BITS Pilani
Pilani | Dubai | Goa | Hyderabad

# DSECL ZG 522: Big Data Systems

## Session 15: Spark -Part 3


**Janardhanan PS**

**Professor**

**janardhanan.ps@wilp.bits-pilani.ac.in**

# Topics for today

- **Spark SQL**
- Spark MLlib
  - ✓ Regression
  - ✓ Classification
  - ✓ Clustering
  - ✓ Collaborative filtering

Code link for each topic provided on Canvas / Impartus

# Spark SQL

- Integrate SQL querying into Spark programs (with other analytical processing) to query structured data
  - ✓ Distributed SQL query engine with in-memory processing
- Work with a variety of data in Hive tables, JSON files, Cassandra etc. via SQL interface
- Write your code in Python, Java, Scala or HiveQL
  - ✓ Can significantly speedup HiveQL queries with in-memory processing
- Can use JDBC/ODBC connectors with Spark SQL (ref Spark Thrift Server that is a port of HiveServer2)
- SparkSQL eliminated the need for firing MapReduce jobs in the background as done in Hive

# DataFrame and Table

- Spark SQL works with DataFrames
- DataFrame is a distributed collection of data similar to RDD but organized into columns - similar to relational tables
- val DF = spark.read.option("header",true).csv("bank.csv") // read CSV file into a DF
- DF.show() // Examine the DF
- **DF.createOrReplaceTempView("BANK")** // Create a Table named BANK from DF
- spark.sql("desc BANK").show() // Display schema
- spark.sql("SELECT age, job, balance FROM BANK").show(5) // SQL select qury
- spark.sql("SELECT age, job, balance FROM BANK").where("job == 'admin.'").show(10)
- // SORT by age

  spark.sql(""" SELECT age, job, balance FROM BANK WHERE job in ('admin.','services') order by age""").show(10)
- // SQL GROUP BY clause

  spark.sql(""" SELECT job, count(*) as count FROM BANK GROUP BY job""").show()
- //SQL JOIN

  val joinDF = spark.sql("select * from EMP e, DEPT d where e.emp_dept_id == d.dept_id")

# Topics for today

- Spark SQL
- **Spark MLlib**
  - ✓ Regression
  - ✓ Classification
  - ✓ Clustering
  - ✓ Collaborative filtering

# Spark MLlib

- MLlib is Spark's machine learning (ML) library.
- Makes practical machine learning scalable and easy.
- At a high level, it provides tools such as:
  - ✓ ML Algorithms: classification, regression, clustering, and collaborative filtering
  - ✓ Featurization: feature extraction, transformation, dimensionality reduction, and selection
  - ✓ Pipelines: tools for constructing, evaluating, and tuning ML Pipelines
  - ✓ Persistence: saving and load algorithms, models, and Pipelines
  - ✓ Utilities: linear algebra, statistics, data handling, etc.
- MLlib RDD-based API (spark.mllib) is now in maintenance mode.
- Primary Machine Learning API for Spark is now DataFrame-based API in spark.ml package.
- ***Spark ML*** is occasionally used to refer to the MLlib DataFrame-based API
- Refer: [https://spark.apache.org/docs/latest/ml-guide.html](https://spark.apache.org/docs/latest/ml-guide.html)

# Sample data for Machine Learning

Folder in spark installation - /spark-3.3.1-bin-hadoop3/data/mllib

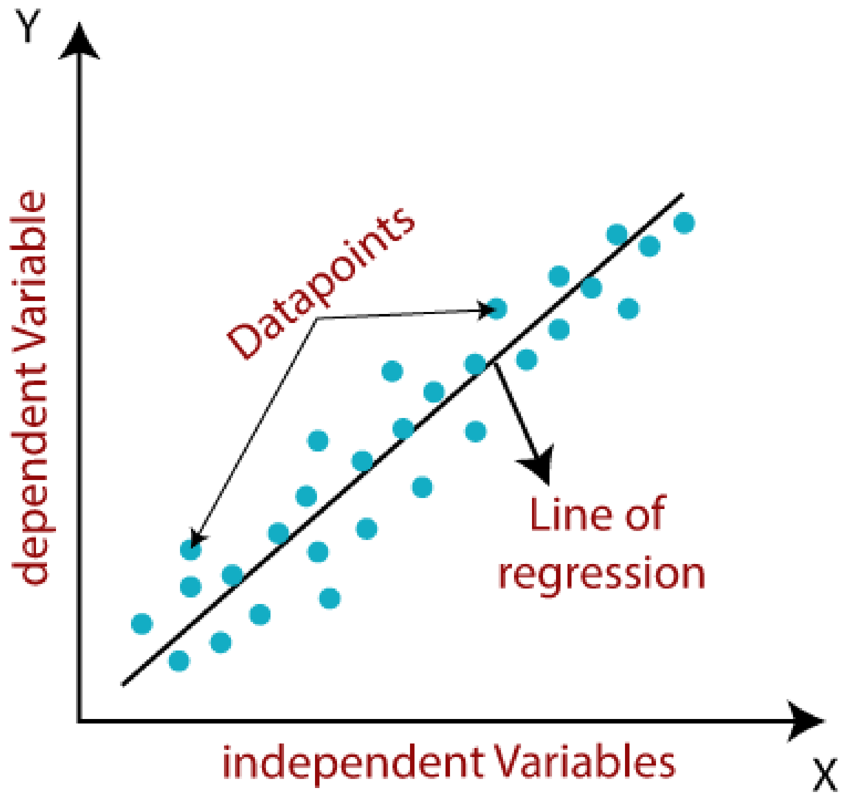
✓ sample_kmeans_data.txt
✓ sample_lda_data.txt
✓ sample_lda_libsvm_data.txt
✓ kmeans_data.txt
✓ sample_libsvm_data.txt
✓ pagerank_data.txt
✓ sample_linear_regression_data.txt
✓ sample_movielens_data.txt
✓ sample_multiclass_classification_data.txt
✓ sample_binary_classification_data.txt
✓ sample_svm_data.txt
✓ streaming_kmeans_data_test.txt
✓ sample_isotonic_regression_libsvm_data.txt

# Topics for today

- Spark SQL
- Spark MLlib
  - ✓ **Regression**
  - ✓ Classification
  - ✓ Clustering
  - ✓ Collaborative filtering

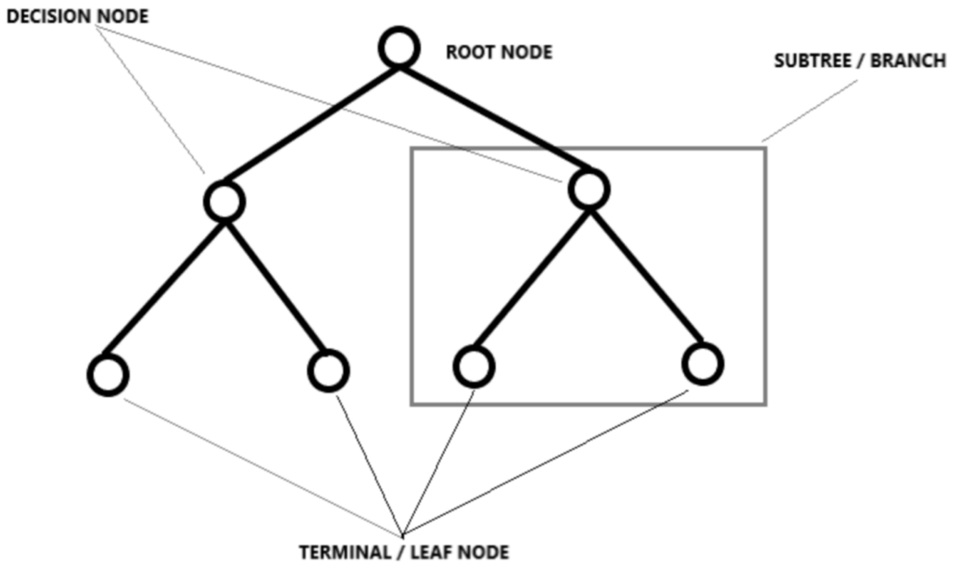
# Linear regression

- Supervised learning
- Predict outcome (dependent variable) based on a set of features (independent variables)
  - ✓ Use a training data set to learn the function
  - ✓ Use new data to predict outcome from features
- Multiple linear regression
  - ✓ Multiple independent variables
- Can extrapolate given it learns a fn

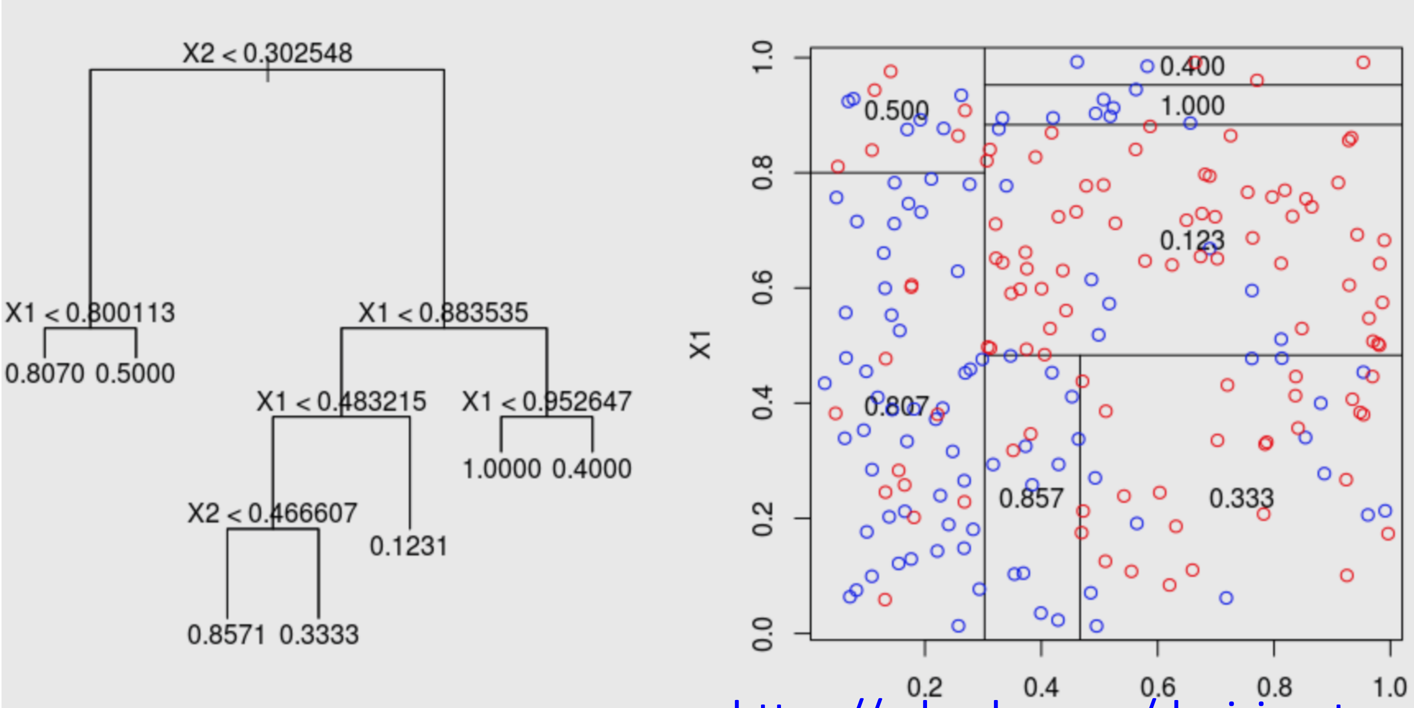


# Non-linear Regression: Decision tree

- Divide data into 2 subsets at each level till information gain is low or max tree depth is reached
- E.g. 15-20 features determine price of a house
  - ✓ Predict price of a house given a set of values for features




- How to determine split points ? Value that minimises MSE in the 2 groups.
- Cannot predict if test data is very different from training data
  - ✓ Unlike linear regression




https://gdcoder.com/decision-tree-regressor-explained-in-depth/

# Topics for today

- Spark SQL
- Spark MLlib
  - ✓ Regression
  - ✓ **Classification**
  - ✓ Clustering
  - ✓ Collaborative filtering

# Classification

- Supervised learning approach
- Train a classifier with labelled data, i.e. with input as a set of features (independent variables) and a label (dependent variable)
- Test the model with an input feature vector and output a label / class
- Similar algorithms as regression
- E.g.
  - ✓ Wine quality is measured by a set of feature attributes, e.g. alcohol content, acidity level, fermentation period, percentages of various components etc.
  - ✓ Train with a set of samples (with feature attributes) and add quality label for each sample
  - ✓ Given a new sample with a set of features predict the quality

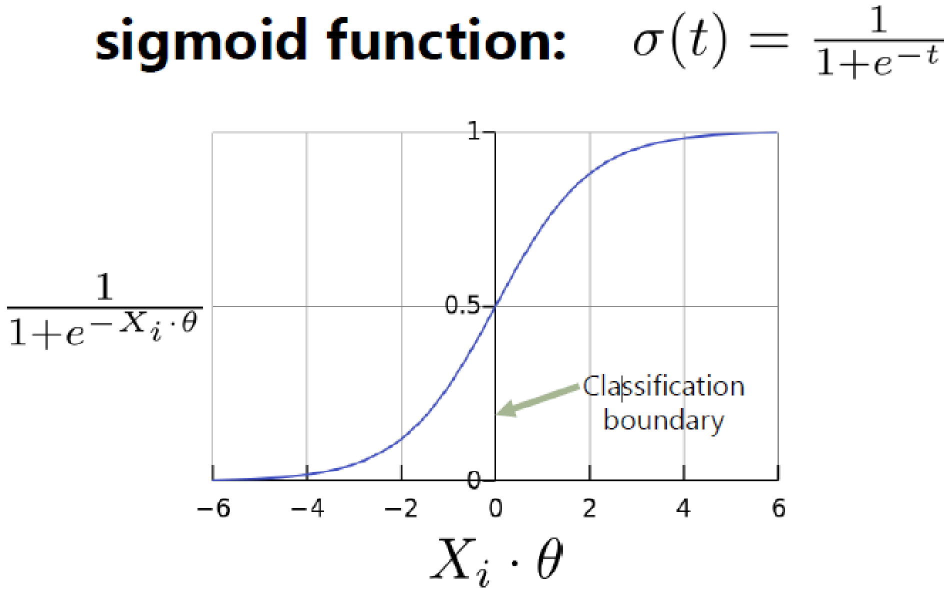
# Common Classifiers

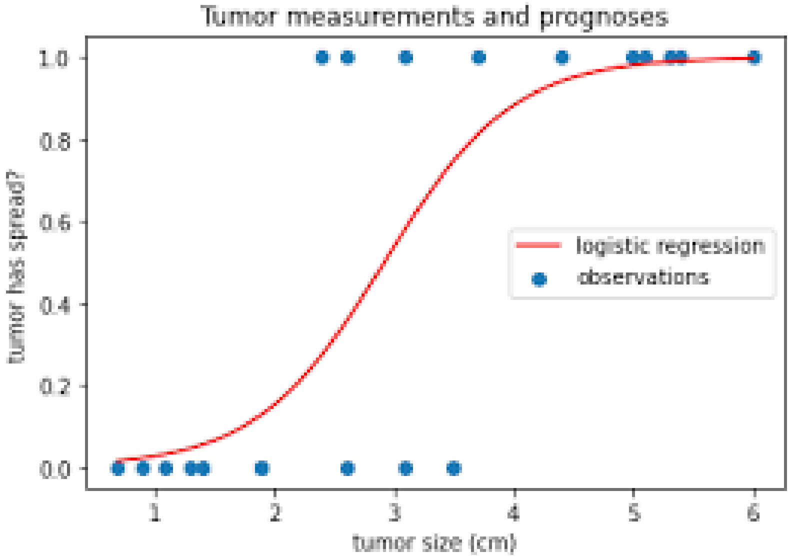
- Logistic regression
- Decision tree classifier
- Random forest classifier
- Gradient-boosted tree classifier
- Multilayer perceptron classifier
- One-vs-Rest classifier (a.k.a. One-vs-All)

# Logistic Regression (Classification)

- A model that generates a probability for each possible discrete label value in classification problems by applying a sigmoid function to a linear prediction.
- Logistic regression is often used in binary classification problems
- Sigmoid function maps logistic or multinomial regression output (log odds) to probabilities, returning a value between 0 and 1.

[Introduction to Logistic Regression | by Ayush Pant | Towards Data Science]



# Topics for today

- Spark SQL
- Spark MLlib
  - ✓ Regression
  - ✓ Classification
  - ✓ **Clustering**
  - ✓ Collaborative filtering

# What is Clustering

- Clustering is the most popular version of unsupervised learning.
- In unsupervised learning, the goal is to identify patterns or structures in the data without any prior knowledge of what to expect.
- In Clustering, the goal is to group data points based on their similarity.

# Use Case of Clustering

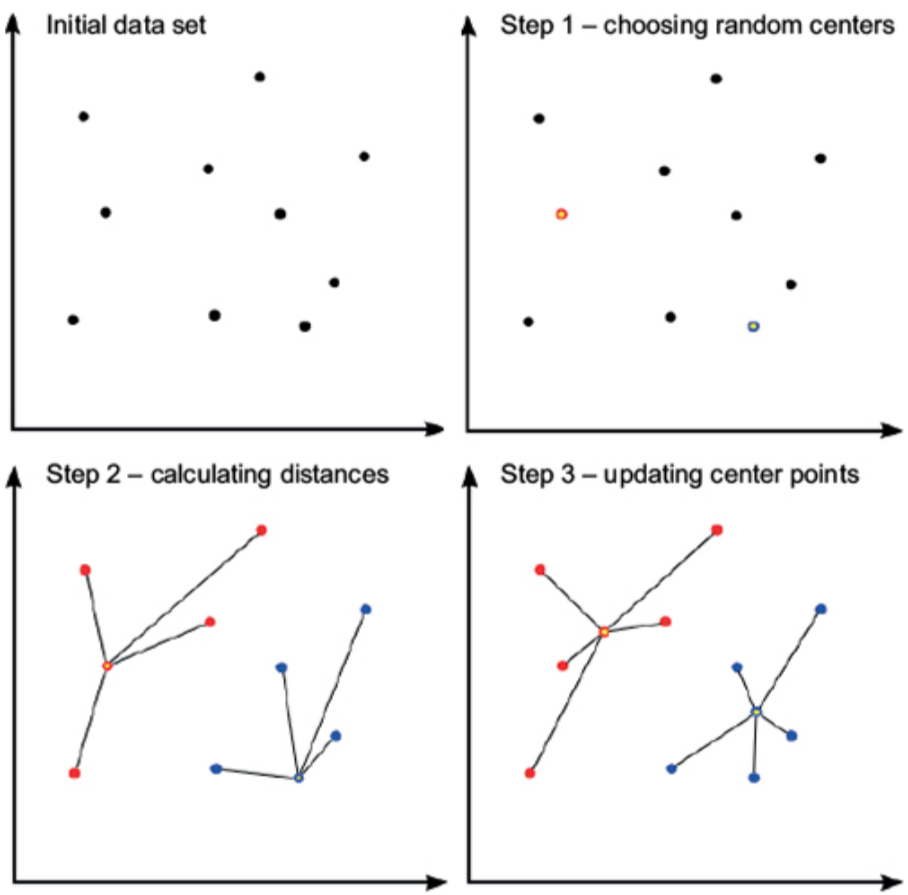
- Suppose you are the head of a retail store and wish to understand the preferences of your customers.
- Can you look at the details of each customer and devise a unique business strategy for each one of them?
- What you can do is cluster all of your customers into, say 5 groups based on their purchasing habits and use a separate strategy for each group.

# Clustering

- An unsupervised machine learning method
  - ✓ Items are not labelled
- Group set of items into clusters, i.e. learn the labels automatically
- Why isn’t data labelled ?
  - ✓ Too expensive, e.g. grouping search results
  - ✓ Not known in advance, e.g. market segmentation, where an algorithm needs to find it
- Can include clustering to label data as part of a classification process
- E.g.: Spam filter, Fake news detection
  - ✓ Use email header, sender, specific content to cluster messages, articles etc.
  - ✓ Users are sometimes asked to label potential SPAM

# K-Means clustering method

- Iteratively choose new centroids till convergence
- Distance computation can run in parallel in each iteration
- How good is the clustering: Silhouette score (mean squared intra cluster distance) - low means tightly coupled clusters found
- Prediction on new data:
  - ✓ Use training data to create cluster centres
  - ✓ Use new data to predict cluster label for each data item
- Other methods in Spark: Gaussian Mixture Model, Power Iteration Clustering etc.




[K-means clustering, starting with 4 left-most points (shabal.in)]

# Types of Clustering

- Exclusive clustering
  - ✓A form of grouping that requires a data point to exist only in one cluster.
  - ✓ This can also be referred to as “hard” clustering.
  - ✓ The K-means clustering algorithm is an example of exclusive clustering
- Overlapping clustering
  - ✓ Allows data points to belong to multiple clusters with separate degrees of membership.
  - ✓ “Soft” or fuzzy k-means clustering is an example of overlapping clustering.

# Use Case of Clustering

- Suppose you are the head of a retail store and wish to understand the preferences of your customers.
- Can you look at the details of each customer and devise a unique business strategy for each one of them?
- What you can do is cluster all of your customers into, say 5 groups based on their purchasing habits and use a separate strategy for each group.

# Topics for today

- Spark SQL
- Spark MLlib
  - ✓ Regression
  - ✓ Classification
  - ✓ Clustering
  - ✓ **Collaborative filtering**

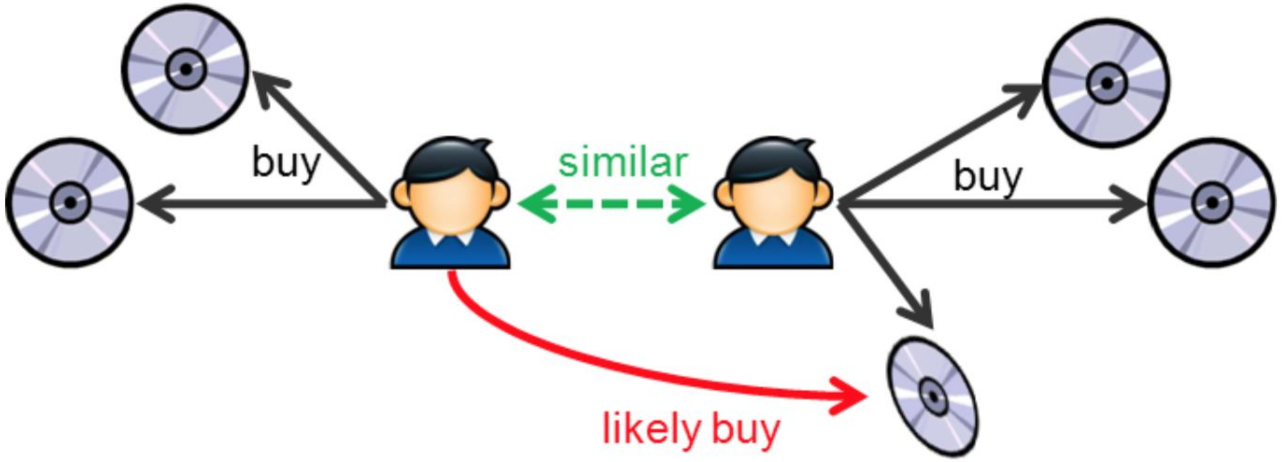
# Recommendation systems

- Try to predict the preference of a user based on past behaviour
  - ✓ Set of items (X)
  - ✓ Set of users (Y)
  - ✓ Learn a function based on past interactions that predicts the likeliness of X to Y
  - ✓ e.g. movie, songs, shopping recommendations
- Broadly 2 types
  - ✓ Content based filtering: uses only attributes of items
    - e.g. Anyone has heard songs a, b, c then likely may want to hear u and v next <- same recommendation for all users
    - Typically items must have multiple attributes to create relationships, e.g. movie genre, actors, director, ..
  - ✓ Collaborative filtering: utilizes user interaction behaviour in addition to item attributes
    - e.g. Specific user’s interactions with songs matters to understand who are similar users what else do they listen to so that next song recommendation can be generated
    - Item attributes are less important
- A hybrid approach is better because Collaborative Filtering has a cold start issue - enough user interactions have to be there for recommendation

# Collaborative Filtering

- Similar to asking friends or people with "similar" preferences
- Most collaborative filtering systems use a "similarity index" wrt active user
  - ✓ Set of "similar user" preferences are aggregated
- What matters is the relationship of users to items rather than only among items
  - ✓ Content based filtering is the latter approach
- So similarity in items is determined by similarity of preferences of those items by the users who have rated both items
- Core technique is to measure similarity or correlation
- Unsupervised technique

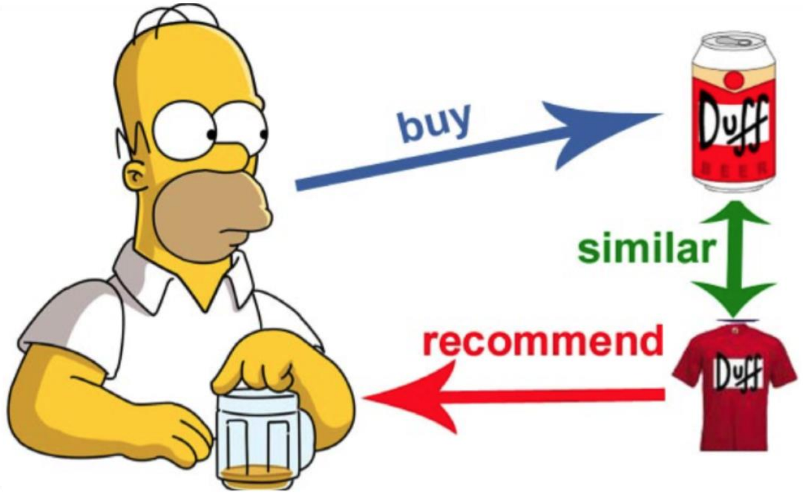
# Nearest-neighborhood approaches




Source: https://dzone.com/articles/recommendation-engine-models

User-based: 2 users are similar because they have similar ratings on same items

Item-based: 2 items are similar because same user has given similar ratings




Source: https://medium.com/tiket-com-dev-team/build-recommendation-engine-using-graph-cbd6d8732e46

# Matrix Factorization approach

- Problem with NN approaches is scalability and data sparsity
  - ✓ e.g. what if you cannot find enough users who have similar rating on same set of movies ?
- So we need lower dimensional spaces to capture user preferences
  - ✓ e.g. A set of movies may belong to the same genre - so instead of movie names we should build user preferences on genre - an example of a 'latent feature'.
- MF approaches deal with sparse data sets and 'latent features'
  - ✓ User similarities may be inferred based on underlying taste in items (e.g. movie genre) instead of specific items
- However, latent features need not be an item attribute - so this is a hard problem
  - ✓ e.g. it is about the kind of movie plot and not the genre
- Spark MLlib uses an implementation of MF called Alternating Least Squares (ALS)

# Additional Reading

- https://www.analyticsvidhya.com/blog/2021/01/a-quick-overview-of-regression-algorithms-in-machine-learning/
- https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f
- https://databricks.com/blog/2014/07/23/scalable-collaborative-filtering-with-spark-mllib.html
- https://www.section.io/engineering-education/sparksql-mllibspark-part-3/
- https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26

Next Session:
Spark - Part 4