Proyecto: Minería de datos

Juan Pablo Echeagaray González, Emily Rebeca Méndez Cruz, Grace Aviance Silva Arostegui

***Resumen*—Referencia perrona de este libro [1]**

***Index Terms*—Data Science, Machine Learning, Data Analysis**

I. Introducción

II. Créditos

- Juan Pablo Echeagaray González - A00830646
- Emily Rebeca Méndez Cruz
- Grace Aviance Silva Aróstegui

III. Modelos de Machine Learning

III-A. Árbol de decisión

[2]

III-B. Support Vector Machine (SVM) Grace Aviance Silva Arostegui

Support vector machine (SVM) es un algoritmo de aprendizaje supervisado que se utiliza en muchos problemas de clasificación y regresión, e incluso para la detección de valores atípicos [1]. Este modelo de aprendizaje automático se basa en una separación de diferentes clases a través de un hiperplano en un espacio de dimensión superior [3].

Dado un conjunto de muestras (ejemplos de entrenamiento) se etiquetan clases y se entrena una SVM para construir un modelo que prediga la clase de una nueva muestra.

Para la base de datos que tenemos y el análisis que queremos llevar a cabo, en donde el enfoque es considerar en cada uno de los registros de las mujeres cual es el riesgo que tienen para desarrollar cáncer cervical. Dado que en la base de datos el rango de riesgos va de 0 a 4, tenemos así 5 clases de las cuales buscaríamos 5 hiperplanos que tengan el margen lo más ancho posible entre las clases, para así poder entregar lo mejor posible al algoritmo y hacer mejores clasificaciones futuras. Cabe mencionar que utilizamos un 80/20 de los datos para entrenamiento y prueba respectivamente.

III-C. Red Neuronal

Después de inspeccionar el mapa generado hemos notado que hay algunos puntos que parecen tener datos geográficos erróneos, descartar la entrega a estos clientes es algo inaceptable, así que una de las siguientes tareas en el proyecto será desarrollar un método de limpieza efectivo que ayude a mejorar la información geográfica que obtengamos de cada punto.

Juan Pablo Echeagaray González, Emily Rebeca Méndez Cruz, Grace Aviance Silva Arostegui pertencen al Tec de Monterrey campus Monterrey, N.L. C.P. 64849, Mexico

III-D. Regresión Logística

La regresión logística es otro de los métodos de *machine learning* usados comúnmente para la clasificación de datos. Este algoritmo se usa regularmente para estimar la posibilidad de que una instancia pertenezca a cierta clase; para el caso de clasificación binaria, si dicha probabilidad es mayor al 50 %, se clasifica a esa instancia como perteneciente a la clase positiva. Para generalizar este modelo a problemas de clasificación con n clases ($n > 2$), se calculan n probabilidades de que la instancia pertenezca a la clase i, al final se escoge la que tenga el valor más alto [1] [4].

Al igual que con una regresión lineal, este algoritmo calcula una suma ponderada de los atributos de la instancia, más un término libre; pero lo que el modelo regresa es el resultado de aplicar la función logística a ese número:

$$\hat{p} = \sigma(\mathbf{x^T \Theta}) \tag{1}$$

La función logística tiene un rango de 0 a 1 (por eso su uso como clasificador binario) a su vez se define como:

$$\sigma(t) = \frac{1}{1 + \exp(-t)} \tag{2}$$

Una vez que se cuenta con la estimación de la probabilidad, la salida del modelo queda definida como:

$$\hat{y} = \begin{cases} 0 & \hat{p} < 0{,}5 \\ 1 & \hat{p} \geq 0{,}5 \end{cases} \tag{3}$$

Para entrenar nuestro modelo hemos de encontrar el vector $\mathbf{\Theta} \in \mathbb{R}^{\mathbf{27}}$ (se tienen 27 atributos después de la limpieza de datos) que minimice la función *log loss*, dicha función tiene la siguiente forma:

$$J(\mathbf{\Theta}) = -\frac{\mathbf{1}}{\mathbf{m}} \sum_{\mathbf{i=1}}^{\mathbf{m}} \left[\mathbf{y^{(i)}} \log\left(\mathbf{\hat{p}^{(i)}}\right) + (\mathbf{1} - \mathbf{y^{(i)}}) \log\left(\mathbf{1} - \mathbf{\hat{p}^{(i)}}\right) \right] \tag{4}$$

El índice m de la función 4 representa el número de instancias que tenemos en la base de datos; lo que hace esta función es calcular el promedio del error producido con el vector $\mathbf{\Theta}$. Al final de la fase de entrenamiento se desea haber encontrado el vector que minimice esta función

III-D1. Generalizando a n clases: La regresión logística que clasifica instancias que podrían tener diferentes etiquetas es conocida como *Regresión Logística Múltiple* o *Regresión Softmax* (similar a la función descrita en la sección III-C). Primero se calcula una suma ponderada como en regresión logística para cada una de las posibles clases:

$$s_k(\mathbf{x}) = \mathbf{x^T \Theta^{(k)}} \tag{5}$$

Se usa este valor en la función sigmoide para obtener una probabilidad estimada:

$$\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_\mathbf{k} \tag{6}$$

Y al final se escoge la clase con la probabilidad más alta:

$$\hat{y} = \arg\max_i \sigma(s_k(\mathbf{x})) \tag{7}$$

IV. Resultados

V. Conclusiones

V-A. *Áreas de mejora*

V-B. *Modelo seleccionado*

VI. Reflexiones

Apéndice A
Datos

Los datos usados en este proyecto pueden descargarse aquí

Apéndice B
Código

El código desarrollado se encuentra en el siguiente repositorio

Apéndice C
Evidencias de trabajo en equipo

Referencias

[1] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Culemborg, Netherlands: Van Duuren Media, 2019.

[2] Sci-kit Learn, "sklearn.tree.DecisionTreeClassifier." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[3] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.

[4] Sci-kit Learn, "sklearn.linear$_model.LogisticRegression.''[Online].Available$: