Práctica 7: Clasificadores y reconocimiento de patrones

Adriana Basurto Vázquez A01323251
Patricia Cañete Leyva A01323992
Luis Alfredo León Villapún A01322275
Diego Osorio Sánchez A01328920
ITESM
Puebla, México

***Resumen*—** En este documento se presentan los resultados de la práctica número siete del curso de Visión para Robots, el cual consiste en utilizar un algoritmo de aprendizaje maquinal con el fin de evaluar un set de datos, y predecir nuevos valores a partir del entrenamiento dado al algoritmo.

Índice de términos: Machine Learning, Validación cruzada, Red de Bayes, Naive Bayes, Matriz de Confusión.

I. Introducción

En la era de los datos es muy importante tener la habilidad de adelantarse a los hechos y ser capaces de predecir, casi con total certeza, nuevas variables que puedan ocurrir en un futuro. Es por aquí donde Machine Learning toma importancia, ya que es una solución muy eficaz en una época donde la inmensa cantidad de datos hace imposible que los humanos sean capaces de procesar por sí mismos. En el presente documento, se presenta un caso de uso clásico en el aprendizaje maquinal, donde hay un set de datos con características de 3 plantas diferentes. Usando un algoritmo conocido como Naive Bayes, se entrenará para que logre predecir el tipo de planta dado un conjunto de datos. Finalmente, se muestra la matriz de confusión de este sistema.

II. Marco teórico

Machine Learning

Machine Learning, o aprendizaje automático, podría definirse como lo cita Samuel: "El área de estudio que provee a las computadoras la habilidad de aprender sin ser explícitamente programadas"(Samuel, 1959).

El aprendizaje maquinal tiene múltiples ramas, dentro de las cuales se encuentran dos grandes grupos: aprendizaje supervisado, y aprendizaje no supervisado. En el aprendizaje supervisado, se cuenta con un set de datos que entrena al algoritmo de aprendizaje. Por el otro lado, el aprendizaje no supervisado funciona al ser entregado un grupo de datos y encontrar patrones y relaciones a partir de estos.

Redes Bayesianas

Modelo gráfico, donde se describe la distribución conjunta de variables aleatorias. (Carballo, 2017). Esto puede ser fácilmente representado en un grafo, donde la probabilidad de un evento está condicionada por sus padres.

De aquí se puede inferir el teorema de Bayes, presentado a continuación:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Clasificador Naive Bayes

Un clasificador naive bayes es un modelo simple de una red bayesiana, donde se asume independencia condicional entre variables, por lo cual es más sencillo calcular la probabilidad de encontrar un evento.

Validación Cruzada

Un método de validación cruzada consiste en dividir el set de datos que se utilizará para un proceso de aprendizaje automático en dos: set de entrenamiento y set de prueba. A continuación se presentan diferentes métodos de validación cruzada:

Enfoque de set de validación: En este método, simplemente se divide el set de datos en 50 por ciento y 50 por ciento. Una gran desventaja es que se puede despreciar alguna información importante.

Leave-one-out: En este método, se itera a través del set de datos utilizando todos los elementos, excepto uno. Así, el algoritmo aprenderá, por supuesto quedándose con la iteración donde su margen de error es menor.

K-folds: Este método de validación cruzada divide el set de datos en k divisiones. Cada división es ocupada como set de prueba, y la unión de las divisiones restantes se utiliza como set de entrenamiento.

Matriz de Confusión

La matriz de confusión es un elemento que organiza los resultados del algoritmo de aprendizaje, de modo que se puede saber cuántos fueron acertados, y cuántas predicciones fueron incorrectas. Sus dimensiones se definen como N*N siendo N el número de clases que puede tomar el clasificador.

Así, observemos un ejemplo:

	SI	NO
SI	5	2
NO	1	10

Aquí, sí fue predecido correctamente por el clasificador cinco veces, y no fue predecido correctamente por el clasificador diez veces. Por otro lado, se predijo un NO cuando era un SI en dos ocasiones, mientras que se predijo un SI cuando era un NO en 1 ocasión.

III. Desarrollo

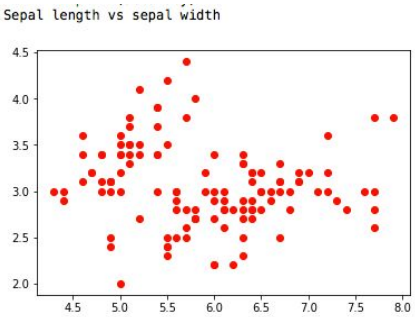
Para el desarrollo de la práctica se decidió utilizar un clasificador Naive Bayes, donde para calcular la probabilidad de que ciertos datos pertenezcan a una clase del dataset, se calcularon las medias y desviaciones estándar por clase. A partir de esto, se calcula la función de probabilidad gaussiana para cada una de las clases, y se selecciona el elemento que devuelve mayor probabilidad.

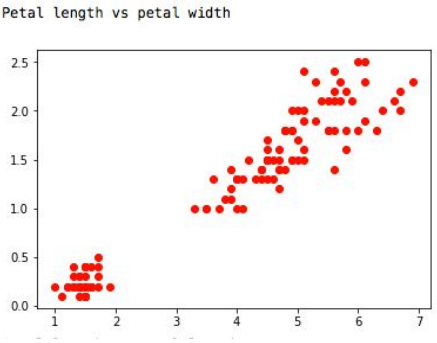
```
def calculateProbability(x, mean, stdev):
    exponent = math.exp(-(math.pow(x-mean,2)/(2*math.pow(stdev,2))))
    return (1 / (math.sqrt(2*math.pi) * stdev)) * exponent
```

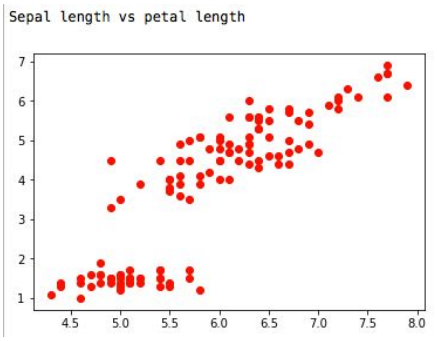
Para la validación cruzada, se eligió el método más simple, el enfoque de set de validación, y se dividió el set de datos en dos partes (ya dadas en la descarga del set de datos).
Finalmente, para calcular la matriz de confusión, se agruparon los datos de acuerdo a su clase, con un contador que incrementa dados los resultados del clasificador.

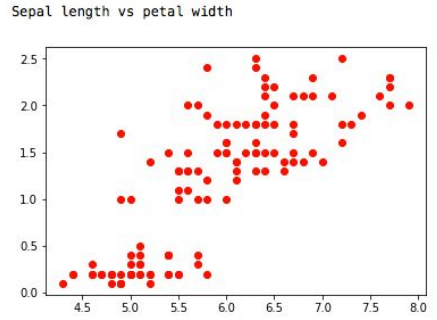
IV. Resultados

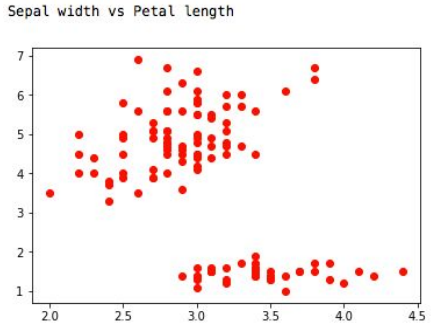
A continuación se presentan los resultados de la práctica, iniciando con una comparación visual de las diferentes variables del set de datos.

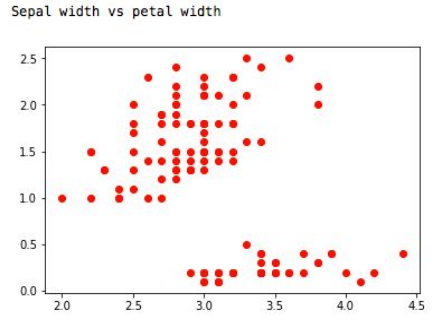












Figuras 1-6. Comparativas de variables del dataset

Posteriormente, se ejecutó el algoritmo, y se almacenaron los resultados devueltos al hacerlo con el set de prueba.

```
53 Iris Versicolor Iris-versicolor
54 Iris Versicolor Iris-versicolor
55 Iris Versicolor Iris-versicolor
56 Iris Versicolor Iris-versicolor
57 Iris Versicolor Iris-versicolor
58 Iris Versicolor Iris-versicolor
59 Iris Versicolor Iris-versicolor
60 Iris Versicolor Iris-versicolor
61 Iris Versicolor Iris-versicolor
62 Iris Versicolor Iris-versicolor
63 Iris Versicolor Iris-versicolor
64 Iris Versicolor Iris-versicolor
65 Iris Versicolor Iris-versicolor
66 Iris Versicolor Iris-versicolor
67 Iris Versicolor Iris-versicolor
68 Iris Versicolor Iris-versicolor
69 Iris Virginica Iris-versicolor
70 Iris Versicolor Iris-versicolor
```

Figura 7. Fragmento de los resultados devueltos por el algoritmo.

Para finalizar, se generó la matriz de confusión correspondiente y se calculó la efectividad del algoritmo, la cual dio un muy aceptable 96.6% de efectividad.

```
49 0 0
0 48 3
0 2 47
96.64429530201343
```

V. Conclusión

Es importante resaltar que este tipo de algoritmos de aprendizaje son muy útiles para automatizar procesos, y mejorarlos, sin embargo, es imperativo tomar en cuenta el margen de error, ya que en aplicaciones como la médica, obtener errores puede costar la vida.

Para esta práctica se contó con un set de datos relativamente corto, el cual se pudo manipular favorablemente, sin embargo, es bueno tomar en cuenta librerías que ya cuentan con funciones para implementar este tipo de cosas en menor tiempo.

VI. Referencias

[1] Neapolitan, R. (2004). Learning Bayesian Networks. Chicago, EU: Prentice Hall.

[2] Ray, S. (2015). Improve Your Model Performance using Cross Validation (in Python and R). 14/11/17, de Analytics Vidhya. Sitio web: https://www.analyticsvidhya.com/blog/2015/11/improve-model-performance-cross-validation-in-python-r/

[3]Mccrea, Nick. (2014). An Introduction to Machine Learning Theory and Its Applications: A Visual Tutorial with Examples. 14/11/17, de Toptal Sitio web: https://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-primer

[4] Brownlee, J. (2014). How To Implement Naive Bayes From Scratch in Python. 14/11/17, de Machine Learning Mastery Sitio web: https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/

[5] Carballo, W. (2017). Clasificación de regiones en imágenes. 14/11/17, de ITESM Puebla. Presentación de PowerPoint.