UNIVERSIDAD DE GRANADA

Tratamiento Inteligente de Datos

Práctica 6

Autor

Antonio José Muriel Gálvez

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Granada, 31 de diciembre de 2024

Índice

Enunciado

Se propone aplicar un tratamiento inteligente de datos para la toma de decisiones respecto a evitar accidentes mortales de tráfico.

El objetivo es construir un modelo para predecir si los accidentes tendrán víctimas mortales o no.

Los datos disponibles se encuentran en el archivo traffic_fatality.csv y constan de 28,390 instancias de accidentes de tráfico. Los atributos a considerar son:

- weekday: Día de la semana del accidente (1: lunes ... 7: domingo).
- Age: Edad del conductor.
- Gender: Sexo del conductor.
- Alcohol Results: Resultados de análisis de alcoholemia.
- Drug Involvement: Resultados de análisis de drogas.
- Atmospheric Condition: Condiciones atmosféricas.
- Roadway: Tipo de carretera.
- Fatality: Clase objetivo:
  - fatal = accidente con víctima mortal.
  - no fatal = accidente sin víctima mortal.

Se debe determinar cuál modelo es más apropiado para la tarea, basado en el análisis comparativo de las métricas obtenidas.

Introducción

Implementar varios modelos de clasificación utilizando KNIME para predecir si un accidente de tráfico tendrá víctimas mortales (Fatality). Se emplearon múltiples algoritmos y se compararon para seleccionar el más adecuado.

Este proyecto involucró la limpieza de datos, preprocesamiento, y entrenamiento de modelos como Decision Tree, Naive Bayes, SVM, y KNN.

Solución Knime

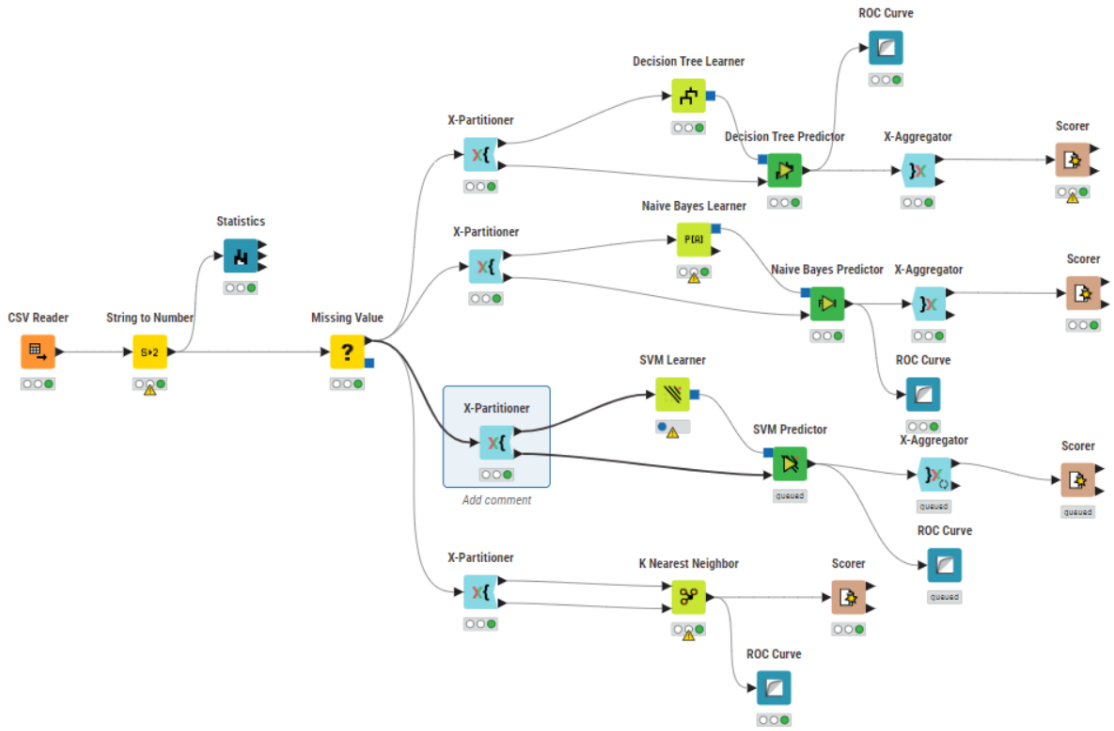

Figura 1: Solución KNIME

Nodo: CSV Reader

Este nodo carga el archivo de entrada (traffic_fatality.csv) en el entorno de trabajo. Permite especificar la ubicación del archivo, interpretar correctamente los delimitadores y asignar nombres a las columnas.

- Se define el delimitador (coma en este caso) y la codificación del archivo (UTF-8).
- Se revisan los tipos de datos iniciales detectados por el nodo para asegurarse de que correspondan con el significado de los atributos.

Nodo: String to Number

Convierte atributos categóricos, almacenados inicialmente como cadenas de texto, en valores numéricos.

Seleccione la columna Alcohol_Results para poder procesar los datos como valores numéricos.

Nodo: Statistics

Exploración inicial para identificar valores faltantes y distribuciones.

- Age: 347 valores faltantes.
- Alcohol_Results: 17,627 valores faltantes (62 %).
- Clase objetivo: Desbalanceada (70 % no_fatal).

Nodo: Missing Value

Este nodo gestiona los valores faltantes en el conjunto de datos, un paso crucial para evitar que los modelos de clasificación se vean afectados por la falta de información.

Para los valores faltantes de la edad se ha optado por usar la mediana y para los valores faltantes de Alcohol_Results la media, la cual tiene un valor de 0.076.

Este enfoque balancea la preservación de la información con la necesidad de evitar el sesgo que podría introducir la eliminación directa de filas o columnas.

Nodo: X-Partitioner

Divide el conjunto de datos en múltiples subconjuntos para realizar validación cruzada. Esto asegura que los modelos sean evaluados de manera robusta y evita el sobreajuste.

Se utilizó una validación cruzada de 10 pliegues. En cada iteración, 80 % de los datos se utilizan para entrenar el modelo y el 20 % restante para evaluar su desempeño.

Este enfoque permite evaluar el rendimiento promedio de los modelos, proporcionando métricas confiables para la comparación.

Construcción de Modelos

Decision Tree Learner y Predictor

Construye un modelo basado en árboles de decisión y lo aplica a los datos de prueba. Los árboles de decisión segmentan iterativamente el espacio de atributos en función de umbrales que maximizan la ganancia de información.

Se utilizó el criterio de "Gini Index"para seleccionar las divisiones en los nodos. Profundidad máxima limitada para evitar el sobreajuste.

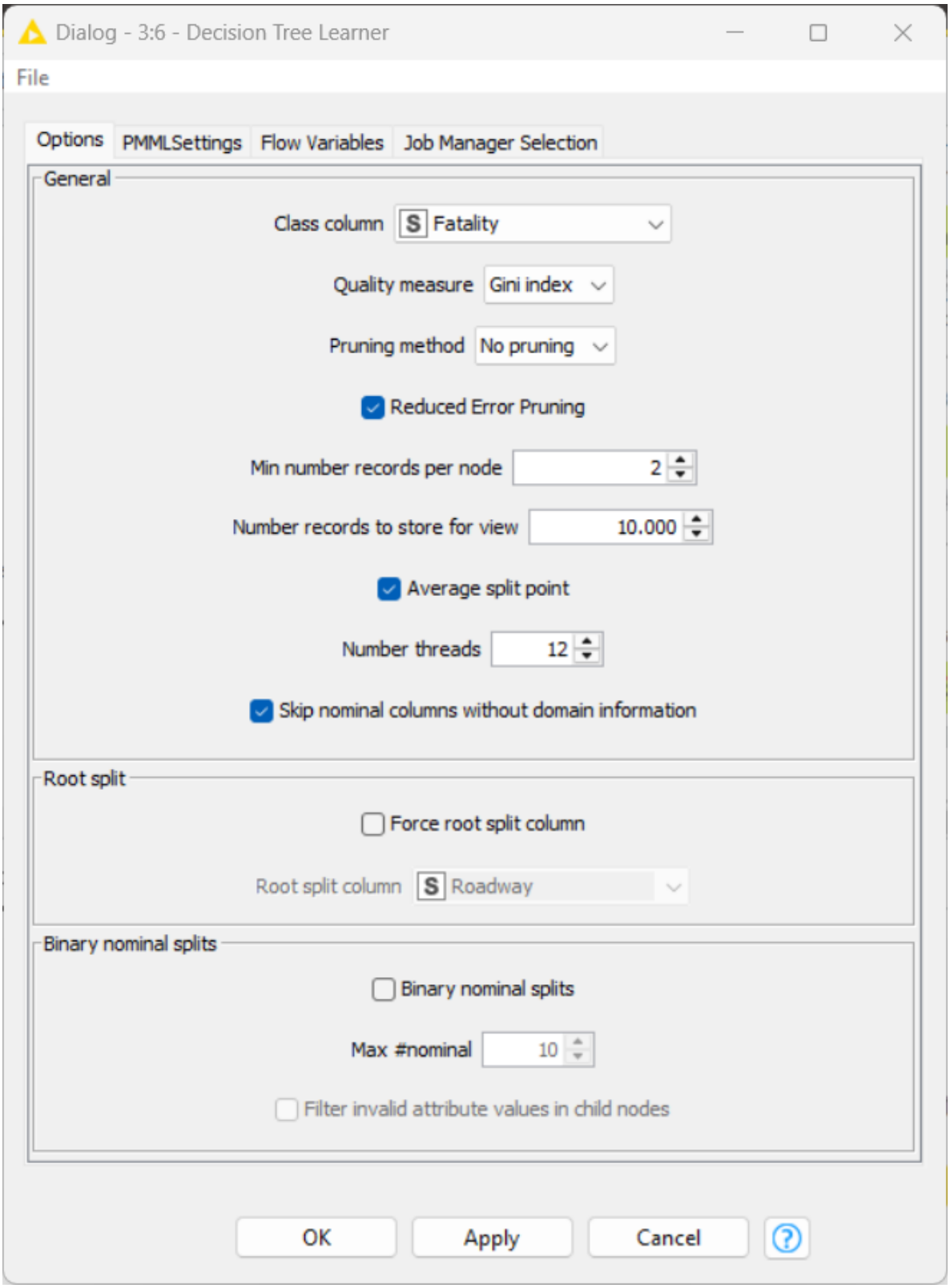

Figura 2: Decision Tree Learner

Naive Bayes Learner y Predictor

Entrena un modelo probabilístico basado en el Teorema de Bayes, asumiendo independencia condicional entre atributos. Este modelo es particularmente eficiente para conjuntos de datos con muchas características categóricas.

Se utilizó la configuración predeterminada, dado que el algoritmo es robusto y requiere poca parametrización.

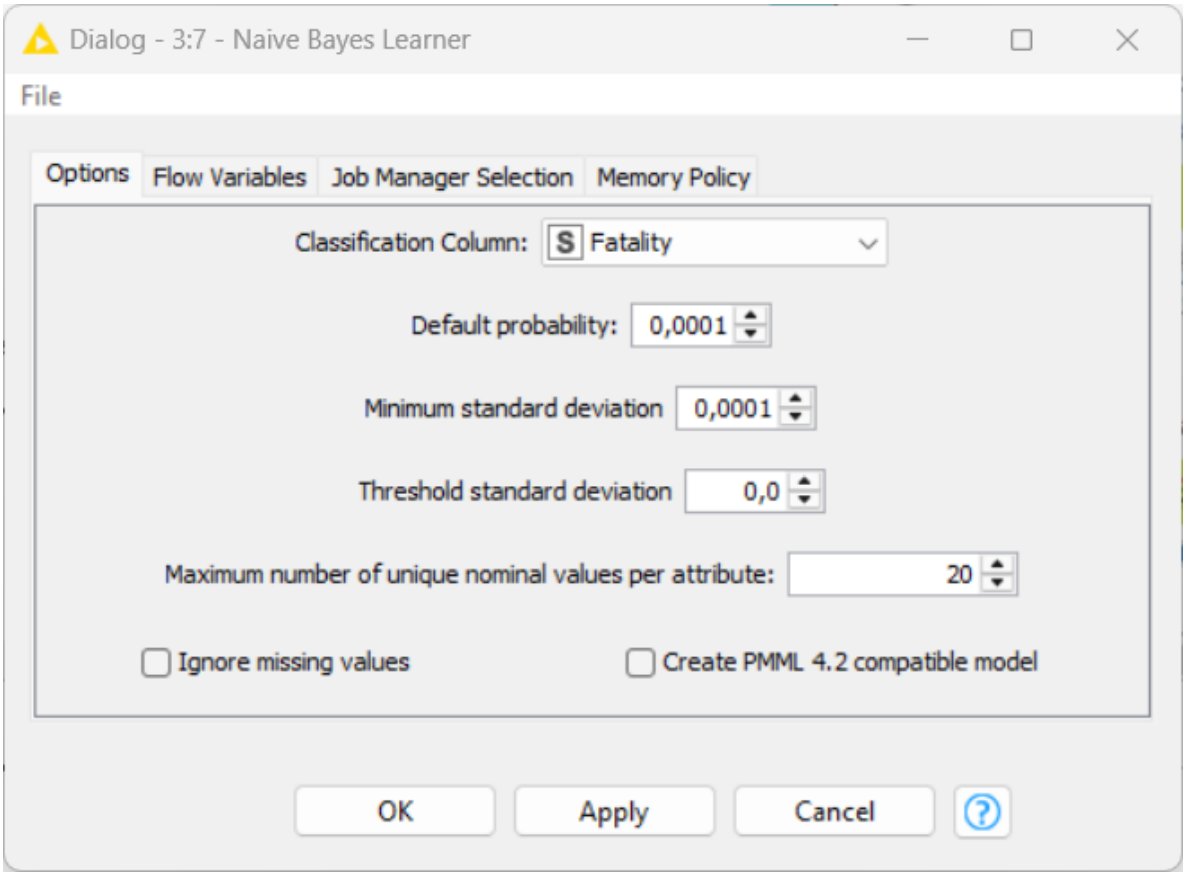

Figura 3: Native Bayes Learner

SVM Learner y Predictor

Implementa Máquinas de Soporte Vectorial (SVM) para encontrar un hiperplano óptimo que separe las clases en el espacio de características.

Se utilizó un kernel RBF (Radial Basis Function) para capturar relaciones no lineales. Parám. Se exploraron valores de gamma para maximizar la capacidad predictiva.

El SVM es especialmente útil en problemas con clases desbalanceadas y características complejas.

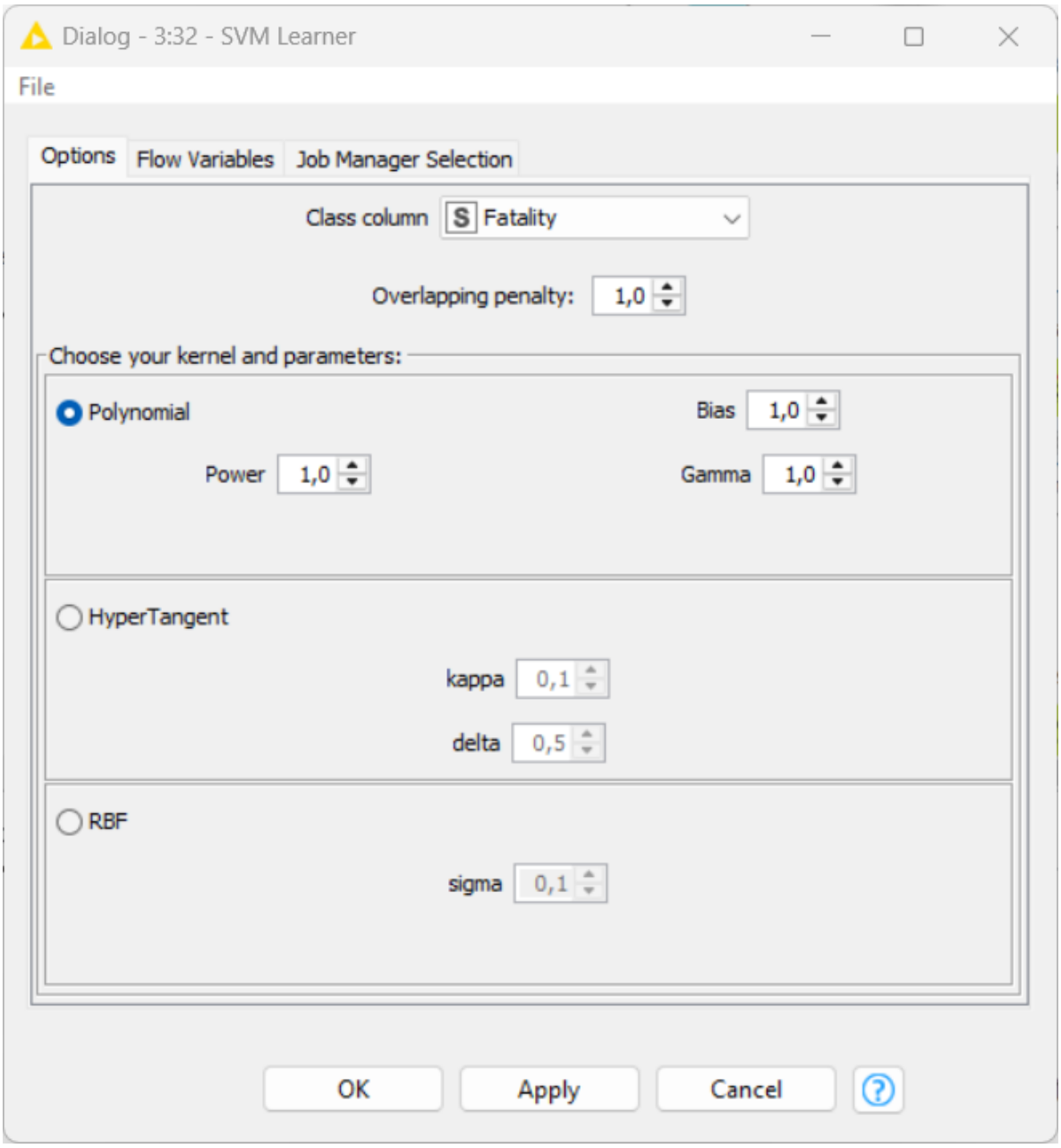

Figura 4: SVM Learner

K Nearest Neighbor

Clasifica instancias en función de la clase mayoritaria entre sus k vecinos más cercanos. Este modelo es sencillo pero efectivo para datos distribuidos espacialmente.

Número de vecinos (k): Se probó con diferentes valores, siendo k=5 el valor que ofreció mejores resultados en términos de precisión.

Métrica de distancia: Se utilizó la distancia euclidiana para determinar la cercanía entre instancias.

Este modelo es particularmente útil cuando los datos presentan una estructura local fuerte.

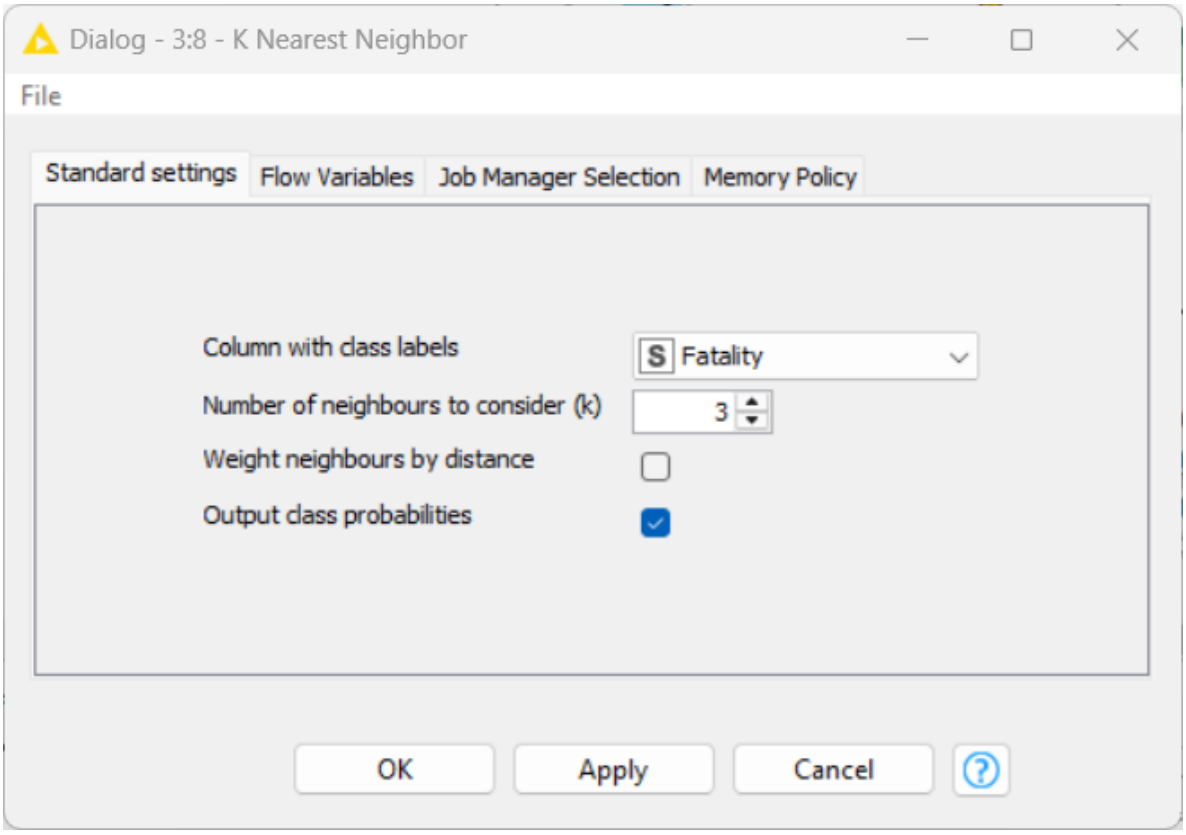

Figura 5: K Nearest Neighbor

Curvas ROC

Decision Tree

Una curva con mayor AUC (cercana a 1) indica mejor desempeño predictivo. Aquí el modelo parece diferenciar bien entre las clases cuando se trata de Fatality=no_fatal, mientras que Fatality=fatal tiene un AUC bajo, indicando dificultades en distinguir correctamente casos fatales.

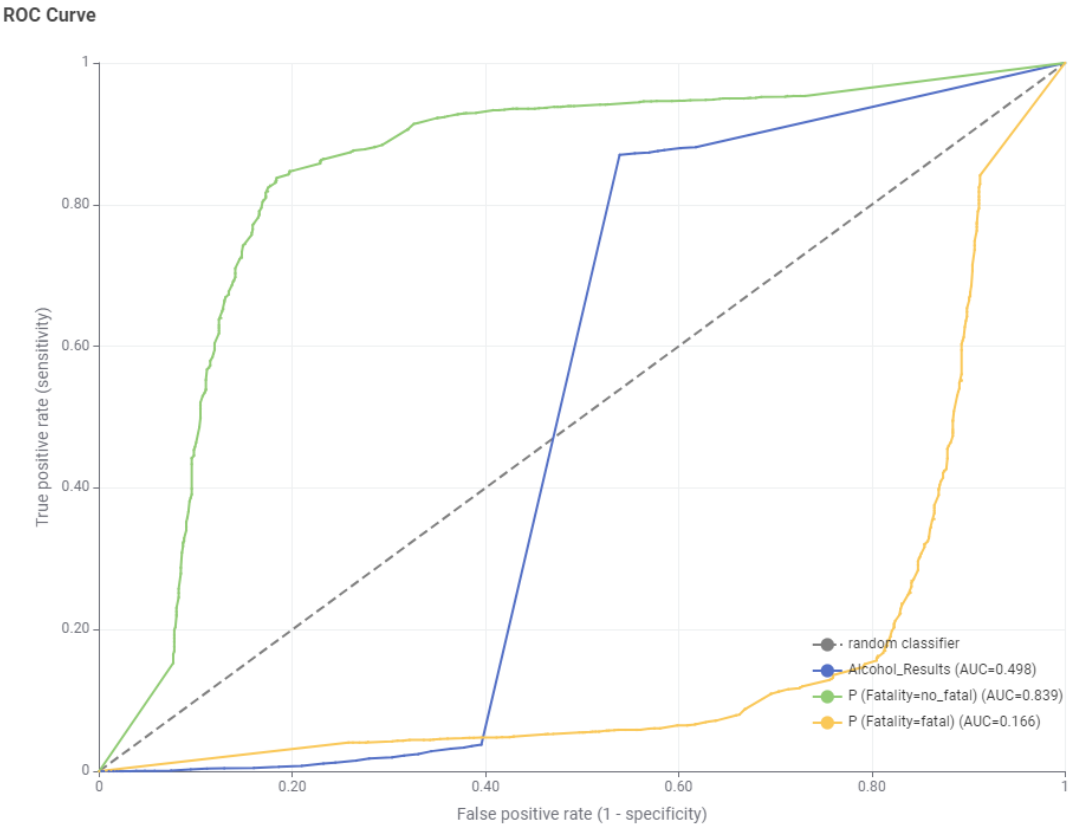

Figura 6: ROC Curve - Decision Tree

Naive Bayes

Similar al árbol de decisión, el modelo tiene un desempeño aceptable para Fatality=no_fatal con un AUC elevado, pero pobre para Fatality=fatal.

La curva de Fatality=no_fatal tiene un buen comportamiento, pero la sensibilidad para Fatality=fatal muestra un aumento gradual con muchos falsos positivos.

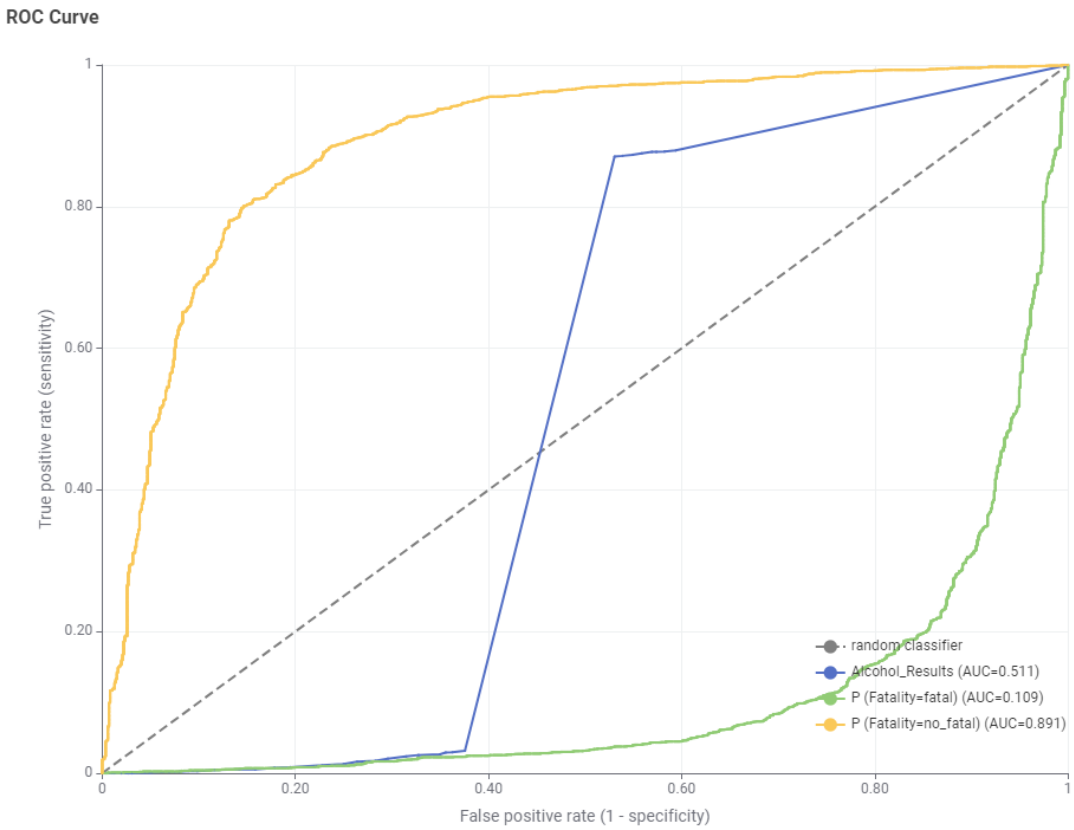

Figura 7: ROC Curve - Naive Bayes

K Nearest Neighbor

La AUC para Fatality=no_fatal es significativamente alta, lo que sugiere que este modelo es bueno para predecir los casos no fatales.

Sin embargo, para Fatality=fatal, el AUC es muy bajo, indicando problemas con la predicción de esta clase.

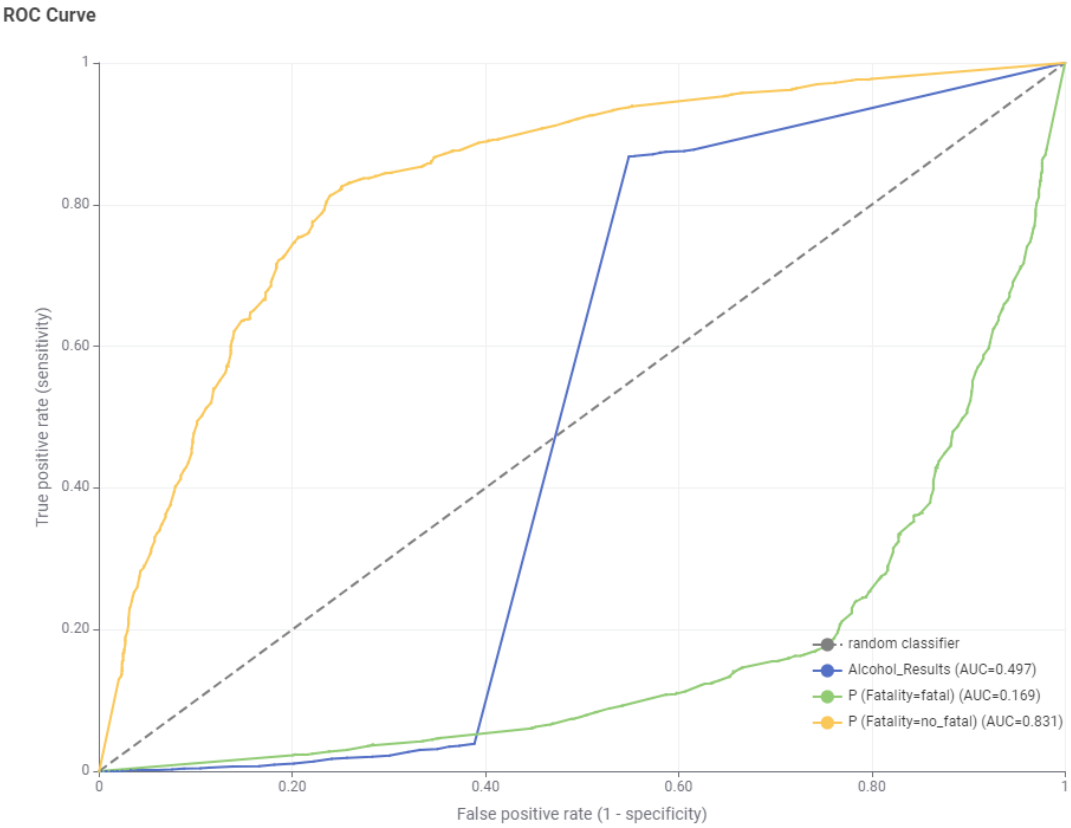

Figura 8: ROC Curve - K Nearest Neighbor

Evaluación de los modelos

Decision Tree

- Verdaderos negativos (no_fatal predicho como no_fatal): 17,497
- Falsos positivos (no_fatal predicho como fatal): 2,111
- Falsos negativos (fatal predicho como no_fatal): 2,662
- Verdaderos positivos (fatal predicho como fatal): 5,917

Naive Bayes

- Verdaderos negativos (no_fatal predicho como no_fatal): 18,258
- Falsos positivos (no_fatal predicho como fatal): 1,454
- Falsos negativos (fatal predicho como no_fatal): 3,036
- Verdaderos positivos (fatal predicho como fatal): 5,643

K Nearest Neighbor

- Verdaderos negativos (no_fatal predicho como no_fatal): 1,771
- Falsos positivos (no_fatal predicho como fatal): 181
- Falsos negativos (fatal predicho como no_fatal): 407
- Verdaderos positivos (fatal predicho como fatal): 481

Análisis comparativo

Modelo	**Accuracy**	**Precision (fatal)**	**Recall (fatal)**	**F1-score**
Decision Tree	82.6 %	73.7 %	69.0 %	71.2 %
Naive Bayes	84.9 %	79.5 %	65.0 %	71.4 %
K Nearest Neighbor	85.0 %	72.6 %	54.2 %	62.1 %

Tabla 1: Métricas de evaluación de modelos en la predicción de fatalidad.

- Naive Bayes tiene el mejor accuracy y precision, pero el recall es moderado.
- Decision Tree tiene el recall más alto, lo que lo hace bueno para capturar casos fatales.
- K Nearest Neighbor muestra un F1-score menor, lo que indica que no es tan equilibrado como los otros.