Understanding Linear Regression from Scratch

1. Overview

This learning document accompanies a Jupyter Notebook that implements Linear Regression completely from scratch using only basic Python, NumPy, and Pandas.No high-level machine learning libraries such as *sklearn* or *statsmodels* are used.

The objective of this learning experience is to help learners understand how linear regression works internally, including data preparation, mathematical formulation, training using gradient descent, and model evaluation. This content is designed for a mixed audience of technical and non-technical learners who are familiar with Python programming, data analysis, and basic machine learning concepts.

2. Learning Path and Video Sequence

The Jupyter Notebook is structured into four major parts. Each part is assumed to be a separately recorded video, and learners are encouraged to follow the videos in the given sequence.

- **Video 1:** Data Loading and Preprocessing
- **Video 2:** Linear Regression Model
- **Video 3:** Training Algorithm (Gradient Descent)
- **Video 4:** Testing and Evaluation

Each video builds on the concepts introduced in the previous one.

3. Video-wise Learning Content

Video 1: Data Loading and Preprocessing

Introduction

In this video, learners are introduced to the dataset and the essential preprocessing steps required before training a machine learning model. Proper data preparation ensures that the model learns meaningful patterns and produces reliable results.

What is Covered

- Loading the housing prices dataset using Pandas
- Inspecting the dataset structure, data types, and summary statistics
- Identifying numerical and categorical features
- Converting categorical variables into numerical form using one-hot encoding
- Performing a manual train–test split without using machine learning libraries
- Applying feature scaling using standardization

- Preventing information leakage by computing scaling parameters only from training data

Summary

By the end of this video, learners understand how raw data is transformed into a clean, numerical format suitable for machine learning and why preprocessing decisions significantly impact model performance.

Video 2: Linear Regression Model

Introduction

This video focuses on the core concept of linear regression and how predictions are generated using a mathematical model.

What is Covered

- The linear regression equation:

$$\hat{y} = Xw + b$$

- Meaning of weights (coefficients) and bias (intercept)
- How multiple input features contribute to a single prediction
- Implementing a prediction function using vectorized operations

Summary

Learners gain a clear understanding of how input features are combined linearly to produce predicted outputs and how model parameters influence predictions.

Video 3: Training Algorithm (Gradient Descent)

Introduction

In this video, learners explore how a linear regression model learns from data by minimizing prediction error through optimization.

What is Covered

- Definition and purpose of a loss function
- Mean Squared Error (MSE) as a measure of prediction error
- Concept of gradient descent
- Role of learning rate and number of epochs
- Iterative updating of weights and bias to minimize loss

Summary

By the end of this video, learners understand how gradient descent enables the model to learn optimal parameters and why training is an iterative process rather than a one-step computation.

Video 4: Testing and Evaluation

Introduction

This video emphasizes the importance of evaluating a trained model on unseen data to measure real-world performance.

What is Covered

- Using the test dataset for evaluation
- Generating predictions on unseen data
- Calculating test Mean Squared Error (MSE)
- Understanding generalization and model performance

Summary

Learners see how evaluation validates whether the model has learned meaningful patterns rather than memorizing training data.

Assessment Questions

Q1. Why must categorical variables be converted into numerical values?

A. To reduce dataset size ✗

B. Because machine learning models require numerical inputs ✓

C. To increase accuracy ✗

D. To remove noise ✗

Correct Answer: B

Feedback: Machine learning algorithms perform mathematical operations and cannot process text values directly.

Q2. Which of the following are numerical features in the dataset? (Multiple correct)

A. Area ✓

B. Bedrooms ✓

C. Mainroad ✗

D. Parking ✓

Correct Answers: A, B, D

Feedback: Numerical features are those represented as numbers and can be directly used in mathematical computations by the model.

Q3. Why is train–test splitting performed before feature scaling?

A. To increase model accuracy ✗

B. To reduce computation time ✗

C. To avoid information leakage ✓

D. To balance the dataset ✗

Correct Answer: C

Feedback: Performing scaling before splitting would allow information from the test set to influence the training process.

Q4. What does the bias term represent in a linear regression model?

A. The error in prediction ✗

B. The intercept of the regression line ✓

C. The importance of a feature ✗

D. Random noise ✗

Correct Answer: B

Feedback: The bias represents the point where the regression line intersects the target axis when all feature values are zero.

Q5. What does Mean Squared Error (MSE) measure?

A. Classification accuracy ✗

B. Average squared difference between actual and predicted values ✓

C. Model complexity ✗

D. Data variance ✗

Correct Answer: B

Feedback: MSE quantifies how far predictions are from actual values by averaging the squared errors.

Q6. Why is feature scaling important for gradient descent?

A. It reduces dataset size ✗

B. It removes outliers ✗

C. It ensures all features contribute equally to learning ✓

D. It prevents overfitting ✗

Correct Answer: C

Feedback: Gradient descent is sensitive to feature scales, and scaling prevents features with large values from dominating the updates.

Q7. What happens if the learning rate is too high?

A. Faster and stable convergence ✗

B. No learning occurs ✗

C. The model may diverge instead of converging ✓

D. The model becomes perfectly accurate ✗

Correct Answer: C

Feedback: A very high learning rate can cause parameter updates to overshoot the minimum loss.

Q8. Gradient descent updates model parameters to minimize which quantity?

A. Accuracy ✗

B. Loss function ✓

C. Feature variance ✗

D. Dataset size ✗

Correct Answer: B

Feedback: Gradient descent iteratively adjusts parameters to minimize the defined loss function.

Q9. Which dataset should be used to evaluate model performance on unseen data?

A. Training dataset ✗

B. Validation dataset ✗

C. Test dataset ✓

D. Entire dataset ✗

Correct Answer: C

Feedback: The test dataset represents unseen data and provides an unbiased estimate of model performance.

Q10. Why is the Mean Squared Error value relatively large in this model?

A. The model implementation is incorrect ✗

B. House prices are large and errors are squared ✓

C. Feature scaling was skipped ✗

D. Gradient descent failed ✗

Correct Answer: B

Feedback: Since house prices are large numerical values, squaring the prediction errors results in a large MSE value.
