UNIVERSIDAD TECNOLÓGICA DE CAMPECHE

INGENIERÍA EN DESARROLLO Y GESTIÓN DE SOFTWARE

ACTIVIDAD:

INVESTIGACIÓN

ASIGNATURA:

EXTRACION DE CONOCIMIENTO EN BASE DE DATOS

PRESENTADO POR:

VALENZUELA LAGUNAS ERICK

NOMBRE DEL PROFESOR:

MTRA. MARIA DE LOURDES CÁRDENAS MALDONADO

GRADO Y GRUPO

9 “B”

PERIODO

MAYO – AGOSTO 2023

JUSTIFICACIÓN DEL ALGORITMO ULTILIZADO

En este caso de estudio, donde se deberán de determinar la si un correo electrónico recibido vía email puede ser considerado como “spam” o no, se ven utilizados una serie de algoritmos que nos permitirán obtener una salida con tiempos y clasificaciones correspondientes a los datos ingresados.

Es por la flexibilidad de la premisa presentada en el caso a estudiar, que se están viendo algoritmos de aprendizaje supervisados de clasificación y de regresión de manera simultanea.

Dentro de estos se ven incluidos aquel de BBR, que servirá como nuestro interprete de datos predictivos por medio de regresión, SVM, que nos permite separar en clases los datos de entrada lo que resulta ideal para el cálculo de estos algoritmos en un entorno de clasificación, y por último se cuenta con el algoritmo de PLAUM que existe como una mejora del algoritmo SVM al compartir los mismos objetivos.

Con esto dicho, vale la pena mencionar que las pruebas de evaluación de estos pueden presentar información diferente a aquella determinada en la descripción de los modelos.

MODELO UTILIZADO

- BBR (Bayesian Binary Regression):

Se trata de una implementación de la regresión logística bayesiana, aplicada a la clasificación binaria. La clave de este algoritmo es la utilización de una distribución de probabilidad previa (ver ecuación 1) y algoritmos de optimización sucesiva de los ejemplos de entrenamiento suministrados.

$$p(y = 1 | \beta, x) = \psi(\beta^T x)$$

Ecuación 1

Este algoritmo inicialmente realiza una regresión logística de los datos de entrenamiento a partir de la distribución de probabilidad elegida (Gausiana o Laplace), por medio de una función de enlace ψ

$$\psi(z) = \exp(z)/(1+\exp(z))$$

Ecuación 2

Una vez obtenido el modelo de regresión, se va optimizando sucesivamente a través de la aplicación de un algoritmo de regresión logística en cadena. Se trata de un algoritmo de optimización de coordenada cíclica descendente. Se comienza poniendo todas las variables a algún valor inicial, y se busca qué valor de la primera variable minimiza la función objetivo, asumiendo que todas las otras variables mantienen constantes sus valores iniciales. Este es un problema de optimización unidimensional. El mismo método se lleva a cabo con la segunda variable, y así sucesivamente hasta que se han cruzado todas las variables. Este proceso se repite varias pasadas hasta encontrar un criterio de convergencia.

1. Inicializar las variables βj=0 (vector de parámetros), ∆j=0 (conjunto de datos, o región explorada), para j=1 hasta d (número de parámetros del modelo); τi =0 (estimador previo de confianza), para i=1 hasta n (número de ejemplos).

2. Desde k=1, 2, ... hasta que se produzca la convergencia, hacer:

2.1. Para j=1 hasta d, hacer:

2.1.1.Calcular

$$\Delta \upsilon_j = -\frac{\left(\sum_{i=1}^{n} x_{ij} y_i \frac{1}{1+\exp(\tau_i)}\right) + 2\beta_j / \tau_j^2}{\left(\sum_{i=1}^{n} x_{ij}^2 F(\tau_j; \Delta_j |x_{ij}|)\right) + 2/\tau_j}$$

2.1.2.Calcular: ∆β j = mín máx ∆υ j − ∆ j ∆ j , en los datos spam del subconjunto tratado.

2.1.3. Calcular:

∆ τ = ∆ βj xij yi , con

τi = τi + ∆τ , para i=1,...,n.

2.1.4. Caulcular: βj = βj + ∆βj

2.1.5. Calcular: ∆i = máx(2 |∆ βj| ∆i /2), ampliando el tamaño del subconjunto de spam tratado.

- SVN:

El objetivo perseguido por este algoritmo es encontrar el hiperplano óptimo que maximice la distancia entre los casos positivos y los casos negativos.

Estos son los pasos principales del algoritmo SVM:

1. Seleccionar el parámetro C como representante de la compensación entre la reducción al mínimo del error de clasificación del conjunto de entrenamiento y maximizar el margen.

2. Seleccionar la función de kernel y cualquier parámetro del mismo.

3. Solucionar el problema cuadrático dual resultante de la ecuación 3 o una formulación alternativa que use la programación cuadrática de forma apropiada o un algoritmo de programación lineal.

$$\min_{\alpha} \tfrac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{m} \alpha_i$$
$$s.t. \quad \sum_{i=1}^{l} y_i \alpha_i = 0$$
$$C \geq \alpha_i \geq 0 \quad i = 1,...,m$$

Ecuación 3

4. Extraer la variable de umbral b, utilizando los vectores de apoyo.

5. Clasificar un punto nuevo x, siguiendo la ecuación 4:

$$f(x) = sign(\sum_{i} y_i \alpha_i K(x, x_i) - b)$$

Ecuación 4

donde, xi (i=1,...,m) son los miembros del conjunto de puntos de entrenamiento; yi=±1 son las etiquetas de la clase; y αi son los vectores de soporte.

- PLAUM (Perceptron Algorithm with Uneven Margins):

Se basa en la idea de encontrar un margen entre hiperplanos, y sus autores aseguran que funciona mejor que SVM para tareas de clasificación de texto. Este algoritmo requiere:

- Un conjunto de entrenamiento linealmente separable de la forma: $z = (x, y) \in (Xx - \{1,+1\})$.
- Un índice de aprendizaje + $\eta \in \Re$.
- Un número máximo de iteraciones T.
- Dos parámetros que limitan los ejemplos negativos y positivos: $\tau_{-1}, \tau_{+1} \in \Re$

El algoritmo funciona de acuerdo a los siguientes pasos:

1. Inicialización de variables:

$\text{iteración} = 0; \text{i} = 1; \text{paso} = \text{m}; \vec{\text{w}} = \vec{0}; \text{b} = 0; \text{R} = \text{máx}_{\text{Z}_1 \in \text{X}} |\vec{\text{x}}_\text{i}||$

2. Repetir.

    2.1. Si $Y_1((\vec{W}, \vec{X}_I) + b) \leq \tau_1$ entonces

        2.1.1. $\vec{w} = \vec{w} + \eta y_i \vec{x}_i$.

        2.1.2. $b = b + \eta y_i R^2$.

        2.1.3. paso = i .

    2.2. i = i +1.

    2.3. Si (i > m) entonces

        2.3.1. i = 1.

        2.3.2. iteración = iteración +1. Hasta (i = paso) o (iteración ≥ T).

3. Devolver (w,b).

EVALUACIÓN Y OPTIMIZACIÓN DEL MODELO

Los experimentos se han realizado utilizando como colección de entrenamiento y prueba de correos electrónicos SPAMBASE.

Esta compuesta por 4601 casos, de los cuales 1813 (39.4%) son clasificados como “Spam”. Cada ejemplo está representado por un vector de 58 atributos,

donde el último contiene la clase a la que pertenece el documento (“Spam” o “NO Spam”). El resto de atributos representan la aparición de determinados símbolos o palabras. La colección SPAMBASE no es linealmente separable, y la selección de atributos claves, permite la separación, casi perfecta de las dos clases. La tabla 1 muestra los tiempos medios expresados en segundos, empleados en el entrenamiento con las particiones tratadas. Los resultados de prueba son inmediatos una vez entrenado el modelo.

	ENTRENAMIENTO
BBR	**5,1 seg.**
SVM	13396,37 seg.
PLAUM	10,61 seg.

Tabla 1

Como método de evaluación de los experimentos realizados hemos utilizado Validación Cruzada o “10-fold Cross Validation”, consistente en hacer diez particiones iguales de la colección y utilizar de forma alternativa nueve partes para entrenamiento y la parte restante para prueba. Este proceso se repite 10 veces variando la partición utilizada en la evaluación. El resultado final obtenido es una media de los diez resultados parciales.

Para cada una de las ejecuciones, hemos obtenido una tabla de contingencia, del estilo mostrado en la tabla 2, donde:

− A es el número de documentos Spam clasificados como Spam;

− B es el número de documentos No Spam, clasificados como Spam;

− C es el número de documentos Spam, clasificados como No Spam;

− D es el número de documentos No Spam, clasificados como No Spam.

Partición P_i	**SPAM**	**NO SPAM**
Asigna SPAM	A	B
Asigna NO SPAM	C	D

Tabla 2

A partir de estos datos se obtienen las métricas de precisión y recall y F medida, expresadas en las ecuaciones 5, 6 y 7:

$$P = \text{Pr}\,ecisión = \frac{A}{A+B}$$

Ecuación 5

$$R = \text{Re}\,call = \frac{A}{A+C}$$

Ecuación 6

$$F1 = \frac{2 * P * R}{P+R}$$

Ecuación 7

La tabla 3 muestra los resultados obtenidos en términos de precisión y recall. Se ha incluido también la métrica F1 con el fin de tener una valoración global del comportamiento de los algoritmos.

	Precisión	**Recall**	**F1**
BBR	87,10%	88,93%	87,99%
SVM	80,52%	40,68%	54,05%
PLAUM	58,10%	67,52%	62,46%

Tabla 3

CONCLUSIONES

Esta actividad marca el final a la tercera unidad correspondiente a la materia de ‘Extracción del conocimiento en bases de datos’ donde se estudian los algoritmos de análisis supervisado, y la práctica busca la creación de un reporte de un caso de estudio sobre el que se puedan aplicar los algoritmos estudiados en la unidad.

El objetivo detallado para está práctica se ve reflejado en el enfoque de la unidad misma como: “El alumno implementará algoritmos de análisis supervisado para aplicarlos en la predicción y clasificación de nuevas entradas de datos.”, este se ve cumplido de una manera bastante eficiente puesto a que analizar un caso de estudio permite aprender como es que se ven los conceptos en un contexto real.

El primer paso que nos llevó al desarrollo de esta actividad fue el de realizar una investigación para aprender cuales son las ocasiones en los que ven aplicados los algoritmos de aprendizaje. Conocer los casos utilizados con mayor frecuencia permitió centrar la búsqueda en un punto donde se pudieran ver utilizados múltiples algoritmos para comparar, evaluar y optimizar por medio de los métodos y nodos ofrecidos en la parte teórica de la unidad.