

# ANÁLISIS DE PARÁMETROS FUNDAMENTALES EN RECONOCIMIENTO DE PATRONES

## ANALYSIS OF FUNDAMENTALS PARAMETERS IN PATTERN RECOGNITION

Oropeza-Méndez, José Martín<sup>a</sup>, García-Rojas, Jesús Alberto<sup>a</sup>.

<sup>a</sup>Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, División de Ingeniería en Tecnologías de la Información y Comunicaciones Mixquiahuala de Juárez Hidalgo, México 42700. jmorpeza@itsoeh.edu.mx

**RESUMEN.** *En el ámbito del reconocimiento de patrones es útil identificar distintas características de las imágenes con el objetivo de utilizar el algoritmo adecuado para cada aplicación, recordando que "patrón" pueden ser imágenes o cualquier otro objeto que sea susceptible de ser clasificado. Considerando que existen diversos algoritmos que realizan esta tarea, en este trabajo se presenta una revisión y análisis de los parámetros utilizados en el algoritmo K-nearest neighbor (KNN) poniendo énfasis en conceptos como la importancia del valor de k (número de vecinos cercanos), la precisión (precision) y la exactitud (accuracy). Así como la interpretación de la matriz de confusión. Para tal efecto se utilizaron datos de prueba que fueron procesados mediante un programa escrito en Python.*

**Palabras clave:** Algoritmo, matriz de confusión, Python

**ABSTRACT.** *In the field of pattern recognition, it is useful to identify different characteristics of the images in order to use the appropriate algorithm for each application, remembering that "pattern" can be images or any other object that is susceptible to be classified. Considering that there are several algorithms that perform this task, this paper presents a review and analysis of the parameters used in the K-nearest neighbor (KNN) algorithm, emphasizing concepts such as the importance of the value of k (number of nearest neighbors), precision and accuracy. As well as the interpretation of the confusion matrix. For this purpose, test data were used and processed by means of a program written in Python.*

**Key words:** algorithm, confusion matrix, Python

### INTRODUCCIÓN

Considerando que para reconocimiento de patrones existen algoritmos de tipo supervisado y no supervisados en este trabajo se utiliza KNN, que es un algoritmo supervisado, lo que significa que se tiene un conjunto de datos iniciales y a partir de estos se intentará clasificar correctamente nuevos datos. A diferencia de otros algoritmos de aprendizaje supervisado, este algoritmo no genera un modelo, en su lugar "aprende" al momento de probar nuevos datos.

Como parte del proceso de clasificación, el algoritmo debe calcular la distancia de un nuevo elemento a cada uno de los existentes y ordenar dichas instancias para seleccionar a cuál grupo pertenece. El algoritmo KNN puede utilizar diferentes distancias, para fines de este trabajo se considera la distancia Minkowski y se analizan los conceptos de precisión y matriz de confusión.

El conjunto de datos que se utilizó proviene de una encuesta sobre la calificación que los usuarios de un

sitio de televisión de paga asignan a las películas, así como la edad, el género y si comprarán la película o no.

Finalmente, es importante definir cuál tipo de algoritmo es el más idóneo para cada tipo de objeto que se desea clasificar para obtener el mejor resultado posible y no menos importante es la correcta interpretación de los resultados.

### METODOLOGÍA

A partir de las encuestas, de cada cliente se puede obtener una serie de atributos como edad y género, para cada uno de ellos se tiene un atributo objetivo que identifica si compraría la película o no.

La información obtenida de las encuestas se procesa utilizando un programa escrito en Python, el cual cuenta con la librería `sklearn.neighbors.KNeighborsClassifier` que por definición utiliza como métrica la distancia Minkowski.

Esta distancia en términos matemáticos se define como la distancia entre dos puntos. Fórmula 1.

$$dist(x_i, x_j) = (|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \dots + |x_{ir} - x_{jr}|^h)^{1/h} \quad (1)$$

El conjunto de datos obtenidos en las encuestas se pasa a una forma matricial con tres características: dm indica cada una de las instancias (clientes), am representa cada uno de los atributos (edad, compra o no, etc) y c indica la clase a predecir.

Como parte del procedimiento experimental es necesario dividir en dos partes el conjunto de datos, en tal división regularmente se utiliza 70% para entrenamiento y 30% para validar, estos términos se denominan Train set y Test set. Tabla 1.

índice	título	edad	compra
1	titanic	19	1
2	cars	35	0
3	rápidosyfuriosos	26	0
4	agente007	27	0
5	avengers	19	1
6	capitanamerica	27	0
7	reyleon	27	1
8	kingkong	32	1
9	gladiador	25	0
10	naufrago	35	1

Tabla 1. Vista parcial del conjunto de datos

Para el análisis se convierte el conjunto de datos a formato .CSV y se obtiene 250 registros, se procesan con el programa en Python y como puede apreciarse se obtiene un accuracy de 0.38 para el set de clasificación y .067 para el set de prueba. Figura 1.

```
In [23]: n_neighbors = 1

knn = KNeighborsClassifier(n_neighbors)
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'
      .format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'
      .format(knn.score(X_test, y_test)))

Accuracy of K-NN classifier on training set: 0.38
Accuracy of K-NN classifier on test set: 0.67
```

Figura 1. Resultado grados de accuracy para el conjunto de datos.

La exactitud se refiere a la dispersión del conjunto de valores obtenidos a partir de mediciones repetidas de una magnitud, en otras palabras, es la cantidad de predicciones positivas que fueron correctas. Fórmula 2.

$$exactitud = \frac{TP + TF}{TP + TF + FP + FN} \quad (2)$$

Donde:  
TP= verdaderos positivos  
TF= verdaderos negativos  
FP= falsos positivos  
FN= falsos negativos

Cuando esta dispersión es menor entonces la precisión es mayor. La precisión se presenta por la proporción entre el número de predicciones correctas (ya sean positivas o negativas) y el total de predicciones, es decir el porcentaje de casos positivos detectados. Fórmula 3.

$$precisión = \frac{TP}{TP + FN} \quad (3)$$

Se obtiene la matriz de confusión con 0 falsos negativos y 1 falso positivo, se observa que la precisión es del 67 %. En este caso la interpretación es que una persona que fue clasificada por el algoritmo como no compradora de la película realmente si adquiere la misma. Por otro lado, no se clasificó a ninguna persona de forma errónea como compradora. Figura 2.

```
In [8]: pred = knn.predict(X_test)
print(confusion_matrix(y_test, pred))
print(classification_report(y_test, pred))

[[1 0]
 [1 1]]

          precision    recall  f1-score   support

     0       0.50      1.00      0.67         1
     1       1.00      0.50      0.67         2

 accuracy          0.67         3
 macro avg          0.75      0.75      0.67         3
 weighted avg          0.83      0.67      0.67         3
```

Figura 2. Matriz de confusión

## RESULTADOS

Como puede observarse se pueden determinar dos valores muy importantes, la exactitud y la precisión, así como la interpretación de la matriz de confusión.

El algoritmo implementado en Python muestra un desempeño adecuado para el tipo de implementación que se describe en el experimento ya que el puntaje f1-score es del 67 %. Cabe recordar que en el programa se utiliza la librería SKLEARN de Python lo cual puede influir en el resultado ya que es posible escribir paso a paso el algoritmo KNN y los resultados podrían variar ligeramente.

## CONCLUSIONES

En este trabajo se realizó la aplicación del algoritmo KNN, utilizando un mínimo de datos, sin embargo, sirve como referencia a los conceptos fundamentales del algoritmo. Para trabajos futuros se recomienda utilizar un conjunto de datos con mayor cantidad de instancias lo que permitirá observar mejor el desempeño del algoritmo

## REFERENCIAS

1. Sudkar (2015). Multi-view face detection using Deep Convolutional Neural Networks. Sitio Oficial de la Association for Computing Machinery.
2. Tefas (2016). Face detection based on deep convolutional neural networks exploiting incremental facial. 23<sup>rd</sup> International Conference on Pattern Recognition(ICPR), Cancun, Mexico.
3. Vázquez (2020). Sistema de reconocimiento facial mediante técnicas de visión tridimensional (Tesis de maestría Centro de Investigación en óptica). Repositorio Institucional
4. Amazon recognition. (2020). Automatice su análisis de imagen y video con aprendizaje automático. Recuperado 12 de mayo 2020 en <https://aws.amazon.com/es/rekognition/>