Ciencia de Datos

Práctico N°4: Clasificación Binaria

Matriz de Confusión

En los problemas de predicción con dos clases, cobran relevancia los siguientes conceptos:

- TP: Positivos verdaderos
- FP: Falsos positivos
- TN: Negativos verdaderos
- FN: Falsos negativos

donde a una de las clases se la considera el *target* para la clasificación y sus valores se los denomina positivos y en contraposición al resto negativos. Cuando al ejemplo de una clase se lo ha clasificado de forma correcta, a la clasificación de ese ejemplo se la toma como verdadera y como falsa en caso contrario. De esta forma, un ejemplo positivo mal clasificado se lo llama falso negativo, mientras que a uno negativo mal clasificado se lo denomina falso positivo.

La manera usual de presentar el número de ejemplos en cada una de estas categorías es mediante la matriz de confusión:

		clasificación	
		$+$	$-$
clase	$+$	$\#TP$	$\#FN$
	$-$	$\#FP$	$\#TN$

Estudiar el matreial de la clase VIII sobre métricas de evaluación, prestando especial atención a las métricas derivadas de la misma para clasificación binaria, en particular:

- Accuracy: $\dfrac{\#TP + \#TN}{\#TP + \#TN + \#FP + \#FN}$ $(= \hat{p})$
- Recall: $\dfrac{\#TP}{\#TP + \#FN}$ (= Sensitivity)
- Precision: $\dfrac{\#TP}{\#TP + \#FP}$
- Specificity: $\dfrac{\#TN}{\#TN + \#FN}$ (= Selectivity)
- F1 score: $\dfrac{2\,\#TP}{2\,\#TP + \#FP + \#FN}$

Notar que estas métricas son sólo útiles para la clasificación binaria, mientras que la matriz de confusión puede construirse para cualquier problema multiclase.

Problema 1: Loan dataset

a) Examinar el dataset Loan Data disponible en Kaggle. Indagar el diccionario de las columnas del dataset y en qué consiste el problema de predicción que puede implementarse con este dataset.

b) Estudiar la forma de disponer los datos directamente en memoria, usando el disco virtual de la instancia de Jupiter Lab (colab) que corre la `.ipynb` en la que se trabaja.

c) Explorar los datos, calcular el desbalance de clases y analizar los valores del único atributo no-numérico.
d) Implementar la función `pandas.get_dummies` de Pandas para convertir el atributo de texto en un conjunto de variables binarias.
e) Separar los datos en un conjunto de entrenamiento, reservando el 33 % para testing.

Problema 2: Clasificador Bayesiano

a) Calcular la media y la matriz de covarianza correspondiente a los datos de entrenamiento de cada una de las clases, para fijar los parámetros del clasificador bayesiano construído en el práctico anterior.
b) Predecir la clase de cada ejemplo del conjunto de test y evaluar la clasificación usando las funciones: `accuracy_score`, `recall_score`, `precision_score` y `confusion_matrix` de la librería `sklearn.metrics` y analizar los resultados.
c) Ignorar las correlaciones presentes entre las variables, reteniendo sólo los elementos diagonales (varianzas) en las matrices de covarianza calculadas en el ítem (a).
d) Repetir la predicción sobre el conjunto de test, usando el clasificador bayesiano sin correlaciones. Evaluar el resultado repitiendo el ítem (b).

Problema 3: Naïve Bayes

a) Estudiar el tutorial de `datacamp` para aprender a implementar un clasificador Naïve Bayes usando Scikit-learn.
b) Usar este clasificador para predecir la clase del Loan dataset y evaluar con las mismas métricas usadas en el ejercicio anterior.
c) Comparar las tres evaluaciones obtenidas y discutir la conveniencia de cada clasificador.

FaMAF 2024