Machine Learning Basics

Session 1

Manu K. Gupta

Department of Management Studies,

MFS of data science and AI

IIT Roorkee.

Machine Learning Module

Previous Modules

- Mathematics and Statistics
- Coding in Python
- Data pre-processing and visualization

This Module

- Linear Regression (Concepts, Demo, implementation etc.)
  - Multiple linear regression, polynomial regression, R^2 score, Regularization.
- Logistic Regression, Support Vector Machines, Naive Bayes Classification, KNN Classifier
  - Softmax Regression, SVR, Confusion Matrix, ROC - AUC, Accuracy, Precision, Recall, F1 Score.
- K-means clustering, Hierarchical clustering, Decision Trees
  - Gini Index, Silhouette score, linkage, Information Gain, Entropy.
- Multi-armed bandit Problem, Reinforcement Learning
  - ϵ-Greedy Strategies, UCB, Thompson Sampling, Q-learning, SARSA
- Linear and Integer Optimization.

Resources: Books

- *Pattern Recognition and Machine Learning*
  - by Christopher M. Bishop
- *Pattern Classification*
  - by Richard Duda, Peter Hart, David Stork.
- *Deep Learning*
  - by Ian Goodfellow, Yoshua Bengio and Aaron Courville
- *Reinforcement Learning*
  - by Richard S. Sutton and Andrew G. Barto,
- *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow*
  - by Aurelien Geron.

Resources: Youtube channels

1. Deepmind
2. Henry AI Labs
3. Simons Institute
4. Talks at Google
5. PyData

This Session

Machine Learning

Regression

Scikit-learn

Learning outcome

- Machine learning Categorization:
  - Supervised, Unsupervised, Reinforcement
- Simple and Multiple Linear Regression
- Implementation in Scikit-learn

Machine Learning

What is machine learning?

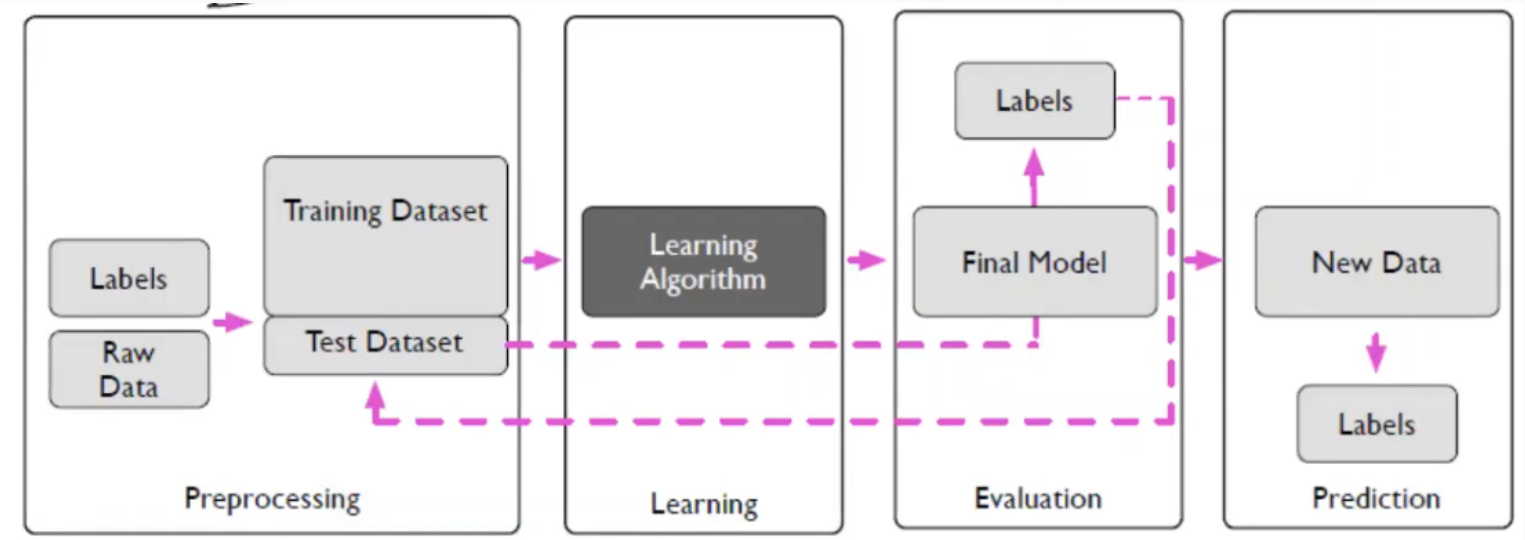
Is this email a spam or not?

- Rule based vs learning algorithm
- Features and labels

Machine Learning Categorization

- Supervised Learning
  - Learn to predict the future events for new data using *labeled data*
- Unsupervised Learning
  - Learn to describe a hidden structure from *unlabeled* data.
- Semi-supervised Learning
  - Typically a small amount of labeled data and a large amount of unlabeled data is available.
- Reinforcement Learning
  - Learning method interacts with the environment by taking actions and receiving feedback (rewards).
  - Actions and the rewards are the main characteristics.

Machine Learning Workflow



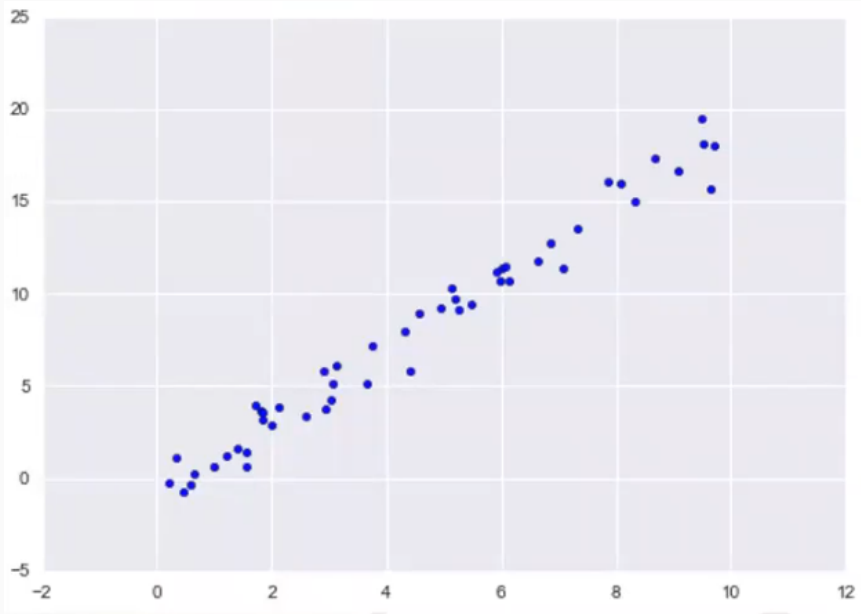
Is this applicable for reinforcement learning?

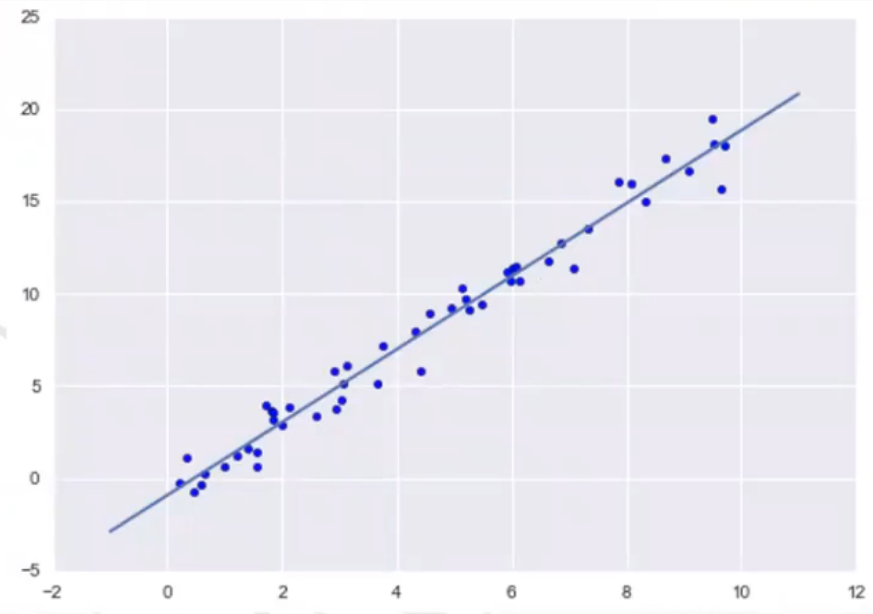
Supervised learning

- Regression
- Classification
- What is the fundamental difference between the above two?

Data is labeled in supervised learning.

Regression





Regression algorithms predict numeric target values.

Regression - Applications

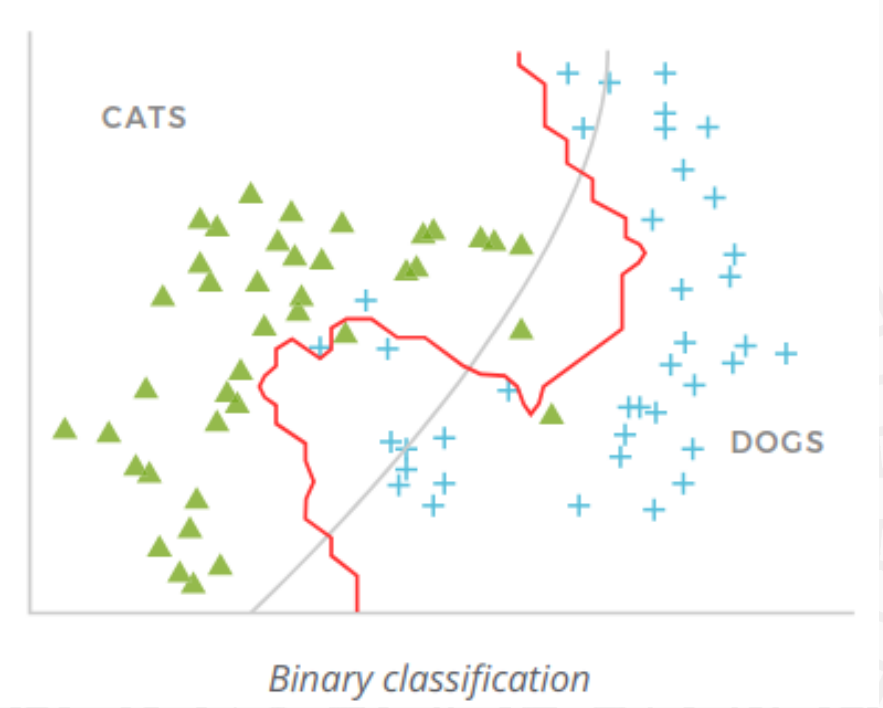
Regression algorithms are used in predicting product demand, sales figures etc.

1. How many items of this product will we be able to sell next month?
2. What will the airfare be for this destination?
3. What's going to be the rental price for this house?
4. What will be the health-care cost vs age?
5. Revenue as a function of price, money spent on promotion, competitior's price and promotional expenses.

Classification

Classification algorithms predict the category of the objects from the dataset.

Classification types



Binary classification

Binary vs multi-class classification.

Classification Examples

Binary classification problems:

- Will this lead convert or not?
- Customer classification based on risk: low risk vs high risk.
- Is this transaction fraudulent or not?
- Customer churn prediction.

Multiclass classification problems:

- Which type of product is this customer more likely to buy: a laptop, a desktop, or a smartphone?
- Sentiment of customers on a product: positive, negative and neutral.

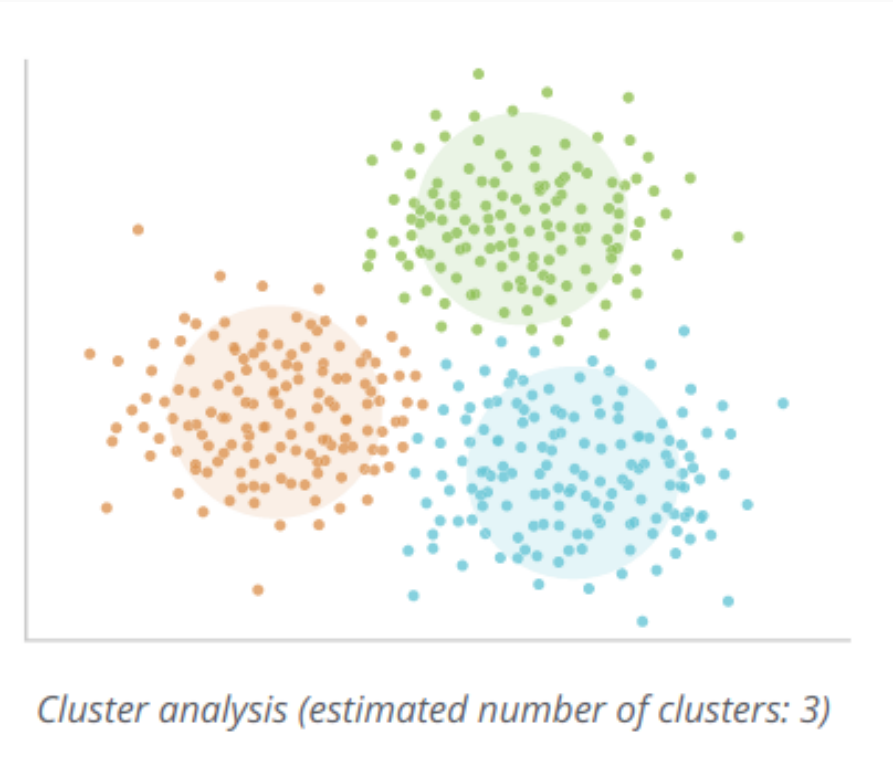
Unsupervised learning

- Unlike supervised learning: labels are **not** known.
- The algorithm is challenged to group items in clusters.
- Descriptive vs predictive analytics.

Clustering Examples

Clustering can solve the following problems:

- What are the main segments of customers we have, considering their demographics and behaviors?
- How can we classify the keywords that people use to reach our website?



Cluster analysis (estimated number of clusters: 3)

Semi-supervised learning

A small amount of labeled data and a large amount of unlabeled data.

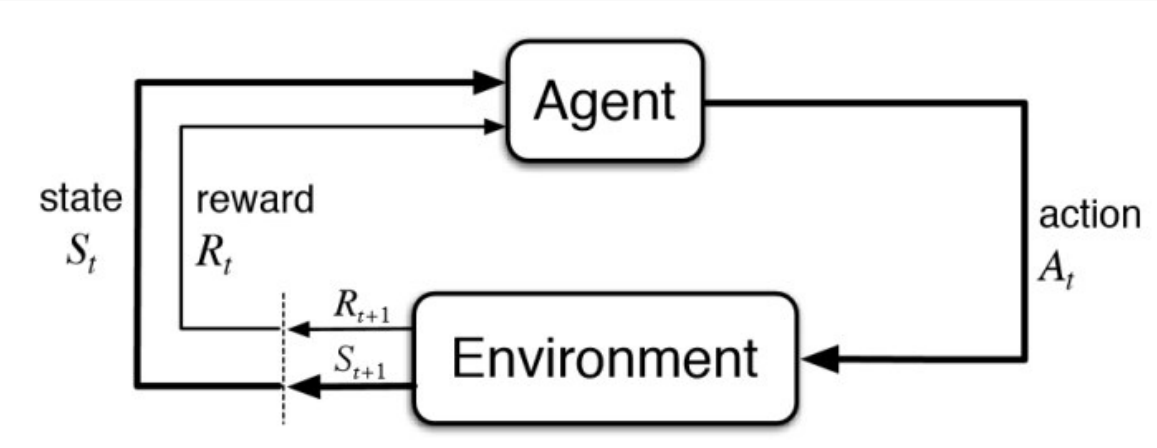
- Supervised learning with additional information.
- Unsupervised learning with constraints.

Reinforcement Learning

Goal-directed learning from *interaction* in an *uncertain* environment.

— Reinforcement Learning: An Introduction, Sutton and Barto, Second Edition

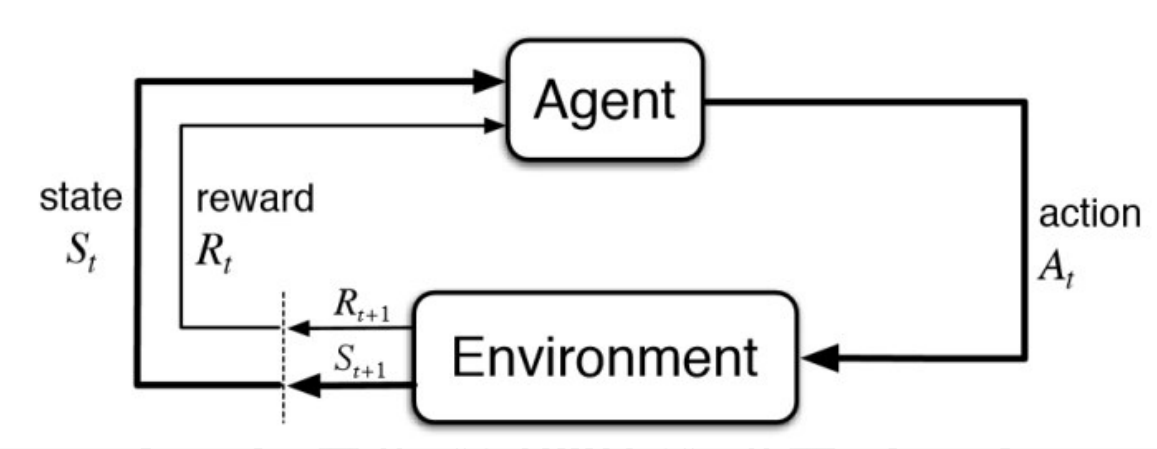
Reinforcement Learning Setting



- **Agent:** The learner or the decision maker.
- **Environment:** The thing the learner interacts with, comprising everything outside the agent.

They interact continually. The agent selects actions. The environment responds to these actions by presenting new situation and giving rewards for the action.

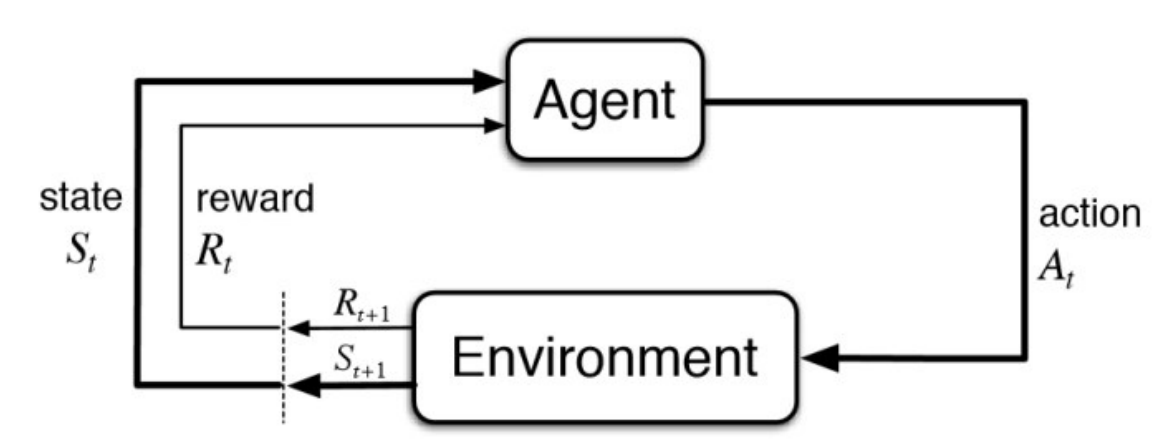
Reinforcement Learning Setting



- **Reward:** Rewards are scalar measures defining what are the good and bad events for the agent.
- **Value:** Value of a state is the total amount of reward an agent is expected to get in future starting from that state.

What is the goal in RL problem?

Reinforcement Learning Setting



- **Goal in RL Problem:** To maximize the total reward "in expectation" over the long run.

Distinguishing Features of Reinforcement Learning

- **Trial and Error:** The learner is not told which actions to take, but instead must discover which actions are most rewarding by trying them.
- **Delayed Rewards:** Actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards.
- **Exploration vs Exploitation:**

> *To obtain a lot of reward, a reinforcement learning agent must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover such actions, it has to try actions that it has not selected before.*
>
> – Sutton and Barto

When to use RL?

- **Trajectories:** Data comes in the form trajectories.
- **Decisions:** Need to take decisions that affect the trajectory data.
- **Feedback:** Need to get feedback about the choice of actions.

Supervised, Unsupervised and Reinforcement Learning

- Supervised learning:
  - Given input-output paired data, the objective of supervised learning is to analyze the input-output relation behind the data.
  - Typical tasks of supervised learning include regression (predicting the real value), classification (predicting the category), and ranking (predicting the order)
  - semi-supervised learning utilizes additional input-only data
  - transfer learning borrows data from other similar learning tasks
  - multi-task learning solves multiple related learning tasks simultaneously.

- Unsupervised learning:
  - Given input-only data, the objective of unsupervised learning is to find something useful in the data.
  - Typical tasks of unsupervised learning include clustering (grouping the data based on their similarity)
  - density estimation (estimating the probability distribution behind the data)
  - anomaly detection (removing outliers from the data)
  - data visualization (reducing the dimensionality of the data to 1-3 dimensions)
  - unsupervised learning methods are sometimes used as data pre-processing tools in supervised learning.

- Reinforcement learning:
  - Supervised learning is a sound approach, but collecting input-output paired data is often too expensive.
  - Unsupervised learning is inexpensive to perform, but it tends to be ad hoc.
  - Reinforcement learning is placed between supervised learning and unsu pervised learning – no explicit supervision (output data) is provided, but we still want to learn the input-output relation behind the data.
  - Instead of output data, reinforcement learning utilizes rewards, which evaluate the validity of predicted outputs.
  - Giving implicit supervision such as rewards is usually much easier and less costly than giving explicit supervision, and therefore reinforcement learning can be a vital approach in modern data analysis.

Points to ponder

- How to obtain a predictor?
- How do we measure the efficacy of the predictor?
- Will the predictor work well in unknown data?
- Implementation with a dataset.

Upcoming topics.

Regression

Regression Problem

- Learning the relationship between some (qualitative and quantitative) input variables

$$X = [x_1, x_2, \cdots x_n]$$

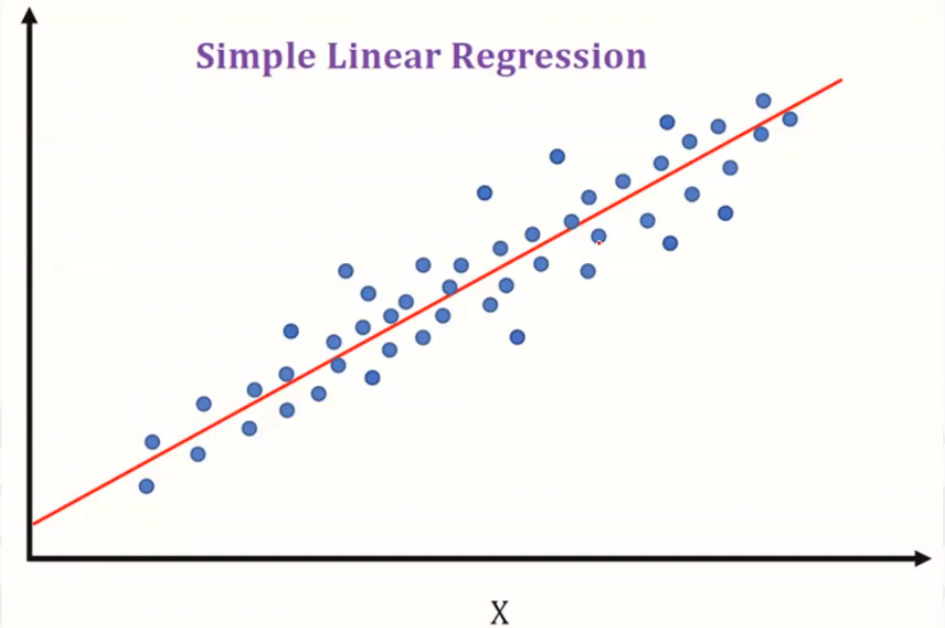
  and a (quantitative) output variable Y.
- Dependent vs independent variables.
- Learning a model $f(.)$ such that

$$Y = f(X) + \epsilon$$

Several Regression Algorithms

- Linear regression
- Polynomial regression
- Ridge and Lasso regression
- Decision Tree
- Random Forest
- Support Vector Machine

Simple Linear Regression



Example: Number of study hours vs score

Linear Regression setup: $Y = b_0 + b_1 X + \epsilon$

Multiple Linear Regression

- Startup data:
  - Profit as function of R&D, Marketing, Admin.
- Say p features $X_1, X_2, \cdots, X_p$.

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_p X_p + \epsilon$$

Points to ponder

- How to learn $b_0, b_1, \cdots, b_p$ from training data?
- Use the above to make prediction.

Linear Regression: Matrix Approach

Closed form solution: $\boxed{b = (M^T M)^{-1} M^T y}$

Efficacy of Regression

How do I know the efficacy of fitted model?

- Mean Square Error
- Coefficient of determination R^2

Mean Square Error

Error or Residual

$$r_i := y_i - \hat{y}_i = y_i - (\hat{b}_0 + \hat{b}_1 x_{i1} + \cdots + \hat{b}_p x_{ip})$$

Can I use sum of errors or mean error?

$$\boxed{MSE = \frac{\sum_i (y_i - \hat{y}_i)^2}{n}}$$

The mean squared error (MSE) tells you how close a regression line is to a set of points.

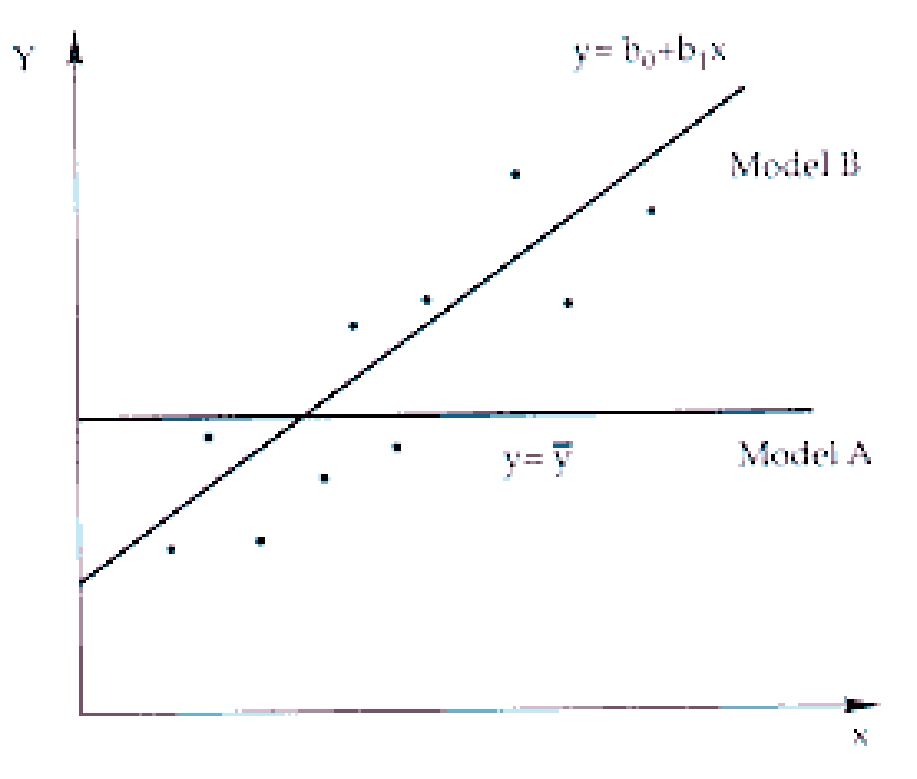
What does MSE zero means?

Goodness of fitting: R^2

Simple linear regression model $Y = b_0 + b_1 X$.

$$R^2 = 1 - \frac{\text{Sum of square residuals from model with } b_0 \text{ and } b_1}{\text{Sum of square residuals from model with mean only}}$$

Interpretation: R^2



Properties of R^2

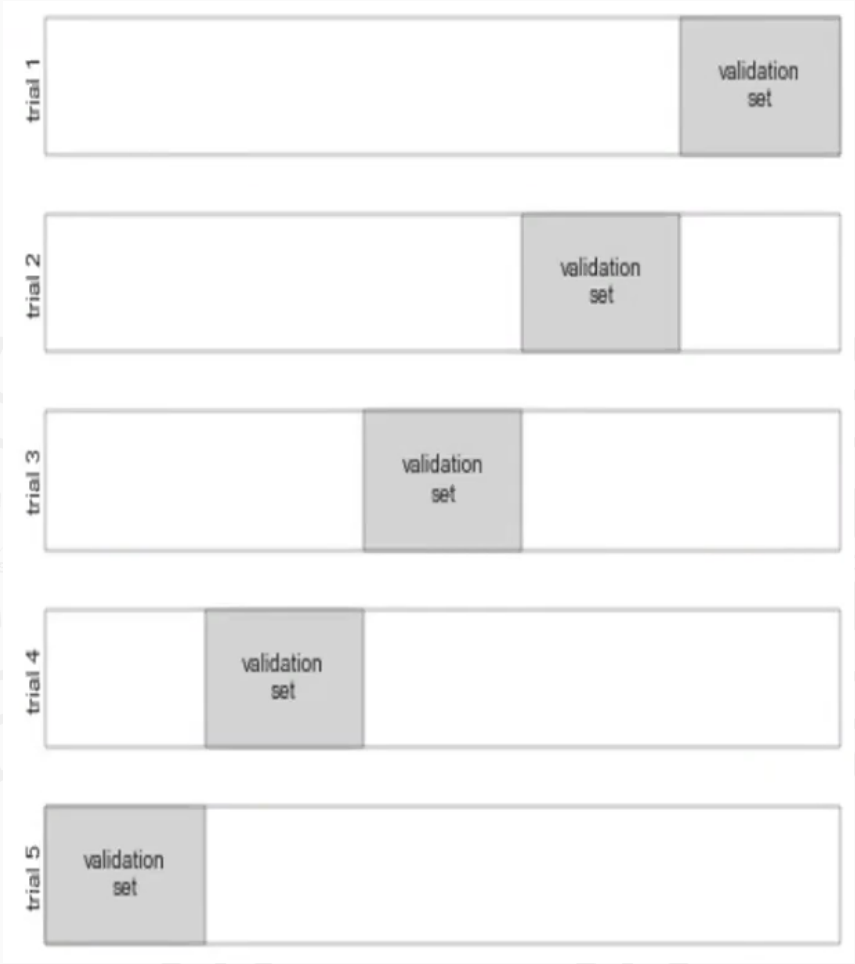
- $0 \leq R^2 \leq 1$
- If SSR = SST, $R^2 = 0$:
  - Model is not useful.
- If SSR = 0, $R^2 = 1$:
  - Model fits all the points perfectly.
- Similar analysis for more than one variable.

How large does R^2 need to be?

Model Validation

- Using one full data set for *model fitting*, and finding a second independent data set for *model validation*.
- Training vs testing for model fitting and model validation.
    - Lost a considerable portion of our data to the model training.
- Cross validation.

Cross Validation

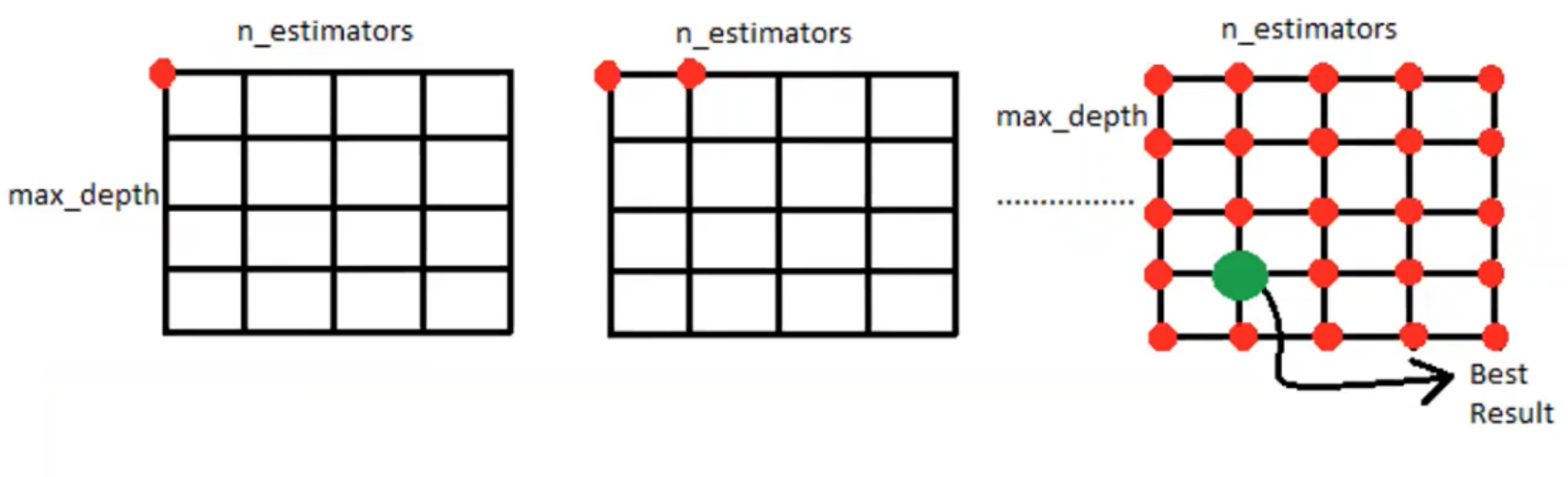


Extreme case: leave-one-out cross validation.

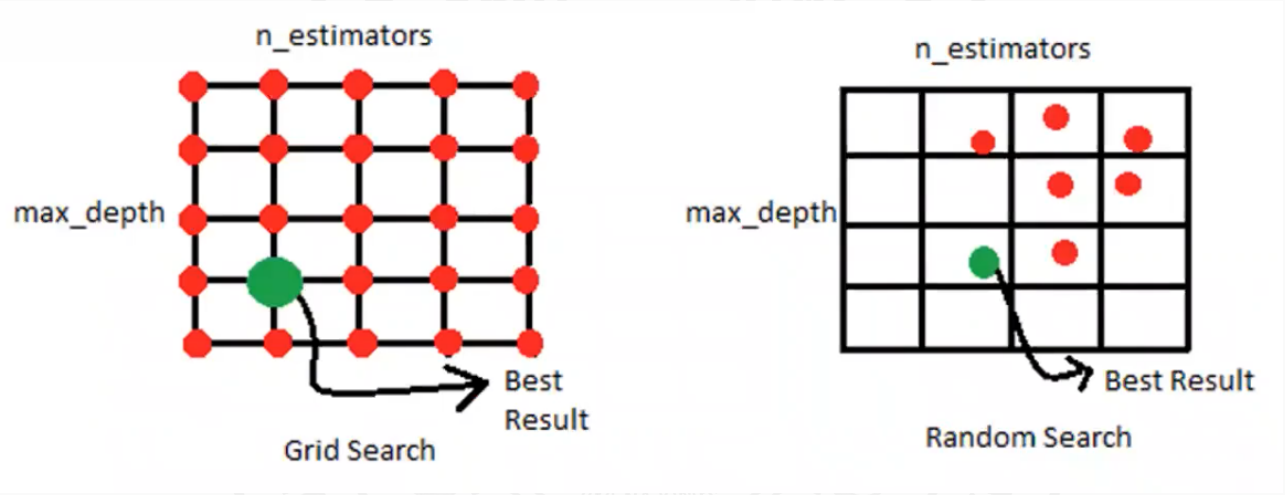
Hyper-parameter tuning

- *Estimators* have parameters that can be tuned.
- Performance critically depends on a few parameters.
  - For example: Random Forest.
  - Hyper-parameters: Number of trees, maximum depth in each tree.
- What parameters to choose?
  - Depends on data-set at hand.
- Automatic parameter search.

Automatic Parameter Search: Grid Search



Automatic Parameter Search: Random Search



How do we implement it?

Scikit-learn

Introduction to sklearn

- Welcome to Scikit learn (or sklearn)!
- Open-source machine learning module with built-in datasets.
- started as a Google summer of code project in 2007.
- INRIA got involved and the first public release was in 2010.
- Currently, more than 30 active contributors.
- Paid sponsorship from INRIA, Google, Tinyclues and the Python Software Foundation.
- Solid implementations of a range of machine learning algorithms.
- Various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.

- A library in python with large number of common algorithms.
- Easy to switch from one model to other.

Implementation

- Built-in machine learning algorithms and models - *estimators*.
- *fit* method – to fit some data.
- *predict* method – to predict target values.

Example

Linear Regression:

```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
X = [[ 1, 2, 3], # 2 samples, 3 features
...
[11, 12, 13]]
y = [0.8, 1.5] # label of each sample
reg.fit(X, y)
reg.predict([4,5,6],[11, 11, 13])
```

Random Forest Classifier:

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier( )
X = [[ 1, 2, 3], # 2 samples, 3 features
...
[11, 12, 13]]
y = [0, 1] # classes of each sample
clf.fit(X, y)
clf.predict([4,5,6],[11, 11, 13])
```

Fit and predict method

- `fit()` method accepts 2 inputs:
  1. Samples matrix X of size `(n_samples, n_features)`
  2. Target value y: real numbers, integers.
  3. X and y are usually expected to be numpy arrays.
- After fitting, `predict()` is used for predicting target values of new data.

sklearn flow

- Import the appropriate estimator class from Scikit-Learn.
- Choose model hyper-parameters.
- Arrange data into a features matrix and target vector.
- Use `fit()` method.
- Apply the model to new data using `predict()` or `transform()`.

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier( )
X = [[ 1, 2, 3], # 2 samples, 3 features
...
[11, 12, 13]]
y = [0, 1] # classes of each sample
clf.fit(X, y)
clf.predict([4,5,6],[11, 11, 13])
```

Simple Linear Regression Demo

Hours	Scores
2.5	21
5.1	47
3.2	27
8.5	75
3.5	50
1.5	20
9.2	88
5.5	60
8.3	81

Simple Linear Regression implementation for scores.csv.

Train-test split automation

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

Multiple Linear Regression Demo

R&D Spend	Administration	Marketing Spend	Profit
165349.20	136897.80	471784.10	192261.83
162597.70	151377.59	443898.53	191792.06
153441.51	101145.55	407934.54	191050.39
144372.41	118671.85	383199.62	182901.99
142107.34	91391.77	366168.42	166187.94

Multiple Linear Regression implementation for startup.csv.

Upcoming topics

- Polynomial regression
- Regularization

Thank you!

manu.gupta@mfs.iitr.ac.in