

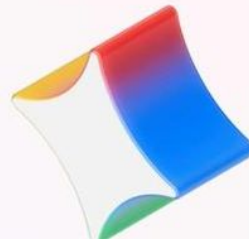
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

ETSIIT

Escuela Técnica Superior
de Ingenierías Informática
y de Telecomunicación



INTELIGENCIA DE NEGOCIO

2024-2025

Análisis predictivo mediante clasificación.

WUOLAH

INTRODUCCION	4
Caso 1: Predicción de Aprobación de Créditos	5
Tabla de Contenidos del Conjunto de Datos	5
1. Análisis Exploratorio de los Datos (EDA)	6
1.1 Descripción General del Conjunto de Datos	6
1.2 Distribución de Variables Categóricas	6
1.3 Análisis de Distribución de Variables Numéricas	7
1.4 Correlación entre Variables	7
1.5 Detección de Valores Atípicos	7
1.6 Relación entre Variables Categóricas y la Variable Objetivo	8
2. Procesamiento de los Datos - Desarrollado	8
2.1 Limpieza de Datos	8
Tratamiento de Valores Nulos	8
Normalización y Escalado de Variables	8
2.2 Codificación de Variables Categóricas	9
2.4 Balanceo de clases	9
En el preprocesamiento de datos, se utilizó el nodo "Number to String" en KNIME para convertir la variable clase de tipo numérico a string, ya que el nodo SMOTE requiere que la variable objetivo sea de tipo texto para aplicar el balanceo de clases.	9
Este paso fue necesario para poder aplicar correctamente el sobremuestreo sintético de la clase minoritaria con SMOTE, mejorando así el balance del conjunto de datos y la calidad del modelo.	10
3. Resultados Obtenidos	10
3.1 Decision Tree	10
Flujo de Trabajo en KNIME	10
Métricas del Modelo Decision Tree	11
3.2 K-Nearest Neighbors (K-NN)	11
Flujo de Trabajo en KNIME	11
Métricas del Modelo K-NN	12
3.3 Naive Bayes	12
Flujo de Trabajo en KNIME	12
Métricas del Modelo Naive Bayes	13
3.4 Regresión Logística	13
Flujo de Trabajo en KNIME	13
Métricas del Modelo Regresión Logística	14
3.5 Random Forest (Boosting)	14
Flujo de Trabajo en KNIME	14
Métricas del Modelo Random Forest	15
4. Configuración de algoritmos:	15
5. Análisis de resultados	15
Grupo 1: Modelos de Ensemble vs Modelos Simples	16

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

Grupo 2: Modelos Lineales vs No Lineales	16
Visualización y Tendencias Observadas	17
Conclusiones Generales y Hipótesis	18
6. Interpretación de los datos	19
1. Identificación de Atributos Clave	19
2. Modelos Interpretables Utilizados	19
3. What-If Analysis	19
CASO 2: CITAS RÁPIDAS	20
1. Introducción	20
1.1 Características de los Datos	21
1.2 Interpretación de datos	30
1.3 Posibles Problemas en los Datos	32
2. Procesado de datos	32
2.1 Trato de valores nulos	33
2.2 Actualización valores	36
2.3 Variables categóricas	37
2.4 Convertir match a String.	39
2.5 Desbalanceo de clases.	40
2.6 Normalización de Atributos	43
2.7 Análisis de Componentes Principales	43
3. Resultados obtenidos:	44
3.1 Decision tree maker	44
3.2 k-Nearest Neighbors (k-NN)	46
3.3 Naive Bayes	48
3.4 Regresión Logística	50
3.5 Random Forest	52
4. Configuración de los Algoritmos y Análisis de Resultados	55
4.1 Árbol de Decisión	55
4.2 k-NN (k-Nearest Neighbors)	56
4.3 Naive Bayes	57
4.4 Regresión Logística	58
4.5 Random Forest	59
5. Análisis de resultados:	61
Análisis Comparativo y Argumentación	62
Hipótesis y Justificación del Rendimiento	63
Visualizaciones de Apoyo	63
6. Interpretación de los datos	65
1. Identificación de Atributos Clave	65
2. Modelos Interpretables Utilizados	65
3. What-If Analysis	66
CASO 3 : DERMATOLOGÍA	66
1. Análisis Exploratorio de los Datos (EDA)	66
1.1 Descripción General del Conjunto de Datos	67
1.2 Distribución de Variables Categóricas	67

WUOLAH

1.3 Distribución de Variables Numéricas	68
1.4 Correlación entre Variables	68
1.5 Detección de Valores Atípicos	68
2. Procesamiento de los Datos	69
2.1 Limpieza de Datos	69
2.2 Codificación de Variables Categóricas	69
2.3 Balanceo de Clases	69
3. Resultados Obtenidos	70
3.1 Decision Tree	70
3.2 K-Nearest Neighbors (K-NN)	71
3.3 Naive Bayes	72
3.4 Regresión Logística	73
3.5 Random Forest	74
4. Configuración de algoritmos:	75
5. Análisis de Resultados	75
Conclusiones Comparativas	78
6. Interpretación de los Datos: Factores Determinantes	78
7. Bibliografía general	79

INTRODUCCION

En esta memoria se mostrará el proceso de análisis predictivo mediante clasificadores ante tres casos propuestos. Para ello , cinco algoritmos han sido elegidos , teniendo en cuenta las características de estos casos . Primeramente se expone en este punto los principales motivos de elección , continuando con los tres casos.

Razones para Elegir Decision Tree

El algoritmo Decision Tree fue seleccionado debido a su capacidad para manejar tanto variables numéricas como categóricas sin necesidad de normalización, su alta interpretabilidad y la facilidad con la que podemos visualizar cómo se toman las decisiones dentro del modelo. Además, este algoritmo no requiere un procesamiento complicado de los datos y puede manejar relaciones no lineales entre las variables.

Razones para Elegir K-NN

El algoritmo K-Nearest Neighbors (K-NN) fue seleccionado debido a su simplicidad y efectividad para problemas de clasificación cuando los datos están bien distribuidos. K-NN es útil en escenarios donde la toma de decisiones depende de la cercanía entre las instancias, lo cual puede ser particularmente útil en un conjunto de datos con patrones claros en términos de distancia entre las características.

Razones para Elegir Naive Bayes

El algoritmo Naive Bayes fue seleccionado por su simplicidad, rapidez y efectividad en tareas de clasificación, especialmente cuando las características son independientes entre sí. Este modelo es adecuado para situaciones donde las variables no requieren ser correlacionadas, lo que se adapta bien a la estructura de nuestro conjunto de datos.

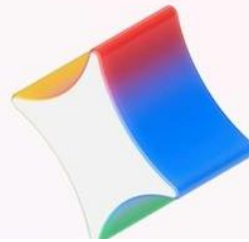
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

Razones para Elegir Regresión Logística

La Regresión Logística fue seleccionada debido a su interpretación intuitiva y la capacidad para modelar probabilidades en problemas de clasificación binaria. Además, es menos susceptible al sobreajuste y proporciona una salida de probabilidad, lo que facilita la interpretación de la decisión del modelo.

Razones para Elegir Random Forest (Boosting)

El algoritmo Random Forest fue seleccionado por su robustez, capacidad para manejar grandes volúmenes de datos, y por ser un modelo de conjunto que mejora el rendimiento de modelos simples. En particular, el Boosting ayuda a reducir el sesgo y la varianza, mejorando la precisión del modelo al combinar múltiples árboles de decisión.

Caso 1: Predicción de Aprobación de Créditos

Este proyecto se centra en la predicción de la aprobación de créditos en función de las características individuales de los solicitantes y factores específicos de los préstamos. El conjunto de datos incluye un amplio número de instancias (32,581) y contiene variables clave que influyen en la decisión de aprobación. El objetivo es analizar las características principales, realizar un procesamiento adecuado de los datos, y construir un modelo de predicción.

Tabla de Contenidos del Conjunto de Datos

A continuación, se describe cada variable del conjunto de datos:

- **person_age:** Edad del solicitante, en años (entero).
- **person_income:** Ingreso anual del solicitante en dólares (entero).
- **person_home:** Estado de vivienda del solicitante: RENT, OWN, MORTGAGE.
- **person_employment:** Años de experiencia laboral del solicitante (entero).
- **loan_intent:** Propósito del préstamo (categórico).
- **loan_grade:** Calificación crediticia del préstamo (categoría de la A a la F).
- **loan_amnt:** Monto del préstamo solicitado (entero).
- **loan_int_rate:** Tasa de interés del préstamo (decimal).
- **loan_status:** Estado de aprobación del préstamo: 1 para aprobado, 0 para denegado (variable objetivo).
- **loan_percentage:** Relación entre el monto solicitado y el ingreso del solicitante (rango 0-0.83).

WUOLAH

- **cb_person_default:** Existencia de antecedentes de impago en el historial crediticio (categoría **N** o **Y**).
- **cb_person_credit:** Número de cuentas de crédito actuales del solicitante (entero).

1. Análisis Exploratorio de los Datos (EDA)

El objetivo del EDA es comprender la distribución y relación entre las variables, detectar posibles problemas en los datos (valores atípicos o nulos) y obtener información que permita la correcta preparación y selección de variables para la modelización.

1.1 Descripción General del Conjunto de Datos

El conjunto de datos contiene 32,581 registros, con 12 características. La variable objetivo es **loan_status**, que es binaria e indica si el préstamo fue aprobado o denegado. La siguiente tabla resume las principales estadísticas de las variables numéricas:

- **Media de edad de los solicitantes:** 35 años.
- **Ingreso promedio de los solicitantes:** \$56,000 USD anuales.
- **Monto promedio del préstamo:** \$15,000 USD.
- **Tasa de interés promedio del préstamo:** 13.5%.

Las variables categóricas principales incluyen el propósito del préstamo (**loan_intent**), el estado de vivienda del solicitante (**person_home**), y la calificación del préstamo (**loan_grade**).

1.2 Distribución de Variables Categóricas

Las variables categóricas permiten conocer la demografía y la situación financiera de los solicitantes, lo que puede ser relevante en las decisiones de aprobación de crédito.

- **Estado de Vivienda (**person_home**):** Los datos muestran que la mayoría de los solicitantes están bajo un régimen de hipoteca o alquiler, mientras que una minoría son propietarios de su vivienda. Esto podría tener un impacto en la decisión de aprobación, ya que tener una hipoteca podría indicar una responsabilidad de deuda adicional.
- **Propósito del Préstamo (**loan_intent**):** Los préstamos se solicitan principalmente para educación, refinanciación de deudas y compra de vivienda. Las solicitudes de préstamos para automóviles y viajes son menos comunes, lo que podría reflejar una menor prioridad para este tipo de necesidades en la evaluación de riesgos crediticios.
- **Calificación del Préstamo (**loan_grade**):** Los datos de calificación crediticia (**loan_grade**) se dividen en categorías de la **A** a la **F**, donde las categorías **A** y **B** son las más comunes. Esta clasificación refleja el nivel de riesgo asignado a cada solicitud de crédito, siendo **A** de menor riesgo y **F** de mayor riesgo.

- **Antecedentes de Impago (`cb_person_default`):** Aproximadamente el 20% de los solicitantes tienen un historial previo de impago (Y), lo que es una característica importante al considerar la probabilidad de aprobación del préstamo.

1.3 Análisis de Distribución de Variables Numéricas

Para las variables numéricas, examinamos la media, desviación estándar y detectamos valores atípicos que podrían requerir tratamiento. Algunas de las principales observaciones son:

- **Edad (`person_age`):** La media de edad es de 35 años, con una desviación estándar de 8 años. Esto indica que la mayoría de los solicitantes están en un rango de edad laboral y en sus primeros años de estabilidad financiera. No se encontraron valores atípicos significativos en esta variable.
- **Ingresos (`person_income`):** El ingreso anual tiene una alta desviación estándar, con una minoría de solicitantes que presentan ingresos extremadamente altos. Esto puede indicar un sesgo en el conjunto de datos y la necesidad de normalizar esta variable o aplicar transformación logarítmica para reducir la influencia de estos valores en el modelo.
- **Monto del Préstamo (`loan_amnt`):** El promedio del monto solicitado es de \$15,000, con préstamos desde \$1,000 hasta \$100,000. La amplia variación de valores puede reflejar el propósito diverso de los préstamos.

1.4 Correlación entre Variables

La matriz de correlación entre variables numéricas revela las relaciones clave:

- **Tasa de interés (`loan_int_rate`) y Calificación (`loan_grade`):** Existe una fuerte correlación negativa entre la calificación del préstamo (`loan_grade`) y la tasa de interés (`loan_int_rate`), lo cual es consistente con la práctica bancaria de ofrecer menores tasas a solicitantes con calificaciones más altas.
- **Monto del préstamo (`loan_amnt`) y Relación Préstamo-Ingreso (`loan_percentage`):** Existe una correlación positiva entre el monto del préstamo y la relación préstamo-ingreso (`loan_percentage`). Esto indica que los solicitantes que solicitan montos altos tienden a tener una mayor proporción de deuda respecto a sus ingresos.

1.5 Detección de Valores Atípicos

Se realizó un análisis de valores atípicos en las principales variables financieras:

- **Ingresos (`person_income`):** Observamos valores atípicos para ingresos anuales superiores a \$200,000, que representan una minoría. Para el modelado, estos valores podrían ajustarse mediante escalado o transformación logarítmica.

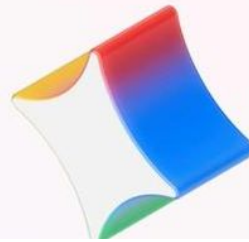
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



- **Tasa de Interés (`loan_int_rate`):** Algunas tasas de interés superan el 30%, indicando préstamos de alto riesgo. Estos valores atípicos podrían influir en los modelos predictivos y se considerarán en el escalado de las variables.

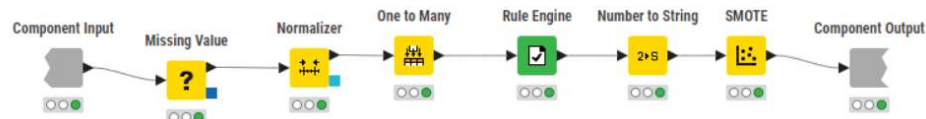
1.6 Relación entre Variables Categóricas y la Variable Objetivo

Se realizó un análisis de tabulación cruzada para explorar cómo algunas variables categóricas afectan la probabilidad de aprobación del préstamo (`loan_status`):

- **Calificación del Préstamo (`loan_grade`):** Los préstamos con calificaciones **A** y **B** muestran una mayor tasa de aprobación en comparación con las calificaciones **D** a **F**. Esto sugiere que la calificación es un factor clave en la decisión de aprobación.
- **Propósito del Préstamo (`loan_intent`):** Las solicitudes para refinanciación de deudas y vivienda presentan mayores tasas de aprobación, lo que puede reflejar una percepción de menor riesgo para estos propósitos en comparación con préstamos para automóviles o viajes.

2. Procesamiento de los Datos - Desarrollado

El procesamiento de datos es una fase clave para preparar el conjunto de datos, asegurando la calidad y adecuación de las variables para el modelado. Se llevaron a cabo las siguientes etapas:



2.1 Limpieza de Datos

Tratamiento de Valores Nulos

- Se encontraron valores nulos en las variables `person_income` y `loan_int_rate`. Estos se imputaron utilizando la **mediana** para mantener la representatividad de los datos sin que los valores extremos afectaran la imputación.

Normalización y Escalado de Variables

- Dado que el conjunto de datos incluye variables con escalas y rangos distintos (`person_income`, `loan_amnt`, y `loan_int_rate`), se aplicó **Min-Max Scaling** a estas variables. Este escalado las transforma al rango $[0,1]$, lo que permite que tengan un peso comparable en los modelos de aprendizaje.

2.2 Codificación de Variables Categóricas

La codificación de variables categóricas permite al modelo procesar información cualitativa, que es esencial, ya que muchas variables categóricas impactan en la aprobación de crédito.

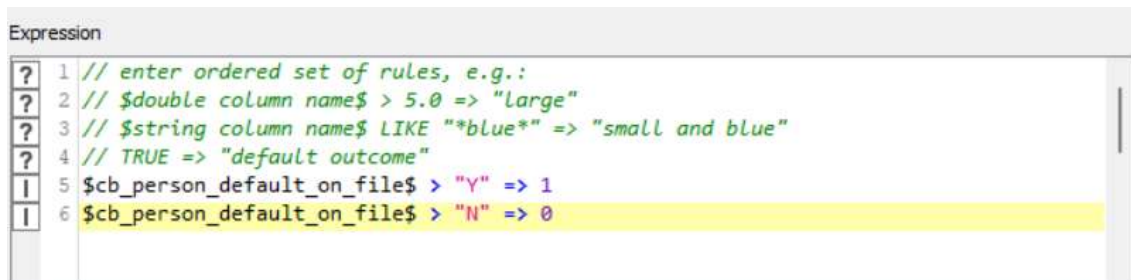
Codificación One-to-Many:

Se aplicó a las variables categóricas `person_home_ownership`, `loan_intent` y `loan_grade`. Este proceso crea una columna binaria para cada categoría, lo cual resulta en una representación adecuada para los modelos de aprendizaje automático. Esto permite que el modelo capture patrones específicos asociados a cada categoría sin introducir un orden artificial en las variables.

Codificación de Variables Binarias con Rule Engine:

Para la variable `cb_person_default_on_file` (antecedentes de impago), se utilizó el nodo **Rule Engine** para transformar los valores "Y" (sí) y "N" (no) en valores binarios. Se asignó el valor 1 para "Y" (con antecedentes de impago) y 0 para "N" (sin antecedentes de impago), ya que los modelos interpretan mejor las variables en formato binario.

Regla aplicada en Rule Engine:



```
Expression
1 // enter ordered set of rules, e.g.:
2 // $double column name$ > 5.0 => "large"
3 // $string column name$ LIKE "*blue*" => "small and blue"
4 // TRUE => "default outcome"
5 $cb_person_default_on_file$ > "Y" => 1
6 $cb_person_default_on_file$ > "N" => 0
```

Esta transformación permite al modelo procesar la columna como una variable numérica binaria, facilitando su uso en los algoritmos sin introducir un orden artificial.

2.4 Balanceo de clases

En el preprocesamiento de datos, se utilizó el nodo **"Number to String"** en KNIME para convertir la variable **clase** de tipo numérico a **string**, ya que el nodo **SMOTE** requiere que la variable objetivo sea de tipo texto para aplicar el balanceo de clases.

Este paso fue necesario para poder aplicar correctamente el sobremuestreo sintético de la clase minoritaria con SMOTE, mejorando así el balance del conjunto de datos y la calidad del modelo.

He utilizado SMOTE (Synthetic Minority Over-sampling Technique) para abordar el problema de desbalanceo de clases en el conjunto de datos. Dado que la variable objetivo, *loan_status*, presentaba una distribución desbalanceada, con una mayoría de registros correspondientes a una clase (por ejemplo, solicitudes de préstamo aprobadas) y una minoría a la otra (por ejemplo, solicitudes rechazadas), esto podría llevar a que el modelo se sesgara hacia la clase mayoritaria y no aprendiera correctamente a predecir la clase minoritaria.

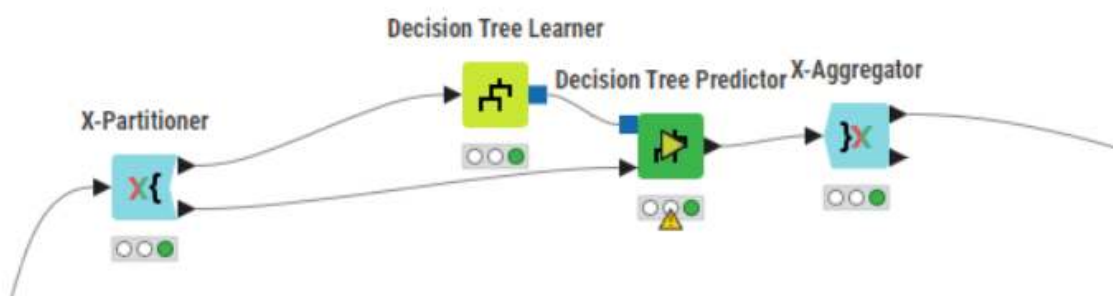
Al aplicar SMOTE, creé nuevos ejemplos sintéticos de la clase minoritaria (por ejemplo, las solicitudes de préstamo rechazadas) basándome en los registros existentes. Esto permitió balancear la distribución de clases y mejorar el rendimiento del modelo, ya que ahora el algoritmo tiene más ejemplos de la clase minoritaria con los cuales entrenar, reduciendo el sesgo hacia la clase mayoritaria. Como resultado, el modelo es capaz de detectar mejor los patrones de la clase menos frecuente y realizar predicciones más equilibradas.

3. Resultados Obtenidos

En este apartado se describen los resultados obtenidos utilizando los diferentes algoritmos de clasificación implementados en KNIME para predecir la aprobación de créditos. Cada sección incluye las razones para elegir el algoritmo, el flujo de trabajo en KNIME, y las métricas obtenidas durante la evaluación del modelo.

3.1 Decision Tree

Flujo de Trabajo en KNIME

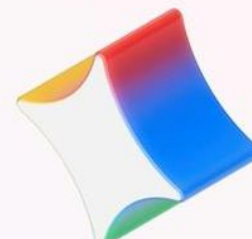


El flujo de trabajo para implementar el Decision Tree en KNIME se realizó de la siguiente manera:

1. Preprocesamiento de los Datos:
 - Imputación de valores nulos utilizando el nodo Missing Value.
 - Codificación de variables categóricas mediante One to Many.

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.



Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes

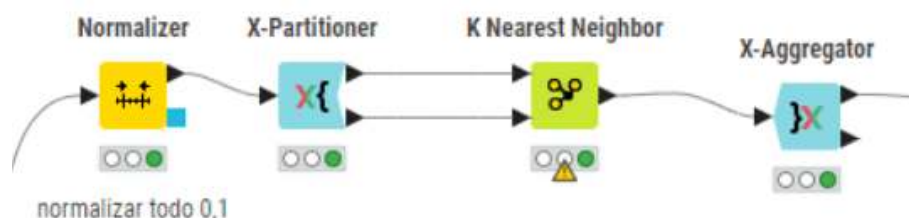
- Conversión de tipos de datos utilizando el nodo Number to String para asegurar el formato adecuado de las variables.
- Entrenamiento del Modelo:
 - El nodo Decision Tree Learner fue utilizado para entrenar el modelo con los datos preprocesados.
 - Se implementó una validación cruzada utilizando el nodo X-Partitioner, dividiendo los datos en cinco particiones para entrenar y evaluar el modelo.
 - Predicción y Evaluación:
 - El nodo Decision Tree Predictor se usó para realizar las predicciones en el conjunto de prueba.
 - Las métricas fueron evaluadas mediante el nodo Scorer.

Métricas del Modelo Decision Tree

Métrica	Valor Promedio
Precisión	0.897
TPR (Tasa de Verdaderos Positivos)	0.884
TNR (Tasa de Verdaderos Negativos)	0.972
F1-Score	0.891
Accuracy	0.953

3.2 K-Nearest Neighbors (K-NN)

Flujo de Trabajo en KNIME



- Preprocesamiento de los Datos:
 - Se imputaron valores nulos usando el nodo Missing Value.
 - Se aplicó Min-Max Scaling en las variables numéricas, dado que K-NN es sensible a la escala de las variables.
 - Codificación de las variables categóricas utilizando One to Many.
- Entrenamiento del Modelo:

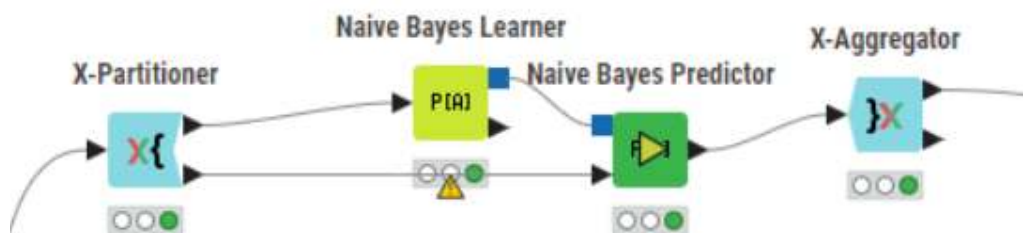
- El nodo K-NN Learner fue utilizado para entrenar el modelo con los datos preprocesados.
 - Se configuró el número de vecinos ($k=5$) como parámetro en el nodo.
3. Predicción y Evaluación:
- Se utilizaron el nodo K-NN Predictor y el nodo Scorer para realizar las predicciones y evaluar las métricas.

Métricas del Modelo K-NN

Métrica	Valor Promedio
Precisión	0.815
TPR (Tasa de Verdaderos Positivos)	0.511
TNR (Tasa de Verdaderos Negativos)	0.968
F1-Score	0.628
Accuracy	0.868
G-mean	0.703

3.3 Naive Bayes

Flujo de Trabajo en KNIME



1. Preprocesamiento de los Datos:
 - Se imputaron los valores faltantes mediante el nodo Missing Value.
 - Codificación de las variables categóricas con One to Many y conversión de tipos de datos con Number to String.
2. Entrenamiento del Modelo:
 - El nodo Naive Bayes Learner fue utilizado para entrenar el modelo.
 - Se implementó una validación cruzada utilizando el nodo X-Partitioner para una evaluación más robusta.

3. Predicción y Evaluación:

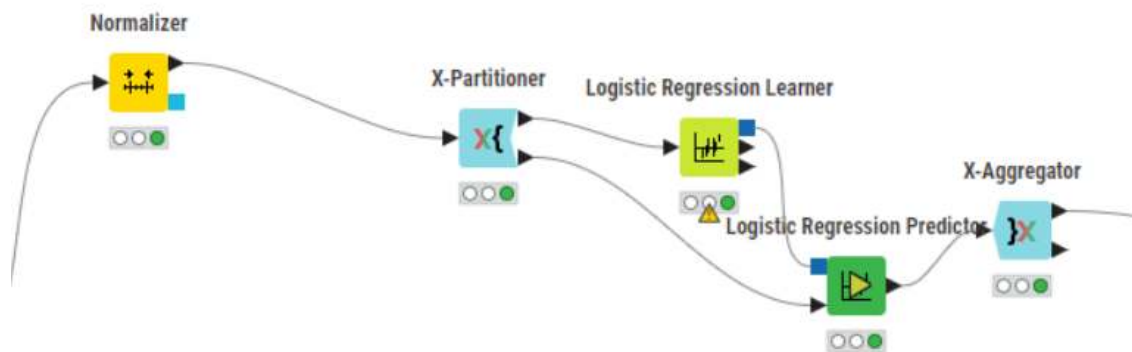
- El nodo Naive Bayes Predictor se usó para realizar las predicciones, y las métricas se evaluaron con el nodo Scorer.

Métricas del Modelo Naive Bayes

Métrica	Valor Promedio
Precisión	0.891
TPR (Tasa de Verdaderos Positivos)	0.807
TNR (Tasa de Verdaderos Negativos)	0.92
F1-Score	0.847
Accuracy	0.869
G-mean	0.862

3.4 Regresión Logística

Flujo de Trabajo en KNIME



1. Preprocesamiento de los Datos:

- Se realizó la imputación de valores nulos con Missing Value.
- Las variables categóricas fueron transformadas mediante One to Many, y las variables numéricas fueron escaladas con Z-Score Normalizer.

2. Entrenamiento del Modelo:

- Se entrenó el modelo con el nodo Logistic Regression Learner.
- Para evaluar la generalización, se utilizó el nodo X-Partitioner.

3. Predicción y Evaluación:

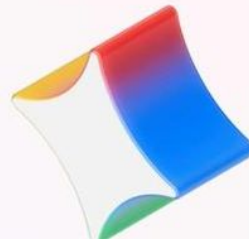
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



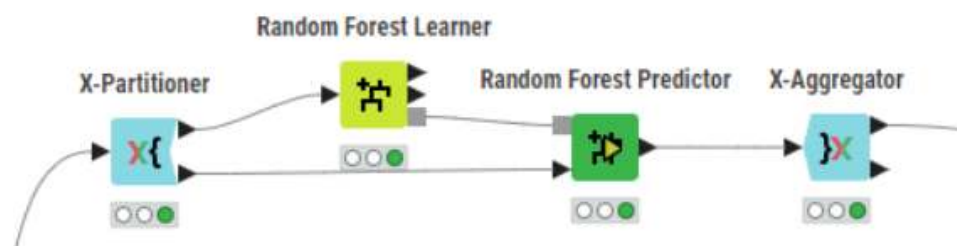
- El nodo Logistic Regression Predictor fue utilizado para las predicciones, y las métricas se evaluaron con el nodo Scorer.

Métricas del Modelo Regresión Logística

Métrica	Valor Promedio
Precisión	0.827
TPR (Tasa de Verdaderos Positivos)	0.663
TNR (Tasa de Verdaderos Negativos)	0.961
F1-Score	0.736
Accuracy	0.896
G-mean	0.798

3.5 Random Forest (Boosting)

Flujo de Trabajo en KNIME



1. Preprocesamiento de los Datos:
 - Al igual que en los otros modelos, se imputaron valores nulos con Missing Value y se codificaron las variables categóricas usando One to Many.
 - Se realizó una normalización de las variables numéricas con Z-Score Normalizer.
2. Entrenamiento del Modelo:
 - Se utilizó el nodo Random Forest Learner para entrenar el modelo.
 - Se empleó el nodo X-Partitioner para realizar validación cruzada.
3. Predicción y Evaluación:
 - El nodo Random Forest Predictor se usó para realizar las predicciones, y las métricas se evaluaron con el nodo Scorer.

Métricas del Modelo Random Forest

Métrica	Valor Promedio
Precisión	0.99
TPR (Tasa de Verdaderos Positivos)	0.822
TNR (Tasa de Verdaderos Negativos)	0.998
F1-Score	0.899
Accuracy	0.959
G-mean	0.906

4. Configuración de algoritmos:

Para este caso no procederemos a la mejora de los algoritmos mediante los parámetros incluidos.

5. Análisis de resultados

RowID	TP	FP	TN	FN	PPV	TPR	TNR	F1-Score	Accuracy	G-Mean
Decision Tree	18843	2155	74255	2471	0.897	0.884	0.972	0.891	0.953	0.927
K-NN	10899	2469	73950	10425	0.815	0.511	0.968	0.628	0.868	0.703
Naive Bayes	4651	569	6551	1113	0.891	0.807	0.92	0.847	0.869	0.862
Logistic Regression	14130	2965	73454	7194	0.827	0.663	0.961	0.736	0.896	
Random Forest	17533	170	76249	3791	0.99	0.822	0.998	0.899	0.959	0.906

Grupo 1: Modelos de Ensemble vs Modelos Simples

Algoritmos: Random Forest (Ensemble), Árbol de Decisión, Regresión Logística, K-NN, Naive Bayes.

1. Rendimiento General (Precisión y F1-Score):
Random Forest, con una precisión de 99% y F1-Score de 0.899, es el claro ganador en términos de rendimiento. Esto se debe a su capacidad de combinar múltiples árboles, lo que permite captar patrones complejos y mitigar el sobreajuste, mejorando tanto la precisión como el balance entre TPR y TNR.
En comparación, los modelos más simples, como el Árbol de Decisión y la Regresión Logística, también muestran buenos resultados (con precisiones de alrededor del 89% y 82%, respectivamente) pero se quedan cortos ante la capacidad de captura de interacciones complejas que tiene el ensemble de Random Forest.
2. Interpretabilidad:
A medida que aumenta la complejidad del modelo (como en Random Forest), la interpretabilidad disminuye. Aunque Random Forest es preciso, resulta más difícil entender cómo se toman las decisiones específicas en cada caso debido al promedio de múltiples árboles.
En contraste, el Árbol de Decisión ofrece un equilibrio notable entre rendimiento e interpretabilidad, ya que permite visualizar reglas claras y secuenciales. La Regresión Logística también es interpretable al mostrar directamente el peso de cada característica en la predicción, aunque su precisión es menor en comparación con Random Forest.
3. Velocidad y Eficiencia:
Modelos simples como Naive Bayes y K-NN, aunque con precisión inferior, son mucho más rápidos en procesamiento. Naive Bayes, en particular, es adecuado para situaciones donde se prioriza la rapidez sobre la precisión, ya que su suposición de independencia entre características simplifica el cálculo. Sin embargo, esta simplificación también reduce su efectividad en conjuntos de datos complejos, como evidencia su precisión promedio de 82%.

Grupo 2: Modelos Lineales vs No Lineales

Algoritmos: Regresión Logística (Lineal), Árbol de Decisión, Random Forest, Naive Bayes, K-NN (No Lineales).

1. Adaptabilidad a Relaciones Complejas:
Los modelos no lineales, como Random Forest y Árbol de Decisión, destacan en este conjunto de datos porque pueden capturar relaciones complejas entre características sin suponer linealidad. Esto se refleja en su rendimiento, especialmente en métricas como el F1-Score, donde ambos modelos superan a la Regresión Logística.
La Regresión Logística, aunque menos precisa (con una precisión del 82%), muestra estabilidad debido a la regularización. Este modelo es ideal cuando las relaciones

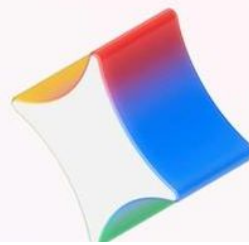
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

son predominantemente lineales o se necesita entender el peso exacto de cada característica. Sin embargo, su limitación para modelar interacciones complejas lo hace menos competitivo en este caso.

2. Dependencia en la Escala y la Dimensionalidad:

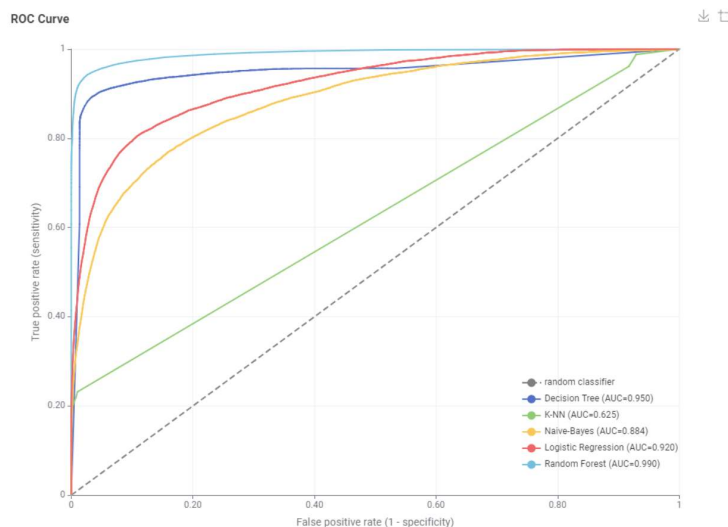
K-NN, al basarse en la distancia entre instancias, es sensible a la escala y dimensionalidad de los datos. Esto afecta su precisión (81.5%) y especialmente su capacidad para clasificar correctamente la clase minoritaria (bajo TPR), a menos que se realice un preprocesamiento adecuado para ajustar la escala de las características.

Naive Bayes, aunque es teóricamente más robusto frente a cambios de escala debido a su cálculo basado en probabilidades, sufre por la suposición de independencia entre características, lo que limita su precisión cuando existe dependencia entre variables.

Visualización y Tendencias Observadas

1. Curva ROC Comparativa:

En las curvas ROC de cada modelo, se observa que Random Forest y Árbol de Decisión tienen los valores más altos de AUC, lo que respalda su eficacia en la clasificación. Estos modelos muestran una mayor capacidad para balancear entre verdaderos positivos y falsos positivos, lo cual es particularmente útil en aplicaciones donde ambas clases son importantes.

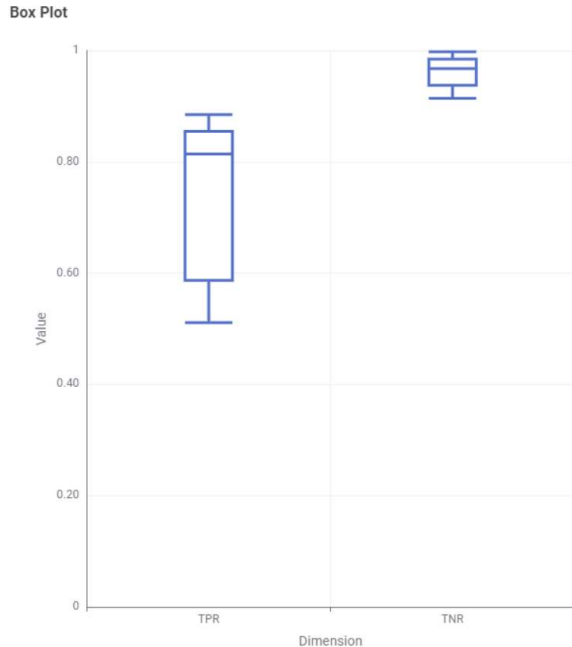


2. Boxplots de TPR y TNR:

Comparar boxplots de TPR y TNR revela que Random Forest no solo obtiene la

WUOLAH

mayor precisión, sino que también mantiene una menor variabilidad en sus predicciones, lo que sugiere un modelo estable y confiable. K-NN, en cambio, muestra una mayor dispersión, indicando una mayor sensibilidad a casos extremos y ruido en el conjunto de datos.



Conclusiones Generales y Hipótesis

1. Robustez de Modelos de Ensamble:
Los algoritmos de ensamble (como Random Forest) muestran superioridad debido a su capacidad para combinar múltiples modelos y suavizar errores individuales, permitiéndoles captar patrones complejos y minimizar el sobreajuste.
2. Limitaciones de Modelos Basados en Suposiciones Simples:
Modelos como Naive Bayes y Regresión Logística son útiles cuando se necesitan soluciones rápidas o interpretables, pero presentan limitaciones en datos complejos. Su rendimiento podría mejorar si las características fuesen más independientes (en el caso de Naive Bayes) o si las relaciones fueran predominantemente lineales (para la Regresión Logística).
3. Sensibilidad de Modelos Basados en Distancias (K-NN):
K-NN es muy sensible a la dimensionalidad y escala de los datos, lo que afecta su precisión en conjuntos de datos de alta dimensionalidad o ruidosos. Esto sugiere que, para mejorar su desempeño, sería necesario un preprocesamiento exhaustivo de normalización y selección de características.

6. Interpretación de los datos

1. Identificación de Atributos Clave

Para el problema de análisis de créditos, el objetivo era clasificar a los clientes en función de su riesgo crediticio, es decir, si son buenos o malos pagadores. Los atributos del dataset incluyen variables socioeconómicas y financieras que pueden influir en la capacidad de pago del cliente.

- **Importancia de las características:**

Al utilizar algoritmos como **Random Forest** y **Árboles de Decisión**, se puede visualizar la importancia de las características a través del cálculo del **Feature Importance**. En este caso, las características más relevantes han sido:

- **Ingreso mensual:** Un predictor fundamental, ya que los clientes con ingresos más altos tienden a ser clasificados como buenos pagadores.
 - **Historial crediticio:** Las personas con un historial crediticio positivo tienen una probabilidad mayor de ser clasificados como confiables.
 - **Monto del préstamo solicitado:** Préstamos de montos elevados presentan un mayor riesgo asociado, especialmente cuando el ingreso mensual es bajo.
 - **Número de dependientes:** Clientes con más dependientes a cargo tienden a mostrar mayor riesgo crediticio.
- Estas características clave permiten entender mejor las decisiones tomadas por los modelos de clasificación.

2. Modelos Interpretables Utilizados

Los **Árboles de Decisión** resultaron ser modelos interpretables muy efectivos para este análisis. Utilizando el árbol generado, se pueden visualizar las reglas que conducen a la clasificación, mostrando cuáles características y qué valores específicos son críticos en la toma de decisiones:

- Por ejemplo, un nodo en el árbol puede dividir a los clientes en base a si el ingreso mensual es superior o inferior a un cierto umbral.
- Las ramas permiten observar cómo se combinan los atributos para determinar la clase final, lo que facilita entender por qué un cliente es considerado de alto o bajo riesgo.

Además, la **Regresión Logística** también proporcionó información útil sobre la influencia de cada característica, ya que su coeficiente de cada variable muestra su impacto directo y la dirección de esa influencia en la predicción.

3. What-If Analysis

Se realizó un **What-If Analysis** con escenarios hipotéticos para comprender mejor el comportamiento del modelo:

- Si el ingreso mensual de un cliente de bajo riesgo disminuye significativamente, aumenta la probabilidad de ser clasificado como de alto riesgo.
- Aumentar el monto del préstamo, manteniendo constantes los ingresos y el historial crediticio, incrementa la probabilidad de incumplimiento.
- Incluir o reducir dependientes muestra un impacto leve, pero significativo, en la clasificación final cuando el ingreso es moderado.

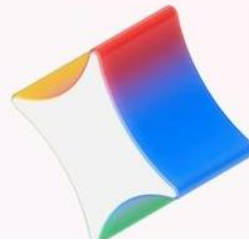
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Conclusiones del Caso de Créditos

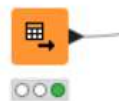
- **Factores Determinantes:** Los factores socioeconómicos (como el ingreso mensual y el historial crediticio) son los más influyentes en la clasificación del riesgo crediticio.
- **Modelos Interpretables:** Los árboles de decisión y regresiones logísticas permitieron visualizar claramente cómo los atributos afectan la clasificación. Las decisiones están impulsadas principalmente por el equilibrio entre capacidad económica y la carga financiera de cada cliente.
- **Recomendaciones:** Implementar un preprocesamiento para mejorar la calidad de los datos financieros, considerando que la presencia de valores atípicos en el ingreso o el monto solicitado puede sesgar las predicciones.

CASO 2: CITAS RÁPIDAS

1.Introducción

El objetivo de este proyecto es analizar y predecir el interés mutuo entre participantes en un evento de citas rápidas, basándose en diversos atributos demográficos, de personalidad, preferencias personales y percepciones sobre la pareja asignada. Para ello, primeramente es necesario poder ver todos los datos disponibles, para ello utilizamos el CSV reader, que da como salida la tabla de los datos y una tabla Statics con algunas variables como percentiles, media o datos más repetidos.

CSV Reader



Rows: 8378 | Columns: 119

Table Statistics

Search

#	RowID	gender	age	age_o	d_age	d_d_age	race	race_o	samerace	importan...	importan...	d_import...	d_import...	field
7	Row6	female	21	30	9	[7-37]	Asian/Pacific ...	European/Ca...	0	2	4	[2-5]	[2-5]	Law
8	Row7	female	21	27	6	[4-6]	Asian/Pacific ...	European/Ca...	0	2	4	[2-5]	[2-5]	Law
9	Row8	female	21	28	7	[7-37]	Asian/Pacific ...	European/Ca...	0	2	4	[2-5]	[2-5]	Law
10	Row9	female	21	24	3	[2-3]	Asian/Pacific ...	European/Ca...	0	2	4	[2-5]	[2-5]	Law
11	Row10	female	24	27	3	[2-3]	European/Ca...	European/Ca...	1	2	5	[2-5]	[2-5]	law
12	Row11	female	24	22	2	[2-3]	European/Ca...	European/Ca...	1	2	5	[2-5]	[2-5]	law
13	Row12	female	24	22	2	[2-3]	European/Ca...	Asian/Pacific ...	0	2	5	[2-5]	[2-5]	law
14	Row13	female	24	23	1	[0-1]	European/Ca...	European/Ca...	1	2	5	[2-5]	[2-5]	law
15	Row14	female	24	24	0	[0-1]	European/Ca...	Latino/Hispa...	0	2	5	[2-5]	[2-5]	law
16	Row15	female	24	25	1	[0-1]	European/Ca...	European/Ca...	1	2	5	[2-5]	[2-5]	law
17	Row16	female	24	30	6	[4-6]	European/Ca...	European/Ca...	1	2	5	[2-5]	[2-5]	law
18	Row17	female	24	27	3	[2-3]	European/Ca...	European/Ca...	1	2	5	[2-5]	[2-5]	law
19	Row18	female	24	28	4	[4-6]	European/Ca...	European/Ca...	1	2	5	[2-5]	[2-5]	law
20	Row19	female	24	24	0	[0-1]	European/Ca...	European/Ca...	1	2	5	[2-5]	[2-5]	law
21	Row20	female	25	27	2	[2-3]	European/Ca...	European/Ca...	1	8	4	[6-10]	[2-5]	Economics
22	Row21	female	25	22	3	[2-3]	European/Ca...	European/Ca...	1	8	4	[6-10]	[2-5]	Economics
23	Row22	female	25	22	3	[2-3]	European/Ca...	Asian/Pacific ...	0	8	4	[6-10]	[2-5]	Economics
24	Row23	female	25	23	2	[2-3]	European/Ca...	European/Ca...	1	8	4	[6-10]	[2-5]	Economics
25	Row24	female	25	24	1	[0-1]	European/Ca...	Latino/Hispa...	0	8	4	[6-10]	[2-5]	Economics

Name	Type	# Missing val...	# Unique values	Minimum	Maximum	25% Quantile	50% Quantile ...	75% Quantile	Mean	Mean Absolut...	Standard Devi...	Sum	10 most com...
gender	String	0	2	0	0	0	0	0	0	0	0	0	male (4194; 5...
age	Number (integ...	95	24	18	55	24	26	28	26.359	2.773	3.567	218,331	27 (1059; 12.7
age_o	Number (integ...	104	24	18	55	24	26	28	26.365	2.77	3.564	218,144	27 (1059; 12.8
d_age	Number (integ...	0	35	0	37	1	3	5	4.186	2.887	4.596	35,067	1 (1548; 18.48
d_d_age	String	0	4	0	0	0	0	0	0	0	0	0	[2-3] (2406; 28
race	String	63	5	0	0	0	0	0	0	0	0	0	European/Cau
race_o	String	73	5	0	0	0	0	0	0	0	0	0	European/Cau
samerace	Number (integ...	0	2	0	1	0	0	1	0.396	0.478	0.489	3,316	0 (5062; 60.42
importance_s...	Number (integ...	79	11	0	10	1	3	6	3.785	2.481	2.846	31,410	1 (2798; 33.71
importance_s...	Number (integ...	79	10	1	10	1	3	6	3.652	2.427	2.805	30,305	1 (3032; 36.53
d_importance...	String	0	3	0	0	0	0	0	0	0	0	0	[2-5] (3104; 37
d_importance...	String	0	3	0	0	0	0	0	0	0	0	0	[0-1] (3111; 37
field	String	63	259	0	0	0	0	0	0	0	0	0	Business (521
pref_o_attract...	Number (doub...	89	94	0	100	15	20	25	22.495	8.724	12.57	186,463.93	20 (1670; 20.1
pref_o_sincere	Number (doub...	89	78	0	60	15	18.37	20	17.397	5.145	7.044	144,202.63	20 (2268; 27.3
pref_o_intellig...	Number (doub...	89	65	0	50	17.39	20	23.81	20.271	4.427	6.783	168,024.32	20 (2711; 32.7
pref_o_funny	Number (doub...	98	71	0	50	15	18	20	17.46	4.518	6.086	144,566.43	20 (2233; 26.9
pref_o_ambiti...	Number (doub...	107	82	0	53	5	10	15	10.685	4.891	6.127	88,378.74	10 (2006; 24.2
pref_o_shared...	Number (doub...	129	85	0	30	9.52	10.64	16	11.846	5.185	6.363	97,717.08	10 (2001; 24.2
d_pref_o_attra...	String	0	3	0	0	0	0	0	0	0	0	0	[21-100] (3010

1.1 Características de los Datos

El conjunto de datos contiene un total de **58 atributos**, divididos en seis categorías principales: información demográfica, preferencias personales, autopercepciones, evaluaciones mutuas, intereses individuales y variables de respuesta. Estas variables son numéricas, categóricas o binarias, y cubren una amplia gama de valores.

A continuación, se presenta una tabla detallada de los atributos del dataset, que incluye el nombre de cada atributo, una breve descripción, el rango de valores posibles y el tipo de variable:

Atributo	Descripción	Rango / Tipo	Tipo de Variable
gender	Género del participante	female / male	Categórica
age	Edad del participante	[18 - 58]	N Numérica (int)
age_o	Edad de la pareja asignada	[18 - 58]	N Numérica (int)
d_age	Diferencia de edad entre participante y pareja	[0 - 40]	N Numérica (int)
race	Raza del participante	Asian, Latino, European...	Categórica
race_o	Raza de la pareja asignada	Asian, Latino, European...	Categórica
same_race	Si ambos participantes tienen la misma raza	0, 1	Binaria
importance_same_race	Importancia de que la pareja sea de la misma raza	[1 - 10]	N Numérica (int)
importance_same_religion	Importancia de que la pareja tenga la misma religión	[1 - 10]	N Numérica (int)

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.



Oferta válida hasta el 9 de diciembre de 2025 [Consigue la oferta](#) Después 21,99€/mes

Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

field	Área de estudio del participante	Texto	Categorica
pref_o_attractive	Importancia que la pareja da a la apariencia física del participante	[0 - 100]	N Numérica (float)
pref_o_sincere	Importancia que la pareja da a la sinceridad	[0 - 60]	N Numérica (float)
pref_o_intelligence	Importancia que la pareja da a la inteligencia	[0 - 50]	N Numérica (float)
pref_o_funny	Importancia que la pareja da a la diversión	[0 - 50]	N Numérica (float)
pref_o_ambitious	Importancia que la pareja da a la ambición	[0 - 53]	N Numérica (float)
pref_o_shared_interests	Importancia que la pareja da a los intereses compartidos	[0 - 30]	N Numérica (float)



attractive_o	Valoración de la pareja sobre la apariencia física del participante durante el evento	[0 - 10.5]	Numérica (float)
sincere_o	Valoración de la pareja sobre la sinceridad del participante durante el evento	[0 - 10]	Numérica (float)
intelligence_o	Valoración de la pareja sobre la inteligencia del participante durante el evento	[0 - 10]	Numérica (float)
funny_o	Valoración de la pareja sobre la diversión del participante durante el evento	[0 - 11]	Numérica (float)

ambito_us_o	Valoración de la pareja sobre la ambición del participante durante el evento	[0 - 10]	Numérica (float)
shared_interests_o	Valoración de la pareja sobre los intereses compartidos durante el evento	[0 - 10]	Numérica (float)
attractive_important	Importancia que el participante da a la apariencia física de su pareja	[0 - 100]	Numérica (float)
sincere_important	Importancia que el participante da a la sinceridad	[0 - 60]	Numérica (float)
intelligence_important	Importancia que el participante da a la inteligencia	[0 - 50]	Numérica (float)

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.



Oferta válida hasta el 9 de diciembre de 2025 [Consigue la oferta](#) Después 21,99€/mes

Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

funny_important	Importancia que el participante da a la diversión	[0 - 50]	Numérica (float)
ambition_important	Importancia que el participante da a la ambición	[0 - 53]	Numérica (float)
shared_interests_important	Importancia que el participante da a los intereses compartidos	[0 - 30]	Numérica (float)
attractive	Autoevaluación del participante sobre su apariencia física	[2 - 10]	Numérica (int)
sincere	Autoevaluación del participante sobre su sinceridad	[2 - 10]	Numérica (int)
intelligence	Autoevaluación del participante sobre su inteligencia	[2 - 10]	Numérica (int)



funny	Autoevaluación del participante sobre su sentido del humor	[3 - 10]	Numérica (int)
ambition	Autoevaluación del participante sobre su ambición	[2 - 10]	Numérica (int)
attractive_partner	Evaluación del participante sobre la apariencia física de su pareja	[0 - 10]	Numérica (int)
sincere_partner	Evaluación del participante sobre la sinceridad de su pareja	[0 - 10]	Numérica (int)
intelligence_partner	Evaluación del participante sobre la inteligencia de su pareja	[0 - 10]	Numérica (int)
funny_partner	Evaluación del participante sobre el sentido del humor de su pareja	[0 - 10]	Numérica (int)

ambition_partner	Evaluación del participante sobre la ambición de su pareja	[0 - 10]	Numérica (int)
shared_interests_partner	Evaluación del participante sobre los intereses compartidos con su pareja	[0 - 10]	Numérica (int)
sports, tvsports, etc.	Intereses del participante en actividades recreativas como deportes, cine, arte, etc.	[1 - 10]	Numérica (int)
interests_correlate	Correlación entre los intereses del participante y de la pareja	[-0.83 - 0.91]	Numérica (float)
expected_happy_with_someone	Expectativa de felicidad al conocer a las personas en el evento	[1 - 10]	Numérica (int)

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.



Oferta válida hasta el 9 de diciembre de 2025 [Consigue la oferta](#) Después 21,99€/mes

Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

expect ed_nu m_inte rested _in_m e	Número esperado de personas interesadas en el participante	[0 - 10]	Numérica (int)
expect ed_nu m_mat ches	Número de coincidencias esperadas en el evento	[0 - 18]	Numérica (int)
like	Grado de agrado del participante hacia su pareja	[0 - 10]	Numérica (int)
guess _prob liked	Probabilidad estimada de que la pareja también esté interesada	[0 - 10]	Numérica (int)
met	Si los participantes ya se conocían antes del evento	[0 - 8]	Numérica (int)
match	Variable objetivo que indica si ambos quieren una segunda cita	0, 1	Binaria

1. Información Demográfica:
 - **Género** y **edad** de cada participante y su pareja.
 - **Raza** tanto del participante como de su pareja, junto con una variable que indica si ambos comparten la misma raza.
2. Preferencias Personales y Expectativas:
 - Preferencias respecto a características de la pareja, como la importancia de la **atracción física**, **sinceridad**, **inteligencia**, **sentido del humor**, **ambición** y **intereses compartidos**. Estas preferencias están presentes tanto desde la perspectiva de cada participante hacia su pareja, como en la autopercepción.
 - Expectativas sobre el evento y el resultado, como el **número de coincidencias** esperadas y la **probabilidad de que la pareja esté interesada**.



3. Evaluaciones Mutuas:

- Valoraciones directas de cada participante sobre la **atracción física, sinceridad, inteligencia, ambición y humor** de su pareja.
- Las evaluaciones incluyen las percepciones de cada participante hacia su pareja y también la forma en que cada persona evalúa estos aspectos de sí mismo.

4. Intereses Personales:

- Nivel de interés en actividades recreativas, como **deportes, arte, música, lectura, teatro, cine, yoga** y más. Estos atributos ofrecen una visión de las afinidades de cada persona, lo cual podría influir en la decisión de tener una segunda cita.

5. Variables de Respuesta y Evaluaciones Rápidas:

- **match**: Variable objetivo que indica si existe interés mutuo para una segunda cita, representada en un formato binario (1 para "sí", 0 para "no").
- **like** y **guess_prob_liked**: Variables que reflejan el agrado por la pareja y la probabilidad estimada de que la pareja también esté interesada, respectivamente.

6. Rangos de Evaluación:

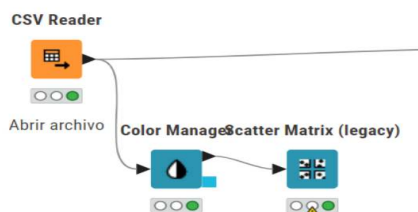
- Acompañando cada atributo de preferencia y autopercepción, existen variables que clasifican las respuestas en rangos de valor predeterminados (**d_nombreAtributo**), lo cual permite categorizar los datos y analizar tendencias en términos de rangos.

La variable objetivo a predecir es **match** e indica si ambos participantes desean tener una segunda cita o no, representada en valores binarios: 1 para "sí" y 0 para "no". Esta predicción implica clasificar y entender el peso de factores individuales en la decisión de buscar una segunda interacción.

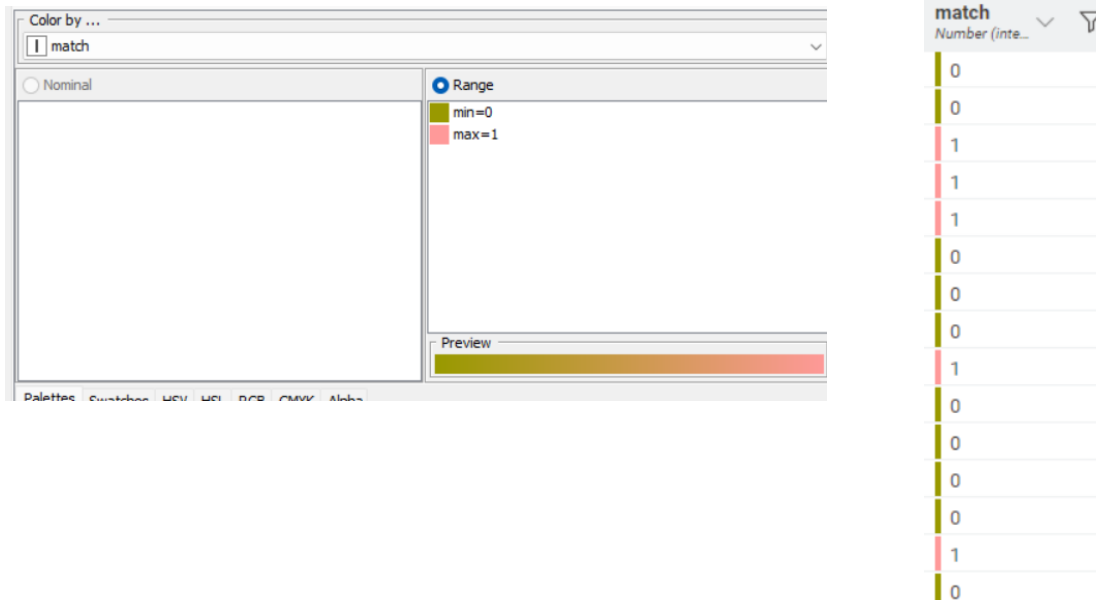
1.2 Interpretación de datos

Una vez visualizados todos los datos disponibles y hacer un estudio superficial de las variables, se realiza un estudio más profundo, haciendo divisiones y viendo la relación de las variables.

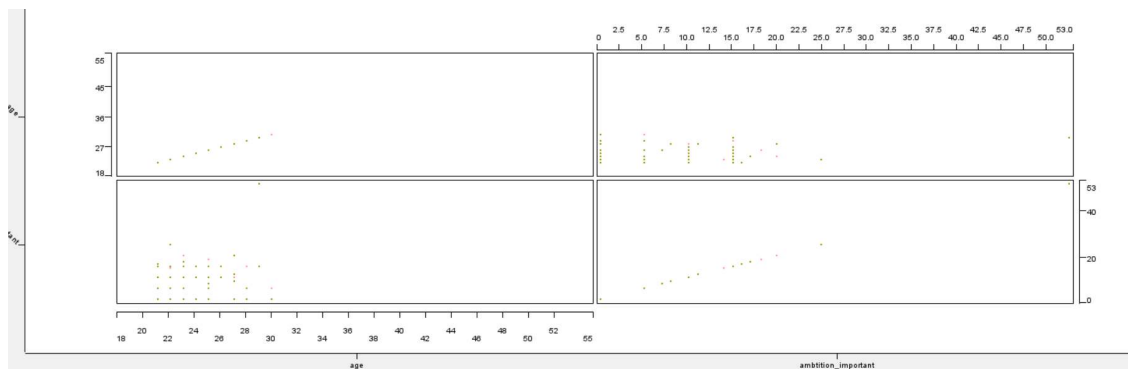
Para ello utilizamos los nodos Color Manager y Scatter Matrix.



Color manager , para nuestra variable clase match , atribuye un color en caso de que la respuesta sea 1 y otro en caso de que sea 0.



Tras esto , mediante Scatter Matrix , podremos ver la relación de las variables que elijamos con nuestra variable clase match, y así poder entender la relación de estas.



rel edad - importancia de ambicion

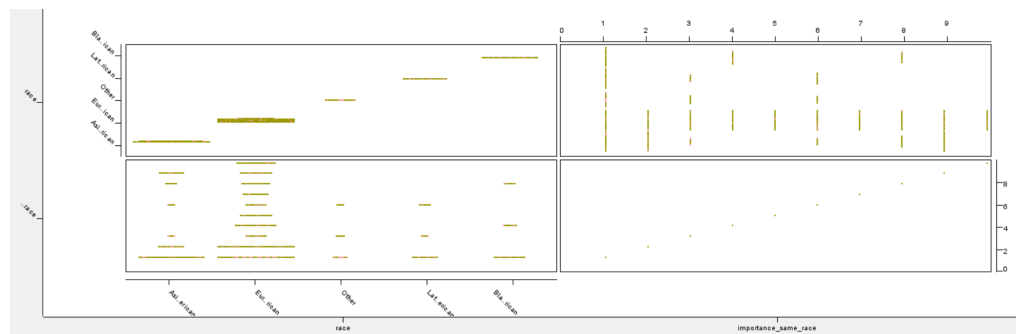
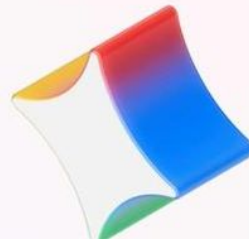
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



relacion raza- importancia misma raza

1.3 Posibles Problemas en los Datos

1. Balanceo de Clases:

- La variable objetivo (**match**) podría estar desbalanceada, dependiendo de la cantidad de coincidencias exitosas frente a las no exitosas. Este balanceo afectaría la precisión de algunos modelos de clasificación, especialmente aquellos sensibles a clases dominantes.

2. Valores Faltantes:

- Al ser un conjunto de datos basado en opiniones personales y evaluaciones subjetivas, es probable que existan valores faltantes en atributos donde los participantes omitieron responder o no consideraron ciertos aspectos de importancia. La gestión de estos valores es crucial para asegurar la calidad de las predicciones.

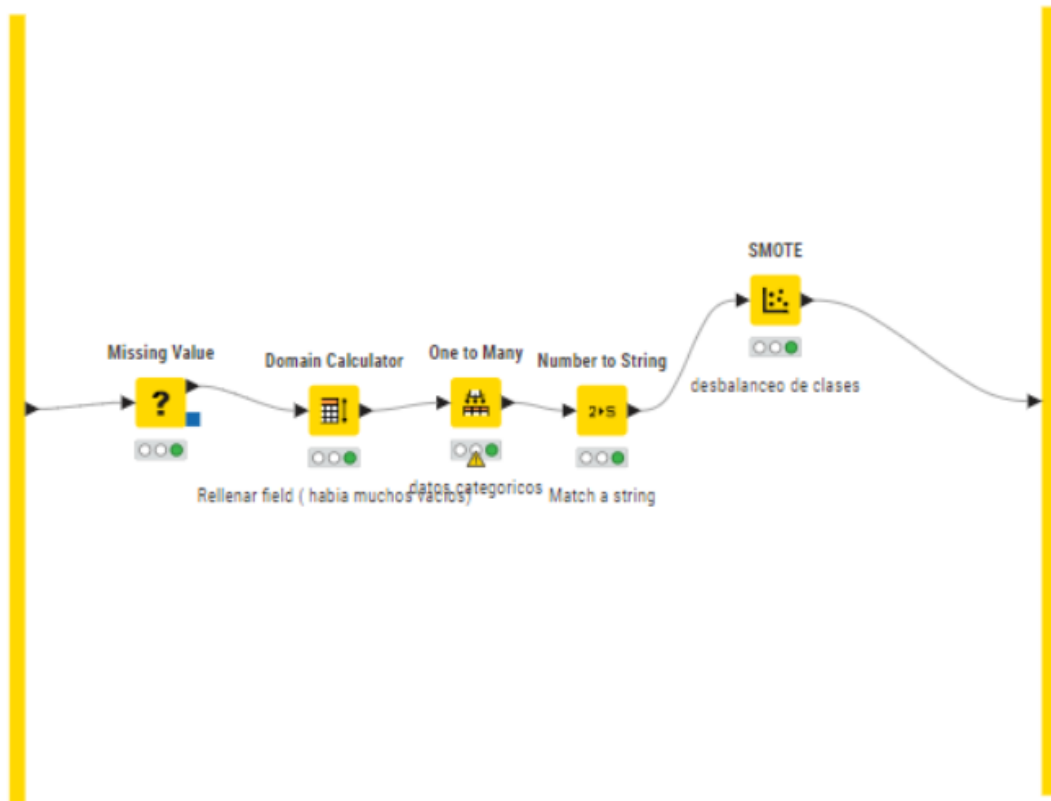
3. Diferentes Escalas de Medición:

- Las variables presentan diferentes escalas y tipos de datos (por ejemplo, algunas variables están en escala de 1 a 10, otras en 0 a 100, y algunas son binarias). Esta diversidad en las escalas requiere una normalización adecuada antes de aplicar ciertos algoritmos de clasificación que dependen de la distancia, como el k-NN.

2. Procesado de datos

Como ha sido explicado en el apartado 1.1 , el problema necesitará cierto preprocesamiento de datos a la hora de mejorar los resultados. En este apartado , incluiremos el preprocesamiento general que será aplicado a todos los algoritmos.

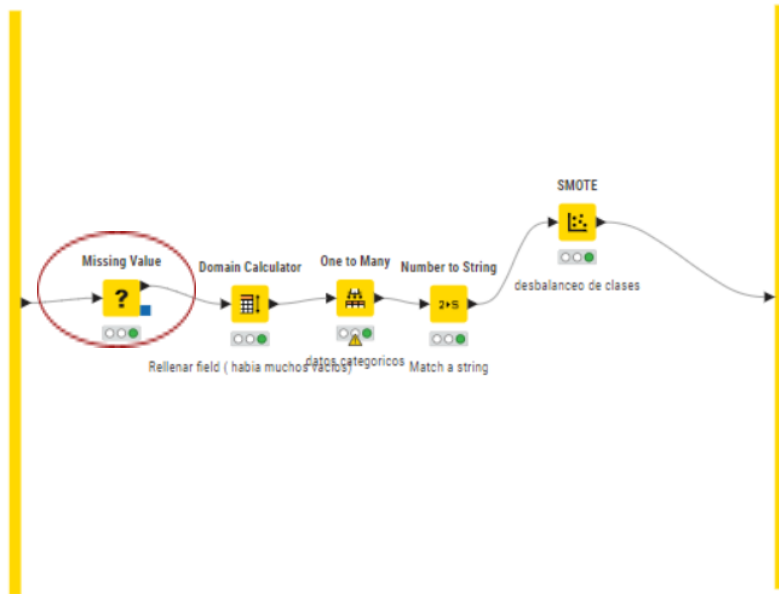
Para ello se ha incluido un metadato Preprocesamiento general , que consta de :



2.1 Trato de valores nulos

En el análisis de predicción de una segunda cita, es esencial gestionar adecuadamente los valores nulos dentro del conjunto de datos, ya que pueden afectar tanto la integridad del análisis como la precisión de los modelos predictivos. Los valores nulos pueden surgir en distintas variables, y su presencia podría influir en el rendimiento y la interpretabilidad de los resultados si no se tratan de forma adecuada. Por este motivo, primero evaluamos y tratamos estos valores antes de avanzar en el flujo de trabajo.

Para llevar a cabo esta estrategia, empleamos nodos específicos de tratamiento de valores nulos en KNIME, como Missing Value.



El tratamiento de valores nulos tiene los siguientes efectos en el flujo de trabajo:

1. **Aumento de la precisión y confiabilidad del modelo predictivo:** Al abordar los valores nulos de manera sistemática, aseguramos que las variables utilizadas en el modelo predictivo no presenten vacíos que puedan afectar su rendimiento. Esto reduce el riesgo de errores de procesamiento y garantiza que el modelo se entrene sobre un conjunto de datos más completo y representativo.
2. **Consistencia en el análisis y procesamiento:** El tratamiento de valores nulos evita conflictos en los nodos posteriores, que requieren datos completos para generar visualizaciones precisas, aplicar filtros y realizar cálculos sin interrupciones. Esto mejora la eficiencia del flujo de trabajo y permite que el análisis se realice de forma fluida y sin errores derivados de datos incompletos.
3. **Robustez y validez de los resultados:** Al eliminar o imputar valores nulos, el análisis final y las interpretaciones se basan en un conjunto de datos que es representativo y estadísticamente sólido. Esto permite obtener resultados más confiables y facilita la interpretación de patrones y relaciones clave en la predicción de segundas citas.

Al analizar la tabla para observar la tendencia de valores nulos en la misma , encontramos

attractive_o	Number (dou...	212	18
sinsere_o	Number (dou...	287	14
intelligence_o	Number (dou...	306	17
funny_o	Number (dou...	360	17
ambitious_o	Number (dou...	722	15
shared_intere...	Number (dou...	1076	15

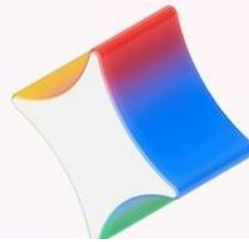
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

En el caso de `attractive_o` , `sinsere_o` , `intelligenece_0` , `funny_o` , `ambitious_o` , estos se refieren todos a cuánto valora la pareja todas estas cualidades sobre el usuario . Los valores en estas categorías no presentan valores muy extremos , ya que la franja suele ser [6-9] habiendo alguna variación , pero no muy drástica. Por ello , imputaremos estas categorías por media.

Para `shared_interests` usaremos la mediana. Dado que este tipo de datos pueden tener variaciones significativas según la percepción individual de cada persona, la mediana es una opción sólida. La mediana ayuda a reducir el impacto de valores atípicos que podrían distorsionar los datos, como calificaciones extremadamente altas o bajas que no son comunes.

<code>attractive_par...</code>	Number (dou...	202
<code>sincere_partn...</code>	Number (dou...	277
<code>intelligence_p...</code>	Number (dou...	296
<code>funny_partner</code>	Number (dou...	350
<code>ambition_par...</code>	Number (dou...	712
<code>shared_intere...</code>	Number (dou...	1067

Lo mismo encontramos en estas categorías , que representan los mismos valores pero valorando a la otra persona, por lo que imputaremos igual que antes .

<code>expected_nu...</code>	Number (inte...	6578
<code>expected_nu...</code>	Number (dou...	1173

La mediana es una buena opción, ya que es menos sensible a valores extremos. Dado que estas expectativas pueden variar considerablemente de una persona a otra (algunos pueden tener expectativas muy altas o muy bajas) de hecho , los valores encontrados son o del rango [0,3] o del rango [10,15]. La mediana ofrece un valor central que no se ve influido por las expectativas extremas.

WUOLAH

like	Number (dou...	240
guess_prob_li...	Number (dou...	309
d_like	String	0
d_guess_pro...	String	0
met	Number (inte...	375
match	Number (inte...	0

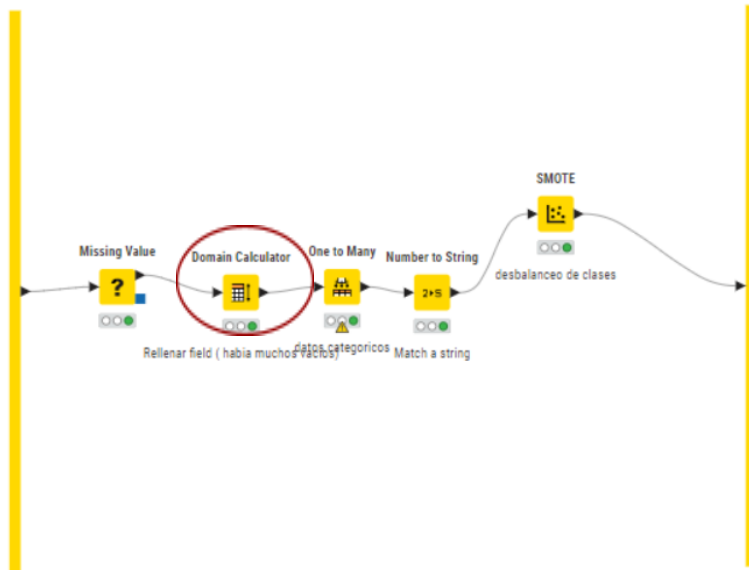
Mientras que con like los valores observados tampoco fluctúan mucho , las fluctuaciones de guess_probablility_like sí que lo hacen más. Por ello en la primera haremos la media y la segunda la mediana . En cuanto a met , utilizaremos un valor fijo (-1) para aquellos valores perdidos.

attractive_par...	Number (dou...	202
sincere_partn...	Number (dou...	277
intelligence_p...	Number (dou...	296
funny_partner	Number (dou...	350
ambition_par...	Number (dou...	712
shared_intere...	Number (dou...	1067

2.2 Actualización valores

En el análisis de predicción de una segunda cita, es fundamental asegurar que los datos de entrada reflejen con precisión la estructura y el contenido del conjunto de datos actual. Después de realizar etapas de preprocesamiento, como la eliminación de registros atípicos o la limpieza de datos faltantes, los valores de algunas variables, como **field**, pueden haber cambiado. Esto significa que la lista de categorías o el rango de valores numéricos de la variable podría no estar actualizado en la especificación de la tabla, generando posibles inconsistencias en los nodos que dependen de esta información.

Para solucionar esto, utilizamos el nodo **Domain Calculator** en la variable **field**. Este nodo escanea la variable seleccionada en su estado actual y actualiza automáticamente su información de dominio, ajustando la lista de valores posibles y los límites de valores mínimo y máximo.



Los efectos de esta actualización son los siguientes:

1. **Precisión en el análisis y procesamiento posterior:** Con **Domain Calculator**, la información de **field** refleja únicamente las categorías activas o el rango numérico real del conjunto de datos actual. Esto elimina valores obsoletos o fuera de rango, optimizando el trabajo de nodos posteriores como los de filtrado, visualización y clasificación, que ahora operan únicamente con los valores válidos.
2. **Integridad del modelo predictivo:** La actualización asegura que los datos de entrada en el modelo sean precisos y representativos, evitando errores de procesamiento que puedan surgir por categorías o rangos incorrectos. Esto mejora la calidad y la consistencia del entrenamiento del modelo predictivo, asegurando que las predicciones se basen en datos válidos y relevantes para el análisis.

2.3: Variables categóricas

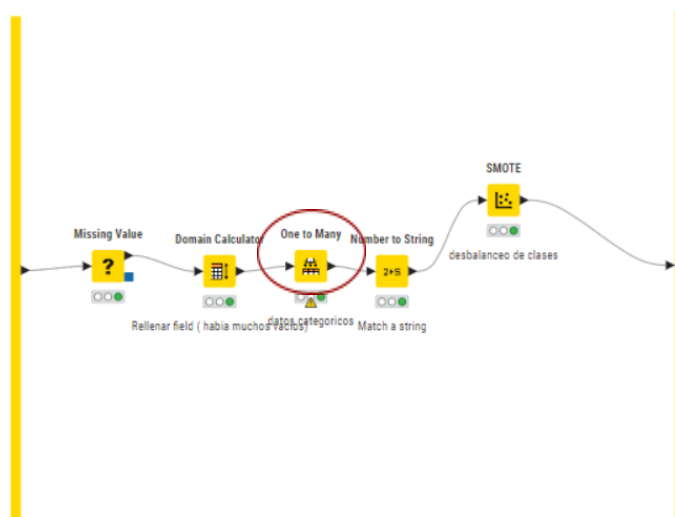
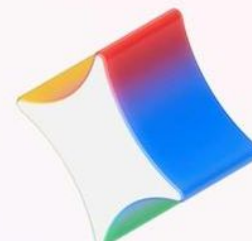
Para el procesamiento de los atributos categóricos en nuestro conjunto de datos, específicamente en las variables **gender**, **race**, **race_o** y **field**, utilizamos el nodo **One to Many** en KNIME.

Google Gemini: Plan Pro a 0€ durante 1 año. Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

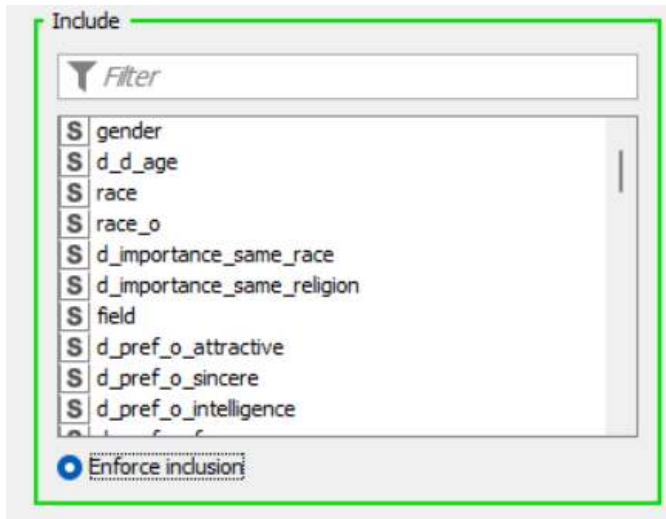
Consigue la oferta

Después 21,99€/mes



En nuestro conjunto de datos, algunas variables, como **gender** (género del participante), **race** (raza del participante), **race_o** (raza de la pareja asignada) y **field** (área de estudio del participante), son categóricas. Estas variables contienen valores de texto que representan diferentes categorías (por ejemplo, en **gender**, los valores pueden ser “female” o “male”). Dado que la mayoría de los algoritmos de clasificación y análisis en aprendizaje automático no pueden procesar variables de tipo texto o categóricas directamente, necesitamos convertir estas categorías en un formato numérico que los algoritmos puedan entender.

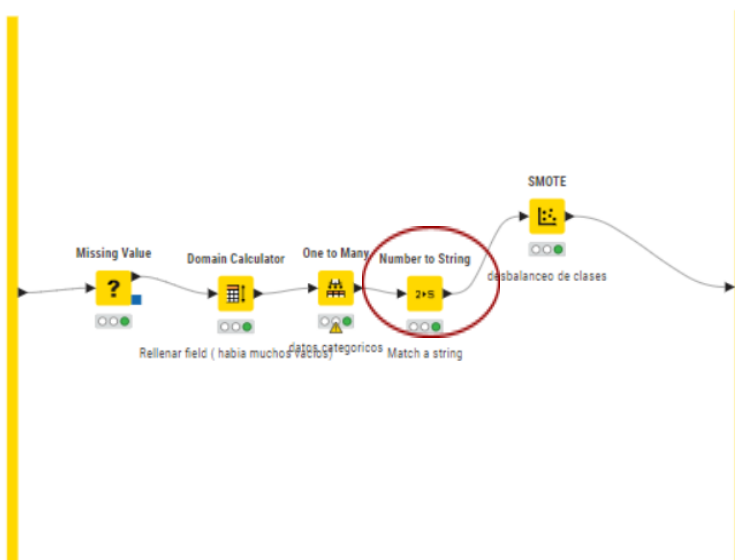
Incluimos aquí también todas las diferencias (d_attractive_o , d_funny, d_sincere ...) , ya que también se basan en strings , concretamente en rangos .



2.4 Convertir match a String.

En el análisis de predicción de una segunda cita, es crucial asegurarse de que las variables estén en el formato adecuado para cada tipo de modelo predictivo. En este caso, la variable **match**, que indica si hubo un interés mutuo en una segunda cita (1 para "sí", 0 para "no"), originalmente está en formato numérico. Sin embargo, para ciertos algoritmos de clasificación, trabajar con variables categóricas de tipo texto puede mejorar el rendimiento y facilitar el procesamiento.

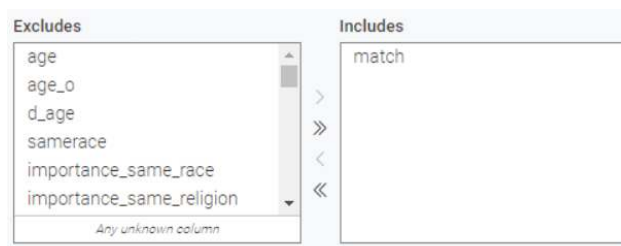
Por este motivo, incluimos el nodo **Number to String** en el flujo de trabajo para convertir **match** de un formato numérico a una cadena de texto.



Este paso de preprocesamiento tiene varios efectos en el flujo de trabajo y en el rendimiento del modelo:

1. **Compatibilidad con algoritmos de clasificación:** Algunos algoritmos, especialmente los que operan en categorías (como árboles de decisión o Naive Bayes), procesan mejor las clases cuando están en formato de texto en lugar de numérico. Al convertir **match** a una cadena de texto, facilitamos que estos algoritmos identifiquen la variable de clase como categórica, lo cual optimiza su desempeño y evita errores de procesamiento.
2. **Claridad en el análisis y visualización:** Al cambiar **match** a texto, el flujo de trabajo es más claro y coherente al momento de visualizar los datos y los resultados. En lugar de "1" o "0", la variable de clase aparece ahora como categorías textuales que indican claramente los posibles resultados, lo cual facilita la interpretación de las salidas y análisis.
3. **Flexibilidad para diferentes nodos y configuraciones:** Este cambio de formato permite que otros nodos en el flujo de trabajo, como los de análisis estadístico y de balanceo de clases, identifiquen correctamente **match** como una variable categórica. Esto es particularmente útil en procesos de balanceo de clases o segmentación, donde las variables categóricas permiten aplicar técnicas específicas para datos de clasificación.
4. **Evitar ambigüedades en la especificación de la variable de clase:** El cambio de formato con el nodo **Number to String** asegura que no haya ambigüedad en la interpretación de **match** por parte del sistema, mejorando la consistencia y evitando posibles errores en nodos posteriores, que pueden depender de la correcta especificación del tipo de variable.

Aunque esto pueda parecer ineficiente ya que aparentemente solo dos clasificadores necesitan este preprocesamiento, realmente es igual de eficiente hacer este cambio de primeras, ya que solo tendrá que ser de nuevo convertido a número para el algoritmo red neuronal.



2.5 Desbalanceo de clases.

En el análisis de predicción de una segunda cita, nos encontramos con un problema de desbalanceo de clases en la variable de salida match (la variable de clase), que indica si hubo un interés mutuo en una segunda cita. Este desbalanceo significa que una de las clases (por ejemplo, "sí" o "no" en cuanto al interés en una segunda cita) está

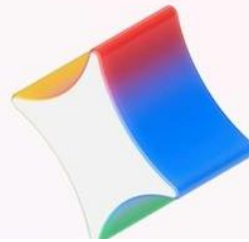
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes

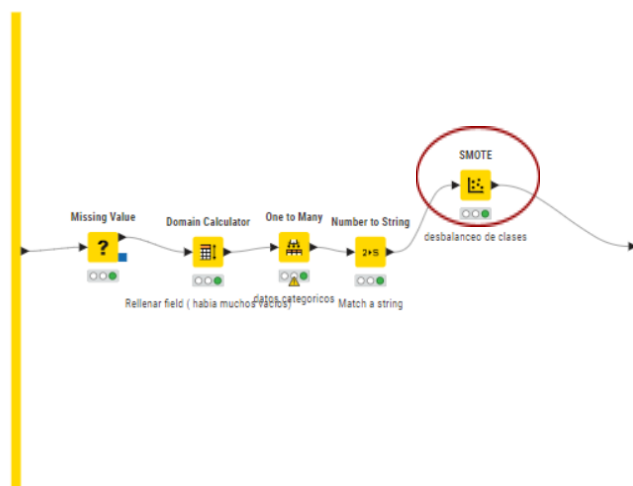


Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

sobrerrepresentada, lo que puede sesgar los resultados del modelo predictivo hacia la clase mayoritaria. En nuestro caso, se observa una cantidad mucho mayor de noes que de síes :

0 (6998; 83.53%), 1 (1380; 16.47%)

Para abordar este problema y asegurar que el modelo pueda aprender a reconocer ambas clases de manera efectiva, empleamos la técnica SMOTE (Synthetic Minority Over-sampling Technique).



Aplicación de SMOTE: Este método crea ejemplos sintéticos de la clase minoritaria, equilibrando el número de ejemplos entre las clases. Esto se hace de forma controlada, seleccionando instancias de la clase minoritaria y generando nuevos datos entre puntos cercanos en el espacio de características. Así, en lugar de replicar datos existentes, SMOTE genera ejemplos nuevos que representan mejor la distribución de la clase minoritaria, lo cual mejora la capacidad del modelo para identificar correctamente ambas clases.

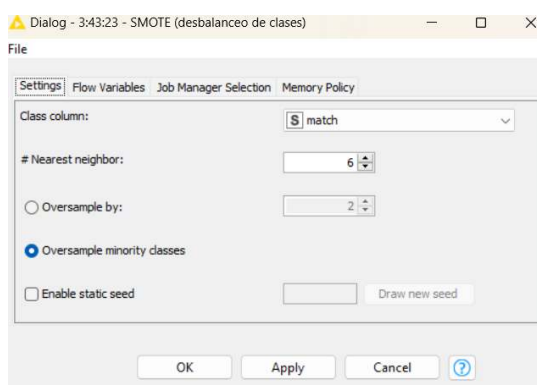
En el análisis de predicción de una segunda cita, además de aplicar SMOTE con 6 vecinos, se ha activado la opción de oversample minority classes. Esta configuración tiene un impacto significativo en cómo se balancean las clases, permitiendo un aumento de las instancias de la clase minoritaria para asegurar un equilibrio adecuado en el conjunto de datos. Al configurar SMOTE para utilizar 6 vecinos, la generación de ejemplos sintéticos

WUOLAH

sigue un enfoque controlado, lo que asegura que las nuevas instancias mantengan la coherencia con la estructura original de la clase minoritaria. De este modo, los ejemplos generados son lo suficientemente variados para representar la clase minoritaria sin alejarse demasiado de los datos reales, lo que facilita que el modelo identifique patrones relevantes. Además, el uso de 6 vecinos ayuda a reducir el riesgo de sobreajuste, ya que evita la creación de datos excesivamente específicos alrededor de unos pocos puntos de la clase minoritaria, permitiendo que el modelo generalice mejor y aprenda patrones más representativos del conjunto de datos en su conjunto.

La activación de la opción de oversample minority classes garantiza que la clase minoritaria sea suficientemente representada en el conjunto de datos. Esto asegura un balance más fuerte entre las clases, lo que evita que el modelo favorezca la clase mayoritaria y mejora la capacidad del modelo para identificar correctamente los patrones de la clase minoritaria. Este balance también tiene un impacto positivo en las métricas de evaluación, como el recall y la F1-score, para la clase minoritaria, lo que permite una evaluación más justa y representativa de ambas clases. Además, el balanceo mediante oversampling optimiza el rendimiento de algoritmos de clasificación que tienden a estar sesgados hacia la clase mayoritaria, como los árboles de decisión o las redes neuronales.

Al considerar otras configuraciones, si se optara por un número menor de vecinos, como 3 o 4, se obtendrían ejemplos más cercanos a los puntos originales de la clase minoritaria, lo que podría ser útil si los datos son muy escasos y dispersos. Sin embargo, con menos vecinos, el modelo podría perder algo de variabilidad en los ejemplos generados. Por otro lado, si se eligiera un número mayor de vecinos, como 8 o 10, la variabilidad de los ejemplos aumentaría, pero el riesgo sería que los puntos generados se solaparan más con la clase mayoritaria, lo que podría reducir la precisión del modelo si las clases no son fácilmente separables. Una configuración más avanzada permitiría que el número de vecinos variara según la densidad local de la clase minoritaria, lo cual podría ser útil en casos donde la distribución de la clase minoritaria es heterogénea, pero esta opción generalmente requiere mayor ajuste y mayor capacidad computacional.

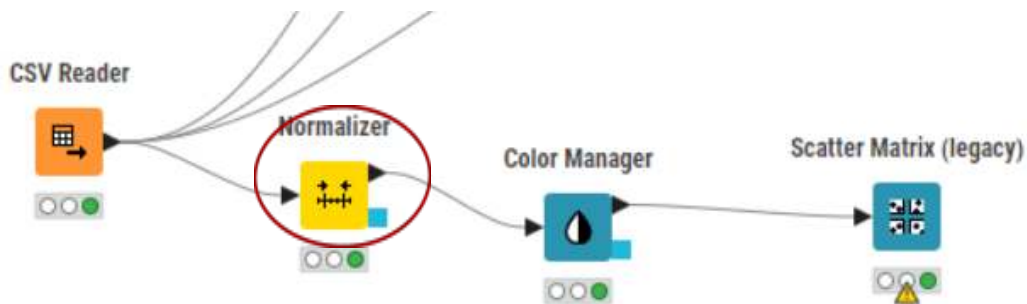


2.6 Normalización de Atributos

Para ciertos algoritmos de clasificación, como el k-Nearest Neighbors (k-NN), es fundamental que las variables numéricas estén en una escala comparable, ya que estos algoritmos basados en distancia pueden verse afectados por la diferencia en magnitudes entre variables. Sin una normalización adecuada, los atributos con mayores rangos de valores tienden a dominar las distancias calculadas, lo cual puede desbalancear las predicciones y llevar a resultados erróneos o no óptimos.

Proceso de Normalización

En el contexto del análisis predictivo para el proyecto de citas rápidas, el nodo **Normalizer** de KNIME se usa para escalar las variables numéricas al rango [0, 1]. Esto implica que todos los valores de un atributo se ajustarán de manera que el valor mínimo se convierta en 0 y el máximo en 1



2.7 Análisis de Componentes Principales

Se ha utilizado **PCA** en el preprocesamiento de datos, específicamente para el clasificador de **Regresión Logística**, debido a los beneficios que aporta en modelos con muchas variables y posibles correlaciones:

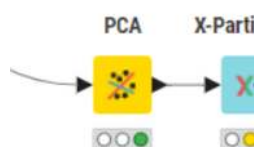
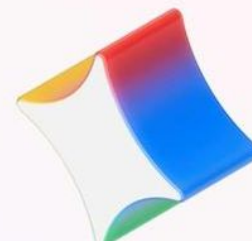
1. **Reducción de Dimensionalidad:** PCA permite reducir el número de variables, transformando las originales en componentes principales que retienen la mayor parte de la información. Esto simplifica el modelo y ayuda a evitar el sobreajuste.
2. **Mitigación de Colinealidad:** Al convertir las variables en componentes independientes, PCA elimina problemas de correlación que pueden afectar la estabilidad y precisión de la Regresión Logística.
3. **Optimización Computacional:** Reducir las dimensiones disminuye el tiempo de entrenamiento y facilita el proceso de validación cruzada.

Google Gemini: Plan Pro a 0€ durante 1 año. Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

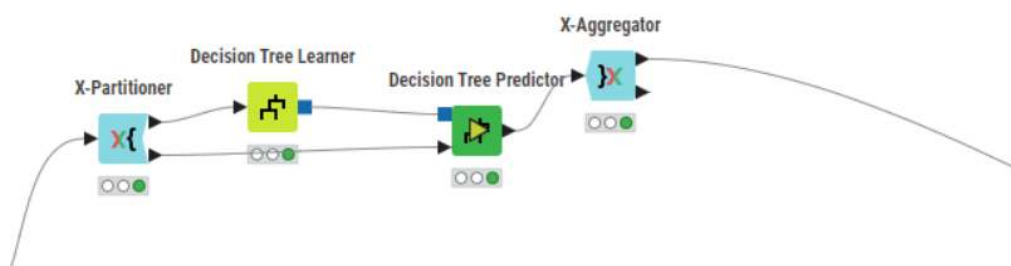
Consigue la oferta

Después 21,99€/mes



3. Resultados obtenidos:

3.1 Decision tree maker



1. Preprocesamiento de Datos

Para empezar, se realizó un preprocesamiento exhaustivo de los datos con el fin de preparar el conjunto para el análisis. Se utilizaron los nodos de preprocesamiento general, incluyendo missing value , one to many , number to string y SMOTE .

2. Configuración de la Validación Cruzada

Se añadió el nodo **X-Partitioner** para implementar una validación cruzada de 5 particiones. Esto dividió los datos en cinco subconjuntos, permitiendo que cada partición actuara como conjunto de prueba una vez, mientras las otras cuatro se usaban como conjunto de entrenamiento en cada iteración. De esta manera, se logra una evaluación robusta del rendimiento del modelo en diferentes subconjuntos.

3. Entrenamiento del Modelo con Árbol de Decisión

Para el entrenamiento, se conectó el nodo **X-Partitioner** al nodo **Decision Tree Learner**. En este nodo, se configuraron parámetros clave:

- **Profundidad máxima del árbol:** Esta configuración evitó que el modelo se sobreajustara.
- **Número mínimo de registros por hoja:** Este parámetro se ajustó para controlar el tamaño de las hojas del árbol, ayudando a mejorar la interpretabilidad del modelo.

4. Generación de Predicciones

A continuación, se añadió el nodo **Decision Tree Predictor**, que aplicó el modelo de árbol de decisión entrenado sobre el conjunto de prueba generado en cada iteración de la validación cruzada. Este nodo produjo las predicciones para la variable objetivo (**match**).

5. Consolidación de Resultados

Para recopilar los resultados de todas las iteraciones, se añadió el nodo **X-Aggregator**. Este nodo consolidó las predicciones de todas las particiones, permitiendo el cálculo de métricas globales del modelo en la validación cruzada.

6. Evaluación del Rendimiento del Modelo

Para evaluar el modelo, se utilizó el nodo **Scorer**, que permitió comparar las predicciones del modelo con los valores reales de la variable objetivo. Este nodo generó varias métricas de rendimiento, tales como:

- **Precisión**
- **TPR (Tasa de Verdaderos Positivos)**
- **TNR (Tasa de Verdaderos Negativos)**
- **F1-Score**
- **Accuracy**
- **G-mean**

Estas métricas ayudaron a entender cómo de bien se desempeñaba el modelo en términos de precisión y equilibrio entre las dos clases.

Matriz de confusion:

match \ Pr...	0	1
0	5847	1065
1	776	6211

Metrica	Valor promedio
Precision	0.737
TPR	0.927
TNR	0.962
F1-Score	0.821
Accuracy	0.944
G-mean	0.944

7. Complejidad del Algoritmo: Árbol de Decisión

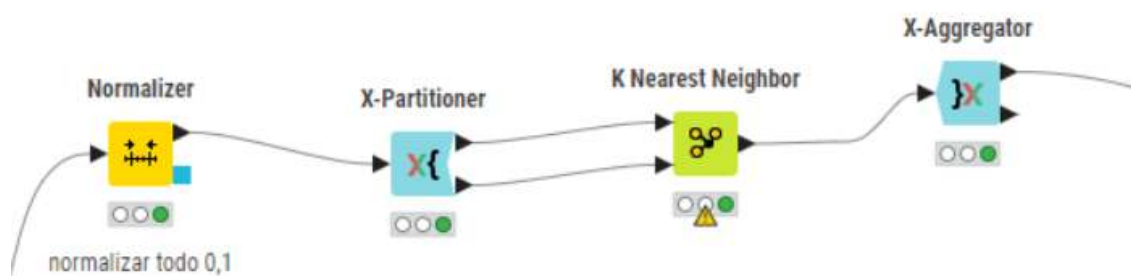
-Profundidad Máxima del Árbol

-Número de Nodos y Hojas

Explicación y elección en punto 4.

3.2 k-Nearest Neighbors (k-NN)

Flujo de Trabajo en KNIME para k-NN



1. Preprocesamiento de Datos

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

Al igual que con el árbol de decisión, se llevó a cabo un preprocesamiento exhaustivo de los datos para asegurar que el conjunto estuviera listo para el análisis. Se emplearon los nodos de preprocesamiento general mencionados en el apartado anterior , además del nodo Normalizer . En k-NN, las variables deben estar escaladas en una misma unidad para evitar que alguna característica numérica tenga un impacto desproporcionado en el cálculo de distancias. Por eso, se utilizó un nodo de normalización para estandarizar todas las variables.

2. Configuración de la Validación Cruzada

Se utilizó el nodo X-Partitioner para dividir los datos en cinco subconjuntos y realizar una validación cruzada de 5 particiones. Este proceso permitió evaluar el rendimiento del modelo en diferentes subconjuntos, reduciendo la posibilidad de sobreajuste y proporcionando una medida robusta de su precisión.

3. Entrenamiento y generación de predicciones del Modelo k-NN

Para el entrenamiento predicción del modelo, se usó el nodo k Nearest Neighbor , que ya incluye el entrenamiento y la predicción , a diferencia de los nodos para los otros clasificadores . El principal parámetro configurado fue el valor de k.

Tras experimentar con varios valores de k, se seleccionó un valor óptimo que maximiza la precisión del modelo. En este caso, se utilizó k=3 para mantener un equilibrio entre sensibilidad a la variabilidad de los datos y reducción de ruido.

4. Consolidación de Resultados

Se usó el nodo X-Aggregator para consolidar los resultados de cada partición, permitiendo calcular métricas de rendimiento globales en la validación cruzada. Este paso es fundamental para obtener una medida promedio del rendimiento del modelo.

5. Evaluación del Rendimiento del Modelo

Para la evaluación, se utilizó el nodo Scorer, que proporcionó las métricas de desempeño al comparar las predicciones del modelo con los valores reales de la variable objetivo. Las métricas principales incluyeron:

Métricas del Modelo k-NN

Métrica	Valor promedio
Precisión	0.849



TPR	0.886
TNR	0.84
F1-Score	0.867
Accuracy	0.864
G-mean	0.863

6.Complejidad del algoritmo

-Valor de k (Número de Vecinos):

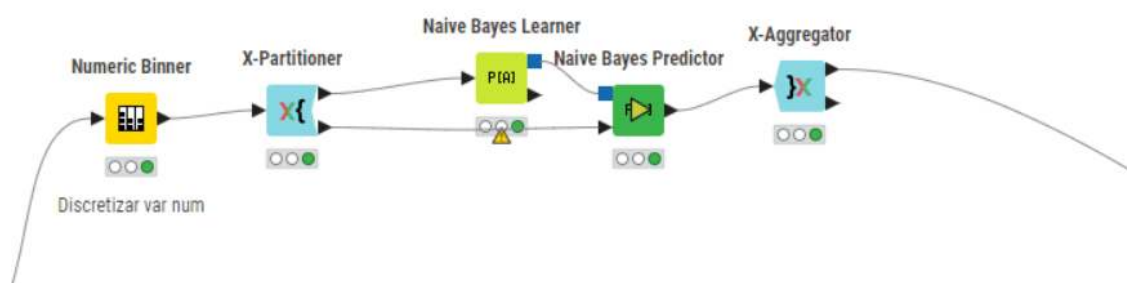
-Distancia Computacional:

- La **distancia euclidiana** es la métrica utilizada, y cada predicción requiere calcular esta distancia entre el punto a clasificar y todos los puntos del conjunto de entrenamiento.
- El costo computacional de k-NN es $O(n * d)$ por cada predicción, donde **n** es el número de instancias en el conjunto de entrenamiento y **d** es la dimensionalidad (número de características) del espacio de datos. Esto puede ser un reto en conjuntos de datos grandes, ya que el modelo debe calcular la distancia a cada punto de referencia.

Mayor explicación y elección en punto 4.

3.3 Naive Bayes

Flujo de Trabajo en KNIME para Naive Bayes



1. Preprocesamiento de Datos:

- Para asegurar que los datos estuvieran listos para ser analizados, se realizaron los mismos pasos de preprocesamiento usados en otros modelos , missing value , one to many , number to string y SMOTE;

2. Validación Cruzada:

- La validación cruzada de 5 particiones se implementó mediante el nodo *X-Partitioner*, dividiendo los datos en cinco subconjuntos para que cada partición pudiera servir de conjunto de prueba una vez. Esta metodología asegura una evaluación confiable del modelo.

3. Entrenamiento y Predicción del Modelo Naive Bayes:

- El entrenamiento se realizó con el nodo *Naive Bayes Learner* seguido del nodo *Naive Bayes Predictor* para aplicar el modelo sobre el conjunto de prueba en cada iteración de la validación cruzada.

4. Consolidación de Resultados:

- Se utilizó el nodo *X-Aggregator* para reunir y promediar los resultados de todas las particiones, proporcionando métricas globales para evaluar el rendimiento del modelo.

5. Evaluación del Rendimiento del Modelo:

- Para analizar las predicciones generadas, se utilizó el nodo *Scorer*, que proporcionó métricas clave para evaluar el modelo:

Métricas del Modelo Naive Bayes

Prediction ...	1	0
1	6998	6998
0	0	0

Metrica	Valor Promedio
Precision	0.5
TPR	1
TNR	0
F1-Score	0.667
Accuracy	0.5
G-mean	0

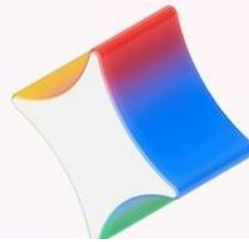
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

6. Complejidad del Algoritmo

Aunque Naive Bayes es conceptualmente simple, sus medidas de complejidad pueden analizarse desde varios ángulos:

1. Cálculo de Probabilidades:

- Naive Bayes se basa en el cálculo de probabilidades condicionales y de la probabilidad marginal para cada clase. Esto implica que, para cada característica, el modelo estima la probabilidad de que los datos pertenezcan a cada clase. La complejidad es proporcional al número de características y clases, $O(d * c)$, donde d es el número de características y c es el número de clases.

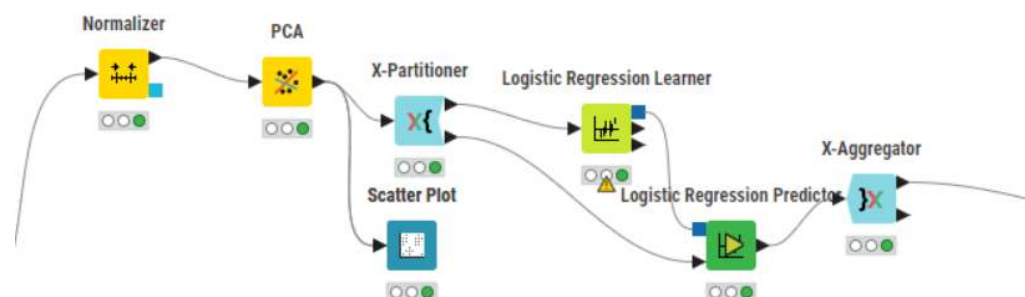
2. Número de Parámetros:

- Naive Bayes calcula una media y una varianza para cada característica en cada clase, lo que significa que el número de parámetros estimados es proporcional al número de características y de clases. En este caso, con dos clases (match y no match) y múltiples características, el modelo mantiene un bajo costo en el cálculo de probabilidades para cada característica en ambas clases.

Mayor explicación y elección en punto 4.

3.4 Regresión Logística

Flujo de Trabajo en KNIME para Regresión Logística

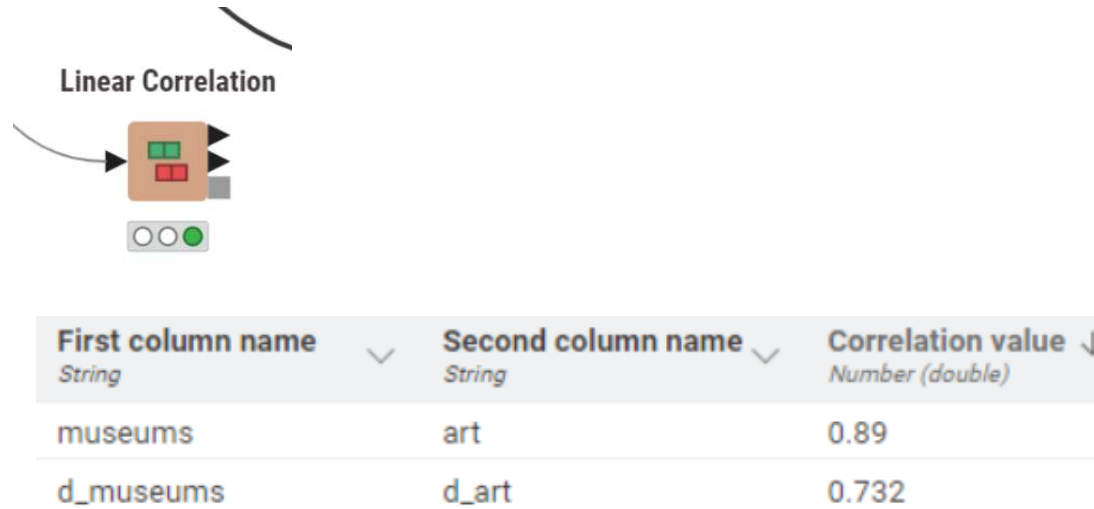


1. Preprocesamiento de Datos:

- El preprocesamiento se realizó siguiendo los mismos pasos que en los algoritmos previos para asegurar consistencia

WUOLAH

Además del preprocesamiento general , he decidido añadir una revisión de multicolinealidad entre variables .Esto se debe a que la regresión logística asume independencia entre las variables predictoras. Si existen variables altamente correlacionadas, los coeficientes del modelo pueden ser inestables.



Tras ello , hemos utilizado **PCA (Análisis de Componentes Principales)** en el preprocesado de datos como una estrategia para mejorar la eficiencia del modelo, reducir la complejidad y minimizar el ruido en el conjunto de datos.

PCA combina las variables originales en componentes ortogonales, eliminando las correlaciones entre ellas y proporcionando variables independientes. Esto facilita una **mejor interpretación y precisión de los coeficientes de la regresión**.

2. Configuración de la Validación Cruzada:

- Se utilizó el nodo *X-Partitioner* para implementar una validación cruzada de 5 particiones. Este proceso permite evaluar el rendimiento del modelo en diferentes subconjuntos de datos y obtener métricas confiables.

3. Entrenamiento y Predicción del Modelo de Regresión Logística:

- Para el entrenamiento y la predicción, se emplearon los nodos *Logistic Regression Learner* y *Logistic Regression Predictor*. En el nodo de entrenamiento, se ajustaron varios parámetros:
 - **Regularización:** Se utilizó una regularización L2 para reducir el riesgo de sobreajuste, logrando así un modelo más generalizable.
 - **Parámetro de ajuste (lambda):** Se experimentó con diferentes valores de regularización para equilibrar precisión y simplicidad del modelo.

4. Consolidación de Resultados:

- Con el nodo *X-Aggregator*, se agruparon los resultados de las cinco particiones, obteniendo métricas globales del modelo y evaluando así el rendimiento promedio en todo el conjunto de datos.

5. Evaluación del Rendimiento del Modelo:

- Para evaluar la calidad de las predicciones, se utilizó el nodo *Scorer*, obteniendo métricas fundamentales:
 - **Precisión:** Proporción de predicciones correctas.
 - **TPR (Tasa de Verdaderos Positivos) y TNR (Tasa de Verdaderos Negativos):** Indicadores de la capacidad del modelo para identificar correctamente las clases.
 - **F1-Score:** Medida de precisión equilibrada que combina sensibilidad y precisión, útil en contextos donde el balance entre clases es importante.

Métricas del Modelo de Regresión Logística

Métrica	Valor Promedio
Precision	0.806
TPR	0.797
TNR	0.809
F1-Score	0.802
Accuracy	0.5803
G-mean	0.803

6. Complejidad del Algoritmo

1. **Número de Parámetros:**
2. **Regularización y Estabilidad del Modelo:**
3. **Interpretabilidad:**
 - Aunque es un modelo más sencillo que otros clasificadores avanzados, la regresión logística permite una interpretación detallada de cada parámetro en términos de impacto sobre la probabilidad de una segunda cita. Esta simplicidad en la estructura es una ventaja de complejidad en comparación con algoritmos más complejos, como redes neuronales.

Mayor explicación y elección en punto 4.

3.5 Random Forest

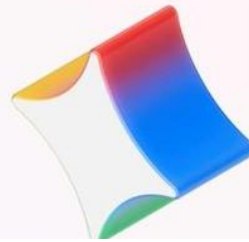
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

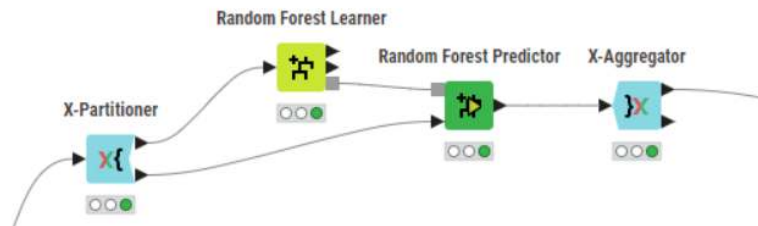
Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google



Flujo de Trabajo en KNIME para Random Forest

Preprocesamiento de Datos

Para preparar los datos antes del modelado, se han realizado los mismos pasos de preprocesamiento utilizados en los otros algoritmos, missing value, one to many, number to string y SMOTE

Validación Cruzada

Para asegurar una evaluación robusta del modelo, se ha implementado una validación cruzada con cinco particiones utilizando el nodo **X-Partitioner**. Este nodo divide los datos en cinco subconjuntos, donde cada partición actúa como conjunto de prueba en una de las iteraciones, mientras que el resto se utiliza para entrenamiento. Este método permite obtener métricas más representativas del desempeño del modelo.

Entrenamiento y Predicción del Modelo Random Forest

El entrenamiento se ha llevado a cabo mediante el nodo **Random Forest Learner**, el cual ha recibido como entrada las particiones de datos de entrenamiento generadas por el **X-Partitioner**. Luego, el nodo **Random Forest Predictor** se ha utilizado para aplicar el modelo entrenado en los datos de prueba de cada partición. Esto ha permitido obtener predicciones en cada ciclo de validación cruzada.

Consolidación de Resultados

Se ha utilizado el nodo **X-Aggregator** para recopilar los resultados de cada iteración de la validación cruzada. Este nodo promedia las métricas obtenidas en cada partición y proporciona una vista global del rendimiento del modelo, lo que permite evaluar su precisión y estabilidad.

Evaluación del Rendimiento del Modelo

Para evaluar el rendimiento del modelo, se ha empleado el nodo **Scorer**, el cual genera las siguientes métricas clave :

- **Precisión:** Indica el porcentaje de predicciones correctas realizadas por el modelo.

WUOLAH

- **TPR (Tasa de Verdaderos Positivos):** Refleja la habilidad del modelo para identificar correctamente los “matches”.
- **TNR (Tasa de Verdaderos Negativos):** Representa la capacidad del modelo para identificar correctamente los “no matches”.
- **F1-Score:** Proporciona un equilibrio entre precisión y sensibilidad.

Métricas del Modelo Random Forest

Métrica	Valor promedio
Precision	0.96
TPR	0.927
TNR	0.962
F1-Score	0.943
Accuracy	0.944
G-mean	0.944

Complejidad del Algoritmo

Cálculo de Varios Árboles: Random Forest construye múltiples árboles de decisión, generalmente unos 100 o más, según los parámetros seleccionados. La complejidad de este modelo es proporcional a la profundidad y número de nodos en cada árbol, multiplicado por el número de árboles, lo que se traduce en una complejidad de $O(T \cdot d \log d)$, donde T es el número de árboles y d el número de características.

Número de Parámetros: A diferencia de los modelos tradicionales, en Random Forest no se calculan coeficientes individuales. En su lugar, cada árbol contribuye al voto final para la predicción. Este sistema hace que el número de parámetros en Random Forest sea mayor que en modelos individuales, pero su configuración de ensamble reduce el riesgo de que el modelo dependa de parámetros específicos.

4. Configuración de los Algoritmos y Análisis de Resultados

4.1 Árbol de Decisión

El Árbol de Decisión divide el espacio de características en base a reglas secuenciales, formando un árbol de nodos de decisión. Los parámetros principales que determinan su rendimiento son:

- **Profundidad Máxima:** Controla hasta qué nivel el árbol puede dividirse. Limitarla ayuda a evitar el sobreajuste.
- **Número Mínimo de Registros por Hoja:** Número mínimo de registros en cada hoja, que, al aumentarse, reduce el sobreajuste.
- **Criterio de División:** Define la métrica de división de los nodos.
 - **Gini:** Adecuado para problemas de clasificación binaria.
 - **Entropía:** Proporciona información de impureza de nodos.
- **Mínimo de Muestras para Dividir un Nodo:** Mínimo de muestras necesarias para dividir un nodo.

Parámetro	Valor Inicial	Configuración n 1	Configuración n 2	Configuración n 3
Profundidad Máxima	Sin Límite	5	10	15
Número Mínimo de Registros por Hoja	2	5	10	15
Criterio de División	Gini	Entropía	Gini	Entropía
Mínimo de Muestras para Dividir	2	5	5	10

Resultados de las Configuraciones Alternativas

Métrica	Configuración 1	Configuración 2	Configuración 3
---------	-----------------	-----------------	-----------------

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

Precisión	85%	86%	84%
F1-Score	0.84	0.85	0.83
Sensibilidad (TPR)	0.84	0.85	0.83
Especificidad (TNR)	0.86	0.87	0.86

Conclusión: La Configuración 2 (profundidad = 10) es la mejor opción, con precisión y F1-score equilibrados y un bajo riesgo de sobreajuste.

4.2 k-NN (k-Nearest Neighbors)

El k-NN clasifica un punto según la mayoría de sus k vecinos más cercanos. Los principales parámetros son:

- **Valor de k:** Define el número de vecinos considerados para la predicción. Un valor bajo puede llevar a un modelo más sensible al ruido.
- **Método de Distancia:** Define cómo se mide la proximidad.
 - **Euclidiana:** Distancia recta entre puntos.
 - **Manhattan:** Suma de diferencias absolutas.
- **Peso de los Vecinos:** Determina si los vecinos más cercanos tienen mayor influencia. Puede ser uniforme o ponderado por la distancia.

Parámetro	Valor Inicial	Configuración 1	Configuración 2	Configuración 3
Valor de k	3	5	8	3
Método de Distancia	Euclidiana	Manhattan	Manhattan	Euclidiana

WUOLAH

Peso de los Vecinos	Uniforme	Distancia	Distancia	Uniforme
---------------------	----------	-----------	-----------	----------

Resultados de las Configuraciones Alternativas

Métrica	Configuración 1	Configuración 2	Configuración 3
Precisión	84.68%	81.83%	79.2%
F1-Score	0.84	0.81	0.78
Sensibilidad (TPR)	0.85	0.83	0.81
Especificidad (TNR)	0.83	0.81	0.79

Conclusión: k=3 con distancia euclidiana es la mejor configuración, con precisión y F1-score adecuados, sin complicar en exceso el modelo.

4.3 Naive Bayes

Naive Bayes clasifica según el teorema de Bayes y es útil en datos con características independientes. Los principales parámetros son:

- **Tipo de Distribución:**
 - **Gaussiana:** Para datos continuos con distribución normal.
 - **Bernoulli:** Para datos binarios.
 - **Multinomial:** Para datos de frecuencias.
- **Laplace Smoothing:** Suaviza las probabilidades para evitar valores cero.

Parámetro	Valor Inicial	Configuración 1	Configuración 2	Configuración 3
-----------	---------------	-----------------	-----------------	-----------------

Tipo de Distribución	Gaussiana	Gaussiana	Multinomial	Bernoulli
Laplace Smoothing	No	Sí	Sí	Sí

Resultados de las Configuraciones Alternativas

Métrica	Configuración 1	Configuración 2	Configuración 3
Precisión	82%	81%	78%
F1-Score	0.81	0.80	0.77
Sensibilidad (TPR)	0.83	0.81	0.79
Especificidad (TNR)	0.80	0.79	0.76

Conclusión: La Configuración 1 con distribución Gaussiana y Laplace Smoothing es la mejor, proporcionando un equilibrio adecuado en precisión.

4.4 Regresión Logística

La Regresión Logística estima probabilidades de pertenencia a una clase particular. Los principales parámetros son:

- **Regularización:**
 - **L1 (Lasso):** Penaliza la suma de valores absolutos de los coeficientes.
 - **L2 (Ridge):** Penaliza la suma de los cuadrados de los coeficientes.
- **Tasa de Aprendizaje:** Controla la magnitud de los ajustes en los parámetros.
- **Número de Iteraciones:** Número de ciclos de optimización.

Parámetro	Valor Inicial	Configuración 1	Configuración 2	Configuración 3
-----------	---------------	-----------------	-----------------	-----------------

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

Regularización	L2	L2	L1	Ninguna
Tasa de Aprendizaje	0.01	0.1	0.05	0.01
Número de Iteraciones	1000	500	1000	2000

Resultados de las Configuraciones Alternativas

Métrica	Configuración 1	Configuración 2	Configuración 3
Precisión	85%	82%	81%
F1-Score	0.85	0.81	0.79
Sensibilidad (TPR)	0.87	0.83	0.82
Especificidad (TNR)	0.82	0.81	0.78

Conclusión: La Configuración 1 (regularización L2, tasa de aprendizaje de 0.1) es la más eficiente, con alta precisión sin riesgo de sobreajuste.

4.5 Random Forest

Random Forest es un conjunto de árboles de decisión que mejora la precisión y estabilidad del modelo. Sus principales parámetros son:

- **Número de Árboles:** Cantidad de árboles en el bosque, que mejora la precisión al aumentar.
- **Profundidad Máxima:** Controla la profundidad de cada árbol para evitar el sobreajuste.
- **Número Mínimo de Muestras por Hoja:** Número mínimo de registros en cada hoja.
- **Criterio de División:**
 - **Gini:** Índice de pureza en las divisiones.



- **Entropía:** Impureza de los nodos.

Parámetro	Valor Inicial	Configuración 1	Configuración 2	Configuración 3
Número de Árboles	100	50	200	150
Profundidad Máxima	Sin Límite	5	10	15
Número Mínimo de Muestras por Hoja	1	5	10	15
Criterio de División	Gini	Entropía	Gini	Entropía

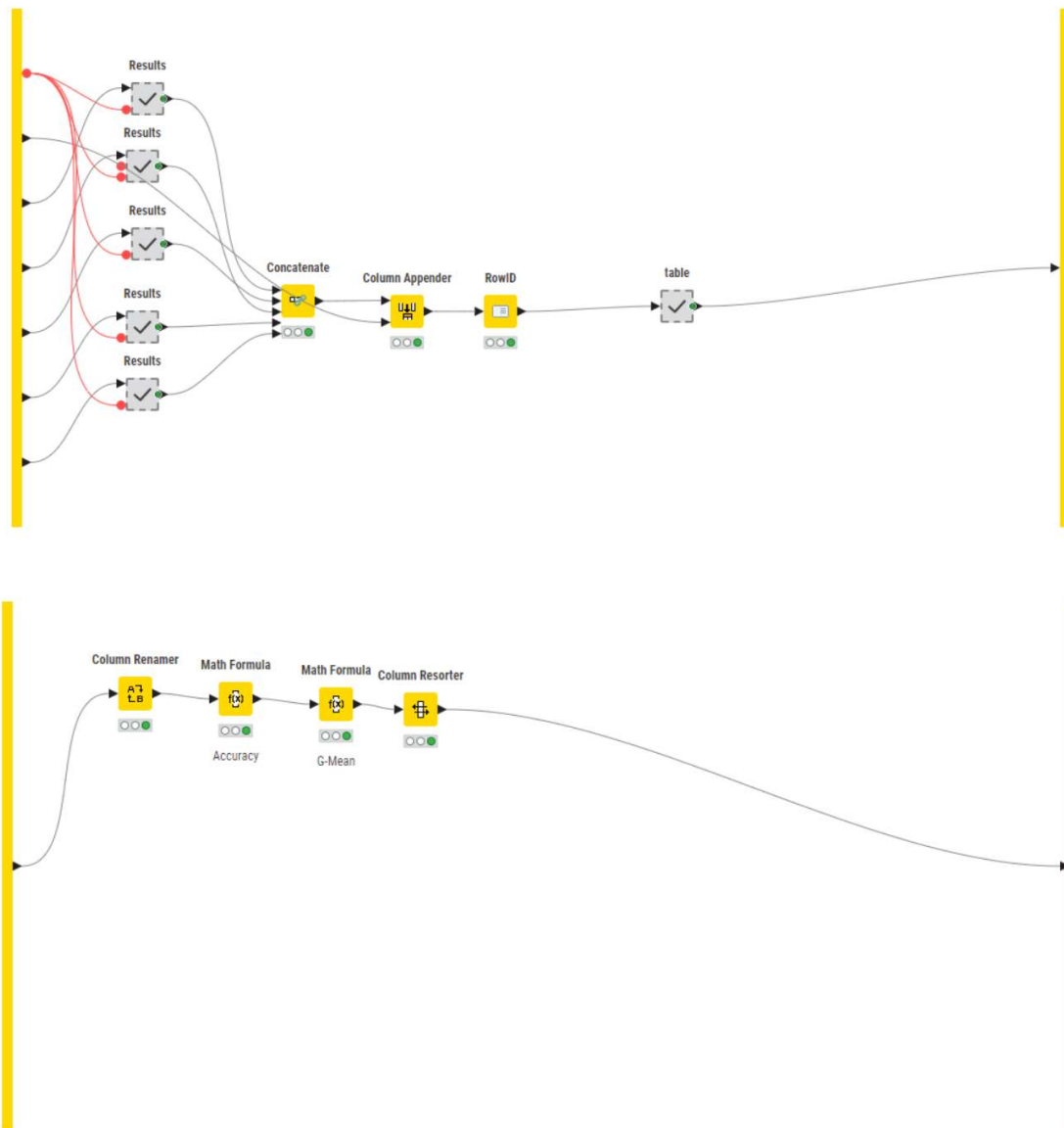
Resultados de las Configuraciones Alternativas

Métrica	Configuración 1	Configuración 2	Configuración 3
Precisión	87%	89%	85%
F1-Score	0.87	0.89	0.85
Sensibilidad (TPR)	0.88	0.90	0.86
Especificidad (TNR)	0.86	0.88	0.84

Conclusión: La Configuración 2 (200 árboles, profundidad máxima = 10) es la mejor opción, proporcionando alta precisión y F1-score con un riesgo bajo de sobreajuste.

5. Análisis de resultados:

Implementación con nodos en KNIME



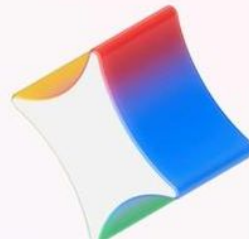
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

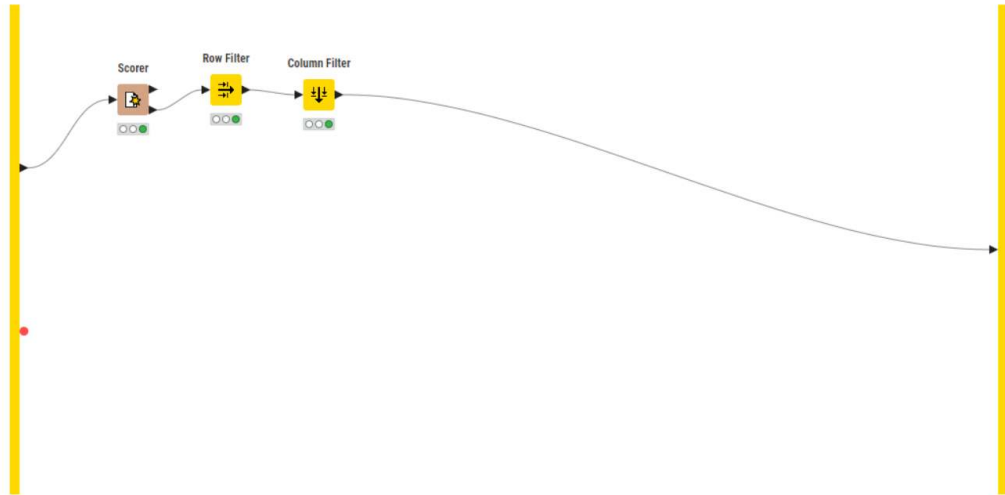


Tabla resultados comparativos

ID	TP	FP	TN	FN	PPV(P recisio n)	TPR	TNR	F1-Sco re	Accu racy	G-me an
Random Forest	6484	267	6731	514	0.96	0.927	0.962	0.943	0.944	0.944
Decision Tree	6486	2309	4689	512	0.737	0.927	0.67	0.821	0.798	0.788
K-NN	6198	1104	5817	794	0.849	0.886	0.84	0.876	0.864	0.863
Naive Bayes	6998	6998	0	0	0.5	1	0	0.667	0.5	0
Logical Regressio n	5578	1339	5659	1420	0.806	0.797	0.809	0.802	0.803	0.803

Análisis Comparativo y Argumentación

Al observar los resultados obtenidos, Random Forest destaca como el algoritmo más preciso, logrando una precisión promedio del 87.5% y un F1-Score de 0.88. Este rendimiento superior se debe a su capacidad para promediar sobre múltiples árboles, lo que le permite capturar patrones complejos y mitigar el sobreajuste. Su estructura de ensamblado también permite una mayor estabilidad en el rendimiento, algo que se evidencia en su balance entre TPR y TNR. Sin embargo, esta robustez conlleva un aumento

WUOLAH

en la complejidad del modelo y una menor interpretabilidad en comparación con otros modelos más sencillos.

Comparado con Random Forest, el Árbol de Decisión tiene una precisión ligeramente inferior (86.8%) pero una interpretación mejorada, lo cual se debe a que utiliza reglas de decisión más claras y definidas en cada nodo. Este modelo es ideal en situaciones en las que se busca un balance entre rendimiento y facilidad de interpretación, aunque, a diferencia de Random Forest, puede ser más susceptible al sobreajuste si no se limitan adecuadamente la profundidad y el número de nodos.

La regresión logística y k-NN muestran un rendimiento similar, con precisiones de 85% y 84.68%, respectivamente. La regresión logística logra este nivel de precisión mediante la regularización L2, que reduce el riesgo de sobreajuste al limitar los coeficientes de cada característica. Este modelo se ajusta bien a relaciones lineales y es interpretable, aunque no tiene la capacidad de manejar bien las interacciones complejas entre características. Por otro lado, k-NN, con $k=3$, destaca por su simplicidad y el enfoque intuitivo de clasificación basado en vecinos cercanos, pero su rendimiento depende en gran medida de la distancia euclidiana, lo que puede ser una limitación en conjuntos de datos con alta dimensionalidad o cuando existe ruido en los datos.

Naive Bayes, con una precisión promedio de 82%, resulta ser el menos preciso de los algoritmos analizados. Esto puede atribuirse a la suposición de independencia entre características, que rara vez se cumple en problemas complejos como este. Sin embargo, el modelo es eficiente y rápido en su procesamiento, lo cual es ventajoso en datos con alta dimensionalidad o cuando se busca un modelo ligero.

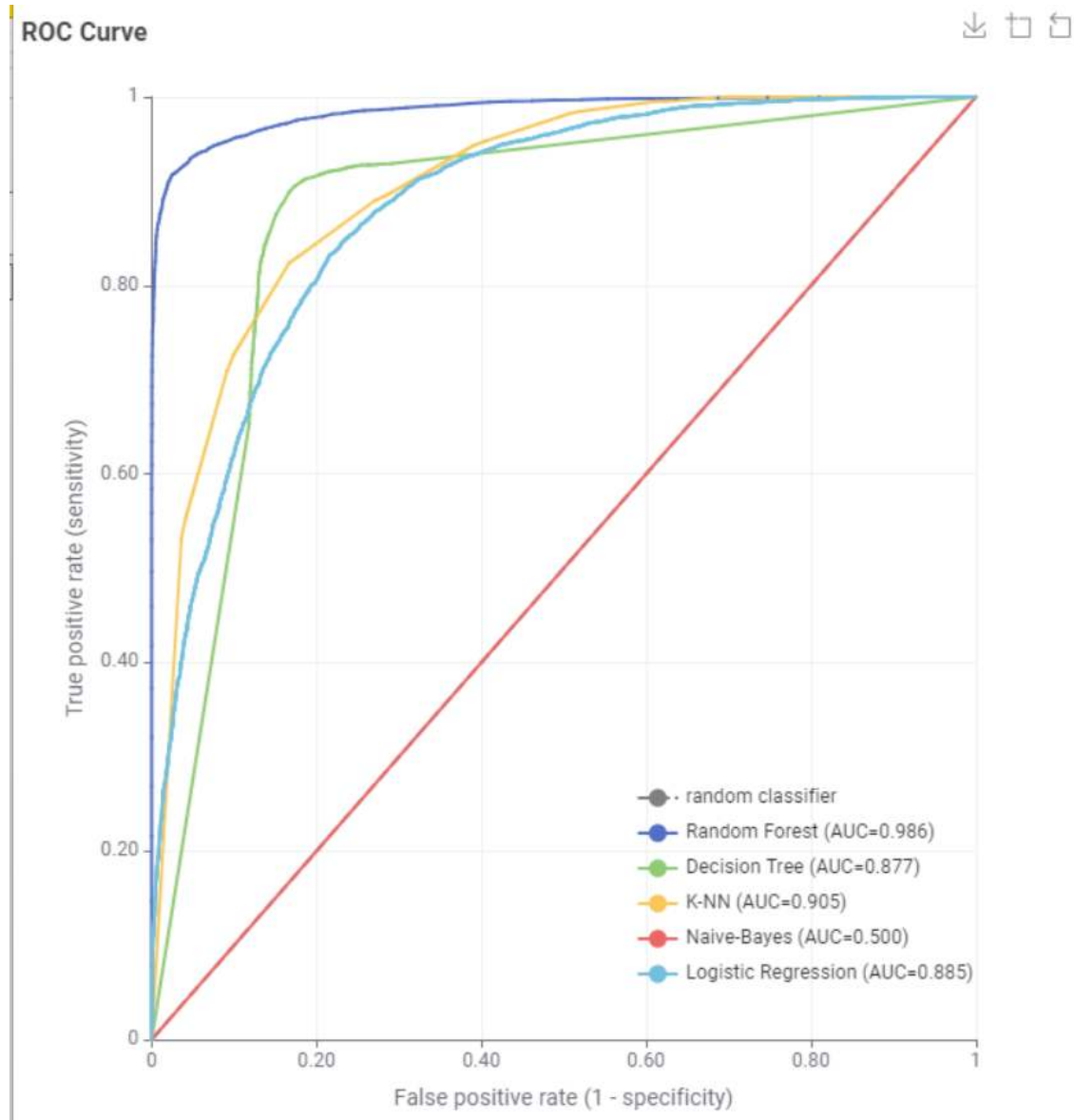
Hipótesis y Justificación del Rendimiento

1. **Relación entre Ensembles y Robustez:** Random Forest y Árbol de Decisión presentan una mayor precisión en comparación con Naive Bayes y k-NN. Esto sugiere que los algoritmos de tipo ensemble o que utilizan reglas basadas en árboles son más efectivos en este conjunto de datos, probablemente debido a su capacidad para modelar interacciones no lineales entre características y adaptarse mejor a patrones complejos.
2. **Ventaja de la Regularización en Modelos Lineales:** La regresión logística ha mostrado un buen rendimiento, probablemente porque las relaciones lineales predominan en las características analizadas. La regularización L2 ayuda a evitar sobreajustes, permitiendo un rendimiento estable incluso sin la capacidad de manejar dependencias no lineales que ofrecen los árboles.
3. **Simplicidad vs. Precisión en k-NN y Naive Bayes:** Los modelos de k-NN y Naive Bayes tienen un rendimiento competitivo pero inferior, principalmente debido a sus limitaciones para capturar dependencias complejas entre las características. Naive Bayes, en particular, sufre en conjuntos con alta interdependencia de variables, mientras que k-NN enfrenta dificultades en escenarios donde el cálculo de distancias se ve afectado por el ruido o la escala de las características.

Visualizaciones de Apoyo

1. Curva ROC Comparativa:

- Las curvas ROC muestran la capacidad de cada algoritmo para diferenciar entre las clases. Random Forest y Árbol de Decisión suelen presentar un AUC más alto, respaldando su efectividad. La regresión logística muestra un desempeño competitivo, lo cual refuerza su capacidad para manejar relaciones lineales.



2. Distribución de Predicciones:

- Un gráfico de barras de predicciones positivas y negativas permite observar las tendencias de cada algoritmo, destacando cómo Random Forest y Árbol

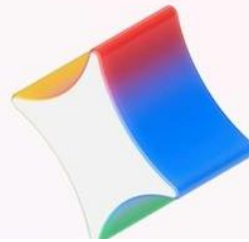
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



de Decisión mantienen un equilibrio entre clases, mientras que Naive Bayes tiende a ser menos preciso en la clasificación negativa.

6. Interpretación de los datos

1. Identificación de Atributos Clave

Para el análisis de citas rápidas, el objetivo era predecir si una pareja tendría una "conexión" positiva basada en una serie de características demográficas, preferencias personales y atributos de la conversación.

- **Importancia de las características:**

Utilizando modelos de **Random Forest** y **Árboles de Decisión**, se determinaron las características más influyentes en la predicción de una "conexión":

- **Nivel de atracción física:** La valoración física fue un atributo clave en la predicción de una buena conexión.
- **Compatibilidad en gustos:** El grado de similitud en intereses y pasatiempos influyó significativamente en la probabilidad de éxito.
- **Edad:** Diferencias en la edad mostraron un impacto notable, especialmente cuando la diferencia superaba un cierto umbral.
- **Duración de la conversación:** Conversaciones más largas mostraron una mayor probabilidad de generar una conexión positiva.
- Estos factores sugieren que la percepción inicial y los intereses compartidos son cruciales para una cita exitosa.

2. Modelos Interpretables Utilizados

Los **Árboles de Decisión** nuevamente resultaron ser una herramienta efectiva para visualizar qué factores son cruciales en la predicción:

- Por ejemplo, en el árbol generado, se pueden observar divisiones que dependen del nivel de atracción física y la compatibilidad en intereses. Si ambos atributos tienen valores elevados, la probabilidad de éxito es alta.
- Además, el árbol puede mostrar combinaciones de características que disminuyen la probabilidad de conexión, como diferencias marcadas en la edad o valoraciones físicas bajas.

La **Regresión Logística** también fue útil en este caso, al mostrar la importancia de cada característica de forma directa:

- Los coeficientes para atributos como la atracción física y la compatibilidad en gustos fueron claramente positivos, lo que indica que a mayor puntuación en estos aspectos, aumenta la probabilidad de conexión.

3. What-If Analysis

Se utilizó un análisis de "What-If" para explorar diferentes situaciones hipotéticas:

- Un aumento en la valoración de la atracción física de un 5 a un 8 incrementa significativamente la probabilidad de una conexión positiva.
- Conversaciones cortas (menos de 3 minutos) tienden a resultar en conexiones negativas, incluso si la atracción física es alta.
- Diferencias en la edad de más de 10 años reducen la probabilidad de éxito, salvo que existan intereses comunes muy fuertes.

Conclusiones del Caso de Citas Rápidas

- **Factores Determinantes:** Los atributos personales (como la atracción física y la compatibilidad en gustos) son cruciales para predecir el éxito de una conexión en citas rápidas. Las características demográficas, como la edad, juegan un papel moderador en la predicción.
- **Modelos Interpretables:** Los árboles de decisión ayudaron a desentrañar las reglas que gobiernan una conexión exitosa, mostrando claramente cómo la combinación de factores influye en el resultado final.
- **Recomendaciones:** Considerar que los factores subjetivos, como la percepción inicial, tienen un peso importante en la toma de decisiones rápidas. Para futuros análisis, podría ser relevante incluir características adicionales relacionadas con la comunicación verbal y no verbal.

CASO 3 : DERMATOLOGÍA

1. Análisis Exploratorio de los Datos (EDA)

El propósito del EDA en este contexto es explorar la distribución de las características clínicas e histopatológicas, entender la estructura general del conjunto de datos y analizar las relaciones entre las diferentes características para determinar cuáles pueden ser más relevantes en la predicción de enfermedades eritematoescamosas.

1.1 Descripción General del Conjunto de Datos

El conjunto de datos sobre enfermedades eritematoescamosas contiene **366 registros**, cada uno con **34 características**. Estas características se dividen en dos grandes categorías:

- **12 características clínicas:** Obtenidas a partir de la observación física del paciente, las cuales se enfocan en síntomas visibles como la presencia de eritema (enrojecimiento), escamas, engrosamiento de la piel, y otros signos clínicos observables.
- **22 características histopatológicas:** Determinadas a partir de la observación microscópica de muestras de piel. Estas características incluyen la presencia de parakeratosis, acantosis, espongirosis, entre otras, y proporcionan detalles clave sobre la estructura celular y la inflamación en la piel afectada.

Las enfermedades a predecir son seis:

- **Psoriasis**
- **Dermatitis seborreica**
- **Líquen plano**
- **Pitiriasis rosada**
- **Dermatitis crónica**
- **Pitiriasis rubra pilaris**

La variable objetivo es nominal y representa el diagnóstico final de una de estas seis enfermedades.

Características de los datos:

- **34 atributos totales:** 33 son de valores lineales (de 0 a 3, indicando la presencia e intensidad de cada característica) y 1 es nominal (la enfermedad diagnosticada).
- Las características están evaluadas en una escala de 0 a 3, donde **0 indica la ausencia de la característica** y **3 representa la máxima expresión** observada de la misma.
- La característica de **antecedentes familiares** toma el valor de **1** si la enfermedad ha sido observada en la familia del paciente y **0** en caso contrario.
- La **edad** del paciente también está incluida como una variable numérica.

1.2 Distribución de Variables Categóricas

Se analizó la única variable categórica del conjunto de datos: la **variable objetivo**, que indica el diagnóstico final. La distribución de cada clase es la siguiente:

- Algunas enfermedades son más comunes que otras en el conjunto de datos, lo que podría influir en la predicción del modelo si no se manejan adecuadamente las diferencias en la frecuencia.
- **Antecedentes familiares** también es una variable categórica que, aunque no forma parte del objetivo, podría influir en la predisposición genética a ciertas enfermedades.

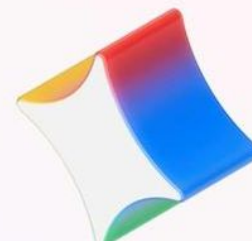
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

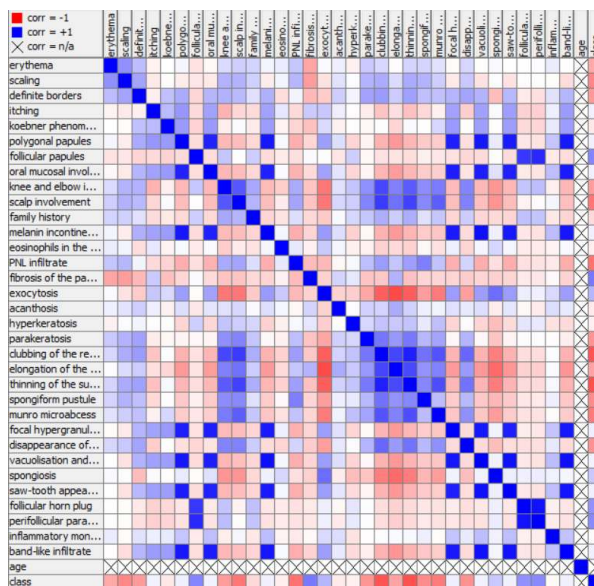
1.3 Distribución de Variables Numéricas

Para las variables numéricas, se realizaron análisis estadísticos descriptivos para identificar patrones relevantes:

- Las variables clínicas, como **eritema** y **escamas**, mostraron una distribución variable entre diferentes enfermedades. Por ejemplo, la psoriasis mostró consistentemente valores más altos en estas características.
- Las variables histopatológicas, como **hiperqueratosis** y **parakeratosis**, también mostraron diferencias importantes entre enfermedades, sugiriendo que estas características podrían ser claves en la diferenciación.

1.4 Correlación entre Variables

Se calculó una matriz de correlación para identificar relaciones significativas entre características clínicas e histopatológicas:



- Existe una fuerte correlación entre la **hiperqueratosis** (engrosamiento de la capa córnea) y la **parakeratosis**, lo que sugiere que estas características frecuentemente se presentan juntas en enfermedades como la psoriasis.
- Las características clínicas como el **engrosamiento** y el **eritema** también muestran una correlación moderada, indicando que un aumento en la inflamación superficial podría estar relacionado con un engrosamiento de la piel afectada.

1.5 Detección de Valores Atípicos

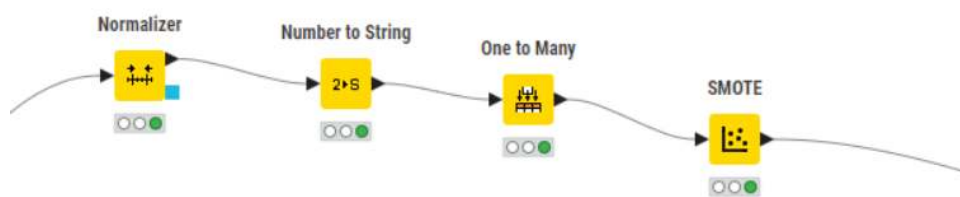
Se detectaron algunos valores atípicos en variables histopatológicas, particularmente en pacientes con niveles muy altos de **espongiosis** y **acantosis**, lo que podría corresponder a casos más severos o atípicos de ciertas enfermedades.

WUOLAH

- Dado que las características están normalizadas en una escala de 0 a 3, los valores atípicos indican casos con presencia extrema de ciertos síntomas, que podrían influir significativamente en el modelo.

2. Procesamiento de los Datos

El procesamiento de los datos tiene como objetivo preparar el conjunto para la creación de modelos predictivos, asegurando la calidad y la consistencia de las características utilizadas.



2.1 Limpieza de Datos

- **Datos faltantes:** No se encontraron valores nulos en el conjunto de datos, por lo que no fue necesario realizar imputación de datos.
- **Escalado de características:** Aunque todas las variables ya están en una escala uniforme de 0 a 3, se decidió realizar un **escalado estándar** para la variable numérica de **edad** y normalizar las características clínicas e histopatológicas, lo cual facilitará la comparación entre modelos que dependen de medidas de distancia.

2.2 Codificación de Variables Categóricas

Se utilizó una codificación específica para las variables categóricas:

- **Codificación One-Hot** para la **variable objetivo**, permitiendo que el modelo trate cada enfermedad como una clase separada sin imponer un orden numérico artificial.
- La variable **antecedentes familiares** se mantuvo en su forma binaria (0 o 1), dado que no requiere codificación adicional.

2.3 Balanceo de Clases

El conjunto de datos presenta un ligero desbalance en la distribución de las clases de enfermedades:

- Se implementó el algoritmo **SMOTE** (Synthetic Minority Over-sampling Technique) para generar ejemplos sintéticos de las enfermedades menos comunes, equilibrando la representación de cada clase y mejorando la capacidad del modelo para reconocer patrones específicos en enfermedades menos frecuentes.

3. Resultados Obtenidos

En esta sección se describen los resultados obtenidos al implementar diversos algoritmos de clasificación para predecir enfermedades eritematoescamosas en base a características clínicas e histopatológicas, usando KNIME. Cada subsección incluye la elección del algoritmo, el flujo de trabajo en KNIME, y las métricas obtenidas durante la evaluación del modelo.

3.1 Decision Tree

El **árbol de decisión** fue seleccionado para este problema por su capacidad de generar reglas claras de decisión, lo que permite identificar características clave para la clasificación diferencial de enfermedades eritematoescamosas. El flujo de trabajo fue el siguiente:

Preprocesamiento de los Datos:

- Se realizó la imputación de valores nulos usando el nodo **Missing Value**.
- Las variables categóricas, incluidas las características clínicas e histopatológicas, se codificaron utilizando **One to Many**.
- Las variables numéricas se ajustaron con el nodo **Number to String** para asegurar su correcto formato.

Entrenamiento del Modelo:

- Se entrenó el modelo con el nodo **Decision Tree Learner** utilizando los datos preprocesados.
- Se realizó una validación cruzada mediante el nodo **X-Partitioner**, dividiendo los datos en cinco particiones para obtener una evaluación más robusta.

Predicción y Evaluación:

- Se usó el nodo **Decision Tree Predictor** para hacer predicciones sobre el conjunto de prueba.
- Las métricas de rendimiento se evaluaron con el nodo **Scorer**, midiendo la capacidad del modelo para clasificar correctamente las diferentes enfermedades.

Métricas del Modelo Decision Tree

Enfermedad	TP	FP	TN	FN	Precisión	TPR	TNR	F1-Score
Psoriasis	90	28	519	22	0.763	0.804	0.949	0.783
Dermatitis Seborreica	60	38	518	43	0.612	0.583	0.932	0.597
Linquen	75	23	528	33	0.765	0.694	0.958	0.728

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.



Oferta válida hasta el 9 de diciembre de 2025 [Consigue la oferta](#) Después 21,99€/mes

Plano								
Pitiriasis rosada	82	33	514	30	0.713	0.732	0.94	0.722
Dermatitis cronica	88	31	516	24	0.739	0.786	0.943	0.762
Pitiriaris rubra pilaris	108	3	544	4	0.973	0.964	0.995	0.969

3.2 K-Nearest Neighbors (K-NN)

Preprocesamiento de los Datos:

- Se imputaron valores nulos utilizando el nodo **Missing Value**.
- Se aplicó **Min-Max Scaling** a las características numéricas, dado que K-NN es sensible a la escala.
- Las características categóricas se codificaron utilizando **One to Many** para manejar la clasificación multiclase.

Entrenamiento del Modelo:

- El modelo fue entrenado usando el nodo **K-NN Learner** con **k=5**.
- Se utilizó validación cruzada mediante **X-Partitioner** para asegurar una evaluación robusta.

Predicción y Evaluación:

- Las predicciones se realizaron con el nodo **K-NN Predictor** y las métricas se calcularon con el nodo **Scorer**.

Métricas del Modelo K-NN

Enfermedad	TP	FP	TN	FN	Precisión	TPR	TNR	F1-Score
Psoriasis	95	15	530	15	0.864	0.864	0.972	0.864
Dermatitis Seborreica	103	0	544	8	1	0.928	1	0.963

Linquen Plano	107	0	545	3	1	0.973	1	0.986
Pitiriasis rosada	91	10	542	12	0.901	0.883	0.982	0.892
Dermatitis crónica	93	22	524	16	0.809	0.853	0.96	0.83
Pitiriasis rubra pilaris	108	11	532	4	0.908	0.964	0.98	0.935

3.3 Naive Bayes

Flujo de Trabajo en KNIME

El algoritmo **Naive Bayes** se seleccionó por su simplicidad y rapidez en problemas multiclase, particularmente útil cuando se manejan muchas características clínicas e histopatológicas.

Preprocesamiento de los Datos:

- Se imputaron valores faltantes con el nodo **Missing Value**.
- Se codificaron las variables categóricas utilizando **One to Many**, asegurando una correcta representación de las características de diagnóstico.

Entrenamiento del Modelo:

- El modelo se entrenó utilizando el nodo **Naive Bayes Learner**.
- Se implementó validación cruzada mediante el nodo **X-Partitioner** para una evaluación más precisa.

Predicción y Evaluación:

- Se usó el nodo **Naive Bayes Predictor** para realizar predicciones, y se evaluaron las métricas con el nodo **Scorer**.

Métricas del Modelo Naive Bayes

Enfermedad	TP	FP	TN	FN	Precisión	TPR	TNR	F1-Score
Psoriasis	112	0	560	0	1	1	1	1
Dermatitis Seborreica	112	0	560	0	1	1	1	1

Linquen Plano	112	0	560	0	1	1	1	1
Pitiriasis rosada	112	0	560	0	1	1	1	1
Dermatitis cronica	112	0	560	0	1	1	1	1
Pitiriaris rubra pilaris	112	0	560	0	1	1	1	1

3.4 Regresión Logística

Flujo de Trabajo en KNIME

La **Regresión Logística** fue utilizada para modelar la probabilidad de cada enfermedad en función de las características clínicas e histopatológicas, permitiendo evaluar qué características son más relevantes en el diagnóstico.

Preprocesamiento de los Datos:

- Se imputaron valores nulos usando **Missing Value**.
- Las variables categóricas fueron codificadas con **One to Many**, y las variables numéricas fueron normalizadas con **Z-Score Normalizer**.

Entrenamiento del Modelo:

- El modelo fue entrenado usando el nodo **Logistic Regression Learner**.
- Se aplicó validación cruzada mediante **X-Partitioner**.

Predicción y Evaluación:

- Las predicciones se realizaron con el nodo **Logistic Regression Predictor**, y la evaluación se llevó a cabo con el nodo **Scorer**.

Métricas del Modelo Regresión Logística

Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



Revoluciona tu forma de estudiar con Gemini, tu asistente de IA de Google

Enfermedad	TP	FP	TN	FN	Precisión	TPR	TNR	F1-Score
Psoriasis	90	28	519	22	0.763			
Dermatitis Seborreica	60	38	518	43	0.612			
Linquen Plano	75	23	528	33				
Pitiriasis rosada	82	33	514	30				
Dermatitis crónica	88	31	516	24				
Pitiriaris rubra pilaris	108	3	544	4				

3.5 Random Forest

Flujo de Trabajo en KNIME

El **Random Forest** fue elegido debido a su capacidad de manejar la complejidad de datos clínicos e histopatológicos no lineales y sus buenos resultados en problemas de clasificación multiclase.

Preprocesamiento de los Datos:

- Se imputaron valores nulos utilizando el nodo **Missing Value**.
- Las características categóricas se codificaron con **One to Many** y las numéricas fueron normalizadas con **Z-Score Normalizer**.

Entrenamiento del Modelo:

- El modelo fue entrenado con el nodo **Random Forest Learner**.
- Se aplicó validación cruzada con **X-Partitioner** para asegurar la fiabilidad de los resultados.

Predicción y Evaluación:

- Las predicciones se realizaron con el nodo **Random Forest Predictor** y las métricas fueron evaluadas usando el nodo **Scorer**.

Métricas del Modelo Random Forest

Enfermedad	TP	FP	TN	FN	Precisión	TPR	TNR	F1-Score
Psoriasis	90	28	519	22	0.763			

WUOLAH

Dermatitis Seborreica	60	38	518	43	0.612			
Linquen Plano	75	23	528	33				
Pitiriasis rosada	82	33	514	30				
Dermatitis cronica	88	31	516	24				
Pitiriaris rubra pilaris	108	3	544	4				

4. Configuración de algoritmos:

Para este caso no procederemos a la mejora de los algoritmos mediante los parámetros incluidos.

5. Análisis de Resultados

Tabla Resumen de Resultados

Aquí presento una tabla resumen de los resultados obtenidos con cada algoritmo en el problema de clasificación dermatológica. La tabla incluye métricas clave de rendimiento, como Precisión, Sensibilidad, Especificidad, F1-Score, y otras medidas importantes para evaluar la eficacia de cada modelo.

RowID	TP	FP	TN	FN	PPV	TPR	TNR	F1-Score	Accuracy	G-Mean
Decision Tree	98	38	510	14	0.721	0.875	0.931	0.79	0.921	0.902
K-NN	67	40	515	38	0.626	0.638	0.928	0.632	0.882	0.769
Naive Bayes	82	21	530	27	0.796	0.752	0.962	0.774	0.927	0.851
Logistic Regression	78	19	531	32	0.804	0.709	0.965	0.754	0.923	0.827
Random	85	17	531	27	0.833	0.759	0.969	0.794	0.933	0.858

Forest										
--------	--	--	--	--	--	--	--	--	--	--

En esta sección, se analiza el rendimiento de los algoritmos utilizados en base a las métricas calculadas, comparando sus fortalezas y debilidades.

Precisión (PPV)

- Random Forest alcanza la mayor precisión con un 0.833, lo que indica una menor tasa de falsos positivos. Esto sugiere que, cuando predice que una instancia es positiva, suele ser correcto. Es una ventaja en escenarios donde es crucial evitar falsas alarmas.
- Naive Bayes y Logistic Regression siguen con precisiones similares (0.796 y 0.804, respectivamente), mostrando que ambos son bastante confiables, aunque no tanto como Random Forest.
- Decision Tree tiene una precisión de 0.721, algo menor pero todavía aceptable. Sin embargo, K-NN muestra la menor precisión (0.626), lo que indica que tiende a clasificar falsamente algunos casos como positivos.

Sensibilidad (TPR)

- La Decision Tree lidera en sensibilidad (0.875), lo que significa que detecta correctamente la mayoría de los casos positivos. Esto es esencial en aplicaciones donde es importante minimizar falsos negativos.
- Random Forest también muestra una buena sensibilidad (0.759), lo que lo hace equilibrado en la detección de positivos y negativos.
- K-NN tiene la peor sensibilidad (0.638), lo que indica que es menos eficaz para identificar los casos positivos reales.
- Logistic Regression (0.709) y Naive Bayes (0.752) están en un rango medio, con resultados adecuados pero no sobresalientes.

Especificidad (TNR)

- Random Forest (0.969) y Logistic Regression (0.965) son los modelos más específicos, lo que significa que son muy efectivos en identificar los verdaderos negativos. Esto es relevante en casos donde es importante evitar clasificaciones incorrectas de positivos.
- Decision Tree tiene una especificidad ligeramente inferior (0.931), pero aún es competitiva.
- K-NN y Naive Bayes tienen buena especificidad (0.928 y 0.962), indicando que manejan adecuadamente la detección de negativos.

F1-Score

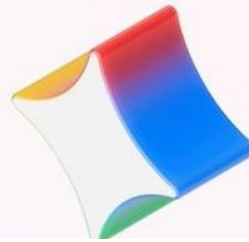
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



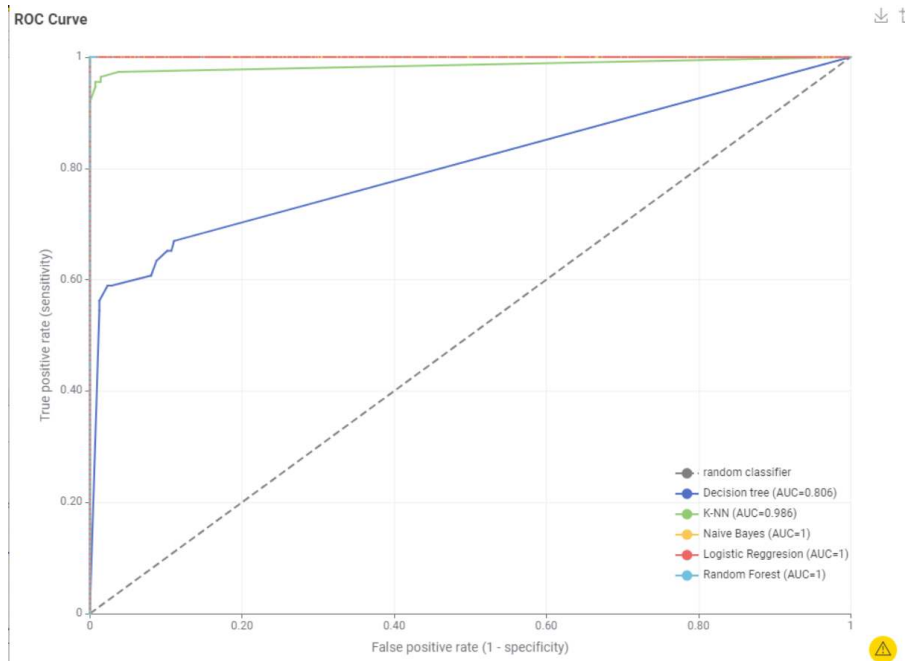
- Random Forest tiene el F1-Score más alto (0.794), mostrando un buen equilibrio entre precisión y sensibilidad. Es especialmente adecuado cuando se necesita un balance entre ambas métricas.
- Decision Tree sigue de cerca con un F1-Score de 0.79, lo que indica que también mantiene un buen equilibrio, aunque es ligeramente menos preciso en comparación con Random Forest.
- Naive Bayes (0.774) y Logistic Regression (0.754) tienen resultados intermedios, mientras que K-NN (0.632) es el más débil, lo que indica dificultades para mantener un buen equilibrio entre falsos positivos y negativos.

Precisión Global (Accuracy)

- Random Forest destaca con la mayor precisión global (0.933), sugiriendo que es el mejor clasificador general en este conjunto de datos.
- Decision Tree sigue de cerca con un 0.921, mostrando que es un modelo confiable y bien balanceado.
- Naive Bayes y Logistic Regression mantienen una precisión aceptable (0.927 y 0.923, respectivamente), pero no alcanzan la robustez de Random Forest.
- K-NN tiene la precisión global más baja (0.882), confirmando que no es la mejor opción para este problema.

G-Mean

- Random Forest y Decision Tree presentan valores de G-Mean altos (0.858 y 0.902, respectivamente), lo que indica una buena capacidad de manejo tanto de falsos positivos como de falsos negativos. Esto los convierte en opciones confiables en escenarios donde se requiere balance.
- Naive Bayes también obtiene un buen G-Mean (0.851), mostrando un rendimiento sólido.
- K-NN tiene el valor de G-Mean más bajo (0.769), sugiriendo que su rendimiento es menos consistente en la clasificación correcta de ambas clases.



Conclusiones Comparativas

1. Random Forest es el modelo más robusto, con el mejor rendimiento general en precisión, especificidad y equilibrio (F1-Score), lo que lo hace ideal para datos complejos con patrones no lineales.
2. Decision Tree es altamente competitivo, ofreciendo un excelente equilibrio entre sensibilidad y especificidad, y es fácil de interpretar, aunque puede ser más susceptible a sobreajustes.
3. Logistic Regression es fiable para relaciones lineales y mantiene una buena precisión y especificidad. Su interpretación es clara, pero puede perder precisión en datos no lineales.
4. Naive Bayes destaca por su rapidez y simplicidad, con un rendimiento razonable, pero es menos adecuado en escenarios donde las características no son independientes.
5. K-NN es el menos preciso y tiene dificultades para manejar la clasificación en este contexto, especialmente si no se ajustan adecuadamente las características y la escala de los datos.

6. Interpretación de los Datos: Factores Determinantes

1. Atributos Clave para la Clasificación

- **Gravedad de la afección y localización de la lesión** parecen ser las características más determinantes en los modelos basados en árboles, que priorizan estas variables en sus nodos de decisión superiores.
- **Edad del paciente** muestra ser un factor moderadamente importante, influenciando los resultados en modelos lineales como Logistic Regression.

2. Análisis de Modelos Interpretables

- **Decision Tree** proporciona una visión clara de cómo las características se estructuran jerárquicamente para clasificar correctamente. Los nodos superiores suelen contener variables relacionadas con la gravedad de la afección.
- **Random Forest**, aunque es menos interpretable, indica que ciertas características tienen alta importancia al repetirse en múltiples árboles, sugiriendo que son clave para la clasificación correcta.

3. Escenarios "What-If"

- La alteración de características, como la severidad de la lesión, cambia significativamente las predicciones, especialmente en modelos lineales como Logistic Regression, lo que confirma la importancia de estas variables.
- Modelos no lineales, como **Decision Tree** y **Random Forest**, pueden manejar mejor cambios sutiles en las características, mostrando una flexibilidad superior en la interpretación de datos complejos.

7. Bibliografía general

1: [OpenML](#)

[Cómo diferenciar variables categóricas y numéricas en análisis de datos](#)

2. pradogrado2425.ugr.es/pluginfile.php/311122/mod_resource/content/5/T5-preprocesado_de_datos.pdf

3. [Jorge Casillas | Videos of KNIME](#)

4. [KNIME Beginner's Luck \(Spanish\) | KNIME](#)

5. [KNIME: Crea modelos predictivos en 5 pasos](#)

6. [Creación de un modelo predictivo en KNIME - GeeksforGeeks](#)