

## Técnicas e Algoritmos em Ciência de Dados

### Tarefa 2

Esta tarefa deve ser enviada até 6 de Abril 2023, às 11:00 da manhã.  
As submissões tardias serão penalizadas em 10% por hora de atraso.

### Tópicos avaliados

Este curso visa proporcionar aos alunos uma boa compreensão da regressão linear com funções de base polinomial e classificação com vizinhos K-mais próximos. Além disso, os alunos aprenderão conceitos-chave relacionados a como a capacidade de generalização dos modelos se relaciona com a complexidade do modelo e o tamanho do conjunto de dados. Os alunos também serão introduzidos ao conceito de usar um conjunto de validação para selecionar parâmetros ótimos. No final deste curso, os alunos terão adquirido uma sólida compreensão desses conceitos-chave e estarão equipados com as habilidades para aplicá-los em cenários práticos - o problema de classificação usando os vizinhos K-mais próximos já usa dados do mundo real.

### Instruções

#### Identificador

Por favor, use o número aleatório de 6 dígitos que você usou para o primeiro curso e escreva-o na primeira célula do notebook. Certifique-se de manter uma cópia desse número, pois ele será usado para fornecer o feedback.

#### Submissão

Envie seus arquivos através do ECLASS. Os arquivos que você envia não podem ser substituídos por mais ninguém e não podem ser lidos por nenhum outro aluno. No entanto, você pode substituir seu envio quantas vezes quiser, reenviando, embora apenas a última versão enviada seja mantida.

*Se você tiver problemas, no último minuto, envie sua tarefa por e-mail como um anexo em [alberto.paccanaro@fgv.br](mailto:alberto.paccanaro@fgv.br) com o assunto "URGENTE – SUBMISSÃO TAREFA 2". No corpo da mensagem, explique o motivo de não enviar através do ECLASS.*

### IMPORTANTE

- Seu envio consistirá em um único notebook do Python implementando suas soluções.
- **O nome do arquivo será o número aleatório que o identifica (por exemplo, 568423.ipynb)**
- Esta tarefa consiste em 2 partes. Certifique-se de que o código de ambas as partes é colocado nas células de código relevantes no notebook.
- **NÃO ENVIE NENHUM CONJUNTO DE DADOS**, apenas o código.
- Qualquer função auxiliar que você usará deve ser incluída no notebook – não envie scripts adicionais.

<p><b>Todo o trabalho que você enviar deve ser exclusivamente seu próprio trabalho. As submissões serão verificadas para isso.</b></p>
--

### Critérios de avaliação

Esta tarefa é obrigatória e vale 10% da sua nota final total para este curso. Para obter notas máximas para cada pergunta, você deve respondê-la corretamente, mas também completamente. Daremos pontos para códigos bem estruturados.

## Parte 1 – Ajuste Polinomial (valor: 50%)

Faça o download no ECLASS do arquivo:

- CW2\_template.ipynb

Neste exercício, você explorará o efeito que o grau do polinômio e o tamanho dos dados de treinamento têm sobre o desempenho e a capacidade de generalização do modelo.

1. Gere um conjunto de dados 2D de 15 pontos  $(x_i, y_i)$ , para  $i = 1..15$ , usando uma onda senoidal perturbada por um pequeno ruído gaussiano— isso é muito semelhante ao que você fez na aula de laboratório.
    - a. O  $x_i$  deve ser igualmente espaçado no intervalo  $[0 - 10]$ .
    - b. Use  $y_i = 4 * \sin(x_i) + \epsilon$  onde  $\epsilon$  é o ruído gaussiano (com  $\mu = 0$  e  $\sigma = 1$ ).
  2. Divida os pontos aleatoriamente em conjuntos de treinamento e teste de tamanhos 10 e 5, respectivamente.
  3. Aprenda os pesos da regressão linear para modelos polinomiais de grau  $M$  para  $M = 0..9$ . Para cada valor de  $M$  calcular a raiz do erro quadrático médio (RMSE - Root Mean Squared Error) para os conjuntos de treinamento e teste e plotar esses valores em relação ao  $M$ . Sua figura deve ser semelhante à Figura 1.
  4. Na última parte deste exercício, você terá que criar mais pontos para treinamento (usando a mesma onda senoidal perturbada por um pequeno ruído gaussiano conforme descrito no ponto 1); para o teste, você continuará a usar os mesmos 5 pontos que você usou nos itens 1 a 3. Aprenda os pesos da regressão linear para modelos polinomiais de grau  $M = 9$  for conjuntos de treinamento de tamanho  $N$  com  $N = 10: 500: 10$  (isto é, de  $N = 10$  a  $N = 500$  com passos de tamanho 10). Para cada valor de  $N$  calcular o erro quadrático médio (RMSE - Root Mean Squared Error) para os conjuntos de treinamento e teste e plotar esses valores em relação a  $N$ . Sua figura deve ser semelhante à Figura 2.
- Observe que, para as partes 3 e 4, você deve escrever seu código do zero e não pode usar funções existentes, como PolynomialFeatures, LinearRegression ou mean\_squared\_error.
  - *Observação: neste exercício não estamos usando o conjunto de validação, porque nosso objetivo não é escolher um modelo específico, mas sim analisar o comportamento da família de modelos.*

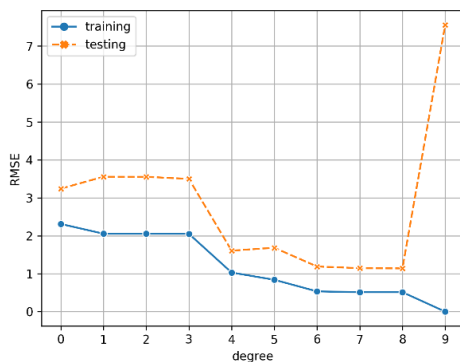


Figure 1

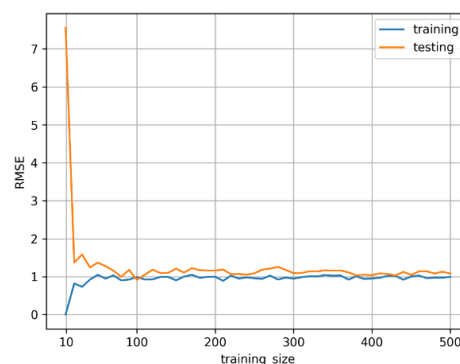


Figure 2

## Parte 2 – Classificação com K-NN (valor: 50%)

Baixe os dados do ECLASS:

- `ionosphere_data_cleaned.csv`

Neste exercício, você implementará o algoritmo K-nearest neighbours (K-NN) (K vizinhos mais próximos) para classificação binária e o utilizará para classificar o conjunto de dados Ionsphere. Você também precisará determinar o valor ideal de K usando o F1-score em um conjunto de validação e deverá relatar o desempenho do conjunto de testes com base no F1-score.

**Conjunto de dados:** O conjunto de dados da ionosphere contém dados de radar coletados de um sistema de radar em Goose Bay, Canadá. O conjunto de dados contém 351 observações, com 34 atributos cada. A última coluna representa a classe, que é "boa" (1) ou "ruim" (0).

1. Carregue o conjunto de dados do Ionsphere em um data frame e divida-o em um conjunto de treinamento (80%), um conjunto de validação (10%) e um conjunto de testes (10%).
2. Você escreverá um código que implementa o algoritmo K-NN para classificação binária. Use a distância euclidiana como a métrica de distância.
3. Use o algoritmo K-NN no conjunto de treinamento para  $K = 1:31:2$  (isto é, de  $K = 1$  até  $K = 31$  com passos de tamanho 2) e avalie seu desempenho no conjunto de validação usando o F1-score. Plote o F1-score do conjunto de validação em relação aos valores que foram tentados para  $K$ . Seu gráfico deve ser semelhante ao da Figura 3.
  - a. *Observação: é melhor se você considerar apenas valores ímpares de  $K$ , considerando que você está prevendo a classe com base na maioria dos vizinhos.*
  - b. *Você também terá que implementar o F1-score do zero.*
4. Escolha o valor de  $K$  que dá a melhor pontuação F1 no item 3 e reporte o F1-score do algoritmo K-NN no conjunto de testes usando o valor selecionado de  $K$ .

**Nota:** Você não tem permissão para usar nenhuma biblioteca Python, como scikit-learn, para implementar o algoritmo K-NN ou para calcular distâncias ou o F1-score. Você pode usar *numpy* ou outras bibliotecas básicas para operações de matriz.

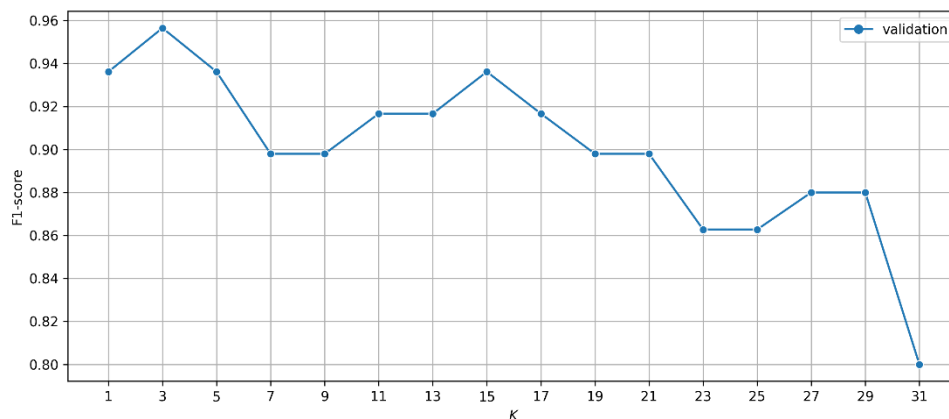


Figura 3