Trabalho Final
CPE727 – Aprendizado Profundo

Miguel Fernandes de Sousa

Universidade Federal do Rio de Janeiro
UFRJ/COPPE/PEE

Dezembro de 2025

Sumário

Tokenização (AG_NEWS)

- Word-level, vocabulário de 10.000 termos (+ <PAD>=0, <UNK>=1)
- Sequência fixa de 200 tokens: padding/truncamento no fim
- Comprimento médio ˜40 tokens (mín. 12, máx. 183) → ˜80% padding
- Preserva ordem das palavras; adequado para LSTM capturar dependências sequenciais

Introdução

- Comparativo entre modelos baseline (generativos e discriminativos) e de aprendizado profundo
- Dois datasets: Fashion MNIST (imagens) e AG_NEWS (texto)
- Modelos deep: CNN (Fashion MNIST) e LSTM (AG_NEWS)

Fashion MNIST

- 70.000 imagens (60k treino, 10k teste)
- 10 classes de vestuário
- 28×28 pixels = 784 features
- Normalização min-max

AG_NEWS

- 127.600 notícias (120k treino, 7.6k teste)
- 4 classes: World, Sports, Business, Sci/Tech
- TF-IDF com 10.000 features
- Remoção de stop words

Representação TF-IDF – Vetores de Entrada

Cada documento → vetor de 10.000 dimensões:

```
Documento: "Apple launches new iPhone"

Vocabulário (exemplo simplificado):
  "apple" → posição 0
  "launches" → posição 1
  "new" → posição 2
  "iphone" → posição 3
  ... (até 10.000 palavras)

Vetor TF-IDF (denso):
[0.234, 0.189, 0.156, 0.201, 0.0, 0.0, ..., 0.0]
  ↑      ↑      ↑      ↑      ↑    ↑           ↑
 apple launches new  iphone  outras palavras (zeros)
```

Características:

- Cada slot = uma palavra do vocabulário com posição fixa
- Valores = scores TF-IDF são floats entre 0.0 e ∼1.0
- Representação esparsa∼98% zeros

Fórmula TF-IDF – Cálculo dos Valores

TF-IDF(word, document) = TF $\times$ IDF

$$\text{TF}(w,d) = \begin{cases} 1 + \log(\text{count}(w,d)) & \text{se count}(w,d) > 0 \\ 0 & \text{caso contrário} \end{cases}$$

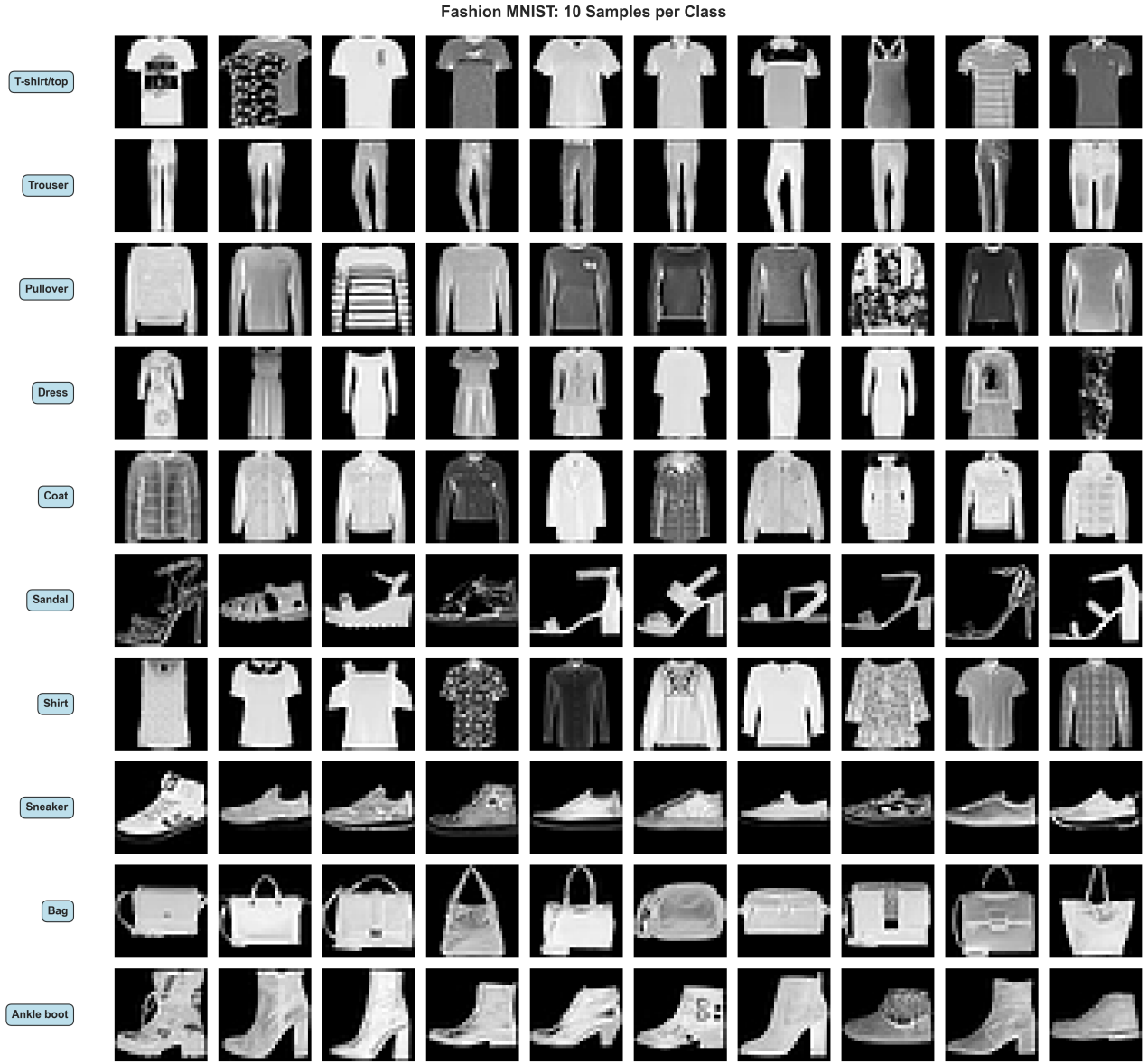
$$\text{IDF}(w) = \log\left(\frac{N+1}{df(w)+1}\right) + 1$$

Onde:

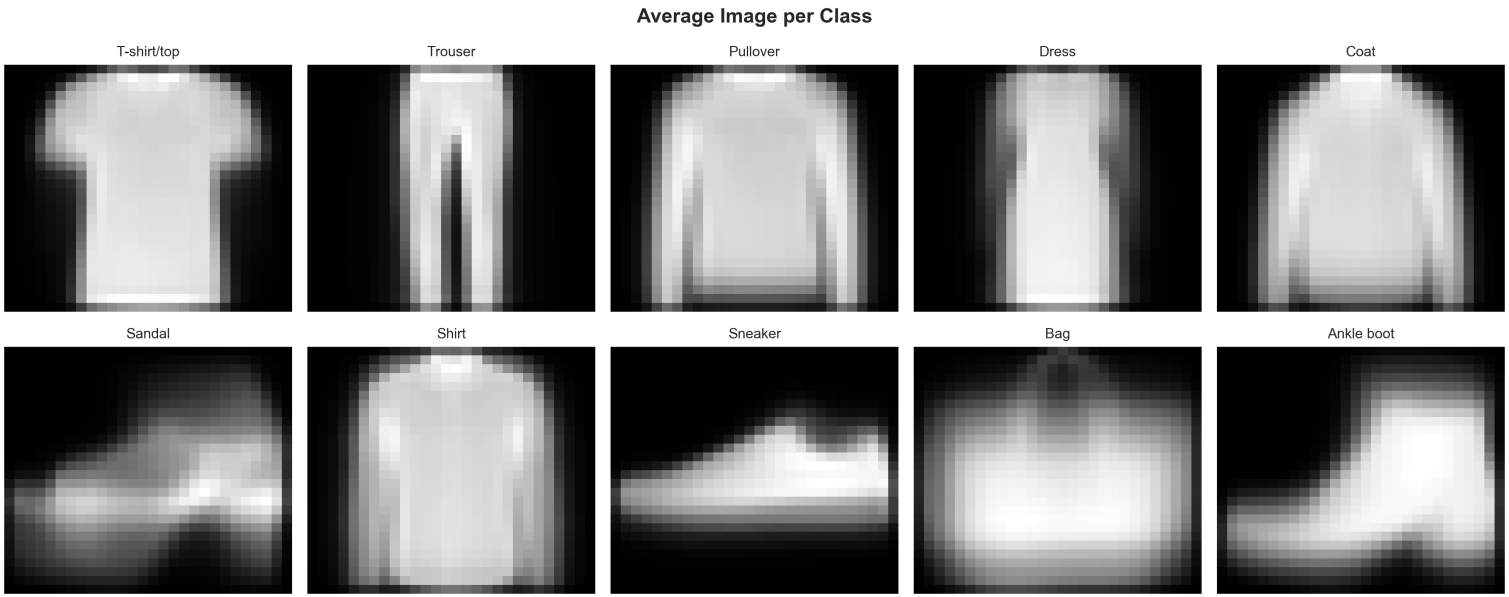
- $\text{count}(w,d)$ = frequência da palavra w no documento d
- N = número total de documentos (120.000 no treino AG_NEWS)
- $df(w)$ = número de documentos que contêm a palavra w

Faixa de valores: Após normalização L2, cada documento tem norma = 1, com valores individuais tipicamente entre 0 e 0.5.

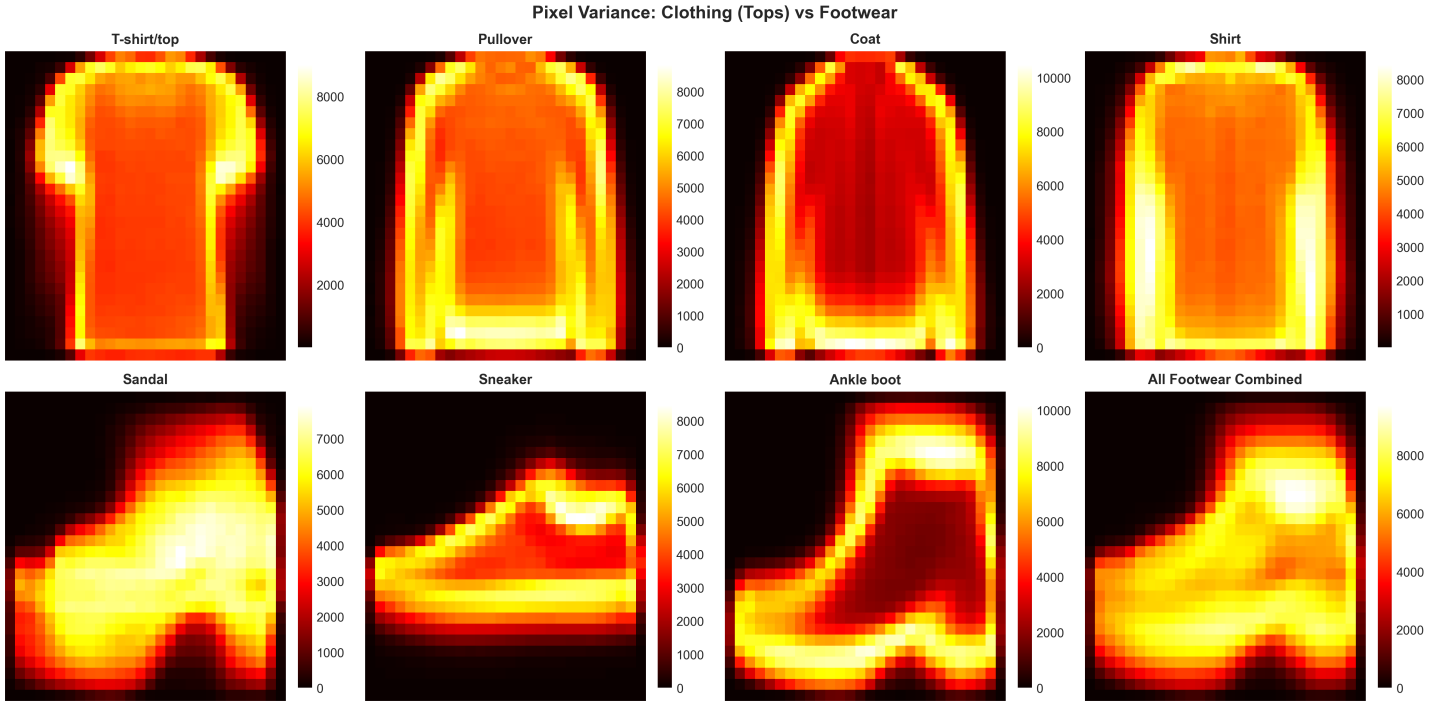
Distribuição de Amostras



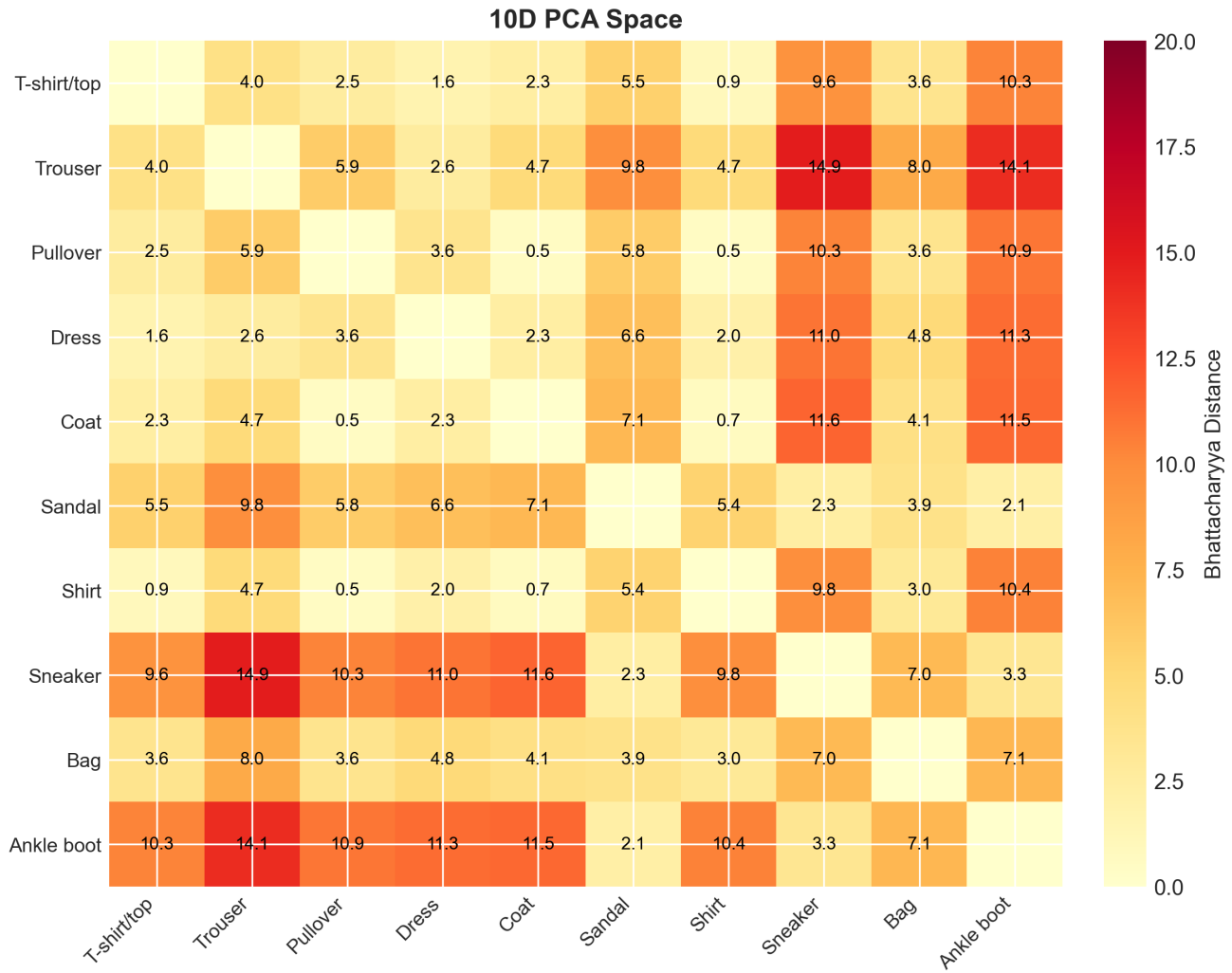
Imagens Médias por Classe



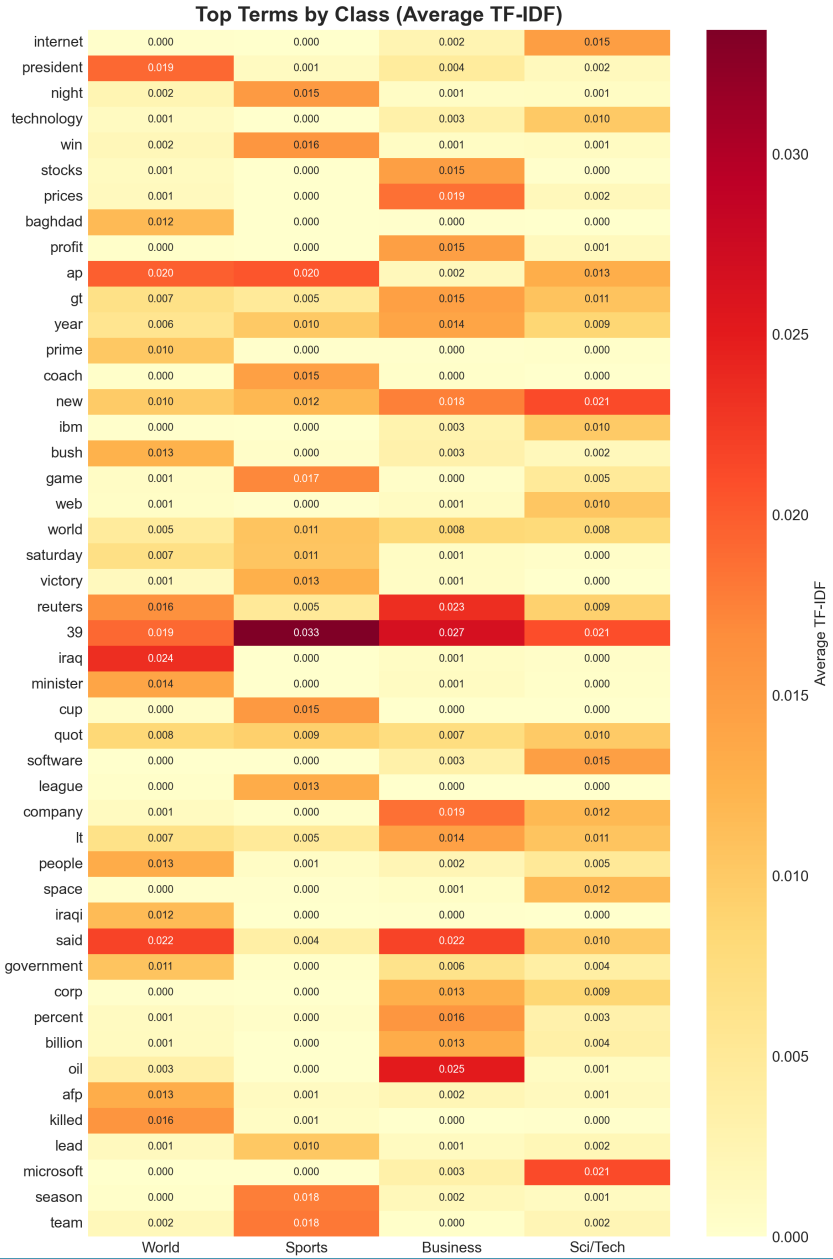
Variância: Roupas vs Calçados



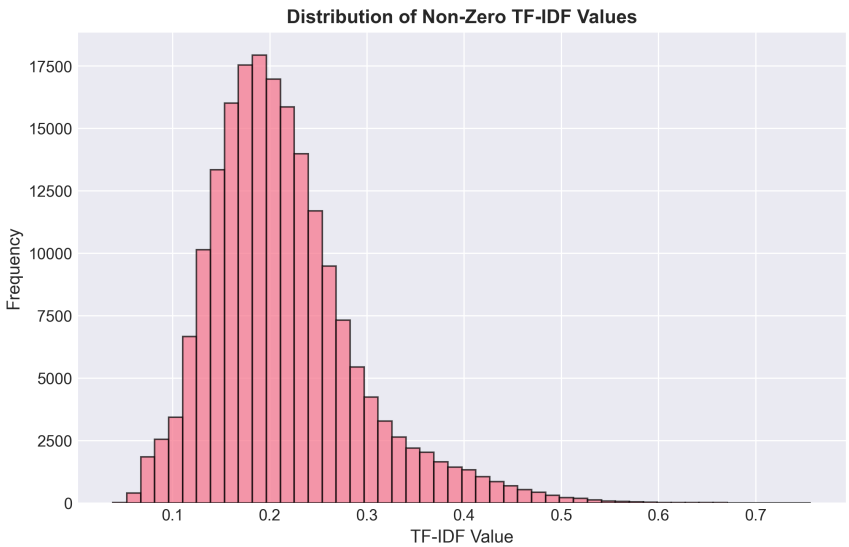
Matriz de Separabilidade entre Classes

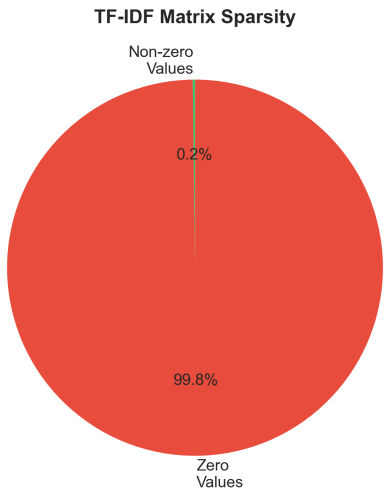


Mapa de Calor – Termos Mais Frequentes



Esparsidade da Matriz TF-IDF





Modelos Implementados

Generativos:

- **Naive Bayes** (3 variantes):
  - GaussianNB: distribuição normal (features contínuas)
  - BernoulliNB: distribuição binomial (features binárias)
  - MultinomialNB: distribuição multinomial (contagem/frequência)
- **GMM**: 1-4 componentes por classe
  - Covariância: completa (Fashion MNIST) / diagonal (AG_NEWS)
  - 5 inicializações, max 100-200 iterações

Discriminativos:

- **Regressão Logística**: norma L2, parâmetro C
  - Softmax (multinomial)
  - One-vs-Rest (OvR)
- **Random Forest**: ensemble de árvores de decisão

Otimização de Hiperparâmetros – Setup

- Grid Search com validação cruzada estratificada
- Fashion MNIST: 5-fold CV
- AG_NEWS: 2-3-fold CV (dataset maior)
- Métrica: Accuracy (com registro de Precision, Recall, F1)
- MLFlow para rastreamento de experimentos

Otimização

- Grid search em 2 estágios: amostra pequena para finalizar mais rápido, em seguida com amostra maior.
- CNN (Fashion MNIST): variação de lr, dropout, batch_size, conv_channels, kernel, padding, uso de batchnorm, FC
- LSTM (AG_NEWS): variação de lr, embedding_dim, hidden_dim, dropout, bidirecionalidade, batch_size
- MLflow para rastreamento de métricas, curvas e matrizes de confusão

Hiperparâmetros Finais (Deep)

CNN (Fashion MNIST)

lr=0.001, dropout=0.4, batch=32, epochs=20,
conv_channels=(32,64), kernel=3, padding=same, FC=128,
batchnorm=True, loss: CrossEntropy

LSTM (AG_NEWS)

lr=0.005, embedding_dim=100, hidden_dim=128, dropout=0.4,
bidirectional=True, batch=32, epochs=20, max_seq=200,
vocab=10k, loss: CrossEntropy

Melhores Hiperparâmetros (Fashion MNIST)

Configurações ótimas encontradas:

Modelo	Melhores Hiperparâmetros
Naive Bayes Gaussiano	var_smoothing = 1e-09 (F1 CV: 0.5065)
GMM	n_components = 1, covariance_type = 'diag'
Logistic Softmax	C = 1.0 (Accuracy CV: ∼84%)
Logistic OvR	C = 0.1 (Accuracy CV: ∼83%)
Random Forest	n_estimators = 200, max_depth = None, min_samples_split = 2, max_features = 'sqrt' (Accuracy CV: 88.05%)

Observação: Algoritmo EM da GMM com convergência instável em algumas execuções

Melhores Hiperparâmetros (AG_NEWS)

Configurações ótimas encontradas:

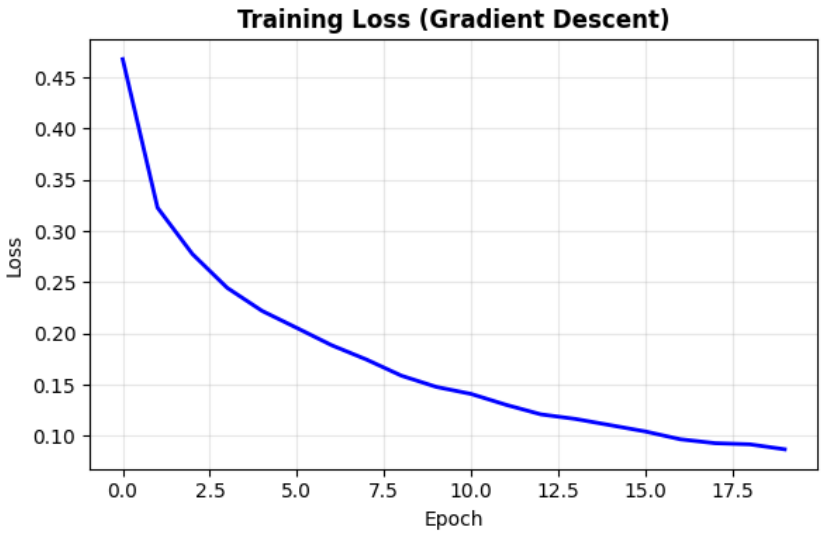
Modelo	Melhores Hiperparâmetros
NB Gaussiano	var_smoothing = 1e-05 (Acc CV: 80.31%)
NB Bernoulli	alpha = 0.1, binarize = 0.0 (Acc CV: 89.17%)
NB Multinomial	alpha = 0.01 (Acc CV: 87.92%)
GMM	n_components = 1, covariance_type = 'diag', n_init = 5, max_iter = 100 Acc CV: 80.97%)
Logistic Softmax	C = 1.0, max_iter = 1000 (Acc CV: 89.90%)
Logistic OvR	C = 10.0, max_iter = 1000 (Acc CV: 89.76%)
Random Forest	n_estimators = 300, max_depth = None, min_samples_split = 5, max_features = 'log2' (Acc CV: 87.70%)

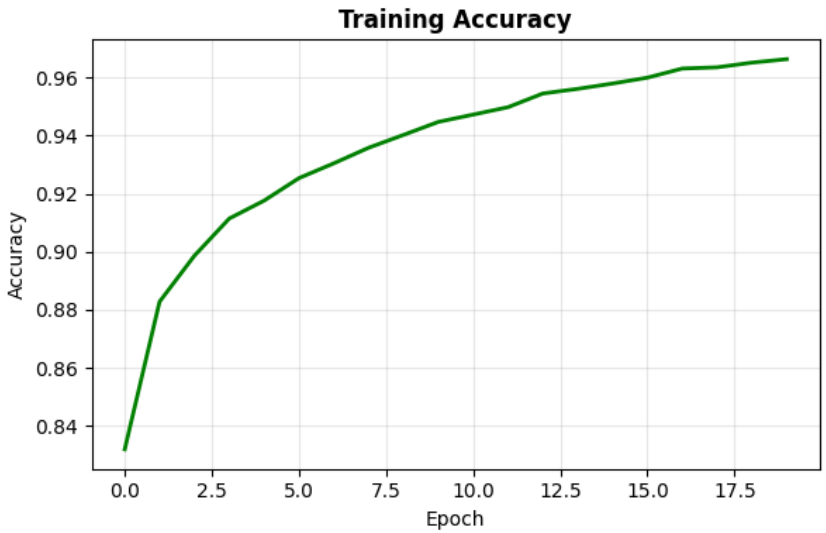
Resultados – Fashion MNIST

Modelo	Acc	Prec	Rec	F1
Random Forest	**87.84%**	87.70%	87.84%	87.70%
Logistic OvR	83.50%	83.74%	83.85%	83.78%
Logistic Softmax	83.40%	83.85%	83.75%	83.76%
NB Multinomial	65.55%	65.42%	65.55%	62.76%
NB Bernoulli	64.82%	66.58%	64.82%	63.98%
NB Gaussiano	59.10%	62.03%	58.40%	55.74%

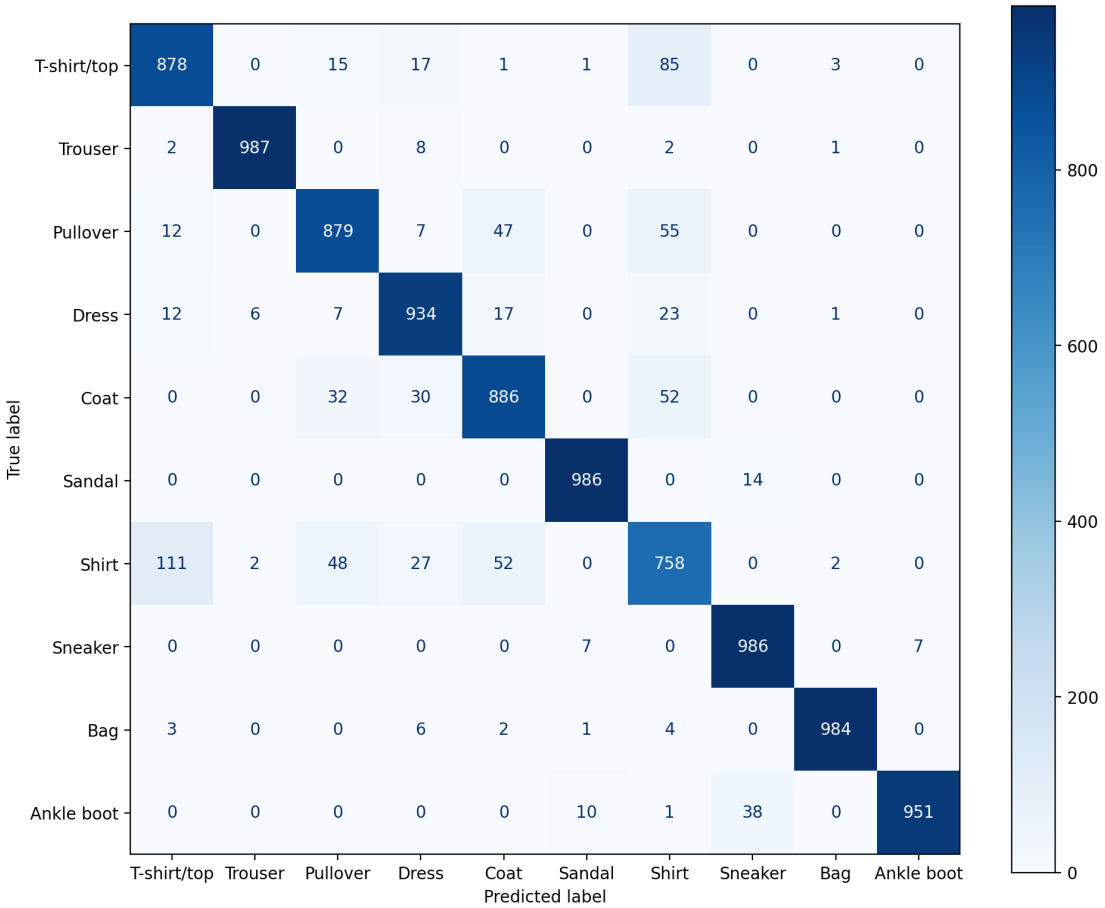
Resultados Deep – CNN (Fashion MNIST)

- Acurácia de teste: **92.29%**
- Precision macro: **0.9230**, Recall macro: **0.9229**, F1 macro: **0.9228**
- Melhor que todos os baselines, inclusive Random Forest com 87.84%





Resultados Deep – CNN (Fashion MNIST)



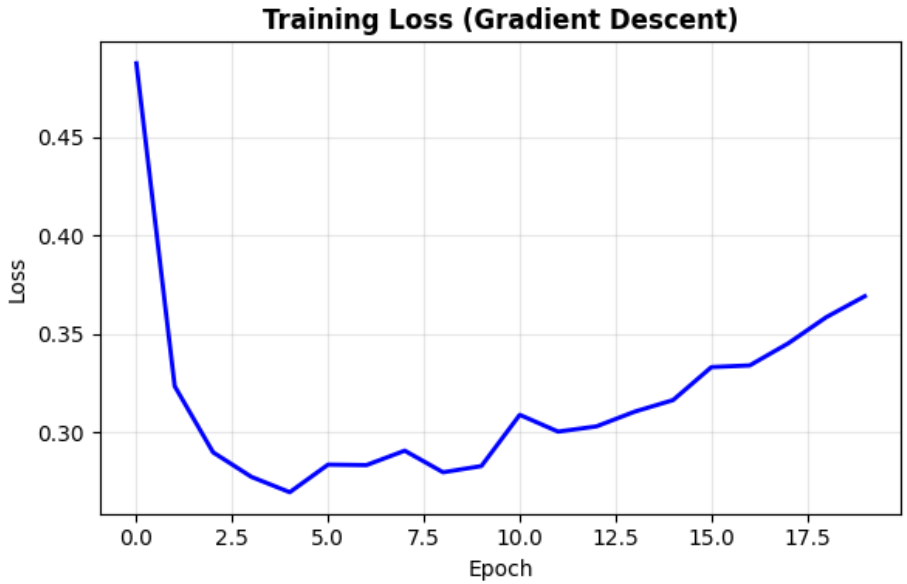
Resultados – AG_NEWS (Conjunto de Teste)

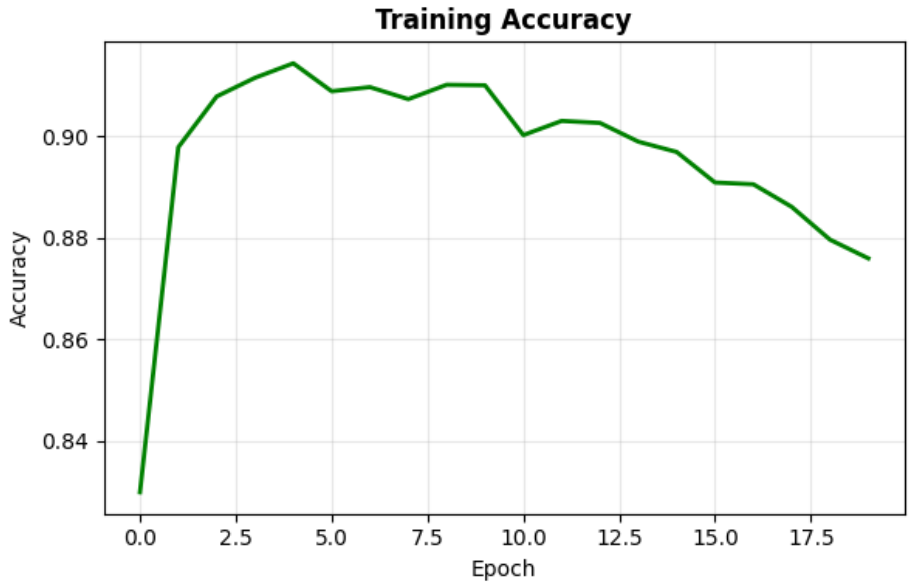
Modelo	Acc	Prec	Rec	F1
Logistic OvR	**91.34%**	91.32%	91.34%	91.32%
Logistic Softmax	**91.24%**	91.22%	91.24%	91.22%
Random Forest	90.96%	90.94%	90.96%	90.92%
NB Multinomial	89.66%	89.61%	89.66%	89.62%
NB Bernoulli	89.36%	89.31%	89.36%	89.31%
GMM	86.89%	86.91%	86.89%	86.90%
NB Gaussiano	86.64%	86.68%	86.64%	86.64%

O gap entre os discriminativos e o naive bayes diminuiu bastante.

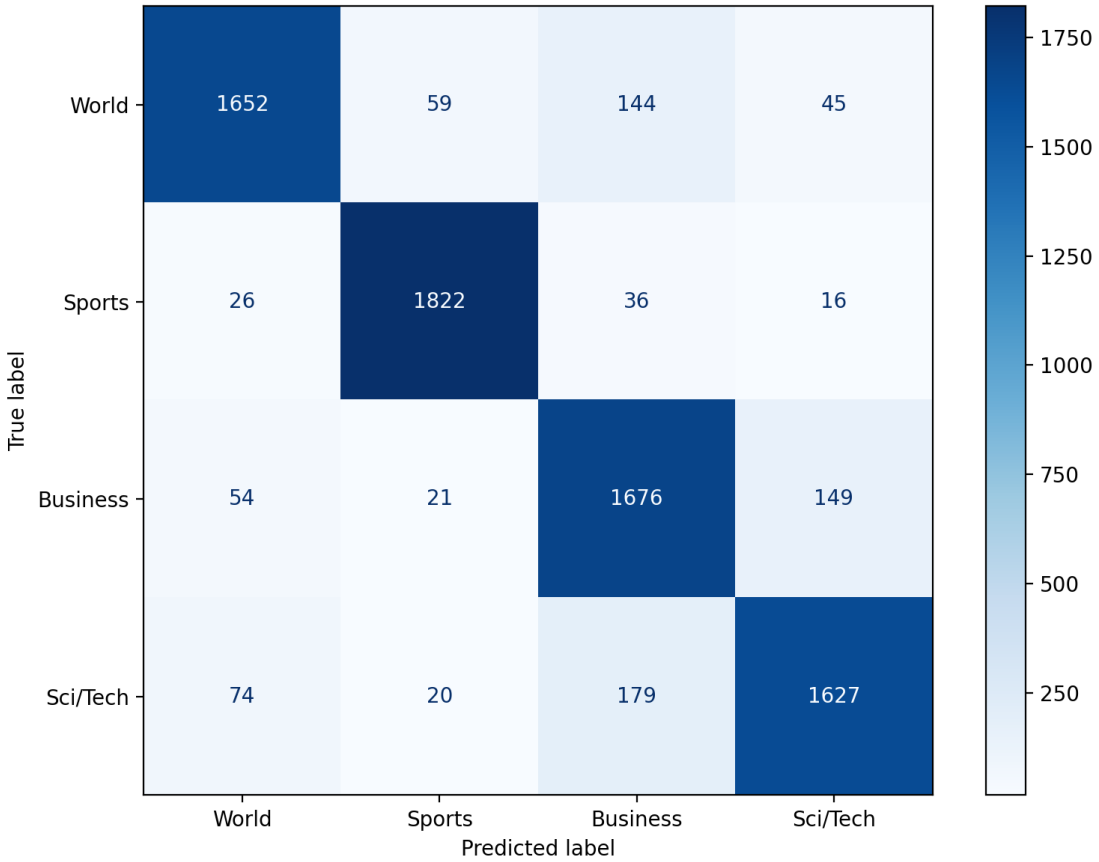
Resultados Deep – LSTM (AG_NEWS)

- Acurácia de teste: **0.8917**
- Precision macro: **0.8930**, Recall macro: **0.8917**, F1 macro: **0.8919**
- Margem menor vs. baselines: Logistic OvR 91.34% vs LSTM 89.17%





Resultados Deep – LSTM (AG_NEWS)



Comparação: Naive Bayes

Fashion MNIST:

Variant	Acc
Multinomial	65.55%
Bernoulli	64.82%
Gaussiano	59.10%

AG_NEWS:

Variant	Acc
Multinomial	89.66%
Bernoulli	89.36%
Gaussiano	86.64%

Observação: Multinomial $>$ Bernoulli $>$ Gaussiano em ambos datasets

Gap Generativo vs. Discriminativo

Dataset	Gap	Possível motivo
Fashion MNIST	**22.29%**	Pixels muito correlacionados
AG_NEWS	**1.68%**	TF-IDF gera vetor esparso

Gap = Melhor Discriminativo - Melhor Generativo
Fashion: RF (87.84%) - NB Multinomial (65.55%) = 22.29%
AG_NEWS: Logistic OvR (91.34%) - NB Multinomial (89.66%) = 1.68%

- A modalide dos dados pode ter influenciado.
- Features densas e correlacionadas como imagens violam suposição de independência do Naive Bayes
- Features esparsas e menos correlacionadas são mais adequadas para modelos generativos justamente por supor a independência

Modelos por Dataset

Fashion MNIST (Imagens):

1. Random Forest: 87.84%
2. Logistic OvR: 83.50%
3. Logistic Softmax: 83.40%
4. NB Multinomial: 65.55%
5. NB Bernoulli: 64.82%
6. NB Gaussiano: 59.10%

AG_NEWS (Texto):

1. Logistic OvR: 91.34%
2. Logistic Softmax: 91.24%
3. Random Forest: 90.96%
4. NB Multinomial: 89.66%
5. NB Bernoulli: 89.36%
6. NB Gaussiano: 86.64%

Problema no Paralelismo da Validação Cruzada

Problema: Random Forest no AG_NEWS consumiu **96 GB RAM**

Paralelismo da Validação Cruzada exigiu memória RAM demais

- GridSearchCV(n_jobs=-1): 16 workers
- Cada worker cria uma RandomForest, sendo que cada RandomForest também cria 16 workers
- Total: $16 \times 16 = 256$ processos paralelos
- Cada um dos 256 modelos copia os dados em memória

Solução: RandomForest(n_jobs=1), sem paralelismo interno

- GridSearchCV ainda paralelo

Estratégia de Otimização em Duas Fases

Tipo	Samples	CV	Tempo
Inviável	96k	3	8-12h
Exploração inicial	5k	2	10 min
Validação Cruzada final	30k	2	40 min

A estimativa cai de 8h-12h para 50min. A exploração inicial é útil para estimar quanto tempo será necessário para a validação cruzada final, além de identificar hiperparâmetros persistentes.

Conclusões Principais

1. Modalidade parece ter influenciado: Gap de 22.29% (imagens) vs 1.68% (texto)
2. Em ambos datasets, MultinomialNB $\approx$ BernoulliNB $>$ GaussianNB
3. Não-linearidade do Random Forest foi melhor para imagens. Classificação de texto, por ser esparso e de alta dimensionalidade, facilitaria a separabilidade no espaço com separadores lineares.
4. CNN (Fashion MNIST) supera Random Forest em 4.45%, foi capaz de generalizar bem e aprender os padrões.
5. LSTM (AG_NEWS) ficou 2.17% abaixo do Logistic OvR, mas competitivo. Ajustes podem ser feitos para tornar mais estável a convergência na minimização da função de custo.

Dúvidas?

Miguel Fernandes de Sousa
PEE/COPPE/UFRJ