**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**

**FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN**

**MAESTRÍA EN GESTIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN**

****

**Big Data**

**Alumnos:**

**Ledezma Ibarra Juan Francisco**

**López Ramírez Flavio Hiram**

**Práctica Final – Modelo Multilayer Perceptron Classifier.**

**Noviembre 2023, Tijuana Baja California, México**

Contenido

[Introducción 3](#_Toc151883293)

[Implementación 4](#_Toc151883294)

[Resultados 7](#_Toc151883295)

[Conclusiones 8](#_Toc151883296)

## Introducción

Los datos del dataset con que se trabajó, están relacionados con campañas de marketing directo (llamadas telefónicas) de una institución bancaria portuguesa. El objetivo es predecir la probabilidad de que un cliente abra una cuenta de inversión implementando el modelo Multilayer Perceptron Classifier.

## Implementación

Para este proyecto que consiste en predecir la probabilidad de que un cliente abra una cuenta de inversión implementando el modelo Multilayer Perceptron Classifier, para la limpieza de datos se utilizó el programa de Microsoft Excel, para la generación del código se utilizó Scala, finalmente este código se ejecuto Spark-Shell.

Limpieza de datos:

1.- Se realizó limpieza de dataset, eliminando las columnas “contact” y “poutcome”, ya que se consideran irrelevantes para el objetivo de la clasificación.

2.- Se reemplazaron las “;” a “,” para su correcta importación a scala.

3.- Se convirtieron los datos categóricos a numéricos dando como resultado la estructura mostrada en la siguiente imagen.

Texto

Descripción generada automáticamente

Dataset Limpio

.

4.- En las siguientes cuatro imágenes se ejecuta el código de escala el cual contiene las 10 corridas con las diferentes Seeds utilizada; de igual manera se presenta la evidencia del archivo readme.md y de los commits realizados en github:

Captura de pantalla de un celular

Descripción generada automáticamente

Imagen - Archivo Readme en Github

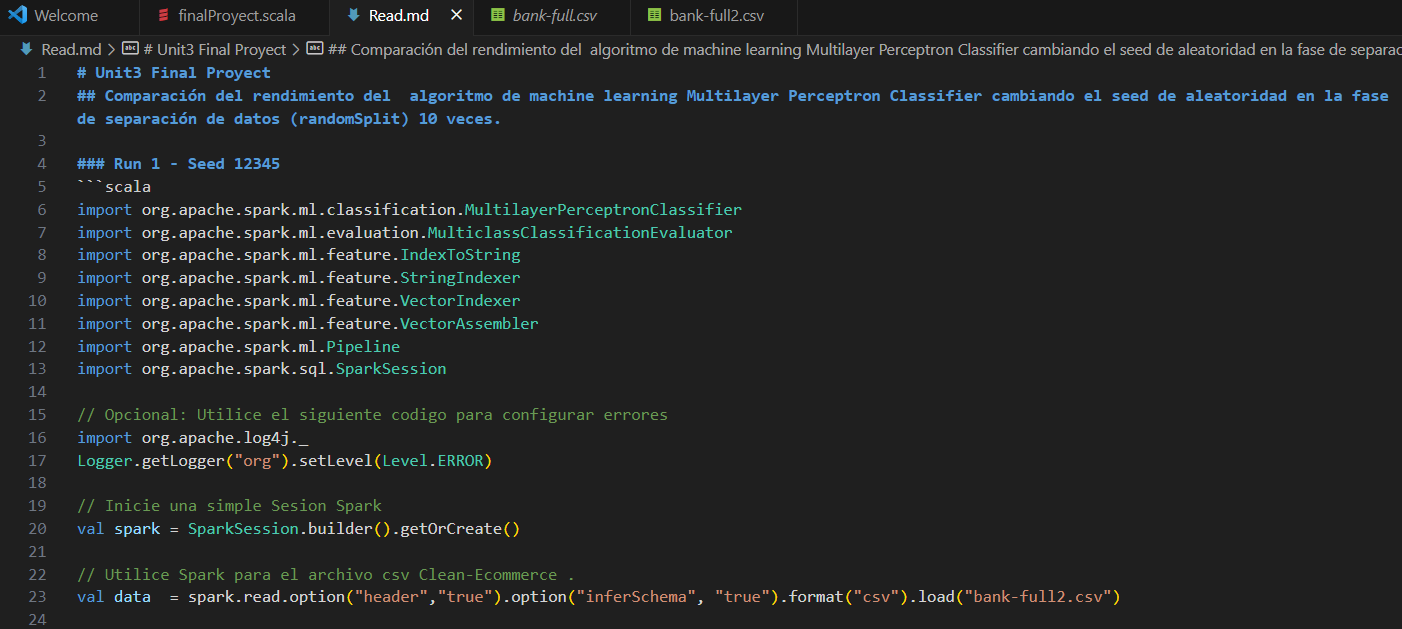


Imagen - Archivo Fuente de Readme (Visual Code)

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteImagen - Historial de Commits Generados en Github

Imagen - Código en Scala

## Resultados

En los resultados se puede observar como la exactitud del modelo cambia respecto a los Seeds utilizados.



## Conclusiones

Se llegó a la conclusión que el modelo entrenado con el Seed 24947 (run 4) fue el que nos arrojó el mejor porcentaje de exactitud con el 88.99%.