Datathon 2023: Fashion Compatibility Challenge

Mango

Joan Saurina i Ricós
Sergi Tomàs Martínez
Jofre Moseguí Monterde
Marc Matute Juan

November 12, 2023

Contents

1 Introducción **2**

2 Modelo de CNN Siamés para estimar el Matching de 2 Productos **3**

3 Modelo de grafo **4**
1 Modelo 4
2 Funcionamiento 5

4 Modelo Tabular: Features "product_data.csv" **7**

5 Resultados **8**
1 Modelo grafo 8
2 Modelo Tabular 10
3 Modelo Siamés 11

6 Futuras Mejoras **13**

7 Conclusiones **14**

8 Implementación **15**

1 Introducción

La convergencia de moda y tecnología abre un abanico de posibilidades que desafía los límites de la creatividad y la innovación. Como participantes del *UPC Datathon 2023*, hemos abrazado el desafío de fusionar estos dos mundos a través del **Fashion Compatibility Challenge**. Este desafío nos ha brindado la oportunidad única de explorar la complejidad del estilo y el diseño de moda a través de las lentes del análisis de datos y la inteligencia artificial, con el objetivo de desarrollar un modelo que genere recomendaciones de atuendos a partir de una prenda inicial.

En este reto, la moda se revela como un dominio rico y multifacético, donde la elección de un conjunto adecuado va más allá de la coincidencia de colores y estilos; es una orquestación de texturas, formas y tendencias que resuenan con la personalidad y las preferencias del individuo. Hemos abordado la tarea de construir un sistema que no solo entienda los datos detrás de la moda, sino que también capture la esencia estética que hace que un atuendo sea considerado armonioso y atractivo.

Nuestro enfoque colectivo ha sido el de desarrollar un modelo integrado que no solo interprete las características tabulares y visuales, sino que también las sintetice para predecir combinaciones de prendas que se complementan entre sí. Este documento detalla el proceso colaborativo de nuestro equipo, desde el análisis exploratorio de datos hasta la implementación y ajuste del modelo, culminando con una evaluación rigurosa de las recomendaciones de moda resultantes.

2 Modelo de CNN Siamés para estimar el Matching de 2 Productos

Descripción del Modelo: Red Neuronal Convolucional Siamesa para Regresión

La arquitectura Siamesa es una red neuronal que consta de subredes gemelas que comparten pesos. Este diseño compara y mapea pares de entradas a un espacio de características común. En este caso, la arquitectura Siamesa se utiliza para la regresión, con el objetivo de predecir valores de 'cantidad' al procesar pares de imágenes. El objetivo a grandes rasgos es entrar al modelo dos pares de imágenes con sus respectivos valores de 'matching'. Estos valores de matching realmente son los outfits en los que ambos productos coinciden. Entonces con las convoluciones de las imagenes y los labels (matching) podremos entrenar el modelo de forma que siga con los criterios de 'matchin' vistos en el dataframe de Outfits.

Preprocesamiento de Imágenes

El modelo toma dos imágenes como entrada, que se cargan desde los paths encontrados en el dataset de Productos. Estas imágenes se procesan para crear pares para la red Siamesa. La columna 'imagenes' contiene tuplas, donde cada tupla contiene dos imágenes. Y éstas serán los inputs para el modelo en pares. Además para ajustar el modelo y conseguir que el modelo estime las relaciones correctas con otras prendas, se implementará un regresor para ajustar las dos convoluciones a el 'matching' que hacen los productos.

Estructura del Modelo Siamés

1. **Capa de Entrada**: El modelo comienza con dos capas de entrada, cada una recibiendo la forma de una sola imagen.
2. **Red Convolucional (CNN)**: Las capas convolucionales procesan cada imagen. Este modelo utiliza dos capas convolucionales con funciones de activación de unidad lineal rectificada (ReLU). Las capas de agrupación máxima siguen a cada capa convolucional para reducir las dimensiones espaciales.
3. **Aplanamiento y Fusión**: La salida de la CNN de cada imagen se aplana y se concatena. Esta representación fusionada combina las características aprendidas de ambas imágenes.
4. **Capas Densas para Regresión**: Las características fusionadas pasan por capas completamente conectadas (densas). El modelo utiliza activación lineal rectificada en capas intermedias y una activación lineal en la capa de salida para la regresión.

Entrenamiento del Modelo

- El modelo se compila utilizando el error cuadrático medio como función de pérdida y el optimizador Adam.
- Los pesos de clase se utilizan para manejar el desequilibrio de clases durante el entrenamiento.
- Después del entrenamiento del modelo, se grafica la evolución de la pérdida tanto para los conjuntos de entrenamiento como para los de validación a lo largo de las épocas utilizando Matplotlib.

3 Modelo de grafo

La motivación para emplear un modelo de grafo en radica en la capacidad intrínseca de los grafos para representar relaciones complejas entre artículos de moda. A diferencia de los modelos tabulares que pueden ignorar la interconexión y la influencia mutua entre productos, los grafos permiten capturar la naturaleza complementaria de las prendas donde los productos no se seleccionan por similitud sino por cómo se realzan entre sí. Al modelar el espacio de productos como un grafo, se puede aprovechar la estructura de datos para extraer y predecir características visuales y patrones de compatibilidad, lo que es fundamental para recomendar combinaciones de moda que van más allá de los metadatos, abordando así el reto de la compatibilidad de moda con una solución más alineada con la complejidad y la riqueza visual del dominio de la moda.

Una vez tenemos el modelo de grafo entrenado somos capaces de crear outfits a partir de una pieza nueva siguiendo un camino de 5 nodos del grafo con approach best-first.

Junto a las características de cada producto se agregado una representación vectorial de una dimensión de la imágen correspondiente a la prenda mediante una transformación de embeddings.

1 Modelo

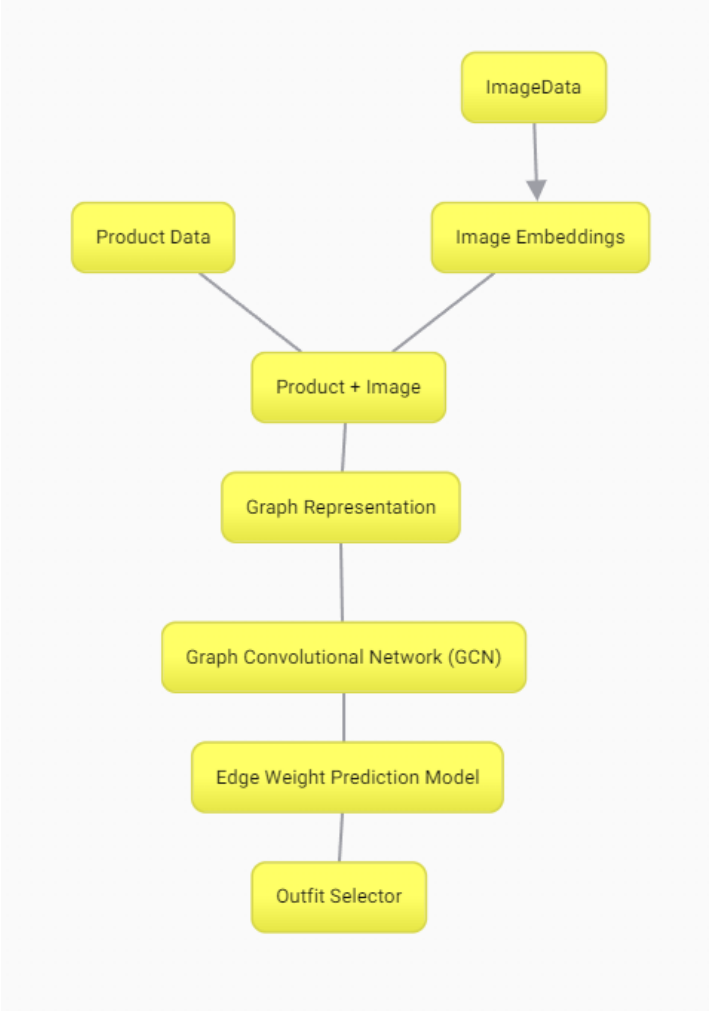

Figure 1: Modelo grafo + embedings imágenes

2 Funcionamiento

Para crear el conjuto Product + Imagen se tiene que procesar la imágene de cada producto de manera numérica. Para hacer eso el proceso de incrustación de la imagen se lleva a cabo mediante un autoencoder. Este método comienza con la definición de un codificador que toma la imagen de entrada y aplica sucesivas capas convolucionales y de agrupamiento máximo para reducir la dimensionalidad, seguido de un decodificador que realiza el proceso inverso, reconstruyendo la imagen a partir de la representación codificada.

La representación de cada iamgen que se añade como columnas al producto es la capa de menor dimensionalidad del Auto-encoder.

En segundo lugar, definimos un grafo $G = (V, E)$ donde V es el conjunto de nodos que representan productos y E es el conjunto de aristas que representan la co-ocurrencia de productos.

A cada arista $e \in E$ se le asigna un peso w_e que indica la frecuencia de co-ocurrencia de los productos conectados por e.

Este peso se ajusta con la función

$$f(x) = e^{0.2(x-2)}$$

e.j. $f(1)$ y $f(15)$:

Para $x = 1$:

$$f(1) = e^{0.5(1-2)} = e^{-0.5} \approx 0.606$$

Para $x = 15$:

$$f(15) = e^{0.5(15-2)} = e^{6.5} \approx 665.141$$

El objetivo de ello es dar mucha más impotancia a las conexiones que aparecen más veces.

La matriz de características de los nodos se denota como $X \in R^{n \times d}$, donde n es el número de nodos y d es el número de características por nodo.

En tercer lugar el modelo GCN está definido por las siguientes ecuaciones:

1. Inicialización:
$$H^{(0)} = X$$

2. Primera capa convolucional:
$$H^{(1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(0)}W^{(0)}\right)$$

3. Segunda capa convolucional:
$$H^{(2)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(1)}W^{(1)}\right)$$

4. Función de pérdida para entrenamiento:
$$\mathcal{L} = -\sum_{(i,j)\in E} \log(\sigma(\text{sim}(h_i^{(2)}, h_j^{(2)}))) + \sum_{(i,j)\notin E} \log(1 - \sigma(\text{sim}(h_i^{(2)}, h_j^{(2)})))$$

Donde:

- $\tilde{A} = A + I_N$ es la matriz de adyacencia del grafo G con auto-conexiones agregadas.
- I_N es la matriz identidad de $N \times N$.
- $\tilde{D}$ es la matriz diagonal de grados de $\tilde{A}$.
- σ es una función de activación no lineal, como ReLU.

- $W^{(0)}$ y $W^{(1)}$ son matrices de pesos entrenables para cada capa convolucional.
- sim es una función de similitud, como el producto punto.
- $h_i^{(2)}$ es el embedding del nodo i después de la segunda capa convolucional.
- E es el conjunto de aristas del grafo, y $(i, j) \in E$ denota que existe una arista entre los nodos i y j.

Por último, este modelo devuelve una probabilidad de existencia de una arista y ha sido entrenado y testeado tanto con aristas positivas (existentes en el grafo) como con aristas negativas, con etiqueta = 0.

Una vez somos capaces de predecir si una arista existe o no, se implementa un segundo modelo que determina el peso de dicha arista. La estructura del modelo es parecida pero realliza una regresión final que determina la relación entre 2 nodos. El objetivo de utilizar 2 modelos separados es intentar que este segundo modelo se entrene para una tarea mas concreta y con muestras menos ruidosas.

Pese a que este segundo modelo no ha sido implementado completamente, un esqueleto de este se ha adjuntado en el notebook *graph.ipynb*, en el apartado *Model Aresta*.

4 Modelo Tabular: Features "product_data.csv"

Después del preprocessing de "product_data.csv", nos quedan un total de 30 variables, puesto que és el merge de las features de dos productos.

Sin embargo, las variables para cada producto son los siguientes:

- **R,G,B:** 3 variables para el color del producto
- **One-hot de des_product_category**

Se ha intentado obtener resultados mediante los siguientes modelos de regresión:

- **Modelo Neuronal:** Se implementó un modelo neuronal utilizando técnicas de Deep Learning. Se utilizaron capas densas y funciones de activación como ReLU para aprender patrones complejos en los datos. La red neuronal se entrenó mediante un conjunto de datos etiquetado, ajustando los pesos de las conexiones entre las neuronas mediante el algoritmo de retropropagación.
- **XGBoost:** Se empleó XGBoost, una técnica de Machine Learning basada en árboles de decisión, para realizar la regresión. Este modelo utiliza un conjunto de árboles débiles que se combinan para formar un modelo más robusto. Se ajustaron los hiperparámetros del modelo para optimizar el rendimiento y se realizaron iteraciones para mejorar la precisión de las predicciones.

Aunque el modelo de XGBoost fue mejor, el fit que obtenia en el modelo del train era muy bajo, así que el resultado en el test tampoco fue bueno.

Así pues, finalmente, los dos modelos no fueron lo suficientemente buenos para la predicción de la variable objetivo: Número de outfits que ambos productos podrían ir bien [0, inf).

5 Resultados

1 Modelo grafo

Para el segundo approach, que es el del grafo, tenemos los siguientes resultados:

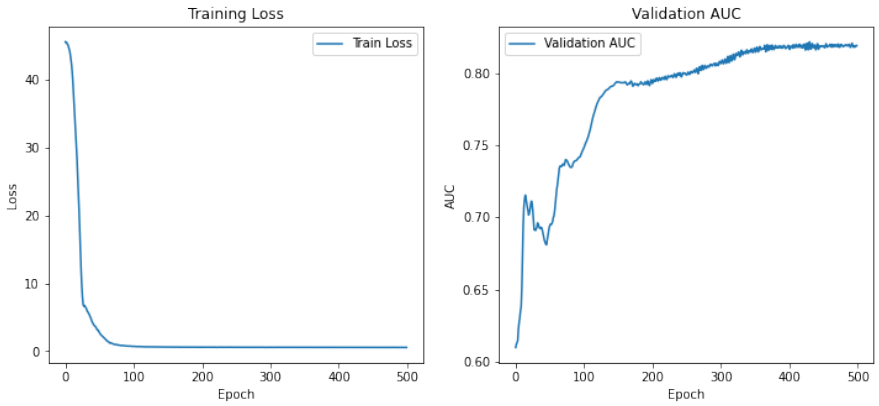

Figure 2: AUC y loss modelo grafo

En este caso, después de un aumento inicial y una fluctuación, la curva AUC de validación muestra una tendencia al alza a medida que avanza el número de epochs, lo que indica que el modelo está aprendiendo y mejorando su capacidad de hacer generalizaciones a partir de los datos de validación. Al final de los 500 epochs, el AUC se estabiliza en un valor alto (aproximadamente 0.85 o más), lo que es una señal prometedora de que el modelo tiene una buena capacidad de generalización. Esta estabilización sugiere que el modelo podría haber alcanzado su punto de convergencia y que entrenar por más epochs puede no llevar a mejoras significativas en la capacidad de discriminación del modelo.

Para un modelo que trabaja con datos de grafos, un AUC alto es particularmente impresionante ya que las relaciones y estructuras complejas dentro de los datos de grafos pueden hacer que la tarea de clasificación sea desafiante. Esto podría implicar que el modelo es efectivo en capturar la estructura y las características del grafo que son relevantes para la tarea de clasificación.

En resumen, basándose en esta curva AUC, parece que el modelo de grafo ha aprendido bien y podría proporcionar predicciones confiables sobre datos no vistos.

Sin embargo, después de comprobar los resultados de manera más detallada se observa que el model predice correctamente la mayoría de las muestras positivas pero

Outfit Generado

Utilizando una técninca de Best First Search se ha generado un outfit que maximiza las probabilidades de output del modelo. Pese a que se ha generado con las predicciones de aristas positivas, se ha comprobado que el outfit es nuevo y no existía con anterioridad en la base de datos. Es el siguiente:

2 Modelo Tabular

Para el tercer approach, tenemos el siguiente plot que muestra el entrenamiento del modelo:

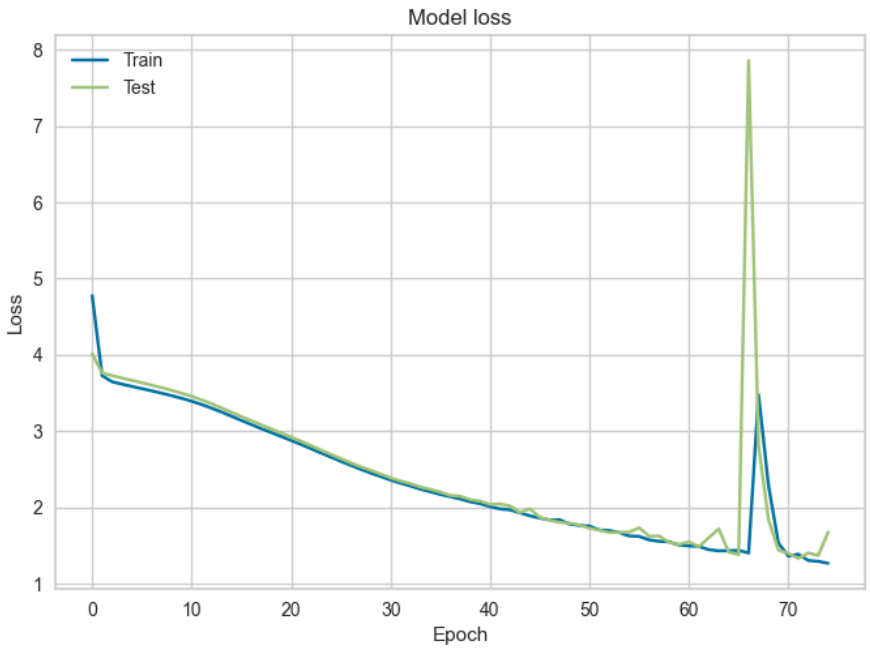

Figure 3: Pérdida del Modelo

Como vemos, el entrenamiento forma un overfitting que resulta en un 0.6 en r2 en el train y un 0.55 en el test.

En este sentido, este 3r approach no hemos podido generar un outfit por falta de tiempo, sin embargo, podemos generar las parejas de máxima similitud y la mínima para ver si tiene sentido. Veamos primero la de máxima:

Como vemos, tiene sentido. Ahora veamos la de mínima similitud:

En realidad, esta pareja de mínima similitud és correcta ya que no combinan y son las dos de la zona alta del cuerpo (por lo tanto, en un outfit solo entraría uno).

3 Modelo Siamés

El modelo siamés no ha dado los resultados esperados. La regresión final no ha conseguido que el modelo se ajuste a los labels intencionados. El modelo ha predecido productos de aspecto similar en

vez de basarse en los valores del dataset de outfits. Esto se puede deber a que el modelo de regresión no está lo suficientemente entrenado y no consigue ajustarse a los labels.

6 Futuras Mejoras

En primer lugar, para mejorar las aproximaciones existentes en este desafío, sería bueno explorar la posibilidad de integrar d contextual adicional en cada enfoque. Por ejemplo, en el modelo Siamese, podríamos considerar integrar datos sobre la ocasión o el estilo preferido del usuario para refinar aún más las recomendaciones. Esta personalización podría lograrse mediante la incorporación de datos de usuario en el entrenamiento del modelo. Esto sería, evidentemente, un approach personalizado para el cliente.

En segundo lugar, para la mejora de la aproximación basada en la similitud de imágenes mediante grafos, podríamos incorporar técnicas de aprendizaje semi-supervisado para aprovechar aún más la información presente en el conjunto de datos. Al permitir que el modelo aprenda de datos no etiquetados adicionales y ajustar sus pesos en consecuencia, podríamos mejorar la generalización del modelo y, por ende, su capacidad para predecir las aristas del grafo y, por tanto, las relaciones entre productos.

Finalmente, la tercera aproximación basada en datos tabulares podría beneficiarse de la inclusión de una de las otras aproximaciones, obteniendo un resultado con ponderación.

En realidad, combinando las salidas ponderadas de estos enfoques mejorados, podríamos obtener una medida de similitud definitiva que integre las fortalezas de cada método, proporcionando recomendaciones más sólidas y más relevantes en este Fashion Compatibility Challenge.

En cuanto al modelo siamés se tendría que haber estudiado más la arquitectura del regresor. Si se hubiera mejorado esta parte del modelo, el modelo siamés podría generar buenos resultados ya que aporta información tanto de las imágenes de similaridad como de los valores de 'matching' que se encuentran en el dataset de outfits. Si se tuviera que continuar en este trabajo se trabajaría en la estructura y se augmentaría su complejidad del regresor.

7 Conclusiones

Los resultados obtenidos hasta ahora sugieren que, si bien aún no hemos alcanzado un rendimiento óptimo / deseado, hemos identificado enfoques prometedores que, con mejoras y refinamientos, podrían ofrecer soluciones en el futuro. La complejidad de la base de datos y la naturaleza del problema presentan desafíos significativos, pero mantenemos la confianza en la capacidad de nuestros modelos para evolucionar y mejorar con el tiempo.

A pesar de las limitaciones actuales, creemos que nuestros enfoques son sólidos y sobre los cuales se podría construir. Si se nos otorgara más tiempo para iterar, ajustar parámetros y expandir nuestras estrategias de entrenamiento, estaríamos en una posición más fuerte para abordar las complejidades a la predicción de la compatibilidad de moda. A medida que continuamos refinando y perfeccionando nuestros modelos, mantenemos una perspectiva optimista y reconocemos que el proceso de mejora es esencial en el desarrollo de soluciones para un desafío como el actual: el Fashion Compatibility Challenge, planteado por Mango.

8 Implementación

El código fuente utilizado para los modelos, así como los procedimientos de procesamiento de datos y los ficheros resultantes, están disponibles de manera pública para su revisión y uso. Todos los recursos se han alojado en un repositorio de GitHub dedicado al proyecto. Los interesados pueden acceder al repositorio siguiendo la siguiente URL:

https://github.com/joansaurina/Datathon-2023

Se recomienda a los usuarios que se familiaricen con los archivos del repositorio para comprender mejor las técnicas implementadas y la estructura de los datos utilizados en este proyecto.