Generación de Modelos Inteligentes Usando PySpark y MLLib

Arturo Cristián Díaz López
Instituto Tecnológico y de Estudios Superiores de Monterrey

29-Oct-2024

Introduction

En este proyecto, se ha empleado el dataset Fashion MNIST con el fin de construir un modelo de clasificación de imágenes de ropa y accesorios. Este dataset ha sido seleccionado debido a su popularidad en tareas de clasificación de imágenes, así como su similitud estructural con el clásico MNIST, pero con un mayor desafío visual al contener diez categorías de productos de moda. Este reporte resume las etapas clave del proyecto, incluyendo la visualización del dataset en Tableau, el desarrollo de un modelo en PySpark y su evaluación final.

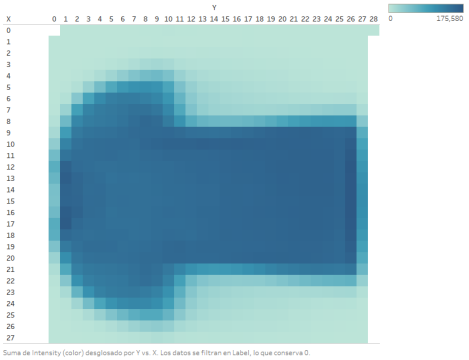

Figure 1: *Heatmap de imágenes de la clase T-Shirt.*

Dataset

El dataset Fashion MNIST contiene imágenes en escala de grises de 28x28 píxeles de 10 clases de artículos de moda. Cada imagen está etiquetada con un número de clase correspondiente a un tipo específico de artículo.

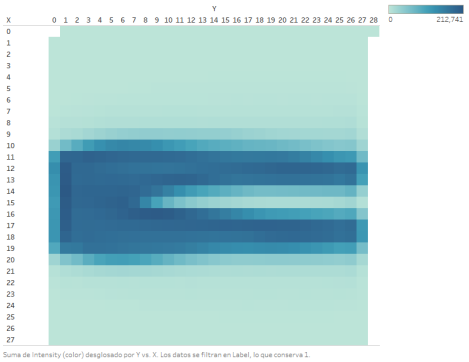

Figure 2: *Heatmap de imágenes de la clase Trouser.*

Estructura

El dataset cuenta con las siguientes 10 clases, pertenecientes a una categoría de artículo de moda.

- 0. T-shirt
- 1. Trouser
- 2. Pullover
- 3. Dress
- 4. Coat
- 5. Sandal
- 6. Shirt
- 7. Sneaker
- 8. Bag
- 9. Ankle boot

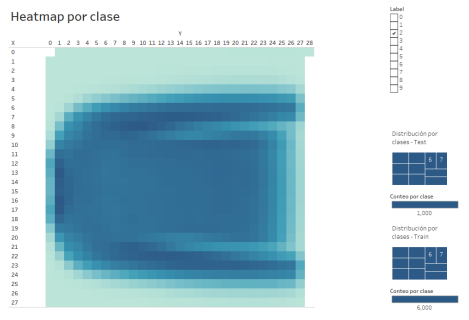

Figure 3: *Heatmap de imágenes de la clase Pullover.*

Visualización

Para mejorar la comprensión de las características visuales del dataset, se realizó una visualización en Tableau, explorando la distribución y las características de las imágenes en cada categoría. En esta visualización se observan diferencias entre clases.

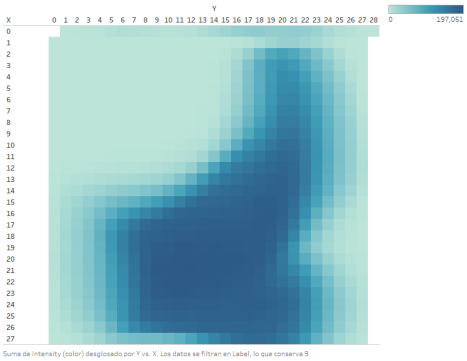

Figure 4: *Heatmap de imágenes de la clase Ankle boot.*

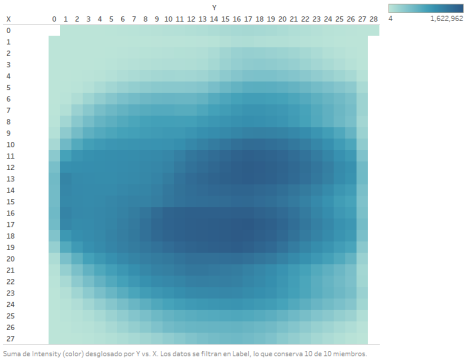

Figure 5: *Heatmap de imágenes de todas las clases.*

Adicionalmente, se realizó una visualización para verificar el número de ítems por clase. Esta visualización permitió encontrar que las clases se encuentran distribuidas equitativamente para todas las imágenes.

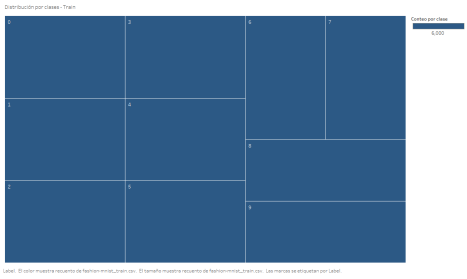

Figure 6: *Distribución por clases para el set de entrenamiento.*

Modelo

En este proyecto, se utiliza un Random Forest Classifier para la clasificación de imágenes en el conjunto de datos de Fashion MNIST. Este modelo es una elección popular para tareas de clasificación debido a su capacidad de manejar grandes conjuntos de datos y su flexibilidad para ajustarse a distintos tipos de problemas.

Implementación en Apache Spark

El modelo fue implementado usando PySpark, la interfaz Python de Apache Spark. A continuación se describen los principales pasos de esta implementación:

1. Configuración de Spark: Se estableció una sesión de Spark optimizada para el manejo de datos en paralelo, asignando memoria suficiente para la carga y el procesamiento del conjunto de datos.
2. Carga y Preparación de Datos: El conjunto de entrenamiento se almacenó en formato Parquet y se particionó en 100 bloques, mejorando así la eficiencia del entrenamiento distribuido.
3. Entrenamiento del Modelo: El clasificador se entrenó utilizando 100 árboles en el bosque aleatorio, configurado para equilibrar la precisión y la rapidez.
4. Almacenamiento del Modelo: El modelo entrenado se guardó para su evaluación y aplicación en futuras predicciones.

El Random Forest fue elegido por su capacidad de evitar el sobreajuste y por su buen rendimiento en entornos de datos complejos. La elección de PySpark permitió un manejo eficiente del volumen de datos y del entrenamiento en paralelo, logrando un modelo escalable y robusto.

Evaluación

La evaluación del modelo se llevó a cabo utilizando métricas estándar de clasificación multiclase. El modelo fue probado en el conjunto de datos de prueba, aplicando el modelo entrenado para generar predicciones sobre cada clase de Fashion MNIST.

Carga y Transformación de Datos

Se cargaron los datos de prueba y el modelo entrenado. Los datos de prueba fueron transformados para obtener predicciones y probabilidades asociadas a cada clase.

Métricas de Evaluación

1. **Exactitud (Accuracy)**: Medida del porcentaje de predicciones correctas realizadas por el modelo. **Valor obtenido**: 0.7648
2. **Precisión (Weighted Precision)**: Evalúa la precisión general del modelo ponderada por el número de muestras en cada clase. **Valor obtenido**: 0.7907

3. **Recall (Weighted Recall)**: Calcula el porcentaje de predicciones correctas sobre el total de instancias verdaderas por clase. **Valor obtenido**: 0.7648
4. **F1-Score**: Una medida combinada que equilibra precisión y recall, ponderada por el soporte de cada clase. **Valor obtenido**: 0.7434

Resultados

Se calcularon y almacenaron las métricas de evaluación, que son indicativas de un rendimiento aceptable del modelo sobre el conjunto de datos de prueba.

Almacenamiento de Predicciones

Las predicciones, junto con sus probabilidades y características originales, fueron exportadas a un archivo CSV para un análisis más detallado y posible visualización de resultados.

Este enfoque evaluativo permite identificar el rendimiento global del modelo y verificar la eficacia de sus predicciones en un conjunto de datos nunca antes visto, asegurando así su capacidad de generalización.

Conclusión

En conclusión, el modelo de clasificación desarrollado con el dataset Fashion MNIST ha demostrado una capacidad razonable para identificar las diferentes categorías de prendas y accesorios. La visualización en Tableau contribuyó al análisis exploratorio y facilitó una mejor comprensión de las características distintivas de cada clase. A través del uso de PySpark, se logró procesar grandes volúmenes de datos de manera eficiente, y se ha dejado abierta la posibilidad de optimizaciones futuras en función de los resultados obtenidos en la evaluación. Este proyecto sirve como una base sólida para el desarrollo de sistemas de clasificación más complejos en el campo de la moda y otras aplicaciones de visión por computadora.

Bibliografía

[1] Zalando. *Fashion MNIST Dataset*. https://github.com/zalandoresearch/fashion-mnist

[2] Apache Spark. *Apache Spark Documentation*. https://spark.apache.org/docs/latest/

[3] Apache Spark. *PySpark Documentation*. https://spark.apache.org/docs/latest/api/python/

[4] Apache Spark. *Machine Learning Pipelines*. https://spark.apache.org/docs/latest/ml-guide.html

[5] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32. https://link.springer.com/article/10.1023/A:1010933404324