ISCTE-IUL
Mestrado em Ciência de Dados

Desafio 1 - Redes neuronais clássicas para classificação de mini-imagens

Trabalho de Grupo realizado no âmbito da Unidade Curricular de Aprendizagem Profunda para Visão por Computador do 1º ano do Mestrado em Ciência de Dados

Diogo Freitas, 104841, MCD-LCD-A1
Diogo_Alexandre_Freitas@iscte-iul.pt

João Francisco Botas, 104782, MCD-LCD-A1
Joao_Botas@iscte-iul.pt

Miguel Gonçalves, 105944, MCD-LCD-A1
Miguel_Goncalves_Pereira@iscte-iul.pt

Ricardo Galvão, 105285, MCD-LCD-A1
Araujo_Galvao@iscte-iul.pt

09 de março 2025
Versão 1.0.0

Índice

Parte 1 : Classificação Multi-classe

Pretende-se fazer a classificação de imagens do *dataset* FASHION_MNIST[1], que é composto por imagens, a preto e branco, de 10 tipos de peças de roupa (60000 imagens no conjunto de treino e 10000 imagens no conjunto de teste). Contudo, para validar e testar o modelo foi utilizado ainda um conjunto de validação com 10000 observações retiradas dos dados de treino.

Para construir a rede foi utilizada uma camada de input de 28 por 28 *features*/píxeis da imagem e depois transformada num vetor (1D), ou seja, com $28 \times 28 = 784$ *features*; uma camada escondida com 128 neurónios e função de ativação ReLu e uma camada de *output* com 10 neurónios, correspondente ao número de classes. Para além destas camadas foi utilizado um Dropout com um fator de 20% para evitar *overfitting* e aumentar a robustez do modelo. A rede foi ainda compilada com o otimizador Adam[2] (valor de `learning_rate` de 0.001 - ver outros valores testados no Anexo A); com a função de perda `categorical_crossentropy`, ideal para problemas de classificação multi-classe; e por métricas, entre elas a *accuracy*, *precision* e *recall*.

Construída a rede, o modelo foi posteriormente treinado com *batch size* de 64 e com "50 *epochs*", mas com duas *callbacks*: BEST_MODEL_CHECKPOINT (guarda os pesos do modelo com menor valor da função de perda) e EARLY_STOPPING (quando não há melhorias na função de perda).

Com isto, o nosso modelo terá $(784 \times 128) + 128 = 100480$ neurónios na camada escondida e $(128 \times 10) + 10 = 1290$ neurónios na camada de *output*. No total são 101770 parâmetros treináveis e 203542 não treináveis.

Os resultados obtidos foram os seguintes:

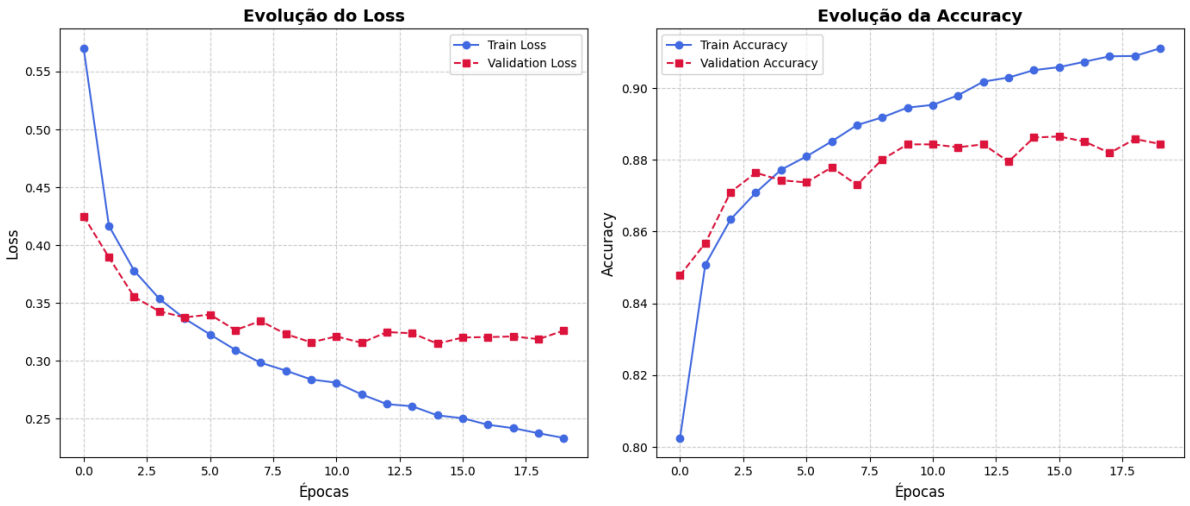

Figura 1: Evolução da Perda e da Acurácia ao longo das Épocas: Classificação multi-classe

Na Figura 1 podemos observar a evolução da função de perda e da acurácia ao longo das épocas do modelo. No gráfico da perda vemos que, a partir de uma determinada época, a perda começa a estagnar para o conjunto de validação e não acompanha o conjunto de treino. Após um certo número de épocas, o modelo acaba por parar pela *callback* de `early_stopping`, pois o valor da perda para a validação começa a decrescer pouco/a aumentar significativamente e a distanciar-se bastante da perda no conjunto de treino. Já no gráfico com a evolução da *accuracy*, observamos a mesma tendência, onde o modelo acaba por parar com de diferença entre o conjunto de treino e validação.

Após treinado o modelo e este ter sido validado fazemos a previsão para o conjunto de teste. Para avaliar a performance utilizamos a matriz de confusão da Tabela 1, juntamente com as métricas tradicionais. Entre estas destacamos a *accuracy*, as métricas individuais de cada classe e as métricas macro, que correspondem à média das métricas individuais de todas as classes, de forma a garantir que cada classe tenha o mesmo peso na avaliação.

[1] https://en.wikipedia.org/wiki/Fashion_MNIST

[2] Valor padrão da biblioteca: https://keras.io/api/optimizers/adam/

Tabela 1: Classificação multi-classe

Classe	Precision	Recall	F1-Score
T-Shirt/Top	0.86	0.79	0.82
Trouser	0.98	0.97	0.98
Pullover	0.81	0.76	0.78
Dress	0.86	0.90	0.88
Coat	0.78	0.80	0.79
Sandal	0.97	0.96	0.97
Shirt	0.68	0.71	0.69
Sneaker	0.94	0.95	0.94
Bag	0.95	0.98	0.96
Boot	0.96	0.96	0.96
Macr. avg	0.88	0.88	0.88
Accuracy		0.88	

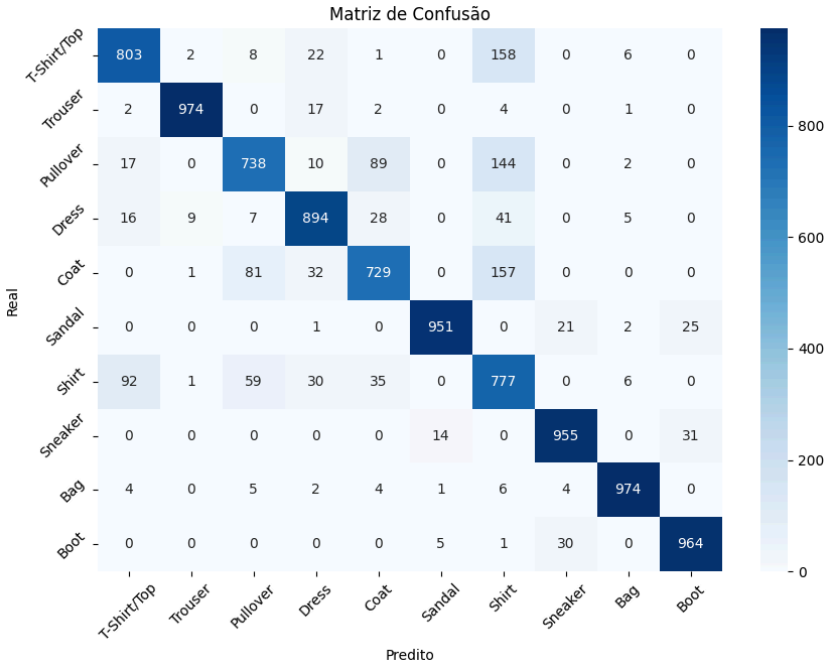

Figura 2: Matriz de confusão: Classificação multi-classe

Analisando os resultados do modelo de forma mais detalhada, é evidente que este apresenta um desempenho global bastante robusto. O Accuracy atinge um valor de 0.88, um resultado bastante positivo na identificação de peças de vestuário e acessórios. No entanto, ao observar a Tabela 1, verificamos que o modelo demonstra dificuldades na previsão de algumas categorias específicas. Um exemplo claro é a classe Shirt, onde a taxa de acerto é inferior à de outras peças. Em contrapartida, no que diz respeito a acessórios como Bag e Boot, o modelo apresenta um desempenho consistente, sem dificuldades notáveis. Na Parte 3, este modelo será binarizado de forma a responder à questão central deste trabalho.

Parte 2 : Classificação Binária

Para esta parte foram divididas as 10 classes originais do *dataset* para 2 grupos distintos: um de peças de Vestuário e outro de Calçado/Malas. As classes "T-Shirt/Top", "Trouser", "Pullover", "Dress", "Coat" e "Shirt" foram consideradas Vestuário, enquanto que as classes "Sandal", "Sneaker", "Bag" e "Boot" foram incluídas em Calçado/Malas. Estas classes foram utilizadas num modelo com os mesmos argumentos e *callbacks* da Parte 1, com as seguintes diferenças: a camada de *output* passou a ter um único neurónio com função de ativação *sigmoid*, enquanto que a função de perda foi alterada para `binary_crossentropy`. Ambas as alterações foram com o intuito de servir melhor um problema de classificação binária. Com a alteração na camada de *output*, o número de parâmetros treináveis nesta camada passará a ser $(128 \times 1) + 1 = 129$ e o número de parâmetros na camada escondida permanecerá igual (100480).

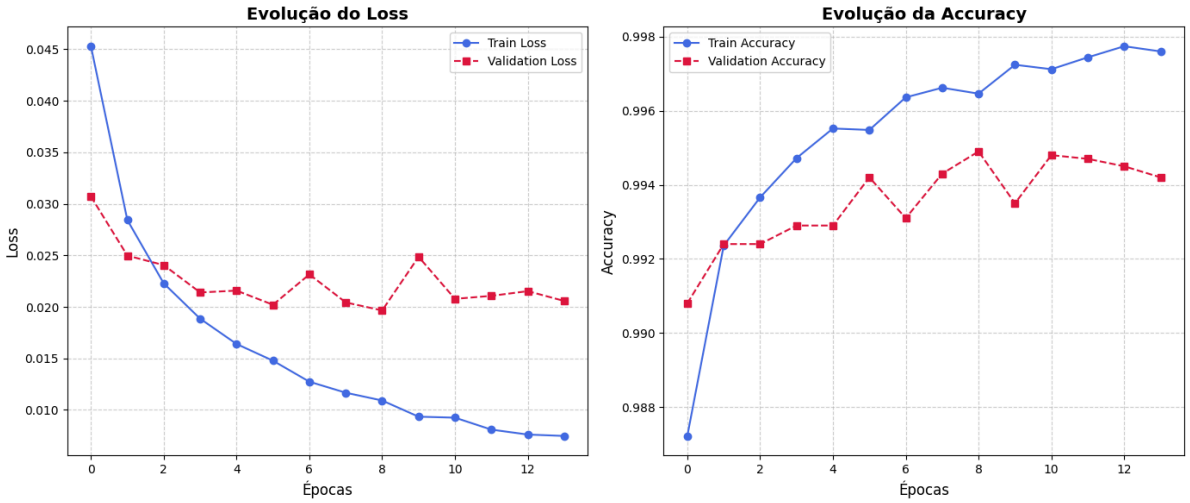

Figura 3: Evolução da Perda e da Acurácia ao longo das Épocas: Classificação binária

Pela Figura 3 podemos perceber que o modelo em poucas épocas de treino chega a uma acurácia no conjunto de validação de cerca de 0.992, acabando de treinar na época 13 devido a dar *overfit* no conjunto de treino em compromisso do conjunto de validação. Foram utilizados os pesos obtidos na época 8 como os finais.

Tabela 2: Relatório de Classificação Binary

Classe	Precision	Recall	F1-Score
Calçado/Malas	0.99	1.00	0.99
Vestuário	1.00	0.99	1.00
Macro avg	0.99	0.99	0.99
Accuracy		0.99	

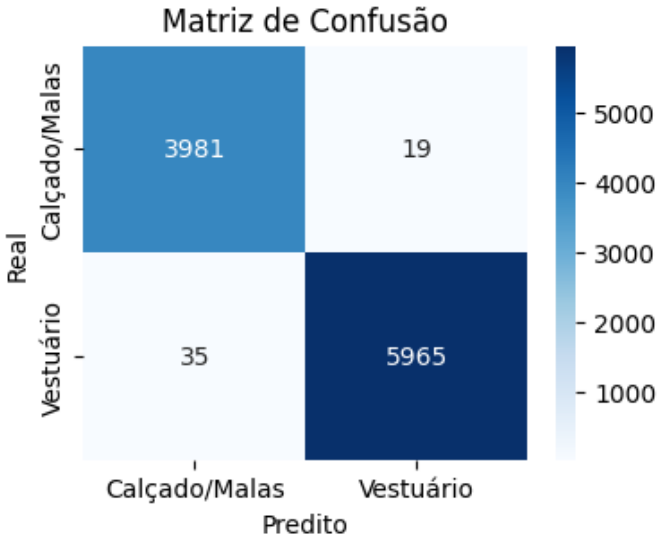

Figura 4: Classificação binária

Neste paradigma, o modelo pouco confunde as duas classes do problema: 35 imagens de Vestuário do conjunto de teste são classificadas como Calçado/Malas, enquanto que 19 imagens de Calçado/Malas do conjunto de teste são classificadas como Vestuário. Desta forma, o modelo apresenta uma acurácia de 0.99 na tarefa de classificação.

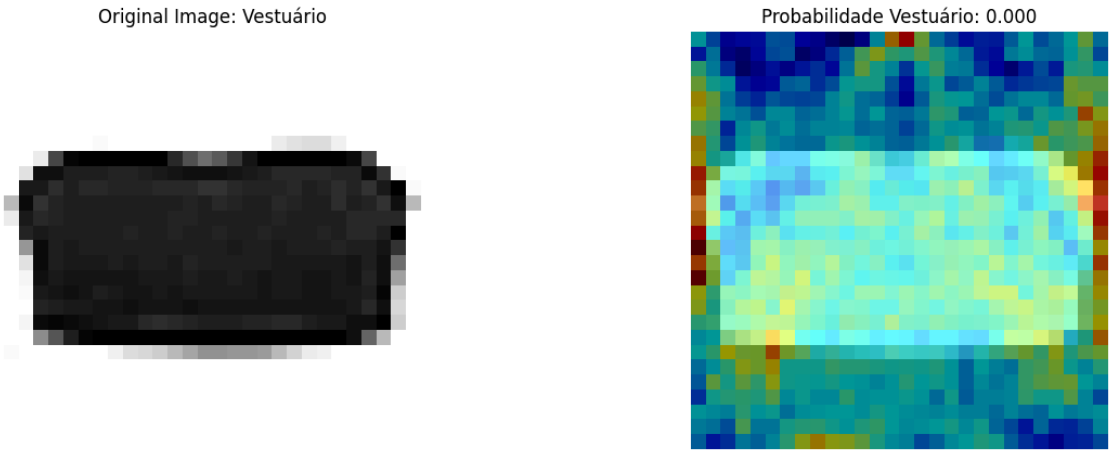

Figura 5: Exemplo de imagem de Vestuário classificada como Calçado/Malas. À direita, vemos um mapa de saliência do modelo para o exemplo. Os píxeis mais vermelhos são aqueles cujos valores o modelo mais importância deu para o resultado que obteve.

Parte 3 : Comparação de resultados

Após terem sido realizadas as classificações multi-classe e binária nas partes 1 e 2, respetivamente, prossegue-se à comparação de resultados e a resposta à seguinte questão:

"**Para a classificação binária Vestuário/Calçado e Malas será melhor usar uma rede para classificação multiclasse e depois binarizar as predições da rede neuronal, ou será melhor usar uma rede neuronal projetada para realizar diretamente a classificação binária?**"

Primeiramente, é necessário binarizar o modelo da Multi-classe (Parte 1). Para isso, iremos utilizar como base a matriz de confusão do modelo multi-classe original, que pode ser visualizada na Tabela 1, e iremos demonstrar os cálculos que serão realizados para a binarizar. A Figura 6 demonstra os cálculos que vão ser realizados.

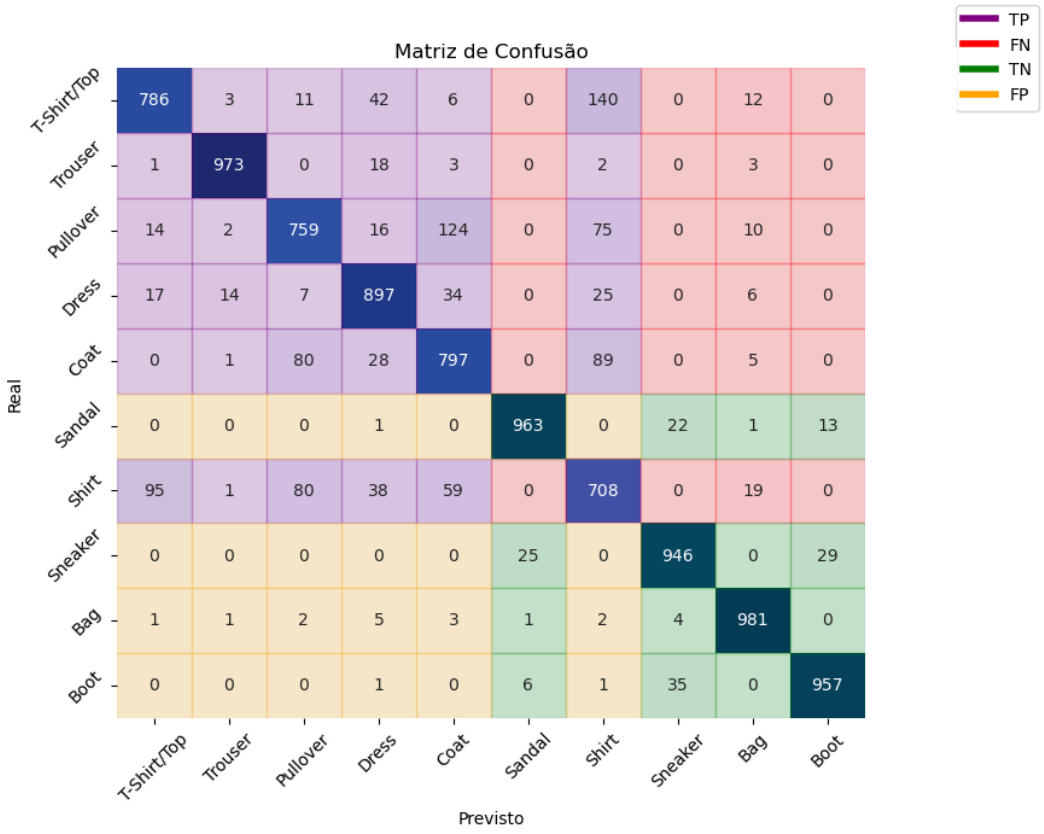

Figura 6: Matriz de confusão: Classificação multi-classe para Binarizar

Na matriz de confusão $M[i, j]$ da Figura 6, podemos identificar quatro blocos de cores distintas, que são somados para a construção da matriz de confusão binarizada. Para uma breve explicação da transformação, vamos utilizar os `True Negative`, que correspondem aos elementos $[i, j]$ com um fundo esverdeado na matriz. Esta vai corresponder aos casos em que o modelo previu classes do novo grupo Calçado/Malas (classe 0) corretamente, ou seja, $\sum_{i \in \text{Calçado/Malas}} \sum_{j \in \text{Calçado/Malas}} M[i, j] = 3983$. Este valor corresponderá aos `True Negatives` da nova matriz de confusão binarizada, na Figura 7. Para os cálculos completos pode-se consultar o Anexo B.

A matriz de confusão dos modelos mostra resultados semelhantes. A Tabela 3 apresenta a matriz dos modelos multi-classe após binarização, e a Figura 4 apresenta valores próximos, com uma ligeira melhoria no F1-score. Os resultados indicam que o desempenho dos dois modelos é quase idêntico.

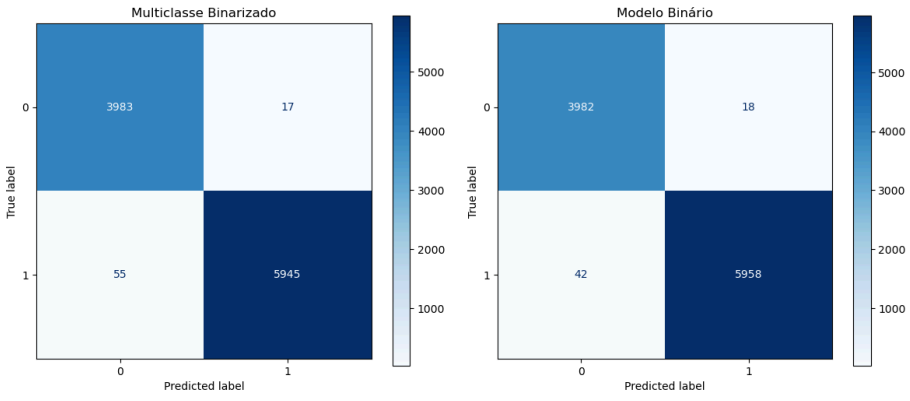

Figura 7: Comparação dos modelos: Matriz de Confusão

Tabela 3: Relatório de Classificação Binary

Classe	Precision	Recall	F1-Score
Calçado/Malas	0.99	1.00	0.99
Vestuário	1.00	0.99	0.99
Macro avg	0.99	0.99	0.99
Accuracy		0.99	

Para além dos valores da tabela, ambos os métodos possuem ROC-AUC de 0.99 (ver Anexo C).

Anexos

Anexo A - Tentativas com outros valores de `learning_rate` no Otimizador Adam do modelo de classificação multi-classe

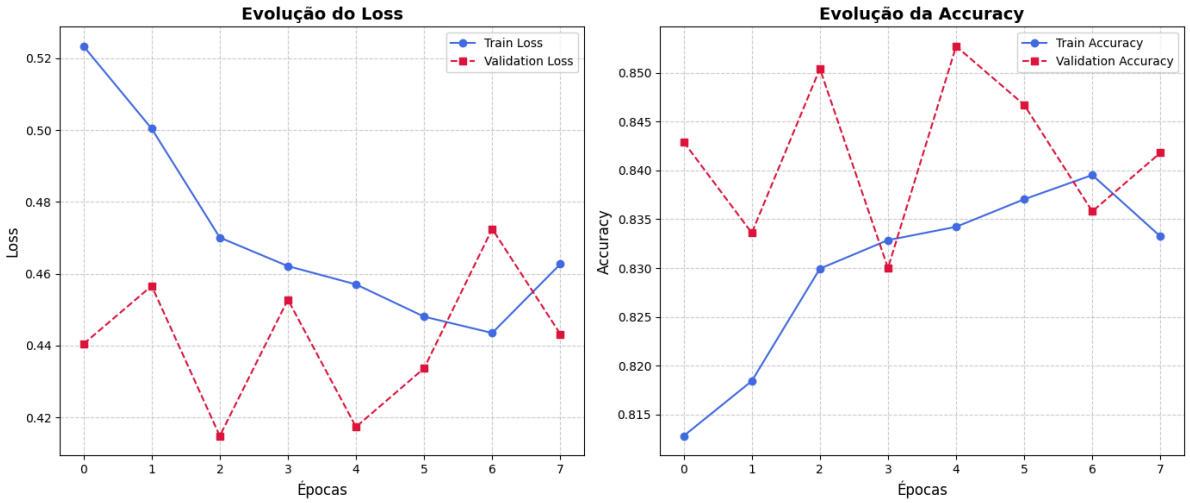

Figura 8: Modelo de classificação multi-classe com valor de **0.01** para o `learning_rate`

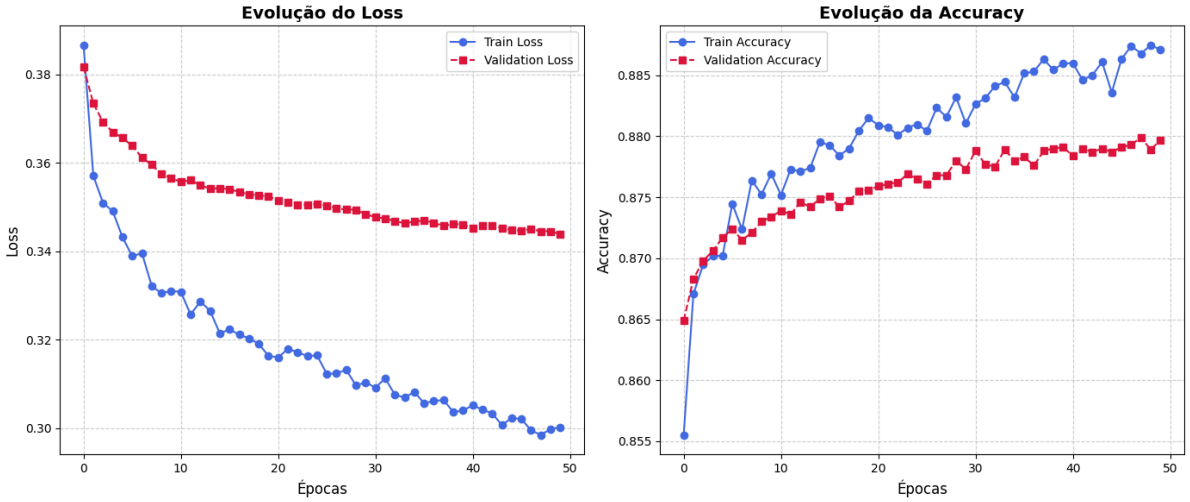

Figura 9: Modelo de classificação multi-classe com valor de **0.0001** para o `learning_rate`

Anexo B - Cálculos para binarização do modelo de classificação multi-classe

$$\textbf{FN} = \sum_{i \in \text{Vestuário}} \sum_{j \in \text{Calçado/Malas}} M[i,j] = \tag{1}$$

$$= 19 + 5 + 6 + 10 + 3 + 12 = 55$$

$$\textbf{FP} = \sum_{i \in \text{Calçado/Malas}} \sum_{j \in \text{Vestuário}} M[i,j] = \tag{2}$$

$$= 1 + 1 + 1 + 2 + 5 + 3 + 2 + 1 + 1 = 17$$

$$\textbf{TN} = \sum_{i \in \text{Calçado/Malas}} \sum_{j \in \text{Calçado/Malas}} M[i,j] = \tag{3}$$

$$= 963 + 946 + 981 + 957 + 22 + 1 + 13 + 29 + 4 + 35 + 25 + 1 + 6 = 3983$$

$$\textbf{TP} = \sum_{i \in \text{Vestuário}} \sum_{j \in \text{Vestuário}} M[i,j] = \sum_{i=0}^{9} \sum_{j=0}^{9} M[i,j] - (\text{FN} + \text{FP} + \text{TN}) = \tag{4}$$

$$= 10000 - (3983 + 17 + 55) = 5945$$

Anexo C - ROC-AUC para os dois métodos realizados

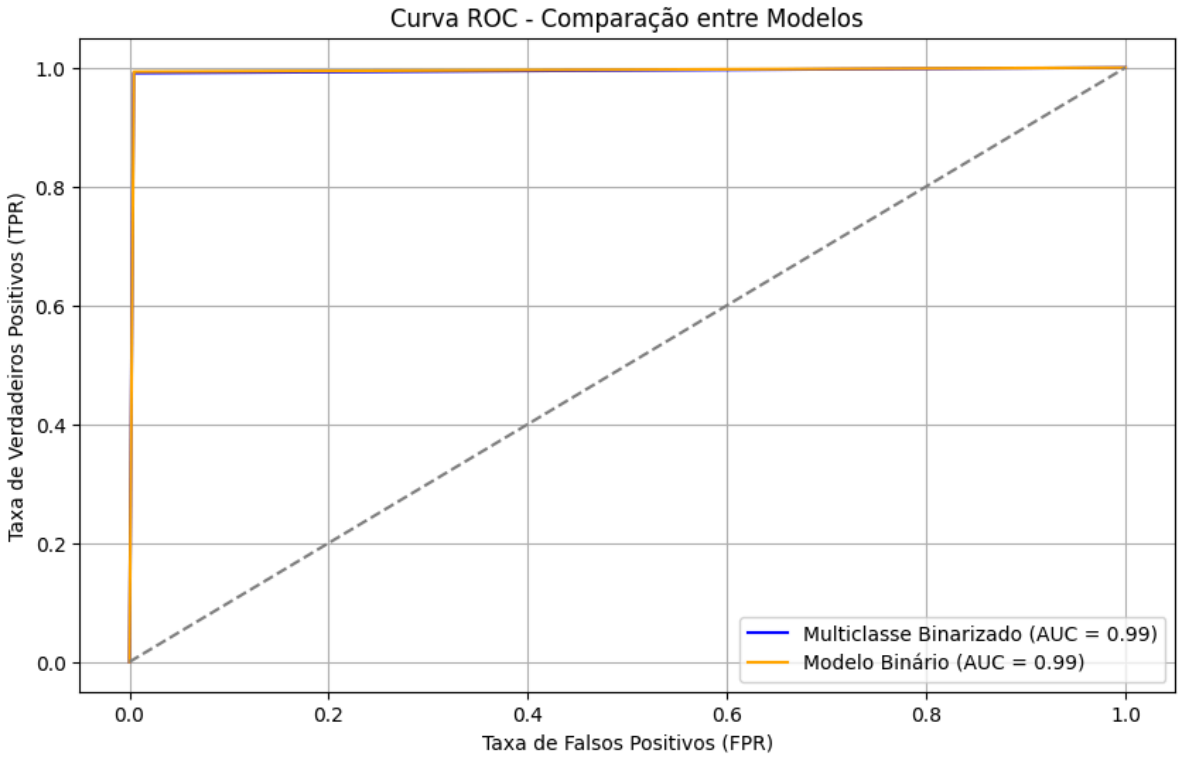

Figura 10: Comparação dos modelos com curva ROC