ESTUDIO COMPARATIVO ENTRE DIFERENTES ALGORITMOS DE CLASIFICACIÓN PARA REALIZAR LA PREDICCIÓN DE LAS CANCELACIÓNES DE RESERVAS EN UN HOTEL EN CIENAGA MAGDALENA.

LUDVIN EDUARDO RUEDA RINCON

UNIVERSIDAD PONTIFICIA BOLIVARIANA

ESCUELA DE INGENIERIAS Y ARQUITECTURA

PROGRAMA DE INGENIERIA INDUSTRIAL

MONTERIA

2022

ESTUDIO COMPARATIVO ENTRE DIFERENTES ALGORITMOS DE CLASIFICACIÓN PARA REALIZAR LA PREDICCIÓN DE LAS CANCELACIÓNES DE RESERVAS EN UN HOTEL EN CIENAGA MAGDALENA

LUDVIN EDUARDO RUEDA RINCON

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO INDUSTRIA

ASESOR

CESAR LOPEZ MARTINEZ

INGENIERO INDUSTRIAL

UNIVERSIDAD PONTIFICIA BOLIVARIANA

ESCUELA DE INGENIERIAS Y ARQUITECTURA

PROGRAMA DE INGENIERIA INDUSTRIAL

MONTERIA

2022

TABLA DE CONTENIDO

LISTA DE ILUSTRACIONES

LISTADO DE TABLAS

RESUMEN

Uno de los problemas que se presentan a diario en los hoteles, radica en el hecho de que los huéspedes hacen una reserva y no se presentan en el hotel. Los administradores de los hoteles hacen uso del concepto de "Sobre Reservar" las habitaciones u Overbooking para controlar los efectos negativos de lo anteriormente explicado. (Pineda, 2015)

Esto es simplemente, la confirmación de más habitaciones que la capacidad disponible del hotel. De esta forma, se anticipan a las cancelaciones que podrían realizarse, en lugar de dejar escapar a un cliente cuando el hotel está sin capacidad, se le sigue brindando una habitación para que, si otro cliente no se presenta, no permanezca esa habitación libre. Esto ayuda a garantizar que se alcance el máximo volumen en ingresos y ocupación en el hotel.

Al realizar esta práctica sin ningún control, se pueden incurrir en distintas ineficiencias. El caso más llamativo es cuando llegan dos clientes a los cuales se les ha ofrecido la misma habitación, en este punto, las directivas optan por incurrir en costos adicionales para la reubicación del cliente en otro hotel, además de una compensación para no disminuir su nivel de servicio, o, por el contrario, en el caso más extremo, se asume la pérdida del cliente, dañando así su credibilidad. (RESCO, 2013)

En este punto, es de vital importancia que los hoteleros tengan un nivel de certeza en saber si un cliente cancelará o no cancelará su reserva. Para tal fin, se propone implementar técnicas de Aprendizaje Automático o Machine Learning. Este trabajo, buscó encontrar un algoritmo de clasificación que permitiera determinar con alto nivel de certeza si un cliente cancelará o no su reserva. Para tal fin se compararon diferentes algoritmos de clasificación tales como: Arboles de decisión, Clasificador Bayesiano, Bosques Aleatorios, K vecinos más cercanos y Máquina de soporte vectorial. Se utilizaron métodos para balancear la base de datos original, puesto que, en esta, la clase de Cancelaciones se presentaba en menor proporciones. Se utilizaron los métodos de balanceo Smote y Adasyn.

La métrica para comparar los modelos fue el indicador F1. Bajo este enfoque, se concluye que los métodos de balanceo utilizados mejoran la capacidad predictiva de los modelos. El algoritmo máquina de soporte vectorial con kernel radial, fue el que mejor rendimiento obtuvo, teniendo un alto grado de predicción en la clase Cancelaciones y reduciendo los falsos positivos.

Palabras clave: Sobre reservar, Aprendizaje automático, Curva ROC, Métrica F1 Score, Matriz de confusión.

ABSTRACT

One of the problems that arise daily in hotels lies in the fact that guests make a reservation and do not show up at the hotel. Hotel administrators make use of the concept of "Overbooking" rooms or overbooking to control the negative effects of what has been explained above. (Pineda, 2015)

This is simply the confirmation of more rooms than the available capacity of the hotel. In this way, they anticipate the cancellations that could be made, instead of letting a client escape when the hotel is out of capacity, they continue to provide a room so that if another client does not show up, that room will not remain free. This helps ensure that the maximum volume of revenue and occupancy is achieved at the hotel.

When carrying out this practice without any control, different inefficiencies can be incurred. The most striking case is when two clients arrive who have been offered the same room, at this point, the directors choose to incur additional costs for the relocation of the client in another hotel, in addition to compensation so as not to decrease their level. of service, or, on the contrary, in the most extreme case, the loss of the client is assumed, thus damaging its credibility. (RESCO, 2013)

At this point, it is vitally important that hoteliers have a level of certainty in knowing whether or not a client will cancel their reservation. For this purpose, it is proposed to implement Automatic Learning or Machine Learning techniques. This work sought to find a classification algorithm that would allow determining with a high level of certainty whether or not a client will cancel their reservation. For this purpose, different classification algorithms were compared, such as: Decision Trees, Bayesian Classifier, Random Forests, K Nearest Neighbors, and Support Vector Machine. Methods were used to balance the original database, since, in this, the Cancellations class was presented in smaller proportions. The Smote and Adasyn balancing methods were used.

The metric to compare the models was the indicator F1. Under this approach, it is concluded that the balancing methods used improve the predictive capacity of the models. The support vector machine algorithm with radial kernel was the one that obtained the best performance, having a high degree of prediction in the Cancellations class and reducing false positives.

Keywords: Overbooking, Machine Learning, ROC Curve, F1 Score Metric, Confusion Matrix.

INTRODUCCION

En la siguiente propuesta se identificó una metodología para predecir las reservas canceladas en un hotel ubicado en Ciénaga Magdalena por medio de la comparación de diferentes tipos de algoritmos de clasificación basados en Machine Learning.

Se buscó predecir si una reserva es cancelada con el fin de que esta información pueda ser utilizado por los gerentes hoteleros como apoyo en la toma de decisiones, enfocado a la estrategia de maximizar los ingresos y la planificación de recursos, de modo que pueda ser una herramienta para mejorar la competitividad del departamento. Además, se estudiaron los aportes de la rama de la inteligencia artificial Machine Learning en la predicción de reservas utilizando técnicas de algoritmos de clasificación, para ello se requiere el aprovechamiento de los grandes volúmenes de datos disponibles, para una adecuada implementación en el sector hotelero.

Para tal fin, se identificaron las características de mayor impacto, que influyeron en la decisión para que un usuario cancele una reserva de hotel. Para esto, también es necesario entender el impacto que existe por la cancelación de las reservas y comprender el proceso de tarificación del costo de pérdida de una reserva. El estudio se enfocó en un análisis del histórico de reservas emitidas desde de 2017 hasta 2019 donde se tienen aproximadamente 17000 registros. La información de la base de datos esta segmentada de acuerdo con:

- ✓ Mes
- ✓ Habitación
- ✓ Días de reserva.
- ✓ Número de personas
- ✓ Tipo de pago
- ✓ Cliente frecuente
- ✓ Tiempo
- ✓ Solicita Parqueadero
- ✓ Cancelaciones previas
- ✓ Solicita Comida

Con esta información, se compararon diferentes algoritmos tradicionales de clasificación para pronosticar con anticipación las cancelaciones, con el fin de encontrar aquel método que mejor resultado obtenga para que con base en este, se pueden tomar las mejores decisiones.

1. PLANTEAMIENTO DEL PROBLEMA

El pronóstico de ocupación en hoteles es de gran importancia en la toma de decisiones, anticipar la ocupación futura permitirá a los gerentes planificar mejor los inventarios, la producción, la mano de obra, las compras, el presupuesto financiero, la gestión de tarifas, todo con un enfoque en la maximización de los ingresos.

Hoy en día el comportamiento de la demanda en una organización es un tema de vital importancia para la toma de decisiones, pues él no conocer la demanda ciertamente trae consigo problemas como; no estimar con precisión el nivel de ventas, déficit o excedente de inventario, entre otros. Actualmente, la experiencia y la intuición, así como el análisis de los datos históricos más recientes, son algunos de los métodos más utilizados por las organizaciones para hacer frente a este problema. (ÁGUADA, 2008)

Actualmente, la industria hotelera ha adoptado estrategias para maximizar los ingresos por habitación, esta disciplina se conoce como Revenue Management, una herramienta que hace posible vender cada habitación al cliente dispuesto a pagar el precio más alto con el fin de obtener los mayores ingresos (GARAY, 2008), Esta estrategia busca vender el producto correcto, al precio correcto y al cliente correcto en el momento correcto. El Revenue Management se basa en establecer segmentos de clientes, definiendo tarifas adecuadas para cada segmento, y en la previsión de la demanda, determinando a qué precios conviene vender, a quién y cuándo; todo respaldado por modelos matemáticos para optimizar los ingresos, además ofrece flexibilidad para rechazar reservas si no ofrecen los mejores beneficios, todo respaldado por pronósticos.

De acuerdo a (PALLARES, 2014) muchas veces, la aplicación de estas técnicas tiene un impacto negativo en los ingresos debido a cálculos de pronóstico erróneos, lo que contribuye a tomar decisiones erróneas en la planificación y administración de tarifas, por ejemplo, si el pronóstico indica que la ocupación será alta. probablemente la estrategia sea subir el precio y si es bajo, probablemente sea el momento de bajar el precio para promover la venta. Si la previsión no es correcta, puede existir el riesgo de que se venda a un precio excesivamente bajo o de que no se produzca la venta.

Podemos encontrar algunas ventajas y desventajas al momento de aplicar la estrategia de overbooking.

VENTAJAS	DESVENTAJAS
✓ Reducir perdidas. ✓ Lograr plena ocupación: Esto implica que no se perderá ninguna transacción financiera ya que todas las habitaciones siempre estarán ocupadas. ✓ La compensación es más económica que tener una habitación desocupa.	✓ Experiencia negativa del cliente. ✓ Posible mala publicidad: Muchos clientes se asegurarán de revisar las reseñas de otros clientes para saber las opiniones sobre el hotel antes de hacer una reserva.

Tabla 1. ventajas y desventajas al aplicar overbooking. Fuente: Rodrigo Ricardo, 2020.

Por lo tanto, la gestión de overbookings es un proceso complejo, profundamente relacionado con la gestión de ingresos y rentabilidad en establecimientos hoteleros, por este motivo los gerentes de hoteles intentan pronosticar la demanda turística para definir el número óptimo de overbookings, pero desde un punto de vista empírico es extremadamente difícil saber cuántas habitaciones estarán ocupadas en una determinada fecha. (RESCO, 2013)

Ante esta situación surgen diferentes interrogantes; ¿Hay alguna forma más técnica de hacer que estas predicciones sean más precisas? ¿Es suficiente solo considerar el promedio de los datos históricos más recientes para establecer un pronóstico para un período futuro? Bajo estas interrogantes se plantea este trabajo de investigación, por tal motivo surge la necesidad de proponer un modelo de pronóstico en el cual se pueda brindar a la organización toda la información necesaria para la estimación de cancelaciones de futuras reservas. Ante esto se plantea la siguiente pregunta problema que se busca responder con el desarrollo del trabajo de investigación.

Pregunta problema: ¿Qué resultados se pueden obtener al aplicar un algoritmo de Machine Learning en cuanto a su calidad de predicción para estimar las cancelaciones de reservas en el HOTEL PLAYA VERDE CIENAGA -MAGDALENA?

2. OBJETIVOS

2.1. OBJETIVO GENERAL:

Determinar una metodología para predecir las cancelaciones de las reservas en un hotel por medio de la comparación de diferentes algoritmos basados en Machine Learning.

2.2. OBJETIVOS ESPECIFICOS:

- ✓ Identificar diferentes modelos de clasificación basados en Machine Learning por medio de una revisión de literatura que tengan la capacidad de predecir las reservas canceladas en la organización de estudio.

- ✓ Comparar de manera descriptiva los algoritmos de clasificación seleccionados por medio de una métrica de desempeño habitual, con el fin de seleccionar aquel modelo que mejor se ajuste a la situación analizada.

- ✓ Proponer los pasos para una integración del modelo con mejor desempeño a los procesos de la organización por medio de un procedimiento que permita una ruta clara y eficiente para mejorar la toma de decisiones en este ámbito.

3. ESTADO DEL ARTE

En la literatura se encontraron varias investigaciones que tratan sobre la predicción de la cancelación de reservas, pero gran parte de las investigaciones proviene de orientaciones desarrolladas para pronosticar reservas de aerolíneas.

Existe una semejanza entre el problema de predicción de las cancelaciones de las reservas y el de los puestos de un vuelo, sin embargo, existen variables en el problema del hotel que son diferentes al problema de la aerolínea. (CANALIS, 2019)

El trabajo de (Afrianto, 2020) propone un Modelos de predicción de reservas para listados de alojamiento entre pares mediante regresión logística, decision tree, K-NN y clasificadores de bosque aleatorio, utilizaron un conjunto de datos de Airbnb que contan de 77,096 registros y 96 variables. El modelo decision Tree obtuvo la puntuación AUC-ROC más baja, y tambien tuvo el tiempo de procesamiento más bajo. El rendimiento de los modelos de bosque aleatorio en la predicción de la probabilidad de reserva de los listados de alojamiento es el más superior.

Por ejemplo (LEE, 1990) utilizaron el método ARIMA Box-Jenkins y el método de suavizado exponencial de Holt Winter para predecir la ocupación hotelera mensual, muestran que ambos métodos producen un error cuadrático medio (ECM).

En el trabajo de (RAJOPADHYE, 2001) Combina el pronóstico a largo plazo utilizando el método Holt Winter con pronósticos a corto plazo utilizando los datos de las reservas realizadas, al final combina las previsiones para obtener la cifra final. Los métodos de Holt Winter son populares en el modelado de series de tiempo, ya que son métodos simples con resultados exitosos.

Otros ejemplos se encuentran en los métodos ARIMA los cuales son frecuentemente usados en el modelamiento de series de tiempo, ejemplos de la aplicación en el turismo lo encontramos en los trabajos de (JIMENEZ, 2006) en este trabajo se realiza un análisis de la capacidad predictiva del método en una serie con estacionalidad procedente del sector turístico.

Un ejemplo del uso del método "pick-up" basado en información de reserva observada se puede encontrar en el trabajo que desarrolla el pronóstico de habitaciones vendidas y utiliza en su modelo el "pickup" a 7, 14, 30 y 60 díasde reservas, también incluye los días de la semana como variable independiente, en esta investigación se aplican varios modelos de pronóstico como (Último día del año anterior, Media Móvil, Suavizado Exponencial, Método Aditivo y Multiplicativo de Recolección, Regresión Múltiple). Calculan varias medidas de error como MAD (error absoluto medio), MAPE (error porcentual absoluto medio).

En el trabajo de (GARAY, 2008) proponen un modelo de previsión de llegadas y ocupación utilizando la técnica de simulación de Montecarlo. El objetivo del modelo es dar mejores resultados que los enfoques existentes, los resultados del método de Montecarlo se comparan con 5 técnicas:

- ✓ Suavizado exponencial de Holt.
- ✓ Pick-up aditiva clásica mediante media móvil.
- ✓ Pick-up aditiva avanzada usando promedio móvil.
- ✓ Pick-up aditiva clásica mediante suavizado exponencial.
- ✓ Pick-up multiplicativa clásica usando suavizado exponencial.

El trabajo de (VELASQUEZ, 2010) es quizás el trabajo más significativo encontrado, donde se muestra la aplicación de algoritmos de Machine Learning para la predicción de series temporales. En este caso, se aplica un algoritmo de máquina de vectores de soporte y se muestra un procedimiento para encontrar un valor óptimo para los parámetros del modelo de kernel radial. Los resultados muestran que el algoritmo utilizado presenta resultados favorables debido a que presenta un bajo error en la predicción, con una particularidad; que los próximos pronósticos deberán realizarse en una ventana temporal no superior a 6 periodos.

La investigación realizada por (CABRERA, 2014) utiliza cuatro técnicas de aprendizaje automático para entrenamiento y validación: Ridge Regresión, Kernel Ridge Regresión, Redes Neuronales Artificiales perceptrón multicapa y de Función Base Radial. Separaron los datos en tres conjuntos que fueron: entrenamiento, validación y prueba. Los conjuntos de datos se construyeron utilizando 3 esquemas diferentes, en el primero se desarrolló un modelo de series de tiempo, donde las entradas del modelo se basaron en observaciones de ocupación de los días anteriores, en el segundo esquema los conjuntos de datos se basaron en estas mismas observaciones y también tomaron en cuenta otras variables como día de la semana, festivos y temporada; En el tercer esquema, se tomó como variable de entrada del modelo la información de los días de anticipación de las reservas, incluyendo además los días de la semana, los meses del año y los feriados. Luego de la construcción del data-sets mediante los tres esquemas, utilizan la técnica MAPE (Mean Absolute Percentage Error), el lenguaje de programación que utilizaron para el desarrollo del algoritmo fue Pytho.

El trabajo realizado por (SANCHEZ A. J., 2020) propone un medio para permitir la previsión de cancelaciones de reservas de hotel utilizando solo 13 variables independientes, número reducido en comparación con investigaciones afines en el área, que además coinciden con los que más solicitan los clientes al realizar una reserva. En este asunto, utilizó técnicas de aprendizaje automático, entre otras redes neuronales artificiales optimizadas con algoritmos genéticos fueron aplicados logrando una tasa de cancelación de hasta el 98%. La metodología propuesta nos permite no sólo conocer sobre tasas de cancelación, sino también para identificar qué cliente es probable que cancele. Este enfoque permitiría a las organizaciones reforzar sus protocolos de actuación frente a la llegada de turistas.

A continuación, la **Tabla 2** presenta el resumen de las diferentes aplicaciones de los algoritmos de Machine Learning utilizados en diferentes campos.

TITULO DEL TRABAJO	REFERENCIA BIBLIOGRAFICA	PROBLEMA ABORDADO
Pronóstico de la demanda diaria de hoteles para datos de alta frecuencia y estacionalidad compleja: un estudio de caso en Tailandia.	Phumchusri, n. &. (2019).	Desarrolló el pronóstico de habitaciones vendidas, En esta investigación se aplican varios modelos de pronóstico, tales como (Último día del año anterior, Media Móvil, Suavizado Exponencial, Método de pickup Aditivo y Multiplicativo, Regresión Múltiple).

Uso de aprendizaje automático y big data para la previsión eficiente de cancelaciones de reservas de hoteles.	Sanchez, a. J. (2020).	Propone un medio para permitir la previsión de cancelaciones de reservas de hotel utilizando solo 13 variables independientes, en este asunto, utilizó técnicas de aprendizaje automático, entre otras redes neuronales artificiales optimizadas con algoritmos genéticos fueron aplicado logrando una tasa de cancelación de hasta el 98%.
Predecir la cancelación de reservas de hotel con un modelo de clasificación de aprendizaje automático	(ANTONIO & DE ALMEIDA, 2019)	se desarrolló un prototipo de sistema basado en aprendizaje automático, haciendo uso de los datos de los sistemas de gestión de propiedades del hotel y se entrena un modelo de clasificación todos los días para predecir qué reservas son "probables de cancelar" y con eso calcular la demanda neta. Los algoritmos utilizados fueron: Boosted Decision Tree, Decision Forest, Decision Jungle, Locally Deep Support Vector Machine y Neural Network.
Modelos de predicción de reservas para listados de alojamiento entre pares mediante regresión logística, árbol de decisión, K-vecino más cercano y clasificadores de bosque aleatorio	(Afrianto, 2020)	El objetivo fue desarrollar modelos de predicción para determinar la probabilidad de reserva de los listados de alojamiento. utilizaron un conjunto de datos de Airbnb, desarrollaron cuatro modelos de aprendizaje automático, regresión logística, árbol de decisión, K-vecino más cercano (KNN) y clasificadores de bosque aleatorio. Evaluaron los modelos usando la puntuación AUC-ROC y el tiempo de desarrollo del modelo usando los procedimientos de validación cruzada. En términos de puntaje promedio de AUC-ROC, los Clasificadores de Bosques

		Aleatorios superaron a otros modelos evaluados.
Desarrollo de un modelo basado en Machine Learning para la predicción de la demanda de habitaciones y ocupación en el sector hotelero.	Cabrera, f. (2014).	Facilitar a los administradores hoteleros la toma de decisiones enfocadas a la optimización de los recursos e incrementos en los beneficios del hotel. Utilizaron cuatro técnicas de Machine Learning para el entrenamiento y la validación las cuales fueron: Ridge Regresión, Kernel Ridge Regresión, Redes Neuronales Artificiales perceptrón multicapa y de Función Base Radial.
Un nuevo enfoque para la maximización de los ingresos por habitación de hotel utilizando métodos avanzados de pronóstico y optimización.	Garay, n. &. (2008).	Propuso un modelo para el pronóstico de llegadas y ocupación utilizando la técnica de simulación Monte Carlo, con parámetros estocásticos como reservas, cancelaciones, duración de la estadía, ausencias, estacionalidad, y tendencia
Prediccion series temporales usando maquinas de vectores de soporte.	Velasquez, j. D. (2010).	Aplicó un algoritmo de Machine Learning para la predicción de series temporales. En este caso, se aplica un algoritmo de máquina de soporte vectorial y se muestra un procedimiento para encontrar un valor óptimo para los parámetros del modelo con kernel radial.
La capacidad predictiva en los metodos box-jenkins y holt winters: una aplicacion al sector turistico.	Jimenez, j. &. (2006).	En este trabajo se realiza un análisis de la capacidad predictiva del método en una serie con estacionalidad procedente del sector turístico. Se Aplicaron métodos ARIMA los cuales son frecuentemente usados en el modelamiento de series de tiempo.

Predicción de cancelación de reservas de hotel usando Algoritmo de regresión logística y redes neuronales.	(RENDI, 2021)	Este estudio, utilizando datos recopilados del Sitio web de Kaggle con el nombre hotel-booking demand dataset. El objetivo de la investigación fue ver el rendimiento de la red neuronal y el algoritmo de regresión logística. La precisión de estos algoritmos aumenta cuando se eliminan las variables de la base de datos que no aportan valor.
Pronóstico de tasas de ocupación hotelera con modelo de serie temporal: un análisis empírico.	Lee, c. K. (1990).	Usaron el método ARIMA de Box-Jenkins y el método de suavizado exponencial Holt Winter, para pronosticar la ocupación mensual del hotel, ellos muestran que ambos métodos producen un bajo error cuadrático medio.
Aprendizaje Automático aplicado al sector hotelero	(Embarec, 2020)	El objetivo de este proyecto fue, aprovechando las herramientas del Aprendizaje Automático, estudiar varios problemas comunes de la industria, como predecir el precio diario de alojamiento dadas unas variables que influyen como la antelación, el número de noches contratadas, la probabilidad de cancelación de una reserva, utilizaron los siguientes algoritmos: **random forest**, regresión lineal, k-vecinos cercanos, regresión polinomial, regulación de tijonov, lasso.

Tabla 2. Aplicaciones de los métodos de Machine Learning. Fuente: Fuente: elabocion propia.

Al examinar los trabajos revisados, se puede evidenciar una ruta para la aplicación de modelos de machine learning para la estimación de reservas canceladas. Esta, consiste en realizar comparaciones tanto de diferentes métodos tradicionales como de diferentes modelos de machine learning estableciendo un indicador AUC.

Los algoritmos de aprendizaje automático más utilizados son: las redes neuronales, Naïve Bayes (NB), Máquina de vectores de soporte (SVM), clasificador de máxima

entropía (ME), análisis de regresión (RA), árboles de decisión, K-vecinos más cercanos (KNN). Así también muestra de qué forma se pueden tratar los datos para los modelos de machine learning, división en tres (3) conjuntos de datos: entrenamiento, Prueba y validación, es la que se ve con mayor frecuencia en estos trabajos. La mayoría de los trabajos se orientan a problemas de regresión, dejando una brecha para comenzar a trabajar problemas de clasificación.

4. MARCO TEÓRICO

4.1. INTELIGENCIA ARTIFICIAL

Dentro de la tecnología, se encuentra la inteligencia artificial, la cual es la rapidez y destreza de las computadoras para poder realizar las actividades que usualmente realizaban las personas en un trabajo (Rouhiainenl, 2018). La inteligencia artificial (IA) Se define como la capacidad de un sistema para interpretar correctamente datos externos, aprender de esos datos y utilizar ese aprendizaje para lograr objetivos específicos a través de una adaptación flexible.. (Kaplan, 2018)

La inteligencia artificial es una tecnología que proporciona la capacidad para que una máquina realice funciones cognitivas, como percibir, razonar, aprender e interactuar con personas. Rápidamente entró en nuestras vidas resolviendo problemas gracias a tres desarrollos tecnológicos que alcanzaron la suficiente madurez y convergencia: el avance de los algoritmos, la masificación de datos y el aumento de la potencia de cómputo y el almacenamiento a bajo costo. (Mustafa, 2019)

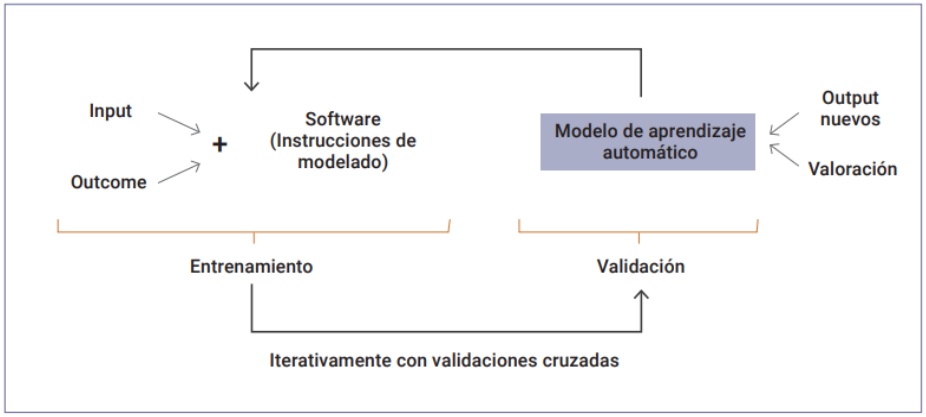

Ilustración 1. Esquema general del proceso de aprendizaje automático. (Adaptado. Brain Commun. 2020)

4.2. MACHINE LEARNING

El machine learning (ML) se define como el conjunto de herramientas o técnicas computacionales que permiten aprender en una computadora a resolver un problema específico a través de la experiencia sin necesidad de ser programado explícitamente. Es un método de análisis de datos que automatiza la construcción de modelos analíticos. Esta es una rama de la IA basada en la idea de que los sistemas pueden aprender de los datos, identificar patrones y tomar decisiones con una mínima intervención humana.. A continuación, se hablará sobre distintos algoritmos de machine learning, los cuales ayudarán a que se obtengan previsiones de manera más exacta. (SANCHEZ & MIR, 2021)

Este conjunto de herramientas de aprendizaje permite el ajuste de un modelo usando unas instrucciones matemáticas y datos que pueden ser estructurados y no estructurados, que contemplen información de calidad para la resolución del problema. El conjunto de instrucciones de modelado matemático se denomina algoritmo. El conjunto de datos que alimenta el algoritmo se organizará en una serie de variables de entrada y tendrá como objetivo encajar en una o varias variables de salida. En base en estas variables de salida, se define lo que puede ser un diagnóstico u otros resultados relevantes. El proceso de aprendizaje en el que se ajustan los parámetros matemáticos del modelo se denomina entrenamiento y la evaluación del modelo ajustado se denomina validación. (Roberto López Blanco, 2021)

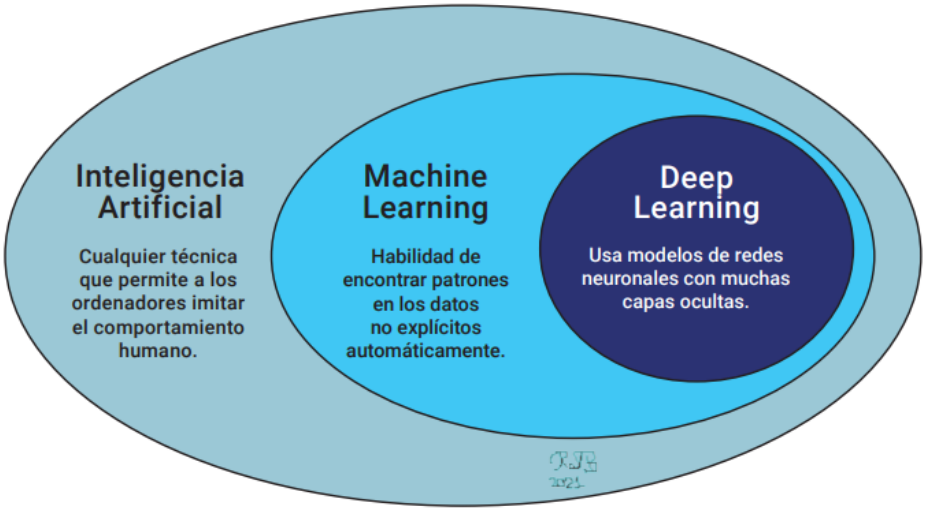

Ilustración 2. Esquema de las características principales de la Inteligencia Artificial, Machine Learning y Deep Learning. (Adaptado Pedersen M et al. Brain Commun. 2020).

4.3. MODELOS DE CLASIFICACIÓN

Cuando usamos el aprendizaje automático, podemos realizar tareas de clasificación o regresión. La diferencia radica en el tipo de resultados obtenidos con los métodos de aprendizaje automático utilizados. La Clasificación Supervisada es una de las tareas más comunes realizadas por los llamados Sistemas Inteligentes. Por lo tanto, es posible generar una gran cantidad de modelos con la ayuda de estadísticas (regresión logística, análisis discriminante) o inteligencia artificial (redes neuronales, árboles de decisión, redes bayesianas y otros modelos) realizar tareas que coincidan con la clasificación.

Para entrenar modelo de Machine Learning es importante dividir el conjunto de datos en dos conjuntos de datos más pequeños que serán utilizadas con los siguientes fines: entrenamiento y test. El subconjunto de datos de entrenamiento es empleado para estimar los parámetros del modelo y el subconjunto de datos de test se utiliza para evidenciar el comportamiento del modelo estimado. Cada registro de la base de datos debe aparecer en uno de los dos subconjuntos, y para dividir el conjunto de datos en ambos subconjuntos, se utiliza un procedimiento de muestreo: muestreo aleatorio simple. Lo ideal es entrenar el modelo con un conjunto de datos independiente de los datos con los que se realiza el test. (PARRA, FRANCISCO, 2019)

Como resultado de aplicar un método de clasificación, se cometerán dos errores, en el caso de una variable binaria que toma valores 0 y 1, habrá ceros que se clasifiquen incorrectamente como unos y unos que se clasifiquen incorrectamente como ceros. (Parra, 2019)

4.4. MATRIZ DE CONFUSIÓN

La matriz de confusión es una de las métricas más relevantes en el aprendizaje automático, ya que describe el rendimiento de cualquier modelo implementado en todos los datos de prueba, donde se desconocen las clases a las que pertenecen, recibe el nombre de matriz de confusión porque se refiere a la facilidad de detección donde el sistema está confundiendo las clases evaluada. (Arias, 2020).

En la siguiente ilustración (6) se muestra el esquema de la matriz de confusión:

		Predicción	
		Positivos	**Negativos**
Verdaderos	**Positivos**	Verdadero Positivos	Falsos Negativos
	Negativos	Falsos Positivos	Verdaderos Negativos

Ilustración 3. Matriz de confusión. Fuente: *(Arias, 2020)**.***

Al realizar una predicción en esta matriz de confusión, se pueden observar 4 resultados diferentes: Verdaderos positivos (VP) son los casos que son correctamente predichos por el modelo, es decir, son positivos y efectivamente clasificados como positivos; los falsos positivos (FP) son casos en los que el modelo clasificó como positivo, pero en realidad son negativos; falsos negativos (FN) es el número de muestras a las que el modelo dio la etiqueta negativa y en realidad son positivas y verdaderos negativos (TN) son los casos que el modelo predice como negativos y en realidad lo son.. (Arias, 2020)

Con estos 4 resultados se pueden calcular diferentes métricas, de las cuales hay 2 fundamentales para valorar el rendimiento del modelo, que son la especificidad y la sensibilidad, aunque también se considera la métrica correspondiente a la puntuación F1. (Arias, 2020)

4.4.1 MÉTRICA DE SENSIBILIDAD

Recall o sensibilidad es una medida que permite conocer la proporción de casos positivos que fueron correctamente clasificados. En un modelo perfecto el recall es igual a 1 para cada clase. Desde el punto de vista analítico un investigador busca aumentar el recall sin afectar el valor de la accuracy. *(DRZEWIECKI, 2017)*

$$\boldsymbol{Sensibilidad = \frac{VP}{VP + FN}}$$

Ecuación 1. Métrica de sensibilidad. Fuente *(DRZEWIECKI, 2017)*

4.4.2 MÉTRICA DE ESPECIFIDAD

Corresponde a la proporción del total de casos negativos que son considerados positivos por el modelo, es decir cuánto es el error del modelo al predecir casos negativos (MISHRA, 2018).

$$Especifidad = \frac{VN}{VN+FP}$$

Ecuación 2. Métrica de especifidad. Fuente: (MISHRA, 2018)

4.4.3 MÉTRICA DE EXHAUSTIVIDAD

Es la métrica que brinda información sobre el desempeño del modelo en relación con la cantidad de muestras clasificadas como falsos negativos, es decir, cuántas predicciones fallaron, esto también se conoce como recordación del clasificador, significa que, si desea minimizar los falsos negativos, el porcentaje de recuperación del modelo debe estar cerca del 100 %(Sitiobigdata.com, 2019).

$$accuracy = \frac{TP+TN}{(TP+TN+FP+FN)}$$

Ecuación 3. Métrica de exhaustividad. Fuente: *(Barrio, 2019)**.***

4.4.4 MÉTRICA PUNTUACION F1-SCORE

Es la medida que permite comparar el rendimiento combinado entre la precisión del modelo y su memoria (Recall), y la relación se basa en que a mayor porcentaje de F1, el modelo es mucho mejor. (BORJA, 2020)

La medida F1-score mezcla las métricas de precisión y recuperación, presentando diferencias en el desempeño de un clasificador que no son reveladas únicamente por la precisión. Es directamente proporcional al aumento de las dos medidas, por lo tanto, loa valores altos de F1-score muestran que el algoritmo de clasificación predice mejor la clase positiva. (BEKKAR, 2017)

$$F1 = 2 * (\frac{precision*recall}{precision+recall})$$

Ecuación 4. Métrica de F1-score. Fuente: *(BEKKAR, 2017)*

4.5 MÉTRICAS PARA EVALUAR MODELOS CLASIFICACIÓN, CURVA (ROC)

Un método para evaluar la capacidad de clasificar de un método es la curva ROC (Receiver Operating Characteristic). La curva ROC es una métrica que se usa con frecuencia para evaluar el desempeño en clasificadores dicotómicos, la curva ROC es un gráfico bidimensional de sensibilidad versus (1 – especificidad) para cada clase. La medida de comparación en este gráfico es el AUC que corresponde al área bajo la curva ROC y sus valores están entre 0 y 1, considerando que en el caso totalmente aleatorio se tiene un AUC igual a 0.5 (Fawcett, 2016)

De la curva ROC Se pueden hacer 3 tipos de interpretaciones o análisis, la primera es si el AUC (Area Under Curve) es 1 (Ilustración 3), significa que el modelo distingue perfectamente todas las muestras o características de cada una de las clases. (clase 1 y clase 2) al asignarles la clase correcta, esto denota que el modelo está sobreentrenado y solo es útil para los datos que se usaron para su entrenamiento, haciendo que su desempeño no sea óptimo para nuevos datos, un valor entre 0.8 y 0.9 para una curva ROC es generalmente un excelente resultado, si el AUC es 0.5 (ilustración 4), el modelo no puede diferenciar entre las dos clases y finalmente si el AUC=0 (ilustración 5), representa que el modelo confunde entre las dos clases, dando clase 1 clase 2 y viceversa (Arias, 2020)

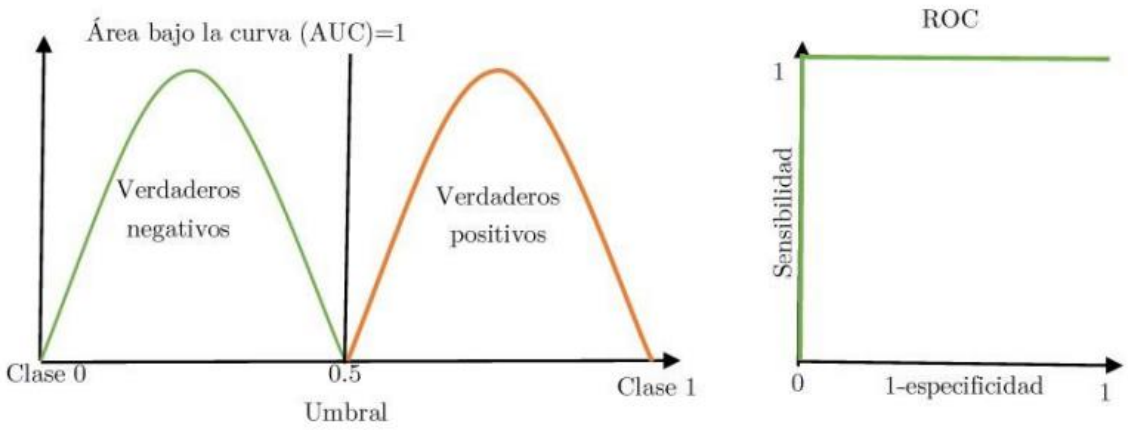

Ilustración 4. Curva ROC, con AUC=1. Fuente: Arias, Erika 2020.

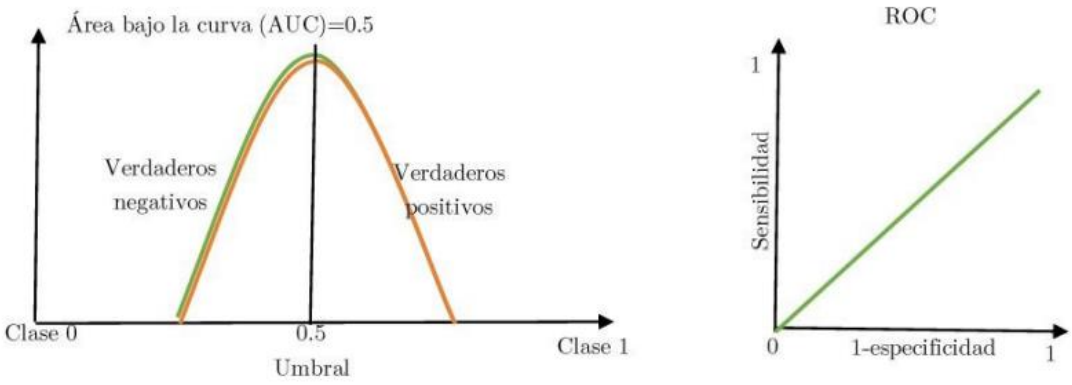

Ilustración 5. Curva ROC, con AUC=0.5. Fuente: Arias, Erika 2020.

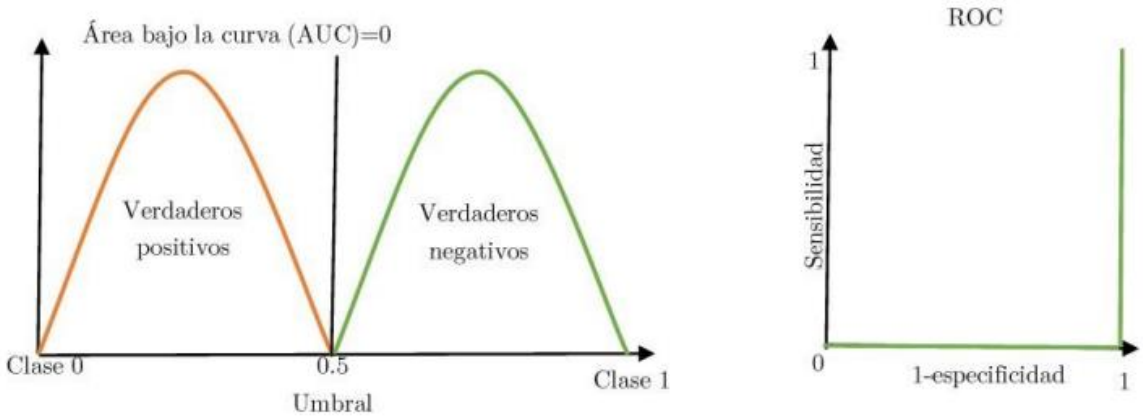

Ilustración 6. Curva ROC, con AUC=0. Fuente: Arias, Erika 2020.

4.6 MODELOS DE CLASIFICACIÓN MACHINE LEARNING

4.6.1 CLASIFICADOR BAYESIANO

El clasificador bayesiano es uno de los métodos más utilizados en el aprendizaje automático, se basa en el teorema de probabilidad de Bayes, el cual tiene como principio Las redes bayesianas modelan un fenómeno mediante un conjunto de variables y las relaciones de dependencia entre ellas. Dado este modelo, se puede hacer inferencia bayesiana; es decir, estimar la probabilidad posterior de las variables no conocidas, con base en las variables conocidas.

el modelo bayesiano ingenuo es el clasificador bayesiano clásico. Este está construido bajo el supuesto de que todas las variables predictoras son

condicionalmente independientes dado el valor de la variable de clase. El modelo naive de Bayes tiene una estructura fija que no depende de datos, por lo que no podemos decidir que él aprendizaje estructural de este modelo criterios generativos o discriminativos. Sin embargo, el aprendizaje se suele utilizar para obtener los parámetros del modelo ingenuo de Bayes, así como la mayoría de los clasificadores bayesianos en general, se basa en las estimaciones máximas probables (ML) o las máximas a posteriori (MAP) (Enrique, 2003).

El algoritmo de clasificación Naïve-Bayes (NBC) es un clasificador probabilístico simple con un fuerte supuesto de independencia. Sin embargo, la suposición de independencia de atributos suele ser una mala suposición y viola el menú de conjuntos de datos verdaderos. Por lo general, proporciona una mejor precisión de clasificación en conjuntos de datos en tiempo real que cualquier otro clasificador. También requiere una pequeña cantidad de datos de entrenamiento. El clasificador Naïve-Bayes aprende de los datos de entrenamiento y rendimiento y predice la clase de instancia de prueba con la mayor probabilidad posterior.(Rodriguez, 2014)

Esto en términos matemáticos se puede explicar de la siguiente manera, mostrando que el enfoque bayesiano en una clasificación supervisada se basa en la atribución a un conjunto de atributos o características., $X_1, X_2, \dots, X_n$ una de las m clases posibles, $C_1, C_2, \dots, C_m$, con el objetivo que la probabilidad calculada para cada clase existente, a través de los atributos sea la máxima posible (Ec.5) (Arias, 2020).

$$\boldsymbol{Argc}[\boldsymbol{MaxP}(\boldsymbol{C}|\boldsymbol{X_1}, \boldsymbol{X_2}, \dots, \boldsymbol{X_n})] \qquad \textit{Ecuación 5.}$$

Si los atributos del problema se representan como $X = [X_1, X_2, \dots, X_n]$, la (Ec.6 Y 7) Se puede reducir a: $\boldsymbol{Argc}[\boldsymbol{MaxP}(\boldsymbol{C}|\boldsymbol{X})]$. La base matemática de los clasificadores bayesianos utiliza la regla de Bayes para encontrar la probabilidad posterior de cada una de las clases trabajadas en función de los atributos (Arias, 2020).

$$\boldsymbol{P(C|X_1, X_2, \dots, X_n)} = \frac{\boldsymbol{P(C) * P(X_1, X_2, \dots, X_n|C)}}{\boldsymbol{P(X_1, X_2, \dots, X_n)}} \qquad \textit{Ecuación 6.}$$

$$\boldsymbol{P(C\ |X)} = \frac{\boldsymbol{P(C) * P(X\ |C)}}{\boldsymbol{P(X)}} \qquad \textit{Ecuación 7.}$$

4.6.2 MÁQUINAS DE SOPORTE VECTORIAL (SVM)

SVM ha mostrado ser uno de los mejores clasificadores para un amplio abanico de situaciones, por lo que es considerado uno de los referentes en el área de aprendizaje estadístico y aprendizaje automático. (RODRIGO J. A., 2017)

Este método está directamente relacionado con problemas de clasificación y regresión, con SVM se construye un modelo de predicción, ya que este método representa los puntos muestrales en el espacio, separados por un hiperplano, que se define como el vector entre los dos puntos, de las dos clases más cercanas a lo que se conoce como vector de soporte. (Soporte M, 2019)

El modelo de Support Vector Machine (SVM). Mmaquina de aprendizaje, las máquinas de vectores de soporte son modelos de aprendizaje supervisado con algoritmos de aprendizaje asociados que analizan los datos utilizados para la clasificación y el análisis de regresión. Las maquinas de soporte vectorial pueden ejecutar de manera eficiente una clasificación no lineal empleando lo que se denomina el truco del kernel, fundamentalmente, dibuja márgenes entre las clases. Los márgenes se dibujan de tal manera que la distancia entre el margen y las clases sea máxima y, por lo tanto, minimice el error de clasificación. (Batta, 2018)

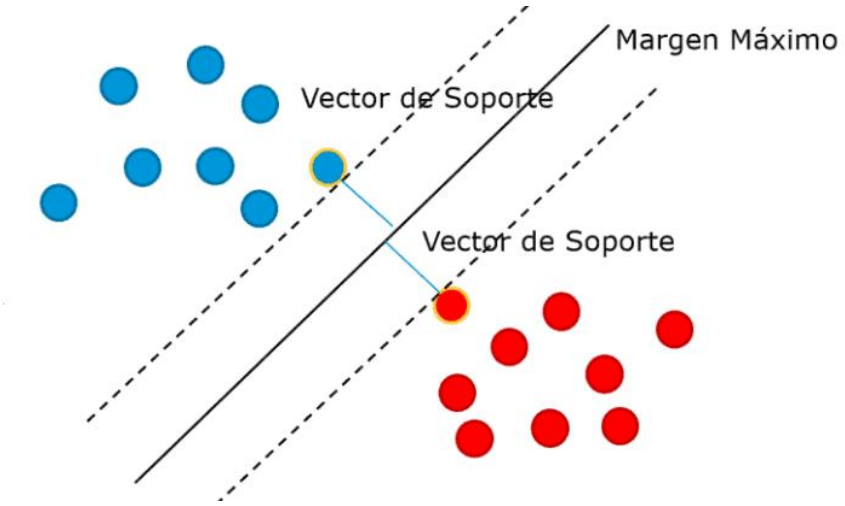

Ilustración 7. Maquina Vectores de soporte. Fuente: iartificial.net.

4.6.3 ARBOLES DE DECISION

Los árboles de decisión son técnicas de aprendizaje automático que permiten construir modelos predictivos de análisis de datos en base a su clasificación según determinadas características o propiedades, o en regresión a través de la relación entre distintas variables para predecir el valor de otra.

Los árboles de decisión a menudo se usan en para predecir la probabilidad de lograr un determinado resultado en función de ciertas condiciones (incertidumbre). Ejemplos típicos del uso de este tipo de algoritmo son:

- ✓ Estimación de las primas de seguros a cobrar a los asegurados.
- ✓ Predecir si se debe ofrecer un determinado producto a una persona

En términos matemáticos, los árboles de clasificación se pueden explicar de la siguiente manera, si C es una variable de respuesta y A es un conjunto de variables predictoras $X_1, X_2, \dots, X_A$, en la cual el conjunto A son etiquetas fijas y la variable Y es aleatoria el problema estadístico a resolver se basa en encontrar y establecer una relación entre C y A, de tal forma que sea posible predecir el valor de C a través de los valores que tiene A, encontrando una función que estudie la probabilidad condicional que existe de la variable aleatoria C (Ec.8) (Sitiobigdata.com, 2019)

$$\boldsymbol{A[C = c\ |X_1, X_2, \dots, X_A]} \qquad \textit{Ecuación 8.}$$

4.6.3.1 ELEMENTOS DEL ARBOL

Cualquier árbol de decisión se divide básicamente en 3 niveles, el primero corresponde al nodo raíz que es la primera división que hace el modelo y es la parte superior del árbol, el segundo nivel corresponde a uno o más nodos internos y está asociado a uno de los atributos y de él salen 2 o más ramas, cada una de ellas representa los diferentes valores que puede tomar el atributo correspondiente, finalmente está el tercer y último nivel que pertenece a los nodos terminales, el cual da la clasificación adecuada y devuelve la decisión del árbol respecto a los datos de entrada.(Arias, 2020).

4.6.3.2 HOMOGENEIDAD

Está asociado a nodos terminales e idealmente las variables resultantes de estos nodos tienen la mayor homogeneidad que se mide por la noción de impureza (Arias, 2020).

4.6.3.3 IMPUREZA DEL NODO

Se define diciendo que C es una variable dicotómica que solo asume los valores de 0 y 1, un nodo tiene mayor impureza cuando su impureza es máxima con$P(C = correcto) = \frac{1}{2}$. La función de impurezas para los nodos toma una forma cóncava y se define como: (Arias, 2020).

$$\boldsymbol{i(\tau) = \varphi(P(C = correcto))}$$ ***Ecuación 9. Impureza del nodo. Fuente: arias,2020***

Entonces $\boldsymbol{\tau}$ se refiere a la impureza y $\boldsymbol{\varphi}$ tiene propiedades como:

$\varphi \geq 0\ y\ (no\ negativa)$Entonces

$p \in (0,1)$

$\varphi(p) = \varphi(1-p)(simetrica)$

$\varphi(0) = \varphi(1) < \varphi(p)$

$mínima\ para\ el\ éxito\ o\ el\ fracaso\ absoluto$

Las opciones más comunes para la función de impurezas en la construcción de un árbol de clasificación son:

- ✓ $\varphi(p) = min(p, 1-p)$: $Mínimo\ error\ o\ error\ de\ bayes$
- ✓ $\varphi(p) = (-p * log(p) -)1-p) * \log(1-p)$: $Entropía$
- ✓ $\varphi(p) = p(1-p)$: $índice\ de\ gini$

Donde se define $0\log(0) = 0$

4.6.3.4 DIVISIÓN DE UN NODO

El nodo raíz se divide en dos nodos homogéneos, estos nodos se seleccionan encontrando el valor entre el rango de variables predictoras que más se aproxima al límite de pureza para cada uno de los nodos internos, el objetivo es que, si el nodo N se divide en dos, Nl y NR, la pureza de estos dos nodos debe ser mayor que la pureza del nodo N, o incluso tener una impureza menor, que suele medirse con la probabilidad más baja, el índice de Gini o entropía (Arias, 2020)

La estructura general de un árbol de decisión se muestra en la (ilustración 8), en la que se puede observar que las variables predictoras están representadas por círculos, y los días del árbol donde los datos pertenecen a una sola clase están representados por un rectángulo. (Arias, 2020)

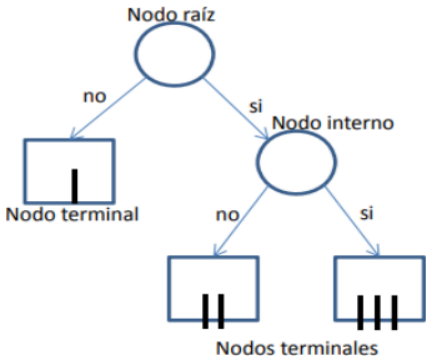

Ilustración 8. Estructura general de un árbol de decisión. Fuente: Arias Erika 2020.

4.6.4 VECINOS MÁS CERCANOS (KNN O K-NN)

El algoritmo de k vecinos más cercanos, también conocido como KNN o k-NN, es un clasificador de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. se puede usar para problemas de regresión o clasificación, generalmente se usa como un algoritmo de clasificación, partiendo de la suposición de que se pueden encontrar puntos similares cerca uno del otro. (GONZALEZ, 2019)

Simplemente busca en las observaciones más cercanas a la que se está tratando de predecir y clasifica el punto de interés basado en la mayoría de los datos que le rodean (Bagnato, 2018).

SUPERVISADO: *quiere decir que tenemos etiquetado nuestro conjunto de datos de entrenamiento, con la clase o resultado esperado dada "una fila" de datos. (GONZALEZ, 2019)*

BASADO EN INSTANCIA: *quiere decir que nuestro algoritmo no aprende explícitamente un modelo (como por ejemplo en Regresión Logística o árboles de decisión). En cambio, memoriza las instancias de entrenamiento que son usadas como "base de conocimiento" para la fase de predicción. (GONZALEZ, 2019)*

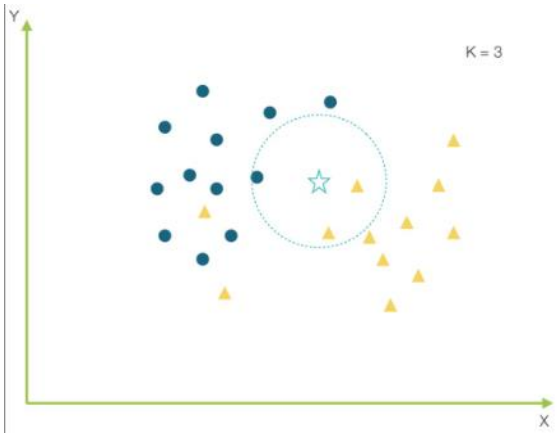

Ilustración 9. Método K vecinos más cercanos. Fuente: aprendeia.com – 2018.

4.6.4.1 DISTANCIA EUCLIDIANA

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Ecuación 10.Distancia euclidiana. Fuente: *(GONZALEZ, 2019)*

4.6.4.2 DISTANCIA MANHATTAN

$$\sum_{i=1}^{k}|x_i - y_i|$$

Ecuación 11. Distancia manhattan. Fuente: *(GONZALEZ, 2019)*

4.6.5 RANDOM FOREST

Es una técnica mejorada de bagging, que ayuda a lograr una mayor precisión en la clasificación al incorporar la aleatoriedad en la construcción de cada clasificador individual. Esta aleatorización se puede introducir en la partición del espacio (construcción de árboles) así como en la muestra de entrenamiento. El algoritmo Random Forest, a diferencia del bagging, introduce aleatoriamente en cada nodo una lata de p variables entre todas las originales, y de estas selecciona las mejores para realizar la partición. (Cardenas, 2019)

Se presenta a continuación el proceso del algoritmo:

1. Seleccione individuos al azar (usando una muestra con reemplazo) para crear diferentes conjuntos de datos.
2. Para crear los árboles si elige variables aleatorias en cada nodo del árbol, dejando que el árbol crezca (sin podar).
3. Crear un árbol de decisión con cada conjunto de datos, obteniendo diferentes árboles, ya que cada conjunto contiene diferentes individuos y diferentes variables.
4. Prediga los nuevos datos utilizando el "voto mayoritario", que se clasificará como "positivo" si la mayoría de los árboles predicen la observación como positiva. (Cardenas, 2019)

Asimismo, el algoritmo Random Forest es uno de los algoritmos de clasificación más utilizados, ya que una de sus mejores virtudes es proporcionar una estimación de precisión interna a través de una forma de validación cruzada, proporcionando conocimiento sobre el moderno procedimiento de trabajo en paralelo que es precisamente la distribución en Núcleos informáticos para grandes volúmenes de datos (Cardenas, 2019).

4.6.5.1 ESTIMACIÓN DEL ERROR CON RANDOM FOREST

La tasa de error fuera de muestra (OOBi) de una observación se define como el error obtenido cuando se clasifica por los árboles en el bosque edificado sin su intervención, es decir, dejando fuera la muestra no modelada. La estimación del error OOB es el promedio de todos los OOBi para todas las observaciones en el conjunto de datos y es un mejor estimador que el error aparente. Similar a la estimación de validación cruzada, la medida se puede extrapolar al problema de regresión describiéndolo en términos del error cuadrático medio (MSE). (RODRIGO J. , 2017)

4.7 BASES DE DATOS DESBALANCEDAS

El objetivo de un algoritmo de clasificación es intentar aprender un separador o clasificador, que pueda distinguir las dos clases del target. Hay muchas maneras de hacerlo, basadas en varias suposiciones matemáticas o estadísticas. Pero cuando comienzas a trabajar con datos reales, una de las primeras observaciones que resalta la desigualdad en la proporción de las clases, como se muestra en la siguiente ilustración: (Fawcett, 2016)

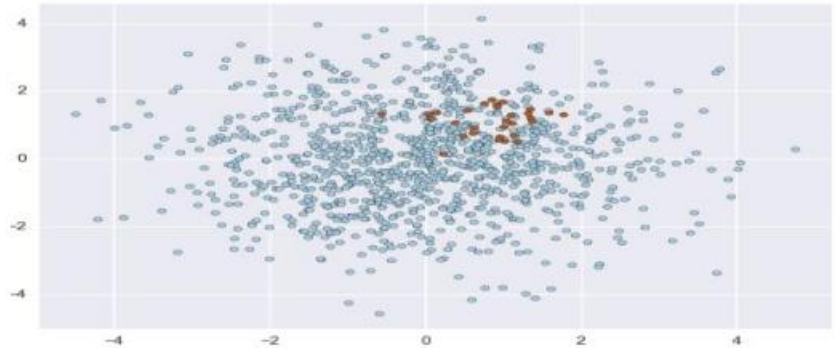

Ilustración 10. Clases desproporcionada del target. Fuente: Fawcett, 2016.

El principal problema es que estas clases están desequilibradas: los puntos azules son mucho más numerosos que los rojos.

Los algoritmos convencionales o tradicionales suelen estar sesgados hacia la clase mayoritaria porque sus funciones de pérdida intentan optimizar cantidades como la tasa de error, sin tener en cuenta la distribución de datos. En el peor de los casos, los ejemplos minoritarios se tratan como valores atípicos de la clase mayoritaria y se ignoran.El algoritmo de aprendizaje simplemente genera un clasificador trivial que clasifica cada ejemplo como la clase mayoritaria. La solución según (Fawcett, 2016), para este problema es:

1. Equilibre el conjunto de entrenamiento de alguna manera:

- ✓ Sobremuestreo de la clase de la minoría.
- ✓ submuestreo de la clase mayoritaria.
- ✓ Sintetizar nuevas clases de minorías.

2. Reproducir los ejemplos minoritarios y cambiar a un marco de detección de anomalías.

3. A nivel del algoritmo, entonces:

- ✓ Ajuste el peso de la clase (costos de clasificación errónea).
- ✓ Ajusta el umbral de decisión.
- ✓ Modifique un algoritmo existente para que sea más sensible a las clases raras.

2. Cree un algoritmo completamente nuevo para obtener buenos resultados en datos desequilibrados.

Las técnicas más comunes utilizadas en este tipo de problemas son: oversampling y undersampling. Los enfoques más fáciles requieren pocos cambios en los pasos de procesamiento y simplemente implican ajustar conjuntos de muestras hasta que estén equilibrados. (Fawcett, 2016).

4.7.1 OVERSAMPLING

La técnica replica aleatoriamente instancias minoritarias (clase menor en proporción del target) para aumentar su población y así equilibrar la clase mayoritaria. Ver ilustración. (Fawcett, 2016)

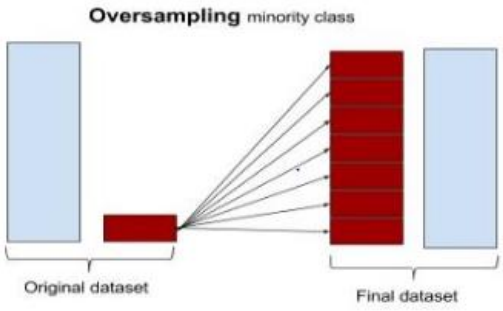

Ilustración 11. Funcionamiento del oversampling. Fuente: Fawcett, 2016.

4.7.2 UNDERSAMPLING

En comparación con la técnica anterior, este proceso hace lo contrario y elige aleatoriamente reducir la clase mayoritaria para llenar la clase minoritaria, y luego equilibra las muestras para entrenar el modelo de aprendizaje automático. (Fawcett, 2016).

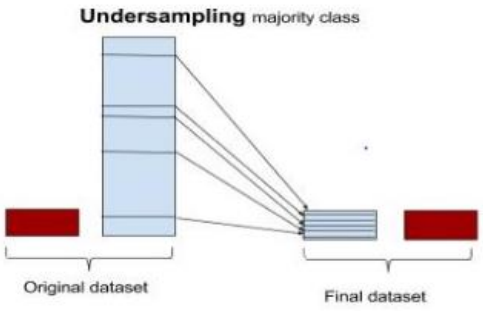

Ilustración 12. Funcionamiento del Undersampling. Fuente: Fawcett, 2016.

4.8 MÉTODOS PARA EL BALANCEO DE LA BASE DE DATOS

4.8.1 MÉTODO SMOTE

SMOTE (Synthetic Minorities About Showcase Technique) consiste en la síntesis de elementos para la clase minoritaria, a partir de lo ya existente. Funciona eligiendo al azar un punto de clase minoritaria y calculando los K- vecinos más cercanos ese punto. Se añaden puntos sintéticos entre el punto elegido y sus vecinos. (Cardenas, 2019)

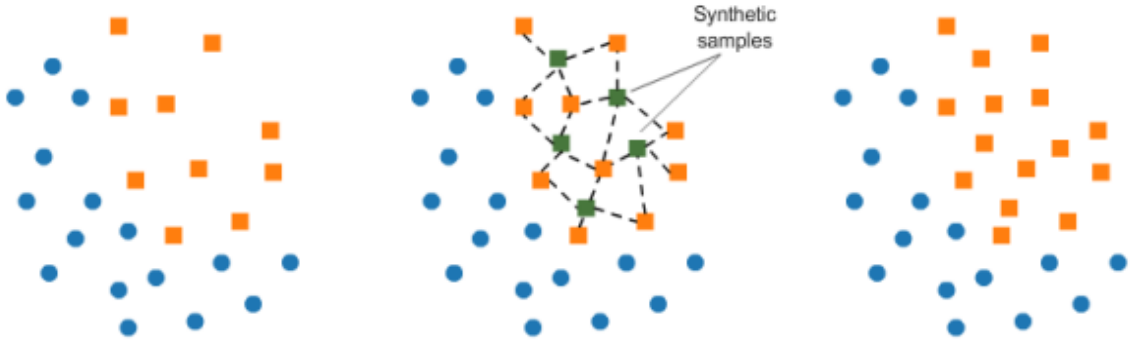

Ilustración 13. Funcionamiento del SMOTE. Fuente: Fawcett, 2016.

Es un método que ha tenido éxito en varias aplicaciones que involucran bases de datos no balanceadas. (Chawla, Bowyer, & Kegelmeyer, 2002). El algoritmo SMOTE crea datos artificiales entre muestras de la clase minoritaria. Específicamente, para el subconjunto $x_- \in x_1$, considere los k-vecinos más cercanos de cada muestra $x_{11} \in x_-$, para algún entero k; los k-vecinos más cercanos son definidos como las k

muestras de $x\_$ cuya distancia euclídea entre ellos y la muestra xi bajo consideración presenta las menores magnitudes. Para crear una muestra sintética, se selecciona aleatoriamente uno de los k-vecinos, luego se multiplica el correspondiente vector de diferencia por un número aleatorio entre el rango [0, 1], y finalmente, se suma el anterior resultado al vector x_i

$$\boldsymbol{X_{new} = X_i + (\hat{x}_i - x_i) * \delta}$$ ***Ecuación 12. Formula del método smote. Fuente: Chawla, 2002***

donde $\hat{x}_i$ es uno de los k-vecinos más cercanos de x_i y $\delta \in [0,1]$ es un número aleatorio. Aunque mostró resultados positivos, el algoritmo SMOTE también tiene desventajas, incluida la generalización y la varianza. (Chawla, Bowyer, & Kegelmeyer, 2002).

4.8.2 MÉTODO ADASYN

Es una versión mejorada de SMOTE. Lo que hace es lo mismo que SMOTE solo con una pequeña mejora. Después de crear esa muestra, agrega pequeños valores aleatorios a los puntos, lo que la hace más realista. En otras palabras, en lugar de que toda la muestra se correlacione linealmente con el origen, tienen un poco más de variación, es decir, están un poco dispersas. (Bhattacharyya, 2018)

Encuentra los n vecinos más cercanos en la clase minoritaria para cada una de las muestras en la clase. Luego dibuja una línea entre los vecinos y genera puntos aleatorios en las líneas. (MONROY, 2016)

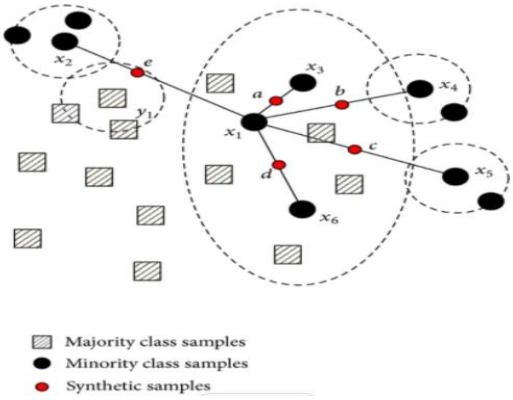

Ilustración 14. Método ADASYN. Fuente: Coinmonks.com.

5 FASES METODOLOGICAS

Para iniciar esta investigación se realizó un estudio de cada una de las técnicas de aprendizaje automático utilizadas en los modelos de clasificación, utilizando como fuente de información bases de datos científicas y el conocimiento de profesionales que trabajaron con estas técnicas, lo que nos permitió determinar las técnicas para utilizar en el transcurso de esta investigación.

5.1 ELECCIÓN DE MODELOS DE CLASIFICACIÓN MACHINE LEARNING

En base a la revisión bibliográfica realizada, se encontró que efectivamente existen aplicaciones para la predicción utilizando modelos de clasificación de aprendizaje automático. Un hallazgo importante es el uso del modelo Vector Support Machines, arboles de decisión, clasificador bayesiano, vecinos más cercanos y Random Forest. Estos modelos, generan diferentes resultados con la capacidad de hacer predicciones. Para este estudio se propone analizar el comportamiento de estos 5 modelos. En la literatura se encontró que los modelos más utilizados para la estimación de las cancelaciones fueron: k- vecinos cercanos, arboles de decision, Random Forest, XG Boost y redes neuronales.

5.2 SELECCIÓN DE LA MÉTRICA DE DESEMPEÑO

En total se tiene una base de datos con 17807 registro de reservas canceladas y no canceladas, esta base de datos fue utilizada para entrenar los modelos de machine learning. El 70% (Total números) de estos registros se tomaron para entrenar los modelos y el 30% restante para validar el ajuste de estos. Además, para evaluar el ajuste de los modelos se tomó los indicadores curva ROC (AUC), sensibilidad, Recall y el F1 Score, conjuntamente este último indicador (F1) se utilizó como medida de desempeño para comparar los modelos con mejor resultado.

Para el entrenamiento de los modelos se utilizó el entorno y lenguaje de programación R el cual es de libre distribución facilitando su uso, además, R tiene muchas funciones para el análisis estadístico y creación de gráficas y lo más importante contiene muchas librerías creadas por la comunidad enfocadas al aprendizaje automático, estas librerias creadas por la comunidad nos facilitan el trabajo a la hora de estudiar los modelos y gracias a esto podemos centrarnos en analizar los resultados y tratar de mejorar los modelos. Al entrenar los modelos

dividiremos la base de datos en dos conjuntos de datos, uno para entrenar el modelo (train) y otro para evaluarlo (test). Para evitar que esta partición influya en la evaluación final de los modelos, utilizaremos la validación cruzada para garantizar que los resultados sean independientes de la partición.

Una vez que se ha construido y entrenado el modelo elegido, se debe verificar la confiabilidad y precisión del modelo. Para ello se utilizaron datos de prueba que previamente habíamos dividido. Con esta selección de caso de prueba se comprueba si los modelos son capaces de predecir correctamente la clase.

Para comparar las predicciones de los modelos se emplea la matriz de confusión. La matriz de confusión genera una matriz que permite la visualización y la precisión de los modelos. La columna de la matriz representa el número de elementos reales por la clase y cada fila el número de elementos predichos.

	Cancelaciones	**Check_In**
Cancelaciones	**VP**	**FP**
Check_In	**FN**	**VN**

Tabla 3. Matriz de confusión adaptada a nuestro problema. Fuente: Elaboración propia, (basado en el diagrama elaborado por aprende IA).

A partir de la matriz de confusión se puede extraer información variada con las que podremos calcular la precisión, el Recall, F1 score entre otras métricas.

Los elementos de la primera (1.1) y última (2.2) casilla corresponden al número de elementos correctamente clasificados. Los modelos predicen correctamente la cantidad de elementos positivos y la cantidad de elementos negativos que en realidad eran positivos y negativos. Los elementos clasificados en la clase positiva (CANCELACIONES) y que realmente pertenecieran a la clase positiva serían considerados verdaderos positivos, y los elementos clasificados en la clase negativa (CHECK IN) y que realmente pertenecieran a la clase negativa serían clasificados como verdaderos negativos.

Los elementos de la segunda (2,1) y tercera (1,2) casillas corresponden al número de elementos clasificados incorrectamente. Los modelos predijeron erróneamente el número de elementos positivos y los clasificaron como negativos

(CANCELACIONES como CHECK IN) y el número de elementos negativos clasificados como positivos (CHECK IN como CANCELACIONES).

El hecho de que tengamos una mayor precisión no indica que el algoritmo sea mejor y que solo sea útil cuando tenemos el mismo número de observaciones para cada clase, y cuando las predicciones que hacemos sobre las clases tienen la misma importancia. Estos datos de precisión no son una métrica adecuada cuando los datos están desequilibrados y, por lo tanto, debemos configurarlos para otras métricas como F1, que nos permite combinar las medidas de precisión y recall, esto es útil porque es más fácil de compare la precisión combinada y el rendimiento entre diferentes soluciones.

5.3 DESCRIPCION DE LA BASE DE DATOS

Para realizar una comparación de modelos efectiva, es necesario crear una base de datos común para todos los modelos y entrenarlos con ella. El tamaño de los datos es un componente importante para considerar para el almacenamiento y procesamiento de datos en la actualidad., ya que no solo ralentiza el entrenamiento, sino que muchos atributos pueden llevar al algoritmo a encontrar la mejor solución. La extracción y proyección se puede generar para un nuevo dataset con menor número de variables y así lograr métricas muy similares al set original, pero con un mínimo costo en términos de procesamiento de datos y almacenamiento. Es importante señalar que en algunos casos perderemos calidad de datos, y aunque en algunos casos el entrenamiento será más rápido, es posible que no consigamos la misma precisión. Por eso es necesario saber cómo tratar la base de datos y ver si merece la pena o no aplicar una reducción en el número de atributos. Además, permite una mejor visualización de los datos.

A la hora de utilizar el aprendizaje automático para la clasificación, siempre nos encontraremos con dos escenarios a los que debemos enfrentarnos 1) elegir el algoritmo adecuado y 2) que los datos que estamos tratando no tengan la suficiente calidad para tratarlos.

Por lo tanto, requieren una gran cantidad de datos para entrenar y hay muchos algoritmos disponibles para una clasificación precisa. Además, los datos del conjunto de entrenamiento deben ser suficientes y representativos de la base de datos que desea probar. En otras palabras, el modelo obtenido del conjunto de entrenamiento sirve para poder clasificar plenamente nuevos datos que no se han encontrado antes. Si el modelo entrenado sigue una estructura lineal, pero los

nuevos datos a clasificar no están relacionados con los datos de entrenamiento, no se logrará la precisión adecuada para identificar o clasificar esos datos.

También hay que tener en cuenta que los datos que tenemos que entrenar tengan una calidad aceptable, de lo contrario el entrenamiento no será el adecuado y el algoritmo no encontrará un modelo adecuado que sirva para clasificar como debe ser.

5.4 CARACTERISTICAS DE LOS DATOS

Se pudo observar la base de datos históricos que suministró el hotel, de la cual se puede decir que:

- ✓ El conjunto de datos es de tipo numérico y cualitativos.
- ✓ Hay un total de 17.807 datos.
- ✓ Se Tiene un total de 8 variables en la base de datos, de las cuales se tomaron las 4 más relevantes para los algoritmos.

Todos estos datos servirán para llevar a cabo los ajustes de los distintos algoritmos que forman parte del estudio, de esta forma poder examinar los diferentes valores que podemos extraer de todos ellos y realizar un análisis de ellos.

5.5 DEPURACIÓN DE VARIABLES

A la hora de realizar un problema de clasificación es adecuado y conveniente conocer el conjunto de datos a tratar, qué valores tiene, si se pueden eliminar algunos, si falta datos en la base de datos o si todos los datos son numéricos o no. En primer lugar, se analiza el tipo de características de los datos iniciales, ya que hay que tener en cuenta que todos los algoritmos trabajan con datos numéricos. Si fueran categóricos, habría que codificarlos o depurarlos para que todas sean numéricas.

Después de examinar el comportamiento de las variables, se puede diseñar un modelo de reducción de variables si es necesario. Este es un proceso enfocado a reducir el número de variables aleatorias que ingresan en los modelos. Hay varias razones por las que nos puede interesar reducir variables:

- No siempre el mejor modelo es el que más variables contiene.

- Mejora el rendimiento computacional (ahorro de tiempo).
- Reducir la complejidad, lo que lleva a facilitar la comprensión del modelo y sus resultados.

Se utilizó el algoritmo de Arboles de decisión para seleccionar las variables porque puede manejar hasta miles de variables de entrada e identificar las más significativas, además, son muy útiles en la exploración de datos, permiten identificar de forma rápida y eficiente las variables (predictores) más significativas. Para la selección se tomó la base de datos inicial, se dividió está en entrenamiento (70%) y validación (30%). Bajo modalidad de validación cruzada, se entrenó el algoritmo de Árbol de decisión. El resultado de este entrenamiento resultó en la conformación del árbol de decisión que se muestra en el anexo 1, el cual nos muestra las variables en orden de importancia desde el nodo raíz hasta los nodos terminales.

5.5.1 VARIABLES SELECCIONADAS

- ✓ **Mes:** variable de tipo numérico, expresa los diferentes meses del año, un índice que va en el rango de 1 hasta 12.
- ✓ **Tiempo:** Variable de tipo numérico, expresa los días estimados entre la realización de la reserva y la llegada al hotel.
- ✓ **Cancelaciones Previas:** Variable binaria, expresa si el visitante ha cancelado el servicio en el pasado.
- ✓ **Cliente Frecuente:** Variable Binaria, expresa si el visitante ha utilizado los servicios de hotel con frecuencia.
- ✓ **Cancelaciones:** variable de tipo binaria, expresa si una persona cancela la reserva de habitación.

De esta forma, se conforma una estructura de base de datos con 17807 registros con cinco columnas, en donde se detalla cuatro variables de entrada y una variable de respuesta.

Por otro lado, analizando la variable de respuesta, se muestra como están distribuidas las clases en ellas. La clase “Canceló” se muestra en menor proporción que la clase mayoritaria “Check in”. La proporción numérica para la clase minoritaria (Canceló) es de 2893 registros, equivalentes a 16.52% del total, en cambio, para la clase mayoritaria, se tiene un total 14863 registros, para un total de 83.46%.

Es evidente que existe un desbalanceo muy marcado en las clases de la variable de respuesta, situación que propicia la utilización de métodos de balanceo para equilibrar las clases y poder entrenar modelos de forma más robusta.

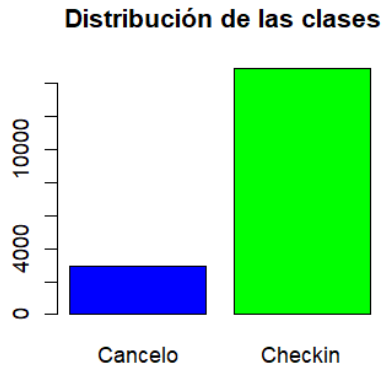

Ilustración 15. Proporción de clases en variable de respuesta. Fuente: elaboración propia.

5.6 MODELOS CON DATOS BALANCEADOS POR MEDIO DE LAS DOS METODOLOGIAS

Al construir el conjunto final, intenta garantizar la confiabilidad de los datos para que la predictibilidad de los modelos utilizados tenga un buen punto de partida, para mejorar bien sea mediante técnicas de balanceo (**SMOTE** y **ADASYM**) o por ajustes de hiper parámetros.

Se consolida la estructura de los datos al momento de aplicar las técnicas ya descritas y se analizan las diferencias de rendimiento de las técnicas de clasificación aplicando los diferentes métodos de balanceo de datos. Se presentan los resultados obtenidos por sobremuestreo en combinación con diferentes modelos de clasificación. Todas las actividades relacionadas con la construcción y aplicación de modelos se realizan en el lenguaje de programación R. La aplicación de los métodos de clasificación y extracción de datos se realiza mediante el paquete de análisis y procesamiento de datos. “caret”, una biblioteca que le permite integrar aplicación de modelado y extracción de métricas. Para los métodos de balanceo de datos se trabaja exclusivamente con el paquete “Smotefamily”, librería que al igual que la anterior permite realizar oversampling en función de un paquete. Para soporte gráfico, la herramienta de visualización seleccionada es "ggplot2".

Ya aplicado los modelos con la data desbalanceada, se propone el empleo de 2 técnicas de balanceo en combinación con los 5 modelos de clasificación ya explicados:

- ✓ Árbol de decisión
- ✓ Naive bayes (clasificador bayesiano)
- ✓ Bosques aleatorios
- ✓ Vecinos más cercanos
- ✓ Máquina de soporte vectorial

Al igual que con los modelos de clasificación y la data original se divide los datos en datos de entrenamiento y datos de test. El conjunto de datos de entrenamiento representa el 70% total de los datos y sobre se realiza la construcción de los modelos. El 30% restante, corresponde al conjunto de test sobre el que se validan los modelos. Realizar esta división se procede iterativamente en un bucle a realizar a una secuencia de operaciones con cada modelo de clasificación. Cada iteración corresponde a una técnica de balanceo diferente. La secuencia es:

1. Balanceo de datos por el método **SMOTE** y **ADASYN** para cada modelo de clasificación.
2. Construcción de los modelos de clasificación.
3. Validación datos de prueba (predicciones).
    - Extracción de métricas de rendimiento
    - Visualización

La secuencia anteriormente explicada se realiza por separado para cada una de las 5 modelos de clasificación. De esta forma es posible observar la influencia de las técnicas de balanceo de datos tienen sobre el rendimiento de los modelos de clasificación.

6 ANALSIS DE RESULTADOS.

6.1 MODELOS DESBALANCEADOS DATOS ORIGINALES

En primer lugar, se trabajarán todos los algoritmos de clasificación con la base de datos desbalanceadas, esto para verificar el rendimiento de estos, además, servirá para evaluar si los métodos de balanceo propuestos inciden en la mejora de los modelos estudiados. Para todos los modelos, se entrenó con una base de datos

correspondientes al 70% de los registros y se realizó la validación con el 30% restante. En el entrenamiento se realizó con validación cruzada garantiza independencia y esto a su vez evita sobre ajuste.

6.1.1 ÁRBOL DE DECISIÓN

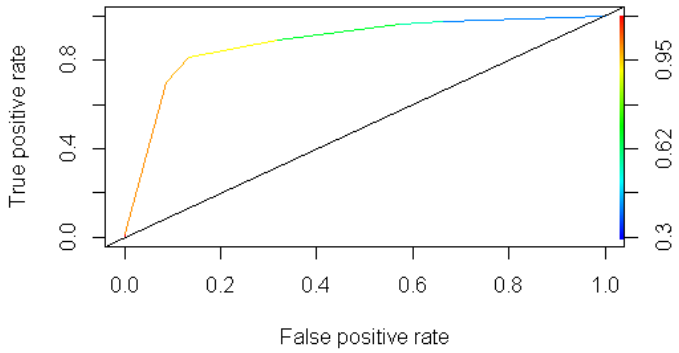

Ilustración 16. Curva ROC del modelo de árbol decisión / decisión tree. Fuente: R studio.

Albol Decision	Resultados
Roc	0.87
F1-score	0.52
Accuracy	0.86
Precision	0.70
Recall	0.42
AUC	0.88

Tabla 4. Variable de eficiencia del modelo de árbol decisión / decisión tree. Fuente: Elaboración propia.

Cp	ROC	Sens	Spec
0.005337215	0.8751441	0.4099995	0.9611706
0.014070839	0.8689084	0.3381924	0.9709751
0.024017467	0.8681667	0.3236293	0.9736660

Tabla 6. Resultados de re-muestreo a través de parámetros de ajuste del modelo árbol de decisión. Fuente: Elaboración Propia.

	Cancelo	Check_In
Cancelo	355	168
Check_In	527	4290

Tabla 5. Matriz de confusión para el modelo de árbol de decisión. Fuente: Elaboración propia.

En la fase de entrenamiento, se obtuvo por medio de validación cruzada, el mejor modelo de árbol de decisión. Con un parámetro Cp de 0.0053, el modelo seleccionado tiene un ROC de 0.8751. Cp es un parámetro global que viene incorporado en el paquete “rpart” del software R, la función principal de este parámetro es ahorrar tiempo de cálculo eliminando divisiones que obviamente no valen la pena, por lo tanto, optimizar este parámetro dará como resultado el mejor modelo entrenado.

En la fase de validación, con el mejor modelo escogido anteriormente, se obtiene los siguientes resultados: El modelo de Árbol de decisión presentó un valor de F1-

score de 0.52, un accuracy o exactitud de 0.86 y una precisión más elevada que el recall o sensibilidad, 0.70 y 0.42 respectivamente. Este modelo fue capaz de predecir educadamente el 86% de los casos de prueba.

Ahora bien, en esta fase, se tienen 882 casos de clase “Canceló” y 4458 casos de clase “Checkin”. El modelo de árbol de decisión entrenado logró clasificar correctamente el 40.24% las observaciones en la clase minoritaria, un poder predictivo bajo. Por otra parte, logró clasificar un 96.23% de las observaciones de la clase mayoritaria, es un resultado esperado, puesto que el sistema se vuelve miope al momento de entrenarse con una clase predominante en los datos. Por lo que no se debe basar el análisis solo en la métrica de exactitud, puesto que esta, como se demostró en los resultados, es alta. La métrica F1, da un análisis más globalizado del rendimiento, con un valor de 0.52, muestra que existe un número alto de observaciones clasificadas ya sea como falsos positivos o como falsos negativos, en este caso, esa mala clasificación se da en los falsos negativos. Bajo esta premisa, el clasificador basado en este método tiene un poder predictivo bajo.

6.1.2 MODELO NAIVE BAYES

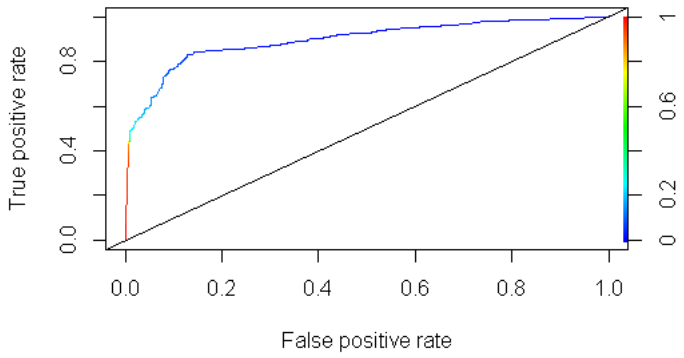

Ilustración 17. Curva ROC modelo Naive Bayes. Fuente: R studio.

Naive Bayes	Resultados
Roc	0.89
F1-score	0.41
Accuracy	0.55
Precision	0.29
Recall	0.99
AUC	0.896

Tabla 7. Variable de eficiencia del modelo Naive Bayes. Fuente: Elaboración propia.

UserKernel	ROC	Sens	Spec
False	0.7802323	0.9941818	0.09514557
True	0.8939690	0.9912692	0.4548759

Tabla 9. Resultados de re-muestreo a través de parámetros de ajuste del modelo Naive Bayes. Fuente: Elaboración propia.

	Cancelo	Check_In
Cancelo	875	2382
Check_In	7	2076

Tabla 8. Matriz de confusión para el modelo de Naive Bayes. Fuente: Elaboración propia

En la fase de entrenamiento, se obtuvo por medio de validación cruzada, el mejor modelo de Naive bayes. El parámetro de giro laplace se mantuvo constante en un valor de 0. El ajuste se mantuvo constante en un valor de 1, la métrica ROC se utilizó para seleccionar el modelo óptimo utilizando el valor más grande, el valor final para el modelo fue lapace = 0, userkernel = True y ajuste = 1 que da como resultado un ROC de 0.89.

En la fase de validación, el modelo de Naive bayes presento un valor de F1-score de 0.41, un Accuracy de 0.55 y una precisión más baja que el Recall, 0.26 y 0.99 respectivamente. Este modelo fue capaz de predecir adecuadamente el 55% de los casos de prueba(positivos). La métrica F1-score, muestra que el clasificador reconoce muy bien la clase minoritaria, pero en contraparte, la clase mayoritaria no la reconoce muy bien, equivocándose en más del 50% de las observaciones, clasificándolas como falsos positivos. En términos generales, un clasificador de bajo rendimiento.

6.1.3 BOSQUES ALEATORIOS

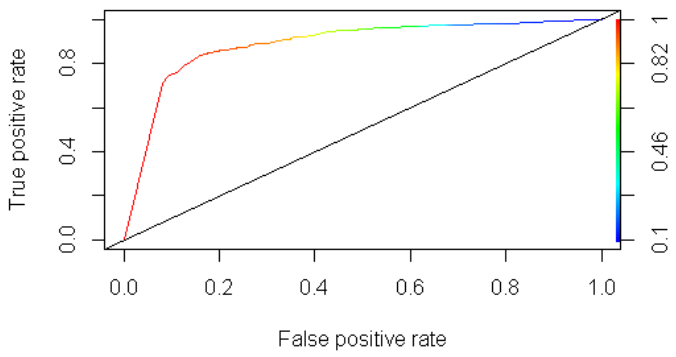

Ilustración 18. Curva ROC modelo bosques aleatórios. Fuente: R studio.

mtry	ROC	Sens	Spec
2	o.8818629	0.3905609	0.9676115
3	0.8810663	0.4216266	0.9646320
4	0.8719100	0.5895314	0.9220577

Tabla 12. Resultados de re-muestreo a través de parámetros de ajuste del modelo Bosques Aleatorios. Fuente: Elaboración propia.

Bosques Aleatorios	Resultados
Roc	0.88
F1-score	0.55
Accuracy	0.87
Precision	0.72
Recall	0.39
AUC	0.8882

Tabla 10. Variable de eficiencia del modelo Bosques Aleatorios. Fuente: Elaboración Propia.

	Cancelo	Check_In
Cancelo	379	176
Check_In	503	4282

Tabla 11. Matriz de confusión para el modelo de Bosques Aleatorios. Fuente: Elaboración propia

En la fase de entrenamiento, el valor de la métrica ROC mostrada en la figura anterior (ilustración 18) se utilizó para seleccionar el modelo óptimo utilizando el valor más grande, el valor final utilizado para el modelo fue mtry = 2, el cual nos arrojó un ROC de 0.88.

En la fase de validación, el modelo de Bosques Aleatorios presento un valor de F1-score de 0.55, un Accuracy de 0.87 y una precisión más elevada que el Recall, 0.72 y 0.39 respectivamente. Este modelo fue capaz de predecir adecuadamente el 87% de los casos de prueba. En esta fase el modelo de bosques aleatorios logro clasificar adecuadamente el 42.97% de las observaciones de la clase (cancelo), también logró clasificar un 96.05% de las observaciones de la clase mayoritaria, este resultado es esperado ya que es la clase predominante. La métrica F1-score, nos muestra que existe un número alto de observaciones clasificadas ya sea como falsos positivos o como falsos negativos, en este caso, esa mala clasificación se da en los falsos negativo. Bajo esta premisa, el clasificador basado en este método tiene un poder predictivo bajo

6.1.4 VECINOS MAS CERCANOS (K-NEAREST NEIGHBORS)

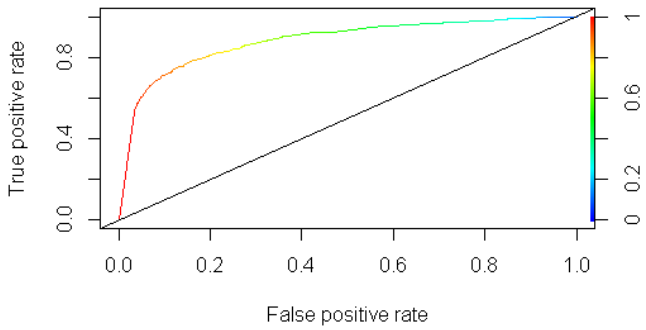

Ilustración 19. Curva ROC modelo Vecinos más cercanos (K-Nearest Neighbors). Fuente: R studio.

K-NN	Resultados
Roc	0.87
F1-score	0.54
Accuracy	0.86
Precision	0.60
Recall	0.46
AUC	0.8803

Tabla 13. Variable de eficiencia del modelo (K-Nearest Neighbors). Fuente: Elaboración Propia.

K	ROC	Sens	Spec
5	0.8778477	0.4861592	0.9418528
7	0.8790075	0.4643263	0.9433915
9	0.8789921	0.4400614	0.9474280

Tabla 15. Resultados de re-muestreo a través de parámetros de ajuste del modelo (K-Nearest Neighbors). Fuente: Elaboración Propia.

	Cancelo	Check_In
Cancelo	425	248
Check_In	457	4210

Tabla 14. Matriz de confusión para el modelo de K-Nearrest Neighbors. Fuente: Elaboración propia

En la fase de entrenamiento la métrica ROC mostrada en la figura anterior (ilustracion19) se utilizó para seleccionar el modelo óptimo utilizando el valor más grande, el valor final utilizado para el modelo fue K =7, el cual nos arrojó un ROC de 0.87.

En la fase de validación el modelo de Vecinos más cercanos (K-Nearest Neighbors) presento un valor de F1-score de 0.54, un Accuracy de 0.86 y una precisión más elevada que el Recall, 0.60 y 0.46 respectivamente. Este modelo fue capaz de predecir adecuadamente el 86% de los casos de prueba. En esta fase el modelo logro clasificar adecuadamente 48.18% de las observaciones de la clase cancelo, también logro clasificar el 94%.43% de las observaciones de la clase mayoritaria, este resultado es esperado ya que es la clase predominante. La métrica F1-score, nos muestra que existe un número alto de observaciones clasificadas ya sea como falsos positivos o como falsos negativos, en este caso, esa mala clasificación se da en los falsos negativos. Bajo esta premisa, el clasificador basado en este método tiene un poder predictivo bajo

6.1.5 MÁQUINA DE SOPORTE VECTORIAL (SVM).

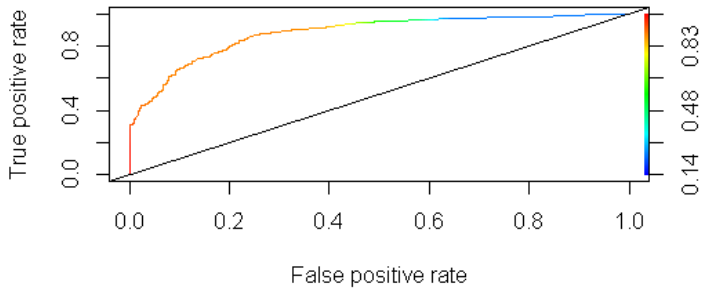

Ilustración 20. Curva ROC modelo Máquina de soporte vectorial (SVM). Fuente: R studio.

SVM	Resultados
Roc	0.86
F1-score	0.51
Accuracy	0.86
Precision	0.70
Recall	0.43
AUC	0.88

Tabla 16. Variable de eficiencia del modelo Maquina de soporte vectorial (SVM). Fuente: Elaboración propia.

C	ROC	Sens	Spec
0.25	0.8577904	0.3910933	0.9646340
0.50	0.86322552	0.4032151	0.9633846
1.00	0.8603205	0.4119530	0.9627119

Tabla 17. Resultados de re-muestreo a través de parámetros de ajuste del modelo Máquina de suporte vectorial (SVM). Fuente: Elaboración propia.

	Cancelo	Check_In
Cancelo	373	189
Check_In	509	4269

Tabla 18. Matriz de confusión para el modelo de Maquina de soporte vectorial. Fuente: Elaboración propia.

En la fase de entrenamiento, el parámetro de ajuste sigma se mantuvo constante en un valor de 3,77, ROC se utilizó para seleccionar el modelo óptimo utilizando el valor más grande. el valor final utilizado para el modelo fue sigma = 3.77 y C=0.5, el cual nos arrojó un ROC de 0.86.

En la fase de validación, el modelo de Máquina de suporte vectorial (SVM) presentó un valor de F1-score de 0.51, un Accuracy de 0.86 y una precisión más elevada que el Recall, 0.70 y 0.43 respectivamente. Este modelo fue capaz de predecir adecuadamente el 86% de los casos de prueba. Este modelo fue capaz de predecir correctamente el 42.29% de las observaciones de la clase (cancelo), también logró clasificar un 95.76% de las observaciones de la clase mayoritaria, este resultado es esperado ya que es la clase predominante de los datos. La métrica F1-score, nos muestra que existe un número alto de observaciones clasificadas ya sea como falsos positivos o como falsos negativos, en este caso, esa mala clasificación se da en los falsos positivos. Bajo esta premisa, el clasificador basado en este método tiene un poder predictivo bajo

6.1.6 COMPARACIÓN DE MODELOS DESBALANCEADOS

Tabla 19. Tabla de variables de efectividad de todos los modelos. Fuente: Elaboración propia.

	Roc	F1-score	Accuracy	Precisión	Recall	AUC
Arbol de Decision	0.87	0.52	0.86	0.70	0.42	0.88
Naive Bayes	0.89	0.41	0.55	0.26	0.99	0.896
Bosques Aleatorios	0.88	0.55	0.87	0.72	0.39	0.8882.
K-Nearest Neighbors	0.87	0.54	0.86	0.60	0.46	0.8803.
SVM	0.86	0.51	0.86	0.70	0.43	0.8810.

En la fase de validación, los modelos mostraron una Accuracy entre 0.55 y 0.87 (tabla 19) donde el valor de Accuracy más elevado lo obtuvo el modelo de bosques aleatorios, seguido por el modelo de vecinos más cercanos y el modelo de máquina de soporte vectorial. El modelo que mostró peores resultados en Accuracy fue el modelo de clasificador bayesiano.

La métrica de precisión y Recall presentó valores más diversos, el modelo que mostró un valor más elevado en la variable de precisión fue el modelo de bosques aleatorios y el modelo que presentó valor más bajo fue el modelo de Naive Bayes. En términos generales, aunque tanto la métrica ROC como el AUC, se muestran elevados, La clasificación de falsos negativos y de falsos positivos para los modelos utilizados es de igual manera alta. No existe un equilibrio entre reconocer de manera

correcta las clases, algunos modelos, reconocen muy bien la clase mayoritaria, otros la clase minoritaria, pero, en general ninguno de los modelos en las condiciones estudiadas, se puede catalogar como aceptable para intentar predecir las cancelaciones.

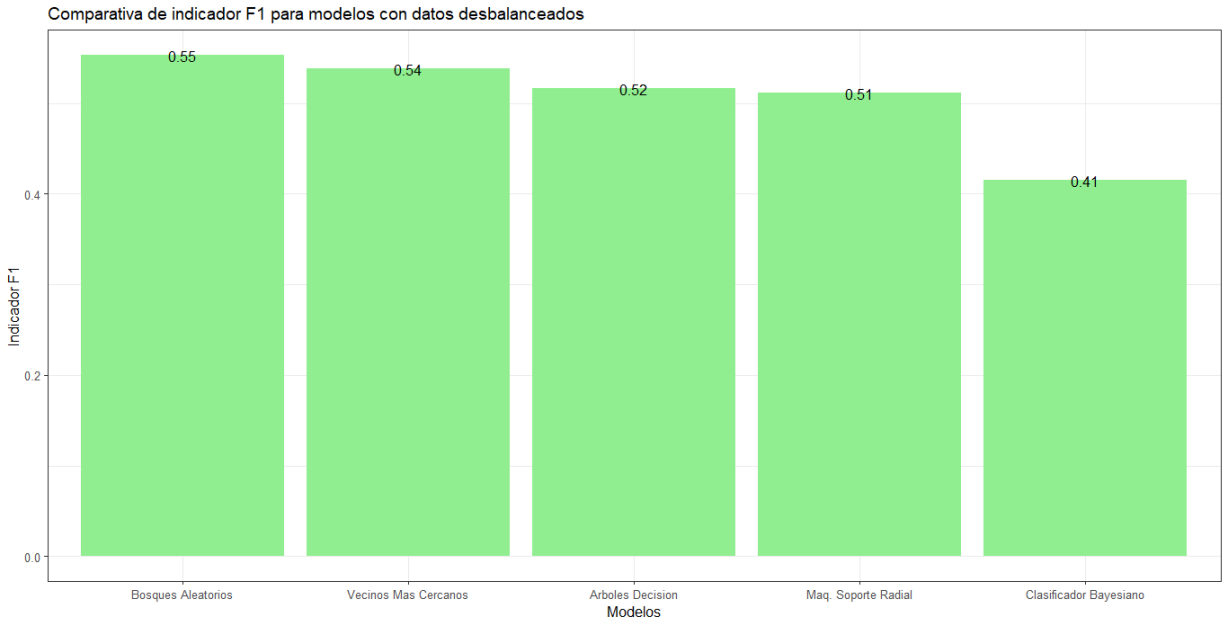

Ilustración 21. Métrica F1-score de todos los modelos. Fuente: R studio.

La métrica F1, da un análisis más profundo del rendimiento, nos muestra un número alto de observaciones clasificadas ya sea como falsos positivos o como falsos negativos, la cual engloba la precisión y el Recall (, con el objetivo de mantener alejado los falsos negativos y los falsos positivos. El modelo que mostró el valor más elevado de bosques aleatorios y el modelo que presento el valor más bajo fue el de Naive Bayes.

6.2 MODELOS BALANCEADOS POR EL MÉTODO SMOTE

Después de entrenar y validar los diferentes modelos con la base de datos original, se prosigue a entrenar y validar estos con el primer método de balanceo. Los resultados encontrados fueron los siguientes:

6.2.1 ARBOL DE DECISION

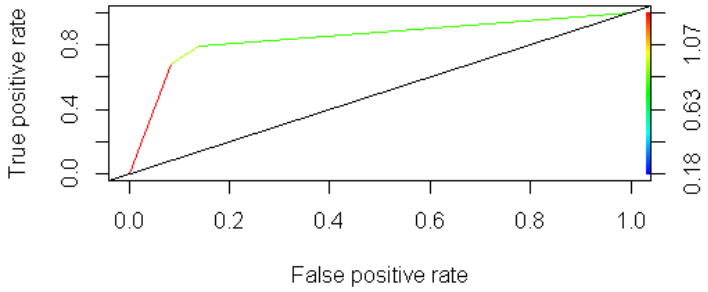

Ilustración 22. Curva ROC del modelo de árbol decisión decisión tree balanceado por método smote. Fuente: R studio.

Decision Tree	Resultados
ROC	0.8624
F1-Score	0.61
Accuracy	0.8046
Precision	0.4529
Recall	0.8616
AUC	0.8423

Tabla 20. Variable de eficiencia del modelo de árbol decisión / decisión tree balanceado por método smote.

CP	ROC	Sens	Spec
0.1397380	0.8624324	0.9110139	0.8001936
0.08306647	0.8439239	0.9193581	0.7587765
0.62008734	0.6216035	0.3751629	0.8680441

Tabla 22. Resultados de re-muestreo a través de Parámetros de ajuste del modelo árbol de decisión balanceado por el método smote. Fuente: Elaboración Propia.

	Cancelo	Check_In
Cancelo	761	922
Check_In	121	3536

Tabla 21. Matriz de confusión para el modelo de árbol de decisión balanceado por el método smote. Fuente: Elaboración propia

En la fase de entrenamiento, se obtuvo por medio de validación cruzada, el mejor modelo de árbol de decisión. Con un parámetro CP = 0.1397380, el modelo seleccionado tiene un ROC de 0.8624.

En la fase de validación, con el mejor modelo escogido anteriormente, se obtiene los siguientes resultados: el modelo de árbol de decisión balanceado por el método SMOTE presentó un valor de F1-score de 0.61, un Accuracy de 0.8046 y una precisión menor que el Recall, 0.4529 y 0.8616 respectivamente. Este modelo fue capaz de predecir adecuadamente e el 80% de los casos de prueba.

Ahora bien, en esta fase, se tienen 882 casos de clase “Canceló” y 4458 casos de clase “Checkin”. El modelo de árbol de decisión entrenado por el método SMOTE logró clasificar correctamente el 86.28% las observaciones en la clase minoritaria, un poder predictivo alto. Por otra parte, logró clasificar un 79.31% de las observaciones de la clase mayoritaria. Se puede evidenciar la mejora sustancial que

obtuvo el modelo con el balanceo por el método SMOTE. La métrica F1, da un análisis más globalizado del rendimiento, con un valor de 0.61, muestra que existe un número de observaciones clasificadas ya sea como falsos positivos o como falsos negativos. Se puede notar que mejora la clasificación en la tasa minoritaria, reduciendo la clasificación de falsos negativos, pero aún, presenta dificultades para reconocer la clase mayoritaria puesto que la tasa de falsos positivos es considerable. Bajo esta premisa, al balancear el modelo bajo el método SMOTE se considera un clasificador aceptable.

6.2.2 CLASIFICADOR BAYESIANO

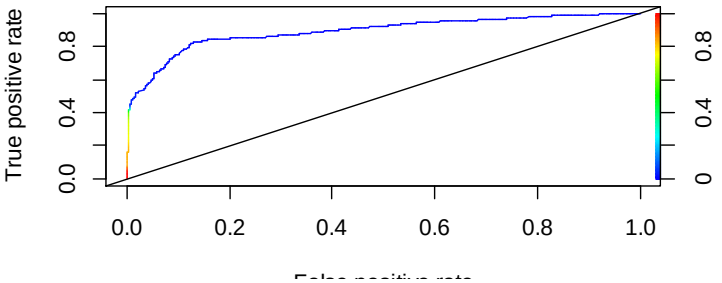

Ilustración 23. Curva ROC modelo Naive Bayes balanceado por el método smote. Fuente: R studio.

clasificador Bayesiano	Resultados
ROC	0.8876
F1-Score	0.34
Accuracy	0.4357
Precision	0.2511
Recall	0.9954
AUC	0.8919

Tabla 23. Variable de eficiencia del modelo Naive Bayes balanceado por el método smote. Fuente: Elaboración propia.

Usekernel	ROC	Sens	Spec
False	0.7836865	0.99301335	0.08302926
True	0.8876896	0.9926248	0.2076872

Tabla 24. Resultados de re-muestreo a través de parámetros de ajuste del modelo Naive Bayes balanceado por el método smote. Fuente: Elaboración propia.

	Cancelo	Check_In
Cancelo	874	3005
Check_In	8	1453

Tabla 25. Matriz de confusión para el modelo de Naive Bayes balanceado por el método smote. Fuente: Elaboración propia

En la fase de entrenamiento, se obtuvo por medio de validación cruzada, el mejor modelo de Naive bayes, el parámetro de giro laplace se mantuvo constante en un valor de 0. El ajuste se mantuvo constante en un valor de 1, la métrica ROC se utilizó para seleccionar el modelo óptimo utilizando el valor más grande, el valor final para el modelo fue lapace = 0, userkernel = True y adjunto = 1, R studio arrojó como resultado un ROC de 0.8876.

En la fase de validación, el modelo de Naive bayes balanceado por el método SMOTE presentó un valor de F1-score de 0.34, un Accuracy de 0.4357 y una precisión más baja que el Recall, 0.2511 y 0.9954 respectivamente. Este modelo fue capaz de predecir correctamente el 43% de los casos de prueba. La métrica F1-score, muestra que el clasificador reconoce muy bien la clase minoritaria, pero en contraparte, la clase mayoritaria no la reconoce muy bien, equivocándose en el 67.41% de las observaciones, clasificándolas como falsos positivos.

6.2.3 BOSQUES ALEATORIOS

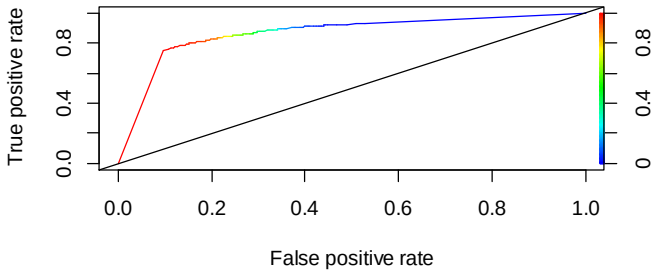

Ilustración 24. Curva ROC modelo bosques aleatorios balanceado por el medo smote. Fuente: R studio.

Bosques Aleatorios	Resultados
ROC	0.9360
F1-Score	0.62
Accuracy	0.8455
Precision	0.5076
Recall	0.7131
AUC	0.8664

Tabla 26. Variable de eficiencia del modelo Bosques Aleatorios balanceado por el método smote. Fuente:

mtry	ROC	Sens	Spec
2	0.9269304	0.9112070	0.8410403
3	0.9341663	0.9132447	0.8422895
4	0.9360136	0.9207186	0.8678527

Tabla 27. Resultados de re-muestreo a través de parámetros de ajuste del modelo Bosques Aleatorios balanceado por el método smote. Fuente: Elaboración propia.

	Cancelo	Check_In
Cancelo	654	597
Check_In	228	3861

Tabla 28. Matriz de confusión para el modelo de Bosques aleatorios balanceado por el método smote. Fuente: Elaboración propia

En la fase de entrenamiento, el valor de la métrica ROC mostrada en la figura anterior se utilizó para seleccionar el modelo óptimo utilizando el valor más grande, el valor final utilizado para el modelo fue mtry = 4, el cual nos arrojó un ROC de 0.9360.

En la fase de validación, el modelo de Bosques Aleatorios balanceado por el método SMOTE presentó un valor de F1-score de 0.62, un Accuracy de 0.8455 y una

precisión menor que el Recall, 0.5076 y 0.7131 respectivamente. Este modelo fue capaz de predecir adecuadamente el 84% de los casos de prueba. En esta fase el modelo de bosques aleatorios logro clasificar adecuadamente el 74.14% de las observaciones de la clase (cancelo), también logró clasificar un 86.60% de las observaciones de la clase mayoritaria, este modelo balanceado por el método SMOTE mejoró el 31.17% con respecto al mismo modelo sin balancear. La métrica F1, da un análisis más globalizado del rendimiento, con un valor de 0.62, muestra que existe un número alto de observaciones clasificadas ya sea como falsos positivos o como falsos negativos, en este caso, esa mala clasificación se da en los falsos positivos. Bajo esta premisa, al balancear el modelo bajo el método smote tiene un poder predictivo aceptable.

6.2.4 VECINOS MAS CERCANOS (K-NEAREST NEIGHBORS)

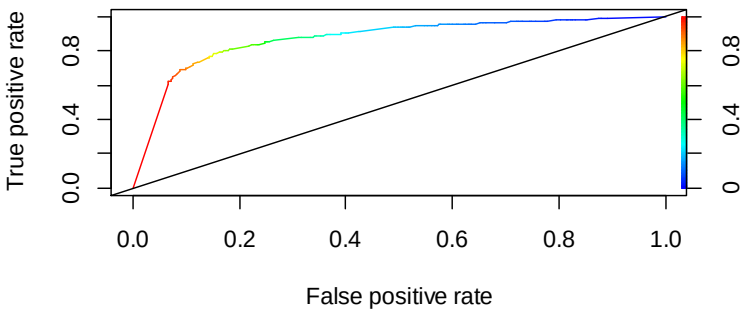

Ilustración 25. Curva ROC modelo Vecinos más cercanos balanceado por el método smote. Fuente: R studio.

KNN	Resultados
ROC	0.9244
F1-Score	0.61
Accuracy	0.8275
Precision	0.4922
Recall	0.7517
AUC	0.8716

Tabla 29. Variable de eficiencia del modelo (K-Nearest Neighbors) balanceado por el método smote.

K	ROC	Sens	Spec
5	0.9244695	0.8834547	0.8336410
7	0.9230007	0.8866559	0.8229736
9	0.9214386	0.8908291	0.8135543

Tabla 30. Resultados de re-muestreo a través de parámetros de ajuste del modelo (K-Nearest Neighbors) balanceado por el método smote. Fuente: Elaboración Propia.

	Cancelo	Check_In
Cancelo	680	719
Check_In	202	3739

Tabla 31. Matriz de confusión para el modelo de KNN balanceado por el método smote. Fuente: Elaboración propia

En la fase de entrenamiento, la métrica ROC mostrada en la figura anterior se utilizó para seleccionar el modelo óptimo utilizando el valor más grande, el valor final utilizado para el modelo fue K =5, el cual nos arrojó un ROC de 0.9244.

En la fase de validación, el modelo de Vecinos más cercanos (K-Nearest Neighbors) balanceado por el método SMOTE presento un valor de F1-score de 0.61, un Accuracy de 0.8272 y una precisión más elevada que el Recall, 0.4922 y 0.7517 respectivamente. Este modelo fue capaz de predecir adecuadamente el 82% de los casos de prueba. En esta fase el modelo logro clasificar adecuadamente 77.09% de las observaciones de la clase (cancelo), también logro clasificar el 83.87% de las observaciones de la clase mayoritaria (check in), se puede evidenciar la mejora sustancial que obtuvo el modelo con el balanceo por el método smote el cual fue del 28.91%. La métrica F1, da un análisis más globalizado del rendimiento, con un valor de 0.61, bajo esta premisa, al balancear el modelo bajo el método smote genera un clasificador aceptable.

6.2.5 MÁQUINA DE SOPORTE VECTORIAL (SVM)

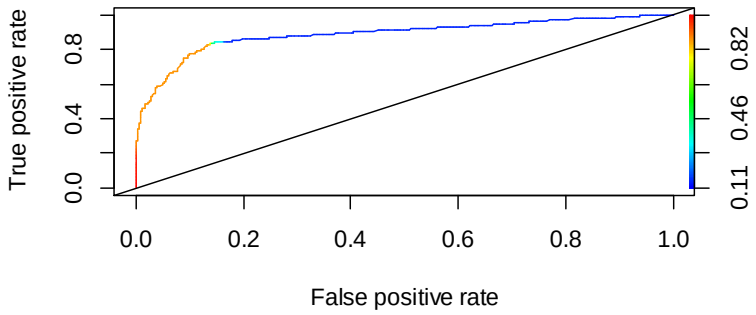

Ilustración 26. Curva ROC modelo Máquina de soporte vectorial (SVM) balanceado por el método smote. Fuente: R studio.

SVM	Resultados
ROC	0.9042
F1-Score	0.65
Accuracy	0.8357
Precision	0.5087
Recall	0.8594
Especificity	0.9661
AUC	0.8890

Tabla 32. Variable de eficiencia del modelo Maquina de soporte vectorial (SVM) balanceado por el método smote. Fuente: Elaboración propia.

C	ROC	Sens	Spec
0.25	0.9001006	0.8797670	0.8329646
0.50	0.8999188	0.8846191	0.8323881
1.00	0.9042821	0.8877243	0.8334453

	Cancelo	Check_In
Cancelo	752	747
Check_In	130	3711

Tabla 34. Resultados de re-muestreo a través de parámetros de ajuste del modelo (SVM) balanceado por el método smote. Fuente: Elaboración propia.

Tabla 33. Matriz de confusión para el modelo de SVM balanceado por el método smote. Fuente: Elaboración propia.

En la fase de entrenamiento, el parámetro de ajuste sigma se mantuvo constante en un valor de 3,27577, ROC se utilizó para seleccionar el modelo óptimo utilizando el valor más grande. el valor final utilizado para el modelo fue sigma = 3.27577 y C=1, el cual nos arrojó un ROC de 0.9042.

En la fase de validación, el modelo de Máquina de suporte vectorial (SVM) balanceado por el método SMOTE presentó un valor de F1-score de 0.65, un Accuracy de 0.8353 una precisión más elevada que el Recall, 0.5087 y 0.8594 respectivamente. Este modelo fue capaz de predecir adecuadamente el 83% de los casos de prueba. Este modelo fue capaz de predecir correctamente el 85.26% de las observaciones de la clase (cancelo), también logró clasificar un 83.24% de las observaciones de la clase mayoritaria, Se puede evidenciar la mejora sustancial que obtuvo el modelo con el balanceo por el método SMOTE obteniendo una mejora del 40.95% en la clase minoritaria. La métrica F1, da un análisis más globalizado del rendimiento, con un valor de 0.65, lo cual es una mejora del 15% aproximadamente con respecto al modelo desbalanceado. Bajo esta premisa, al balancear el modelo bajo el método SMOTE genera un clasificador aceptable.

6.2.6 COMPARACION DE MODELOS BALANCEADO POR EL METODO SMOTE

	Roc	F1-score	Accuracy	Precisión	Recall	AUC
Arbol de Decision	0.8624	0.61	0.8046	0.4529	0.8616	0.8423
Naive Bayes	0.8876	0.34	0.4357	0.2511	0.9954	0.8919
Bosques Aleatorios	0.9360	0.62	0.8455	0.5076	0.7131	0.8664
K-Nearest Neighbors	0.9244	0.61	0.8275	0.4922	0.7517	0.8716
SVM	0.9042	0.65	0.8357	0.5087	0.8594	0.8890

Tabla 35. Tabla de variables de efectividad de todos los modelos balanceados por el método smote. Fuente: Elaboración propia.

En la fase de validación, los modelos de clasificación balanceados por el método SMOTE presentaron una Accuracy entre 0.80 y 0.85, exceptuando el método Naive Bayes el cual presentó el menor valor en comparación a los demás modelos. El valor de Accuracy más elevado lo obtuvo el modelo de bosques aleatorios, seguido por el modelo de máquina de soporte vectorial y el modelo vecino más cercanos. El

modelo que mostró peores resultados en Accuracy fue el modelo de clasificador bayesiano.

La métrica de precisión presentó valores más diversos, el modelo que presentó un valor más elevado fue el modelo de máquinas de soporte vectorial seguido por el modelo de bosques aleatorios, y el modelo que presentó valor más bajo fue el modelo de Naive Bayes.

Para el Recall el cual es la proporción de casos positivos (Cancelaciones) que fueron correctamente reconocidas por el algoritmo. Los modelos presentaron unos valores entre 0.70 y 0.99, el valor más grande lo obtuvo el modelo de clasificador bayesiano, seguido por el modelo de árbol de decisión y máquinas de soporte vectorial. El modelo que mostró peores resultados en Recall fue el modelo de bosques aleatorios.

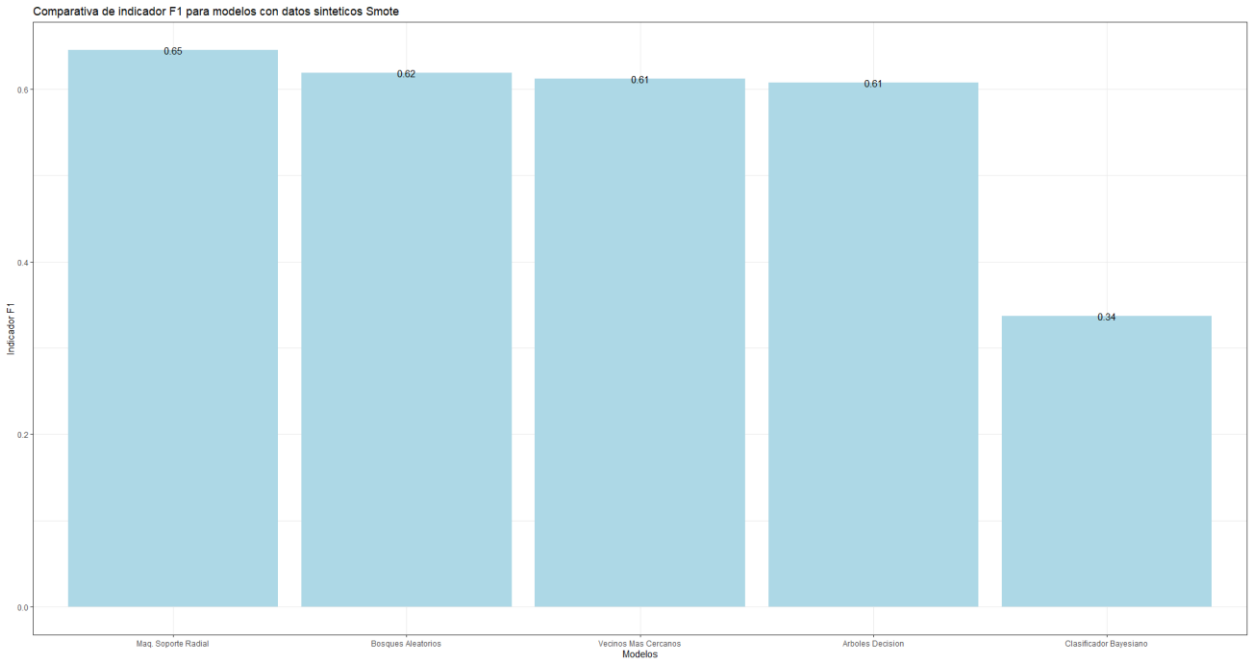

Ilustración 27. Métrica F1-score de todos los modelos balanceados por el método smote. Fuente: R studio.

Por otro lado, la métrica de estudio en nuestro caso F1-score, la cual engloba la precisión y sensibilidad en una sola métrica. Por ello es de gran utilidad cuando la distribución de las clases es desigual, con el objetivo de mantener alejado los falsos negativos y los falsos positivos. En general casi todos los modelos balanceados por el método SMOTE mantuvieron un valor por encima del 0.60, exceptuando el

modelo de Naive Bayes que obtuvo un valor de 0.34. El modelo que mostró el valor más elevado en dicha métrica fue el modelo de SMV máquina de soporte vectorial.

6.3 MODELOS BALANCEADOS POR EL MÉTODO ADASYM

Después de entrenar y validar los diferentes modelos con la base de datos balanceada por el método SMOTE, se prosigue a entrenar y validar estos con el segundo método de balanceo. Los resultados encontrados fueron los siguientes:

6.3.1 ARBOL DE DECISION

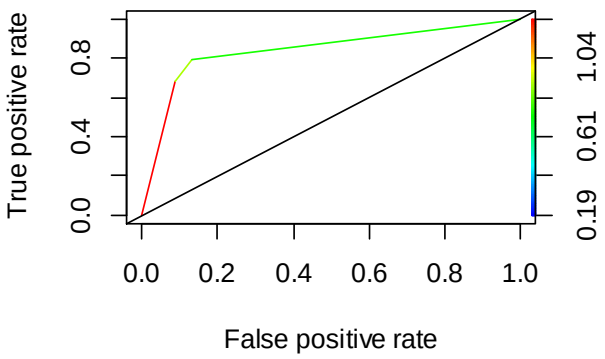

Ilustración 28. Curva ROC modelo árbol de decisión balanceado por el método adasym. Fuente: R studio.

Decision Tree	Resultados
ROC	0.8424
F1-Score	0.61
Accuracy	0.8046
Precision	0.4577
Recall	0.8662
AUC	0.8440

Tabla 36. Variable de eficiencia del modelo árbol de decisión balanceado por el método adasym. Fuente: Elaboración propia.

CP	ROC	Sens	Spec
0.1387952	0.8440987	0.8700734	0.8005788
0.07209639	0.8164206	0.8881947	0.7350259
0.59132530	0.6455960	0.4557826	0.8364095

Tabla 37. Resultados de re-muestreo a través de parámetros de ajuste del modelo árbol de decisión balanceado por el método adasym. Fuente: Elaboración propia.

	Cancelo	Check_In
Cancelo	761	922
Check_In	121	3536

Tabla 38. Matriz de confusión para el modelo de árbol de decisión balanceado por el método adasym. Fuente: Elaboración propia.

En la fase de entrenamiento, se obtuvo por medio de validación cruzada, el mejor modelo de árbol de decisión. Con un parámetro CP = 0.13987952, el modelo seleccionado tiene un ROC de 0.8440.

En la fase de validación, con el mejor modelo escogido anteriormente, se obtiene los siguientes resultados: el modelo de decisión tree balanceado por el meto ADASYM presentó un valor de F1-score de 0.61, un Accuracy de 0.8046 y una precisión menor que el Recall, 0.4577 y 0.8662 respectivamente. Este modelo fue capaz de predecir adecuadamente el 80% de los casos de prueba.

Ahora bien, en esta fase, se tienen 882 casos de clase “Canceló” y 4458 casos de clase “Checkin”. El modelo de árbol de decisión entrenado por el método ADASYM logró clasificar correctamente el 86.28% las observaciones en la clase minoritaria, un poder predictivo alto. Por otra parte, logró clasificar un 79.31% de las observaciones de la clase mayoritaria, se puede evidenciar la mejora sustancial que obtuvo el modelo con el balanceo por el método ADASYM. La métrica F1, da un análisis más globalizado del rendimiento, con un valor de 0.61, muestra que existe un número de observaciones clasificadas ya sea como falsos positivos o como falsos negativos. pero aún, presenta dificultades para reconocer la clase mayoritaria puesto que la tasa de falsos positivos es considerable. Bajo esta premisa, al balancear el modelo bajo el método ADASYM se considera un clasificador aceptable.

6.3.2 MODELO CLASIFICADOR BAYESIANO

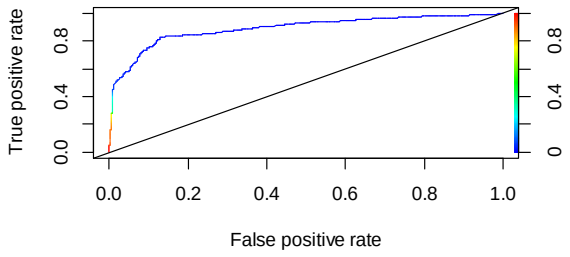

Ilustración 29. Curva ROC modelo clasificador bayesiano balanceado por el método adasym. Fuente: R studio.

clasificador Bayesiano	Resultados
ROC	0.8816
F1-Score	0.36
Accuracy	0.4194
Precision	0.2160
Recall	0.9931
AUC	0.8906

Tabla 39. Variable de eficiencia del modelo clasificador bayesiano balanceado por el método adasym. Fuente: Elaboración propia.

Usekernel	ROC	Sens	Spec
False	0.7623471	0.9900084	0.09985332
True	0.8892541	0.9919485	0.30206597

Tabla 40. Resultados de re-muestreo a través de parámetros de ajuste del modelo clasificador bayesiano balanceado por el método adasym. Fuente: Elaboración propia.

Cancelo	874	3092
Check_In	8	1366

Tabla 41. Matriz de confusión para el modelo de clasificador bayesiano balanceado por el método adasym. Fuente: Elaboración propia.

En la fase de entrenamiento, se obtuvo por medio de validación cruzada, el mejor modelo de Naive bayes, el parámetro de giro laplace se mantuvo constante en un valor de. El ajuste se mantuvo constante en un valor de 1, la métrica ROC se utilizó para seleccionar el modelo óptimo utilizando el valor más grande, el valor final para el modelo fue lapace = 0, userkernel = True y adjunto = 1, R arrojó como resultado un ROC de 0.8892.

En la fase de validación, el modelo de Naive bayes balanceado por el método ADASYM presentó un valor de F1-score de 0.36, un Accuracy de 0.4194 y una precisión más baja que el Recall, 0.2160 y 0.9931 respectivamente. Este modelo fue capaz de predecir adecuadamente el 41% de los casos de prueba. La métrica F1-score, muestra que el clasificador reconoce muy bien la clase minoritaria, pero en contraparte, la clase mayoritaria no la reconoce muy bien, equivocándose en el 69.36% de las observaciones, clasificándolas como falsos positivos.

6.3.3 MODELO BOSQUES ALEATORIOS

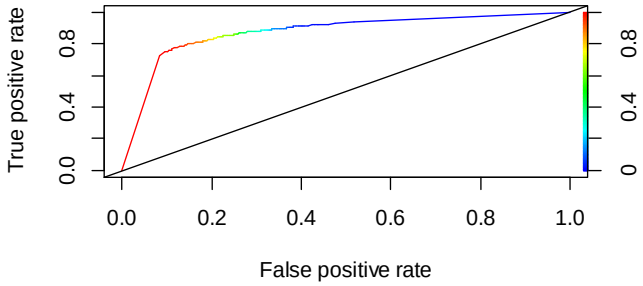

Ilustración 30. Curva ROC modelo bosques aleatorios balanceado por el método adasym. Fuente: R studio.

Bosques Aleatorios	Resultados
ROC	0.9312
F1-Score	0.61
Accuracy	0.8455
Precision	0.5215
Recall	0.7414
AUC	0.8742

Tabla 42. Variable de eficiencia del modelo bosques aleatorios balanceado por el método adasym. Fuente: Elaboración propia.

mtry	ROC	Sens	Spec
2	0.9207759	0.8822371	0.8409437
3	0.9298579	0.8981459	0.8401743
4	0.9312873	0.9174500	0.8709279

Tabla 43. Resultados de re-muestreo a través de parámetros de ajuste del modelo bosques aleatorios balanceado por el método adasym. Fuente: Elaboración propia.

	Cancelo	Check_In
Cancelo	654	600
Check_In	228	3858

Tabla 44. Matriz de confusión para el modelo de bosques aleatorios balanceado por el método adasym. Fuente: Elaboración propia.

En la fase de entrenamiento, la curva ROC mostrada en la figura anterior se utilizó para seleccionar el modelo óptimo utilizando el valor más grande, el valor final utilizado para el modelo fue mtry = 4, el cual nos arrojó un ROC de 0.9312.

En la fase de validación, el modelo de Bosques Aleatorios balanceado por el método Adasym presentó un valor de F1-score de 0.61, un Accuracy de 0.8455 y una precisión menor que el Recall, 0.5215 y 0.7414 respectivamente. Este modelo fue capaz de predecir adecuadamente el 84% de los casos de prueba. En esta fase el modelo de bosques aleatorios logro clasificar adecuadamente el 74.14% de las observaciones de la clase (cancelo), también logró clasificar un 86.60% de las observaciones de la clase mayoritaria, este modelo balanceado por el método ADASYM mejoró el 31.17% con respecto al mismo modelo sin balancear. La métrica F1, da un análisis más globalizado del rendimiento, con un valor de 0.61, muestra que existe un número alto de observaciones clasificadas ya sea como falsos positivos o como falsos negativos. Bajo esta premisa, al balancear el modelo bajo el método ADASYM tiene un predictivo aceptable.

6.3.4 MODELO KNN (VECINOS MAS CERCANOS)

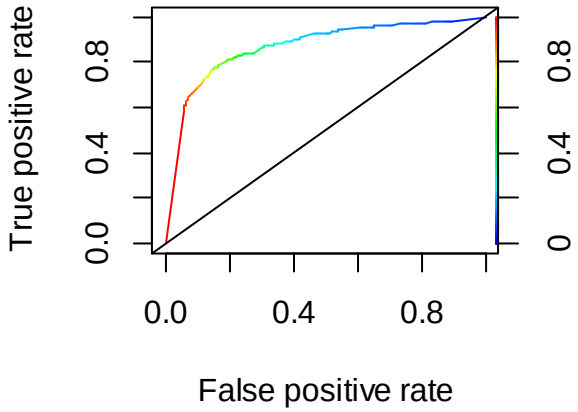

Ilustración 31. Curva ROC modelo KNN balanceado por el método adasym. Fuente: R studio.

KNN	Resultados
ROC	0.9140
F1-Score	0.60
Accuracy	0.8187
Precision	0.4702
Recall	0.7698
AUC	0.8679

Tabla 45. Variable de eficiencia del modelo bosques aleatorios balanceado por el método adasym. Fuente: Elaboración propia.

K	ROC	Sens	Spec
5	0.9140382	0.8706951	0.8271028
7	0.9083797	0.8705982	0.8164351
9	0.9052426	0.8701133	0.8054801

Tabla 47. Resultados de re-muestreo a través de parámetros de ajuste del modelo KNN balanceado por el método adasym. Fuente: Elaboración propia.

	Cancelo	Check_In
Cancelo	679	765
Check_In	203	3693

Tabla 46. Matriz de confusión para el modelo de bosques aleatorios balanceado por el método adasym. Fuente: Elaboración propia.

En la fase de entrenamiento, la curva ROC mostrada en la figura anterior (ilustración 31) se utilizó para seleccionar el modelo óptimo utilizando el valor más grande, el valor final utilizado para el modelo fue K =5, el cual nos arrojó un ROC de 0.9140.

En la fase de validación, el modelo de Vecinos más cercanos (K-Nearest Neighbors) balanceado por el método ADASYM presento un valor de F1-score de 0.60, un Accuracy de 0.8187 y una precisión más elevada que el Recall, 0.4702 y 0.7698 respectivamente. Este modelo fue capaz de predecir adecuadamente el 81% de los casos de prueba(positivos). En esta fase el modelo logro clasificar adecuadamente 76.94% de las observaciones de la clase (cancelo), también logro clasificar el 82.83% de las observaciones de la clase mayoritaria (check in), se puede evidenciar la mejora sustancial que obtuvo el modelo con el balanceo por el método adasym el cual fue del 28.76%. La métrica F1, da un análisis más globalizado del rendimiento, con un valor de 0.60, bajo esta premisa, al balancear el modelo bajo el método smote genera un clasificador aceptable.

6.3.5 MODELO MÁQUINA DE SOPORTE VECTORIAL (SVM)

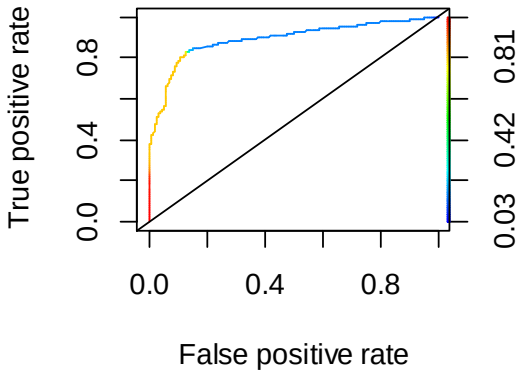

Ilustración 32. Curva ROC modelo SVM balanceado por el método adasym. Fuente: R.

SVM	Resultados
ROC	0.8905
F1-Score	0.65
Accuracy	0.8374
Precision	0.5046
Recall	0.8639
AUC	0.8897

Tabla 48. Variable de eficiencia del modelo SVM balanceado por el método adasym. Fuente: Elaboración propia.

C	ROC	Sens	Spec
0.25	0.8892952	0.8455703	0.8293142
0.50	0.8888375	0.8507118	0.8306601
1.00	0.8905640	0.8571775	0.8336151

Tabla 50. Resultados de re-muestreo a través de parámetros de ajuste del modelo SVM balanceado por el método adasym. Fuente: Elaboración propia.

	Cancelo	Check_In
Cancelo	762	748
Check_In	120	3710

Tabla 49. Matriz de confusión para el modelo de SVM balanceado por el método adasym. Fuente: Elaboración propia.

En la fase de entrenamiento, el parámetro de ajuste sigma se mantuvo constante en un valor de 3 3.386161, ROC se utilizó para seleccionar el modelo óptimo utilizando el valor más grande. el valor final utilizado para el modelo fue sigma = 3.27577 y C=1, el cual nos arrojó un ROC de 0.8905.

En la fase de validación, el modelo de Máquina de suporte vectorial (SVM) balanceado por el método ADASYM presento un valor de F1-score de 0.65, un Accuracy de 0.8374 una precisión menor que el Recall, 0.5046 y 0.8639 respectivamente. Este modelo fue capaz de predecir adecuadamente el 83% de los casos de prueba(positivos). En esta fase el modelo logro clasificar adecuadamente 86.39% de las observaciones de la clase (cancelo), también logro clasificar el 8322% de las observaciones de la clase mayoritaria (check in), se puede evidenciar la mejora sustancial que obtuvo el modelo con el balanceo por el método ADASYM. La métrica F1, da un análisis más globalizado del rendimiento, con un

valor de 0.65, bajo esta premisa, al balancear el modelo bajo el método ADASYM genera un clasificador aceptable

6.3.6 COMPARACION DE MODELOS BALANCEADO POR EL METODO ADASYM

	Roc	F1-score	Accuracy	Precisión	Recall	AUC
Arbol de Decision	0.8424	0.61	0.8046	0.4577	0.8662	0.8440
Naive Bayes	0.8816	0.36	0.4194	0.2160	0.9931	0.8906
Bosques Aleatorios	0.9312	0.61	0.8455	0.5215	0.7414	0.8742
K-Nearest Neighbors	0.9140	0.60	0.8187	0.4702	0.7698	0.8679
SVM	0.8905	0.65	0.8374	0.5087	0.8639	0.8897

Tabla 51. Tabla de variables de efectividad de todos los modelos balanceados por el método adasym. Fuente: Elaboración propia.

En la fase de validación, los modelos de clasificación balanceados por el método adasym presentaron una Accuracy entre 0.80 y 0.84, excluyendo el método Naive Bayes el cual presentó un valor atípico en comparación a los demás modelos. El valor de Accuracy más elevado lo logró el modelo de bosques aleatorios, seguido por el modelo de máquina de soporte vectorial y el modelo vecino más cercanos. El modelo que mostró peores resultados en Accuracy fue el modelo de clasificador bayesiano obteniendo un resultado de 0.4194.

La variable de precisión presento valores más variados, el modelo que mostró un valor más elevado fue el modelo de bosques aleatorios, seguido por el modelo de SVM y el modelo que presentó valor más bajo fue el modelo de Naive Bayes obteniendo un valor de 0.2160

Para el Recall el cual es la proporción de casos positivos (Cancelaciones) que fueron correctamente identificadas por el algoritmo. los modelos presentaron unos valores entre 0.74 y 0.99 (tabla 51), el valor más grande lo obtuvo el modelo de clasificador bayesiano, seguido por el modelo de árbol de decisión y máquinas de soporte vectorial. El modelo que presento peores resultados en Recall fue el modelo de bosques aleatorios.

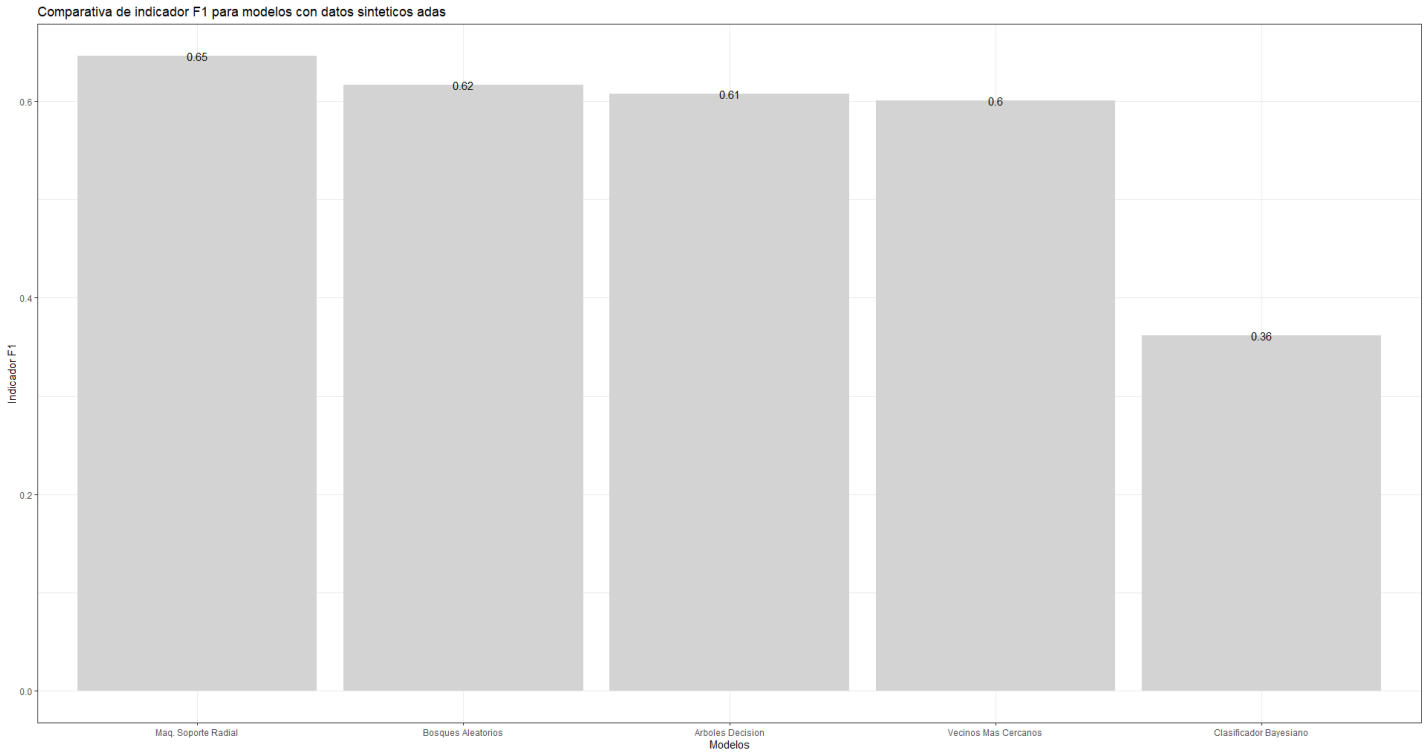

Ilustración 33. Métrica F1-score de todos los modelos balanceados por el método adasym. Fuente: R studio.

La métrica de estudio en nuestro caso F1-score, la cual engloba la precisión y el Recall (ilustración 32), con el objetivo de mantener alejado los falsos negativos y los falsos positivos. El modelo que presento el valor más elevado en dicha métrica fue el modelo de máquina de soporte vectorial con un valor de 0.65, y el modelo que presento el valor más bajo fue el modelo de Naive Bayes obteniendo un valor de 0.36

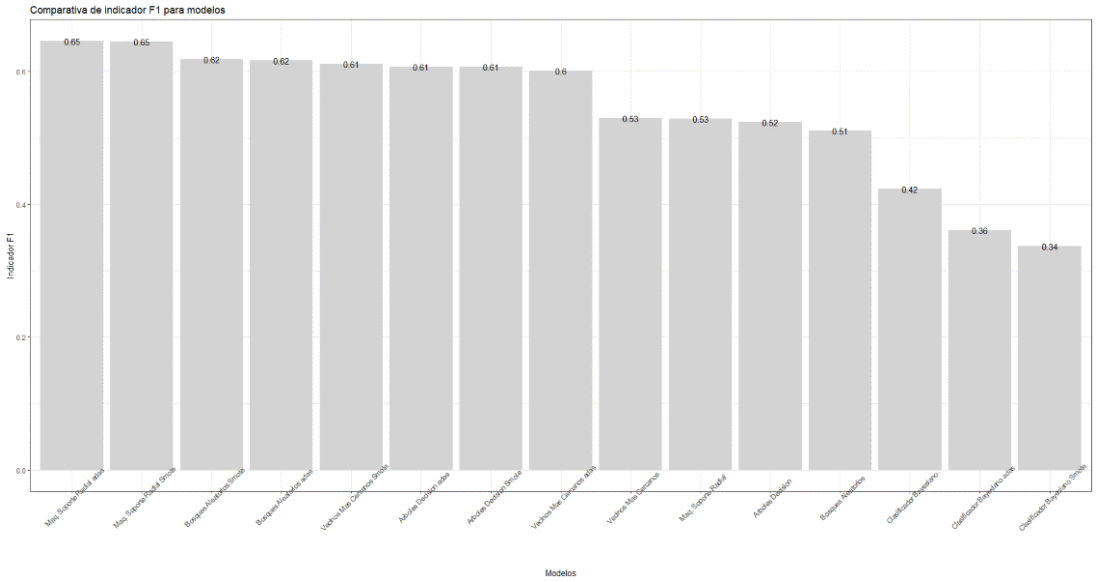

Ilustración 34. Métrica F1-Score todos los modelos vistos con sus respectivos balanceos. Fuente: R.

En la (ilustración 34) se muestran los modelos que mejores resultados obtuvieron en la métrica F1-score para el problema propuesto.

SVM DESBALANCEADO		
	Cancelo	**Check_In**
Cancelo	373	189
Check_In	509	4269

SVM BALANCEADO METODO SMOTE		
	Cancelo	**Check_In**
Cancelo	752	747
Check_In	130	3711

Ilustración 35. Comparación de la matriz de confusión del mejor modelo obtenido. Fuente: Elaboración propia.

En la (ilustración 34 y 35) se aprecia con mayor claridad el comportamiento del modelo que obtuvo mejores resultados en la métrica F1-socre, en las matrices de confusión se observa como al aplicar oversampling con la técnica SMOTE mejora sistemáticamente respecto a los distintos modelos estudiados y las distintas técnicas de balanceo. El modelo máquina de soporte vectorial paso de tener 373 cancelaciones, a tener 752 cancelaciones que corresponde a los verdaderos positivos, y paso de tener 509 falsos positivos a 130, lo cual corresponde a un margen de mejora del 42.97%.

7. PASOS PARA LA INTEGRACIÓN DEL MEJOR MODELO AL PROCESO DEL HOTEL

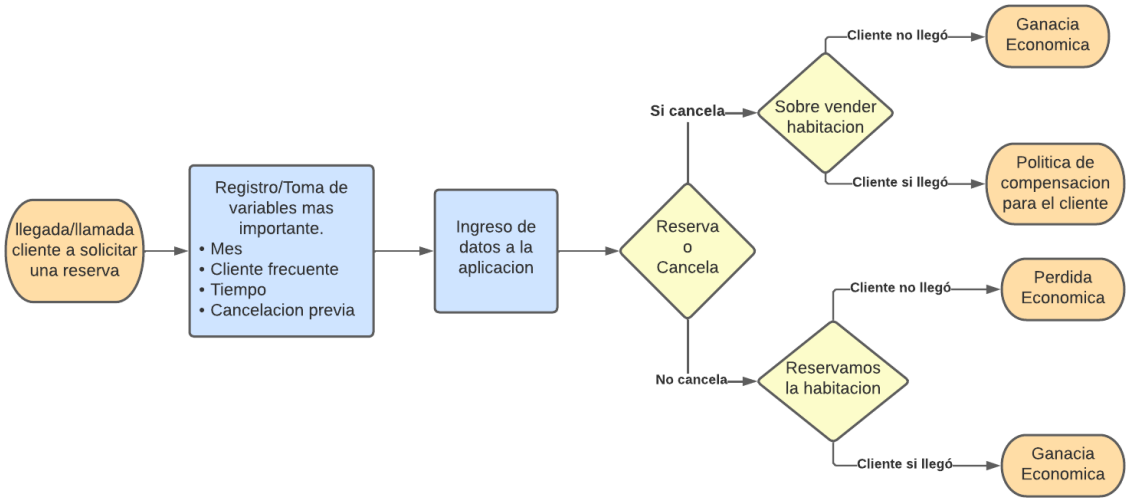

Ilustración 36. Flujo grama de los pasos para la integración del modelo con mejores resultados. Fuente: elaboración propia.

Paso 1: El cliente solicita una reserva al hotel esta puede hacerse tanto presencial como por vía telefónica.

Paso 2: la persona encargada de la recepción le toma los datos al cliente incluido las variables más importantes que necesita el modelo para su correcto funcionamiento.

Paso 3: ingresar las variables al aplicativo, que previamente debe estar cargado con los datos históricos para que así nos arroje la predicción del estatus del cliente.

- La aplicación nos proyecta dos decisiones que puede suceder con la reserva:
  1 El cliente cancela.
  2 El cliente hace check in.
- Si la aplicación nos dice que el cliente cancela la reserva (opción 1) podemos tomar la decisión de sobrevender la habitación, si el cliente no llego esto beneficiaria al hotel económicamente,
  en cambio, si el cliente hace check in (opción 2) deberíamos compensar al cliente ya sea buscándole algún hotel con las mismas características o mejores condiciones.
- Si el cliente hace check in tomamos la decisión de reservar la habitación, corriendo el riesgo de que el cliente no llegue al hotel, lo que incurriría en una pérdida económica, por el contrario, si el cliente si llega al hotel generaría una ganancia económica para este.

8. CONCLUSIONES Y RECOMENDACIONES

El objetivo de este estudio fue realizar una comparación de los diferentes modelos de clasificación basados en aprendizaje automático que podrían utilizarse para la predicción de cancelaciones en un hotel y, a su vez, cuantificar y comparar los modelos construidos para detectar qué modelos tienen mayor precisión a la hora de identificar interacciones entre una persona que va a cancelar o no una reserva.

Para este estudio se logró adaptar 5 modelos con metodología machine learning para la predicción de cancelaciones. Se utilizó la misma base de datos para todos los modelos. Todos los modelos mostraron valores significativos de efectividad para predecir cancelaciones en base a los datos utilizados y con los diferentes métodos de balanceo. El modelo que peores resultados obtuvo fue el modelo clasificador bayesiano, que en 3 de las 6 variables de eficiencia estudiadas presenta los niveles más bajos en relación al resto de modelos balanceados con las 2 técnicas utilizadas (smote y adasyn). Por lo contrario, el modelo basado en máquinas de soporte vectorial presento los mejores resultados, con el F1-score más elevado de todos los modelos al igual que con las dos técnicas de balanceo, las curvas ROC indicaron que los modelos más sólidos fueron SVM, KNN y bosques aleatorios.

Podríamos afirmar que con la base de datos suministrada por la organización el modelo que obtuvo mejores predicciones fue el modelo de SVM balanceado por el método smote, dado los buenos resultados del modelo SVM se puede considerar este modelo como el más adecuado para el estudio de predicciones de cancelaciones. En contraposición con el resto de los modelos. Los bosques aleatorios y vecinos más cercanos (KNN) presentan un buen funcionamiento en la mayoría de los problemas de clasificación que involucran grandes bases de datos.

Es importante desarrollar algoritmos eficientes para la predicción de cancelaciones de reservas en los hoteles. Con el estudio realizado se ha confirmado que los modelos basados en técnicas de machine learning son viables para el estudio de predicciones de cancelaciones y pueden aportar una vía de investigación muy potente e importante. Estos métodos facilitan la identificación de nuevos objetivos en el tema de reservas canceladas en los hoteles, sin los enormes costes que conlleva identificar de forma experimental.

9. TRABAJOS FUTUROS

En vista de que se puede seguir estudiando la aplicación de modelos de aprendizaje automático para predecir cancelaciones de reservas, el estudio puede escalarse a la aplicación de modelos de Aprendizaje Profundo (Deep learning), como las redes neuronales, que tienen un alto grado de aplicación para examinar problemas tanto de regresión como de clasificación. Asimismo, se pueden establecer estudios comparativos entre los modelos de pronósticos tradicionales, los modelos ARIMA, los modelos de aprendizaje automático y el Modelo de Aprendizaje Profundo (Deep learning), en donde se puedan colocar a prueba y evaluar el compartimento de cada uno de ellos.

10. BIBLIOGRAFIA

AFRIANTO, M. (2020). BOOKING PREDICTION MODELS FOR PEER-TO-PEER ACCOMMODATION LISTINGS USING LOGISTICS REGRESSION, DECISION TREE, K-NEAREST NEIGHBOR, AND RANDOM FOREST CLASSIFIERS. *JOURNAL OF INFORMATION SYSTEMS ENGINEERING AND BUSINESS INTELLIGENCE*, 123. OBTENIDO DE HTTPS://E-JOURNAL.UNAIR.AC.ID/JISEBI/ARTICLE/VIEW/20098/12443

AGUEDA, E. (2008). ANÁLISIS DEL COMPORTAMIENTO DE LA DEMANDA HOTELERA Y SU POSIBLE INTERFERENCIA POR EL SISTEMA DE. *ÁREA DE COMERCIALIZACIÓN E INVESTIGACIÓN DE MERCADOS.* UNIVERSIDAD COMPLUTENSE DE MADRID, MADRID.

ANTONIO, N., & DE ALMEIDA, A. (2019). UN SISTEMA AUTOMATIZADO DE SOPORTE DE DECISIONES BASADO EN APRENDIZAJE AUTOMÁTICO PARA PREDECIR CANCELACIONES DE RESERVAS DE HOTELES. *DATA SCIENCE JOURNAL*.

ARIAS, E. (2020). DESARROLLO DE UN MODELO PREDICTIVO CON INTELIGENCIA ARTIFICIAL PARA ESTABLECER CLASIFICACIÓN. *TESIS DE INVESTIGACIÓN PRESENTADA COMO REQUISITO PARCIAL PARA OPTAR AL TITULO DE BIOINGENIERO.* UNIVERSIDAD DE ANTIOQUIA, MEDELLIN, COLOMBIA. RECUPERADO EL 26 DE 03 DE 2022, DE HTTP://BIBLIOTECADIGITAL.UDEA.EDU.CO/BITSTREAM/10495/15251/1/ARIASERIKA_2020_DESARROLLOMODELOPREDICTIVO.PDF

BAGNATO, J. I. (2018). APRENDE MACHINE LEARNING. RECUPERADO EL 7 DE 05 DE 2022, DE HTTPS://WWW.APRENDEMACHINELEARNING.COM/CLASIFICAR-CON-K-NEAREST-NEIGHBOR-EJEMPLO-EN-PYTHON/#:~:TEXT=%C2%BFQU%C3%A9%20ES%20EL%20ALGORITMO%20K,DE%20DATOS%20QUE%20LE%20RODEAN.

BARRIO, J. (26 DE 07 DE 2019). *HEALTH BIG DATA*. OBTENIDO DE HTTPS://WWW.JUANBARRIOS.COM/LA-MATRIZ-DE-CONFUSION-Y-SUS-METRICAS/

BATTA, M. (2018). MACHINE LEARNING ALGORITHMS - A REVIEW. *INTERNATIONAL JOURNAL OF SCIENCE AND RESEARCH (IJSR)*, 381-382.

BEKKAR, M. &. (2017). EVALUATION MEASURES FOR MODELS ASSESSMENT OVER IMBALANCED DATA SETS . *JOURNAL OF INFORMATION ENGINEERING AND APPLICATIONS*, 27-38.

BHATTACHARYYA, I. (2018). SMOTE Y ADASYN (MANEJO DE CONJUNTOS DE DATOS DESEQUILIBRADOS).

BORJA, R. (2020). ESTANDARIZACIÓN DE MÉTRICAS DE RENDIMIENTO PARA CLASIFICADORES MACHINE Y DEEP LEARNING. *RISTI*, 187-188.

CABRERA, F. (2014). MODELO BASADO EN MACHINE LEARNING PARA LA PREDICCION DE LA DEMANDA DE HABITACION Y OCUPACION EN EL SECTOR HOTELERO. *(TESIS DE GRADO PARA OPTAR EL TÍTULO DE MAGISTER EN INGENIERÍA).* UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR, CARTAGENA.

CANALIS, X. (29 DE ABRIL DE 2019). OVERBOOKINH: LOS HOTELES YA PUEDEN COPIAR A LAS AEROLINEAS. *HOSTELTUR*.

CARDENAS, A. (2019). *CLASIFICACIÓN DE ACEPTACIÓN DE CAMPAÑAS PARA UNA ENTIDAD FINANCIERA, USANDO RANDOM FOREST CON DATOS BALANCEADOS Y DATOS NO BALANCEADOS.* RICARDO PALMA, LIMA, PERU. RECUPERADO EL 28 DE 03 DE 28, DE HTTP://REPOSITORIO.URP.EDU.PE/BITSTREAM/HANDLE/URP/2307/T030_47199993_M%20%20%20CARDENAS%20GARRO%20JOS%C3%89%20ANTONIO.PDF?SEQUENCE=1&ISALLOWED=Y

CHAWLA, N., BOWYER, K., & KEGELMEYER, P. (2002). SMOTE: SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE. RECUPERADO EL 28 DE 03 DE 2002, DE HTTPS://WWW.JAIR.ORG/INDEX.PHP/JAIR/ARTICLE/VIEW/10302/24590

DRZEWIECKI, W. (2017). THOROUGH STATISTICAL COMPARISON OF MACHINE LEARNING REGRESSION MODELS AND THEIR ENSEMBLES FOR . *GEODESY AND CARTOGRAPHY*, 171-209.

EID, A. (2020). APLICACIÓN DE MACHINE LEARNING EN LA INDUSTRIA HOTELERA: UNA REVISIÓN CRÍTICA. *ARTICULO.* UNIVERSIDAD DE HAIL, GRANIZO, ARABIA SAUDITA.

EMBAREC, R. (2020). APRENDIZAJE AUTOMÁTICO AAPLICADO AL SECTOR HOTELERO. *TRABAJO DE GRADO PARA OPTAR AL TITULO DE INGENIERIA INFORMATICA.* UNIVERSIDAD DE LA LAGUNA, SAN CRISTOBAL DE LA LAGUNA.

ENRIQUE, F. (14 DE 05 DE 2003). *LABORATORIO DE SISTEMAS INTELIGENTES*. (A. D. BAYESIANOS, PRODUCTOR) RECUPERADO EL 26 DE 03 DE 2022, DE AVAILABLE: WWW.FI.UBA.AR/LABORATORIOS/LSI.

FAWCETT, T. (2016). DATOS DESBALANCEADOS. *DATOS DESBALANCEADOS.* RECUPERADO EL 28 DE 03 DE 2022, DE HTTPS://WWW.SVDS.COM/LEARNING-IMBALANCED-CLASSES/

GARAY, N. &. (2008). A NEW APROACH FOR HOTEL ROOM REVENUE MAXIMIZATION USING ADVANCED FORECASTING AND OPTIMIZATION METHODS. *FACULTY OF COMPUTERS AND INFORMATION*.

GONZALEZ, L. (2019). *K VECINOS MÁS CERCANOS - TEORÍA | #39 CURSO MACHINE LEARNING CON PYTHON*. OBTENIDO DE HTTPS://APRENDEIA.COM/K-VECINOS-MAS-CERCANOS-TEORIA-MACHINE-LEARNING/#:~:TEXT=K%20VECINOS%20M%C3%A1S%20CERCANOS%20ES,Y%20LA%20DETECCI%C3%B3N%20DE%20INTRUSOS.

JIMENEZ, J. &. (2006). LA CAPACIDAD PREDICTIVA EN LOS METODOS BOX-JENKINS Y HOLT WINTERS: UNA APLICACION AL SECTOR TURISTICO. *REVISTA EUROPEA DE DIRECCION Y ECONOMIA DE LA EMPRESA*, 185-198.

KAPLAN, A. (2018). SIRI, SIRI, IN MY HAND: WHO'S THE FAIREST IN THE LAND? ON THE INTERPRETATIONS, ILLUSTRATIONS, AND IMPLICATIONS OF ARTIFICIAL INTELLIGENCE. *BUSINESS HORIZONS* , 15-25.

LEE, C. K. (1990). FORECASTING HOTEL OCCUPANCY RATES WITH TIME SERIES MODEL: AN EMPIRICAL ANALYSIS. *REVISTA DE INVESTIGACION EN HOTELERIA Y TURISMO* , 173-181.

MALAGON, L. (14 DE 05 DE 2003). OBTENIDO DE HTTPS://WWW.NEBRIJA.ES/~CMALAGON/INCO/APUNTES/BAYESIAN_LEARNING.PDF

MISHRA, A. (24 DE 02 DE 2018). *METRICS TI EVALUATE YOUR MACHINE LEARNING ALGORITHM*. RECUPERADO EL 26 DE 03 DE 2022, DE HTTPS://TOWARDSDATASCIENCE.COM/METRICS-TO-EVALUATE-YOUR-MACHINE-LEARNING-ALGORITHM-F10BA6E38234

MONROY, J. (2016). *TÉCNICAS DE MUESTREO PARA MEJORAR EL RENDIMIENTO DEL ALGORITMO BACKPROPAGATION EN PROBLEMAS DE DESBALANCE DE CLASES: UN ESTUDIO.*

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO, ATLACOMULCO, MEXICO.

MUSTAFA, E. (2019). WHAT IS ARTIFICIAL INTELLIGENCE? TECHNICAL CONSIDERATIONS AND FUTURE PERCEPTION. *ANATOLJCARDIOL.COM*, 5-6.

PALLARES, F. (2014). DESARROLLO DE UN MODELO BASADO EN MACHINE LEARNING PARA LA PREDICCIÓN DE LA DEMANDA DE. *MAESTRÍA EN INGENIERÍA.* TECNOLÓGICA DE BOLÍVAR, CARTAGENA.

PARRA, F. (25 DE ENERO DE 2019). *ESTADÍSTICA Y MACHINE LEARNING CON R*. OBTENIDO DE ESTADÍSTICA Y MACHINE LEARNING CON R: HTTPS://BOOKDOWN.ORG/CONTENT/2274/METODOS-DE-CLASIFICACION.HTML

PARRA, FRANCISCO. (25 DE 01 DE 2019). OBTENIDO DE BOOKDOWN.ORG: HTTPS://BOOKDOWN.ORG/CONTENT/2274/PORTADA.HTML

PINEDA, J. (2015). CÁLCULO DE LA TASA ÓPTIMA DE OVERBOOKING PARA UN HOTEL EN LA CIUDAD DE BOGOTA. *TESIS DE PREGRADO.* UNIVERSIDAD DE LOS ANDES, BOGOTA.

RAJOPADHYE, M. &. (2001). FORECASTING UNCERTAIN HOTEL ROOM DEMAND. *ELSEVIER SCIENCE INC.*, 1-11.

RENDI, S. (2021). PREDICTION OF HOTEL BOOKING CANCELLATION USING DEEP NEURAL NETWORK AND LOGISTIC REGRESSION ALGORITHM. *STMIK NUSA MANDIRI*.

RESCO, L. (2013). LA ESTRATEGIA DE OVERBOOKING EN LA INDUSTRIA HOTELERA: UN ANALISIS COMPARADO. *LIC. EN ECONOMIA.* UNIVERSIDAD DE SANANDRES, BUENOS ARES.

ROBERTO LÓPEZ BLANCO, M. M. (2021). *MANUAL SEN DE NUEVAS TECNOLOGÍAS EN TRASTORNOS DEL MOVIMIENTO.* MADRID: SEN.

RODRIGO, J. (10 DE 2017). *CIENCIADEDATOS.NET*. OBTENIDO DE ÁRBOLES DE DECISIÓN, RANDOM FOREST, GRADIENT BOOSTING Y C5.0: HTTPS://WWW.CIENCIADEDATOS.NET/DOCUMENTOS/33_ARBOLES_DE_PREDICCION_BAGGING_RANDOM_FOREST_BOOSTING

RODRIGO, J. A. (2017). SUPPORT VECTOR MACHINES, SVM . RECUPERADO EL 7 DE 05 DE 2022, DE HTTPS://WWW.CIENCIADEDATOS.NET/DOCUMENTOS/34_MAQUINAS_DE_VECTOR_SOPOR

RODRIGUEZ, E. (2014). VALORACIÓN PREANESTÉSICA IMPORTANCIA EN EL PACIENTE QUIRÚRGICO. *ANESTESIOLOGÍA*, 193-198.

ROUHIAINENL, P. (2018). *INTELIGENCIA ARTIFICIAL 101 COSAS QUE DEBES SABER HOY SOBRE NUESTRO FUTURO.* BARCELONA: ALIENTA. OBTENIDO DE HTTPS://STATIC0PLANETADELIBROSCOM.CDNSTATICS.COM/LIBROS_CONTENIDO_EXTRA/40/39308_INTELIGENCIA_ARTIFICIAL.PDF

SANCHEZ, A. J. (2020). USING MACHINE LEARNING AND BIG DATA FOR EFFICIENT FORECASTING OF HOTEL BOOKING CANCELLATIONS. *ELSEVIER LTD.* , 2-21.

SANCHEZ, A., & MIR, P. (28 DE 06 DE 2021). MANUAL SEN DE NUEVAS TECNOLOGÍAS EN TRASTORNOS DEL MOVIMIENTO. *SEN - SOCIEDAD ESPAÑOLA DE NEUROLOGIA*.

SITIOBIGDATA.COM. (19 DE 01 DE 2019). *MACHINE LEARNING: ARBOLES DE DECISION*. RECUPERADO EL 26 DE 03 DE 2022, DE HTTPS://SITIOBIGDATA.COM/2019/01/19/MACHINE-LEARNING-METRICA-CLASIFICACION-PARTE-3/

SOPORTE M, V. (2019). MÁQUINAS DE VECTORES DE SOPORTE. RECUPERADO EL 7 DE 05 DE 2022, DE HTTPS://WWW.CIENCIADEDATOS.NET/DOCUMENTOS/34_MAQUINAS_DE_VECTOR_SOPOR

VELASQUEZ, J. D. (2010). PREDICCION SERIES TEMPORALES USANDO MAQUINAS DE VECTORES DE SOPORTE . *INGENIERE. REVISTA CHILENA DE INGENIERIA* , 64-75.

11. ANEXO 1

Árbol de decisión obtenido para la selección de variables:

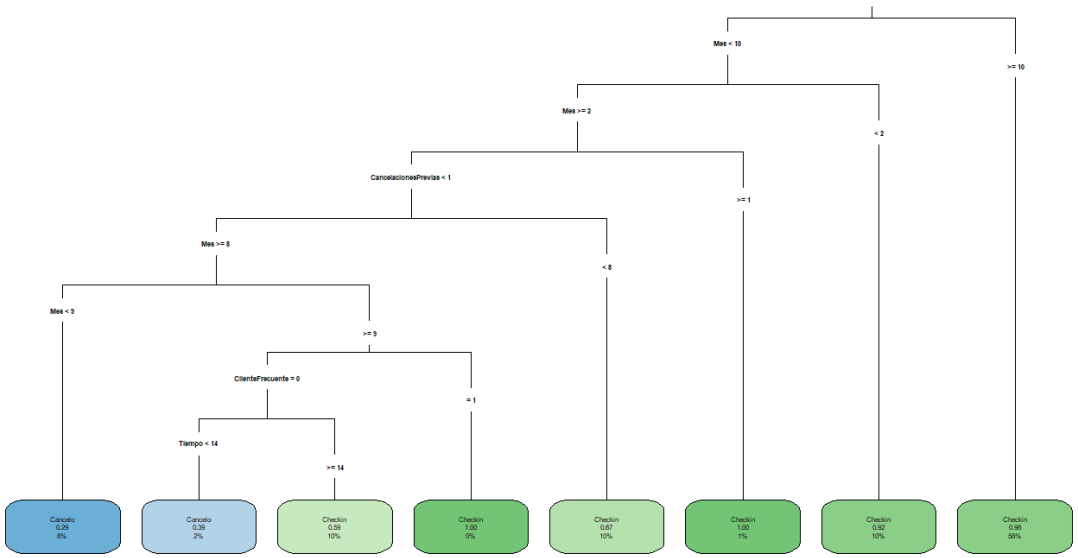

Ilustración 37. Árbol de decisión selección d variables. Fuente: R.