Minicurso de Ciência de Dados

Aula 2 - Validação Cruzada, Overfitting e Underfitting

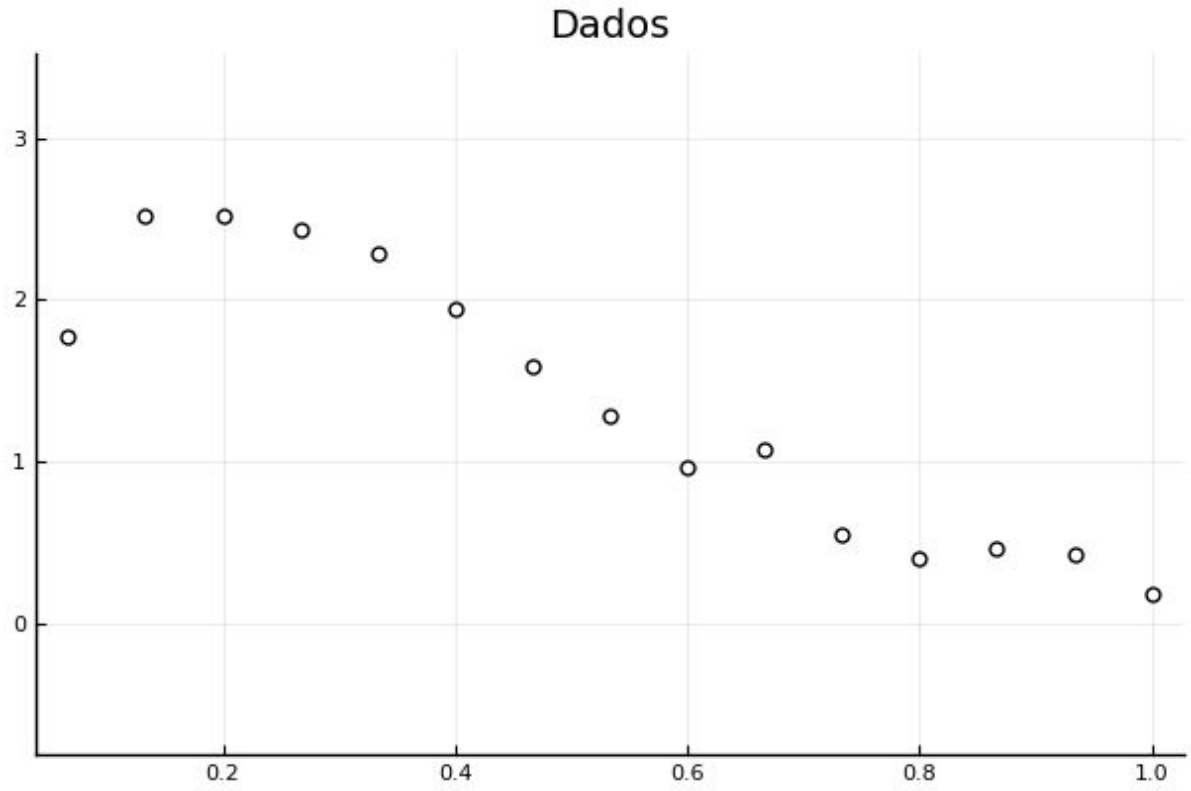
Abel Soares Siqueira

3 de Fevereiro de 2020

I CiDWeek

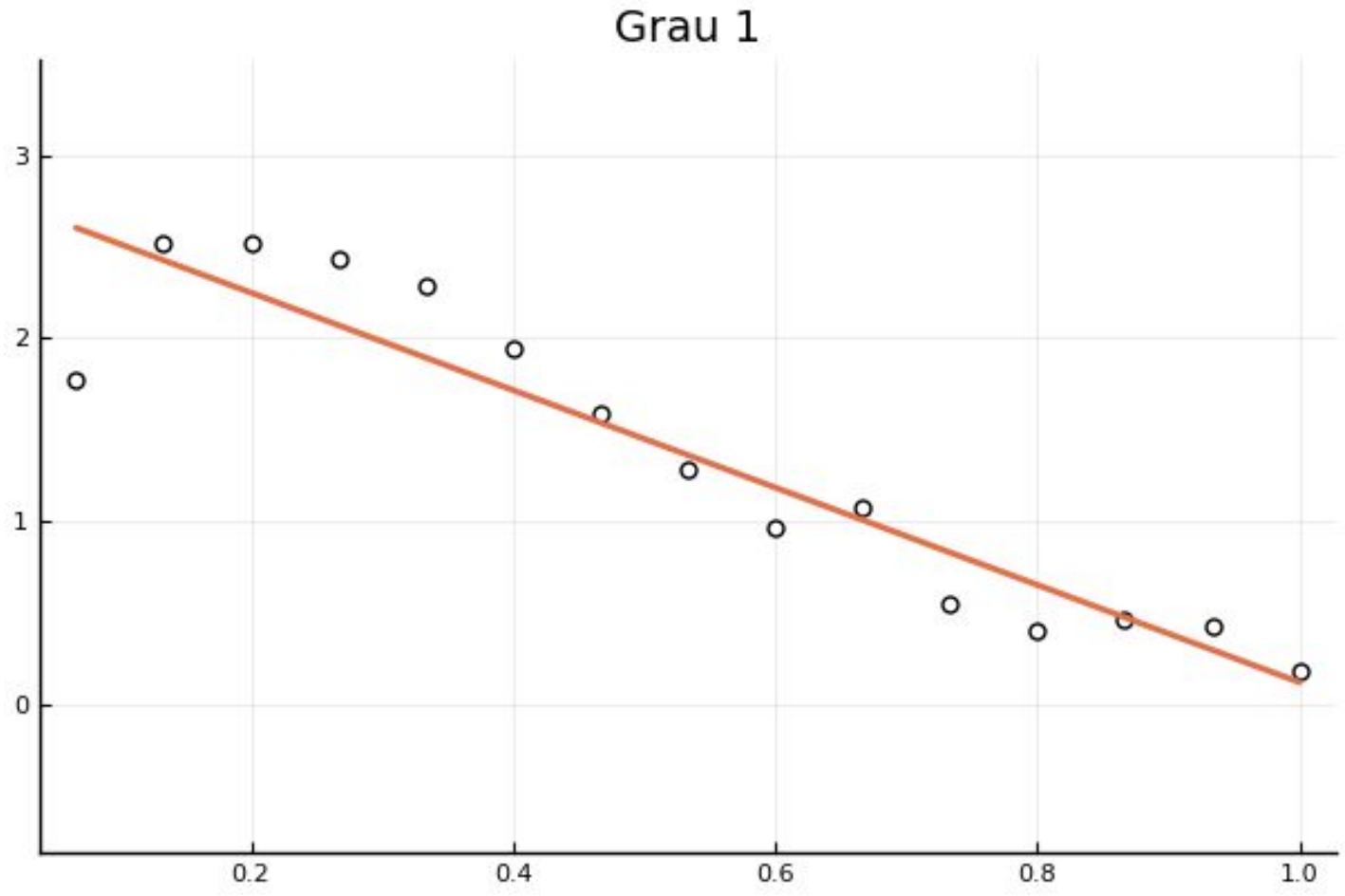
CiDAMO

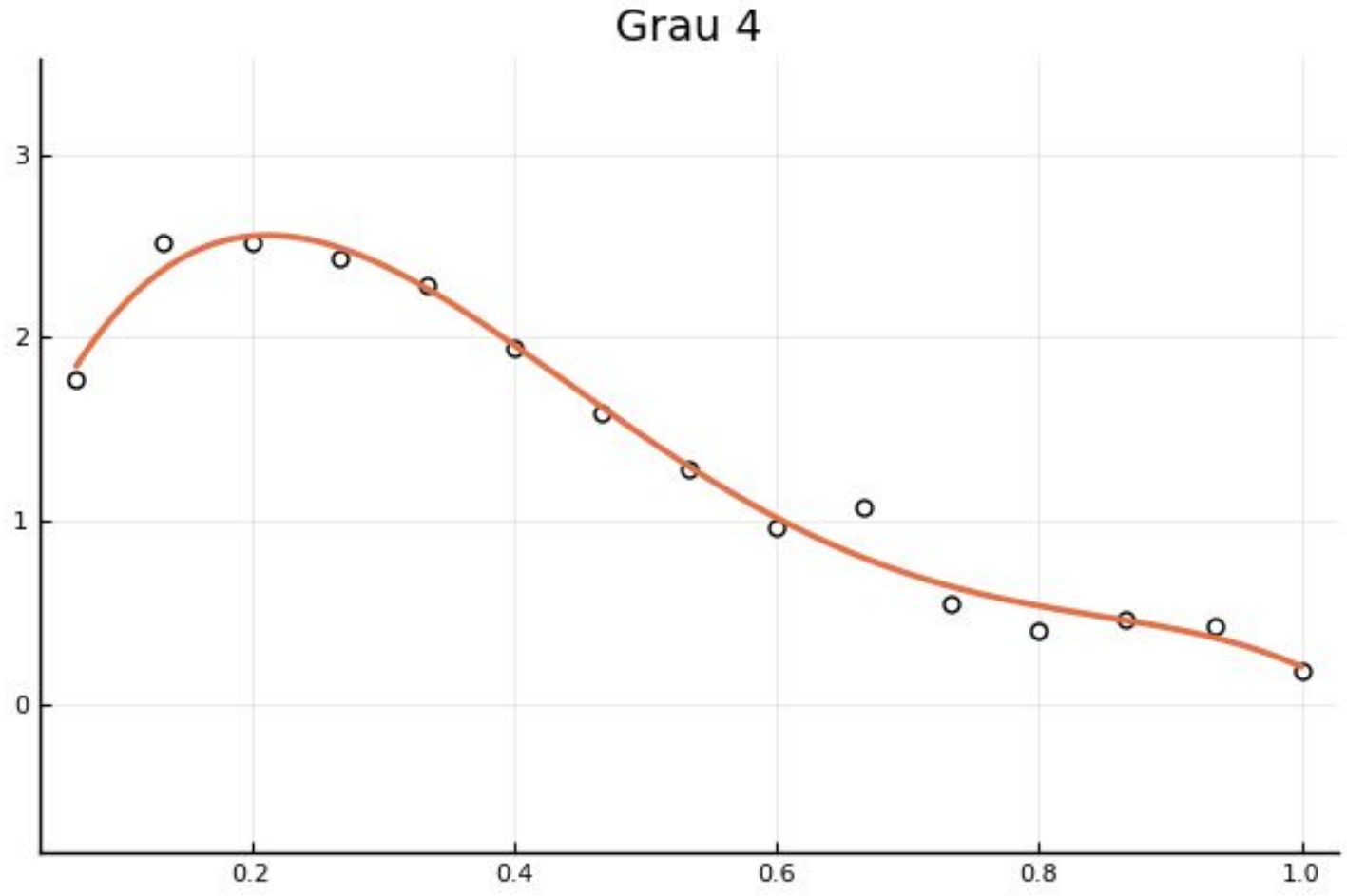
Seleção do modelo

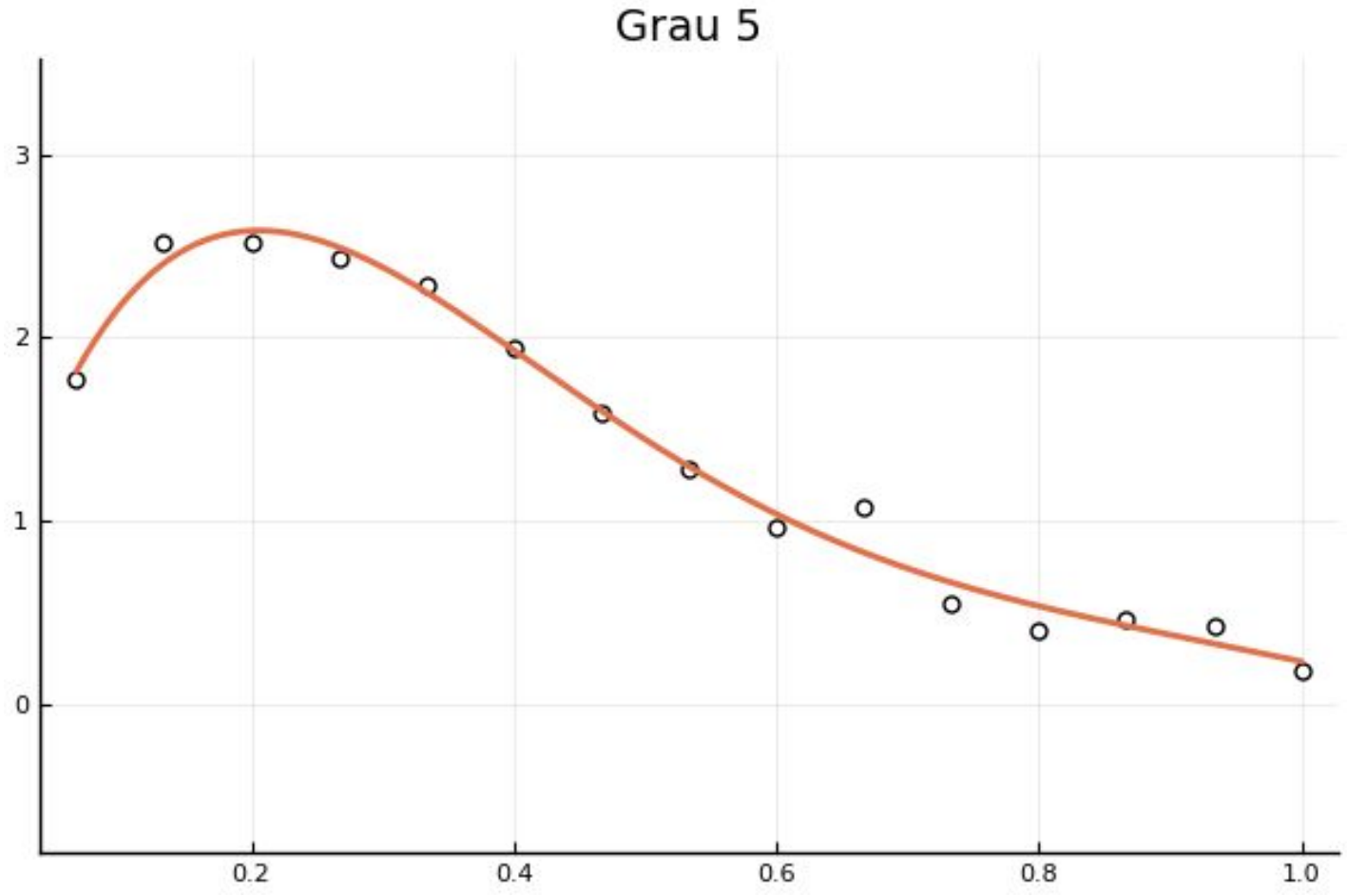


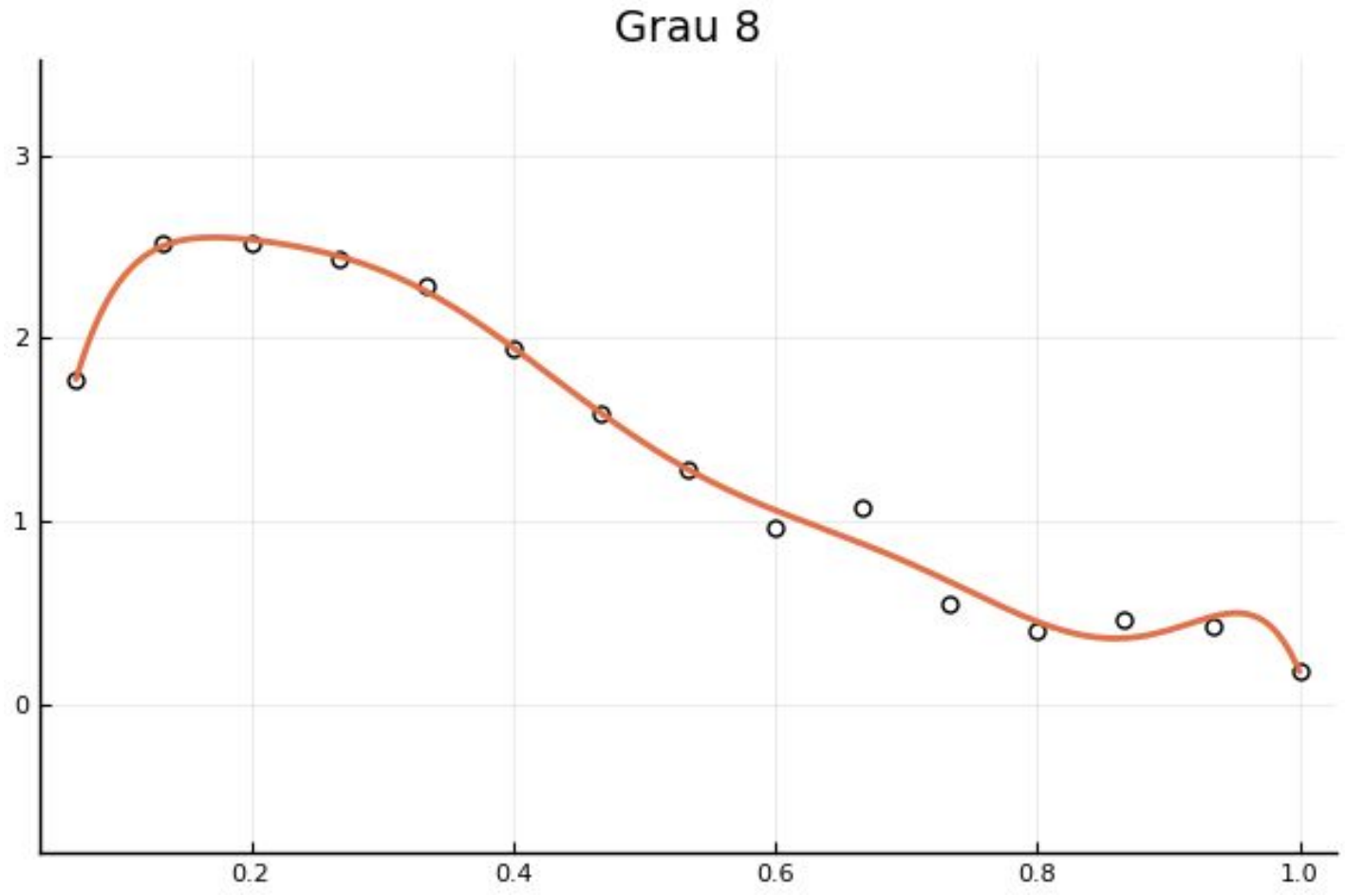
Seleção do modelo

- Ajustar os dados dá uma descrição **(inferência)**.
- Com a hipótese de como é a cara do modelo (linear, polinomial, etc.), encontramos os parâmetros que melhor descrevem os dados.
- Se não temos hipóteses e queremos fazer **predição**, i.e. utilizar o modelo para prever novos valores, o que fazer?
- Cada família de modelo usada tem aquele com melhores parâmetros









Seleção do modelo

CiDAMO

- O modelo linear não explica muito bem os dados, mas pega a ideia geral, a tendência.
- O modelo de grau 14 explica 100% dos dados, mas perde o fio.
- Como encontrar o equilíbrio?

Seleção do modelo

CiDAMO

- Alice te contrata para fazer a previsão da demanda do mês seguinte
- Seu modelo mais simples (reta) têm um R^2 de 25% - Alice não gosta
- Seu modelo mais complexo têm um R^2 de 100% - Alice fica feliz
- Passa o mês, ambos modelo acertam por volta de 20%

Seleção do modelo

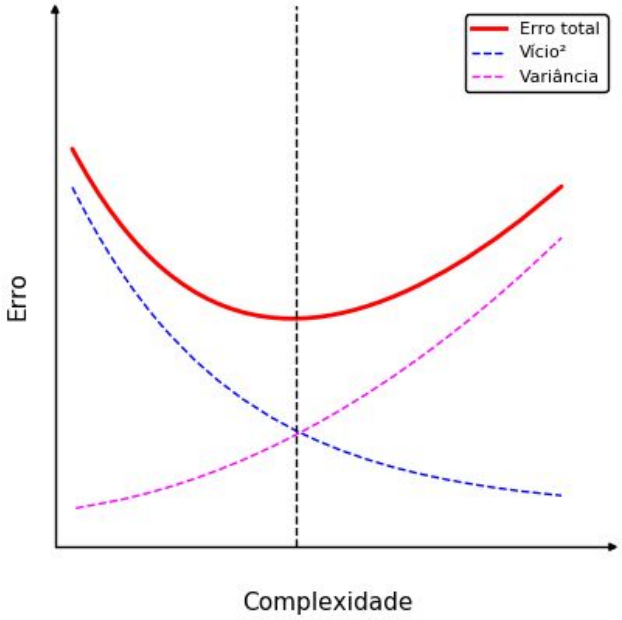
- Um modelo intermediário, acerta 60% nos dados anteriores e no mês novo - **como encontrar esse modelo?**
- Poderíamos ver o acerto no novo mês - mas ele está no futuro
- Podemos simular essa situação

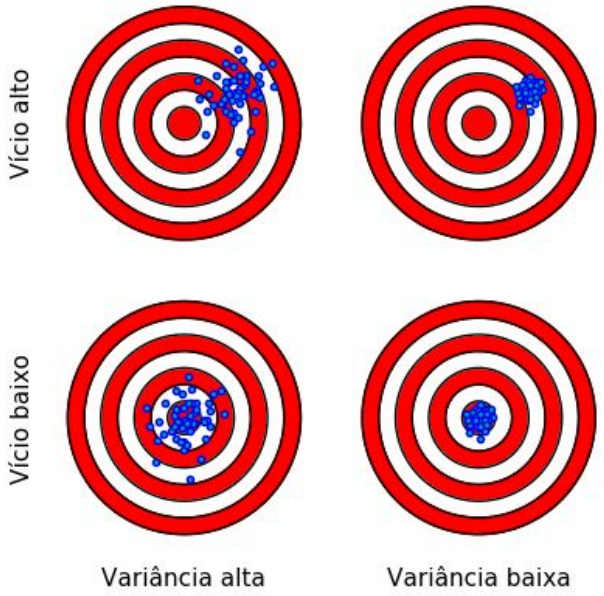
Seleção do modelo

- O que fazemos é obter dois conjuntos de dados: um para **Treino** e um para **Teste**
- Os dados de treinamento são usados para obtenção do modelo, e os dados de teste são usados para verificar a generalização do modelo
- Baixa complexidade não aprende direito - **Underfitting**
- Alta complexidade pega ruído - **Overfitting**

Dilema Vício-Variância

- **Erro** = (Vício/Viés/"Defeito")^2 + Variância + Erro irredutível

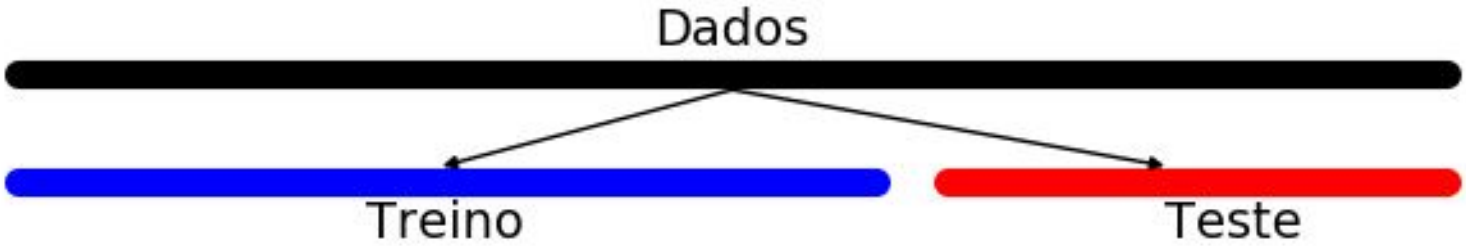




Validação cruzada e Bootstrap

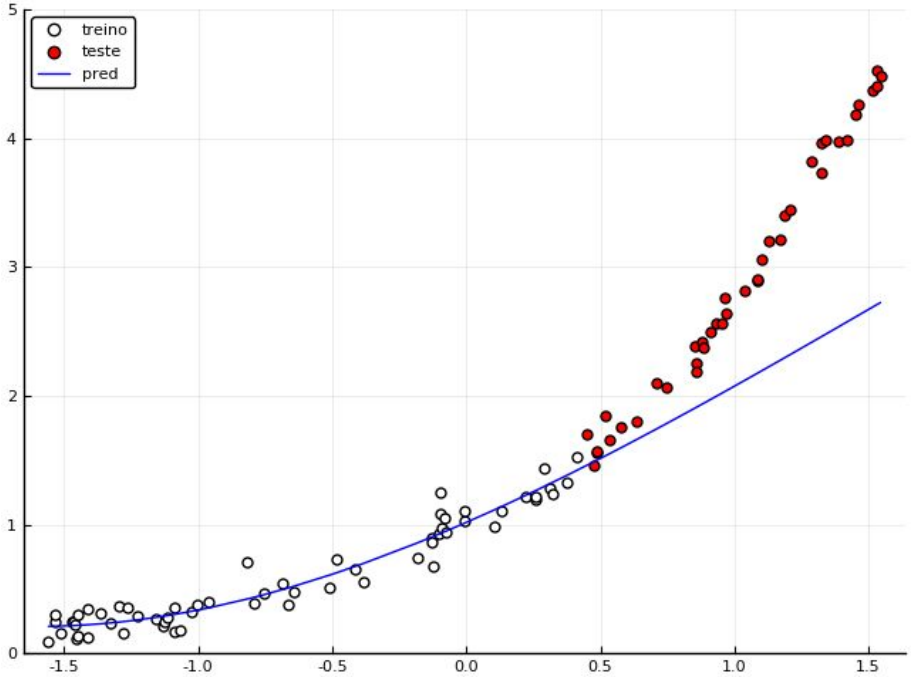
- Existem algumas maneiras de criar conjuntos de treino e teste
- A **validação cruzada** consiste em separar o conjunto
  - Estratégia **Holdout**: escolhe aleatoriamente uma partição
  - Estratégia **Leave-one-out** ou exaustiva: testa com cada elemento de teste
  - Estratégia **K-fold**: escolhe aleatoriamente k partições e cada uma é teste
- **Bootstrap** reutiliza elementos do próprio conjunto para criar um conjunto de testes - a seleção é aleatória e com repetição

Holdout

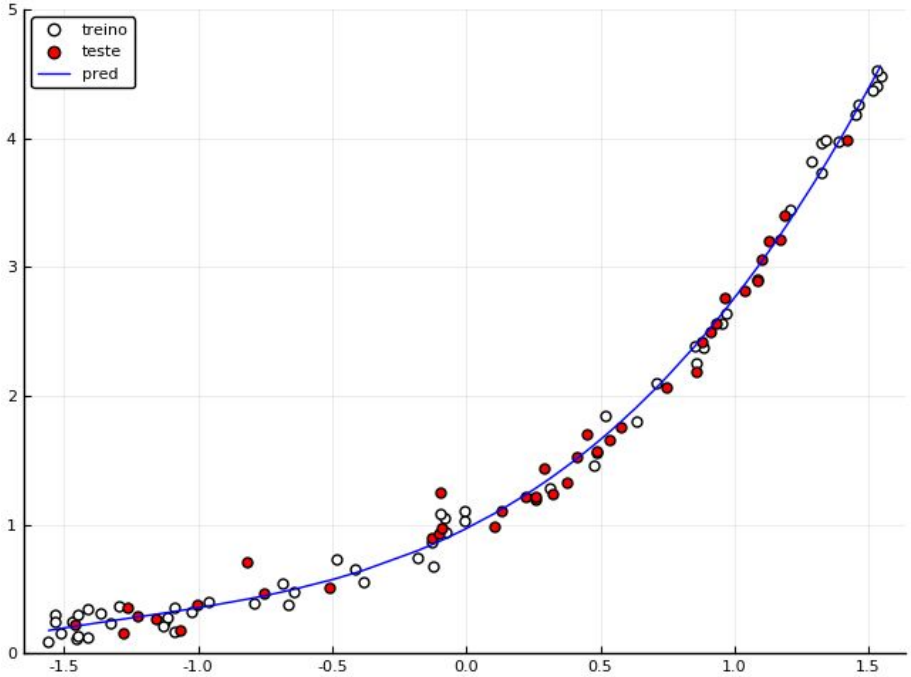


- A quantidade de dados de treino é menor logo a variância é maior, na tentativa de diminuir o viés
- Escolha aleatoriamente

Holdout

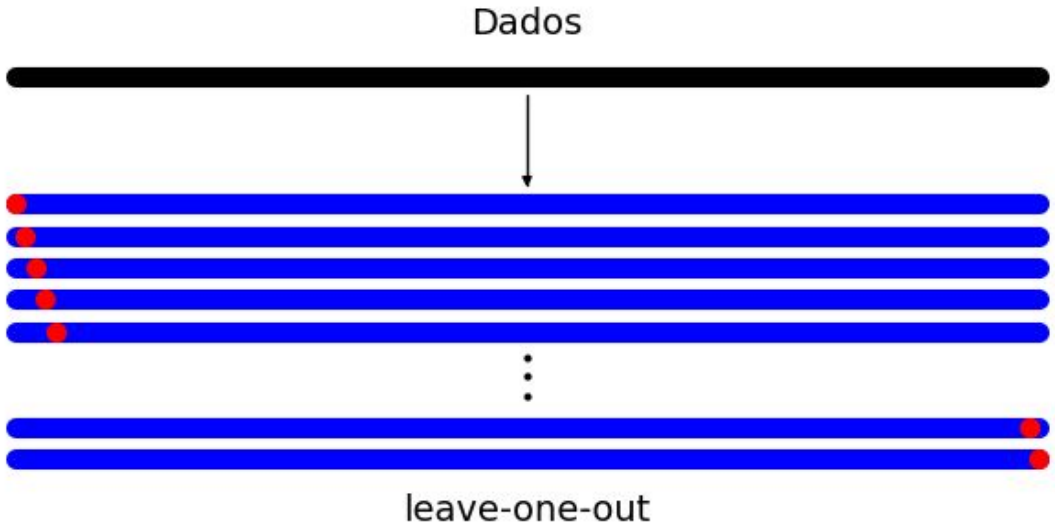


Holdout



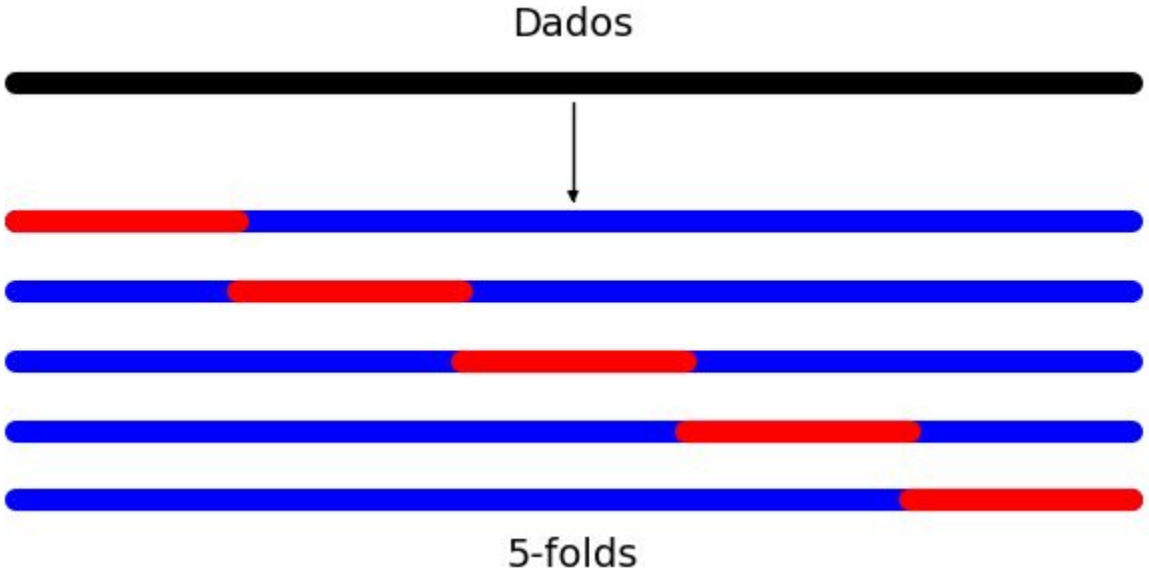
Leave-one-out

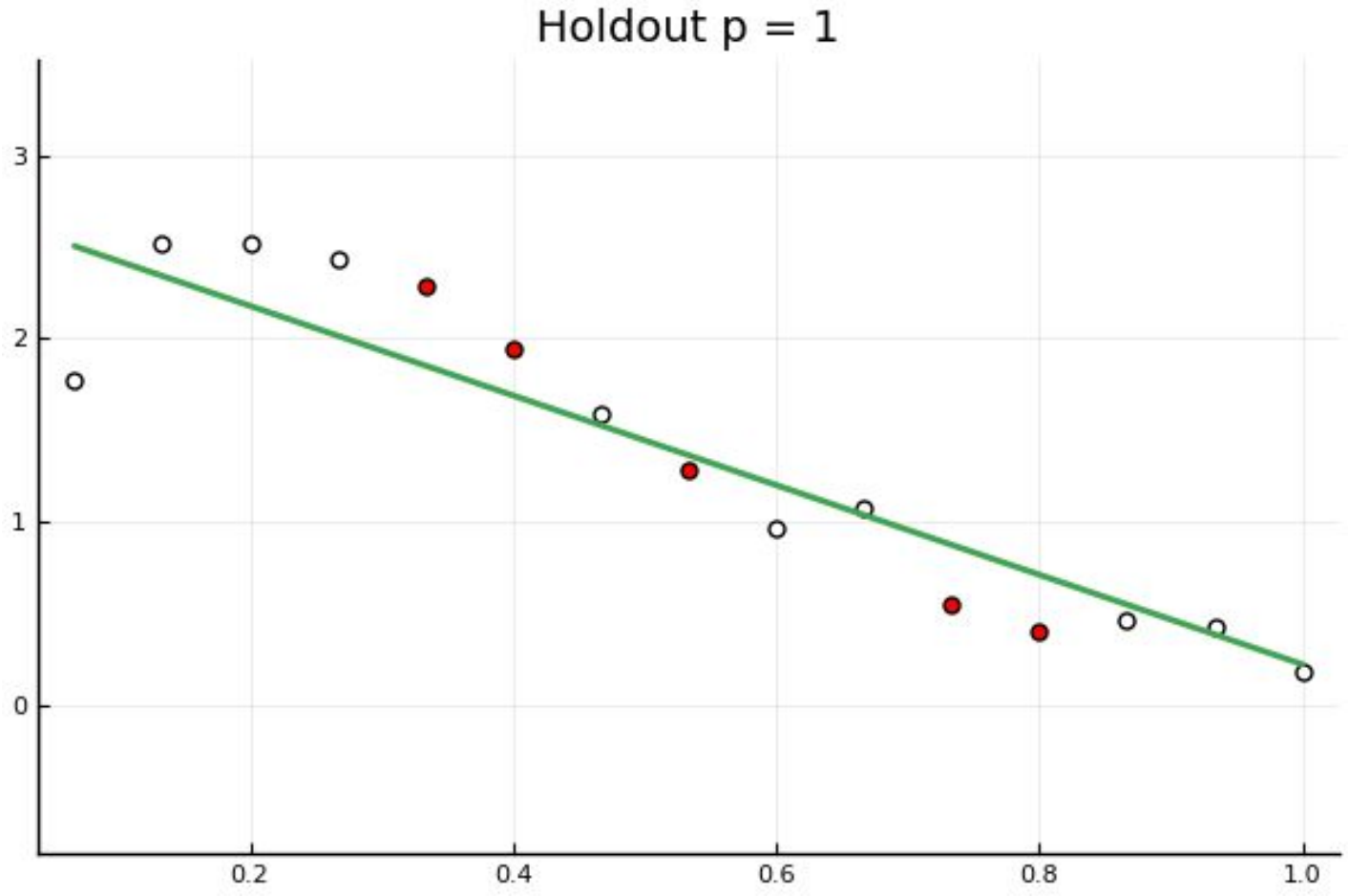
- LOOCV faz n separações, e usa-se a média
- Todos os dados são usados, e usa-se o máximo no treino, logo o vício e variância são minimizados, mas é lenta

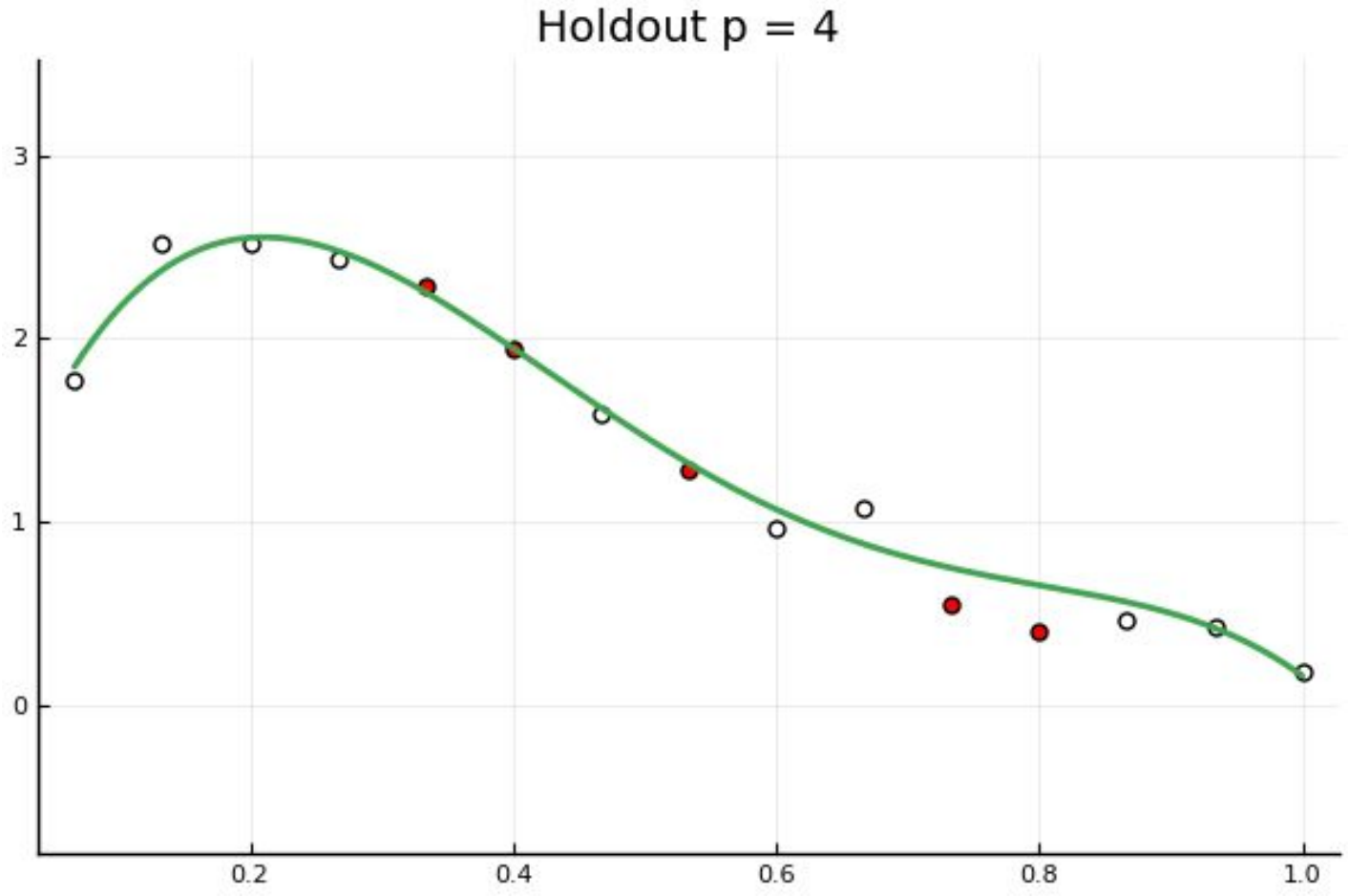


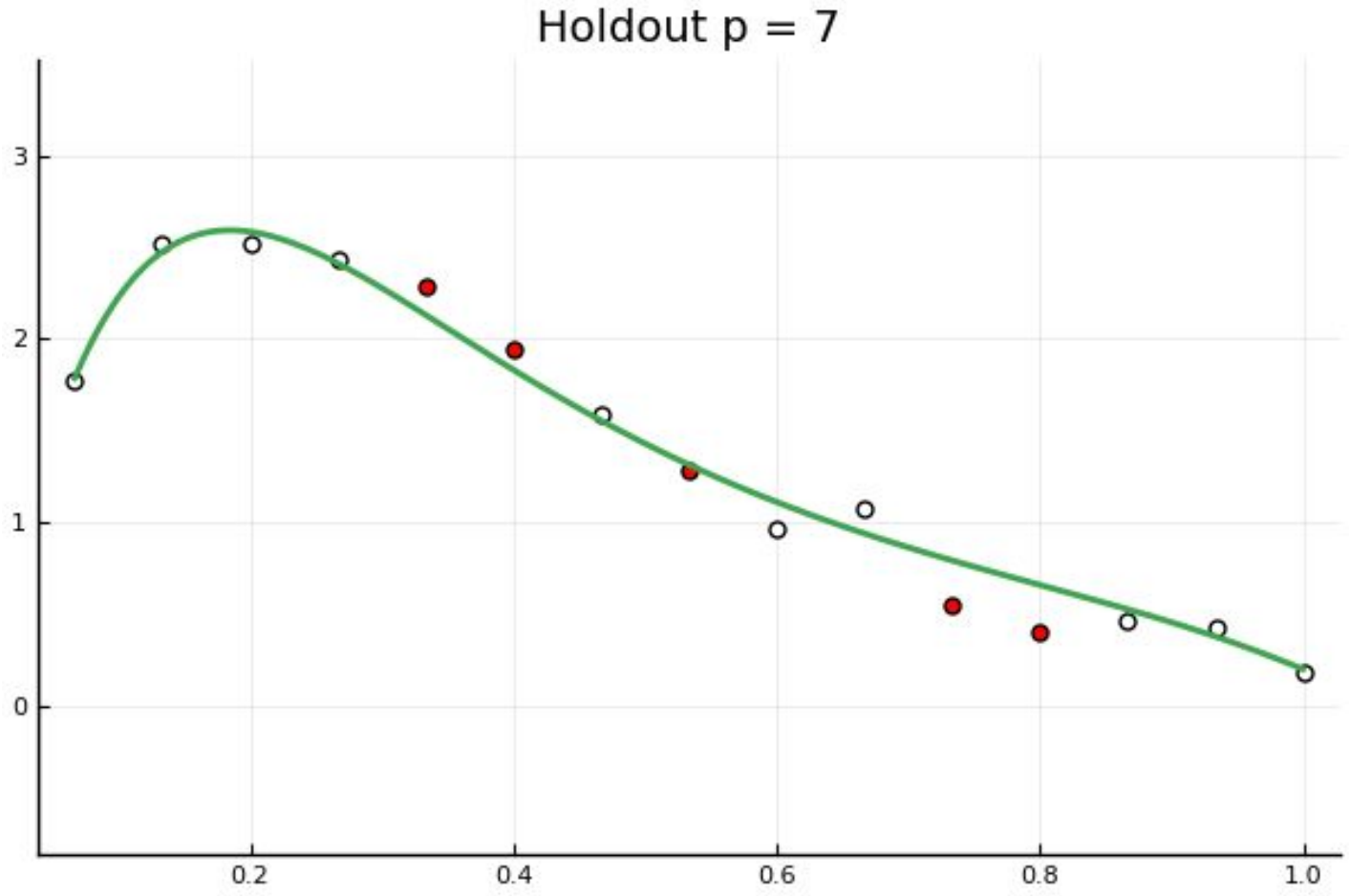
K-fold

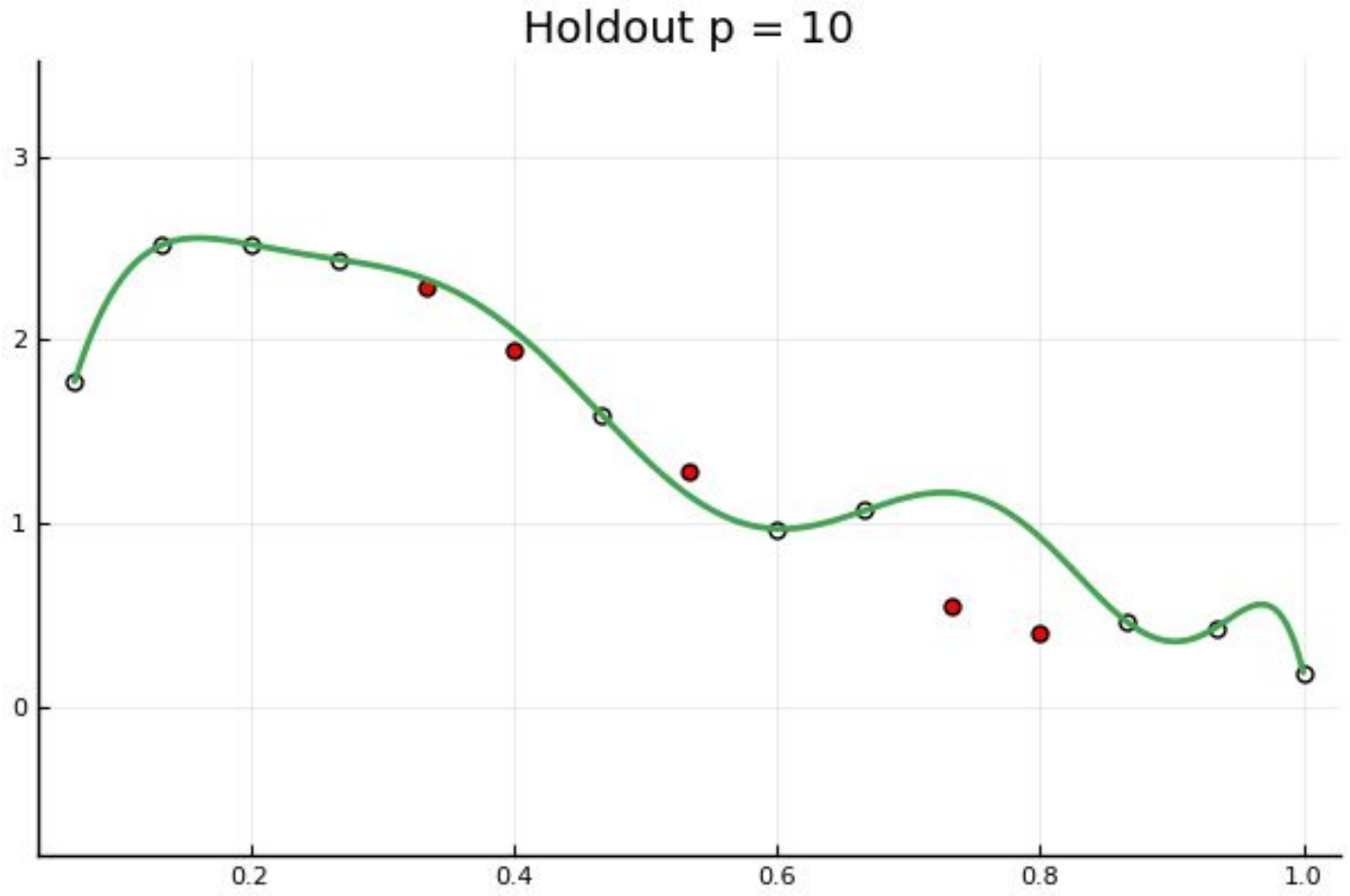
- K-fold faz k separações (5 ou 10 são comum), e usa-se a média
- Equilíbrio entre velocidade e diminuição de variância e vício

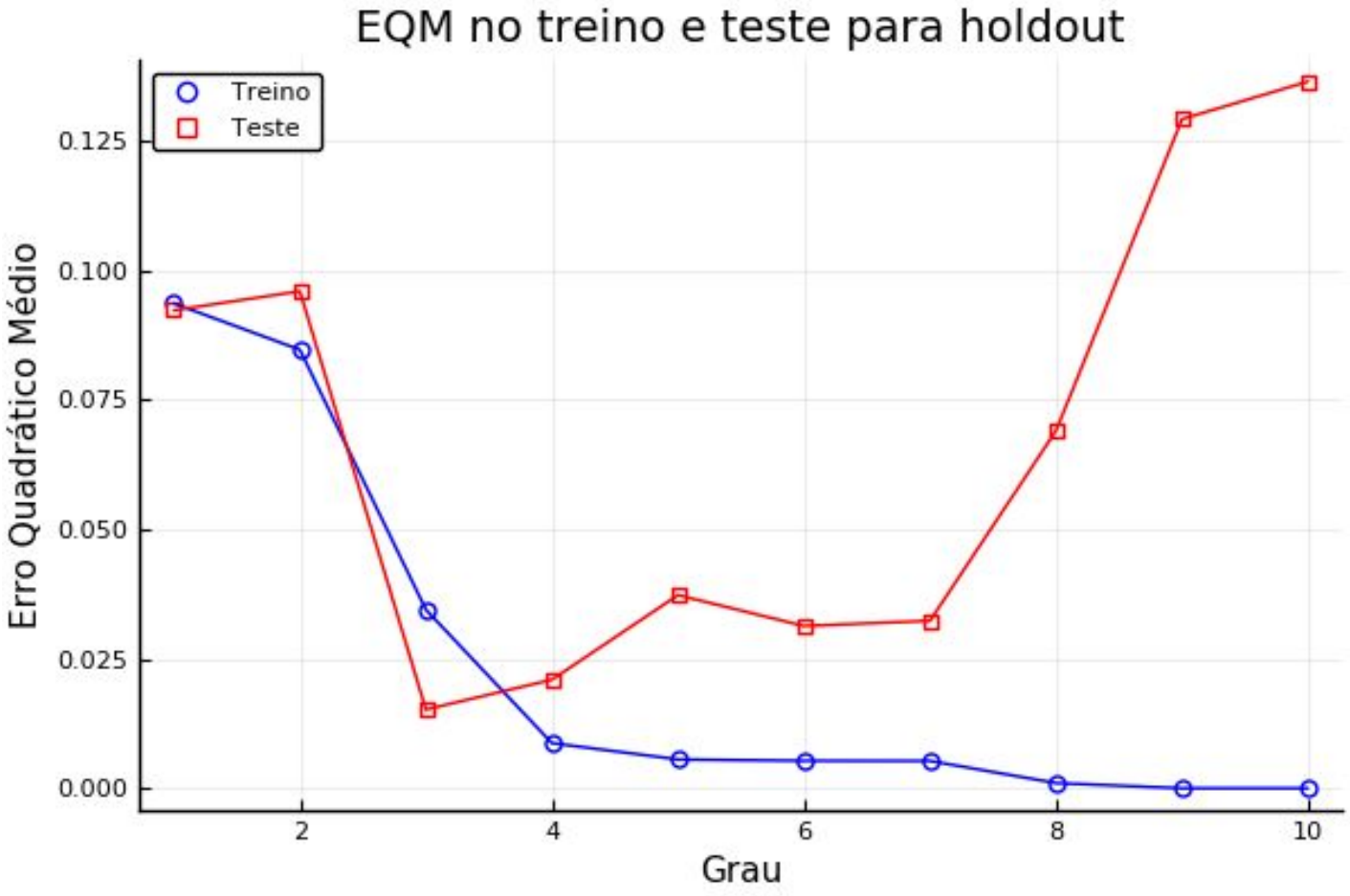


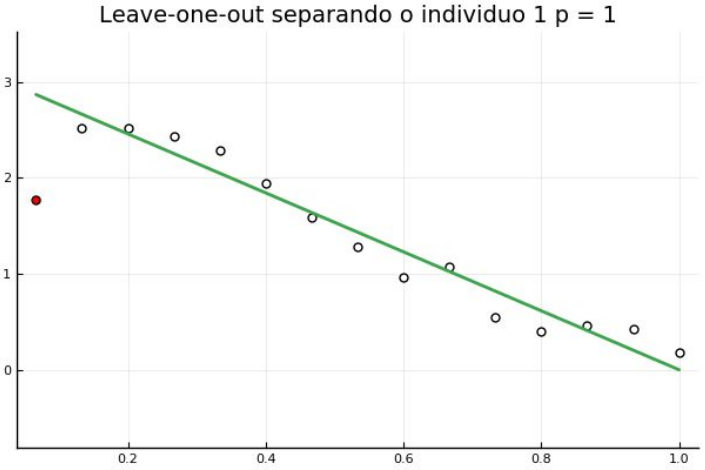


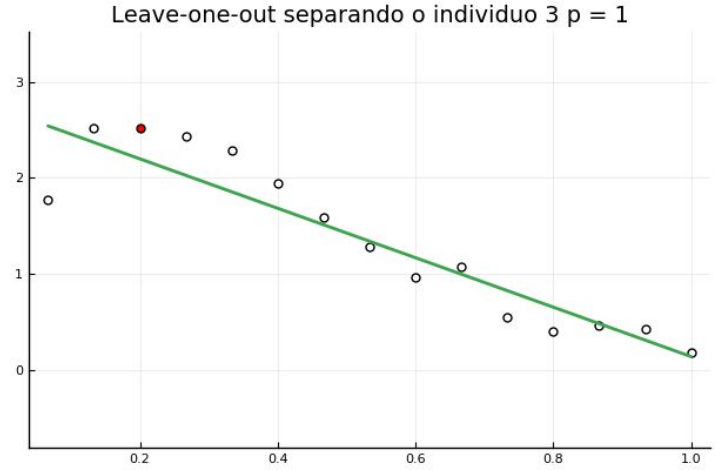


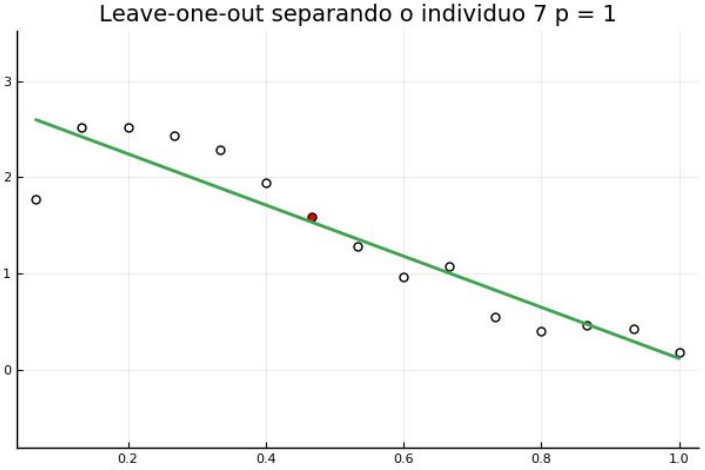


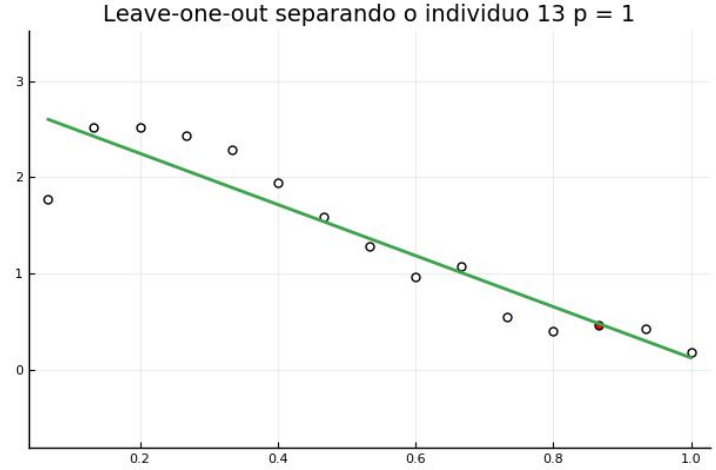


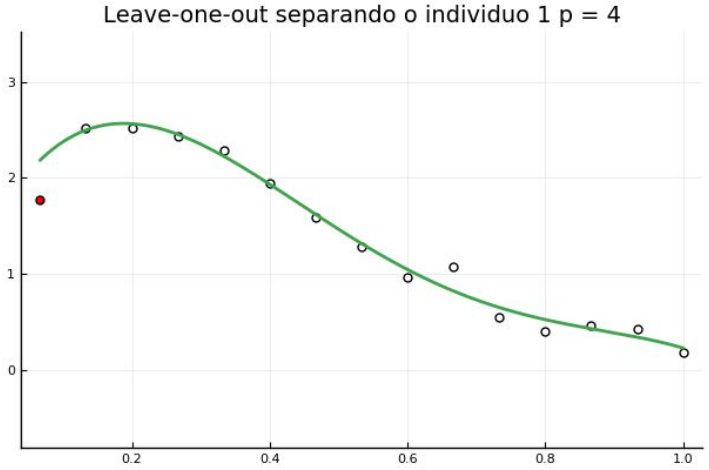

EQM no treino e teste para holdout
Treino
Teste
Erro Quadrático Médio
0.125
0.100
0.075
0.050
0.025
0.000
2
4
6
8
10
Grau

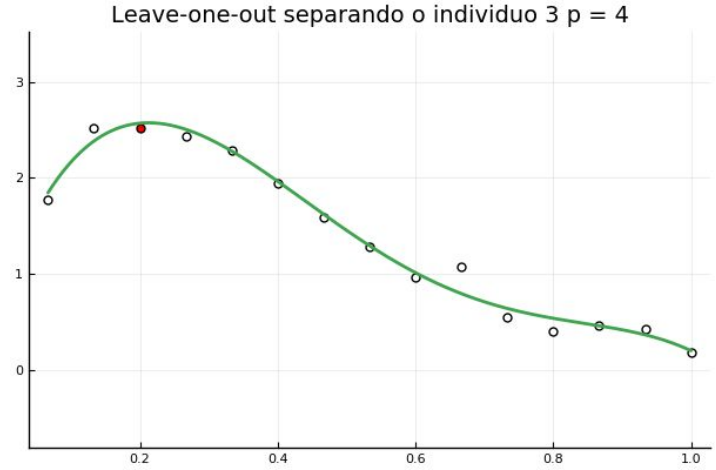

Leave-one-out separando o individuo 1 p = 1
3
2
1
0
0.2
0.4
0.6
0.8
1.0

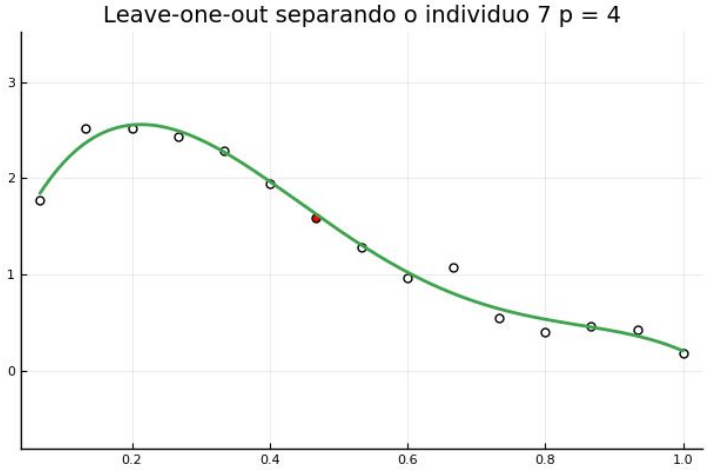

Leave-one-out separando o individuo 3 p = 1
3
2
1
0
0.2
0.4
0.6
0.8
1.0

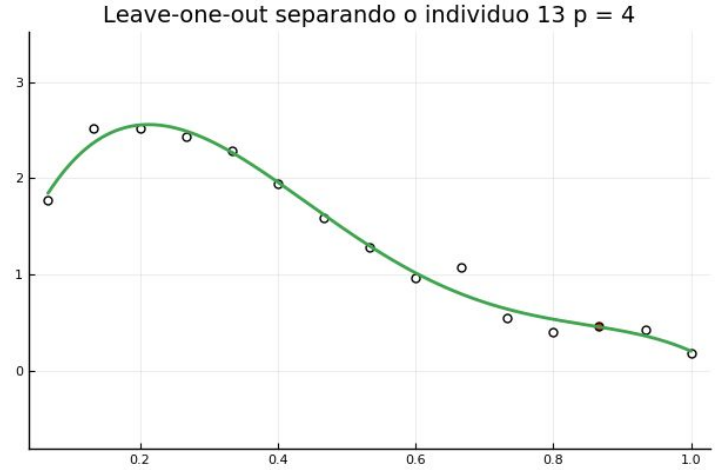

Leave-one-out separando o individuo 7 p = 1
3
2
1
0
0.2
0.4
0.6
0.8
1.0

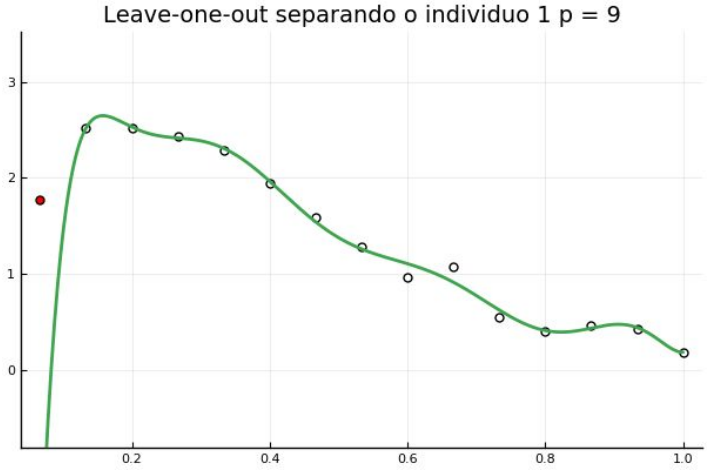

Leave-one-out separando o individuo 13 p = 1
3
2
1
0
0.2
0.4
0.6
0.8
1.0

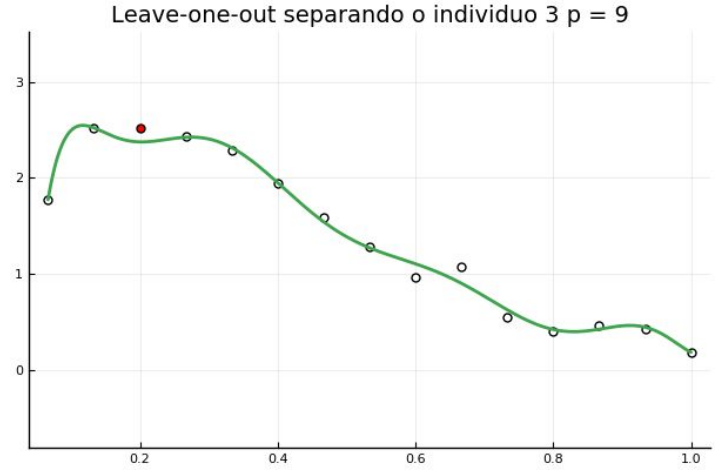

Leave-one-out separando o individuo 1 p = 4
3
2
1
0
0.2
0.4
0.6
0.8
1.0

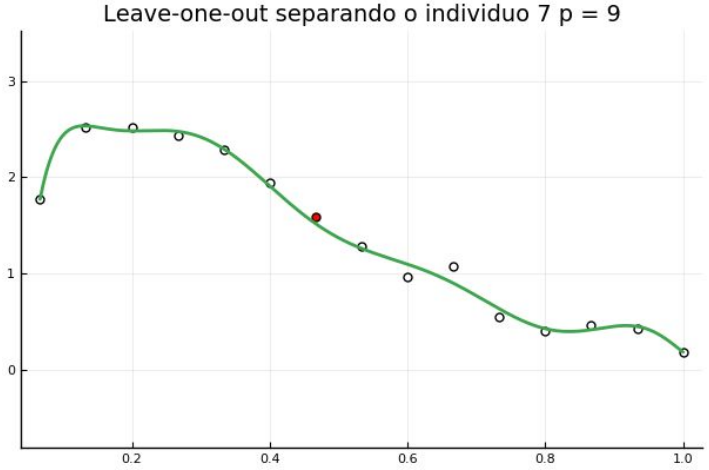

Leave-one-out separando o individuo 3 p = 4
3
2
1
0
0.2
0.4
0.6
0.8
1.0

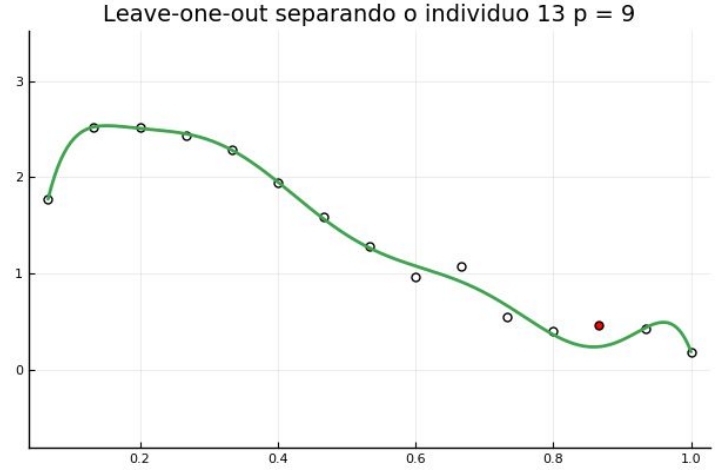

Leave-one-out separando o individuo 7 p = 4
3
2
1
0
0.2
0.4
0.6
0.8
1.0

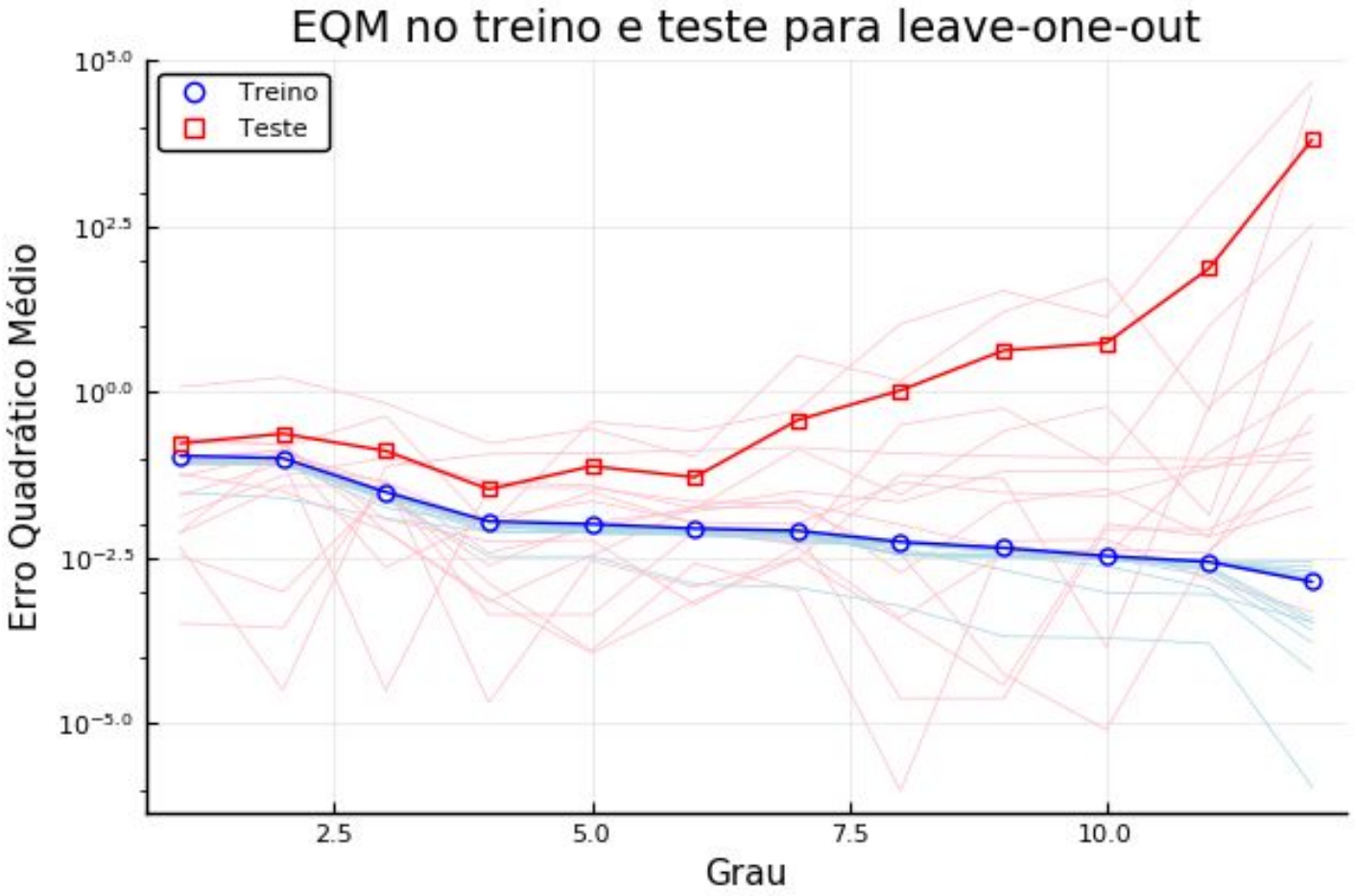

Leave-one-out separando o individuo 13 p = 4
3
2
1
0
0.2
0.4
0.6
0.8
1.0


Leave-one-out separando o individuo 1 p = 9
3
2
1
0
0.2
0.4
0.6
0.8
1.0

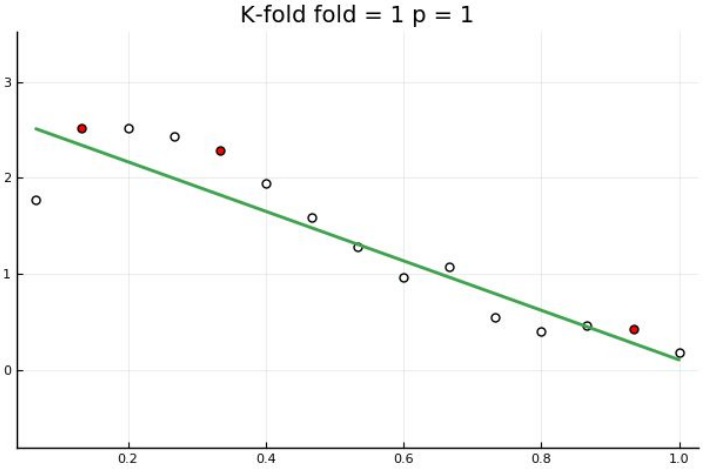

Leave-one-out separando o individuo 3 p = 9
3
2
1
0
0.2
0.4
0.6
0.8
1.0

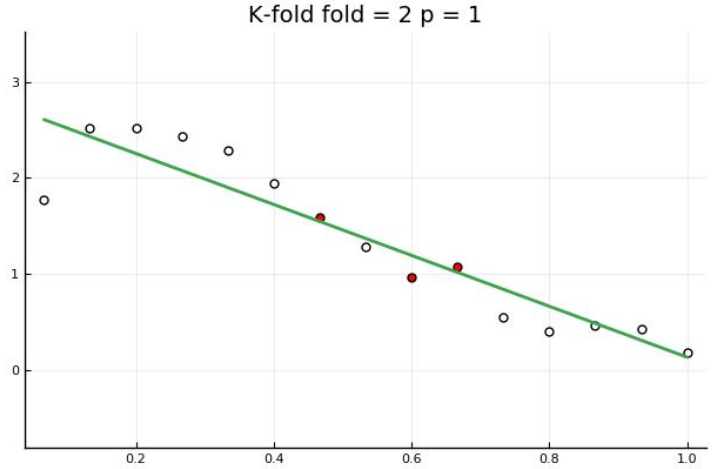

Leave-one-out separando o individuo 7 p = 9
3
2
1
0
0.2
0.4
0.6
0.8
1.0

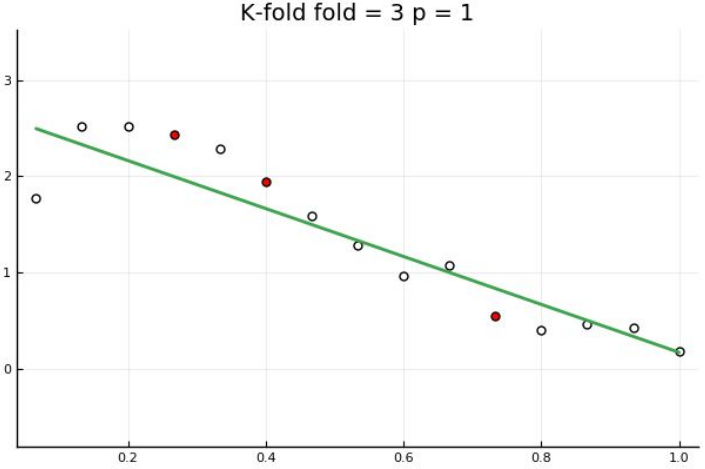

Leave-one-out separando o individuo 13 p = 9
3
2
1
0
0.2
0.4
0.6
0.8
1.0

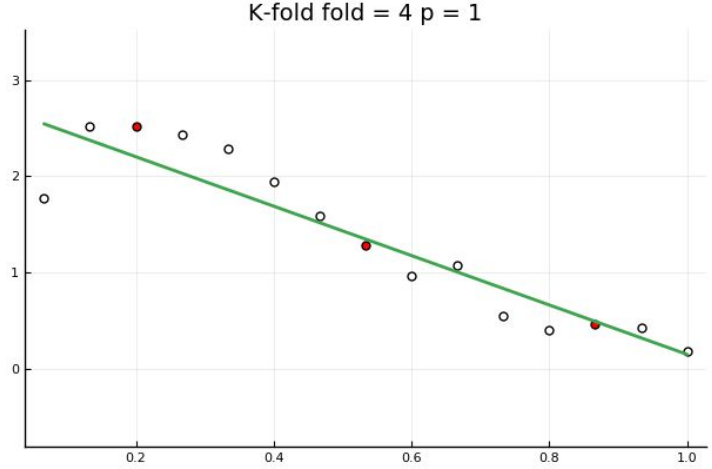

EQM no treino e teste para leave-one-out
Treino
Teste
Erro Quadrático Médio
$10^{5.0}$
$10^{2.5}$
$10^{0.0}$
$10^{-2.5}$
$10^{-5.0}$
2.5
5.0
7.5
10.0
Grau

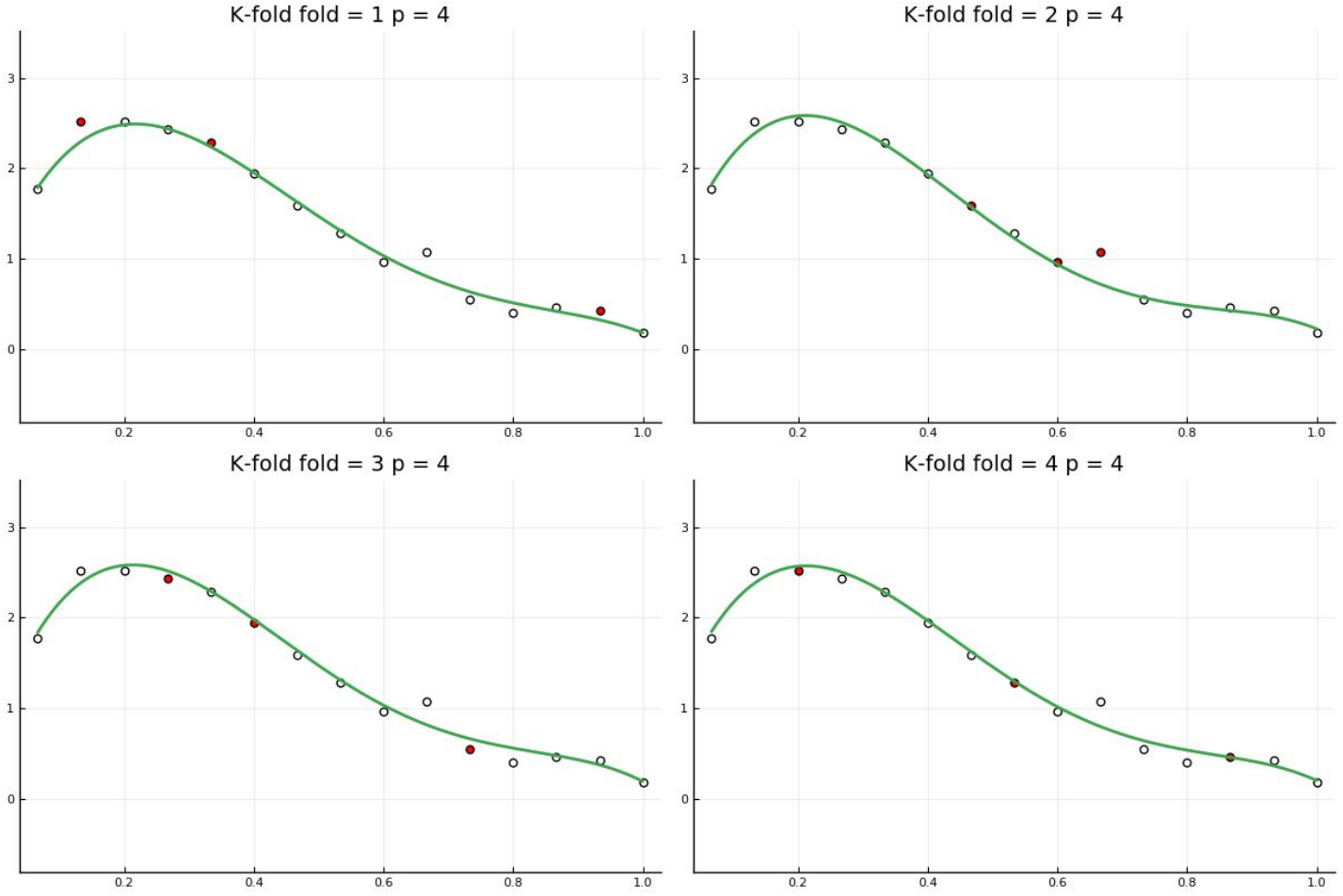
CiDAMO


K-fold fold = 1 p = 1
3
2
1
0
0.2
0.4
0.6
0.8
1.0

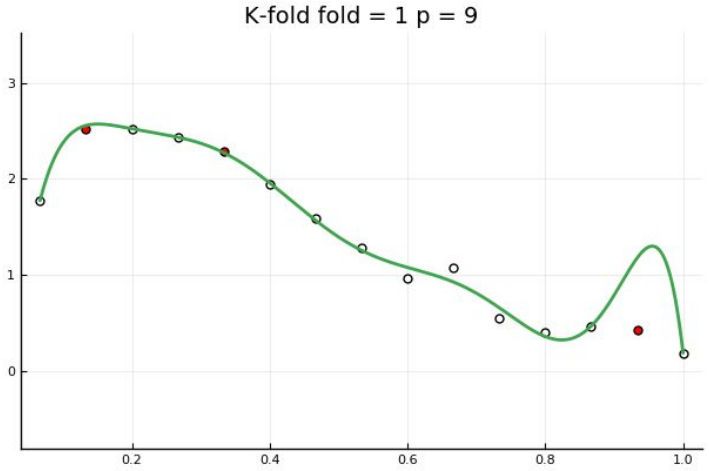

K-fold fold = 2 p = 1
3
2
1
0
0.2
0.4
0.6
0.8
1.0

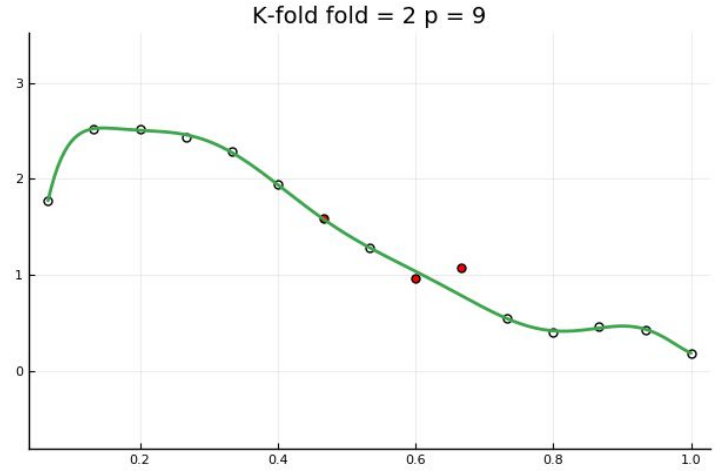

K-fold fold = 3 p = 1
3
2
1
0
0.2
0.4
0.6
0.8
1.0

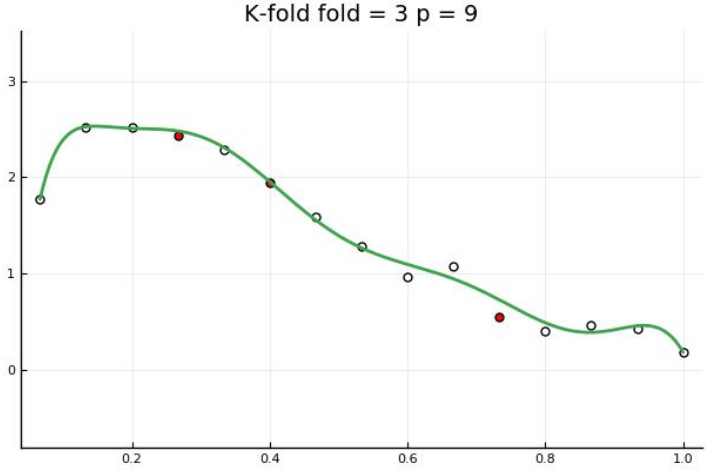

K-fold fold = 4 p = 1
3
2
1
0
0.2
0.4
0.6
0.8
1.0

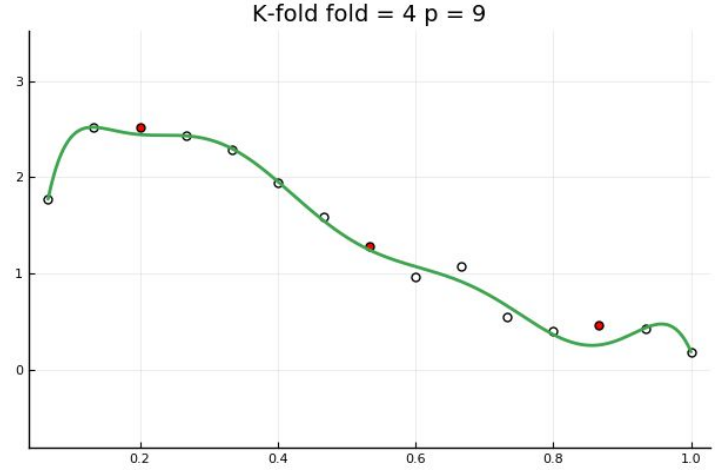

K-fold fold = 1 p = 4
K-fold fold = 2 p = 4
K-fold fold = 3 p = 4
K-fold fold = 4 p = 4
0
1
2
3
0.2
0.4
0.6
0.8
1.0

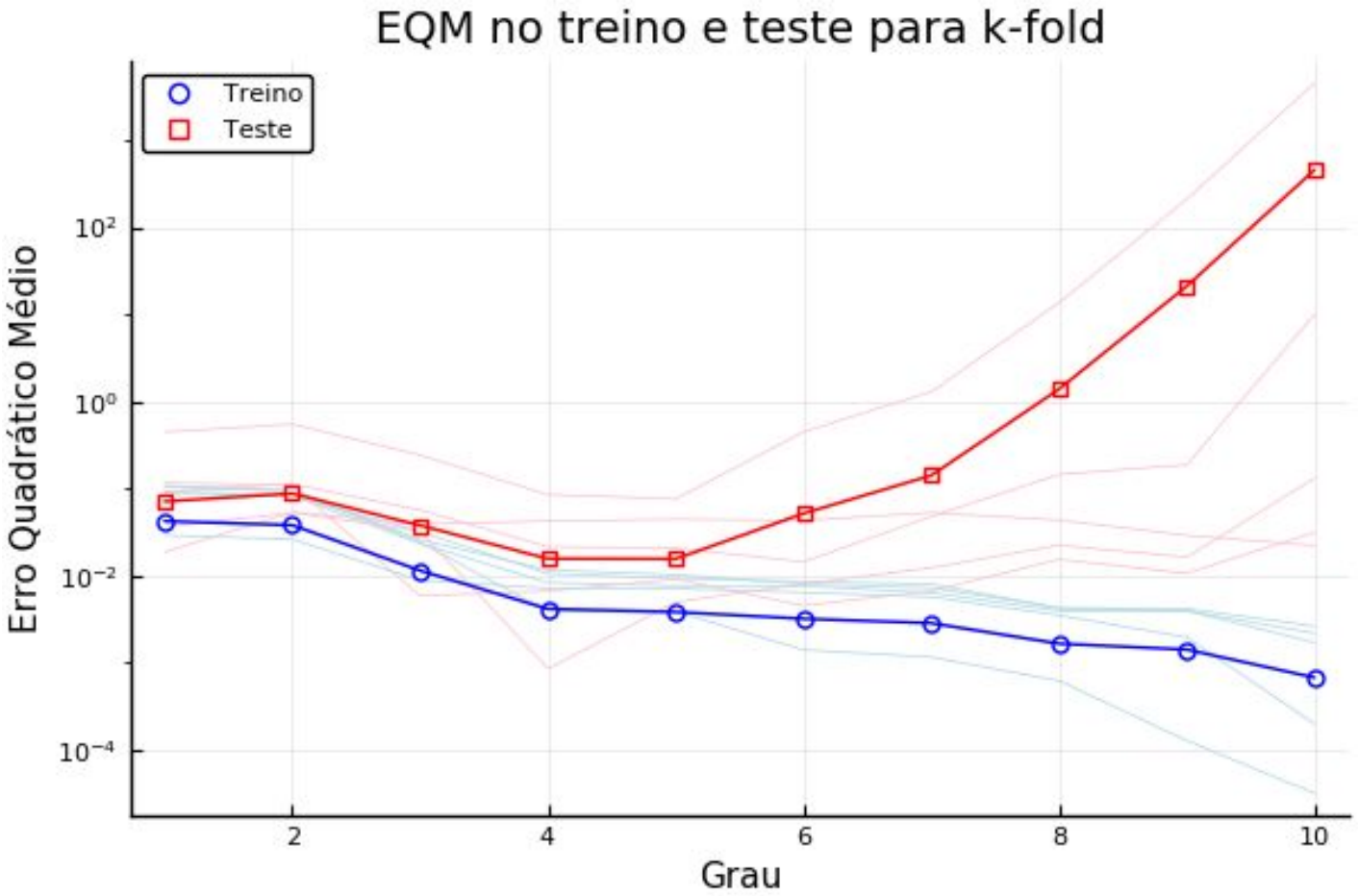
CiDAMO


K-fold fold = 1 p = 9
3
2
1
0
0.2
0.4
0.6
0.8
1.0


K-fold fold = 2 p = 9
3
2
1
0
0.2
0.4
0.6
0.8
1.0

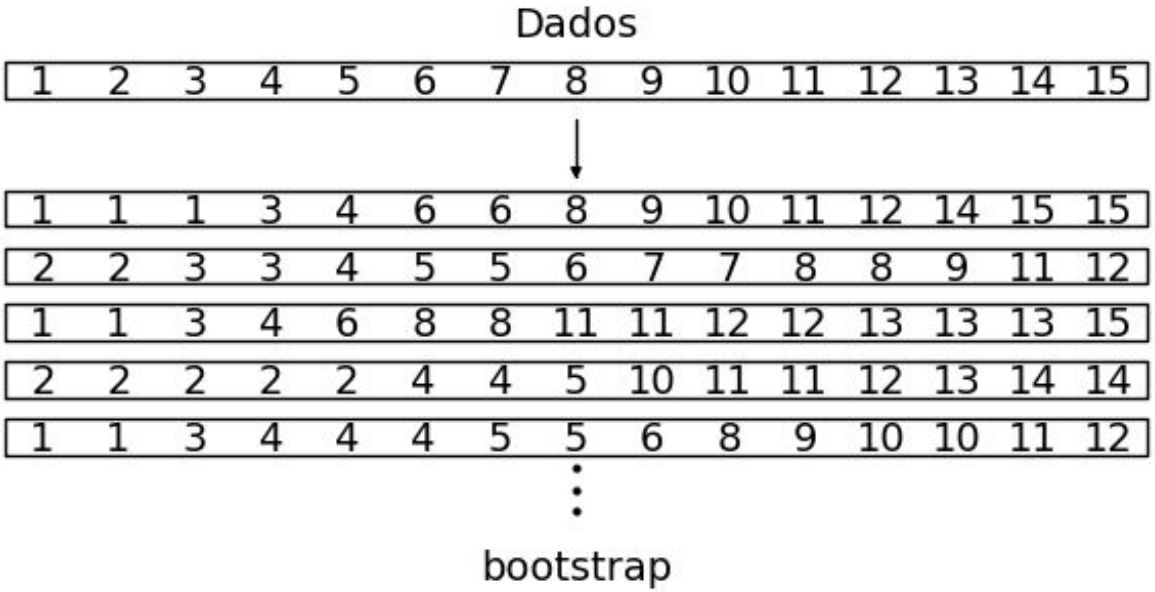

K-fold fold = 3 p = 9
3
2
1
0
0.2
0.4
0.6
0.8
1.0

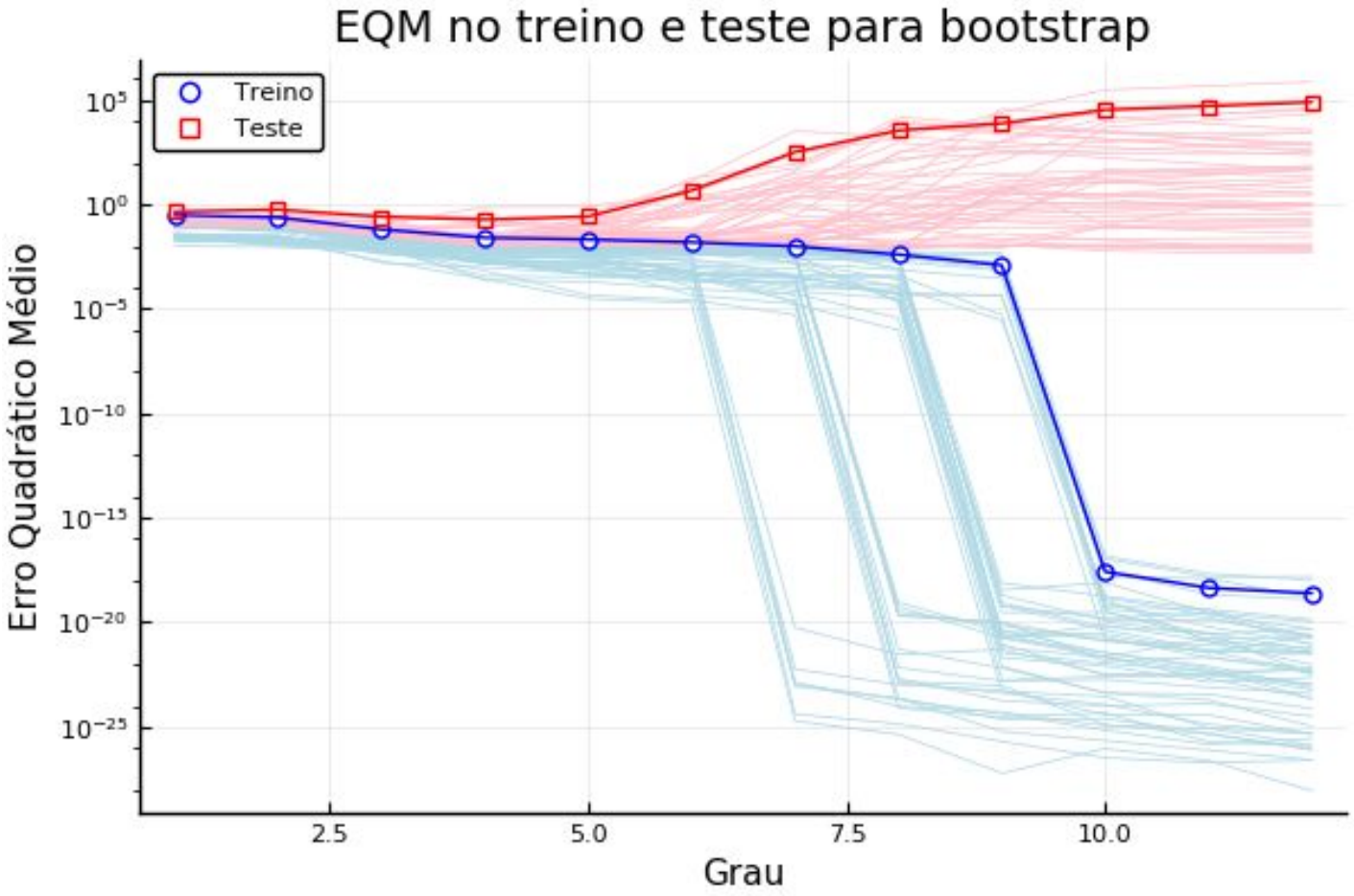

K-fold fold = 4 p = 9
3
2
1
0
0.2
0.4
0.6
0.8
1.0


EQM no treino e teste para k-fold
Treino
Teste
Erro Quadrático Médio
10^{2}
10^{0}
10^{-2}
10^{-4}
2
4
6
8
10
Grau

Bootstrap

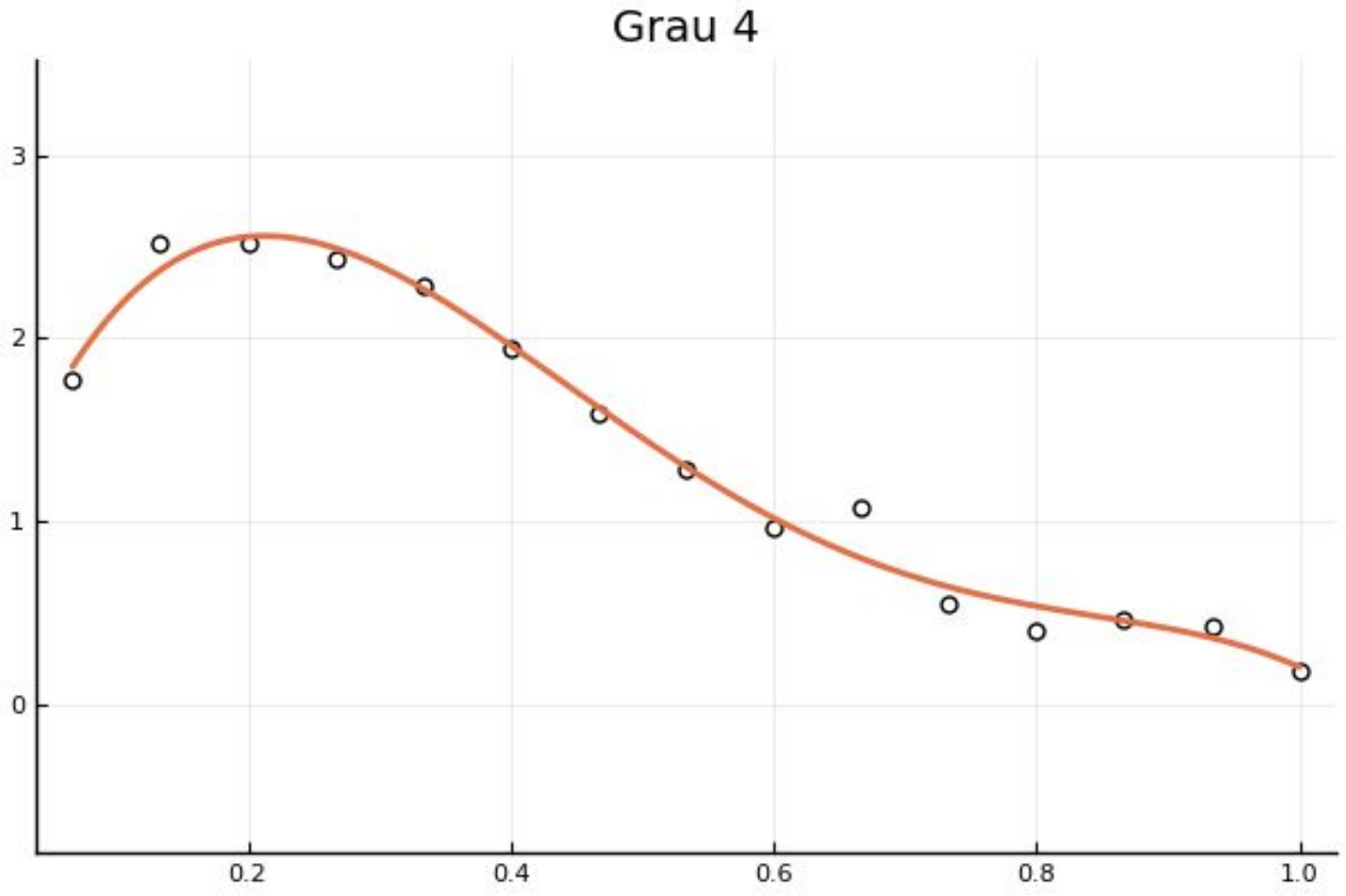
- Cria conjunto com dados originais aleatório e com repetição.
- É caro e causa subestimação do erro. Use com cuidado.




EQM no treino e teste para bootstrap
Treino
Teste
Erro Quadrático Médio
10^{5}
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-20}
10^{-25}
2.5
5.0
7.5
10.0
Grau

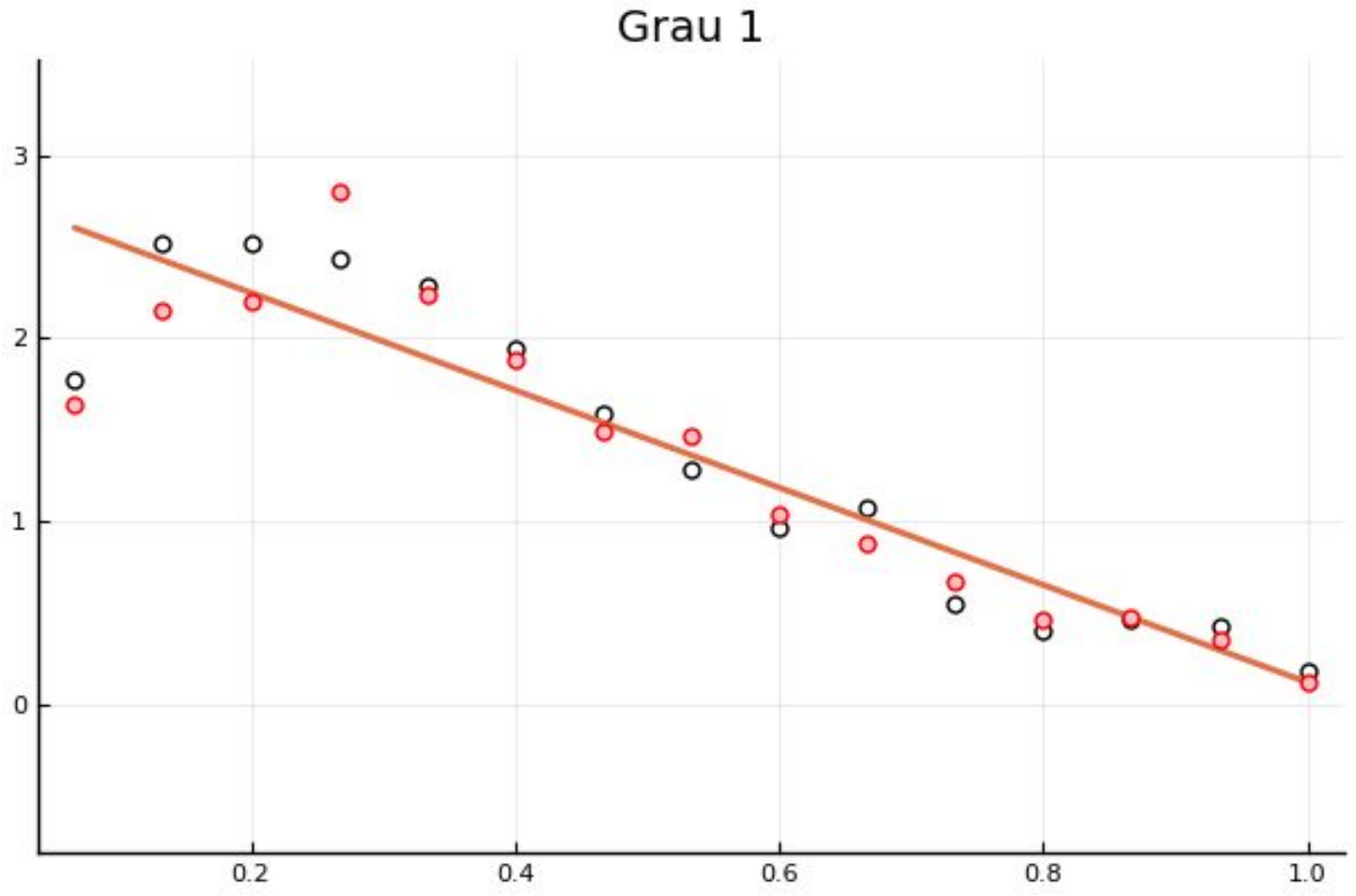
Seleção do modelo

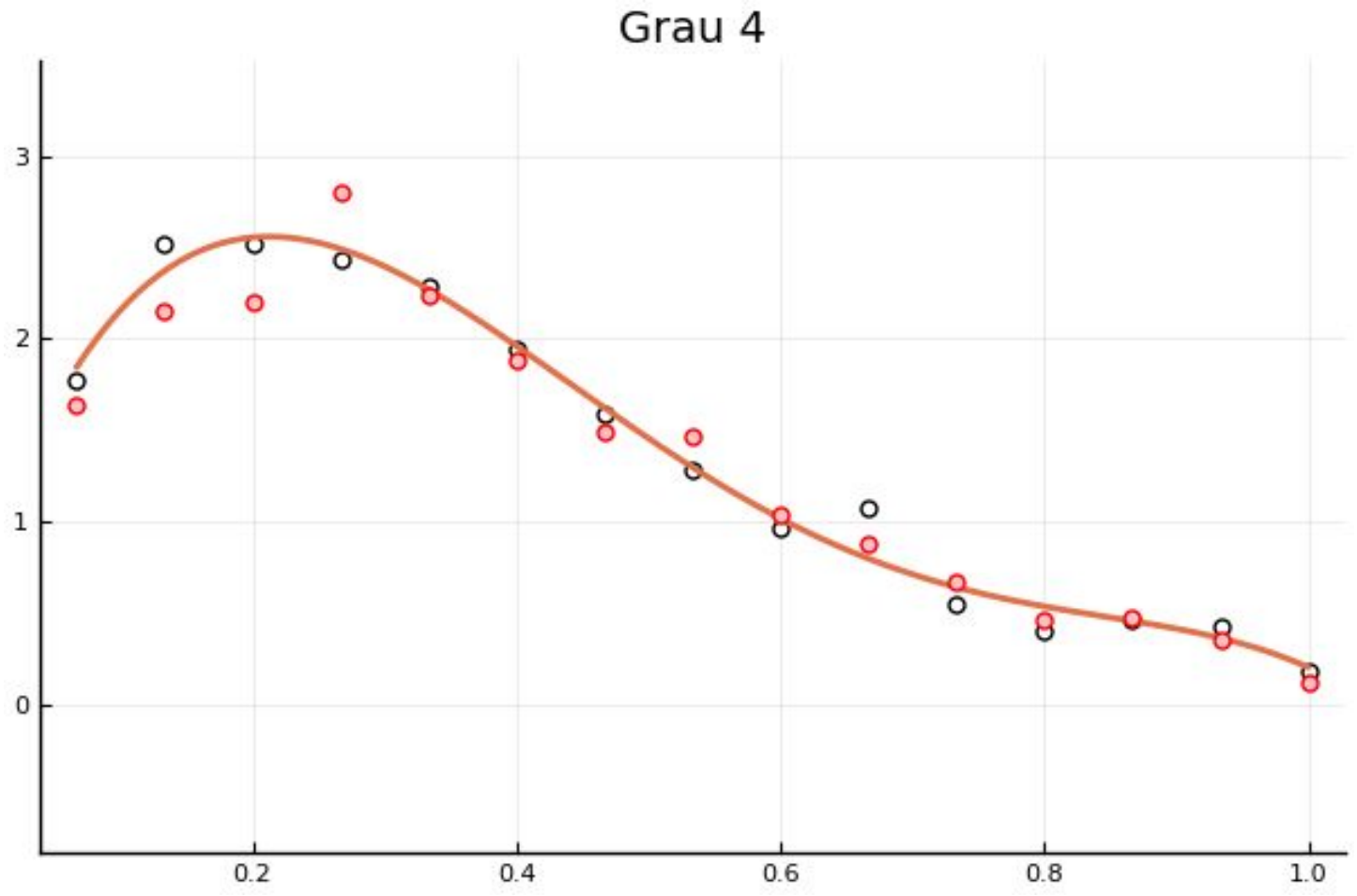
- Todas essas técnicas são usadas para escolher **a família do modelo.**
- Depois de escolhido o grau do polinômio, treina-se com todos os dados.
- Escolhemos aquele que minimiza o erro no teste. (Nos ex. 4 ou 5)

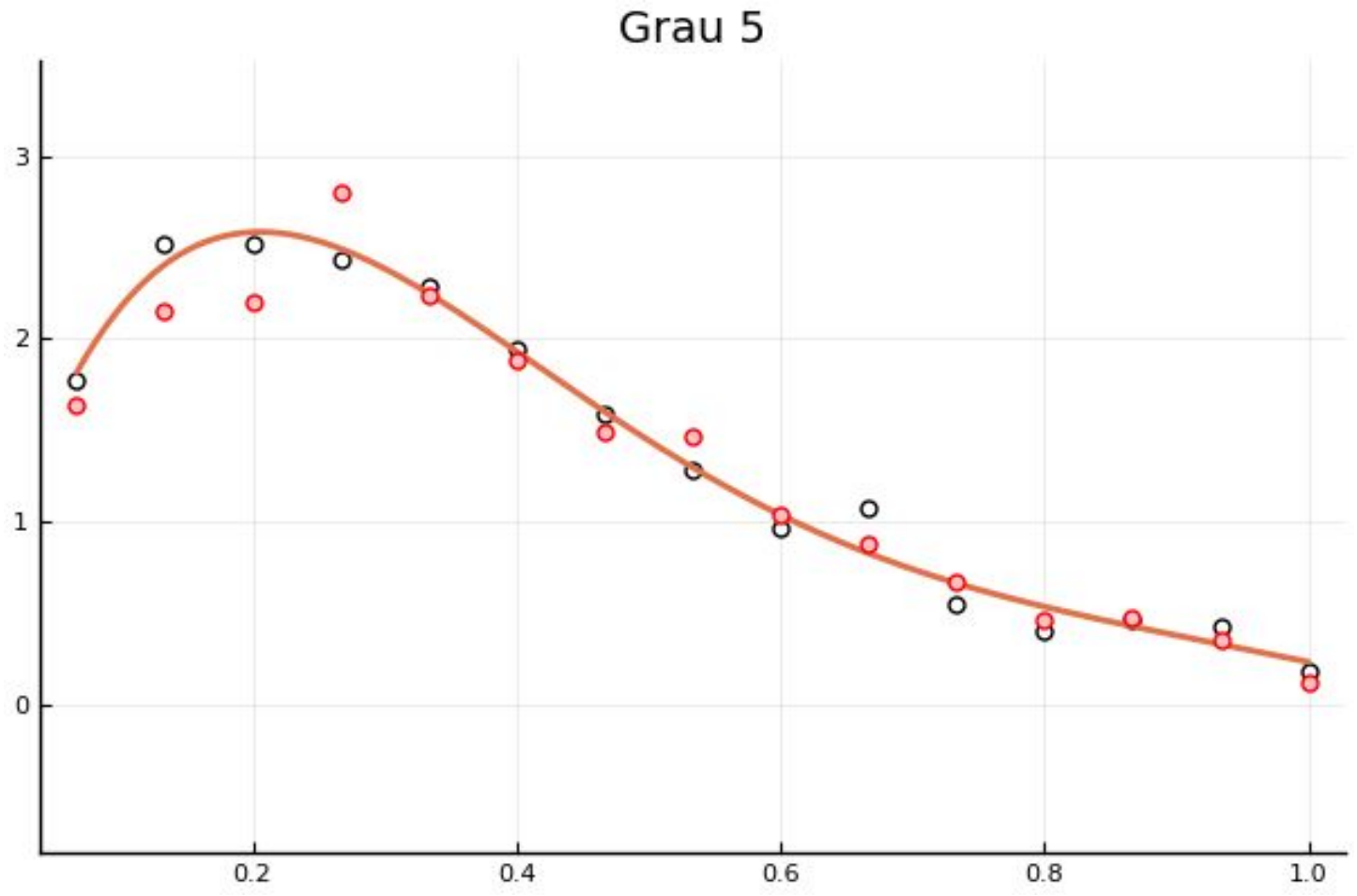


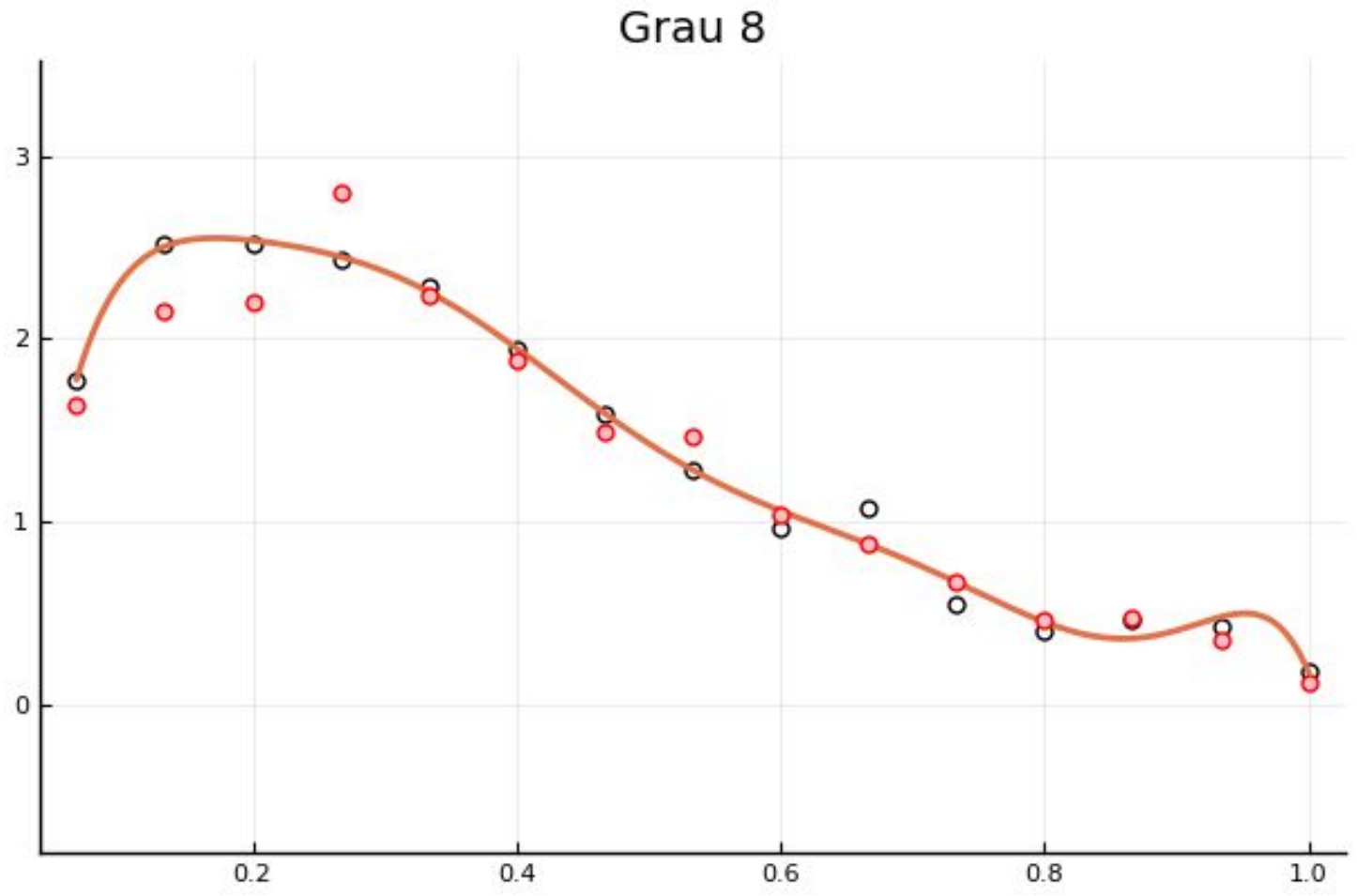
Seleção do modelo

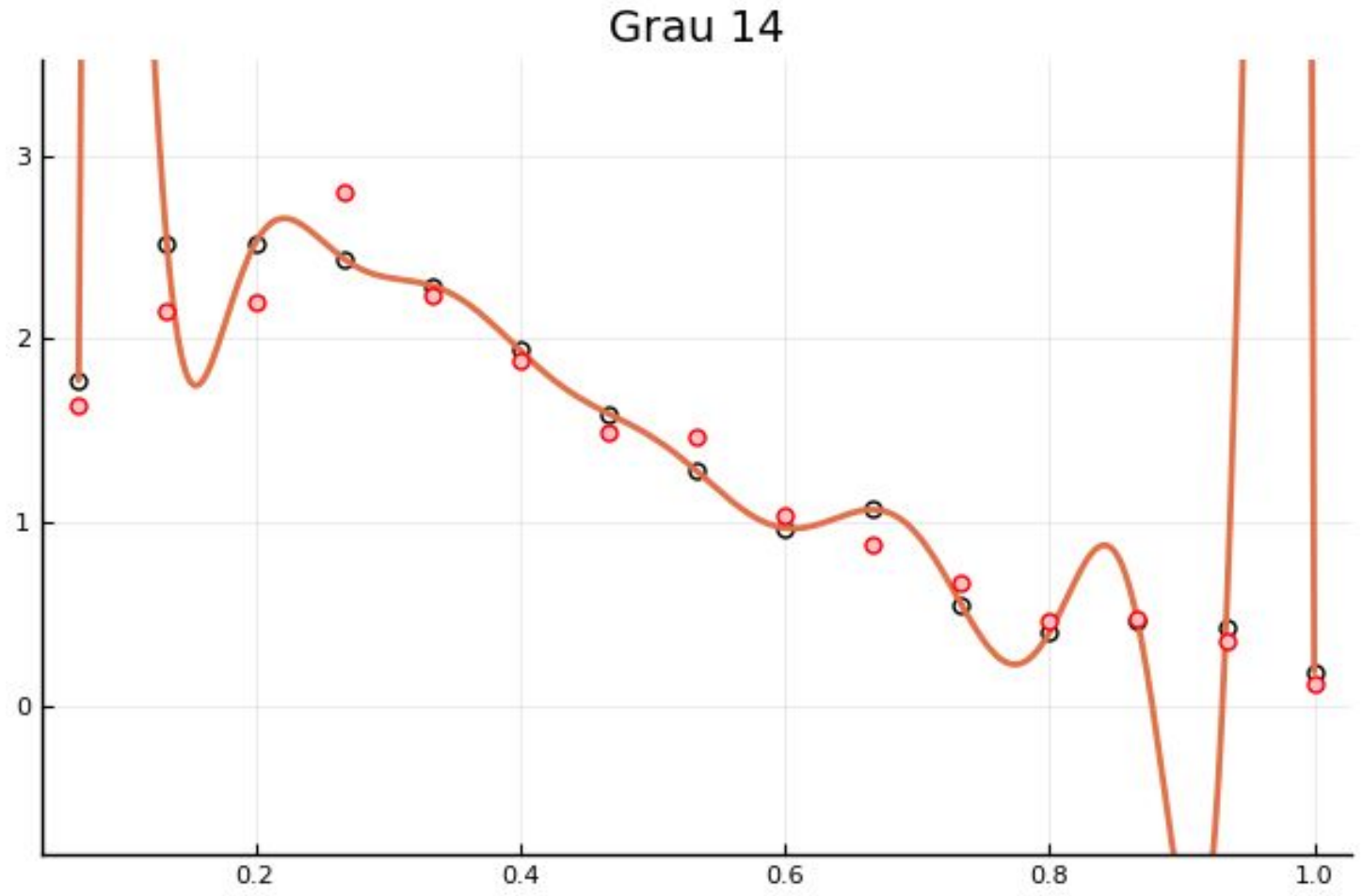
- Mais dados chegaram, vamos ver os resultados

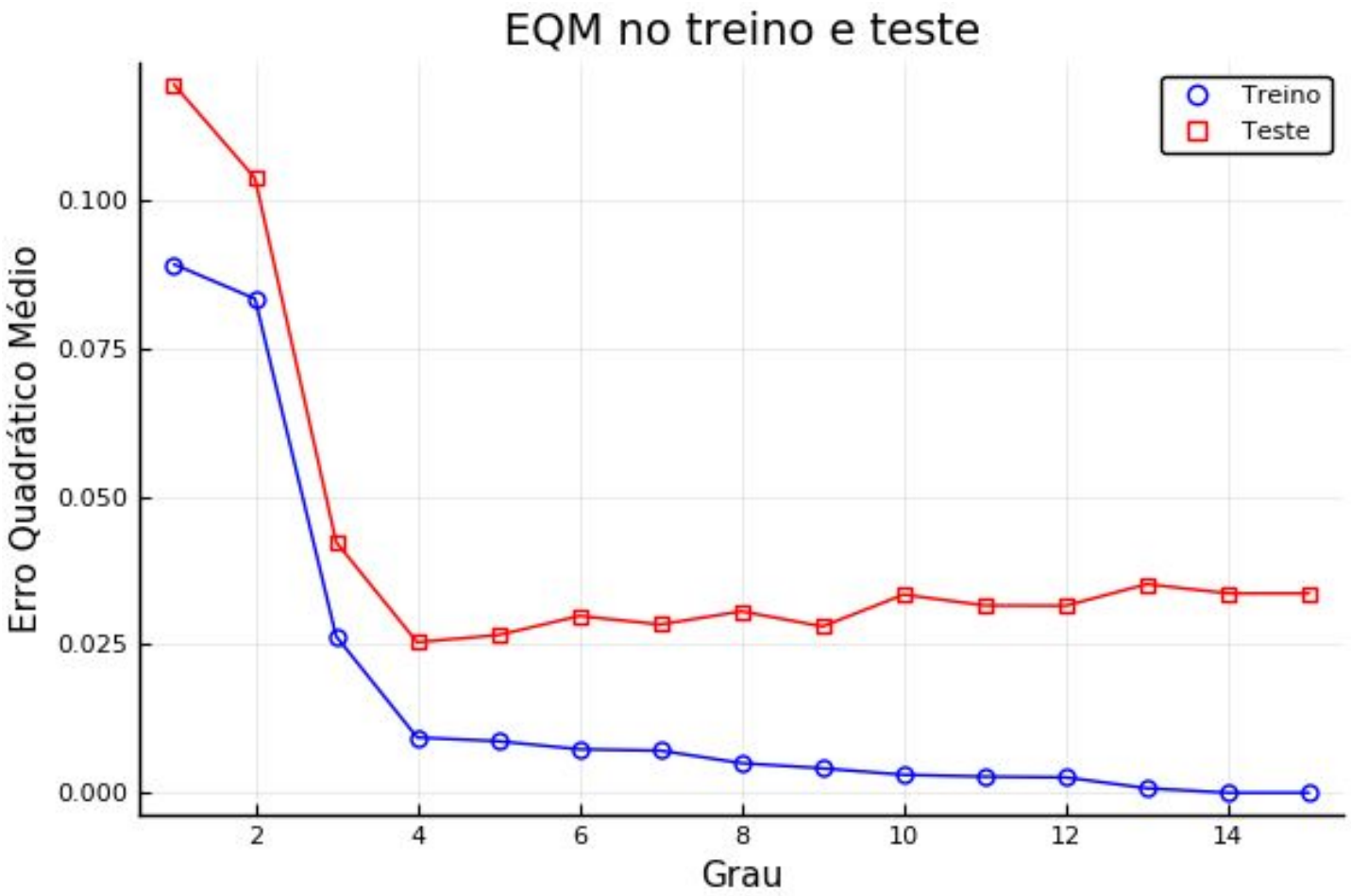











EQM no treino e teste
Treino
Teste
Erro Quadrático Médio
0.100
0.075
0.050
0.025
0.000
2
4
6
8
10
12
14
Grau

Seleção do modelo

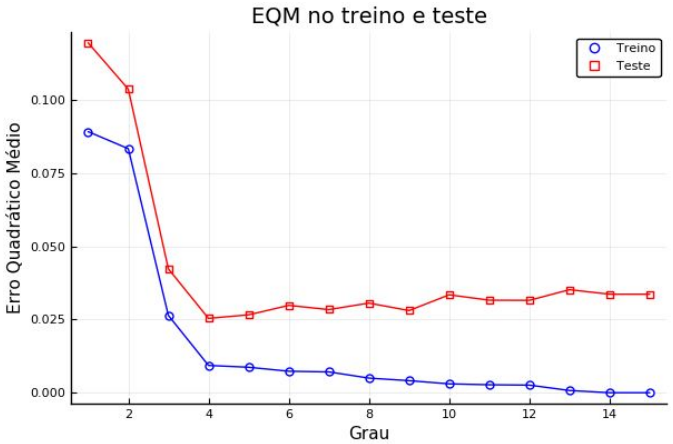
- Muitas vezes teremos os dados já separados (ex.: Kaggle, Hackathon)
- Esses dados são usados para validação final
- Além dessa separação fazemos separação local
- Também vale notar que o que fizemos aqui se aplica para classificação

Otimização de Hiper-parâmetros

- Com um esquema de validação escolhido, definimos uma função

$$\text{grau} \rightarrow \text{Erro no teste para esse grau}$$

- Podemos **otimizar** essa função, i.e., encontrar seu mínimo.
- O grau, neste caso, é chamado de **hiper-parâmetro**, enquanto os coeficientes do modelo são chamados de parâmetros.



Otimização de Hiper-parâmetros

- Existem outros hiper-parâmetros, e.g.
  - Ramificações nas árvores de decisão
  - Valor do parâmetro de regularização
  - Kernel no SVM
- NÃO são hiper-parâmetros:
  - K no k-fold
  - A semente de randomização

Grid Search

- Grid Search faz uma busca exaustiva para todas as combinações
- Testar todos RandomForest:
  - n_estimators: [10, 100]
  - max_depth: [1, 5]
  - max_leaf_nodes: [10, None]

Grid Search

CiDAMO

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, GridSearchCV

# ...

X_train, X_test, y_train, y_test = train_test_split(X, y)

param_grid = [
    {'n_estimators': [10, 100], 'max_depth': [1, 5], 'max_leaf_nodes': [10, None]}
]

search = GridSearchCV(RandomForestClassifier(), param_grid, cv=5)

search.fit(X_train, y_train)
```

Grid Search

- Max_depth = 1
  - Max_leaf_nodes = 10
    - N_estimator = 10
      - **ERRO = 0.015**
    - N_estimator = 100
      - **ERRO = 0016**
  - Max_leaf_nodes = None
    - N_estimator = 10
      - **ERRO = 0.015**
    - N_estimator = 100
      - **ERRO = 0017**

- Max_depth = 5
  - Max_leaf_nodes = 10
    - N_estimator = 10
      - **ERRO = 0.033**
    - N_estimator = 100
      - **ERRO = 0047**
  - Max_leaf_nodes = None
    - N_estimator = 10
      - **ERRO = 0.044**
    - N_estimator = 100
      - **ERRO = 0072**

Ex. contínuo: Regularização

CiDAMO

- Regressão linear com regularização Ridge/LASSO

$$\min \ \frac{1}{2}\|X\beta - y\|^2 + \frac{1}{2}\lambda\|\beta\|^2 \qquad \min \ \frac{1}{2}\|X\beta - y\|^2 + \lambda\|\beta\|_1$$

- O parâmetro de regularização pode assumir qualquer valor não-negativo
- GridSearch se limita a um conjunto finito, mas fornece uma aproximação
- Existem outras estratégias - e toda uma área

Sumário

CiDAMO

- Dilema vício-variância
- Use validação cruzada
- Holdout para conjuntos grandes
- Leave-one-out para pequenos
- K-fold é um meio termo adequado
- Grid Search para avaliar combinações de hiper-parâmetros

Obrigado

CiDAMO