HYPOTHESIS IN MACHINE LEARNING

A hypothesis in machine learning is the model's presumption regarding the connection between the input features and the result. It is an illustration of the mapping function that the algorithm is attempting to discover using the training set. To minimize the discrepancy between the expected and actual outputs, the learning process involves modifying the weights that parameterize the hypothesis. The objective is to optimize the model's parameters to achieve the best predictive performance on new, unseen data, and a cost function is used to assess the hypothesis' accuracy.

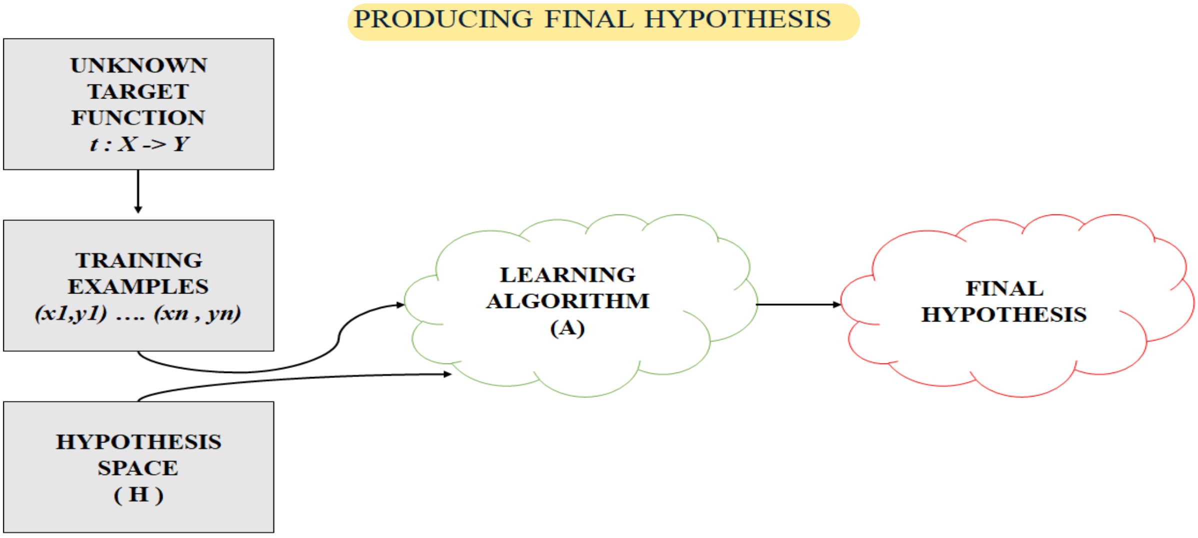
What is Hypothesis Testing?

Researchers must consider the possibility that their findings could have happened accidentally before interpreting them. The systematic process of determining whether the findings of a study validate a specific theory that pertains to a population is known as hypothesis testing.To assess a hypothesis about a population, hypothesis testing is done using sample data. A hypothesis test evaluates the degree of unusualness of the result, determines whether it is a reasonable chance variation, or determines whether the result is too extreme to be attributed to chance.

How does a Hypothesis work?

In most supervised machine learning algorithms, our main goal is to find a possible hypothesis from the hypothesis space that could map out the inputs to the proper outputs.

The following figure shows the common method to find out the possible hypothesis from the Hypothesis space:



Hypothesis Space (H)

Hypothesis space is the set of all the possible legal hypothesis. This is the set from which the machine learning algorithm would determine the best possible (only one) which would best describe the target function or the outputs.

Hypothesis (h)

A hypothesis is a function that best describes the target in supervised machine learning. The hypothesis that an algorithm would come up depends upon the data and also depends upon the restrictions and bias that we have imposed on the data.

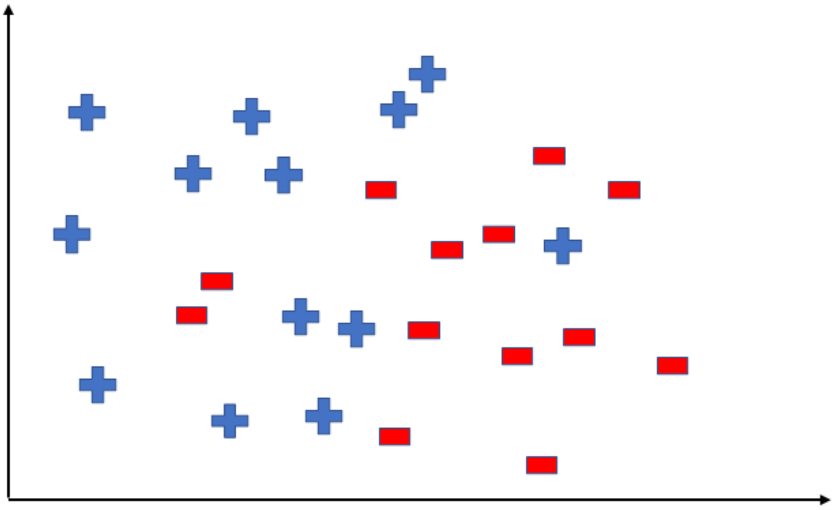
The Hypothesis can be calculated as:

$y = mx + b$

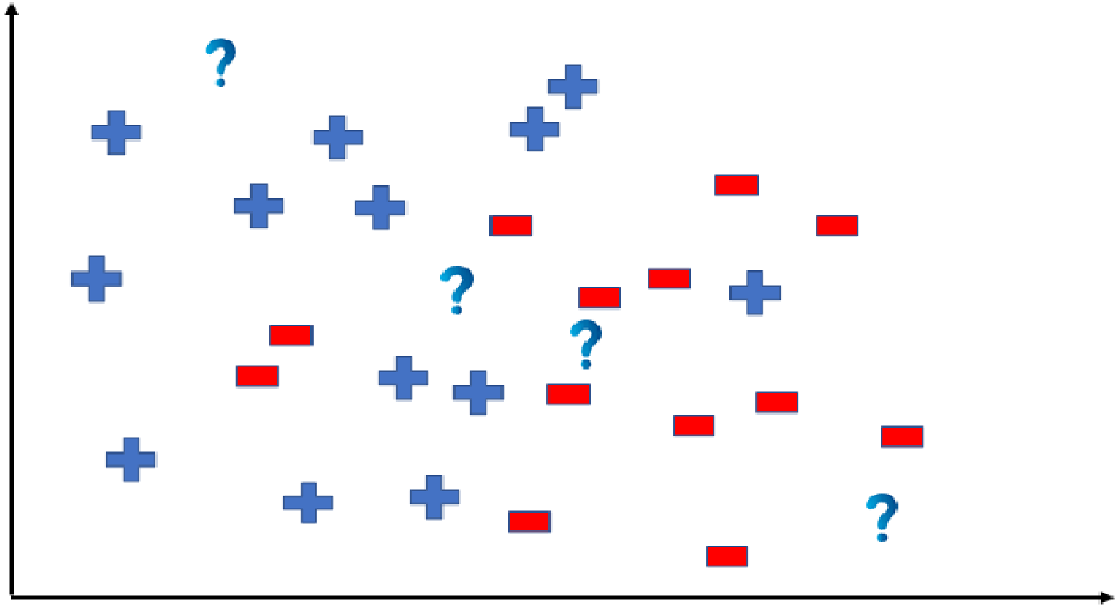
Where,

- y = range
- m = slope of the lines
- x = domain
- b = intercept

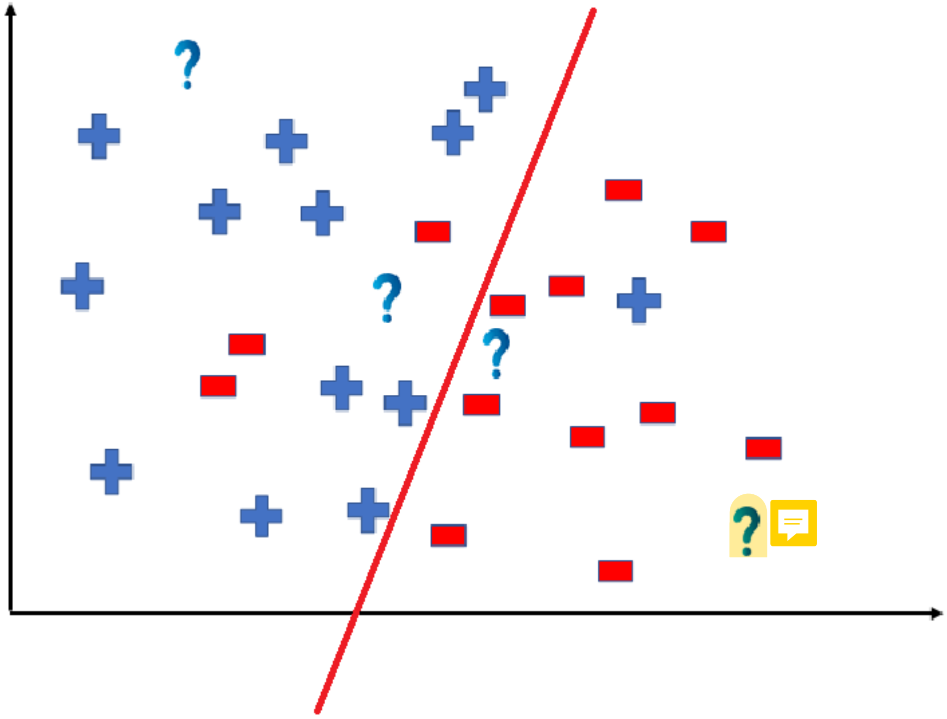
To better understand the Hypothesis Space and Hypothesis consider the following coordinate that shows the distribution of some data:



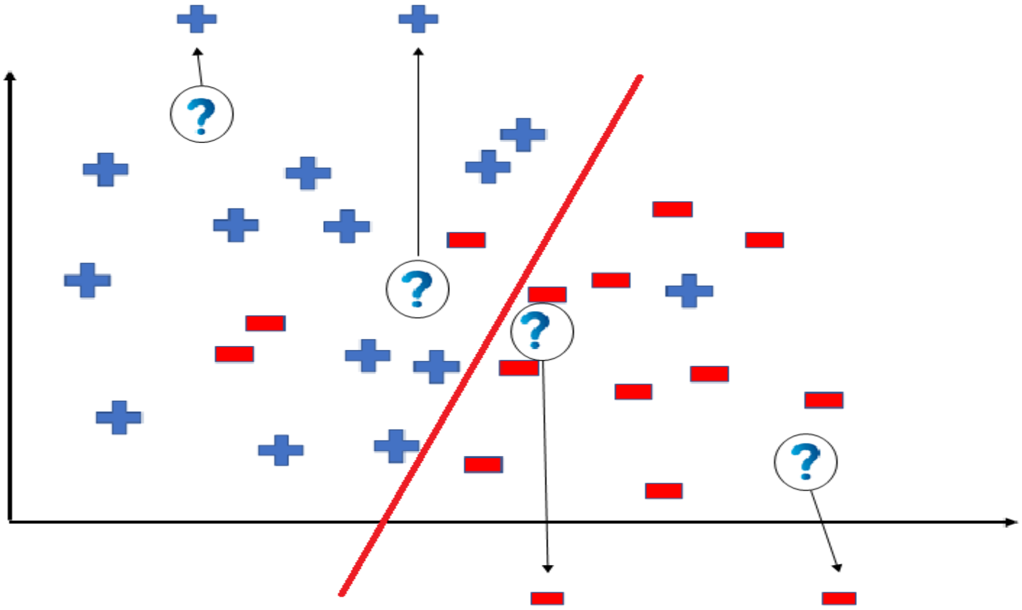
Say suppose we have test data for which we have to determine the outputs or results. The test data is as shown below:



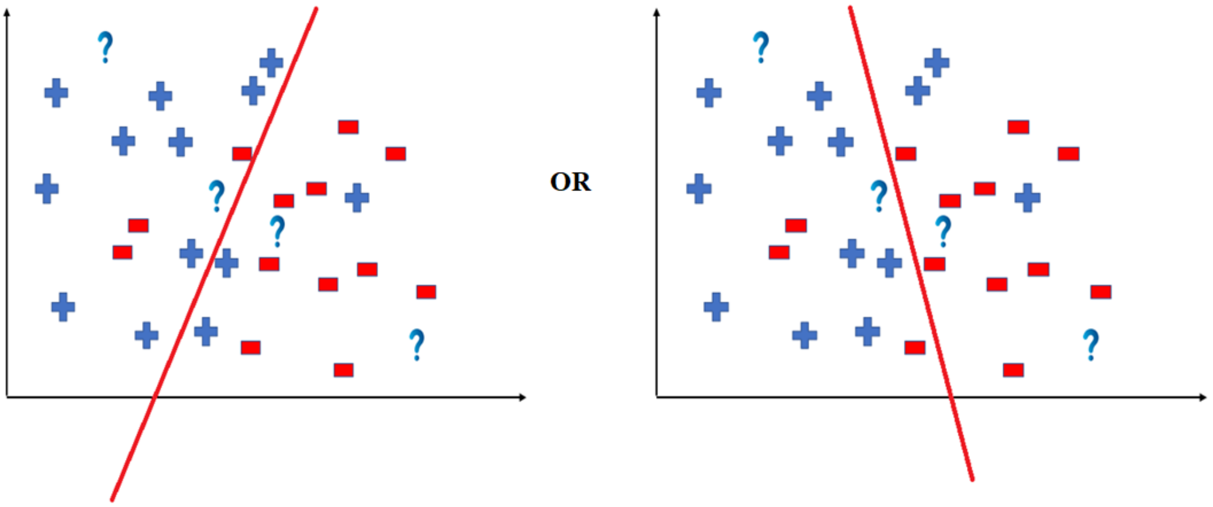
We can predict the outcomes by dividing the coordinate as shown below:



So the test data would yield the following result:



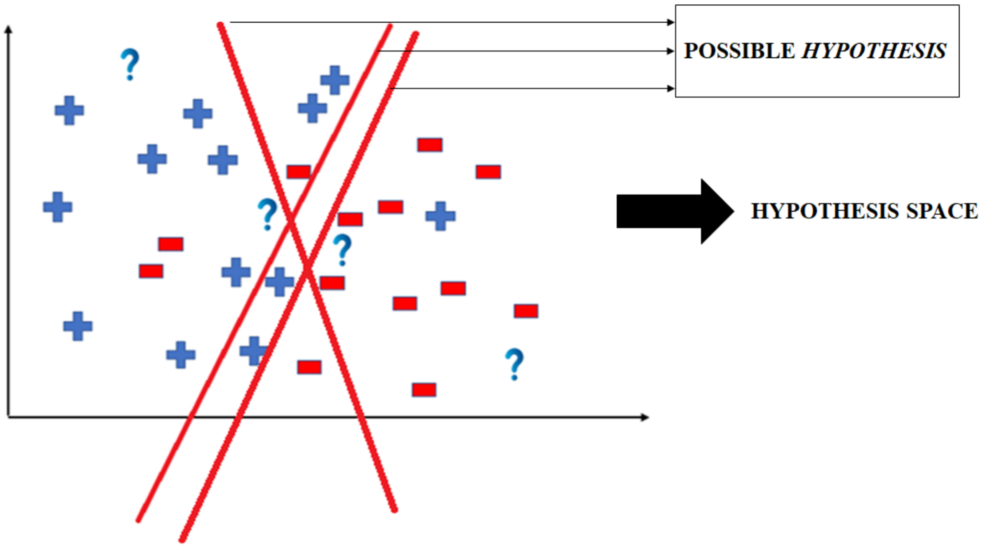
But note here that we could have divided the coordinate plane as:



The way in which the coordinate would be divided depends on the data, algorithm and constraints.

- All these legal possible ways in which we can divide the coordinate plane to predict the outcome of the test data composes of the Hypothesis Space.
- Each individual possible way is known as the hypothesis.

Hence, in this example the hypothesis space would be like:



Hypothesis in Statistics

In statistics, a hypothesis refers to a statement or assumption about a population parameter. It is a proposition or educated guess that helps guide statistical analyses. There are two types of hypotheses: the null hypothesis (H0) and the alternative hypothesis (H1 or Ha).

- **Null Hypothesis(H0):** This hypothesis suggests that there is no significant difference or effect, and any observed results are due to chance. It often represents the status quo or a baseline assumption.
- **Aternative Hypothesis(H1 or Ha):** This hypothesis contradicts the null hypothesis, proposing that there is a significant difference or effect in the population. It is what researchers aim to support with evidence.

INDUCTIVE BIAS

Definition

At its core, inductive bias refers to the set of assumptions that a learning algorithm makes to predict outputs for inputs it has never seen. It's the bias or inclination of a model towards making a particular kind of assumption in order to generalize from its training data to unseen situations.

Why is Inductive Bias Important?

Learning from Limited Data: In real-world scenarios, it's practically impossible to have training data for every possible input. Inductive bias helps models generalize to unseen data based on the assumptions they carry.

Guiding Learning: Given a dataset, there can be countless hypotheses that fit the data. Inductive bias helps the algorithm choose one plausible hypothesis over another.

Preventing Overfitting: A model with no bias or assumptions might fit the training data perfectly, capturing every minute detail, including noise. This is known as overfitting. An inductive bias can prevent a model from overfitting by making it favour simpler hypotheses.

Types of Inductive Bias

Preference Bias: It expresses a preference for some hypotheses over others. For example, in decision tree algorithms like ID3, the preference is for shorter trees over longer trees.

Restriction Bias: It restricts the set of hypotheses considered by the algorithm. For instance, a linear regression algorithm restricts its hypothesis to linear relationships between variables.

Examples of Inductive Bias in Common Algorithms

Decision Trees: Decision tree algorithms, like ID3 or C4.5, have a bias towards shorter trees and splits that categorize the data most distinctly at each level.

k-Nearest Neighbors (k-NN): The algorithm assumes that instances that are close to each other in the feature space have similar outputs.

Neural Networks: They have a bias towards smooth functions. The architecture itself (number of layers, number of neurons) can also impose bias.

Linear Regression: Assumes a linear relationship between the input features and the output.

Trade-offs

While inductive bias helps models generalize from training data, there's a trade-off. A strong inductive bias means the model might not be flexible enough to capture all patterns in the data. On the other hand, too weak a bias could lead the model to overfit the training data.

Generalization in Machine Learning

Have you ever noticed that your model false predictions over your testing data? Even though you have trained your model with enough data still you get false negatives or false positives for your test data. Why is that?

Either your model is underfitting or overfitting to your training data. Generalization is a measure of how your model performs on predicting unseen data. So, it is important to come up with the best-generalized model to give better performance against future data. Let us first understand what is underfitting and overfitting, and then see what are the best practices to train a generalized model.

A: Underfitting, B: Generalized, C: Overfitting

What is Underfitting?

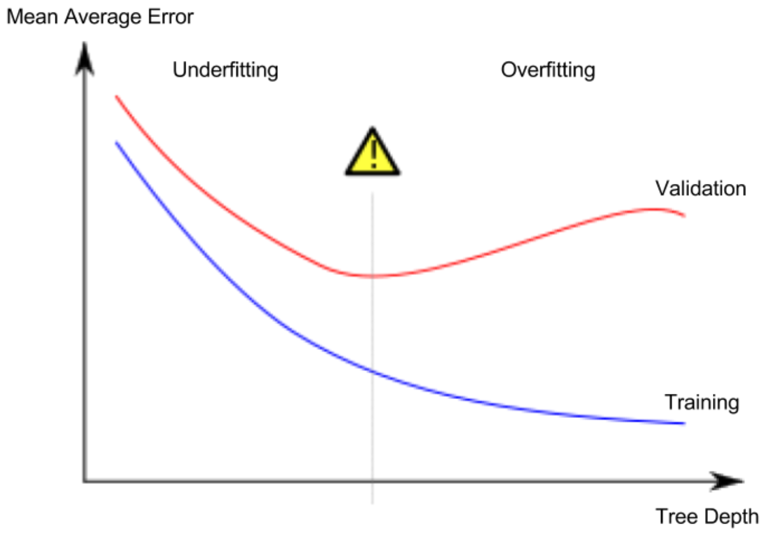
Underfitting is a state where the model cannot model itself on the training data. And also not able to generalize new data. You can notice it with the help of loss function during your training. A simple rule of thumb is if both training loss and cross-validation loss are high, then your model is underfitting.

Lack of data, not enough features, lack of variance in training data or high regularization rate can cause underfitting. A simple solution is to add more shuffled data to your training. Depending on what causes underfitting to your model, you can try introducing more meaningful features, feature crossing and introducing higher order polynomials as features or reducing regularization rate if you are using regularization. In some cases trying out with different training algorithm will work fine.

What is Overfitting?

Overfitting is a situation where your model force learns the whole variance. Experts say it as model starts to memorize all the noise instead of learning. A simple rule of thumb to identify the overfitting is if your training loss is low and cross-validation loss is high then your model is overfitting.

Uncleaned data, fewer steps in training, higher complexity of the model (due to higher weights in data) can cause overfitting. It is always recommended to preprocess data and create a good data pipeline. Select only necessary and meaningful features with good variance. Reduce the complexity of the model using good regularization algorithm (L1 norm or L2 norm).



Comparison

What are the best practices to get a Generalized model?

It is important to have a training dataset with good variance (i.e. a shuffled data set). The Best way to do this is computing the hash for an appropriate feature and split data into training, evaluation and test sets based on the computed hash value. Here the evaluation set is used to cross-validate the trained model. It is always good to ensure that the distribution in all the dataset is stationary(same).

Handling outliers also important, it always depends on the task you are working around. If you are training the model to detect anomalies you should consider outliers, in such case, these anomalies may be the labels you need to identify. So you cannot classify or detect without outliers. On the other hand, if you are modeling a regression-based classification it is good to remove outliers.

Using resampling during the training. Resampling enables you to reconstruct your sample dataset in different ways for each iteration. One of the most popular resampling technique is k-fold cross-validation. It does training and testing on the model for k times with different subsets of your testing data.

It is always good to know when to stop training. It is a common human insight to determine when to stop training. When you reach a good training loss and a good validation loss at that point stop training.

Learn to do some feature engineering when needed. In some cases, your model cannot be able to converge, there may be not a meaningful relation found on the raw features you have. Doing Feature crosses and introducing new features with meaningful relation helps the model to converge.

In addition to these parameters tunings, Hyper parameter tunings, using regularization algorithms also helps to generalize the model for better performance.

Hope you all get a basic idea of generalization, underfitting, and overfitting. Use this as a base and keep exploring on subtopics for deeper understandings.

Don’t forget to applaud if you find this article useful. Your doubts and feedbacks are always welcomed.