UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Estudios Superiores
Acatlán

Practica 4:

K folds

Aprendizaje de maquina y minería de datos avanzados

Integrantes:

Gustavo Adolfo Alvarez Hernández

Profesor:

Eduardo Eloy Loza Pacheco

FES Acatlán, 13 de febrero 2023

Objetivo

Que el alumno utilice métodos de evaluación de modelos como K folds para entender la importancia de dividir los datos para entrenamiento, validación y test. Así como aprenda a usar estos métodos para asegurar la generalidad de su modelo y evitar el sobre ajuste en sus modelos.

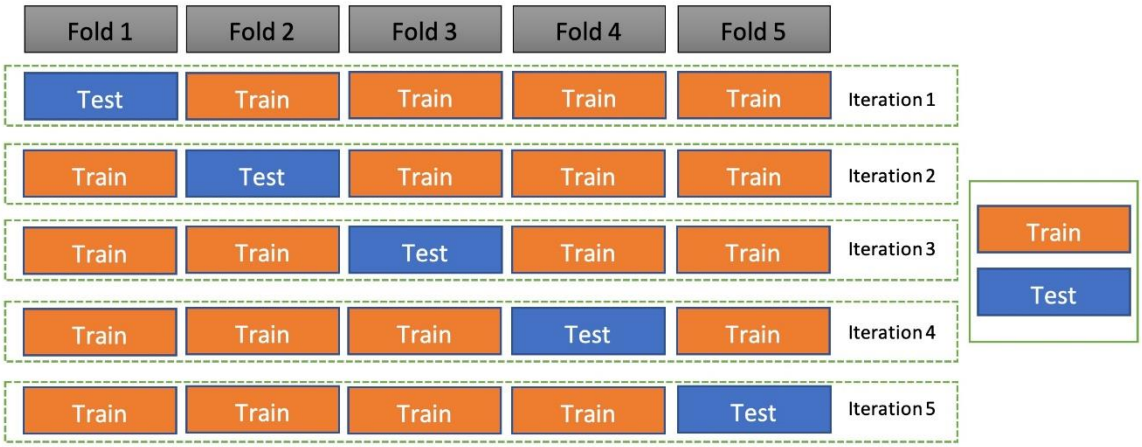
Resumen

En la presente practica se utilizarán datos sobre individuos que compraron o no un producto, dichos datos contienen el sexo de la persona y su nivel de ingresos. Se aplicarán 3 modelos de clasificación ya utilizados en otras prácticas como lo son regresión logística, SVM y arboles de clasificación, con el fin de hacer cross validation sobre los modelos y ver una métrica más general.

Antecedentes

Cross Validation K-fold, es un método de evaluación que se suele usar para tener métricas de desempeño del modelo que se haya implementado, la gran variante radica en que mide el error en diferentes iteraciones sobre los datos, siendo ahora el promedio de los resultados de la perdida nuestra nueva métrica. Se sigue el siguiente algoritmo:

1. Se divide los datos totales en dos (entrenamiento y test) en k diferentes formas.
2. Se toma una de las k formas de separar los datos y se entrena con sus respectivos datos de entrenamiento.
3. Se mide su rendimiento con sus respectivos datos de test y se guarda el valor de la métrica obtenida.
4. Se realiza los pasos anteriores para cada uno de los $k-1$ acomodos.
5. Se promedian las métricas obtenidas en las k interacciones y se toma como la métrica de validación final.



Así se tiene una métrica más general, puesto que, al probar con distintas configuraciones de los datos, el algoritmo ya utilizo toda la información disponible y en cada interacción podemos tener mejor o peor desempeño, pero esto nos ayuda a ver que en determinados casos con una sola validación pudimos quedarnos con el peor o el mejor desempeño y diríamos que ese sería el desempeño final.

Desarrollo

1. Importar los módulos necesarios

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
```

2. Cargamos los datos de redes sociales y determinamos el comprar o no como nuestra variable a predecir.

```
data = pd.read_csv("AnunciosRedesSociales.csv")
```

```
X= data.iloc[:,[2,3]].values
y= data.iloc[:,4].values
```

3. Dividimos los datos en entrenamiento y prueba para posteriormente escalarlos, esto con el fin de que los algoritmos que usan distancia tengan un mejor desempeño.

```
x_train,x_test,y_train,y_test= train_test_split(X,y,test_size=.25, random_state=0)
```

```
sx= StandardScaler()
x_train = sx.fit_transform(x_train)
x_test= sx.fit_transform(x_test)
```

4. Procedemos a instanciar la maquinada de soporte vectorial de clasificación

```
classificador= SVC(kernel="poly",degree=5,random_state=0)
classificador.fit(x_train, y_train)
```

```
SVC
SVC(degree=5, kernel='poly', random_state=0)
```

5. Realizamos una predicción con test y calculamos su matriz de confusión.

```
y_predice= classificador.predict(x_test)
```

```
confusion_matrix(y_test,y_predice)
```

```
array([[65,  3],
       [13, 19]], dtype=int64)
```

6. Y aplicamos cross validation con 10 folds sobre el modelo.

```
precisionSVC= cross_val_score(estimator=classificador, X=x_train, y= y_train, cv=10)
```

7. Se realiza un procedimiento de modelado similar al enlistado ahora para una regresión logística.

Logistica

```
regLog= LogisticRegression()
regLog.fit(x_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
y_log= regLog.predict(x_test)
```

```
confusion_matrix(y_test,y_log)
```

```
array([[63,  5],
       [ 8, 24]], dtype=int64)
```

```
precisionLog= cross_val_score(estimator=regLog, X=x_train, y= y_train, cv=10)
```

8. Y su símil para el árbol de clasificación

Arboles

```
tree= DecisionTreeClassifier()
tree.fit(x_train,y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
y_tree= tree.predict(x_test)
```

```
confusion_matrix(y_test,y_tree)
```

```
array([[61,  7],
       [ 5, 27]], dtype=int64)
```

```
precisionTree= cross_val_score(estimator=tree, X=x_train, y= y_train, cv=10)
```

9. Ahora ya tenemos los resultados de los tres algoritmos, sacamos promedio y desviación estándar para saber cual es en nuestro caso el acurracy y que tanto varía de iteración en interacción

```
SVC mean: 0.8033333333333333, SVC std: 0.06574360974438676
Logistic regression mean: 0.8233333333333335, Logistic regression std: 0.09666666666666669
Tree mean: 0.8566666666666667, Tree std: 0.051747248987533426
```

Conclusiones

Al realizar k folds sobre nuestros 3 algoritmos, generamos en esencia 10 modelos para cada tipo de algoritmo, medimos su desempeño y vemos en general o en promedio que tan bueno es. Este proceso de generalización nos sirve para tener modelos más robustos.

En nuestro caso el árbol de clasificación fue el que acertó en 85.6% de las veces y con un mas menos 5% de variación entre cada uno de los modelados, entonces es nuestro mejor modelo al tener mejor promedio y la menor de las variaciones entre experimento y experimento. El peor respectivamente sería la máquina de soporte vectorial.

Por último, este modelo también es usado para encontrar hiper parámetros para los modelos sin generar sobreajuste, puesto que al probarse sobre varias porciones de datos, no es posible que los parámetros se sobre ajusten a cada una de las porciones.

Referencias

Nellihela, P. (2022 de Jun de 2022). *Towards Data Science*. Obtenido de What is K-fold Cross Validation?: https://towardsdatascience.com/what-is-k-fold-cross-validation-5a7bb241d82f

Zhou, Z.-H. (2016). *Machine Learning.* Singapore: Springer Nature Singapore.