|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Generative AI Consortium (Ltd)**  **AI/ML Internship: Assignment 1 (Simple Machine Learning Problem) Assignment)**  **Name: Arun Karthik S** | | **Email: mailto:arun261104@gmail.com** | | | | | | |  |
| **House ID** | **Square Feet** | **Number of Bedrooms** | **Age of House (Years)** | **Price (Label)** |
| 1 | 1500 | 3 | 10 | 300,000 |
| 2 | 2000 | 4 | 15 | 400,000 |
| 3 | 850 | 2 | 5 | 150,000 |
| 4 | 1200 | 3 | 20 | 220,000 |
| 5 | 1750 | 4 | 12 | 360,000 |
| 6 | 1300 | 3 | 8 | 280,000 |

**Feature**: These are the input variables used to predict the output. In our example, "Square Feet," "Number of Bedrooms," and "Age of House" are features.

**Label**: This is the output variable that we want to predict. In our example, "Price" is the label.

**Prediction**: This is the output that our model generates based on the input features. For example, given a house with 1600 square feet, 3 bedrooms, and 10 years old, the model might predict a price of $310,000.

**Outlier**: These are data points that are significantly different from others. For instance, if a house with 1500 square feet and 3 bedrooms was priced at $600,000, it might be considered an outlier.

**Test Data**: This is the subset of data used to evaluate the performance of the model. It is not used during the training process. In our example, rows 4 and 5 could be our test data.

**Training Data**: This is the subset of data used to train the model. In our example, rows 1, 2, 3, and 6 could be our training data.

**Model**: This is the mathematical representation (e.g., linear regression, decision tree) that learns from the training data and makes predictions on new data.

**Validation Data**: This is a subset of data used to tune the model parameters. It helps in assessing the model performance while training. For example, rows 3 and 6 could be our validation data.

**Hyperparameter**: These are parameters set before training the model and are not learned from the data. Examples include the learning rate, number of epochs, and regularization strength.

**Epoch**: One complete pass through the entire training dataset. If we train our model for 10 epochs, it means the model has seen the entire training dataset 10 times.

**Loss Function**: This is a function that measures the difference between the predicted value and the actual value. For example, Mean Squared Error (MSE) is a common loss function used for regression tasks.

**Learning Rate**: The learning rate is a hyperparameter that controls how much the model's weights are adjusted in response to the estimated error each time the model weights are updated.

 With a **small learning rate**, you take small steps, ensuring you don't overshoot the bottom, but it will take you longer to get there.

 With a **large learning rate**, you take big steps, reaching the bottom faster, but you risk overshooting and oscillating around the bottom or even moving away from it.

**Overfitting:** This occurs when the model learns the training data too well, including the noise, and performs poorly on new data. If our model predicts the training data with 100% accuracy but fails on test data, it is overfitting.

**Underfitting:** Underfitting occurs when a machine learning model is too simple to capture the underlying patterns in the data. As a result, the model performs poorly on both the training data and new, unseen data.

**Regularization**: This is a technique used to prevent overfitting by adding a penalty to the loss function for large coefficients. Examples include L1 and L2 regularization.

**Cross-validation:** This is a technique used to assess the model's performance by partitioning the data into multiple subsets, training the model on some subsets, and validating it on the remaining ones. A common method is k-fold cross-validation.

**Feature Engineering:** This is the process of creating new features or modifying existing ones to improve model performance. For instance, we might create a new feature "Price per Square Foot."

**Dimensional Reduction:** This is the process of reducing the number of features while retaining important information. Techniques include Principal Component Analysis (PCA) and t-SNE.

**Bias**: Bias is the error introduced by approximating a real-world problem, which might be complex, by a simplified model.

 **High Bias**:

* Occurs when a model is too simple and cannot capture the underlying patterns in the data.
* Results in high errors on both the training data and the test data.
* Leads to **underfitting**.

 **Low Bias**:

* Indicates that the model is more complex and can capture the underlying patterns in the data more accurately.
* Reduces errors on the training data, though not necessarily on the test data if the model overfits.

**Variance**: Variance is the error introduced by the model's sensitivity to small fluctuations in the training set.

* **High Variance**:
  + Occurs when a model is too complex and captures noise or random fluctuations in the training data.
  + Results in low errors on the training data but high errors on the test data.
  + Leads to **overfitting**.
* **Low Variance**:
  + Indicates that the model is less sensitive to the specific details of the training data and can generalize better to unseen data.
  + Reduces the risk of overfitting, though the model might underfit if it is too simple.