

## **Etapas 4 – Aprendizaje Supervisado**

Andrés Felipe Riveros Valero

Tutor

José Alfarr Morales Barrera

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Ingeniería de sistemas

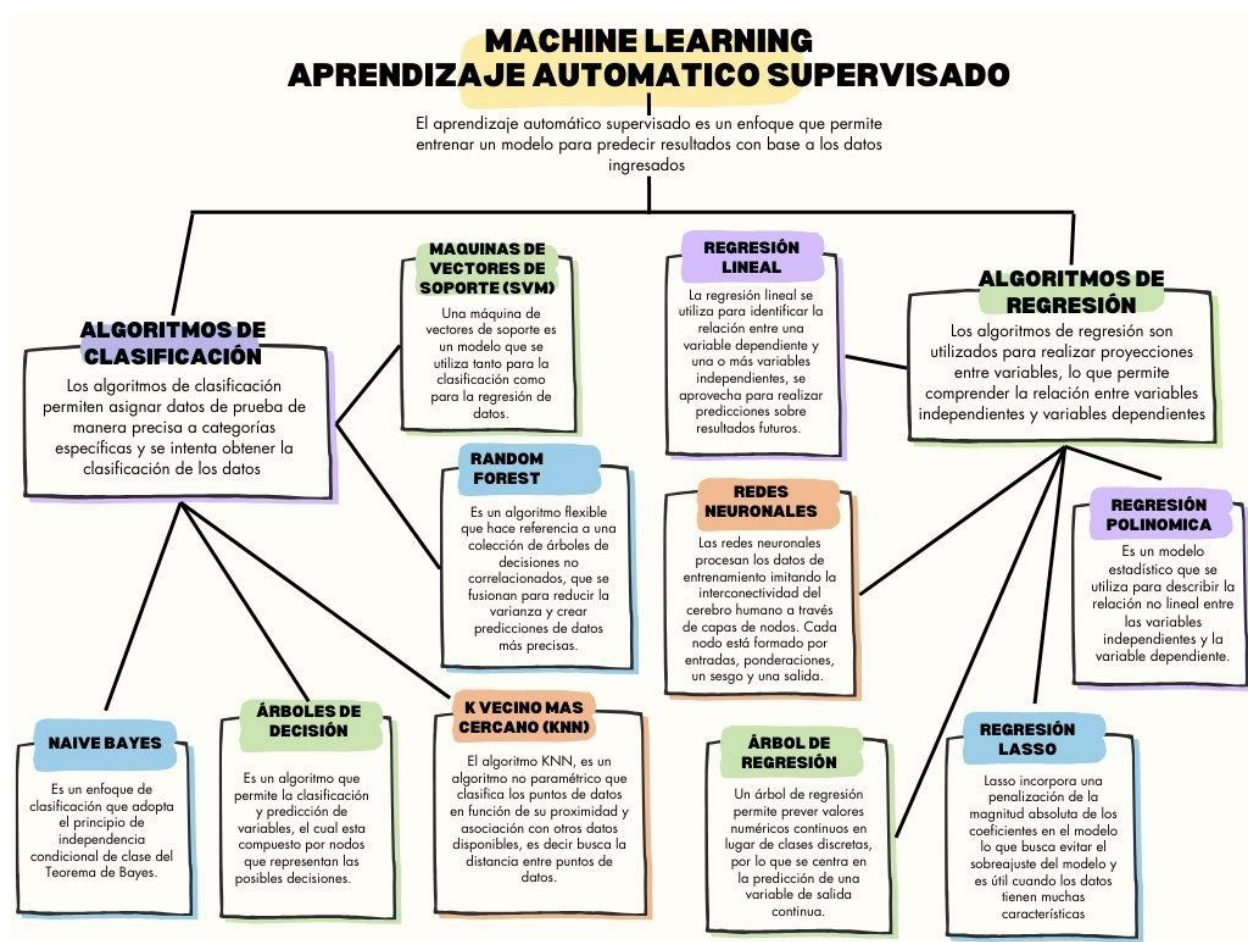
2023

## Mapa Conceptual

[https://www.canva.com/design/DAF0VoXfErY/\\_a-IK7Y0pIVTj5s4ocfOUw/view?utm\\_content=DAF0VoXfErY&utm\\_campaign=designshare&utm\\_medium=link&utm\\_source=editor](https://www.canva.com/design/DAF0VoXfErY/_a-IK7Y0pIVTj5s4ocfOUw/view?utm_content=DAF0VoXfErY&utm_campaign=designshare&utm_medium=link&utm_source=editor)

**Figura 1**

*Mapa conceptual*

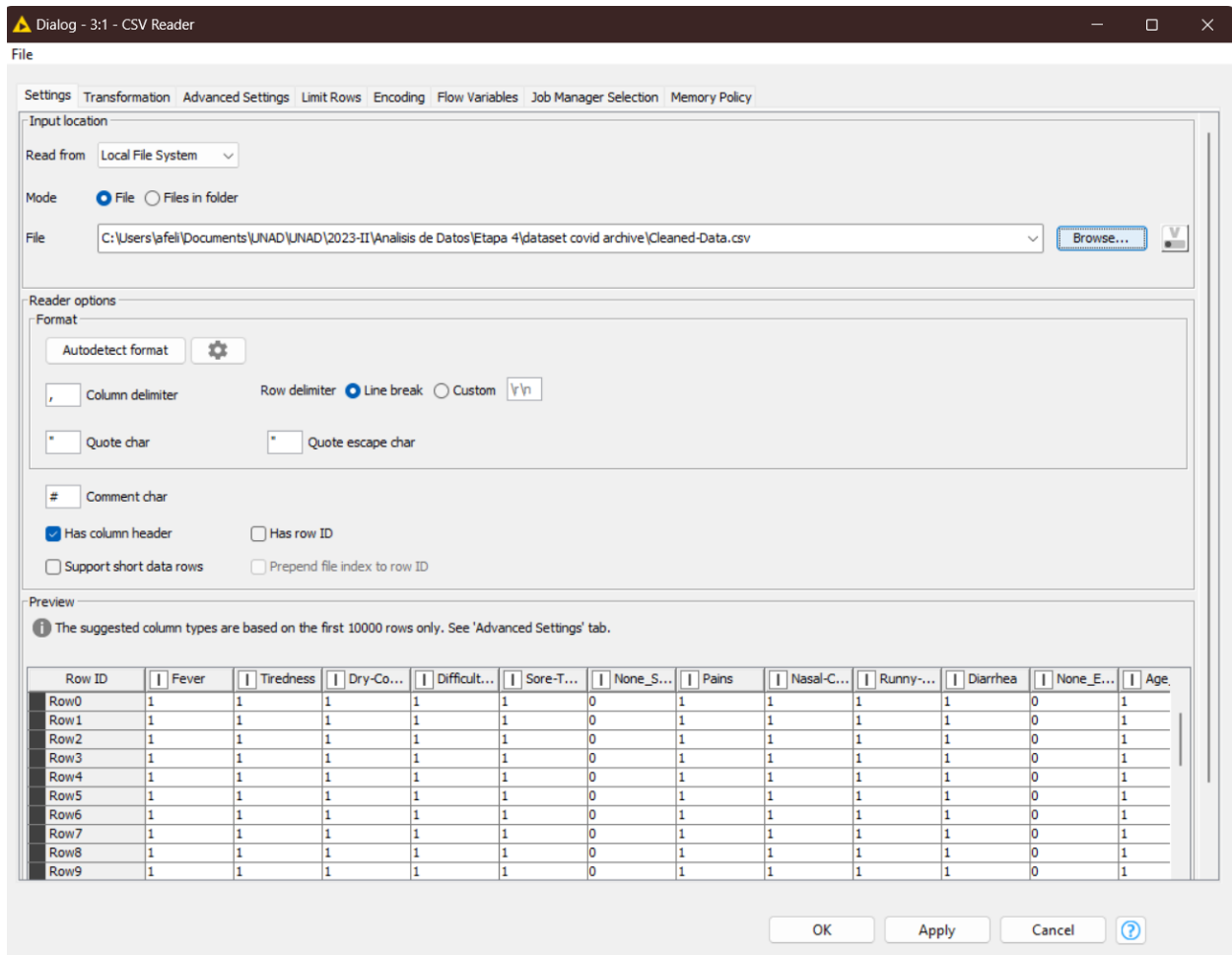


*Nota.* Imagen de autoría propia donde se muestra el mapa conceptual con los diferentes modelos de aprendizaje automático supervisado.

## Configuración del Dataset

El presente informe se basa en la dataset sobre el Covid, el cual fue tomado de <https://www.kaggle.com/datasets/iamhungundji/covid19-symptoms-checker/data> para el análisis y estudio de los casos de Covid, para esto se utilizará la herramienta Knime la cual permitirá realizar el cargue de información del dataset y a partir de él realizar el proceso para análisis de datos mediante arboles de decisión, KNN y Naive Bayes.

En primera instancia se debe configurar el dataset para el manejo de los datos, pues este dataset trae información que se puede agrupar en una sola columna por tener información similar pero separada, estos datos son “edad”, “severidad”, “contacto” y “genero”. Para esto se debe cargar el dataset en knime utilizando la opción de “CSV Reader” buscando la ruta donde se encuentra el dataset para cargar y allí se muestra previamente como se ve la información como se observa en la figura 2, se confirma y ya estará cargada la información del dataset en knime.

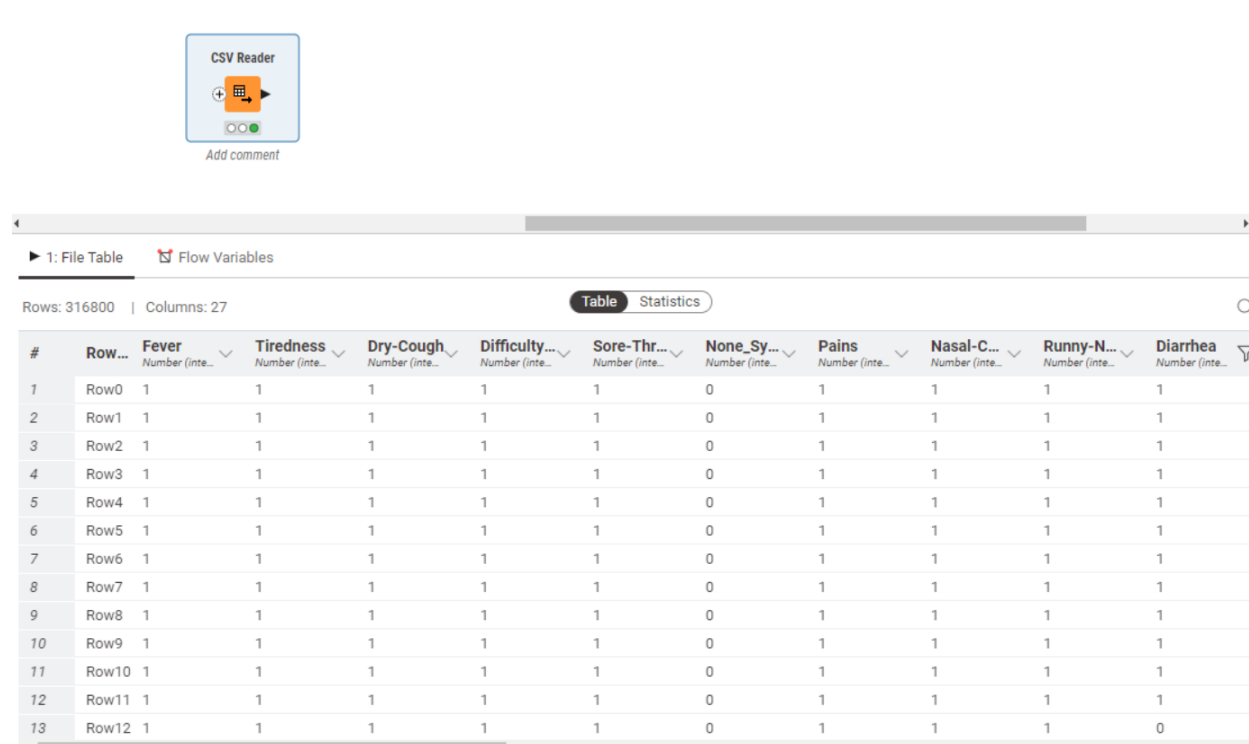
**Figura 2***Dataset Covid*

**Nota.** Imagen de autoría propia donde se muestra el cargue del archivo con la información del dataset del covid.

Después de cargada la información se ejecuta el proceso de CSV Reader y en Knime en la parte inferior se mostrará la data completa, donde además se indica el tipo de dato que tiene cada campo (number o string), como se ve en la figura 3.

**Figura 3**

*Información cargada del dataset covid*

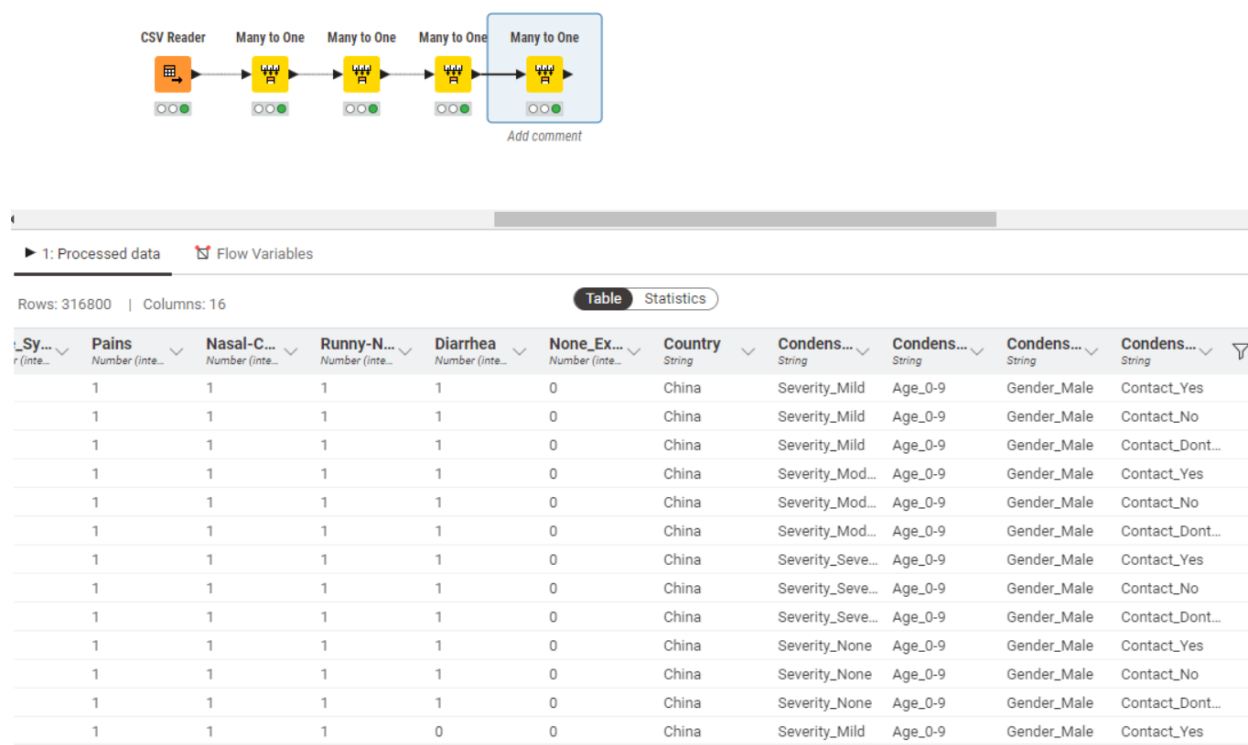


The screenshot shows the Knime CSV Reader interface. At the top, there is a 'CSV Reader' widget icon with a plus sign and a comment button. Below it, the interface displays a table with 12 columns and 13 rows. The columns are: #, Row..., Fever, Tiredness, Dry-Cough, Difficulty..., Sore-Thr..., None\_Sy..., Pains, Nasal-C..., Runny-N..., and Diarrhea. Each column has a dropdown menu for data type. The table contains numerical data representing the count of each symptom across different rows.

#	Row...	Fever Number (inte...)	Tiredness Number (inte...)	Dry-Cough Number (inte...)	Difficulty... Number (inte...)	Sore-Thr... Number (inte...)	None_Sy... Number (inte...)	Pains Number (inte...)	Nasal-C... Number (inte...)	Runny-N... Number (inte...)	Diarrhea Number (inte...)
1	Row0	1	1	1	1	1	0	1	1	1	1
2	Row1	1	1	1	1	1	0	1	1	1	1
3	Row2	1	1	1	1	1	0	1	1	1	1
4	Row3	1	1	1	1	1	0	1	1	1	1
5	Row4	1	1	1	1	1	0	1	1	1	1
6	Row5	1	1	1	1	1	0	1	1	1	1
7	Row6	1	1	1	1	1	0	1	1	1	1
8	Row7	1	1	1	1	1	0	1	1	1	1
9	Row8	1	1	1	1	1	0	1	1	1	1
10	Row9	1	1	1	1	1	0	1	1	1	1
11	Row10	1	1	1	1	1	0	1	1	1	1
12	Row11	1	1	1	1	1	0	1	1	1	1
13	Row12	1	1	1	1	1	0	1	1	1	0

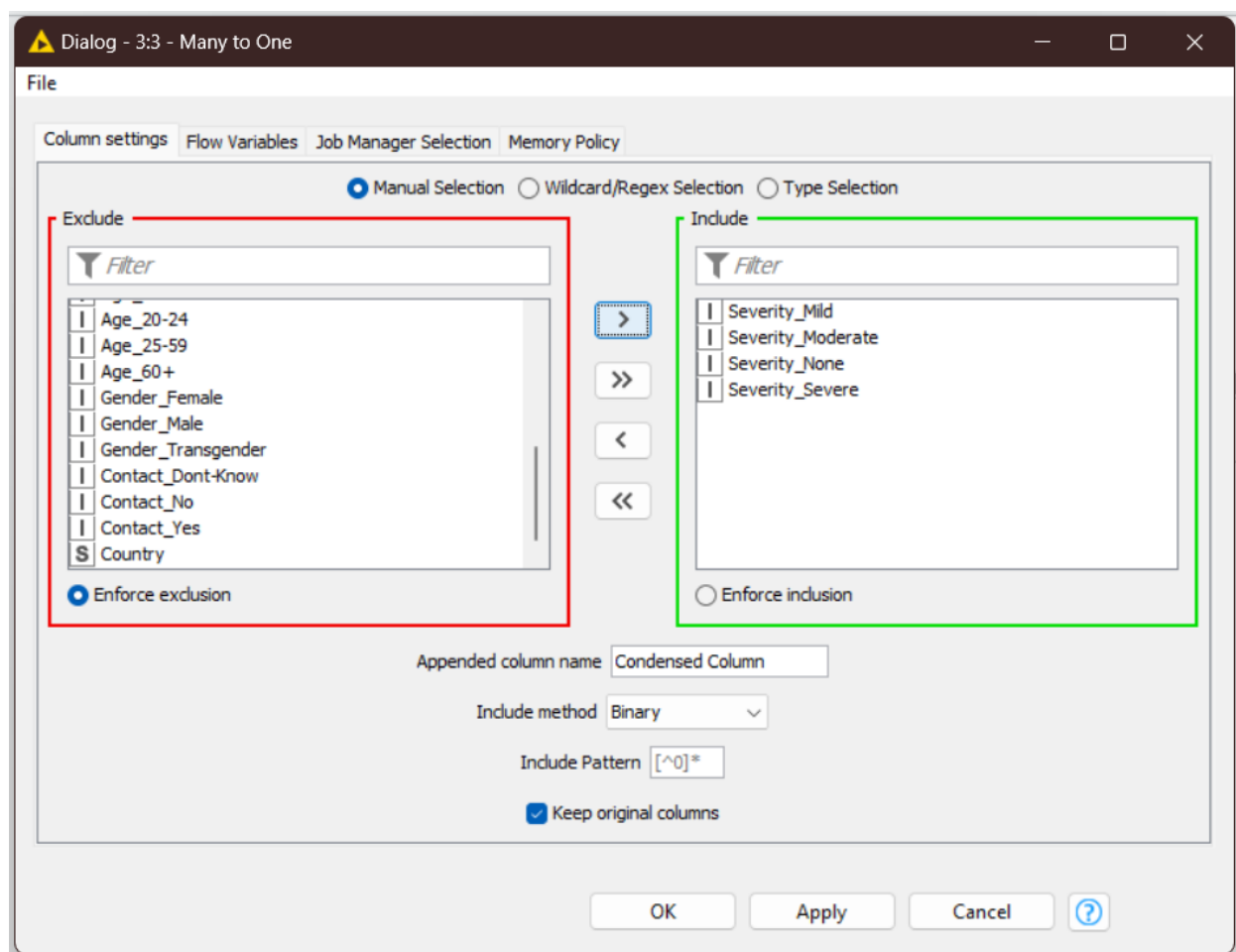
*Nota.* Imagen de autoría propia donde se muestra la información del dataset de Covid en Knime.

Luego se procede a agrupar los campos anteriormente mencionados con el fin de reducir la información en una sola columna para que permita trabajar y analizar de manera más precisa la variable objetivo que en este caso será “severidad”, para esto se utiliza la opción de “many to one” de Knime y como se ve en la figura 4 se realiza este procedimiento cuatro veces, una para cada grupo de variables.

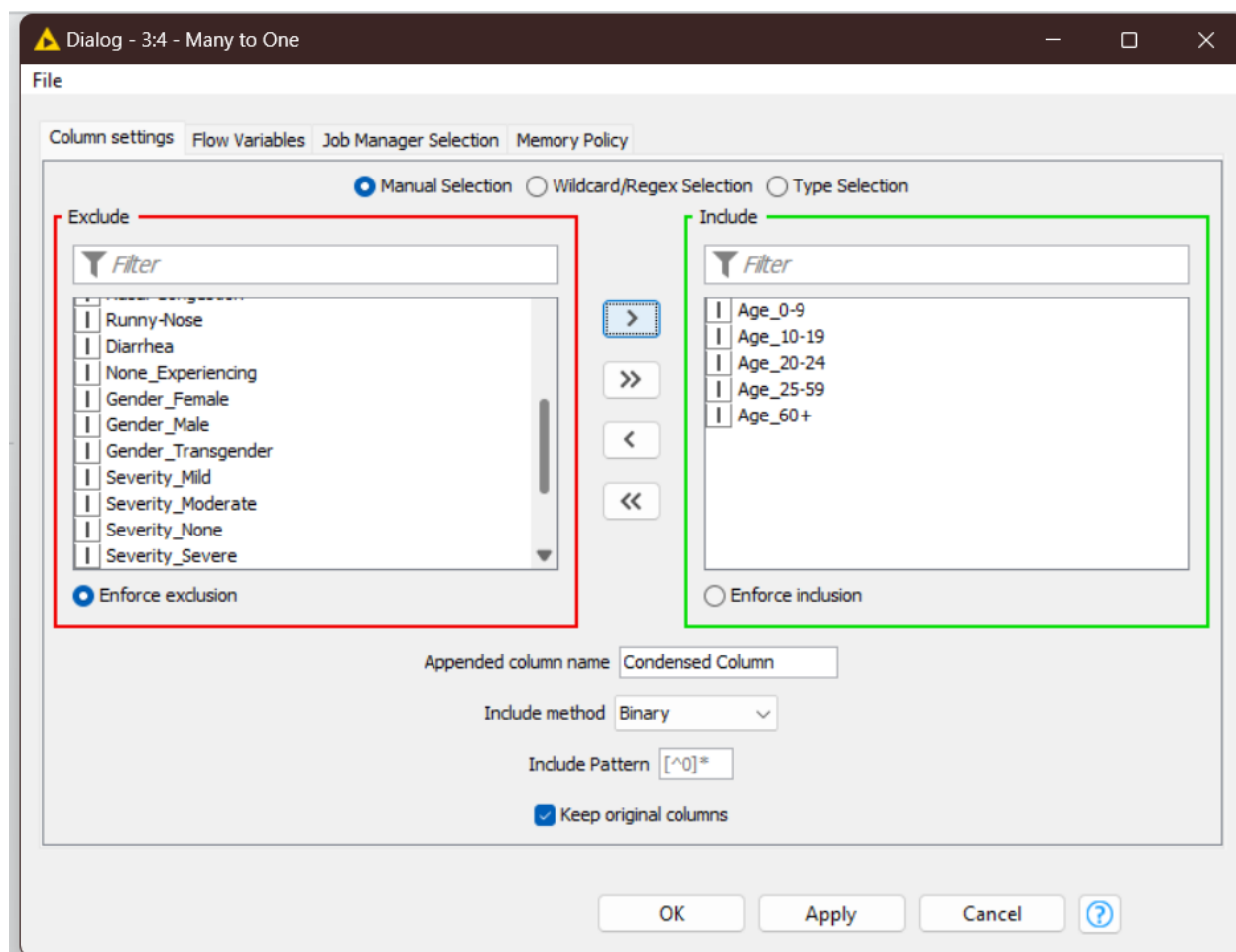
**Figura 4***Función many to one*

*Nota.* Imagen de autoría propia donde se muestra la agrupación de variables utilizando la función “many to one” de Knime.

En las figuras 5, 6, 7 y 8, se observa la variables que fueron agrupadas en cada función “many to one”, donde la figura 5 muestra el grupo de variables “severity” en una nueva columna llamada “condensed column”, la figura 6 muestra el grupo de variables “age” en una columna llamada “condensed column #1”, la figura 7 muestra el grupo de variables “gender” en una nueva columna llamada “condensed column #2” y la figura 8 muestra el grupo de variables “contact” en una nueva columna llamada “condensed column #3”.

**Figura 5***Grupo de variables severity*

*Nota.* Imagen de autoría propia donde se muestra la agrupación de las variables “severity” utilizando la función “many to one” de Knime.

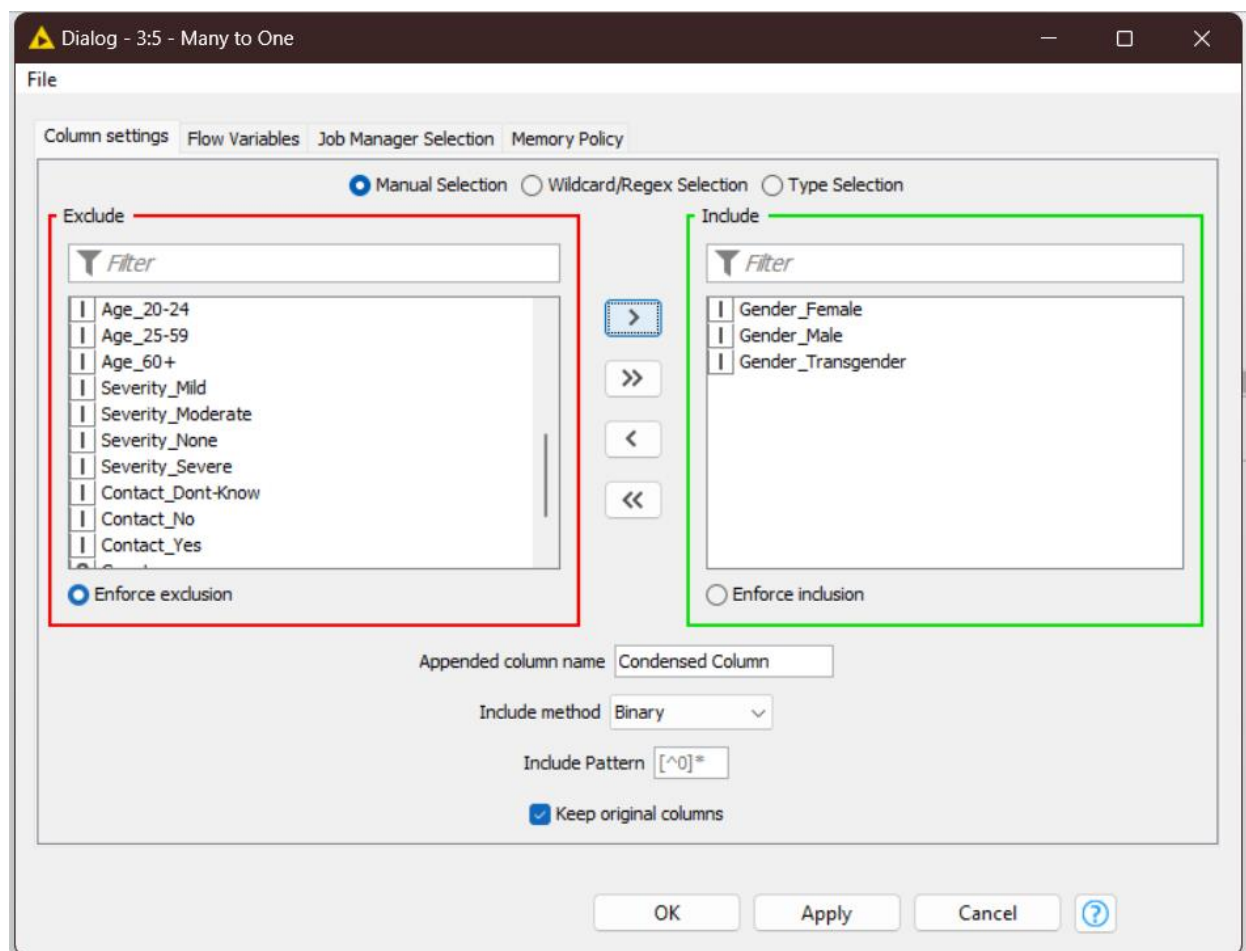
**Figura 6***Grupo de variables age*

*Nota.* Imagen de autoría propia donde se muestra la agrupación de las variables “age” utilizando la función “many to one” de Knime.

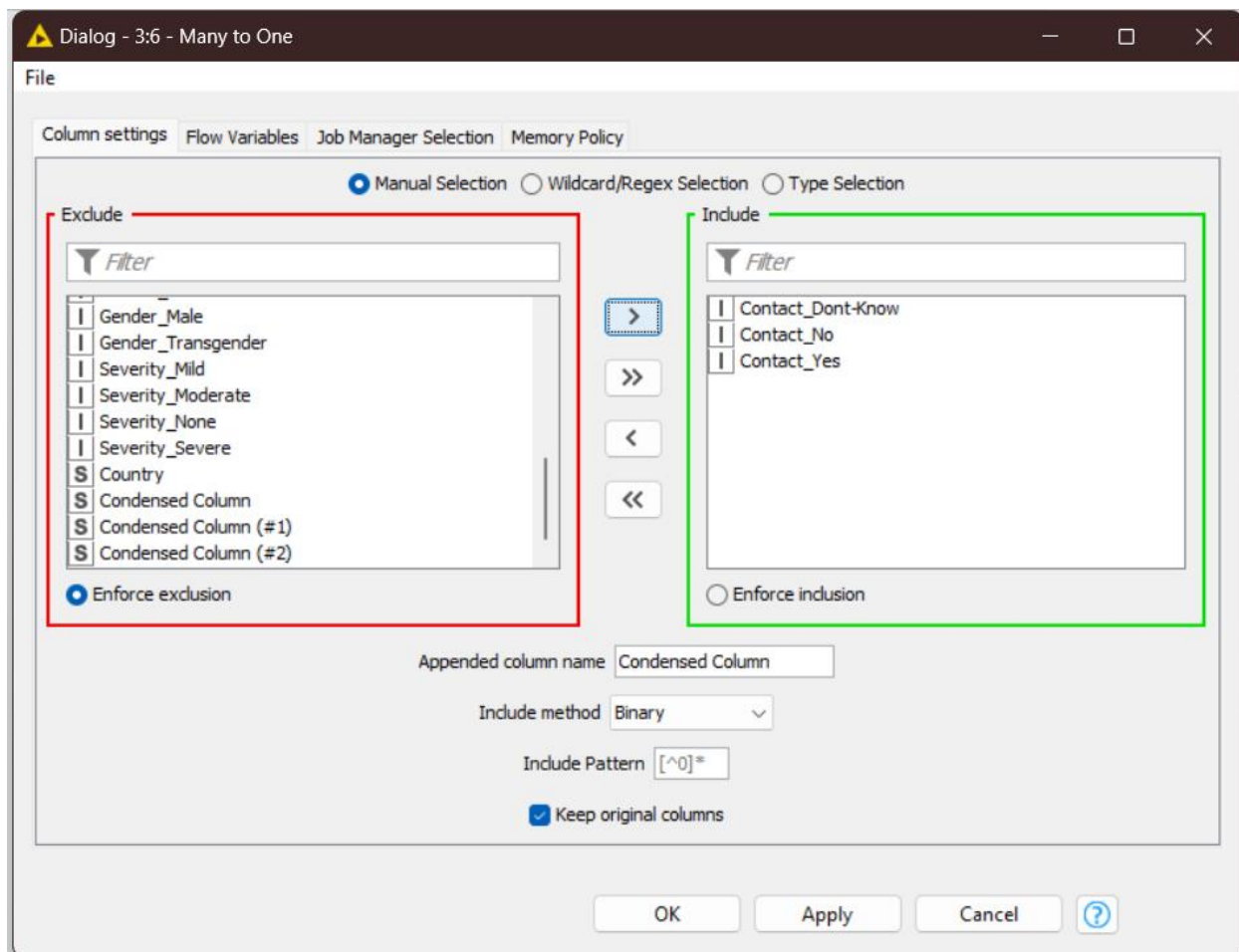


**Figura 7**

*Grupo de variables gender*



*Nota.* Imagen de autoría propia donde se muestra la agrupación de las variables “gender” utilizando la función “many to one” de Knime.

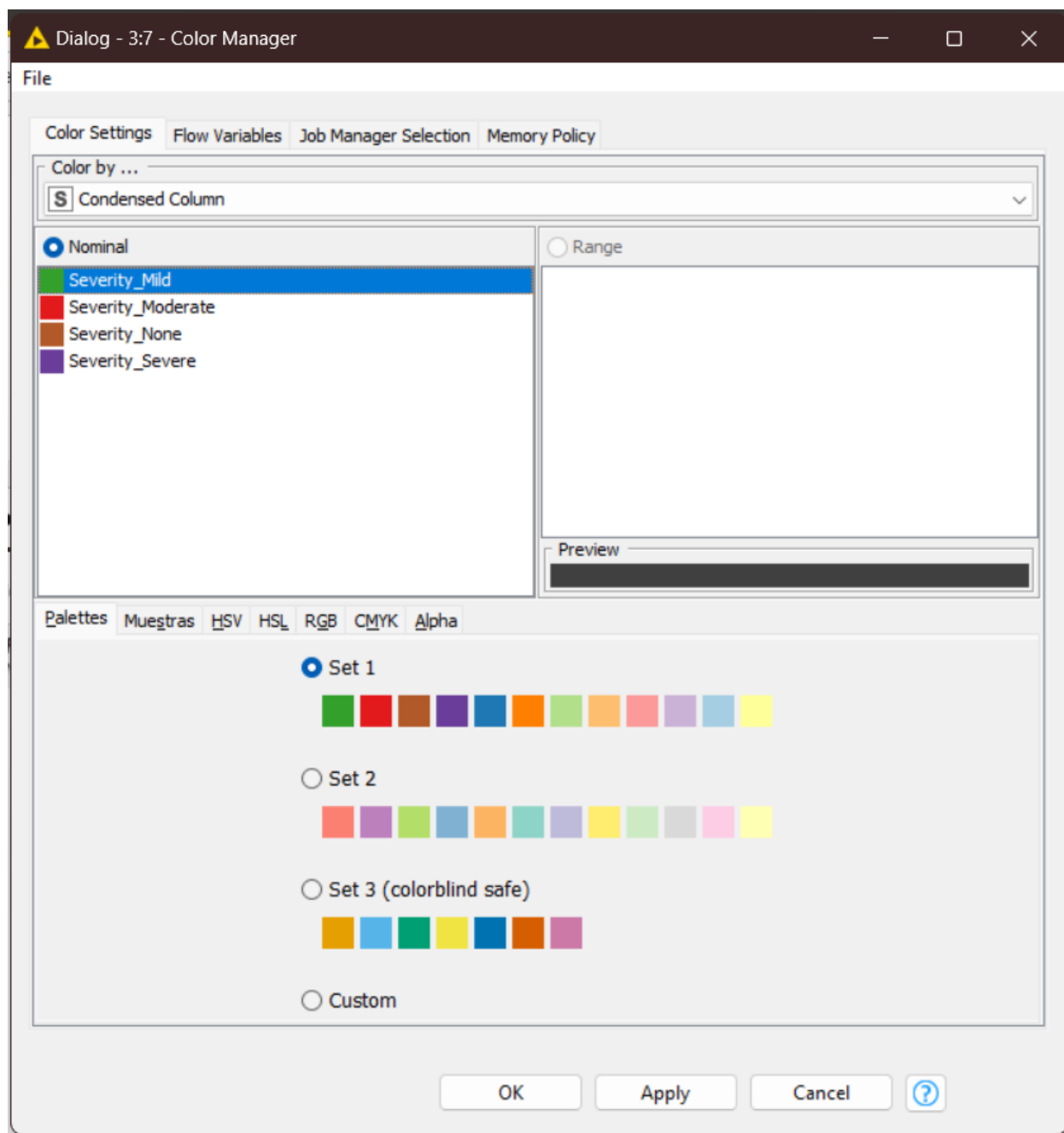
**Figura 8***Grupo de variables contact*

*Nota.* Imagen de autoría propia donde se muestra la agrupación de las variables “contact” utilizando la función “many to one” de Knime.

Después de agrupar las variables, se utiliza la opción “color manager” como se ve en la figura 9, sobre el grupo de variables de severidad, es decir la columna “condensed column”, las cuales son el objetivo del estudio con el fin de identificar claramente los diferentes tipos de severidad como se ve en la figura 10.

**Figura 9**

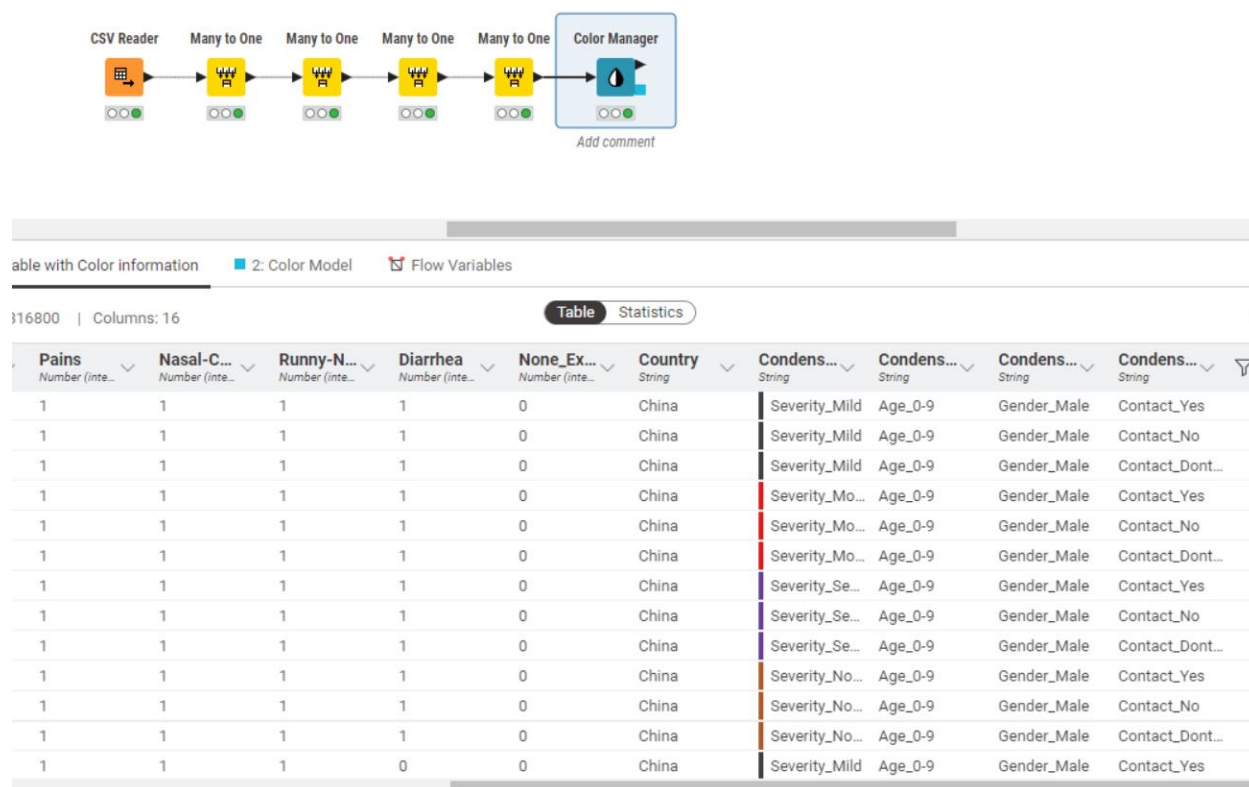
*Función color manager*



*Nota.* Imagen de autoría propia donde se muestra la aplicación de “color manager” sobre la variable objetivo “severidad”.

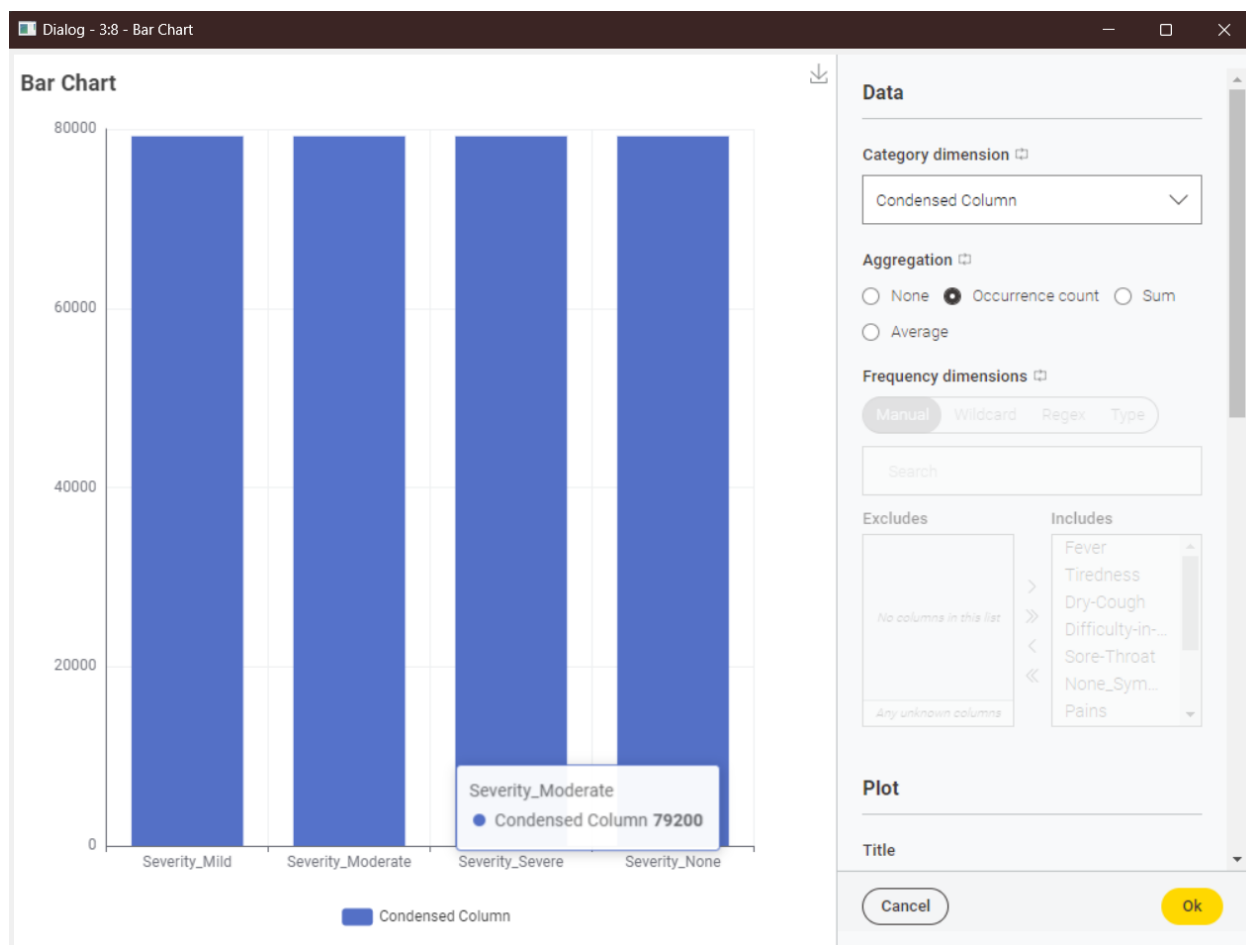
**Figura 10**

*Aplicación de color manager sobre la variable objetivo*



*Nota.* Imagen de autoría propia donde se muestra la aplicación de “color manager” sobre la variable objetivo “severidad”.

Se puede verificar la información de la variable objetivo “severidad” a través de un grafico de barras donde se muestra el total de los datos (316.800), representados en los cuatro tipos de severidad de covid, los cuales tiene 79.200 casos para cada tipo de severidad como se ve en la figura 11.

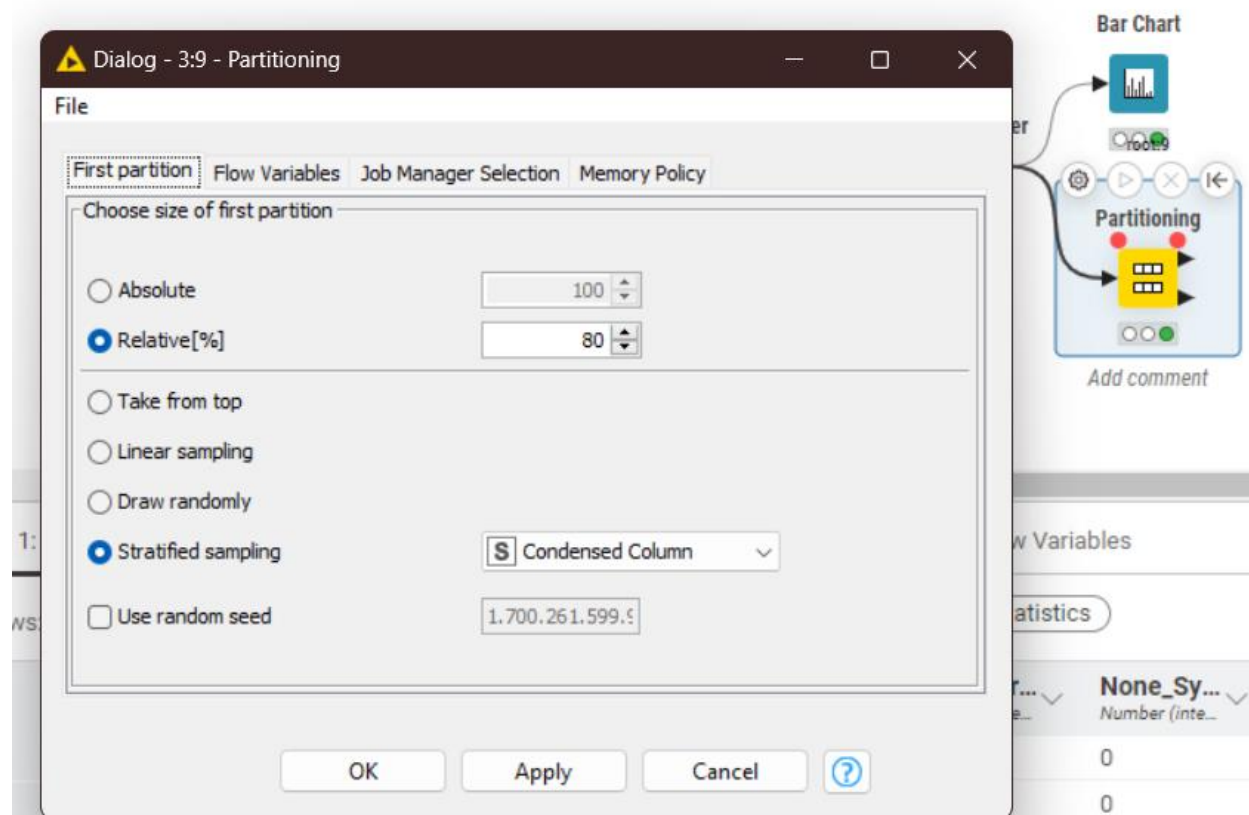
**Figura 11***Grafica de barras de severidad*

*Nota.* Imagen de autoría propia donde se muestra la grafica de barras que representa los datos de severidad por cada tipo.

Finalmente como parte de la configuración del dataset se utiliza la opción “partitioning” para tomar una parte de los datos para entrenar el modelo, en este caso se tomará el 80% de los datos para el entrenamiento como se ve en la figura 12.

**Figura 12**

*Partición de los datos*



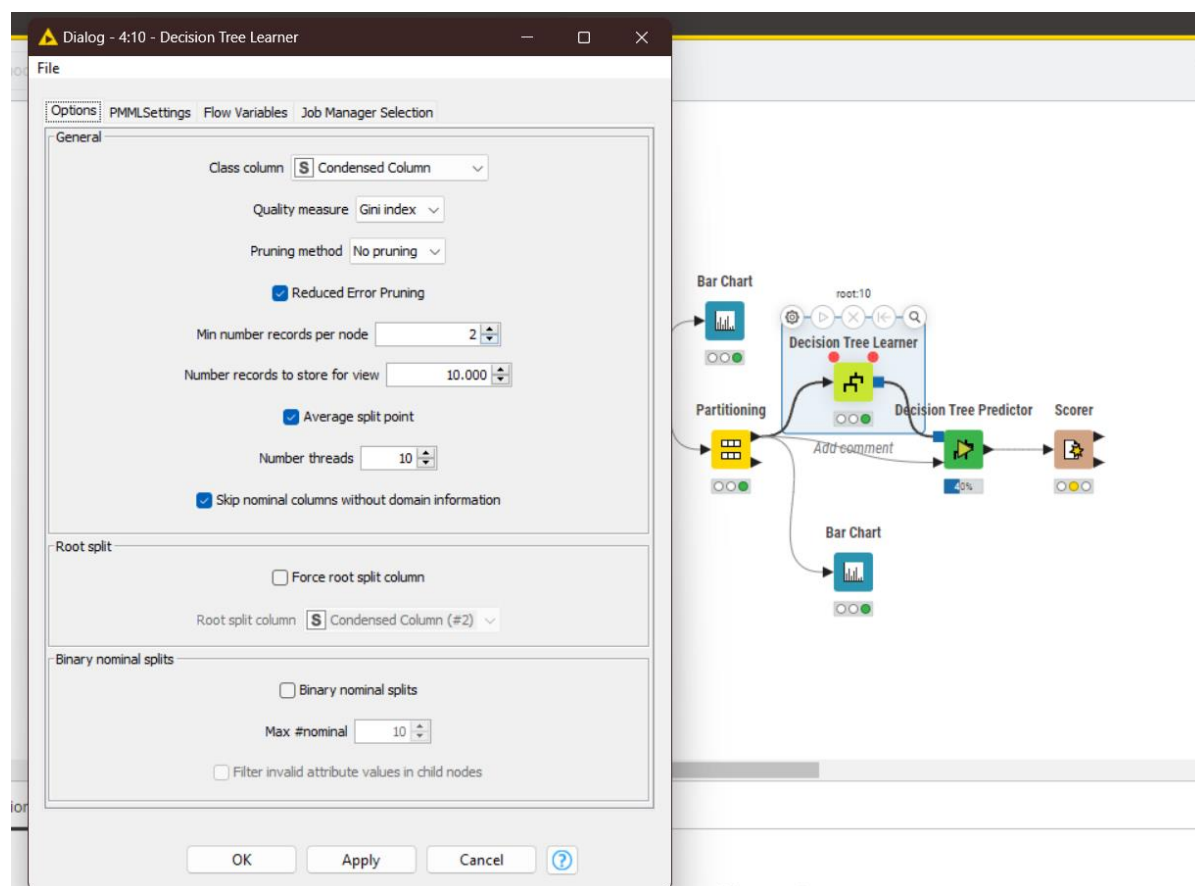
*Nota.* Imagen de autoría propia donde se muestra la partición de los datos en un 80% para el entrenamiento del modelo.

## Árbol de Decisión

Con el dataset ya configurado, se realizará el árbol de decisión sobre la variable objetivo “severidad”, para esto se utiliza la opción **“Decision tree learner”** de Knime y se debe conectar el 80% de los datos que se van a utilizar para el entrenamiento del modelo y en la opción de decision tree learner se debe elegir en el campo “class column” la columna “condensed column” como se ve en la figura 13, la cual contiene la variable objetivo de severidad con sus cuatro tipos de severidad,

**Figura 13**

*Configuración de decision tree learner*

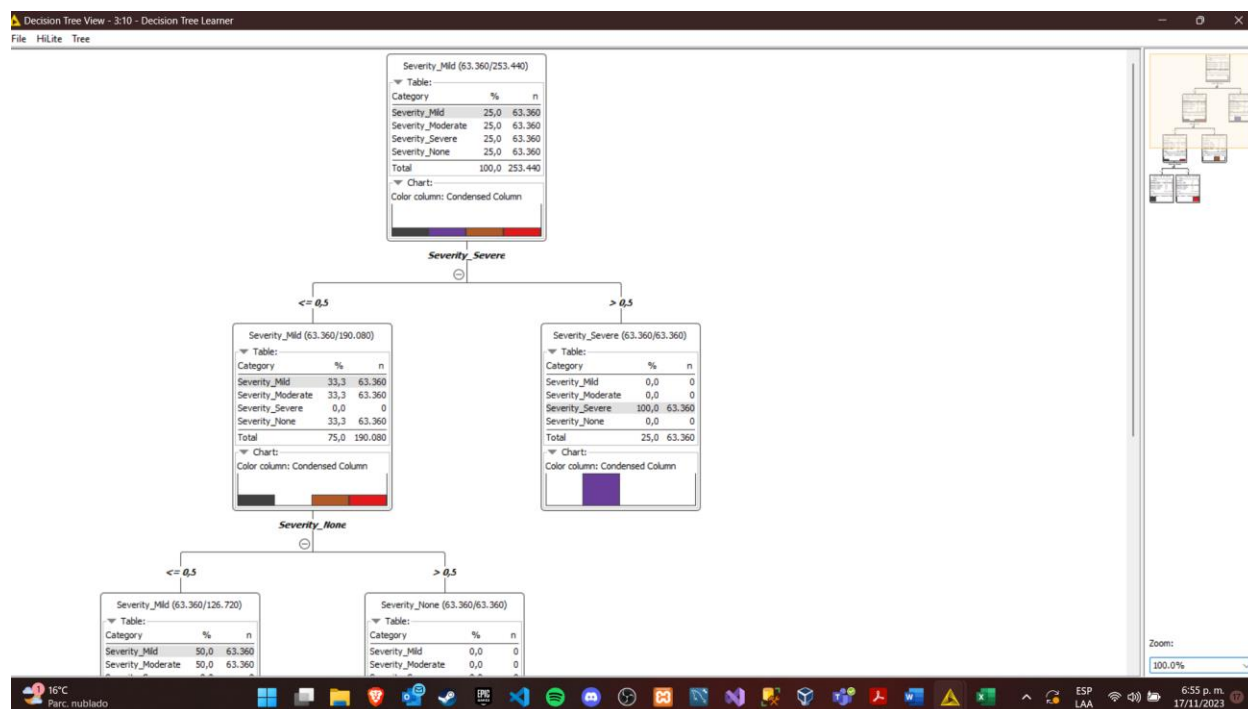


*Nota.* Imagen de autoría propia donde se muestra la configuración del modelo “decision tree learner” con los datos de entrenamiento.

Después de cargados los datos en el modelo, se ejecuta el proceso para obtener el árbol de decisión como se ve en la figura 14, que muestran los datos obtenidos de la variable objetivo con el fin de realizar el análisis de los datos para concluir como el modelo clasifica los pacientes con covid.

**Figura 14**

*Decision tree learner*



*Nota.* Imagen de autoría propia donde se muestra el árbol de decisión arrojado por el modelo.

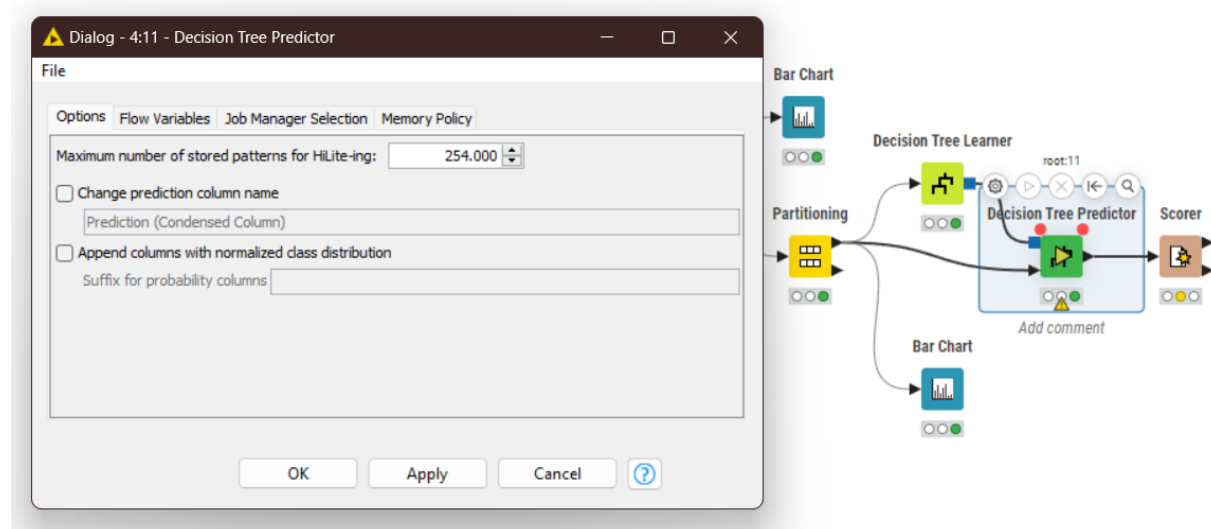
Revisando la información arrojada por el árbol de decisión se encuentra que el total de datos estudiados es de 253.440 que corresponde al 80% del total de datos del dataset y donde se evidencia que el 25% que corresponde a 63.360 de 253.440 del total, corresponde a la cantidad de pacientes con covid en cada tipo de severidad, es decir, que para cada tipo de severidad hay exactamente la misma cantidad de casos.



Luego de entrenar y analizar el árbol de decisión, se procede a crear el árbol de decisiones de predicciones, por medio de la función *“Decision tree predictor”* como se ve en la figura 15, el cual será el modelo final.

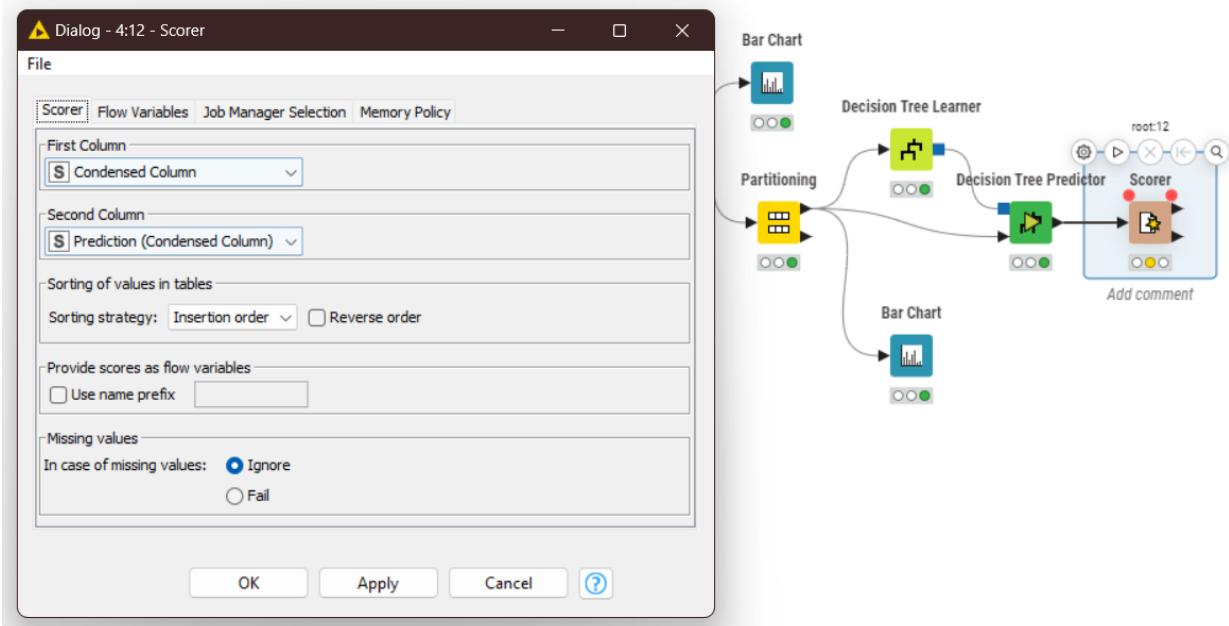
**Figura 15**

*Decision tree predictor*

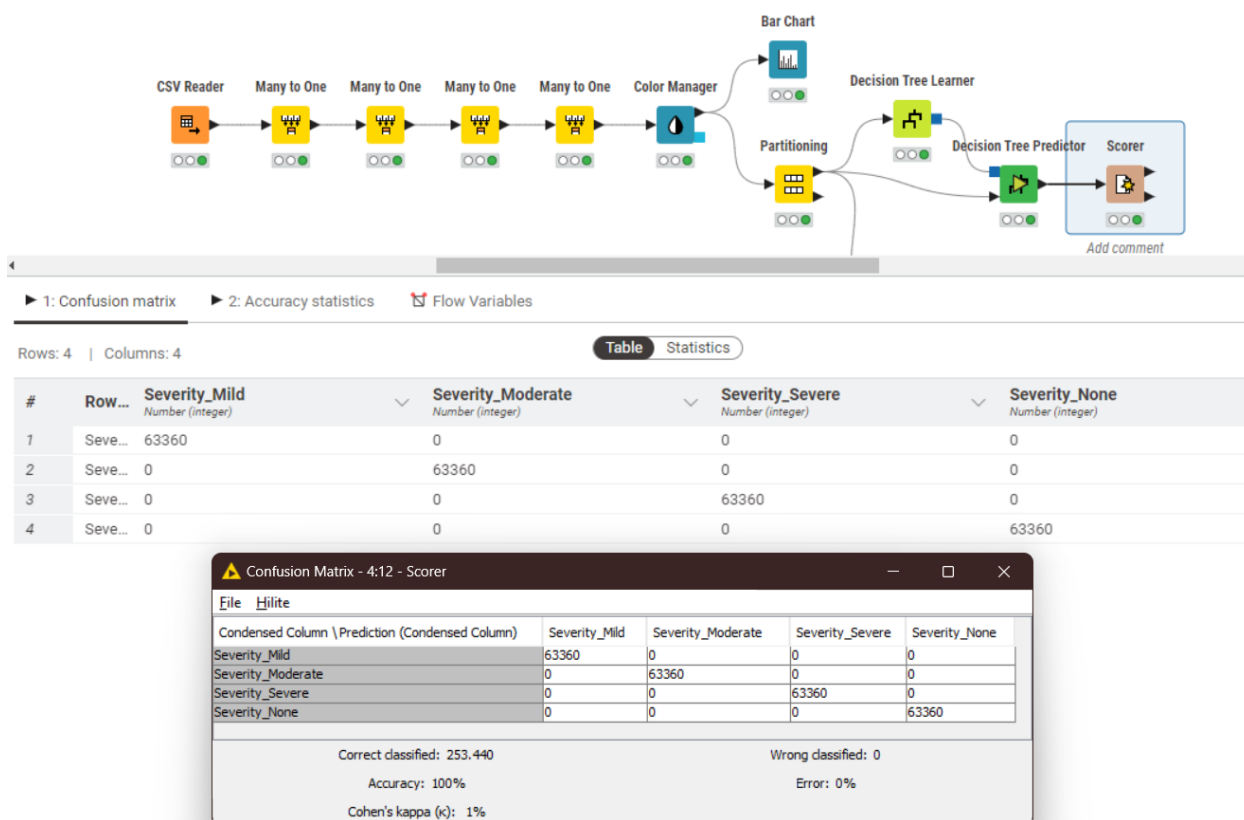


*Nota.* Imagen de autoría propia donde se muestra el modelo decision tree predictor.

Finalmente para visualizar un resumen de los datos encontrados y de manera más clara, se hace uso de la función *“Scorer”* en la cual pasa la variable objetivo severidad que esta contenida en la columna “condensed column” como se ve en la figura 16 y se ejecuta el proceso para obtener la matriz de confusión con el detalle de la precisión del modelo realizado, donde se encuentra que el *accuracy* fue de 100% siendo este un modelo confiable, pues logro clasificar correctamente todos los datos con un total de 253.440 y encontrando que para cada tipo de severidad hay 63.360 casos verdaderos positivos como se ve en la figura 17.

**Figura 16***Scorer*

*Nota.* Imagen de autoría propia donde se muestra la configuración de la matriz de confusión a través de la opción scorer.

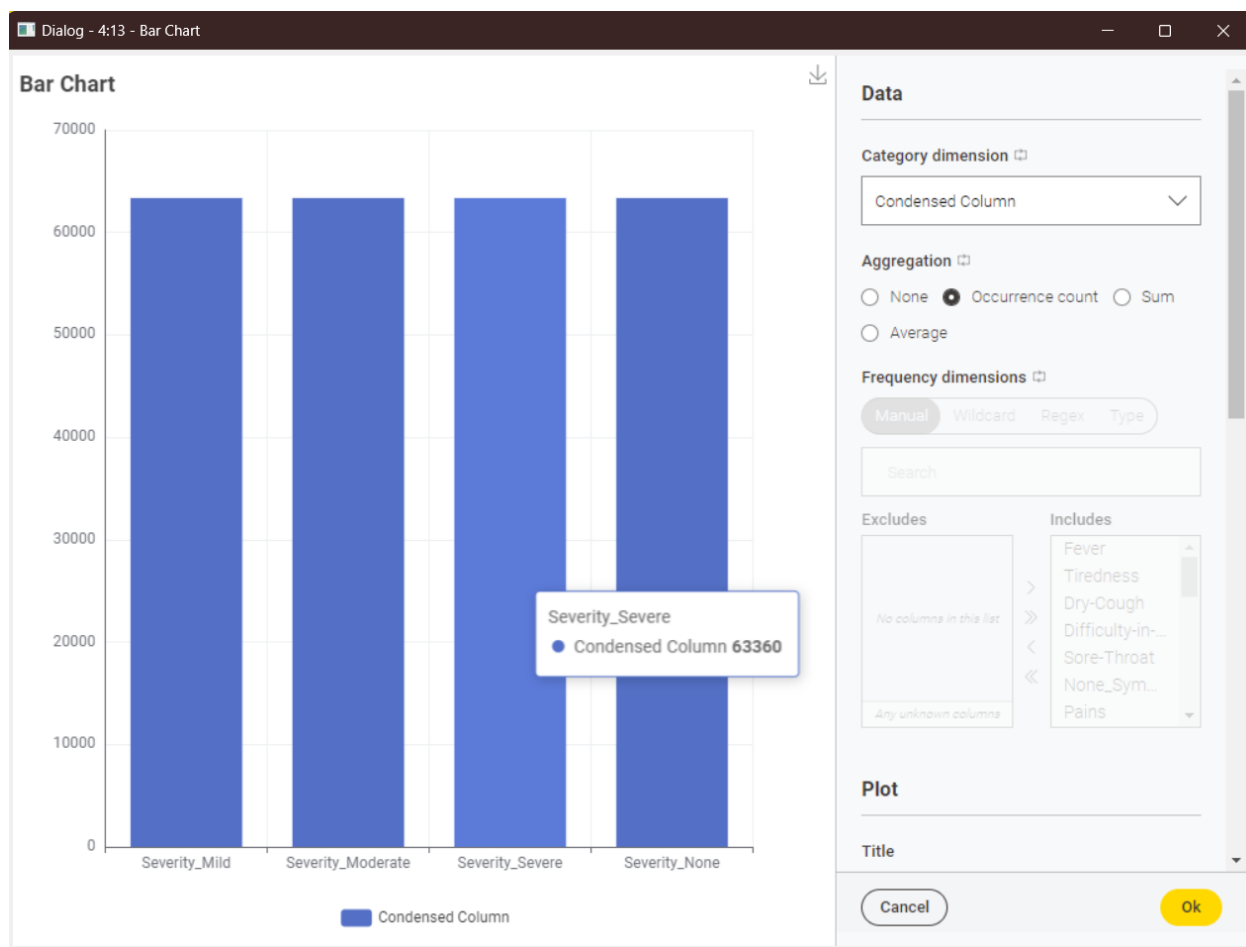
**Figura 17***Matriz de confusión*

*Nota.* Imagen de autoría propia donde se muestra la matriz de confusión con la conclusión del modelo de árbol de decisión.

El modelo se puede comprobar a través de un gráfico de barras que represente los datos encontrados y donde se evidencia que para cada tipo de severidad se tiene un total de 63.360 casos de covid como se ve en la figura 18.

**Figura 18**

*Grafico de barras con el 80% de los datos*



*Nota.* Imagen de autoría propia donde se muestra el gráfico de barras que representa el total de los datos correspondientes al 80%, distribuidos en cada tipo de severidad, obteniendo un 63.360 para cada uno.

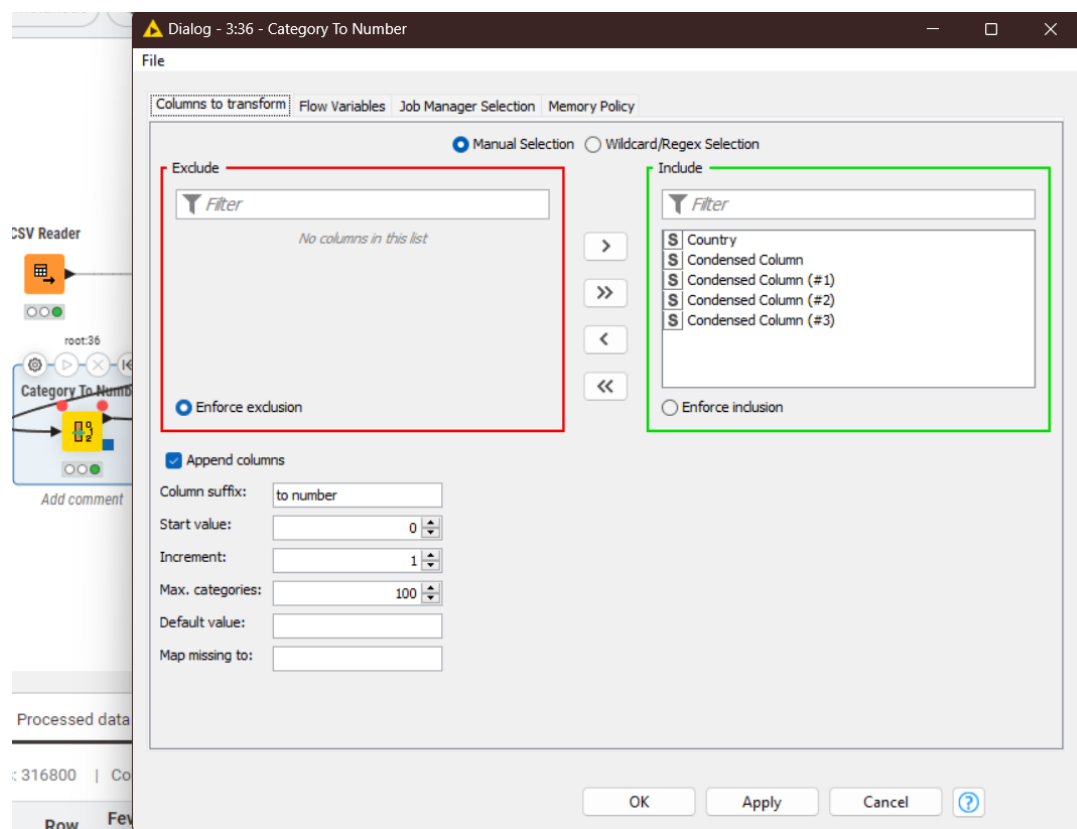
## KNN (K-Nearest Neighbors)

Para el algoritmo de K-Nearest Neighbors se utilizará la misma configuración del dataset tomando como variable objetivo la severidad del caso para determinar si un paciente tiene o no covid.

Antes de continuar con el algoritmo se realiza la conversión de las variables categóricas a un tipo de dato numérico, para ello se utiliza el nodo “**Category To Number**” de Knime como se ve en la figura 19.

**Figura 19**

*Category to Number*

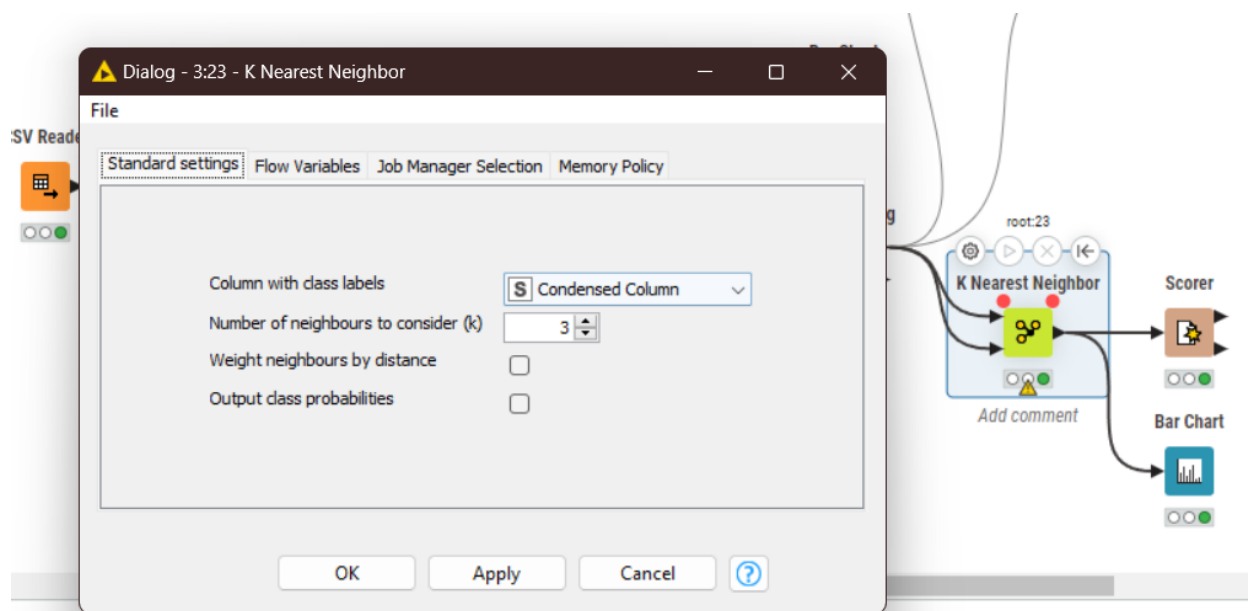


*Nota.* Imagen de autoría propia donde se muestra la configuración del nodo “Category to number” de Knime, para conversión de las variable categóricas.

Luego mediante el nodo “**K Nearest Neighbor**” de Knime se encontrará la clasificación de los pacientes con covid de acuerdo su severidad, en la configuración del nodo en la opción “column with class labels” se le indicará el tipo de severidad que se encuentra agrupado en la columna “condensed column” y el número de K vecinos que se asignará será de 3 como se ve en la figura 20, una vez configurado esto se ejecuta el nodo para obtener la clasificación.

**Figura 20**

*Configuración del nodo K Nearest Neighbor*



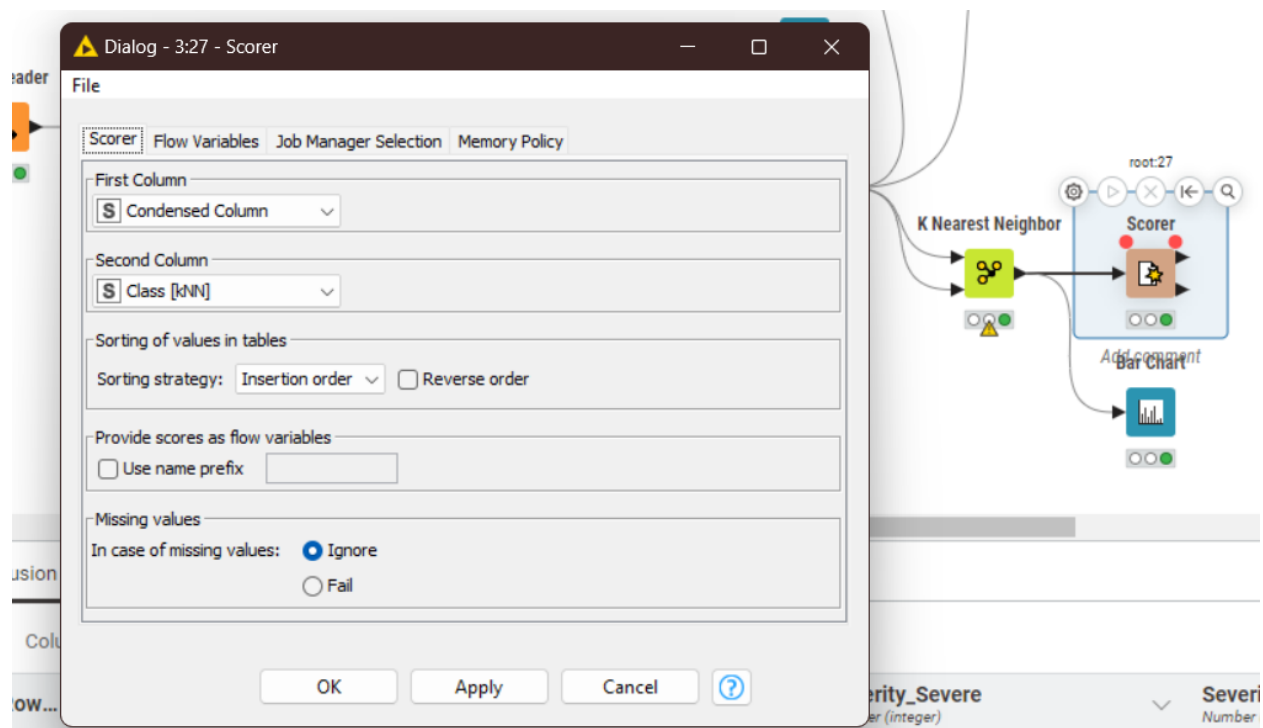
*Nota.* Imagen de autoría propia donde se muestra la configuración del nodo “K Nearest Neighbor” de Knime.

Después de ejecutar el nodo KNN, se procede a visualizar la información obtenida por el algoritmo a través de la función “**Scorer**” en la cual pasa la variable objetivo severidad que está contenida en la columna “condensed column” como se ve en la figura 21 y se ejecuta el proceso para obtener la matriz de confusión con el detalle de la precisión del modelo realizado, donde se encuentra que el *accuracy* fue de 100% siendo este un modelo confiable, pues logro clasificar correctamente todos los datos con un total de 253.440 y encontrando que para cada tipo de severidad hay 63.247

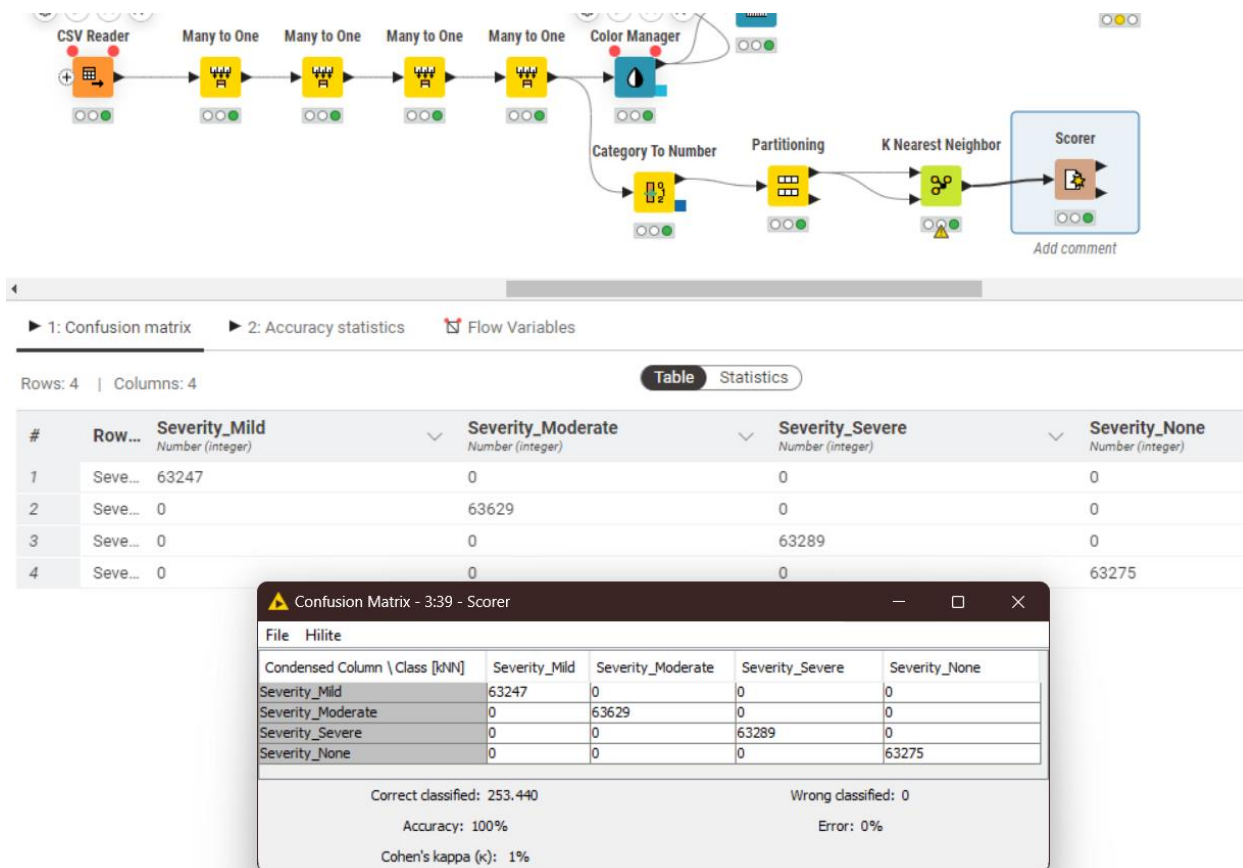
casos verdaderos positivos con severidad baja, 63.629 casos verdaderos positivos con severidad moderada, 63.289 caso verdaderos positivos con severidad severa y 63.275 casos verdaderos positivos con ninguna severidad como se ve en la figura 22.

**Figura 21**

*Scorer KNN*



*Nota.* Imagen de autoría propia donde se muestra la configuración de la matriz de confusión a través de la opción scorer.

**Figura 22***Matriz de confusión KNN*

*Nota.* Imagen de autoría propia donde se muestra la matriz de confusión con la conclusión del modelo KNN realizado.



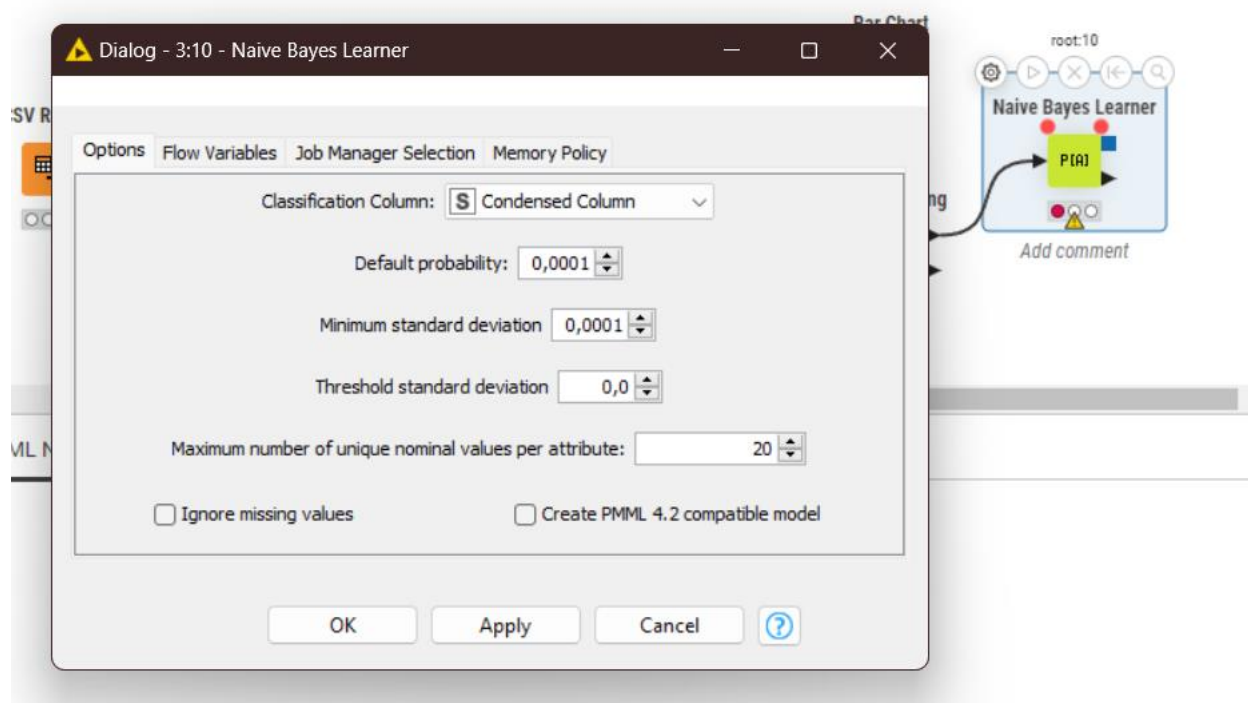
## Naive Bayes

Para el algoritmo de Naive Bayes se utilizará la misma configuración del dataset tomando como variable objetivo la severidad del caso para determinar si un paciente tiene o no covid.

Mediante el nodo “**Naive Bayes Learner**” de Knime se encontrará la clasificación de los pacientes con covid de acuerdo su severidad, en la configuración del nodo en la opción “classification column” se le indicará el tipo de severidad que se encuentra agrupado en la columna “condensed column” como se ve en la figura 23, una vez configurado esto se ejecuta el nodo para obtener la clasificación.

**Figura 23**

*Configuración del nodo Naive Bayes*



*Nota.* Imagen de autoría propia donde se muestra la configuración del nodo “Naive Bayes” de Knime.

Después de ejecutar el nodo Naive Bayes, se procede a visualizar la información obtenida por el algoritmo, en donde encontramos como el algoritmo utiliza la distribución Gaussiana para determinar la predicción con base a la severidad como se ve en la figura 24 donde se demuestra que con el 80% de los datos del dataset se encuentra un 25% para cada tipo de severidad, lo que corresponde a 63.360 casos, además que con relación a variables como los síntomas como se ve en la figura 25, se evidencia el mismo 25% que corresponde al rango para cada tipo de severidad y donde se calcula la mediana y desviación estándar respecto a los datos.

**Figura 24**

*Resultado del nodo Naive Bayes*

Naive Bayes Learner View - 3:10 - Naive Bayes Learner					
File					
Class counts for Condensed Column					
Class:	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe	
Count:	63360	63360	63360	63360	
Total count: 253440					
Threshold to used for zero probabilities: 1.0E-4					
P(Condensed Column (#1)   class=?)					
Class/Condensed Column (#1)	Age_0-9	Age_10-19	Age_20-24	Age_25-59	Age_60+
Severity_Mild	12663	12659	12645	12761	12632
Severity_Moderate	12631	12672	12657	12721	12679
Severity_None	12696	12581	12677	12683	12723
Severity_Severe	12772	12690	12617	12621	12660
Rate:	20 %	20 %	20 %	20 %	20 %
P(Condensed Column (#2)   class=?)					
Class/Condensed Column (#2)	Gender_Female	Gender_Male	Gender_Transgender		
Severity_Mild	21199	21046	21115		
Severity_Moderate	21154	21125	21081		
Severity_None	21152	21104	21104		
Severity_Severe	21136	20981	21243		
Rate:	33 %	33 %	33 %		
P(Condensed Column (#3)   class=?)					
Class/Condensed Column (#3)	Contact_Dont-Know	Contact_No	Contact_Yes		
Severity_Mild	21121	21008	21231		
Severity_Moderate	21083	21084	21193		
Severity_None	21143	21135	21082		
Severity_Severe	21156	21035	21169		
Rate:	33 %	33 %	33 %		

*Nota.* Imagen de autoría propia donde se muestra el resultado de del nodo “Naive Bayes” donde se evidencia la distribucion gaussiana utilizada por el algoritmo.

**Figura 255**

*Resultado del nodo Naive Bayes con respecto a los síntomas*

Gaussian distribution for Diarrhea per class value				
	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe
Count:	63360	63360	63360	63360
Mean:	0,36192	0,36294	0,36324	0,36406
Std. Deviation:	0,48056	0,48085	0,48094	0,48117
Rate:	25 %	25 %	25 %	25 %

Gaussian distribution for Difficulty-in-Breathing per class value				
	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe
Count:	63360	63360	63360	63360
Mean:	0,49975	0,50079	0,49932	0,50103
Std. Deviation:	0,5	0,5	0,5	0,5
Rate:	25 %	25 %	25 %	25 %

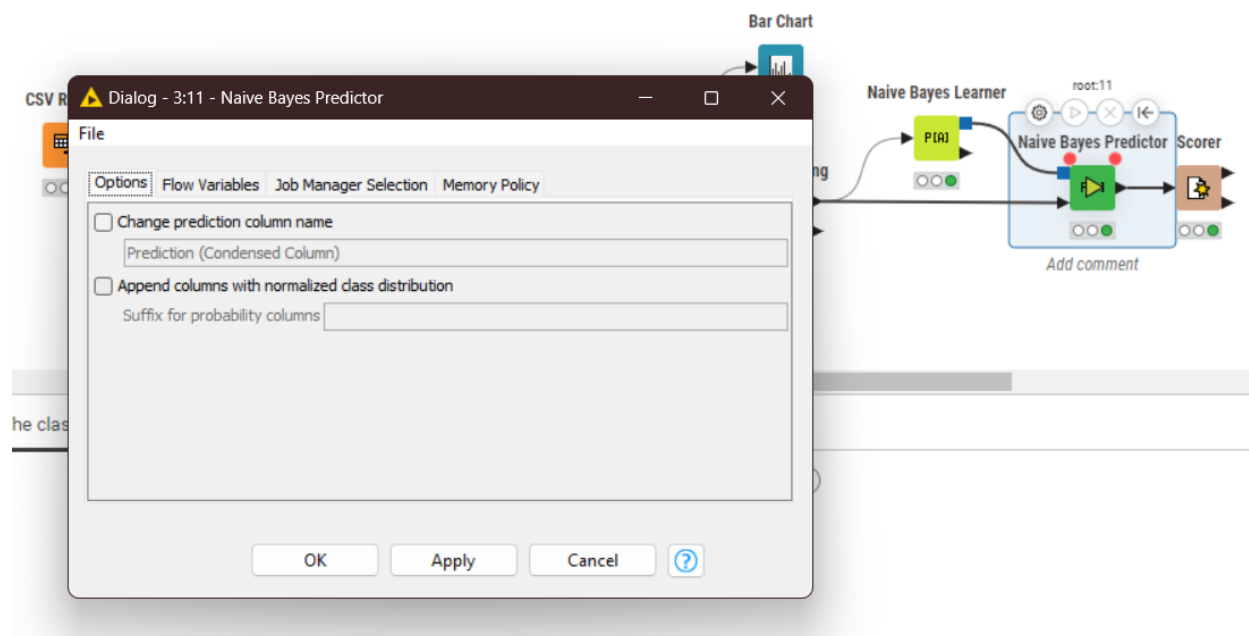
Gaussian distribution for Dry-Cough per class value				
	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe
Count:	63360	63360	63360	63360
Mean:	0,56286	0,56316	0,56237	0,56433
Std. Deviation:	0,49604	0,496	0,4961	0,49585
Rate:	25 %	25 %	25 %	25 %

Gaussian distribution for Fever per class value				
	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe
Count:	63360	63360	63360	63360
Mean:	0,31343	0,31296	0,3137	0,31135
Std. Deviation:	0,46389	0,4637	0,464	0,46305
Rate:	25 %	25 %	25 %	25 %

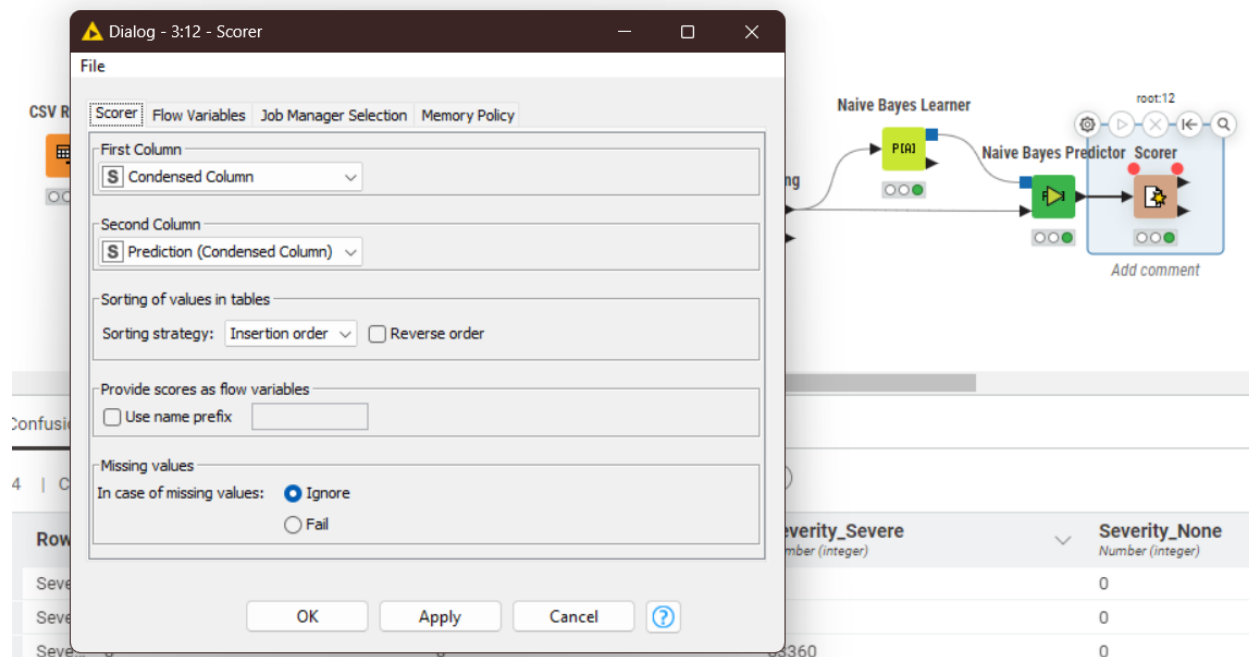
*Nota.* Imagen de autoría propia donde se muestra el resultado de del nodo “Naive Bayes” donde se evidencia la distribucion gausaina utilizada por el algoritmo, con respecto a los síntomas.

Luego de entrenar y analizar el modelo, se procede a probar el modelo de Naive Bayes por medio de la función **“Naive Bayes Predictor”** como se ve en la figura 26, pasando como parámetro a predecir la severidad del caso y allí se obtendrá el modelo final.

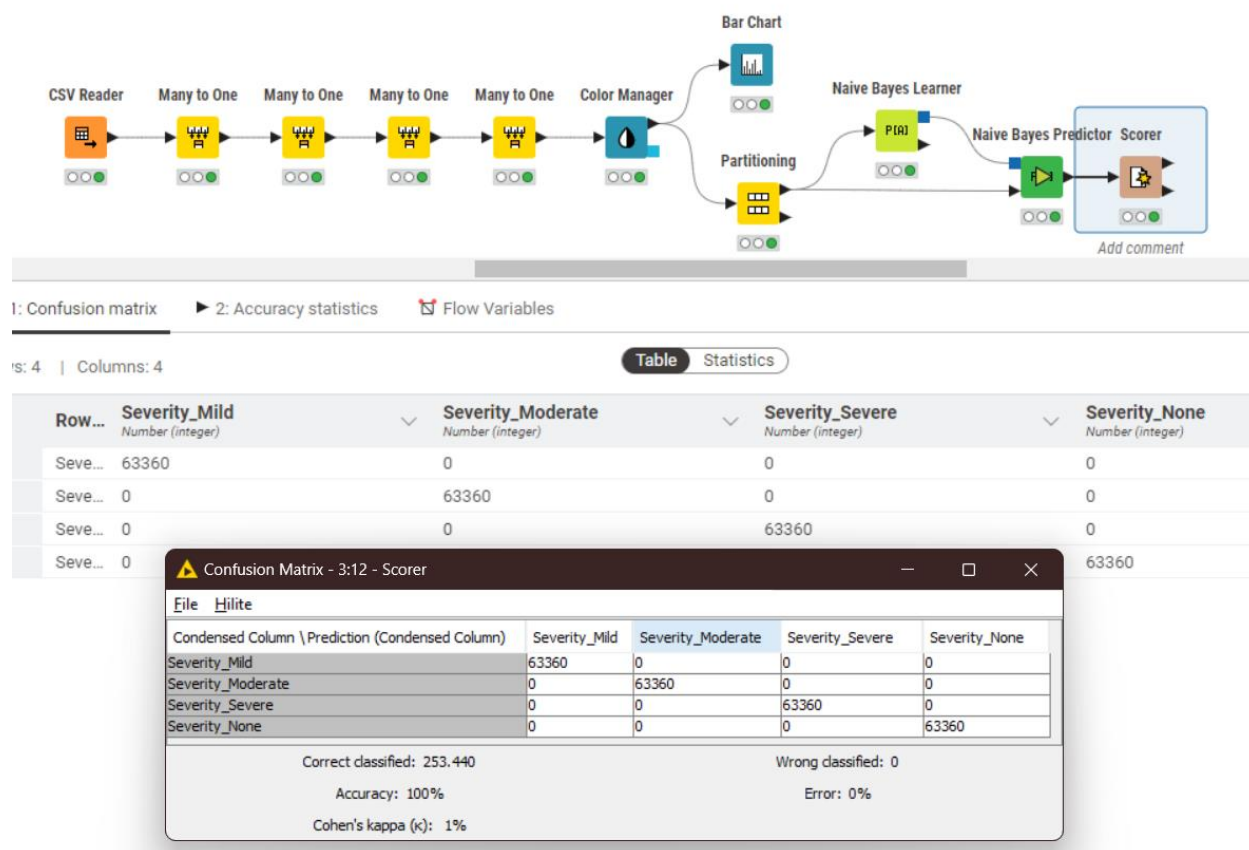
**Figura 26***Naive Bayes Predictor*

*Nota.* Imagen de autoría propia donde se muestra la configuración del nodo “Naive Bayes Predictor” de Knime para el modelo final.

Finalmente para visualizar un resumen de los datos encontrados y de manera más clara, se hace uso de la función “**Scorer**” en la cual pasa la variable objetivo severidad que está contenida en la columna “condensed column” como se ve en la figura 27 y se ejecuta el proceso para obtener la matriz de confusión con el detalle de la precisión del modelo realizado, donde se encuentra que el *accuracy* fue de 100% siendo este un modelo confiable, pues logro clasificar correctamente todos los datos con un total de 253.440 y encontrando que para cada tipo de severidad hay 63.360 casos verdaderos positivos como se ve en la figura 28.

**Figura 27***Scorer Naive Bayes*

*Nota.* Imagen de autoría propia donde se muestra la configuración de la matriz de confusión a través de la opción “scorer”.

**Figura 28***Matriz de confusión Naive Bayes*

*Nota.* Imagen de autoría propia donde se muestra la matriz de confusión con la conclusión del modelo de Naive Bayes.

### Referencias Bibliográficas

- Taylor Smith. (2019). Supervised Machine Learning with Python : Develop Rich Python Coding Practices While Exploring Supervised Machine Learning. Packt Publishing.  
[https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2145644&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp\\_5](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2145644&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_5) Capitulo 1
- Díaz Monroy, L. G. y Morales Rivera, M. A. (2012). Análisis estadístico de datos multivariados. Editorial Universidad Nacional de Colombia. <https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/127592?page=407>
- Pardo, C. E., & Del Campo, P. C. (2007). Combinación de métodos factoriales y de análisis de conglomerados en R: el paquete FactoClass. Revista colombiana de estadística, 30(2), 231-245. <https://www.redalyc.org/pdf/899/89930206.pdf>
- Posada Hernández, G. J. (2016). Elementos básicos de estadística descriptiva para el análisis de datos. Universidad Católica Luis Amigó. Recuperado de <https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/127436?page=128>