

# Trabalho de Aplicação

Yugo Oyama NUSP: 9297784

14/11/2021

## Introdução

Este trabalho consiste na aplicação de técnicas estatísticas abordadas no curso de Estatística em Altas Dimensões - 2021 em dois bancos de dados que serão explicados detalhadamente mais a frente.

## Dados de dígitos manuscritos

O banco de dados é composto por:

x\_treino: matriz com as 60000 imagens do conjunto de treino;

y\_treino: vetor com os reais valores dos dígitos escritos nas imagens do conjunto de treino (etiquetas);

Cada linha da matriz x\_treino é uma imagem. Aqui, cada imagem está representada como um vetor de dimensão 1x784. Cada coluna indica o tom de cinza do respectivo pixel da imagem (entre 0 - preto e 255 - branco).

## Objetivo

O objetivo desta análise é propor um modelo que consiga realizar boas previsões para a qual número corresponde cada uma das imagens.

## Técnicas

Para esse projeto serão usados modelos lineares com regularização, modelos baseados em árvores, e redes neurais.

Para comparar os modelos entre si, será utilizada a acurácia e ao final de cada método testado, será adicionado o resultado a uma tabela comparativa.

Por motivos de tempo de processamento, apenas o modelo escolhido de cada categoria será apresentado no relatório.

## Parâmetros

Para o ajuste do modelo, foi definido que o conjunto de treino seria dividido em conjunto de treino e validação na proporção 7,5:1,5, ou seja, 75% do conjunto originalmente de treino foi usado para validação. Foi definida também uma semente por questão de reprodutibilidade (12345).

## Lasso

Para criar um modelo de regressão com a penalização lasso, foi utilizada a biblioteca glmnet. Com ela, dentro da amostra de treino, foi ajustado o modelo de classificação com penalização lasso utilizando validação cruzada. Em seguida, foi testado o modelo obtido no conjunto de validação e calculado o respectivo erro dentro e fora

Para o ajuste do modelo, inicialmente testou-se os definir para a função os valores (0.01, 0.1, 1, 2, 10, 25, 50, 75, 100) e em seguida testou-se definir que o modelo escolhesse 30 lambdas diferentes.

Existem dois lambdas que são popularmente usados: o que minimiza o erro gerado pela validação cruzada e o no qual o erro não ultrapassa um desvio padrão do melhor modelo. Com isso, foram testados modelos com cada um dos lambdas e calculados os erros dentro e fora respectivos de cada um.

O modelo cujos valores de lambda foram escolhidos pelo `cv.glmnet` apresentou resultados significativamente melhores.

O cálculo de previsão foi realizado diretamente do modelo resultante da validação cruzada conforme recomendado na documentação do pacote `glmnet` por questão de convergência e otimização.

```
##                modelo acuracia_dentro acuracia_fora
## 1          Lasso - Lambda minimo          0.9364          0.9219
## 2 Lasso - Lambda 1 desvio padrao          0.9342          0.9207
```

Percebemos que usar o modelo com lambda mínimo ou o modelo com lambda 1 desvio padrão não resulta em muita diferença na acurácia.

O ajuste do modelo demorou algumas horas para ser realizado.

## Modelo de Árvores

Foi escolhido o modelo de Floresta Aleatória já que dessa forma existe variabilidade na escolha das variáveis explicativas e na escolha das observações para a construção de cada árvore. Essa variabilidade por sua vez pode levar a uma melhor predição no conjunto de validação e no de teste.

Para realizar o ajuste do modelo, foi utilizado a função `randomForest` do pacote `randomForest`.

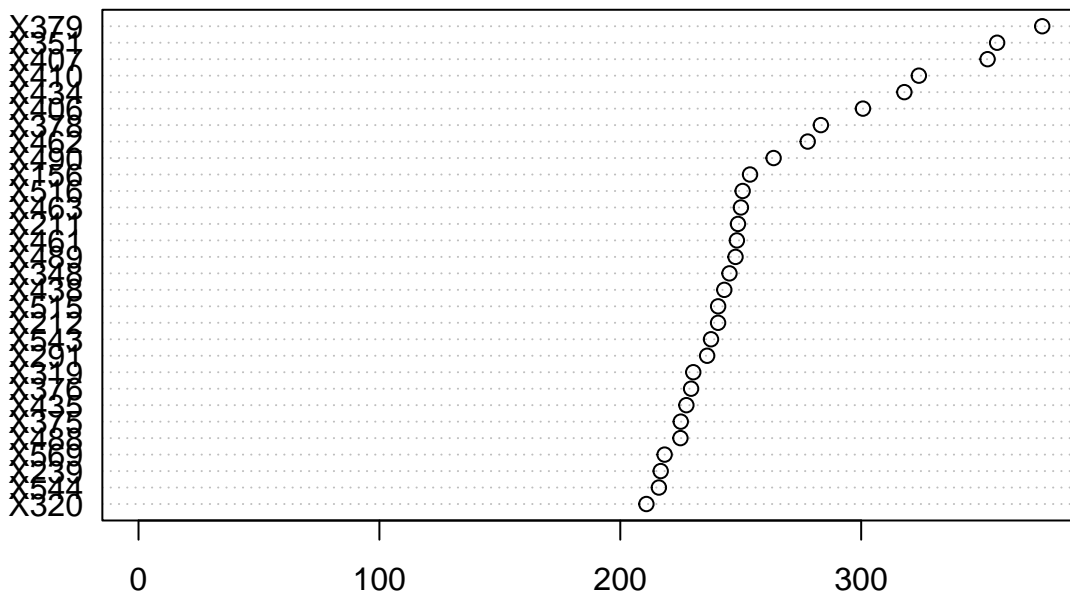
A seguir, podemos ver os cinco pixels mais importantes e as cinco menos importantes para a predição.

```
##      MeanDecreaseGini
## X379          375.1617
## X351          356.4352
## X407          352.4710
## X410          323.9503
## X434          317.9197

##      MeanDecreaseGini
## X780              0
## X781              0
## X782              0
## X783              0
## X784              0
```

Percebemos, como esperado, que os pixels mais periféricos (mais próximos às bordas das imagens) são os que menos contribuem para o ajuste de previsão e os pixels mais centrais são os que mais contribuem.

Ainda, é possível visualizar melhor no gráfico a seguir a importância de cada pixel.



Pelo gráfico, percebemos que o ajuste do modelo não é feito a partir de alguns pixels isolados e sim a partir, principalmente, do conjunto de pixels mais próximos do centro.

##	modelo	acuracia_dentro	acuracia_fora
## 1	Lasso - Lambda minimo	0.9364	0.9219
## 2	Lasso - Lambda 1 desvio padrao	0.9342	0.9207
## 3	Floresta aleatoria	1.0000	0.9698

Percebemos que o modelo de floresta aleatória obteve acurácia maior que os modelos de regressão com penalidade Lasso tanto na acurácia dentro como na acurácia fora.

Impressionantemente, a acurácia dentro foi de 100%, ou seja, não errou nenhuma predição.

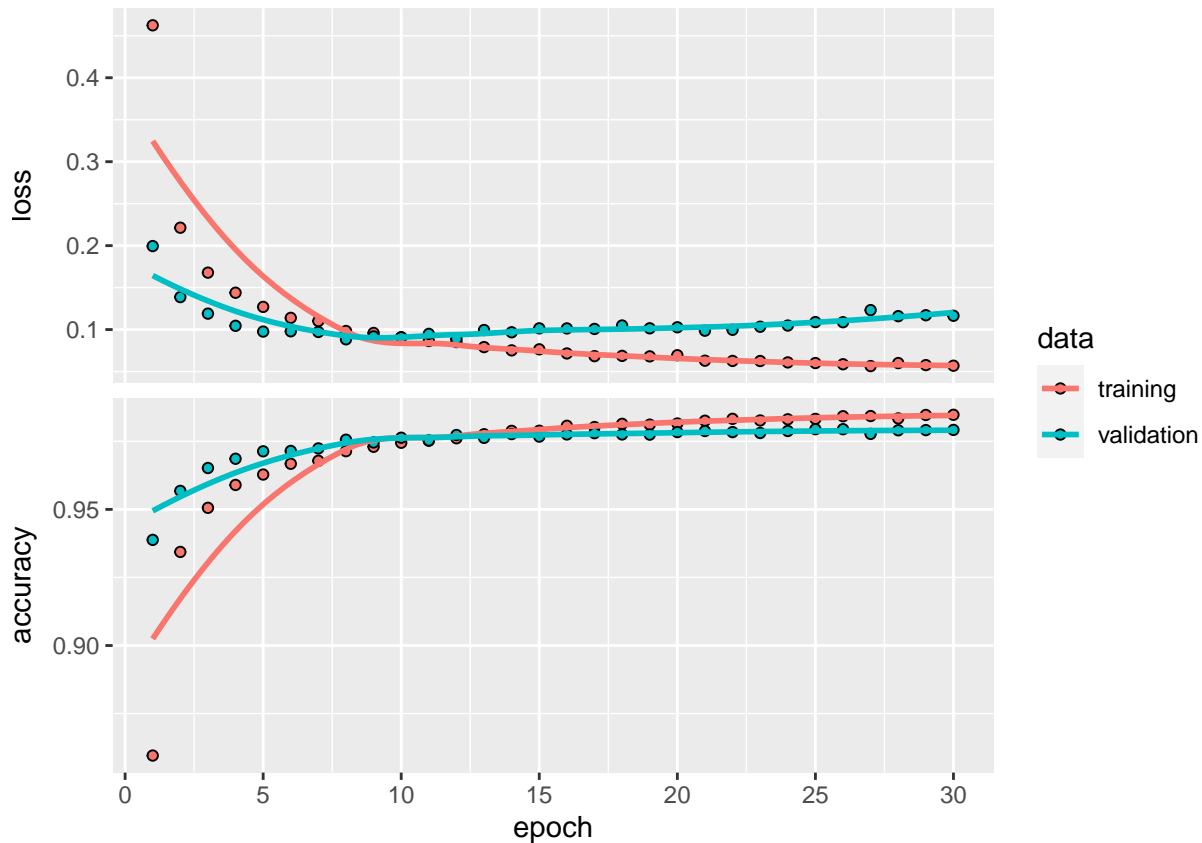
### Redes neurais

Para o modelo de redes neurais, foi considerado um modelo com duas camadas ocultas com 256 e 128 unidades respectivamente com função de ativação ReLU.

Como temos mais de uma classificação, foi utilizada a função softmax como função de ativação na camada de saída.

Ainda, foi considerada supressão de 40%, 40 epochs e tamanho de batch 128 de forma a tentar otimizar o tempo de processamento do modelo.

```
## Loaded Tensorflow version 2.5.0
```



##	modelo	acuracia_dentro	acuracia_fora
## 1	Lasso - Lambda minimo	0.9364	0.9219
## 2	Lasso - Lambda 1 desvio padrao	0.9342	0.9207
## 3	Floresta aleatoria	1.0000	0.9698
## accuracy	Redes Neurais	0.9974	0.9792

O modelo de redes neurais também levou bem menos tempo que o modelo de regressão com penalidade lasso para ser ajustado. Ainda, obteve a maior acurácia fora (conjunto de validação) e acurácia bastante elevada dentro apesar de não tão alta como no modelo de floresta aleatória.

## Resultados

Como o objetivo desse projeto é prever o número em um conjunto de dados nunca antes visto no modelo, a acurácia fora tem um peso maior que a dentro. Dessa forma, o modelo de redes neurais foi o escolhido para realizar as previsões do conjunto de teste.

Ainda, como os modelos testados já obtiveram acurácia bastante satisfatória, foi considerado que não havia necessidade de se testar mais modelos diferentes ou variações dos apresentados.

Para o conjunto de dados mnist, o modelo de redes neurais foi o melhor modelo, seguido pelo de floresta aleatória e por último o modelo de regressão logística com penalidade LASSO, sendo que o último não se mostrou um bom modelo tanto em tempo de processamento quanto em acurácia.