# Aprendizado de Máquina Supervisionado I - Teste 01 (INF0615)

Leonardo Cesar Silva dos Santos, Fernando Augusto Cardoso Candalaft

Março 2024

## Contents

# 1 Exploração dos dados

Inicialmente recebemos três arquivos correspondendo aos dados de treinamento, validação e teste. Os dados de treinamento tinham 9336 linhas e 19 colunas, colunas essas divididas entre dados contínuos, categóricos (dias da semana) e a variável a ser predita (*target*).

Ao inspecionarmos os dados de treinamento notamos que os mesmos não possuíam valores duplicados e também não possuíam valores nulos. Mas caso encontrássemos valores nulos nos dados de treinamento poderíamos tomar alguma ação com relação a isso. Por exemplo, se tivéssemos poucas linhas com valores nulos com relação ao número total de linhas poderíamos simplesmente descartar as mesmas, ou se tivéssemos o caso de uma coluna ter muitos valores nulos, poderíamos optar por não usar a mesma. Uma última alternativa com relação ao tratamento de valores nulos poderia ser substituir os mesmos pela média, mínimo, máximo ou outro valor a depender do contexto do problema, por exemplo, *forward fill*, caso os dados da tabela fossem temporais. Vale comentar como um possível ponto negativo que ao preencher os valores com a média altera-se a variância da *feature*.

# 2 Normalização

Nesta etapa normalizamos nossos dados contínuos (não usamos os valores categóricos, dado que não faz sentido e foi dito várias vezes em aula para não fazermos isso) via normalização *Z-Score* e também convertemos nossa coluna categórica *weekday* em colunas binárias por meio do método *One-Hot-Encoding*, obtendo assim 7 novas colunas, cada uma referente a um dia da semana e descartamos a coluna *weekday*. Aplicamos o mesmo processo para os dados de validação e teste, porém aqui tomamos cuidado para somente transformar os dados contínuos do conjunto de validação usando os dados de média e desvio padrão calculados para os dados de treino, isto é, aplicamos somente o "*transform*" dos dados contínuos de validação.

# 3 Definição do modelo *baseline*

Antes de aplicarmos o modelo baseline definimos uma "semente" (*seed*) para garantir a reprodutibilidade do experimento.

Treinamos um modelo *baseline* usando uma regressão linear (polinomio de grau 1) com as *features* contínuas e categóricas (*one-hot*). Avaliamos a performance do modelo em questão usando as métricas *MAE*, *MSE* e *R2*, que estão listadas na Tabela 1.

Table 1: Resultados do modelo

| **Conjunto** | **MAE** | **MSE** | **R2** |
|---|---|---|---|
| Treino | 716.382 | 873307.044 | 0.0719 |
| Validação | 718.937 | 890337.640 | 0.0702 |

**Obs.:** Avaliamos todos os modelos usando as métricas acima.

Algo perceptível com o modelo *baseline* é que o mesmo tem uma performance baixa. Podemos ver isso pela Figura 1 de valores preditos versus valores reais sobre o conjunto de validação. A expectativa era de que tivéssemos uma reta se nosso modelo fosse razoavelmente bom. O valor baixo de *R2* também confirma a nossa constatação, dado que um baixo valor nessa métrica indica que que a explicabilidade do nosso modelo com relação aos dados de treino e validação é baixa.

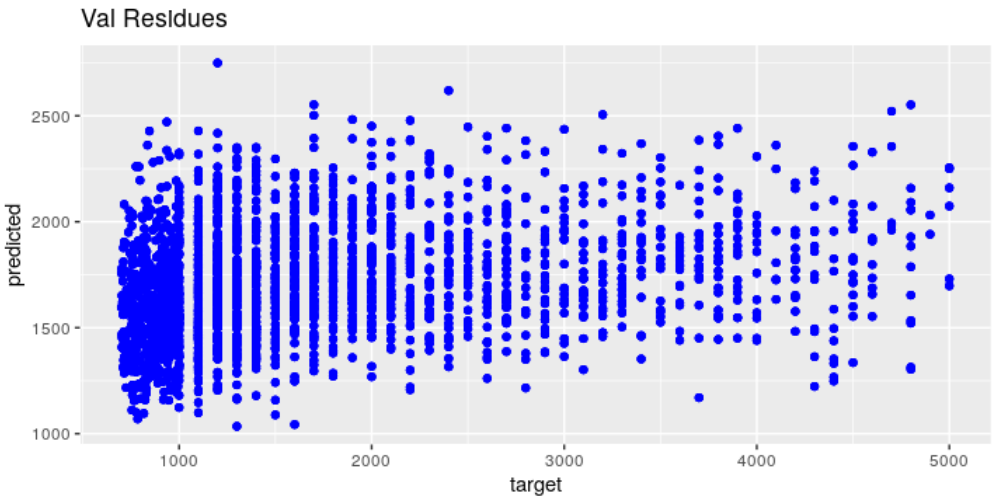


Figure 1: Valores preditos sobre o conjunto de validação x valores reais

# 4 Implementação de soluções alternativas

Para as soluções alternativas utilizamos o auxílio do *ChatGPT* para realizar a combinação das variáveis. Após isso avaliações diversas combinações dos valores contínuos e também os mesmos mais os valores categóricos. A melhor combinação que encontramos foi a nomeada "*Combination 07*" dada por

$$
\begin{aligned}
\text{target} \sim & \left(\frac{\text{global\_sentiment\_polarity}}{\text{log\_n\_tokens\_content}}\right) + \\
& (\text{num\_keywords} \times \text{global\_rate\_positive\_words}) + \\
& \left(\frac{\text{avg\_positive\_polarity}}{\text{avg\_negative\_polarity}}\right) + \\
& \text{log\_num\_hrefs} + \\
& (\text{log\_n\_tokens\_content} \times \text{log\_self\_reference\_max\_shares}) + \\
& (\text{global\_subjectivity} \times \text{global\_rate\_negative\_words}) + \\
& \left(\frac{\text{avg\_negative\_polarity}}{\text{global\_subjectivity}}\right) + \\
& (\text{rate\_positive\_words} \times \text{rate\_negative\_words}) + \\
& \left(\frac{\text{log\_self\_reference\_avg\_sharess}}{\text{n\_tokens\_title}}\right) + \\
& \text{Saturday} + \text{Friday} + \text{Tuesday} + \text{Monday} + \text{Wednesday} + \text{Thursday} + \text{Sunday}
\end{aligned}
$$

e cujas métricas encontradas estão na Tabela 2

Table 2: Resultados do modelo

| **Conjunto** | **MAE** | **MSE** | **R2** |
|---|---|---|---|
| Treino | 717.286 | 873973.873 | 0.0712 |
| Validação | 720.641 | 891991.600 | 0.0684 |

# 5 Exploração de modelos polinomiais

Exploramos também modelos polinomiais com diferentes graus (2 a 11). Encontramos os seguintes valores de *MAE* nos conjuntos de treino e validação. Fica claro pela curva de Viés e Variância da Figura 2 que o menor

Table 3: Resultados do modelo com diferentes graus polinomiais

| polyDegree | TrainMAE | ValMAE |
|---|---|---|
| 2 | 712.9499 | 716.5688 |
| 3 | 712.2570 | 717.1419 |
| 4 | 710.8237 | 715.8935 |
| 5 | 710.1103 | 716.0732 |
| 6 | 709.0994 | 717.6095 |
| 7 | 708.6519 | 717.8417 |
| 8 | 708.1572 | 718.6752 |
| 9 | 707.1297 | 722.0124 |
| 10 | 706.5625 | 722.3649 |
| 11 | 706.3792 | 722.4782 |

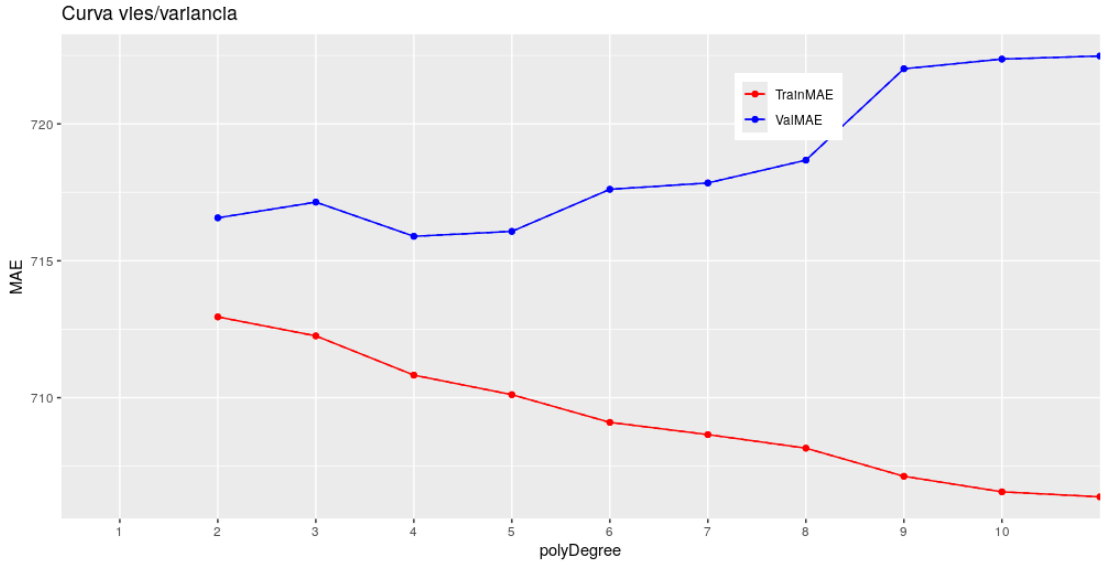


Figure 2: Curva de viés e variância para diferentes graus polinomiais.

valor de *MAE* no conjunto de validação é encontrado para *degree* = 4. Após isso vemos que nosso modelo começa a entrar em *overfitting*, ou seja, seu erro sobre o conjunto de treinamento começa a ficar muito distante do conjunto de validação, indicando que nosso modelo é super-especializado no conjunto de treino para graus muito alto de polinômios.

# 6 Escolha do melhor modelo e avaliação no conjunto de teste

Dentre os modelos avaliados anteriormente constatamos que o **melhor foi o modelo polinomial com grau** 4, cujo valor de *MAE* (Tabela 3) **sobre o conjunto de validação** foi o menor se comparado com o modelo *baseline* e com o modelo *Combination 07*. Porém, vale comentar que o valor de *R2* não teve uma melhora perceptiva. Os valores das métricas obtidas pelo melhor modelo estão presentes na tabela 4.

Table 4: Métricas dos modelos sobre o conjunto de teste

| MAE | MSE | R2 | Model |
|---|---|---|---|
| 727.6314 | 903732.3 | 0.06543409 | Baseline |
| 729.6173 | 908892.9 | 0.06009741 | Combination 07 |
| 726.8799 | 901666.0 | 0.06757085 | Polynomial degree = 4 |

# 7 Conclusão

De modo geral exploramos três modelos principais. O modelo *baseline* foi o nosso ponto de partida e o modelo mais simples implementado. Qualquer modelo que escolhêssemos deveria ser melhor que este. Já o modelo polinomial de grau 4 foi um modelo mais complexo que explorou tanto as *features* contínuas e diferentes níveis como também as categóricas. Tal modelo é mais complexo que o nosso *baseline* e é esperado que performe melhor. Por fim, o modelo *Combination 07* foi um modelo gerado a partir de diversas combinações entre as *features* do problema, mas tal combinação não levou em consideração diversos fatores que poderíamos analisar para obter uma performance melhor caso houvesse mais tempo, por exemplo, correlação de *features* de modo a perceber multicolinearidade nos dados e se possível eliminá-la, melhor combinação de *features*, análise de redução de dimensionalidade, etc. De modo que no final, nosso melhor modelo foi o intermediário (polinômio de grau 4).