

# Planul Cursului

---

Overfitting/Supraadaptare

Selectarea Modelului

Cross Validation/Validare încrucișată

Bias vs Varianță

Regularizare: LASSO și Ridge

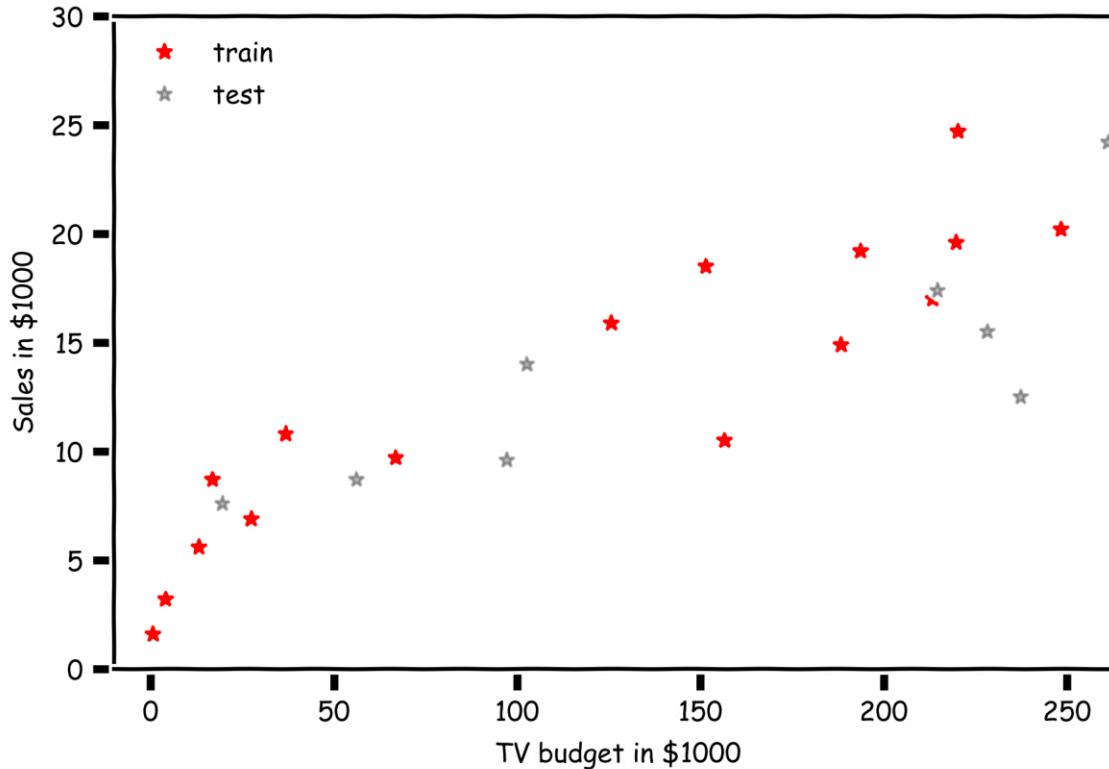
Metod de Regularizare: O Comparatie



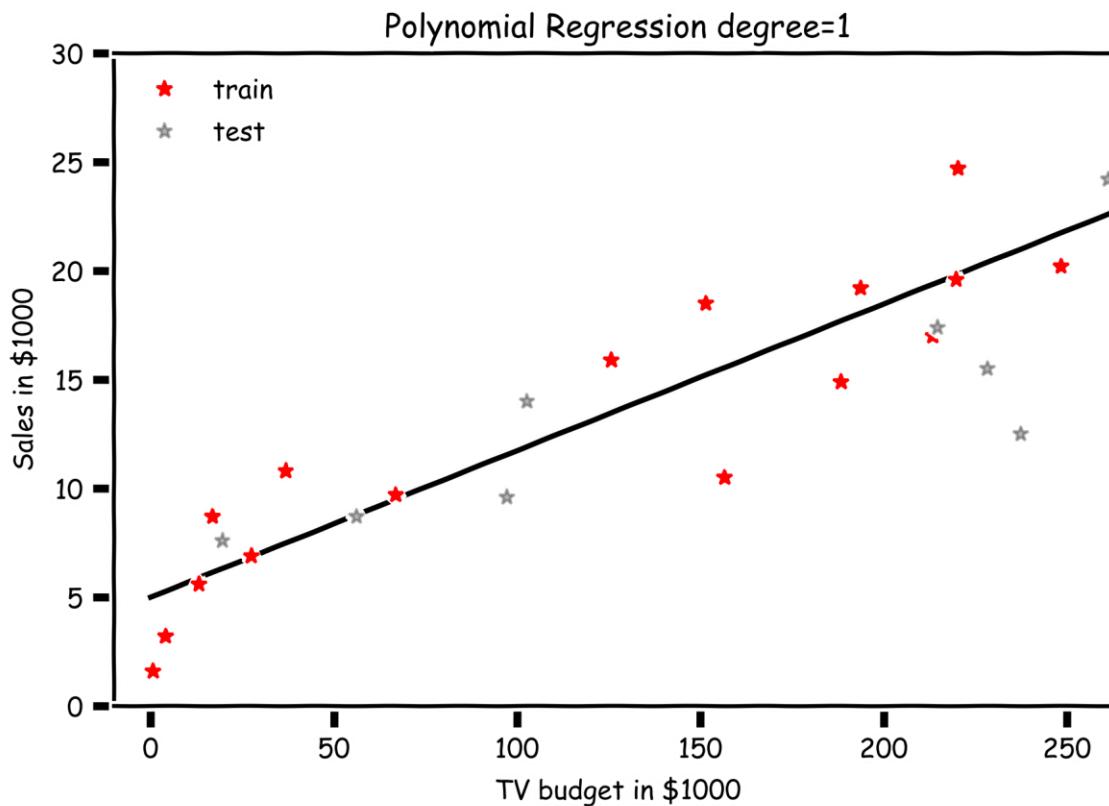


## **OVERFITTING/SUPRAADAPTARE**

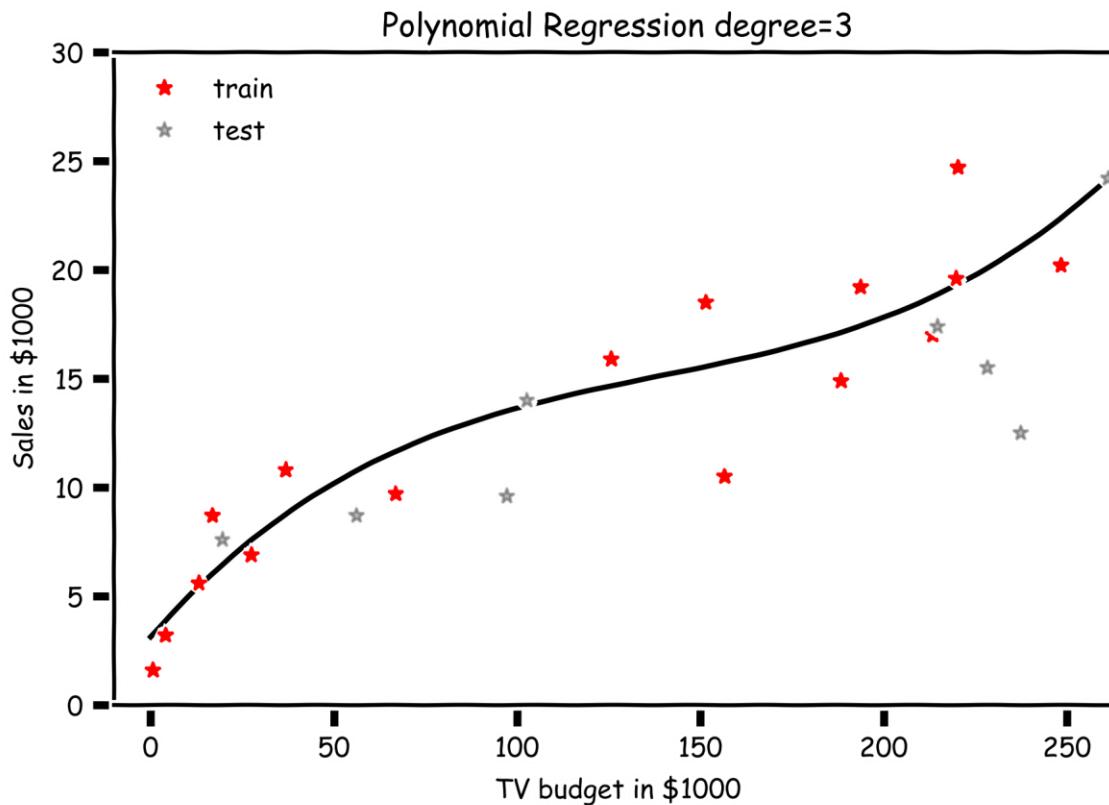
# Overfitting



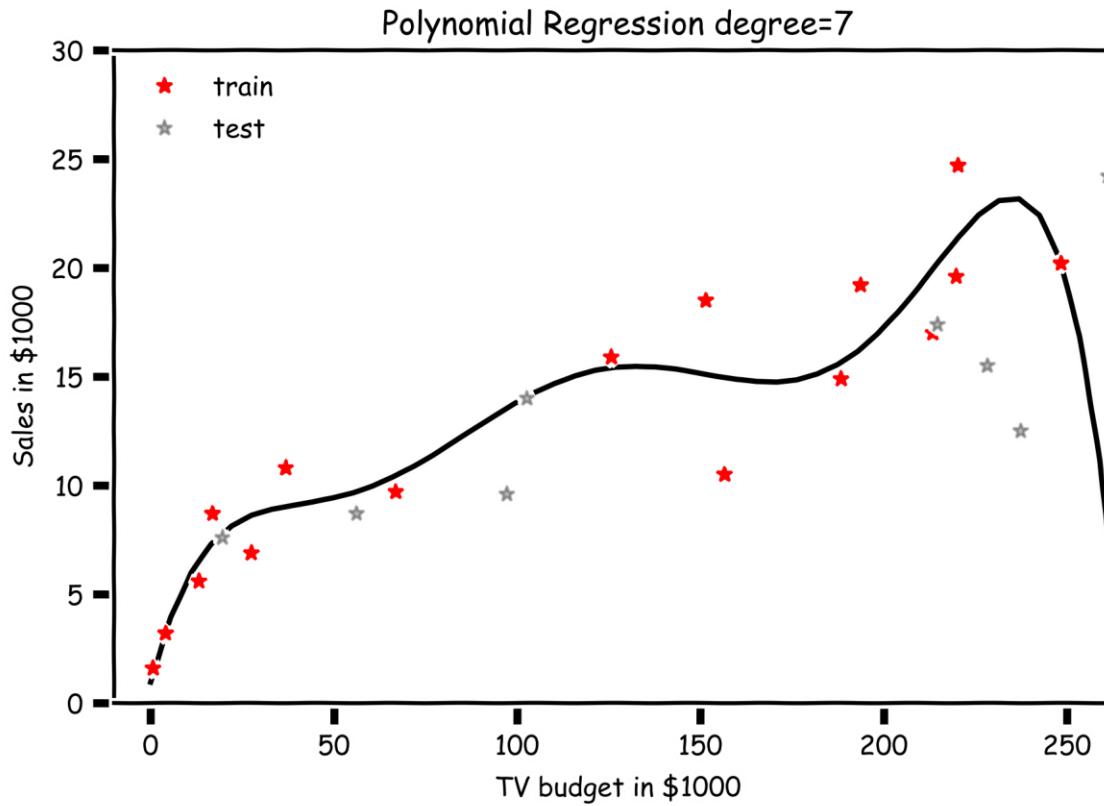
# Overfitting



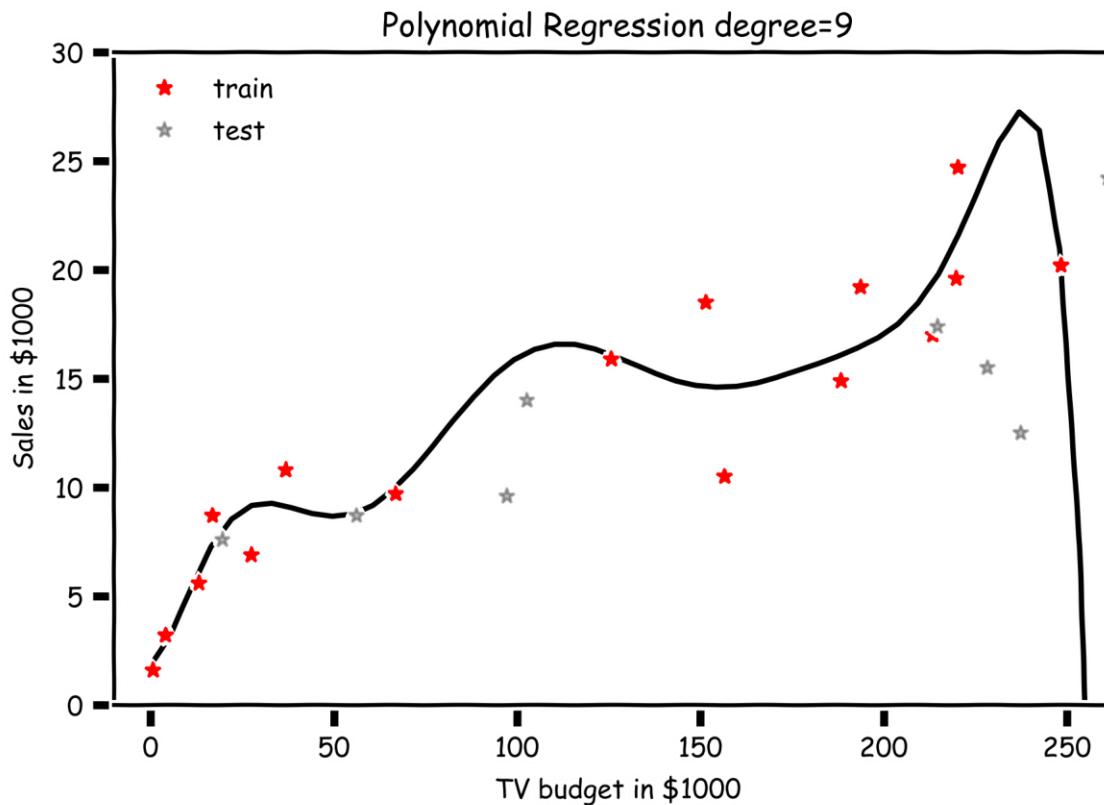
# Overfitting



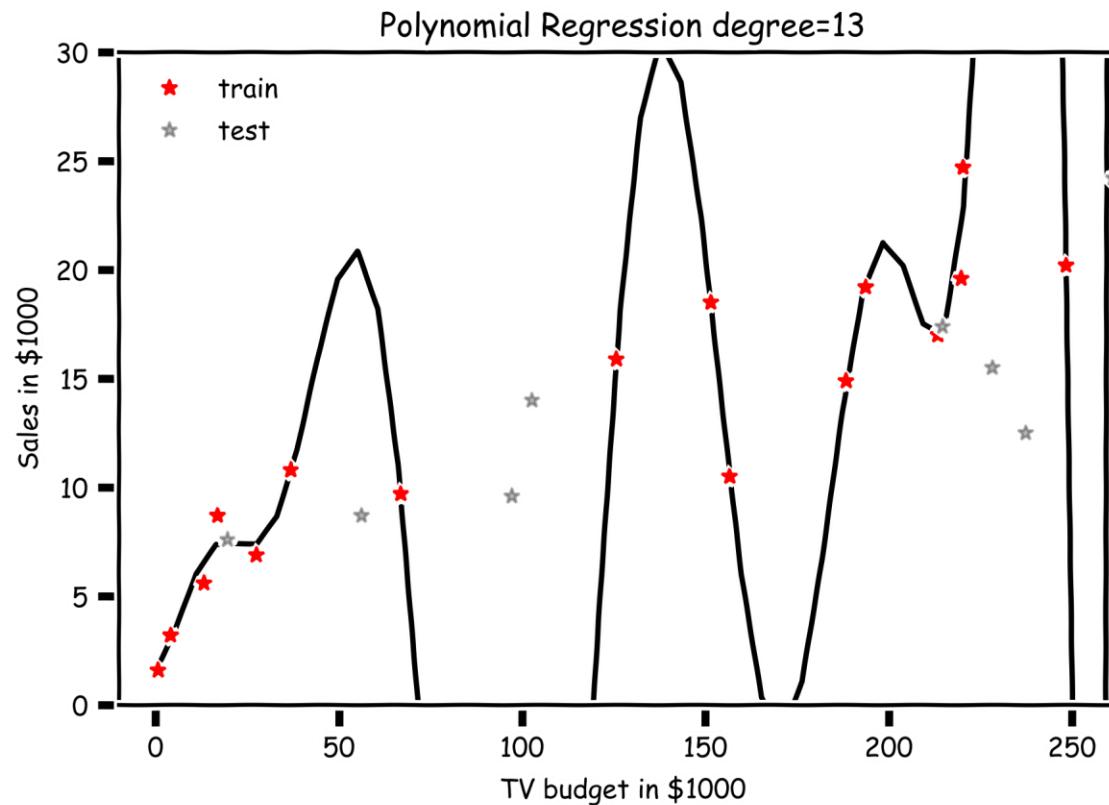
# Overfitting



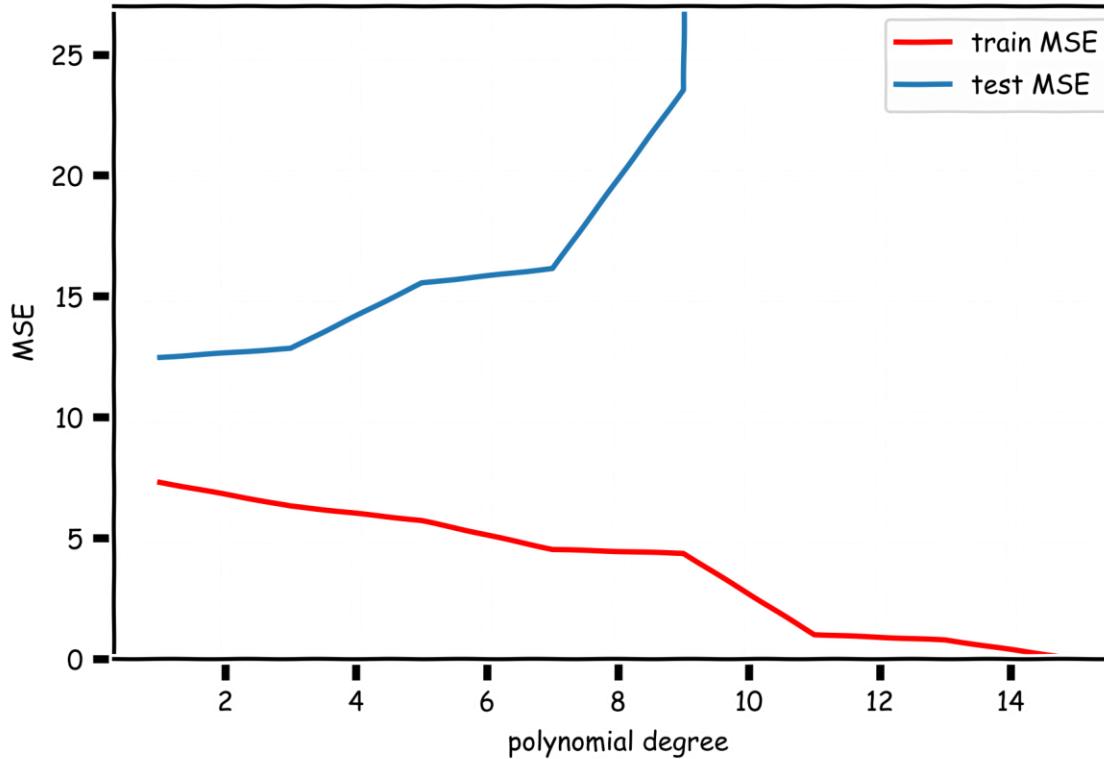
# Overfitting



# Overfitting



# Validation



# Antrenare-Validare-Testare

## Înrebare:

Cum ați raportat performanța modelului?

$R^2_{test} = 0.52$

$R^2_{train}(\text{grad}=1) = 0.83$





## **SELECAREA MODELULUI**

# Selectarea Modelului

---

**Selectia modelului** este aplicarea unei metode bazate pe principii pentru a determina complexitatea modelului, de ex. alegerea unui subset de predictori, alegerea gradului modelului polinomial etc.

O motivație puternică pentru efectuarea selecției modelului este de a evita supraadaptarea, ceea ce am văzut că se poate întâmpla când:

- sunt prea mulți predictori:
  - spațiul caracteristicilor are o dimensionalitate ridicată (multe coloane)
  - gradul polinomial este prea mare
  - se iau în considerare prea mulți termeni încrucișăți
- **valorile** coeficienților sunt **prea extreme (nu am văzut asta încă)**

# Selectarea Modelului

---

**Întrebare:**

Câte modele diferite atunci când se iau în considerare un număr de  $J$  predictori?



# Selectarea Modelului

**Exemplu: 3 predictori ( $X_1, X_2, X_3$ )**

- Model cu 0 predictori (de ex. Media):

M0:

- Modele cu 1 predictor:

M1:  $X_1$

M2:  $X_2$

M3:  $X_3$

- Modele cu 2 predictori:

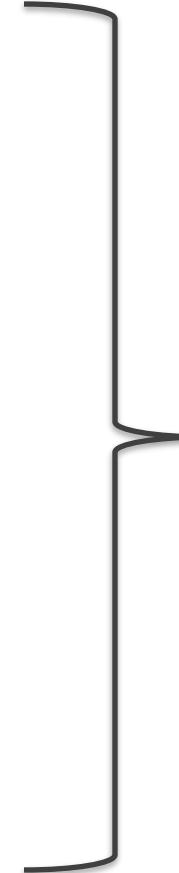
M4:  $\{X_1, X_2\}$

M5:  $\{X_2, X_3\}$

M6:  $\{X_3, X_1\}$

- Modele cu 3 predictori:

M7:  $\{X_1, X_2, X_3\}$



$2^J$  Modele

# Alegere pas cu pas a predictorilor și validare încrucișată

---

Selectarea subseturilor optime de predictori (inclusiv alegerea gradului pentru modele polinomiale) prin:

- selecție pas cu pas - construirea iterativă a unui subset optim de predictori prin optimizarea unei metrii fixate de evaluare a modelului,
- validare - selectarea unui model optim prin evaluarea fiecărui model pe setul de validare.

De asemenea, vom aborda problema descurajării valorilor extreme în parametrii modelului mai târziu.

# Alegere pas cu pas a predictorilor: metoda selecției directe

## *Forward Selection*

În selecția directă, găsim un set „optim” de predictori prin construirea iterativă a setului nostru.

**1.** Pornim cu mulțimea vidă  $P_0$ , construim modelul nul  $M_0$ .

**2.** Pentru  $k = 1, \dots, J$ :

**2.1** Fie  $M_{k-1}$  modelul construit pe cea mai bună selecție de  $k - 1$  predictori,  $P_{k-1}$ .

**2.2** Selectăm predictorul  $X_{n_k}$ , care să nu fie în  $P_{k-1}$ , așa încât  $P_k = X_{n_k} \cup P_{k-1}$  optimizează o metrică fixată (poate fi una din p -value, F -stat; MSE pe setul de validare,  $R^2$ , sau AIC/BIC pe setul de testare).

**2.3** Fie  $M_k$  modelul construit din submulțimea optimă  $P_k$ .

**3.** Alegem modelul  $M$  din  $\{M_0, M_1, \dots, M_J\}$  care optimizează metrica aleasă

# Alegere pas cu pas a predictorilor: Complexitate de Calcul

---

Câte modele vom evalua?

- Pas 1,  **$J$  Modele**
- Pas 2,  **$J-1$  Modele** (adăugam 1 predictor din  $J-1$  posibile)
- Pas 3,  **$J-2$  Modele** (adăugam 1 predictor din  $J-2$  posibile)
- ...

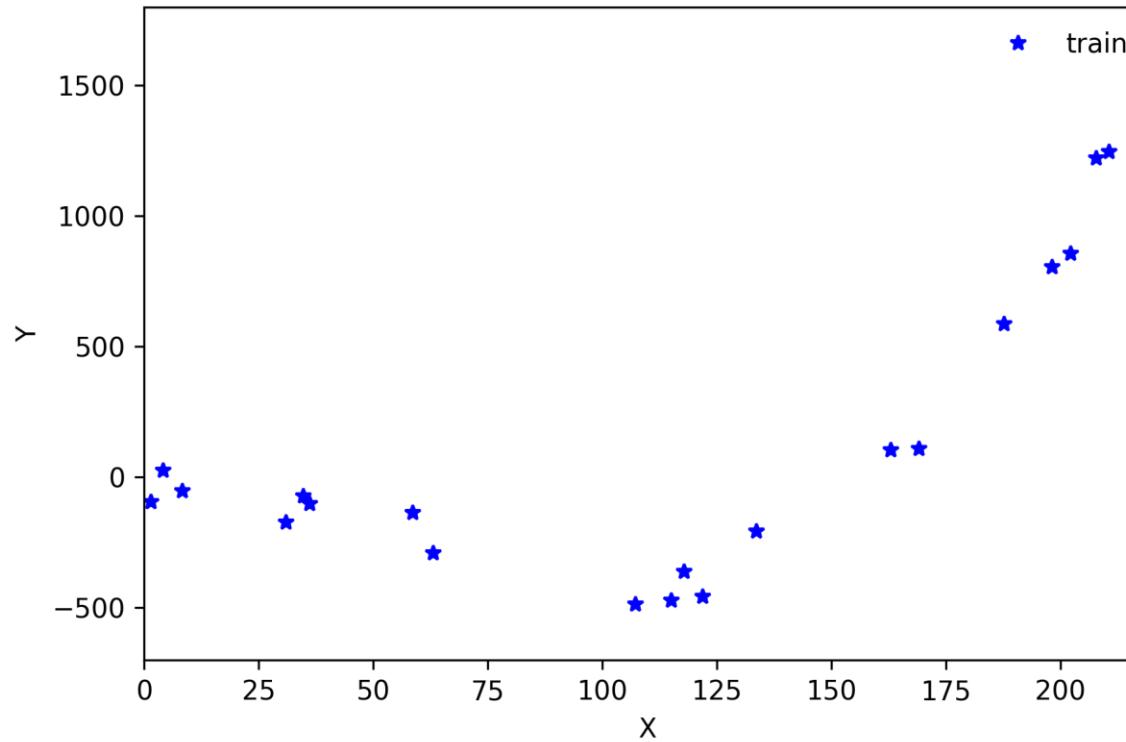
$O(J^2) \ll 2^J$  pentru valori mari ale lui  $J$



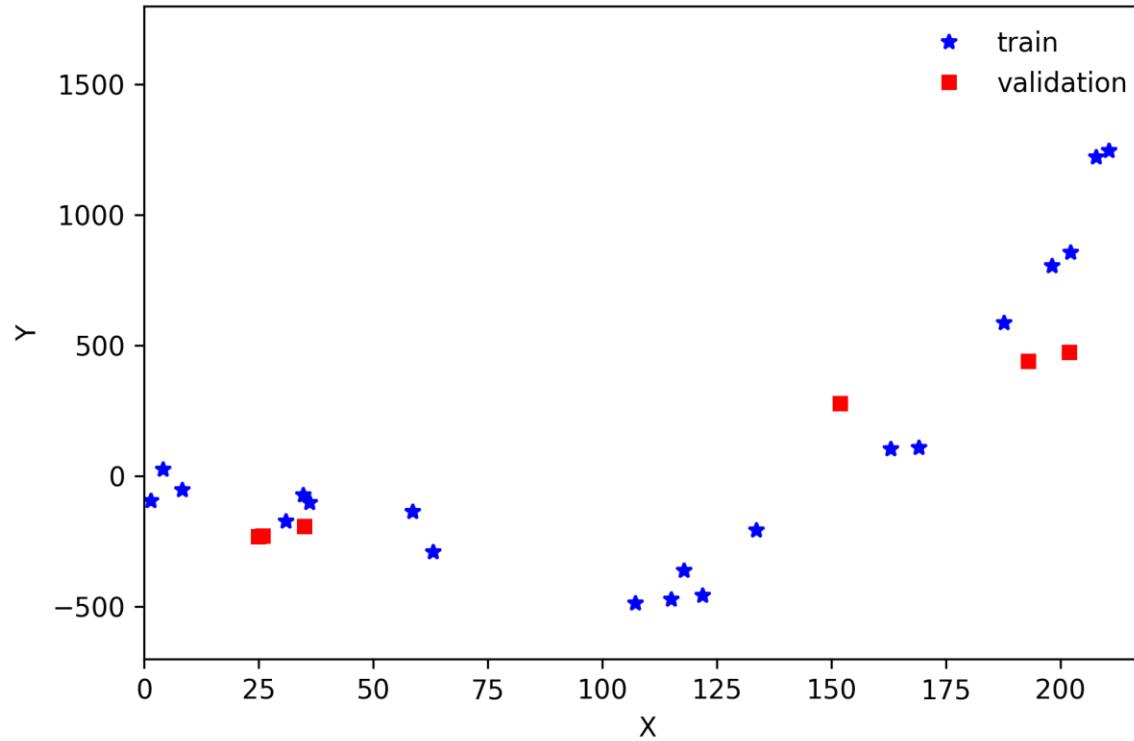


## **CROSS VALIDATION/ VALIDARE ÎNCRUCIŞATĂ**

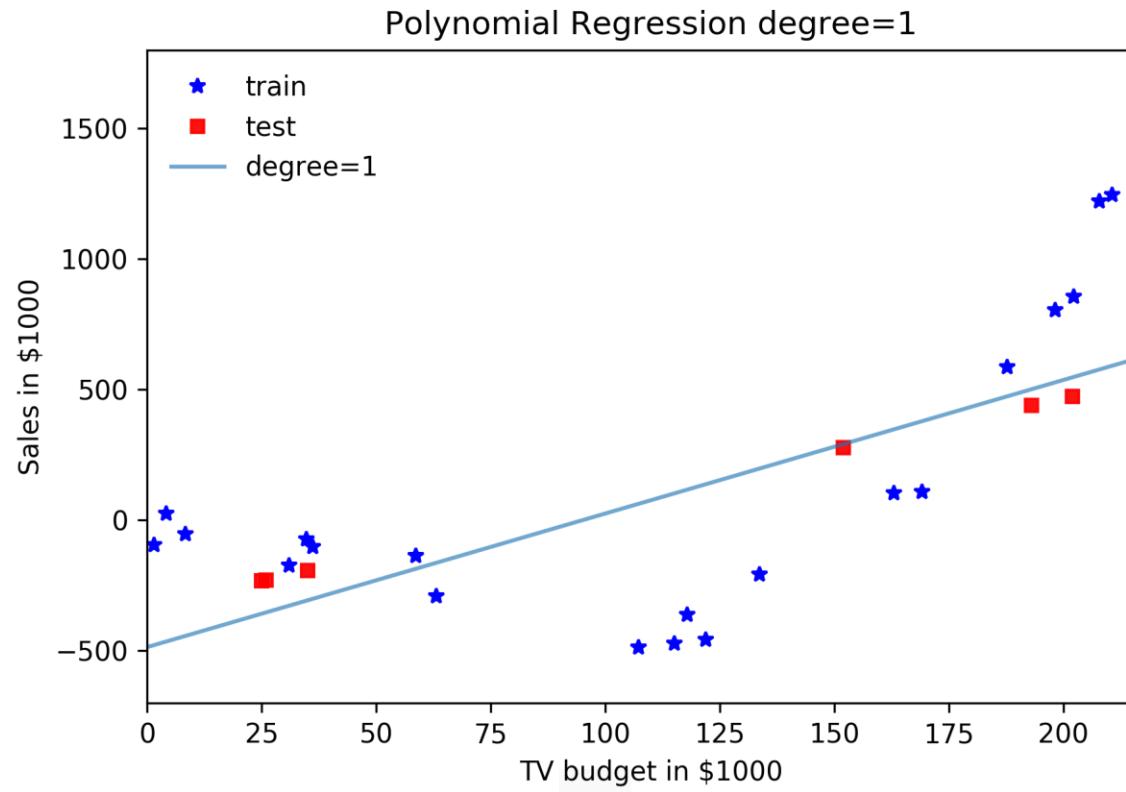
# Cross Validation/Validare Încrucișată



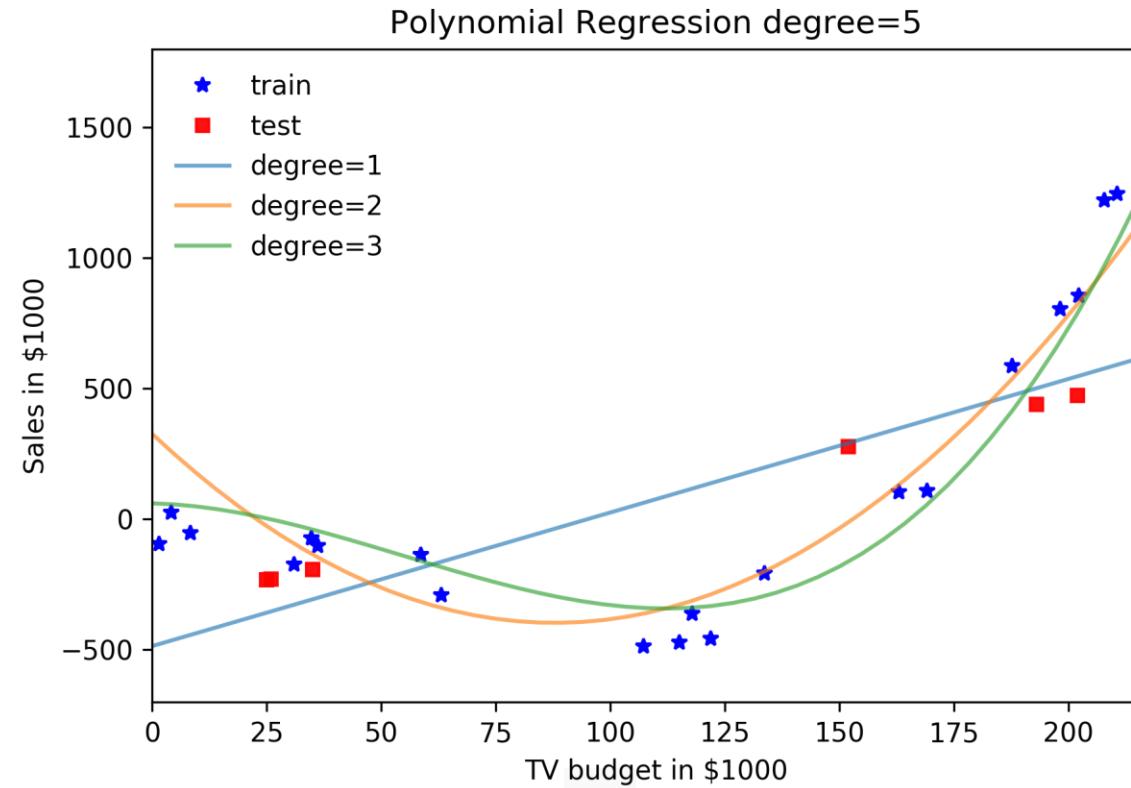
# Cross Cross Validation/Validare Încrucișată



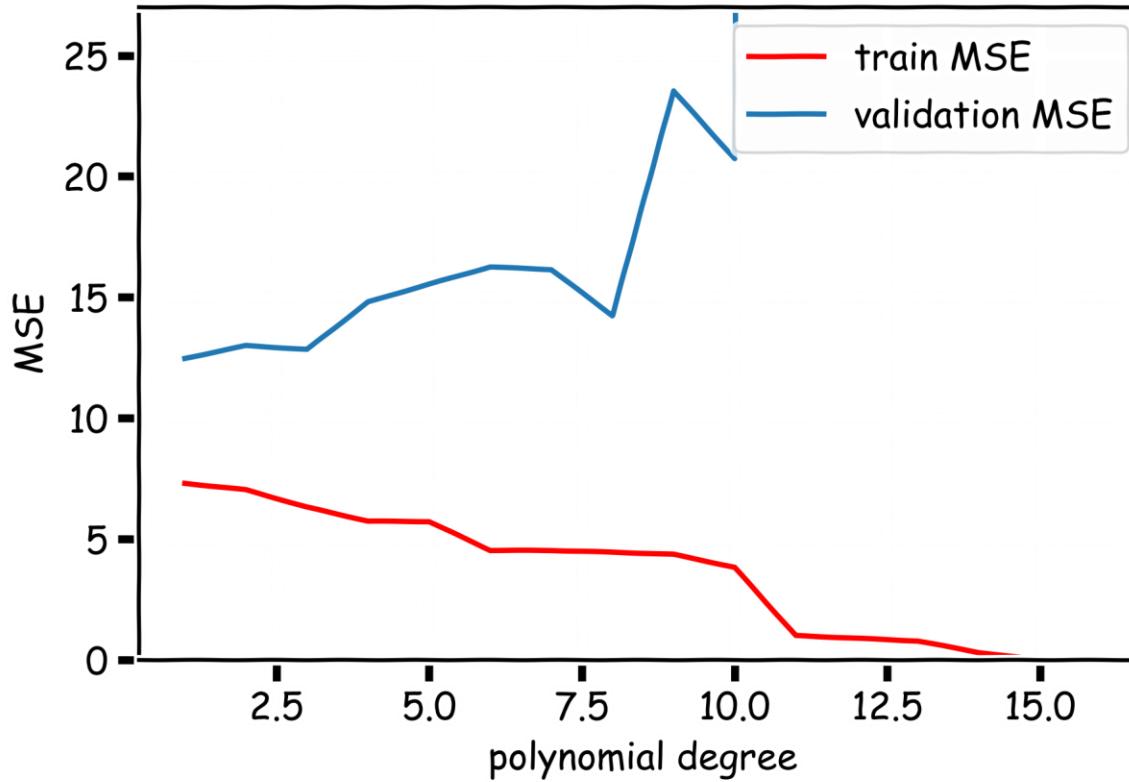
# Cross Validation/Validare Încrucișată Validation



# Cross Validation/Validare Încrucișată



# Validare



# Cross Validation/Validare Încrucișată: Motivație

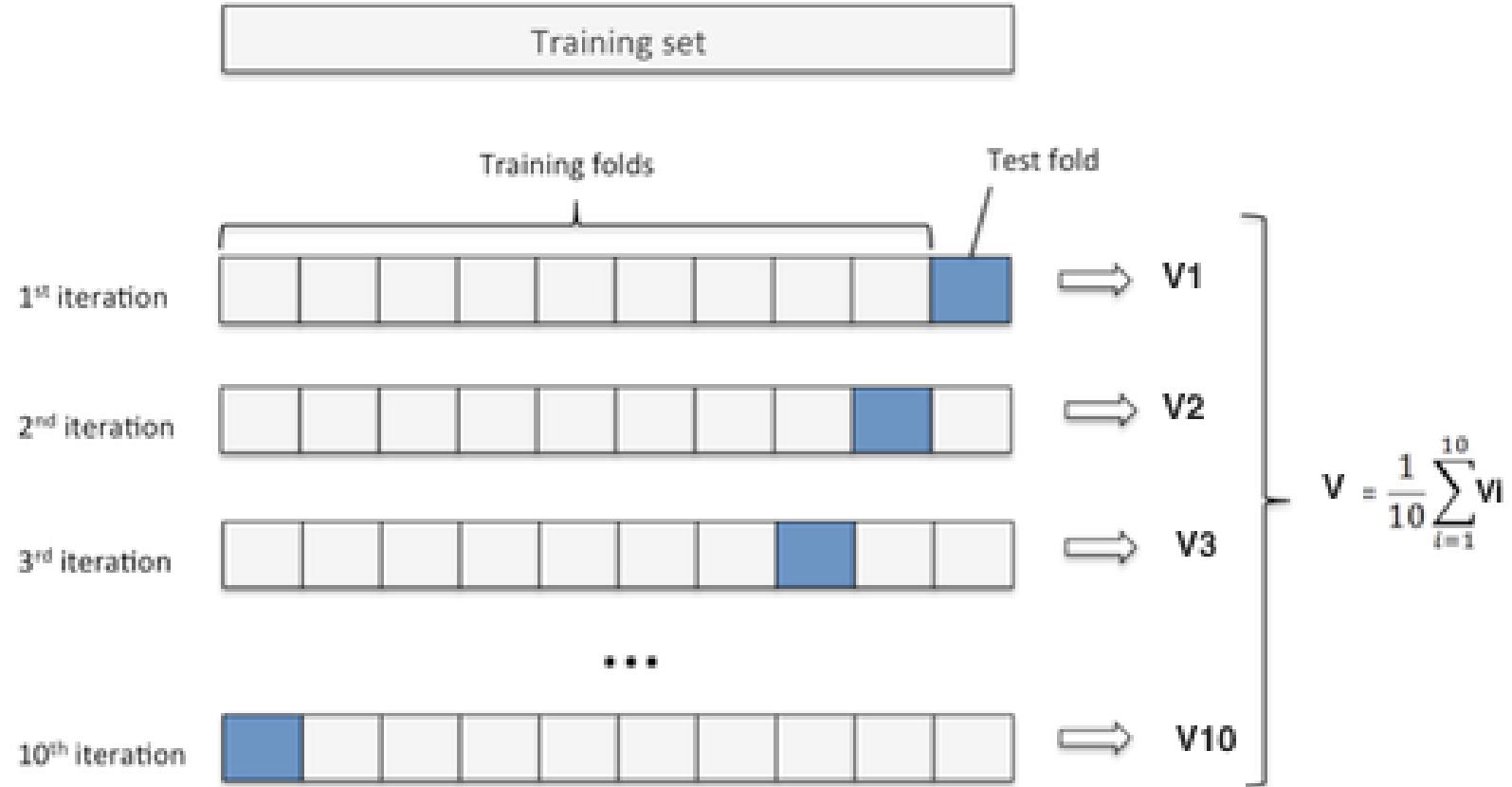
---

Utilizarea unui **singur set de validare** pentru a selecta dintre mai multe modele poate fi problematică - există posibilitatea de a se **supraadapta setului de validare (overfitting)**.

O soluție la problemele ridicate prin utilizarea unui singur set de validare este evaluarea fiecărui model pe **mai multe seturi de validare** și media performanței validării.

Se poate împărți **aleatoriu** setul de antrenament în antrenament și validare de mai multe ori, dar crearea aleatorie a acestor seturi poate crea scenariul în care caracteristicile importante ale datelor nu apar niciodată în extragerile noastre aleatorii.

# Cross Validation/Validare Încrucișată



# K-Fold Cross Validation/Validare Încrucișată de K ori

Pentru a ne asigura că fiecare observație din setul de date este inclusă în cel puțin un set de antrenament și cel puțin un set de validare, utilizăm **validarea de K ori**:

- Se împart datele în  $K$  submulțimi de înregistrări în mod uniform,  $\{C_1, \dots, C_K\}$  (toate cam aceeași proporție de date)
- creăm numărul  $K$  de perechi set de antrenament / validare, folosind una dintre cele  $K$  bucăți pentru validare și restul pentru antrenament.

Modelul antrenat pe fiecare set de antrenament în parte, notat  $\hat{f}_{C_{-i}}$ , este apoi evaluat pe setul de validare corespunzător,  $\hat{f}_{C_{-i}}(C_i)$ . **Validarea încrucișată** este performanța medie a modelului pentru toate seturile de validare:

$$CV(\text{Model}) = \frac{1}{K} \sum_{i=1}^K L(\hat{f}_{C_{-i}}(C_i))$$

unde  $L$  este funcția ce va fi optimizată.



# Selectarea Predictorilor: Cross Validation/Validare Încrucișată

**Întrebare:** Care este raportul corect dintre mărimele mulțimilor de anternament / validare / testare, cum aleg K?

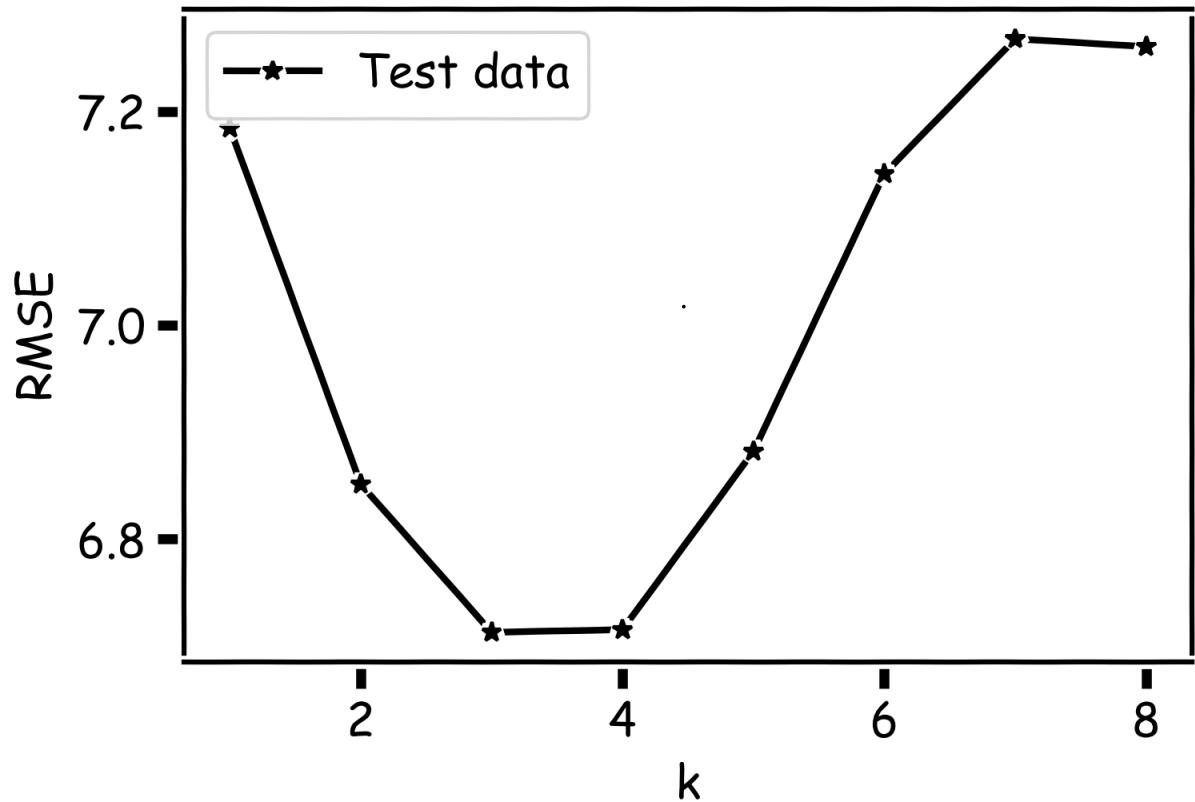
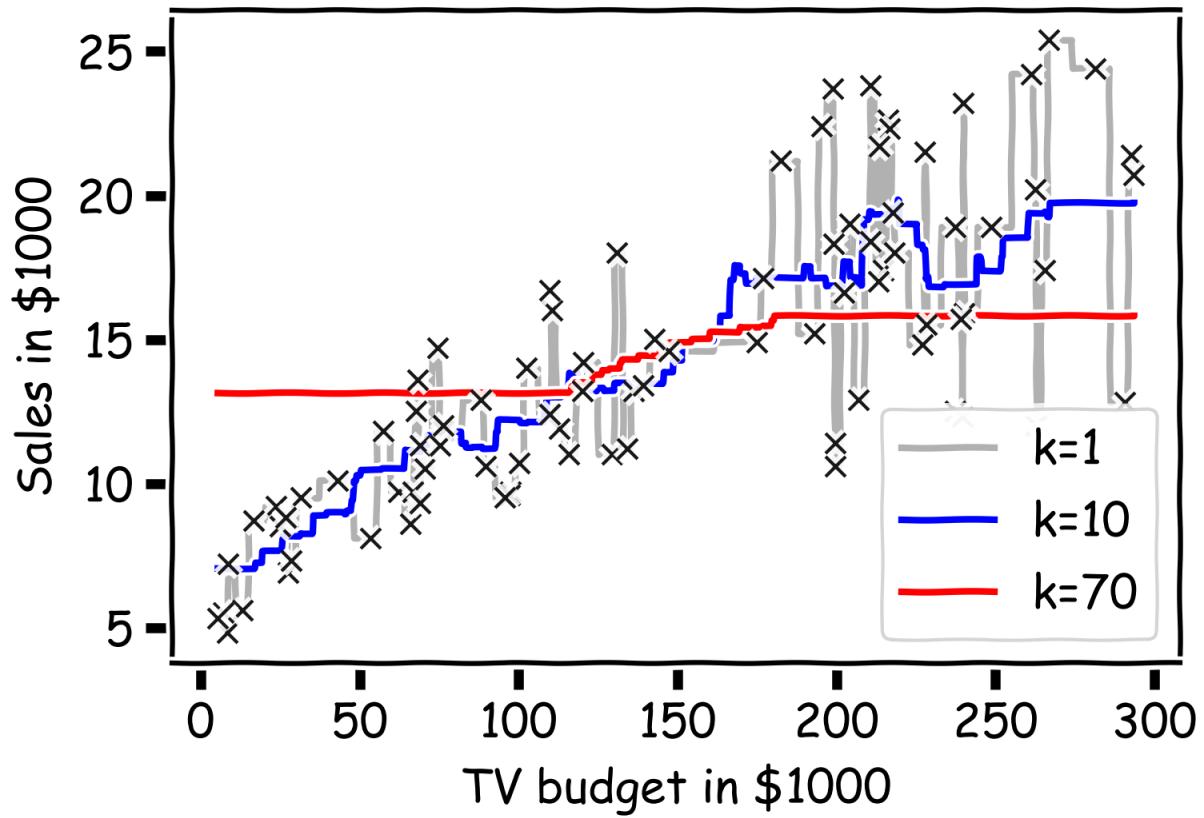
**Întrebare:** Care este diferența dintre predictorii multipli și regresia polinomială în selecția modelului?

Putem încadra problema selecției gradelor pentru modelele polinomiale ca o problemă de selecție a predictorilor:

care din predictorii  $\{x, x^2, \dots, x^m\}$ , are trebui selectați pentru model?

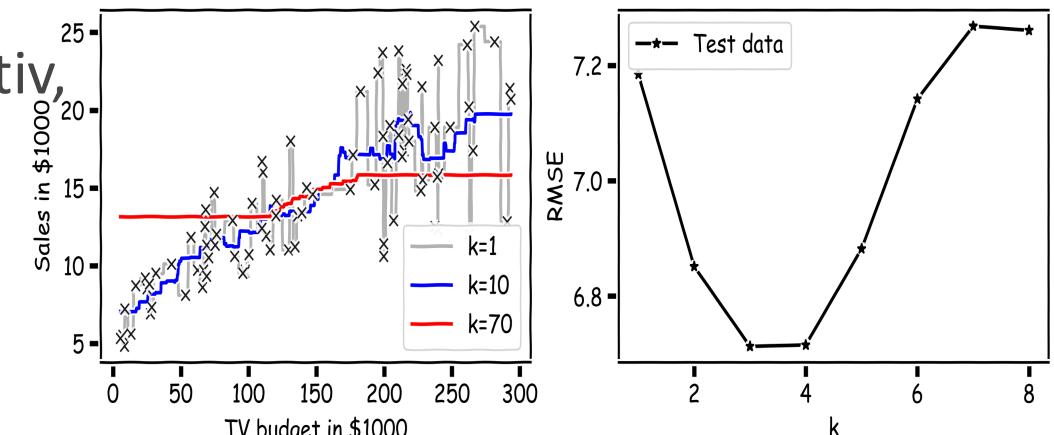


# kNN Revisited



# Din nou kNN

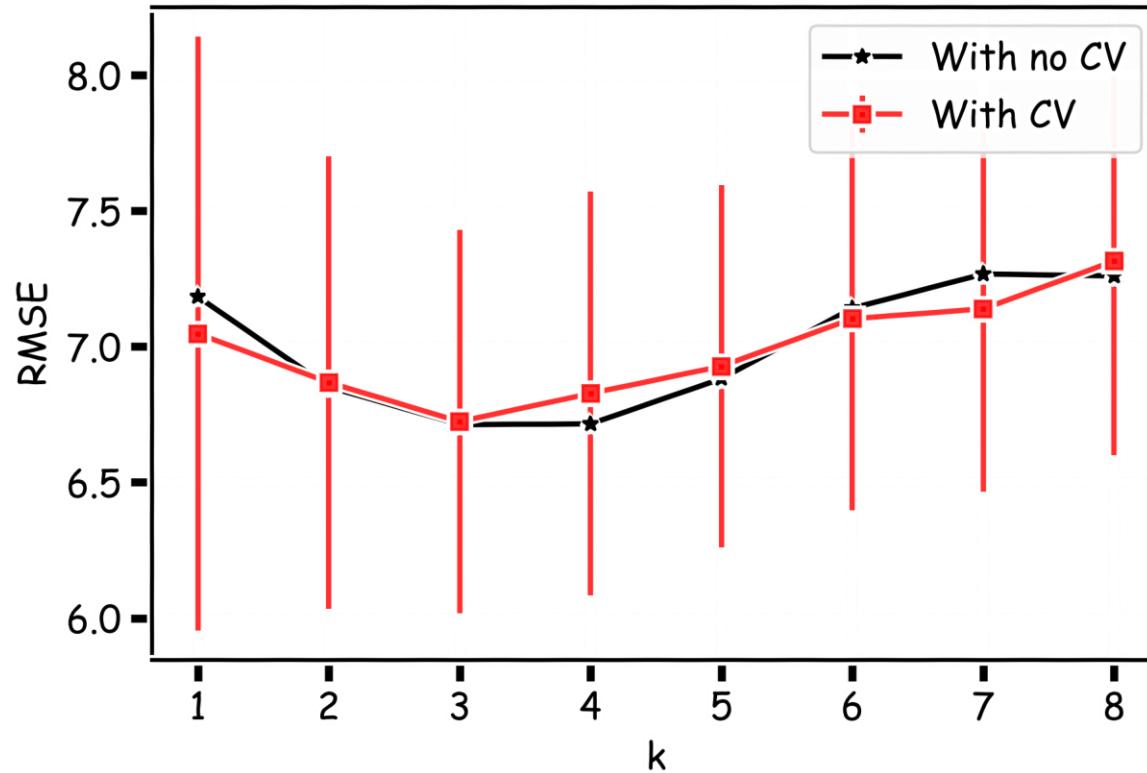
Recapitulăm primul nostru model simplu, intuitiv, non-parametric pentru regresie – modelul kNN. Am văzut că este foarte important să selectăm un  $k$  adecvat pentru date.



Când  $k$  este prea mic, modelul este foarte sensibil la zgomot (deoarece o nouă predicție se bazează pe foarte puțini vecini observați), iar când  $k$  este prea mare, modelul tinde să facă predicții constante.

Un mod principal de a alege  $k$  este prin **validare încrucișată de K ori**.

# De K ori cu $k=100$





# BIAS VS VARIANȚĂ

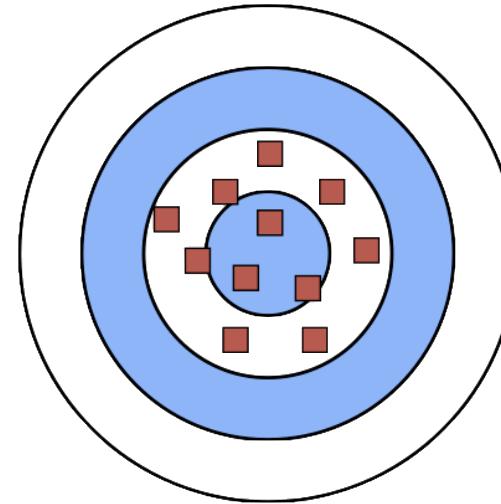
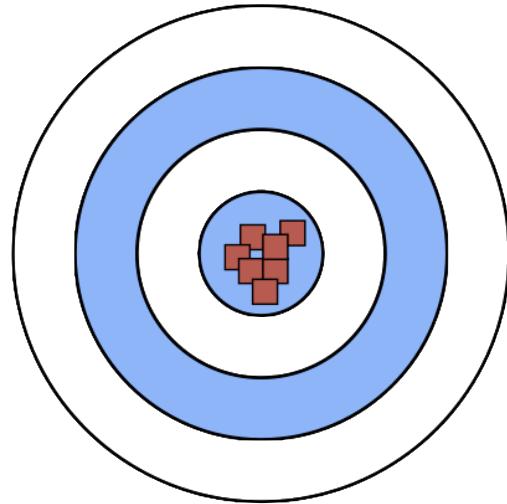




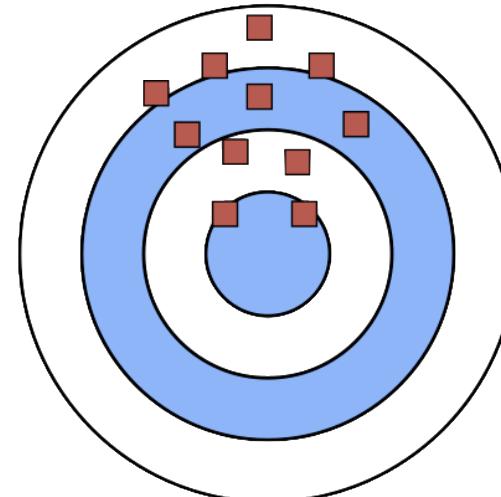
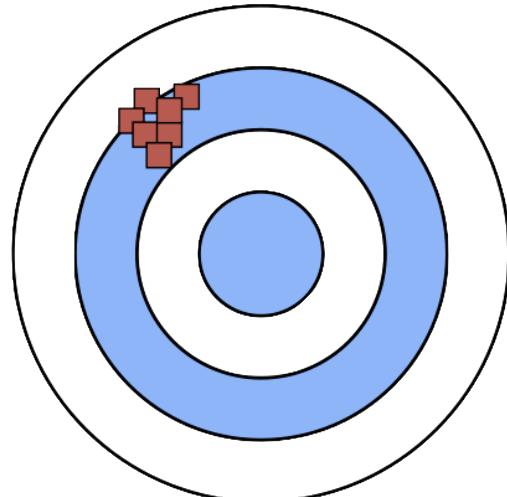
**Low Variance**  
(Precise)

**High Variance**  
(Not Precise)

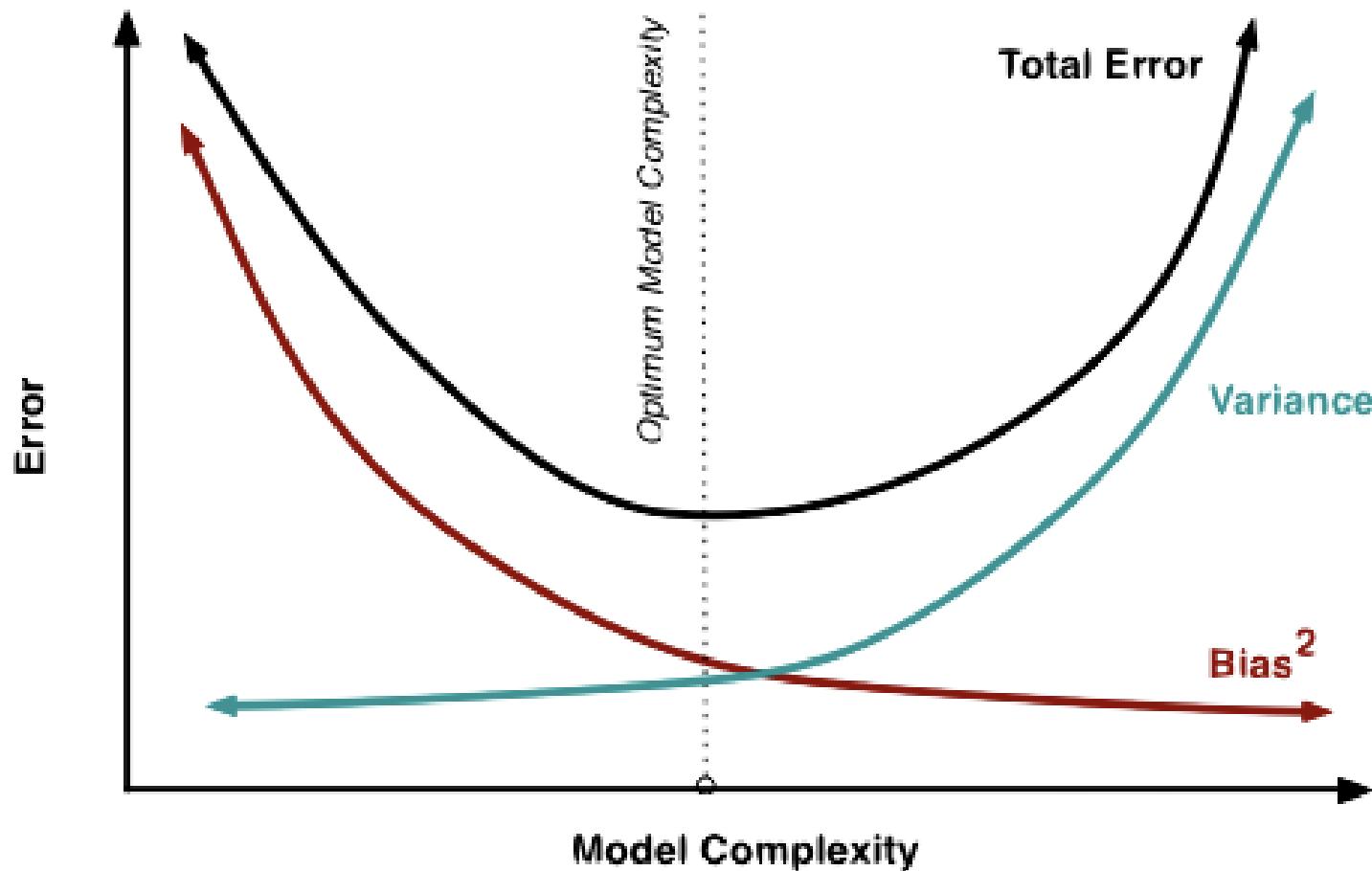
**Low Bias**  
(Accurate)



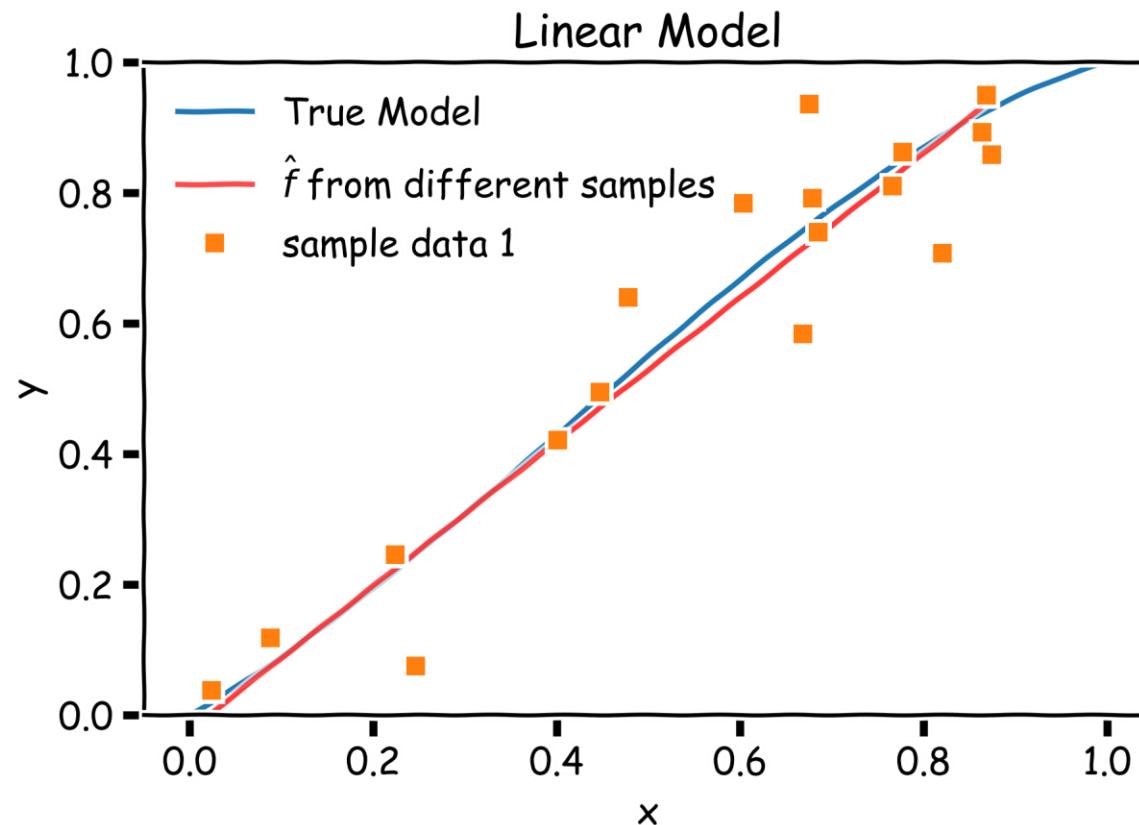
**High Bias**  
(Not Accurate)



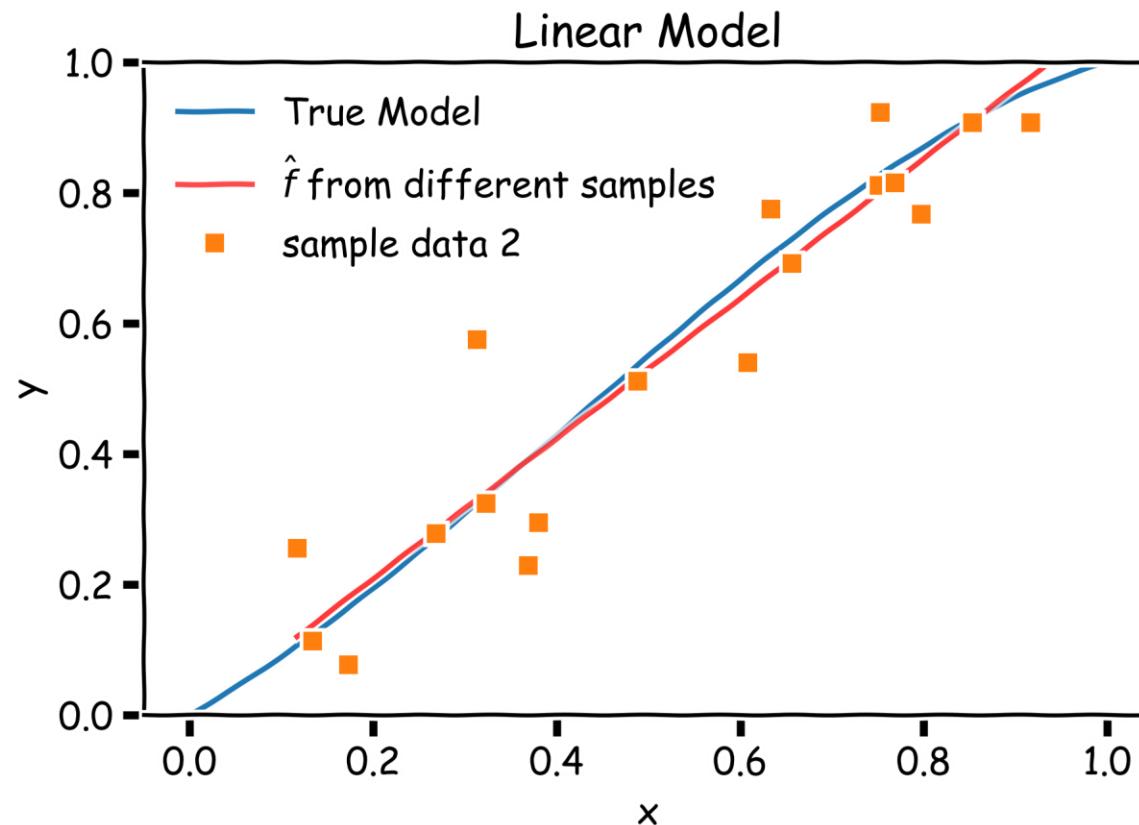
# Bias vs Variantă



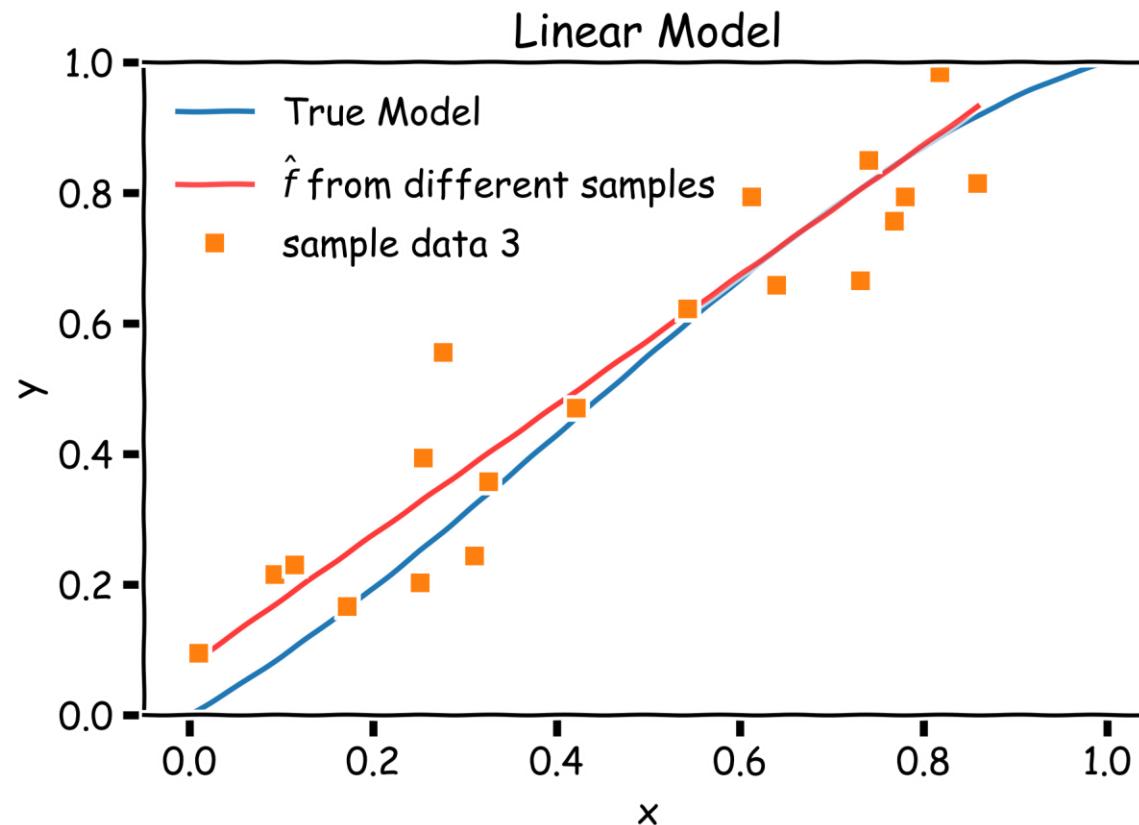
# Bias vs Variantă



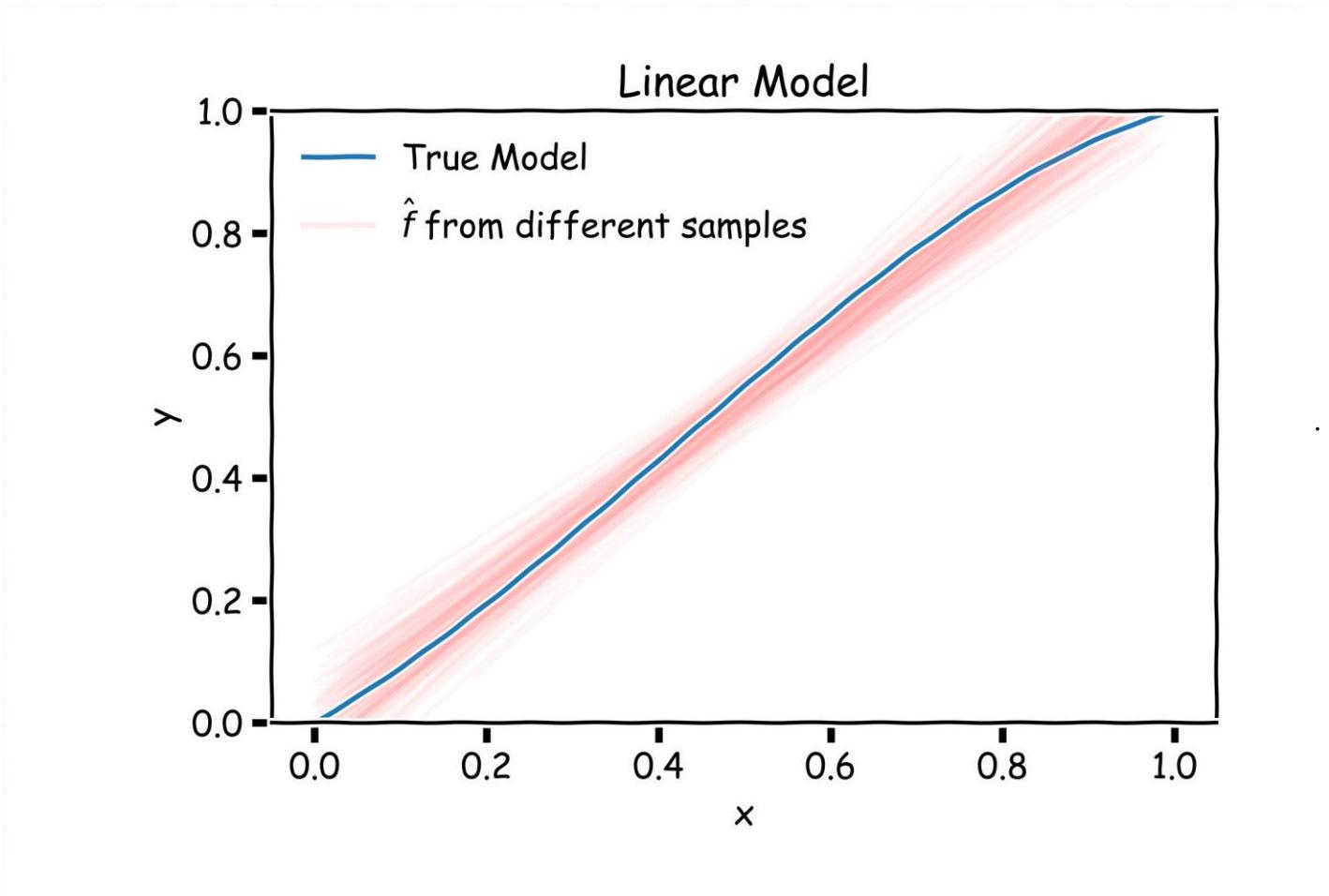
# Bias vs Variance



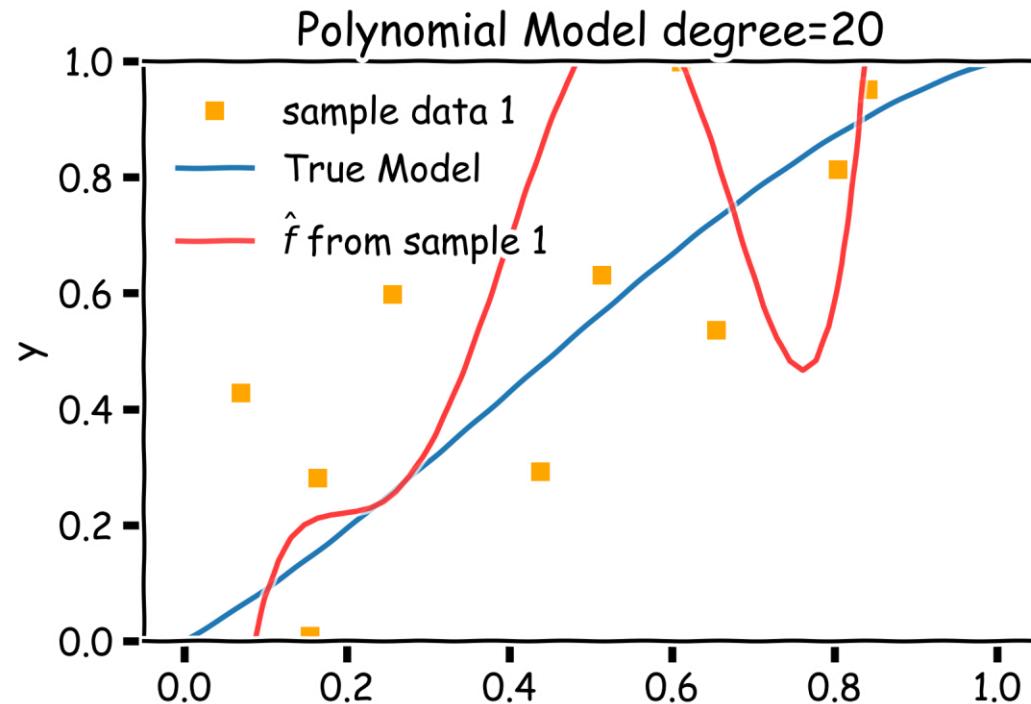
# Bias vs Variance



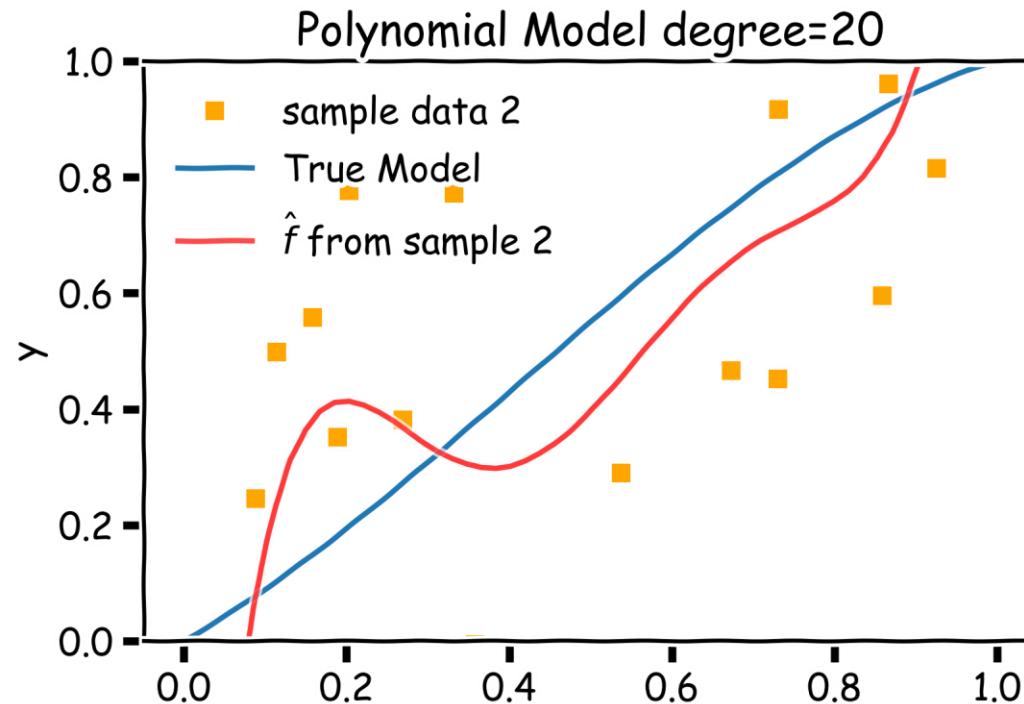
# Modele liniare: 20 de data points, 2000 simulations



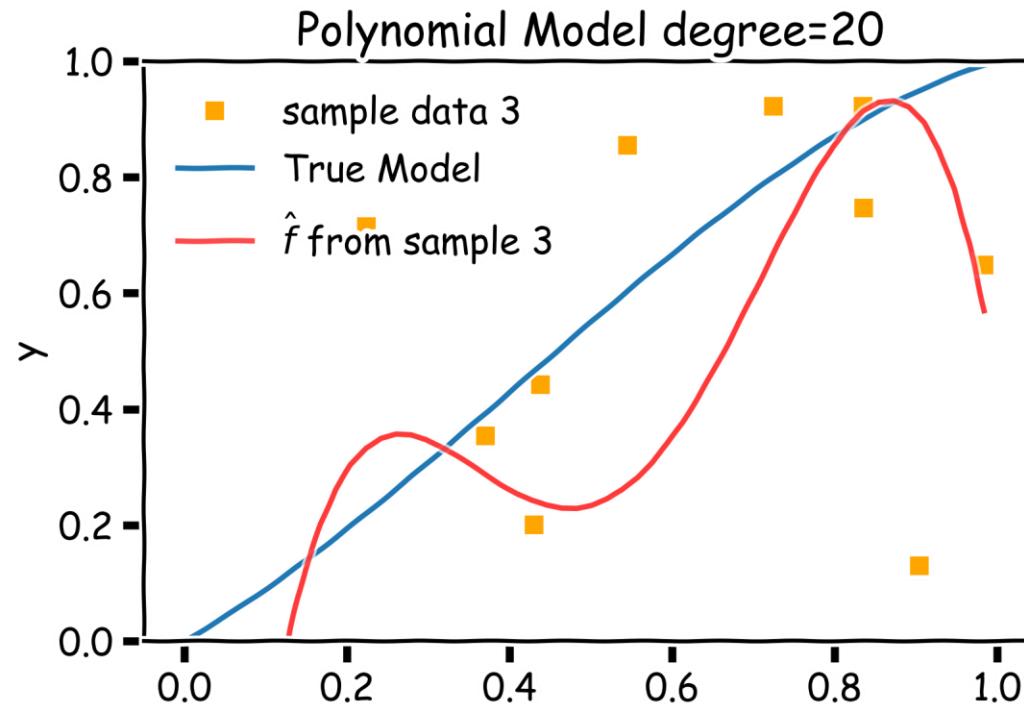
# Bias vs Variance



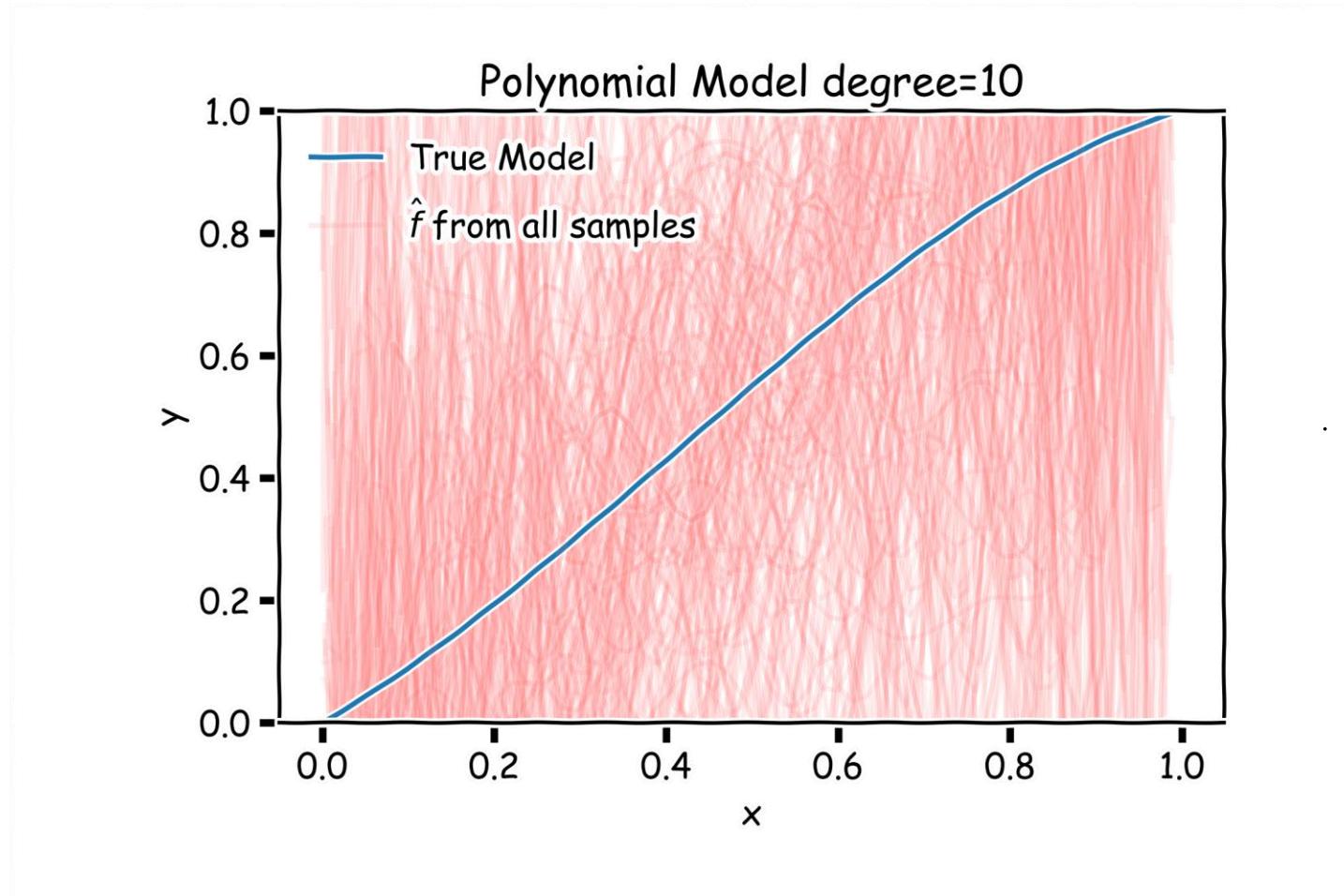
# Bias vs Variance



# Bias vs Variance



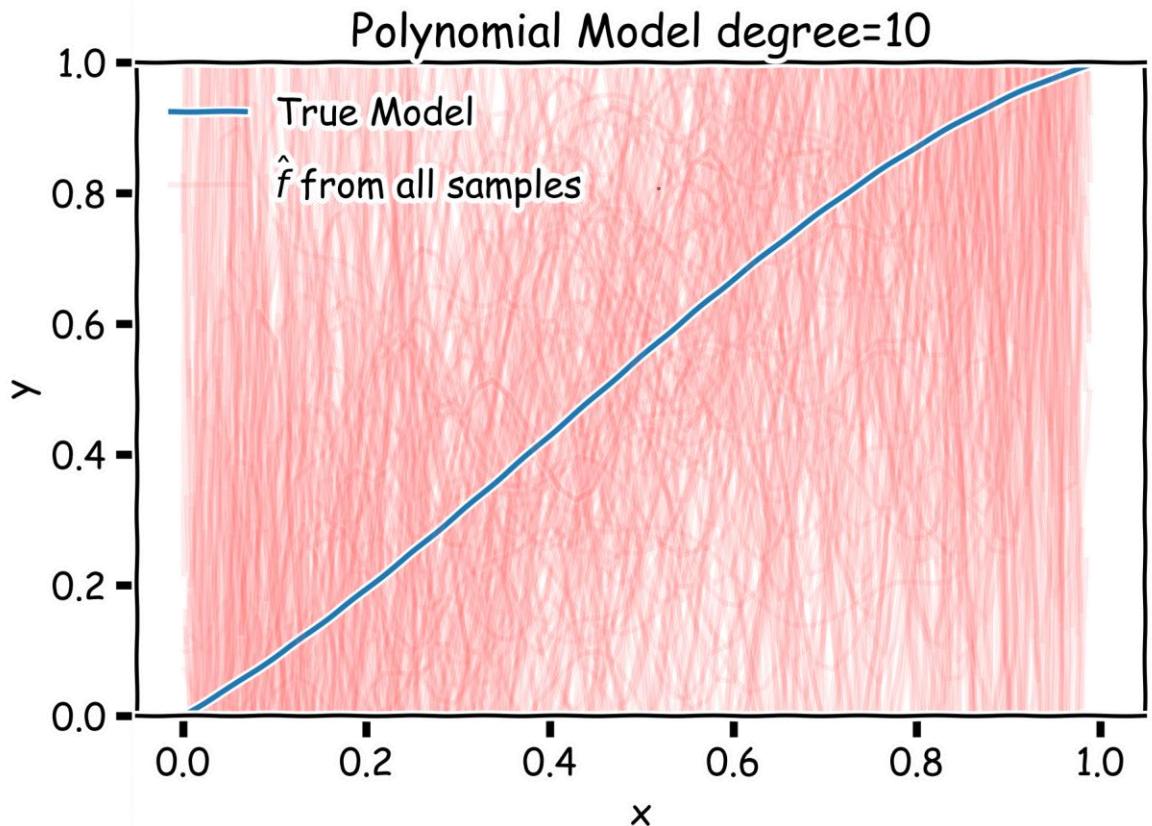
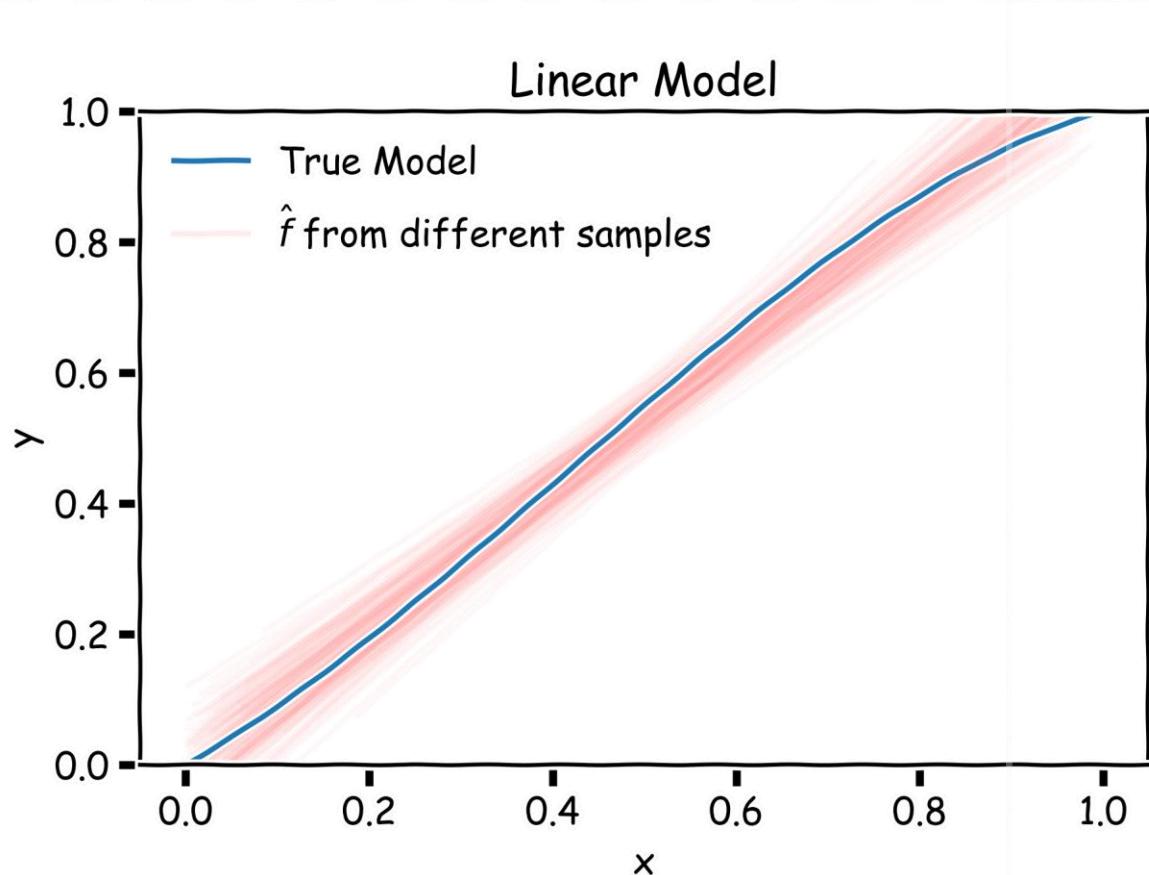
# Poly 10 degree models : 20 de data points, 2000 simulations



# Bias vs Variance

**Left:** 2000 best fit straight lines, each fitted on a different 20 point training set.

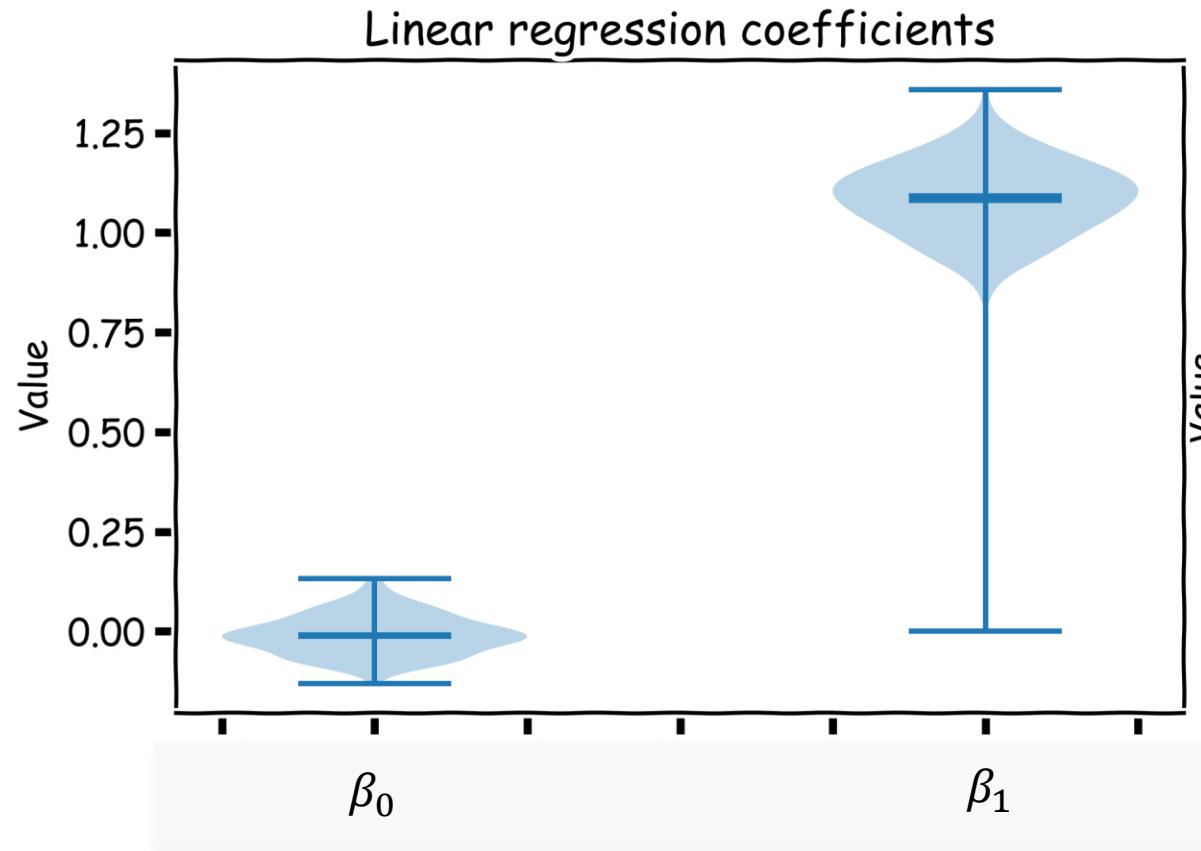
**Right:** Best-fit models using degree 10 polynomial



# Bias vs Variance

Stânga: Coeficienți regresie liniară

Dreapta: Coeficienți regresie polinomială de ordin 10





## **REGULARIZARE: LASSO ȘI RIDGE**

# Regularizare: O imagine de ansamblu

---

Ideea de regularizare se învârte în jurul modificării funcției de pierdere  $L$  (loss function; de ex MSE); în special, adăugăm un termen de regularizare care penalizează unele proprietăți specificate ale parametrilor modelului.

$$L_{reg}(\beta) = L(\beta) + \lambda R(\beta),$$

unde  $\lambda$  este un scalar care dă greutatea (sau importanța) termenului de regularizare  $R$ .

Antrenarea modelului utilizând funcția de pierdere modificată  $L_{reg}$  ar duce la parametrii ai modelului cu proprietăți dorite (specificate de  $R$ ).

# Regresie LASSO

---

Deoarece dorim să descurajăm valorile extreme în parametrii modelului, trebuie să alegem un termen de regularizare care să penalizeze magnitudinea parametrilor. Pentru funcția noastră de pierdere vom folosi din nou MSE:

$$L_{LASSO}(\beta) = \frac{1}{n} \sum_{i=1}^n |y_i - \beta^\top \mathbf{x}_i|^2 + \lambda \sum_{j=1}^J |\beta_j|.$$

Aici  $\sum_{j=1}^J |\beta_j|$  este norma  $l_1$  a vectorului  $\beta$

$$\sum_{j=1}^J |\beta_j| = \|\beta\|_1$$



# Regresie LASSO

Prin urmare, vom spune în mod ușual că  $L_{\text{LASSO}}$  este funcția de pierderi (loss-function) pentru regularizare  $L_1$ .

Găsirea parametrilor modelului,  $\beta_{\text{LASSO}}$ , care minimizează funcția de pierdere pentru regularizarea  $L_1$  se numește regresie **LASSO**.

```
In [ ]: from sklearn.linear_model import Lasso
```

```
In [22]: lasso_regression = Lasso(alpha=1.0, fit_intercept=True)
lasso_regression.fit(np.vstack((X_train, X_val)), np.hstack((y_train, y_val)))

print('Lasso regression model:\n {} + {}^T . x'.format(lasso_regression.intercept_, lasso_regression.coef_))
```

```
Lasso regression model:
10.424895873901445 + [ 0.24482603  3.48164594  1.84836859 -0.06864603 -0.          -0.
-0.02249766 -0.          0.          0.          0.          ]^T . x
```

```
In [23]: print('Train R^2: {}, test R^2: {}'.format(lasso_regression.score(np.vstack((X_train, X_val)),
                                                               np.hstack((y_train, y_val))),
                                                               lasso_regression.score(X_test, y_test)))
```

```
Train R^2: 0.48154992527975765, test R^2: 0.6846451270316087
```

# Regresie Ridge

---

Alternativ, putem alege un termen de regularizare care să penalizeze pătratele magitudinilor parametrilor. Astfel funcția noastră de pierdere regularizată este:

$$L_{Ridge}(\beta) = \frac{1}{n} \sum_{i=1}^n |y_i - \beta^\top \mathbf{x}_i|^2 + \lambda \sum_{j=1}^J \beta_j^2.$$

Aici  $\sum_{j=1}^J \beta_j^2$  este norma  $L_2$  a vectorului  $\beta$

$$\sum_{j=1}^J \beta_j^2 = \|\beta\|_2^2$$

# Ridge Regression

Prin urmare, vom spune în mod ușual că  $L_{\text{ridge}}$  este funcția de pierderi (loss-function) pentru regularizare  $L_2$ .

Găsirea parametrilor modelului  $\beta_{\text{ridge}}$  care minimizează funcția de pierdere pentru regularizarea  $L_2$  se numește regresie *ridge*.

```
In [ ]: from sklearn.linear_model import Ridge
```

```
In [20]: x_train = train[all_predictors].values
x_val = validation[all_predictors].values
x_test = test[all_predictors].values

ridge_regression = Ridge(alpha=1.0, fit_intercept=True)
ridge_regression.fit(np.vstack((X_train, X_val)), np.hstack((y_train, y_val)))

print('Ridge regression model:\n {} + {}^T . x'.format(ridge_regression.intercept_, ridge_regression.coef_))
```

```
Ridge regression model:
-525.7662550875951 + [ 0.24007312  8.42566029  2.04098593 -0.04449172 -0.01227935  0.41902475
 -0.50397312 -4.47065168  4.99834262  0.           0.           0.29892679]^T . x
```

```
In [21]: print('Train R^2: {}, test R^2: {}'.format(ridge_regression.score(np.vstack((X_train, X_val)),
                                                               np.hstack((y_train, y_val))),
                                                               ridge_regression.score(X_test, y_test)))
```

```
Train R^2: 0.5319764744847737, test R^2: 0.7881798111697319
```

# Alegerea lui $\lambda$

---

Atât la regresia ridge cât și la cea LASSO, vedem că cu cât este mai mare alegerea parametrului de regularizare  $\lambda$ , cu atât penalizăm mai mult valorile mari în  $\beta$

- Dacă  $\lambda$  e aproape de zero, regăsim MSE, adică regresiile ridge și LASSO regression se apropie de regresia obișnuită.
- Dacă  $\lambda$  este suficient de mare, termenul MSE în funcția de pierdere regularizată va fi nesemnificativ și termenul de regularizare  $\beta_{\text{ridge}}$  sau  $\beta_{\text{LASSO}}$  va fi forțat să se apropie de zero.

Pentru a evita alegeri ad-hoc pentru  $\lambda$  putem folosi validare încrușitată.

# Ridge – Complexitate Computatională

---

Soluția algebrică pentru regresia Ridge:

$$\beta = (X^T X + \lambda I)^{-1} X^T Y$$

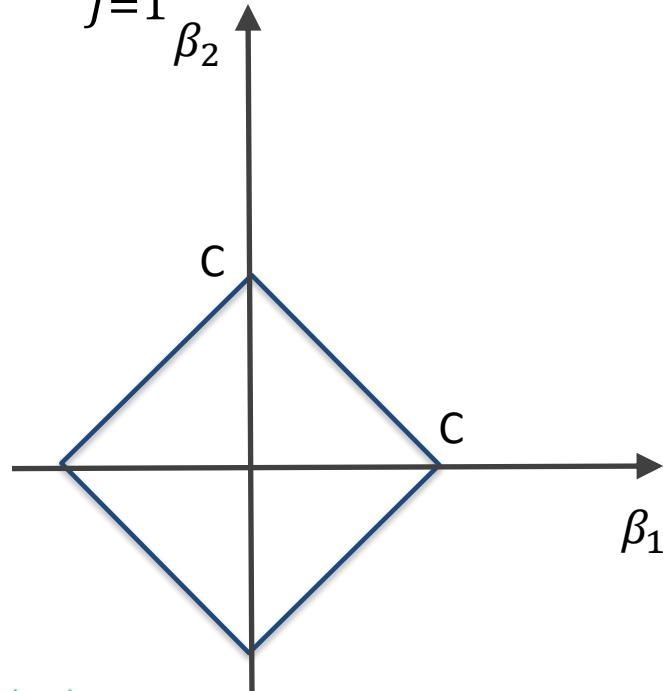
Soluția pentru regreisa Ridge/Lasso presupune 3 pași

- Selectarea lui  $\lambda$
- Găsirea minimului pentru regre funcția de pierdere ridge/Lasso regression (algebră liniară) și reținerea valorii  $R^2$  pe setul de date de test.
- Găsirea acelui  $\lambda$  care ne dă cea mai mare valoare  $R^2$

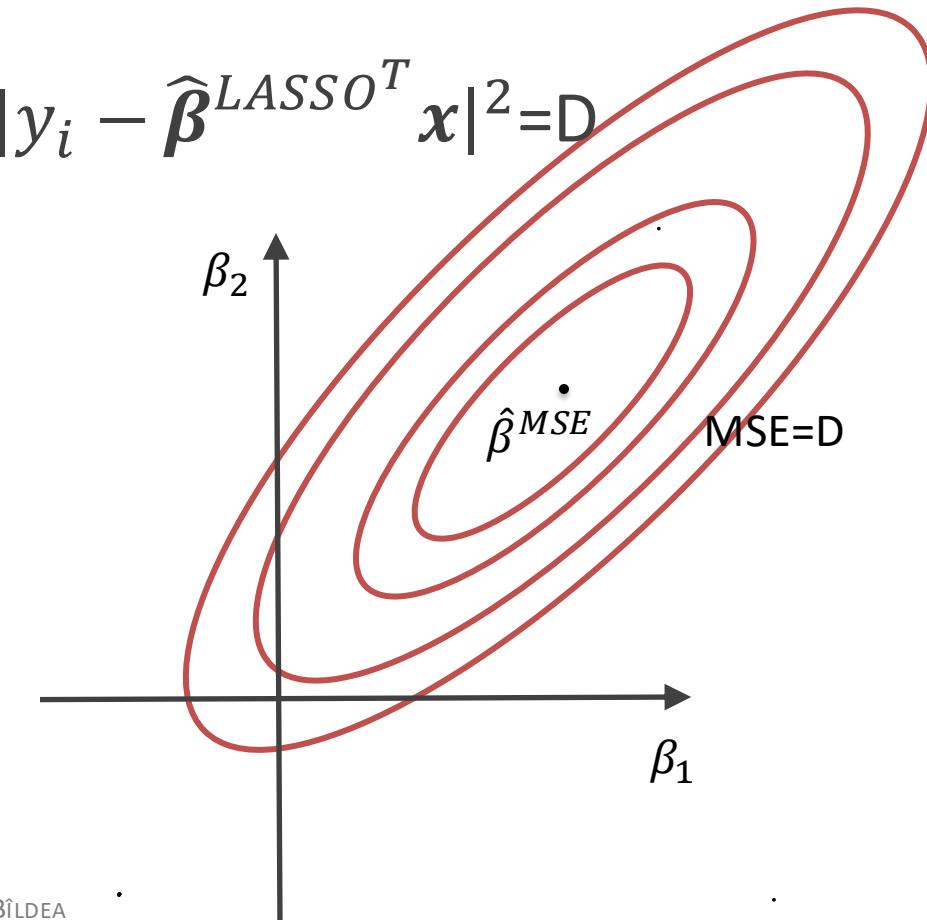
# Geometria regularizării (LASSO)

$$L_{LASSO}(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^n |y_i - \boldsymbol{\beta}^T \mathbf{x}|^2 + \lambda \sum_{j=1}^J |\beta_j| \quad \hat{\boldsymbol{\beta}}^{LASSO} = \operatorname{argmin} L_{LASSO}(\boldsymbol{\beta})$$

$$\lambda \sum_{j=1}^J |\hat{\beta}_j^{LASSO}| = C$$



$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{\boldsymbol{\beta}}^{LASSO T} \mathbf{x}|^2 = D$$

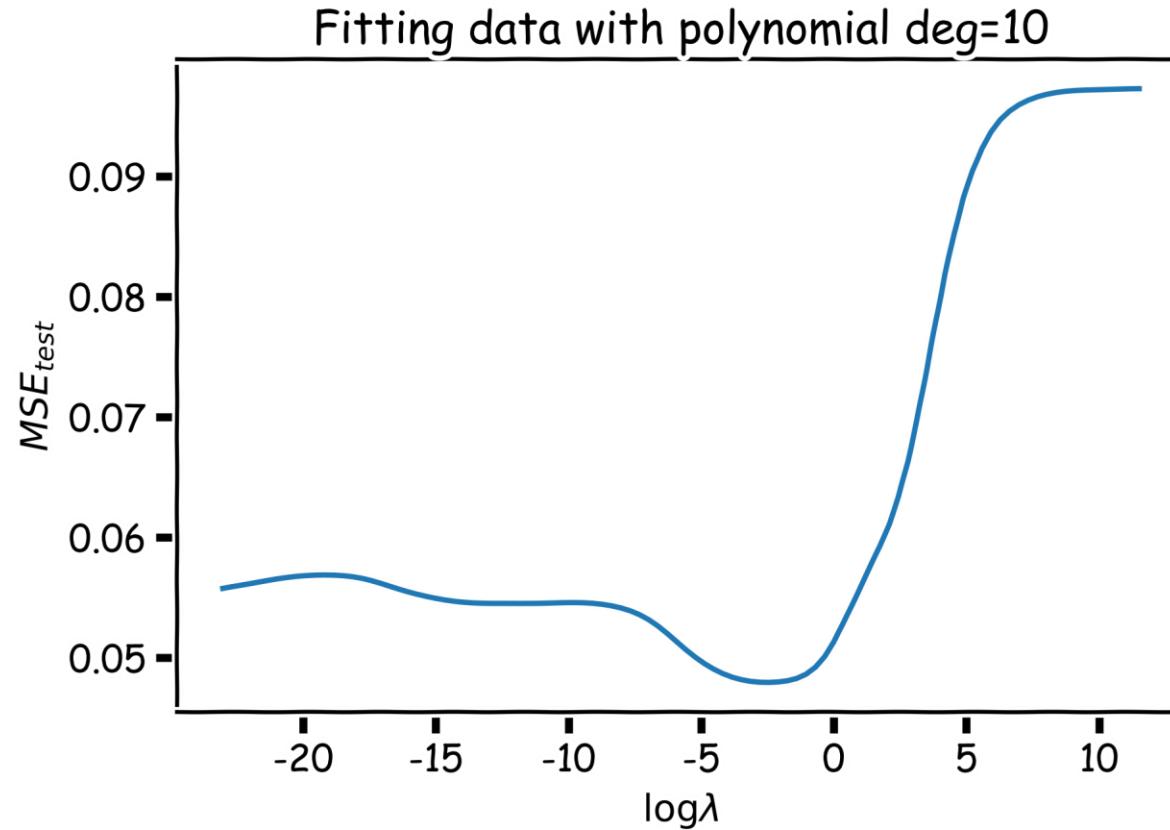


# Regularizare ridge doar pe **validare**: pas cu pas

1. Împărțire date în  $\{\{X, Y\}_{train}, \{X, Y\}_{validation}, \{X, Y\}_{test}\}$
2. Pentru  $\lambda$  in  $\{\lambda_{min}, \dots, \lambda_{max}\}$ :
  1. determină  $\beta$  care minimizează  $L_{ridge}$ ,  $\hat{\beta}_{Ridge}(\lambda) = (X^T X + \lambda I)^{-1} X^T Y$ , folosind datele de antrenament.
  2. Reține  $L_{MSE}(\lambda)$  folosind datele de validare.
3. selectarea lui  $\lambda$  care minimizează pierderea pe setul de validare,  
$$\lambda_{ridge} = \operatorname{argmin}_{\lambda} L_{MSE}(\lambda)$$
4. Reantrenarea modelului folosind reuniunea dintre setul de antrenament și validare,  $\{\{X, Y\}_{train}, \{X, Y\}_{validation}\}$ , obținem  $\hat{\beta}_{ridge}(\lambda_{ridge})$
5. raportăm MSE sau  $R^2$  pe  $\{X, Y\}_{test}$  folosind  $\hat{\beta}_{ridge}(\lambda_{ridge})$ .



# Ridge regularization with **validation** only: step by step



# Regularizare LASSO doar pe **validare**: pas cu pas

1. Împărțire date în  $\{\{X, Y\}_{train}, \{X, Y\}_{validation}, \{X, Y\}_{test}\}$
2. Pentru  $\lambda$  in  $\{\lambda_{min}, \dots, \lambda_{max}\}$ :
  1. determină  $\beta$  care minimizează  $L_{lasso}$ ,  $\hat{\beta}_{lasso}(\lambda)$ , , folosind datele de antrenament.
  2. Reține  $L_{MSE}(\lambda)$  folosind datele de validare.
3. selectarea lui  $\lambda$  care minimizează pierderea pe setul de validare  
$$\lambda_{lasso} = \operatorname{argmin}_{\lambda} L_{MSE}(\lambda)$$
4. Reantrenarea modelului folosind reuniunea dintre setul de antrenament și validare,  $\{\{X, Y\}_{train}, \{X, Y\}_{validation}\}$ , obținem  $\hat{\beta}_{lasso}(\lambda_{lasso})$
5. raportăm MSE sau  $R^2$  pe  $\{X, Y\}_{test}$  folosind  $\hat{\beta}_{lasso}(\lambda_{lasso})$

# Regularizare Ridge cu CV(Valid. Încr.): pas cu pas

	$\lambda_1$	$\lambda_2$	...	$\lambda_n$
$k_1$	$L_{11}$	$L_{12}$	..	..
$k_2$	$L_{21}$	...	..	..
...	..	...	..	..
$k_n$	..	...	..	..
$E[]$	$\bar{L}_1$	$\bar{L}_2$	..	$\bar{L}_n$

1. Se elimină  $\{X, Y\}_{test}$  din date
2. Spargerea datelor în K părți,  $\{\{X, Y\}_{train}^{-k}, \{X, Y\}_{val}^k\}$
3. pentru  $k$  in  $\{1, \dots, K\}$

1. pentru  $\lambda$  in  $\{\lambda_0, \dots, \lambda_n\}$ :

A. determină  $\beta$  care minimizează  $L_{ridge}$ ,  $\hat{\beta}_{ridge}(\lambda, k) = (X^T X + \lambda I)^{-1} X^T Y$ , folosind datele de antrenament,  $\{X, Y\}_{train}^{-k}$ .

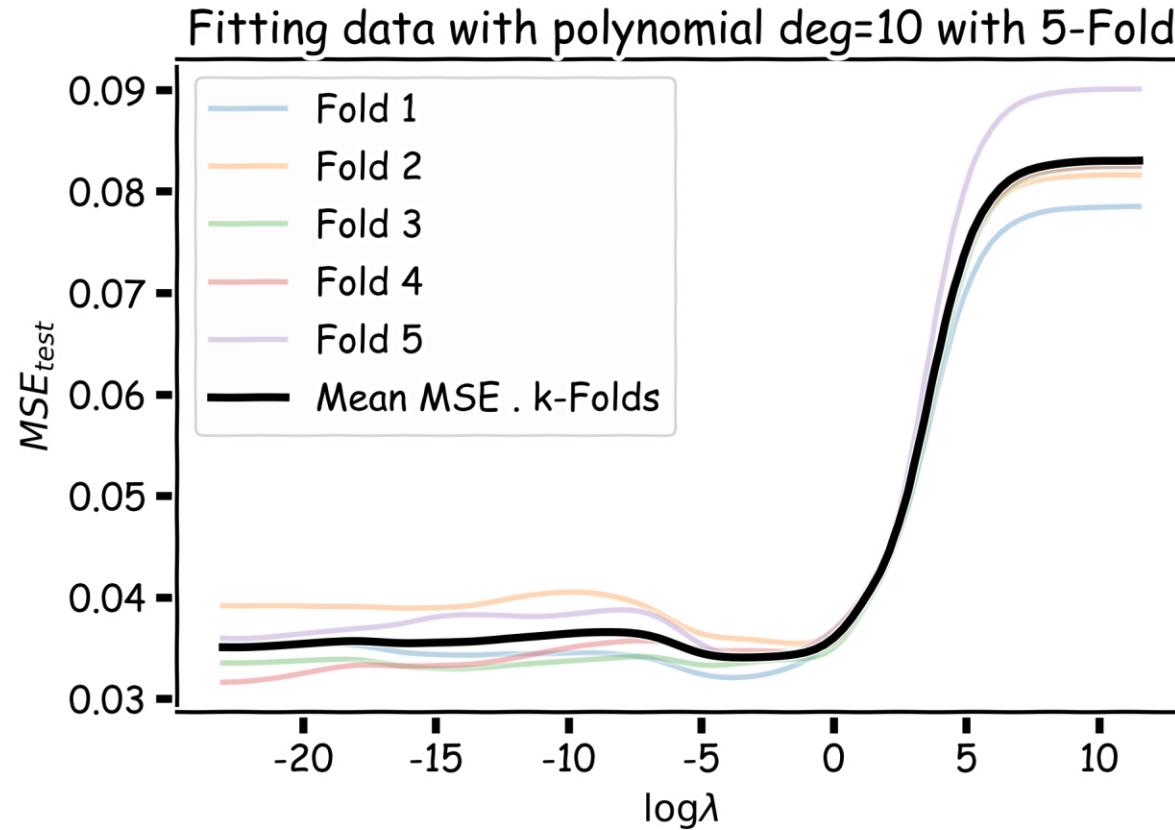
B. Reține  $L_{MSE}(\lambda, k)$  datele de validare  $\{X, Y\}_{val}^k$

Aici vom avea o matrice 2-D, pe linii cu valori diferite ale lui  $k$ , coloanele pentru diferite valori ale lui  $\lambda$ .

4. Facem media  $L_{MSE}(\lambda, k)$  pentru fiecare  $\lambda$ ,  $\bar{L}_{MSE}(\lambda)$ .
5. Găsim  $\lambda$  care minimizează  $\bar{L}_{MSE}(\lambda)$ ,  $\rightarrow \lambda_{ridge}$ .
6. Reantrenăm modelul folosind întregul set de antrenamenta,  $\{\{X, Y\}_{train}, \{X, Y\}_{val}\}$ ,  $\rightarrow \hat{\beta}_{ridge}(\lambda_{ridge})$
7. Raportează MSE sau  $R^2$  pe  $\{X, Y\}_{test}$  pentru  $\hat{\beta}_{ridge}(\lambda_{ridge})$



# Regularizare Ridge cu CV(Valid. Încr.): pas cu pas



# Alegerea predictorilor prin regularizare

---

Deoarece regresia LASSO are tendința de a produce estimări zero pentru un număr de parametri ai modelului - spunem că soluțiile LASSO sunt rare - considerăm că **LASSO este o metodă pentru selecția predictorilor**.

Mulți preferă să utilizeze LASSO pentru selecția variabilă (precum și pentru suprimarea valorilor extreme ale parametrilor) în loc de selecție treptată, deoarece LASSO evită problemele statistice care apar în selecția treptată.

**Întrebare:**

Care sunt avantajele și dezavantajele celor două abordări?



