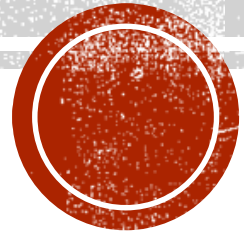
CIENCIA DE DATOS PARA DETECCIÓN DE MALWARE EN ANDROID



Christian Camilo Urcuqui López, MSc

UNIVERSIDAD ICESI | A OTRO NIVEL

PRESENTACIÓN

Christian Camilo Urcuqui López

Ing. Sistemas, Magister en Informática y Telecomunicaciones

Big Data Professional

Big Data Scientist

Deep Learning Specialization

Cyber Security Data Scientist, LUMU Technologies

Líder de investigación y desarrollo, laboratorio i2t – U ICESI.

ulcamilo@gmail.com

OBJETIVOS DE APRENDIZAJE

Al final de esta actividad, podrá realizar las siguientes operaciones en KNIME:

- Crea un flujo de trabajo
- Importar un conjunto de datos
- Explore un conjunto de datos mediante el uso de parcelas

CONTEXTO - ANDROID

- Sistema operativo para dispositivos móviles
- Cuenta con más de mil millones de usuarios activos [1]
- Código abierto basado en el kernel de Linux
- Arquitectura de cinco componentes [2]

[1] Pichai, S. Google I/O 2014 - Keynote [video. 6:43m], https://www.google.com/events/io. June 2014.
[2] Elenkov, N. Android Security Internals: An In-depth Guide to Android's Security Architecture. No Starch Press. October 2014.

ANDROID

Las aplicaciones se encuentran compiladas en un archivo Android Application Package (APK).

Dentro de un APK podremos entrar los elementos que permiten ejecutar una aplicación Android en un dispositivo, entre los archivos podemos encontrar:

- Código fuente (archivos .dex)
- Recursos
- AndroidManifest.xml

ANDROID

Mecanismos de seguridad

- Entorno sandbox a nivel del kernel para prevenir el acceso al file-system
- API de permisos a nivel de la aplicación
- Herramientas de seguridad a nivel del desarrollo de aplicaciones
- Plataforma de distribución digital Google Play

ANDROID – API DE PERMISOS

Existe una medida de seguridad implementada por Android llamada "Sistema de permisos" (Permission system), esta es la encargada de controlar los accesos de las aplicaciones a elementos del dispositivo (por ejemplo, accesos a Internet, a la lista de contactos, a la cámara, etc.)

Los accesos a los permisos los podemos encontrar en un archivo único en cada APK (llamado AndroidManifest.xml).

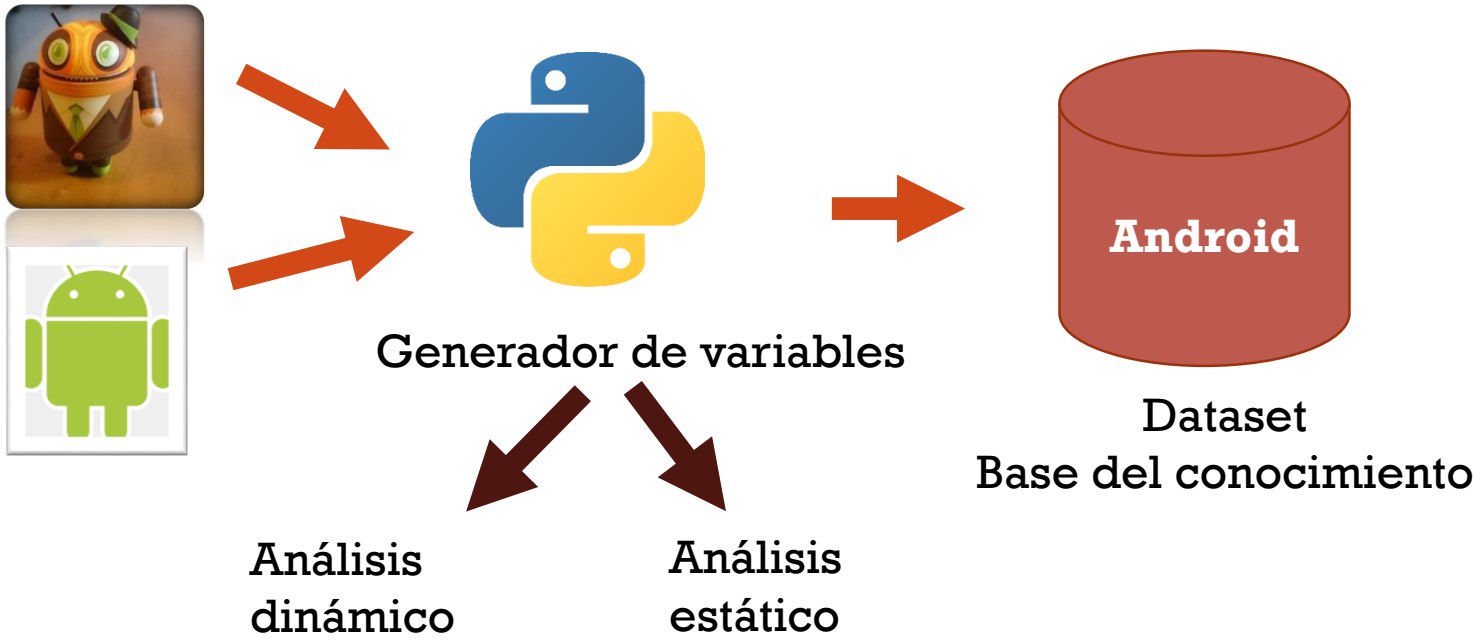
ANDROID - DATASET

- En [3] se realizo una recolección de aplicaciones maliciosas y benignas de estas se extrajeron los permisos de cada APK.
- En [4] se propuso un marco de trabajo para extracción de variables para entrenamiento de modelos de machine learning para detección de ciberamenazas. De los resultados se consiguió modelos clasificatorios que cuentan con un desempeño que van desde el 92%-94% en identificación.

[3] Urcuqui. Christian, & Cadavid, A. N. (2016, April). Machine learning classifiers for android malware analysis. In Communications and Computing (COLCOM), 2016 IEEE Colombian Conference on(pp. 1-6). IEEE.

[4] López, U., Camilo, C., García Peña, M., Osorio Quintero, J. L., & Navarro Cadavid, A. (2018). Ciberseguridad: un enfoque desde la ciencia de datos-Primera edición.

ANDROID - DATASET



TALLER GUÍA

Paso 1.

Como primer paso descargue el conjunto de datos (archivo csv) de alguno de los siguientes enlaces:

- https://ieee-dataport.org/documents/dataset-malwarebeningn-permissions-android
- https://www.kaggle.com/xwolf12/datasetandroidpermissions

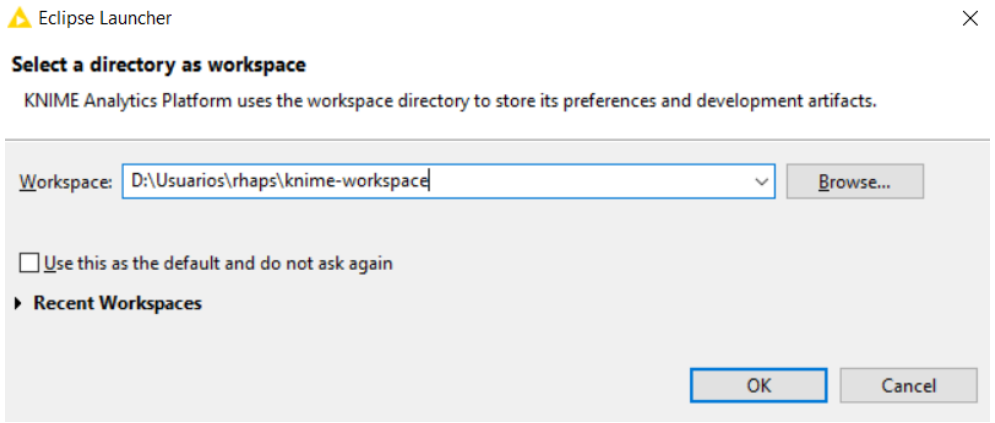
TALLER GUÍA

Paso 2.

- Descargue e instale el software KNIME Analytics

Paso 3.

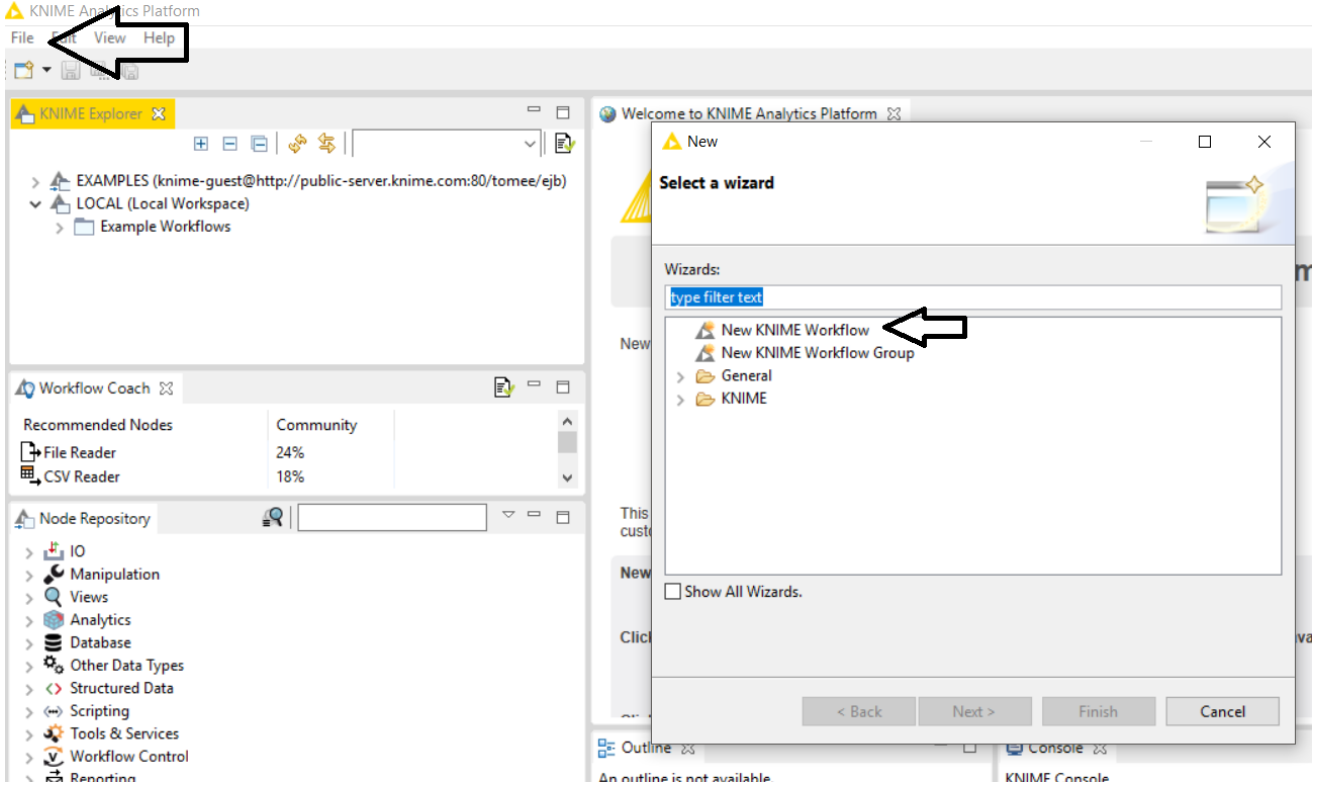
- Ejecute el software KNIME y asigne un entorno de trabajo



TALLER GUÍA

Paso 4.

- En la pestaña ***File*** seleccione la opción ***new workflow***



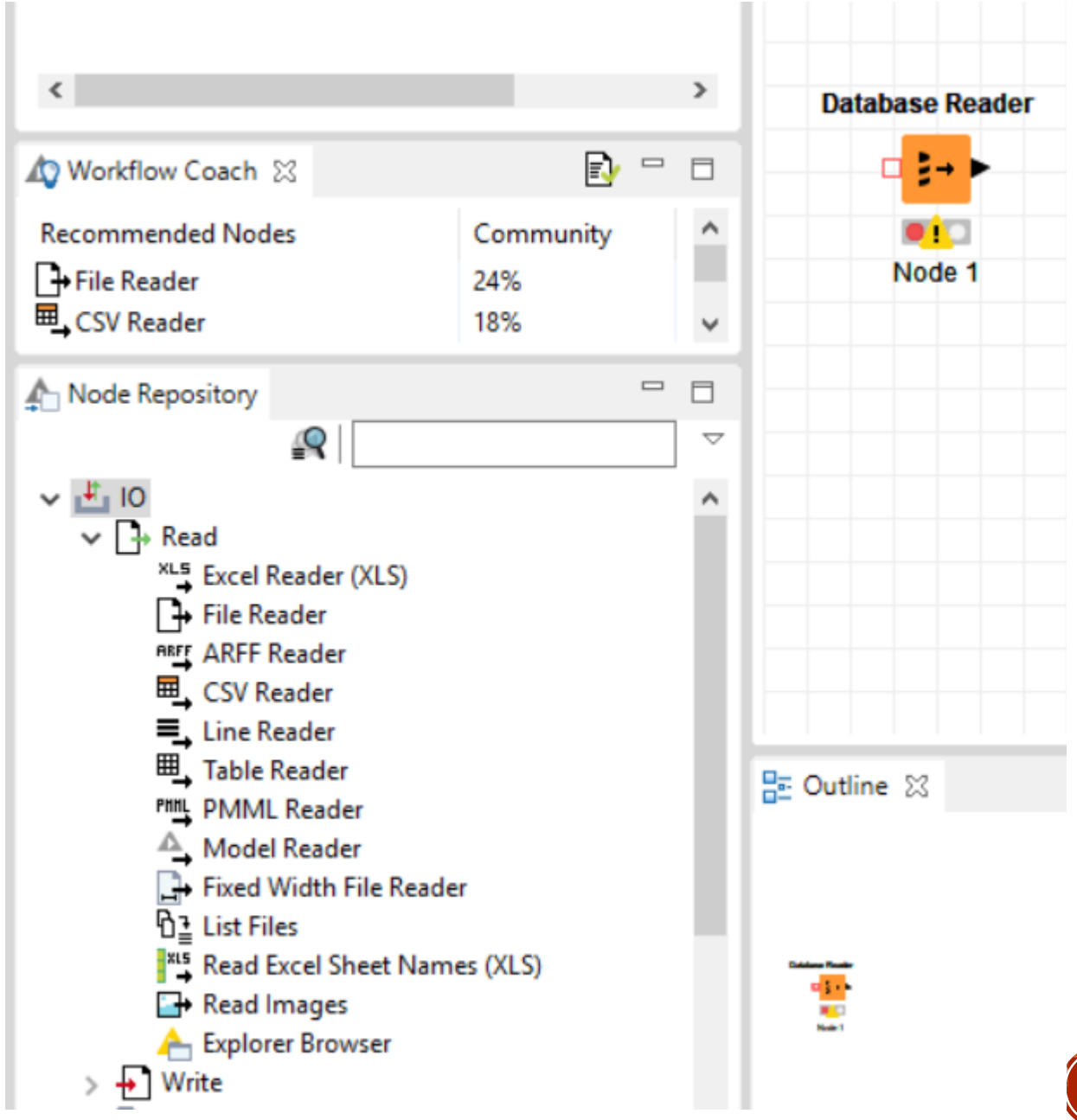
TALLER GUÍA

Paso 5.

- Asigne un nombre a su flujo de trabajo

Paso 6.

- Seleccione y arrastre al entorno de trabajo un ***CSV Reader*** que lo puede encontrar en ***Node Repository -> IO -> Read -> CSV Reader***



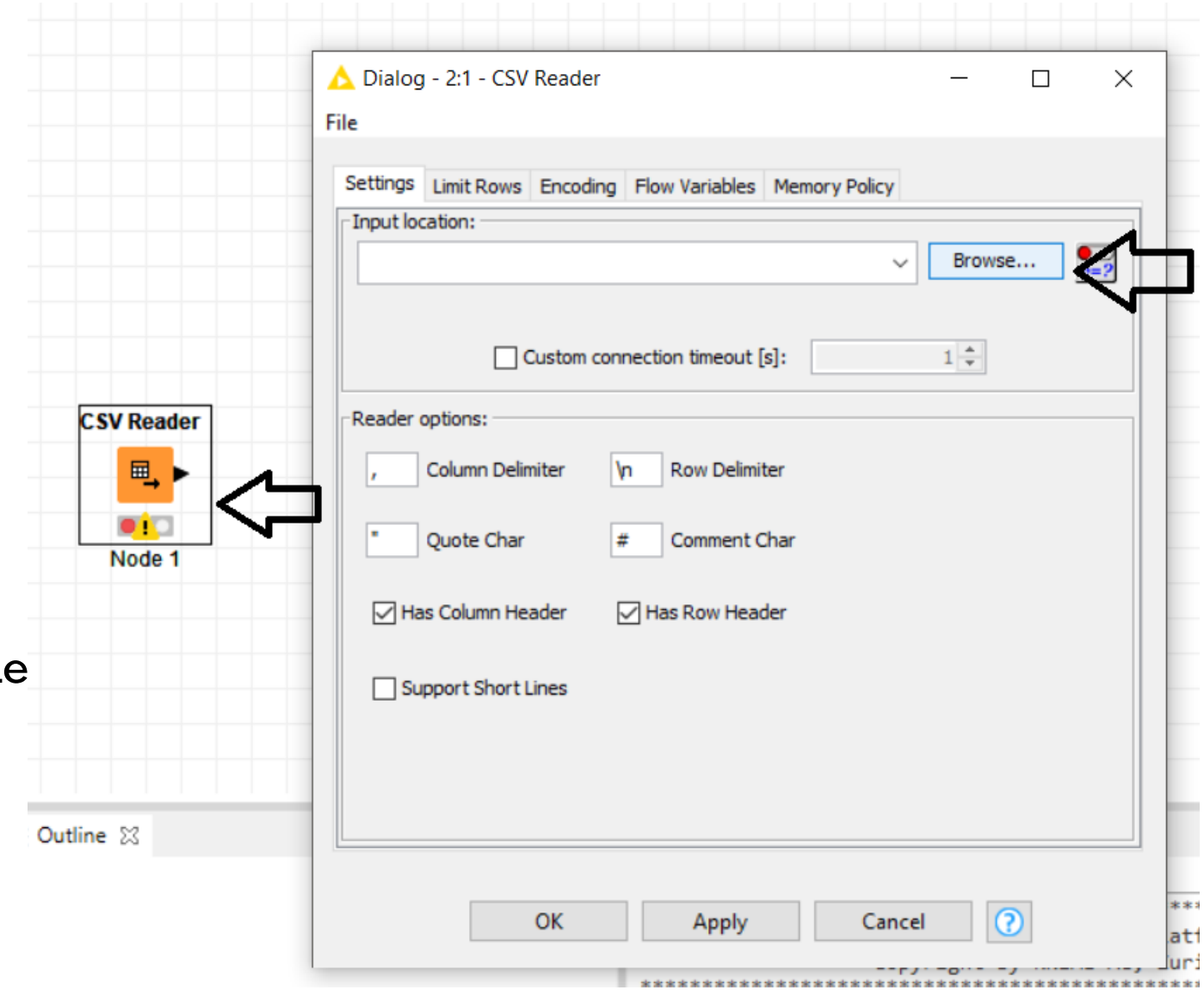
TALLER GUÍA

Paso 7.

- Presione click derecho sobre el nodo ***Database Reader*** y luego seleccione la opción ***Configure***

Paso 8.

- En la opción ***Browse…*** busque el directorio y el archivo train.csv que descargo



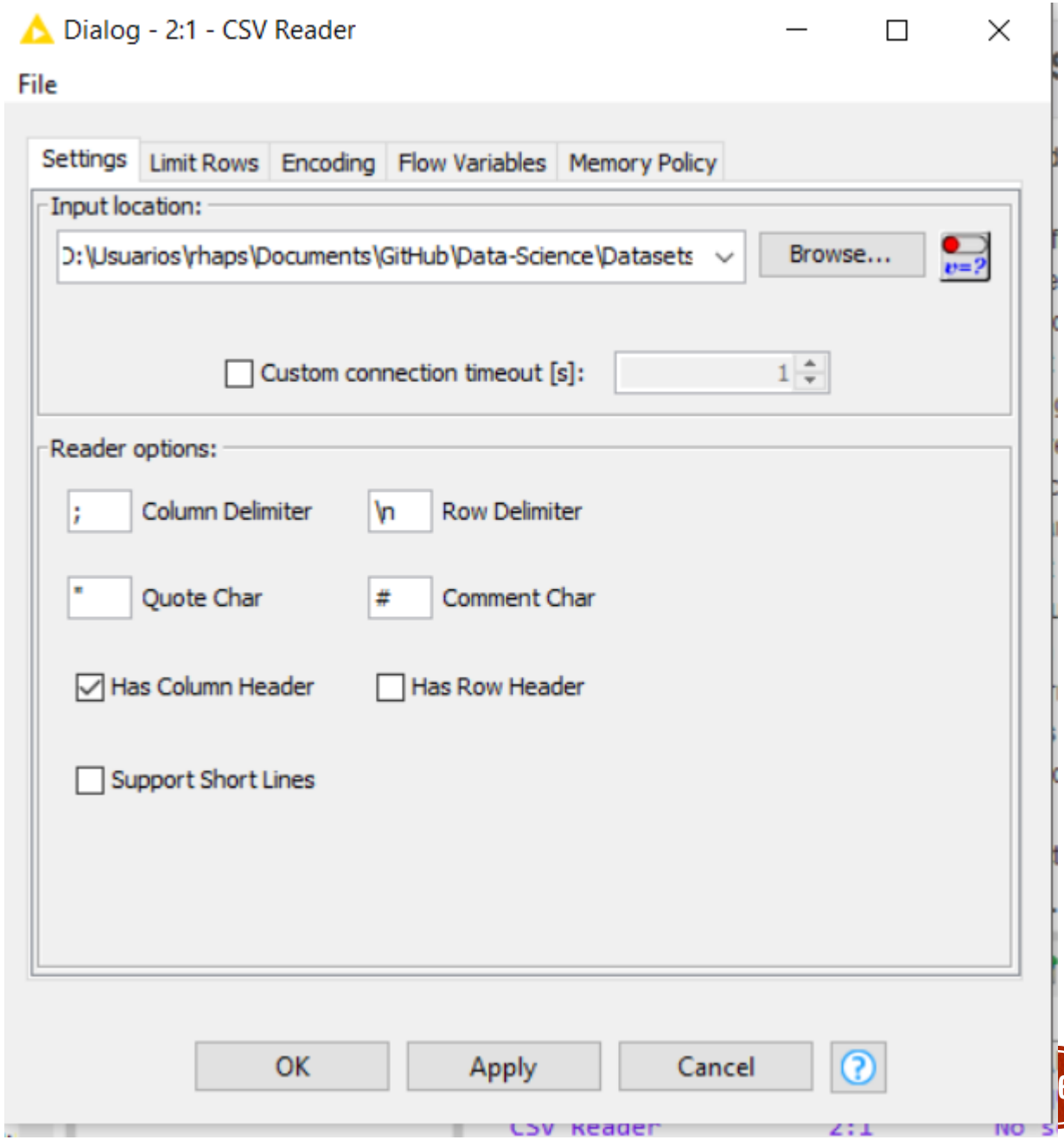
TALLER GUÍA

Paso 9.

- El archivo csv cuenta con cada variable separada por punto y coma, verifique que cuenta con la siguiente configuración

Paso 10.

- Luego de click en ***Apply*** y ***Ok***



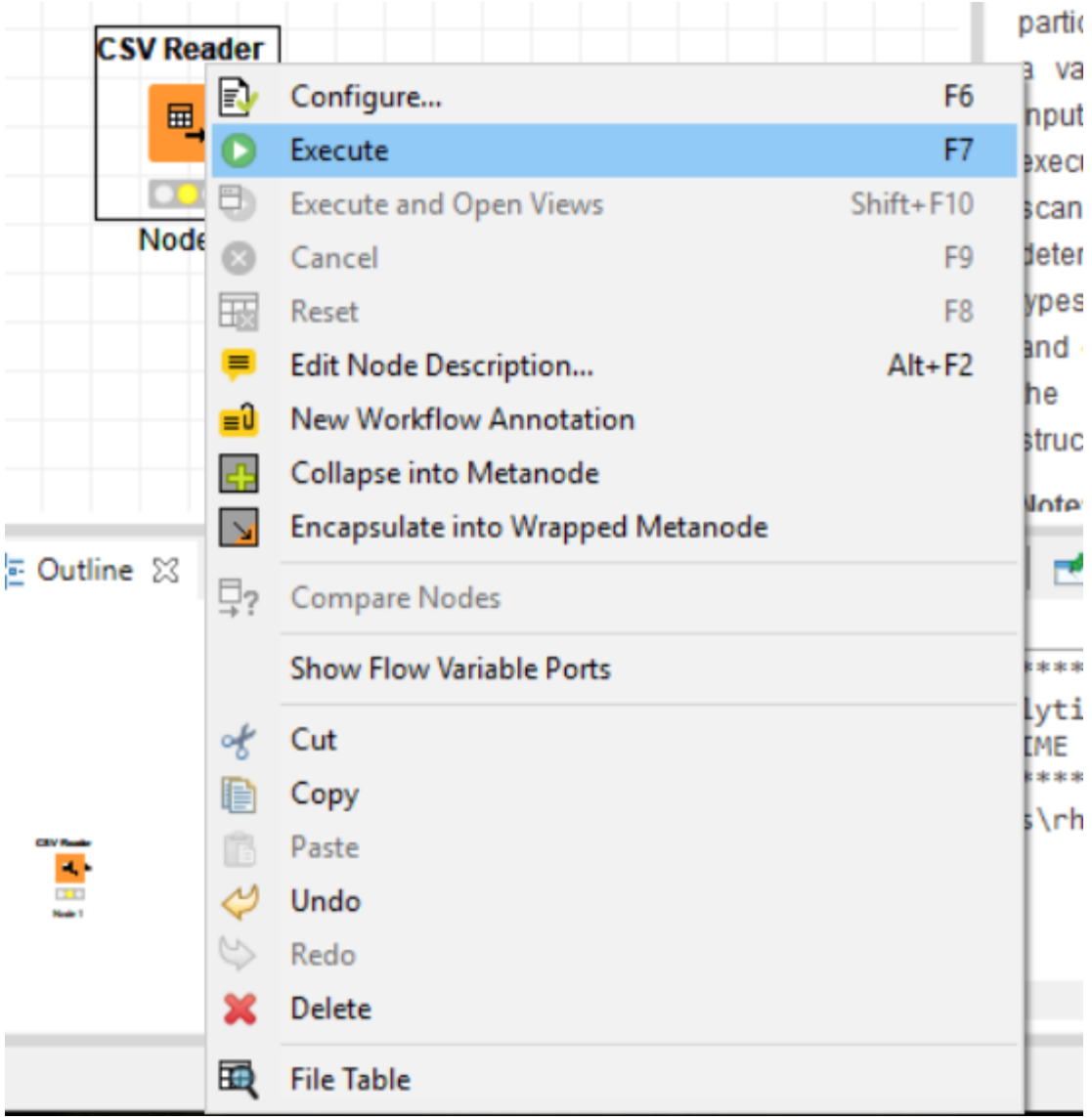
TALLER GUÍA

Paso 11.

- Vuelva y de click sobre el nodo ***CSV Reader*** *(note que el color del semáforo ahora esta en amarillo)*

Paso 12.

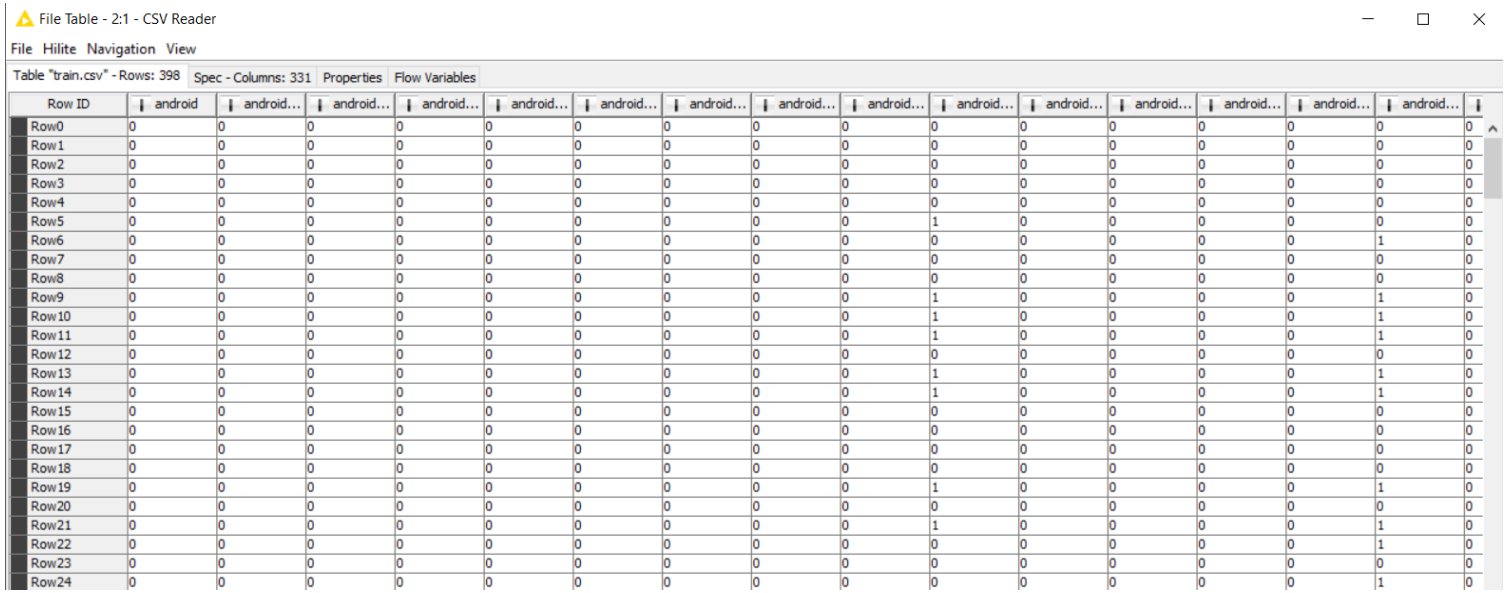
- Ahora de la opción ***Execute*** con el fin de que el semáforo ahora pase al estado listo, es decir, en color verde.



TALLER GUÍA

Paso 13.

- Ahora vuelva a dar click derecho sobre el nodo pero ahora utilice la opción ***File Table***



TALLER GUÍA

Ahora tenemos un vistazo de los datos cargados en nuestro entorno de trabajo.

El csv se encuentra conformado por un conjunto de variables binarias (columnas) donde sus valores representan:

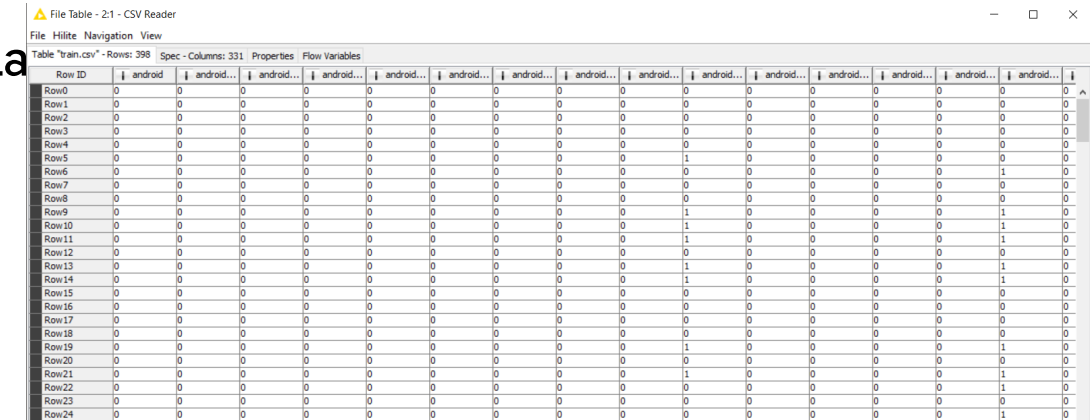
- 1, Si una aplicación accedió al permiso
- 0 , Si una aplicación no accedió al permiso

El dataset cuenta con una columna llamada ***type*** que representa

- 1, si el registro (fila) es de una aplicación maliciosa
- 0, si el registro (fila) es de una aplicación benigna

En conclusión, podemos decir que tenemos

un conjunto de datos de permisos por aplicación



TALLER GUÍA

Ahora la pregunta de ciencia de datos es:

- ¿Es posible determinar si un APK es un malware o no a través de sus permisos?

TALLER GUÍA – ANÁLISIS EXPLORATORIO

Paso 14.

Recuerde que podemos utilizar técnicas estadísticas y de visualización.

Para un primer acercamiento utilizaremos una tabla de frecuencias para conocer cuales son los permisos más accedidos tanto para aplicaciones maliciosas y benignas.

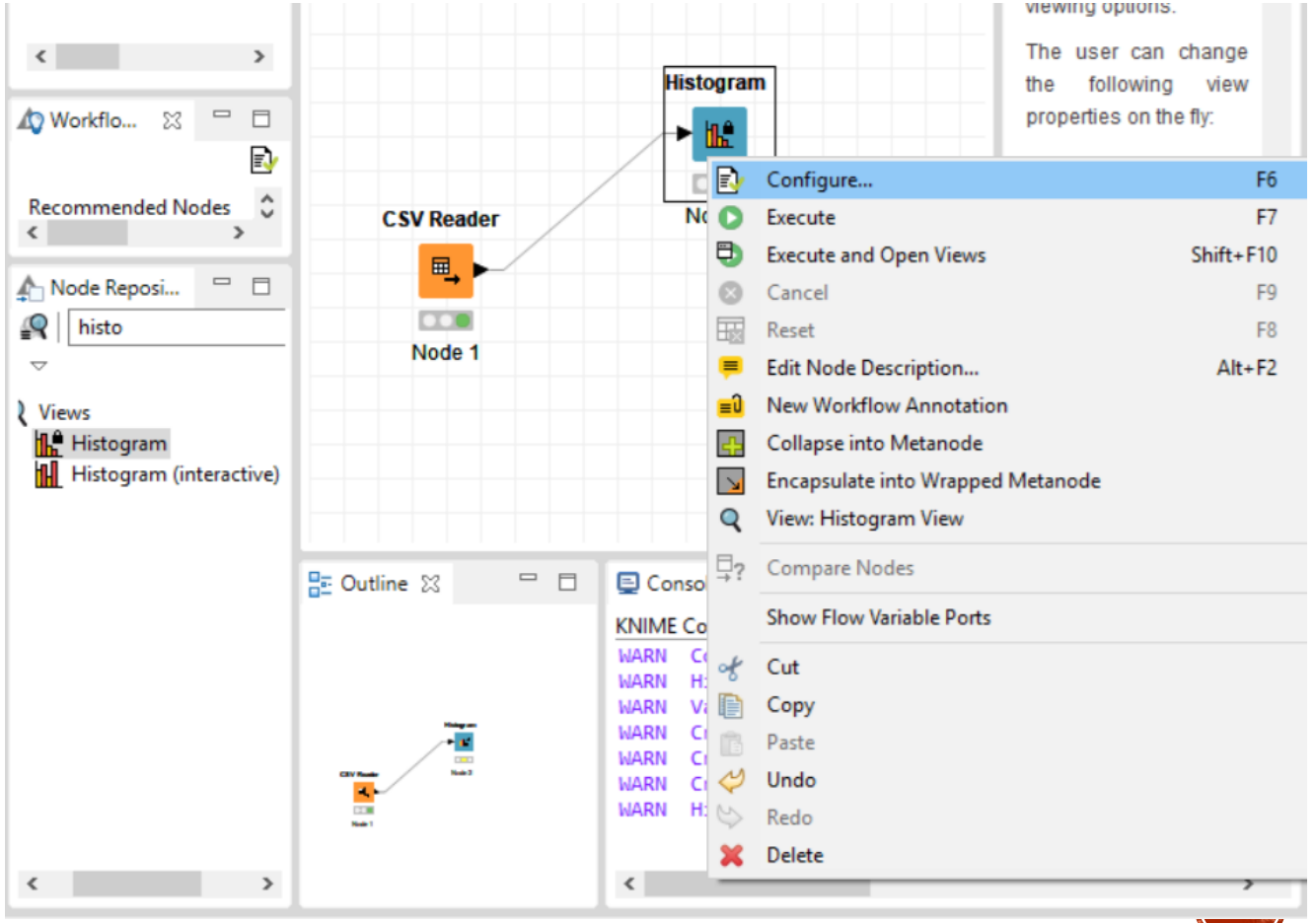
TALLER GUÍA – ANÁLISIS EXPLORATORIO

Paso 14.

Vamos a agregar un nodo ***Histogram***

Conecte la salida del nodo ***CSV Reader*** al ***Histogram.***

Luego accedamos a las configuraciones del nodo ***Histogram.***



TALLER GUÍA – ANÁLISIS EXPLORATORIO

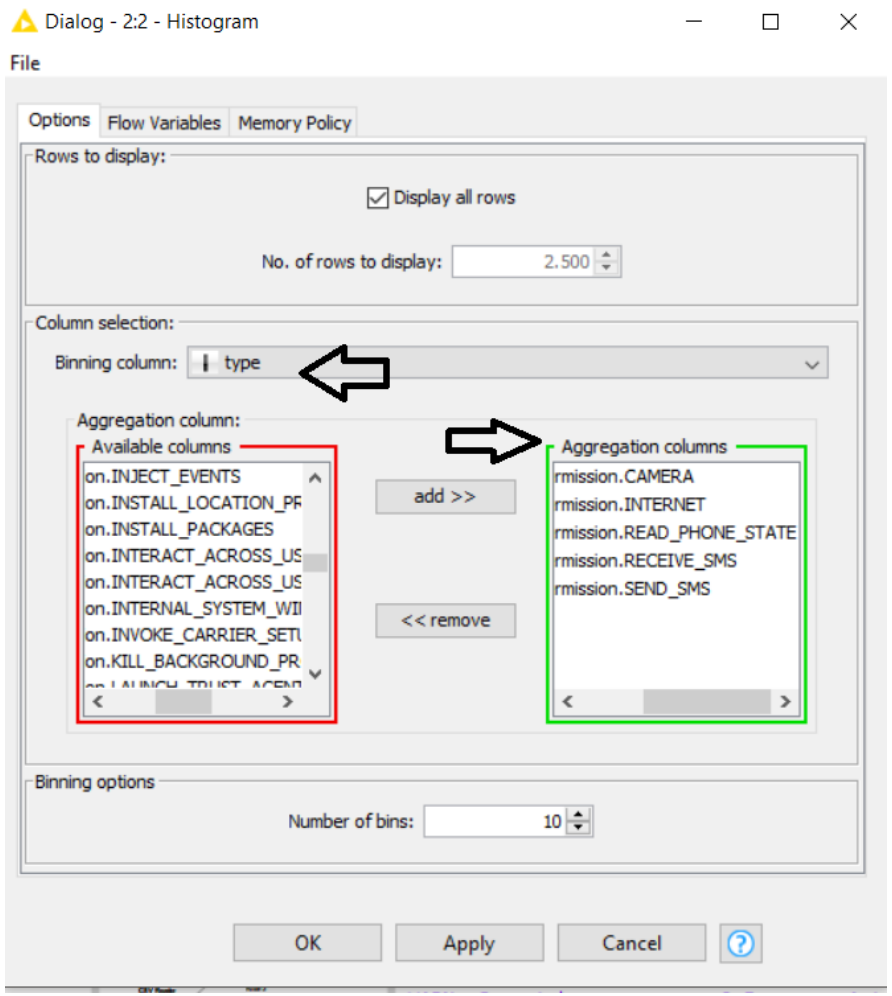
Paso 15.

Observe las configuraciones, el ***Binning column*** esta asignada a la variable objetivo ***Type***.

Vamos a realizar un análisis sobre las variables:

- Camera
- Internet
- Read_Phone_State
- Receive_SMS
- Send_SMS

Seleccione las variables y presione ***add >>***



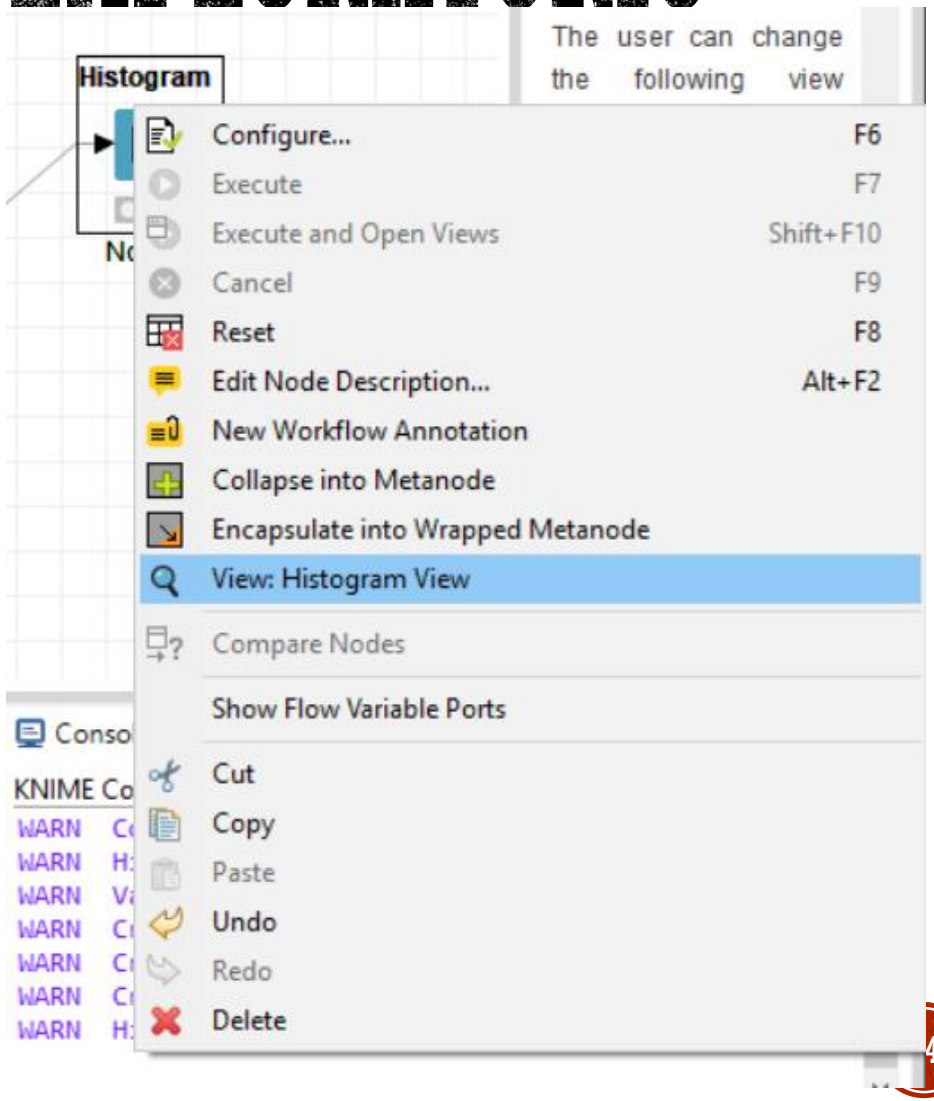
TALLER GUÍA – ANÁLISIS EXPLORATORIO

Paso 16.

Luego de click en ***Apply*** y en ***Ok***. Posteriormente, ejecute el nodo de Histogram para que su estado se encuentre en listo (color verde).

Paso 17.

Acceda a las opciones del nodo ***Histogram*** y de click sobre ***View: Histogram View***



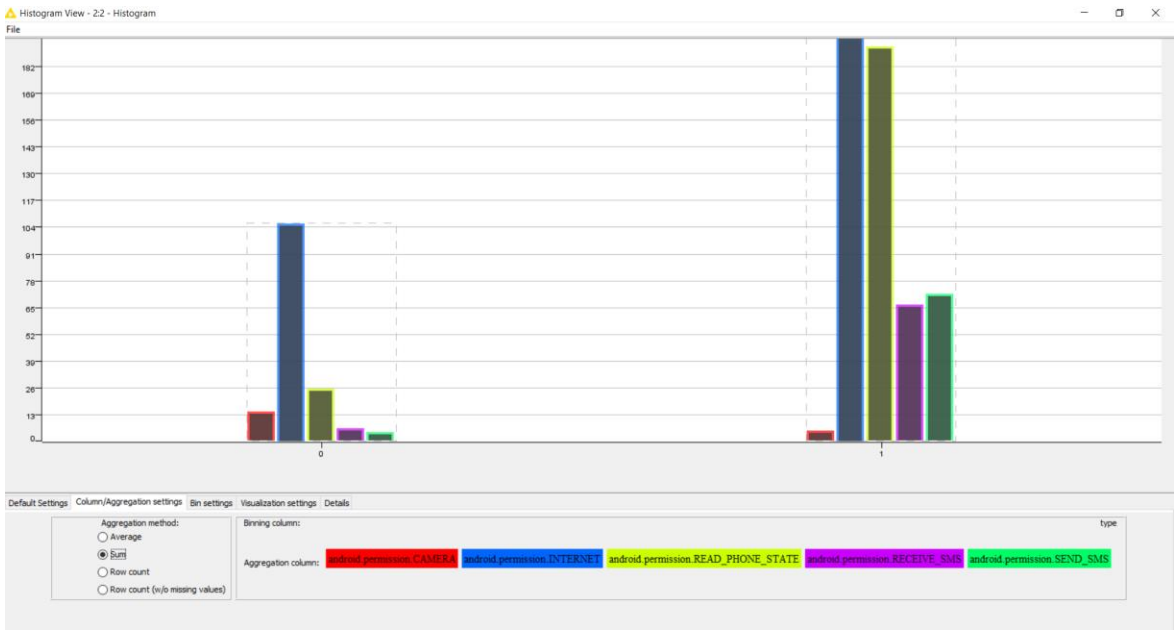
TALLER GUÍA – ANÁLISIS EXPLORATORIO

Paso 18.

En las opciones del Histogram vaya hasta la pestaña ***Column/Aggregation settings*** y cambie la opción del ***Aggregation Method*** al modo ***Sum.***

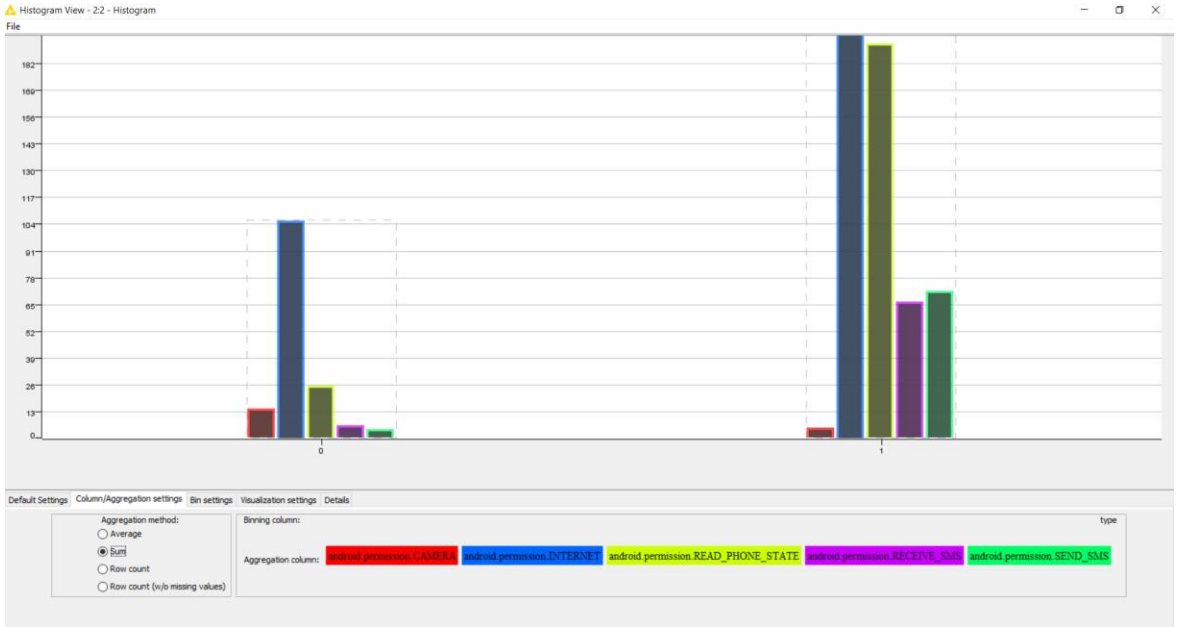
Paso 19.

Podrá conseguir un gráfico de barras de permisos accedidos por tipo de aplicación.



TALLER GUÍA – ANÁLISIS EXPLORATORIO

- ¿Las aplicaciones maliciosas tienden a conectarse más Internet, enviar mensajes SMS y ver el estado del dispositivo?
- Según los datos la respuesta es ***Si***



TALLER GUÍA – MACHINE LEARNING

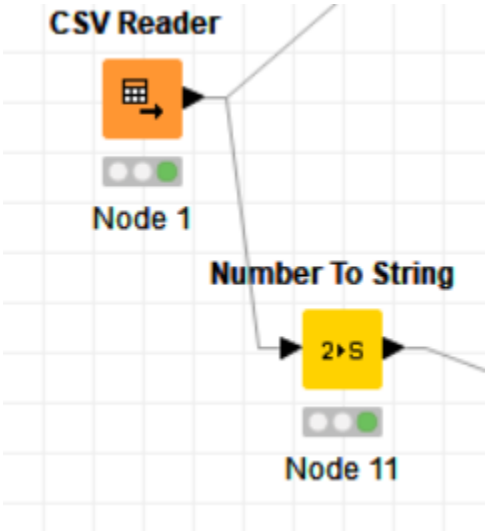
- Para esta tarea aplicaremos un entrenamiento supervisado, es decir, de antemano conocemos que aplicaciones son malware y cuales son benignas.

Paso 20.

- Adicione y asigne a la salida de CSV un nodo tipo

Number to String que nos permitirá transformar

nuestros datos a tipo categóricos



TALLER GUÍA – MACHINE LEARNING

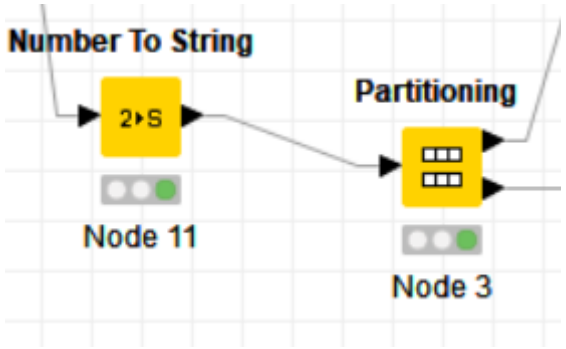
- Para esta tarea aplicaremos un entrenamiento supervisado, es decir, de antemano conocemos que aplicaciones son malware y cuales son benignas.

Paso 20.

- Adicione un nodo ***Partitioning***
- Conecte la entrada con la salida

del nodo ***Number To String***

- Configure relative en 80%

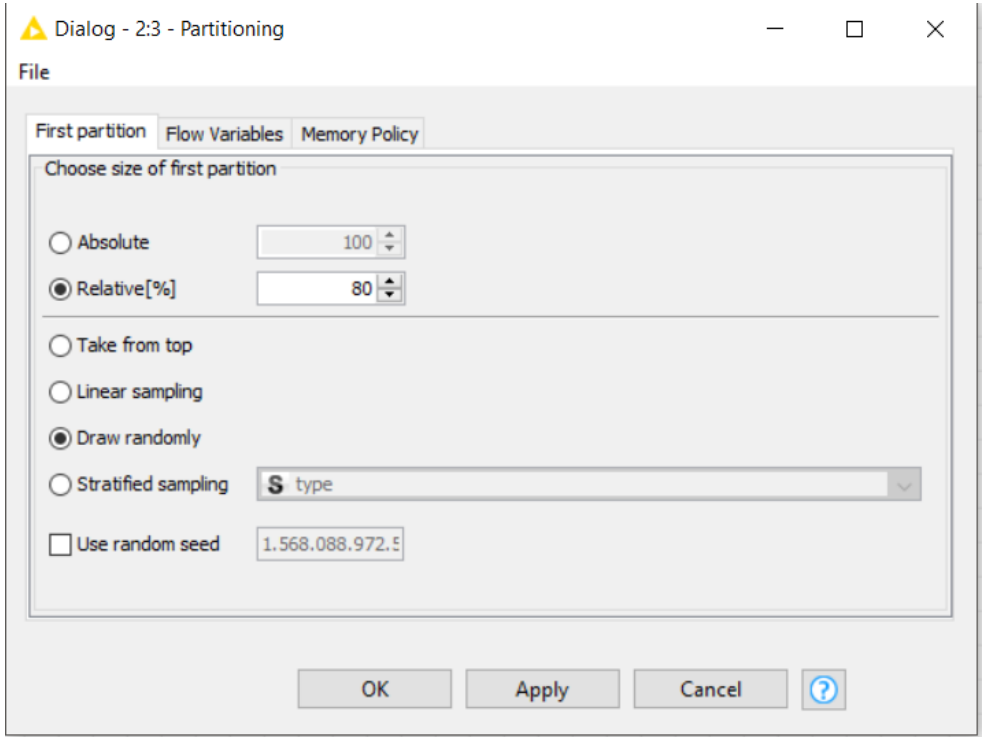


TALLER GUÍA – MACHINE LEARNING

- Con la configuración en relative, estamos diciendo que los datos los vamos a dividir aleatoriamente en dos conjuntos, uno para entrenamiento (80%) y otro para testeo (20%)
- El conjunto de entrenamiento

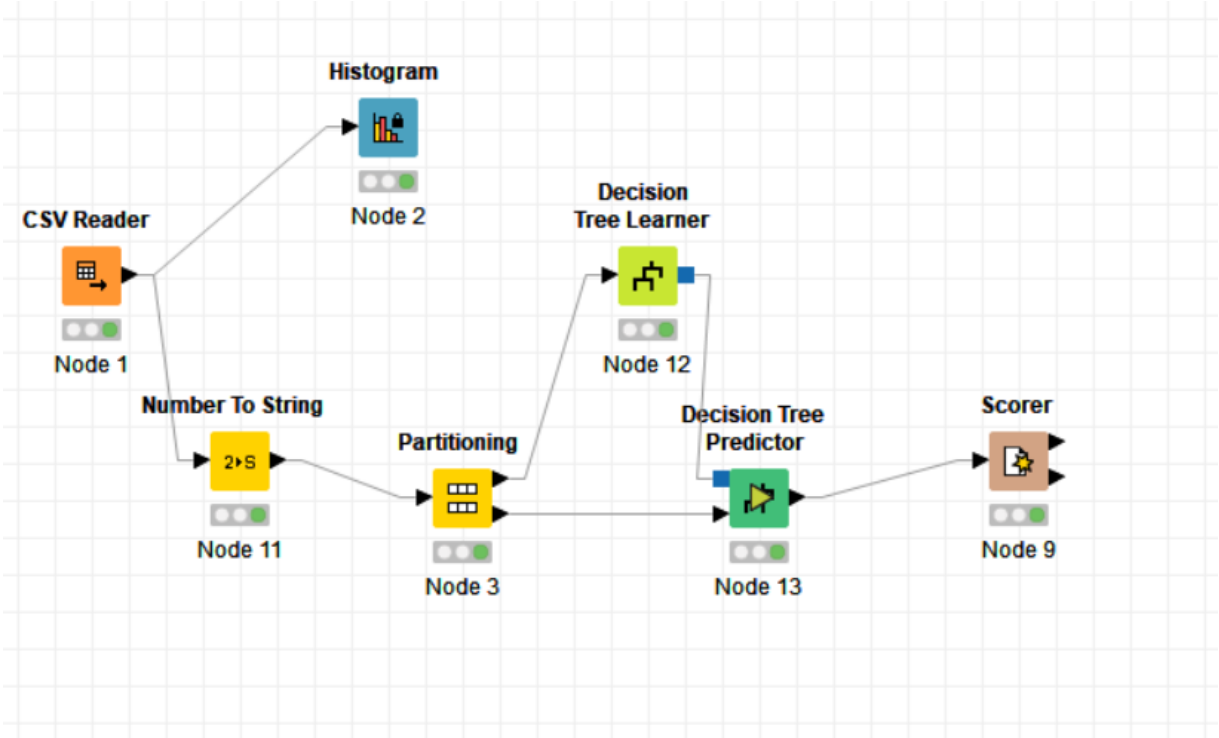
tendrá la variable ***Type*** mientras

que el de testeo no.



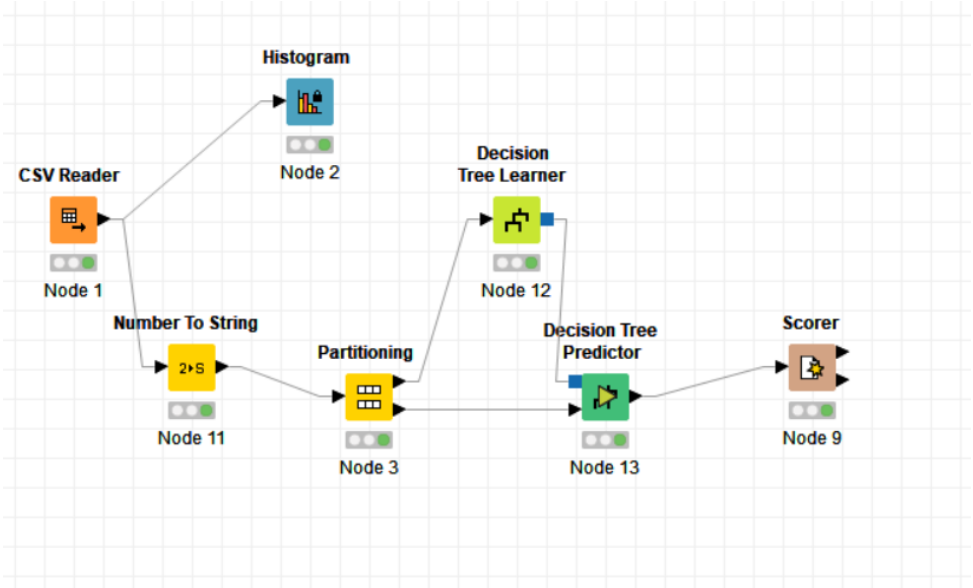
TALLER GUÍA – MACHINE LEARNING

- Complete el flujo de trabajo agregando los nodos ***Decision Tree Learner***, ***Decision Tree Predictor*** y ***Scorer***. Conecte cada nodo mencionado como aparece en la siguiente figura.



TALLER GUÍA – MACHINE LEARNING

Para este caso estamos utilizando un tipo de clasificador de machine learning conocido como decisión tree o arboles de decisión, hemos agregado un nodo ***learner*** encargado del entrenamiento del modelo y un ***predictor*** para la evaluación del modelo entrenado contra el conjunto de testeo.



TALLER GUÍA – MACHINE LEARNING

Ejecute los nodos y observe las propiedades del nodo ***Scorer.***

- De los resultados podemos conocer que el modelo tuvo un desempeño del 91% y tuvo un error del 0,87%.