Lecture 01 Notes [16/08/2019] **Machine Learning basics**

[Ashish Verma 15MI423]

What is Machine Learning?

Machine learning (**ML**) is the study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.

Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task

Types of learning algorithms

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

Supervised Learning:

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and a desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs.

Supervised learning algorithms include classification and regression.

Classification: Classification algorithms are used when the outputs are restricted to a limited set of values

Regression: Regression algorithms are used when the outputs may have any numerical value within a range.

Unsupervised Learning:

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms therefore learn from test data that has not been labeled, classified or categorized.

Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data.

Reinforcement Learning:

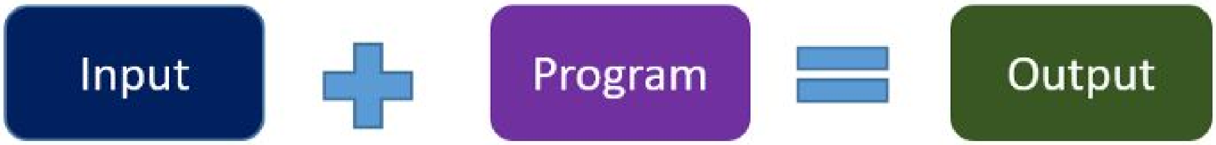
Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.

Applications:

Handwriting Recognition –convert written letters into digital letters
Language Translation –translate spoken and or written languages (e.g. Google Translate)
Speech Recognition –convert voice snippets to text (e.g. Siri, Cortana, and Alexa)
Image Classification –label images with appropriate categories (e.g. Google Photos)
Autonomous Driving –enable cars to drive

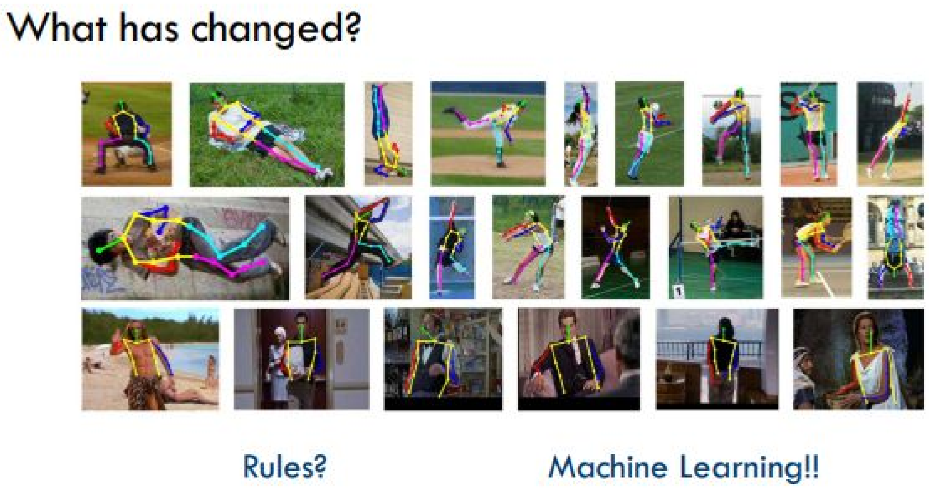
How Machine Learning is different from Traditional Logical Programming?

Traditional programming is a manual process — meaning a person (programmer) creates the program. But without anyone programming the logic, one has to manually formulate or code rules. We have the input data, and someone (programmer) coded a program that uses that data and runs on a computer to produce the desired output.



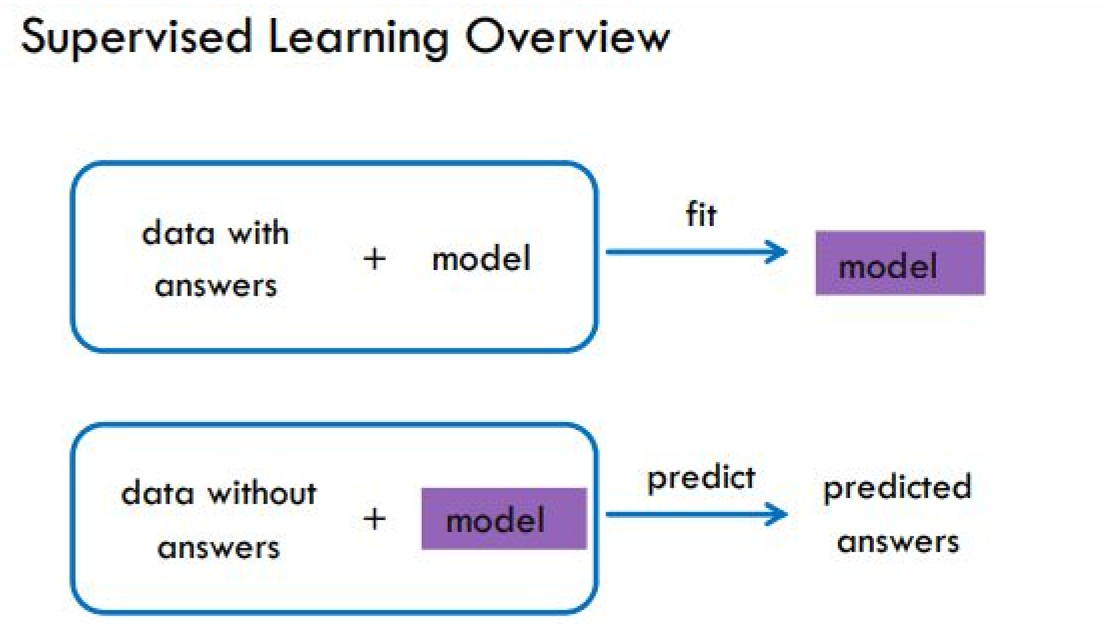
Machine Learning, on the other hand, the input data and output are fed to an algorithm to create a program.In Traditional programming one has to manually formulate/code rules while in Machine Learning the algorithms automatically formulate the rules from the data.

For example look at following image

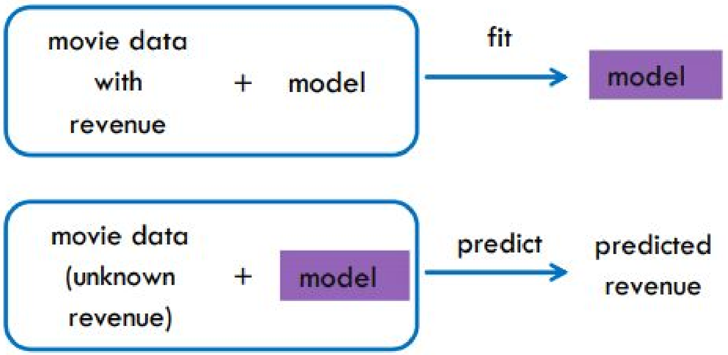


Now if we have to write a program to detect the posture of human then we have have to code the posture manually and there a huge lot of positions of libs possible so it is not possible to code each orientation so we take machine learning approach in which we just give the point mark to the joints as the training data and train the model on lot of samples and then later use this model for the inference of posture of an image which was not included in training sample.

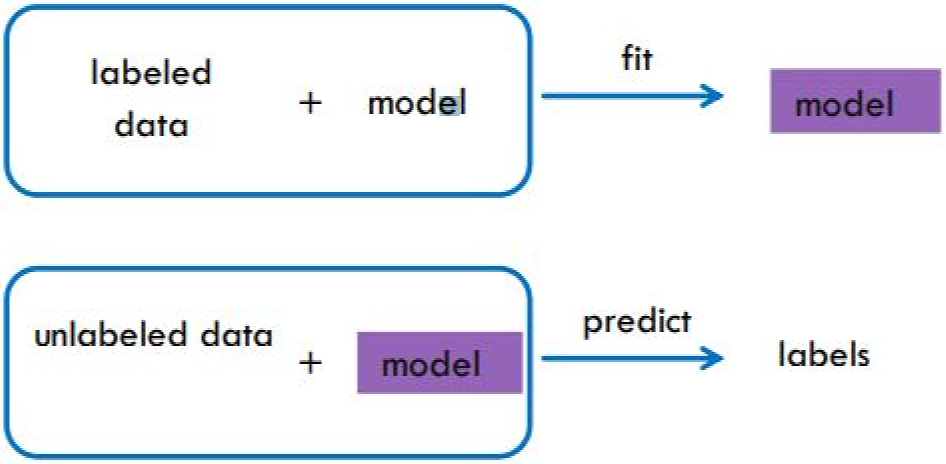
Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.



Regression analysis is a set of statistical processes for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed.



Classification however give non numerical answer. In this the sample data or the training data is provided with a string label. So when the same kind of data is fed into trained model the output is corresponding label.

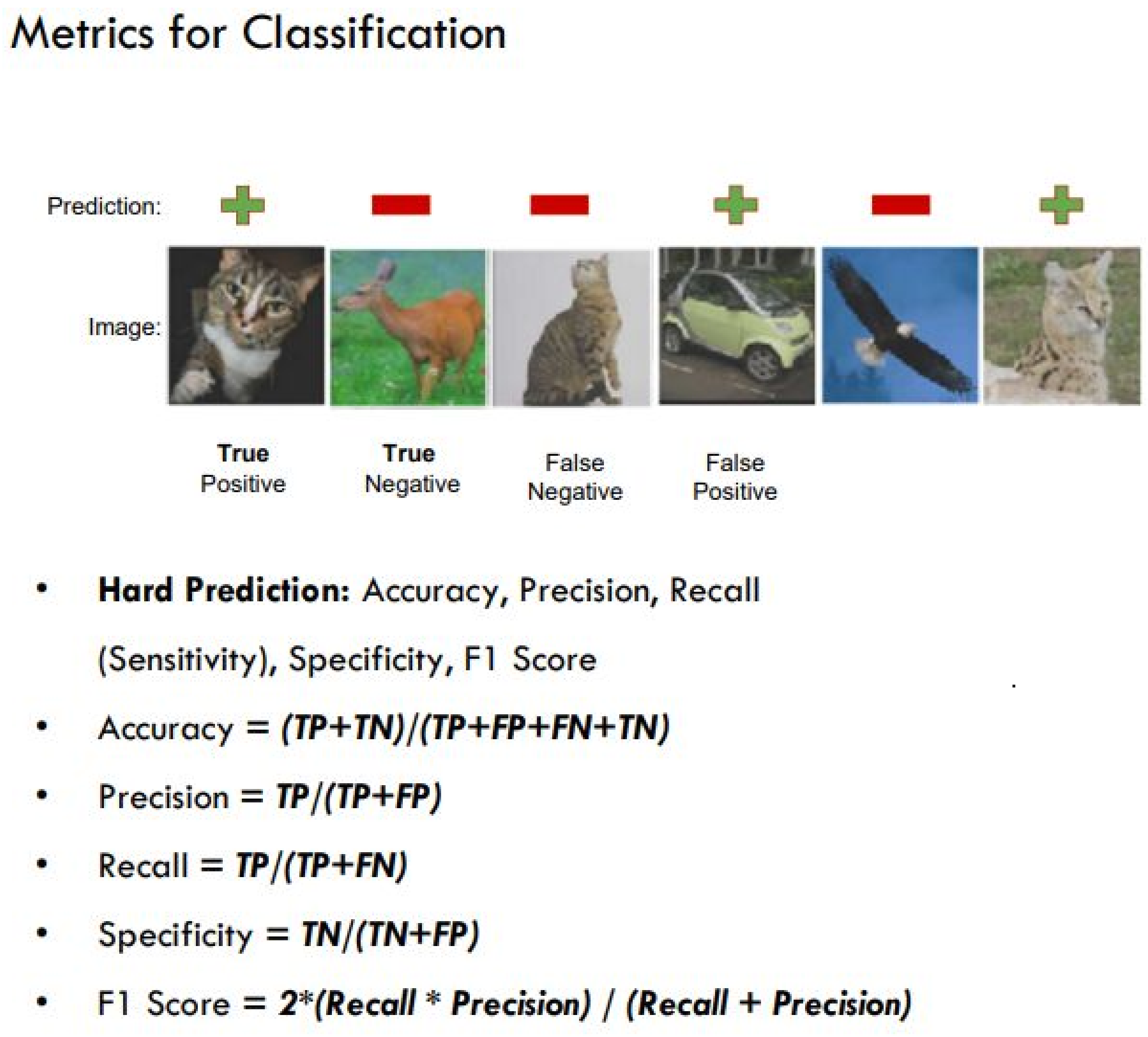


Three Types of Classification Predictions

Hard Prediction: Predict a single category for each instance. So the output is the category or the label associated with the input data.Here previously some kind of predefined threshold can be considered to make hard decision
Parameters for Hard decisions: Accuracy, Precision, Recall (Sensitivity), Specificity, F1 Score

Ranking Prediction: Rank the instances from most likely to least likely. (binary classification)
Parameters:AUC (ROC), Precision-Recall Curves

Probability Prediction: Assign a probability distribution across the classes to each instance.So the each class or the category the model is trained is assigned a probability and top most are chosen to be shown at output.
Parameters: Log-loss (aka CrossEntropy), Brier Score



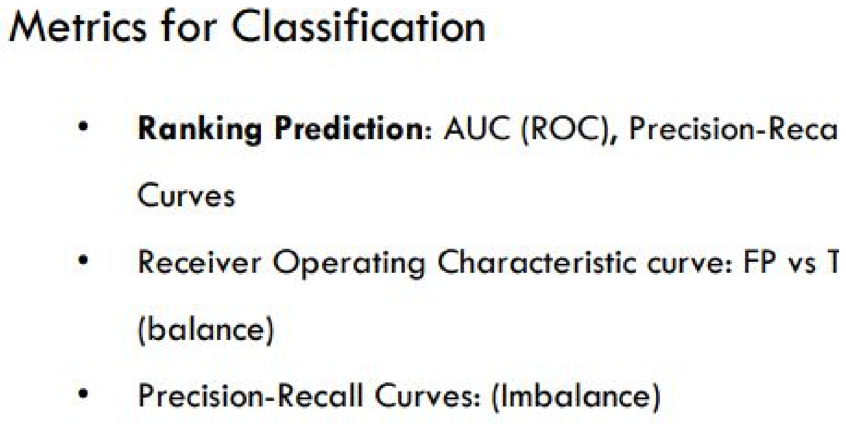
Metrics for Classification

- **Hard Prediction:** Accuracy, Precision, Recall (Sensitivity), Specificity, F1 Score
- Accuracy = ***(TP+TN)/(TP+FP+FN+TN)***
- Precision = ***TP/(TP+FP)***
- Recall = ***TP/(TP+FN)***
- Specificity = ***TN/(TN+FP)***
- F1 Score = ***2*(Recall * Precision) / (Recall + Precision)***

In the above example we are finding the cat.
TP: True Positive : if the image is of cat and the model predicted in positive i.e. it predicted cat.
TN: the 2nd image is of deer not cat and the output of the model is negative(corresponding to not cat) .So prediction right implied :True and the output is negative.
FN: This condition happens when the the image is of cat but the model predicted wrong and output is negative

The any of the above i.e. accuracy, reacall , specificity can be used as a parameter for thresholding and hard decisioning

Recall:
Recall is a measure that tells us what proportion of patients that actually had cancer was diagnosed by the algorithm as having cancer. The actual positives (People having cancer are TP and FN) and the people diagnosed by the model having a cancer are TP. (Note: FN is included because the Person actually had a cancer even though the model predicted otherwise).



Metrics for Classification

- **Ranking Prediction**: AUC (ROC), Precision-Reca Curves
- Receiver Operating Characteristic curve: FP vs T (balance)
- Precision-Recall Curves: (Imbalance)

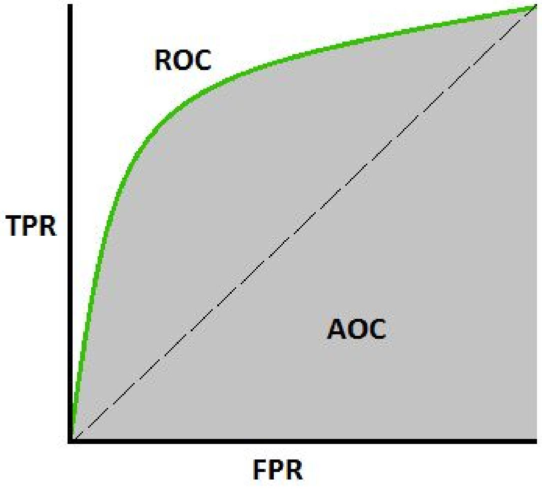
Precision Recall Curve:

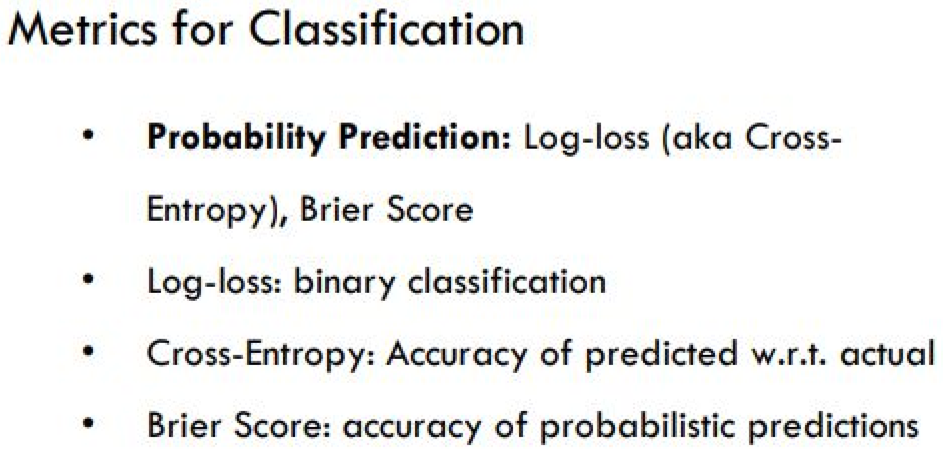
Precision-Recall is a useful measure of success of prediction when the classes are very imbalanced. In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned.

The precision-recall curve shows the tradeoff between precision and recall for different threshold. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

What is AUC - ROC Curve?

AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease.The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.

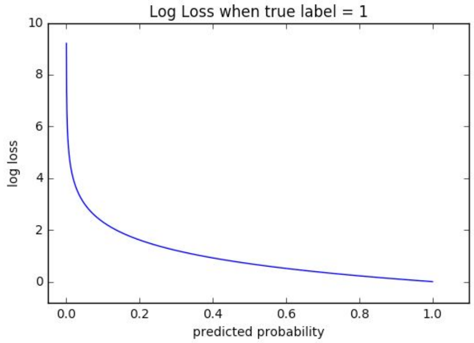




Metrics for Classification

- **Probability Prediction:** Log-loss (aka Cross-Entropy), Brier Score
- Log-loss: binary classification
- Cross-Entropy: Accuracy of predicted w.r.t. actual
- Brier Score: accuracy of probabilistic predictions

Log Loss(Cross Entropy): Logarithmic loss (related to cross-entropy) measures the performance of a classification model where the prediction input is a probability value between 0 and 1. The goal of our machine learning models is to minimize this value. A perfect model would have a log loss of 0. Log loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high log loss.



The **Brier score** is a proper score function that measures the accuracy of probabilistic predictions. It is applicable to tasks in which predictions must assign probabilities to a set of mutually exclusive discrete outcomes.

Metrics for Regression: RMSE (Root Mean Square Error)

It represents the sample standard deviation of the differences between predicted values and observed values (called residuals). Mathematically, it is calculated using this formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

MAE

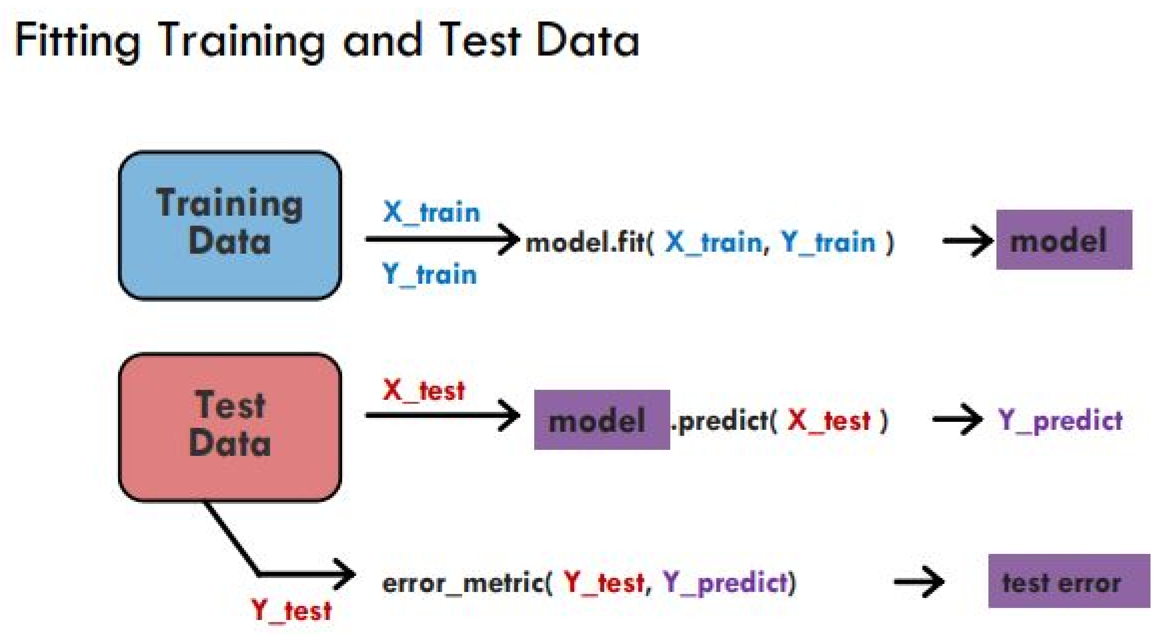
MAE is the average of the absolute difference between the predicted values and observed value. The MAE is a linear score which means that all the individual differences are weighted equally in the average. For example, the difference between 10 and 0 will be twice the difference between 5 and 0. However, same is not true for RMSE which we will discuss more in details further. Mathematically, it is calculated using this formula:

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

How Training and Inference Takes place is shown in following image: The two data parameter the input and associated output label is fed. The X-train is kind of input signal and the Y-train is associated class or label. These both are fed into a model and the model is trained and used for inference.

For inference the x-train is fed and its corresponding label is predicted.

While in training phases the output is compared with what output should be and the error is found . Which is used as feed back and algorithms like root mean square are used to minimise the error.



Fitting Training and Test Data

Training Data

X_train

Y_train

model.fit(X_train, Y_train) → model

Test Data

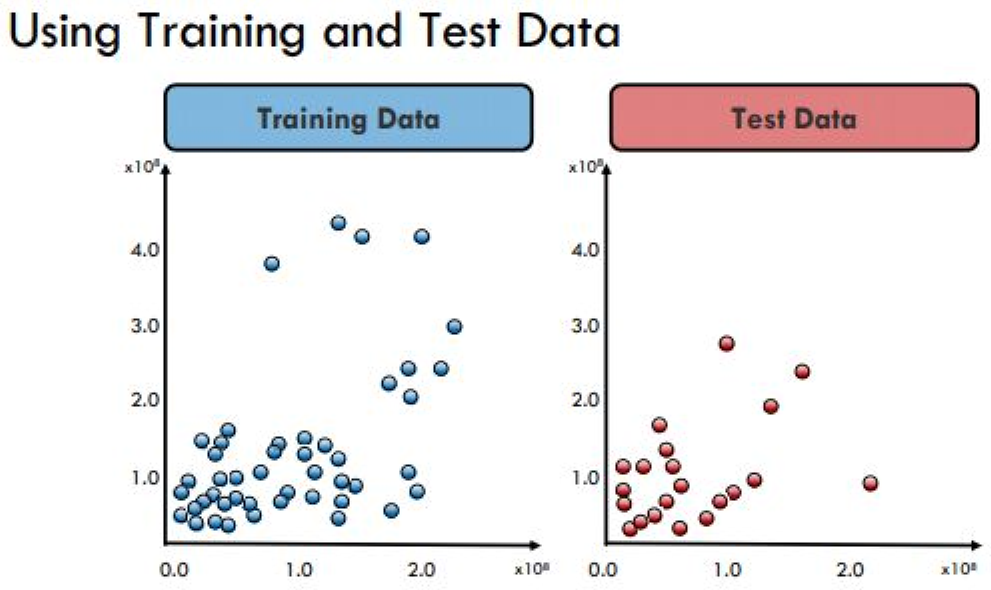
X_test

model .predict(X_test) → Y_predict

Y_test

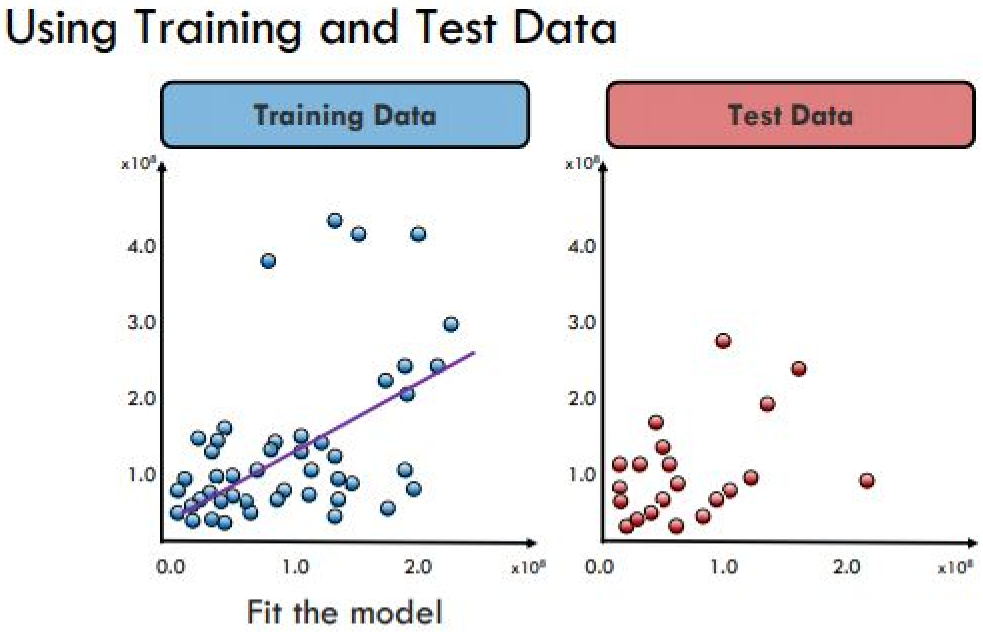
error_metric(Y_test, Y_predict) → test error

Initial plot of training data and Test data



Using Training and Test Data

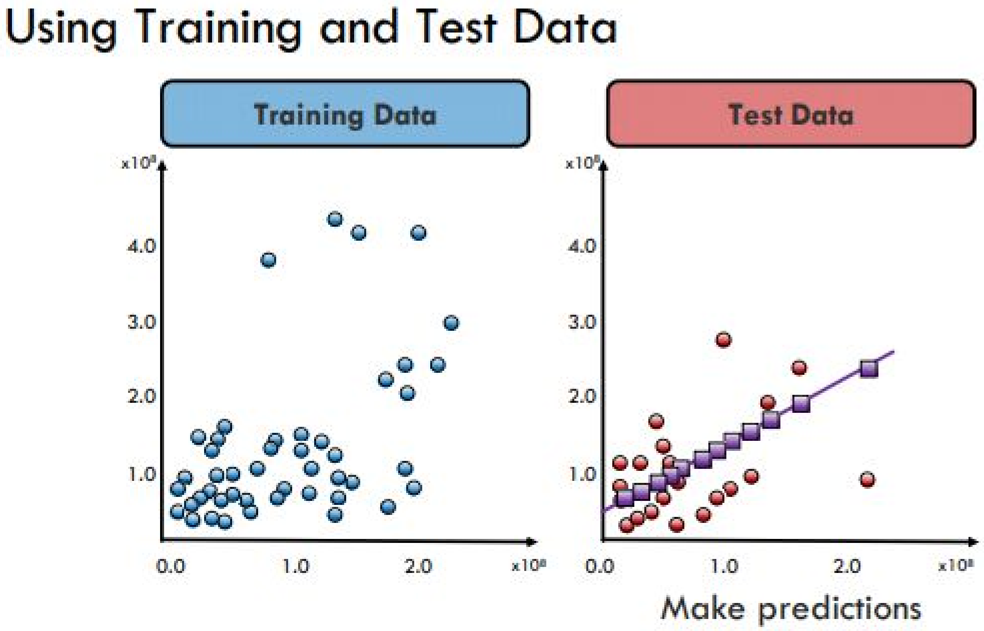
Now we are using linear model first . which would be line that is closest to each point in the training data so the error is minimum this is represented in following image.



Using Training and Test Data

Fit the model

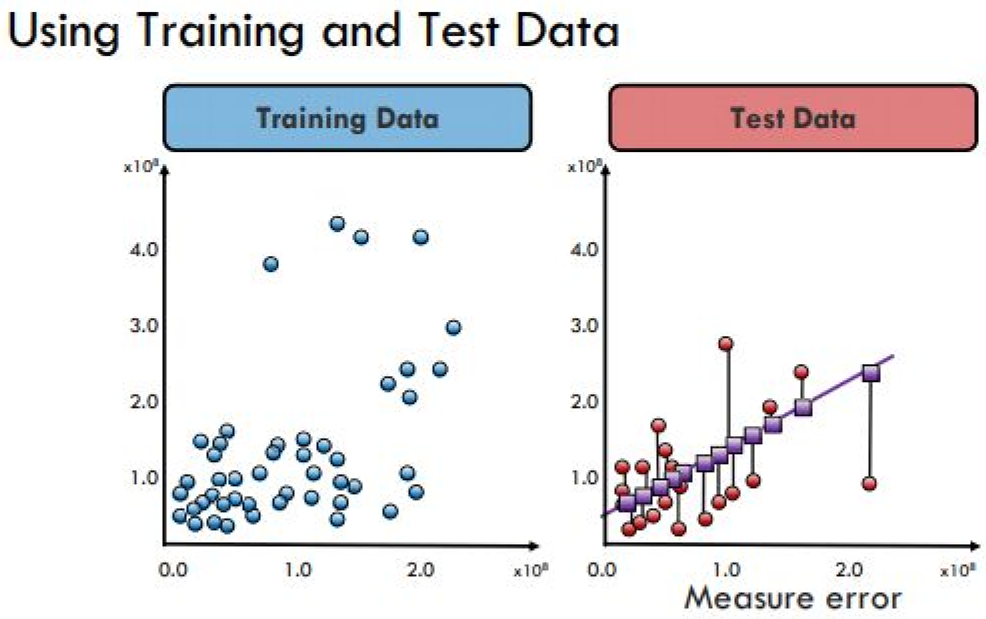
The model training or fitting the the process in with the model tries to find the gradient and bias of the line to minimise the error. As we see there are some points still far from the line so after words we introduce the non liner models. Some samples from the training data is taken as test data to find the performance of the model. The model testing is shown below.



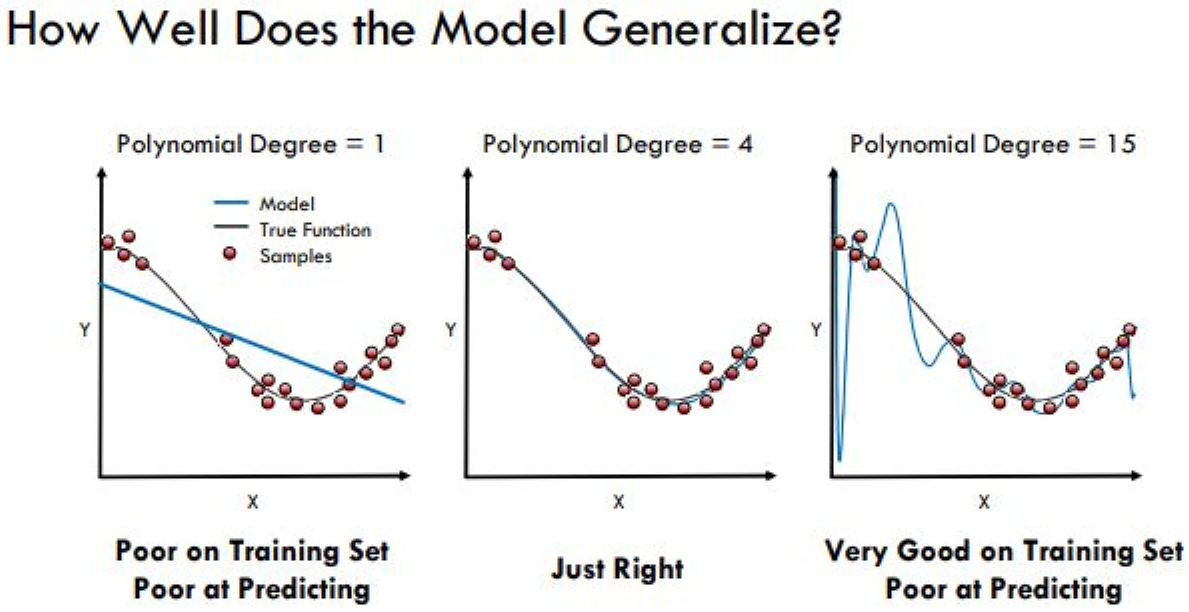
Using Training and Test Data

Make predictions

The error is measured in terms of the distance from the from the the line whose gradient is found during training phase so it is little bit adjusted to perform better on test data.

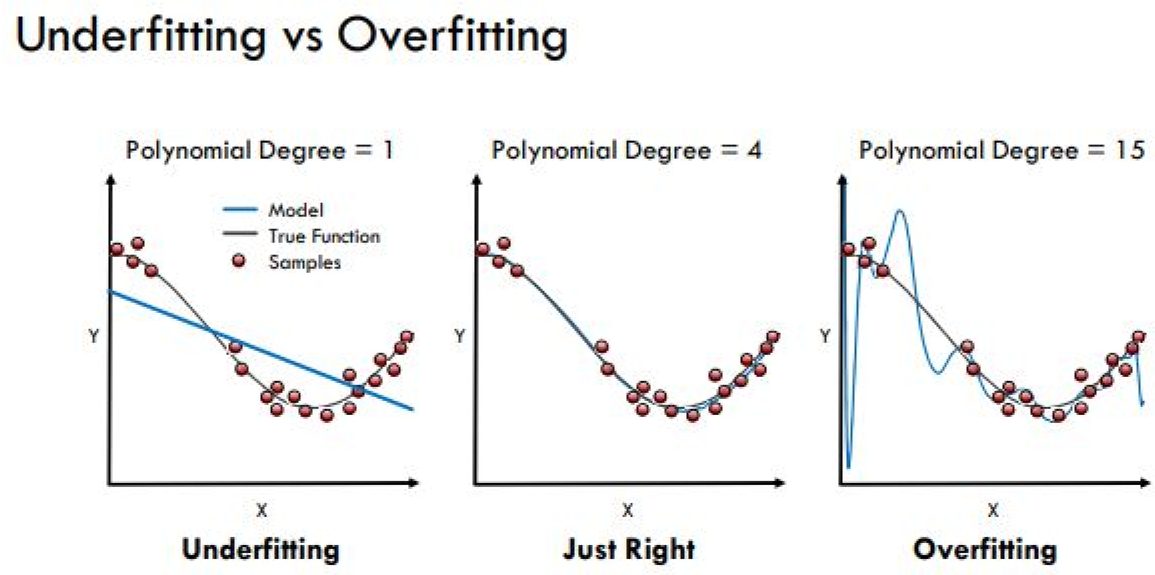


Model Generalization: when the linear model does not fits the data efficiently we take higher degree polynomial but if we do excessive fitting the model me not be generalised anymore now it will be more specific to that training data and produce larger error on generalized input.

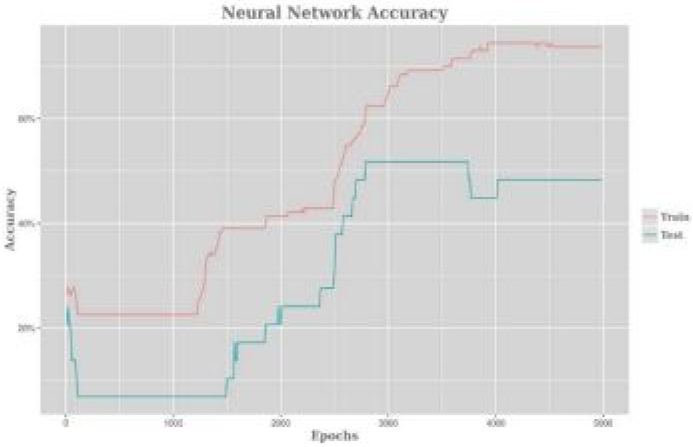


Underfitting:
A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data. *(It's just like trying to fit undersized pants!)*Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough. It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data. In such cases the rules of the machine learning model are too easy and flexible to be applied on such a minimal data and therefore the model will probably make a lot of wrong predictions. Underfitting can be avoided by using more data and also reducing the features by feature selection.

Overfitting:
A statistical model is said to be overfitted, when we train it with a lot of data *(just like fitting ourselves in an oversized pants!)*. When a model gets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set. Then the model does not categorize the data correctly, because of too much of details and noise. The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.



As we see in the first curve the error is still significant so model does not fit the data yet. But as we increased the polynomial degree the fitting is way better. In the polynomial the polynomial is very high degree which causes overfitting over the training data so it performs worse on the test data. Following curve explains how the error or the accuracy varies with overfitting.

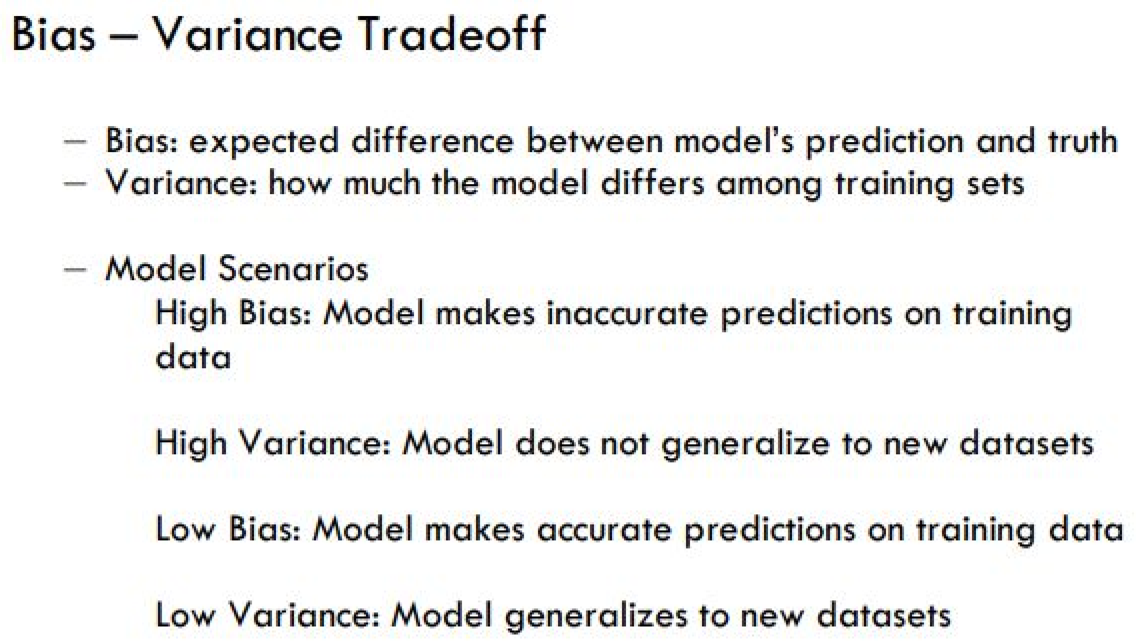


As we see as the epoc increase the accuracy over the training data increases but the accuracy becomes almost constant after a while which indicates that the overfitting is happening since the over alll performance of the model is not increasing .

Bias:
Bias is how far are the predicted values from the actual values. If the **average predicted values are far off from the actual values then the bias is high**.
High bias causes algorithm to miss relevant relationship between input and output variable. When a model has a high bias then it implies that the model is too simple and does not capture the complexity of data thus **underfitting the data**.

Variance: Variance occurs when the model performs good on the trained dataset but does not do well on a dataset that it is not trained on, like a test dataset or validation dataset. **Variance tells us how scattered are the predicted value from the actual value**.
High variance causes overfitting that implies that the algorithm models random noise present in the training data.
when a model has a high variance then the model becomes very flexible and tune itself to the data points of the training set. when a high variance model encounters a different data point that it has not learnt then it cannot make right prediction.

As the variance of the curve increase the error is high because the data distribution is relatively far from the model that is supposed to fit to the data so low variance model is preferred.



Bias – Variance Tradeoff

- Bias: expected difference between model's prediction and truth
- Variance: how much the model differs among training sets

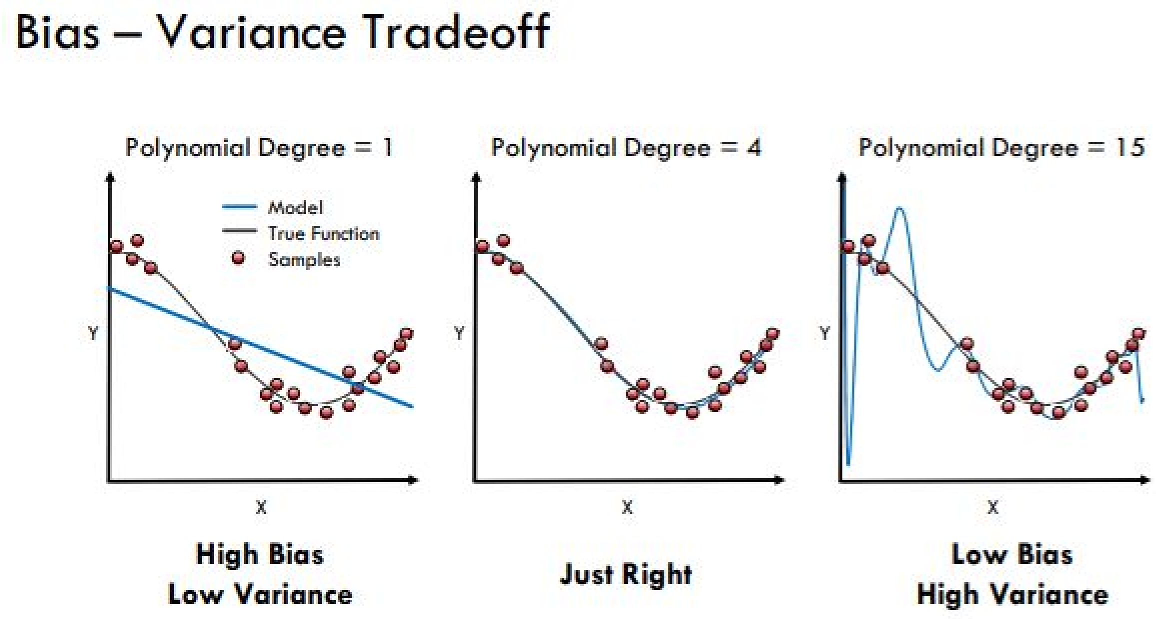
- Model Scenarios

  High Bias: Model makes inaccurate predictions on training data

  High Variance: Model does not generalize to new datasets

  Low Bias: Model makes accurate predictions on training data

  Low Variance: Model generalizes to new datasets



Bias – Variance Tradeoff

High Bias
Low Variance

Just Right

Low Bias
High Variance

High bias is due to a simple model and we also see a high training error. To fix that we can do following things

- Add more input features
- Add more complexity by introducing polynomial features

- Decrease Regularization term

High variance is due to a model that tries to fit most of the training dataset points and hence gets more complex. To resolve high variance issue we need to work on

- Getting more training data
- Reduce input features
- Increase Regularization term