas que se tenga en custodia o se les haga mantención, se convierte en una actividad de valor dado que genera lazos de confianza con los clientes, sobre todo en modelos de negocios B2C.

4.8. Red de Partners

Como partners clave, se visualizan los siguientes:

- Universidades: Al requerir conocimientos técnicos para el cuidado de las plantas exóticas, se establece como partner clave a las universidades que imparten carreras relacionadas con el área, con el fin de proveer un foco de investigación y generar economías de escala en cuanto a compartir costos y conocimientos.
- Proveedores de artículos de jardín: los proveedores de fertilizantes, abonos, tierras, maceteros y todo lo necesario para el cuidado de una planta se convierte en un tema crítico para el mantenimiento de ese modelo de negocios.

4.9. Estructura de Costos

Los costos del presente modelo de negocio se entregarán en el modelo Canvas de forma referencial ya que no es el foco del estudio. Para estos, se estima como crítico estimar el costo de mantenimiento de una planta al mes (costo marginal) con el objetivo de extrapolar los resultados. Este dato lo podremos obtener de la encuesta que se utilizará para obtener los datos que se utilizarán para el árbol de decisión.

5. Metodología de Investigación

5.1. Diseño de Metodología y tipo de investigación

El presente estudio de prefactibilidad será abordado a través de diferentes metodologías dependiendo del componente del estudio que se esté abordando. Dado que este estudio busca inferir el comportamiento de una población a partir de una muestra, se utilizará un método cuantitativo de carácter exploratorio, los cuales son un tipo de investigación utilizada para estudiar un problema que no está claramente definido, por lo que se lleva a cabo para comprenderlo mejor, pero sin proporcionar resultados concluyentes, lo cual nos proporcionará, inicialmente, la data para el modelo de decisión que se armará más adelante.

Dentro de los métodos de investigación cualitativa exploratoria se encuentran las encuestas, las cuales son catalogadas como investigación primaria. La investigación primaria es la información recopilada directamente del sujeto. Puede ser a través de un grupo de personas o incluso de un individuo. Puede ser llevada a cabo directamente por el investigador o encargarse a un tercero para que la realice en su nombre. La investigación primaria se lleva a cabo específicamente para explorar un determinado problema que requiere un estudio a profundidad.

Tal y como se mencionó anteriormente, para esta etapa del estudio se definió la utilización de una encuesta. Las encuestas se utilizan para recopilar información de un grupo determinado. Es uno de los métodos cuantitativos más importantes. Se pueden utilizar varios tipos de encuestas o sondeos para explorar opiniones, tendencias, etc.

La encuesta desarrollada para este estudio fue generada a través de la utilización de la herramienta Google Form, la cual permite al encuestador el diseño de un formulario en línea, fácil de distribuir, con resultados en línea y exportables en formatos con los cuales se puede trabajar utilizando diversas herramientas.

Luego de extraída la data, se analizará utilizando estadística descriptiva con el fin de abordar los problemas de balanceo de los datos, correlaciones, etc. Para este trabajo, utilizaremos la herramienta Google Colab, también conocido como "Collaboratory", permite programar y ejecutar Python en cualquier navegador. Este lenguaje de programación, entre sus cualidades más destacables, permite de manera simple a través de la instalación de librerías, el trabajo en los ámbitos de ciencia de datos y Machine Learning, siendo este último ámbito en el cual desarrollaremos el modelo predictivo para el presente estudio.

Precisamente en la línea del Machine Learning, utilizaremos una metodología de investigación basada en un modelo de aprendizaje supervisado a través de la generación de un árbol de decisión de clasificación. En estos tipos de algoritmos, existe una variable objetivo-predefinida y se utilizan, principalmente, para resolver problemas de clasificación. Las variables de entrada y salida pueden ser categóricas o continuas con el objetivo de dividir el espacio de predictores (variables independientes) en regiones distintas y no sobrepuestas con el objetivo de dividir la población o muestra en conjuntos homogéneos basados en la variable de entrada más significativa para la mejor división de la población o muestra.

Como se mencionó anteriormente, el tipo de árbol utilizado será de clasificación, en este tipo de algoritmos, la variable dependiente es categórica y el valor del nodo terminal se reduce a la moda de las observaciones del conjunto de entrenamiento que han “caído” en esa región. En términos de las ventajas y desventajas que podemos apreciar en este tipo de algoritmos, como ventajas, señalar que:

- Son modelos fáciles de entender.
- Útiles en la exploración de datos a la hora de identificar la importancia de variables a partir de cientos de variables.
- Se necesita menos limpieza de datos (outliers y valores faltantes no influyen el modelo en un cierto grado).
- El tipo de datos no es una restricción.
- Es un método no paramétrico (no hay suposición acerca del espacio de distribución y la estructura del clasificador).

Como desventajas, es posible apreciar:

- Sobreajuste.
- Pérdida de información al categorizar variables continuas
- Precisión: métodos como SVM y clasificadores tipo ensamblador a menudo tienen tasas de error un 30% más bajas que los modelos de árboles de clasificación y regresión.
- Inestabilidad, esto, dado que un pequeño cambio en los datos puede modificar ampliamente la estructura del árbol por lo que la interpretación podría no ser tan directa como parece.

Sin embargo, a pesar de las desventajas presentadas anteriormente, estas pueden ser abordadas a través de diferentes técnicas para trabajar los datos dentro de los árboles de decisión, así como también, por diferentes etapas del proceso de construcción del modelo, las cuales revisaremos a continuación.

5.1.1. Fuente de recolección de datos

Para la recolección de datos, se utilizó el diseño de un formulario de Google que contemplaba recolección de datos de caracterización sociodemográfica, caracterización de comportamiento referido a cuidados de plantas y comportamiento de pago con respecto a los servicios entregados. Dicho formulario se disponibilizó a través de redes sociales, mediante canales personales y similares.

Para la definición de la muestra, se utilizó el universo de 2.259.792⁸ y teniendo como consideración los siguientes conceptos:

- **Tamaño de la población:** La cantidad total de personas en el grupo que se desea estudiar.
- **Margen de error:** porcentaje que determina en qué medida es posible que los resultados de la encuesta reflejen la opinión de la población general.
- **Nivel de confianza del muestreo:** porcentaje que revela cuánta confianza se puede tener en que la población seleccione una respuesta dentro de un rango determinado.

Para el cálculo de la muestra a encuestar a fin de que sea representativo, se utilizó la siguiente fórmula:

$$\text{Tamaño de la muestra} = \frac{\frac{z^2 \times p(1-p)}{e^2}}{1 + \left(\frac{z^2 \times p(1-p)}{e^2 N} \right)}$$

Ilustración 16: Fórmula para calcular el tamaño de la muestra

Teniendo como referencia un 95% de confianza y un margen de error de 7,5%⁹.

El cuestionario mencionado en Google Drive contiene las siguientes consultas:

- **Pregunta N°1:** “Sexo”, posibles respuestas: Hombre, Mujer, Intersex.
- **Pregunta N°2:** “Género”, posibles respuestas: Femenino, Masculino, Femenino trans, Masculino trans.

⁸ Dado el universo definido en el punto N°4 modelo de negocios del presente documento.

⁹ La justificación de dichos valores está fundamentada en los resultados de captura de la herramienta de recolección de datos, el cual fue de un total de 201 casos permitiendo obtener dichos intervalos de confianza y margen de error.

- Pregunta N°3: “¿qué edad tienes?”, posibles respuestas: Entre 20 y 24; entre 25 y 29; entre 30 y 34; entre 35 y 39; entre 40 y 44; entre 45 y 49; entre 50 y 54; entre 55 y 60; más de 60.
- Pregunta N°4: “Estado Civil”, posibles respuestas: Soltera/o; En pareja; Unión Civil; Casada/o; Separada/o; Viuda/o.
- Pregunta N°5: “Tipo de Vivienda”, posibles respuestas: Casa; Departamento; Town House.
- Pregunta N°6: “Tienes trabajo”, posibles respuestas: Sí; No.
- Pregunta N°7: “Si tu respuesta anterior fue 'Sí', ¿Tu trabajo es dependiente o independiente?, si tu respuesta anterior fue 'No', favor no contestar esta pregunta.”, Posibles respuestas: Dependiente; Independiente.

Esto corresponde a la primera sección del cuestionario, la cual se centra en caracterizar sociodemográficamente a las personas encuestadas. A continuación, comienza la sección de “gustos por las plantas”.

- Pregunta N°8: “¿Te gustan las plantas?”, posibles respuestas: Sí; No; Imparcial (no me gustan y tampoco me molestan).
- Pregunta N°9: “¿Prefieres Plantas de Interior o Exterior?, posibles respuestas: Interior; Exterior; Ambas; Ninguna.
- Pregunta N°10: “¿Conoces de plantas?, Posibles respuestas: Sí; No.
- Pregunta N°11: “¿Cuántas plantas tienes en tu hogar aproximadamente?”, posible respuesta: entrada libre de texto numérico.
- Pregunta N°12: “¿Tienes plantas "caras" en tu hogar? (Caras desde su propio punto de vista)”, posibles respuestas: Sí; No.

Esta sección del formulario se generó con el objetivo de recolectar información relativa a los gustos por las plantas de los encuestados.

- Pregunta N°13: “¿Pagarías por un servicio de Cuidado de Plantas mientras estás de vacaciones? (Servicio que incluye almacenamiento de la planta en un lugar adecuado por el periodo de sus vacaciones, riego, fertilizantes, limpieza, control de plagas y reparaciones en caso de requerirlas)”, posibles respuestas: Sí (en este caso se habilitaba la pregunta N°14); No (en este caso se habilitaba la pregunta N°15 sin pasar por la pregunta N°14).
- Pregunta N°14: “¿Cuánto pagarías (por día) por el servicio de Cuidado de Plantas mientras estás de vacaciones? (Pensar en lo que se pagaría por planta, es decir, costo unitario para el servicio antes descrito. (valor por planta por día))”, posibles respuestas: entrada libre de texto numérico.

- Pregunta N°15: “¿Pagarías por un servicio de mantenimiento de las plantas de tu hogar? (Visitas periódicas en función de sus necesidades en las cuales se haría limpieza, cambio de tierra, mantención de las hojas, fertilizantes, control de plagas y entrega de recomendaciones)”, posibles respuestas: Sí (en este caso se habilitaba la pregunta N°16); No (en este caso se envía el formulario sin pasar por la pregunta N°16 ni N°17).
- Pregunta N°16: “¿Cuánto pagarías por un servicio de mantenimiento de las plantas de tu hogar? (Pensar en lo que se pagaría de forma mensual por el servicio considerando todas las plantas del hogar (jardín incluido en caso de que aplique))”, posibles respuestas: entrada libre de texto numérico.
- Pregunta N°17: “¿Te gustaría que el servicio de mantenimiento fuera en tu hogar o que fueran a retirar tus plantas y luego te las devuelvan?”, posibles respuestas: Mantenimiento en el hogar; Mantenimiento con retiro y devolución de plantas.

Como se mencionó anteriormente, el formulario de recolección de datos fue contestado por 201 personas entregando los siguientes resultados preliminares:

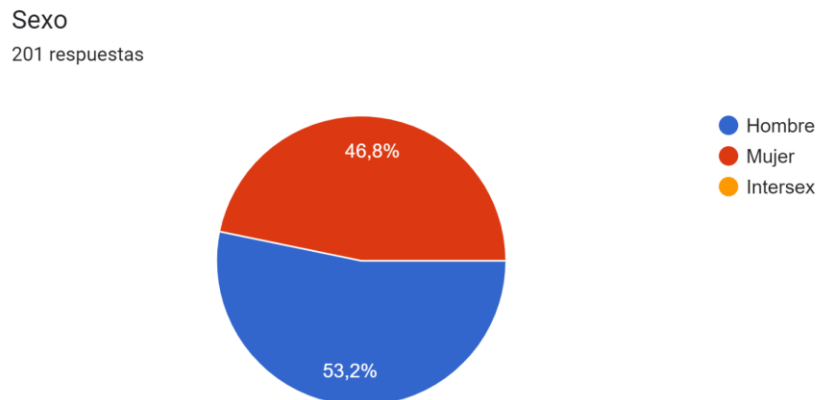


Ilustración 17: Resultados encuesta pregunta Sexo

La muestra fue bastante equitativa en términos del sexo de las personas encuestadas.

¿Qué edad Tienes?

201 respuestas

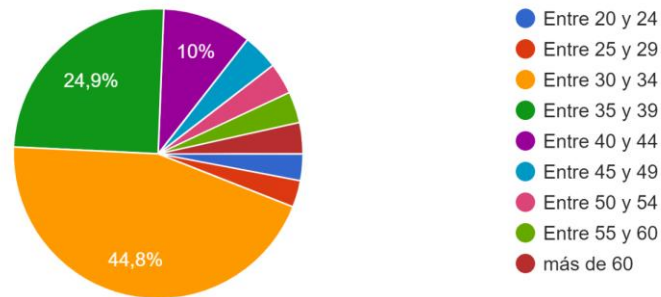


Ilustración 18: Resultados encuesta pregunta Edad

La edad de los participantes del estudio presenta un sesgo para este, debido a que la mayor cantidad de participantes (90) se encuentra en el rango etáreo de entre 30 y 34 años.

Estado Civil

201 respuestas

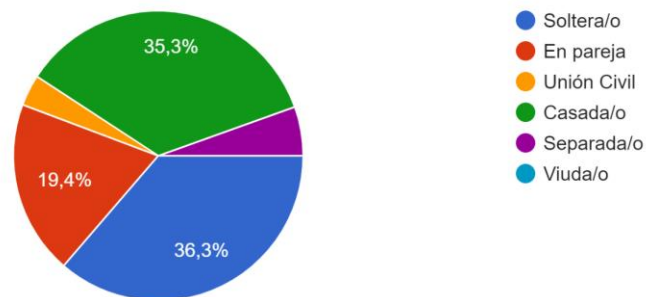


Ilustración 19: Resultados encuesta pregunta Estado Civil

Respecto del estado civil de los participantes, la muestra presenta un comportamiento bastante equitativo respecto de 3 grandes grupos: Solteros/as, Casados/as y En pareja.

En relación con el tipo de vivienda de las personas encuestadas, encontramos 2 grandes grupos que representan el 99% de los encuestados/as:

Tipo de vivienda
201 respuestas

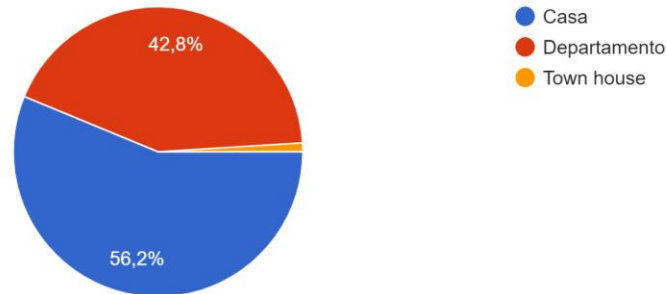


Ilustración 20: Resultados encuesta pregunta Tipo de Vivienda

Siendo estos los y las personas que viven en casas y departamentos, los que, por cierto, se distribuyen de una forma bastante equitativa.

Otro sesgo que se puede producir en el estudio es referente a la condición laboral de las personas encuestadas:

¿Tienes Trabajo?
201 respuestas

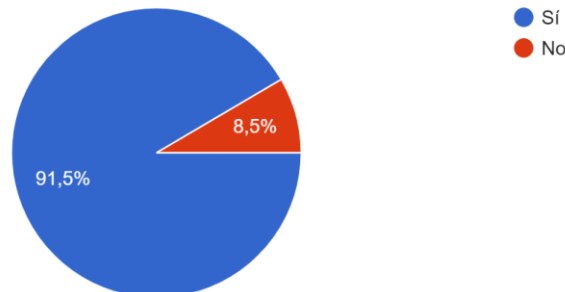


Ilustración 21: Resultados encuesta pregunta empleabilidad

En este aspecto, claramente se observa una tendencia a personas que tienen trabajo y que, además, son trabajadores dependientes como se muestra en la siguiente gráfica:

Si tu respuesta anterior fue 'Sí', ¿Tu trabajo es dependiente o independiente?, si tu respuesta anterior fue 'No', favor no contestar esta pregunta.

183 respuestas

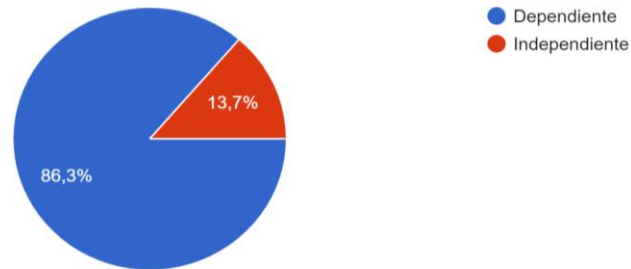


Ilustración 22: Resultados encuesta pregunta Tipo de Trabajo

En torno a las características de los gustos de las personas encuestadas, podemos observar que también existe una clara tendencia hacia la preferencia por las plantas:

¿Te gustan las plantas?

201 respuestas

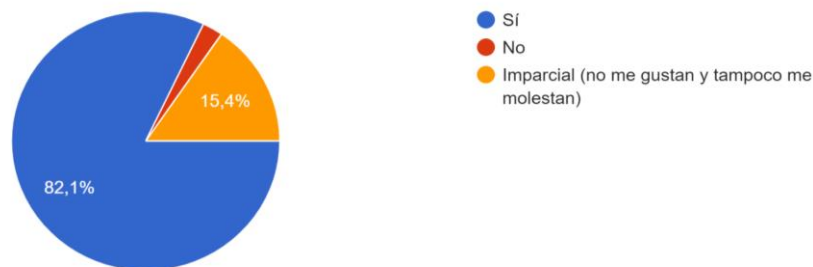


Ilustración 23: Resultados encuesta pregunta gusto por las plantas

A su vez, se aprecia una indiferencia en el tipo de plantas:

¿Prefieres Plantas de Interior o Exterior?

201 respuestas

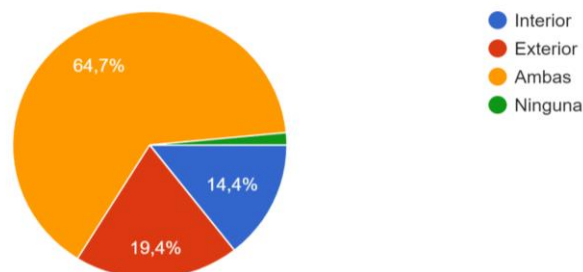


Ilustración 24: Resultados encuesta pregunta plantas de interior o exterior

No obstante, lo anterior, más de un 30% de la muestra evidencia preferencias claras por plantas de interior o exterior.

En cuanto al conocimiento de plantas, las personas encuestadas presentan una leve tendencia a que sí conocen de plantas, pero la distribución en torno a esta respuesta es bastante equitativa como se puede apreciar en la gráfica siguiente:

¿Conoces de plantas?
201 respuestas

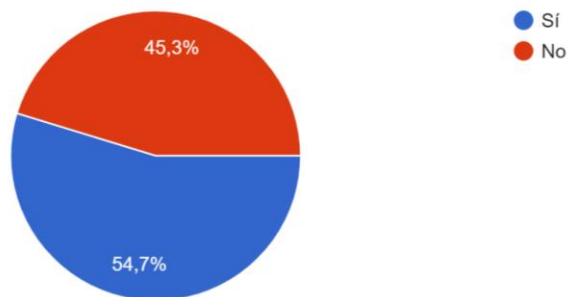


Ilustración 25: Resultados encuesta pregunta conocimiento sobre plantas

Respecto de cuantas plantas por hogar tienen las personas, existe un promedio de 119 plantas por hogar, esto considerando que hubo gente que dijo tener más de diez mil plantas:

¿Cuántas plantas tienes en tu hogar aproximadamente?
201 respuestas

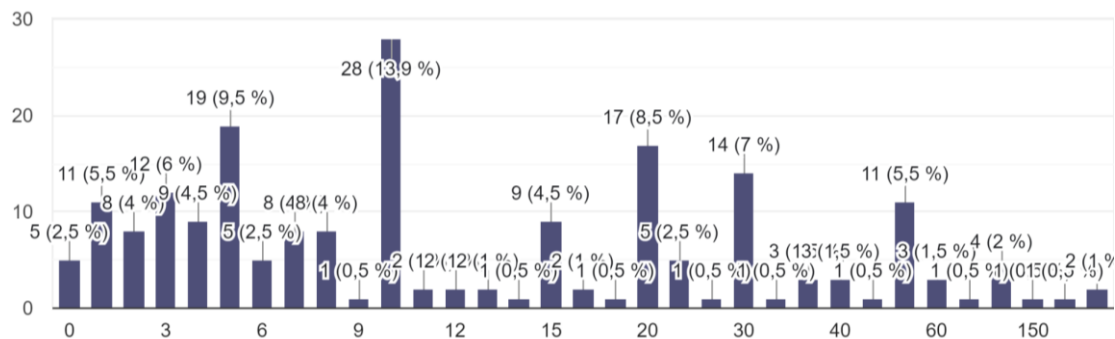


Ilustración 26: Resultados encuesta pregunta cantidad de plantas en el hogar

5.1.2. Construcción del árbol de decisión

Las etapas para la construcción del árbol de decisión utilizando el lenguaje de programación Python en la plataforma Google Colab, serán las siguientes:

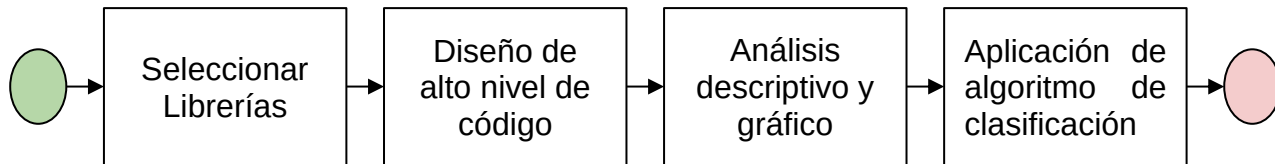


Ilustración 27: Etapas para la construcción del árbol de decisión

5.1.2.1. Seleccíonar Librerías

Una librería Python, corresponde al conjunto de funcionalidades que permiten al usuario llevar a cabo nuevas tareas que antes no se podían realizar. Es decir, las librerías de Python responden al conjunto de implementaciones que permiten codificar este lenguaje, con el objeto de crear una interfaz independiente. Cada una de las librerías de Python disponen de diferentes módulos, con funciones específicas y que también varían en función del sistema operativo con el que se trabaje.

Existe una gran cantidad de librerías disponibles, dentro de las cuales, seleccionamos las que nos permitirán trabajar con foco en algoritmos de machine learning. estas son:

- Librería Pandas: Librería de Python para el manejo y operaciones con data sets (conjuntos de datos), nos ofrece estructuras potentes y flexibles que nos facilitan la manipulación y el tratamiento de datos, muy usada para la ciencia de datos y machine learning.
- Librería PyDrive: Dado que la Google Colab trabaja en la nube, se instaló esta librería la cual nos permite tener acceso a Google Drive, lugar en el cual dejamos todos los archivos de datos.
- Librería PyOD: Es la biblioteca de Python más completa y escalable para detectar objetos atípicos en datos multivariados. Este campo se conoce comúnmente como Detección de valores atípicos o Detección de anomalías.
- Librería Numpy: es una librería especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos. Incorpora

una nueva clase de objetos llamados *arrays*¹⁰ que permite representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación. La ventaja de Numpy frente a las listas predefinidas en Python es que el procesamiento de los arrays se realiza mucho más rápido (hasta 50 veces más) que las listas, lo cual la hace ideal para el procesamiento de vectores y matrices de grandes dimensiones.

- **Librería Matplotlib:** Es una librería especializada en la creación de gráficos en dos dimensiones. Permite crear y personalizar los tipos de gráficos más comunes, entre ellos:
 - Diagramas de barras
 - Histograma
 - Diagramas de sectores
 - Diagramas de caja y bigotes
 - Diagramas de violín
 - Diagramas de dispersión o puntos
 - Diagramas de líneas
 - Diagramas de áreas
 - Diagramas de contorno
 - Mapas de color
 - y combinaciones de todos ellos.
- **Librería subprocess:** La librería “subproceso” permite lanzar nuevos procesos, conectarse a sus pipes de entrada/salida/error y obtener sus códigos de resultado. Este módulo está destinado a reemplazar múltiples módulos y funciones previamente existentes. Para el caso del presente estudio, se utilizará para manejar procesos en segundo plano.
- **Librería Seaborn:** La librería Seaborn proporciona varias funciones para personalizar los gráficos. Para el caso del presente estudio, se utilizará para graficar la estadística descriptiva.
- **Librería Imbalanced-learn:** Es una librería licenciada por el MIT que proporciona diferentes técnicas para el balanceo de datos en muestras que presentan desequilibrio entre sus variables.
- **Librería IPython:** Esta librería permite que Python pueda convivir con múltiples lenguajes de programación. Dado que python es de código abierto, es posible encontrar algoritmos cuyos códigos estén en JSON, HTML y otros. A su vez, permite incorporarlos en el código.

¹⁰ Los arrays o arreglos se utilizan para almacenar múltiples valores en una sola variable.

- Librería Graphviz: Esta librería nos permite graficar códigos como los del árbol de decisión ya que fue diseñada para expresar de forma gráfica flujos estructurados como redes entre otras cosas.
- Librería Scikit Learn: Librería open source que posee herramientas para análisis predictivo de datos. Esta librería es la que cuenta con las clases y funciones asociadas al diseño de árboles de decisión.

5.1.2.2. Diseño del Código de alto Nivel

Las librerías antes mencionadas, se cargan en la herramienta Google Colab mediante el código del Anexo N°1: instalación de librerías, del presente documento. En el bloque de código antes mencionado, se importan todas las bibliotecas y paquetes necesarios para realizar el análisis estadístico y el modelado de un árbol de decisión en Python.

Las primeras líneas de código, importan todas las librerías al notebook de Google Colab para poder utilizarlas, luego la línea de código `%matplotlib inline` es una función mágica en Jupyter Notebook que permite que las visualizaciones y gráficos se muestren directamente en el notebook de Google Colab. La segunda línea importa la biblioteca Seaborn, que se basa en Matplotlib y se utiliza para crear visualizaciones de datos más complejas y atractivas.

Las siguientes líneas de código importan varios paquetes y módulos de Python necesarios para la autenticación de Google Drive, la creación de un modelo de árbol de decisión, la manipulación y el análisis de datos, la visualización de gráficos, la evaluación de modelos y la gestión de imágenes.

En resumen, este bloque de código es necesario para importar todas las bibliotecas y paquetes necesarios para realizar el análisis estadístico y el modelado de un árbol de decisión en Python.

Luego comenzamos con la definición del código necesario para generar los análisis estadísticos provenientes de la muestra utilizando el código del Anexo N°2: Configuración para análisis gráfico, del presente documento. Este bloque de código es utilizado para configurar el ambiente en el que se ejecutará el análisis estadístico de la muestra, incluyendo la configuración de matplotlib para la visualización de gráficos, la autenticación con Google Drive y la creación de una instancia de la clase Google Drive para interactuar con los archivos almacenados en Google Drive.

Para comenzar, se utiliza la "magic function" de Jupyter Notebook `%matplotlib inline` para permitir que los gráficos y visualizaciones se muestren directamente en el notebook de Google Colab, en lugar de abrir una ventana emergente para la visualización.

A continuación, se establece el tamaño predeterminado de las figuras que se crearán usando matplotlib. En este caso, se ha establecido el tamaño en 16 pulgadas de ancho por 9 pulgadas de alto, utilizando la línea de código `plt.rcParams['figure.figsize'] = (16, 9)`.

Después de establecer la configuración de matplotlib, se configura el estilo de los gráficos generados por la biblioteca. En este caso, se utiliza el estilo 'ggplot', que emula el estilo de las gráficas en el paquete R ggplot, con la línea de código `plt.style.use('ggplot')`.

Para acceder a los archivos de Google Drive (el CSV de las respuestas de la encuesta), es necesario autenticarse utilizando credenciales de Google ya que Google Colab opera utilizando Google drive de forma nativa. Para hacerlo, se llama a la función `auth.authenticate_user()`, que solicita al usuario que proporcione sus credenciales de Google a través de una ventana emergente. Luego se crea una instancia de la clase `GoogleAuth`, que se utiliza para autenticarse con Google y obtener acceso a los servicios de Google Drive.

Una vez que se ha creado la instancia de la clase `GoogleAuth`, se establecen las credenciales de la aplicación actual utilizando la línea de código `gauth.credentials = GoogleCredentials.get_application_default()`. Finalmente, se crea una instancia de la clase `Google Drive`, que se utiliza para interactuar con los archivos almacenados en Google Drive a través de la API de Google Drive.

En resumen, este bloque de código se utiliza para configurar el ambiente de trabajo para el análisis estadístico de una muestra, estableciendo las opciones de visualización de matplotlib y autenticándose con Google Drive para acceder a los archivos almacenados en la nube.

Una vez autorizado el Notebook de colab para operar con los datos de la encuesta, se comienza con la clasificación de las preguntas con el objetivo de generar las ramificaciones del árbol de decisión a través del siguiente código del Anexo N°3: Parametrización de respuestas, del presente documento. Este código lee un archivo CSV y realiza algunas transformaciones de datos en él. El objetivo es codificar las respuestas de la encuesta de manera que se puedan analizar posteriormente.

Primero, el código descarga el archivo CSV desde un enlace en Google Drive y lo almacena en una variable llamada `df` como un marco de datos (dataframe) de Pandas. A continuación, el código elimina la columna llamada "Marca temporal" del marco de datos y reemplaza cualquier valor "#N/A" (no disponible) por 0.

Luego, el código convierte algunas variables categóricas en variables numéricas. Por ejemplo, para la variable "Sexo", el programa asigna el valor 1 a "Mujer" y el valor 2 a "Hombre", lo mismo se hace para otras variables. Por último, el código

muestra una matriz de correlación y estadísticas descriptivas del marco de datos. A continuación, se presentan tres matrices de correlación:

Tabla 3: Análisis estadístico pregunta de cantidad de plantas en el hogar

Matriz de Correlación	¿Cuántas plantas tienes en tu hogar aproximadamente?
¿Cuántas plantas tienes en tu hogar aproximadamente?	1,000000
¿Cuánto pagarías (por día) por el servicio de Cuidado de Plantas mientras estás de vacaciones?	-0,003455
¿Cuánto pagarías por un servicio de mantenimiento de las plantas de tu hogar?	0,065249

Tabla 4: Análisis estadístico pregunta por el pago diario de un servicio de cuidado de plantas

Matriz de Correlación	¿Cuánto pagarías (por día) por el servicio de Cuidado de Plantas mientras estás de vacaciones?
¿Cuántas plantas tienes en tu hogar aproximadamente?	-0,03455
¿Cuánto pagarías (por día) por el servicio de Cuidado de Plantas mientras estás de vacaciones?	1,000000
¿Cuánto pagarías por un servicio de mantenimiento de las plantas de tu hogar?	0,206273

Tabla 5: Análisis estadístico pregunta por el pago diario de un servicio de mantención de plantas

Matriz de Correlación	¿Cuánto pagarías por un servicio de mantenimiento de las plantas de tu hogar?
¿Cuántas plantas tienes en tu hogar aproximadamente?	0,065249
¿Cuánto pagarías (por día) por el servicio de Cuidado de Plantas mientras estás de vacaciones?	0,206273
¿Cuánto pagarías por un servicio de mantenimiento de las plantas de tu hogar?	1,000000

Dado que, en la matriz de correlación los valores en la diagonal principal (1.0) indican la correlación perfecta de cada variable consigo misma, los valores fuera de la diagonal principal indican la correlación entre las variables. Los valores positivos indican una correlación positiva, lo que significa que a medida que una variable aumenta, la otra variable también tiende a aumentar. Los valores negativos indican una correlación negativa, lo que significa que a medida que una variable aumenta,

la otra variable tiende a disminuir. Los valores más cercanos a cero indican una correlación débil, mientras que los valores más cercanos a 1.0 (o -1.0) indican una correlación fuerte.

Las matrices de correlación muestran que no hay una correlación significativa entre el número de plantas en el hogar y la disposición a pagar por un servicio de cuidado o mantenimiento de plantas. Sin embargo, hay una correlación positiva moderada entre la disposición a pagar por el servicio de cuidado y el servicio de mantenimiento de plantas.

Los descriptivos iniciales muestran que la mayoría de los participantes tienen menos de 100 plantas en su hogar y que la disposición a pagar por un servicio de cuidado y mantenimiento de plantas varía ampliamente, con una gran proporción de participantes que no estarían dispuestos a pagar nada.

Al analizar la estadística descriptiva en relación con las 3 preguntas fundamentales, podemos notar lo siguiente:

Tabla 6: Estadística descriptiva de cantidad de plantas en el hogar

Estadígrafo	¿Cuántas plantas tienes en tu hogar aproximadamente?
Cantidad (N)	201
Media	20,0348
Desviación Estándar	32,4346
Valor mínimo	0
25%	5
50%	10
75%	25
Valor Máximo	347

Muestra que la media de plantas en el hogar es de alrededor de 20, con un rango amplio, desde 0 hasta 347. La desviación estándar de 32,4346 indica una gran variabilidad en el número de plantas en los hogares de los encuestados.

Tabla 7: Estadística descriptiva de precio del servicio de cuidados por día

Estadígrafo	¿Cuánto pagarías (por día) por el servicio de Cuidado de Plantas mientras estás de vacaciones?
Cantidad (N)	201
Media	\$1.494
Desviación Estándar	\$4.695
Valor mínimo	\$0
25%	\$0
50%	\$0
75%	\$500
Valor Máximo	\$38.000

La media de la cantidad que las personas están dispuestas a pagar es de 1.494 pesos por día, con una desviación estándar de \$4.695. También se observa que el 50% de los encuestados no estaría dispuesto a pagar nada.

Tabla 8: Estadística descriptiva de precio por el servicio de mantenimiento de plantas

Estadígrafo	¿Cuánto pagarías por un servicio de mantenimiento de las plantas de tu hogar?
Cantidad (N)	201
Media	\$10.798
Desviación Estándar	\$18.099
Valor mínimo	\$0
25%	\$0
50%	\$0
75%	\$20.000
Valor Máximo	\$120.000

La tercera pregunta, "¿Cuánto pagarías por un servicio de mantenimiento de las plantas de tu hogar?" muestra una media de \$10.798 pesos, con una gran desviación estándar de \$18.099 pesos. Esto indica una gran variabilidad en la cantidad que las personas están dispuestas a pagar por el servicio de mantenimiento de sus plantas en el hogar. Además, se puede observar que el 50% de los encuestados no estaría dispuesto a pagar nada por este servicio.

5.1.2.3. Análisis Descriptivo y Gráfico

La siguiente etapa, corresponde a la generación del análisis descriptivo y gráfico, para ello utilizaremos el código del Anexo N°4: Generación de estadística gráfica, del presente documento. Este código utiliza la biblioteca seaborn (sb) de Python para crear gráficos de barras que muestran la distribución de las respuestas a las preguntas de la encuesta en función de la variable "cuidados", que indica si la persona encuestada cuida o no plantas.

Primero, se agrupa el dataframe "df" por la variable "cuidados" y se cuenta el número de respuestas en cada grupo con la función ".size()". Esto nos permite tener una idea de la distribución de las respuestas entre las personas que cuidan plantas y las que no. A continuación, se utilizan varias funciones de la biblioteca seaborn para crear diferentes gráficos de barras:

- `sb.catplot(data=df,x=cuidados,kind="count")`: este gráfico muestra la cantidad de personas que cuidan o no plantas en el dataset "df".

- `sb.catplot(x='Sexo',data=df,hue=cuidados,kind="count")`: este gráfico muestra la distribución de las respuestas a la pregunta sobre la variable "sexo" en función de si la persona cuida o no plantas.
- `sb.catplot(x=edad,data=df,hue=cuidados,kind="count")`: este gráfico muestra la distribución de las respuestas a la pregunta sobre la variable "edad" en función de si la persona cuida o no plantas.
- `sb.catplot(x='Estado Civil',hue=cuidados,data=df,kind="count", aspect=6)`: este gráfico muestra la distribución de las respuestas a la pregunta sobre la variable "estado civil" en función de si la persona cuida o no plantas.
- `sb.catplot(x=vivienda,hue=cuidados,data=df,kind="count", aspect=3)`: este gráfico muestra la distribución de las respuestas a la pregunta sobre la variable "vivienda" en función de si la persona cuida o no plantas.
- `sb.catplot(x=trabajo,hue=cuidados, data=df,kind="count", aspect=2)`: este gráfico muestra la distribución de las respuestas a la pregunta sobre la variable "trabajo" en función de si la persona cuida o no plantas.
- `sb.catplot(x=tipotrabajo,data=df,hue=cuidados,kind="count", aspect=2)`: este gráfico muestra la distribución de las respuestas a la pregunta sobre la variable "tipo de trabajo" en función de si la persona cuida o no plantas.
- `sb.catplot(x=tipoplanta,data=df,kind="count",hue=cuidados, aspect=3)`: este gráfico muestra la distribución de las respuestas a la pregunta sobre la variable "tipo de planta" en función de si la persona cuida o no plantas.

En resumen, este código nos permite visualizar la distribución de las respuestas a las preguntas de la encuesta en función de si la persona está interesada en el servicio de cuidado de plantas o no. Esto puede ayudarnos a identificar patrones o tendencias en la forma en que las diferentes variables están relacionadas con el cuidado de las plantas.

Por ejemplo, en el siguiente gráfico se muestra la correlación entre las variables de si la persona tiene o no un trabajo y si está interesada o no en pagar por el cuidado de las plantas durante sus vacaciones:

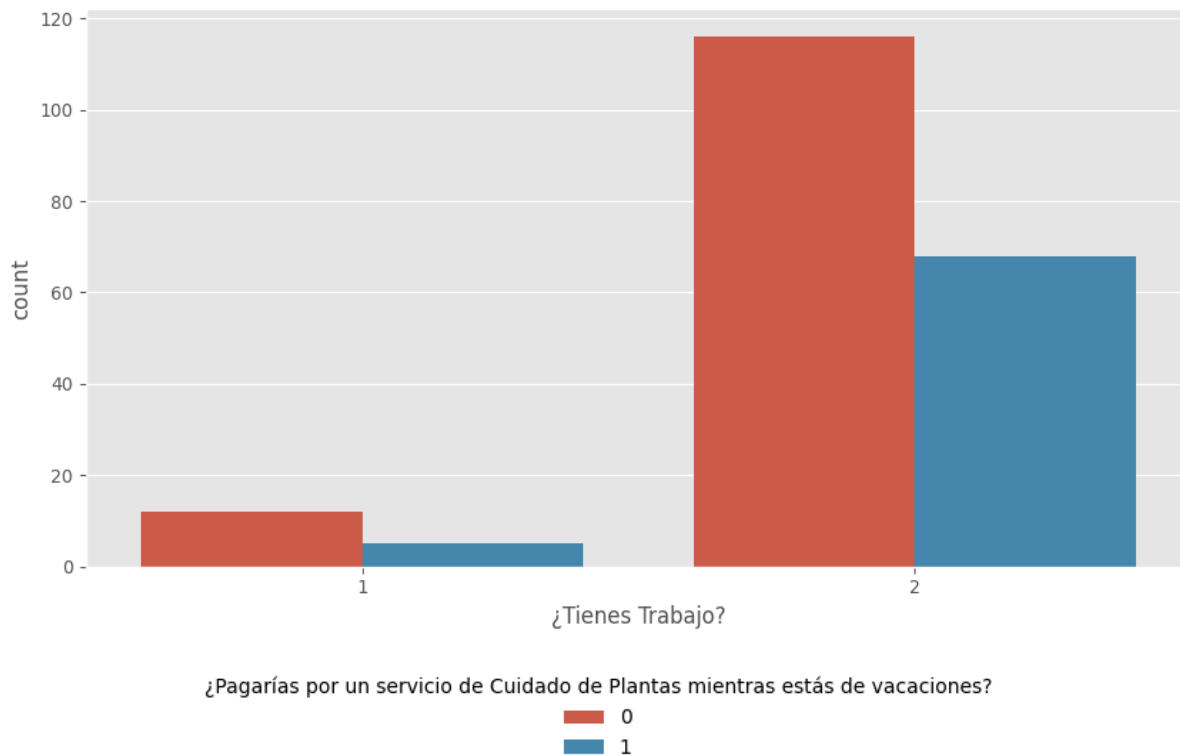


Ilustración 28: Histograma de correlación multivariable

Para este caso en particular, como se mencionó anteriormente, a cada respuesta se le asignó una clase, para las personas que no tienen trabajo, se les asignó el N°1 y para los que sí lo tienen, se les asignó el N°2. Esto se correlaciona con las clases asignadas a las respuestas de las preguntas de si pagarían o no por un servicio de cuidado de plantas durante las vacaciones, siendo el N°0 para las que no pagarían y el N°1 para las que sí pagarían.

De este gráfico podemos desprender que la variable de si tienen trabajo o no es irrelevante para el estudio, puesto que el 91,5% de la muestra declaró tener trabajo, en estos casos, habría que utilizar técnicas de oversampling o subsampling, las cuales consisten en generar o eliminar datos para equiparar variables necesarias para el estudio, este no es el caso puesto que está dirigido a personas con ingresos que pudiesen pagar el servicio.

A su vez, es posible observar que, de las personas con trabajo, la respuesta de que si pagaría o no por un servicio de cuidado de plantas durante sus vacaciones, es alrededor de un 50% más dominante que las que no pagarían. Esto es una variable explicativa y sirve para la ramificación del árbol de clasificación.

Este ejercicio se realiza con el objetivo de balancear los datos de la muestra antes de generar el árbol de decisión, a fin de que este árbol clasifique de la mejor manera

posible los datos al momento de suprimir una variable. Luego del balanceo de cada variable, se procede con la generación del árbol.

5.1.2.4. Aplicación del Algoritmo de Clasificación

Antes de generar el árbol de decisión, es recomendable evaluar la mejor profundidad del árbol en función de la cantidad de datos que se tiene utilizando validación cruzada. Esta técnica entrena y evalúa un modelo de árbol de decisión para cada nivel de profundidad posible y almacena las precisiones promedio obtenidas en cada nivel. Luego, muestra los resultados en una tabla. Esto se hace utilizando el código del Anexo N°5 del presente documento y la variable “mantenimiento” que es, en definitiva, si paga o no paga por un servicio de mantenimiento de plantas. Una vez que se ejecuta el código, comienza asignando el valor “mantenimiento” a una variable llamada “variable”. Luego, se crea un objeto “KFold” que se utilizará más adelante para dividir los datos en 4 segmentos en la validación cruzada. A continuación, se inicializa una lista llamada “accuracies” que se utilizará para almacenar las precisiones promedio obtenidas en cada nivel de profundidad del árbol, se determina el número máximo de atributos en un dataframe llamado “df” y se crea un rango de números que representa los posibles valores de profundidad del árbol.

Luego, comienza un bucle que recorre cada nivel de profundidad posible. Para cada nivel de profundidad, se inicia una lista llamada “fold_accuracy” para almacenar las precisiones obtenidas en cada fold de validación cruzada. A continuación, se crea un objeto “DecisionTreeClassifier” con ciertos parámetros predefinidos. Este objeto será utilizado para entrenar un modelo de árbol de decisión en cada iteración del bucle externo.

El siguiente bucle itera sobre cada par de folds de entrenamiento y validación generados por el objeto “KFold”. En cada iteración, se seleccionan las filas correspondientes a los folds de entrenamiento y validación actuales. A continuación, se entrena el modelo de árbol de decisión utilizando el conjunto de entrenamiento actual y se evalúa en el conjunto de validación actual para calcular la precisión del modelo.

La precisión obtenida en el fold de validación actual se agrega a la lista “fold_accuracy”. Este proceso se repite para todos los folds de validación en el nivel de profundidad actual. Una vez que se completan todos los folds de validación para el nivel de profundidad actual, se calcula el promedio de las precisiones obtenidas y se agrega a la lista “accuracies”.

Después de que se han recorrido todos los niveles de profundidad, se crea un nuevo Data Frame (o tabla) llamado "df2" que almacena los resultados obtenidos. El Data Frame tiene dos columnas: "Max Depth", que representa el nivel de profundidad del árbol, y "Average Accuracy", que representa la precisión promedio obtenida en ese nivel de profundidad. Finalmente, se imprime la tabla "df2" sin mostrar los índices de fila.

Tabla 9: Exactitud del modelo

Max Depth	Average Accuracy
1	0,995
2	0,995
3	0,995
4	0,995
5	0,995
6	0,995

Luego de verificada la precisión promedio del árbol, se procede a generar la representación gráfica de este utilizando el código del Anexo N°6 del presente documento. El resultado corresponde a la representación gráfica de nuestro árbol de decisión sin ajustes de rebalanceo de datos u otro método.

5.2. Análisis de Resultados

5.2.1. Resultados sin ajustes

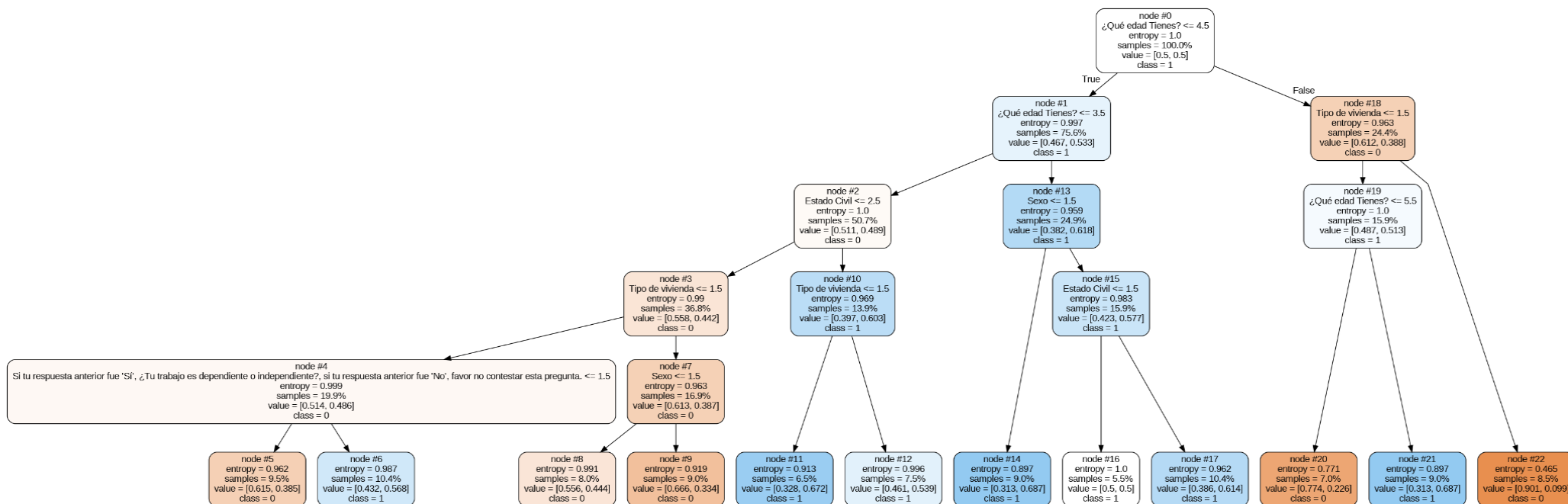


Ilustración 29: Árbol de decisión sin ajustes

En la gráfica vemos, un nodo raíz que hace una primera subdivisión por la edad y las salidas van a izquierda por True que sea menor o igual a 3.5, es decir los grupos de edades 1, 2 y 3 (eran los grupos etáreos de 20 a 34 que menos pagaban por el servicio) y a derecha en False van los grupos etarios del 3 en adelante que sí contestaron que pagarían por el servicio

En el segundo nivel vemos que la cantidad de muestras (samples) queda repartida en 151 y 50 respectivamente. A medida que bajamos de nivel veremos que los valores de entropía se aproximan más a 1 cuando el nodo tiene más muestras que sí pagarían por el servicio (color azul) y se acercan a 0 cuando hay mayoría de muestras que no pagaría por el servicio (color naranja).

En los diversos niveles veremos divisiones por sexo, edad, tipo de vivienda, estado civil o tipo de trabajo.

Debido a la exactitud del modelo sin ajustes (95%), los nodos del último nivel tienen una entropía cercana a 1, lo cual hace que sean muy exactos en la combinación de las variables para su predicción. En este caso, la cantidad total de personas que pagarían por el servicio es de 43 personas basadas en las diferentes combinaciones de sexo, estado civil, edad, vivienda y trabajo.

Cabe destacar que, el valor de exactitud del modelo, se debe a que esta versión sin ajustes utiliza todas las variables de la muestra seleccionada para predecir, por lo que es muy exacto, ahora, para efectos de ajustar el modelo y efectuar un ejercicio de predicción de una muestra dada, se eliminarán las variables de la encuesta que no se pueden obtener sin hacer un sondeo como el realizado, básicamente, se mantendrán las variables de sexo, trabajo, tipo de vivienda, edad, estado civil, plantas caras con el objetivo de predecir la variable de si paga o no por el servicio de cuidado de plantas.

5.2.2. Resultados con ajustes

Se propone eliminar ciertas variables para la construcción del árbol de decisión, debido a que estas condicionan o están directamente relacionadas con lo que queremos predecir, en este caso, si pagarían o no por un servicio de cuidado de plantas durante la ausencia prolongada. Estas variables que se decidieron dropear¹¹ son básicamente variables que se explican entre otras, como por ejemplo el género frente al sexo¹², la preferencia por plantas de interior y/o exterior y lo relacionado al pago por el servicio de mantenimiento y a su vez, no son fáciles de conseguir en base a los estudios disponibles de una población. Se mantuvieron variables que tienen que ver, principalmente, con la caracterización del público potencial y la decisión fundamental de si pagarían o no por un servicio de cuidado de plantas. Las variables en torno a las cuales se generará el árbol de decisión finalmente son:

- **Sexo:** Este atributo nos ayuda a caracterizar a la población para poder predecir temas como, por ejemplo, “los hombres son los que pagan principalmente por el servicio ofrecido”.
- **Edad:** Este atributo nos permite generar una segmentación en torno a la edad del público objetivo y además predecir en qué tramo etáreo podremos vender el servicio
- **Estado civil:** El estado civil, al igual que los dos atributos anteriores, nos sirve para segmentar e identificar si los usuarios que adquieran el servicio se encuentran casados o solteros. Para este caso en particular, en nuestra muestra, se categorizaron los usuarios que se identificaron como viudos dentro de la categoría de solteros con el objetivo de balancear la variable de estado civil.
- **Tipo de vivienda:** El tipo de vivienda, nos permite categorizar si es que nuestro público objetivo vive en casa o departamento. Para el caso de nuestra muestra, los encuestados que dijeron vivir en townhouse, se los categorizó como casa, con el objetivo de balancear la variable de tipo de vivienda.
- **Trabajo:** Esta es una variable cualitativa muy decidora puesto que las personas deberían tener trabajo para pagar por el servicio, por lo que se mantuvo y se categoriza en sí tiene trabajo o no.

¹¹ "Dropear" o "eliminar" variables en el contexto de análisis de datos se refiere a eliminar una columna o conjunto de columnas de un conjunto de datos. Cuando se realiza el proceso de "drop" en Python, se elimina la columna especificada del DataFrame, lo que implica que esa columna ya no estará disponible para su uso en el análisis o modelado.

¹² Esto se transformó en una variable irrelevante debido a que en todos los casos encuestados, el género, se condice con el sexo de forma que son variables directamente relacionadas y se representan entre ellas.

- **Tipo trabajo:** Esta variable busca identificar si la persona usuaria del servicio tiene un trabajo dependiente o independiente, con el objetivo de segmentar aún más nuestro público objetivo.
- **Planta cara:** Esta variable nos habla de si la persona le gustan las plantas o no, por lo que es representativa respecto a la variable del gusto de las plantas y además permite segmentar.
- **Cuidados:** Finalmente, la variable cuidados, la cual representa si la persona pagaría o no por un servicio de cuidado de plantas, es la variable fundamental en la que se centra el estudio y es la pregunta que se predecirá luego de entrenado el modelo.

El modelo de clasificación, con las variables antes mencionadas, muestra precisiones promedio obtenidas para cada profundidad en torno a lo siguiente:

Tabla 10: Exactitud del modelo preajustado

Max Depth	Average Accuracy
1	0,5623
2	0,5623
3	0,5623
4	0,5623
5	0,5623
6	0,5623

Esto, utilizando el código explicado en el capítulo 5.1.2.4 Aplicación del Algoritmo de Clasificación del presente documento. En este caso, podemos apreciar que la precisión promedio del modelo de clasificación, disminuyó en un 43,27% esto se debe a que al quitar variables del modelo se está impactando la exactitud para la predicción.

la precisión promedio por nivel, aún puede ser mejorada utilizando la matriz de confusión. La matriz de confusión es una herramienta fundamental en la evaluación de modelos de clasificación. Proporciona una representación visual de la precisión de un modelo al mostrar cómo los datos de prueba se clasifican en diferentes categorías en comparación con las etiquetas reales.

La matriz de confusión es una tabla de 2x2 (en el caso de clasificación binaria) o de $n \times n$ (en el caso de clasificación multiclase), donde n es el número de clases. En una matriz de confusión binaria, los elementos de la matriz representan lo siguiente:

- Verdaderos positivos (True Positives, TP): Casos en los que el modelo predijo correctamente una muestra como positiva cuando realmente lo era.
- Verdaderos negativos (True Negatives, TN): Casos en los que el modelo predijo correctamente una muestra como negativa cuando realmente lo era.
- Falsos positivos (False Positives, FP): Casos en los que el modelo predijo incorrectamente una muestra como positiva cuando en realidad era negativa (también conocidos como "error tipo I").
- Falsos negativos (False Negatives, FN): Casos en los que el modelo predijo incorrectamente una muestra como negativa cuando en realidad era positiva (también conocidos como "error tipo II").

La matriz de confusión proporciona información valiosa para evaluar el rendimiento del modelo en términos de precisión, recall (también conocido como sensibilidad o tasa de verdaderos positivos) y especificidad (tasa de verdaderos negativos). Estas métricas se calculan utilizando los valores de la matriz de confusión.

Además, la matriz de confusión permite visualizar los errores de clasificación del modelo, lo que puede ayudar a identificar patrones y áreas en las que el modelo puede tener dificultades. Esto es especialmente útil para comprender los tipos de errores que el modelo está cometiendo y ajustar o mejorar el modelo en consecuencia.

En resumen, la matriz de confusión es una herramienta que proporciona información detallada sobre el rendimiento de un modelo de clasificación y permite evaluar su capacidad para predecir correctamente las diferentes clases en un problema de clasificación.

Para aplicar dicha matriz de confusión a nuestro modelo, utilizamos el código del Anexo N°7 del presente documento el cual nos arroja el siguiente resultado:

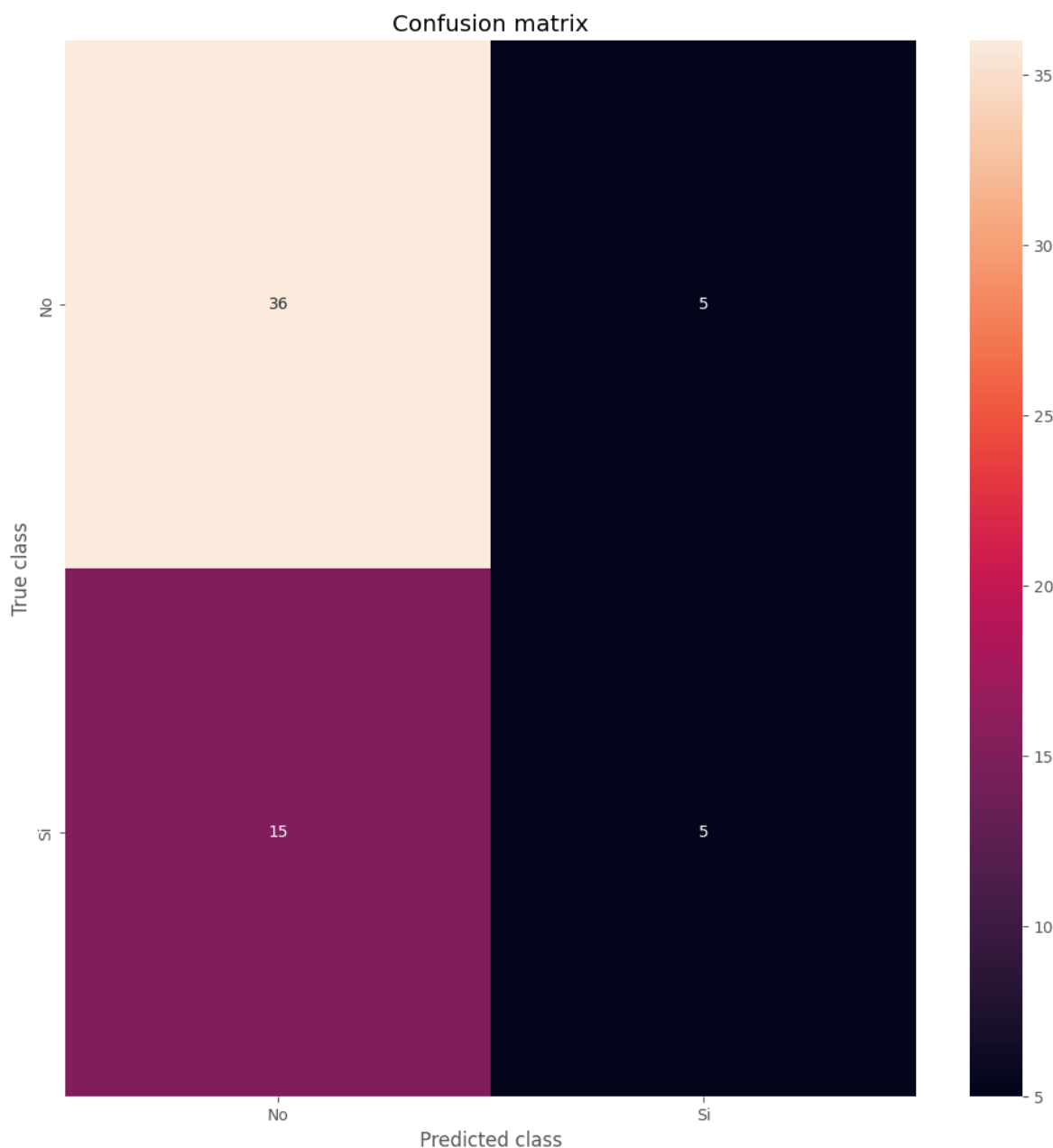


Ilustración 30: Matriz de confusión del modelo

Como se puede observar en la imagen, al comparar los valores reales con la capacidad de predicción del modelo, el modelo sólo fue capaz de predecir sólo cinco valores reales cuando realmente lo fueron. Si aplicamos un código de penalización (ver Anexo N°8 del presente documento) el cual corresponde a un modelo de regresión logística utilizando el algoritmo de penalización "L2" (ridge) y la opción de "balanced class_weight" (peso de clase equilibrado) en la regresión logística.

Después de entrenar el modelo utilizando los datos de entrenamiento y las etiquetas correspondientes, se hace una predicción utilizando el conjunto de prueba. Esto se realiza mediante el uso del método predict del modelo entrenado, pasando como argumento las características (X_{test}) del conjunto de prueba. A continuación, se menciona la función mostrar_resultados(y_{test} , $pred_y$):

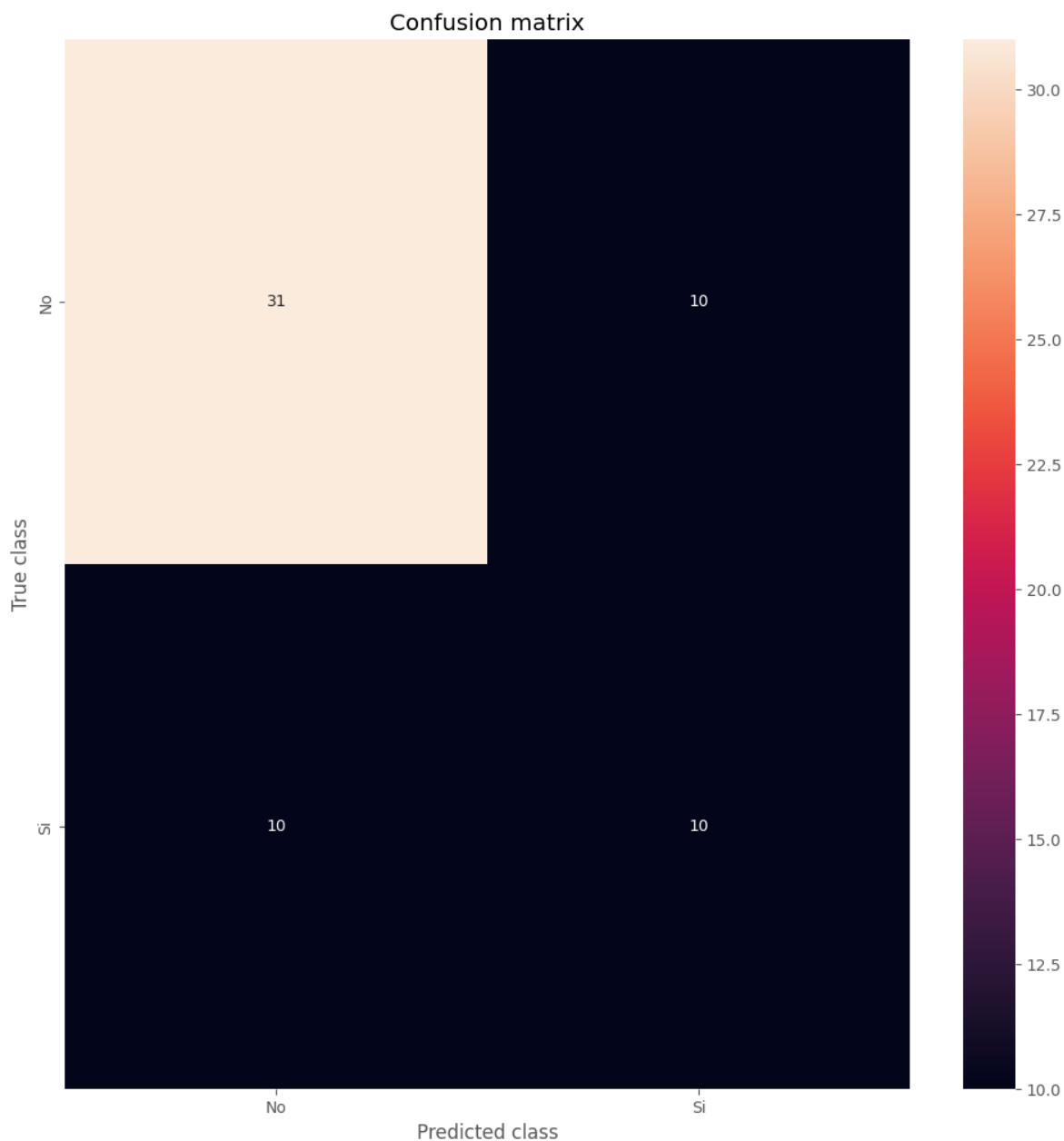


Ilustración 31: Matriz de confusión penalizada

La capacidad de predicción de la muestra aumenta en un 50% al lograr predecir de la misma muestra diez casos verdaderamente correctos. Esto en conjunto con la aplicación del código del Anexo N°9 el cual, a diferencia del que se utilizó para

obtener la precisión promedio en el capítulo 5.1.2.4 – Anexo N°5 del presente documento, realiza una búsqueda aleatoria de los mejores parámetros para el modelo de árbol de decisión utilizando la métrica de exactitud. Luego, muestra el promedio de la exactitud obtenida en cada profundidad del árbol. Esto permite evaluar el rendimiento del modelo con diferentes combinaciones de parámetros y encontrar los valores óptimos para la predicción de la variable "cuidados". Todo lo anterior, mejora la precisión promedio obtenida para cada profundidad en torno a un 6% como se aprecia en la siguiente tabla:

Tabla 11: Exactitud del modelo ajustado

Max Depth	Average Accuracy
1	0,6216
2	0,6216
3	0,6216
4	0,6216
5	0,6216
6	0,6216

Sin embargo, el accuracy o exactitud del modelo aún puede ser optimizado, utilizando una técnica denominada "RandomizedSearchCV" la cual, es una técnica de optimización de hiperparámetros que se utiliza para buscar la mejor combinación de parámetros en un modelo de aprendizaje automático.

La técnica "RandomizedSearchCV" es una variante del enfoque de búsqueda exhaustiva llamado "GridSearchCV". A diferencia de GridSearchCV, que prueba todas las combinaciones posibles de parámetros, RandomizedSearchCV selecciona aleatoriamente un subconjunto de combinaciones de parámetros para evaluar de la siguiente manera:

- Primero, se define un espacio de búsqueda especificando los hiperparámetros que se desean optimizar y los rangos de valores posibles para cada hiperparámetro, el espacio de búsqueda está definido en la variable "params".
- Luego, se realiza una selección aleatoria de combinaciones de parámetros, en lugar de evaluar todas las combinaciones posibles, RandomizedSearchCV selecciona aleatoriamente un número determinado

de combinaciones de parámetros para probar, esto se controla mediante los parámetros "n_iter" y "random_state".

- Después se realiza una evaluación del modelo para cada combinación de parámetros seleccionada aleatoriamente, se entrena y evalúa el modelo utilizando validación cruzada. El rendimiento del modelo se mide utilizando una métrica específica, como precisión, precisión-recall, AUC-ROC, para nuestro caso, utilizaremos la métrica "accuracy".
- Finalmente se selecciona el mejor conjunto de parámetros, una vez que se han evaluado todas las combinaciones de parámetros, se selecciona el conjunto de parámetros que proporcionó el mejor rendimiento según la métrica especificada.

La principal ventaja de "RandomizedSearchCV" sobre "GridSearchCV" es que permite explorar un espacio de búsqueda más amplio en menos tiempo, ya que no requiere probar todas las combinaciones posibles. Sin embargo, existe una compensación entre exploración exhaustiva y eficiencia computacional.

En resumen, "RandomizedSearchCV" es una técnica que realiza una búsqueda aleatoria de los mejores hiperparámetros para un modelo de aprendizaje automático. Ayuda a encontrar combinaciones óptimas de parámetros de manera eficiente, evitando la necesidad de evaluar todas las combinaciones posibles.

Para aplicar la técnica RandomizedSearchCV, utilizaremos el código del Anexo N°10 del presente documento. El resultado obtenido, corresponde al siguiente:

Tabla 12: Valores óptimos del modelo ajustado

Parámetro	Valor
n_folds	8.000000
n_iter	9.000000
random_state	82.000000
average_accuracy	0.636538
Name:	6881
dtype:	float64

Al insertar los parámetros de número de iteraciones (n_iter), número de folds (n_folds) y el estado aleatorio (random_state), pudimos mejorar el porcentaje de exactitud del modelo de un 62% a un 64%.

Se hizo correr el modelo en varias iteraciones identificando que su accuracy va desde un 60% a un 64% en su mejor puntuación.

En cuanto a la representación gráfica se encuentra expresa en el siguiente diagrama:

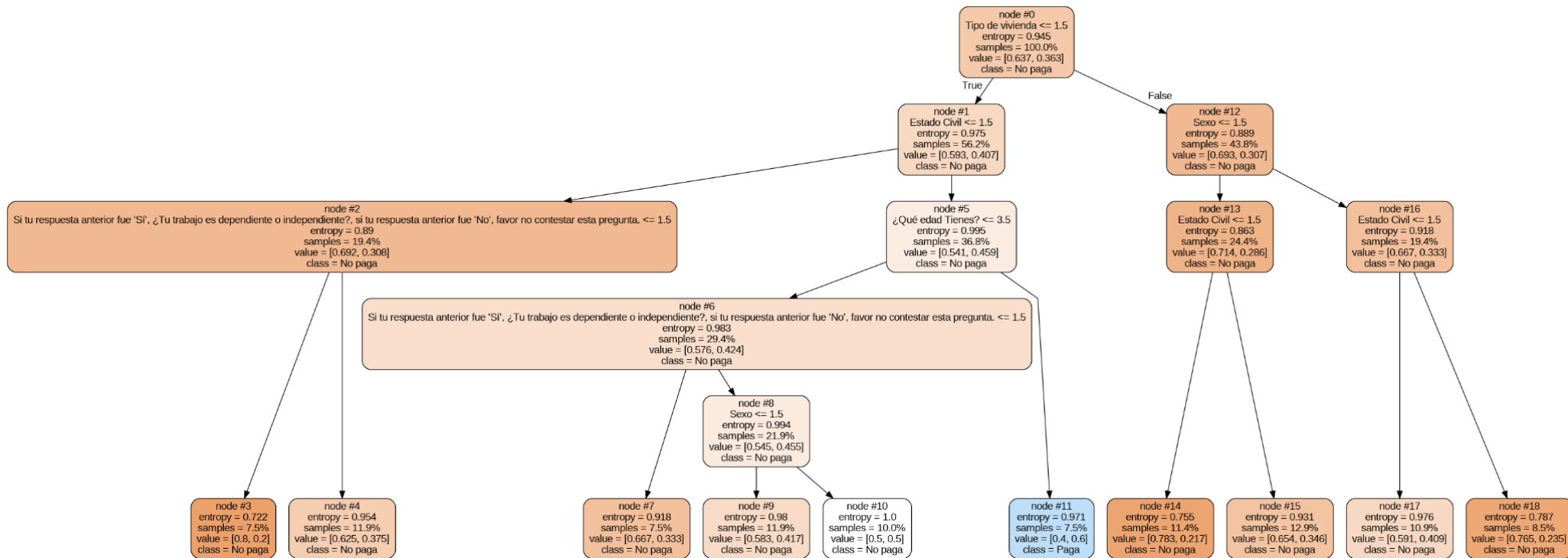


Ilustración 32: Árbol de decisión del modelo ajustado

6. Testeo del Modelo

6.1. Criterios para la Creación de la Muestra

Para el desarrollo del testeo se utilizará la generación de una muestra basada en los datos de censo y otras fuentes¹³ utilizando el método de muestreo estratificado utilizando una muestra de un N de 17.574.003 y un n de muestra de 6.727.117 utilizando las variables identificadas.

La caracterización de la muestra se realizó en función de los principales atributos sobre los cuales el árbol de decisión genera los nodos y tomando en consideración, que estos criterios son de los cuales se encuentra información disponible sin la necesidad de realizar encuestas. Estos son:

- **Edad:** Basada en el último censo de la población chilena realizado el año 2017 se utiliza una distribución etárea con los siguientes valores:

Tabla 13: Población Objetivo por edad y sexo

Rango Etáreo	Hombres	Mujeres
20 - 24	687.110	667.753
25 - 30	747.701	734.933
30 - 34	658.648	658.336
35 - 39	612.595	627.258
40 - 44	578.676	601.714
45 - 49	567.649	601.745
50 - 54	569.365	612.711
55 - 59	520.184	570.017
> 60	1.356.960	1.688.862

Fuente: INE

Por lo tanto, para la distribución de nuestra muestra de 17.574.003 de personas, utilizaremos un universo muestral de 74,33%, lo que corresponde a **13.062.217** personas.

¹³ Ver en Bibliografía del presente documento.

- **Sexo de la Población Chilena:** Para la distribución del sexo de la población chilena, se calculará utilizando la distribución de hombres y mujeres existente del censo 2017, la cual se distribuye de la siguiente manera:

Tabla 14: Distribución Porcentual por sexo de la población chilena el 2017

Censo	Total	Hombres	Mujeres
2017	17.574.003	48,95%	51,05%

Fuente: INE

Por lo tanto, de los 13.062.217 integrantes del universo muestral, un 48,95% serán hombres, es decir 6.394.219 personas y un 51,05% mujeres, es decir 6.668.261 personas.

- **Empleo:** Para establecer esta variable, se utilizó el boletín estadístico de empleo trimestral emitido por el Instituto Nacional de Estadística (INE) el cual establece la tasa de participación en la fuerza de trabajo de la población chilena, la cual, para el año 2017 era 65,1% de la cual un 58,8% corresponde a la tasa de ocupación y un 9,6% correspondiente a la tasa de desocupación. Por lo tanto:

De un universo de 17.574.003 personas, un 65,1% componen la tasa de participación en la fuerza de trabajo correspondiente a 11.440.676 personas, de las cuales un 58,8% están “ocupadas” o trabajando formalmente¹⁴ (6.727.117 personas).

- **Tipo de Empleo:** Para establecer el tipo de empleo, utilizaremos como universo de trabajo el cálculo anterior, es decir, 6.727.117 personas ocupadas formalmente, de los cuales, nos interesa saber la distribución de empleos dependientes e independientes. Para ello, utilizaremos las estadísticas del trabajo publicadas por el INE, en las cuales, se puede apreciar que para el año 2017, 2.787.045 trabajadores son independientes (41,43% del universo de trabajadores ocupados) y 3.940.072 corresponden a trabajadores dependientes (58,57% del universo de trabajadores ocupados).
- **Tipo de Vivienda:** Se utilizará una distribución de casas o departamentos según lo que establece el último censo del 2017 realizado por el INE, en el cual se verificó que de 6.499.355 viviendas existentes a nivel nacional,

¹⁴ Se excluye de la muestra el trabajo informal.

5.167.728 corresponden a casas (79,51%) y 1.138.062 corresponden a departamentos (17,51%).¹⁵

- **Estado civil:** Para establecer esta variable, se utilizarán los datos del censo de 2017, el cual menciona que el estado civil de los chilenos se distribuye en un 44,31% de casados y un 47,23% en otro estado civil.

Con estas definiciones, se procede con la construcción de la muestra para posteriormente poder predecir la variable de si pagaría o no por un servicio de cuidado de plantas.

6.2. Creación de la muestra

Para la generación del Data Frame con los criterios antes mencionados, se utilizó un método de muestreo estratificado y jerarquizado, el cual consiste en una combinación del muestreo estratificado y el muestreo jerarquizado.

El muestreo estratificado es una técnica de muestreo utilizada para obtener una muestra representativa de una población dividida en subgrupos o estratos. En lugar de tomar una muestra aleatoria simple de toda la población, primero se divide la población en grupos homogéneos llamados estratos. Luego, se realiza una muestra aleatoria simple dentro de cada estrato.

Por otra parte, el muestreo jerarquizado (o por conglomerados) es una técnica de muestreo en la que se toma una muestra aleatoria de grupos más grandes (conglomerados) y luego se seleccionan submuestras aleatorias de estos conglomerados para obtener la muestra final.

En algunos casos, es posible que se desee utilizar una combinación de ambas técnicas para obtener una muestra representativa de una población compleja y jerárquica. En este enfoque combinado, primero se estratifica la población en grupos homogéneos, y luego se seleccionan conglomerados al azar dentro de cada estrato. Finalmente, se toman muestras aleatorias de las unidades individuales dentro de los conglomerados seleccionados.

Este enfoque puede ser útil cuando se enfrenta a datos que presentan múltiples niveles jerárquicos como por ejemplo en el caso de la muestra conglomerados por edades, trabajo y tipo de trabajo en nuestro caso

¹⁵ el 3% restante, corresponde a otros tipos de vivienda tales como viviendas tradicionales indígenas, piezas en edificios, mediagua, móvil o viviendas colectivas las que no serán consideradas para el presente estudio.

En resumen, el muestreo estratificado y jerarquizado son técnicas que permiten obtener muestras representativas de poblaciones complejas y ayudan a mejorar la precisión de las estimaciones para subgrupos específicos dentro de la población.

Para la generación de la muestra utilizando el muestreo estratificado y jerarquizado sobre los conjuntos de datos antes mencionados, se utilizó el código del Anexo N°11 del presente documento. Este código nos entrega un dataframe de 17.574 filas y 6 columnas las cuales están organizadas de la misma forma que el data frame original (con el que se entrenó el modelo para realizar las predicciones).

6.3. Predicción de la Variable “Cuidados”

Hay que recordar que la variable que se definió para aperturar el árbol de decisión, fue la denominada variable de “Cuidados”, la cual, corresponde a la pregunta N°13 de la encuesta realizada¹⁶. Esta variable responde a si un determinado individuo pagaría o no por el servicio de cuidado de sus plantas durante una ausencia prolongada¹⁷. Para predecir dicha variable, se utilizó el código del Anexo N°12 del presente documento, el cual, es el mismo código utilizado para generar la representación gráfica del árbol de decisión pero con la diferencia de que se está utilizando como variable de entrenamiento el data frame generado anteriormente. Luego, se le solicita que agregue la variable predicha como una columna más y que grafique la cantidad de “Sí” con el fin de visualizar cuantos individuos pagarían por el servicio. El resultado fue el siguiente:

¹⁶ ver capítulo 5.1.1 fuente de recolección de datos.

¹⁷ Para efectos del estudio, se propuso como ejemplo de ausencia prolongada las vacaciones.

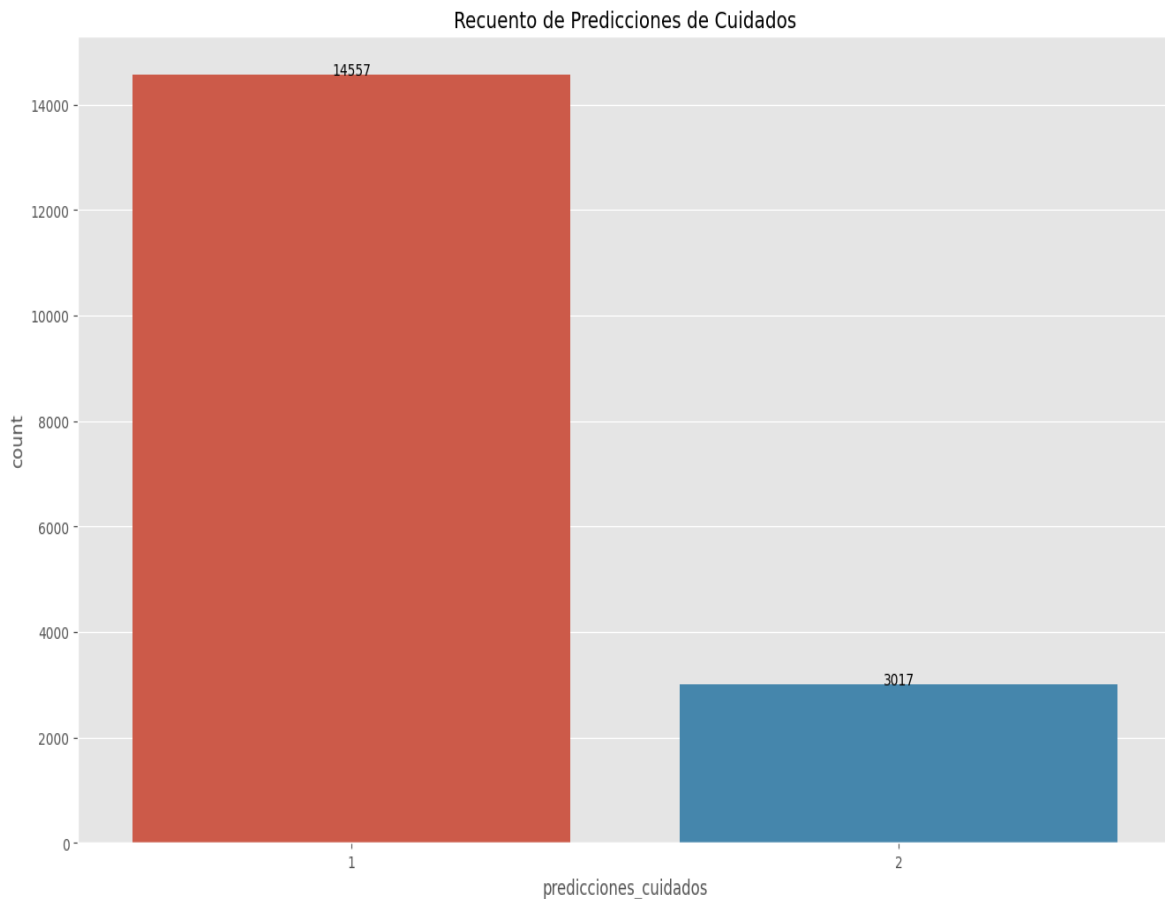


Ilustración 33: Histograma de predicciones hechas por el modelo

El modelo fue capaz de predecir con un 64% de exactitud que, para una muestra de 17.574 individuos con características definidas en torno a sus atributos de sexo, edad, trabajo, tipo de trabajo, estado civil, vivienda y tipo de vivienda, un 17,17% (3.017) pagarían por un servicio de cuidado de plantas durante una ausencia prolongada.

7. Conclusiones y Recomendaciones

A modo de contexto general, el estudio realizado se basó en una exploración alternativa para la evaluación clásica de un proyecto, tomando en cuenta una encuesta realizada en un tiempo acotado la cual puede denotar ciertos sesgos como lo son la edad y el sexo de los encuestados. En definitiva, el presente estudio muestra un acercamiento a las nuevas metodologías de análisis de los proyectos basadas en métodos predictivos como input para perfeccionar las decisiones.

Como conclusión primaria, es posible decir que sí existen individuos que pagarían por un servicio de cuidado de plantas durante un periodo de ausencia prolongado, en ese sentido, se corrobora que el modelo de negocios planteado podría ser exitoso dado que fue posible predecir clientes utilizando el modelo predictivo planteado en el presente estudio.

En torno al comportamiento y familiaridad del concepto “Plantsitter”, pudimos observar durante el estudio que existe una cultura creciente en torno a la tenencia de plantas, puesto que más de un 50% de los encuestados asume conocer de plantas y por otra parte, alrededor de un 43% comentó que posee plantas “caras” en su hogar. Esto plantea que los negocios en torno a las plantas van en aumento y probablemente exista mucho campo por explotar.

Respecto de la fuente de recolección de datos, es evidente que a mayor cantidad de datos o encuestados, la predicción del modelo alcanzará mejores niveles que un 64%, esto considerando que se consultó por variables que no impactan directamente en el estudio como lo fueron:

- Género del participante
- Gusto por las plantas
- Si prefiere plantas de interior o exterior
- Si conoce o no de plantas
- Cantidad de plantas que posee
- Si tiene o no plantas caras
- Precios
- entre otros

Estas preguntas en particular pudieron haber marcado algún tipo de sesgo en los participantes del estudio, a modo de recomendación, se debe ahondar más en el desarrollo del modelo de negocio con el fin de recolectar variables y/o decisiones que afecten la decisión de pago en particular por sobre las apreciaciones y/o gustos.

en cuanto al modelo seleccionado para construir el árbol de decisión de clasificación, al momento de generar el árbol utilizando el conjunto de prueba correspondiente a la data extraída desde la encuesta, podemos observar que el

modelo clasifica los resultados con un 99,5% de exactitud, esto se debe principalmente a la disponibilidad de información del modelo al momento de generar los nodos de decisión, lo que en comparación al modelo con ajustes, el cual evidencia como máximo un 64% de exactitud, es bastante superior, pero sigue siendo complejo alcanzar dichos niveles con la información disponible realmente como lo son los datos del censo (por ejemplo).

A modo de recomendación, se plantea que posiblemente el estudio podría haber mostrado resultados más ajustados (o no) ajustándolo a través de la utilización de otros métodos de predicción como por ejemplo random forest), regresión logística múltiple, redes neuronales, support vector machine, entre otros. Sin embargo, no fue la finalidad del estudio realizado, puesto que el foco era la predicción de una variable en función de ciertas preferencias detectadas concluyendo satisfactoriamente.

Otra recomendación, viene por el lado de mejorar la encuesta de recolección de datos, dado que, como se mencionó anteriormente, se generó mucha información que no se utilizó para el estudio.

8. Bibliografía

- Publishing Ethics Resource Kit. (s/f). Pattern Recognition. Elsevier.com. Recuperado el 27 de marzo de 2023, de <http://www.elsevier.com/locate/pr>
- Laboratorio de invasiones biológicas (LIB). (s/f). Udec.cl. Recuperado el 27 de marzo de 2023, de <http://www.lib.udec.cl/>
- Campbell & Stanley, (1966). Diseños experimentales y cuasiexperimentales en la investigación social. Amorrortu Editores.
- R2D3. (s/f). R2d3.us. Recuperado el 17 de abril de 2023, de <http://www.r2d3.us/>
- Alvear, J. O. (s/f). Árboles de decisión y Random Forest. Bookdown.org. Recuperado el 17 de abril de 2023, de <https://bookdown.org/content/2031/arboles-de-decision-parte-i.html>
- Proyecciones de Población. (s/f). Default. Recuperado el 17 de abril de 2023, de <https://www.ine.gob.cl/estadisticas/sociales/demografia-y-vitales/proyecciones-de-poblacion>
- Censo de Población y Vivienda. (s/f). Default. Recuperado el 17 de abril de 2023, de <https://www.ine.gob.cl/estadisticas/sociales/censos-de-poblacion-y-vivienda/centso-de-poblacion-y-vivienda>
- Na. (2018, abril 13). Árbol de Decisión en Python: Clasificación y predicción. Aprende Machine Learning. <https://www.aprendemachinelearning.com/arbol-de-decision-en-python-clasificacion-y-prediccion/>
- 1.10. Decision Trees. (s/f). Scikit-Learn. Recuperado el 17 de abril de 2023, de <https://scikit-learn.org/stable/modules/tree.html>
- Na. (2019, mayo 16). Clasificación con datos desbalanceados. Aprende Machine Learning. <https://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/>
- Wikipedia contributors. (s/f). Strelitziaceae. Wikipedia, The Free Encyclopedia. <https://es.wikipedia.org/w/index.php?title=Strelitziaceae&oldid=136618052>
- Calculadora del tamaño de muestra. (s/f). SurveyMonkey. Recuperado el 17 de abril de 2023, de <https://es.surveymonkey.com/mp/sample-size-calculator/>

- CEPALSTAT Bases de Datos y Publicaciones Estadísticas. (s/f). Cepal.org. Recuperado el 17 de abril de 2023, de <https://statistics.cepal.org/portal/cepalstat/index.html?lang=es>
- Censo. (s/f). Default. Recuperado el 8 de junio de 2023, de <https://www.ine.gob.cl/ine-ciudadano/definiciones-estadisticas/censo>
- Metropolitana, R. (s/f). BOLETÍN ESTADÍSTICO: EMPLEO TRIMESTRAL. Gob.cl. Recuperado el 8 de junio de 2023, de [https://www.ine.gob.cl/docs/default-source/ocupacion-y-desocupacion/boletines/2023/regi%C3%B3n-metropolitana-y-gran-santiago/boletin-de-empleo-region-metropolitana-\(febrero-abril-2023\).pdf?sfvrsn=1b7768be_6](https://www.ine.gob.cl/docs/default-source/ocupacion-y-desocupacion/boletines/2023/regi%C3%B3n-metropolitana-y-gran-santiago/boletin-de-empleo-region-metropolitana-(febrero-abril-2023).pdf?sfvrsn=1b7768be_6)
- (S/f). Gob.cl. Recuperado el 8 de junio de 2023, de <https://www.ine.gob.cl/docs/default-source/ocupacion-y-desocupacion/comites-y-notas-tecnicas>
- CChC - Cámara Chilena de la Construcción. (s/f). Cchc.cl. Recuperado el 8 de junio de 2023, de <https://cchc.cl/centro-de-informacion/indicadores/mercado-inmobiliario-venta-nacional>
- Chile - Pirámide de población. (2022, julio 8). Datosmacro.com. <https://datosmacro.expansion.com/demografia/estructura-poblacion/chile>

9. Anexos

9.1. Anexo N°1: Instalación de Librerías Python

```
!pip install PyDrive
!pip install pyod
!pip install numpy
!pip install matplotlib
!pip install subprocess
!pip install -U seaborn
!pip install imbalanced-learn
!pip install -U IPython
!pip install IPython
!pip install graphviz
%matplotlib inline
import seaborn as sb
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
from sklearn import tree
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
```

```
from IPython.display import Image as PImage
from subprocess import check_call
from subprocess import call
from PIL import Image, ImageDraw, ImageFont
from imblearn.over_sampling import RandomOverSampler
from collections import Counter
```

9.2. Anexo N°2: Configuración para análisis gráfico

```
%matplotlib inline
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

9.3. Anexo N°3: Parametrización de las respuestas

```
link = 'https://drive.google.com/file/d/16-d3wfNL7PoFpyUPo9AFJneXPDlgldbX/view'
id = link.split("/")[-2]
nombrebase = 'Encuesta Proyecto Tesina (respuestas) - Respuestas de formulario 1'
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile(nombrebase)
with open(nombrebase) as f:
```

```
encoding = f.encoding

df = pd.read_csv(nombrebase,sep=",", encoding=encoding,header=0,
index_col=False)

df = df.drop(['Marca temporal'], axis=1)

df = df.fillna(0)

print(df.columns.tolist())

print("Matriz de correlación:",df.corr())

print("Descriptivos inicial;",df.describe())

sexo = 'Sexo'

df[sexo] = np.where(df[sexo] == "Mujer", 1, df[sexo])

df[sexo] = np.where(df[sexo] == "Hombre", 2, df[sexo])

genero = 'Género'

df[genero] = np.where(df[genero] == "Femenino", 1, df[genero])

df[genero] = np.where(df[genero] == "Masculino", 2, df[genero])

edad = '¿Qué edad Tienes?'

df[edad] = np.where(df[edad] == "Entre 20 y 24", 1, df[edad])

df[edad] = np.where(df[edad] == "Entre 25 y 29", 2, df[edad])

df[edad] = np.where(df[edad] == "Entre 30 y 34", 3, df[edad])

df[edad] = np.where(df[edad] == "Entre 35 y 39", 4, df[edad])

df[edad] = np.where(df[edad] == "Entre 40 y 44", 5, df[edad])

df[edad] = np.where(df[edad] == "Entre 45 y 49", 6, df[edad])

df[edad] = np.where(df[edad] == "Entre 50 y 54", 7, df[edad])

df[edad] = np.where(df[edad] == "Entre 55 y 60", 8, df[edad])

df[edad] = np.where(df[edad] == "más de 60", 9, df[edad])

estado_civil = 'Estado Civil'
```

```
df['Estado Civil'] = np.where(df['Estado Civil'] == "Soltera/o", 1,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "En pareja", 2,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "Unión Civil", 3,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "Casada/o", 4,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "Separada/o", 5,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "Viuda/o", 6,
df['Estado Civil'])

vivienda = 'Tipo de vivienda'

df[vivienda] = np.where(df[vivienda] == "Casa", 1, df[vivienda])

df[vivienda] = np.where(df[vivienda] == "Departamento", 2,
df[vivienda])

df[vivienda] = np.where(df[vivienda] == "Town house", 3, df[vivienda])

trabajo = '¿Tienes Trabajo?'

df[trabajo] = np.where(df[trabajo] == "Sí", 2, df[trabajo])

df[trabajo] = np.where(df[trabajo] == "No", 1, df[trabajo])

tipotrabajo = "Si tu respuesta anterior fue 'Sí', ¿Tu trabajo es
dependiente o independiente?, si tu respuesta anterior fue 'No', favor
no contestar esta pregunta."

df[tipotrabajo] = np.where(df[tipotrabajo] == "Dependiente", 2,
df[tipotrabajo])
```

```
df[tipotrabajo] = np.where(df[tipotrabajo] == "Independiente", 1,
df[tipotrabajo])

gusto = '¿Te gustan las plantas?'

df['¿Te gustan las plantas?'] = np.where(df['¿Te gustan las plantas?']
== "Sí", 2, df['¿Te gustan las plantas?'])

df['¿Te gustan las plantas?'] = np.where(df['¿Te gustan las plantas?']
== "No", 1, df['¿Te gustan las plantas?'])

df['¿Te gustan las plantas?'] = np.where(df['¿Te gustan las plantas?']
== "Imparcial (no me gustan y tampoco me molestan)", 3, df['¿Te gustan
las plantas?'])

tipoplanta = '¿Prefieres Plantas de Interior o Exterior?'

df[tipoplanta] = np.where(df[tipoplanta] == "Interior", 4,
df[tipoplanta])

df[tipoplanta] = np.where(df[tipoplanta] == "Exterior", 3,
df[tipoplanta])

df[tipoplanta] = np.where(df[tipoplanta] == "Ambas",2, df[tipoplanta])

df[tipoplanta] = np.where(df[tipoplanta] == "Ninguna", 1,
df[tipoplanta])

conoce_planta = '¿Conoces de plantas?'

df['¿Conoces de plantas?'] = np.where(df['¿Conoces de plantas?'] ==
"Sí", 2, df['¿Conoces de plantas?'])

df['¿Conoces de plantas?'] = np.where(df['¿Conoces de plantas?'] ==
"No", 1, df['¿Conoces de plantas?'])

plantacara = '¿Tienes plantas "caras" en tu hogar?'

df[plantacara] = np.where(df[plantacara] == "Sí", 2, df[plantacara])

df[plantacara] = np.where(df[plantacara] == "No", 1, df[plantacara])
```

```
cuidados = '¿Pagarías por un servicio de Cuidado de Plantas mientras  
estás de vacaciones?'  
  
df[cuidados] = np.where(df[cuidados] == "Sí", 1, df[cuidados])  
df[cuidados] = np.where(df[cuidados] == "No", 0, df[cuidados])  
  
mantenimiento = '¿Pagarías por un servicio de mantenimiento de las  
plantas de tu hogar?'  
  
df[mantenimiento] = np.where(df[mantenimiento] == "Sí", 2,  
df[mantenimiento])  
df[mantenimiento] = np.where(df[mantenimiento] == "No", 1,  
df[mantenimiento])  
  
tipomantenimiento = "¿Te gustaría que el servicio de mantenimiento fuera  
en tu hogar o que fueran a retirar tus plantas y luego te las devuelvan?"  
  
df[tipomantenimiento] = np.where(df[tipomantenimiento] ==  
"Mantenimiento en el hogar", 2, df[tipomantenimiento])  
df[tipomantenimiento] = np.where(df[tipomantenimiento] ==  
"Mantenimiento con retiro y devolución de plantas", 1,  
df[tipomantenimiento])  
  
qplantas = '¿Cuántas plantas tienes en tu hogar aproximadamente?'  
  
df[qplantas] = np.where((df[qplantas] >= 1) & (df[qplantas] <= 5), 1,  
df[qplantas])  
df[qplantas] = np.where((df[qplantas] >= 6) & (df[qplantas] <= 10), 2,  
df[qplantas])  
df[qplantas] = np.where((df[qplantas] >= 11) & (df[qplantas] <= 20), 3,  
df[qplantas])  
df[qplantas] = np.where((df[qplantas] >= 21) & (df[qplantas] <= 10000),  
4, df[qplantas])
```

```
pagodia = '¿Cuánto pagarías (por día) por el servicio de Cuidado de  
Plantas mientras estás de vacaciones?'

df[pagodia] = np.where((df[pagodia] >= 1) & (df[pagodia] < 1000), 1,  
df[pagodia])

df[pagodia] = np.where((df[pagodia] >= 1000) & (df[pagodia] < 2500), 2,  
df[pagodia])

df[pagodia] = np.where((df[pagodia] >= 2500) & (df[pagodia] < 5000), 3,  
df[pagodia])

df[pagodia] = np.where((df[pagodia] >= 5000) & (df[pagodia] < 7500), 4,  
df[pagodia])

df[pagodia] = np.where((df[pagodia] >= 7500) & (df[pagodia] < 10000),  
5, df[pagodia])

df[pagodia] = np.where((df[pagodia] >= 1000) & (df[pagodia] < 12500),  
6, df[pagodia])

df[pagodia] = np.where((df[pagodia] >= 12500) & (df[pagodia] < 15000),  
7, df[pagodia])

df[pagodia] = np.where((df[pagodia] >= 15000) & (df[pagodia] < 25000),  
8, df[pagodia])

df[pagodia] = np.where((df[pagodia] >= 25000) & (df[pagodia] < 30000),  
9, df[pagodia])

df[pagodia] = np.where((df[pagodia] >= 30000), 10, df[pagodia])

pagomant = '¿Cuánto pagarías por un servicio de mantenimiento de las  
plantas de tu hogar?'

df[pagomant] = np.where((df[pagomant] >= 1) & (df[pagomant] < 2500), 1,  
df[pagomant])

df[pagomant] = np.where((df[pagomant] >= 2500) & (df[pagomant] < 5000),  
2, df[pagomant])
```

```
df[pagomant] = np.where((df[pagomant] >= 5000) & (df[pagomant] < 10000),
3, df[pagomant])

df[pagomant] = np.where((df[pagomant] >= 10000) & (df[pagomant] <
15000), 4, df[pagomant])

df[pagomant] = np.where((df[pagomant] >= 15000) & (df[pagomant] <
30000), 5, df[pagomant])

df[pagomant] = np.where((df[pagomant] >= 30000) & (df[pagomant] <
50000), 6, df[pagomant])

df[pagomant] = np.where((df[pagomant] >= 50000), 7, df[pagomant])

df = df
df = df
df[[sexo,edad,estado_civil,vivienda,tipotrabajo,trabajo,cuidados,pagom
ant]]
```

9.4. Anexo N°4: Generación de estadística gráfica

```
df.groupby(cuidados).size()

sb.catplot(data=df,x=cuidados,kind="count")

sb.catplot(x='Sexo',data=df,hue=cuidados,kind="count")

sb.catplot(x=edad,data=df,hue=cuidados,kind="count")

sb.catplot(x='Estado Civil',hue=cuidados,data=df,kind="count",
aspect=6)

sb.catplot(x=vivienda,hue=cuidados,data=df,kind="count", aspect=3)

sb.catplot(x=trabajo,hue=cuidados, data=df,kind="count", aspect=2)

sb.catplot(x=tipotrabajo,data=df,hue=cuidados,kind="count", aspect=2)

sb.catplot(x=tipoplanta,data=df,kind="count",hue=cuidados, aspect=3)
```

9.5. Anexo N°5: Evaluación del nivel de exactitud en función de la variable de mantenimiento

```
variable =mantenimiento

cv = KFold(n_splits=4)

accuracies = list()

max_attributes = len(list(df))

depth_range = range(1, max_attributes + 1)

for depth in depth_range:

    fold_accuracy = []

    tree_model = tree.DecisionTreeClassifier(criterion='entropy',
min_samples_split=20,min_samples_leaf=5,      max_depth      =      10,
class_weight='balanced')

    for train_fold, valid_fold in cv.split(df):

        f_train = df.loc[train_fold]

        f_valid = df.loc[valid_fold]

        model = tree_model.fit(X = f_train.drop([variable],
axis=1), y = f_train[variable].astype('int'))

        valid_acc = model.score(X = f_valid.drop([variable], axis=1),
y = f_valid[variable].astype('int'))

        fold_accuracy.append(valid_acc)

    avg = sum(fold_accuracy)/len(fold_accuracy)

    accuracies.append(avg)

df2 = pd.DataFrame({"Max Depth": depth_range, "Average Accuracy":
accuracies})

df2 = df2[["Max Depth", "Average Accuracy"]]

print(df2.to_string(index=False))
```

9.6. Anexo N°6: generación de representación gráfica del árbol de decisión

```
variable = cuidadosos

y_train = df.loc[:,[variable]].astype('int')
x_train = df.drop([variable], axis=1)

class_names = df[variable].astype('object').unique().tolist()
print(class_names)

max_depth = 10

decision_tree = tree.DecisionTreeClassifier(criterion='entropy',
min_samples_split=20, min_samples_leaf=11, max_depth = max_depth,
class_weight='balanced')

decision_tree.fit(x_train, y_train)

with open(r"tree1.dot", 'w') as f:

    f = tree.export_graphviz(decision_tree, out_file=f,

                             max_depth = max_depth,

                             impurity = True,

                             feature_names = x_train.columns.tolist(),

                             class_names

= ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10'],

                             rounded = True,

                             filled= True,

                             leaves_parallel= True,

                             proportion= True,

                             node_ids= True)

call(['dot', '-Tpng', r'tree1.dot', '-o', r'tree1.png'])
```

```
PImage("tree1.png")
```

9.7. Anexo N°7: Generación de Matriz de confusión

```
#definimos nuestras etiquetas y features
variable = cuidados
y = df[variable]
X = df.drop(variable, axis=1)
LABELS = ['No', 'Si']
#LABELS= ['No', 'Si']
#dividimos en sets de entrenamiento y test
X_train, X_test, y_train, y_test = train_test_split(X, y,
train_size=0.7)

#creamos una función que crea el modelo que usaremos cada vez
def run_model(X_train, X_test, y_train, y_test):
    clf_base = LogisticRegression(C=1.0,penalty='l2',random_state=2,solver="newton-
cg")
    clf_base.fit(X_train, y_train)
    return clf_base

#ejecutamos el modelo "tal cual"
model = run_model(X_train, X_test, y_train, y_test)

#definimos función para mostrar los resultados
def mostrar_resultados(y_test, pred_y):
```

```
conf_matrix = confusion_matrix(y_test, pred_y)

plt.figure(figsize=(12, 12))

sb.heatmap(conf_matrix, xticklabels=LABELS, yticklabels=LABELS,
annot=True, fmt="d");

plt.title("Confusion matrix")

plt.ylabel('True class')

plt.xlabel('Predicted class')

plt.show()

print (classification_report(y_test, pred_y))

pred_y = model.predict(X_test)

mostrar_resultados(y_test, pred_y)
```

9.8. Anexo N°8: Código de Penalización

```
def run_model_balanced(X_train, X_test, y_train, y_test):

    clf = LogisticRegression(C=1.0,penalty='l2',random_state=1,solver="newton-
cg",class_weight="balanced")

    clf.fit(X_train, y_train)

    return clf

model = run_model_balanced(X_train, X_test, y_train, y_test)

pred_y = model.predict(X_test)

mostrar_resultados(y_test, pred_y)
```

9.9. Anexo N°9: evaluación de la exactitud en función de la variable Cuidados

```
variable =cuidados

max_attributes = len(list(df.columns))

cv = KFold(n_splits=10)

params = {

    'max_depth': range(1, max_attributes + 1),

    'min_samples_split': range(2, 21),

    'min_samples_leaf': range(1, 21),

}

tree_model      =      tree.DecisionTreeClassifier(criterion='entropy',
class_weight='balanced')

random_search    =    RandomizedSearchCV(tree_model,      params,      cv=cv,
scoring='accuracy', n_iter=10, random_state=42)

accuracies = list()

depth_range = range(1, max_attributes + 1)

for train_fold, valid_fold in cv.split(df):

    f_train = df.loc[train_fold]

    f_valid = df.loc[valid_fold]

    random_search.fit(X = f_train.drop([variable], axis=1), y =
f_train[variable].astype('int'))

    model = random_search.best_estimator_
```

```
valid_acc = model.score(X = f_valid.drop([variable], axis=1), y =  
f_valid[variable].astype('int'))  
  
accuracies.append(valid_acc)  
  
avg = sum(accuracies)/len(accuracies)  
  
df2 = pd.DataFrame({"Max Depth": depth_range, "Average Accuracy": avg})  
df2 = df2[["Max Depth", "Average Accuracy"]]  
print(df2.to_string(index=False))
```

9.10. Anexo N°10: Código para la Búsqueda Randomizada

```
# Define los parámetros que deseas optimizar  
variable = cuidados  
max_attributes = len(list(df.columns))  
  
# Definiendo rangos de parámetros  
params = {  
    'max_depth': range(1, max_attributes + 1),  
    'min_samples_split': range(2, 21),  
    'min_samples_leaf': range(1, 21),  
}  
  
tree_model = DecisionTreeClassifier(criterion='entropy',  
class_weight='balanced')  
  
# Definir los rangos para n_folds, n_iter y random_state  
n_folds_range = list(range(2, 11))  
n_iter_range = list(range(1, 11))  
random_state_range = list(range(1, 101))  
  
# Crear una lista para guardar los resultados
```

```

results = []

# Iterar sobre cada combinación de n_folds, n_iter, y random_state
for n_folds in n_folds_range:
    for n_iter in n_iter_range:
        for random_state in random_state_range:
            cv = KFold(n_splits=n_folds)

            # Implementar el RandomizedSearchCV
            random_search = RandomizedSearchCV(tree_model, params,
cv=cv, scoring='accuracy',
                                                    n_iter=n_iter,
random_state=random_state)

            accuracies = []

            for train_fold, valid_fold in cv.split(df):
                f_train = df.loc[train_fold]
                f_valid = df.loc[valid_fold]

                # Ajustar el modelo con los mejores parámetros
                random_search.fit(X = f_train.drop([variable],
axis=1), y = f_train[variable].astype('int'))

                model = random_search.best_estimator_

                valid_acc = model.score(X = f_valid.drop([variable],
axis=1), y = f_valid[variable].astype('int'))

                accuracies.append(valid_acc)

            avg_acc = sum(accuracies)/len(accuracies)

```

```
results.append((n_folds, n_iter, random_state, avg_acc))

# Crear un DataFrame a partir de los resultados
df_results = pd.DataFrame(results, columns=['n_folds', 'n_iter',
'random_state', 'average_accuracy'])

# Ordenar los resultados por la precisión media y mostrar el mejor
conjunto de parámetros

best_parameters = df_results.sort_values(by='average_accuracy',
ascending=False).iloc[0]

print(best_parameters)
```

9.11. Anexo N°11: Generación de la muestra utilizando el muestreo estratificado y jerarquizado sobre los conjuntos de datos

```
import numpy as np

import pandas as pd

# Parámetros

poblacion = 17574003

# Tamaño de la muestra18

n = int(0.001 * poblacion)

# Proporciones para cada categoría

proporciones = {

    'Sexo': {'Hombre': 0.4895, 'Mujer': 0.5105},

    'rango_etareo_Hombre': {

        'Entre 20 y 24': 0.11, 'Entre 25 y 29': 0.12, 'Entre 30 y 34':

0.10, 'Entre 35 y 39': 0.10,
```

¹⁸ Cabe destacar que, si bien se utilizó un universo de 17.574.003 personas conforme a lo planteado anteriormente, para la confección del dataframe se utilizará el 1% de la población debido a que utilizar la muestra completa, genera problemas en la consola para la generación del dataframe.

```

        'Entre 40 y 44': 0.09, 'Entre 45 y 49': 0.09, 'Entre 50 y 54':
0.09, 'Entre 55 y 59': 0.08, 'Más de 60': 0.22

    },

    'rango_etareo_Mujer': {

        'Entre 20 y 24': 0.10, 'Entre 25 y 29': 0.11, 'Entre 30 y 34':
0.10, 'Entre 35 y 39': 0.09,

        'Entre 40 y 44': 0.09, 'Entre 45 y 49': 0.09, 'Entre 50 y 54':
0.09, 'Entre 55 y 59': 0.08, 'Más de 60': 0.25

    },

    '¿Tienes Trabajo?': {'Sí': 0.3827, 'No': 1 - 0.3827},

    'tipo_trabajo': {'Independiente': 0.4143, 'Dependiente': 1 -
0.4143},

    'Tipo de vivienda': {'Casa': 0.79, 'Departamento': 0.175, 'Town
house': 1 - 0.79 - 0.175},

    'Estado Civil': {'Casada/o': 0.4431, 'Soltera/o': 1-0.4431}
}

# Crear el dataframe
df = pd.DataFrame()

# Estratificación jerárquica
for sexo, prop_sexo in proporciones['Sexo'].items():

    n_sexo = int(n * prop_sexo) # Número de individuos para este sexo

    rango_etareo =
np.random.choice(list(proporciones[f'rango_etareo_{sexo}'].keys()),
n_sexo, p=list(proporciones[f'rango_etareo_{sexo}'].values()))

    df_sexo = pd.DataFrame({'Sexo': sexo, 'rango_etareo':
rango_etareo})

    df = pd.concat([df, df_sexo])

# Verificar y ajustar el tamaño de la muestra si es necesario
n_actual = len(df)

```

```

if n_actual != n:

    df = df.sample(n, replace=n_actual < n)

# Añadir las otras categorías

for categoria, subcategorias in {k: v for k, v in proporciones.items()
if k not in ['sexo', 'rango_etareo_Hombre',
'rango_etareo_Mujer']}.items():

    df[categoria] = np.random.choice(list(subcategorias.keys()), n,
p=list(subcategorias.values()))

print(df)

tipotrabajo = "Si tu respuesta anterior fue 'Sí', ¿Tu trabajo es
dependiente o independiente?, si tu respuesta anterior fue 'No', favor
no contestar esta pregunta."

sexo = 'Sexo'

df[sexo] = np.where(df[sexo] == "Mujer", 2, df[sexo])
df[sexo] = np.where(df[sexo] == "Hombre", 1, df[sexo])

df.rename(columns = {'rango_etareo':'¿Qué edad Tienes?'}, inplace =
True)

df.rename(columns = {'tipo_trabajo':tipotrabajo}, inplace = True)

edad = '¿Qué edad Tienes?'

df[edad] = np.where(df[edad] == "Entre 20 y 24", 1, df[edad])
df[edad] = np.where(df[edad] == "Entre 25 y 29", 1, df[edad])
df[edad] = np.where(df[edad] == "Entre 30 y 34", 2, df[edad])
df[edad] = np.where(df[edad] == "Entre 35 y 39", 3, df[edad])
df[edad] = np.where(df[edad] == "Entre 40 y 44", 3, df[edad])
df[edad] = np.where(df[edad] == "Entre 45 y 49", 4, df[edad])
df[edad] = np.where(df[edad] == "Entre 50 y 54", 4, df[edad])
df[edad] = np.where(df[edad] == "Entre 55 y 59", 4, df[edad])
df[edad] = np.where(df[edad] == "Más de 60", 5, df[edad])

```

```
df[edad] = df[edad].astype('int')

#Estado civil: se entendió a solteros como aquellos que no tendría un
compromiso es decir solteros, separados y viudos. Por el lado de casados
quedaron Unión Civil, Casados y en pareja

estado_civil = 'Estado Civil'

df['Estado Civil'] = np.where(df['Estado Civil'] == "Soltera/o", 1,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "En pareja", 2,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "Unión Civil", 2,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "Casada/o", 2,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "Separada/o", 1,
df['Estado Civil'])

df['Estado Civil'] = np.where(df['Estado Civil'] == "Viuda/o", 1,
df['Estado Civil'])

# Se homologó Casa a Townhouse

vivienda = 'Tipo de vivienda'

df[vivienda] = np.where(df[vivienda] == "Casa", 1, df[vivienda])

df[vivienda] = np.where(df[vivienda] == "Departamento", 2,
df[vivienda])

df[vivienda] = np.where(df[vivienda] == "Town house", 2, df[vivienda])

trabajo = '¿Tienes Trabajo?'

df[trabajo] = np.where(df[trabajo] == "Sí", 2, df[trabajo])

df[trabajo] = np.where(df[trabajo] == "No", 1, df[trabajo])

tipotrabajo = "Si tu respuesta anterior fue 'Sí', ¿Tu trabajo es
dependiente o independiente?, si tu respuesta anterior fue 'No', favor
no contestar esta pregunta."
```

```
df[tipotrabajo] = np.where(df[tipotrabajo] == "Dependiente", 2,
df[tipotrabajo])

df[tipotrabajo] = np.where(df[tipotrabajo] == "Independiente", 1,
df[tipotrabajo])

print(df)

df2 = df

df2 = df2[[sexo,edad,estado_civil,vivienda,trabajo,tipotrabajo]]

print("este es el df2:",df2)
```

9.12. Anexo N°12: código para predecir la variable de pago por el servicio de mantenimiento

```
# Crear arrays de entrenamiento y las etiquetas
#df.head(5)

variable = cuidados

y_train = df.loc[:,[variable]].astype('int')

x_train = df.drop([variable], axis=1)

class_names = df[variable].astype('object').unique().tolist()

print(class_names)

max_depth = 10

decision_tree = tree.DecisionTreeClassifier(criterion='entropy',
min_samples_split=8, min_samples_leaf=15, max_depth = max_depth,
class_weight=None)

decision_tree.fit(x_train, y_train)

x_test = df2[x_train.columns]

# Ahora puedes realizar las predicciones

predictions = decision_tree.predict(x_test)

# Agrega las predicciones al DataFrame df2 como una nueva columna
```

```
df2['predicciones_cuidados'] = predictions

print(df2.head(5))

# Crear el gráfico de barras

ax = sb.countplot(x="predicciones_cuidados", data=df2)

# Añadir título

plt.title('Recuento de Predicciones de Cuidados')

# Añadir etiquetas a cada barra

for p in ax.patches:

    height = p.get_height()

    ax.text(x = p.get_x()+(p.get_width()/2),

            y = height+0.2,

            s = '{:.0f}'.format(height),

            ha = 'center')

plt.show()
```