03/05/23 – **Cross-validation (K-folds) & Optimización paramétrica**

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | **Optimización de hiper-parámetros | Hyperparameter Optimization**  Familiarizados con el concepto de Cross Validation vamos a utilizar este mismo principio de fondo para lograr automatizar un poco la selección y optimización de nuestros modelos.  **Problema**: Parece que encontramos un modelo de aprendizaje que parece funcionar, pero esto puede implicar que ahora tenemos que ***encontrar la optimización de cada uno de los parámetros de este modelo***, encontrar el que mejor se ajuste y el que mejor resultado nos dé.   * Es fácil perderse entre los conceptos de tantos parámetros. Tenemos flexibilidad para algoritmos básicos de Machine Learning, pero fácil perderse. * Es difícil medir la sensibilidad de estos manualmente. * Es COSTOSO, en tiempo humano y computacionalmente.   Scikit-Learn nos ofrece enfoques para automatizar el proceso de optimización paramétrica. Existen 3 enfoques principales, estos son:   1. Optimización manual. 2. Optimización por grilla de parámetros | GridSearchCV. 3. Optimización por búsqueda aleatorizada.   **Optimización manual**   1. Escoger el modelo que queremos ajustar. 2. Buscar en la documentación de Scikit-Learn 3. Identificar parámetros y ajustes. Parámetros que vamos a necesitar y cuáles son los posibles ajustes que vamos a requerir para cada uno de estos parámetros. 4. Probar combinaciones una por una iterando a través de listas. |
| Sumario: El **score** es un arreglo de errores negativos medios cuadrados (es decir, cuanto más pequeño en valor absoluto, mejor se ajusta el modelo a los datos) como salida del **cross\_val\_score**, este resultado se da ya que el modelo fue separado **CV** veces (en este caso 5 al principio y luego 3) en set de datos de entrenamiento y prueba, en lo que se puede notar que particiones fueron más satisfactorias.  Ahora al aplicar el promedio y el valor absoluto, puedes observar el error medio cuadrado promedio calculado a partir de las salidas score que evalúan la adaptación promedio del modelo a los datos. | |

03/05/23 - **Optimizacion por grilla de parámetros | GridSearchCV**

|  |  |
| --- | --- |
| Puntos clave | Notas |
| Siempre testear, probar nuevas cosas y más con esta forma.  Todos los modelos son malos, solo que se busca el mejor. | **Optimizacion por grilla de parámetros | GridSearchCV**  Es una forma organizada, exhaustiva y sistematica de probar todos los parametros que le digamos que tenga que probar, con los espectivos rangos de valores que le aportemos.  Definir una o varias métricas que queremos optimizar.  Identificar los posibles valores que pueden tener los parámetros.  Crear un diccionario de parámetros.  Usar Cross Validation.  Entrenar el modelo  La grilla de parámetros nos define GRUPOS DE PARÁMETROS que serán probados en todas sus combinaciones (Un grupo a la vez)  svm-gridsearch-optimized  **Optimizacion por búsqueda aleatorizada | RandomizedSearchCV**  Si no tenemos tanto tiempo para una prueba tan exhaustiva o queremos combinaciones aleatorias usaremos este método. Es lo mismo que el caso anterior, pero busca de forma aleatoria los parámetros y Scikit-Learn selecciona los mejores de las combinaciones aleatorias que se hicieron.  En este método, definimos escalas de valores para cada uno de los parámetros seleccionados, el sistema probará varias iteraciones (Configurables según los recursos) y mostrará la mejor combinación encontrada. Ejemplo:  svm-randomized-search-optimized |
| Sumario: | |

03/05/23 - **GridSearchCV Vs. RandomizedSearchCV**

|  |  |
| --- | --- |
| Key Points | Notes |
|  | **GridSearchCV Vs. RandomizedSearchCV**  **GridSearchCV**   1. Cuando se quiera realizar un estudio a fondo sobre las implicaciones de los parámetros. 2. Se tenga tiempo. 3. Se tenga poder de procesamiento.   **RandomizedSearchCV**   1. Cuando se quiera explorar posibles optimizaciones. 2. Haya poco tiempo. 3. Haya poco poder de procesamiento.   GridSearch-vs-RandomizedSearch  **Automated Machine Learning (AutoML)**:es un concepto relativamente nuevo que en general pretende la completa automatización de todo el proceso de Machine Learning, desde la extracción de los datos hasta su publicación final de cara a los usuarios.  Sin embargo, este ideal aún está en desarrollo en la mayoría de las etapas del proceso de Machine Learning y aún se depende bastante de la intervención humana. Aún con esto, es importante que seamos conscientes de que ya existen varias herramientas que nos acercan un poco a esta meta casi tomada de la ciencia ficción.  **auto-sklearn:** nos ayudará a llevar aún un paso más lejos nuestro proceso de selección y optimización de modelos de machine learning. Dado que automáticamente prueba diferentes modelos predefinidos y configuraciones de parámetros comunes hasta encontrar la que más se ajuste según los datos que le pasemos como entrada. Con esta herramienta podrás entrenar modelos tanto de clasificación como de regresión por igual. |
| Sumario: | |

03/06/23 - **Auto-sklearn**

|  |  |
| --- | --- |
| Key Points | Notes |
|  | **Auto-sklearn**: Esta herramienta es una librería basada en los algoritmos de scikit-learn, aunque hay que tener presente que es una librería externa y se debe instalar siempre por aparte. En todo caso al ser una librería de Python se puede combinar sin ningún problema con el resto de nuestro código desarrollado para scikit-learn, incluso permitiendo la exportación de modelos ya entrenados para su posterior uso.   1. Se requiere un sistema operativo basado en Linux. 2. Python (>=3.5) . 3. Compilador para C++ (con soporte para C++11), por ejemplo GCC. 4. SWIG (versión 3.0 o superior).   La forma de hacer funcionar nuestro algoritmo no podría ser más fácil. Nos resultará bastante familiar a estas alturas después de haber trabajado tanto con sklearn.   1. import autosklearn.classification 2. cls = autosklearn.classification.AutoSklearnClassifier() 3. cls.fit(X\_train, y\_train) 4. predictions = cls.predict(X\_test) |
| Summary: | |

1/4/23

|  |  |
| --- | --- |
| Key Points | Notes |
|  |  |
| Summary: | |

1/4/23

|  |  |
| --- | --- |
| Key Points | Notes |
|  |  |
| Summary: | |

1/4/23

|  |  |
| --- | --- |
| Key Points | Notes |
|  |  |
| Summary: | |

1/4/23

|  |  |
| --- | --- |
| Key Points | Notes |
|  |  |
| Summary: | |

1/4/23

|  |  |
| --- | --- |
| Key Points | Notes |
|  |  |
| Summary: | |

1/4/23

|  |  |
| --- | --- |
| Key Points | Notes |
|  |  |
| Summary: | |