Module: Understanding Linear Regression through a Prediction Problem

Welcome to the module focused on Linear Regression! This is one of the most fundamental and widely used supervised learning algorithms, particularly for tackling **regression problems** where the goal is to predict a continuous numerical outcome.

Structure of this Module

This module will guide you through the core concepts and practical application of Linear Regression. We will cover the following topics:

1. **Introduction to Linear Regression** *(Current Section)*
2. Understanding Cost Functions
3. Linear Regression Assumptions
4. Problem Statement (Project)
5. Data Pre-processing (Project)
6. Building our LR Model (Project)
7. Understanding Model Performance (Project)
8. Model Tuning methods (Project)

We will use a hands-on prediction project to illustrate these concepts.

Introduction to Linear Regression

Linear Regression models the relationship between a dependent (target) variable and one or more independent (predictor) variables by fitting a linear equation to the observed data.

Simple Linear Regression (SLR)

The simplest form is Simple Linear Regression, which involves only **one independent variable (x)** and **one dependent variable (y)**. You might recall the equation of a straight line from mathematics:

**y = mx + c**

Where:

* **y**: The dependent variable (the value we want to predict).
* **x**: The independent variable (the feature used to make the prediction).
* **m**: The **slope** of the line. It represents the change in y for a one-unit change in x.
* **c**: The **y-intercept**. It represents the value of y when x is zero.

*(Consider inserting the y=mx+c graph image here)*

In the context of machine learning, the goal of a Simple Linear Regression algorithm is to find the optimal values for the slope (m) and the intercept (c) that best fit the given data points (x, y).

Key Terminology

Let's define some common terms used in Linear Regression, often using Greek letters (like Beta - β) for coefficients in statistical contexts:

* **Predictors / Features / Independent Variables (X):** These are the input variables (x in SLR, or x₁, x₂, ..., x<0xE2><0x82><0x99> in Multiple LR) used by the model to predict the target variable.
* **Target Variable / Dependent Variable (y):** This is the continuous output variable that the model aims to predict.
* **Beta Coefficients (β) / Parameters / Weights:** These are the values the model learns from the data. They define the line (or hyperplane in higher dimensions) that best fits the data.
  + **Intercept (β₀ or c):** The predicted value of y when all predictors are zero.
  + **Slope Coefficients (β₁, β₂, ..., β<0xE2><0x82><0x99> or m):** Represent the change in the target variable y associated with a one-unit change in the corresponding predictor variable (xᵢ), holding other predictors constant.
* **Making Predictions (ŷ):** Once the model learns the optimal coefficients (βs), it can predict the target variable for new, unseen data points by plugging their feature values (X) into the learned equation. The predicted value is often denoted as ŷ (y-hat).

Multiple Linear Regression (MLR)

Often, the target variable depends on more than one predictor. Multiple Linear Regression extends SLR to handle **two or more independent variables**. The equation becomes:

**y = β₀ + β₁x₁ + β₂x₂ + β₃x₃ + ... + β<0xE2><0x82><0x99>x<0xE2><0x82><0x99>**

Where:

* y is the target variable.
* x₁, x₂, ..., x<0xE2><0x82><0x99> are the n different predictor variables.
* β₀ is the intercept.
* β₁, β₂, ..., β<0xE2><0x82><0x99> are the coefficients for each corresponding predictor variable.

The core idea remains the same: the algorithm's goal is to find the set of Beta coefficients (β₀, β₁, ..., β<0xE2><0x82><0x99>) that results in the best fit to the training data, minimizing the difference between the predicted values (ŷ) and the actual values (y). How we measure this difference and minimize it is the topic of our next section: Cost Functions.

Selection deleted