MTH 8302 - Modèles de Régression et d'Analyse de Variance

Leçon 3 : Méthodes de Sélection de Modèles et Généralisation en Apprentissage Statistique

Polytechnique Montréal - Hiver 2025

Chiheb Trabelsi

March 12, 2025

Table des Matières

Théorie de la Décision et Compromis Biais-Variance

Théorie de la Décision et Compromis Biais-Variance

Théorie de la Décision et Compromis Biais-Variance de $\hat{f}$

- On considère un modèle de régression, ou tout autre modèle de prédiction où l'on a une variable cible Y est reliée à une entrée $\mathbf{X}$ selon l'équation:

$$Y = f(\mathbf{X}) + \boldsymbol{\epsilon}, \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n).$$

- L'objectif est d'analyser comment l'estimateur $\hat{f}$, obtenu à partir d'un jeu de données d'entraînement $\mathcal{D}$, se comporte lorsqu'on fait varier $\mathcal{D}$.
- Nous évaluons la distribution des prédictions $\hat{f}(\boldsymbol{x})$ pour une observation $\boldsymbol{x}$ donnée en moyenne sur différents ensembles $\mathcal{D}$.

Théorie de la Décision et Compromis Biais-Variance de $\hat{f}$

- **Définition :** Dans le contexte d¡un problème de prédiction, un estimateur ponctuel $\hat{\boldsymbol{\theta}}$ D'un paramètre $\boldsymbol{\theta}$, est défini comme :

$$\begin{aligned}\hat{\boldsymbol{\theta}} = g(\mathcal{D}) &= g(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, Y_1, \ldots, Y_n) \\ &= g(\{(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_n, Y_n)\}) \\ &= g(\mathbf{X}, \mathbf{Y}), \quad \text{où :}\end{aligned}$$

  - g est une fonction des données de l'échantillon des données $\mathbf{X}_1, \ldots, \mathbf{X}_n$ de $\mathbf{X}$ et de leur cibles correspondantes $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top$.
  - $\mathcal{D}$ est la paire du jeux de données (*dataset*) constitué par la $(\mathbf{X}, \mathbf{Y})$
- **Exemple de la Régression Linéaire :** Dans le cas de la régression linéaire multiple, où l'on considère le modèle :

$$\mathbf{Y} = \underbrace{\mathbf{X}\boldsymbol{\beta}}_{f_{\boldsymbol{\beta}}(\mathbf{X})} + \boldsymbol{\epsilon}, \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n),$$

  - Le paramètre $\boldsymbol{\theta}$ à estimer est le vecteur $\boldsymbol{\beta} \in \mathbb{R}^n$.
  - $f_{\boldsymbol{\beta}}$ est la fonction de prédiction de $\mathbf{Y}$ tel que $f_{\boldsymbol{\beta}}(\mathbf{X}) = \mathbf{X}\boldsymbol{\beta}$.

Théorie de la Décision et Compromis Biais-Variance de $\hat{f}$

- Les méthodes d'estimation des paramètres du modèle de régression linéaires définissent des fonctions d'estimation de $\boldsymbol{\beta}$ où l'on a :

$$\begin{cases} \hat{\boldsymbol{\beta}}_{\text{MCO}} = g_{\text{MCO}}(\mathbf{X}, \mathbf{Y}) = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} \\ \hat{\boldsymbol{\beta}}_{\text{MV}} = g_{\text{MV}}(\mathbf{X}, \mathbf{Y}) = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} \end{cases}$$

- g_{MCO} et g_{MV} sont respectivement les fonctions d'estimation de $\boldsymbol{\beta}$ par la méthode des moindres carrés ordinaires et par la méthode de maximum de vraisemblance.
- La fonction de prédiction $f_{\boldsymbol{\beta}}$ est elle estimée par $f_{\hat{\boldsymbol{\beta}}_{\text{MCO}}}$ et d'une manière équivalente par $f_{\hat{\boldsymbol{\beta}}_{\text{MV}}}$ où l'on a :

$$\hat{f}_{\boldsymbol{\beta}}(\mathbf{X}) = f_{\hat{\boldsymbol{\beta}}_{\text{MCO}}}(\mathbf{X}) = f_{\hat{\boldsymbol{\beta}}_{\text{MV}}}(\mathbf{X}) = \underbrace{\mathbf{X}\hat{\boldsymbol{\beta}}_{\text{MCO}}}_{\hat{\mathbf{Y}}_{\text{MCO}}} = \underbrace{\mathbf{X}\hat{\boldsymbol{\beta}}_{\text{MV}}}_{\hat{\mathbf{Y}}_{\text{MV}}}.$$

- Comme nous pouvons le constater, $\hat{f}_{\boldsymbol{\beta}}$, l'estimateur de $f_{\boldsymbol{\beta}}$ dépend de la variable cible $\mathbf{Y}$ et des données de l'échantillon $\mathbf{X}_1, \ldots, \mathbf{X}_n$ de $\mathbf{X}$.

Théorie de la Décision et Compromis Biais-Variance de $\hat{f}$

- Pour une réalisation donnée $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}$ de l'échantillon aléatoire $\mathbf{X}_1, \ldots, \mathbf{X}_n$ de $\mathbf{X}$, $\hat{f}_{\boldsymbol{\beta}}$ **dépend de ces réalisations et des observations cibles correspondantes $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$.** On aura :

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{\text{MCO}} = \hat{\boldsymbol{\beta}}_{\text{MV}} &= g_{\text{MCO}}(\mathcal{D} = \boldsymbol{x_1}, \ldots, \boldsymbol{x_n}, \boldsymbol{y}) \\ &= g_{\text{MV}}(\mathcal{D} = \{(\boldsymbol{x_1}, y_1), \ldots (\boldsymbol{x_n}, y_n)\}) \\ &= ({\mathbf{X}_{\text{train}}}^\top \mathbf{X}_{\text{train}})^{-1} {\mathbf{X}_{\text{train}}}^\top \boldsymbol{y}, \quad \text{où :}\end{aligned}$$

- $\mathbf{X}_{\text{train}} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_n})^\top \in \mathbb{R}^{n \times (p+1)}$ ici dénote la matrice contenant les réalisation $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}$ de l'échantillon aléatoire $\mathbf{X} = (\mathbf{X_1}, \ldots, \mathbf{X_n})^\top \in \mathbb{R}^{n \times (p+1)}$.
- L'ensemble $\mathcal{D}_{\text{train}}$ formé par les réalisations et leurs cibles correspondantes $(\mathbf{X}_{\text{train}}, \boldsymbol{y})$ est utilisé pour estimer $\boldsymbol{\beta}$ et obtenir $\hat{\boldsymbol{\beta}}$.
- $\mathcal{D}_{\text{train}} = (\mathbf{X}_{\text{train}}, \boldsymbol{y}) = (\mathbf{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ est appelé ensemble d'ajustement ou encore **ensemble d'entraînement**.
- $\Rightarrow$ Nous avons donc $\hat{\boldsymbol{\beta}}_{\text{MCO}} = \hat{\boldsymbol{\beta}}_{\text{MV}} = g_{\text{MCO}}(\mathcal{D}_{\text{train}}) = g_{\text{MV}}(\mathcal{D}_{\text{train}})$.

Théorie de la Décision et Compromis Biais-Variance de $\hat{f}$

- L'objectif est d'évaluer comment l'estimateur $\hat{f}$ se comporte lorsque ses paramètres sont ajustés sur différents jeux de données $\mathcal{D}$.
- Pour une nouvelle entrée $\boldsymbol{x}$, nous étudions la distribution des prédictions $\hat{f}(\boldsymbol{x})$ lorsque le modèle est entraîné avec différents échantillons de données.
- Puisque $\hat{f}$ dépend du jeu de données d'entraînement $\mathcal{D}$, nous analysons sa performance moyenne en étudiant ses prédictions lorsqu'il est ajusté sur différents ensembles $\mathcal{D}$.
- Nous cherchons à comprendre si $\hat{f}$ est biaisé et/ou si ses prédictions varient trop en fonction des données d'entraînement.
- Pour une nouvelle observation $\boldsymbol{x}$, nous nous intéressons alors à la moyenne et à la dispersion des prédictions $\hat{f}(\boldsymbol{x})$ obtenues sur plusieurs jeux de données.

Théorie de la Décision et Compromis Biais-Variance de $\hat{f}$

- On considère le modèle de régression suivant :

$$Y = f(\mathbf{X}) + \epsilon, \quad \text{où} \quad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n).$$

- On considère les modèles de régression linéaires et tout autre modèle de prédiction.
  - $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$: Matrice de conception contenant les n observations.
  - $\boldsymbol{x} \in \mathbb{R}^{p+1}$: Un vecteur représentant une observation donnée.
  - $f(\mathbf{X})$: La vraie relation inconnue entre $\mathbf{X}$ et Y.
  - $\hat{f}(\boldsymbol{x})$: L'estimateur de $f(\boldsymbol{x})$ obtenu à partir d'un ensemble d'entraînement.
- Objectif : Étudier la performance de $\hat{f}(\boldsymbol{x})$ en variant l'ensemble d'entraînement $\mathcal{D}$ utilisé pour estimer f.

Rappel : Propriétés des Estimateurs

- **Biais :** Le biais d'un estimateur est la différence entre l'espérance de l'estimateur et le vrai paramètre qu'il est censé estimer.
- La fonction de prédiction
  - Biais$(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta$
  - Un estimateur est **non biaisé** si son biais est nul: Biais$(\hat{\theta}) = 0 \Rightarrow \mathbb{E}[\hat{\theta}] = \theta$, pour toutes les tailles d'échantillon.
- **Variance :** La variance d'un estimateur mesure la dispersion de l'estimation d'un échantillon à l'autre.
  - Var$(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]$
  - Un estimateur avec une faible variance a tendance à donner des résultats plus précis.
- **Dans le contexte de la Régression Linéaire :**
  - Nous cherchons à approximer une fonction inconnue f par une fonction estimée $\hat{f}$, obtenue à partir d'un jeu de données d'entraînement $\mathcal{D}_{\text{train}}$.

Théorie de la Décision et Compromis Biais-Variance de $\hat{f}$

- Formellement, le biais en un point donné $\boldsymbol{x} \in \mathbb{R}^{(p+1)}$ est donné par :

$$\text{Biais}(\hat{f}(\boldsymbol{x})) = \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})] - f(\boldsymbol{x}) = \mathbb{E}_{\mathcal{D}}[\hat{y}] - y.$$

- Un estimateur est **non biaisé** si $\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})] = f(\boldsymbol{x})$ **pour tout** $\boldsymbol{x}$. C'est à dire, la moyenne des prédictions sur des échantillons de données différents $\mathbb{E}_{\mathcal{D}}[y]$ est égale à la vraie observation y.
- **Variance :** La variance d'un estimateur $\hat{f}$ mesure la sensibilité de l'estimation aux variations des échantillons de données d'entraînement.
  - Formellement, la variance en un point $\boldsymbol{x}$ est définie par :

$$\text{Var}(\hat{f}(\boldsymbol{x})) = \mathbb{E}_{\mathcal{D}}\left[\left(\hat{f}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})]\right)^2\right] = \mathbb{E}_{\mathcal{D}}\left[(\hat{y} - \mathbb{E}_{\mathcal{D}}[\hat{y}])^2\right] = \text{Var}(\hat{y}).$$

  - Un estimateur avec une variance élevée produit **des prédictions très différentes pour des échantillons de données différents.**

Théorie de la Décision et Compromis Biais-Variance de $\hat{f}$

- On peut démontrer que **l'Erreur Quadratique Moyenne (EQM)** (En anglais *Mean Squared Error (MSE)*) en un point donné $\boldsymbol{x}$ peut être décomposée comme suit : **(voir Annexe pour la preuve détaillée)**

$$\mathbb{E}_{\mathcal{D}}[(y-\hat{f}(\boldsymbol{x}))^2] = \underbrace{(f(\boldsymbol{x})-\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])^2}_{\text{Biais}^2} + \underbrace{\mathbb{E}_{\mathcal{D}}[(\hat{f}(\boldsymbol{x})-\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])^2]}_{\text{Variance}} + \sigma^2$$
$$= (y-\mathbb{E}_{\mathcal{D}}[\hat{y}])^2 + \mathbb{E}_{\mathcal{D}}[\hat{y}-\mathbb{E}_{\mathcal{D}}[\hat{y}]] + \sigma^2$$
$$= \mathbb{E}_{\mathcal{D}}[(y-\hat{y})^2], \quad \text{où } \sigma^2 \text{ la variance de } \epsilon \text{ est irréductible.}$$

- **Interprétation : 1. Modèle à Biais Élevé (Sous-ajustement)** :
  - Un modèle trop simple ne capture pas la complexité des données.
  - **Exemple :** une régression linéaire sur un problème intrinsèquement non linéaire. $\Rightarrow$ L'estimateur $\hat{f}$ est sous-ajusté aux données d'entraînement (*problème de sous-apprentissage, en anglais underfitting*).

Théorie de la Décision et Compromis Biais-Variance de $\hat{f}$

- **Interprétation : 2. Modèle Bien Ajusté** :
  - Un bon compromis entre biais et variance.
  - **Exemple :** une régression polynomiale de degré adapté au problème.
  - $\Rightarrow$ L'estimateur $\hat{f}$ capture bien la structure des données et généralise bien.
- **Interprétation : 3. Modèle à Variance Élevée (Sur-ajustement)** :
  - Un modèle trop complexe apprend trop bien les données d'entraînement, y compris le bruit.
  - **Exemple :** une régression polynomiale de degré trop élevé.
  - $\Rightarrow$ L'estimateur $\hat{f}$ varie fortement en fonction des données (*problème de sur-apprentissage, en anglais overfitting*).

Résumé : Interprétation du Compromis Biais-Variance

- Le compromis biais-variance peut être illustré à travers 3 types de modèles :
  - **Modèle à Biais Élevé (Sous-ajustement)** :
    - Exemple : régression linéaire sur un problème non linéaire.
    - Biais élevé : le modèle est trop simple pour capturer $f(X)$.
    - Variance faible : les prédictions sont similaires quelle que soit la base d'entraînement.
  - **Modèle Bien Ajusté** :
    - Exemple : régression polynomiale d'un bon degré.
    - Biais faible : le modèle capture bien la structure sous-jacente des données.
    - Variance modérée : le modèle ne s'adapte pas excessivement aux fluctuations des données.
  - **Modèle à Variance Élevée (Sur-ajustement)** :
    - Exemple : polynôme de degré trop élevé.
    - Biais faible : le modèle suit parfaitement les données d'entraînement.
    - Variance élevée : une petite variation dans les données change significativement les prédictions.

Exercice : Illustration du Compromis Biais-Variance

Exercice : Illustration du Compromis Biais-Variance

- **Objectif :**
  - Dans cet exercice, vous allez explorer le compromis biais-variance à travers la régression polynomiale appliquée à un ensemble de données simulé.
  - Vous devrez analyser le comportement d'un modèle sous-ajusté, bien ajusté et sur-ajusté.
- **Instructions :**
  - **Comprendre les données :** Vous disposez d'un jeu de données généré artificiellement qui suit une relation non linéaire avec du bruit.
  - **Créer et entraîner des modèles :** Vous devrez ajuster des modèles polynomiaux de différents degrés et observer leur performance.
  - **Analyser les erreurs d'entraînement et de test :** Vous étudierez l'évolution de l'erreur quadratique moyenne (EQM) en fonction de la complexité du modèle.
  - **Visualiser l'effet du biais et de la variance :** Vous observerez comment le degré du polynôme influence le compromis entre biais et variance.

Compromis Biais-Variance : Code à Remplir

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split
import seaborn as sns
# Configuration du style des figures
sns.set_theme()

# 1. Génération des données
np.random.seed(8302)
X = np.linspace(-5, 5, 100).reshape(-1, 1)
y = np.sin(X).ravel() + np.cos(X).ravel() ** 2 + np.sin(X).ravel() ** 3 + \
    np.random.normal(0, 0.6, X.shape[0])  # Fonction sinusoïdale avec bruit

# Séparation en jeu d'entraînement et test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Visualisation des données
plt.scatter(X_train, y_train, label="Train Data", alpha=0.6)
plt.scatter(X_test, y_test, label="Test Data", alpha=0.6)
plt.legend()
plt.xlabel("X")
plt.ylabel("y")
plt.title("Données simulées avec bruit")
plt.show()
```

Compromis Biais-Variance : Code à Remplir

```
# 2. Définition de la fonction d'affichage du compromis biais-variance
def plot_bias_variance_tradeoff(degrees):
    """
    Fonction qui entraîne des modèles polynomiaux de différents degrés
    et affiche leurs performances.
    """
    plt.figure(figsize=(12, 5))
    train_errors, test_errors = [], []
    for d in degrees:
        model = make_pipeline(PolynomialFeatures(d), LinearRegression())
        # Compléter l'entraînement du modèle
        # À COMPLETER: Entraînez le modèle sur les données d'entraînement
        model.fit(______, ______)
        # Prédictions sur l'ensemble d'entraînement et de test
        y_train_pred = model.predict(X_train)
        y_test_pred = model.predict(X_test)
        # Calcul de l'erreur quadratique moyenne
        # À COMPLETER: Calculer l'erreur quadratique moyenne
        # pour les ensembles d'entraînement et de test
        train_error = np.mean((_____ - ______) ** 2)
        test_error = np.mean((_____ - ______) ** 2)
        # Stocker les erreurs
        train_errors.append(train_error)
        test_errors.append(test_error)
```

Compromis Biais-Variance : Code à Remplir

```
        # Visualisation des prédictions du modèle
        plt.scatter(X_train, y_train, label="Train Data", alpha=0.3)
        plt.scatter(X_test, y_test, label="Test Data", alpha=0.3)
        # Générer une grille de valeurs pour les prédictions
        X_plot = np.linspace(-5, 5, 100).reshape(-1, 1)
        y_plot = model.predict(X_plot)
        # À COMPLETER: Ajouter la courbe des prédictions du modèle pour le degré d
        plt.plot(______, ______, label=f"Degree {d}")
        plt.xlabel("X")
        plt.ylabel("y")
        plt.legend()
        plt.title(f"Fit du Modèle pour un polynôme de degré {d}")
        plt.show()
    # Affichage des courbes d'erreur
    plt.figure(figsize=(10, 5))
    plt.plot(degrees, train_errors, label="Training Error", marker='o', linestyle='--')
    plt.plot(degrees, test_errors, label="Test Error", marker='o', linestyle='-')
    plt.xlabel("Complexité du Modèle (Degré du polynôme)")
    plt.ylabel("Erreur Quadratique Moyenne (MSE)")
    plt.title("Compromis Biais-Variance")
    plt.legend()
    plt.show()
# 3. Expérimentation : Tester des modèles avec différents degrés de complexité
# À COMPLETER: Choisir une liste de degrés pertinents pour illustrer les trois cas
# sous-ajustement, bon ajustement, et sur-ajustement :
plot_bias_variance_tradeoff([____, ____, ____, ____, ____, ____])
```

Solution de l'Exercice : Illustration du Compromis Biais-Variance

Compromis Biais-Variance : Solution

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split
import seaborn as sns
# Configuration du style des figures
sns.set_theme()

# 1. Génération des données
np.random.seed(8302)
X = np.linspace(-5, 5, 100).reshape(-1, 1)
y = np.sin(X).ravel() + np.cos(X).ravel() ** 2 + np.sin(X).ravel() ** 3 + \
    np.random.normal(0, 0.6, X.shape[0])  # Fonction sinusoïdale avec bruit

# Séparation en jeu d'entraînement et test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Visualisation des données
plt.scatter(X_train, y_train, label="Train Data", alpha=0.6)
plt.scatter(X_test, y_test, label="Test Data", alpha=0.6)
plt.legend()
plt.xlabel("X")
plt.ylabel("y")
plt.title("Données simulées avec bruit")
plt.show()
```

Compromis Biais-Variance : Solution

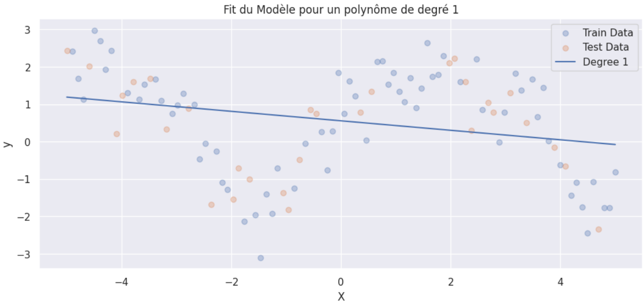
```python
# 2. Définition de la fonction d'affichage du compromis biais-variance
def plot_bias_variance_tradeoff(degrees):
    """
    Fonction qui entraîne des modèles polynomiaux de différents degrés
    et affiche leurs performances.
    """
    plt.figure(figsize=(12, 5))
    train_errors, test_errors = [], []
    for d in degrees:
        model = make_pipeline(PolynomialFeatures(d), LinearRegression())
        # Compléter l'entraînement du modèle
        # À COMPLETER: Entraînez le modèle sur les données d'entraînement
        model.fit(X_train, y_train)
        # Prédictions sur l'ensemble d'entraînement et de test
        y_train_pred = model.predict(X_train)
        y_test_pred = model.predict(X_test)
        # Calcul de l'erreur quadratique moyenne
        # À COMPLETER: Calculer l'erreur quadratique moyenne
        # pour les ensembles d'entraînement et de test
        train_error = np.mean((y_train - y_train_pred) ** 2)
        test_error = np.mean((y_test - y_test_pred) ** 2)
        # Stocker les erreurs
        train_errors.append(train_error)
        test_errors.append(test_error)
```

Compromis Biais-Variance : Code à Remplir

```
        # Visualisation des prédictions du modèle
        plt.scatter(X_train, y_train, label="Train Data", alpha=0.3)
        plt.scatter(X_test, y_test, label="Test Data", alpha=0.3)
        # Générer une grille de valeurs pour les prédictions
        X_plot = np.linspace(-5, 5, 100).reshape(-1, 1)
        y_plot = model.predict(X_plot)
        # À COMPLETER: Ajouter la courbe des prédictions du modèle pour le degré d
        plt.plot(X_plot, y_plot, label=f"Degree {d}")
        plt.xlabel("X")
        plt.ylabel("y")
        plt.legend()
        plt.title(f"Fit du Modèle pour un polynôme de degré {d}")
        plt.show()
    # Affichage des courbes d'erreur
    plt.figure(figsize=(10, 5))
    plt.plot(degrees, train_errors, label="Training Error", marker='o', linestyle='--')
    plt.plot(degrees, test_errors, label="Test Error", marker='o', linestyle='-')
    plt.xlabel("Complexité du Modèle (Degré du polynôme)")
    plt.ylabel("Erreur Quadratique Moyenne (MSE)")
    plt.title("Compromis Biais-Variance")
    plt.legend()
    plt.show()
# 3. Expérimentation : Tester des modèles avec différents degrés de complexité
# À COMPLETER: Choisir une liste de degrés pertinents pour illustrer les trois cas
# sous-ajustement, bon ajustement, et sur-ajustement :
plot_bias_variance_tradeoff([1, 2, 3, 4, 5, 6, 7, 8, 15, 20, 25, 30, 35, 40])
```

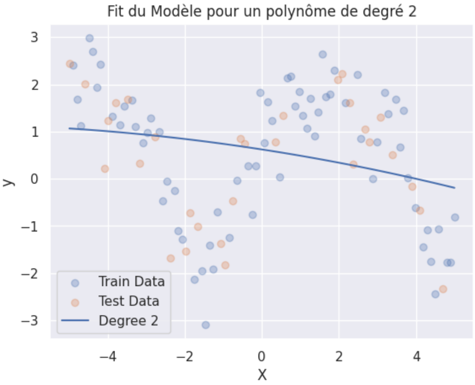
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Biais Élevé (Sous-ajustement)** :
  - Exemple : régression linéaire sur un problème non linéaire.
  - Biais élevé : le modèle est trop simple pour capturer $f(X)$.
  - Variance faible : les prédictions sont similaires quelle que soit la base d'entraînement.



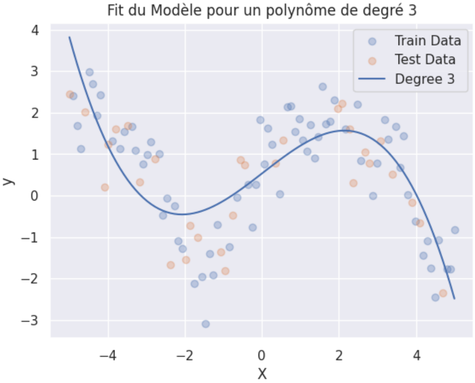
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Biais Élevé (Sous-ajustement)** :
  - Exemple : régression linéaire sur un problème non linéaire.
  - Biais élevé : le modèle est trop simple pour capturer $f(X)$.
  - Variance faible : les prédictions sont similaires quelle que soit la base d'entraînement.



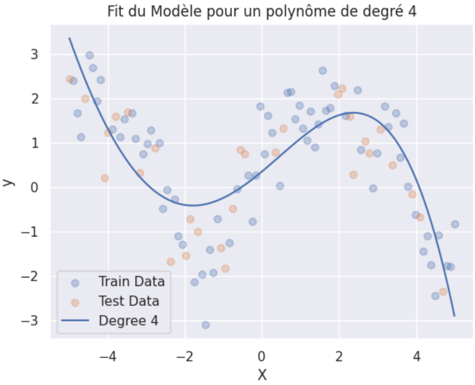
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Exemple : régression polynomiale d'un bon degré.
  - Biais faible : le modèle capture bien la structure sous-jacente des données.
  - Variance modérée : le modèle ne s'adapte pas excessivement aux fluctuations des données.



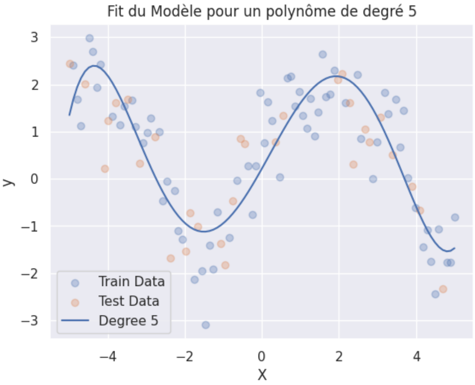
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Exemple : régression polynomiale d'un bon degré.
  - Biais faible : le modèle capture bien la structure sous-jacente des données.
  - Variance modérée : le modèle ne s'adapte pas excessivement aux fluctuations des données.



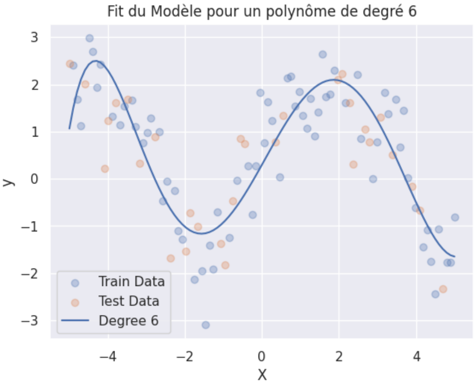
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Exemple : régression polynomiale d'un bon degré.
  - Biais faible : le modèle capture bien la structure sous-jacente des données.
  - Variance modérée : le modèle ne s'adapte pas excessivement aux fluctuations des données.



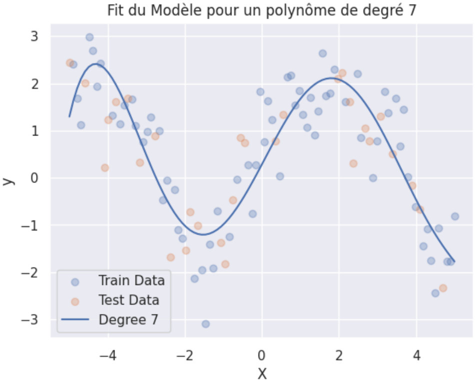
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Exemple : régression polynomiale d'un bon degré.
  - Biais faible : le modèle capture bien la structure sous-jacente des données.
  - Variance modérée : le modèle ne s'adapte pas excessivement aux fluctuations des données.



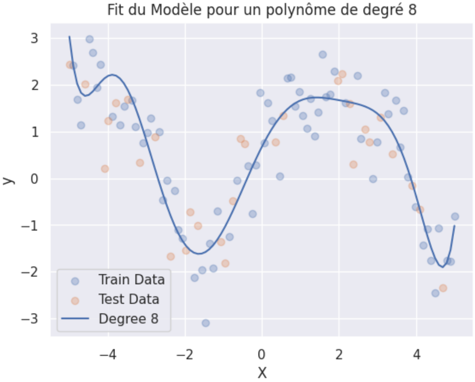
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Exemple : régression polynomiale d'un bon degré.
  - Biais faible : le modèle capture bien la structure sous-jacente des données.
  - Variance modérée : le modèle ne s'adapte pas excessivement aux fluctuations des données.



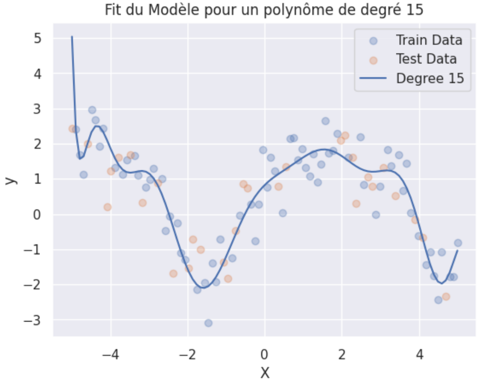
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : le modèle suit parfaitement les données d'entraînement.
  - Variance élevée : une petite variation dans les données change significativement les prédictions.



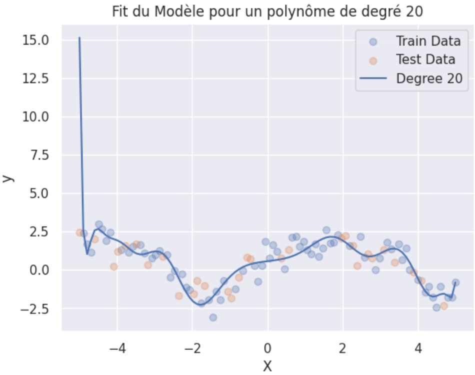
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : le modèle suit parfaitement les données d'entraînement.
  - Variance élevée : une petite variation dans les données change significativement les prédictions.



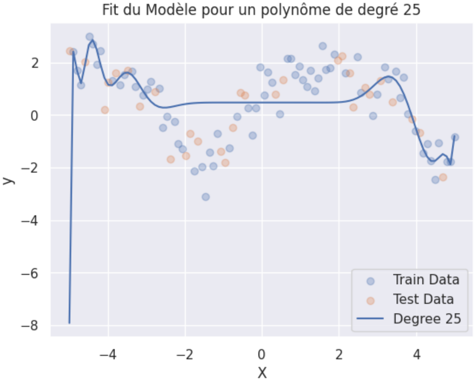
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : le modèle suit parfaitement les données d'entraînement.
  - Variance élevée : une petite variation dans les données change significativement les prédictions.



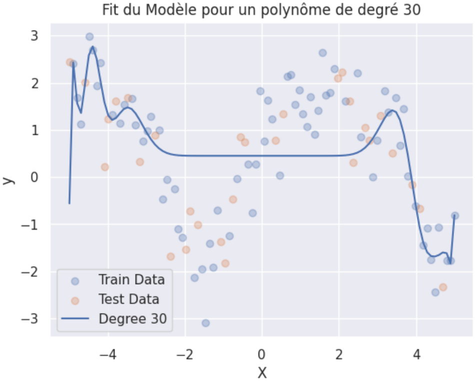
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : le modèle suit parfaitement les données d'entraînement.
  - Variance élevée : une petite variation dans les données change significativement les prédictions.



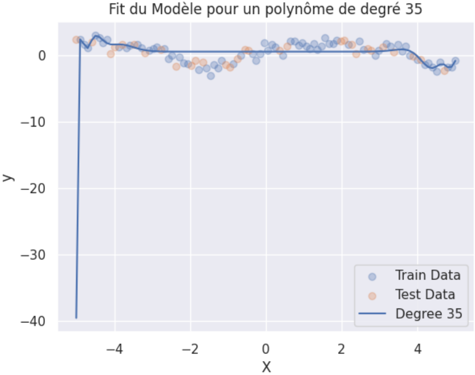
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : le modèle suit parfaitement les données d'entraînement.
  - Variance élevée : une petite variation dans les données change significativement les prédictions.



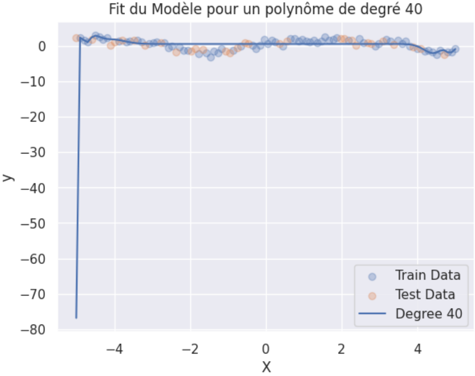
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : le modèle suit parfaitement les données d'entraînement.
  - Variance élevée : une petite variation dans les données change significativement les prédictions.



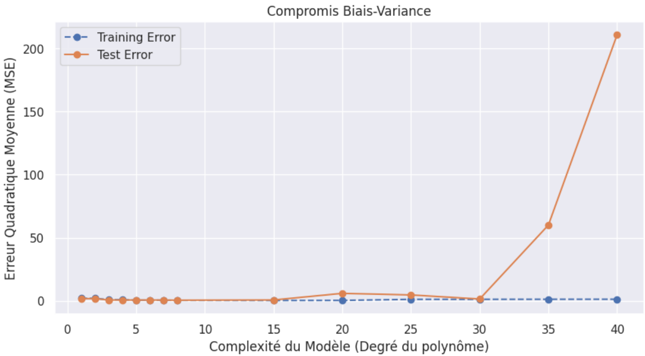
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : le modèle suit parfaitement les données d'entraînement.
  - Variance élevée : une petite variation dans les données change significativement les prédictions.



Solution : Analyse du Compromis Biais-Variance

- On observe une décroissance de l'erreur d'entraînement à mesure que le degré du polynôme augmente.
- Mais l'erreur de test diminue jusqu'à un certain seuil (degré 8), puis augmente rapidement, ce qui reflète le sur-ajustement.
- À partir de degré 25-40, la courbe devient chaotique : le modèle est trop complexe, ce qui conduit à une explosion de l'erreur de test.



Solution : Analyse du Compromis Biais-Variance

- **Figures concernées :** Polynômes de degré 1 et 2.
- **Caractéristiques :**
  - La courbe d'ajustement est quasiment linéaire et ne suit pas du tout la structure non linéaire des données.
  - **Biais élevé** : Le modèle est trop simpliste et ne capture pas les tendances sous-jacentes des données.
  - **Variance faible** : Les prédictions sont similaires pour différents jeux de données d'entraînement.
  - **Performance** : L'erreur quadratique moyenne (EQM) est élevée aussi bien sur les données d'entraînement que de test.
- **Interprétation :**
  - Le modèle est trop rigide et ne permet pas d'exploiter correctement la structure de la relation $y = f(X)$.
  - Cela correspond à un **modèle sous-ajusté (underfitting)**.

Solution : Analyse du Compromis Biais-Variance

- **Figures concernées :** Polynômes de degré 3 à 7.
- **Caractéristiques :**
  - Le modèle capture bien la structure des données sans suivre excessivement le bruit.
  - **Biais faible** : La courbe s'aligne bien sur la tendance réelle des données.
  - **Variance modérée** : Une légère fluctuation dans les données d'entraînement ne modifie pas fortement les prédictions.
  - **Performance** : L'erreur quadratique moyenne est raisonnablement basse et équilibrée entre l'entraînement et le test.
- **Interprétation :**
  - Ce modèle représente un bon **compromis entre biais et variance**.
  - Il est capable de généraliser correctement à de nouvelles données.
  - C'est le type de modèle que l'on recherche en pratique.

Solution : Analyse du Compromis Biais-Variance

- **Figures concernées :** Polynômes de degré 8 à 40.
- **Caractéristiques :**
  - La courbe suit presque parfaitement les points d'entraînement, avec de fortes oscillations.
  - **Biais faible** : La courbe passe par la plupart des points de l'ensemble d'entraînement.
  - **Variance élevée** : La moindre variation dans les données d'entraînement entraîne de grands changements dans la prédiction.
  - **Performance** : Très faible erreur sur l'entraînement, mais forte erreur sur le test (voir la courbe de l'EQM).
- **Interprétation :**
  - Le modèle mémorise les données d'entraînement au lieu de généraliser.
  - Il est trop sensible aux fluctuations du jeu d'entraînement, ce qui le rend mauvais en généralisation.
  - C'est un **modèle sur-ajusté (overfitting)**.

Lien entre le Compromis Biais-Variance et les Intervalles de Confiance et de Prédiction

Lien avec les Intervalles de Confiance et Prédiction

- Le compromis biais-variance concerne la capacité d'un modèle à généraliser à de nouvelles données.
- Cela se manifeste dans :
  - **Les intervalles de confiance et de prédiction :**
    - L'**intervalle de confiance** (**IC**) pour $E(Y_0)$ reflète l'incertitude sur la moyenne prédite, qui diminue à mesure que le modèle est bien ajusté.
    - L'**intervalle de prédiction** (**IP**) pour une nouvelle observation est toujours plus large que l'IC, car il prend en compte la variance des erreurs.
    - Si le modèle est trop flexible, càd avec une faible erreur d'entraînement mais forte variance, l'IP peut être très large, ce qui est une indication de surajustement.
- **En résumé :**
  - **Modèle trop simple** (biais élevé) : IC et IP **très larges**, car le modèle sous-exploite l'information des données.
  - **Modèle trop complexe** (variance élevée) : IC et IP plus grand car le modèle surexploite les données d'entraînement (la largeur des intervalles augmente où il y a moins de données et plus d'incertitude).

Lien avec les Intervalles de Confiance et Prédiction

- La sélection d'un modèle et la construction des intervalles de confiance et de prédiction sont liées à la théorie de la décision.
- Risque et choix du modèle :
  - Un modèle est sélectionné pour minimiser l'erreur quadratique moyenne (EQM).
  - La largeur de l'intervalle de prédiction est un indicateur clé :
    - Un modèle avec une variance élevée aura des IP plus grands, signifiant une incertitude plus grande sur les nouvelles observations.
- Utilisation des intervalles dans la prise de décision :
  - Si intervalle de prédiction est trop large, le modèle est peu fiable pour la prise de décision.
  - Un intervalle de confiance étroit sur $\mathbb{E}(Y_0)$ est un bon signe : cela signifie que l'estimation du modèle est précise.

Rappel : Intervalles de Confiance et de Prédiction

- **Somme des carrés des écarts :**

$$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

- **Estimateur de la variance des résidus :**

$$s^2 = \frac{1}{n-p-1} \sum (y_i - \hat{y}_i)^2, \quad \text{où :}$$

  - n est la taille de l'échantillon,
  - p est le degré du polynôme.
- **Valeurs Critique :** $\pm t_{\alpha/2,\, n-p-1}$ tel que $P(T > t_{\alpha/2,\, n-p-1}) = \frac{\alpha}{2}$ où T suit une loi de Student à $n-p-1$ degrés de liberté.
- **Intervalle de Confiance :** (incertitude sur la moyenne prédite) $IC = \hat{y} \pm t \cdot s \cdot \sqrt{\frac{1}{n} + \frac{(x_0 - \bar{X})^2}{S_{xx}}}$
- **Intervalle de Prédiction :** (incertitude sur une nouvelle observation) $IP = \hat{y} \pm t \cdot s \cdot \sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{X})^2}{S_{xx}}}$.

Exercice : Lien entre le Compromis Biais-Variance et les Intervalles de Confiance et Prédiction

Exercice : Lien avec les IC et IP

- **Objectif :**
  - Explorer le compromis biais-variance à travers la régression polynomiale appliquée à un ensemble de données simulé.
  - Comprendre et interpréter les intervalles de confiance et de prédiction.
  - Analyser le comportement d'un modèle sous-ajusté, bien ajusté et sur-ajusté.
- **Instructions :**
  - **Comprendre les données :** Un jeu de données généré artificiellement suivant une relation non linéaire avec du bruit.
  - **Créer et entraîner des modèles :** Ajuster des modèles polynomiaux de différents degrés et observer leur performance.
  - **Analyser les erreurs d'entraînement et de test :** Étudier l'évolution de l'erreur quadratique moyenne (EQM) en fonction de la complexité du modèle.
  - **Visualiser les intervalles :** Observer l'effet de la complexité du modèle sur la largeur des intervalles de confiance et de prédiction.

Exercice : Lien avec les IC et IP (Code à Compléter)

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split
# Configuration du style des figures
sns.set_theme()

# 1. Génération des données
np.random.seed(8302)
X = np.linspace(-5, 5, 100).reshape(-1, 1)
y = np.sin(X).ravel() + np.cos(X).ravel() ** 2 + np.sin(X).ravel() ** 3 + \
    np.random.normal(0, 0.6, X.shape[0])  # Fonction sinusoïdale avec bruit

# Séparation en jeu d'entraînement et test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Visualisation des données
plt.scatter(X_train, y_train, label="Données d'entraînement", alpha=0.6)
plt.scatter(X_test, y_test, label="Données de test", alpha=0.6)
plt.legend()
plt.xlabel("X")
plt.ylabel("y")
plt.title("Données simulées avec bruit")
plt.show()
```

Exercice : Lien avec les IC et IP (Code à Compléter)

```
from scipy import stats
def plot_model_with_intervals(degree):
    """
    Fonction qui entraîne un modèle polynomial et affiche les intervalles
    de confiance et de prédiction.
    """
    model = make_pipeline(PolynomialFeatures(degree), LinearRegression())
    model.fit(X_train, y_train)

    X_plot = np.linspace(-5, 5, 100).reshape(-1, 1)
    y_pred = model.predict(X_plot)

    # Transformation polynomiale
    poly = PolynomialFeatures(degree)
    X_train_poly = poly.fit_transform(X_train)
    X_plot_poly = poly.transform(X_plot)

    #  À remplir: Calculer le facteur critique de Student
    n = ______
    p = ______.shape[1]
    dof = ______  # À compléter
    t_critical = stats.t.ppf(0.975, df=______)  # À compléter

    # À remplir: Calculer les résidus et l'erreur standard
    residuals = y_train - model.predict(X_train)
    mse = np.sum(residuals**2) / (________)  # À compléter : degrés de liberté
    sigma_hat = np.sqrt(mse)
```

Exercice : Lien avec les IC et IP (Code à Compléter)

```
X_mean = np.mean(X_train_poly, axis=0)
S_xx = np.sum((________ - ________) ** 2, axis=0) # À remplir: Calculer S_xx
# IMPORTANT : Ajout d'un epsilon pour éviter la division par zéro
S_xx[S_xx == 0] += 1e-10
# À remplir : Calculer les intervalles de confiance et de prédiction
SE_confidence = sigma_hat * np.sqrt(
    1/n + np.sum((X_plot_poly - X_mean) ** 2 / S_xx, axis=1))
SE_prediction = sigma_hat * np.sqrt(
    1 + 1/n + np.sum((X_plot_poly - X_mean) ** 2 / S_xx, axis=1))
ci_upper = y_pred + _______ * _______
ci_lower = y_pred - _______ * _______
pi_upper = y_pred + _______ * _______
pi_lower = y_pred - _______ * _______
# Visualisation
plt.figure(figsize=(8, 5))
plt.scatter(X_train, y_train, label="Données d'entraînement", alpha=0.3)
plt.scatter(X_test, y_test, label="Données de test", alpha=0.3)
plt.plot(X_plot, y_pred, label=f"Modèle degré {degree}", color='red')
plt.fill_between(X_plot.ravel(), ci_lower, ci_upper, color='blue',
                 alpha=0.2, label='Intervalle de Confiance (95%)')
plt.fill_between(X_plot.ravel(), pi_lower, pi_upper, color='green',
                 alpha=0.2, label='Intervalle de Prédiction (95%)')
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.title(f"Modèle de degré {degree} : Intervalles de Confiance et de Prédiction")
plt.show()
```

Exercice : Lien avec les IC et IP (Code à Compléter)

```
# À remplir Tester pour différents degrés de polynômes
degrees = [_______, _______, _______, _______, _______, ..., _______]
for d in degrees:
    plot_model_with_intervals(d)
# Comparaison des erreurs
train_errors, test_errors = [], []

for d in degrees:
    model = make_pipeline(PolynomialFeatures(d), LinearRegression())
    model.fit(X_train, y_train)

    train_error = np.mean((y_train - model.predict(X_train)) ** 2)
    test_error = np.mean((y_test - model.predict(X_test)) ** 2)

    train_errors.append(train_error)
    test_errors.append(test_error)

# Visualisation des erreurs
plt.figure(figsize=(8, 5))
plt.plot(degrees, train_errors, marker='o', linestyle='--', label="Erreur d'entraînement")
plt.plot(degrees, test_errors, marker='o', linestyle='-', label="Erreur de test")
plt.xlabel("Degré du polynôme")
plt.ylabel("Erreur Quadratique Moyenne (MSE)")
plt.title("Compromis Biais-Variance et Erreur de Test")
plt.legend()
plt.show()
```

Questions d'Analyse

- Quel est l'effet de l'augmentation du degré du polynôme sur l'erreur d'entraînement et l'erreur de test ?
- Comment la largeur des intervalles de confiance et de prédiction évolue-t-elle avec la complexité du modèle ?
- Pourquoi un modèle sur-ajusté a-t-il des intervalles de confiance plus serrés mais des intervalles de prédiction plus larges ?
- En quoi cette analyse peut-elle aider à choisir un modèle optimal en pratique ?

Solution de l'Exercice : Lien entre le Compromis Biais-Variance et les Intervalles de Confiance et Prédiction

Exercice : Lien avec les IC et IP (Solution)

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split

# Configuration du style des figures
sns.set_theme()

# 1. Génération des données
np.random.seed(8302)
X = np.linspace(-5, 5, 100).reshape(-1, 1)
y = np.sin(X).ravel() + np.cos(X).ravel() ** 2 + np.sin(X).ravel() ** 3 + \
    np.random.normal(0, 0.6, X.shape[0])  # Fonction sinusoïdale avec bruit

# Séparation en jeu d'entraînement et test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Visualisation des données
plt.scatter(X_train, y_train, label="Données d'entraînement", alpha=0.6)
plt.scatter(X_test, y_test, label="Données de test", alpha=0.6)
plt.legend()
plt.xlabel("X")
plt.ylabel("y")
plt.title("Données simulées avec bruit")
plt.show()
```

Exercice : Lien avec les IC et IP (Solution)

```python
from scipy import stats

def plot_model_with_intervals(degree):
    """
    Fonction qui entraîne un modèle polynomial et affiche les intervalles
    de confiance et de prédiction.
    """
    model = make_pipeline(PolynomialFeatures(degree), LinearRegression())
    model.fit(X_train, y_train)

    X_plot = np.linspace(-5, 5, 100).reshape(-1, 1)
    y_pred = model.predict(X_plot)

    # Transformation polynomiale
    poly = PolynomialFeatures(degree)
    X_train_poly = poly.fit_transform(X_train)
    X_plot_poly = poly.transform(X_plot)

    # Calcul du facteur critique de Student
    n = len(y_train)
    p = X_train_poly.shape[1]
    dof = n - p
    t_critical = stats.t.ppf(0.975, df=dof)

    # Calcul des résidus et de l'erreur standard
    residuals = y_train - model.predict(X_train)
    mse = np.sum(residuals**2) / dof
    sigma_hat = np.sqrt(mse)
```

Exercice : Lien avec les IC et IP (Solution)

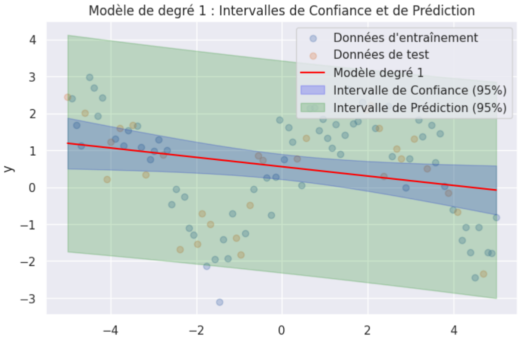
```
# Calcul de S_xx
X_mean = np.mean(X_train_poly, axis=0)
S_xx = np.sum((X_train_poly - X_mean) ** 2, axis=0)
# IMPORTANT : Ajout d'un epsilon pour éviter la division par zéro
S_xx[S_xx == 0] += 1e-10
# Calcul des intervalles de confiance et de prédiction
SE_confidence = sigma_hat * np.sqrt(
    1/n + np.sum((X_plot_poly - X_mean) ** 2 / S_xx, axis=1))
SE_prediction = sigma_hat * np.sqrt(
    1 + 1/n + np.sum((X_plot_poly - X_mean) ** 2 / S_xx, axis=1))
ci_upper = y_pred + t_critical * SE_confidence
ci_lower = y_pred - t_critical * SE_confidence
pi_upper = y_pred + t_critical * SE_prediction
pi_lower = y_pred - t_critical * SE_prediction
# Visualisation
plt.figure(figsize=(8, 5))
plt.scatter(X_train, y_train, label="Données d'entraînement", alpha=0.3)
plt.scatter(X_test, y_test, label="Données de test", alpha=0.3)
plt.plot(X_plot, y_pred, label=f"Modèle degré {degree}", color='red')
plt.fill_between(X_plot.ravel(), ci_lower, ci_upper, color='blue',
                 alpha=0.2, label='Intervalle de Confiance (95%)')
plt.fill_between(X_plot.ravel(), pi_lower, pi_upper, color='green',
                 alpha=0.2, label='Intervalle de Prédiction (95%)')
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.title(f"Modèle de degré {degree} : Intervalles de Confiance et de Prédiction")
plt.show()
```

Exercice : Lien avec les IC et IP (Solution)

```
# Tester pour différents degrés de polynômes
degrees = [1, 2, 3, 4, 5, 6, 7, 8, 15, 20, 25, 30, 35, 40]
for d in degrees:
    plot_model_with_intervals(d)

# Comparaison des erreurs
train_errors, test_errors = [], []

for d in degrees:
    model = make_pipeline(PolynomialFeatures(d), LinearRegression())
    model.fit(X_train, y_train)

    train_error = np.mean((y_train - model.predict(X_train)) ** 2)
    test_error = np.mean((y_test - model.predict(X_test)) ** 2)

    train_errors.append(train_error)
    test_errors.append(test_error)

# Visualisation des erreurs
plt.figure(figsize=(8, 5))
plt.plot(degrees, train_errors, marker='o', linestyle='--', label="Erreur d'entraînement")
plt.plot(degrees, test_errors, marker='o', linestyle='-', label="Erreur de test")
plt.xlabel("Degré du polynôme")
plt.ylabel("Erreur Quadratique Moyenne (MSE)")
plt.title("Compromis Biais-Variance et Erreur de Test")
plt.legend()
plt.show()
```

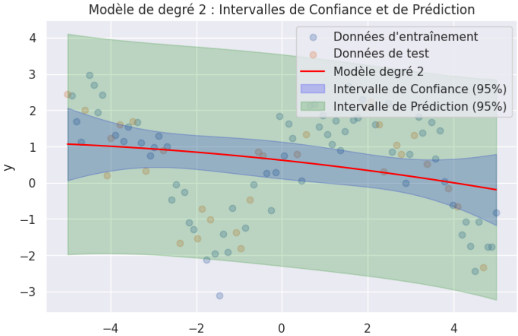
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Biais Élevé (Sous-ajustement)** :
  - Biais élevé : le modèle est trop simple pour capturer $f(X)$.
  - Les intervalles de confiance (bleu) et de prédiction (vert) sont très larges, indiquant une incertitude élevée.
  - Le modèle ne suit pas bien la tendance réelle des données, ce qui est un indicateur de biais élevé.
  - Variance faible : les prédictions sont similaires quelle que soit la base d'entraînement.



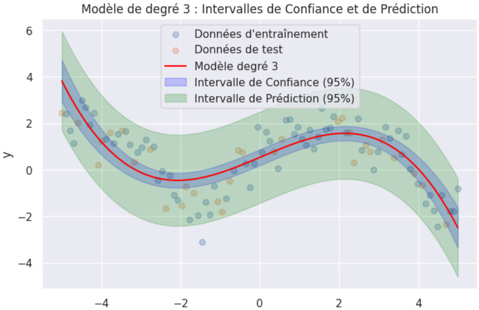
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Biais Élevé (Sous-ajustement)** :
  - Biais élevé : le modèle est trop simple pour capturer $f(X)$.
  - Les intervalles de confiance (bleu) et de prédiction (vert) sont très larges, indiquant une incertitude élevée.
  - Le modèle ne suit pas bien la tendance réelle des données, ce qui est un indicateur de biais élevé.
  - Variance faible : les prédictions sont similaires quelle que soit la base d'entraînement.



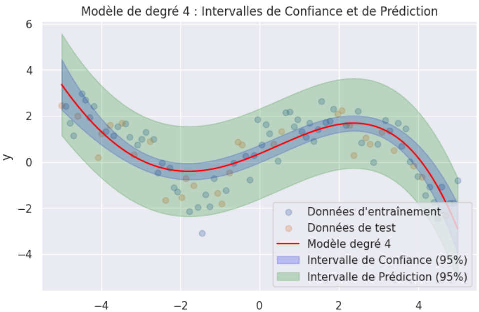
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Régression polynomiale d'un bon degré. Ici, le modèle semble offrir un bon compromis entre biais et variance.
  - Biais faible : La courbe rouge commence à mieux épouser la structure des données. Les intervalles de confiance deviennent plus resserrés, indiquant une meilleure estimation de la moyenne.
  - L'intervalle de prédiction reste large mais raisonnable, ce qui est normal car il prend en compte la variabilité des nouvelles observations.



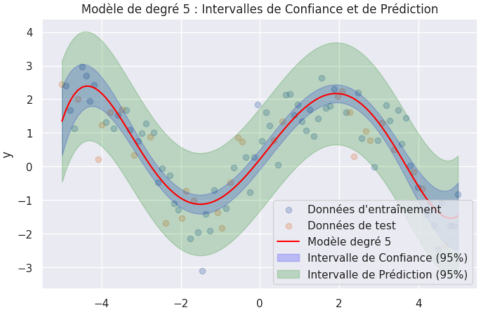
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Régression polynomiale d'un bon degré. Ici, le modèle semble offrir un bon compromis entre biais et variance.
  - Biais faible : La courbe rouge commence à mieux épouser la structure des données. Les intervalles de confiance deviennent plus resserrés, indiquant une meilleure estimation de la moyenne.
  - L'intervalle de prédiction reste large mais raisonnable, ce qui est normal car il prend en compte la variabilité des nouvelles observations.



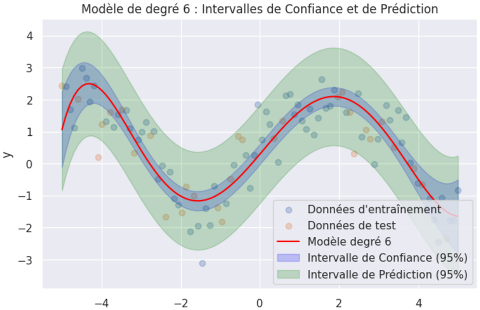
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Régression polynomiale d'un bon degré. Ici, le modèle semble offrir un bon compromis entre biais et variance.
  - Biais faible : La courbe rouge commence à mieux épouser la structure des données. Les intervalles de confiance deviennent plus resserrés, indiquant une meilleure estimation de la moyenne.
  - L'intervalle de prédiction reste large mais raisonnable, ce qui est normal car il prend en compte la variabilité des nouvelles observations.



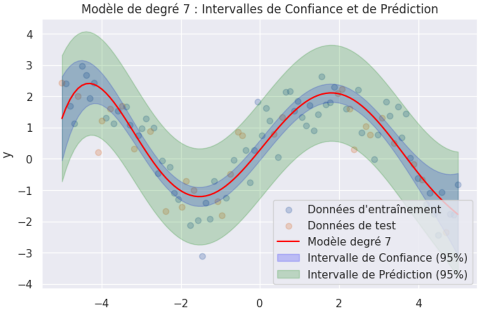
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Régression polynomiale d'un bon degré. Ici, le modèle semble offrir un bon compromis entre biais et variance.
  - Biais faible : La courbe rouge commence à mieux épouser la structure des données. Les intervalles de confiance deviennent plus resserrés, indiquant une meilleure estimation de la moyenne.
  - L'intervalle de prédiction reste large mais raisonnable, ce qui est normal car il prend en compte la variabilité des nouvelles observations.



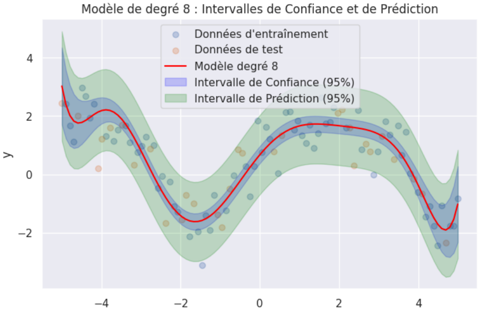
Solution : Interprétation du Compromis Biais-Variance

- **Modèle Bien Ajusté** :
  - Régression polynomiale d'un bon degré. Ici, le modèle semble offrir un bon compromis entre biais et variance.
  - Biais faible : La courbe rouge commence à mieux épouser la structure des données. Les intervalles de confiance deviennent plus resserrés, indiquant une meilleure estimation de la moyenne.
  - L'intervalle de prédiction reste large mais raisonnable, ce qui est normal car il prend en compte la variabilité des nouvelles observations.



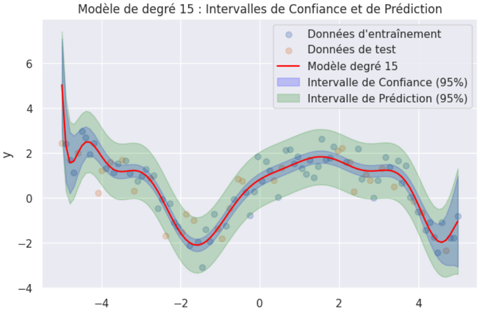
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : La courbe rouge épouse trop fidèlement les points d'entraînement, capturant même le bruit. Les intervalles de confiance sont parfois instables, en particulier aux extrémités.
  - Variance élevée : une petite variation dans les données change significativement les prédictions. Le modèle s'adapte trop aux variations du jeu d'entraînement et risque de mal généraliser sur de nouvelles données.



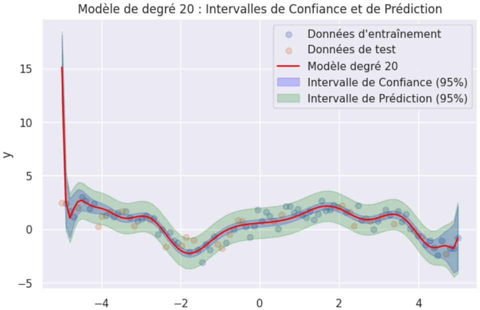
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : La courbe rouge épouse trop fidèlement les points d'entraînement, capturant même le bruit. Les intervalles de confiance sont parfois instables, en particulier aux extrémités.
  - Variance élevée : une petite variation dans les données change significativement les prédictions. Le modèle s'adapte trop aux variations du jeu d'entraînement et risque de mal généraliser sur de nouvelles données.



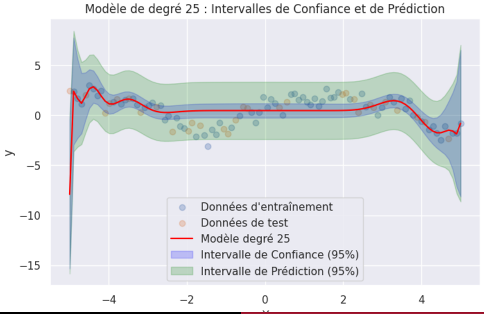
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : La courbe rouge épouse trop fidèlement les points d'entraînement, capturant même le bruit. Les intervalles de confiance sont parfois instables, en particulier aux extrémités.
  - Variance élevée : une petite variation dans les données change significativement les prédictions. Le modèle s'adapte trop aux variations du jeu d'entraînement et risque de mal généraliser sur de nouvelles données.



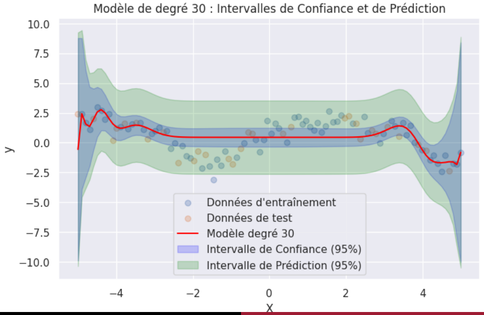
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : La courbe rouge épouse trop fidèlement les points d'entraînement, capturant même le bruit. Les intervalles de confiance sont parfois instables, en particulier aux extrémités.
  - Variance élevée : une petite variation dans les données change significativement les prédictions. Le modèle s'adapte trop aux variations du jeu d'entraînement et risque de mal généraliser sur de nouvelles données.



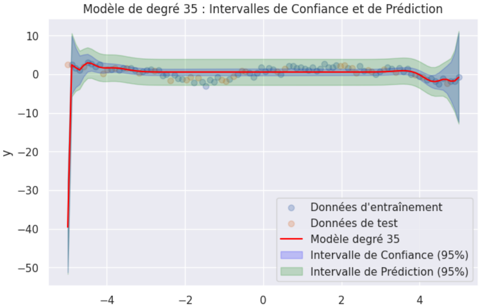
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : La courbe rouge épouse trop fidèlement les points d'entraînement, capturant même le bruit. Les intervalles de confiance sont parfois instables, en particulier aux extrémités.
  - Variance élevée : une petite variation dans les données change significativement les prédictions. Le modèle s'adapte trop aux variations du jeu d'entraînement et risque de mal généraliser sur de nouvelles données.



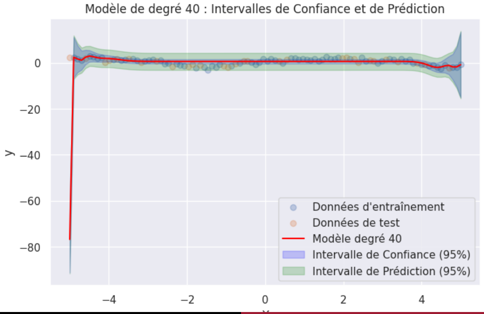
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : La courbe rouge épouse trop fidèlement les points d'entraînement, capturant même le bruit. Les intervalles de confiance sont parfois instables, en particulier aux extrémités.
  - Variance élevée : une petite variation dans les données change significativement les prédictions. Le modèle s'adapte trop aux variations du jeu d'entraînement et risque de mal généraliser sur de nouvelles données.



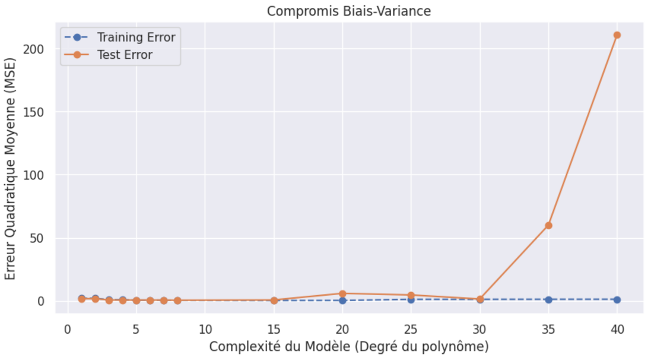
Solution : Interprétation du Compromis Biais-Variance

- **Modèle à Variance Élevée (Sur-ajustement)** :
  - Exemple : polynôme de degré trop élevé.
  - Biais faible : La courbe rouge épouse trop fidèlement les points d'entraînement, capturant même le bruit. Les intervalles de confiance sont parfois instables, en particulier aux extrémités.
  - Variance élevée : une petite variation dans les données change significativement les prédictions. Le modèle s'adapte trop aux variations du jeu d'entraînement et risque de mal généraliser sur de nouvelles données.



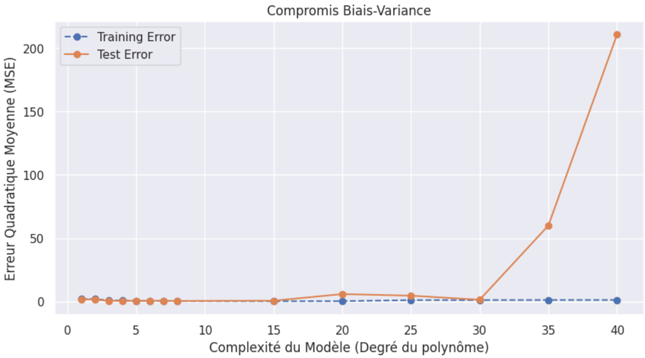
Solution : Analyse du Compromis Biais-Variance

- On observe une décroissance de l'erreur d'entraînement à mesure que le degré du polynôme augmente.
- Mais l'erreur de test diminue jusqu'à un certain seuil (degré 8), puis augmente rapidement, ce qui reflète le sur-ajustement.
- À partir de degré 25-40, la courbe devient chaotique : le modèle est trop complexe, ce qui conduit à une explosion de l'erreur de test.



Solution : Analyse du Compromis Biais-Variance

- Solutions Potentielles aux sur-ajustement :
  - **L'utilisation de validation croisée** permet d'identifier la meilleure complexité de modèle sans sur-ajuster.
  - **Ajouter de la régularisation** (ex : Ridge, Lasso) peut aussi aider à éviter le sur-ajustement.



Lien entre Estimateur Consistant et Erreur Moyenne Quadratique (EMQ)

Consistance et Erreur Moyenne Quadratique (EMQ)

- **Définition de la consistance**
  Un estimateur $\hat{\theta}_n$ d'un paramètre θ est consistant si, lorsque la taille de l'échantillon n tend vers l'infini, il converge en probabilité vers le paramètre réel θ, c'est-à-dire :

$$\hat{\theta}_n \xrightarrow{\mathbb{P}} \theta$$

  Autrement dit, pour tout $\varepsilon > 0$:

$$\lim_{n\to\infty} P(|\hat{\theta}_n - \theta| > \varepsilon) = 0.$$

- **Erreur Moyenne Quadratique (EMQ) pour un estimateur ponctuel**
  L'erreur moyenne quadratique d'un estimateur $\hat{\theta}_n$ est définie par :

$$EMQ(\hat{\theta}_n) = \mathbb{E}[(\hat{\theta}_n - \theta)^2] = \text{Var}(\hat{\theta}_n) + \text{Biais}^2(\hat{\theta}_n),$$

  où :
  - Le biais est $\text{Biais}(\hat{\theta}_n) = \mathbb{E}[\hat{\theta}_n] - \theta$.
  - La variance est $\text{Var}(\hat{\theta}_n) = \mathbb{E}[(\hat{\theta}_n - \mathbb{E}[\hat{\theta}_n])^2]$.

Lien entre Consistance et Erreur Moyenne Quadratique (EMQ)

- **Erreur Moyenne Quadratique (EMQ) pour un modèle de prédiction**
  Dans le cadre d'un modèle de prédiction, l'EMQ inclut un terme supplémentaire lié à l'erreur irréductible σ^2 :

$$\mathbb{E}_{\mathcal{D}}[(y - \hat{f}(\boldsymbol{x}))^2] = \text{Biais}^2 + \text{Variance} + \sigma^2,$$

  où σ^2 représente la variance du bruit aléatoire ϵ.
- Si un estimateur $\hat{\theta}_n$ est consistant, alors :
  - Son biais tend vers 0 : $\lim_{n\to\infty} \mathbb{E}[\hat{\theta}_n] = \theta$.
  - Sa variance tend vers 0 : $\lim_{n\to\infty} \text{Var}(\hat{\theta}_n) = 0$.
  - Donc, l'EMQ tend vers 0 : $\lim_{n\to\infty} EMQ(\hat{\theta}_n) = 0$.
- **Distinction pour les modèles de prédiction**
  Contrairement aux estimateurs ponctuels, dans un modèle de prédiction : $\lim_{n\to\infty} \mathbb{E}_{\mathcal{D}}[(y - \hat{f}(\boldsymbol{x}))^2] = \sigma^2$. Même si $\text{Biais}^2 \to 0$ et Variance $\to 0$, l'erreur irréductible σ^2 persiste (même avec un modèle parfait, il y a une erreur résiduelle due au bruit des données.).

Table des Matières

Décomposition en Biais Variance

Preuve détaillée de la décomposition biais-variance

- On considère un modèle de régression avec une relation entre la variable cible y et l'entrée $\boldsymbol{x}$ définie par :

$$y = f(\boldsymbol{x}) + \epsilon, \quad \text{où} \quad \mathbb{E}[\epsilon] = 0, \quad \mathsf{Var}(\epsilon) = \sigma^2.$$

- Notre objectif est d'étudier l'erreur quadratique moyenne (**EQM**) en un point donné $\boldsymbol{x}$:

$$\mathbb{E}_{\mathcal{D}}[(y - \hat{f}(\boldsymbol{x}))^2].$$

Étape 1 : Développement de l'erreur quadratique moyenne

- On remplace y par son expression en fonction de $f(\boldsymbol{x})$ et du bruit ϵ :

$$\mathbb{E}_{\mathcal{D}}[(y - \hat{f}(\boldsymbol{x}))^2] = \mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) + \epsilon - \hat{f}(\boldsymbol{x}))^2].$$

- En développant le carré :

$$\mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}) + \epsilon)^2].$$

- Décomposons ce carré :

$$\mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}))^2 + 2(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}))\epsilon + \epsilon^2].$$

Étape 2 : Calcul de l'espérance terme par terme

- Par linéarité de l'espérance :

$$\mathbb{E}_{\mathcal{D}}[(y-\hat{f}(\boldsymbol{x}))^2] = \mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x})-\hat{f}(\boldsymbol{x}))^2]+2\mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x})-\hat{f}(\boldsymbol{x}))\epsilon]+\mathbb{E}[\epsilon^2].$$

- Puisque $\mathbb{E}[\epsilon] = 0$ et que ϵ est indépendant de $\hat{f}(\boldsymbol{x})$, on a :

$$\mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}))\epsilon] = \mathbb{E}_{\mathcal{D}}[f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})] \cdot \mathbb{E}[\epsilon] = 0.$$

- Il reste donc :

$$\mathbb{E}_{\mathcal{D}}[(y - \hat{f}(\boldsymbol{x}))^2] = \mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}))^2] + \mathbb{E}[\epsilon^2].$$

- Or, la variance du bruit est $\mathbb{E}[\epsilon^2] = \sigma^2$, donc :

$$\mathbb{E}_{\mathcal{D}}[(y - \hat{f}(\boldsymbol{x}))^2] = \mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}))^2] + \sigma^2.$$

Étape 3 : Décomposition du premier terme (biais et variance)

- On ajoute et soustrait $\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})]$ dans $(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}))$:

$$f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}) = (f(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})]) + (\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})] - \hat{f}(\boldsymbol{x})).$$

- En élevant au carré et en prenant l'espérance :

$$\mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])^2] + 2\mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])(\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})] - \hat{f}(\boldsymbol{x}))] \\ + \mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})] - \hat{f}(\boldsymbol{x}))^2].$$

- Le deuxième terme s'annule car :

$$\begin{aligned} &\mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])(\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})] - \hat{f}(\boldsymbol{x}))] \\ &= (f(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})]) \cdot \mathbb{E}_{\mathcal{D}}[\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})] - \hat{f}(\boldsymbol{x})] \\ &= 0. \end{aligned}$$

- Il reste donc :

$$\mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}))^2] = (f(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])^2 + \mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})] - \hat{f}(\boldsymbol{x}))^2].$$

Conclusion : Décomposition finale

- Ces termes correspondent respectivement :
  - Au biais au carré :

$$\text{Biais}^2 = (f(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])^2.$$

  - À la variance :

$$\text{Variance} = \mathbb{E}_{\mathcal{D}}[(\hat{f}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])^2].$$

- On réinjecte dans l'expression de l'erreur quadratique moyenne :

$$\mathbb{E}_{\mathcal{D}}[(y - \hat{f}(\boldsymbol{x}))^2] = \underbrace{(f(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])^2}_{\text{Biais}^2} + \underbrace{\mathbb{E}_{\mathcal{D}}[(\hat{f}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(\boldsymbol{x})])^2]}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Erreur Irréductible}}.$$