UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS
CIENCIAS DE LA COMPUTACIÓN

MACHINE LEARNING
Laboratorio Ensamble Learning
(Primer Semestre del 2019)

Objetivos de aprendizaje:

- Combinar modelos de clasificación.

---

1. Actividad en Weka

1.1. Predicción usando J48 (10 minutos)

El conjunto de datos `glass.arff` contiene los siguientes atributos:

- RI (refractive index)
- Na (Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10))
- Mg (Magnesium)
- Al (Aluminum)
- Si (Silicon)
- K (Potassium)
- Ca (Calcium)
- Ba (Barium)
- Fe (Iron)

Al atributo meta es `Type` el cual puede tener los siguientes valores:

- building_windows_float_processed
- building_windows_non_float_processed
- vehicle_windows_float_processed
- vehicle_windows_non_float_processed (none in this database)
- containers
- tableware
- headlamps

Se le pide que entrene un modelo de clasificación para este conjunto de datos usando el algoritmo `J48` usando validación cruzada con folds de 10.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)

1.2. Aplicando método ensemble Bagging (20 minutos)

Usando el mismo conjunto de datos `glass.arff` se le pide que realice la clasificación pero ahora usando el clasificador bagging. Para esto realice los siguientes pasos:

- Seleccione el algoritmo `Bagging` en la carpeta de clasificadores `meta`.
- En los parámetros del algoritmo `Bagging`, en el parámetro `classifier` seleccione el algoritmo `J48`.
- En los parámetros del algoritmo `Bagging`, en el parámetro `printClassifiers` seleccione la opción `True`.
- Entrene el modelo usando validación cruzada con folds de 10.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)
- ¿El modelo mejora al usar bagging?
- Pruebe el algoritmo bagging usando otros algoritmos como por ejemplo `Naive Bayes`, `SMO`, entre otros. ¿El modelo mejora al usar bagging?

1.3. Aplicando el algoritmo RandomForest (10 minutos)

Usando el mismo conjunto de datos `glass.arff` se le pide que realice la clasificación pero ahora usando el algoritmo `RandomForest`. Para esto realice los siguientes pasos:

- Seleccione el algoritmo `RandomForest` en la carpeta de clasificadores `trees`.
- En los parámetros del algoritmo `RandomForest`, en el parámetro `printClassifiers` seleccione la opción `True`.
- Entrene el modelo usando validación cruzada con folds de 10.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)
- ¿En qué se diferencia del algoritmo `Bagging`?

1.4. Aplicando método ensemble Boosting (20 minutos)

Usando el mismo conjunto de datos `glass.arff` se le pide que realice la clasificación pero ahora usando el clasificador boosting. Para esto realice los siguientes pasos:

- Seleccione el algoritmo `AdaBoost.M1` en la carpeta de clasificadores `meta`.
- En los parámetros del algoritmo `AdaBoost.M1`, en el parámetro `classifier` seleccione el algoritmo `J48`.
- Entrene el modelo usando validación cruzada con folds de 10.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)
- ¿El modelo mejora al usar boosting?
- Pruebe el algoritmo boosting usando otros algoritmos como por ejemplo `Naive Bayes`, `SMO`, entre otros. ¿El modelo mejora al usar boosting?
- ¿Qué diferencia encuentra entre bagging y boosting? En la práctica, ¿cuál es mejor?

1.5. Aplicando método ensemble Stacking (20 minutos)

Usando el mismo conjunto de datos `glass.arff` se le pide que realice la clasificación pero ahora usando el clasificador stacking. Para esto realice los siguientes pasos:

- Seleccione el algoritmo `Stacking` en la carpeta de clasificadores `meta`.
- En los parámetros del algoritmo `Stacking`, en el parámetro `classifiers` seleccione los algoritmo `NaiveBayes` y `J48`.
- En los parámetros del algoritmo `Stacking`, en el parámetro `MetaClassifier` seleccione el algoritmo `J48`.
- Entrene el modelo usando validación cruzada con folds de 10.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)
- ¿Qué diferencia encuentra entre bagging, boosting y stacking?

2. Actividad en RapidMiner

2.1. Predicción usando Decision Tree (10 minutos)

El conjunto de datos `Pima Indians Diabetes Database`, que se encuentra en el archivo `diabetes.csv`, posee los siguientes atributos, todos ellos numéricos:

- preg (Number of times pregnant)
- plas (Plasma glucose concentration a 2 hours in an oral glucose tolerance test)
- pres (Diastolic blood pressure (mm Hg))
- skin (Triceps skin fold thickness (mm))
- insu (2-Hour serum insulin (mu U/ml))
- mass (Body mass index (weight in kg/(height in m)^2))
- pedi (Diabetes pedigree function)
- age (Age (years))

Al atributo meta es `class` el cual puede tener los siguientes valores:

- tested_positive
- tested_negative

Se le pide que entrene un modelo de clasificación para este conjunto de datos usando el algoritmo `Decision Tree` usando validación cruzada con folds de 10.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)

2.2. Aplicando método ensemble Bagging (20 minutos)

Usando el mismo conjunto de datos `diabetes.csv` se le pide que realice la clasificación pero ahora usando el clasificador bagging. Para esto realice los siguientes pasos: configure el diseño principal conforme la figura 1. Recuerde que al leer el CVS debe indicar el atributo meta.

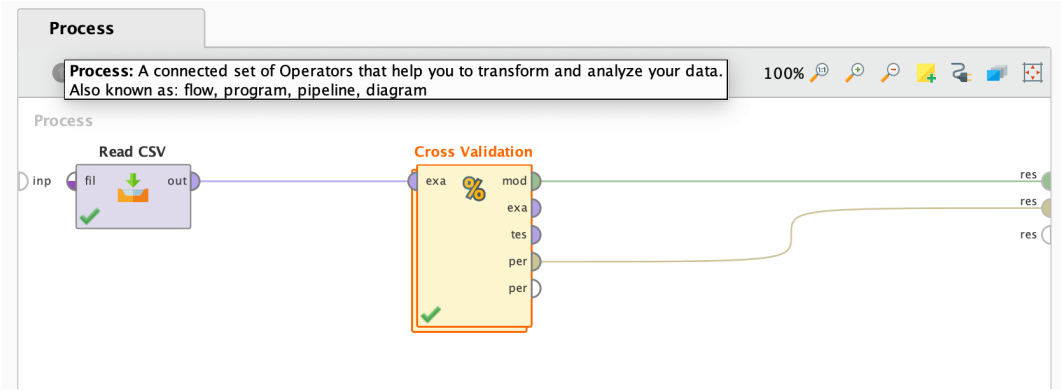

Figura 1: RapidMiner: Bagging - diseño principal.

Configure la validación cruzada conforme la figura 2. Para configurarla debe hacer doble click en el componente `Cross Validation`.

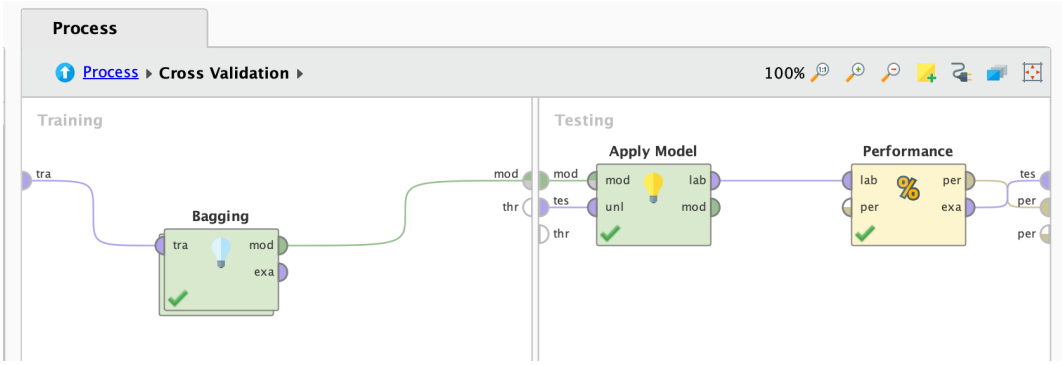

Figura 2: RapidMiner: Bagging - configuración de la validación cruzada.

Finalmente, configure el método bagging conforme la figura 3. Para configurarla debe hacer doble click en el componente `Bagging`.

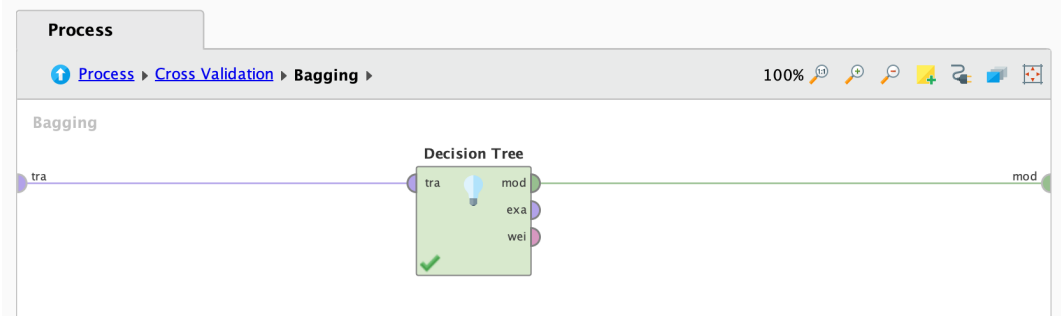

Figura 3: RapidMiner: Bagging - configuración de bagging.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)
- ¿El modelo mejora al usar bagging?
- Pruebe el algoritmo bagging usando otros algoritmos como por ejemplo `Naive Bayes`, `Support Vector Machine`, entre otros. ¿El modelo mejora al usar bagging?

2.3. Aplicando el algoritmo RandomForest (10 minutos)

Usando el mismo conjunto de datos `diabetes.csv` se le pide que realice la clasificación pero ahora usando el algoritmo `RandomForest`. Para esto realice los siguientes pasos:

- Utilice la misma configuración que en el ejercicio anterior.
- En lugar del componente `Bagging` use el componente `Random Forest`.
- Entrene el modelo usando validación cruzada con folds de 10.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)
- ¿En qué se diferencia del algoritmo `Bagging`?

2.4. Aplicando método ensemble Boosting (20 minutos)

Usando el mismo conjunto de datos `diabetes.csv` se le pide que realice la clasificación pero ahora usando el clasificador boosting. Para esto realice los siguientes pasos: configure el diseño principal conforme la figura 4. Recuerde que al leer el CVS debe indicar el atributo meta.

Configure la validación cruzada conforme la figura 5. Para configurarla debe hacer doble click en el componente `Cross Validation`.

Finalmente, configure el método boosting conforme la figura 6. Para configurarla debe hacer doble click en el componente `AdaBoost`.

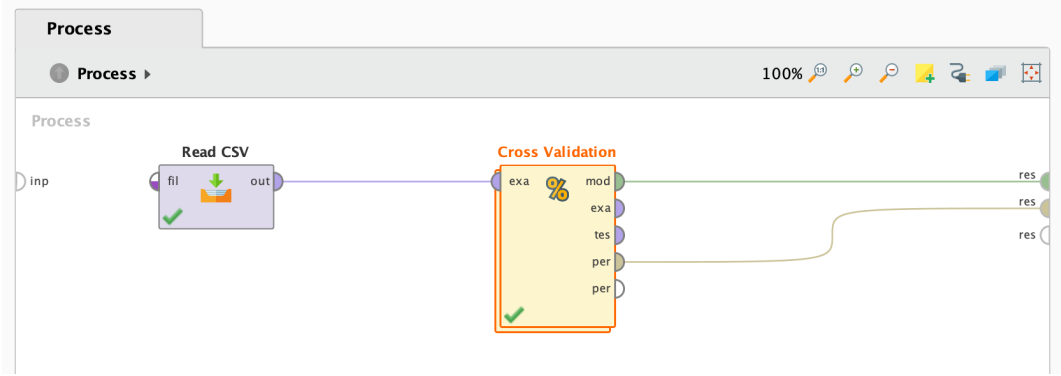

Figura 4: RapidMiner: Boosting - diseño principal.

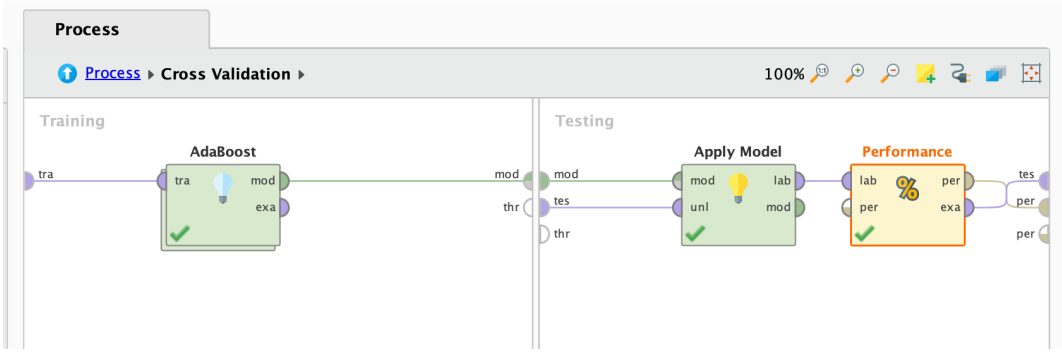

Figura 5: RapidMiner: Boosting - configuración de la validación cruzada.

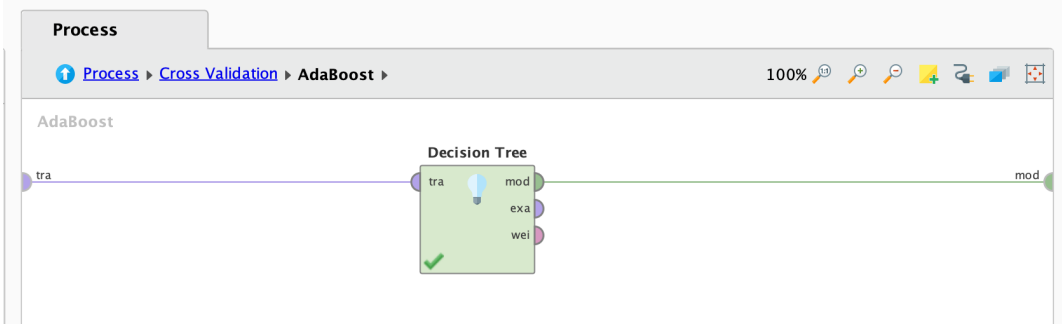

Figura 6: RapidMiner: Boosting - configuración de AdaBoost.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)
- ¿El modelo mejora al usar boosting?
- Pruebe el algoritmo boosting usando otros algoritmos como por ejemplo `Naive Bayes`, `Support Vector Machine`, entre otros. ¿El modelo mejora al usar boosting?
- ¿Qué diferencia encuentra entre bagging y boosting? En la práctica, ¿cuál es mejor?

2.5. Aplicando método ensemble Stacking (20 minutos)

Usando el mismo conjunto de datos `diabetes.csv` se le pide que realice la clasificación pero ahora usando el clasificador stacking. Para esto realice los siguientes pasos: configure el diseño principal conforme

la figura 7. Recuerde que al leer el CVS debe indicar el atributo meta.

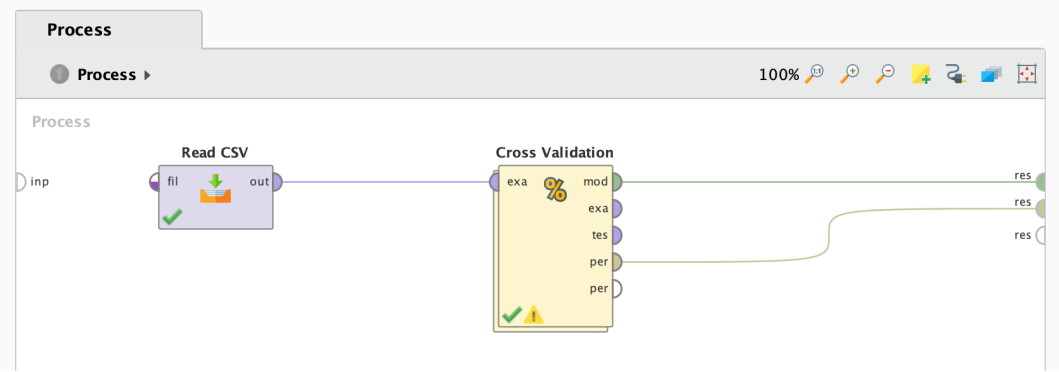

Figura 7: RapidMiner: Stacking - diseño principal.

Configure la validación cruzada conforme la figura 8. Para configurarla debe hacer doble click en el componente `Cross Validation`.

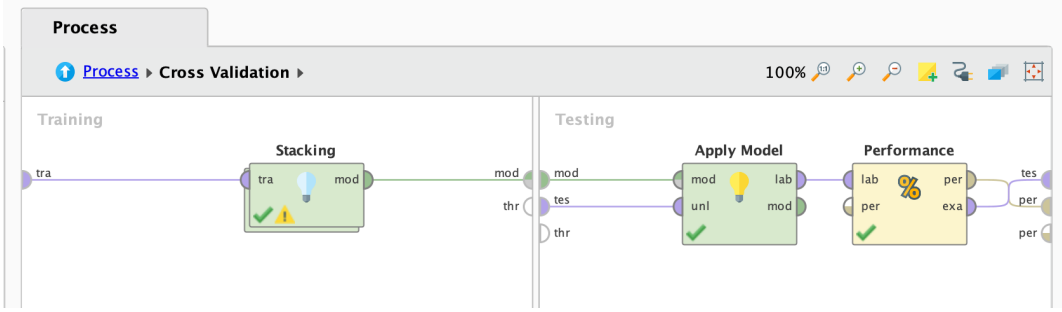

Figura 8: RapidMiner: Stacking - configuración de la validación cruzada.

Finalmente, configure el método stacking conforme la figura 9. Para configurarla debe hacer doble click en el componente `Stacking`.

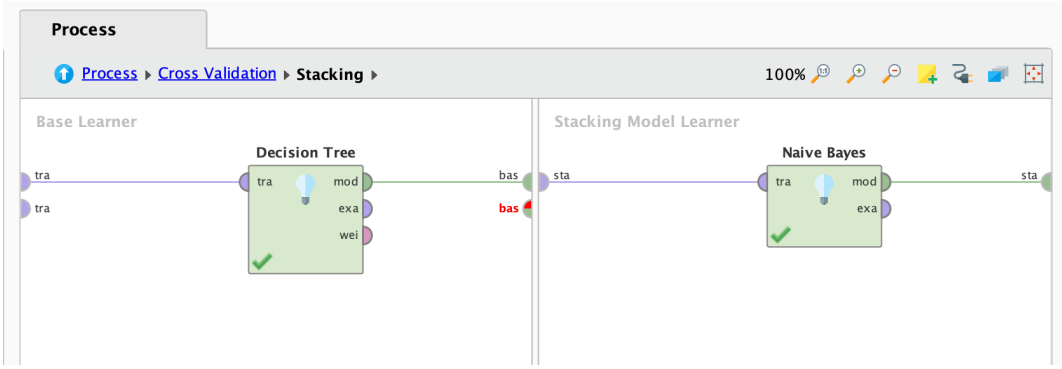

Figura 9: RapidMiner: Stacking - configuración de Stacking.

Preguntas de discusión

- Describa al modelo. ¿Es un buen modelo? Justifique su respuesta.
- Anote las principales métricas de este modelo (i.e., precisión, recall, número F_1)
- ¿Qué diferencia encuentra entre bagging, boosting y stacking?

3. Actividad en Python

3.1. Métodos ensemble en Python

Para ejecutar los métodos ensemble en Python se deben importar los algoritmos de la siguiente manera:

```
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import VotingClassifier
```

Use estos algoritmos para clasificar el conjunto de datos glass usando Python.

Monterrico, 4 de junio de 2019.