Dados de dígitos manuscritos

No conjunto de dados de dígitos manuscritos, temos 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.

Em seguida os pixels foram normalizados para a escala $[0, 1]$, (equivalente a por a imagem na escala de cinza) para que alguns dos modelos utilizados convirja 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. E como diversos pixels geralmente não são preenchidos para escrever os dígitos, é interessante realizar selecionar os pixels importantes para a classificação de cada dígito. Por isso, será ajustado um modelo Lasso Multinomial. 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.

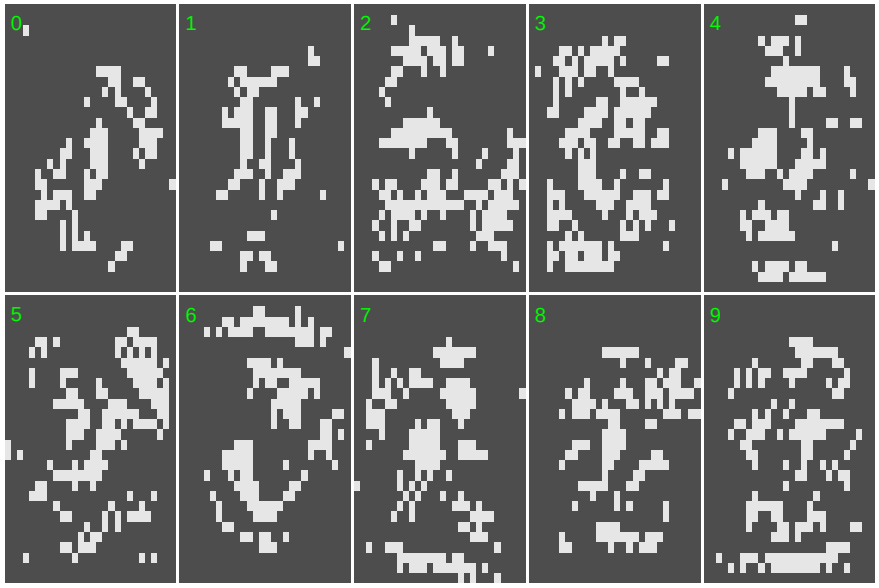


Figura 1: Pixels não nulos para cada dígito pelo modelo lasso.

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

Existe vários tipos possíveis de modelos baseados em árvore que podem ser utilizados. Porém, para casos de classificação de imagem, mais especificamente a classificação de dígitos do MNIST, modelos de *boosting*

conseguem oferecer um bom ajuste. Por conta disso, será ajustado um modelo de boosting utilizando o pacote XGBoost.

Para a seleção dos hiper-parâmetros, foi ajustado 75 modelos para selecionar a taxa de aprendizagem (`eta`), a profundidade máxima da árvore (`max_depth`) e a perda mínima para criar outra partição da árvore (`min_child_weight`). Os parâmetros escolhidos foram:

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

O número de árvores foi controlado pelo conjunto de validação, porém foi definido que o número máximo de árvores seria 500.

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 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).

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	0.9976667	0.9922000

Podemos ver pela Tabela 1 que 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 predições do modelo escolhida, na Figura 2 é apresentado a matriz de confusão do modelo no conjunto de validação.

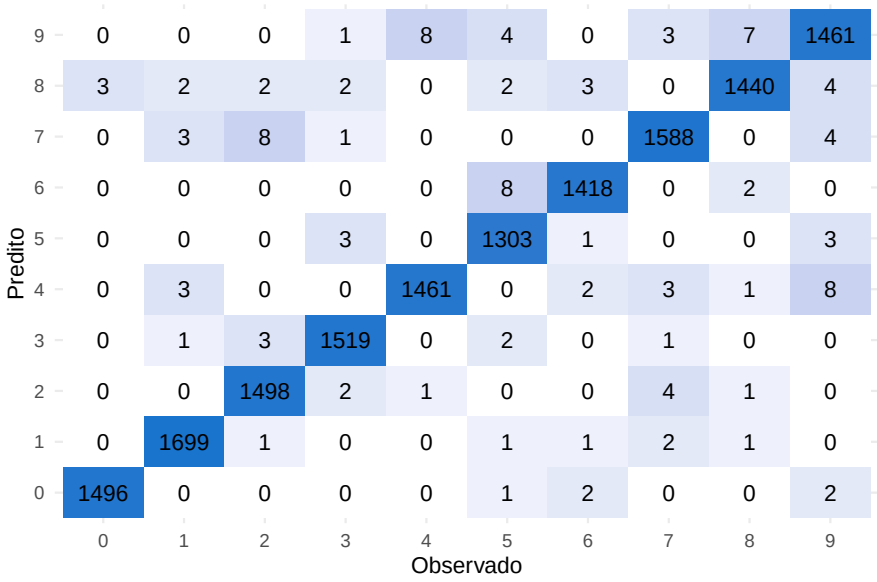


Figura 2: Matriz de confusão para o modelo de rede neural convolucional para o conjunto de dados de validação.

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.

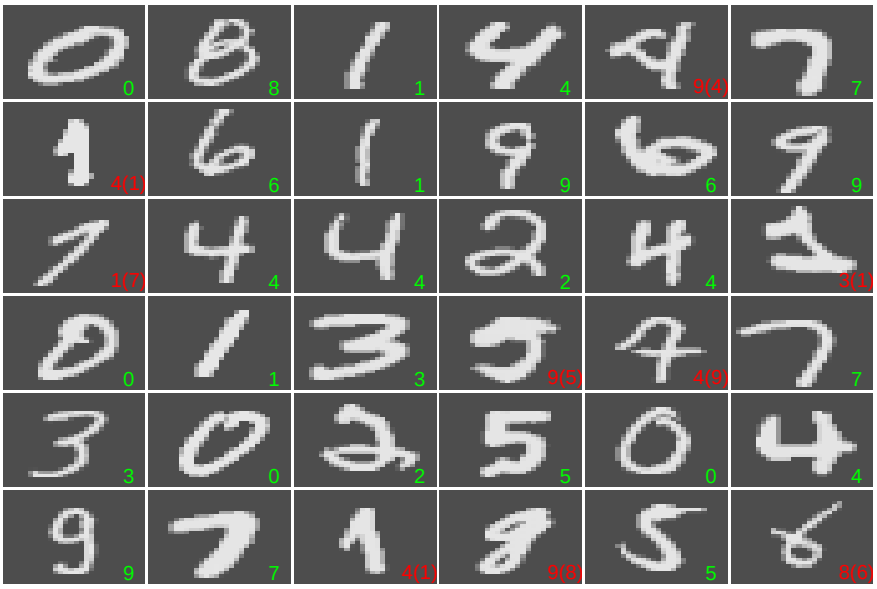


Figura 3: Amostra das classificações da rede neural para o conjunto de validação.

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.

Dados de chuva e vazão no Rio São Francisco

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

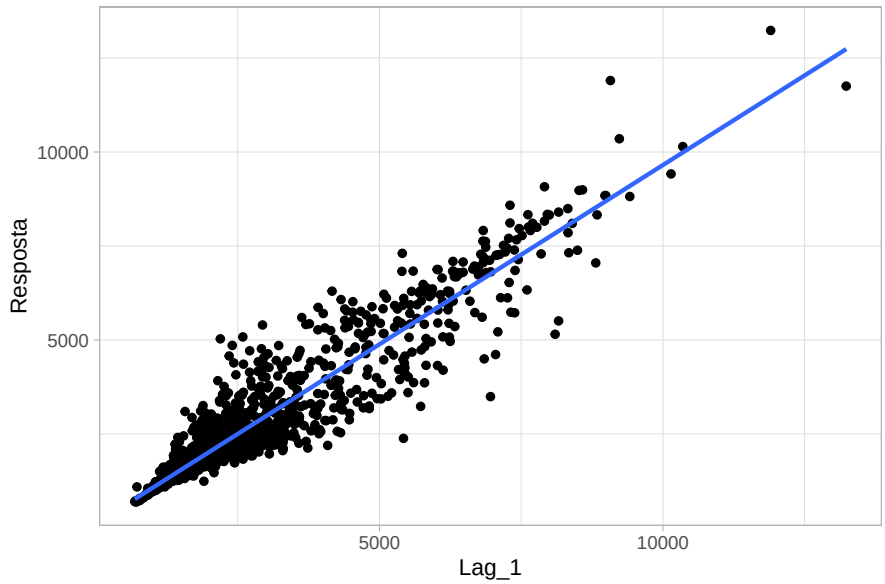


Figura 4: Gráfico da resposta contra a sua defasagem.

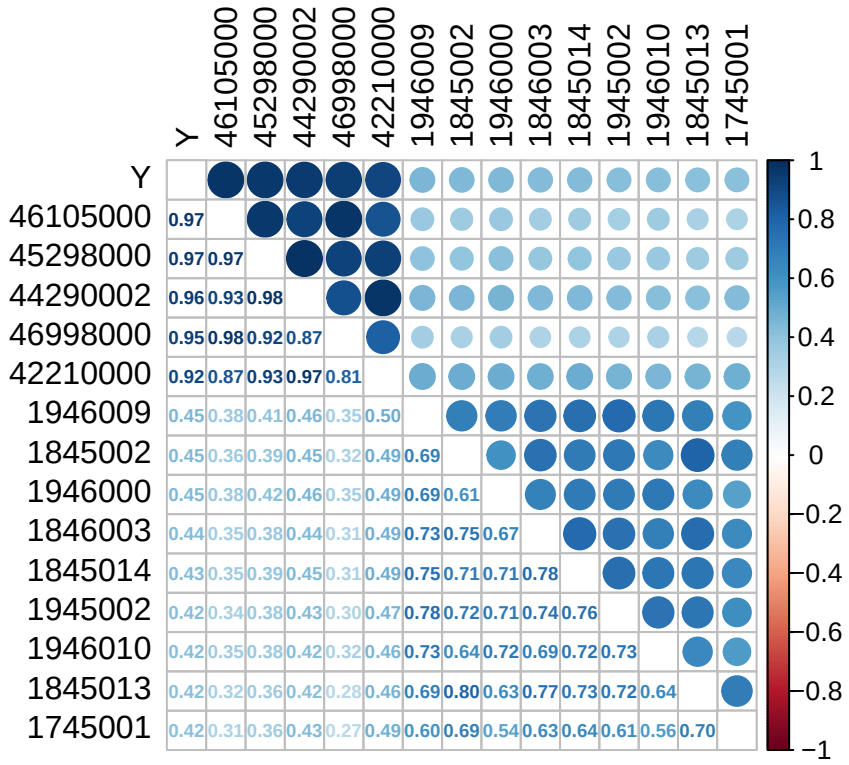


Figura 5: Heatmap das variáveis com a maior correlação com a variável resposta.

Todas as bacias que apresentaram correlação acima de 0.9 com a variável resposta `Y`, pertencem ao rio são francisco.

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

O conjunto de dados possui diversas informações faltando, e comparado ao conjunto de dados de dígitos MNIST, não possui muitas observações. Por isso, para o tratamento e seleção de modelos será considerado a seguinte abordagem:

- Imputação: Será utilizado todo conjunto de treino disponibilizado;
- Tratamento
- Seleção de modelo: Será utilizado validação cruzada para comparar os modelos considerando o erro absoluto médio;
- Embedding

Correção de assimetria e escala

É recomendado realizar transformações nas variáveis para que elas sejam normalmente distribuídas. O pacote *scikit-learn* fez um exemplo para apresentar 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 um peso tão grande no modelo.

A Figura 6 apresenta as densidades das vazões e precipitações das estações até o quantil de 90% para melhor visualização.

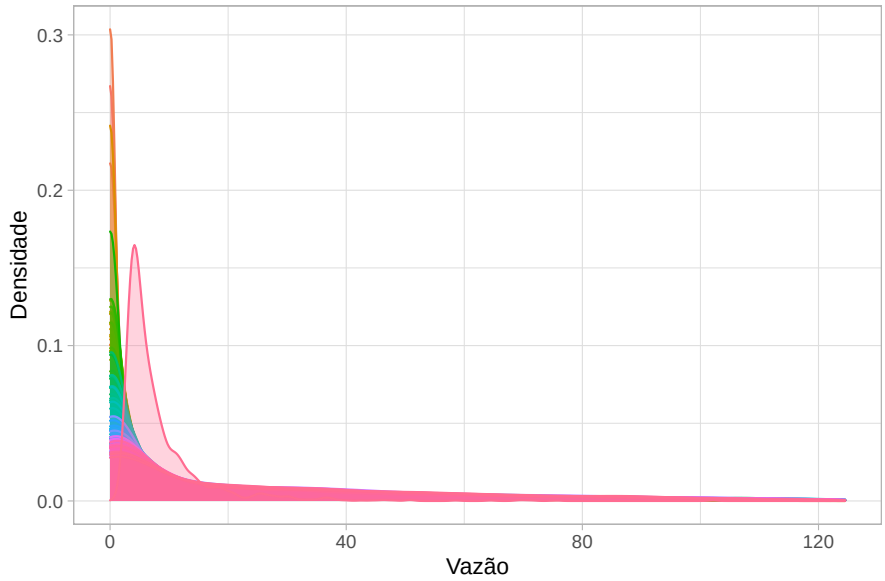


Figura 6: Gráfico de densidades das estações, utilizando até o quantil 90% de todas as vazões.

Na Figura 7 é apresentado a densidade da variável que deseja-se estimar.

Pelas Figuras 6 e 7, nota-se as vazões são bastante assimétricas. Por isso, será aplicado a transformação $\log(x+1)$ em todas as vazões e precipitações para reduzir a assimetria. Na figura 8 são apresentadas todas as vazões e precipitações transformadas das estações preditoras e a vazão da variável resposta.

Nota-se pela 8, que a transformação $\log(x+1)$ conseguiu reduzir bastante a assimetria.

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

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

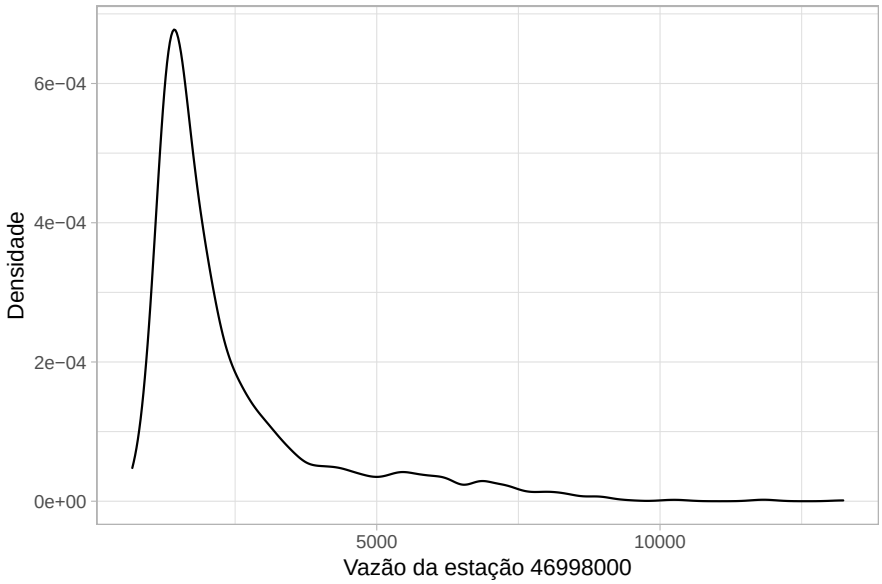
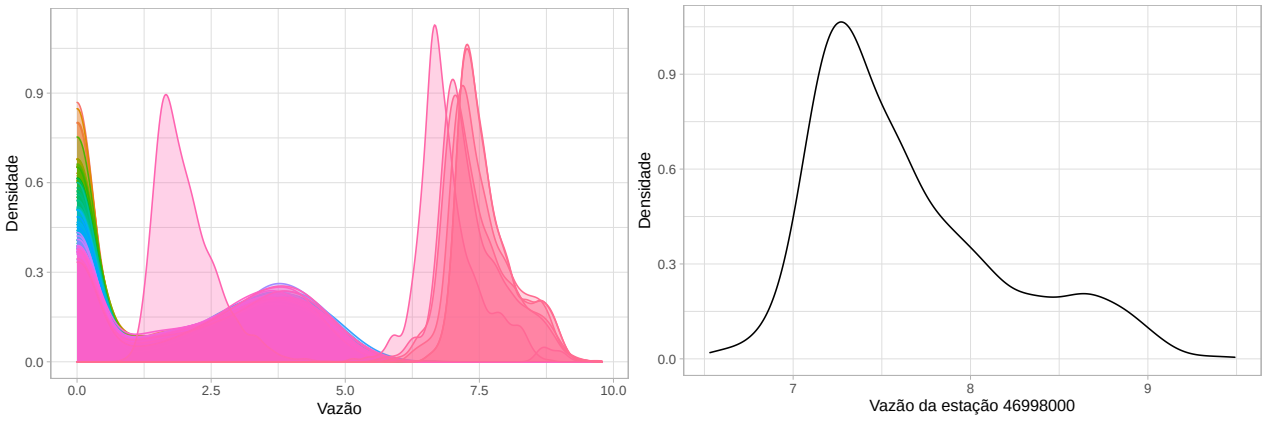


Figura 7: Gráfico de densidade da estação resposta



(a) Todas as estações

(b) Variável resposta

Figura 8: Gráfico de densidade das variáveis transformadas.

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 seja apresentada 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.

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 10 vezes. A medida utilizada para comparar os modelos será o erro médio absoluto.

Os modelos que serão ajustados são:

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

- 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*

Maior parte do modelo de floresta aleatória foi ajustado com os valores padrão da função `randomForest` do pacote com o mesmo nome. Mas para previnir sobreajuste foi definido que a profundidade máxima de cada árvore seria 15.

O modelo de Boosting consegue facilmente sobreajustar os dados de treino. Porém, o pacote XGBoost fornece diversos hiper-parâmetros para controlar o ajuste. Por isso foi utilizada uma taxa de aprendizagem $\eta = 0.01$, profundidade máxima da árvore 4, em cada árvore utilizava apenas 70% das observações e dos preditores. Além disso, a perda mínima para a prolongar a árvore seria 0.6, foi utilizada regularização de primeira ordem e foi utilizado o número de árvores igual a 1000.

Para o modelo de redes neurais, utilizou-se um modelo simples que foi é treinado em 50 épocas em lotes de tamanho 64 (`batch_size`). A estrutura do modelo é:

- camada `dense` com 128 neurônios e ativação sigmóide;
- camada `dense` com 64 neurônios e ativação sigmóide;
- a última camada é uma `dense` com ativação linear;

comparar as estruturas das redes antigas, comparação relu e sigmoide e rede com 2 camadas escondidas ou com 1

Comparação dos modelos

Com a validação cruzada foi obtido diversos erros quadráticos médios para os modelos ajustados. Na Figura 9 essas medidas são apresentadas em um boxplot, com a média sendo apresentada em um ponto em vermelho. E na Tabela 3 é apresentado a média e o desvio padrão estimados para cada modelo.

Nota-se que dos modelos ajustados, o modelo de redes neurais foi o que teve o pior ajuste e a maior variabilidade pela validação cruzada. Os modelos baseados em árvore foram os que apresentaram os erros mais consistentes pela validação cruzada, e os que apresentaram o melhor ajuste no geral foram o modelo de Boosting e o Lasso.

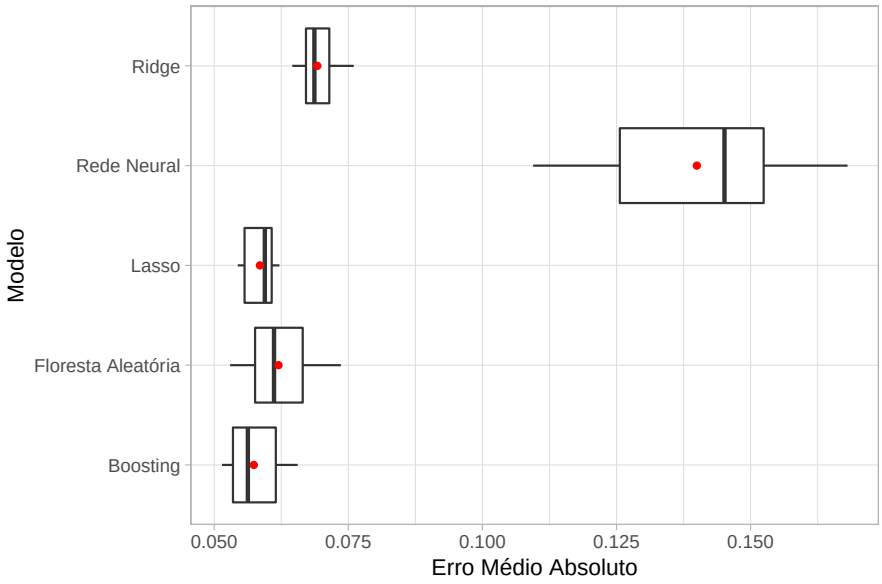


Figura 9: Boxplot do erro médio absoluto por modelo pela validação cruzada.

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

Modelo	Média	Desvio
Boosting	0.0573793	0.0049593
Lasso	0.0585185	0.0029573
Floresta Aleatória	0.0619819	0.0068516
Ridge	0.0692288	0.0034015
Rede Neural	0.1399850	0.0182200

Escolha do modelo

Geralmente para ter melhores predições, as predições dos vários modelos de *machine learning* são misturadas. Existem diversas formas de misturar as predições, desde tomar a média delas, até construir modelos sobre essas predições.

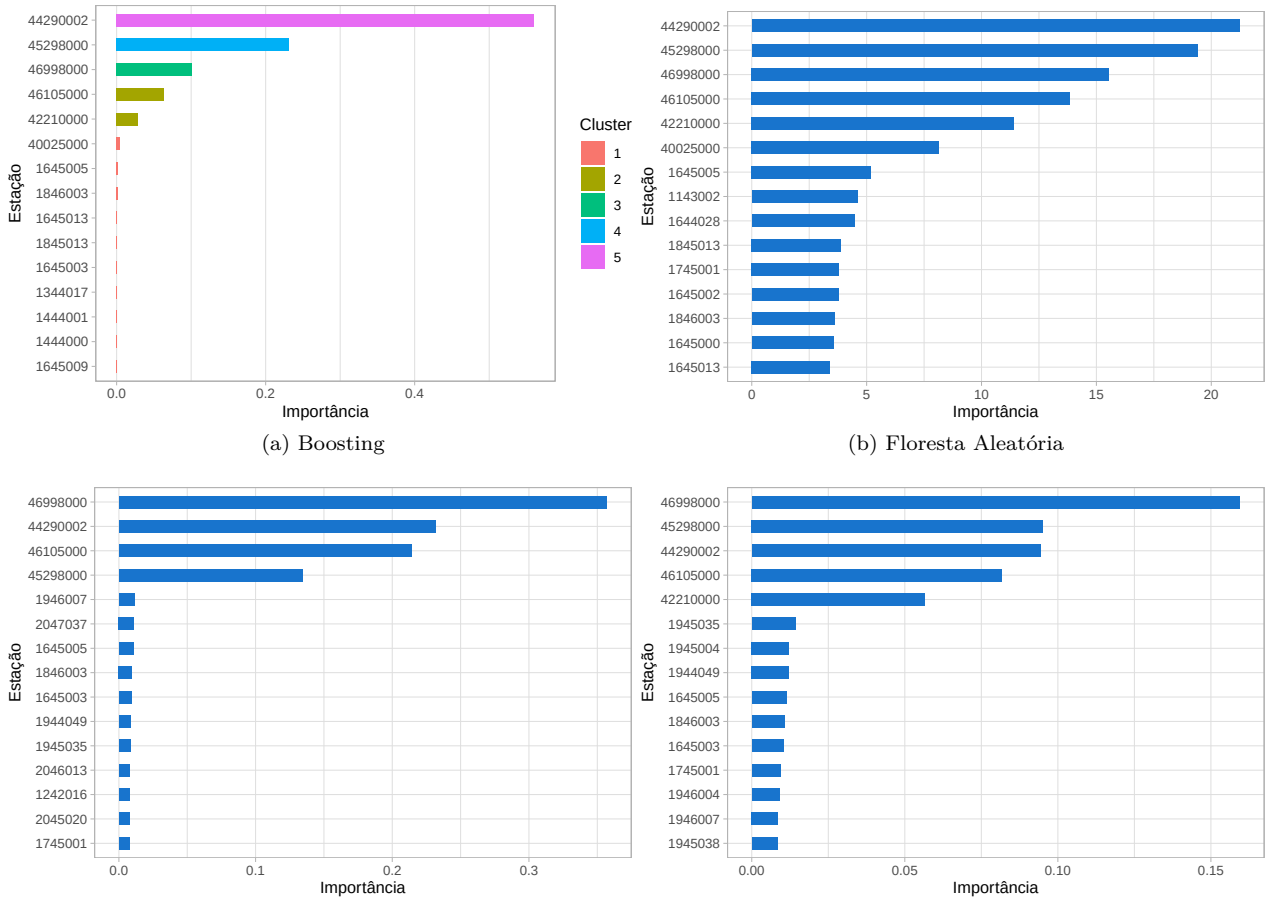
Todos os modelos foram ajustados novamente utilizando todo o conjuto de treino. E para as predições finais, será utilizada a média ponderada de todos os modelos ajustados, em que os pesos serão definidos a partir da performance pela validação.

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:

$$\hat{y} = \exp\{0.25\hat{y}_B + 0.25\hat{y}_L + 0.25\hat{y}_F + 0.2\hat{y}_R + 0.05\hat{y}_{RN}\} - 1,$$

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

```
##      xgb       rf    lasso    ridge       rn  mistura
## 141.8217 158.8751 161.4692 180.0940 385.4719 147.3145
```



(a) Boosting

(b) Floresta Aleatória

(c) Lasso

(d) Ridge

Figura 10: Importância de variável