Universidad De Granada

Apredizaje Automático

Proyecto Final

Pedro Bonilla Nadal
Antonio Martín Ruiz

11 de junio de 2020

Índice

1. Compensión del problema a resolver

Este dataset contiene datos de la oficina del censo[1] relativos a censos de 1995, obtenidos del repositorio UCI de bases de datos [2]. Un total de 48842 personas han sido encuestadas para este censo. De estas personas tenemos una serie de variables con información de carácter socioeconómico. En particular:

- tenemos 6 variables de tipo numérico u ordinal, con valores en rangos distintos.
- 8 variables de tipo categórico.
- Una variable de clase que toma como valores < 50K y > 50k.
- Es un problema de clasificación, ya que no tenemos información para hacer una regresión sobre la ganancia.

El código utilizado para la resolución de la práctica se encuentra en el archivo `main.py`. Del mismo modo, los datos limpiados se encuentran en el archivo `adults.data` y su información relativa procesada en el archivo `adult.names`.

2. División y codificación de los datos

A la hora de obetener los datos en el respositorio se encuentran tres archivos: `adult.data`, `adult.test` y `adult.names`. Estos datos fueron procesados en el año 1996. Nosotros realizamos una operación de formato de los datos para ajustarlos a convenciones más recientes.

- Cambiamos la variable de clase por valores categóricos 0 para < 50K y 1 para > 50K
- Eliminamos el espaciado después de la coma para facilitar la lectura por parte de librerías actuales.
- Añadimos la información del archivo `adult.test` al archivo `adult.data` con el objetivo de tener la información procesada de manera conjunta.
- Añadimos el flag `@attribute` a la lista de atributos provista en `adult.names`.

Una vez leidos los datos haremos una division train/test de 80 %/20 %. Elegimos esta proporción aprovechando que dada la gran cantidad de datos de los que poseemos. Obtenemos por lo tanto un conjunto de train con 39074 instancias, así como un conjuto de test con 9768 instancias.

Además, a la hora de validar la selección de hiperparámetros utilizaremos la técnica cross validation, para la cual diviremos el conjunto de training en 5 subconjuntos. Se utilizará para esta la función `cross_validate`[4].

Como ultimo detalle de codificación de los datos, durante la parte de preprocesado realizamos la técnica de dummy variables. Esta técnica permite entender como variables categóricas

3. Preprocesado

Mostraremos a continuación las siguientes técnicas de preprocesado realizadas. Justificaremos individualmente su uso.

- Normalización de las variables: realizaremos una normalización de los datos, para evitar que su escala afecte a la relevancia de estos. Para ello usaremos la función `StandardScaler`[3] provista en la librería `sklearn`.

- Aumento de dimensionalidad: debido a la baja cantidad de variables, en contraposicición con la alta cantidad de ejemplos podemos considerar aumentos de la dimensionalidad de los datos sin miedo a que se genere un gran sobreajuste.

  - Dummy Variables: una variable dummy es aquella que toma sólo el valor 0 o 1 para indicar la ausencia o la presencia de algún efecto categórico que se pueda esperar que cambie el resultado. Esta técnica ayudará a codificar la entrada para algunos algoritmos, así como poder hacer un estudio de cada categoría como una variable separada a la hora de hacer una valoración del interés de cada variable.

    Incluir inofrmación de como se ha realizado.

  - Polynomial Freatures: con especial interés en el caso lineal, el uso de variaciónes polinómicas de los datos puede ser util para permitir aproximaciones a clases de funciones nuevas. Probaremos variación cuadrática sobre las variables numéricas cuando realizemos un ajuste lineal. Para Random Forest no lo consideramos necesario por la ya conocida complejidad de los arboles, y para SVM utilizaremos variaciones de kernell.

    Incluir información de como se ha realizado

- Valoración de las variables de interés: Para realizar un estudio de las variables de interés

- Datos incompletos y valores perdidos: en relación a los valores perdidos encontramos en tres variables de tipo categórico.

Variable	Valores distintos	Valores Perdidos
`workclass`	9	2799
`occupation`	15	2809
`native-country`	42	857

Tabla 1: Representación de valores perdidos.

Realizamos la sustitución de estos datos mediante la función `replace_lost_categorical_values`. En esta, para cada columna, calculamos su distribución de probabilidad, esto es, sumamos las ocurrencias de cada categoría y dividimos entre el total de valores con valores no perdidos. Obtenemos así la probabilidad de cada categoría. A continuación calculamos las probabilidades acumuladas sumando para categoría su probablidad y la de todas las anteriores. Para cada dato perdido generamos un número aleatorio mediante una distribución uniforme en el intervalo $[0, 1]$. La clase por la que el valor perdido será sustituida será la de cuyo intervalo contenga al valor generado, siendo el intervalo de cada clase el comprendido entre la probabilidad acumulada de la anterior y su probabilidad acumulada (incluyendo en cada uno su extremo inferior, pero no su extremo superior).

- Datos inconsistentes: no encontramos datos inconsistentes.

- Balanceo de clases: nos encontramos ante una situación con un desbalanceo notable. Sin embargo, dada la cantidad de datos provista por la base de datos creemos que no es necesario realizar modificaciones de los datos ni en el conjunto de train ni en el de test.

4. Métricas

Hemmmos decidido conseiderar, en este contexto de clasificación, dos funciones con objetivo de medir el error, ambas halladas en [6].

- *accuracy*: Decidimos utilizar esta medida por su simplicidad y expresividad. Esta métrica nos propocionará una idea general de la bondad de nuestro modelo en un caso general, así como nos permitirá comparar, por ejemplo, con el clasificador modal.
- *f1-score*: Al estar en un modelo de clases desbalanceadas, consideramos que debiamos utilizar una métrica que penara comportamientos de 'asignación modal'. La medida F_1 se define como[7]:

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

  Donde precision es (en terminos de clase positiva/negativa) el número de verdaderos positivos entre los positivos escogidos, y recall es el número de positivos escogidos entre los positivos totales.

5. Técnica de Ajuste para el Modelo Lineal

Selección de las técnica (parámetrica) y valoración de la idoneidad de la misma frente a otras alternativas

6. Hiperparámetros y selección del modelo

Aplicación de la técnica especificando claramente que algoritmos se usan en la estimación de los parámetros, los hiperparámetros y el error de generalización

7. Error de Generalización

8. Argumentar sobre la idoneidad de la función regularización usada

9. Conclusiónes

Valoración de los resultados y justificación (gráficas, métricas de error, análisis de residuos, etc)
que se ha obtenido la mejor de las posibles soluciones con la técnica elegida y la muestra dada. Argumentar en términos de los errores de ajuste y generalización.

Referencias

[1] Página oficial de la Oficina del censo: http://www.census.gov/ftp/pub/DES/www/welcome.html.

[2] Página del Centro para el Apredizaje Automático y Sistemas Inteligentes:http://archive.ics.uci.edu/ml/datasets/Adult.

[3] Función StandardScaler de la librería sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[4] Función cross_validate de la librería sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html

[5] Artículo donde explica el concepto de dummy variable: https://en.wikipedia.org/wiki/Dummy_variable_(statistics)

[6] Métricas de sklearn https://scikit-learn.org/stable/modules/model_evaluation.html#balanced-accuracy-score

[7] Métrica f1-score https://en.wikipedia.org/wiki/F1_score