UNIVERSIDADE FEDERAL DA PARAÍBA

RELATÓRIO
APRENDIZAGEM DE MÁQUINA

MARIA MARCOLINA CADOSO
INGRID DAYANE
PALOMA DUARTE

ESTRUTURA DO PROJETO:

- Projeto_Final_Educ.ipynb: Contém scripts e notebooks relacionados à análise e visualização dos dados educacionais, incluindo gráficos e relatórios descritivos dos modelos de aprendizagem de máquina.

- Projeto_Final_Enem.ipynb: Inclui os modelos de aprendizado de máquina desenvolvidos para predição de desempenho no Enem, bem como os experimentos realizados com diferentes algoritmos.

- modelos.py: módulo com os modelos DecisionTree, SVM e Redes Neurais

- Tratamentos de Dados:

- includingLabels.ipynb: Script responsável por adicionar rótulos e categorias aos dados brutos, facilitando a análise e modelagem.

- dataPB_Selection.ipynb: Realiza a seleção de dados específicos da Paraíba, filtrando informações relevantes para o estudo.

- data_cleaning.ipynb: Contém funções para limpeza e pré-processamento dos dados, como remoção de valores ausentes, normalização e transformação de variáveis.

- Analise Descritiva:

- AnaliseDescritiva.ipynb: scripts com análise geral para entendimento do dado.

DATASET E TRATAMENTO

Foram utilizados dois datasests relativos ao tema educação: um daset do Enem, do governo Federal, e um daset do desempenho de alunos da rede pública Estadual da Paraíba. Todo o projeto pode ser encontrado no github: https://github.com/mariaeco/MachineLearning/tree/main/FinalProject

Dados do Enem

Os dados do governo foram baixados da plataforma de microdados: https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados/enem. Foram baixados os dados de 2011 a 2023.

O dado do Enem contém, entre 50 a 100 variaváveis, que compõe dados de identificação, Informações demográficas, dados escolares, notas das provas, respostas e gabaritos, informações sobre redação e questões socioeconômicas

(Q001 a Q070). O dado não é completamente padronizado entre anos, então foi necessária uma longa padronização de colunas, nomes de variáveis, categorias.

Nós filtramos os dados apenas para o Estado da Paraíba, por ser um banco de dados de grande dimensão, e escolhemos as variáveis de interesse: faixa etária, dependência admnistrativa (federal, estadual, municipal, privado), escolaridade dos pais, renda familiar (em faixas de salários), número de pessoas em casa, acesso a computador, celular, internet, notas em cada área e na redação.

Escolhidas as variáveis, foram realizados os seguintes procedimentos:

- Padronização de nomes de colunas
- Padronização do valor das notas (em alguns anos estava entre 0 a 100 e outros de 0 a 1000)
- Tratamento de dados ausentes
- Padronização dos labels das variáveis categóricas e binárias
- Tratamento das questões socioeconômicas, pois a cada ano, incluiam novas questões e mudaram as numerações.

Este tratamento, de limpeza e padronização foi realizado basicamente para cada ano individualmente, já que os nomes das colunas, labels, tipos de variáveis, estão bastante despadronizados. Todo o tratamento pode ser encontrado em 3 notebooks, que devem ser utilzizados na ordem: 1) **dataPB_selection.ipynb**, **data_cleaning_standardization.ipynb** e **includingLabels.ipynb**.

Estes notebooks são fundamental para garantir a consistência e qualidade dos dados para análises posteriores, lidando com as diferentes estruturas e nomenclaturas ao longo dos anos do ENEM.

Dados da Educação Paraibana

Inicialmente, realizamos a filtragem dos dados para manter apenas os registros dos dados educacionais do último ano (2024). Devido ao grande volume de dados, selecionamos apenas as variáveis de interesse com comandos SELECT, incluindo: idade, porte da escola, se recebe bolsa família, série, frequência, desempenho, turno.

As etapas de tratamento dos dados em SQL incluíram:

- Padronização dos nomes das colunas: utilizando ALTER TABLE e RENAME COLUMN, harmonizando os diferentes nomes entre os anos.

- Tratamento de dados ausentes: substituímos valores nulos com UPDATE e CASE WHEN, ou eliminamos registros incompletos com DELETE.

- Padronização de labels em variáveis categóricas e binárias: utilizando CASE para recodificar as categorias de maneira uniforme entre diferentes anos.

Transformação das variáveis

Em ambos os datasets, a maioria das variáveis selecionadas são binárias, sem a necessidade de normalização ou padronização.

As variáveis categóricas com múltiplos labels, como dependência admnistrativa, modalidade, série, turno, foram transformadas em 0 e 1 pelo método **OneHotEncoder()**.

As variáveis inteiras como idade, número de pessoas em casa, e categóricas ordinais, como porte, escolaridade dos pais, renda, fora normalizadas, usando o método **MinMaxScaler()**, transformando-os para uma escala entre 0 e 1, seguindo a fórmula: X_norm = (X - X_min) / (X_max - X_min)

ANÁLISES

ÁRVORE DE DECISÃO

Foi implementado um modelo de Árvore de Decisão com Poda Otimizada para o ajuste e treinamento dos dados. O processo teve como objetivo principal evitar o overfitting através da técnica de poda baseada no custo-complexidade (Minimal Cost-Complexity Pruning).

Inicialmente, uma árvore de decisão (DecisionTreeClassifier) foi treinada sem restrição de complexidade para calcular o caminho de poda, utilizando o método cost_complexity_pruning_path. A partir desse caminho, foram extraídos os valores de alpha (ccp_alphas), que controlam o grau de poda da árvore. Quando a quantidade de valores de alpha era superior a 20, foi realizada uma amostragem uniforme para limitar o número de alphas avaliados, mantendo o foco nos intervalos mais relevantes e reduzindo o custo computacional. Também foram removidos valores de alpha extremamente pequenos (inferiores a 1e-10), pois representariam podas insignificantes.

Posteriormente, foi realizada uma busca em grade (GridSearchCV) para encontrar o melhor valor de alpha. O processo de busca avaliou diferentes modelos, treinados com diferentes níveis de poda, utilizando validação cruzada com 5 folds (cv=5). O parâmetro n_jobs=-1 foi utilizado para paralelizar a execução e acelerar o processo

de ajuste. Durante todo o processo, o parâmetro random_state=0 foi fixado para garantir a reprodutibilidade dos resultados.

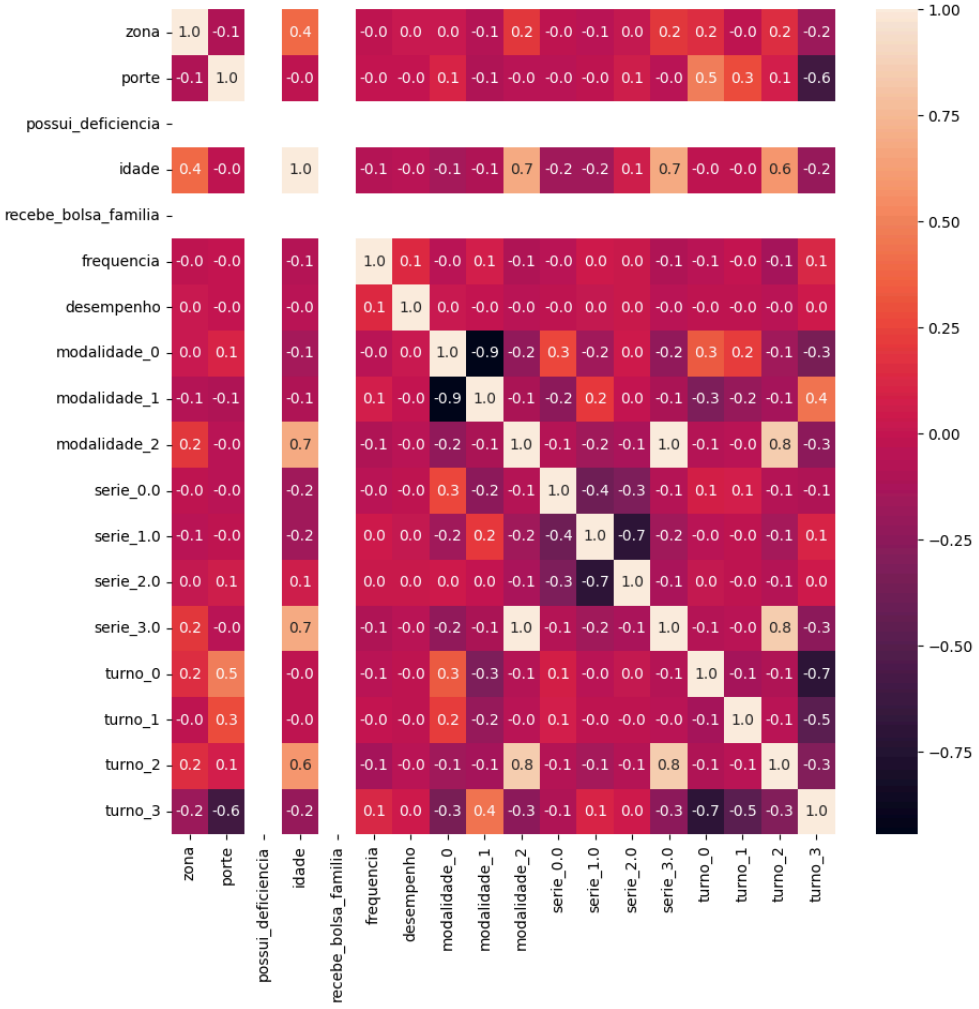

Figura 1: Matriz de Correlação entre as variáveis dos dados da Educação da Paraíba.

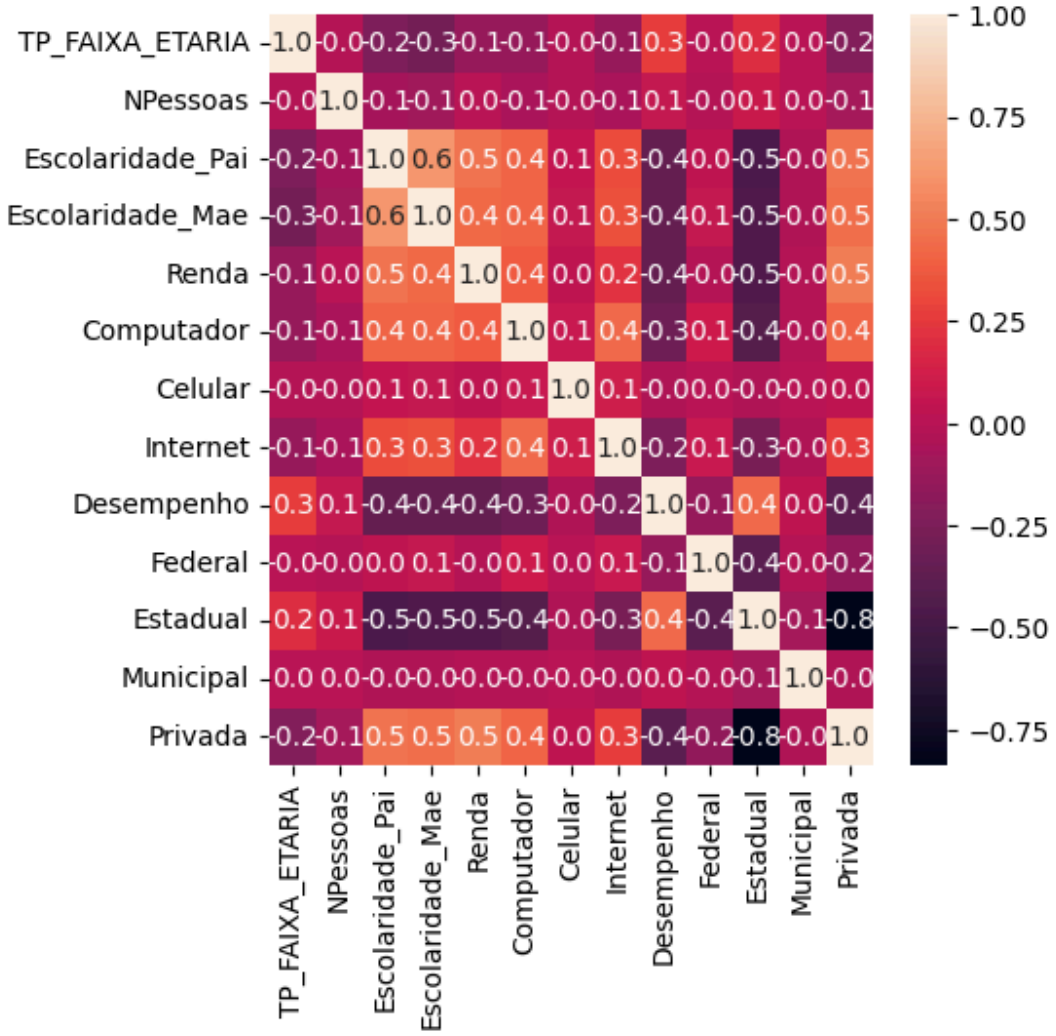

Figura 2: Matriz de Correlação entre as variáveis dos dados do ENEM (B).

Tabela 1: Resultados das medidas de Avaliação do Modelo de Árvore de Decisão para os dados da Educação da Paraíba.

```
Árvore de Decisão Otimizada
Melhor alpha: 0.0005989468708161217
Ein: 0.0859
Eout: 0.0843

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.64      0.03      0.05       341
           1       0.92      1.00      0.96      3659

    accuracy                           0.92      4000
   macro avg       0.78      0.51      0.50      4000
weighted avg       0.89      0.92      0.88      4000

A profundidade da árvore para o melhor alpha é: 2
```

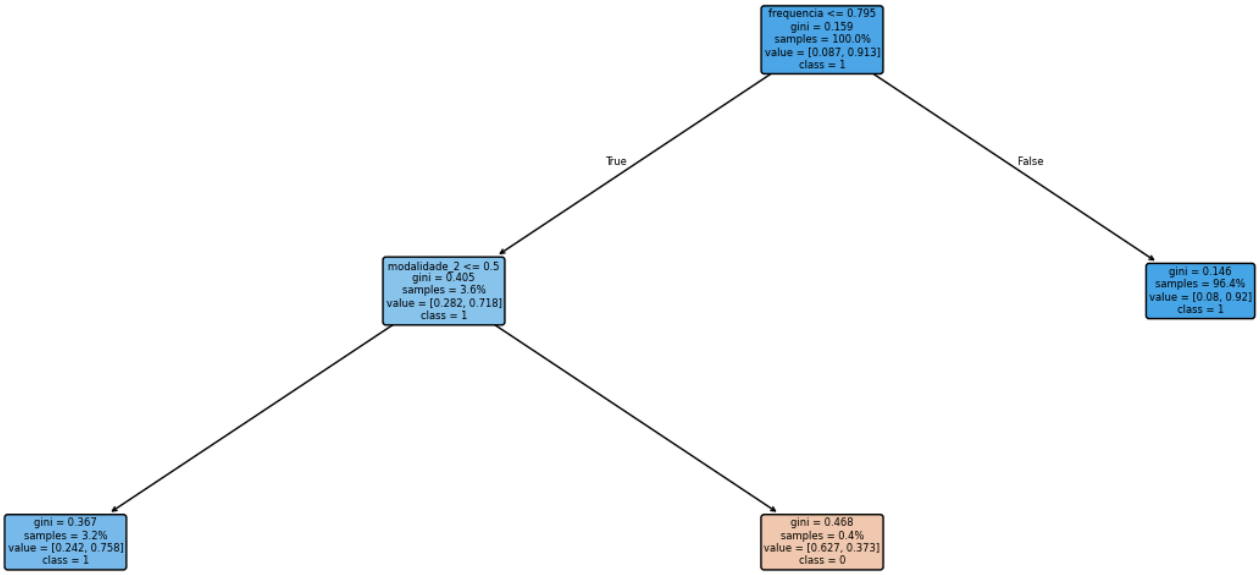

Figura 3: Árvore de decisão para os dados da Educação da Paraíba

Tabela 1: Resultados das medidas de Avaliação do Modelo de Árvore de Decisão para os dados do Enem.

Árvore de Decisão Otimizada
Melhor alpha: 0.0020150466534824114
Ein: 0.3831
Eout: 0.3886

Relatório de Classificação:

	precision	recall	f1-score	support
0	0.50	0.38	0.43	4071
1	0.63	0.80	0.71	17396
2	0.58	0.33	0.42	8520
accuracy			0.61	29987
macro avg	0.57	0.50	0.52	29987
weighted avg	0.60	0.61	0.59	29987

A profundidade da árvore para o melhor alpha é: 3

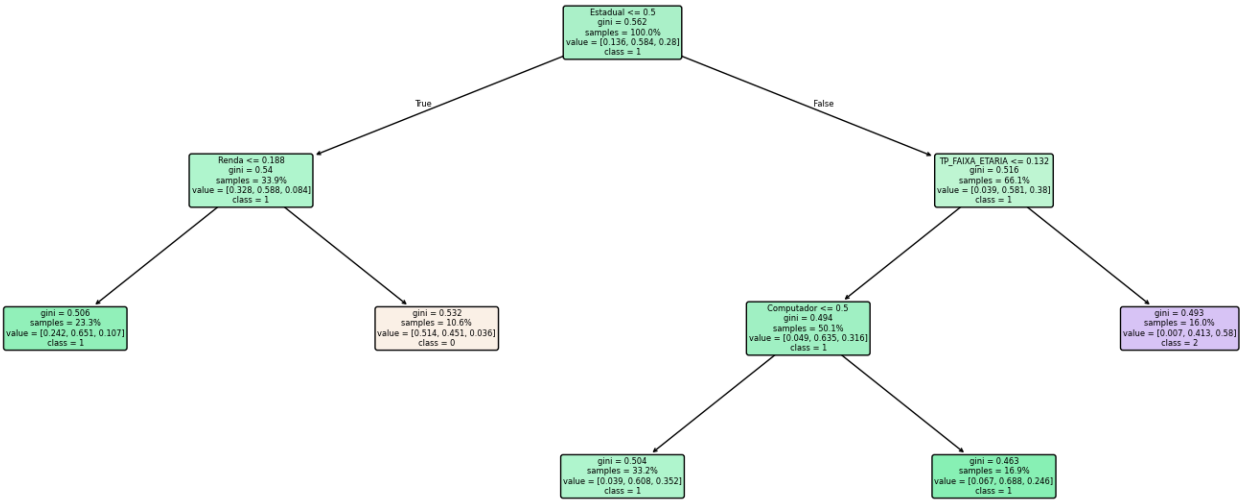

Figura 4: Árvore de decisão para os dados do Enem

SVM

Foi implementado um modelo de Máquinas de Vetores de Suporte (Support Vector Machine, SVM), utilizando kernel do tipo radial basis function (RBF), que é muito eficiente para separar classes que não são linearmente separáveis no espaço original dos dados.

O modelo foi ajustado com a utilização do GridSearchCV, que realizou uma busca exaustiva para encontrar os melhores hiperparâmetros. No caso, foram testados os valores de C e gamma, onde C controla o equilíbrio entre a maximização da margem e a minimização do erro de classificação, enquanto gamma define a influência de um único exemplo de treino sobre a formação da fronteira de decisão. Para este experimento, o grid de busca considerou C=10 e gamma=0.1, realizando validação cruzada para garantir uma avaliação robusta.

Tabela 3: Resultados das medidas de Avaliação do Modelo SVM para os dados da Educação da Paraíba.

```
SVM - Resultados
Melhores parâmetros: {'C': 10, 'gamma': 0.1}
Erro de treino (Ein): 0.0867
Erro de teste (Eout): 0.0850
Número total de vetores de suporte: 3074

Relatório de Classificação:
              precision    recall  f1-score   support
```

0	**1.00**	**0.00**	**0.01**	**341**
1	**0.91**	**1.00**	**0.96**	**3659**
accuracy			**0.92**	**4000**
macro avg	**0.96**	**0.50**	**0.48**	**4000**
weighted avg	**0.92**	**0.92**	**0.87**	**4000**

Tabela 4: Resultados das medidas de Avaliação do Modelo SVM para os dados da Educação da Paraíba.

SVM - Resultados
Melhores parâmetros: {'C': 10, 'gamma': 0.1}
Erro de treino (Ein): 0.3863
Erro de teste (Eout): 0.3919
Número total de vetores de suporte: 18745

Relatório de Classificação:

	precision	recall	f1-score	support
0	0.62	0.45	0.52	1419
1	0.61	0.88	0.72	3412
2	0.64	0.03	0.06	1252
accuracy			0.61	6083
macro avg	0.62	0.46	0.43	6083
weighted avg	0.61	0.61	0.54	6083

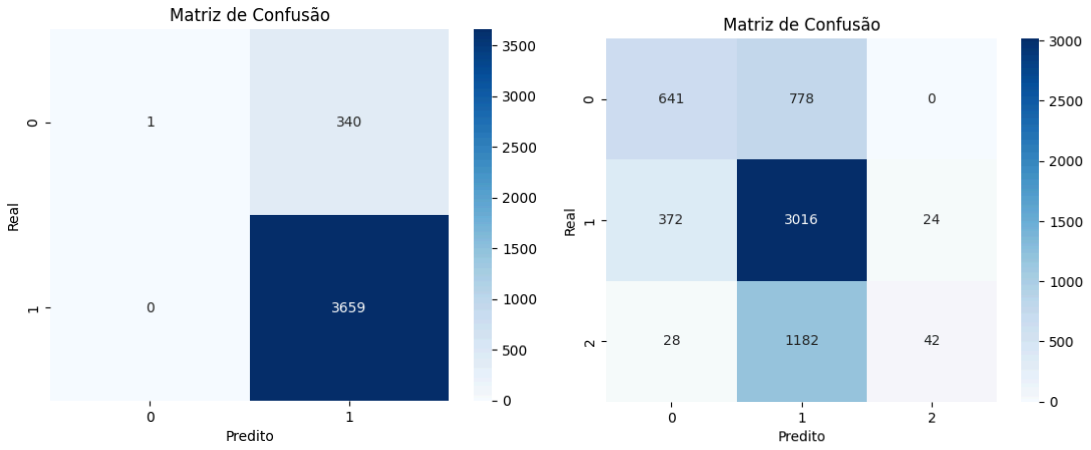

Figura 5: Loss, Acurácia e Matriz de confusão para os dados da educação da PB e do Enem.

REDE NEURAL

O modelo de rede neural implementado consiste em um classificador multicamadas (MLP), para resolver um problema de classificação multiclasse. Sua arquitetura é composta por uma camada de entrada, cuja dimensão é definida pelo

número de variáveis preditoras (X_train.shape[1]), com um número de neurônios parametrizável (n_neurons) e função de ativação ReLU. Em seguida, há uma camada intermediária contendo n neurônios, também utilizando a função de ativação ReLU. A camada de saída possui um número de neurônios correspondente ao número de classes (output_dim) e utiliza a função de ativação Softmax, adequada para problemas de classificação com múltiplas categorias. Os pesos da rede foram inicializados a partir de uma distribuição normal.

O cálculo do número de neurônios foi baseado em uma heurística para determinar o número máximo de neurônios na camada oculta de uma rede neural, considerando a prevenção de overfitting.

Fórmula: (N-10)/(10*(alfa+2))
Onde:

N = número de exemplos de treinamento
α = número de features/variáveis de entrada
10 = fator de escala para prevenir overfitting
2 = termo adicional para considerar o bias
Explicação dos Componentes:

Para o treinamento, foi utilizado o otimizador Adam, com uma taxa de aprendizado de 0,001, e a função de perda escolhida foi a sparse_categorical_crossentropy, apropriada para classificação multiclasse com rótulos inteiros. Como métrica de avaliação, foi adotada a acurácia. Com o objetivo de evitar overfitting, foi implementado o mecanismo de Early Stopping, monitorando a perda de validação (val_loss), com paciência de 10 épocas e restauração dos melhores pesos obtidos durante o treinamento.

O treinamento do modelo foi realizado com um número máximo de 100 épocas, utilizando batches de 256 amostras e com 20% dos dados de treinamento reservados para validação. Ao longo do processo, foram monitorados e calculados o erro de treino (Ein), o erro de teste (Eout), a acurácia de treino e de teste, bem como o valor da função de perda em ambos os conjuntos. Para análise dos resultados, foram gerados gráficos de evolução da Loss e da Acurácia ao longo das épocas (tanto para treino quanto para validação), além da matriz de confusão e de um relatório de classificação completo, contendo métricas como precisão, recall e F1-score para cada classe.

Tabela 5: Resultados das medidas de Avaliação do Modelo de Redes Neurais para os dados da Educação da Paraíba.

Métricas de Avaliação:

```
Acurácia de Treino: 0.9129
Acurácia de Teste: 0.9147
Ein: 0.0871
Eout: 0.0853
625/625 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 766us/step

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.68      0.04      0.07      1740
           1       0.92      1.00      0.96     18260

    accuracy                           0.91     20000
   macro avg       0.80      0.52      0.51     20000
weighted avg       0.90      0.91      0.88     20000
```

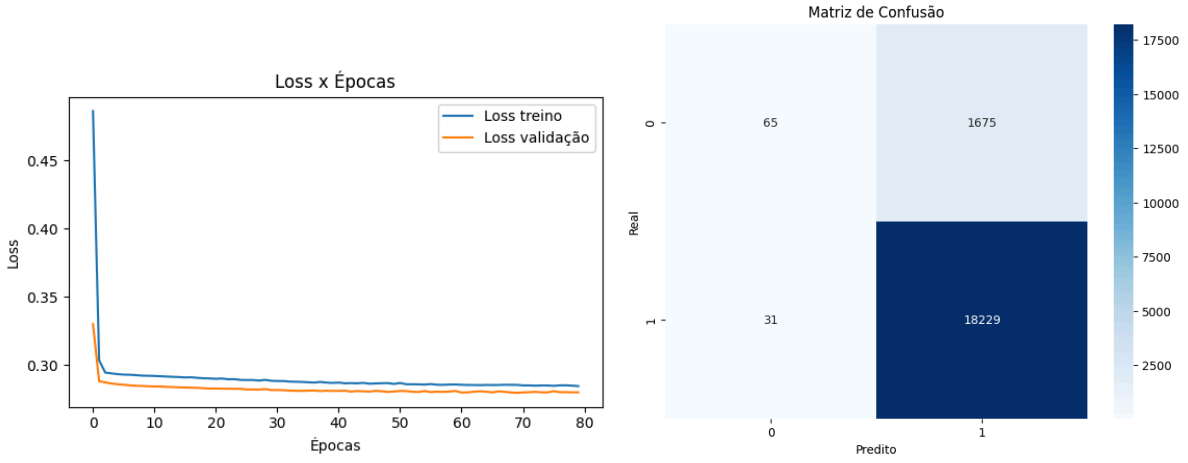

Figura 6: Loss, Acurácia e Matriz de confusão para os dados do Enem

Tabela 6: Resultados das medidas de Avaliação do Modelo de Redes Neurais para os dados do Enem.

```
Métricas de Avaliação:
Acurácia de Treino: 0.6272
Acurácia de Teste: 0.6215
Ein: 0.3728
Eout: 0.3785

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.59      0.26      0.36      4071
           1       0.63      0.86      0.73     17396
           2       0.60      0.30      0.40      8520
```

accuracy			0.62	29987
macro avg	0.61	0.48	0.50	29987
weighted avg	0.61	0.62	0.59	29987

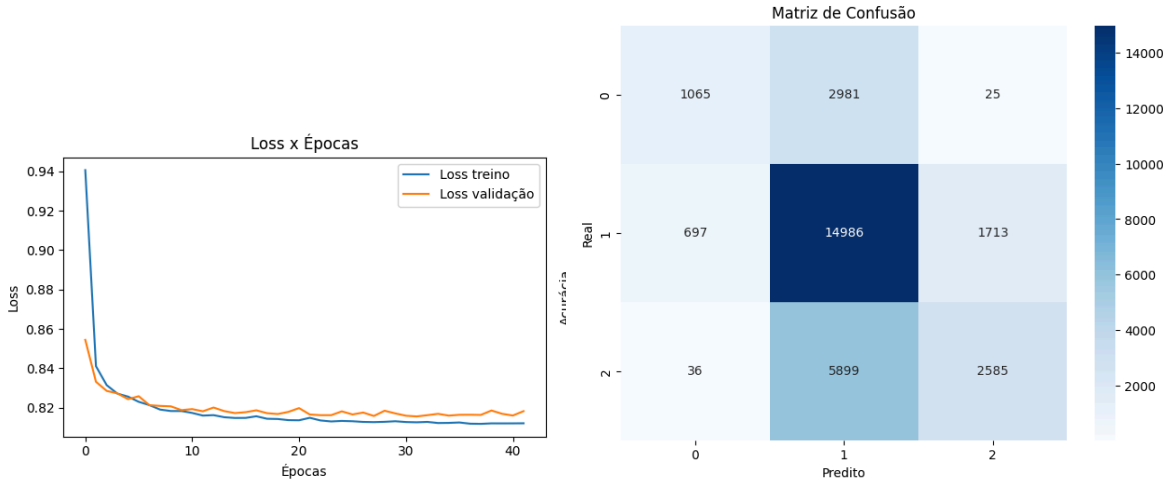

Figura 7: Loss e Matriz de confusão para os dados do Enem

Conclusão:

A diferença de classificação dos 3 modelos para os dados do Enem foi pouca, tendo uma acurácia entre 50 e 65% dos labels.

Os dados da educação da Paraíba, embora tenham resultado em uma acuracia geral maior (aproximadamente 90%), observamos uma grande diferença na acurácia dos labels no modelo SVM para os modelos de Rede Neural e Árvore de Decisão.