Aprendizaje automático TC3020

Práctica 01

Entrega: 26/Febrero/2021 a más tardar a las 23:59 horas

Se deberá generar un reporte en formato **PDF** donde se muestren los detalles de la práctica en **typesetting** (es decir, formato nativamente digital, no escaneo). Subir el reporte a Canvas en el apartado adecuado. Se debe agregar el código fuente del programa generado. El código fuente debe estar estructurado de tal forma que pueda ser fácilmente ejecutado desde una terminal. Agregar un documento README.txt donde es específique la forma de emplear el programa. El código fuente deberá estar debidamente comentado.

Regresión logística

Dataset 1: Credit Card Default Data. Este conjunto de datos contiene 10,000 entradas donde el objeto es predecir cuáles clientes van a incumplir con la deuda de su tarjeta de crédito. Este dataset cuenta con el siguiente formato:

1. ID: número de la entrada (descartar).
2. default: un factor con niveles **No** y **Yes** indicando si el cliente va a incumplir su deuda.
3. student: un factor con niveles **No** y **Yes** indicando si el cliente es un estudiante.
4. balance: el saldo promedio que le queda al cliente en su tarjeta de crédito después de realizar su pago mensual.
5. income: ingreso del cliente.

Este dataset se encuentra disponible en el apartado de la Clase 2 bajo e nombre *Default.txt*. Para la práctica ocupar a *balance*, *income*, y *student* como las features del vector de entrada. Además, *default* será la categoría para el vector de entrada, es decir, serán las etiquetas.

Dataset 2: Identificación de genero.Este conjunto de datos contiene 10,000 entradas donde el objeto es predecir el genero de una persona, es decir, **Male** o **Female**. Este dataset cuenta con el siguiente formato:

1. Gender: genéro de la persona con dos niveles: **Male** o **Female**.
2. Height: altura de la persona.
3. Weight: peso de la persona.

Este dataset se encuentra disponible en el apartado de la Clase 2 bajo e nombre *genero.txt*. Ocupar a *Height* y *Weight* como los features del vector de entrada y a *Gender* como la categoría a ser predecida.

Instrucciones

1. Programar el método de regresión logística visto en clase, ocupando el algoritmo de entrenamiento basado en el gradiente descendente.
2. Ocupar como método de paro del método de gradiente descendente $||\vec{\beta}_{j+1} - \vec{\beta}_j||_2 <$ threshold. Se recomienda usar threshold $= 1 \times 10^{-4}$.
3. Preparar el dataset. Ocupar valores 0 y 1 para las categorías implicadas.
4. Ocupar el 80 % del dataset como training set y el restante 20 % será ocupado como test set. La asignación de las instancias al training set y el test set debe ser de forma aleatoria.

5. Una vez entrenado el modelo, aplicar el test set y medir la efectividad del entrenamiento. Para esto, ocupar la tasa de error: $\epsilon = \frac{1}{N}\sum_{i=1}^{N} I(y_i, \hat{y}_i)$, donde N es la cardinalidad del test set e $I(y, \hat{y}) = 1$ si $y \neq \hat{y}$ o $I(y, \hat{y}) = 0$ si $y = \hat{y}$.

6. El learning rate $\alpha \in (0, 1]$ es un valor que impacta mucho la efectividad del entrenamiento. Buscar el α que les otorgue el mejor valor posible de ϵ.

En el reporte, por cada dataset, presentar la siguiente información de acuerdo con cada intento realizado:

1. Mostrar el vector $\vec{beta}$.

2. Informar cuántas interaciones fueron necesarias para entrenar el modelo.

3. Informar el valor de α empleado.

4. Informar el valor de threshold empleado en el entrenamiento (en caso de usar y no usar 1×10^{-4}).

5. Informar el valor ϵ obtenido, es decir la taza de error que tiene el modelo.

Solo para el caso del segundo dataset, hacer la gráfica de la clasificación hecha por el modelo empleando el test set con el $\vec{\beta}$ asociado al mejor ϵ. Es decir, gráficar $\hat{y}_i$ vs $(\vec{x}), i = 1, 2, \ldots, M$, donde M es el tamaño del test set. Además, comparar esta gráfica contra los valores de categoría verdaderos, es decir, contra cada y_i. Recuerda que $\hat{y} = 0$ si $\hat{p} < 0.5$ y $\hat{y} = 1$ si $\hat{p} \geq 0.5$, donde $\hat{p} = h_{\vec{\beta}}$. Dado que es una gráfica en $\mathbb{R}^3$, te puedes apoyar de hacer proyecciones sobre los planos x_1 vs y y x_2 vs y.

Investigar cómo funciona el método *LogisticRegression* de SciKit-Learn (Python). Una vez investigado, ocúpalo para entrenar un modelo de regresión lineal. Compara los resultados de esta función contra lo que tú obtuviste con tu método. Reporta el valor de $\vec{\beta}$ obtenido con *LogisticRegression* y el valor ϵ. Recuerda ocupar el mismo training set y test set que empleaste con tu método al utilizar *LogisticRegression*. En el reporte, describe cómo ocupaste la funcion *LogisticRegression*, y todos los parámetros necesarios para hacerlo funcionar.

Entregar el código fuente de forma elegante. Se deberá mandar el programa desde una terminal, con los siguientes parámetros:

1. param1: nombre del archivo del dataset.

2. param2: porcentaje de elementos del training set (ocupar valores enteros).

3. param3: porcentaje de elementos del test set (ocupar valores enteros).

4. param4: valor de α.

5. param5: valor del threshold.

6. param6: semilla del generador de números pseudoaleatorios.

El programa deberá mostrar como salida los valores del vector $\vec{\beta}$, ϵ, el número de iteraciones que se ocuparon para el algoritmo de gradiente descendente. Finalmente, se deberá escribir en un archivo los vectores $\vec{x}_i$ junto con $\hat{y}_i$, y_i, y $\hat{p}_i$ para $i = 1, 2, \ldots, M$, donde M es el tamaño del test set. Agregar un README.txt donde se muestre cómo se ejecuta el programa y toda descripción necesaria. Entregar todo en un archivo ZIP.