Ciencia de Datos

Práctico N°6: Regresión lineal, SVMs, ROC, Precision-Recall y κ

Problema 1: Leer el artículo Common pitfalls in the interpretation of coefficients of linear models de scikit-learn, prestando especial atención al análisis de la base de datos, a los `pipelines` construidos y a la interpretación de los coeficientes.
a) Importar los datos de la base "Current Population Survey" citada en el artículo. Construir los conjuntos train y test y configurar el pipeline adecuado para regresión lineal, considerando las columnas numéricas y categóricas.
b) Implementar los modelos `LinearRegression` y `HuberRegressor`. Graficar valores predichos vs. valores reales sobre el conjunto test, y evaluar el error absoluto medio en cada uno.
c) En el artículo se menciona que las columnas AGE y EXPERIENCE están muy correlacionadas y que por lo tanto causan gran variabilidad en los coeficientes del ajuste lineal. Repita el procedimiento anterior con `HuberRegressor` removiendo alguna de estas dos columnas y compare los resultados. ¿Altera esto al error obtenido?

Problema 2: Estudiar las implementaciones de Support Vector Machines (SVMs) provistas por scikit-learn.
a) Estudiar las diferencias entre los modelos Support Vector Classification (SVC), Linear Support Vector Classification (LinearSVC), Nu-Support Vector Classification y Linear classifiers with SGD (SGDClassifier). ¿Cuales son los kernels disponibles? Destacar los pros y contras de cada modelo.
b) Para la clasificación multiclase identificar cuales modelos implementan el esquema `one-vs-one` y/o el esquema `one-vs-rest`.
c) Estudiar el significado de los parámetros `C`, `nu`, `gamma`, `coef0`, `degree` y `class_weight` y averiguar cuándo se aplican.

Problema 3: Usar como guía la entrada de scikit-learn sobre SVMs para estudiar las fronteras de decisión generadas por diferentes máquinas de vectores soporte, utilizando como *toy-model* el iris dataset. Para trabajar en el plano 2D, emplear sólo las dimensiones de sépalo de las flores.

Problema 4: Usar como guía la entrada de se scikit-learn sobre RBF SVM para estudiar los parámetros `C` y `gamma`, implementando una búsqueda sobre grid para optimizarlos y visualizar los resultados aplicando el modelo sobre el iris dataset.

Problema 5:

Recuerde que se usa la curva Precision-Recall para problemas desbalanceados o cuando la clase positiva es prioritaria, seleccionando el umbral que maximice el F1-Score o cumpla con requisitos de precisión/recall, y se usa la curva ROC para problemas más balanceados o cuando el FPR es relevante, seleccionando el umbral que maximice el índice de Youden o minimice la distancia al punto ideal.

Siempre considera el contexto del problema y los costos asociados a los errores para tomar una decisión informada.

Varios clasificadores proveen como salida un *score* entre 0 y 1 para cada ejemplo, salida que puede interpretarse como una probabilidad y es una medida para generar un clasificador binario que asigna etiquetas en base a un umbral (usualmente 0.5). Si la salida del clasificador está por encima del umbral, se etiqueta con p, caso contrario, etiqueta con n. La tabla adjunta muestra 20 ejemplos sintéticos con clase binaria C = {p, n} y el correspondiente *score* asignado por un clasificador hipotético.

	C	Score		C	Score
1	p	0.90	11	p	0.40
2	p	0.80	12	n	0.39
3	n	0.70	13	p	0.38
4	p	0.60	14	n	0.37
5	p	0.55	15	n	0.36
6	p	0.54	16	n	0.35
7	n	0.53	17	p	0.34
8	n	0.52	18	n	0.33
9	p	0.51	19	p	0.30
10	n	0.505	20	n	0.10

a) Construir *a mano* la curva ROC y la curva Precision Recall para este ejemplo. Compare el umbral escogido con la mejor *accuracy* con el umbral que sugiere el mayor F1-score=2*(Precision-Recall)/(Precision+Recall), y el mayor índice de Youden $J = TPR - FPR$.
b) Usando RocCurveDisplay.from_estimator de scikit-learn, graficar la curva ROC de SVC y la de Logistic regression aplicado al Breast cancer dataset.
c) Usando PrecisionRecallDisplay.from_estimator de scikit-learn, graficar la curva Precision-Recall de SVC y la de Logistic regression aplicado al Breast cancer dataset.

Problema 6: Un problema usual en el etiquetado de los ejemplos de una base de datos consiste en medir la concordancia entre dos anotadores. En clasificación binaria, la forma más sencilla es calcular la fracción de ejemplos igualmente clasificados. Sin embargo, esta medida no tiene en cuenta las coincidencias por mero azar. Para contemplar esta posibilidad es que se introduce el Coeficiente Kappa de Cohen según,

$$\kappa = \frac{p_o - p_e}{1 - p_e},$$

donde p_o es el acuerdo relativo entre los dos clasificadores y p_e es la probabilidad de acuerdo hipotético por azar, bajo el supuesto que los clasificadores son independientes. A modo de ejemplo se muestra la matriz de confusión de dos anotadores para 50 ejemplos.

	Yes	No
Yes	20	5
No	10	15

Claramente $p_o = (20 + 15)/50 = 0{,}7$. Por otro lado, uno de los anotadores asigna Yes con probabilidad $(20 + 5)/50 = 0{,}5$ y No con probabilidad $(10 + 15)/50 = 0{,}5$; mientras que el otro asigna Yes con $(20 + 10)/50 = 0{,}6$ y No con probabilidad $(5 + 15)/50 = 0{,}4$.
Bajo la hipótesis de independencia: $p_e = 0{,}5 \times 0{,}6 + 0{,}5 \times 0{,}4 = 0{,}5$ y resulta $\kappa = 0{,}4$.
a) Calcular κ en el siguiente ejemplo:

	Yes	No
Yes	25	35
No	5	35

b) Estudiar la implementación de la función κ de scikit-learn y evaluarla en el problema anterior.

FaMAF 2024