CODE BASICS

MACHINE LEARNING TAKEAWAYS

Chapter:

Supervised Machine Learning: Regression

Linear Regression Single Variable

1. Simple Linear Regression —> y = mx + b
    - Slope (m)
    - Intercept (b)

2. Linear regression helps to establish a relationship between dependent variables and independent variables.

3. Gradient Descent is the most important concept in the world of Supervised Machine Learning, which will be covered in the coming lectures.

Linear Regression Multiple Variables

1. Multiple Linear Regression —>
   y = m1 * x1 + m2 * x2 + ... + b
   - m1, m2 – coefficients
   - x1, x2 – Independent variables
   - **b –** Intercept

2. When you run the model.predict() function, it employs the formula mentioned earlier to forecast values based on the data it was trained on.

Cost Function

1. Understanding the cost function is essential for understanding Gradient Descent

2. Gradient Descent is the most important concept in the world of Supervised Machine Learning

3. Error / Loss – Difference between the predicted and actual Y value.

4. Mean Absolute Error (MAE) – The average of errors, disregarding their direction. It's the average of absolute differences between prediction and actual observation.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Yi - \hat{Y}i|$$

Cost Function

5 Mean Squared Error (MSE) – It is the average of squared differences between predicted and actual observations. It effectively highlights larger errors.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

6 MSE serves as the cost function for Linear Regression.

Derivatives, Partial Derivatives

1. Slope of a line at a given point is Derivative
2. Derivative -> X^n = n X^ (n-1)
3. Slope is used for Linear equations, whereas Derivative is used for non-linear equations.
4. Slope is constant, whereas Derivative is a function.
5. The purpose of a Partial Derivative is to measure how a function changes as one of its variables is varied while keeping the other variables constant.

Chain Rule

1. Chain rule is a technique used to compute the derivative of a function, composed of multiple functions.

2. Chain rule will be used in the Gradient Descent Technique.

Gradient Descent: Theory

1. Gradient Descent is an optimization method used in linear regression to find the best-fit line by iteratively adjusting the slope (m) and intercept (b) to minimize the cost function, usually the mean squared error.

2. Since testing every combination for MSE is impractical, Gradient Descent efficiently minimizes MSE with fewer iterative adjustments.

$$\text{Mean Squared Error (MSE)} = \frac{1}{n}\sum_{i=1}^{n}(y_i - (mx_i + b))^2$$

$$\partial/\partial m = \frac{2}{n}\sum_{i=1}^{n} - x_i\,(y_i - (mx_i + b))$$

$$\partial/\partial m = \frac{2}{n}\sum_{i=1}^{n} - (y_i - (mx_i + b))$$

Gradient Descent: Python Implementation

1. In your role as a Data Scientist or AI Engineer, your daily tasks will not typically involve implementing Gradient Descent. Instead, you'll use ML libraries. However, a solid understanding of this concept is beneficial for both your work and potential interviews.

2. Adjusting the learning rate and epochs based on observed outputs (m, b) will enable you to obtain the desired outcomes in the Gradient Descent Implementation.

Why MSE and not MAE?

1. Mean Squared Error (MSE) is our go-to for calculating Gradient Descent because:
    - It's sensitive to outliers
    - It's continuously differentiable

2. In rare scenarios, Mean Absolute Error (MAE) is our pick for Gradient Descent when we're dealing with lots of outliers.

3. Even though these principles are basic, in real-world scenarios, you'll be using machine learning libraries directly.

Model Evaluation: Train, Test Split

1. Just as obtaining a driver's license involves passing a test, not just learning to drive, machine learning models require splitting the dataset into training and testing parts and evaluating the model's precision.

2. We utilize the train_test_split() function from the sklearn library for this purpose.

Model Evaluation: Metrics

1. To evaluate the performance of a Machine Learning model, we can use metrics such as MSE, MAE, or R2 score.

2. The R2 score is easier to interpret, compared to other metrics.

3. The parameter 'random_state' is used in the train_test_split() function to ensure reproducibility.

Model Evaluation: Metrics

1. To evaluate the performance of a Machine Learning model, we can use metrics such as MSE, MAE, or R2 score.

2. The R2 score is easier to interpret, compared to other metrics.

3. The parameter 'random_state' is used in the train_test_split() function to ensure reproducibility.

Data Preprocessing: One Hot Encoding

1. In simple terms, computers understand numbers, not text. The process of turning text into numbers is known as encoding.

2. We use label encoding for ordinal categories that have a specific order.

3. For nominal categories, which don't have an order, we use one-hot encoding.

4. One-hot encoding transforms categorical data into a binary vector format that's easier for machine learning models to comprehend. Each category is represented by a binary vector, with a '1' at the point corresponding to the category, and '0's everywhere else.

Data Preprocessing: One Hot Encoding

5 Multicollinearity is a situation where two or more independent variables are closely linked, making it hard to differentiate their separate effects.

6 To prevent multicollinearity, we remove one of the columns after/during the one-hot encoding.

7 The Pandas library includes a built-in function named get_dummies() for implementing one-hot encoding. By using the drop_first parameter, we can eliminate the first dummy column.

Polynomial Regression

1. Simple linear regression models a straight-line ($y = b_o + b_1x$) relationship between a dependent variable y and an independent variable x. Polynomial regression extends this by including higher powers of x for complex, non-linear relationships.

   - $y = b_o + b_1x + b_2x^2 + b_3x^3 + ... + b_nx^n$
   - $b_o, b_1, ...b_n$ – coefficients
   - n = degree

2. Polynomial Regression with degree=1 is nothing but a Linear Regression

3. Deciding the degree will be based on trial and error, as well as domain knowledge.

Overfitting and Underfitting

1. **Overfitting:** Occurs when a model learns too much detail and noise from the training data, affecting its performance on new data.

2. **Underfitting:** Happens when a model is too simple and can't learn the data pattern, leading to poor performance on all data.

3. **Balanced Fit:** This is achieved when a model accurately learns the training data's patterns and performs well on unseen data.

Reasons and Remedies For Overfitting / Underfitting

1 Overfitting can happen due to any one or combination of the following points:

- Reason: Poor model, hyperparameters selection
- Solution: Better model, hyperparameters selection

- Reason: Insufficient training data
- Solution: Sufficient training data

- Reason: Poor feature selection
- Solution: Careful feature selection

- Reason: Inadequate validation
- Solution: Adequate validation

- Reason: Lack of regularization
- Solution: Apply regularization

Reasons and Remedies For Overfitting / Underfitting

2 Underfitting can happen due to any one or combination of the following points:

- Reason: Too simple model
- Solution: Use a complex model that can capture data patterns

- Reason: Insufficient training data
- Solution: Sufficient training data

- Reason: Insufficient features / Poor feature engineering
- Solution: Better feature selection/ engineering

- Reason: Insufficient training time
- Solution: Sufficient training time

- Reason: Inadequate validation
- Solution: Adequate validation

- Reason: Excessive regularization
- Solution: Adequate regularization.

L1 and L2 Regularization

1. L1 and L2 regularization are effective tools for minimizing overfitting.

2. When L2 regularization is applied to Linear Regression, it transforms into Ridge Regression.

3. In the same vein, L1 Regularization leads to what we commonly call Lasso Regression.

4. Linear Regression too can encounter overfitting issues if the number of features is excessive.

5. The choice between Ridge or Lasso Regression and parameters like Alpha is contingent on multiple factors and is typically determined through a process of trial and error.

Bias Variance Trade Off

1. Bias occurs when an algorithm misses significant patterns in the data due to its simplicity, while Variance occurs when an algorithm changes significantly based on minor differences in the training data.

2. BUVO: Bias–Underfitting, Variance–Overfitting