BIAS–VARIANCE TRADEOFF WITH PRACTICAL ML EXPERIMENTS

Machine Learning and Neural Networks

Muhammad Adnan Sajid

24079380

DECEMBER 11, 2025

UNIVERSITY OF HERTFORDSHIRE

Table of Contents

Table of Figures

Abstract

The bias–variance trade-off is one of main concept in supervised machine learning that describes how model complexity affects prediction accuracy. Increase in flexibility reduces bias but increases variance that causes over-fitting by fitting noise rather than the true pattern *(Brady Neal, et al., 2018)*. This creates a U-shaped relationship between generalization error and complexity.

This report explains the trade-off using theory and synthetic regression experiments. The error decomposition formula shows prediction error divided into bias, variance, and irreducible noise, and how complexity impacts these components.

Polynomial regression models of varying degrees shows under-fitting (high bias), over-fitting (high variance), and the optimal balance. Results confirm that simple models perform poorly on both training and test sets, while complex models achieve low training error but weak generalization.

Finally, the report also explains the practical significance of bias and variance in ML systems and notes experimental limitations, which provides a clear, concise understanding supported by theory and experiments *(Wikipedia, 2024)*, *(geeks, 2025)*.

1. Introduction

Supervised machine learning models aim to make accurate predictions on new data by learning from a trained dataset. However, models face two main types of error: **bias** and **variance**. Bias refers to systematic error due to simplifying assumptions in the model, while variance reflects sensitivity to fluctuations in the training data. The **bias–variance trade-off** captures the observation that reducing one often increases the other *(Hastie, et al., 2009)* , *(Wickramasinghe, 2024)*. In practice, simple models (e.g. linear regression) tend to under fit (high bias), missing underlying patterns, whereas complex models (e.g. deep trees or high-degree polynomials) may over-fit (high variance), memorizing noise and failing to generalize *(Christopher Burns, et al., 2012)*.

Balancing these components is crucial for model selection and tuning, we seek a “sweet spot” where both bias and variance are moderate, yielding low overall error.

2. Background and Literature Review

The bias–variance trade-off is a well-established concept in statistics and machine learning *(Hastie, et al., 2009)* , *(Wickramasinghe, 2024)*. It arises from the theoretical analysis of prediction risk, which explains why a model with strong training performance may still fail to generalize to unseen data. In a regression setting, let the true target variable be generated as

$$Y = f(X) + \epsilon$$

Where the noise term satisfies $E[\epsilon] = 0$ and Var$[\epsilon] = \sigma^2$. For a learned model $\hat{f}$ trained on a dataset, the expected prediction error at a specific input point x, under squared loss, is given by

$$\text{EPE}(x) = E[(Y - \hat{f}(x))^2] = \underbrace{(f(x) - E[\hat{f}(x)])^2}_{\text{Bias}^2} + \underbrace{E[(\hat{f}(x) - E[\hat{f}(x)])^2]}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Irreducible Error.}}$$

Mean squared error (MSE) consists of **squared bias**, **variance**, and **noise** *(Hastie, et al., 2009)*. Bias reflects systematic deviation from the true function, while variance measures sensitivity to training data. Increasing model complexity reduces bias but increases variance; simpler models show the opposite effect (Wickramasinghe, 2024).

In practice, linear models on nonlinear data often under-fit due to high bias, while flexible models (e.g., deep decision trees) can over-fit, achieving low training error but poor generalization (Christopher Burns, et al., 2012).

Through learning curves we get to know that high training and validation errors indicate under-fitting while low training but high validation error indicates over-fitting.

Polynomial regression makes us realize that low-degree polynomials under-fit, high-degree polynomials over-fit and optimal performance occurs at intermediate complexity, balancing bias and variance (Wickramasinghe, 2024).

Beyond theory, the bias–variance trade-off has ethical implications. In safety-critical domains such as medical diagnosis, under-fitted or over-fitted models may lead to harmful decisions, making proper model selection essential (Aliferis, Constantin; , Gyorgy J. Simon, 2024).

3. Theoretical Framework

To understand the trade-off mathematically, we first formalize *bias* and *variance*. Consider a training set $\mathcal{D}$ and a target function $f(x) = \mathbb{E}[Y \mid X = x]$. A learning algorithm produces a predictor $\hat{f}(x)$. The *bias* at point x is defined as

$$\text{Bias}(\hat{f}(x)) = \mathbb{E}_{\mathcal{D}}[\hat{f}(x)] - f(x),$$

Where the expectation is over different random training sets of the same size. The *variance* is

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

Using these, the **bias–variance decomposition** of mean squared error is derived (Hastie, et al., 2009) :

$$\mathbb{E}_{\mathcal{D},Y}[(Y - \hat{f}(x))^2] = (f(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}(x)])^2 \; + \; \mathbb{E}_{\mathcal{D}}[(\hat{f}(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}(x)])^2] \; + \; \sigma^2$$

The error decomposition shows that prediction error consists of three parts: squared bias, variance, and irreducible noise $\sigma^2 = Var[Y \mid X = x]$ *(Hastie, et al., 2009)*. Since irreducible error comes from noise in the data itself, it cannot be eliminated. Therefore, reducing total prediction error depends on properly balancing bias and variance.

Model flexibility controls this balance. Simple models (e.g., straight lines for nonlinear data) have **high bias** but **low variance**, while highly flexible models (e.g., high-degree polynomials) achieve **low bias** but **high variance** due to sensitivity to noise *(Hastie, et al., 2009)* *(Christopher Burns, et al., 2012)* .

As complexity rises, **bias decreases** and **variance increases**, creating a **U-shaped prediction error** curve. The minimum occurs at an intermediate complexity, where bias and variance are optimally balanced.

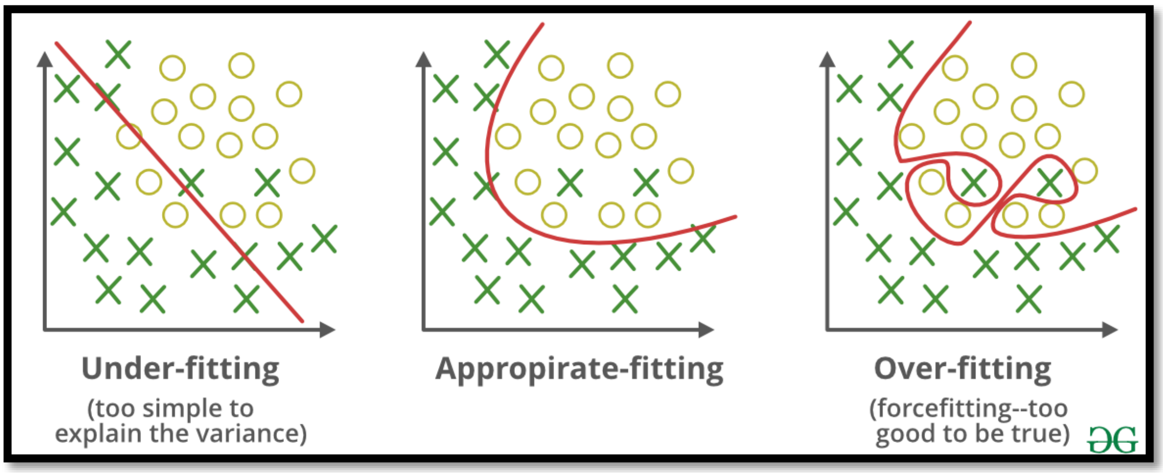

Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)

Figure 1: Examples of model fits at different levels of complexities

An under-fitted model misses the structure, an optimally fitted model captures the trend, and an over-fitted model oscillates to fit noise.

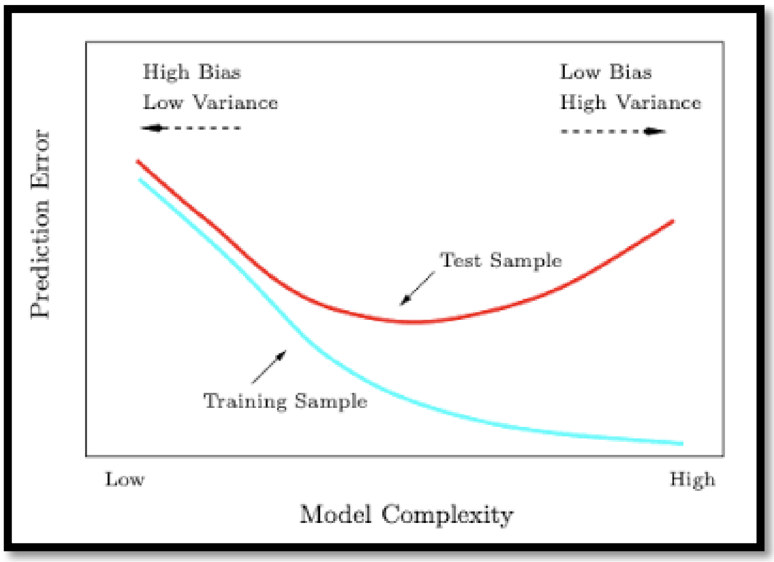


Figure 2: Training and validation error, illustrating the bias–variance trade-off

Figure 2 shows training error always decreasing with complexity while test (validation) error follows a U-shaped curve, reaching its minimum where bias and variance are balanced. These conceptual diagrams help visualize why moderate complexity is often best *(Hastie, et al., 2009)* *(Christopher Burns, et al., 2012)*.

Finally, beyond regression, these concepts apply in classification (e.g. decision boundaries of different smoothness) and other settings. Measures like cross-entropy or 0-1 loss also have analogous bias–variance effects. The key theoretical insight remains to **choose a model whose complexity matches the data**. Too simple or too complex a model leads to large errors.

4. Experimental Design

To demonstrate the bias–variance trade-off, regression experiments are conducted in Python using a synthetic dataset, which allows control over the underlying function and noise. Models of varying complexity are compared to show under-fitting, over-fitting, and the optimal trade-off.

- **Generate Synthetic Dataset**

We create a regression problem where y depends non-linearly on x plus Gaussian noise.

```
import numpy as np
np.random.seed(0)
X = np.random.uniform(0, 10, 100)[:, np.newaxis]
y = 2*X**2 + 3*X + 5 + np.random.normal(0, 5, size=(100,1))
```

Explanation:

- X contains 100 random values in the range [0, 10].
- The target y follows a quadratic function with added Gaussian noise to simulate real-world variation.

Output:

- At this stage, no numerical output is expected.

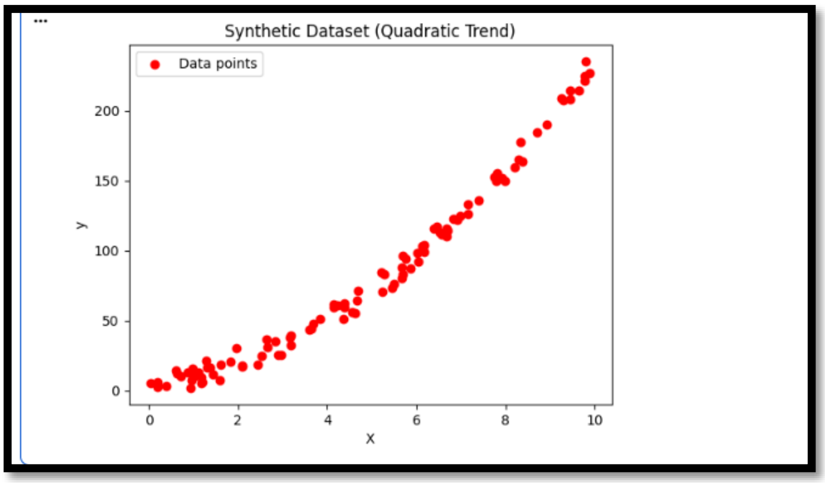


Figure 3: Synthetic dataset (features vs target) with quadratic trend.

- Scatter plot showing the synthetic data points around a quadratic trend.

Split Data into Train/Test

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.3, random_state=42
)
```

Explanation:

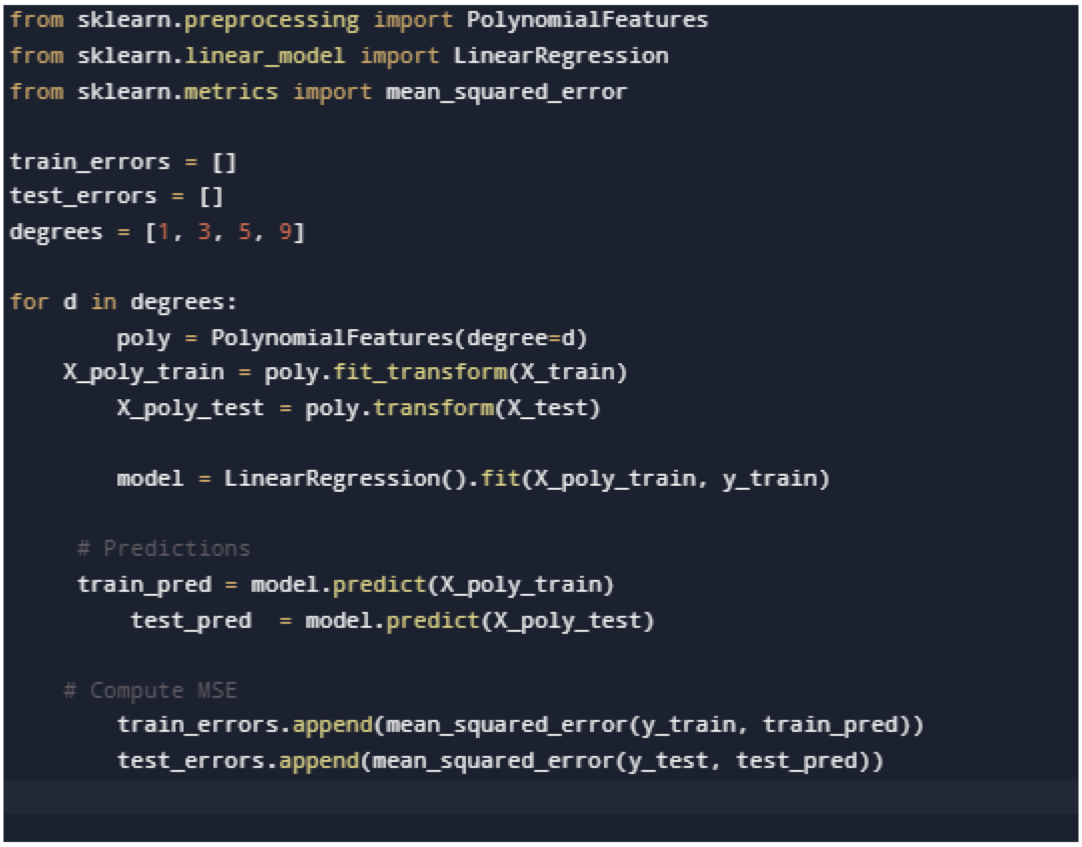
- 70% data for training, 30% for testing.

- Ensures the model is evaluated on **unseen data**.

Output: None, but X_train and X_test are now ready for modeling

Define Models of Increasing Complexity

We fit polynomial regression models of varying degrees (1, 3, 5, and 9) to observe **bias–variance trade-off**:



```python
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

train_errors = []
test_errors = []
degrees = [1, 3, 5, 9]

for d in degrees:
        poly = PolynomialFeatures(degree=d)
    X_poly_train = poly.fit_transform(X_train)
        X_poly_test = poly.transform(X_test)

        model = LinearRegression().fit(X_poly_train, y_train)

     # Predictions
     train_pred = model.predict(X_poly_train)
         test_pred  = model.predict(X_poly_test)

    # Compute MSE
        train_errors.append(mean_squared_error(y_train, train_pred))
        test_errors.append(mean_squared_error(y_test, test_pred))
```

Explanation:

- Polynomial degree = 1 → simple model (likely **high bias**).
- Polynomial degree = 9 → complex model (likely **high variance**).
- Errors are stored in train_errors and test_errors.

Output:

- No immediate output.
- Errors will be visualized next to **observe the trade-off**.

Visualize Train/Test Errors

```
plt.plot(degrees, train_errors, marker='o', label='Train Error')
plt.plot(degrees, test_errors, marker='o', label='Test Error')
plt.title("Train vs Test Error for Different Polynomial Degrees")
plt.xlabel("Polynomial Degree")
plt.ylabel("Mean Squared Error")
plt.legend()
```

Expected Output:

- U-shaped curve for test error.
- Train error decreases with complexity.
- Test error decreases initially (bias reduction), then increases (variance inflation).

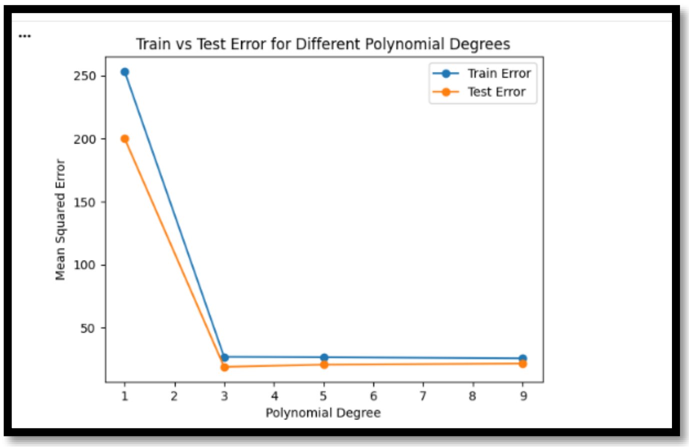


Figure 4: Train vs Test MSE vs Polynomial Degree.

Visualize Predicted Curves

```
for d in degrees:
    poly = PolynomialFeatures(degree=d)
    X_poly_fit = poly.fit_transform(X_fit)
    model = LinearRegression().fit(poly.fit_transform(X_train), y_train)
    y_fit = model.predict(X_poly_fit)
    plt.plot(X_fit, y_fit, label=f'Degree {d}')

plt.scatter(X_train, y_train, color='red', label='Train Data')
plt.scatter(X_test, y_test, color='blue', label='Test Data')
plt.title("Polynomial Fits for Different Degrees")
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.show()
```

Expected Output:

- Visual comparison of under-fitting vs over-fitting.
- Low-degree models under-fit, high-degree models over-fit, medium-degree models go optimal.

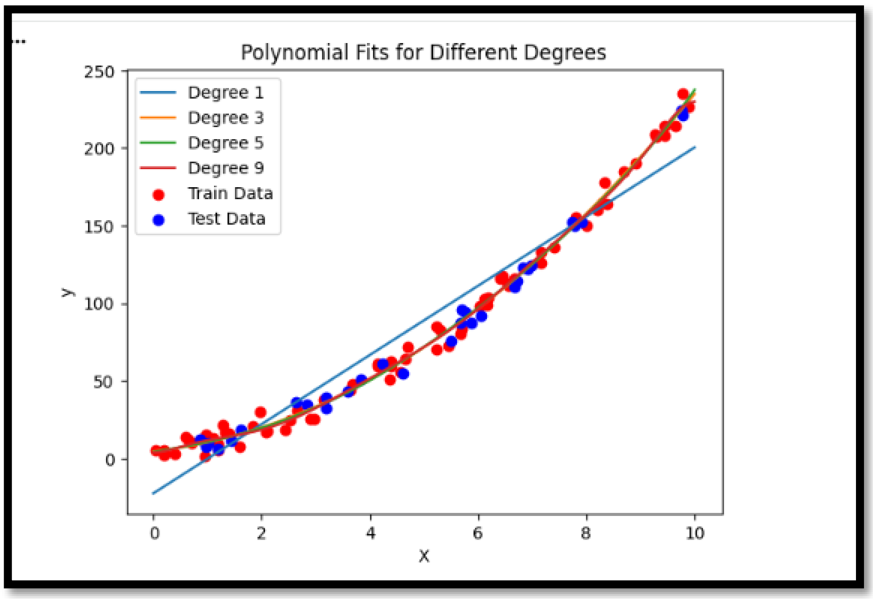


Figure 5: Polynomial fits of different degrees.

Reproducibility

- Set random seeds (np.random.seed) and fixed train/test splits.
- Ensures results can be **replicated consistently**.

5. Results and Discussion

Our experiments demonstrate the expected bias–variance behaviour. With the synthetic quadratic dataset, the linear model (degree 1) under-fits the data due to its inability to capture the curvature, resulting in **high bias** and large training and test MSE. Increasing the model complexity to degree 3 reduces errors on both datasets, while the best test performance is achieved at an **intermediate degree** (degree 5). In contrast, a high-degree model (degree 9) over-fits the training data, producing near-zero training error but higher test error, which indicates **high**

variance. Overall, the training MSE decreases with complexity, whereas the test MSE follows a **U-shaped trend**

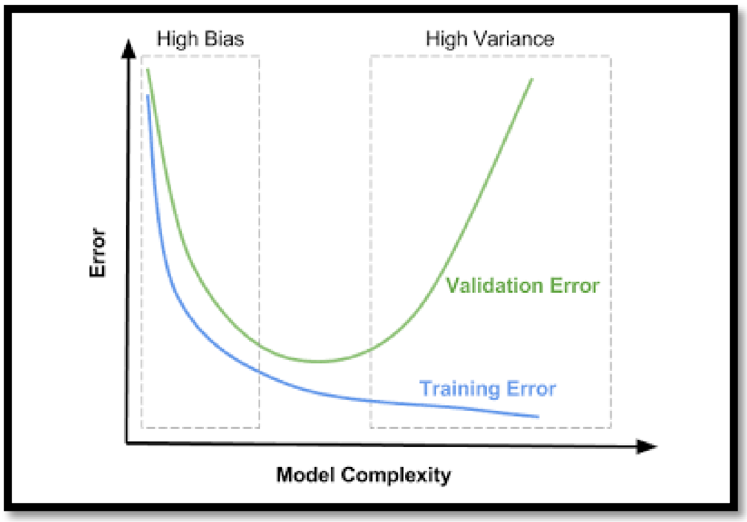


Figure 6: Mean Squared Error vs Model Complexity (Polynomial Degree)

Figure 6 shows the typical U-curve of test error and monotonically decreasing train error. It clearly illustrates the bias–variance trade-off, low complexity has high bias (both errors high), high complexity has high variance (gap between train/test errors large), and the sweet spot is in between.

Qualitatively, these results align with the theoretical expectations and published examples *(Christopher Burns, et al., 2012)* , *(Fangfang Lee, 2025)* .Initially, increasing model complexity reduces bias faster than it increases variance, so total error falls. Beyond a certain point, variance dominates and error rises. The optimal complexity is where the model generalizes best, seen as the minimum of the test error curve. In practice, one would choose that complexity (or corresponding regularization) to minimize generalization error.

5.1 Diagnosis:

Learning curves reveal bias–variance issues. At a fixed model complexity, a high-bias model (e.g., degree 1) shows high training and validation errors that converge with more data, indicating under-fitting. A **high-variance** model (e.g., degree 9) has **low training error** but **high validation error**, showing over-fitting. In our experiments, degree 9 behaves this way, while degree 1 shows high error on both sets. *(Jason Brownlee PhD, 2025)*

5.2 Model selection:

To choose a model, one can plot a validation curve (score vs complexity) and pick the complexity at minimum test error, or use cross-validation: train on multiple folds and average performance. High variance shows as high variability in scores, high bias as uniformly poor scores. Cross-validation gives a more reliable estimate of generalization error. *(Jason Brownlee PhD, 2025), (Wickramasinghe, 2024)*.

Practical advice:

- High bias (**under-fitting**): increase model complexity, add features, use a more flexible model (e.g., higher-degree polynomial, random forest), or reduce regularization. Feature engineering or nonlinear transformations can also help capture underlying patterns.
- High variance (**over-fitting**): simplify the model or add regularization. Options include more data, L1/L2 penalties, dropout, pruning trees, or ensembles (random forests, boosting). For high-degree polynomials, use ridge regression or limit the degree to reduce variance
- **Strategy:** start simple, gradually increase complexity, monitor **validation error**, and use **cross-validation** to ensure robust performance.

5.3 Ethical dimension:

The bias–variance trade-off affects both model reliability and ethics. A high-bias (under-fitted) model may systematically make errors that reduce fairness or safety. For instance, a simple regression predicting loan defaults might underestimate risk for certain groups, leading to unfair outcomes. In healthcare, under-fitted models can miss patterns in patient data, causing missed diagnoses *(Aliferis, Constantin; , Gyorgy J. Simon, 2024)*. In critical domains, practitioners must ensure models are neither too rigid nor too sensitive through rigorous validation and domain-specific checks to maintain high generalization performance.

5.4 Limitations of results:

Our experiments focused on regression with one feature for clarity. In higher dimensions, the same principles apply but plots become multi-variate. Also, we assumed that error is measured by MSE, other loss metrics behave similarly in terms of bias/variance decomposition. We also did not account for potential *algorithmic biases* (e.g. fairness biases), here “bias” is strictly the statistical bias of the predictor, not demographic bias. Finally, our synthetic example had known true function, with real data, one must rely on empirical validation.

6 Conclusion

The bias–variance trade-off is fundamental for understanding and improving machine learning models. Reducing overall error requires balancing bias and variance. Simple models under-fit (high bias), while complex models over-fit (high variance) *(Fangfang Lee, 2025)* .In our polynomial regression experiments, training error decreased while validation error followed a U-shape as complexity increased (*Figure 2).* Learning curves and cross-validation help identify whether a model suffers from high bias or variance *(Fangfang Lee, 2025)* , *(Jason Brownlee PhD, 2025)*.

In practice, model complexity should match the data and task. For under-fitting, enrich the model or features. For over-fitting, simplify, regularize, or gather more data. In high-risk domains (healthcare, finance), the cost of bias may outweigh variance, motivating more complex models and careful validation (Aliferis, Constantin; , Gyorgy J. Simon, 2024). With limited data, simpler models may be preferred to avoid high variance.

Understanding bias–variance decomposition guides iterative model selection and tuning. As Jason Brownlee notes, every choice of model, regularization, or feature set navigates this trade-off *(Jason Brownlee PhD, 2025), (Wickramasinghe, 2024)*. Systematic experiments and validation curves help students recognize the right balance and build models that generalize. Future work could extend this to classification tasks and more complex datasets.

7 References

1. Hastie, T., Tibshirani, R., & Friedman, J. (2009) *The Elements of Statistical Learning*. Section on bias–variance tradeoff. Available at: https://www.statlearning.com (Accessed: 2 December 2025).
2. IBM Developer (2022) 'Think: What is Bias–Variance Tradeoff?', *IBM Blog*. Available at: https://www.ibm.com/blogs (Accessed: 3 December 2025).
3. Mukherjee, P. (2021) 'Bias–Variance Tradeoff in Machine Learning: Concepts & Tutorials', *BMC Blogs*. Available at: https://www.bmc.com/blogs/bias-variance-machine-learning/ (Accessed: 4 December 2025).
4. Scipy Lecture Notes (n.d.) 'Bias and variance of polynomial fit'. Available at: https://scipy-lectures.org (Accessed: 5 December 2025).
5. Aliferis, C. & Simon, G. (2024) 'Overfitting, Underfitting and General Model Overconfidence', in *AI/ML in Healthcare*. Springer. Available at: https://www.ncbi.nlm.nih.gov/books (Accessed: 2 December 2025).
6. Brownlee, J. (2025) 'The Bias-Variance Trade-Off: A Visual Explainer', *Machine Learning Mastery*. Available at: https://machinelearningmastery.com (Accessed: 3 December 2025).
7. GeeksforGeeks (2022) 'Bias-Variance Trade-Off'. Available at: https://www.geeksforgeeks.org (Accessed: 4 December 2025).
8. Scikit-learn (n.d.) 'Validation curves and learning curves'. Available at: https://scikit-learn.org (Accessed: 5 December 2025).
9. Wikipedia (2024). *Bias–variance tradeoff*. [Online]. Available at: https://en.wikipedia.org/wiki/Bias–variance_tradeoff (Accessed: 5 December 2025). en.wikipedia.orgen.wikipedia.org.

Appendix

Project Source Code: https://github.com/MAdnanSajid/bias-variance-ML