Equipe: Antônio Gabriel dos Santos Clemente - agsc

Jose Izaias da Silva Junior - jisj

Leandro Luiz de Lima Freitas - lllf

Tomás Nascimento Santos - tns

19 de fevereiro de 2025

Analisando a Base de Dados SpamBase

A base de dados Spambase é um conjunto de dados amplamente utilizado para tarefas de classificação de spam. Ela contém 4.601 instâncias (emails), sendo 1.813 classificados como spam (39,4%) e 2.788 como não spam (60,6%).

Cada instância é descrita por 58 atributos, sendo:

- 48 atributos contínuos:

Representam a frequência de palavras ou caracteres específicos no email. Esses atributos são expressos como porcentagens do total de palavras no email. Exemplos incluem frequências de palavras como "free", "money", "george" (nome comum em emails não spam), etc.

- 6 atributos contínuos:

  Representam a frequência de caracteres especiais, como pontos de exclamação (!), cifrões ($), hífens (-), etc.

- 3 atributos contínuos:

  Relacionados ao comprimento médio de sequências de letras maiúsculas e o comprimento total de letras maiúsculas no email.

- 1 atributo categórico:

  Representa a classe (target), onde:

  1 indica que o email é spam.

  0 indica que o email não é spam.

Resumo Estatístico: Número de instâncias: 4.601

Número de atributos: 58 (57 contínuos + 1 categórico)

- Distribuição das classes:

  Spam: 1.813 (39,4%)

  Não spam: 2.788 (60,6%)

ÁRVORE DE DECISÃO

Divisão dos Dados

Os dados foram divididos em conjuntos de treino e teste com a seguinte configuração:

- 70% para treinamento
- 30% para teste
- Amostragem estratificada para manter a distribuição das classes.

Ajuste de Hiperparâmetros

Para encontrar a melhor configuração da árvore de decisão, foi utilizada a técnica GridSearchCV com validação cruzada de 5 folds. Os hiperparâmetros testados foram:

- max_depth: [None, 5, 10, 20, 30, 40, 50]
- min_samples_split: [2, 5, 10]
- min_samples_leaf: [1, 2, 4]

Os melhores hiperparâmetros encontrados foram:

max_depth = 20

min_samples_split = 5

min_samples_leaf = 2

Resultados Obtidos

Métrica	Valor
Precisão	0.9119
Recall	0.8750
F1-Score	0.8931

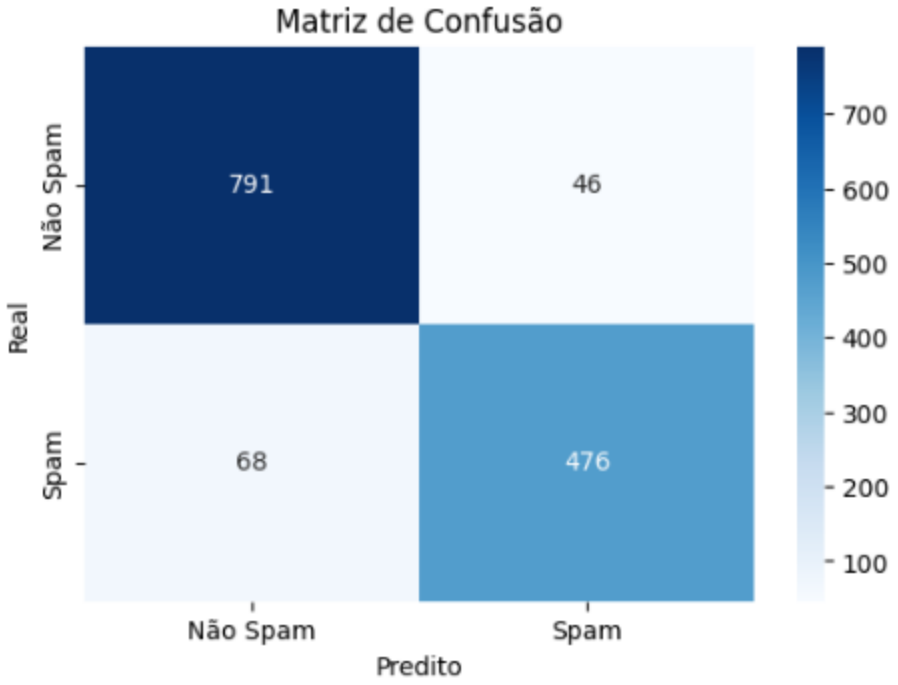
Importância das Features

A análise de importância das features mostrou que os atributos mais relevantes para a classificação foram:

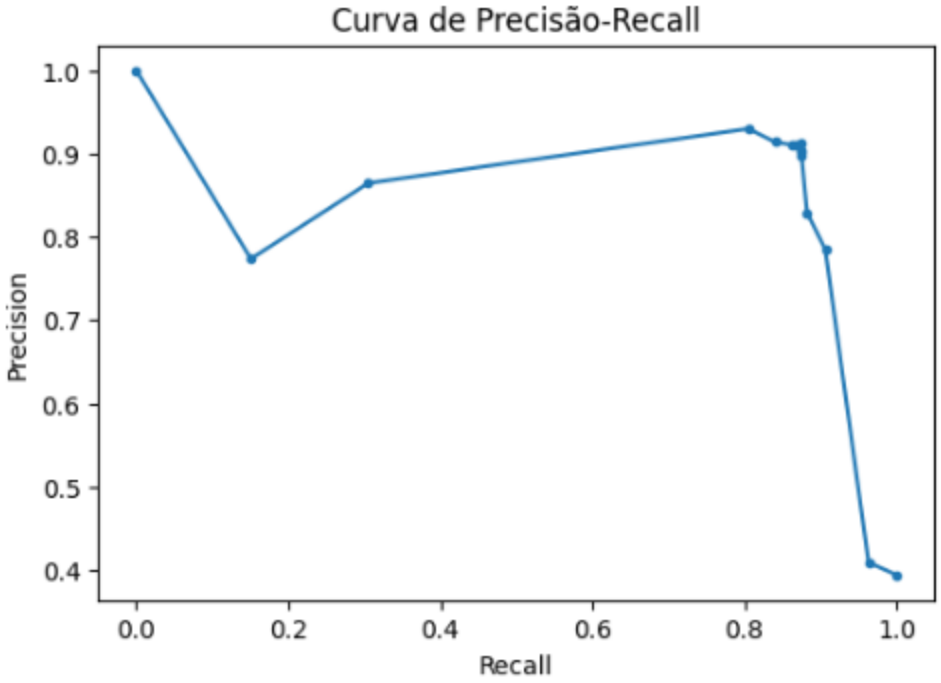
1. char_freq_$
2. word_freq_remove
3. capital_run_length_total

Visualização Gráfica

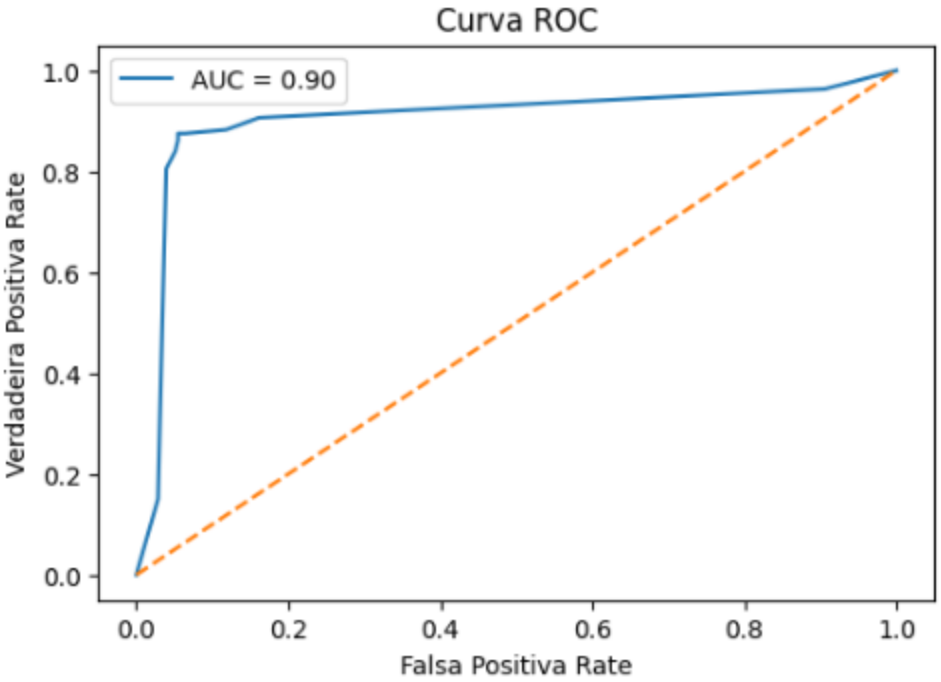
- Matriz de Confusão: Indicou uma boa separação entre classes, com poucos falsos positivos e falsos negativos.



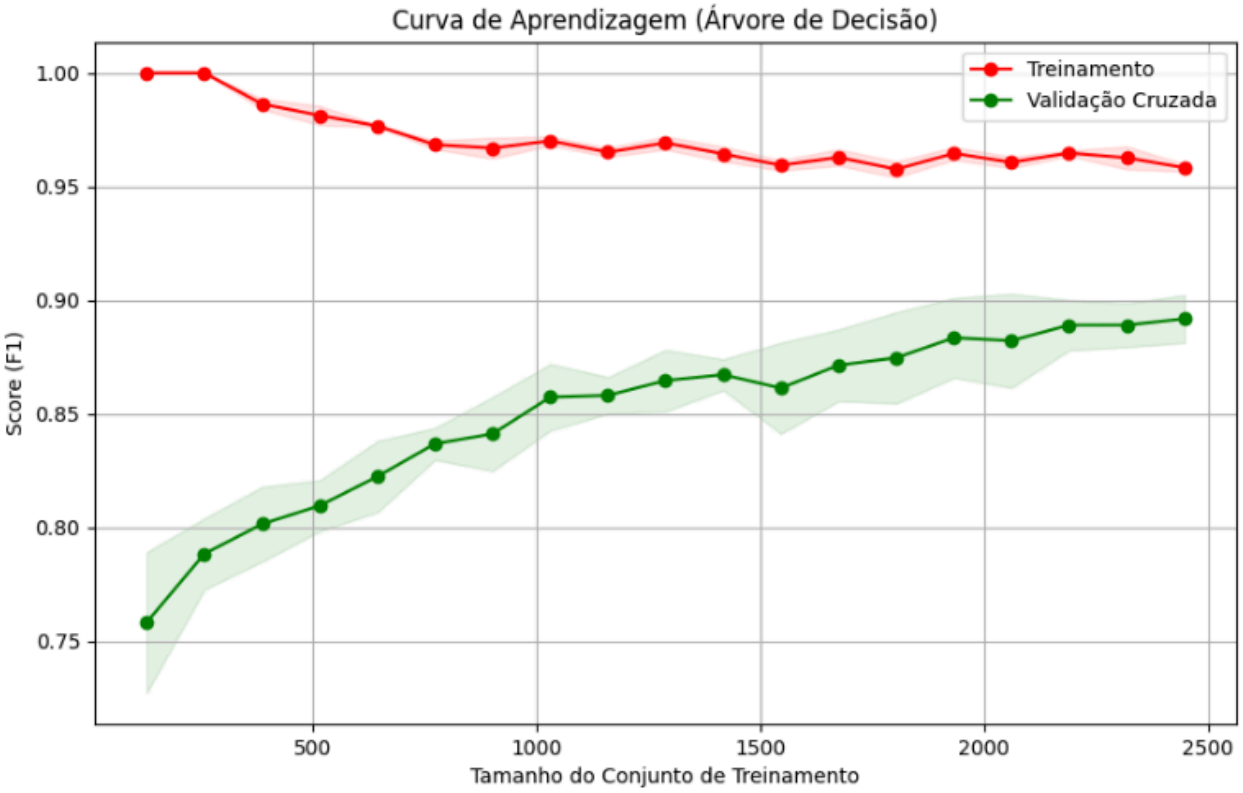
- Curva de Precisão-Recall: Demonstrou bom equilíbrio entre precisão e revocação.



- Curva ROC: Mostrou um AUC elevado, indicando um bom desempenho do classificador.



- Curva de Aprendizagem: Indicou um bom ajuste, sem overfitting significativo.



O modelo de Árvore de Decisão apresentou um desempenho sólido na classificação de e-mails como spam, alcançando uma **acurácia de 92%**. A otimização dos hiperparâmetros contribuiu para melhorar a capacidade de generalização do modelo.

BAYESIANO INGÊNUO

O modelo **Naive Bayes** foi utilizado para a classificação, assumindo a independência condicional entre as variáveis. Como o algoritmo requer um ajuste fino do hiperparâmetro **var_smoothing**, que controla a suavização da variância para evitar probabilidades zero, um **GridSearchCV** foi aplicado para encontrar o melhor valor, usando **Validação Cruzada (cv=5)**, variando o hiperparâmetro `var_smoothing` em uma escala logarítmica. O objetivo foi maximizar o **F1-Score**.

O melhor valor encontrado foi:

- **var_smoothing = 1e-06**

Treinamento e Avaliação

Após a escolha do hiperparâmetro ótimo, o modelo foi treinado no conjunto de treinamento completo e avaliado no conjunto de teste. As métricas reportadas incluem **Precisão (Precision), Revocação (Recall) e F1-Score**.

Métricas no Conjunto de Teste

- **Precision**: 0.8030
- **Recall**: 0.8989
- **F1-Score**: 0.8482

O relatório de classificação detalhado apresenta os seguintes resultados:

Classe	Precision	Recall	F1-Score	Suporte
0 (NEG)	0.93	0.86	0.89	837
1 (POS)	0.80	0.90	0.85	544

- A **classe 0** (NEG) teve **alta precisão (0.93)**, indicando que poucos exemplos negativos foram classificados incorretamente como positivos.
- A **classe 1** (POS) teve um **recall elevado (0.90)**, demonstrando que o modelo conseguiu identificar a maioria dos exemplos positivos corretamente.

A acurácia geral do modelo foi **0.87**.

Curvas de Aprendizagem

A curva de aprendizagem do **Naive Bayes** foi analisada para diferentes tamanhos do conjunto de treinamento.

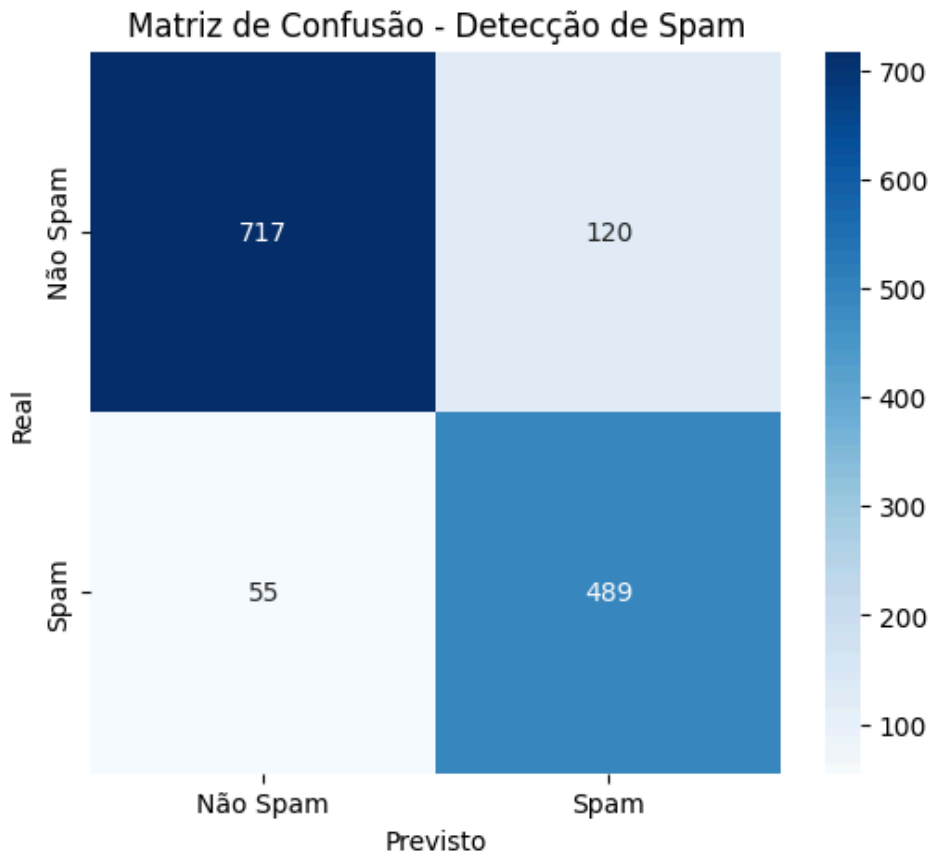
Observações:

- Para **pequenos conjuntos de treinamento**, há uma grande variação nas métricas, sugerindo **alta variabilidade** nos resultados iniciais.
- À medida que o tamanho do conjunto de treinamento aumenta, os scores de **treinamento e validação convergem**, indicando que o modelo melhora sua generalização.

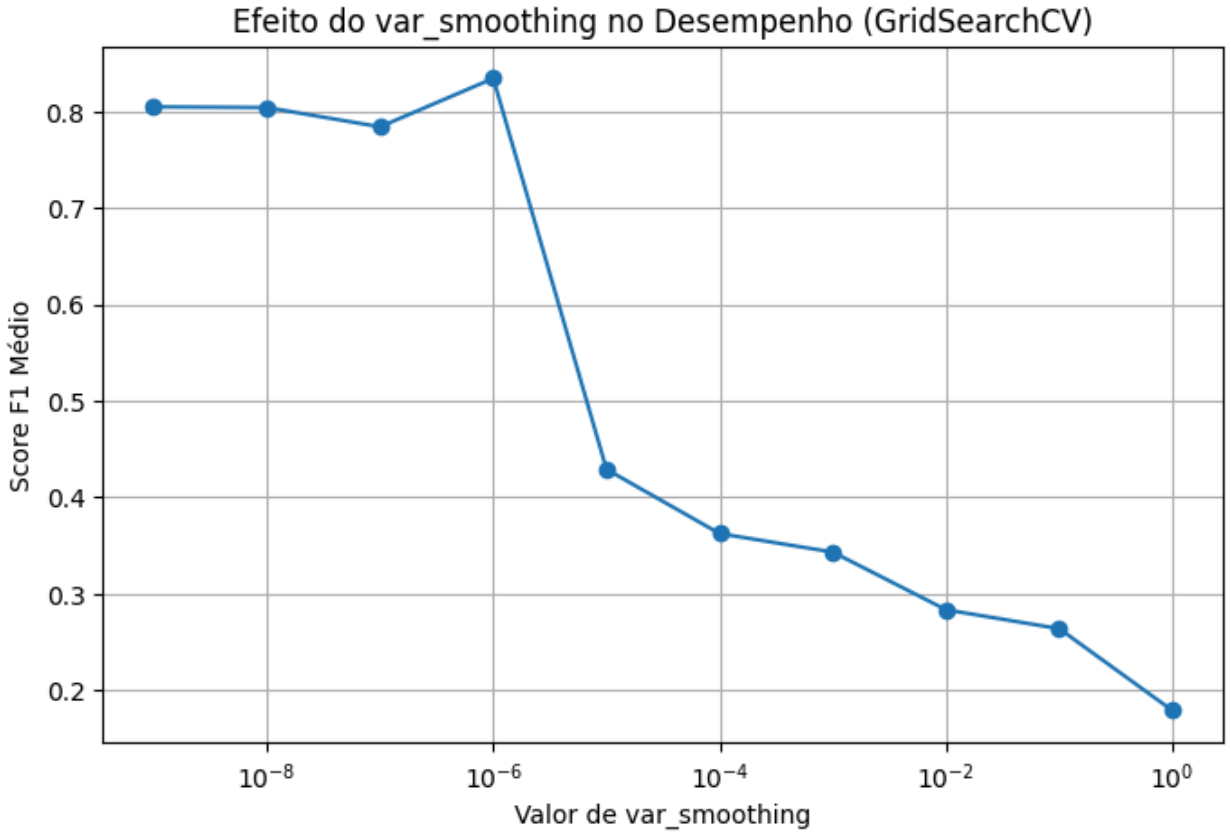
- O modelo atinge **estabilidade a partir de aproximadamente 1500 amostras**, onde **as métricas de validação cruzada se aproximam das métricas de treinamento**.
- Como esperado, o **Naive Bayes tende a ser mais eficaz em grandes conjuntos de dados**, pois se baseia na suposição de independência entre as variáveis.

Visualização Gráfica

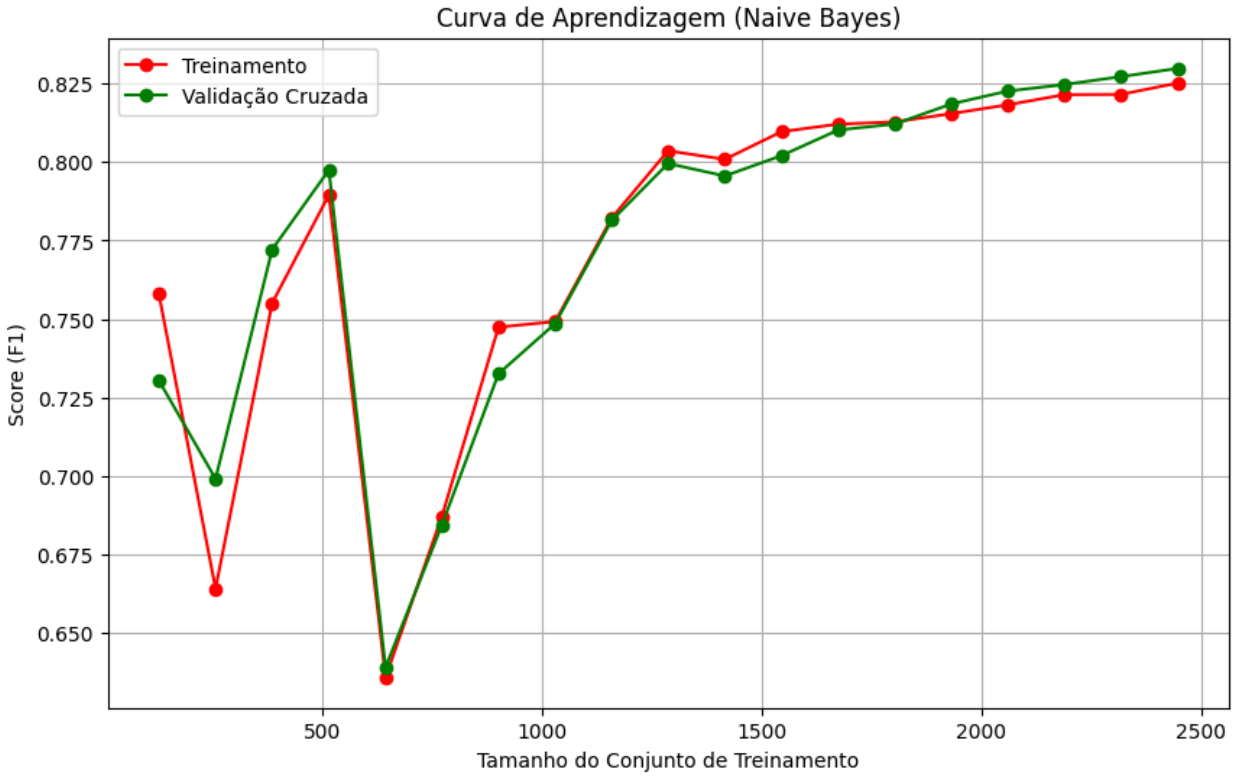
- **Matriz de confusão:** Retornou uma boa separação entre classes, com baixa incidência de falsos positivos e falsos negativos.



- **Mudança de hiperparâmetro:** Fica visível o quão benéfico foi usar o melhor valor possível para o `var_smoothing` para otimizar a aprendizagem do modelo



- **Curva de aprendizagem:** Ajuste e estabilização compatíveis com o esperado



REGRESSÃO LOGÍSTICA

De início, foi treinado um modelo com os parâmetros *default* para comparação futura com as outras estratégias de treinamento. Dessa forma, os resultados surpreendentemente foram bastante positivos:

Métricas no Conjunto de Teste:

- ***Precision*: 0.919**
- ***Recall*: 0.897**
- ***F1 Score*: 0.908**

Após isso, foram treinados outros modelos adotando outras estratégias de tratamento prévio de dados e exploração de diferentes parâmetros:

1. Utilização de métricas de alta correlação entre features visando eliminação de colunas para simplificação do treinamento do modelo

Foi utilizado o **coeficiente de correlação de Kendall** para mensurar a correlação entre colunas.

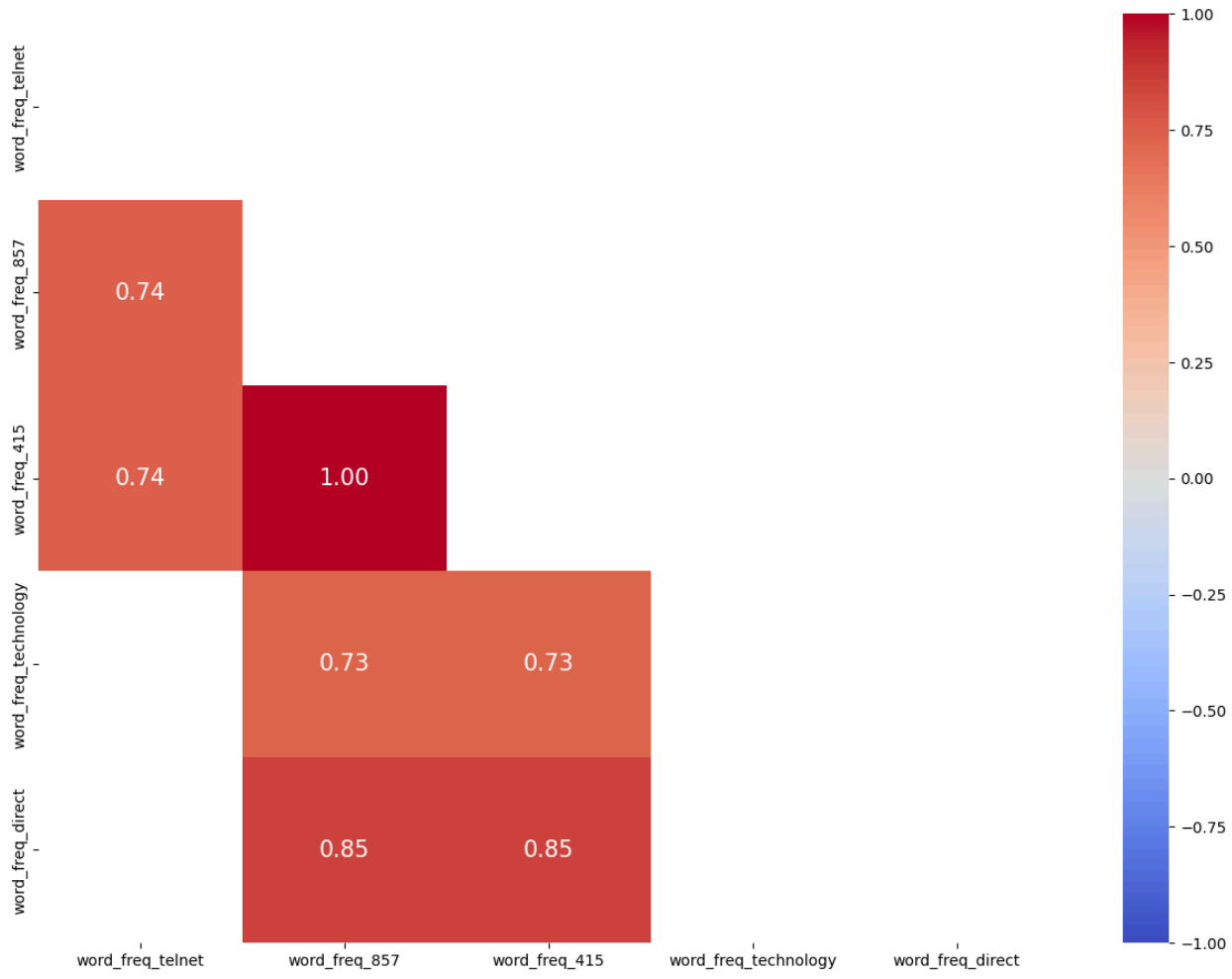
$$\tau = \frac{\{Número\ de\ pares\ concordantes\} - \{Número\ de\ pares\ discordantes\}}{\{Total\ de\ pares\}}$$

Nesse contexto, temos as seguintes definições:

Seja um par de observações (Xi, Yi) e (Xj, Yj), eles são ditos concordantes se

- Quando Xi > Xj, Yi > Yj
- Qando Xi < Xj, Yi < Yj

Quando tal relação não se mostra presente para um par de observações, ele é dito como discordante. Com tal métrica em mãos, buscamos aquelas variáveis com níveis de correlação acima de 70%.



Após excluir 4 das 5 variáveis acima citadas, os resultados do treinamento na base de teste foram os seguintes:

- **Precision: 0.919**
- **Recall: 0.897**
- **F1 Score: 0.908**

Note que **não houve** alteração significativa em comparação com o modelo anterior.

2. Normalização dos dados antes de alimentar o modelo

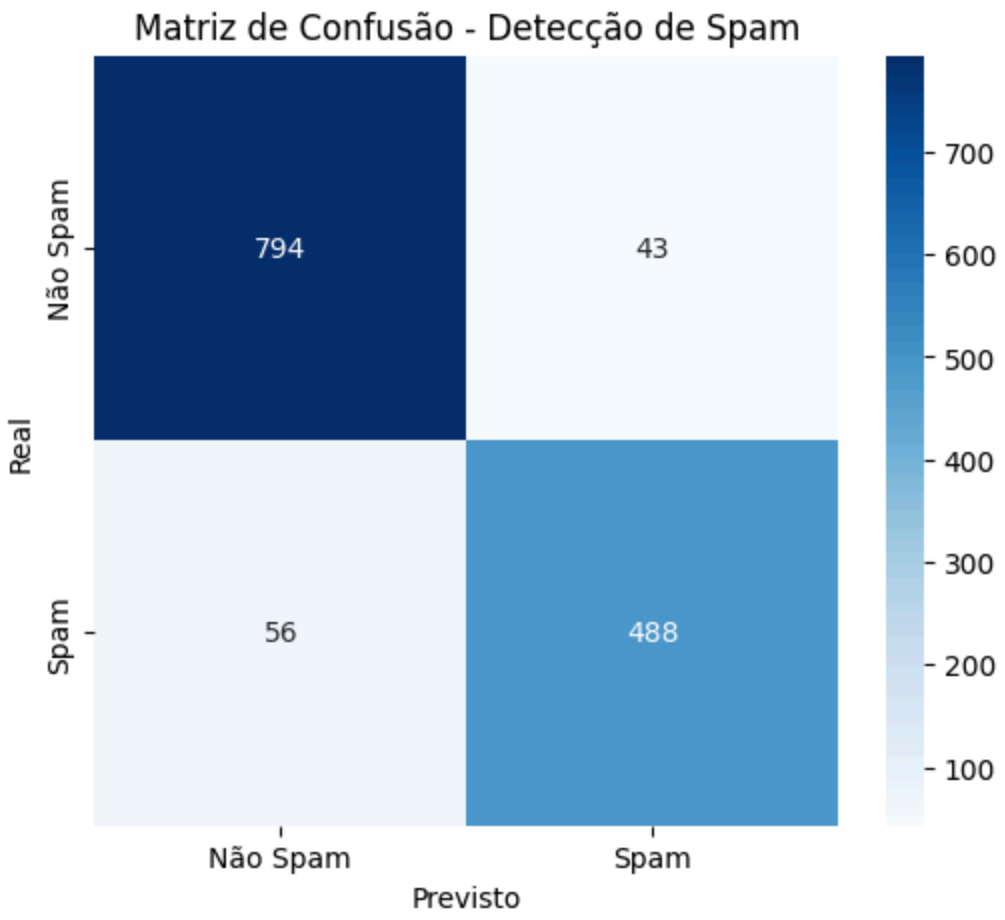
Usando o método *StandardScaler* da biblioteca scikit-learn, buscamos normalizar os dados usados no treinamento do modelo de regressão. Assim foram os resultados:

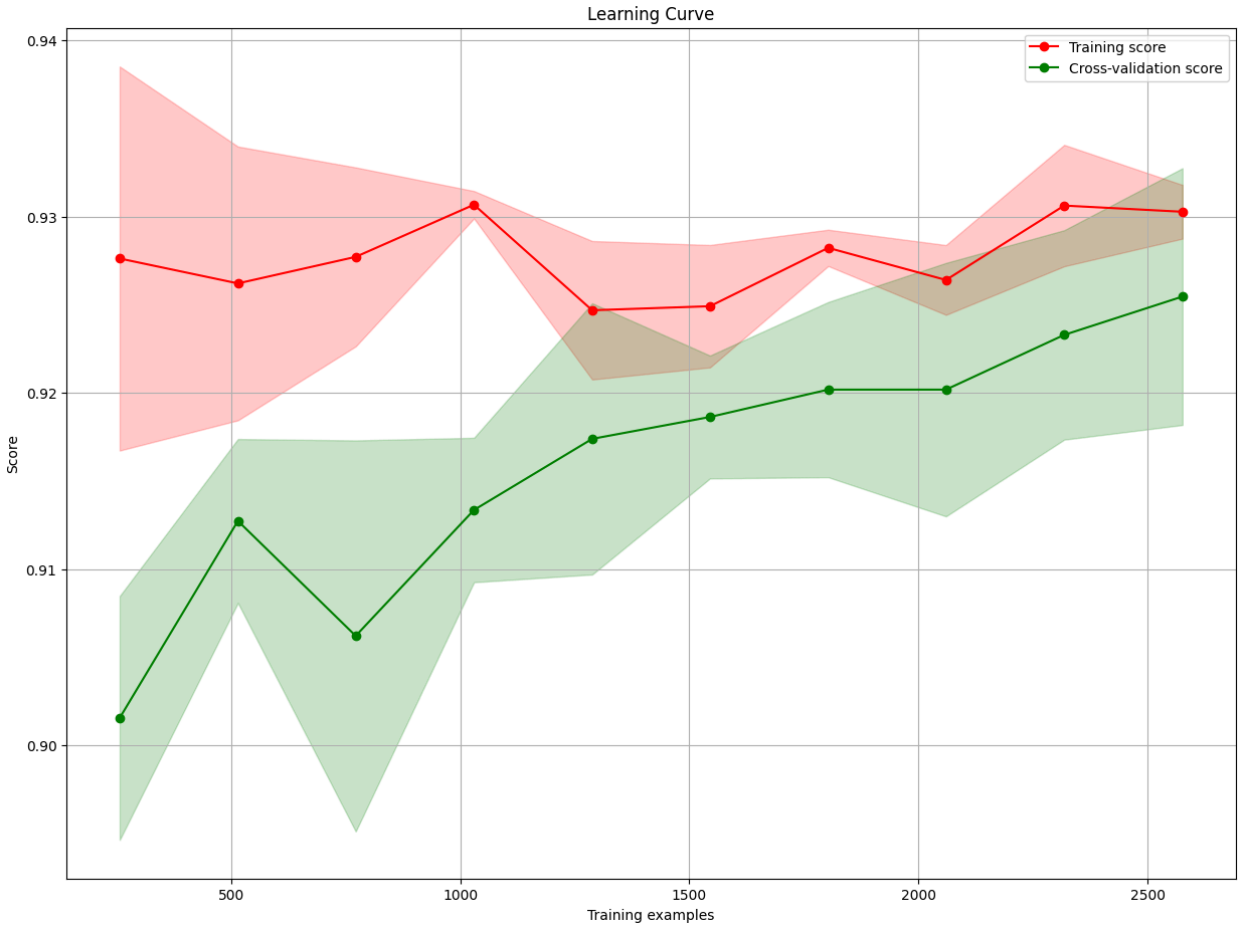
- **Precision: 0.922**
- **Recall: 0.893**
- **F1 Score: 0.908**

3. Busca de parâmetros

Realizando variações nos parâmetros de treinamento “penalty” e “solver” no modelo de regressão linear da biblioteca *scikit-learn*, Foram obtidos resultados **extremamente** semelhantes aos do primeiro modelo treinado.

- **Gráficos de resultados e aprendizagem do primeiro modelo**


Matriz de Confusão - Detecção de Spam
Não Spam
Spam
Real
794
43
56
488
Não Spam
Spam
Previsto
700
600
500
400
300
200
100



K-VIZINHOS

Foi criado um pipeline que primeiro aplica o **Standard Scaler** para normalizar as variáveis, essencial em métodos baseados em distância como o KNN. Em seguida, o **K Neighbors Classifier** é configurado para usar a distância Euclidiana (**p=2**) e pesos proporcionais à distância (**weights='distance'**), fazendo com que vizinhos mais próximos tenham maior influência na decisão.

Por meio do **Grid Search CV**, variamos o número de vizinhos (**n_neighbors**) de 1 a 50, utilizando validação cruzada com 10 *folds* e o *f1-score* como métrica de avaliação. O melhor valor encontrado para **n_neighbors** foi **17** e escolhido para compor o modelo final.

Após selecionar os melhores hiperparâmetros, o modelo é treinado com o conjunto de treinamento completo e testado no conjunto de teste. As métricas reportadas incluem *precision*, *recall*, *f1-score* e o relatório de classificação.

Métricas no Conjunto de Teste:

- *Precision*: 0.901
- *Recall*: 0.884
- *F1 Score*: 0.892

O relatório de classificação mostra que a classe 0 (não spam) obteve uma *precision* de 0.93 e *recall* de 0.94, enquanto a classe 1 (spam) obteve 0.90 de *precision* e 0.88 de *recall*. A acurácia total foi de aproximadamente 0.92.

Curvas de Aprendizagem:

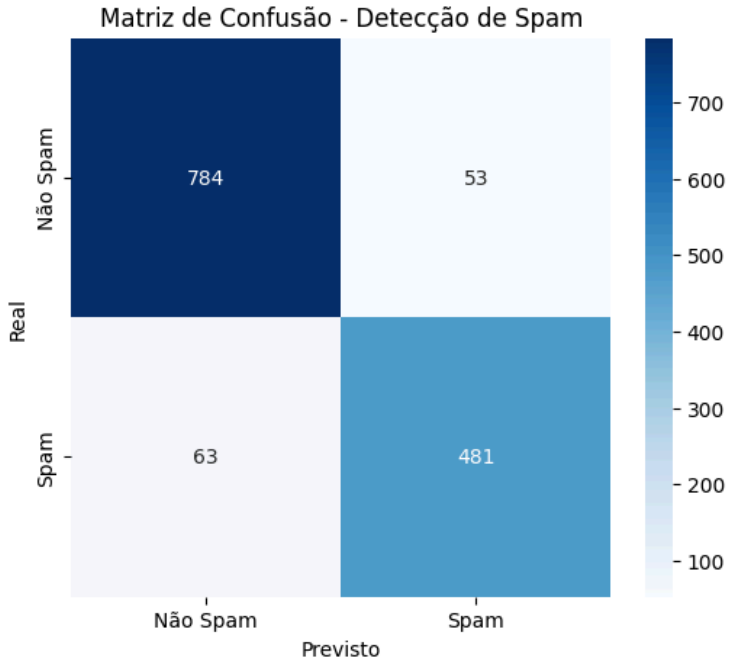
Observa-se que:

- As métricas (precision, recall, F1) começam altas, pois o modelo tende a se ajustar bem a um volume menor de dados.

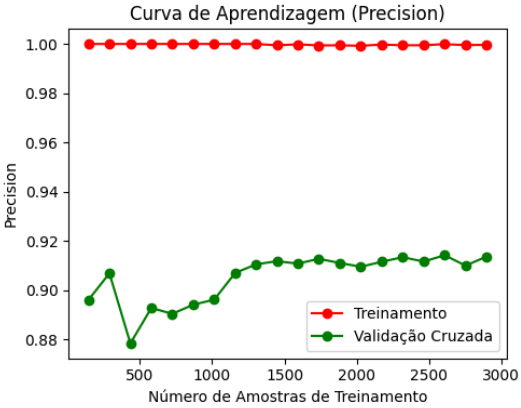
- Conforme o número de amostras de treinamento aumenta, as métricas de validação cruzada melhoram e se aproximam das curvas de treinamento, indicando que o modelo ganha generalização à medida que mais dados são usados.

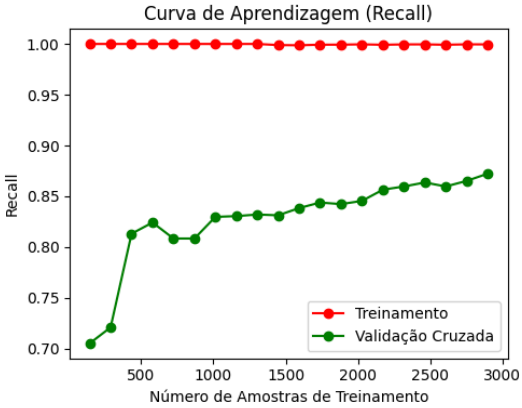
Visualização Gráfica

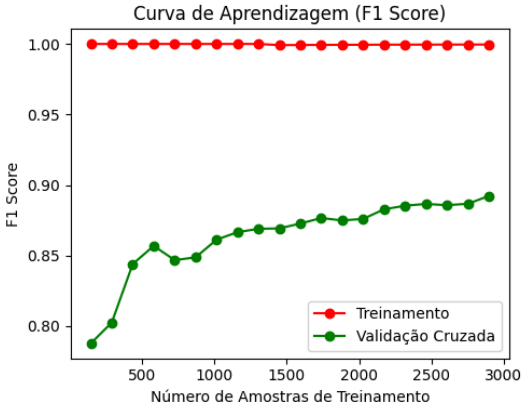
- **Matriz de confusão:** Retornou uma boa separação entre classes, com baixa incidência de falsos positivos e falsos negativos.



- **Curva de aprendizagem:** Conforme visualizado na linha vermelha, vemos que nas três métricas o treinamento se manteve estável próximo do 1, o que indica boa adaptação do modelo ao conjunto de dados. A curva em verde, que representa a validação cruzada, como esperado, se manteve subindo gradativamente, melhorando o acerto conforme a base de dados aumentava.







No geral, o KNN apresentou um bom equilíbrio entre *precision* e *recall*, resultando em um *F1 score* robusto. Isso é desejável em tarefas de detecção de spam, onde se busca identificar a maior quantidade de spam possível (recall) sem classificar erroneamente e-mails legítimos como spam (precision).