UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y MECÁNICA

ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS

Proyecto de investigación semestral titulado:

"Algoritmo K-means con pyspark para clasificación de imágenes"

Asignatura:

Aprendizaje Automático

Docente:

MONTOYA CUBAS, Carlos Fernando

Estudiantes:

- **INCA CRUZ, Carlos Eduardo** **171258**
- **HUAMAN HERMOZA, Antony Isaac** **170434**
- **PEREIRA CHINCHERO, Richard Mikhael** **171916**
- **QUISPE CHAMBILLA, Carlos Enrique** **174447**
- **QUISPE PALOMINO, Luiyi Antony** **174914**

Cusco – Perú
2021

1. Introducción

El presente proyecto aborda la temática del análisis de datos en un entorno big data, a su vez,se implementa el algoritmo de k-means utilizando pyspark para la clasificación de imágenes que contienen números escritos, con el fin de que cualquier persona interesada tenga las nociones necesarias para implementar este ambiente utilizando las herramientas de Hadoop y Apache Spark. En cuanto al análisis de datos, se busca la construcción de un clasificador de imágenes mediante el método de K-means, el cual es un algoritmo de clustering que fue utilizado con el API que ofrece la librería Apache Spark para Python.

Cabe mencionar, que la base de datos utilizada fue descargada de Internet en formato CSV y al trabajar con ETL no se realizó un proceso de depuración, solo se utilizó para cargar el dataset a HDFS. Al no contar con los equipos necesarios, el proyecto se desarrolló de forma stand-alone y además, debido a las características del equipo, la base de datos utilizada no fue de gran tamaño. Los resultados obtenidos al utilizar K-means se muestran solamente por consola.

2. Estado del Arte

En la actualidad, existen numerosos estudios e investigaciones relacionados con el campo de la clasificación de imágenes utilizando diferentes técnicas como grafos, redes neuronales, K-means. En este trabajo nos centraremos en este último algoritmo que es un algoritmo de clasificación no supervisada (clusterización) que agrupa objetos en k grupos basándose en sus características.

Recientes estudios en el campo de la investigación de clasificación de imágenes muestran, que formular los problemas de K-means en tiempo continuo configurado de una manera correcta, evita una conversión predefinida y permite encontrar de manera más rápida unas soluciones globales a través de Convex Optimization.

Sin embargo, los algoritmos de K-means son definidos de la forma que se busca obtener un tiempo continuo, existen muchas propuestas para mejorar el rendimiento como la eficiencia, pero podemos afirmar basándonos en trabajos relacionados, que esto todavía inmerso en investigaciones y desarrollos, ya que existen otras técnicas que le hacen frente a este algoritmo.

3. Marco Teórico

Clustering

El *aprendizaje de máquina* estudia el aprendizaje automático a partir de datos para conseguir hacer predicciones precisas a partir de observaciones con datos previos.

La clasificación automática de objetos o datos es uno de los objetivos del aprendizaje de máquina. Tenemos tres tipos de algoritmos:

En ***clasificación supervisada*** disponemos de un conjunto de datos que vamos a llamar datos de entrenamiento y cada dato está asociado a una etiqueta. Construímos un modelo en la fase de entrenamiento utilizando dichas etiquetas, que nos dicen si una imagen está clasificada correcta o incorrectamente por el modelo. Una vez construído el modelo podemos utilizarlo para clasificar nuevos datos que, en esta fase, ya no necesitan etiqueta para su clasificación, aunque sí la necesitan para evaluar el porcentaje de objetos bien clasificados.

En ***clasificación no supervisada*** los datos no tienen etiquetas (o no queremos utilizarlas) y estos se clasifican a partir de su estructura interna (propiedades, características).

En ***clasificación semi supervisada*** algunos datos de entrenamiento tienen etiquetas, pero no todos. Este último caso es muy típico en clasificación de imágenes, donde es habitual disponer de muchas imágenes mayormente no etiquetadas. Estos se pueden considerar algoritmos supervisados que no necesitan todas las etiquetas de los datos de entrenamiento.

Algoritmo K-means

K-means es un algoritmo de clasificación no supervisada (clusterización) que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. Se suele usar la distancia cuadrática.

El algoritmo consta de tres pasos:

***Inicialización*:** una vez escogido el número de grupos, k, se establecen k=centroides en el espacio de los datos, por ejemplo, escogiendo aleatoriamente.

Asignación objetos a los centroides*:* cada objeto de los datos es asignado a su centroide más cercano.

Actualización centroides*:* se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

Se repiten los pasos 2 y 3 hasta que los centroides no se mueven, o se mueven por debajo de una distancia umbral en cada paso.

El algoritmo *k-means* resuelve un problema de optimización, siendo la función a optimizar (minimizar) la suma de las distancias cuadráticas de cada objeto al centroide de su cluster.

Los objetos de se representan con vectores reales de d dimensiones $(x_1, x_2, \dots, x_n)$ y el algoritmo k-means construye k grupos donde minimiza la suma de distancias de los

objetos, dentro de cada grupo $S = \{S_1, S_2, \dots, S_k\}$, a su centroide. El problema se puede formular de la siguiente forma:

$$\min_{\mathbf{S}} E(\boldsymbol{\mu_i}) = \min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

donde S es el conjunto de datos cuyos elementos son los objetos x_j representados por vectores, donde cada uno de sus elementos representa una característica o atributo. Tendremos k grupos o clusters con su correspondiente centroide μ_i.

En cada actualización de los centroides, desde el punto de vista matemático, imponemos la condición necesaria de extremo a la función $E(\mu_i)$ que, para la función cuadrática anterior es:

$$\frac{\partial E}{\partial \boldsymbol{\mu}_i} = 0 \implies \boldsymbol{\mu}_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j$$

y se toma el promedio de los elementos de cada grupo como nuevo centroide.

Las principales ventajas del método k-means son que es un método sencillo y rápido. Pero es necesario decidir el valor de k y el resultado final depende de la inicialización de los centroides. En principio no converge al mínimo global sino a un mínimo local.

Desventajas del K-Means

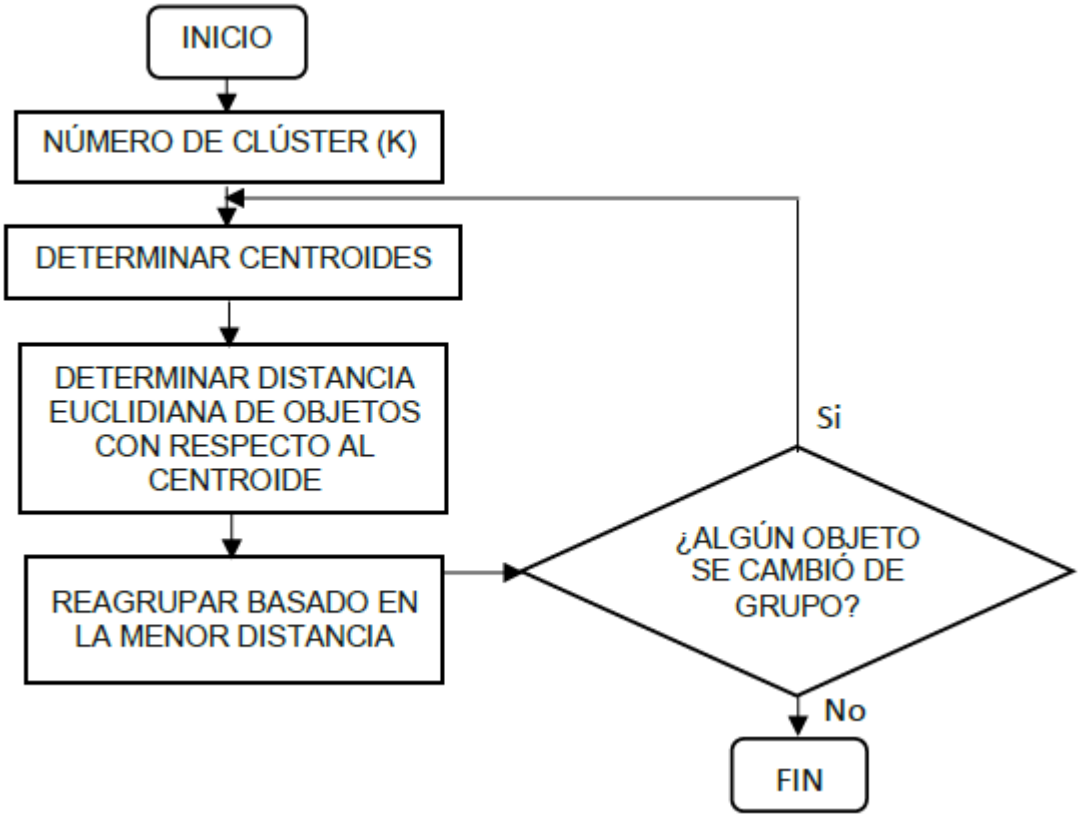
Ya hemos visto la potencia que tiene este algoritmo. Por lo sencillo que es de aplicar y la valiosa información sobre nuestros datos que nos aporta. Como no es oro todo lo que reluce, tengo que comentaros también las desventajas que ofrece:

- Tenemos que elegir k nosotros mismos. Es muy posible que nosotros cometamos un error, o que sea imposible escoger una k óptima.
- Es sensible a outliers. Los casos extremos hacen que el clúster se vea afectado. Aunque esto puede ser algo positivo a la hora de detectar anomalías.
- Es un algoritmo que sufre de la maldición de la dimensionalidad.

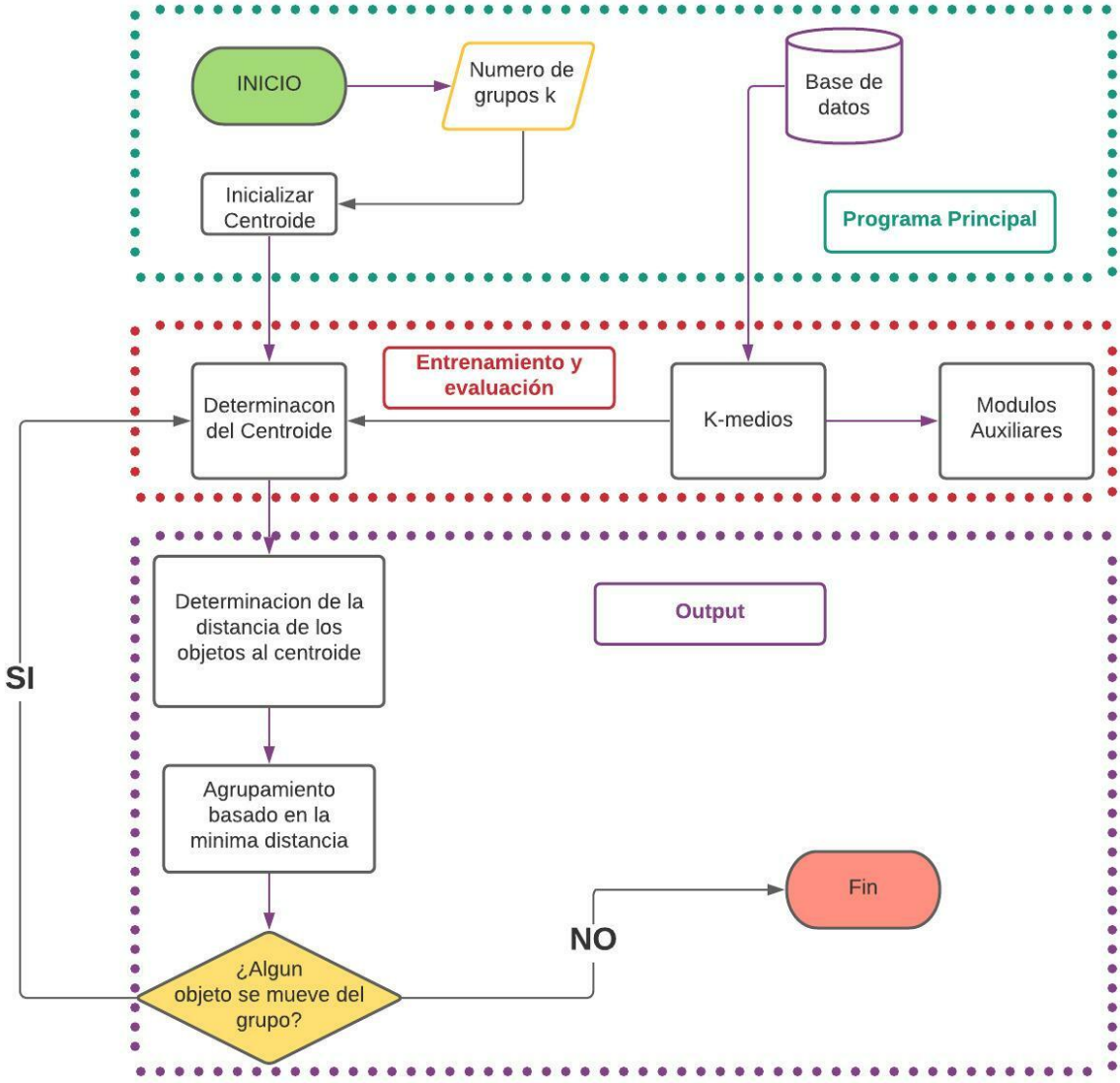
4. Implementación

Antes del proceso de la implementación se tiene que tomar en cuenta la estructura del algoritmo, en este caso hacemos una breve comparación de la estructura inicial del algoritmo k-means en **4.1.1.** con la estructura mejorada del algoritmo implementada en PySpark mostrado en **4.1.2.**

4.1.1. Diagrama de flujo:



4.1.2. Diagrama de flujo:



4.2. Importar Librerías

```
import numpy as np
import pickle
import sys
import time
from numpy.linalg import norm
from matplotlib import pyplot as plt
import pandas as pd
from google.colab import output
from google.colab import data_table
from PIL import Image
```

4.3. Instalación de Pyspark

```
!pip install pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("PySpark en Google
Colab").getOrCreate()
sc=spark.sparkContext
output.clear()
print(spark.sparkContext.appName)
```

4.4. Implementar algoritmo k means

```
def eleccion(p):
    '''
    Módulo que genera una muestra aleatoria de [0, len(p)),
    donde p[i] es la probabilidad asociada con i.
    '''

def kmeans_inicializacion(rdd, K):
    '''
    Módulo que selecciona conjuntos `RUNS` de puntos iniciales
para `K`-means
    '''

def obtener_mascercano(p, centers):
    '''
    Módulo que devuelve los índices a los centroides más cercanos
de 'p'.
    'centers' contiene conjuntos de centroides, donde
'centers[i]' es
    el i-ésimo conjunto de centroides.
    '''

def kmeans(rdd, K, converge_dist=0.1):
      '''
      Módulo que ejecute el algoritmo K-means en 'rdd', donde
      'RUNS' Es el número de conjuntos iniciales a usar.
      '''
def Comprobar(palabra):
```

```
    '''
    Módulo que comprueba si una cadena es número o texto
    '''


def kmeans_fit(rdd,K):
    "Módulo que devuelve un np.array con los clusters del rdd"
```

5. Conclusiones

- En primer lugar, destacar cómo en la mayoría de las imágenes reales, el número de etiquetas de la imagen segmentada, cuyos valores son máximos, no coinciden con la imagen de referencia. Esto es debido a que el ground truth no es totalmente realista al color y por lo tanto, existen problemas de coincidencia de píxeles.
- Dado que el algoritmo de k-medias se calcula en función de la distancia euclidiana, el algoritmo de k-medias es más sensible al rango de datos, por lo que antes de usar el algoritmo de k-medias, los datos deben estandarizarse para garantizar que el algoritmo de k-medias no se vea afectado por La influencia de las dimensiones de las características.
- El clustering es una técnica muy popular para problemas sin etiqueta y también para tareas de EDA. El K-Means es el rey de esta técnica por su sencillez tanto de entender como de aplicar. Se basa en agrupar los datos según la distancia entre ellos. Aunque como todo en esta vida tiene peros.

6. Bibliografía