Trabalho Final de AEAD

Álvaro J. S. Kothe

28-11-2021

Dados de dígitos manuscritos

O conjunto de dados de dígitos manuscritos possui 60000 dígitos escritos a mão em imagens de dimensão 28×28. O conjunto de dados foi separado em conjunto de treino e validação, onde 25% dos dados foi utilizado como conjunto de validação, ou seja, uma amostra de 45000 dígitos será utilizada para construir os modelos, e os dígitos restantes serão utilizados para avaliar a qualidade de ajuste dos modelos.

Os pixels foram normalizados para a escala $[0, 1]$, (equivalente a por a imagem na escala de cinza) para que alguns dos modelos utilizados convirjam mais rápido.

Com os dados normalizados e separados em conjunto de treino e teste, serão construídos três modelos para realizar as predições das imagens:

- Lasso Multinomial
- Boosting
- Rede Neural Convolucional

Modelo linear com regularização

Como o problema é diferenciar 10 dígitos utilizando 784 pixels, isso indica que o problema é de classificação com mais de duas classes. Logo, será ajustado um modelo multinomial. Como diversos pixels não são preenchidos para escrever os dígitos, é interessante selecionar os pixels importantes para a classificação de cada dígito. Por isso, será utilizada a regularização Lasso. Para a escolha do parâmetro de penalização, foi gerado 100 hiper-parâmetros de penalização, e foi escolhido o melhor hiper-parâmetro através de validação cruzada utilizando apenas o conjunto de treino.

Na figura 1 é apresentado os pixels que não foram nulos para a classificação de cada dígito, com o dígito referente em verde no canto superior esquerdo de cada imagem.

Pela Figura 1, é possível ver os principais pixels para a classificação de cada dígito. Pode-se notar que a partir dos pixels não nulos, é possível desenhar o dígito que deseja-se classificar.

Modelo baseado em árvore

Existem vários tipos possíveis de modelos baseados em árvore que podem ser utilizados. Para esse problema de classificação de imagem, será escolhido o modelo de *boosting*. O pacote utilizado para ajustar o modelo é o XGBoost.

Para a seleção dos hiper-parâmetros, mantendo o número de árvores fixo em 100, foi ajustado 75 modelos para selecionar a taxa de aprendizagem (`eta`), a profundidade máxima da árvore (`max_depth`) e o peso mínimo para criar outra partição da árvore (`min_child_weight`). Os parâmetros escolhidos foram:

- eta: 0.1
- max_depth: 6
- min_child_weight: 4

Em seguida, foi selecionado o número de árvores a partir do conjunto de validação, em que o modelo para de ser ajustado após um certo número de iterações em que não teve melhora no conjunto de validação.

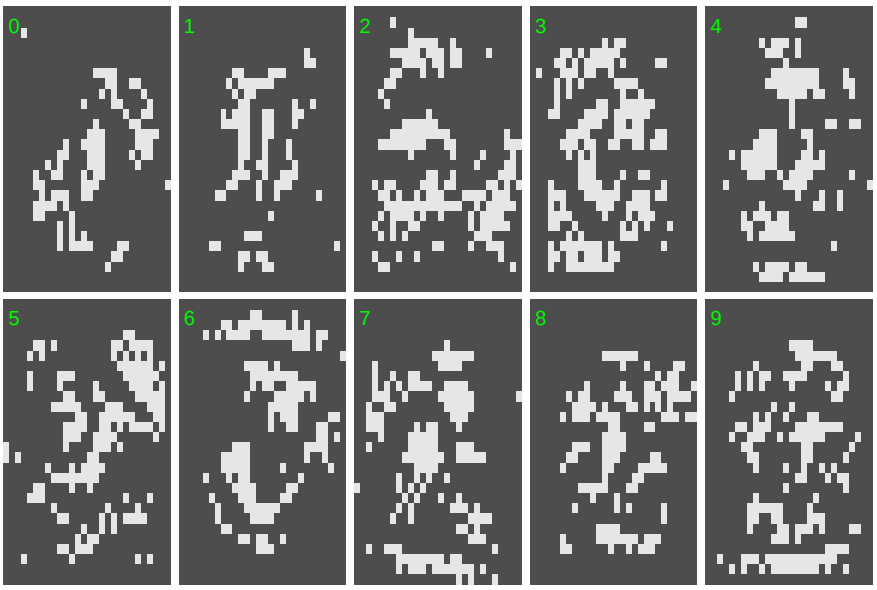


Figura 1: Pixels não nulos para cada dígito pelo modelo lasso.

Redes neurais

Para o modelo de rede neural, foi considerado uma rede neural convolucional base com acurácia 0.9897. Para a rede base, foi utilizada uma camada convolucional e uma camada escondida. Todas camadas tiveram função de ativação `relu`. O otimizador utilizado foi o `Adam` com taxa de aprendizagem 0,0001.

Em seguida foi ajustado diversas redes, manipulando a quantidade de camadas intermediárias, adicionando camada de `dropout` e manipulando o otimizador (Ruder apresentou em um blog diversos otimizadores baseados em gradiente descendente). A estrutura da rede final é:

- camada convolucional com 32 filtros, janela deslizante 5×5 e ativação relu;
- camada de `max_pooling` de dimensão 2×2;
- camada de `dropout` com taxa 0.2;
- camada convolucional com 64 filtros, janela deslizante 5×5 e ativação relu;
- camada de `max_pooling` de dimensão 2×2;
- camada de `dropout` com taxa 0.2;
- camada de achatamento (`flatten`);
- camada `dense` com 512 neurônios e ativação relu;
- camada de `dropout` com taxa 0.2;
- a última camada é uma `dense` com ativação softmax de 10 neurônios para classificar o dígito.

O otimizador escolhido foi o `nadam` que é similar ao `adam` mas contém momento para acelerar a convergência. Foi definido que o ajuste seria parado se houvesse 5 épocas sem melhora, e o modelo foi ajustado em lotes de tamanho 64.

Comparação dos modelos

Para a comparação dos modelos será utilizado a acurácia. Na Tabela 1 é apresentado a acurácia para o conjunto de treino (Acurácia Dentro) e validação (Acurácia Fora).

Pela Tabela 1, o modelo de rede neural foi o que teve as melhores predições no conjunto de validação. Nota-se também que o modelo de Boosting predizeu perfeitamente no conjunto de treino, o que mostra que esse modelo conseguiu decorar o conjunto de treino. Porém, conseguiu predizer corretamente na maior parte no conjunto de validação.

Como o modelo de rede neural convolucional foi o que teve a maior acurácia no conjunto de validação, ele será o modelo escolhido para realizar as predições no conjunto de teste. Para apresentar um pouco das

Tabela 1: Acurácias dos modelos ajustados para o conjunto de dados MNIST.

Modelo	Acurácia Dentro	Acurácia Fora
Multinomial Lasso	0.9200667	0.9089333
Boosting	1.0000000	0.9742667
Rede Neural Convolucional	0.9981333	0.9918000

predições do modelo escolhida, na Figura 2 é apresentado a matriz de confusão do modelo no conjunto de validação.

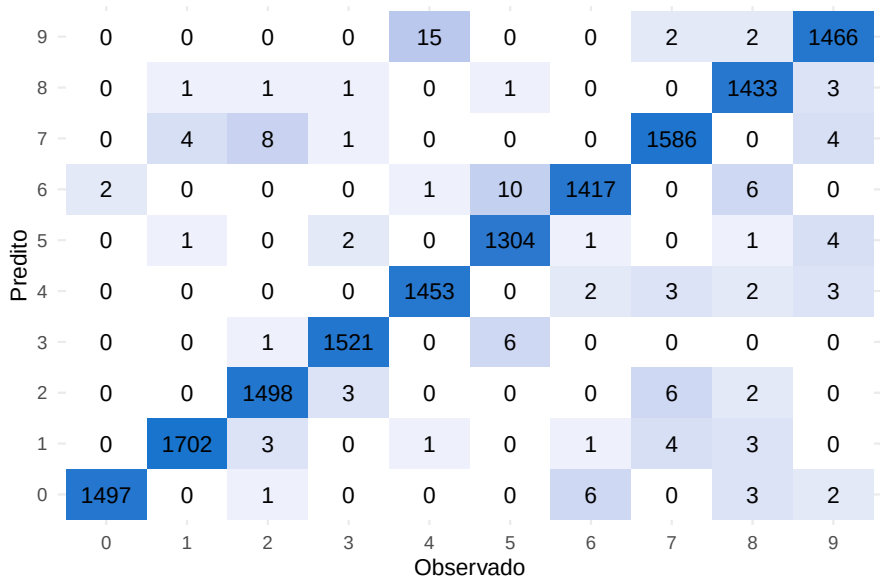


Figura 2: Matriz de confusão para o modelo de rede neural convolucional para o conjunto de dados de validação.

Pela Figura 2, não se percebe um dígito em que o modelo apresentou dificuldades de prever.

Na Figura 3 se encontram algumas predições no conjunto de validação. As predições corretas estão na cor verde, e as incorretas estão na cor vermelha com a predição correta dentro dos parênteses.

Pela Figura 3, nota-se alguns dígitos que são facilmente reconhecíveis, porém o modelo predizeu incorretamente. Mas como não se espera que um modelo consiga acertar em todo o conjunto de validação, e como ele teve uma ótima acurácia, ele será o modelo escolhido para realizar as predições no conjunto de teste.

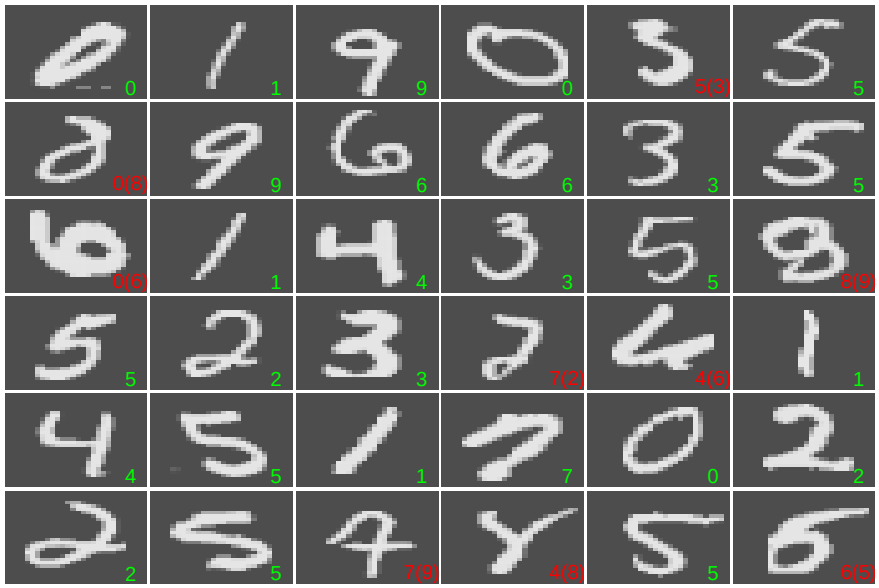


Figura 3: Amostra das classificações da rede neural para o conjunto de validação.

Dados de chuva e vazão no Rio São Francisco

No conjunto de dados sobre a vazão no rio são francisco, tem 1717 semanas de coletas de vazão e precipitação de diversas estações. O objetivo é predizer a vazão na estação 46998000 na semana seguinte.

Comparado ao conjunto de dados de dígitos MNIST, o conjunto de dados não possui muitas observações. Por isso, para a estimação do erro será utilizada a validação cruzada.

Relação da variável resposta com os preditores.

Na Figura 4, é apresentado o gráfico de dispersão da estação resposta contra a sua defasagem de primeira ordem.

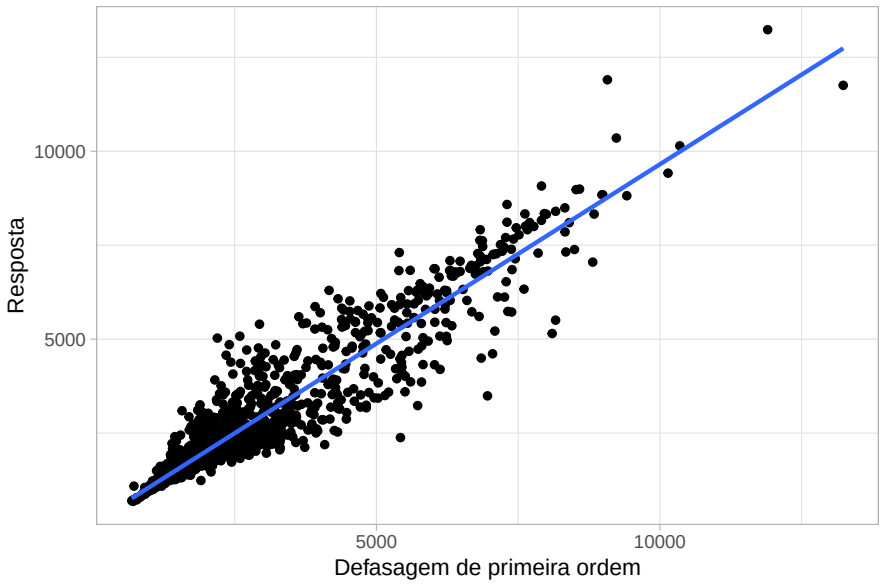


Figura 4: Gráfico da resposta contra a sua defasagem.

Pela Figura 4, nota-se que existe uma forte relação linear entre a resposta e a sua defasagem. Na Figura 5 é apresentado o correlograma das variáveis que tiveram as maiores correlações com a variável resposta.

Todas as bacias que apresentaram correlação acima de 0.9 com a variável resposta `Y`, pertencem ao rio são francisco. Também nota-se que existe uma correlação alta e positiva entre essas bacias.

Correção de assimetria e escala

É recomendado realizar transformações nas variáveis contínuas para que elas estejam simétricas em torno da média. O pacote *scikit-learn* fez um exemplo que apresenta as vantagens de construir os modelos com a variável resposta transformada, e em seguida realiza a transformação inversa para obter melhores predições. Um dos principais motivos para realizar esse tipo de transformação é para que pontos discrepantes não tenham tanta influência no modelo.

A Figura 6a apresenta as densidades das vazões e precipitações das estações até o quantil de 90% de todas as estações. A Figura 6b apresenta a densidade da variável resposta.

Pelas Figuras 6a e 6b, nota-se que as vazões são bastante assimétricas. Por isso, será aplicado a transformação $\log(x+1)$ em todas as vazões e precipitações para reduzir a assimetria, incluindo a variável resposta. Na Figura 7 são apresentadas todas as vazões e precipitações transformadas das estações preditoras e a vazão da variável resposta.

Nota-se pela Figura 7, que a transformação $\log(x+1)$ conseguiu reduzir bastante a assimetria das vazões e precipitações.

Em seguida, para remover o efeito da escala das vazões e precipitações, essas medidas foram padronizadas.

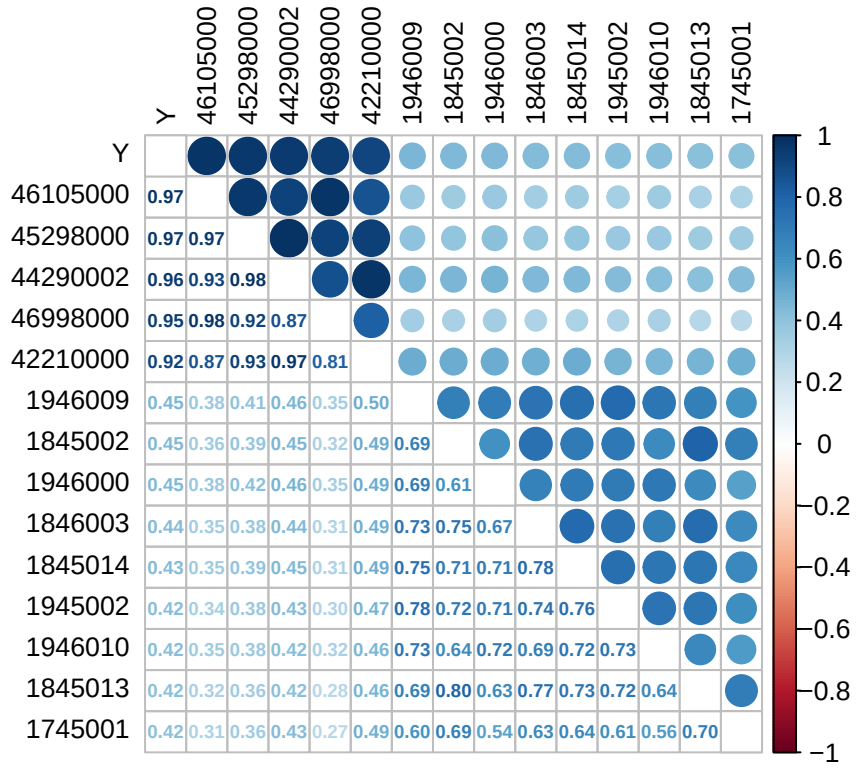
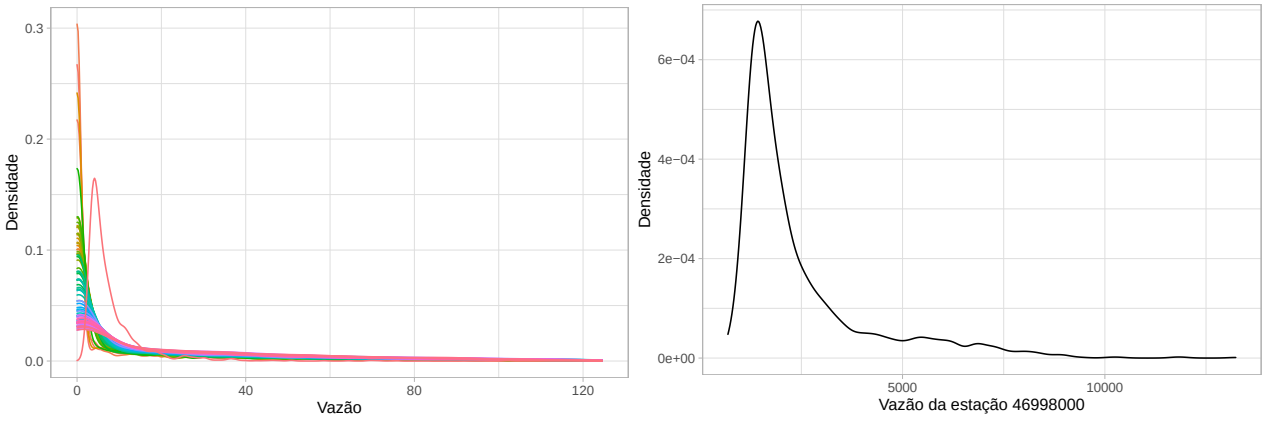


Figura 5: Matriz de correlação



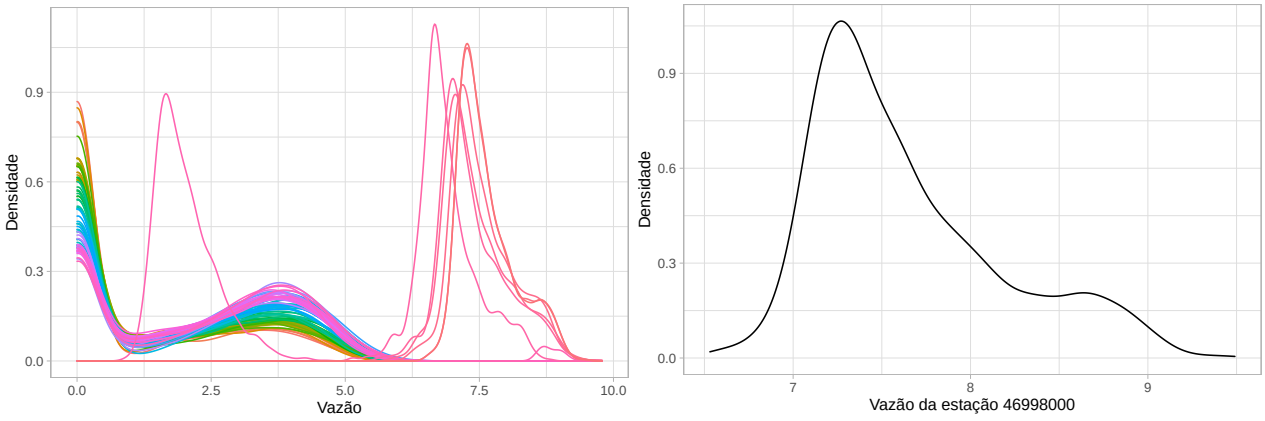
(a) Todas as estações até o quantil 90%

(b) Variável resposta

Figura 6: Gráfico de densidades das estações.

Para a predição da estação 46998000, será utilizado apenas os dados das vazões da semana anterior, incluindo a vazão da própria estação 46998000. Além disso, mesmo que os dados estejam padronizados, a vazão da semana seguinte será predita com a transformação $\log(x+1)$. Para conjunto de teste será aplicado a transformação $\exp(x)-1$ nas predições para que a vazão esteja na escala original.

Para ilustrar as transformações feitas, e o que será utilizado para ajustar os modelos, na Tabela 2 é apresentado as 6 primeiras observações do conjunto de treinamento, em que na primeira coluna tem-se a vazão da semana seguinte que deseja-se prever com a transformação $\log(x+1)$. As demais colunas são as 6 primeiras colunas das vazões da semana anterior já transformadas e padronizadas.



(a) Todas as estações

(b) Variável resposta

Figura 7: Gráfico de densidade das variáveis transformadas.

Tabela 2: Primeiras linhas e colunas do conjunto de treinamento, com a primeira coluna sendo a variável resposta transformada (vazão da semana seguinte).

$\log(Y+1)$	40025000	42210000	44290002	45298000	46105000	46998000
7.261	-0.248	-0.571	-0.536	-0.541	-0.546	-0.543
7.842	0.199	-0.733	-0.455	0.052	0.080	0.414
8.241	0.362	1.120	1.162	1.178	1.259	1.298
7.643	-0.141	0.040	0.095	0.044	0.013	-0.013
7.286	-0.311	-0.017	-0.452	-0.434	-0.502	-0.610
8.162	1.274	1.242	1.184	1.013	0.729	0.367

Ajuste dos modelos

Como já definido, os modelos serão comparados por validação cruzada. O método de validação cruzada escolhido foi o *k-folds*, com $k = 10$, ou seja, será utilizado 9 amostras para ajustar os modelos, e uma amostra para validação. Esse processo será repetido para cada amostra (*fold*). A medida utilizada para comparar os modelos será o erro médio absoluto na variável resposta transformada (com a transformação $\log(x+1)$). Não é necessário comparar com a resposta na escala original pois a transformação logarítmica é uma transformação monótona.

Os modelos ajustados serão:

- Modelos lineares com regularização:
  - Lasso
  - Ridge
- Modelos baseados em árvores:
  - Floresta aleatória
  - Boosting utilizando o pacote XGBoost
- Rede Neural

Para os modelos de Lasso e Ridge, primeiro foi escolhido o melhor parâmetro de penalização através da validação cruzada, utilizando as mesmas amostras já definidas pelo *k-fold*. Em seguida, dado os parâmetros de penalização que apresentaram o menor erro de validação cruzada, foi feito novamente a validação cruzada para obter o erro médio absoluto de cada *fold*

O modelo de floresta aleatória foi ajustado com a função `randomForest` do pacote com o mesmo nome.

De acorto com o pacote `caret`, o principal hiper-parâmetro de floresta aleatória a ser otimizado é o número de variáveis amostradas em cada divisão da árvore (`mtry`). Para isso, foi escolhido aleatóriamente 10 valores para esse hiper-parâmetro, e foram comparados por validação cruzada fixando o número de árvores em 50. O parâmetro escolhido dessa amostra foi testado novamente, só que dessa comparado com 3 valores anteriores e consecutivos a ele, fixando o número de árvores em 100. O hiper-parâmetro `mtry` escolhido foi 54.

Após a seleção do hiper-parâmetro `mtry`, o modelo foi ajustado utilizando 500 árvores, que é o padrão do pacote `randomForest`.

Para a seleção dos hiper-parâmetros do modelo de boosting, manteve-se o número de árvores fixo em 100, foi ajustado 216 modelos para selecionar a taxa de aprendizagem (`eta`), a profundidade máxima da árvore (`max_depth`), a perda mínima para criar outra partição da árvore (`gamma`), amostragem de observações e variáveis (`subsample` e `colsample_bytree`, respectivamente) e regularização L1 (`alpha`). Os parâmetros que tiveram o menor erro médio absoluto pela validação cruzada foram:

- eta: 0.1
- max_depth: 6
- gamma: 0
- subsample: 1
- colsample_bytree: 1
- reg_alpha: 0

Maior parte dos hiper-parâmetros selecionados são o padrão da função `xgboost`. Em seguida, foi selecionado o número de árvores `ntrees`, também pela validação cruzada. O valor escolhido foi 500.

A estrutura do modelo de redes neurais foi definida a partir da tentativa e erro, utilizando validação cruzada. onde primeiramente foi definido que a primeira camada teria 128 neurônios, e as seguintes teriam a metade da camada anterior. Além disso, foi definido que todas as camadas teriam ativação sigmóide, o otimizador utilizado foi o `adam`. O modelo foi treinado em 100 épocas em lotes de tamanho 64. Com isso, encontrou que o número ótimo de camadas escondidas seria 4. Em seguida foi escolhida a ativação de cada camada.

Depois de definido a estrutura da rede, foi escolhido a quantidade de épocas para treino, em que foi escolhido 200 épocas. A estrutura final da rede é

- camada `dense` com 128 neurônios e ativação sigmóide;
- camada `dense` com 64 neurônios e ativação sigmóide;
- camada `dense` com 32 neurônios e ativação relu;
- camada `dense` com 16 neurônios e ativação relu;
- a camada de saída é uma `dense` com ativação linear;

Comparação dos modelos

Selecionado os hiper-parâmetros de cada modelo, eles serão comparados pela validação cruzada. Na Figura 8 os erros médios absolutos estimados são apresentados em um boxplot, com a média como um ponto em vermelho. Na Tabela 3 é apresentado a média e o desvio padrão estimados para cada modelo.

Tabela 3: Média e desvio do erro médio absoluto estimado dos modelos pela validação cruzada

Modelo	Média	Erro padrão
Floresta Aleatória	0.0442830	0.0039404
Boosting	0.0452870	0.0032188
Rede Neural	0.0573199	0.0028580
Lasso	0.0585185	0.0029573
Ridge	0.0692288	0.0034015

Todos os modelos apresentaram erros estimados pela validação cruzada próximos, mas os modelos baseados em árvore foram os que apresentaram os menores erros.

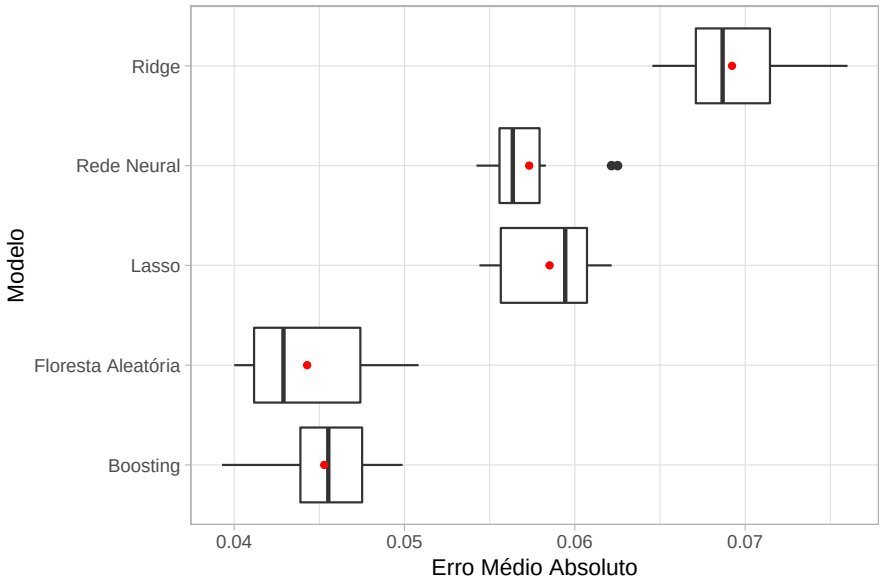


Figura 8: Boxplot do erro médio absoluto estimado por modelo pela validação cruzada.

Escolha do modelo

Geralmente para ter melhores predições, as predições dos vários modelos de *machine learning* são misturadas. Uma das principais vantagens de misturar as predições é que evita o sobreajuste, e as predições são muito mais robustas. Existem diversas formas de misturar as predições, desde tomar a média delas, até construir modelos sobre essas predições.

Todos os modelos foram ajustados novamente utilizando todo o conjuto de treino. Para as predições finais, será utilizada a média ponderada de todos os modelos ajustados, em que os pesos serão definidos a partir da performance dos modelos apresentadas na Figura 8 e Tabela 3.

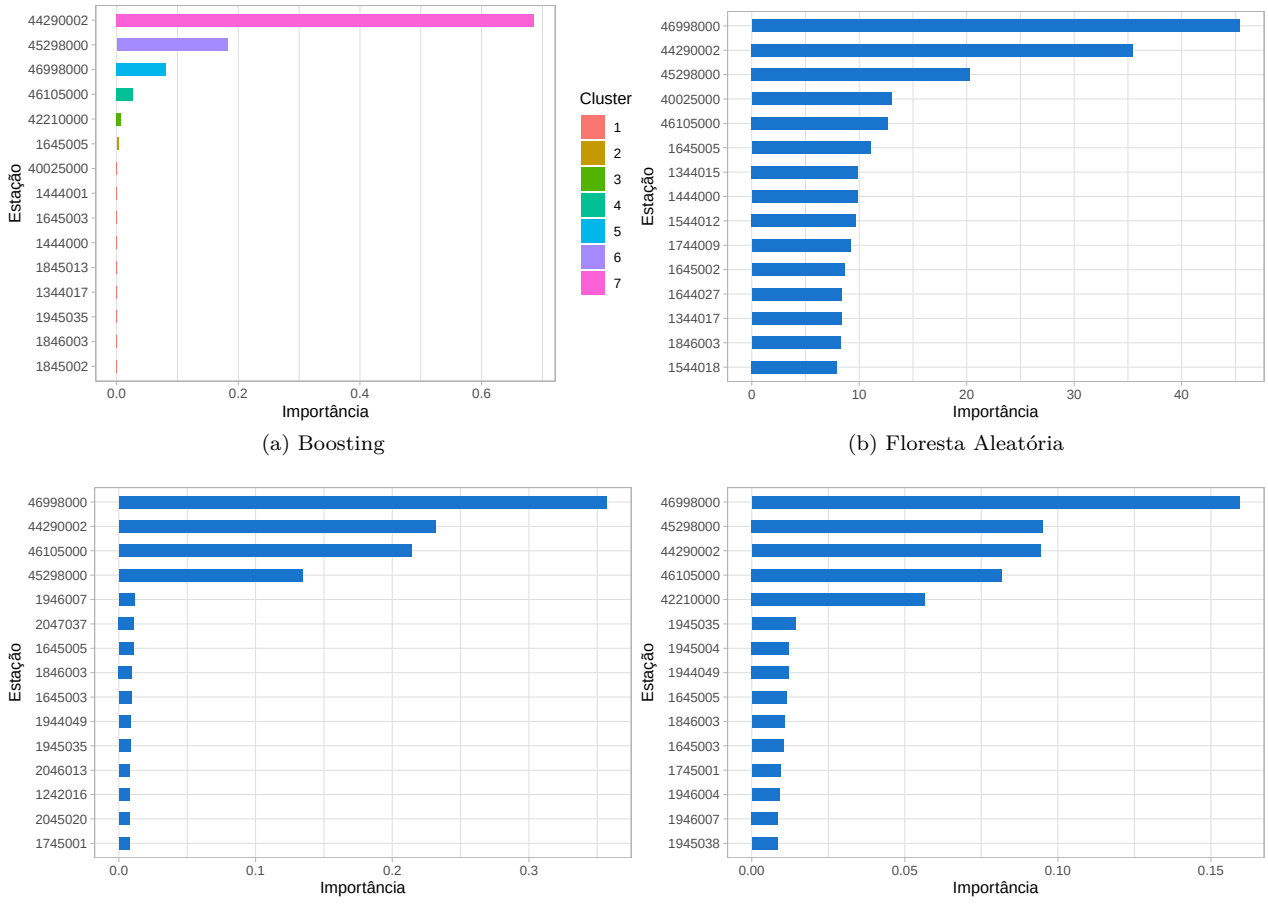
Após tomar a média ponderada será utilizada a transformação $\exp(x) - 1$ na média ponderada para que a vazão predita esteja na escala original da variável resposta. Logo, as predições no conjunto de teste são dadas por:

$$\hat{y} = \exp\{0.240\hat{y}_F + 0.235\hat{y}_B + 0.189\hat{y}_{RN} + 0.182\hat{y}_L + 0.154\hat{y}_R\} - 1,$$

onde $\hat{y}_F, \hat{y}_B, \hat{y}_{RN}, \hat{y}_L, \hat{y}_R$ são as predições de Floresta aleatória, Boosting, Rede Neural, Lasso e Ridge, respectivamente.

A Figura 9 apresenta a importância de cada preditor para os modelos de Boosting, Floresta Aleatória, Lasso e Ridge. A importância dos modelos de Lasso e Ridge é basicamente o valor absoluto do coeficinete.

Pela Figura 9, as estações que pertencem ao rio são francisco foram as que tiveram a maior importância em todos os modelos. O modelo de boosting foi o único que não teve a vazão da semana anterior da estação 46998000 como o preditor mais importante.



(a) Boosting

(b) Floresta Aleatória

(c) Lasso

(d) Ridge

Figura 9: Importância de variável