Aprendizaje Automático con Scikit-learn

Taller de Data Science - Intermedio

Contenido

1. Aprendizaje automático con Scikit-learn
    - Algoritmos de clasificación: regresión logística, árboles de decisión, SVM.
    - Algoritmos de regresión: regresión lineal, regresión polinómica.
    - Algoritmos de agrupamiento: k-means, clustering jerárquico.
2. Evaluación de modelos
    - Métricas de evaluación de modelos de clasificación: precisión, recall, F1-score.
    - Métricas de evaluación de modelos de regresión: error cuadrático medio, coeficiente de determinación.
3. Validación cruzada y técnicas de particionamiento

Aprendizaje automático con Scikit-learn

Scikit-learn: Una biblioteca de aprendizaje automático en Python

- Scikit-learn es una biblioteca de aprendizaje automático en Python.
- Proporciona una amplia gama de algoritmos y herramientas.
- Permite realizar tareas de aprendizaje automático de manera sencilla y eficiente.
- Es una de las bibliotecas más utilizadas y populares en la comunidad de Python.

Datasets de ejemplo

Dataframes prácticos para usar modelos de regresión y de clasificación

https://scikit-learn.org/stable/datasets/toy_dataset.html

Algoritmos de clasificación en Scikit-learn

- Regresión logística: Clasificación basada en la estimación de probabilidades.
- Árboles de decisión: Modelo en forma de estructura de árbol para clasificación.
- Máquinas de vectores de soporte (SVM): Clasificación basada en hiperplanos de separación.

https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html

https://ml-cheatsheet.readthedocs.io/en/latest/classification_algos.html

https://www.kaggle.com/getting-started/161325

Algoritmos de regresión en Scikit-learn

- Regresión lineal: Modelo lineal para predecir valores continuos.
- Regresión polinómica: Modelo no lineal que considera relaciones polinómicas.

https://ml-cheatsheet.readthedocs.io/en/latest/regression_algos.html

Algoritmos de agrupamiento en Scikit-learn

- K-means: Algoritmo de agrupamiento basado en la asignación a k clústeres.
- Clustering jerárquico: Construcción de clústeres en forma de jerarquía.

https://towardsdatascience.com/clustering-cheat-sheet-dcf72259abb6

Scikit-learn en resumen

- Biblioteca de aprendizaje automático en Python.
- Amplia variedad de algoritmos y herramientas.
- Facilita la implementación de tareas de aprendizaje automático.
- Popular y ampliamente utilizada en la comunidad de Python.

Evaluación de modelos

Métricas para modelos de clasificación

Las métricas de evaluación de modelos de clasificación más comunes son **precisión**, **recall** y **F1-score**.

Matriz de confusión		Estimado por el modelo	
		Negativo (N)	*Positivo (P)*
Real	*Negativo*	a: (TN)	b: (FP)
	Positivo	c: (FN)	d: (TP)

Precisión ***("precision")*** Porcentaje predicciones positivas correctas:	d/(b+d)	
Sensibilidad, exhaustividad ***("Recall")*** Porcentaje casos positivos detectados	**Especifidad** ***(Specifity)*** Porcentaje casos negativos detectados	**Exactitud ("accuracy")** Porcentaje de predicciones correctas *(No sirve en datasets poco equilibrados)*
d/(d+c)	a/(a+b)	(a+d)/(a+b+c+d)

Imagen tomada de https://empresas.blogthinkbig.com/ml-a-tu-alcance-matriz-confusion/

Precisión

Es la proporción de instancias positivas correctamente clasificadas sobre el total de instancias clasificadas como positivas. Se calcula como:

precisión = verdaderos positivos / (verdaderos positivos + falsos positivos)

La precisión mide la exactitud del modelo al predecir las instancias positivas. Es útil cuando se desea minimizar los falsos positivos.

Recall

También conocido como sensibilidad o tasa de verdaderos positivos, es la proporción de instancias positivas correctamente clasificadas sobre el total de instancias positivas en los datos reales. Se calcula como:

recall = verdaderos positivos / (verdaderos positivos + falsos negativos)

El recall mide la capacidad del modelo para encontrar todas las instancias positivas. Es útil cuando se desea minimizar los falsos negativos.

F1-score

Es la media armónica de precisión y recall. Combina estas dos métricas en una sola medida que proporciona una visión equilibrada del rendimiento del modelo. Se calcula como:

F1-score = 2 * (precisión * recall) / (precisión + recall)

El F1-score es útil cuando se desea tener en cuenta tanto la precisión como el recall de manera equilibrada.

Importancia de las métricas

Dependiendo del problema y las necesidades específicas, se puede dar mayor importancia a la precisión, al recall o al equilibrio entre ambos, lo cual se reflejará en la elección de la métrica principal a considerar en la evaluación.

Es importante tener en cuenta que estas métricas son aplicables a problemas de clasificación binaria, pero pueden ser extendidas a problemas de clasificación multiclase mediante la utilización de técnicas como la macro o micro promediación.

Métricas para modelos de regresión

Error cuadrático medio (MSE), es una medida que calcula el promedio de los errores al cuadrado entre las predicciones del modelo y los valores reales de la variable objetivo. Es una métrica comúnmente utilizada para evaluar la precisión de un modelo de regresión.

El MSE se expresa en las unidades cuadradas de la variable objetivo. Un valor de MSE más bajo indica que las predicciones del modelo se ajustan mejor a los datos reales.

El coeficiente de determinación, también conocido como R^2, es una medida que proporciona información sobre qué proporción de la variabilidad de la variable objetivo puede ser explicada por el modelo. R^2 varía entre 0 y 1, y se interpreta de la siguiente manera:

Un valor de R^2 más alto indica que el modelo es capaz de explicar una mayor proporción de la variabilidad de la variable objetivo.

Error cuadrático medio (MSE)

Se calcula de la siguiente manera:

$MSE = (1/n) * \Sigma(y - y\_pred)^2$

Donde:

- n es el número de muestras en el conjunto de datos.
- y son los valores reales de la variable objetivo.
- y_pred son las predicciones del modelo.

El MSE se expresa en las unidades cuadradas de la variable objetivo. Un valor de MSE más bajo indica que las predicciones del modelo se ajustan mejor a los datos reales.

Coeficiente de determinación (R^2)

R^2 varía entre 0 y 1, y se interpreta de la siguiente manera:

- $R^2 = 1$: El modelo explica perfectamente la variabilidad de la variable objetivo.
- $R^2 = 0$: El modelo no explica la variabilidad de la variable objetivo, es decir, no es mejor que predecir simplemente el valor medio de la variable objetivo.
- $R^2 < 0$: El modelo es peor que predecir el valor medio de la variable objetivo.

El coeficiente de determinación se calcula de la siguiente manera:

$R^2 = 1 - (SSR / SST)$

Donde:

- SSR (Sum of Squares of Residuals) representa la suma de los residuos al cuadrado.
- SST (Total Sum of Squares) representa la suma total de las desviaciones al cuadrado de los valores reales con respecto a su media.

Un valor de R^2 más alto indica que el modelo es capaz de explicar una mayor proporción de la variabilidad de la variable objetivo.

Validación cruzada y técnicas de particionamiento

Validación cruzada (Cross-validation)

La validación cruzada es una técnica que se utiliza para estimar el rendimiento de un modelo en datos no vistos. Consiste en dividir el conjunto de datos en múltiples subconjuntos llamados "folds" o "pliegues". El modelo se entrena y evalúa repetidamente, utilizando diferentes combinaciones de folds como conjunto de entrenamiento y prueba. La validación cruzada proporciona una estimación más robusta del rendimiento del modelo al promediar los resultados de las diferentes iteraciones.

Un ejemplo común de validación cruzada es la validación cruzada k-fold. En este enfoque, el conjunto de datos se divide en k pliegues. El modelo se entrena k veces, cada vez utilizando k-1 pliegues como conjunto de entrenamiento y el pliegue restante como conjunto de prueba. Los resultados se promedian para obtener una medida de rendimiento global.

Técnicas de particionamiento de datos

métodos para dividir el conjunto de datos en conjuntos de entrenamiento y prueba de manera determinística. Estas técnicas incluyen:

- **División en entrenamiento/prueba**: El conjunto de datos se divide en dos partes, una para entrenar el modelo y la otra para evaluar su rendimiento.
- **Validación cruzada estratificada**: Similar a la validación cruzada k-fold, pero se asegura que cada fold contenga una proporción equilibrada de muestras de cada clase. Es útil cuando el conjunto de datos está desequilibrado.
- **División temporal:** Se utiliza cuando los datos tienen una dimensión temporal, donde se divide el conjunto de datos en función del tiempo. Los datos más antiguos se utilizan para entrenar el modelo y los datos más recientes se utilizan para evaluar su rendimiento.
- **División por grupos**: Cuando los datos contienen grupos o clústeres, se puede dividir el conjunto de datos de manera que los grupos se asignen exclusivamente al conjunto de entrenamiento o prueba. Esto evita que se introduzca sesgo debido a la similitud de las muestras en el mismo grupo.