UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS
CIENCIAS DE LA COMPUTACIÓN

MACHINE LEARNING
Laboratorio 1
(Primer Semestre del 2019)

Objetivos de aprendizaje:

- Generar modelos algorítmicos y evaluarlos.

---

1. Actividad en Weka

1.1. Opciones de Prueba

Weka dispone de varias opciones para realizar pruebas que permiten medir el rendimiento de los modelos algorítmicos. Entre ellos se tienen:

Use training set Con esta opción, se prueba el modelo usando las mismas instancias que se usaron para el entrenamiento.

Supplied test set Con esta opción, se prueba el modelo usando un conjunto de datos adicional el cual debe ser suministrado por el usuario.

Cross-validation Con esta opción, se prueba el modelo usando la técnica de la validación cruzada. El usuario debe informar el número de *folds*. Por defecto, Weka presenta un valor de *folds* igual a 10.

Percentage split Con esta opción, se retiene un porcentaje de las instancias para realizar el entrenamiento del modelo. El resto del las instancias se utilizan para las pruebas.

Para escoger la opción de prueba que se desea utilizar, basta seleccionarla del panel `Test options`, la cual se encuentra en la parte superior izquierda en la pestaña `Classify`. Este panel de opciones se puede visualizar en la figura 1.

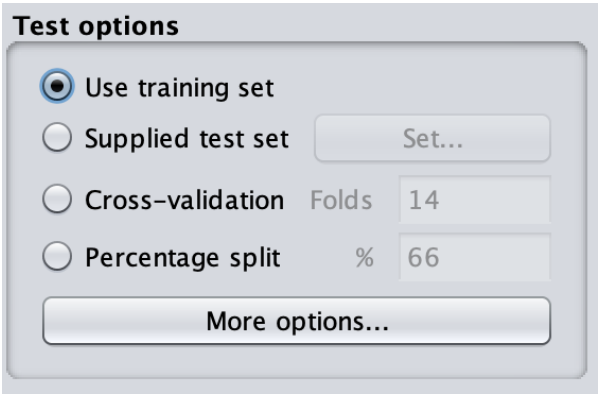

Figura 1: Weka: Opciones de Prueba.

A continuación se procederá a probar cada una de estas opciones. Se utilizarán los algoritmos `OneR`, `Naïve Bayes` y `J48`.

Use training set (10 minutos) Usando los conjuntos de datos `iris`, `weather` y `diabetes`, entrene los algoritmos `OneR`, `Naïve Bayes` y `J48` usando la opción de prueba `Use training set` y luego anote la métrica F_1 en la tabla 1.

Tabla 1: Casos de prueba para la opción Use training set.

Algoritmo	**Conjunto de Datos**	F_1
OneR	iris.arff	
Naïve Bayes	iris.arff	
J48	iris.arff	
OneR	weather.nominal.arff	
Naïve Bayes	weather.nominal.arff	
J48	weather.nominal.arff	
OneR	diabetes.arff	
Naïve Bayes	diabetes.arff	
J48	diabetes.arff	

Preguntas de discusión

- ¿Qué puede decir sobre los algoritmos en relación a la métrica usada?
- ¿Se puede afirmar que uno es mejor que otro? ¿Por qué?

Cross-validation (10 minutos) Usando los conjuntos de datos `iris`, `weather` y `diabetes`, entrene los algoritmos `OneR`, `Naïve Bayes` y `J48` usando la opción de prueba `Cross-validation` y luego anote la métrica F_1 en la tabla 2 usando diferentes valores para el *fold*. Use los valores de *fold* que se indican en la tabla 2 en la variable k.

Tabla 2: Casos de prueba para la opción Cross-validation.

Algoritmo	**Conjunto de Datos**	F_1 **k=2**	F_1 **k=5**	F_1 **k=7**	F_1 **k=10**	F_1 **k=12**
OneR	iris.arff					
Naïve Bayes	iris.arff					
J48	iris.arff					
OneR	weather.nominal.arff					
Naïve Bayes	weather.nominal.arff					
J48	weather.nominal.arff					
OneR	diabetes.arff					
Naïve Bayes	diabetes.arff					
J48	diabetes.arff					

Preguntas de discusión

- ¿Qué pasa cuando el número del *fold* aumenta?
- ¿Qué diferencia encuentra entre la métrica F_1 usando la opción de prueba `Use training set` y `Cross-validation`? ¿Por qué existe esta diferencia?

Percentage split (10 minutos) Usando los conjuntos de datos `iris`, `weather` y `diabetes`, entrene los algoritmos `OneR`, `Naïve Bayes` y `J48` usando la opción de prueba `Percentage split` y luego anote la métrica F_1 en la tabla 3 usando diferentes valores para el porcentaje. Use los valores de porcentaje que se indican en la tabla 3.

Tabla 3: Casos de prueba para la opción Percentage split.

Algoritmo	Conjunto de Datos	F_1 %=10	F_1 %=25	F_1 %=50	F_1 %=66	F_1 %=85
OneR	iris.arff					
Naïve Bayes	iris.arff					
J48	iris.arff					
OneR	weather.nominal.arff					
Naïve Bayes	weather.nominal.arff					
J48	weather.nominal.arff					
OneR	diabetes.arff					
Naïve Bayes	diabetes.arff					
J48	diabetes.arff					

Preguntas de discusión

- ¿Qué pasa cuando el porcentaje de división aumenta?
- ¿Qué diferencia encuentra entre la métrica F_1 usando la opción de prueba `Use training set`, `Cross-validation` y `Percentage split`? ¿Por qué existe esta diferencia?
- ¿Existe alguna relación entre el valor del F_1 y la cantidad de instancias del conjunto de datos?

Supplied test set (20 minutos) Se desea encontrar el modelo más óptimo para predecir el conjunto de datos denominado `Adult`. La descripción completa de este conjunto de datos la puede encontrar en la URL `https://archive.ics.uci.edu/ml/datasets/Adult`. Se requiere además que use la opción `Supplied test set`. Deberá usar como datos de entrenamiento el archivo `adult.arff` y como datos de prueba el archivo `adult_test.arff`. Ambos archivos se encuentran en el aula virtual.

Preguntas de discusión

- ¿Qué algoritmo consigue el mejor modelo? ¿Cómo justifica su elección?
- ¿Qué atributos seleccionó para entrenar el modelo? ¿Cómo justifica su elección?

2. Actividad en RapidMiner

En las sesiones previas se ha visto como ejecutar modelos algorítmicos entrenados y probados con el mismo conjunto de datos. Ahora se verá como se ejecutan algoritmos en `RapidMiner` que usan datos de prueba y entrenamiento diferentes y la técnica de cross-validation.

Usando datos de prueba (10 minutos) Se desea encontrar el modelo más óptimo para predecir el conjunto de datos denominado `Adult`. La descripción completa de este conjunto de datos la puede encontrar en la URL `https://archive.ics.uci.edu/ml/datasets/Adult`. Deberá usar como datos de entrenamiento el archivo `adult.csv` y como datos de prueba el archivo `adult_test.csv`. Ambos archivos se encuentran en el aula virtual.

Utilice tanto el algoritmo `DecisionTree` como el algoritmo `Naïve Bayes`. Para conseguir esto deberá definir un diseño similar al que se muestra en la figura 2.

Considere que:

- El componente `ReadCSV` deberá leer el archivo `adult.csv`. Recuerde que debe indicar la columna que es la etiqueta (`label`), es decir la clase predictora.
- El componente `ReadCSV (2)` deberá leer el archivo `adult_test.csv`. Recuerde que debe indicar la columna que es la etiqueta (`label`), es decir la clase predictora.

- El componente `Performance` corresponde al componente textttPerformance (Classification).
- El componente `Performance (2)` corresponde al componente textttPerformance.
- El componente `Select Attributes` podrá ser usado para seleccionar los atributos del conjunto de datos.

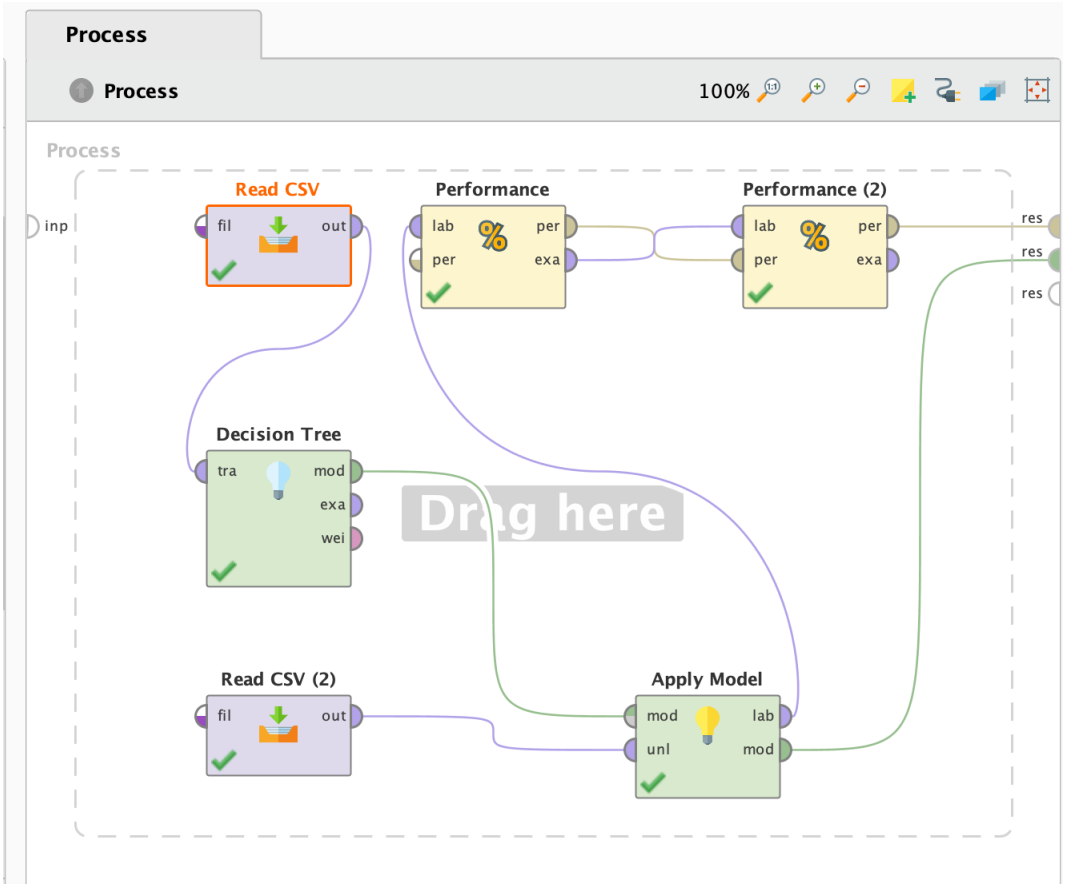

Figura 2: RapidMiner: Diseño con datos de prueba.

Preguntas de discusión

- ¿Qué algoritmo consigue el mejor modelo? ¿Cómo justifica su elección?
- ¿Los datos son consistente con los algoritmos ejecutados en Weka? Compare los valores de la precisión.

Validación cruzada (20 minutos) Se desea encontrar el modelo más óptimo para predecir la clase del conjunto de datos denominado `Car Evaluation` usando la técnica de la validación cruzada. La descripción completa de este conjunto de datos la puede encontrar en la URL `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`. El conjunto de datos se encuentra en el aula virtual.

Utilice tanto el algoritmo `ID3`, `DecisionTree` como el algoritmo `Naïve Bayes`. Para conseguir esto deberá definir un diseño similar al que se muestra en la figura 3 y 4.

En la figura 3 se puede apreciar la configuración de primer nivel de la validación cruzada. El componente `Cross Validation` posee como parámetro a `number of folds` en donde podrá configurar el número de `folds` que requiere.

Para detallar la configuración de la validación cruzada, debe hacer doble click en el componente `Cross Validation`. Aparecerá una pantalla la cual debe configurar conforme la figura 4. El componente `Performance` corresponde a `Performance (Classification)`.

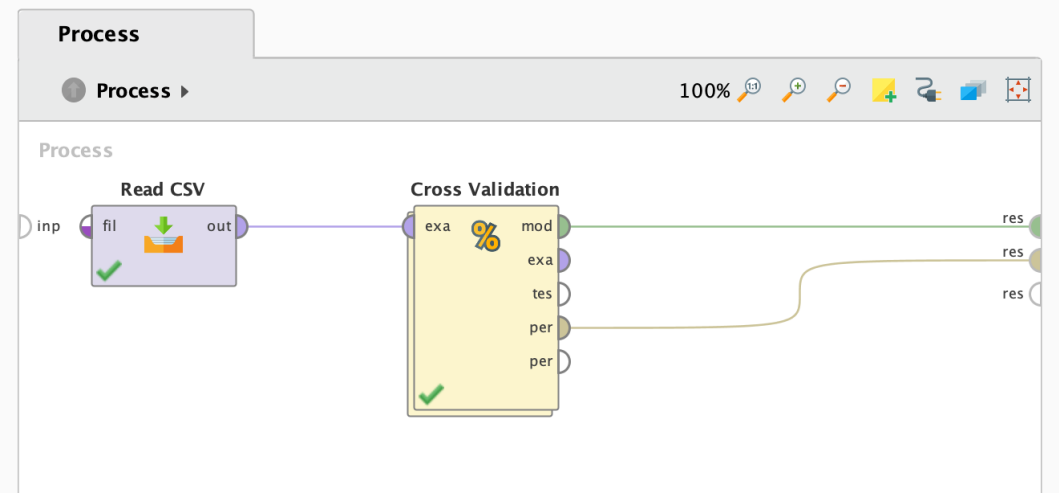

Figura 3: RapidMiner: Validación cruzada - paso 1.

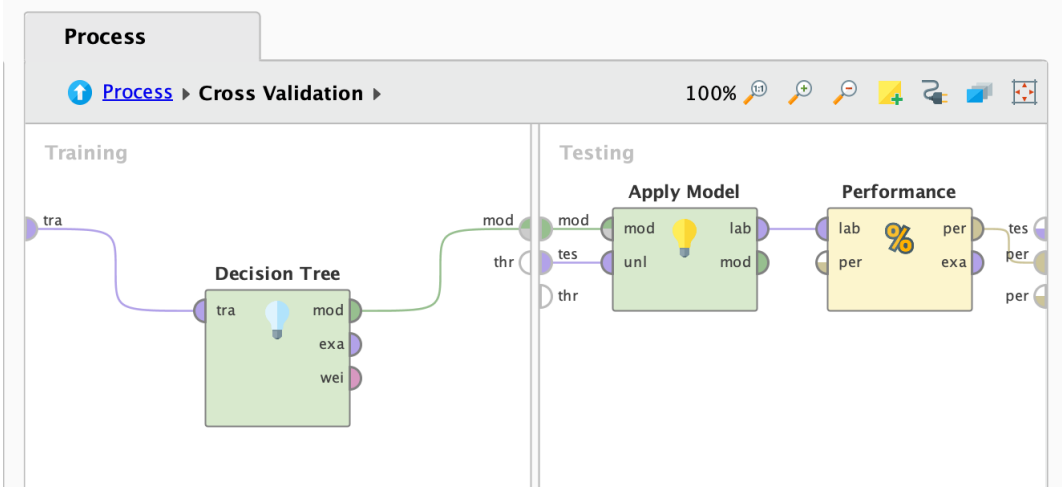

Figura 4: RapidMiner: Validación cruzada - paso 2.

Preguntas de discusión

- ¿Qué algoritmo consigue el mejor modelo? ¿Cómo justifica su elección?
- ¿Fue necesario seleccionar atributos? En caso afirmativo, ¿en qué algoritmo fue necesario?

3. Actividad en Python

En las sesiones previas se ha visto como ejecutar modelos algorítmicos entrenados y probados con el mismo conjunto de datos. Ahora se verá como se ejecutan algoritmos en `Python` que usan datos de prueba y entrenamiento diferentes y la técnica de cross-validation.

Usando datos de prueba (10 minutos) Se desea encontrar el modelo más óptimo para predecir el conjunto de datos denominado `iris.data` usando parte de los datos para entrenamiento y otra parte para las pruebas. Utilice tanto el algoritmo `GaussianNB` como el algoritmo `DecisionTreeClassifier`.

A continuación se presenta el script que puede ser usado para el algoritmo `GaussianNB`.

```
1 import pandas
2 from sklearn.naive_bayes import GaussianNB
3 from sklearn.model_selection import train_test_split
4 from sklearn.metrics import accuracy_score
5 from sklearn.metrics import classification_report
```

```
6
7  archivo="iris.data"
8  columnas=['longitud-sepalo', 'ancho-sepalo', 'longitud-petalo', 'ancho-petalo', 'clase']
9  conjunto_de_datos = pandas.read_csv(archivo, names=columnas)
10
11 X = conjunto_de_datos.iloc[:,0:4].values
12 y = conjunto_de_datos.iloc[:,4].values
13
14 X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size
       =0.66)
15
16 gnb = GaussianNB()
17 modelo = gnb.fit(X_entrenamiento, y_entrenamiento)
18 y_predecido = modelo.predict(X_prueba)
19
20 print(accuracy_score(y_prueba, y_predecido))
21 print(classification_report(y_prueba, y_predecido))
```

A continuación se presenta el script que puede ser usado para el algoritmo `DecisionTreeClassifier`.

```
1  import pandas
2  from sklearn.tree import DecisionTreeClassifier
3  from sklearn.model_selection import train_test_split
4  from sklearn.metrics import accuracy_score
5  from sklearn.metrics import classification_report
6
7  archivo="iris.data"
8  columnas=['longitud-sepalo', 'ancho-sepalo', 'longitud-petalo', 'ancho-petalo', 'clase']
9  conjunto_de_datos = pandas.read_csv(archivo, names=columnas)
10
11 X = conjunto_de_datos.iloc[:,0:4].values
12 y = conjunto_de_datos.iloc[:,4].values
13
14 X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(X, y, test_size
       =0.66)
15
16 arbol = DecisionTreeClassifier(criterion="entropy")
17 modelo = arbol.fit(X_entrenamiento, y_entrenamiento)
18 y_predecido = modelo.predict(X_prueba)
19
20 print(accuracy_score(y_prueba, y_predecido))
21 print(classification_report(y_prueba, y_predecido))
```

Preguntas de discusión

- Pruebe diferentes valores para `test_size`.
- ¿Los datos son consistente con los algoritmos ejecutados en Weka y RapidMiner? Compare los valores de la F_1.

Validación cruzada (15 minutos) Se desea encontrar el modelo más óptimo para predecir el conjunto de datos denominado `iris.data` usando la técnica de la validación cruzada. Utilice tanto el algoritmo `GaussianNB` como el algoritmo `DecisionTreeClassifier`.

A continuación se presenta un script que presenta el uso de la validación cruzada usando el algoritmo `DecisionTreeClassifier`.

```
1  import pandas
2  from sklearn.tree import DecisionTreeClassifier
3  from sklearn.model_selection import KFold
4  from sklearn.model_selection import cross_val_score
5  from sklearn.metrics import accuracy_score
6  from sklearn.metrics import classification_report
7
```

```
archivo="iris.data"
columnas=['longitud-sepalo', 'ancho-sepalo', 'longitud-petalo', 'ancho-petalo', 'clase']
conjunto_de_datos = pandas.read_csv(archivo, names=columnas)

X = conjunto_de_datos.iloc[:,0:4].values
y = conjunto_de_datos.iloc[:,4].values

modelo = DecisionTreeClassifier(criterion="entropy")
kfold=KFold(n_splits=10)

resultado=cross_val_score(modelo, X, y, cv=kfold, scoring="accuracy")
print("accuracy: ", resultado.mean())
resultado=cross_val_score(modelo, X, y, cv=kfold, scoring="f1_weighted")
print("F1: ", resultado.mean())
```

Preguntas de discusión

- ¿Qué algoritmo consigue el mejor modelo? ¿Cómo justifica su elección?
- ¿Los datos son consistente con los algoritmos ejecutados en Weka y RapidMiner? Compare los valores de la F_1.

Validación cruzada (15 minutos) Se desea encontrar el modelo más óptimo para predecir el conjunto de datos denominado `Fertility`. La descripción completa de este conjunto de datos la puede encontrar en la URL `https://archive.ics.uci.edu/ml/datasets/Fertility`. Deberá usar como datos de entrenamiento el archivo `fertility_Diagnosis.csv` usando la técnica de la validación cruzada. El archivo se encuentra en el aula virtual.

Preguntas de discusión

- ¿Qué algoritmo consigue el mejor modelo? ¿Cómo justifica su elección?
- ¿Qué atributos seleccionó para entrenar el modelo? ¿Cómo justifica su elección?

Pando, 25 de abril de 2019.