
Université de Liège

---

Introduction to machine learning (ELEN062-1)

# Projet 2 : Bias and variance analysis

---

*Authors :*
Louis HOGGE s192814
Simon LOUVEAU s194100
Tom WEBER s203806

*Professor :* L. Wehenkel
*Professor :* P. Geurts
*Year :* 2023-2024

# 1 Anatical derivation

## 1.1 Analytical expression

Data of the ridge regression problem :

- Learning sample of N pairs, $LS = \{(x_1, y_1), ..., (x_N, y_N)\}$ with $(x_i, y_i) \in \mathbb{R}x\mathbb{R}$
- Values of input variables $x_i$ are fixed while ouput variables are $y_i = wx_i + \epsilon_i$, where $e_i$'s are independent and identically drawn from a normal distribution $\mathcal{N}(0, \sigma^2)$.
- $\hat{w}_{LS} = \arg\min_{w'}\{\sum_{i=1}^{N}(y_i - w'x_i)^2 + \lambda w'^2\}$, with $\lambda \geq 0$ the regularization hyper-parameter
- w' is the weight we want to estimate
- Let us call the cost function F(w') such that $F(w') = \sum_{i=0}^{N}(y_i - w'x_i)^2 + \lambda w'^2$
- $x_0$ will be referring to a specific test point.
- $s_{xx} = \sum_i x_i^2$ and $s_{xy} = \sum_i x_i y_i$

Now, analytical expressions are given with explainations here-under.

1. $\hat{w}_{LS}$
   To find the $\hat{w}_{LS}$ that minimizes the cost function F(w') defined in the preamble, we take the derivative of F(w') with respect to w', set it to zero and solve for w'.
   Taking the derivative :
   $$\frac{dF(w')}{dw'} = \frac{d}{dw'}(\sum_{i=1}^{N}(y_i - w'x_i)^2 + \lambda w'^2)$$
   Expanding the derivative :
   $$= \sum_{i=1}^{N} 2(y_i - w'x_i)(-x_i) + 2\lambda w'$$
   $$= -2\sum_{i=1}^{N} x_i y_i + 2w' \sum_{i=1}^{N} x_i^2 + 2\lambda w'$$
   Setting this derivative equal to zero to find the minimium :
   $$-2\sum_{i=1}^{N} x_i y_i + 2w' \sum_{i=1}^{N} x_i^2 + 2\lambda w' = 0$$
   Simplifying and solving for w' :
   $$w'(\sum_{i=1}^{N} x_i^2 + \lambda) = \sum_{i=1}^{N} x_i y_i$$
   $$w' = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} x_i^2 + \lambda} = \hat{w}_{LS} = \frac{s_{xy}}{s_{xx} + \lambda}$$
2. Bayes model, $h_B(x_0)$
   To find $h_B(x_0)$, the model that predicts the expected value of y given x is to be determined, using the distribution of y conditioned on x. In our case, $y = wx + \epsilon$, where $\epsilon$ is normally distributed as $\mathcal{N}(0, \sigma^2)$. The Bayes model, $h_B(x)$, is the model that minimizes the expected prediction error for a new input x, which in the case of regression is typically the expected value of y given x. Given that $\epsilon$ is normally distributed with mean 0, the expected value of y given x simplifies to the deterministic part of the relationship as $E\{\epsilon\} = 0$ as $\epsilon$ is from a $\mathcal{N}(0, \sigma^2)$.
   Therefore, the Bayes model, $h_B(x_0)$ for a new input $x_0$ in this case is $h_B(x_0) = E\{y|x = x_0\} = wx_0$.
   $h_B(x_0)$ represents the theoretical best prediction under the given model of data.

3. Residual error, $noise(x_0)$
   The residual error in a regression context typically refers to the difference between the observed values and the predicted values. As seen from the theoretical notes, $noise(x_0) = E_{y|x_0}\{(y - h_B(x_0))^2\}$ quantifies how much y varies from $h_B(x_0) = E_{y|x_0}\{y\}$ (Bayes model). We know that $y = wx + \epsilon$ where $\epsilon$ is the only source of randomness in the learning sample and drawn from a normal distribution $\mathcal{N}(O, \sigma^2)$ meaning its expectation is 0 (will be used when simplifying expressions). Thus, the analytical expression simply becomes :

$$noise(x_0) = E_{y|x_0}\{((wx_0 + \epsilon)) - wx_0)^2\} = E_{y|x_0}\{\epsilon^2\} = noise(x_0)$$

   We know in general that for any random variable X, $Var\{X\} = E\{X^2\} - E\{X\}^2$. So, in our case,

$$E_{y|x_0}\{\epsilon^2\} = Var\{\epsilon\} + E\{\epsilon\}^2 = Var\{\epsilon\} = \sigma^2$$

4. Squared bias, $bias^2(x_0)$
   Bias measures the error between the Bayes model and the average model.
   It is given by $bias^2(x_0) = (h_B(x_0) - E_{LS}\{\hat{y}(x_0)\})^2$ In our case, the true value of the model is given by $y = wx + \epsilon$ where $\epsilon$ is normally distributed with a mean of 0. The Bayes model $h_B(x) = wx$ and for the ridge regression, the prediction at $x_0$ is given by $\hat{y}(x_0) = \hat{w}_{LS}x_0$ where $\hat{w}_{LS}$ is the estimate for the weight w using the ridge regression.
   Thus,

$$bias^2(x_0) = (h_B(x_0) - E_{LS}\{\hat{y}(x_0)\})^2$$

   Given that $E_{LS}\{\hat{y}(x_0)\}$ is $E_{LS}\{\hat{w}_{LS}\}x_0$ as as $E\{\epsilon\} = 0$ and $h_B(x_0)$ is $wx_0$, the expression becomes :

$$bias^2(x_0) = (wx_0 - E_{LS}\{\hat{w}_{LS}\}x_0)^2$$

   In addition, the expectation with respect to $\epsilon_i$ is considered as $x_i$ are fixed and w, $\lambda$ are constants, and we can substitute $y_i = wx_i + \epsilon_i$, so we have

$$E_{LS}\{\hat{w}_{LS}\} = E_{LS}\left\{\frac{\sum_{i=1}^N x_i y_i}{\sum_{i=1}^N x_i^2 + \lambda}\right\} = E_{LS}\left\{\frac{\sum_{i=1}^N x_i(wx_i + \epsilon_i)}{\sum_{i=1}^N x_i^2 + \lambda}\right\} = E_{LS}\left\{\frac{w\sum_{i=1}^N x_i^2 + \sum_{i=1}^N x_i\epsilon_i}{\sum_{i=1}^N x_i^2 + \lambda}\right\}$$

   But since $\sum_{i=1}^N x_i^2$ and $\lambda$ are constants, they come out of the expectation and the expectation of the sum involving $\epsilon_i$ is zero, we have

$$E_{LS}\{\hat{w}_{LS}\} = \frac{w\sum_{i=1}^N x_i^2}{\sum_{i=1}^N x_i^2 + \lambda} = \frac{ws_{xx}}{s_{xx} + \lambda}$$

$$bias^2(x_0) = (wx_0 - E_{LS}\{\hat{w}_{LS}\}x_0)^2 = (wx_0 - \frac{ws_{xx}}{s_{xx} + \lambda}x_0)^2$$

   This can be even more simplified by and it is as follows

$$bias^2(x_0) = \left(\frac{wx_0\lambda}{s_{xx} + \lambda}\right)^2$$

5. Variance, $variance(x_0)$
   The $variance(x_0)$ quantifies how much the average over the whole input space varies from one learning sample to another. In our ridge regression model, the prediction for a given input $x_0$ is : $\hat{y}(x_0) = \hat{w}_{LS}x_0$. Furthermore, the variance is concerned with how much $\hat{y}(x_0)$ varies as the training

data varies while $x_0$ remains fixed! This variance is due to the randomness in the training data particularly the noise terms $\epsilon_i$. Variance of the prediction at $x_0$ is :

$$variance(x_0) = E_{LS}[(\hat{y}(x_0) - E_{LS}[\hat{y}(x_0)])^2]$$

Since $E[\hat{y}(x_0)]$ is $E[\hat{w}_{LS}]x_0$, we have :

$$variance(x_0) = E_{LS}[(\hat{w}_{LS}x_0 - E[\hat{w}_{LS}]x_0)^2]$$

which simplifies with the help of above results to :

$$variance(x_0) = E_{LS}[(\hat{w}_{LS}x_0 - \frac{ws_{xy}}{s_{xx}+\lambda}x_0)^2]$$

By using the general rules $Var(cX) = c^2Var(X)$ and afterwards $E(x^2) = Var(X) + E(X)^2)$, we get

$$variance(x_0) = x_0^2 E_{LS}\left\{(\hat{w}_{LS} - \frac{ws_{xx}}{s_{xx}+\lambda})^2\right\}$$

$$variance(x_0) = x_0^2\left(Var\{\hat{w}_{LS} - \frac{ws_{xx}}{s_{xx}+\lambda}\} + E_{LS}\left\{\hat{w}_{LS} - \frac{ws_{xx}}{s_{xx}+\lambda}\right\}^2\right)$$

The fact that $\frac{ws_{xx}}{s_{xx}+\lambda}$ is a constant allows the simplification using the general rule $Var(X + c) = Var(X)$ as follows

$$variance(x_0) = x_0^2\left(Var\{\hat{w}_{LS}\} + E_{LS}\left\{\hat{w}_{LS} - \frac{ws_{xx}}{s_{xx}+\lambda}\right\}^2\right)$$

By using $E(X^2) - Var(X) = E(X)^2$, we get

$$variance(x_0) = x_0^2\left(Var\{\hat{w}_{LS}\} + E_{LS}\left\{\left(\hat{w}_{LS} - \frac{ws_{xx}}{s_{xx}+\lambda}\right)^2\right\} - Var\{\hat{w}_{LS} - \frac{ws_{xx}}{s_{xx}+\lambda}\}\right)$$

Using again the general rule $Var(X + c) = Var(X)$, we have

$$variance(x_0) = x_0^2\left(Var\{\hat{w}_{LS}\} + E_{LS}\left\{\left(\hat{w}_{LS} - \frac{ws_{xx}}{s_{xx}+\lambda}\right)^2\right\} - Var\{\hat{w}_{LS}\}\right)$$

Which gives us

$$variance(x_0) = x_0^2 E_{LS}\left\{\left(\hat{w}_{LS} - \frac{ws_{xx}}{s_{xx}+\lambda}\right)^2\right\}$$

By expanding the square, we obtain

$$variance(x_0) = x_0^2 E_{LS}\left\{\hat{w}_{LS}^2 - \frac{2ws_{xx}\hat{w}_{LS}}{s_{xx}+\lambda} + \frac{w^2 s_{xx}^2}{(s_{xx}+\lambda)^2}\right\}$$

Again, this simplifies to

$$variance(x_0) = x_0^2\left\{E_{LS}\left\{\hat{w}_{LS}^2\right\} - \frac{2ws_{xx}}{s_{xx}+\lambda}E_{LS}\left\{\hat{w}_{LS}\right\} + \frac{w^2 s_{xx}^2}{(s_{xx}+\lambda)^2}\right\}$$

Since $E_{LS}\{\hat{w}_{LS}\} = \frac{ws_{xx}}{s_{xx}+\lambda}$, we have

$$variance(x_0) = x_0^2 \left\{ E_{LS}\left\{\hat{w}_{LS}^2\right\} - 2w^2 \left(\frac{s_{xx}}{s_{xx}+\lambda}\right)^2 + \frac{w^2 s_{xx}^2}{(s_{xx}+\lambda)^2} \right\}$$

Let's consider the $E_{LS}\left\{\hat{w}_{LS}^2\right\}$,

$$E_{LS}\left\{\hat{w}_{LS}^2\right\} = E_{LS}\left\{\hat{w}_{LS}\right\}^2 + Var\{\hat{w}_{LS}\} = \left(\frac{ws_{xx}}{s_{xx}+\lambda}\right)^2 + Var\{\hat{w}_{LS}\}$$

Putting back the value into the equation of the variance we found gives us

$$variance(x_0) = x_0^2 \left\{ \left(\frac{ws_{xx}}{s_{xx}+\lambda}\right)^2 + Var\{\hat{w}_{LS}\} - 2w^2 \left(\frac{s_{xx}}{s_{xx}+\lambda}\right)^2 + \frac{w^2 s_{xx}^2}{(s_{xx}+\lambda)^2} \right\}$$

Which simplifies to

$$variance(x_0) = x_0^2 \left\{ Var\{\hat{w}_{LS}\} \right\}$$

Let's consider $Var\{\hat{w}_{LS}\}$, knowing that the variance of a constant is zero, thet $e_i$ have a variance of $\sigma^2$ by definition and that $Var(cX) = c^2 Var(X)$,

$$Var\{\hat{w}_{LS}\} = Var\left\{ \frac{w\sum_{i=1}^N x_i^2 + \sum_{i=1}^N x_i\epsilon_i}{\sum_{i=1}^N x_i^2 + \lambda} \right\} = \frac{\sum_{i=0}^N x_i^2}{(s_{xx}+\lambda)^2} Var(e_i) = \frac{\sum_{i=0}^N x_i^2}{(s_{xx}+\lambda)^2}\sigma^2$$

Thus, the final expression of the variance is

$$variance(x_0) = x_0^2 \frac{\sum_{i=0}^N x_i^2}{(s_{xx}+\lambda)^2}\sigma^2$$

$$variance(x_0) = x_0^2\sigma^2 \frac{s_{xx}}{(s_{xx}+\lambda)^2}$$

## 1.2 Impact of paramaters on bias and variance

Now let us discuss the impact of the following parameters on both bias and variance.
Recall that $\hat{w}_{LS} = \frac{s_{xy}}{s_{xx}+\lambda}$, $bias^2(x_0) = \left(\frac{wx_0\lambda}{s_{xx}+\lambda}\right)^2$ and $variance(x_0) = x_0^2\sigma^2\frac{s_{xx}}{(s_{xx}+\lambda)^2}$

a. **Regularization level $\lambda$**
$\lambda$ is a non-negative parameter so it makes sense to understand how bias and variance act when $\lambda$ tends towards $+\infty$ or 0.

I. $\lambda \to +\infty$

A. Bias
In the limit, the bias term approaches $wx_0$ and $bias^2$ approaches $wx_0^2$ which becomes large if w and $x_0$ are non-zero.

B. Variance
The variance approaches zero as $\lambda$ grows because the denominator grows much faster than the numerator. Thus, a very large $\lambda$ allows model's prediction to become very stable with low variance but at the cost of a high bias if true weights are not near zero.

II. $\lambda \to 0$

A. Bias
bias is 0. The ridge estimator is unbiased and this is typical to the classical least squares estimator.

B. Variance
Variance becomes $\frac{x_0^2\sigma^2}{s_{xx}}$. There is no regularization to constrain the weights. This can lead to high variance.

III. Discussion

It seems that in the context of the ridge regression, the $\lambda$ parameter has a high impact on multiple aspects. First, when $\lambda$ is too small, overfitting occurs as the model is too complex. There is no regularization so the model may fit the training data very well but it will likely perform poorly on the unseen data because of its low bias and high variance.

Then, when $\lambda$ is too large, it can lead to underfitting. Underfitting is caused by the shrinkage of coefficients (coefficients tends to become so small due to the optimization of the cost function that they may approach zero resulting in a model that is too simple and doesn't fit the training data).

In addition to this, there is a clear bias-variance trade-off as increasing $\lambda$ introduces bias with a significant reduction in variance and conversely when $\lambda$ is decreasing.

b. **Noise** $\sigma$

I. Bias

The bias doesn't depend on $\sigma$ as can be seen by looking at its expression. So, $\sigma$ has no effect on the bias.

II. Variance

Variance is directly proportional to $\sigma^2$.

A. $\sigma \to +\infty$
As $\sigma$ increases, variance increases with no bound.

B. $\lambda \to +0$
As $\sigma$ decreases, so does the variance.

III. Discussion

As $\sigma$ increases, so does the variances, bringing more noise. Thus, the model's predictions are less reliable with higher uncertainty.

c. **Learning sample size** $N$

Following the suggestion, $x_i$ are assumed to have been drawn from a normal distribution $\mathcal{N}(0,1)$. We know that $s_{xx}$ is defined as

$$s_{xx} = \sum_{i=1}^{N} x_i^2$$

Each $x_i^2$ is a random variable that follows a chi-squared distribution with degree 1 of freedom because it is the square of a standard variable following the normal distribution as assumed by the suggestion. The expectation of such variable is its degree of freedom which in our case is 1. Therefore,

$$E\{x_i^2 = 1\}$$

For a large number of $x_i$, by the law of large numbers, the average of the $x_i^2$'s will converge to the expected value of a single $x_i^2$ as N tends to $+\infty$. As a reminder, the law of large numbers states that as the size of the sample increases, the sample mean (in our cas, sample mean of $x_i^2$) will converge to the expected value of the population mean. This means

$$\frac{s_{xx}}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i^2 \approx E\{x_i^2 = 1\}$$

$$s_{xx} \approx N$$

The larger the N, the closer $s_{xx}$ will be equal to N.
With that in mind we can then analyse how will the bias and variance be influenced by N.

A. Bias

$$bias^2(x_0) = \left(\frac{wx_0\lambda}{s_{xx}+\lambda}\right)^2$$

If N increases, $s_{xx}$ increases, thus $s_{xx}+\lambda$ increases too resulting in a smaller ratio $\frac{\lambda}{s_{xx}+\lambda}$ meaning a smaller bias.

B. Variance

$$variance(x_0) = x_0^2\sigma^2\frac{s_{xx}}{(s_{xx}+\lambda)^2}$$

Numerator is directly proportional to N but the denominator increases faster as N grows so the expression will decrease as N grows.

C. Discussion
As N increases, both bias and variance decreases. This shows that model's predictions are becoming more consistent across different samples.

## 1.3 Analytical expression of $\lambda_*$

The goal of the following section is to derive an analytical expression of the optimal value of $\lambda_*$, i.e. the value of $\lambda$ that leads to the smallest expected generalization error.
As seen from the theoretical lecture, the expected generalized error is given by

$$E = var_y\{y\} + bias^2 + var_{LS}\{\hat{y}\}$$

Which in our case becomes

$$E(x_0) = Var_y\{y(x_0)\} + bias^2(x_0) + Var_{LS}\{\hat{y}(x_0)\}$$

where we replace $x_0$, the test point, by $\lambda$.
This expression contains :

1. $Var_y\{y(x_0)\}$ is the irreducible error
2. $bias^2(x_0)$ is the squared bias
3. $Var_{LS}\{\hat{y}(x_0)\}$ is the variance of the model's predictions in the learning sample

Given that only the squared bias and the variance of the model's predictions in the learning sample depend on $\lambda$, to find the optimal $\lambda$, $\lambda^*$, we need to minimize the sum of the squared bias and the mentioned variance. To do so, We take the derivative of $E(\lambda)$ with respect to $\lambda$ and we equal it to 0. Solving for $\lambda$ gives $\lambda^*$. Therefore

$$\frac{\partial E(\lambda)}{\partial \lambda} = 0$$

$$\frac{\partial}{\partial \lambda}\left(\left(\frac{wx_0\lambda}{s_{xx}+\lambda}\right)^2 + \frac{x_0^2\sigma^2 s_{xx}}{(s_{xx}+\lambda)^2}\right) = 0$$

$$\frac{2\lambda s_{xx}w^2x_0^2}{\lambda^3+s_{xx}^3+3\lambda s_{xx}^2+3\lambda^2 s_{xx}} - \frac{2s_{xx}\sigma^2x_0^2}{\lambda^3+s_{xx}^3+3\lambda s_{xx}^2+3\lambda^2 s_{xx}} = 0$$

$$\frac{2\lambda s_{xx} w^2 x_0^2}{(\lambda + s_{xx})^3} - \frac{2 s_{xx} \sigma^2 x_0^2}{(\lambda + s_{xx})^3} = 0$$

Solving this for $\lambda$ gives

$$\lambda^* = \frac{\sigma^2}{w^2}$$

In order to discuss the expression, we can consider two things :

1. The numerator $\sigma^2$
   This is the noise. If the noise is high, it would need a higher value of $\lambda$ to prevent overfitting.
2. The denominator $w^2$
   This is the true weight also referenced as the signal.

The optimization process for finding $\lambda^2$ finally showed that $\lambda^2$ is inversely proportional to the well known signal-to-noise ratio but applied in the context of the ridge regression. We can adapt the traditional way of thinking to this new "noise-to-signal" ratio. A high "noise-to-signal" ratio calls for more regularization to prevent the model from learning noise as if it were a true signal. Conversely, a low "noise-to-signal" ratio means the true underlying pattern is clear and the model doesn't need as much regularization to make accurate predictions.

# 2 Empirical analysis

## 2.1 Residual error estimation

Let's start by observing that estimating the residual error will be quite hard in this setting.

The residual error is represented as $var_{y|x}(y)$, which is difficult to estimate because it's a variance calculated at specific input points. Given that our dataset is finite and consists of continuous values, it's highly improbable to have a significant number of outputs for the same input. Since our learning set has a high dimension makes it even more unlikely. In the vast majority of cases, we'll have a unique output for each input, which makes the computation of $var_{y|x}(y)$ excessively difficult.

## 2.2 Estimation protocol

Protocol to estimate variance, the expected error and the sum of the bias and the residual error in few steps.

1. Create Learning Sample :
   We decide to take the same separation ratio as the previous project, so 80% of it to create 50 random training sets (shuffled from the big training set) of size 500 (size for next question) and 20% to test the model.
2. Fit the model to the learning samples and test it on the test set.
3. Compute the expected error :
   Take the mean of the squares of the differences between all true values of output and the outputs which is computed with the model. Then compute the mean of this value over all 50 learning samples.

$$E_{LS}\Big\{ E_{y|\underline{x}}\{(y - \hat{y}(\underline{x}))^2\} \Big\}$$

4. Compute the variance :
   Compute the mean over all learning samples of the squares of the differences between the predicted values and the mean of the predicted values.

$$E_{LS}\Big\{(\hat{y}(\underline{x}) - E_{LS}\{\hat{y}(\underline{x})\})^2\Big\}$$

5. Compute the sum of the squared bias and the residual error :
   Simply subtract the variance from the expected error.

$$\text{noise}(\bar{x}) + \text{bias}^2(\bar{x}) = E_{LS}\Big\{E_{y|\underline{x}}\{(y - \hat{y}(\underline{x}))^2\}\Big\} - \text{Var}(\underline{x})$$

## 2.3 Impact of the complexity on bias and variance

For all three models, we observe the impact of their complexity (related to the value of their respective hyper-parameter : $\alpha, k, max\_depth$) on the expected error, the variance and the bias.

Generally, we should observe an increase in variance and a drop of bias as the complexity of the model increases. Indeed, as the model gets more and more complex, it starts to better fit our learning data which leads to more possible outputs for each inputs of the test set and thus a higher variance.
As the complexity increases, the mean output of the model also gets closer to the real one which leads to a lower bias. On the other hand, as a model gets simpler, the bias gets bigger and bigger (and the variance gets smaller).

- Overfitting :
At some point, it might even start fitting the noise in our data which will significantly increase the error of the model.

- Underfitting :
At some point, the model can even become too simple. The corresponding increase in bias will then significantly increase the error and prevent the model from being accurate.
⇒ underfitting and overfitting are bad to the generalization error of the model.

Since the expected error is the sum of the bias and the variance, it is expected to first decrease with the decrease of bias and then increase because of the variance.

The bias-variance tradeoff is the fact we cannot decrease as much as wanted the bias and the variance at the same time.

### 2.3.1 Ridge regression

For the ridge model, it's important to understand that high values of $\alpha$ means low complexity of the model.

Ridge regression is a modified version of the classic regression algorithm. Instead of minimizing the residual sum of squares ($RSS$), it wants to add the product of $\alpha$ by the sum of the squared coefficient to this loss function. It will allow ridge regression to penalize large coefficients $\xi$.

$$\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \alpha\sum_{j=1}^{M}\xi_j^2 \quad \alpha \in [0; \infty]$$

Minimizing this loss function is equivalent to this :

$$\text{argmin}(RSS) \rightarrow \sum_{j=1}^{M} \xi_j^2 < \frac{1}{\alpha}$$

With equation above and the Figure 1 :

— For small complexity (larger $\alpha$ gets), the smaller the sum of the square of the coefficient must be in order to be selected to minimize the loss function. They (the $\xi$'s) can less vary in the values they can take. This induces less variability and a higher bias.
— For high complexity ($\alpha$ gets really small), the $\xi$'s can take more possible values to possibly take which directly leads to higher variance and lower bias.

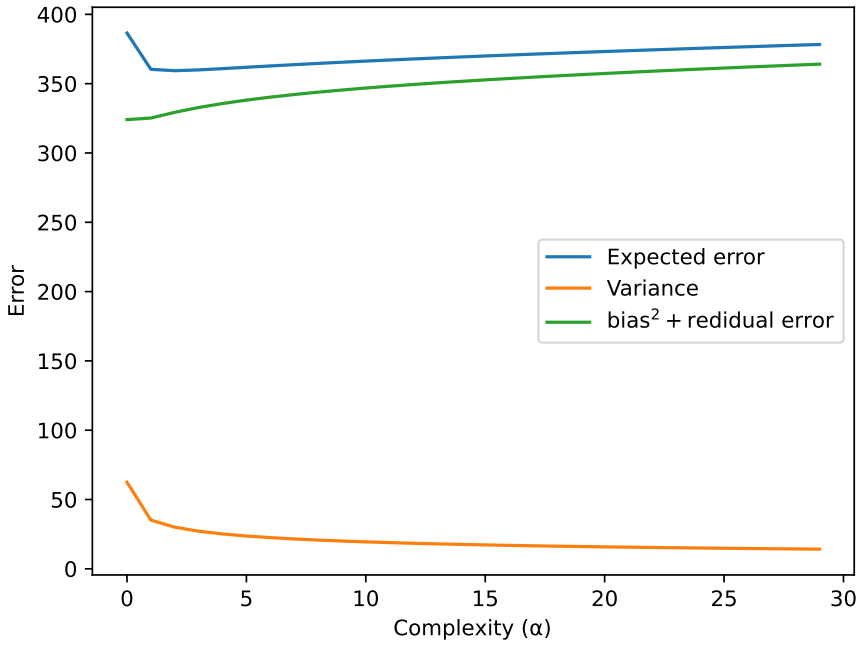



FIGURE 1 – Error of Ridge regression related to complexity

### 2.3.2 k-NN

For the k-NN model, k represents the number of neighbours taken into account in order to make predictions.
- If $k$ is small, the model will get extremely complex in order to correctly predict every value given in the learning set (sensitive to noise), leading to high variance and low bias. Case of overfitting
- If $k$ increases, the model becomes less complex and its predictions become more stable, but it may start to underfit, leading to higher bias and lower variance.
- If $k$ is the total number of inputs in the learning set, this model yields the same value for any input from the test set, the variance in such an extreme case is not zero because it depends on the learning set (it will reach some constant)

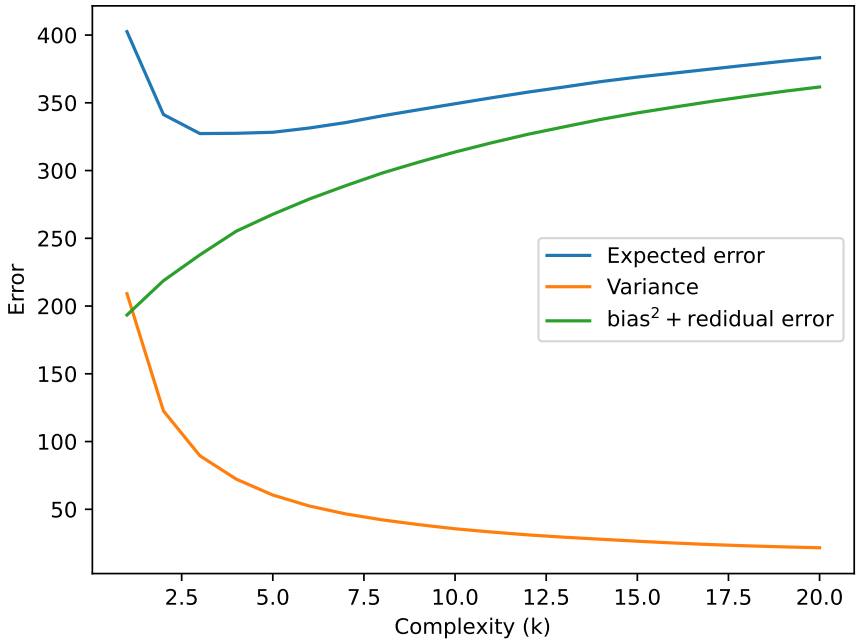



FIGURE 2 – Error of k-NN related to complexity

### 2.3.3 Regression trees

For the regression tree, the maximal depth of the tree is a measure of the complexity of the model. Observation made in Figure :3 :

— For very low complexity (very low depth), the bias dominates the variance. The model is underfitting and the error is important. For example, with a maximum depth of only 1, the tree makes a single separation in the learning set and cannot include details that the learning set may have such as a more complex curve that cannot be approximated by a single line (low complexity).

— When increasing the depth of the tree (tends towards a fully grown tree), the variance increases and the model starts to overfit. This leads to a smaller bias but an increasing error. If we use, the maximal depth that will allow to perfectly predict the learning set up to the smallest detail (high complexity), it will not predict the outputs of the test set very well because the noise of the learning sample is fit by the mode.

$\Rightarrow$ In this model, the complexity does not need to be high to obtain less error and a better bias.

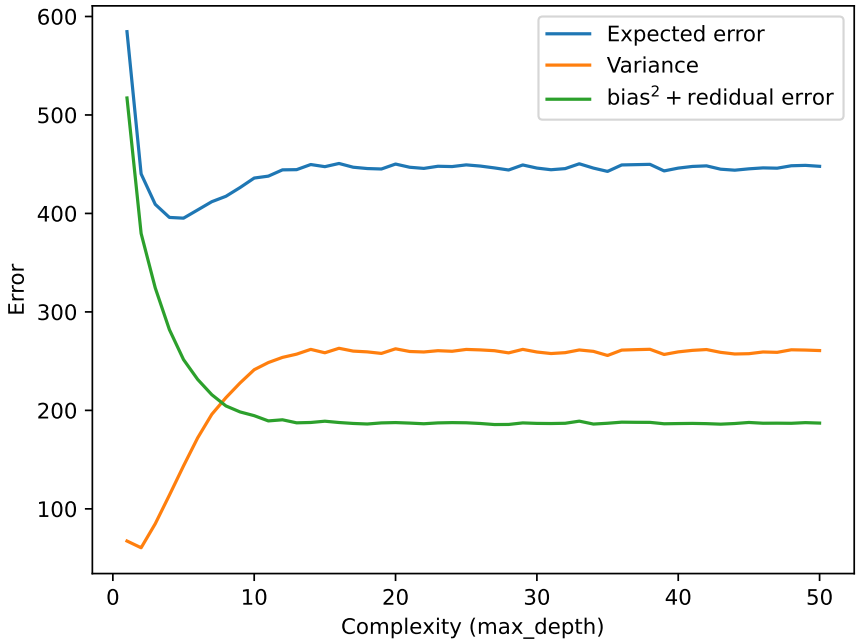



FIGURE 3 – Error of tree regression related to complexity

## 2.4 Impact of the size of LS on bias and variance

### 2.4.1 Ridge regression

Ridge regression is a type of linear regression that penalizes large coefficients by adding a term proportional to their squared sum to the loss function. The complexity of ridge regression depends on the size of the learning set and the value of the penalty parameter $\alpha$ (see equation 2.3.1).

In Figure 4 :

— For larger learning sets, the variability of the coefficients is reduced and increase the bias, while larger $\alpha$ values constrain the coefficients more and reduce the variance. When $\alpha$ is null, we are back with the classic linear regression whose complexity is independent from the size of the learning set.

Indeed, the loss function of ridge regression, as expressed in equation (2.3.1), indicates that increasing the number of data points only increases the Residual Sum of Squares (RSS) and not the second term. This restricts the coefficients from changing significantly under fluctuations in the learning set. The set of possible $\xi$ (and hence the set of possible hyperplanes) coming from different datasets is much more limited when the number of data points is large.

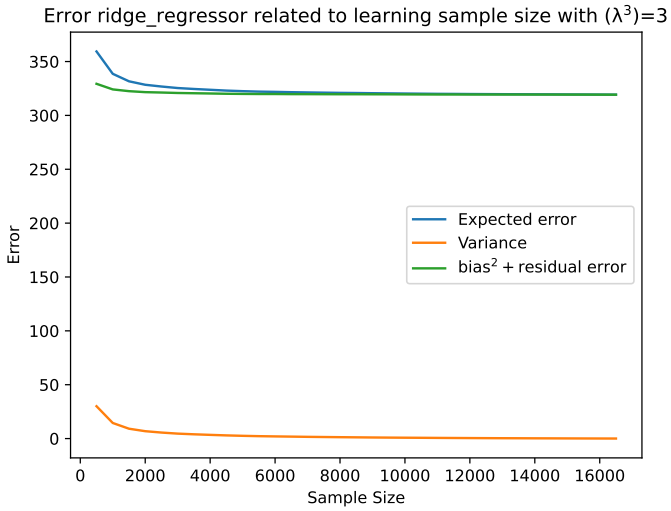



FIGURE 4 – Ridge regression model : impact of the size of the learning samples on error, variance and bias

### 2.4.2 k-NN

In Figure 5 :

— The variance of a kNN model tends to decrease as the size of the learning sample increases. This is because with more data, the model has a larger pool of neighbors to choose from, which can lead to more stable and consistent predictions. However, the effect on variance also depends on the value of k. With k set to 7, the model is somewhat robust to noise in the data, but it might still be sensitive to local fluctuations in the data distribution.

— The bias decreases but the model complexity is fixed, it means our parameter value is poorly choose. Indeed, a fixed $k$ value means the model always considers the same number of neighbors, regardless of the size of the learning sample. When the learning sample size increases, the nearest neighbors are likely to be closer to the point chosen in the feature space, leading to more accurate predictions and lower bias.

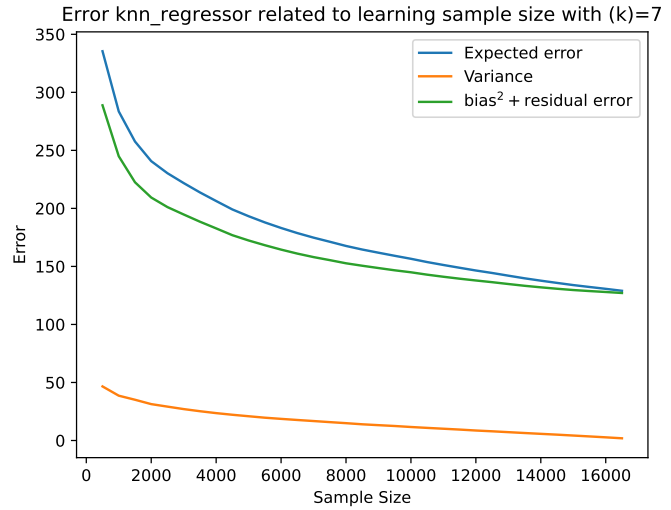


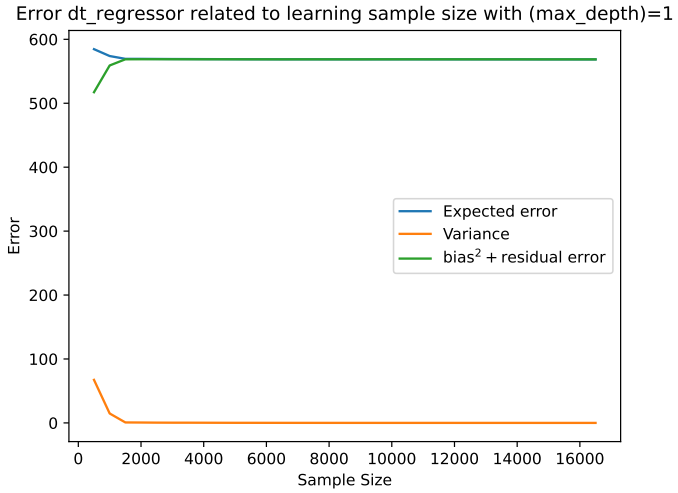

FIGURE 5 – k-NN Model : impact of the size of the learning samples on error, variance and bias

### 2.4.3 Regression trees

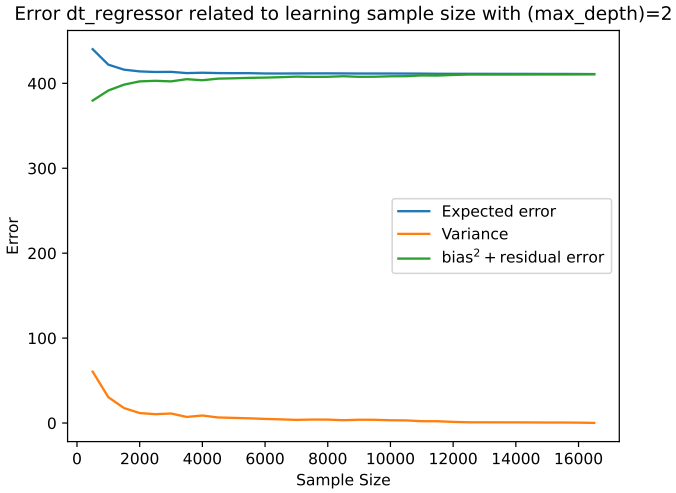
The complexity of this algorithm is dependent on the size of the learning set which means we'll observe an evolution of all the values. Indeed, it can be seen in figure 6 that the bigger the learning set, the lower the values of bias, variance and expected error. In other words, the model get better with a bigger learning set.
For regression trees, two cases are possible :

— For fixed depths, we observe on Figure 6 that the bias is constant and the variance drops when increasing the leaning sample size. As explained above, in this case the complexity of the model doesn't depend on the learning sample size. Another thing we notice is that as said in Question 2.3, when the depth increases, the variance increases and the bias decreases.

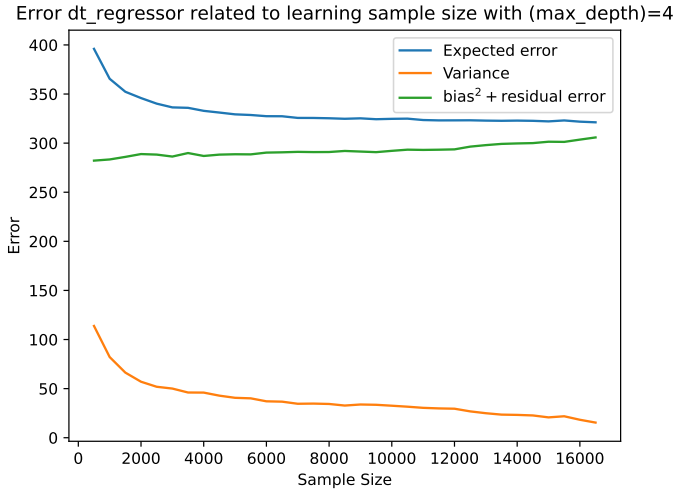
— For fully grown trees, the complexity depends on the learning sample size. Indeed, the depth of the tree will depend on the number of samples in the learning sample. We observe that the bias as the variance decreases with the number of LS.



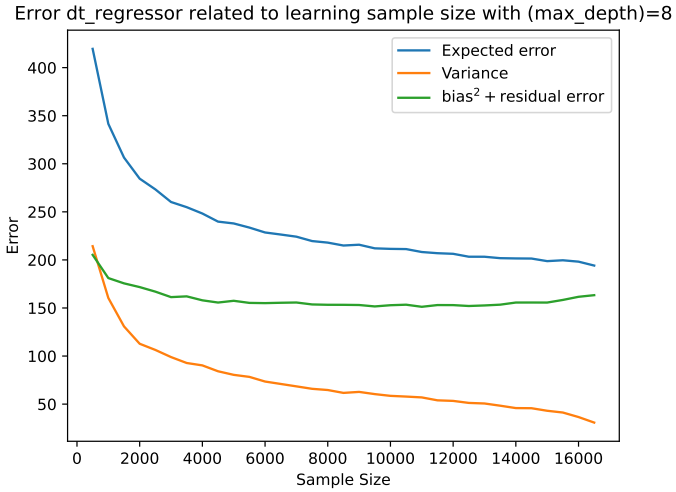

(a) Regression tree model : $max\_depth = 1$




(b) Regression tree model : $max\_depth = 2$

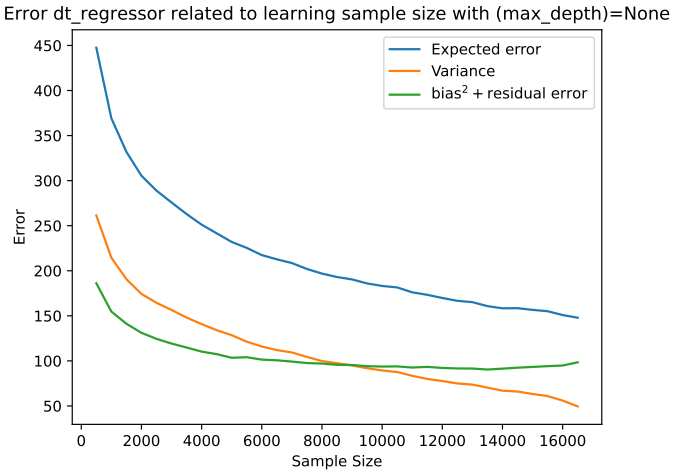



(c) Regression tree model : $max\_depth = 4$




(d) Regression tree model : $max\_depth = 8$

FIGURE 6 – Error of Decision tree regressor related to learning sample size with different depth size




(e) Regression tree model : $max\_depth = None$

FIGURE 6 – Error of Decision tree regressor related to learning sample size with different depth size

### 2.4.4 Conclusion

Overall in all 3 cases,we see that when we increase the number of learning samples, the error and the variance automatically decreases and ends up by stabilizing.

## 2.5 Bootstrap aggregating (bagging method)

Bagging, or bootstrap aggregating, is an ensemble machine learning method that involves creating multiple subsets of the original training data with replacement (bootstrap samples), training a model on each subset, and then averaging the results to form an ensemble prediction. It is particularly effective to reduce variance and improve prediction stability, often with a slight increase in bias.

We are implementing bagging with fully grown regression trees, ridge regression with $\lambda$=30 and k-Nearest Neighbors with k=1. Our approach uses 80% of the instances for training and 20% for testing. Here are the results obtained :

| **Model** | **Bias** | | **Variance** | |
|---|---|---|---|---|
| | w/o Bag. | w/ Bag. | w/o Bag. | w/ Bag. |
| Ridge Regression | 323.2082 | 323.1924 | 838.0827 | 0.7438 |
| kNN | 121.2344 | 104.4242 | 1169.0972 | 47.5971 |
| Regression Tree | 141.3675 | 100.2985 | 1128.2149 | 80.1095 |

TABLE 1 – Comparison of Bias and Variance for Models With and Without Bagging

### 2.5.1 Ridge regression (with fixed $\lambda$=30)

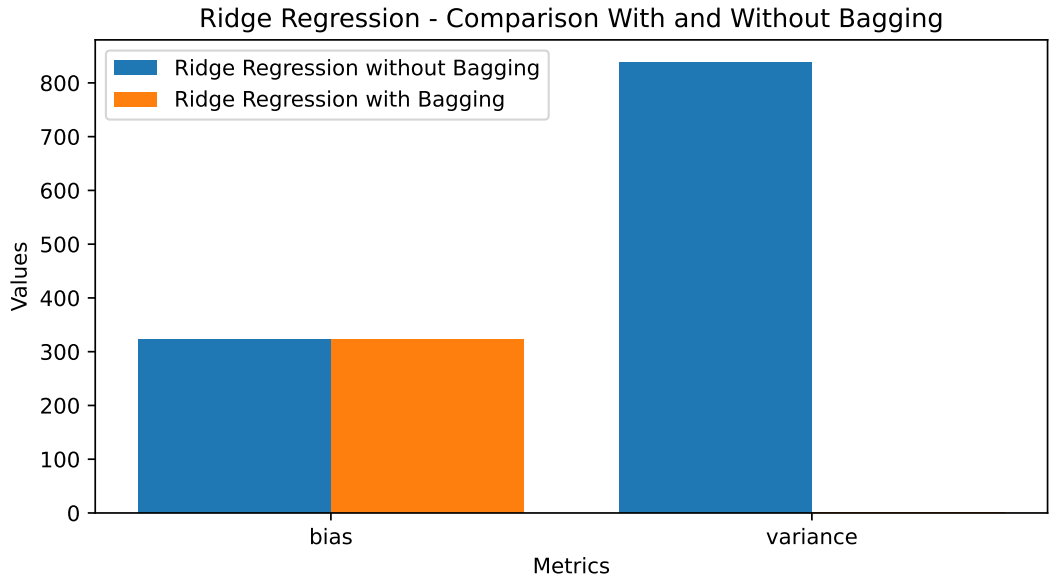



FIGURE 7 – Comparison of Ridge Regression Model With and Without Bagging

The application of bagging on Ridge Regression showed a remarkable reduction in variance. This decrease in variance is a significant outcome considering that Ridge Regression, a regularized form of linear regression, is already designed to manage the trade-off between bias and variance. Contrary to typical expectations with bagging, the bias remained largely unaffected, indicating that the averaging effect did not introduce the slight increase in bias often associated with this technique.

### 2.5.2 k-NN (w/ k=1)

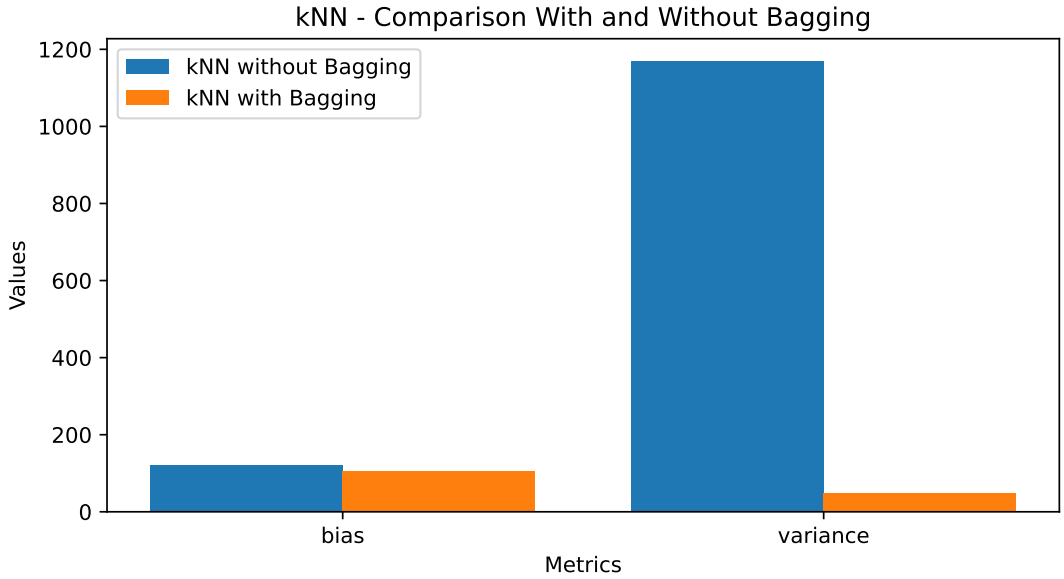



FIGURE 8 – Comparison of kNN Model With and Without Bagging

The application of bagging on kNN significantly reduced the variance, addressing the model's tendency for high variance, especially when k=1. While there was also a reduction in bias, it was less pronounced compared to the reduction in variance. This reduction in bias might be partly attributed to the averaging effect of bagging, which can smoothen the model's sensitivity to noise in the training data.

### 2.5.3 Regression trees (fully grown, w/ depth=None)

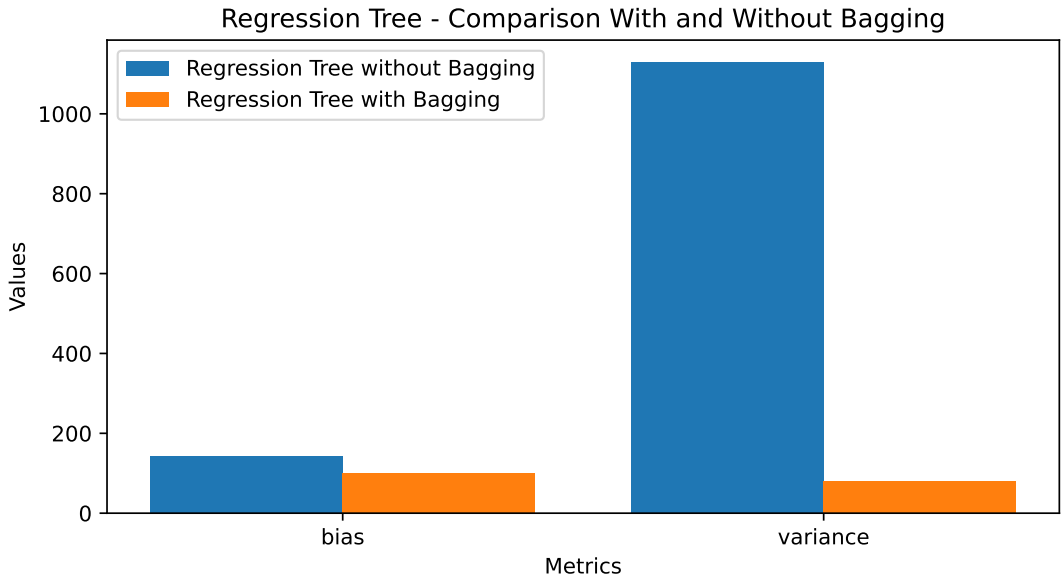



FIGURE 9 – Comparison of Regression Tree Model With and Without Bagging

For Regression Trees, bagging led to a substantial reduction in variance, which is particularly important given the model's propensity for high variance due to overfitting in its fully grown state. The reduction in bias, though present, was less substantial than the reduction in variance.

### 2.5.4 Conclusion

Bagging proved to be highly effective in reducing variance across all three models. This is consistent with the theoretical understanding of bagging, which aims to reduce overfitting through averaging multiple predictions from models trained on different subsets of data.