Understanding Linear Regression Internally: A Guided Learning Experience

Introduction

This learning experience is designed to help learners understand how Linear Regression works internally by building the algorithm from scratch without relying on machine learning libraries such as sklearn or statsmodels.

Through four structured modules (represented as videos), learners will move from raw data to a fully trained regression model optimized using gradient descent.

The goal is not just to use Linear Regression — but to understand how it learns.

🎥 Video 1 – Data Loading and Preprocessing

Introduction

Before any model can learn, the data must be prepared. Machine learning models operate entirely on numerical matrices, not raw tables or text-based categories.

This module focuses on transforming raw housing data into structured numerical arrays suitable for mathematical optimization.

Key Learning Outcomes

- Load CSV data using pandas
- Convert categorical variables using one-hot encoding
- Perform train-test split
- Prevent information leakage
- Standardize features

Important Concept: Information Leakage

Scaling was performed after train-test splitting to prevent test data statistics from influencing training.

This ensures honest evaluation.

Summary

By the end of this module:

- Features are numeric
- Data is split properly
- Inputs are standardized
- The dataset is ready for model training

🎥 Video 2 – The Linear Regression Model

Introduction

Linear Regression models the relationship between features and target using a linear equation:

$$y^{\wedge}=XW+b$$

Where:

- X → feature matrix
- W → weights (feature importance)
- b → bias (intercept)

Conceptual Insight

Each weight represents how much a feature influences house price.

The model computes predictions using a weighted sum of features.

Summary

At this stage, we defined how predictions are computed — but the weights are not yet optimized.

🎥 Video 3 – Training with Gradient Descent

Introduction

To make the model useful, we must adjust weights to minimize prediction error.

This module introduces:

- Mean Squared Error (MSE)
- Gradient computation
- Learning rate
- Iterative parameter updates

Why MSE?

MSE penalizes larger errors more heavily due to squaring, making it sensitive to large deviations.

Gradient Descent Intuition

Gradient descent updates parameters in the opposite direction of the gradient to reduce loss.

Loss decreasing across epochs indicates successful learning.

Practical Insight

Exploding loss (inf/nan) demonstrated:

- Importance of scaling
- Sensitivity to learning rate
- Numerical stability considerations

Summary

The model gradually improved its predictions by minimizing MSE through iterative updates.

🎥 Video 4 – Testing and Generalization

Introduction

After training, the model must be evaluated on unseen data.

This ensures that the model generalizes beyond training samples.

Key Concepts

- Test MSE
- Generalization
- Overfitting vs underfitting
- Interpretation of large-scale loss values

Summary

Evaluation confirmed that the model learned meaningful patterns from the housing dataset.

Conclusion

By implementing Linear Regression from scratch, learners gain:

- Mathematical understanding
- Algorithmic intuition
- Practical debugging experience
- Insight into optimization dynamics

This approach transforms Linear Regression from a black-box tool into a transparent learning system.

10 Assessment Questions

1. Why must categorical variables be converted into numeric format before training a linear regression model?

A) To reduce dataset size
B) Because mathematical models operate only on numerical inputs
C) To automatically improve model accuracy
D) To eliminate multicollinearity

Correct Answer: B

Feedback:

- A ❌ Encoding does not reduce dataset size.
- B ✅ Linear regression performs mathematical operations that require numeric input.
- C ❌ Encoding alone does not guarantee improved accuracy.
- D ❌ Multicollinearity is a separate statistical issues.

---

2. Which steps help prevent information leakage?

A) Splitting the dataset before scaling
B) Using training set statistics for scaling test data
C) Scaling the entire dataset before splitting
D) Evaluating model only on training data

Correct Answers: A, B

Feedback:

- A ✅ Splitting first prevents test data influencing training.
- B ✅ Using training mean and std ensures fair evaluation.
- C ❌ Scaling before splitting leaks test data information.
- D ❌ Evaluating only on training data hides generalization issues

---

3. What does the bias term (b) represent in the equation ŷ = XW + b?

A) Feature importance
B) Regularization strength
C) The intercept of the regression line
D) The model error

Correct Answer: C

Feedback:

- A ❌ Feature importance is represented by weights.
- B ❌ Regularization is a separate concept.
- C ✅ Bias shifts the regression line vertically (intercept).
- D ❌ Error is measured using the loss function.

---

4. Why can gradient descent produce inf or nan values during training?

A) Learning rate is too large
B) Gradients explode
C) Features are properly scaled
D) Target values are very large

Correct Answers: A, B, D

Feedback:

- A ✅ Large learning rate causes unstable updates.
- B ✅ Large gradients lead to overflow.
- C ❌ Proper scaling actually prevents instability.
- D ✅ Large target values can produce large gradients.

---

5. What is minimized during training in this implementation?

A) Accuracy
B) Mean Absolute Error
C) Mean Squared Error
D) Variance

Correct Answer: C

Feedback:

- A ❌ Accuracy is for classification tasks.
- B ❌ MAE is another metric but not used here.
- C ✅ MSE is explicitly defined and minimized.
- D ❌ Variance is not directly minimized.

---

6. What happens if the learning rate is too small?

A) Model diverges
B) Training becomes very slow
C) Model overfits
D) Gradients vanish

Correct Answer: B

Feedback:

- A ❌ Divergence usually happens when learning rate is too large.
- B ✅ Small learning rate leads to slow convergence.
- C ❌ Overfitting relates to model complexity.
- D ❌ Vanishing gradients occur in deep networks, not simple regression.

---

7. Why is feature scaling important in gradient descent?

A) It improves numerical stability
B) It prevents exploding gradients
C) It removes the need for bias
D) It ensures features contribute proportionally

Correct Answers: A, B, D

Feedback:

- A ✅ Scaling stabilizes calculations.
- B ✅ It reduces risk of large gradient updates.
- C ❌ Bias is still required.
- D ✅ Scaling prevents one feature dominating due to magnitude.

---

8. What does the dot product np.dot(X, weights) compute?

A) The mean of features
B) A weighted combination of input features
C) The gradient
D) The variance

Correct Answer: B

Feedback:

- A ❌ Mean is not computed here.
- B ✅ Dot product computes weighted sum.
- C ❌ Gradient is computed separately.
- D ❌ Variance is unrelated.

---

9. Why do we evaluate the model on test data?

A) To adjust the weights further
B) To measure generalization performance
C) To improve training accuracy
D) To normalize predictions

Correct Answer: B

Feedback:

- A ❌ We do not update weights using test data.
- B ✅ Test data measures unseen performance.
- C ❌ Training accuracy is measured on training data.
- D ❌ Normalization is separate from evaluation

---

10. Which assumptions are inherent in linear regression?

A) Linear relationship between features and target
B) Numerical input features
C) Categorical output variable
D) Continuous target variable

Correct Answers: A, B, D

Feedback:

- A ✅ Linear regression assumes linearity.
- B ✅ Inputs must be numeric.
- C ❌ Categorical outputs require classification models.
- D ✅ Linear regression predicts continuous values.

Document By :

Amoolya M Shanbhag

1MP22AD004@gmail.com

AI&DS, BGSCET