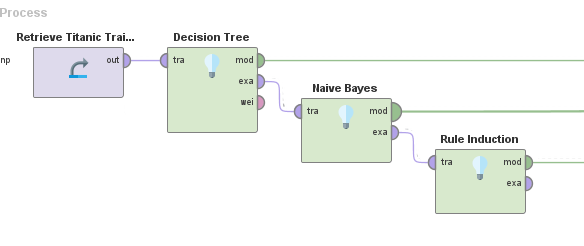
|  |
| --- |
| **Introducción a los Métodos de Aprendizaje Automático** |
| **Ejercicios domiciliarios**  **UT2 – PD2** |
| **Autor:** Gerardo Fernández - CI: 2858230-7  **Docente:** Ernesto Ocampo |
| **04/09/2021** |
|  |

**Ejercicio 1**

**1. Modeling**

En este tutorial se parte de un dataset de Titanic “limpio” y etiquetado, se aplican 3 modelos para ver que relaciones se pueden descubrir sobre los datos y las chances de sobrevivir de un pasajero, que de sólo mirar los datos no hubiéramos descubierto.

*The decision tree clearly shows that family size matters more than passenger class for women. This behavioral pattern could not be detected for men. In general, men had a lower likelihood to survive ("women and children first"). The easiest way to see this is in the Chart visualization of the Naive Bayes model. Although usually not the most accurate type of models, in general, the rule set is an easy to read format which can be useful when you want to interpret the models.*

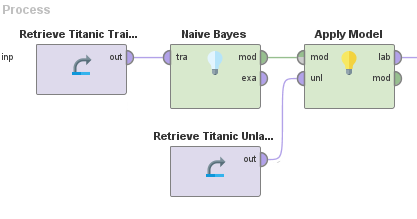


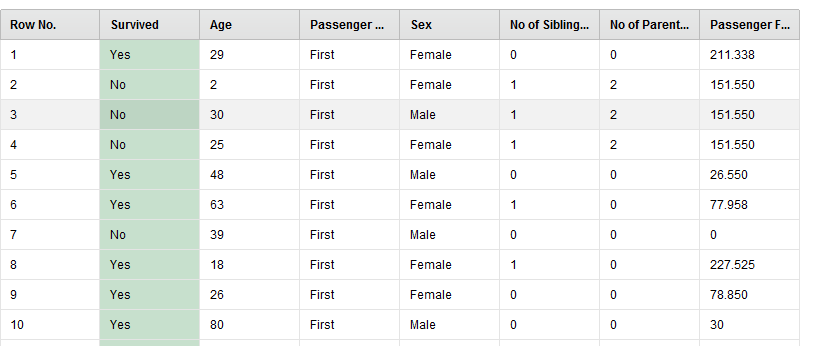
**2. Scoring**

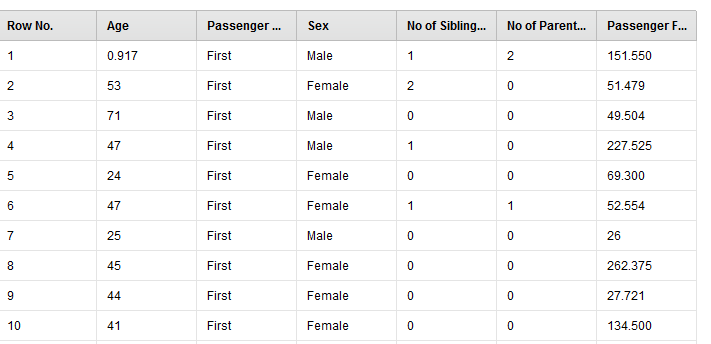
En este caso, se entrena un modelo con ciertos datos etiquetados, y luego se se aplica a un un test data set, es decir un conjunto de datos sin etiquetar para predecir la clase Survived (yes, no) de cada pasajero con su respectivo nivel de confianza (probabilidad de Yes y No).

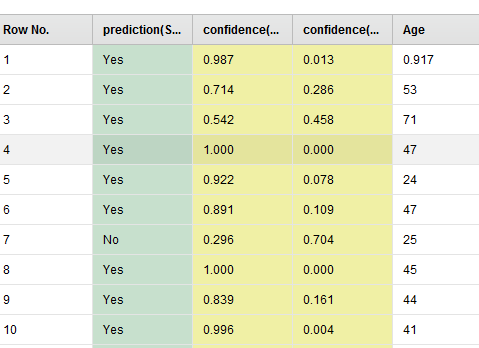
El Apply Model crea predicciones para cada ejemplo del conjunto sin etiquetar. Lo que se obtiene de Rapidminer es el mismo conjunto de Test pero con una columna “Survived” y dos columnas adicionales con el grado de confianza de cada clase de la predicción.

Importante: El formato de ambos conjuntos de datos (training y test) debe ser el mismo. Utilizar los mismo atributos y en lo posible los mismos rangos de valores. Algunos modelos son más sensibles que otros a los cambios en los datos.



**Training dataset**

**Testing dataset**



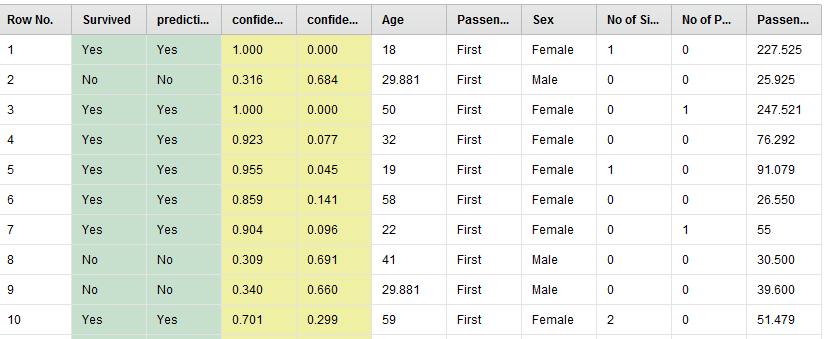
**Predicción**

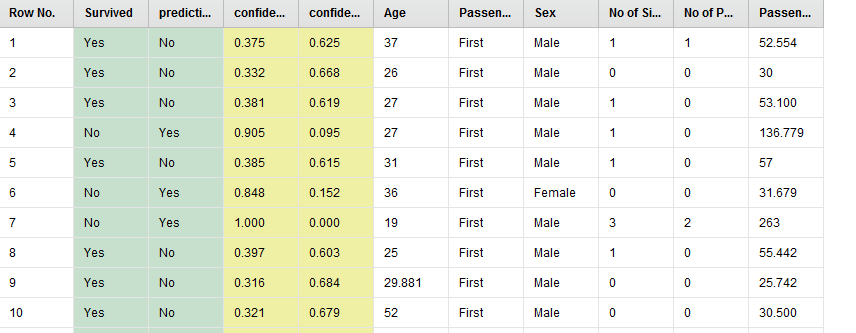
**3. Test Splits and Validation**

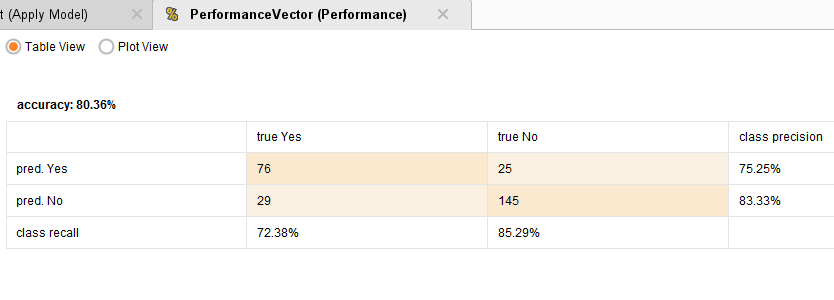
En este caso se plantea la pregunta de qué tan bien ejecutará mi modelo con datos de producción, es decir datos que nunca vio ?

Para averiguarlo se debe validar el modelo. Esto se logra guardando parte de los datos etiquetados y no utilizarlos para crear el modelo. Como los datos estan etiquetados, al final puedo comparar la predicción obtenida con el valor real y obtener el error de clasificación o la precisión del modelo.

En este caso, el dataset de test viene con las mismas columnas del scoring pero además se agrega la etiqueta original “Survived” que contiene los valores reales. Con esto se calcula la exactitud del modelo.

**Predicciones correctas**

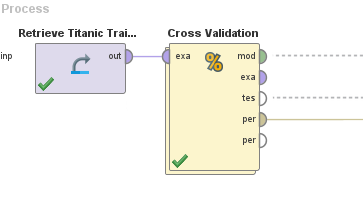
**Predicciones incorrectas**

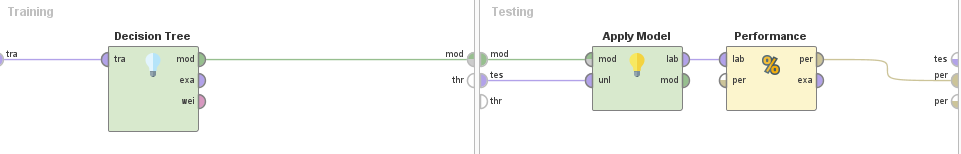
**Performance del modelo**

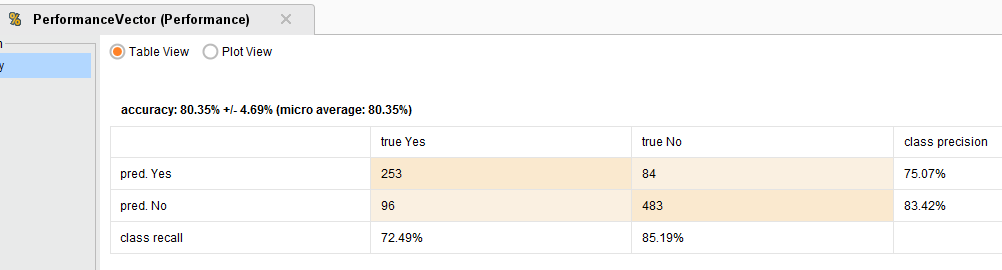
**4. Cross Validation**

Para asegurarnos que no haya diferencias en los datos utilizados tanto en el training como en el test, y asegurarnos que cada ejemplo se haya usado en ambos conjuntos se utiliza la técnica de cross validation.

En este caso, la performance tiene un dato adicional, la desviación estándar de la exactitud de la cross validation. Cuanto más chica más robusto es el modelo, es decir, menos sensible a los cambios en los datos.







**5. Visual Model Comparison**