What is a Recommendation System

Recommendation engines are a subclass of machine learning which generally deal with ranking or rating products / users. Loosely defined, a recommender system is a system which predicts ratings a user might give to a specific item. These predictions will then be ranked and returned back to the user.

They're used by various large name companies like Google, Instagram, Spotify, Amazon, Reddit, Netflix etc. often to increase engagement with users and the platform. For example, Spotify would recommend songs similar to the ones you've repeatedly listened to or liked so that you can continue using their platform to listen to music. Amazon uses recommendations to suggest products to various users based on the data they have collected for that user.

Recommender systems are often seen as a "black box", the model created by these large companies are not very easily interpretable. The results which are generated are often recommendations for the user for things that they need / want but are unaware that they need / want it until they've been recommended to them.

There are many different ways to build recommender systems, some use algorithmic and formulaic approaches like Page Rank while others use more modelling centric approaches like collaborative filtering, content based, link prediction, etc. All of these approaches can vary in

complexity, but complexity does not translate to “good” performance. Often simple solutions and implementations yield the strongest results. For example, large companies like Reddit, Hacker News and Google have used simple formulaic implementations of recommendation engines to promote content on their platform. In this article, I’ll provide an intuitive and technical overview of the recommendation system architecture and the implementation of a few different variations on a sample generated dataset.

What Defines a Good Recommendation?

Identifying what defines a good recommendation is a problem in itself that many companies struggle with. This definition of “good” recommendations help evaluate the performance of the recommender you built. The quality of a recommendation can be assessed through various tactics which measure coverage and accuracy. Accuracy is the fraction of correct recommendations out of total possible recommendations while coverage measures the fraction of objects in the search space the system is able to provide recommendations for. The method of evaluation of a recommendation is solely dependent on the dataset and approach used to generate the recommendation. Recommender systems share several conceptual similarities with the classification and regression modelling problem. In an ideal situation, you would want to see how real users react to recommendations and track metrics around the user to improve your recommendation, however, this is quite difficult to accomplish. Common statistical

accuracy measures to evaluate accuracy of a recommender are RMSD, MAE, and k fold cross validation.

K Fold Cross Validation

- Imagine you've built a model which will predict how well a user will rate an item based on a set of features. K fold cross validation can be used to infer the results of the model through accuracy metrics
- Same idea as a train test split, except we create K many randomly assigned training and test sets
- Each individual training set / fold is used to train on the recommendation system independently and then measure the accuracy of the resulting systems against the test set
- We take the average of accuracy score to see how well the recommendation system is learning
- This method is beneficial to prevent your model from overfitting, however it is a computationally extensive process

MAE (Mean Absolute Error)

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error
y_i = prediction
x_i = true value
n = total number of data points

- Represents the average absolute value of each error in rating prediction
- Lower the MAE score the better

RMSD (Root Mean Square Deviation)

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

RMSD = root-mean-square deviation
i = variable i
N = number of non-missing data points
x_i = actual observations time series
$\hat{x}_i$ = estimated time series

- A similar metric to MAE but has a stronger penalty for when the prediction is very far from the true value and weaker penalty for when the prediction is closer to the true value
- Taking the squares off the difference of true and predicted values instead of the sum of the absolute values. This ensures that the resulting value is always positive and is larger when the difference is high and smaller when the difference is low.
- The lower the RMSD score the better

These metrics are often used to evaluate the quality of a recommendation but they lack various components. Having user data associated with recommendations is essential to know the true quality of a recommendation. Being able to track hit rates of recommendations, engagement to the platform, responsiveness etc. will provide a clearer viewpoint of recommendation quality. Other components to be aware of is to know when to change recommendations when the user hasn't interacted with them for an X amount of time, or when to re-train recommenders based on new ratings or interactions from the users. You would also want to pay attention to whether or not these recommendations are limiting the user to a subsection of the products, how does the recommender deal with novelty, diversity and selection bias. A/B testing is often the method used to keep track of these metrics (check out my articles on bayesian and frequentist a/b testing).

Data

In the following sections we are going to go more in depth about different methods of creating recommendation engines and the associated implementations in Python. This section will provide a script which will synthesize a dataset associated with books. The dataset will be used for applications of recommendation systems in the following sections, the goal of this article is not to get meaningful results but to show the user the intuition and implementation behind various types of recommendation engines. Hence the results of these recommendations will be meaningless but the methodologies will be similar to those in production grade environments in industry.

Collaborative Filtering Systems

Intuition

Collaborative filtering is the process of predicting the interests of a user by identifying preferences and information from many users. This is done by filtering data for information or patterns using techniques involving collaboration among multiple agents, data sources, etc. The underlying intuition behind collaborative filtering is that if users A and B have similar taste in a product, then A and B are likely to have similar taste in other products as well.

There are two common types of approaches in collaborative filtering, memory based and model based approach.

1. Memory based approaches — also often referred to as neighbourhood collaborative filtering. Essentially, ratings of user-item combinations are predicted on the basis of their neighbourhoods. This can be further split into user based collaborative filtering and item based collaborative filtering. User based essentially means that like minded users are going to yield strong and similar recommendations. Item based collaborative filtering recommends items based on the similarity between items calculated using user ratings of those items.
2. Model based approaches — are predictive models using machine learning. Features associated to the dataset are parameterized as inputs of the model to try to solve an optimization related problem. Model based approaches include using things like decision trees, rule based approaches, latent factor models etc.

Advantages

The main advantage to using collaborative filtering models is its simplicity to implement and the high level coverage they provide. It is also beneficial because it captures subtle characteristics (very true for latent factor models) and does not require understanding of the item content.

Disadvantages

The main disadvantage to this model is that it's not friendly for recommending new items, this is because there has been no user/item interaction with it. This is referred to as the cold start problem.

Memory based algorithms are known to perform poorly on highly sparse datasets.

Examples

Some examples of collaborative filtering algorithms :

- YouTube content recommendation to users — recommending you videos based on other users who have subscribed / watched similar videos as yourself.
- CourseEra course recommendation — recommending you courses based on other individuals who have finished existing courses you've finished.

Content Based Systems

Intuition

Content based systems generate recommendations based on the users preferences and profile. They try to match users to items which they've liked previously. The level of similarity between items is generally established based on attributes of items liked by the user. Unlike most collaborative filtering models which leverage ratings between target user and other users, content based models focus on the ratings

provided by the target user themselves. In essence, the content based approach leverages different sources of data to generate recommendations.

The simplest forms of content based systems require the following sources of data (these requirements can increase based on the complexity of the system you're trying to build):

1. Item level data source — you need a strong source of data associated to the attributes of the item. For our scenario, we have things like book price, num_pages, published_year, etc. The more information you know regarding the item, the more beneficial it will be for your system.
2. User level data source — you need some sort of user feedback based on the item you're providing recommendations for. This level of feedback can be either implicit or explicit. In our sample data, we're working with user ratings of books they've read. The more user feedback you can track, the more beneficial it will be for your system.

Advantages

Content based models are most advantageous for recommending items when there is an insufficient amount of rating data available. This is because other items with similar attributes might have been rated by the user. Hence, a model should be able to leverage the ratings along

with the item attributes to generate recommendations even when there isn't a lot of data.

Disadvantages

There are two main disadvantages of content based systems.

1. The recommendations provided are "obvious" based on the items / content the user has consumed. This is a disadvantage because if the user has never interacted with a particular type of item, that item will never be recommended to the user. For example, if you've never read mystery books, then through this approach, you will never be recommended mystery books. This is because the model is user specific and doesn't leverage knowledge from similar users. This reduces the diversity of the recommendations, this is a negative outcome for many businesses.
2. They're ineffective for providing recommendations for new users. When building a model you require a history of explicit / implicit user level data for the items. It's generally important to have a large dataset of ratings available to make robust predictions without overfitting.

Examples

Some examples of content based systems are :

- Amazon product feed (you're being recommended products similar to what you've previously purchased)
- Spotify music recommendations

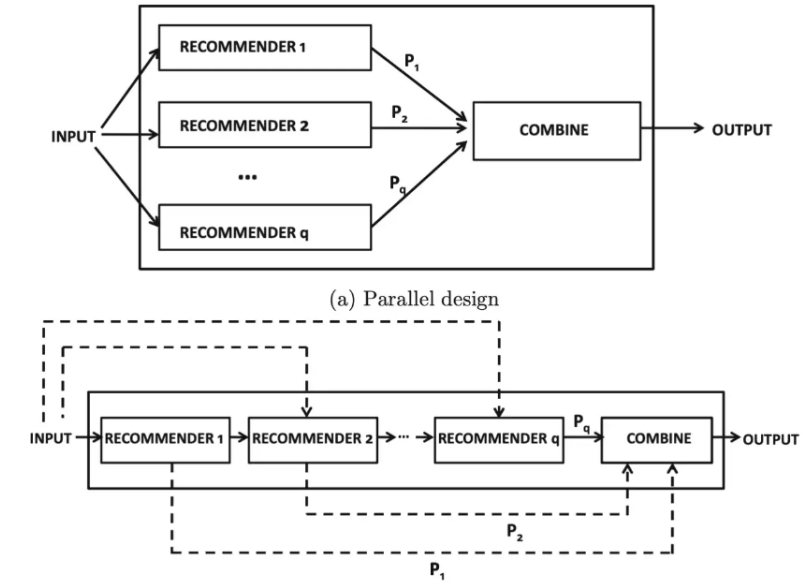
There are many excellent content based systems which are built algorithmically without the dependency on a model based approach. For example companies like Hacker Rank and Reddit have been known to previously used algorithmic approaches to recommend new posts on their platform to users. The key to building an algorithmic approach to content based recommenders lies in defining a set of rules for your business which can be used to rank items. In the case of Reddit, their recommendations are bounded by time of post, number of likes, number of dislikes, number of comments, etc. This can be factored into a formula to generate a score for a post, a high score would yield a high recommendation and vice versa.

Hybrid Recommendation System

Intuition

Various methods of recommendation systems have their own benefits and flaws. Often, many of these methods may seem restrictive when used in isolation, especially when multiple sources of data are available for the problem. Hybrid recommender systems are ones designed to use different available data sources to generate robust inferences.

Hybrid recommendation systems have two predominant designs, parallel and sequential. The parallel design provides the input to multiple recommendation systems, each of those recommendations are combined to generate one output. The sequential design provides the input parameters to a single recommendation engine, the output is passed on to the following recommender in a sequence. Refer to the figure below for a visual representation of both designs.



Advantages

Hybrid systems combine different models to combat the disadvantages of one model with another. This overall reduces the weaknesses of using individual models and aids in generating more robust recommendations. This yields more robust and personalized recommendations for users.

Disadvantages

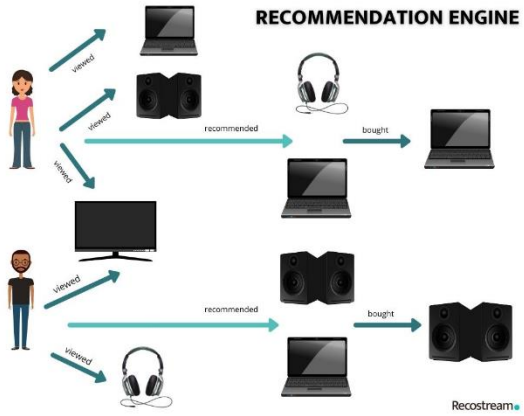
These types of models generally have high computational complexity and require a large database of ratings and other attributes to keep up to date. Without up to date metrics (user engagement, ratings, etc.) it makes it difficult to retrain and provide new recommendations with updated items and ratings from various users.

Example

Netflix is a company which uses a hybrid recommendation system, they generate recommendations to users based on the watch and search style of similar users (collaborative filtering) in conjunction with movies which share similar characteristics who've been rated by users (content based).

Stages of creating a recommendation and how to improve them

The first step is called candidate generation. Depending on the query given, the system generates a set of the most relevant candidate items to potentially suggest to the user.



he next stage is narrowing down the data by ranking the candidates. Optimizing the process as shown in the above image is what makes a recommendation successful.

This is a task for artificial intelligence. The most precise recommendation systems utilize self-learning models that register, analyze and interpret everything there is to know about user preferences.

Machine learning algorithms pave the way for personalized recommendations.

Recommendation system machine learning algorithms

Machine learning, a subset of artificial intelligence, is a process through which a system explores patterns and connections occurring in vast historical data volumes (e.g. through association rules). This way it can delve deep into complex matters, such as human behavior, and understand them better.

To produce personalized content, recommendation systems must be trained by algorithms. Let us picture this by comparing the machine to a human.

A creative writing student receives from a tutor instructions on how to self-educate. They are specific guidelines that the teacher arrived at through trial and error. Such instructions can be compared to machine learning algorithms; the teacher is the creator of the algorithm; the student is a recommendation system.

In order for the student to quickly learn at home and, consequently, produce high-quality and engaging texts, the teacher’s self-learning instructions must be extremely precise and effective.

Similarly, if we want our systems to generate recommendations that boost user engagement, the recommendation algorithms have to be efficient.

Unlike the complicated deep learning models using deep neural networks, traditional machine learning models allow systems to learn without being explicitly programmed.

Recommendation system machine learning does not require a neural network (or deep learning advancements like natural language processing or computer vision) to make accurate product recommendation technology for a user.

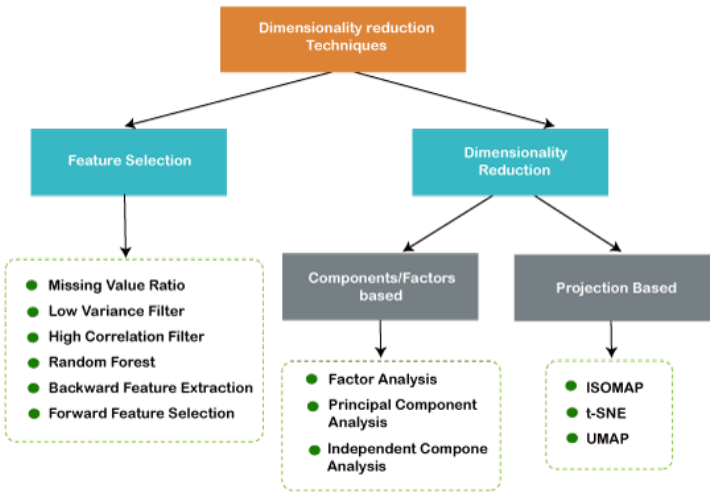
What is Dimensionality Reduction?

The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction.

A dataset contains a huge number of input features in various cases, which makes the predictive modeling task more complicated. Because it is very difficult to visualize or make predictions for the training dataset with a high number of features, for such cases, dimensionality reduction techniques are required to use.

Dimensionality reduction technique can be defined as, ***"It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information."*** These techniques are widely used in machine learning for obtaining a better fit predictive model while solving the classification and regression problems.

It is commonly used in the fields that deal with high-dimensional data, such as **speech recognition, signal processing, bioinformatics, etc. It can also be used for data visualization, noise reduction, cluster analysis**, etc.



The Curse of Dimensionality

Handling the high-dimensional data is very difficult in practice, commonly known as the *curse of dimensionality*. If the dimensionality of the input dataset increases, any machine learning algorithm and model becomes more complex. As the number of features increases, the number of samples also gets increased proportionally, and the chance of overfitting also increases. If the machine learning model is trained on high-dimensional data, it becomes overfitted and results in poor performance.

Hence, it is often required to reduce the number of features, which can be done with dimensionality reduction.

Benefits of applying Dimensionality Reduction

Some benefits of applying dimensionality reduction technique to the given dataset are given below:

- By reducing the dimensions of the features, the space required to store the dataset also gets reduced.
- Less Computation training time is required for reduced dimensions of features.
- Reduced dimensions of features of the dataset help in visualizing the data quickly.
- **I**t removes the redundant features (if present) by taking care of multicollinearity.

Disadvantages of dimensionality Reduction

There are also some disadvantages of applying the dimensionality reduction, which are given below:

- Some data may be lost due to dimensionality reduction.
- In the PCA dimensionality reduction technique, sometimes the principal components required to consider are unknown.

Approaches of Dimension Reduction

There are two ways to apply the dimension reduction technique, which are given below:

Feature Selection

Feature selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy. In other words, it is a way of selecting the optimal features from the input dataset.

Three methods are used for the feature selection:

1. Filters Methods

In this method, the dataset is filtered, and a subset that contains only the relevant features is taken. Some common techniques of filters method are:

- **Correlation**
- **Chi-Square Test**
- **ANOVA**
- **Information Gain, etc.**

2. Wrappers Methods

The wrapper method has the same goal as the filter method, but it takes a machine learning model for its evaluation. In this method, some features are fed to the ML model, and evaluate the performance. The performance decides whether to add those features or remove to increase the accuracy of the model. This method is more accurate than the filtering method but complex to work. Some common techniques of wrapper methods are:

- Forward Selection
- Backward Selection
- Bi-directional Elimination

3. Embedded Methods: Embedded methods check the different training iterations of the machine learning model and evaluate the importance of each feature. Some common techniques of Embedded methods are:

- **LASSO**
- **Elastic Net**
- **Ridge Regression, etc.**

- ## Feature Extraction:
- Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions. This approach is useful when we want to keep the whole information but use fewer resources while processing the information.

Some common feature extraction techniques are:

a. Principal Component Analysis
b. Linear Discriminant Analysis
c. Kernel PCA
d. Quadratic Discriminant Analysis

Common techniques of Dimensionality Reduction

a. **Principal Component Analysis**
b. **Backward Elimination**
c. **Forward Selection**
d. **Score comparison**
e. **Missing Value Ratio**
f. **Low Variance Filter**
g. **High Correlation Filter**
h. **Random Forest**
i. **Factor Analysis**
j. **Auto-Encoder**

Principal Component Analysis (PCA)

Principal Component Analysis is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modeling.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-

world applications of PCA are ***image processing, movie recommendation system, optimizing the power allocation in various communication channels.***

Backward Feature Elimination

The backward feature elimination technique is mainly used while developing Linear Regression or Logistic Regression model. Below steps are performed in this technique to reduce the dimensionality or in feature selection:

- In this technique, firstly, all the n variables of the given dataset are taken to train the model.
- The performance of the model is checked.
- Now we will remove one feature each time and train the model on n-1 features for n times, and will compute the performance of the model.
- We will check the variable that has made the smallest or no change in the performance of the model, and then we will drop that variable or features; after that, we will be left with n-1 features.
- Repeat the complete process until no feature can be dropped.

In this technique, by selecting the optimum performance of the model and maximum tolerable error rate, we can define the optimal number of features require for the machine learning algorithms.

Forward Feature Selection

Forward feature selection follows the inverse process of the backward elimination process. It means, in this technique, we don't eliminate the feature; instead, we will find the best features that can produce the highest increase in the performance of the model. Below steps are performed in this technique:

- We start with a single feature only, and progressively we will add each feature at a time.
- Here we will train the model on each feature separately.
- The feature with the best performance is selected.
- The process will be repeated until we get a significant increase in the performance of the model.

Missing Value Ratio

If a dataset has too many missing values, then we drop those variables as they do not carry much useful information. To perform this, we can set a threshold level, and if a variable has missing values more than that threshold, we will drop that variable. The higher the threshold value, the more efficient the reduction.

Low Variance Filter

As same as missing value ratio technique, data columns with some changes in the data have less information. Therefore, we need to calculate the variance of each variable, and all data columns with variance lower than a given threshold are dropped because low variance features will not affect the target variable.

High Correlation Filter

High Correlation refers to the case when two variables carry approximately similar information. Due to this factor, the performance of the model can be degraded. This correlation between the independent numerical variable gives the calculated value of the correlation coefficient. If this value is higher than the threshold value, we can remove one of the variables from the dataset. We can consider those variables or features that show a high correlation with the target variable.

Random Forest

Random Forest is a popular and very useful feature selection algorithm in machine learning. This algorithm contains an in-built feature importance package, so we do not need to program it separately. In this technique, we need to generate a large set of trees against the target variable, and with the help of usage statistics of each attribute, we need to find the subset of features.

Random forest algorithm takes only numerical variables, so we need to convert the input data into numeric data using **hot encoding**.

Factor Analysis

Factor analysis is a technique in which each variable is kept within a group according to the correlation with other variables, it means variables within a group can have a high correlation between themselves, but they have a low correlation with variables of other groups.

We can understand it by an example, such as if we have two variables Income and spend. These two variables have a high correlation, which means people with high income spends more, and vice versa. So, such variables are put into a group, and that group is known as the **factor**. The number of these factors will be reduced as compared to the original dimension of the dataset.

Auto-encoders

One of the popular methods of dimensionality reduction is auto-encoder, which is a type of ANN or artificial neural network, and its main aim is to copy the inputs to their outputs. In this, the input is compressed into latent-space representation, and output is occurred using this representation. It has mainly two parts:

- **Encoder:** The function of the encoder is to compress the input to form the latent-space representation.
- **Decoder:** The function of the decoder is to recreate the output from the latent-space representation.

Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices. It has some interesting algebraic properties and conveys important geometrical and theoretical insights about linear transformations. It also has some important applications in data science. In this article, I will try to explain the mathematical intuition behind SVD and its geometrical meaning.

Mathematics behind SVD

The SVD of mxn matrix A is given by the formula :

A = UWV^{T}

where:

- U: *mxn* matrix of the orthonormal eigenvectors of .
- V^T: transpose of a *nxn* matrix containing the orthonormal eigenvectors of A^{T}A.

- W: a *nxn* diagonal matrix of the singular values which are the square roots of the eigenvalues of .

Singular decomposition analysis(SVD)

$$C_{m\,x\,n} = U_{m\,x\,r} \times \sum_{r\,x\,r} \times V^{1}_{r\,x\,n}$$

Examples

- Find the SVD for the matrix A = $\begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$
- To calculate the SVD, First, we need to compute the singular values by finding eigenvalues of AA^{T}.

$$A \cdot A^T = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix} \cdot \begin{bmatrix} 3 & 2 \\ 2 & 3 \\ 2 & -2 \end{bmatrix} = \begin{bmatrix} 17 & 8 \\ 8 & 17 \end{bmatrix}$$

- The characteristic equation for the above matrix is:

$$W - \lambda I = 0$$
$$AA^T - \lambda I = 0$$

$$\lambda^2 - 34\lambda + 225 = 0$$

= (\lambda-25)(\lambda -9) so our singular values are: $\sigma_1 = 5 ; \sigma_2 = 3$

- Now we find the right singular vectors i.e orthonormal set of eigenvectors of A^TA. The eigenvalues of A^TA are 25, 9, and 0, and since A^TA is symmetric we know that the eigenvectors will be orthogonal.

For $\lambda = 25$,

$$A^{T}A - 25 \cdot I = \begin{bmatrix} -12 & 12& 2\\ 12 & -12 & -2\\ 2& -2 & -17 \end{bmatrix}$$

which can be row-reduces to :

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

A unit vector in the direction of it is:

$$v_1 = \begin{bmatrix} \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\\ 0 \end{bmatrix}$$

Similarly, for $\lambda = 9$, the eigenvector is:

$$v_2 = \begin{bmatrix} \frac{1}{\sqrt{18}} \\ \frac{-1}{\sqrt{18}} \\ \frac{4}{\sqrt{18}} \end{bmatrix}$$

For the 3rd eigenvector, we could use the property that it is perpendicular to v1 and v2 such that:

$$v_1^T v_3 = 0$$
$$v_2^T v_3 = 0$$

Solving the above equation to generate the third eigenvector

$$v_3 = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a \\ -a \\ -a/2 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ \frac{-2}{3} \\ \frac{-1}{3} \end{bmatrix}$$

Now, we calculate U using the formula $u_i = \frac{1}{\sigma} A v_i$ and this gives $U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$. Hence, our final SVD equation becomes:

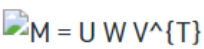
$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} &\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}& \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 5 & 0& 0 \\ 0 & 3& 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}& \frac{1}{\sqrt{2}} &0 \\ \frac{1}{\sqrt{18}}& \frac{-1}{\sqrt{18}} & \frac{4}{\sqrt{18}}\\ \frac{2}{3}&\frac{-2}{3} &\frac{1}{3} \end{bmatrix}$$

Applications

- **Calculation of Pseudo-inverse:** Pseudo inverse or Moore-Penrose inverse is the generalization of the matrix inverse that may not be invertible (such as low-rank matrices). If the matrix is invertible then its inverse will be equal to Pseudo inverse but pseudo inverse exists for the matrix that is not invertible. It is denoted by A^{+}.

Suppose, we need to calculate the pseudo-inverse of a matrix M:

Then, the SVD of M can be given as:



Multiply both sides by M^{-1}.

$$M^{-1}M = M^{-1}UWV^T$$

$$I = M^{-1}UWV^T$$

Multiply both side by V:

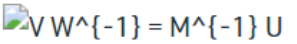
$$V = M^{-1}UWV^TV$$

$$V = M^{-1}UW$$

Multiply by W^{-1}. Since the W is the singular matrix, the inverse of W $= diag(a_1, a_2, a_3, ...a_n)^{-1}$ is

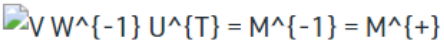
$= diag(1/a_1, 1/a_2, 1/a_3, ...1/a_n)$

$$VW^{-1} = M^{-1}UWW^{-1}$$



Multiply by U^T

$$VW^{-1}U^T = M^{-1}UU^T$$



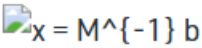
The above equation gives the pseudo-inverse.

- **Solving a set of Homogeneous Linear Equation (Mx =b):** if b=0, calculate SVD and take any column of V^T associated with a singular value (in *W*) equal to 0.

If $b \neq 0$, $Mx = b$

Multiply by M^{-1}

$$M^{-1}Mx = M^{-1}b$$

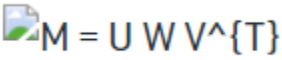


From the Pseudo-inverse, we know that $M^{-1} = VW^{-1}U^T$

Hence,

$$x = VW^{-1}U^Tb$$

- **Rank, Range, and Null space:**
    - The rank of matrix M can be calculated from SVD by the number of nonzero singular values.
    - The range of matrix M is The left singular vectors of U corresponding to the non-zero singular values.
    - The null space of matrix M is The right singular vectors of V corresponding to the zeroed singular values.



- **Curve Fitting Problem:** Singular value decomposition can be used to minimize the least square error. It uses the pseudo inverse to approximate it.
- Besides the above application, singular value decomposition and pseudo-inverse can also be used in Digital signal processing and image processing

Principal Component Analysis

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are ***image processing, movie recommendation system, optimizing the power allocation in various communication channels.*** It is a feature extraction technique, so it contains the important variables and drops the least important variable.

The PCA algorithm is based on some mathematical concepts such as:

- Variance and Covariance

- Eigenvalues and Eigen factors

Some common terms used in PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Principal Components in PCA

As described above, the transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:

- The principal component must be the linear combination of the original features.
- These components are orthogonal, i.e., the correlation between a pair of variables is zero.
- The importance of each component decreases when going to 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance.

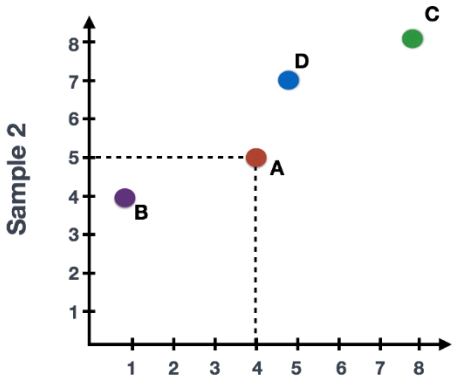
Steps for PCA algorithm

1. **Gettingthedataset**
   Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.

2. **Representingdataintoastructure**
   Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.
3. **Standardizingthedata**
   In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lowervariance.
   If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.
4. **CalculatingtheCovarianceofZ**
   To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.
5. **Calculating the Eigen Values and Eigen Vectors**
   Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.
6. **Sorting the Eigen Vectors**
   In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.
7. **Calculating the new features Or Principal Components**
   Here we will calculate the new features. To do this, we will multiply the P* matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrix is independent of each other.
8. **Remove less or unimportant features from the new dataset.**
   The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed out.

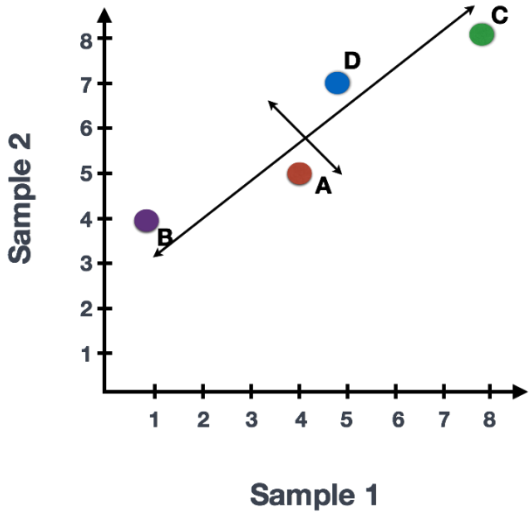
Applications of Principal Component Analysis

- PCA is mainly used as the dimensionality reduction technique in various AI applications such **as computer vision, image compression, etc.**
- It can also be used for finding hidden patterns if data has high dimensions. Some fields where PCA is used are Finance, data mining, Psychology, etc.
- If you had two samples and wanted to plot the counts of one sample versus another, you could plot the counts of one sample on the x-axis and the other sample on the y-axis as shown below:

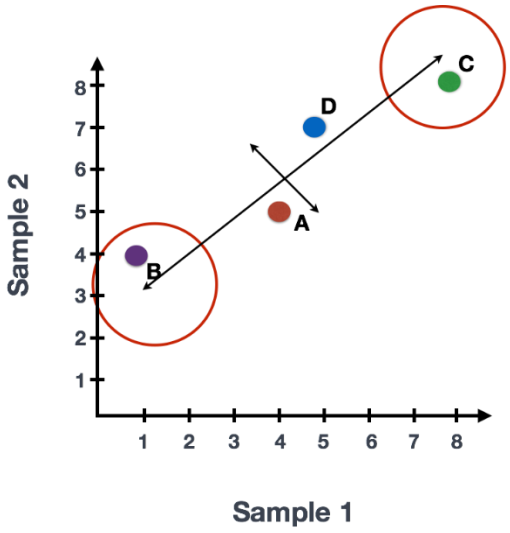


	Sample 1	Sample 2
Gene A	4	5
Gene B	1	4
Gene C	8	8
Gene D	5	7

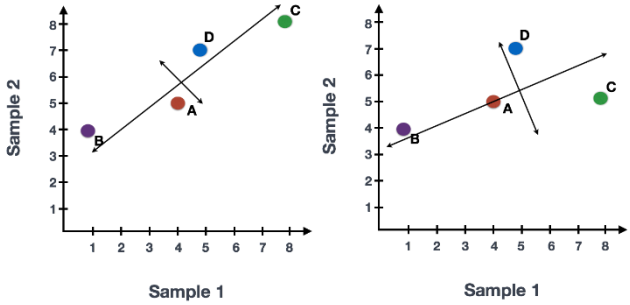
- You could draw a line through the data in the direction representing the most variation, which is on the diagonal in this example. The maximum variation in the data is between the two endpoints of this line.
- We also see the genes vary somewhat above and below the line. We could draw another line through the data representing the second most amount of variation in the data.



- The genes near the ends of the line, which would include those genes with the highest variation between samples (high expression in one sample and low expression in the other), have the greatest influence on the direction of the line.

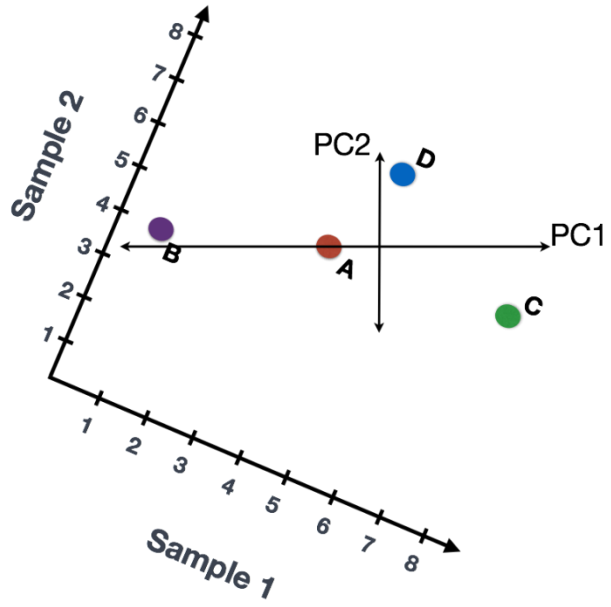


-
- For example, a small change in the value of *Gene C* would greatly change the direction of the line, whereas a small change in *Gene A* or *Gene D* would have little affect.

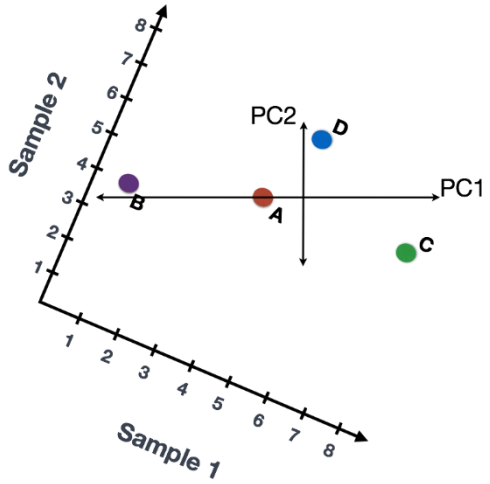


-
- We could just rotate the entire plot and view the lines representing the variation as left-to-right and up-and-down. We see most of the variation in the data is left-to-right; this is and the second most variation in the data is up-and-down. These axes that represent the variation are “Principal Components”, with PC1 representing the most variation in the data and PC2 representing the second most variation in the data.

- If we had three samples, then we would have an extra direction in which



  we could have variation. Therefore, if we have *N* samples we would have *N*-directions of variation or principal components.
-
- We could give quantitative scores to genes based on how much they influence PC1 and PC2. Genes with little influence would get scores near zero, while genes with more influence would receive larger scores. Genes on opposite ends of the lines have a large influence, so they would receive large scores, but with opposite signs.
-



	Sample 1	Sample 2	Influence on PC1	Influence on PC2
Gene A	4	5	-2	0.5
Gene B	1	4	-10	1
Gene C	8	8	8	-5
Gene D	5	7	1	6

Introduction

Recommendation systems are built to predict what users might like, especially when there are lots of choices available. They can explicitly offer those recommendations to users (e.g., Amazon or Netflix, the classic examples), or they might work behind the scenes to choose which content to surface without giving the user a choice.

Either way, the "why" is clear: they're critical for certain types of businesses because they can expose a user to content they may not have otherwise found or keep a user engaged for longer than they otherwise would have been. While building a simple recommendation system can be quite straightforward, the real challenge is to actually build one that works and where the business sees real uplift and value from its output.

Recommendation systems can be built using a variety of techniques, from simple (e.g., based only on other rated items from the same user) to extremely complex. Complex recommendation systems leverage a variety of different data sources (one challenge is using unstructured data, especially images, as the input) and machine learning (including deep learning) techniques. Thus, they are well suited for the world of artificial intelligence and more specifically unsupervised learning; as users continue to consume content and provide more data, these systems can be built to provide better and better recommendations.

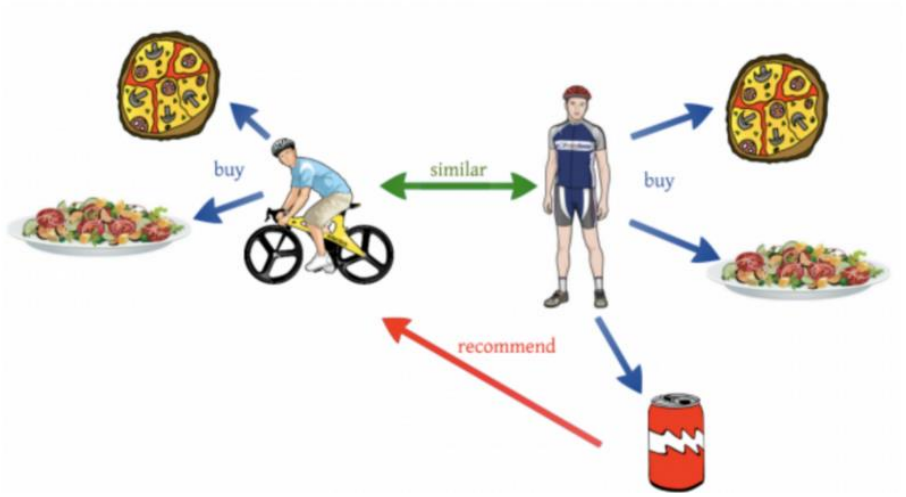
In this post and those to follow, I will be walking through the creation and training of recommendation systems, as I am currently working on this topic for Master Thesis. Part 1 provides a high-level overview of recommendation systems, how they are built, and how they can be used to improve businesses across industries.

The 2 Types of Recommendation System

There are two primary types of recommendation systems, each with different sub-types. Depending on goals, audience, the platform, and what you're recommending, these different approaches can be employed individually, though generally, the best results come from using them in combination:

1 — Collaborative Filtering

It primarily makes recommendations based on inputs or actions from other people (rather than only the user for whom a recommendation is being made).



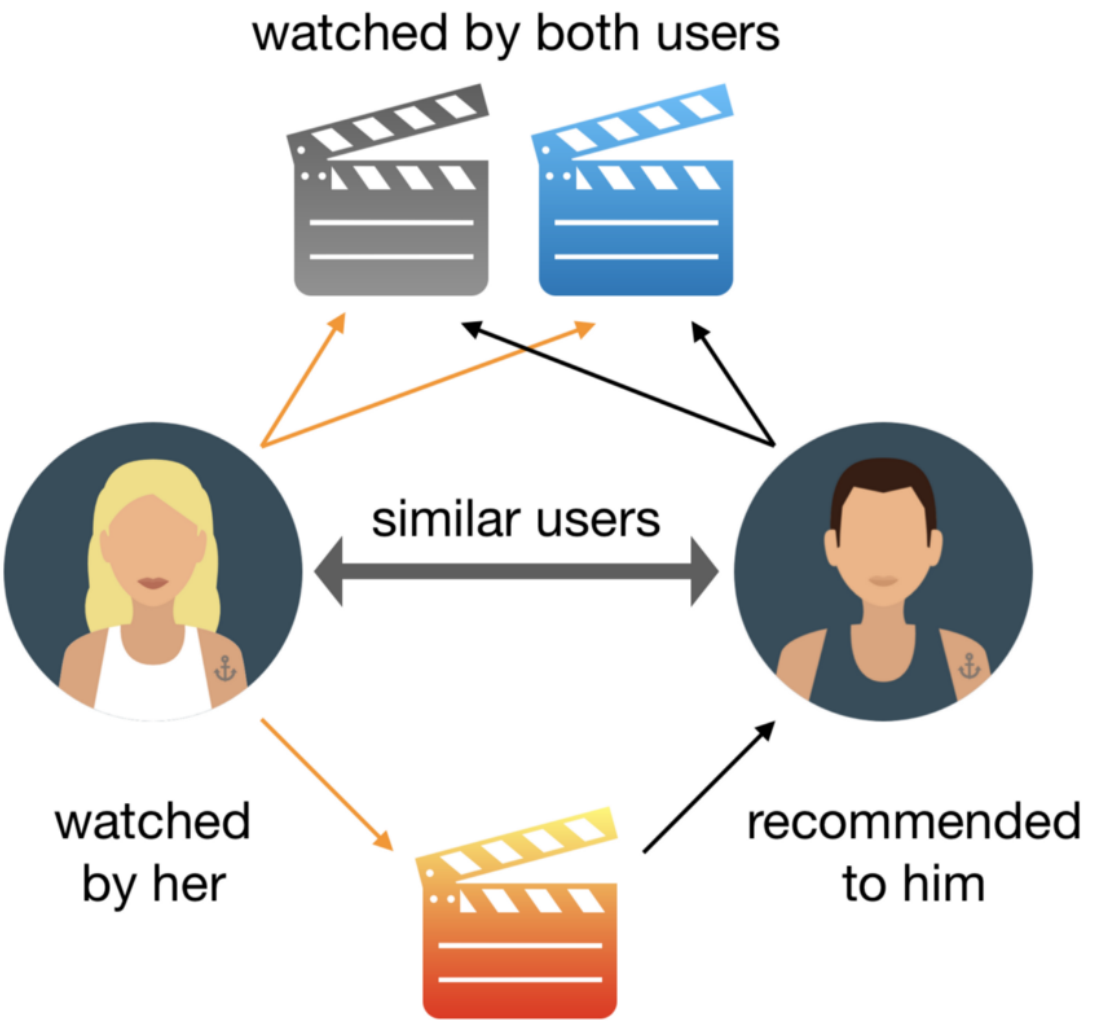
Variations on this type of recommendation system include:

- **By User Similarity:** This strategy involves creating user groups by comparing users' activities and providing recommendations that are popular among other members of the group. It is useful on sites with a strong but versatile audience to quickly provide recommendations for a user on which little information is available.
- **By Association:** This is a specific type of the one mentioned above, otherwise known as "Users who looked at X also looked at Y." Implementing this type of recommendation system is a matter of looking at purchasing sequences or purchasing groups, and showing similar content. This strategy is useful for capturing

recommendations related to naturally complementary content as well as at a certain point in the life of the user.

2 — Content-Based

Content-based systems make recommendations based on the user's purchase or consumption history and generally become more accurate



the more actions (inputs) the user takes.

More specific types of content-based recommendation systems include:

- **By Content Similarity:** As the most basic type of content-based recommendation system, this strategy involves recommending content that is close based on its metadata. This approach makes sense for catalogs with a lot of rich metadata and where traffic is low compared to the number of products in the catalog.
- **By Latent Factor Modeling:** Going one step further than the content similarity approach, the crux of this strategy is inferring individuals' inherent interests by assuming that previous choices are indicative of certain tastes or hobbies. Where the previous strategy is based on explicit, manually filled catalog metadata, this strategy hinges on discovering implicit relationships. This is done by using the history of users' larger interactions (e.g., movie watched, item purchased, etc.) to learn these tastes.
- **By Topic Modeling:** This is a variant of the Latent Factor Modeling strategy, whereby instead of considering users' larger actions, one would infer interests by analyzing unstructured text to detect particular topics of interest. It is particularly interesting for use cases with rich but unstructured textual information (such as news articles).
- **By Popular Content Promotion:** This involves highlighting product recommendations based on the product's intrinsic features that may make it interesting to a wide audience: price, feature, popularity, etc. This strategy can also take into account the freshness or age of the content and thus enable using the most

trendy content for recommendations. This is often used in cases where new content is the majority.

The 6 Steps to Build a Recommendation System

Building a successful and robust recommendation system can be relatively straightforward if you're following the basic steps to grow from raw data to a prediction. That being said, there are some particularities to consider when it comes to recommendation systems that often go overlooked and that, for the most efficient process and best predictions, are worth introducing (or reiterating).

This section will walk through the six fundamental steps to completing a data project in the context of building a recommendation system.

1 — Understand the Business

Extremely simple and critical but often overlooked, the first step in building a recommendation system is defining the goals and parameters of the project. This will most definitely involve discussions between and input from both the data team as well as business teams (which might be product managers, operations teams, even partnership or advertising teams, depending on your product).

Here are some specific topics to consider to understand the business need more deeply and kickstart the discussion between these teams:

- *What is the end goal of the project?* Is the idea to build a recommendation system to directly increase sales / achieve a higher average basket size / reduce browsing time and make a purchase happen faster / reduce the long tail of unconsumed content / improve user engagement time with your product?
- *Is a recommendation really necessary?* This is perhaps an obvious question, but since they can be expensive to build and maintain, it's worth asking. Can the business achieve its end goal by driving discovery via a static set of content instead (like staff/editor picks or most popular content)?

- *At what point will recommendations occur?* If recommendations make sense in multiple places (i.e., on a home screen upon first visiting the app or site as well as after purchasing or consuming content), will the same system be used in both places, or are the parameters and needs distinct for each?
- *What data is available on which to base recommendations?* At the time of recommendation, approximately what percentage of users are logged in (in which case there may be much more data available) vs. anonymous (which could complicate things for building the recommendation system)?
- *Are there product changes that must be made first?* If the team wants to build the recommendation system using more robust data, are there product changes that must be made first to identify users earlier (i.e., invite them to log in sooner), and if so, are they reasonable changes from a business perspective?
- *Should all content or products be treated equally?* That is, are there particular products or pieces of content that the business team wants to (or has to) promote aside from organic recommendations?
- *How can users with similar tastes be segmented?* In other words, if employing the model based on user similarity, how will you decide what makes users similar?

2 — Get the Data

The best recommendation systems use terabyte(s) of data. So when it comes to rounding up data to use for your recommendation systems, in general, the more the better. This can be difficult if users are unknown

when you're trying to make a recommendation for them — i.e., they're not logged in or, even more challenging, they're brand new. If you have a business where most users are unknown, you may need to rely on external data sources or general data not explicitly tied to preferences, like demographics, browsing history, etc.

When it comes to user preferences, there are two kinds of feedback: explicit and implicit.

- **Explicit user feedback** is anything that requires user effort, like leaving a review/rating or initiating a complaint or product return (often from customer relationship management, CRM, data).
- By contrast, **implicit user feedback** is information that can be gathered about a user's preferences without them actually specifying those preferences. For example, past purchase history, time spent looking at certain offers, products, or content, data from social networks, etc.

Good recommendation systems usually employ a combination of these types of feedback since there are advantages and disadvantages to each.

- Explicit feedback can be very clear: a user has literally stated their preferences, likes, or dislikes. But by the same token, it's inherently biased; a user doesn't know what he doesn't know (in other words, he might like something but has never tried it and therefore wouldn't list it as a preference or interact with that type of item or content normally).
- By contrast, implicit feedback is the opposite — it can reveal preferences that a user didn't — or wouldn't — otherwise, admit to in a profile (or perhaps their profile information is stale). On the other hand, implicit feedback can be more complicated to interpret; just because a user spent time on a given item doesn't mean that (s)he likes it, so it's best to rely on a combination of implicit signals to determine preference.

3 — Explore, Clean, and Augment the Data

One thing to consider when exploring and cleaning your data for a recommendation system, in particular, is changing user tastes. Depending on what you're recommending, the older reviews, actions, etc., may not be the most relevant on which to base a recommendation. Consider only looking at features that are more likely to represent the user's current tastes and removing older data that might no longer be relevant or adding a weight factor to give more importance to recent actions compared to older ones.

Datasets for recommendation systems can be challenging to work with because they are commonly high dimensional, but at the same time, it's also common that many of the features don't have any values, which can make clustering and outlier detection difficult.

4 — Predict the Ranking

Given the work done in the previous steps, you could have already built a recommendation system, simply by ranking those scores by users and you'll have products to recommend. This strategy doesn't use machine learning or a predictive element, but that's totally fine. For some use cases, this is sufficient.

But if you do want to build something more complex, there are lots of subtasks that can be done after users consume recommended content that can be used to further refine the system. There are several ways to leverage the hybrid approach to try for the highest-quality recommendations:

- Presenting recommendations from different types of systems together side-by-side.
- Maintaining multiple algorithms in parallel where the decision of which algorithm is preferred over another is itself subject to machine learning (e.g., multi-armed bandit).
- Using a pure machine learning approach to combine multiple

recommendation systems (logistic regression or other weighted regression methods). One specific example would be using a

weighted average of two (or more) recommendations using different techniques.

It's also possible that different models will work better in different parts of the product or website. For example, the homepage where the user has yet to take action vs. after the user has clicked or consumed content in some way.

5 — Visualize the Data

In the context of recommendation systems, visualization serves 2 primary purposes:

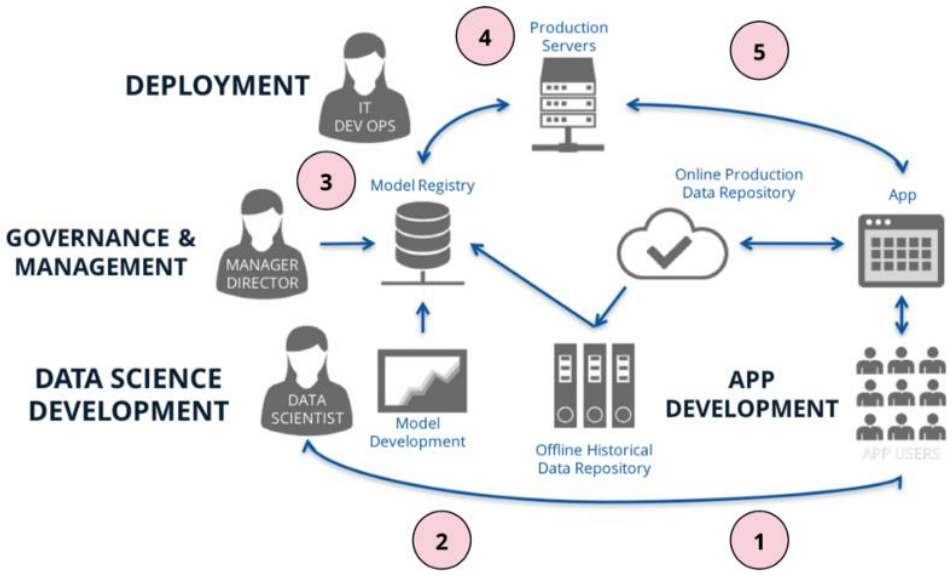
1. When still in the exploration phases, visualizations can help reveal things about the data set or give feedback on model performance that would otherwise be difficult to see.
2. After putting the recommendation system in place, visualizations can help convey useful information to the business or product teams (e.g., which content does well but isn't being discovered, similarities between users' tastes, content or products commonly consumed together, etc.) so they can make changes or decisions based on this information.

The primary issue with visualizing this type of data is the amount of data present, which can make it difficult to cut through the noise in a meaningful way. But by the same token, a good visualization will help make sense out of lots of data from which it would be otherwise difficult to derive meaningful insights.

6 — Iterate and Deploy Models

Recommendation systems that are working in a development environment or sandbox don't do any good. It's all about putting the system into production so that you can begin to see the effect on the business goals you've laid out in the beginning.

Additionally, keep in mind that the more data you have with which to feed the recommendation system, the better it can become. So with this type of data project perhaps more so than others, it's critical to evaluate performance and continue to fine-tune, like adding new data sources to see if they have a positive effect.



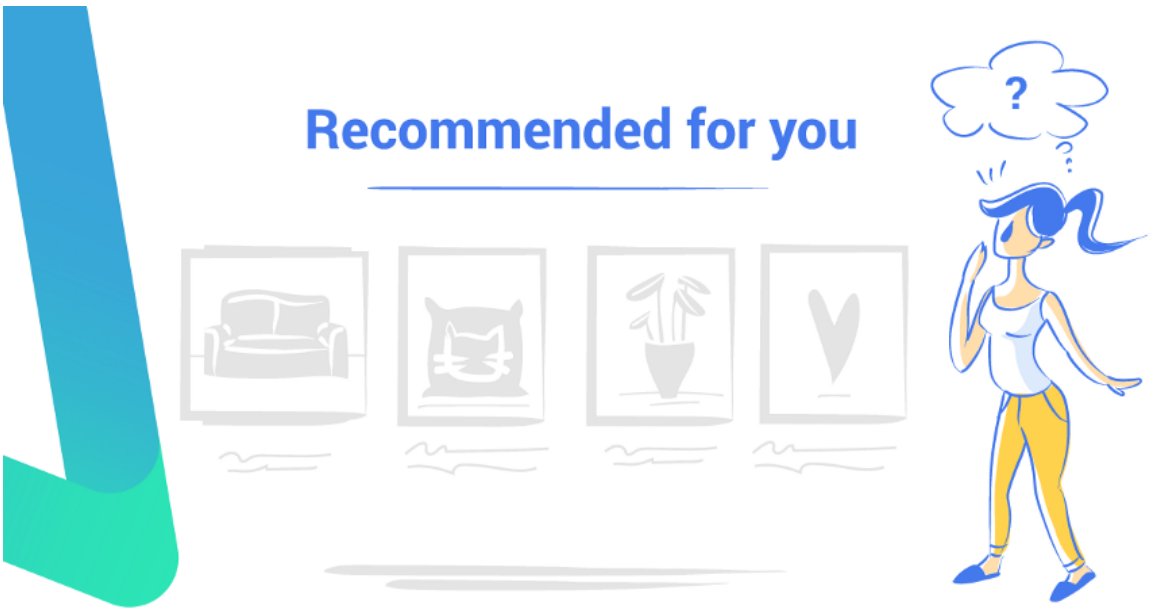
In fact, making sure your recommendation system is built to adapt and evolve by regularly monitoring its performance is one of the most important parts of the process — a recommendation system that isn't properly adjusting to tastes or new data over time likely will not help you ultimately achieve your initial project goal, even if the system performed well at first. Building a feedback loop to understand whether or not users care about recommendations will be helpful and provide a good metric for making refinements and decisions going forward.

If recommendations are core to your business, constantly trying new things and evolving the initial model you've created will be an ongoing task; recommendation systems are not something you can create and cast aside.

Challenges

It's important to create a recommendation system that will **scale** with the amount of data you have. If it's built for a limited dataset and that dataset grows, computation costs grow exponentially, and the system will be unable to handle the amount of data. To avoid having to rebuild your recommendation system later on, you must ensure from the beginning it is built to scale to expected data volumes.

It's also possible that after spending time, energy, and resources on building a recommendation system (and even after having enough data and good initial results) that the recommendation system only makes very **obvious recommendations**. The crux of avoiding this pitfall really harkens back to the first of the seven steps: understand the business need. If there isn't enough of a content long-tail or no need for the system, perhaps you need to reconsider the need to build a recommendation system in the first place.



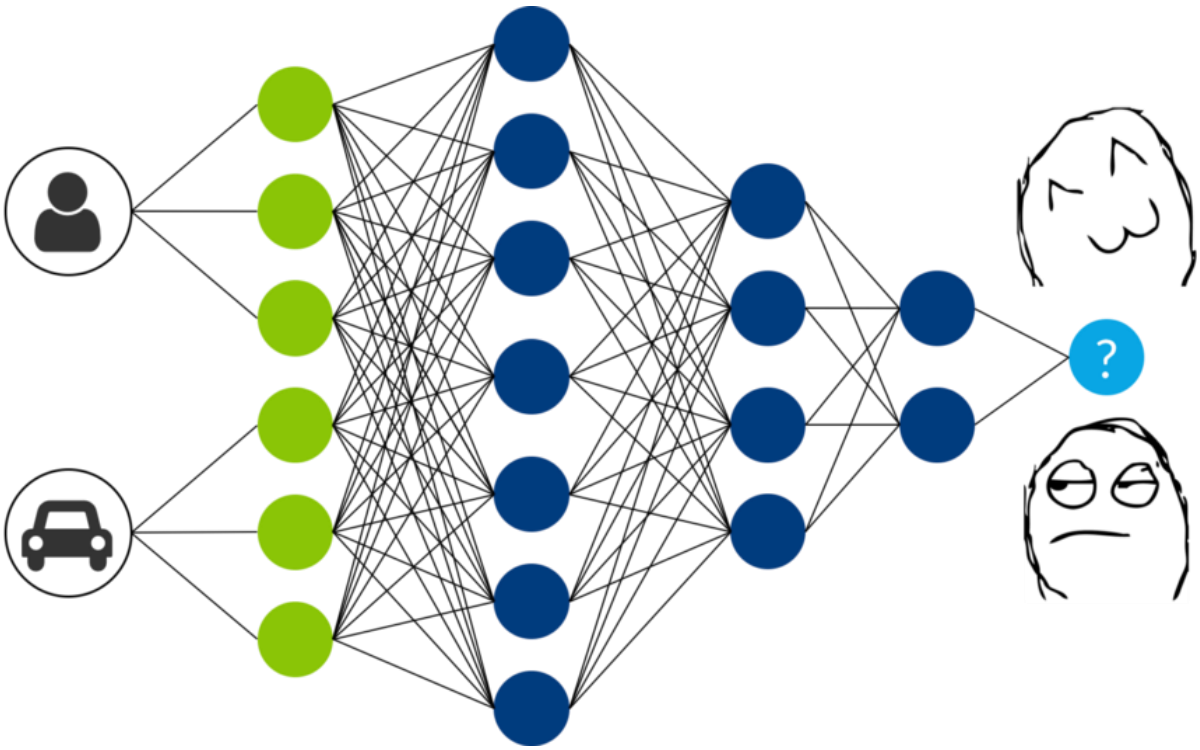
Finally, people's tastes don't stay static over time, and if a recommendation system isn't built to consider this fact, it may never be as accurate as it could be. Similarly, there is a risk of building a recommendation system that doesn't get better over time. As users continue to consume content and more data is available, your recommendation system should learn more about users and adapt to their tastes. A recommendation system **not agile enough** to continue to adapt can quickly become obsolete and won't serve its purpose.

Future Work

Basic recommendation systems have been around for quite some time, though they continue to get more complex and have been perfected by retail and content giants. But what's next? What are the latest trends

and developments that businesses should consider if they are looking to develop a truly cutting-edge system?

Context-aware recommendation systems represent an emerging area of experimentation and research, aiming to provide even more precise content given the context of the user in a particular moment in time. For example, is the user at home, or on the go? Using a larger or smaller screen? Is it morning or night? Given the data available on a certain user, context-aware systems may be able to provide recommendations a user is more likely to take in those scenarios.



Deep learning is already in use by some of the biggest and most powerful recommendation systems in the world (like YouTube and Spotify). But as the amount of data continues to skyrocket and more

businesses find themselves up against a huge corpus of content and struggling to scale, deep learning will become the de facto methodology for not only recommendation systems but all learning problems.

Solving the cold-start problem is also something that cutting-edge researchers are starting to look at so that recommendations can be made for items on which there is little data. This is a critically important area for businesses with lots of turnover in content to examine so that they can successfully push items that will sell well (even before they know how that item will perform).

Conclusion

Recommendation systems can be an effective way to expose users to content they may not have otherwise found, which in turn can forward larger business goals like increasing sales, advertising revenues, or user engagement. But there are a few key points to find success with recommendation systems. Namely, recommendation systems should be, above all, necessary.

Building a complex system that requires experienced staff and ongoing maintenance when a simpler solution will do is a waste of data team resources that could be spent elsewhere for more impact. The challenge lies in building a system that will actually have a business impact; building the system in and of itself shouldn’t be the end goal.

social network graph

A **social network graph** is a graph where the nodes represent people and the lines between nodes, called edges, represent social connections between them, such as friendship or working together on a project. These graphs can be either undirected or directed. For instance, Facebook can be described with an undirected graph since the friendship is bidirectional, Alice and Bob being friends is the same as Bob and Alice being friends. On the other hand, Twitter can be described with a directed graph: Alice can follow Bob without Bob following Alice.

Social networks tend to have characteristic network properties. For instance, there tends to be a short distance between any two nodes (as in the famous six degrees of separation study where everyone in the world is at most six degrees away from any other), and a tendency to form "triangles" (if Alice is friends with Bob and Carol, Bob and Carol are more likely to be friends with each other.)

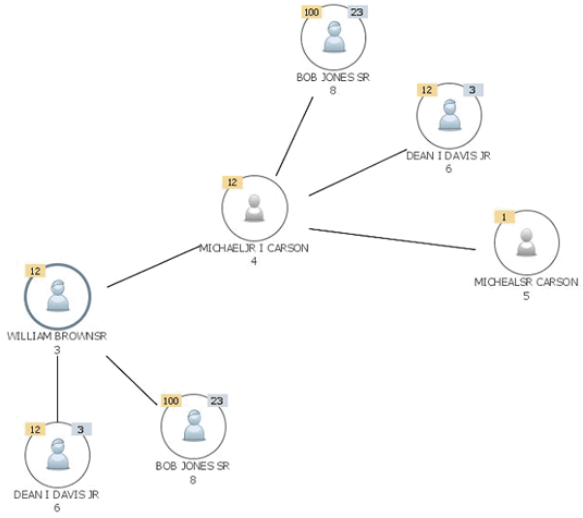
Social networks are important to social scientists interested in how people interact as well as companies trying to target consumers for advertising. For instance if advertisers connect up three people as friends, co-workers, or family members, and two of them buy the advertiser's product, then they may choose to spend more in advertising to the third hold-out, on the belief that this target has a high propensity to buy their product.

Social scientists can also use social networks to model the way things made by people connect. Pages on the internet and the links between them form a social network in much the same way as people form networks with other people. Also, counter-intelligence agencies have used cell-phone data and calls to map out terrorist cells.

The image to the right shows the connections between different physicians who co-author papers on hepatitis C, for instance showing that two people who coauthored one paper, also mutually coauthored separate papers with another physician.

Tips for using the Social Network graph

- Use the **Show remaining related Entities** right-click option to expand the related entities for one or more entities on the graph. Each expansion creates another relationship cluster. Look for patterns between the clusters.
- If multiple relationship clusters are graphed, try zooming out to look for the bigger patterns and context in the clusters. For example, if a particular entity shows up in every cluster or many clusters, then that entity might be a big influencer within a particular sphere. Or that entity might be key to connecting multiple relationship clusters.
- Use the **Attribute Explorer** to see which attributes link the related entities. Select a particular attribute row to highlight every entity on the graph that shares that attribute. The value in the **Entities** column can show you which attributes are shared by the most entities.



What is a Social Network?

When we think of a social network, we think of Facebook, Twitter, Google+, or another website that is called a "social network," and indeed this kind of network is representative of the broader class of networks called "social." The essential characteristics of a social network are:

1. There is a collection of entities that participate in the network. Typically, these entities are people, but they could be something else entirely. We shall discuss some other examples in

2. There is at least one relationship between entities of the network. On Facebook or its ilk, this relationship is called friends. Sometimes the relationship is all-or-nothing; two people are either friends or they are not. However, in other examples of social networks, the relationship has a degree. This degree could be discrete; e.g., friends, family, acquaintances, or none as in Google+. It could be a real number; an example would be the fraction of the average day that two people spend talking to each other.

3. There is an assumption of nonrandomness or locality. This condition is the hardest to formalize, but the intuition is that relationships tend to cluster. That is, if entity A is related to both B and C, then there is a higher probability than average that B and C are related. 10.1.2 Social Networks as Graphs Social networks are naturally modeled as graphs, which we sometimes refer to as a

social graph. The entities are the nodes, and an edge connects two nodes if the nodes are related by the relationship that characterizes the network. If there is a degree associated with the relationship, this degree is represented by labeling the edges. Often, social graphs are undirected, as for the Facebook friends graph. But they can be directed graphs, as for example the graphs of followers on Twitter or Google+.

Example 10.1 : Figure 10.1 is an example of a tiny social network. The entities are the nodes A through G. The relationship, which we might think of as "friends," is represented by the edges. For instance, B is friends with A, C, and D. Is this graph really typical of a social network, in the sense that it exhibits locality of relationships? First, note that the graph has nine edges out of the 10.1. SOCIAL NETWORKS AS GRAPHS 357 A B D E G F C Figure 10.1: Example of a small social network 7 2 = 21 pairs of nodes that could have had an edge between them. Suppose X, Y , and Z are nodes of Fig. 10.1, with edges between X and Y and also between X and Z. What would we expect the probability of an edge between Y and Z to be? If the graph were large, that probability would be very close to the fraction of the pairs of nodes that have edges between

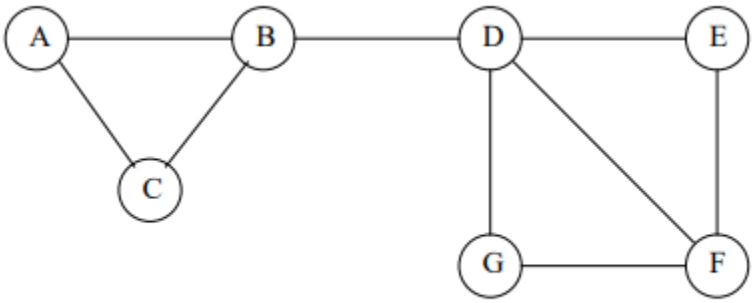


Figure 10.1: Example of a small social network

Varieties of Social Networks

There are many examples of social networks other than "friends" networks. Here, let us enumerate some of the other examples of networks that also exhibit locality of relationships.

Telephone Networks

Here the nodes represent phone numbers, which are really individuals. There is an edge between two nodes if a call has been placed between those phones in some fixed period of time, such as last month, or "ever." The edges could be weighted by the number of calls made between these phones during the period. Communities in a telephone network will form from groups of people that communicate frequently: groups of friends, members of a club, or people working at the same company, for example.

Email Networks

The nodes represent email addresses, which are again individuals. An edge represents the fact that there was at least one email in at least one direction between the two addresses. Alternatively, we may only place an edge if there were emails in both directions. In that way, we avoid viewing spammers as "friends" with all their victims. Another approach is to label edges as weak or strong. Strong edges represent communication in both directions, while weak edges indicate that the communication was in one direction only. The communities seen in email networks come from the same sorts of groupings we mentioned in connection with telephone networks. A similar sort of network involves people who text other people through their cell phones.

Collaboration Networks

Nodes represent individuals who have published research papers. There is an edge between two individuals who published one or more papers jointly. Optionally, we can label edges by the number of joint publications. The communities in this network are authors working on a particular topic. An alternative view of the same data is as a graph in which the nodes are papers. Two papers are connected by an edge if they have at least one author in common. Now, we form communities that are collections of papers on the same topic. There are several other kinds of data that form two networks in a similar way. For example, we can look at the people who edit Wikipedia articles and the articles

that they edit. Two editors are connected if they have edited an article in common. The communities are groups of editors that are interested in the same subject. Dually, we can build a network of articles, and connect articles if they have been edited by the same person. Here, we get communities of articles on similar or related subjects.

Other Examples of Social Graphs Many other phenomena give rise to graphs that look something like social graphs, especially exhibiting locality. Examples include: information networks (documents, web graphs, patents), infrastructure networks (roads, planes, water pipes, powergrids), biological networks (genes, proteins, food-webs of animals eating each other), as well as other types, like product co-purchasing networks.

Graphs With Several Node Types There are other social phenomena that involve entities of different types. We just discussed under the heading of "collaboration networks," several kinds of graphs that are really formed from two types of nodes. Authorship networks can be seen to have author nodes and paper nodes. In the discussion above, we built two social networks by eliminating the nodes of one of the two types, but we do not have to do that. We can rather think of the structure as a whole. For a more complex example, users at a site like del.icio.us place tags on Web pages. There are thus three different kinds of entities: users, tags, and pages. We might think that users were somehow connected if they tended to use the same tags frequently, or if they tended to tag the same pages. Similarly, tags could be considered related if they appeared on the same pages or were used by the same users, and pages could be considered similar if they had many of the same tags or were tagged by many of the same users. The natural way to represent such information is as a k-partite graph for some $k > 1$. We met bipartite graphs, the case $k = 2$, in Section 8.3. In general, a k-partite graph consists of k disjoint sets of nodes, with no edges between nodes of the same set.

Example 10.2 : Figure 10.2 is an example of a tripartite graph (the case $k = 3$ of a k-partite graph). There are three sets of nodes, which we may think of as users {U1, U2}, tags {T1, T2, T3, T4}, and Web pages {W1, W2, W3}. Notice that all edges connect nodes from two different sets. We may assume this graph represents information about the three kinds of entities. For example, the edge (U1, T2) means that user U1 has placed the tag T2 on at least one page. Note that the graph does not tell us a detail that

could be important: who placed which tag on which page? To represent such ternary information would require a more complex representation, such as a database relation with three columns corresponding to users, tags, and pages.

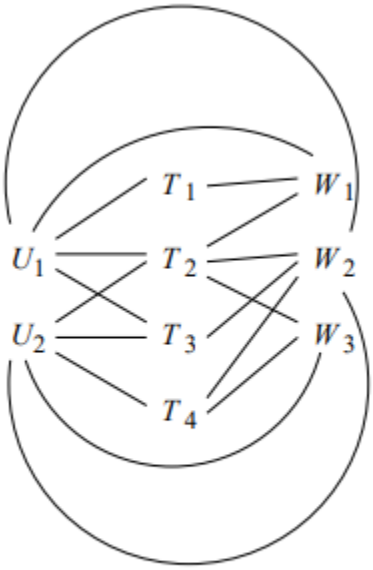


Figure 10.2: A tripartite graph representing users, tags, and Web pages

Clustering of Social-Network Graphs

An important aspect of social networks is that they contain communities of entities that are connected by many edges. These typically correspond to groups of friends at school or groups of researchers interested in the same topic, for example. In this section, we shall consider clustering of the graph as a way to identify communities. It turns out that the techniques we learned in Chapter 7 are generally unsuitable for the problem of clustering social-network graphs.

Distance Measures for Social-Network Graphs

If we were to apply standard clustering techniques to a social-network graph, our first step would be to define a distance measure. When the edges of the graph have labels, these labels might be usable as a distance measure, depending on what they represented. But when the edges are unlabeled, as in a “friends” graph, there is not much we can do to define a suitable distance. Our first instinct is to assume that nodes are close if they have an edge between them and distant if

not. Thus, we could say that the distance d(x, y) is 0 if there is an edge (x, y) and 1 if there is no such edge. We could use any other two values, such as 1 and ∞, as long as the distance is closer when there is an edge. Neither of these two-valued "distance measures" – 0 and 1 or 1 and ∞ – is a true distance measure. The reason is that they violate the triangle inequality when there are three nodes, with two edges between them. That is, if there are edges (A, B) and (B, C), but no edge (A, C), then the distance from A to C exceeds the sum of the distances from A to B to C. We could fix this problem by using, say, distance 1 for an edge and distance 1.5 for a missing edge. But the problem with two-valued distance functions is not limited to the triangle inequality, as we shall see in the next section.

Applying Standard Clustering Methods

Recall from Section 7.1.2 that there are two general approaches to clustering: hierarchical (agglomerative) and point-assignment. Let us consider how each of these would work on a social-network graph. First, consider the hierarchical methods covered in Section 7.2. In particular, suppose we use as the intercluster distance the minimum distance between nodes of the two clusters. Hierarchical clustering of a social-network graph starts by combining some two nodes that are connected by an edge. Successively, edges that are not between two nodes of the same cluster would be chosen randomly to combine the clusters to which their two nodes belong. The choices would be random, because all distances represented by an edge are the same.

Example 10.3 : Consider again the graph of Fig. 10.1, repeated here as Fig. 10.3. First, let us agree on what the communities are. At the highest level, it appears that there are two communities {A, B, C} and {D, E, F, G}. However, we could also view {D, E, F} and {D, F, G} as two subcommunities of {D, E, F, G}; these two subcommunities overlap in two of their members, and thus could never be identified by a pure clustering algorithm. Finally, we could consider each pair of individuals that are connected by an edge as a community of size 2, although such communities are uninteresting.

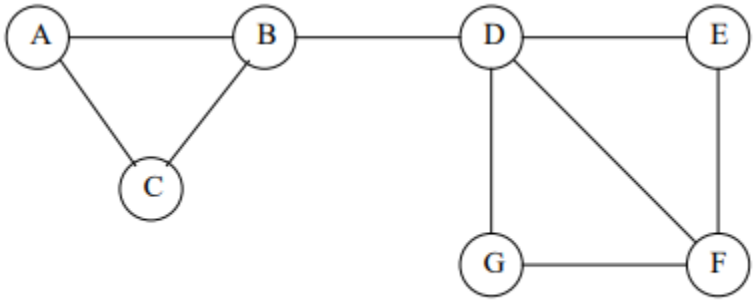


Figure 10.3: Repeat of Fig. 10.1

The problem with hierarchical clustering of a graph like that of Fig. 10.3 is that at some point we are likely to chose to combine B and D, even though they surely belong in different clusters. The reason we are likely to combine B and D is that D, and any cluster containing it, is as close to B and any cluster containing it, as A and C are to B. There is even a 1/9 probability that the first thing we do is to combine B and D into one cluster. There are things we can do to reduce the probability of error. We can run hierarchical clustering several times and pick the run that gives the most coherent clusters. We can use a more sophisticated method for measuring the distance between clusters of more than one node, as discussed in Section 7.2.3. But no matter what we do, in a large graph with many communities there is a significant chance that in the initial phases we shall use some edges that connect two nodes that do not belong together in any large community. ✷ Now, consider a point-assignment approach to clustering social networks. Again, the fact that all edges are at the same distance will introduce a number of random factors that will lead to some nodes being assigned to the wrong cluster. An example should illustrate the point. Example 10.4 : Suppose we try a k-means approach to clustering Fig. 10.3. As we want two clusters, we pick k = 2. If we pick two starting nodes at random, they might both be in the same cluster. If, as suggested in Section 7.3.2, we start with one randomly chosen node and then pick another as far away as possible, we don't do much better; we could thereby pick any pair of nodes not connected by an edge, e.g., E and G in Fig. 10.3. However, suppose we do get two suitable starting nodes, such as B and F. We shall then assign A and C to the cluster of B and assign E and G to the cluster

of F. But D is as close to B as it is to F, so it could go either way, even though it is "obvious" that D belongs with F. If the decision about where to place D is deferred until we have assigned some other nodes to the clusters, then we shall probably make the right decision. For instance, if we assign a node to the cluster with the shortest average distance to all the nodes of the cluster, then D should be assigned to the cluster of F, as long as we do not try to place D before any other nodes are assigned. However, in large graphs, we shall surely make mistakes on some of the first nodes we place.

Getting Started with Community Detection in Graphs and Networks

Introduction

The word "community" has entered mainstream conversations around the world this year thanks in no large part to the ongoing coronavirus pandemic. Given my experience and interest in graphs and graph theory in general, I wanted to understand and explore how I could leverage that in terms of a community.

That's how I landed on the topic of community detection. While the current pandemic is beyond the scope of this article, I feel community detection is quite a nuanced topic everyone should at least know a bit about. This is an excellent extension of graph theory – the topic taking the data science community by storm there days.

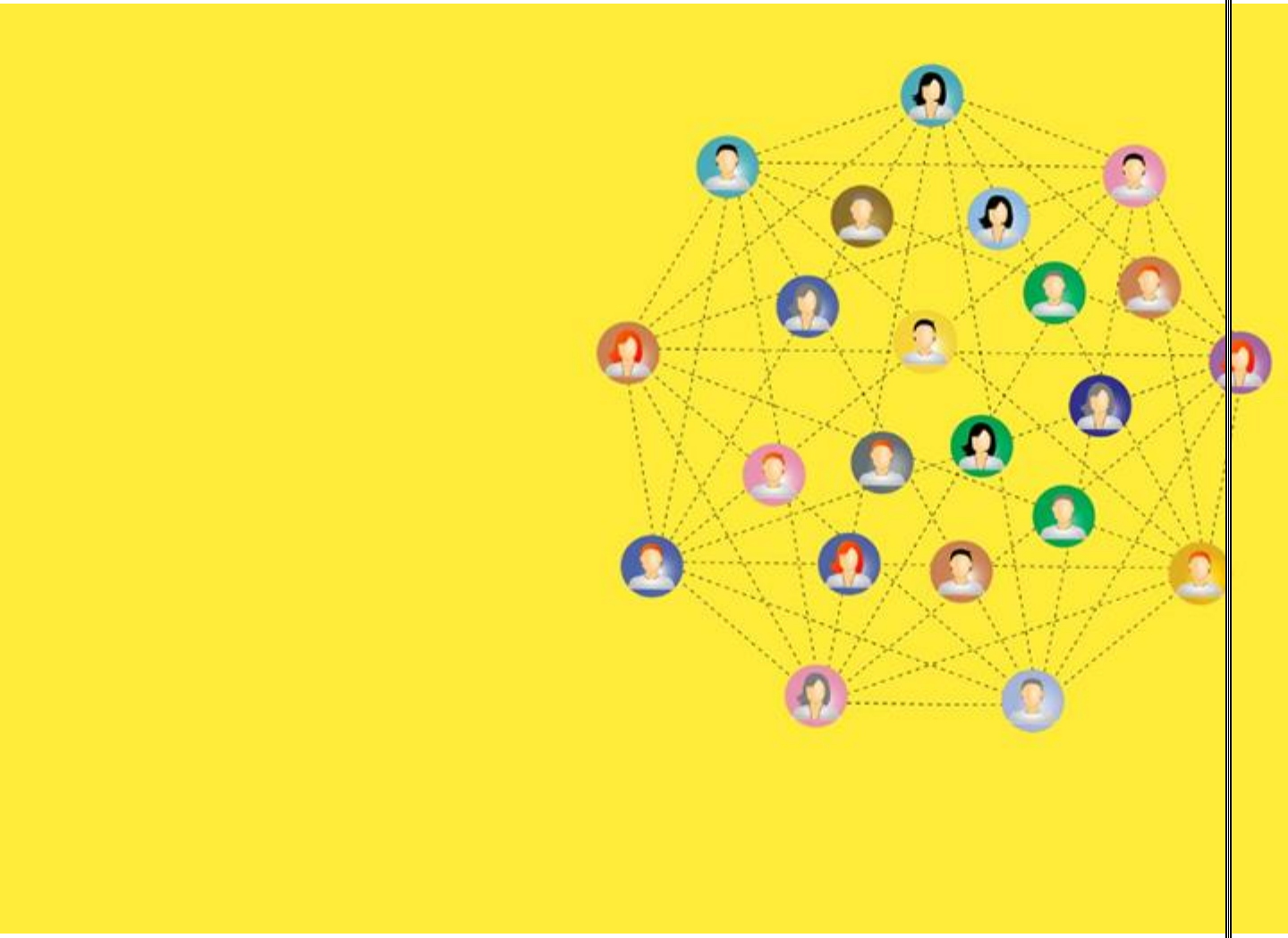
My focus in this article is to help you get started with community detection. This will, of course, rely on an underlying understanding of graph theory as well (link to learn about it is provided below). We'll talk about community detection in

detail, including the Girvan-Newman algorithm and how to implement it in Python. There's a lot to learn so let's get the ball rolling!

Introduction

The word "community" has entered mainstream conversations around the world this year thanks in no large part to the ongoing coronavirus pandemic. Given my experience and interest in graphs and graph theory in general, I wanted to understand and explore how I could leverage that in terms of a community.

That's how I landed on the topic of community detection. While the current pandemic is beyond the scope of this article, I feel community detection is quite a nuanced topic everyone should at least know a bit about. This is an excellent extension of graph theory – the topic taking the data science community by storm there days.



My focus in this article is to help you get started with community detection. This will, of course, rely on an underlying understanding of graph theory as well (link to learn about it is provided below). We'll talk about community detection in detail, including the Girvan-Newman algorithm and how to implement it in Python. There's a lot to learn so let's get the ball rolling!

This article, as you might have surmised already, assumes familiarity with graph theory. If you're new to this topic, here's a handy starting point:

- Let’s Think in Graphs: Introduction to Graph Theory and its Applications using Python

Table of Contents

What is a Community?

Let’s first put a definition to the word “community”. It’s a broad term, right? We need to define what exactly it means in the context of this article.

A community, with respect to graphs, can be defined as a subset of nodes that are densely connected to each other and loosely connected to the nodes in the other communities in the same graph.

Let me break that down using an example. Think about social media platforms such as Facebook, Instagram, or Twitter, where we try to connect with other people. Eventually, after a while, we end up being connected with people belonging to different social circles. These social circles can be a group of relatives, school mates, colleagues, etc.

These social circles are nothing but communities!

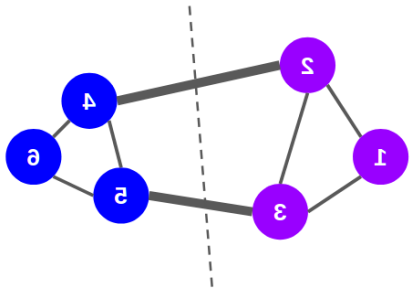
What is Community Detection?

Detecting communities in a network is one of the most important tasks in network analysis. In a large scale network, such as an online social network, we could have millions of nodes and edges. Detecting communities in such networks becomes a herculean task.

Therefore, we need community detection algorithms that can partition the network into multiple communities.

here are primarily two types of methods for detecting communities in graphs:

- (a) Agglomerative Methods
- (b) Divisive Methods



(a) Agglomerative Methods

In agglomerative methods, we start with an empty graph that consists of nodes of the original graph but no edges. Next, the edges are added one-by-one to the graph, starting from “stronger” to “weaker” edges. This strength of the edge, or the weight of the edge, can be calculated in different ways.

Don’t worry – we will discuss this later. Keep in mind that new communities are formed in the consecutive steps of the algorithm.

(b) Divisive Methods

In divisive methods, we go the other way round. We start with the complete graph and take off the edges iteratively. The edge with the highest weight is removed first. At every step, the edge-weight calculation is repeated, since the weight of the remaining edges changes after an edge is removed. After a certain number of steps, **we get clusters of densely connected nodes.**

In this article, we will cover the Girvan-Newman algorithm – an example of the divisive method.

Girvan-Newman Algorithm for Community Detection

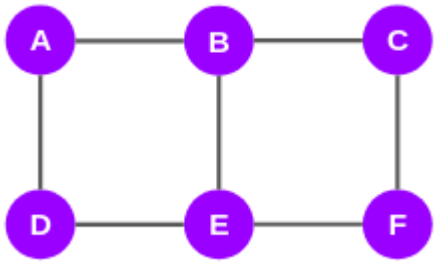
Under the Girvan-Newman algorithm, the communities in a graph are discovered by iteratively removing the edges of the graph, based on the edge betweenness centrality value.

The edge with the highest edge betweenness is removed first. We will cover this algorithm later in the article, but first, let's understand the concept of "edge betweenness centrality".

Edge Betweenness Centrality (EBC)

The edge betweenness centrality (EBC) can be defined as the number of shortest paths that pass through an edge in a network. Each and every edge is given an EBC score based on the shortest paths among all the nodes in the graph.

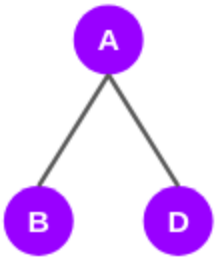
With respect to graphs and networks, the shortest path means the path between any two nodes covering the least amount of distance. Let's take an example to find how EBC scores are calculated. Consider this graph below:



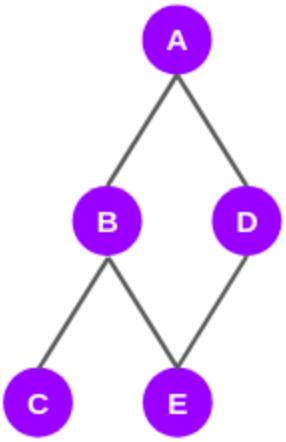
It has 6 nodes and 7 edges, right? Now, we will try to find the EBC scores for all the edges in this graph. Note that it is an iterative process and I've given an outline of it here:

- We will take one node at a time and plot the shortest paths to the other nodes from the selected node
- Based on the shortest paths, we will compute the EBC scores for all the edges
- We need to repeat this process for every node in the graph. As you can see, we have 6 nodes in the graph above. Therefore, there will be 6 iterations of this process
- This means every edge will get 6 scores. These scores will be added edge-wise
- Finally, the total score of each edge will be divided by 2 to get the EBC score

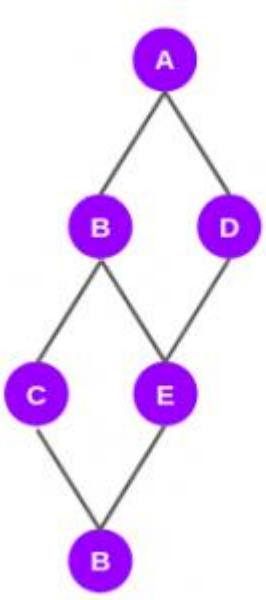
Now, let's start with node A. The directly connected nodes to node A are nodes B and D. So, the shortest paths to B and D from A are AB and AD respectively:



It turns out that the shortest paths to nodes C and E from A go through B and D:



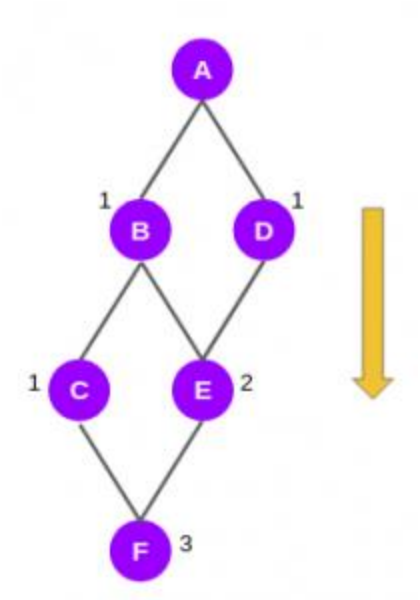
The shortest paths to the last node F from node A, pass through nodes B, D, C, and E:



The graph above depicts only the shortest paths from node A to all the other nodes. Now we will see how edges are scored.

Before giving scores to the edges, we will assign a score to the nodes in the shortest-path-graph. To assign these scores, we will have to traverse the graph from the root node, i.e., node A to the last node (node F).

Assigning Scores to Nodes



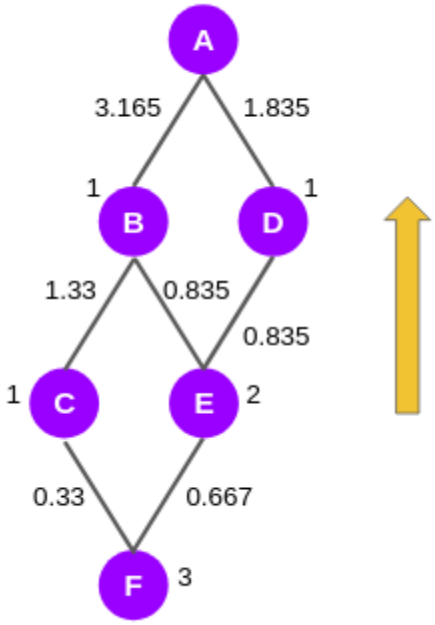
As you can see in the graph above, nodes B and D have been given a score of 1 each. This is because the shortest path to either node from node A is only one. For the very same reason, node C has been given a score of 1 as there is only one shortest path from node A to node C.

Moving on to node E. It is connected to node A through two shortest paths, ABE and ADE. Hence, it gets a score of 2.

The last node F is connected to A through three shortest paths — ABCF, ABEF, and ADEF. So, it gets a score of 3.

Computing Scores for Edges

Next, we will proceed with computing scores for the edges. Here we will move in the backward direction, from node F to node A:



FC = 1/3 = 0.33
FE = 2/3 = 0.667

CB = 1 + 0.33 = 1.33
EB = (1 + 0.667)/2 = 0.835
ED = (1 + 0.667)/2 = 0.835

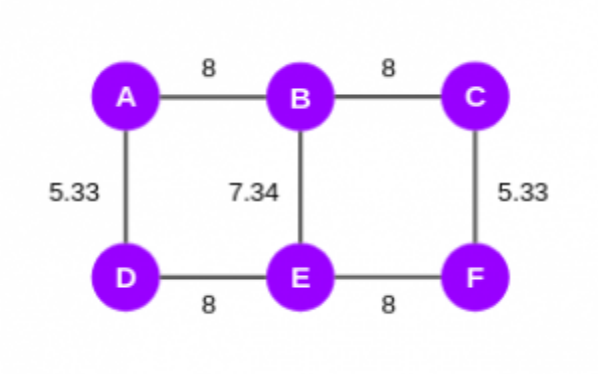
BA = (1 + 1.33 + 0.835)/1 = 3.165
DA = (1 + 0.835)/1 = 1.835

We first compute the score for the edges FC and FE. As you can see, the edge score for edge FC is the ratio of the node scores of C and F, i.e. 1/3 or 0.33. Similarly, for FE the edge score is 2/3.

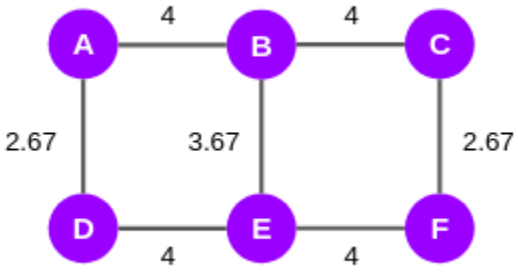
Now we have to calculate the edge score for the edges CB, EB, and ED. According to the Girvan-Newman algorithm, from this level onwards, every node will have a default value of 1 and the edge scores computed in the previous step will be added to this value.

So, the edge score of CB is (1 + 0.33)/1. Similarly, edge score EB or ED is (1 + 0.667)/2. Then we move to the next level to calculate the edge scores for BA and DA.

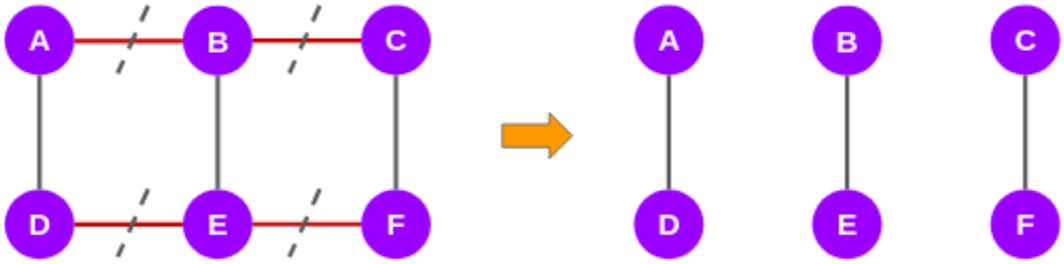
So far, we have computed the edge scores of the shortest paths with respect to node A. We will have to repeat the same steps again from the other remaining five nodes. In the end, we will get a set of six scores for all the edges in the network. We will add these scores and assign them to the original graph as shown below:



Since it is an undirected graph, we will divide these scores by two and finally, we will get the EBC scores:



According to the Girvan-Newman algorithm, after computing the EBC scores, the edges with the highest scores will be taken off till the point the graph splits into two. So, in the graph above, we can see that the edges AB, BC, DE, and EF have the highest score, i.e., 4. We will strike off these edges and it gives us 3 subgraphs that we can call communities:



If you prefer learning all of this in video form, this will help you out:

Neighbourhood (graph theory)

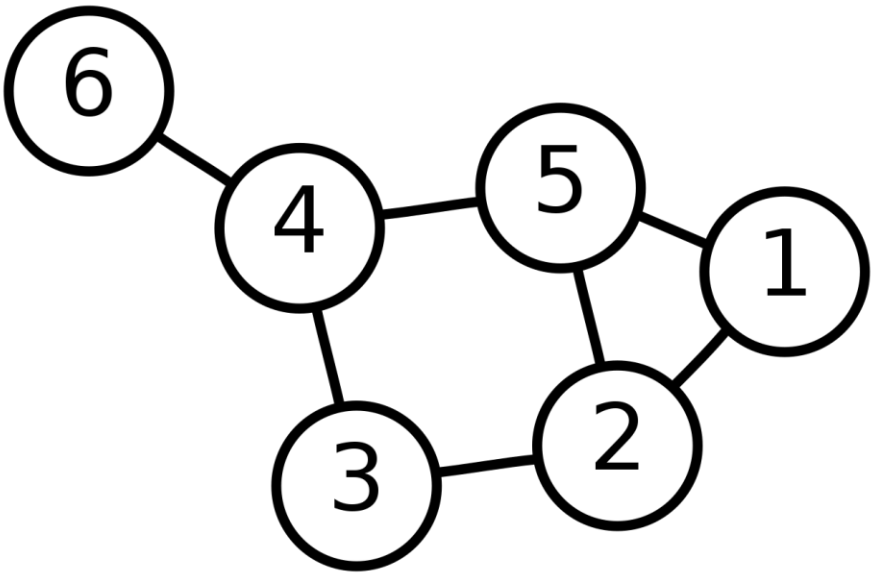
For other meanings of neighbourhoods in mathematics, see Neighbourhood (mathematics).

In graph theory, an **adjacent vertex** of a vertex v in a graph is a vertex that is connected to v by an edge. The **neighbourhood** of a vertex v in a graph G is the subgraph of G induced by all vertices adjacent to v, i.e., the graph composed of the vertices adjacent to v and all edges connecting vertices adjacent to v.

The neighbourhood is often denoted or (when the graph is unambiguous) . The same neighbourhood notation may also be used to refer to sets of adjacent vertices rather than the corresponding induced subgraphs. The neighbourhood described above does not include v itself, and is more specifically the **open neighbourhood** of v; it is also possible to define a neighbourhood in which v itself is included, called the **closed neighbourhood** and denoted by . When stated without any qualification, a neighbourhood is assumed to be open.

Neighbourhoods may be used to represent graphs in computer algorithms, via the adjacency list and adjacency matrix representations. Neighbourhoods are also used in the clustering coefficient of a graph, which is a measure of the average density of its neighbourhoods. In addition, many important classes of graphs may be defined by properties of their neighbourhoods, or by symmetries that relate neighbourhoods to each other.

An isolated vertex has no adjacent vertices. The degree of a vertex is equal to the number of adjacent vertices. A special case is a loop that connects a vertex to itself; if such an edge exists, the vertex belongs to its own neighbourhood.

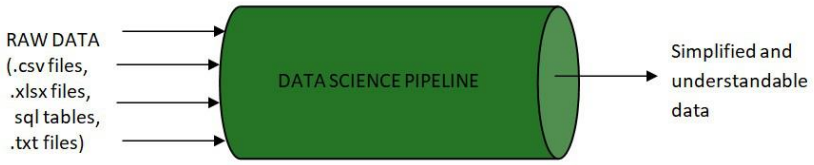


Data Visualization

Before jumping into the term “Data Visualization”, let’s have a brief discussion on the term “Data Science” because these two terms are interrelated. But how? Let’s understand. So, in simple terms, “**Data Science is the science of analyzing raw data using statistics and machine learning techniques with the purpose of drawing conclusions about that information**“.

what is **Data Science Pipeline?**

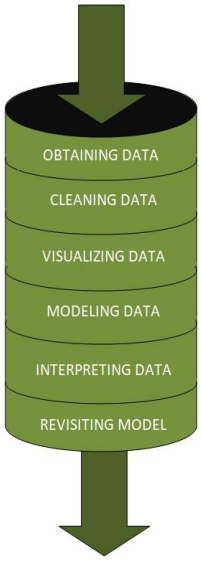
In simple words, a pipeline in data science is “a set of actions which changes the raw (and confusing) data from various sources (surveys, feedback, list of purchases, votes, etc.), to an understandable format so that we can store it and use it for analysis.”



Data science pipeline in a simplified way

The raw data undergoes different stages within a pipeline, which are:

1. Fetching/Obtaining the Data
2. Scrubbing/Cleaning the Data
3. **Data Visualization**
4. Modeling the Data
5. Interpreting the Data
6. Revision

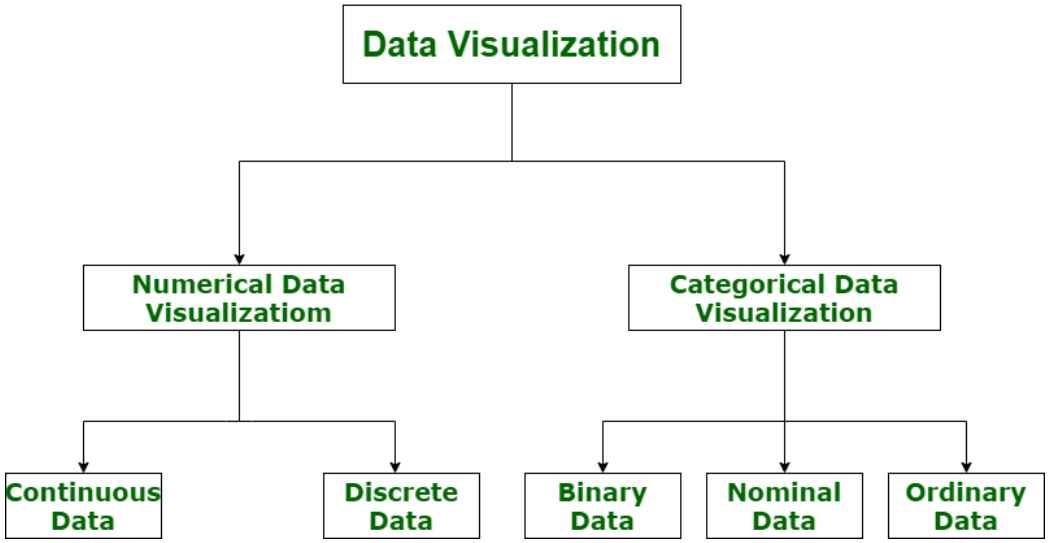


So now we are jumping to the **Data Visualization term**.

Data visualization is the graphical representation of information and data in a pictorial or graphical format(Example: charts, graphs, and maps). Data visualization tools provide an accessible way to see and understand trends, patterns in data, and outliers. Data visualization tools and technologies are essential to analyzing massive amounts of information and making data-driven decisions. The concept of using pictures is to understand data that has been used for centuries. General types of data visualization are Charts, Tables, Graphs, Maps, Dashboards.

Categories of Data Visualization

Data visualization is very critical to market research where both numerical and categorical data can be visualized, which helps in an increase in the impact of insights and also helps in reducing the risk of analysis paralysis. So, data visualization is categorized into the following categories:



Advantages of Data Visualization

1. Better Agreement: In business, for numerous periods, it happens that we need to look at the exhibitions of two components or two situations. A

conventional methodology is to experience the massive information of both the circumstances and afterward examine it. This will clearly take a great deal of time.

2. A Superior Method: It can tackle the difficulty of placing the information of both perspectives into the pictorial structure. This will unquestionably give a superior comprehension of the circumstances. For instance, Google patterns assist us with understanding information identified with top ventures or inquiries in pictorial or graphical structures.

3. Simple Sharing of Data: With the representation of the information, organizations present another arrangement of correspondence. Rather than sharing the cumbersome information, sharing the visual data will draw in and pass on across the data which is more absorbable.

4. Deals Investigation: With the assistance of information representation, a salesman can, without much of a stretch, comprehend the business chart of items. With information perception instruments like warmth maps, he will have the option to comprehend the causes that are pushing the business numbers up just as the reasons that are debasing the business numbers. Information representation helps in understanding the patterns and furthermore, different variables like sorts of clients keen on purchasing, rehash clients, the impact of topography, and so forth.

5. Discovering Relations Between Occasions: A business is influenced by a lot of elements. Finding a relationship between these elements or occasions encourages chiefs to comprehend the issues identified with their business. For instance, the online business market is anything but another thing today. Each time during certain happy seasons, like Christmas or Thanksgiving, the diagrams of online organizations go up. Along these lines, state if an online organization is doing a normal $1 million business in a specific quarter and the business ascends straightaway, at that point they can rapidly discover the occasions compared to it.

6. Investigating Openings and Patterns: With the huge loads of information present, business chiefs can discover the profundity of information in regard to the patterns and openings around them. Utilizing information representation, the specialists can discover examples of the conduct of their clients, subsequently preparing for them to investigate patterns and open doors for business.

Now the most important question arises. **Why is Data Visualization So Important?**

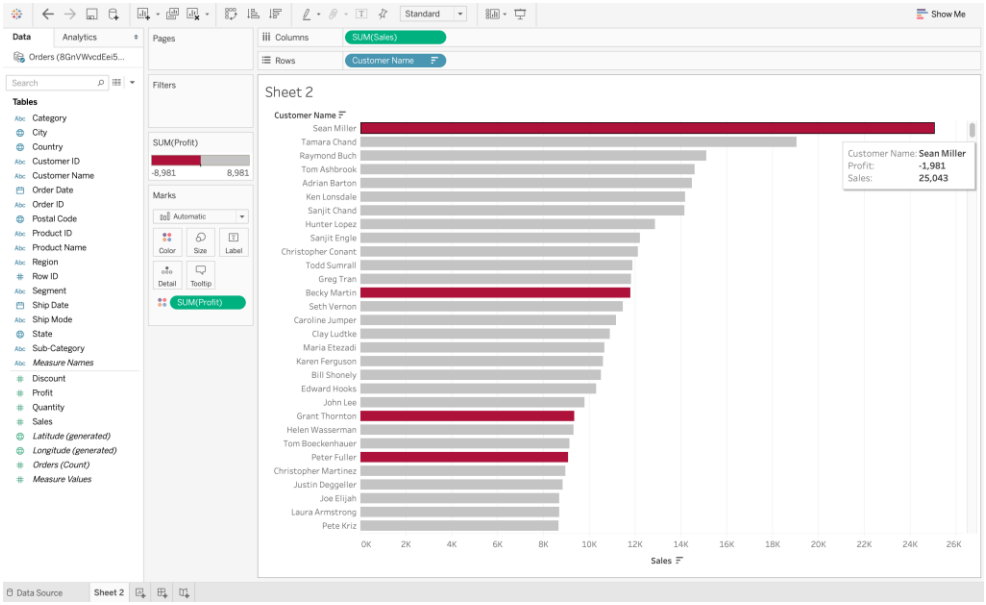
Why is Data Visualization Important?

Let’s take an example. Suppose you compile a data visualization of the company’s profits from 2010 to 2020 and create a line chart. It would be very easy to see the line going constantly up with a drop in just 2018. So you can observe in a second that the company has had continuous profits in all the years

except a loss in 2018. It would not be that easy to get this information so fast from a data table. This is just one demonstration of the usefulness of data visualization. Let's see some more reasons why data visualization is so important.

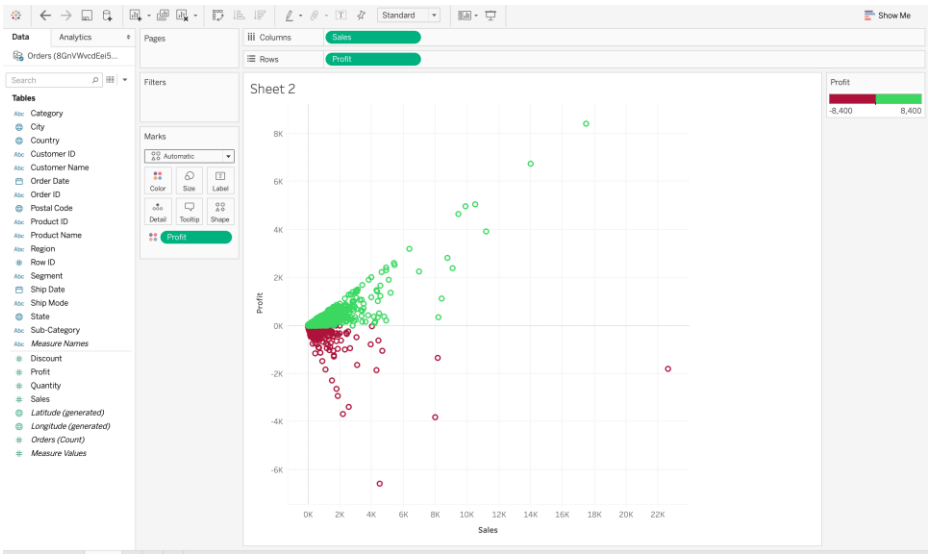
1. Data Visualization Discovers the Trends in Data

The most important thing that data visualization does is discover the trends in data. After all, it is much easier to observe data trends when all the data is laid out in front of you in a visual form as compared to data in a table. For example, the screenshot below on Tableau demonstrates the sum of sales made by each customer in descending order. However, the color red denotes loss while grey denotes profits. So it is very easy to observe from this visualization that even though some customers may have huge sales, they are still at a loss. This would be very difficult to observe from a table.



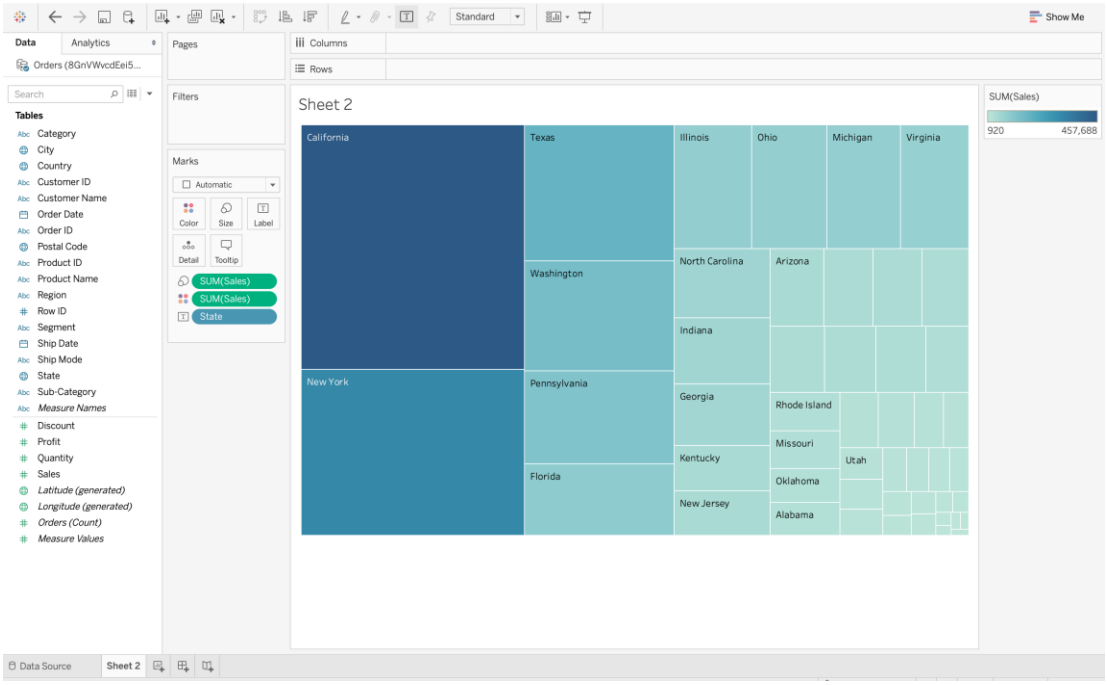
2. Data Visualization Provides a Perspective on the Data

Data Visualization provides a perspective on data by showing its meaning in the larger scheme of things. It demonstrates how particular data references stand with respect to the overall data picture. In the data visualization below, the data between sales and profit provides a data perspective with respect to these two measures. It also demonstrates that there are very few sales above 12K and higher sales do not necessarily mean a higher profit.



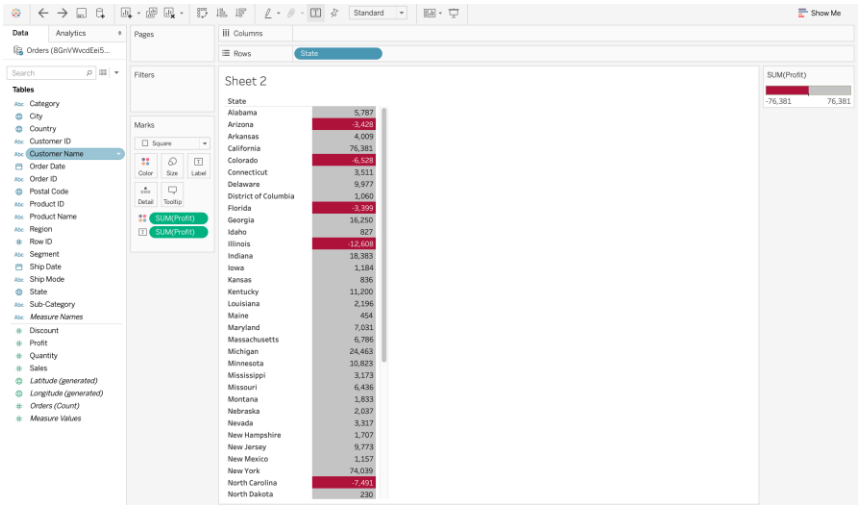
3. Data Visualization Puts the Data into the Correct Context

It is very difficult to understand the context of the data with data visualization. Since context provides the whole circumstances of the data, it is very difficult to grasp by just reading numbers in a table. In the below data visualization on Tableau, a TreeMap is used to demonstrate the number of sales in each region of the United States. It is very easy to understand from this data visualization that California has the largest number of sales out of the total number since the rectangle for California is the largest. But this information is not easy to understand outside of context without data visualization.



4. Data Visualization Saves Time

It is definitely faster to gather some insights from the data using data visualization rather than just studying a chart. In the screenshot below on Tableau, it is very easy to identify the states that have suffered a net loss rather than a profit. This is because all the cells with a loss are colored red using a heat map, so it is obvious states have suffered a loss. Compare this to a normal table where you would need to check each cell to see if it has a negative value to determine a loss. Obviously, data visualization saves a lot of time in this situation!



5. Data Visualization Tells a Data Story

Data visualization is also a medium to tell a data story to the viewers. The visualization can be used to present the data facts in an easy-to-understand form while telling a story and leading the viewers to an inevitable conclusion. This data story, like any other type of story, should have a good beginning, a basic plot, and an ending that it is leading towards. For example, if a data analyst has to craft a data visualization for company executives detailing the profits on various products, then the data story can start with the profits and losses of various products and move on to recommendations on how to tackle the losses.

Top Data Visualization Tools

The following are the 10 best Data Visualization Tools

1. Tableau
2. Looker
3. Zoho Analytics
4. Sisense

5. IBM Cognos Analytics
6. Qlik Sense
7. Domo
8. Microsoft Power BI
9. Klipfolio
10. SAP Analytics Cloud

Data Visualization Projects

An Astronomical Viz:

National Geographic has always been a leader in photography. They are also innovators in the field of data visualization. The above display, The Atlas of Moons, shows the various moons present in our solar system and it starts with our moon. The visualization is scrollable, which makes it more immersive and enjoyable. You can navigate every moon and their orbits, finding out more information about them.

This visualization is the perfect combination of art and data. You can try to imitate this project and create a scrollable visualization yourself. You can pick a similar topic, such as the planets in our solar system. This project can help you try out unique methods of showing data and understanding how you can represent comparisons between multiple objects.

Show the Wilderness of Australia

Nature is beautiful in itself, but in the above visualization, Jonni Walker has shown its beauty in the form of data visualization. The above viz indicates the location and extent of bioluminescence present on the coast of Australia. All the relevant data and legend is present on the map, making it easy to read and comprehend. Jonni had created it on Tableau.

You can create a similar visualization for bioluminescence on other coasts of the world (there are many). Or, you can simply try to replicate this visualization with your tools and see how it turns out. If you're interested in nature studies and want to use your data visualization skills in this sector, then this project will help you get ahead. This project would help you in understanding how you can use data visualization to study nature and relevant topics. Try mimicking this project.

Visualizing Complex Data Sets; The Art Of Storytelling

Storytelling is an art that is simple and complex at the same time. Stories provoke thought and bring out insights and it is similar for large and complex data sets as well. Companies collect a lot of information through various data sources and store it in a single file. Through years this data keeps on increasing and by the time an organization realizes; the data becomes unmanageable. What organizations fail to understand here is that the best stories not presented well, end up being useless. Therefore, it is very important to start dealing with complex data sets using analytics and visualization; the art of storytelling.

Challenges in managing complex data:

- Data is big
- Data is unstructured
- Data is dirty or of bad quality
- Data is incomplete, inaccurate and contains duplicates

Usually, because of such complexities in data, most of the companies are unsure whether they are travelling the right direction or not. Appropriate data cleansing activity could help here, but companies can't afford to do it on their own and hence get benefitted by delegating it to outsourcing firms.

Operating the data using analytics

The complexity of data indicates the level of difficulty a company faces while trying to translate that data into business value. Analysis of that data includes data cleansing,

rearrangement, categorization, and visualization; which predominantly simplify complex data sets.

- Schindler leveraged intelligent analytics to perform predictive maintenance of elevators. Thus, the company justified every dollar clients spend on service and replacement.
- General motors with cost-saving predictions covered about a quarter of its 30k factory robots and managed to foresee and avoid about 100 potential failures in 2 years.
    - Companies willing to improve the decision-making process for increased profitability achieved through improved productivity, need to brush up their data with help of data cleansing experts. It ultimately helps them to get on to descriptive, predictive and prescriptive analytics for better insights from complex data sets.

Benefits of data visualization

- **Absorb information quickly:** Data visualization simplifies makes complex data into simpler ones. This further simplifies the gleaning of insights and information from large datasets.
- **Find the outliers:** Outliners skew your results and hence it is necessary to eliminate these outliners before they predict something which is incorrect and harmful to your business. As outliners tend to pull down the data averages and portray something unusual, visualization helps in finding and eradicating them.
- **Hold your audience's interests longer:** Appealing graphical images have the quality to attract readers as nowadays grappling reader's attention is getting complicated.
- **Understand your next steps:** Visualization helps in identifying trends and patterns to easily understand the next best move with less time and energy.
- **User-friendly data visuals:** The best data visuals are the one which are easy to understand, real time and quickly upgradeable with new technologies.

Conclude

- Companies are getting perplexed with how to manage huge unstructured dirty datasets. Wise use of cleansing techniques, analytics and visualization processes prove to be a savior. Data analytics has shifted its way to a whole new level just by enabling a better service of converting your complex data into interesting visuals. Visualization, the art of storytelling is not only about charts, images or graphs; but is actually revealing some anonymous insights, patterns, trends, and consequently making your organization – wiser.

Ethics in Data Science and Proper Privacy and Usage of Data

Data may be utilized to make decisions and have a large influence on businesses. However, this valuable resource is not without its drawbacks. How can businesses acquire, keep, and use data in an ethical manner? What are the rights that must be protected? Some ethical practices must be followed by data-handling business personnel. Data is someone's personal information and there must be a proper way to use the data and maintain privacy.

What is Ethics?

The term "ethics" comes from the Greek word Ethos, which means "habit" or "custom." Ethics instructs us on what is good and wrong. Philosophers have pondered this crucial topic for a long time and have a lot to say about it. Most people associate ethics with morality: a natural sense of what is "good." We as humans live in a society, and society has rules and regulations. We must be able to decide what is right and what is wrong. Ethics deals with feelings, laws, and social norms which determine right from wrong. Our ways of life must be reasonable and live up to the standards of society.

Why Ethics in Data Science is important?

Today, data science has a significant impact on how businesses are conducted in disciplines as diverse as medical sciences, smart cities, and transportation. Whether it's the protection of personally identifiable data, implicit bias in automated decision-making, the illusion of free choice in psychographics, the social impacts of automation, or the apparent divorce of truth and trust in virtual

communication, the dangers of data science without ethical considerations are as clear as ever. The need for a focus on data science ethics extends beyond a balance sheet of these potential problems because data science practices challenge our understanding of what it means to be human.

Algorithms, when implemented correctly, offer enormous potential for good in the world. When we employ them to perform jobs that previously required a person, the benefits may be enormous: cost savings, scalability, speed, accuracy, and consistency, to name a few. And because the system is more precise and reliable than a human, the outcomes are more balanced and less prone to social prejudice.

A Digital World

We are all living in a digital world, where our day-to-day life is dependent on applications, run by tech companies. We need to take a taxi, we call an Uber. We need to order food, we use Zomato and so on. These companies have our personal data. Our email ID, phone numbers, address, purchase history, etc, and so on. The protection of personal data is thus an important aspect in the present day. Perhaps no aspect of data science ethics has gotten greater attention in recent years than the safeguarding of personal data. Our relationships with social and economic networks have undergone a digital revolution, revealing who we are, what we believe, and what we do.

In India, the Personal Data Protection Bill affirms the rights of digital citizens and addresses the hazards of commercial exploitation of personal and personally identifiable data. The Data Protection Bill is a long-awaited and

desperately needed piece of legislation that would replace India's present antiquated, obsolete, and inadequate data protection policy. It has the potential to raise user understanding of their privacy and hold data custodians and processors accountable. Read more about it here.

Who regulates and owns our Data?

In codifying ethical benchmarks such as the right to be informed, the right to object, the right to access, the right to rectification, and the right to be forgotten, these legal frameworks attempt to rebalance the inequitable relationships of power and influence between organizations and individuals.

The divisions between public and private, individuals and society, and the resource wealthy and resource-poor are being redefined as data becomes the new currency of the international economy. Which rights can be allocated with express or implicit permission, and who owns personal data? To what degree should governmental and commercial institutions be permitted to gather and control enormous databases of human interaction? How much should these data controllers and processors be held liable for the loss or abuse of our personal information?

Data Science Ethics

Analysts, data scientists, and information technology professionals must be concerned about data science ethics. Anyone who works with data must understand the fundamentals. Anyone dealing with any type of data must report any instances of data theft, unethical data collection, storage, use, etc.

For example, from the first time a consumer enters their email address on your website to the time they purchase your goods, your organization may gather and keep data about their trips. People in the marketing team might be dealing with the data. The data of the person must be preserved.

Protected data has been made public on the internet in the past, resulting in harm to persons whose information has been made available. Misconfigured databases, spyware, theft, or publishing on a public forum can all lead to data leaks. Individuals and organizations must use safe computing practices, conduct frequent system audits, and adopt policies to address computer and data security. Companies must take appropriate cybersecurity steps to prevent the leakage of data and information. This is more important for banks and financial institutions which deal with customers' money. Protections must be maintained even when equipment is transferred or disposed of, according to policies.

Some Ethical Practices

Making Decisions:

Data scientists should never make judgments without contacting a client, even if the decision is for the interest of the project. The aims and objectives of projects must be understood by both data scientists and clients.

Let's say a data scientist wishes to take action on behalf of a customer on a certain ongoing project. Even if the action is advantageous to the client and the project, it must be explained to the client, and no choice should be made on

their behalf. Data scientists should only make decisions when it is expressly stated in the contract or when their authority allows them to.

Privacy and Confidentiality of Data:

Data scientists are continually involved in producing, developing, and receiving information. Data concerning client affiliates, customers, workers, or other parties with whom the clients have a confidentiality agreement is often included in this category. Then, regardless of the sort of sensitive information, it is the data scientist's responsibility to protect it. Only when the customer provides permission for data scientists to share or talk about this type of information should it be disclosed or spoken about. Complete privacy of clients' or customers' data must be maintained.

Even if a consumer consents to your organization collecting, storing, and analyzing their personally identifiable information (PII), that doesn't mean they want it made public.

Personally, identifiable information includes:

Phone Number, Address, Full Name, PAN card number, and so on.

To preserve people's privacy, make sure you're keeping the information in a secure database so it doesn't get into the wrong hands. Dual-authentication password protection and file encryption are two data security solutions that assist safeguard privacy.

Data Ownership:

One of the important concepts of ethics in Data Science is that the individual has data ownership. Collecting someone's personal data without their agreement is illegal and immoral. As a result, consent is required to acquire someone's data.

Signed written agreements, digital privacy policies that require users to accept a company's terms and conditions, and pop-ups with checkboxes that allow websites to track users' online behavior using cookies are all typical approaches to get consent. To prevent ethical and legal issues, never assume a consumer agrees to you gathering their data; always ask for permission.

Good intentions with Data:

Intentions of data collection and analyzing data must be good. Data professionals must be clear about how and why they use the data. If a team is collecting data regarding users' spending habits, to make an app to manage expenses, then the intention is good.

Transparency:

Data subjects have a right to know how you plan to acquire, keep, and utilize their personal information, in addition to owning it. Transparency should be used when acquiring data. You should create a policy that explains how cookies are used to track user's activity and how the information gathered is kept in a secure database, as well as train an algorithm that gives a tailored online experience. It is a user's right to have access to this information so that they may choose whether or not to accept your site's cookies.

Some Real-Life Examples:

OK Cupid Data Release:

In 2016, Emil Kirkegaard and Julius Daugbjerg Bjerrekr of Denmark shared a dataset on the Open Science Framework that included information on over 70,000 members of the online dating service OkCupid. The researchers scraped information from OkCupid's site, including user names (but not actual names), ages, gender, religion, and personality characteristics, as well as the answers to the questions the site asks new members to help discover prospective matches, to construct their own dataset.

The information, which was gathered between November 2014 and March 2015, is not anonymous and is quite personal. The only reason the researchers haven't shared users' images is that it would take up too much hard disc space, according to the researchers.

Anyone who has repeated a username from one site to another, or who has used a name that may be traced back to them, may suddenly be severely vulnerable. The data was scraped and uploaded in violation of the basic ethical norms that social scientists observe. When questioned on Twitter, the researchers claimed that the data was already public because it had been submitted on OkCupid.

This was a case of unethical behavior with data. Even though the data was public, collecting it and sharing it explicitly was not right.

Robinhood Data Breach:

American financial services company Robinhood announced a data breach in November 2021, affecting over five million users of the trading app. A customer support system was used to get email addresses, names, phone numbers, and other information. According to the firm, no Social Security numbers were disclosed throughout the probe. Bank accounts and debit cards were not included.

This was a case of data theft and occurred due to security issues in data storage. Steps should be taken to prevent such cases.

Data Science in the fight against Covid-19:

Outbreak analytics, a data science methodology aimed to guide outbreak response, has risen in response to the rising complexity of outbreak data.

The South Korean government used real-time analytics to improve preventative plan design and Covid-positive patient surveillance. It incorporates data from IoT and AI systems that underpin real smart city networks, as well as personal data supplied by confirmed patients. With the use of big data analytics, researchers can follow the patients' travels, identify their contacts, and anticipate the possible outbreak magnitude in a specific location. The information is also utilized to create prevention plans and instructions.

This is an example of how data is used for a good purpose.

Conclusion:

Data Science Ethics is an important topic of discussion in today's world. Organizations and companies using data and implementing data science must

follow a set of ethics while dealing with data. When used ethically, data may help you make better decisions and make a difference in the world.

A look back in data science:

The story of how data scientists became sexy is mostly the story of the coupling of the mature discipline of statistics with a very young one--computer science. The term “Data Science” has emerged only recently to specifically designate a new profession that is expected to make sense of the vast stores of big data. But making sense of data has a long history and has been discussed by scientists, statisticians, librarians, computer scientists and others for years. The following timeline traces the evolution of the term “Data Science” and its use, attempts to define it, and related terms.

1962 John W. Tukey writes in “The Future of Data Analysis”: “For a long time I thought I was a statistician, interested in inferences from the particular to the general. But as I have watched mathematical statistics evolve, I have had cause to wonder and doubt… I have come to feel that my central interest is in *data analysis*… Data analysis, and the parts of statistics which adhere to it, must…take on the characteristics of science rather than those of mathematics… data analysis is intrinsically an empirical science… How vital and how important… is the rise of the stored-program electronic computer? In many instances the answer may surprise many by being ‘important but not vital,’ although in others there is no doubt but what the computer has been ‘vital.’” In 1947, Tukey coined the term “bit” which Claude Shannon used in his 1948 paper “A Mathematical Theory of Communications.” In 1977, Tukey published *Exploratory Data Analysis*, arguing that more emphasis needed to be placed on using data to suggest hypotheses to test and that Exploratory Data Analysis and Confirmatory Data Analysis "can—and should—proceed side by side."

1974 Peter Naur publishes *Concise Survey of Computer Methods* in Sweden and the United States. The book is a survey of contemporary data processing methods that are used in a wide range of applications. It is organized around the concept of data as defined in the *IFIP Guide to Concepts and Terms in Data Processing*: “[Data is] a representation of facts or ideas in a formalized manner capable of being communicated or manipulated by some process.“ The Preface to the book tells the reader that a course plan was presented at the IFIP Congress in 1968, titled “Datalogy, the science of data and of data

processes and its place in education," and that in the text of the book, "the term 'data science' has been used freely." Naur offers the following definition of data science: "The science of dealing with data, once they have been established, while the relation of the data to what they represent is delegated to other fields and sciences."

1977 The International Association for Statistical Computing (IASC) is established as a Section of the ISI. "It is the mission of the IASC to link traditional statistical methodology, modern computer technology, and the knowledge of domain experts in order to convert data into information and knowledge."

1989 Gregory Piatetsky-Shapiro organizes and chairs the first Knowledge Discovery in Databases (KDD) workshop. In 1995, it became the annual ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).

September 1994 *BusinessWeek* publishes a cover story on "Database Marketing": "Companies are collecting mountains of information about you, crunching it to predict how likely you are to buy a product, and using that knowledge to craft a marketing message precisely calibrated to get you to do so... An earlier flush of enthusiasm prompted by the spread of checkout scanners in the 1980s ended in widespread disappointment: Many companies were too overwhelmed by the sheer quantity of data to do anything useful with the information... Still, many companies believe they have no choice but to brave the database-marketing frontier."

1996 Members of the *International Federation of Classification Societies (IFCS)* meet in Kobe, Japan, for their biennial conference. For the first time, the term "data science" is included in the title of the conference ("Data science, classification, and related methods"). The IFCS was founded in 1985 by six country- and language-specific classification societies, one of which, *The Classification Society*, was founded in 1964. The classification societies have variously used the terms data analysis, data mining, and data science in their publications.

1996 Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth publish "From Data Mining to Knowledge Discovery in Databases." They write: "Historically, the notion of finding useful patterns in data has been given a variety of names, including data mining, knowledge extraction, information discovery, information harvesting, data archeology, and data pattern processing... In our view, KDD [Knowledge Discovery in Databases] refers to

the overall process of discovering useful knowledge from data, and data mining refers to a particular step in this process. *Data mining* is the application of specific algorithms for extracting patterns from data… the additional steps in the KDD process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, are essential to ensure that useful knowledge is derived from the data. Blind application of data-mining methods (rightly criticized as data dredging in the statistical literature) can be a dangerous activity, easily leading to the discovery of meaningless and invalid patterns."

1997 In his inaugural lecture for the H. C. Carver Chair in Statistics at the University of Michigan, Professor C. F. Jeff Wu (currently at the Georgia Institute of Technology), calls for statistics to be renamed data science and statisticians to be renamed data scientists.

1997 The journal Data Mining and Knowledge Discovery is launched; the reversal of the order of the two terms in its title reflecting the ascendance of "data mining" as the more popular way to designate "extracting information from large databases."

December 1999 Jacob Zahavi is quoted in "Mining Data for Nuggets of Knowledge" in Knowledge@Wharton: "Conventional statistical methods work well with small data sets. Today's databases, however, can involve millions of rows and scores of columns of data… Scalability is a huge issue in data mining. Another technical challenge is developing models that can do a better job analyzing data, detecting non-linear relationships and interaction between elements… Special data mining tools may have to be developed to address web-site decisions."

2001 William S. Cleveland publishes "Data Science: An Action Plan for Expanding the Technical Areas of the Field of Statistics." It is a plan "to enlarge the major areas of technical work of the field of statistics. Because the plan is ambitious and implies substantial change, the altered field will be called 'data science.'" Cleveland puts the proposed new discipline in the context of computer science and the contemporary work in data mining: "…the benefit to the data analyst has been limited, because the knowledge among computer scientists about how to think of and approach the analysis of data is limited, just as the knowledge of computing environments by statisticians is limited. A merger of knowledge bases would produce a powerful force for innovation. This suggests that statisticians should look to computing for knowledge today just as data science looked to mathematics in the past. …

departments of data science should contain faculty members who devote their careers to advances in computing with data and who form partnership with computer scientists."

2001 Leo Breiman publishes "Statistical Modeling: The Two Cultures" (PDF): "There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools."

April 2002 Launch of *Data Science Journal*, publishing papers on "the management of data and databases in Science and Technology. The scope of the Journal includes descriptions of data systems, their publication on the internet, applications and legal issues." The journal is published by the Committee on Data for Science and Technology (CODATA) of the International Council for Science (ICSU).

January 2003 Launch of *Journal of Data Science*: "By 'Data Science' we mean almost everything that has something to do with data: Collecting, analyzing, modeling..... yet the most important part is its applications--all sorts of applications. This journal is devoted to applications of statistical methods at large.... The *Journal of Data Science* will provide a platform for all data workers to present their views and exchange ideas."

May 2005 Thomas H. Davenport, Don Cohen, and Al Jacobson publish "Competing on Analytics," a Babson College Working Knowledge Research Center report, describing "the emergence of a new form of competition based on the extensive use of analytics, data, and fact-based decision making... Instead of competing on traditional factors, companies are beginning to employ statistical and quantitative analysis and predictive modeling as primary elements of competition. " The research is later published by Davenport in the *Harvard Business Review* (January 2006) and is expanded (with Jeanne G. Harris) into the book *Competing on Analytics: The New Science of Winning* (March 2007).

September 2005 The National Science Board publishes "Long-lived Digital Data Collections: Enabling Research and Education in the 21st Century." One of the recommendations of the report reads: "The NSF, working in partnership with collection managers and the community at large, should act to develop and mature the career path for data scientists and to ensure that the research enterprise includes a sufficient number of high-quality data scientists." The report defines data scientists as "the information and computer scientists, database and software engineers and programmers, disciplinary experts, curators and expert annotators, librarians, archivists, and others, who are crucial to the successful management of a digital data collection."

2007 The Research Center for Dataology and Data Science is established at Fudan University, Shanghai, China. In 2009, two of the center's researchers, Yangyong Zhu and Yun Xiong, publish "Introduction to Dataology and Data Science," in which they state "Different from natural science and social science, Dataology and Data Science takes data in cyberspace as its research object. It is a new science." The center holds annual symposiums on Dataology and Data Science.

July 2008 The JISC publishes the final report of a study it commissioned to "examine and make recommendations on the role and career development of data scientists and the associated supply of specialist data curation skills to the research community. " The study's final report, "The Skills, Role & Career Structure of Data Scientists & Curators: Assessment of Current Practice & Future Needs," defines data scientists as "people who work where the research is carried out--or, in the case of data centre personnel, in close collaboration with the creators of the data--and may be involved in creative enquiry and analysis, enabling others to work with digital data, and developments in data base technology."

January 2009 *Harnessing the Power of Digital Data for Science and Society* is published. This report of the Interagency Working Group on Digital Data to the Committee on Science of the National Science and Technology Council states that "The nation needs to identify and promote the emergence of new disciplines and specialists expert in addressing the complex and dynamic challenges of digital preservation, sustained access, reuse and repurposing of data. Many disciplines are seeing the emergence of a new type of data science and management expert, accomplished in the computer, information, and data sciences arenas and in another domain science. These individuals are key to the current and future success of the scientific

enterprise. However, these individuals often receive little recognition for their contributions and have limited career paths."

January 2009 Hal Varian, Google's Chief Economist, tells the *McKinsey Quarterly*: "I keep saying the sexy job in the next ten years will be statisticians. People think I'm joking, but who would've guessed that computer engineers would've been the sexy job of the 1990s? The ability to take data—to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it—that's going to be a hugely important skill in the next decades… Because now we really do have essentially free and ubiquitous data. So the complimentary scarce factor is the ability to understand that data and extract value from it… I do think those skills—of being able to access, understand, and communicate the insights you get from data analysis—are going to be extremely important. Managers need to be able to access and understand the data themselves."

March 2009 Kirk D. Borne and other astrophysicists submit to the Astro2010 Decadal Survey a paper titled "The Revolution in Astronomy Education: Data Science for the Masses "(PDF): "Training the next generation in the fine art of deriving intelligent understanding from data is needed for the success of sciences, communities, projects, agencies, businesses, and economies. This is true for both specialists (scientists) and non-specialists (everyone else: the public, educators and students, workforce). Specialists must learn and apply new data science research techniques in order to advance our understanding of the Universe. Non-specialists require information literacy skills as productive members of the 21st century workforce, integrating foundational skills for lifelong learning in a world increasingly dominated by data."

May 2009 Mike Driscoll writes in "The Three Sexy Skills of Data Geeks": "…with the Age of Data upon us, those who can model, munge, and visually communicate data—call us statisticians or data geeks—are a hot commodity." [Driscoll will follow up with The Seven Secrets of Successful Data Scientists in August 2010]

June 2009 Nathan Yau writes in "Rise of the Data Scientist": "As we've all read by now, Google's chief economist Hal Varian commented in January that the next sexy job in the next 10 years would be statisticians. Obviously, I whole-heartedly agree. Heck, I'd go a step further and say they're sexy now—mentally and physically. However, if you went on to read the rest of Varian's interview, you'd know that by statisticians, he actually meant it as a general title for someone who is able to extract information from large datasets and

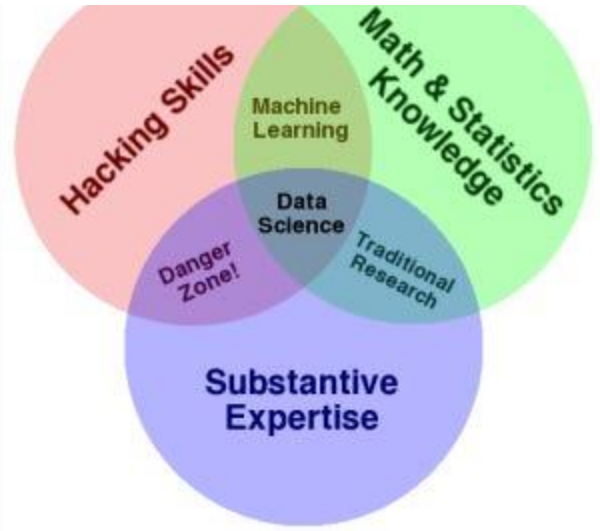
then present something of use to non-data experts… [Ben] Fry… argues for an entirely new field that combines the skills and talents from often disjoint areas of expertise… [computer science; mathematics, statistics, and data mining; graphic design; infovis and human-computer interaction]. And after two years of highlighting visualization on FlowingData, it seems collaborations between the fields are growing more common, but more importantly, computational information design edges closer to reality. We're seeing *data scientists*— people who can do it all— emerge from the rest of the pack."

June 2009 Troy Sadkowsky creates the data scientists group on LinkedIn as a companion to his website, datasceintists.com (which later became datascientists.net).

February 2010 Kenneth Cukier writes in *The Economist* Special Report "Data, Data Everywhere": "… a new kind of professional has emerged, the data scientist, who combines the skills of software programmer, statistician and storyteller/artist to extract the nuggets of gold hidden under mountains of data."

June 2010 Mike Loukides writes in "What is Data Science?": "Data scientists combine entrepreneurship with patience, the willingness to build data products incrementally, the ability to explore, and the ability to iterate over a solution. They are inherently interdisciplinary. They can tackle all aspects of a problem, from initial data collection and data conditioning to drawing conclusions. They can think outside the box to come up with new ways to view the problem, or to work with very broadly defined problems: 'here's a lot of data, what can you make from it?'"

September 2010 Hilary Mason and Chris Wiggins write in "A Taxonomy of Data Science": "…we thought it would be useful to propose one possible taxonomy… of what a data scientist does, in roughly chronological order: Obtain, Scrub, Explore, Model, and iNterpret…. Data science is clearly a blend of the hackers' arts… statistics and machine learning… and the expertise in mathematics and the domain of the data for the analysis to be interpretable… It requires creative decisions and open-mindedness in a scientific context."



Source: Drew Conway

September 2010 Drew Conway writes in "The Data Science Venn Diagram": "...one needs to learn a lot as they aspire to become a fully competent data scientist. Unfortunately, simply enumerating texts and tutorials does not untangle the knots. Therefore, in an effort to simplify the discussion, and add my own thoughts to what is already a crowded market of ideas, I present the Data Science Venn Diagram... hacking skills, math and stats knowledge, and substantive expertise."

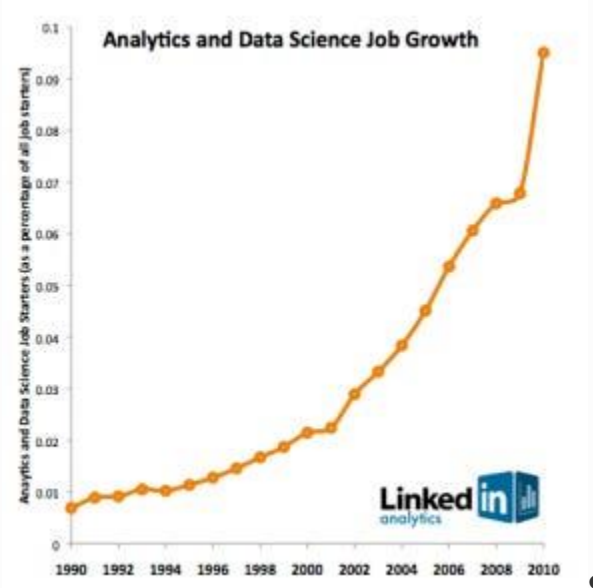
May 2011 Pete Warden writes in "Why the term 'data science' is flawed but useful": "There is no widely accepted boundary for what's inside and outside of data science's scope. Is it just a faddish rebranding of statistics? I don't think so, but I also don't have a full definition. I believe that the recent abundance of data has sparked something new in the world, and when I look around I see people with shared characteristics who don't fit into traditional categories. These people tend to work beyond the narrow specialties that dominate the corporate and institutional world, handling everything from finding the data, processing it at scale, visualizing it and writing it up as a story. They also seem to start by looking at what the data can tell them, and then picking interesting threads to follow, rather than the traditional scientist's approach of choosing the problem first and then finding data to shed light on it."

May 2011 David Smith writes in "'Data Science': What's in a name?": "The terms 'Data Science' and 'Data Scientist' have only been in common usage for a little over a year, but they've really taken off since then: many companies are now hiring for 'data scientists', and entire conferences are run under the name of 'data science'. But despite the widespread adoption, some have resisted the change from the more traditional terms like 'statistician' or 'quant' or 'data

analyst'…. I think 'Data Science' better describes what we actually do: a combination of computer hacking, data analysis, and problem solving."

June 2011 Matthew J. Graham talks at the Astrostatistics and Data Mining in Large Astronomical Databases workshop about "The Art of Data Science" (PDF). He says: "To flourish in the new data-intensive environment of 21st century science, we need to evolve new skills… We need to understand what rules [data] obeys, how it is symbolized and communicated and what its relationship to physical space and time is."

September 2011 Harlan Harris writes in "Data Science, Moore's Law, and Moneyball" : "'Data Science' is defined as what 'Data Scientists' do. What Data Scientists do has been very well covered, and it runs the gamut from data collection and munging, through application of statistics and machine learning and related techniques, to interpretation, communication, and visualization of the results. Who Data Scientists are may be the more fundamental question… I tend to like the idea that Data Science is defined by its practitioners, that it's a career path rather than a category of activities. In my conversations with people, it seems that people who consider themselves Data Scientists typically have eclectic career paths, that might in some ways seem not to make much sense."



September 2011 D.J. Patil writes in "Building Data Science Teams": "Starting in 2008, Jeff Hammerbacher (@hackingdata) and I sat down to share our experiences building the data and analytics groups at Facebook and LinkedIn. In many ways, that meeting was the start of data science as a distinct professional specialization…. we realized that as our organizations grew, we both had to figure out what to call the

people on our teams. ‘Business analyst’ seemed too limiting. ‘Data analyst’ was a contender, but we felt that title might limit what people could do. After all, many of the people on our teams had deep engineering expertise. ‘Research scientist’ was a reasonable job title used by companies like Sun, HP, Xerox, Yahoo, and IBM. However, we felt that most research scientists worked on projects that were futuristic and abstract, and the work was done in labs that were isolated from the product development teams. It might take years for lab research to affect key products, if it ever did. Instead, the focus of our teams was to work on data applications that would have an immediate and massive impact on the business. The term that seemed to fit best was data scientist: those who use both data and science to create something new. “

September 2012 Tom Davenport and D.J. Patil publish “Data Scientist: The Sexiest Job of the 21st Century” in the *Harvard Business Review*.

How does Data Science Work?

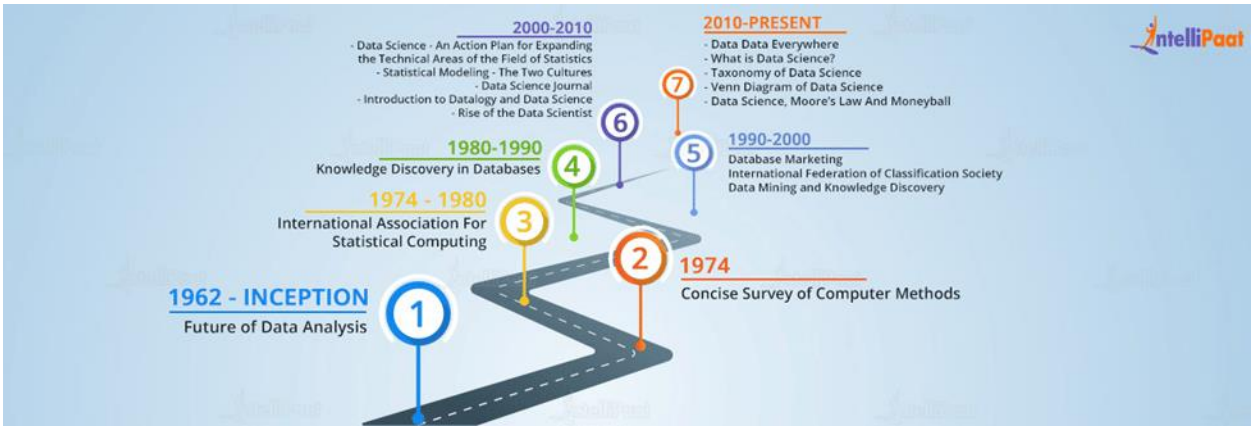
The working of data science can be explained as follows:

1. Raw data is gathered from various sources that explain the business problem.
2. Using various statistical analysis, and machine learning approaches, data modeling is performed to get the optimum solutions that best explain the business problem.
3. Actionable insights that will serve as a solution for the business problems gathered through data science.

Let’s understand this with an example, Suppose there is an organization that is working towards finding out potential leads for their sales team. They can follow the following approach to get an optimal solution using Data Science:

1. Gather the previous data on the sales that were closed.
2. Use statistical analysis to find out the patterns that were followed by the leads that were closed.
3. Use machine learning to get actionable insights for finding out potential leads.
4. Use the new data on sales lead to segregate potential leads that will be highly likely to be closed.

Now that we have discussed data science and how it works, let's discuss the history of data science, and how it has evolved into an emerging domain for the years to come.



1. **1962 – Inception**

   a. Future of Data Analysis – In 1962, John W Tukey wrote the "Future of Data Analysis" where he first mentioned the importance of data analysis with respect to science rather than mathematics.

2. **1974**

   a. Concise Survey of Computer Methods – In 1974, Peter Naur published the "Concise Survey of Computer methods that surveys the contemporary methods of data processing in various applications.

3. **1974 – 1980**

   a. International Association For Statistical Computing – In 1997, The committee was formed whose sole purpose is to link traditional statistical

methodology with modern computer technology to extract useful information and knowledge from the data.

4. **1980-1990**

   a. Knowledge Discovery in Databases – In 1989, Gregory Piatetsky-Shapiro chaired the Knowledge Discovery in Databases that later went on to become the annual conference on knowledge discovery and data mining.

5. **1990-2000**

   a. Database Marketing – In 1994, BusinessWeek published a cover story that explains how big organizations are using the customer data to predict the likelihood of a customer buying a specific product or not. Kind of like how targeted ads work in the modern era for social media campaigns.

   b. International Federation of Classification Society – For the first time in 1996, the term “Data Science” was used in a conference held in Japan.

6. **2000-2010**

   a. Data Science – An Action Plan for Expanding the Technical Areas of the Field of Statistics – In 2001, William S Cleveland published the action plan, that majorly focused on major areas of the technical work in the field of statistics and coined the term Data Science.

   b. Statistical Modeling – The Two Cultures – In 2001, Leo Breiman wrote “There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown”.

   c. Data Science Journal – April 2002 saw the launch of a journal that focused on management of data and databases in science and technology.

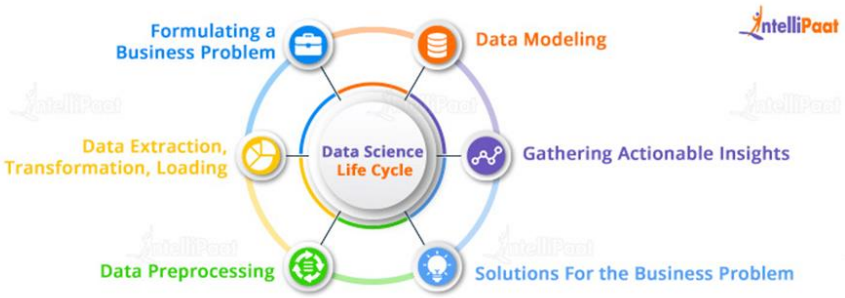
7. **2010-Present**

   a. Data Everywhere – In February 2010, Kenneth Cukier wrote a special report for The Economist that said a new professional has arrived – a data scientist. Who combines the skills of software programmer, statistician and storyteller/artist to extract the nuggets of gold hidden under mountains of data.

b. What is Data Science? – In June 2010, Mike Loukides described data science as combining entrepreneurship with patience, the willingness to build data products incrementally, the ability to explore, and the ability to iterate over a solution.

Data Science Life Cycle

The Data Science lifecycle comprises of the following:



1. **Formulating a Business Problem**

   Any data science problem will start their journey from formulating a business problem. A business problem explains the issues that may be fixed with insights gathered from an efficient Data Science solution. A simple example of a business problem is – You have past 1 year's sales data for a retail store. Using machine learning approaches, you have to predict or forecast the sales for the next 3 months that will help the store to create an inventory that will help in reducing the wastage of products that have lesser shelf life than the other products.

2. **Data Extraction, Transformation, Loading**

   The next step in the data science life cycle is to create a data pipeline where the relevant data is extracted from the source and transformed into machine readable format, and eventually the data is loaded into the program or the machine learning pipeline to get things started.

   For the above example – To forecast the sales, we will need data from the store that will be useful for formulating an efficient machine learning model. Keeping this in mind, we would create separate data points that may or may not be affecting the sales for that particular store.

3. **Data Preprocessing**

   The third step is where the magic happens. Using statistical analysis, Exploratory data analysis, data wrangling and manipulation, we will create meaningful data. The preprocessing is done to assess the various data points and formulate hypotheses that best explain the relationship between the various features in the data.

   For example – The store sales problem will require the data to be in a time series format to be able to forecast the sales. The hypothesis testing will test the stationarity of the series and further computations will show various trends, seasonality and other relationship patterns in the data.

4. **Data Modeling**

   This step involves advanced machine learning concepts that will be used for feature selection, feature transformation, standardization of the data, data normalization, etc. Choosing the best algorithms based on evidence from the above steps will help you create a model that will efficiently create a forecast for the said months in the above example.

   For example – We can use the Time Series forecasting approach for the business problem where the presence of high dimensional data could be the case. We will use various dimensionality reduction techniques, and create a Forecasting model using AR, MA, or ARIMA model and forecast the sales for the next quarter.

5. **Gathering Actionable Insights**

   The final step from the data science life cycle is gathering insights from the said problem statement. We create inferences and findings from the entire process that would best explain the business problem.

   For example – From the above Time series model, we will get the monthly or weekly sales for the next 3 months. These insights will in turn help the professionals create a strategy plan to overcome the problem at hand.

6. **Solutions For the Business Problem**

   The solutions for the business problem are nothing but actionable insights that

will solve the problem using evidence based information. For example – Our forecast from the Time series model will give an efficient estimate for the store sales in the next 3 months. Using those insights, the store can plan their inventory to reduce the wastage of perishable goods.

Prerequisites for Data Science

There are several prerequisites that must be fulfilled in order to efficiently drive data science solutions in an organization. Some of the prerequisites are as follows:

1. **Programming Knowledge**

   For the statistical analysis and computations that are required for the Data Science processes, it is necessary for the professionals to be familiar with Programming languages such as Python or R programming. The library support and scripting knowledge helps you create machine learning models from scratch with ease. Scikit-learn, Tensorflow, pandas, matplotlib, seaborn, scipy, numpy, etc, are some of the inbuilt python programming libraries that can be used for Data Science using Python.

2. **Statistics, Probability, And Linear Algebra**

   The knowledge of descriptive statistics, inferential statistics is a must if you really want to make a career in data science. With the help of statistical analysis, you are able to create various inferences and understand the data at hand. One example would be how we discussed performing hypothesis testing to test whether a time series is stationary or not.

Probability and linear algebra also plays an important role in shaping the understanding of complex machine learning algorithms. If you're familiar with these concepts, it will be easier for you to understand the internal functioning of various machine learning algorithms.

Want to learn more about Statistics for Data Science check out our course on Statistics for Data Science Course.

3. **SQL, Excel And Visualization Tools**

   The visualization tools such as PowerBI, Tableau, etc, can provide a great interactive interface to represent various data points, that can help in performing initial analysis or just to understand the data.

SQL and Excel on the other hand can help you in understanding the representation of data in tabular format or data frames that help in data manipulation, wrangling, etc.

4. **Big Data And Cloud**

   A machine learning model deployed at scale is where the cloud comes into the picture, to be able to magnify the learnings and outcomes for any business problem we use machine learning on cloud. And big data gives a better perspective on how to handle large and complex data for out business problems and for creating data pipelines for continuous development and training of various machine learning models at scale.

Who is a Data Scientist?

Data scientists are IT professionals whose main role in an organization is to perform data wrangling on a large volume of data—structured and unstructured—after gathering and analyzing it. Data scientists need this voluminous data for multiple reasons including building hypotheses, analyzing market and customer patterns, and making inferences.

Roles and Responsibilities of a Data Scientist

The role and responsibilities of a data scientist can vary from organization to organization, based on this, we can segregate them in the following manner.

A data scientist's role in any organization will involve the following:

1. Data Extraction, Loading, Transformation
2. Exploratory Data Analysis
3. Data Manipulation
4. Statistical Analysis
5. Visualization
6. Data Modeling

7. Gathering Actionable Insights

This modified data is further used for the prediction of results that can help organizations to come up with efficient plans that need to be executed for the growth of the organizations.

Although in some organizations, some of the responsibilities will be divided amongst the data engineers and data analysts who will wrangle the data and transform the features for it to be provided to the machine learning engineers to perform various modeling techniques to get the solutions.

And finally, the data scientist will make sense of the inferences to get the solutions for the business problems. But, in some organizations, a data scientist might have to cover all these aspects in order to drive solutions for the business problems.

Real-World Examples

1. A team of statisticians and data scientists was able to somewhat predict the various waves and their outcomes in the world using the data from previous catastrophic events of the same scale when the world encountered COVID-19 for the first time. As more data was available, they were able to predict the outcomes with more precision and were forecasting the COVID-19 outbreak on a daily basis with much more efficiency and accuracy.

2. Recommendation engines for various streaming websites take account of the historical data of the users that has various features. Based on these data points, data scientists have built recommendation engines using machine learning algorithms that can give the users recommendations that they are most likely to watch based on their previous choices.

3. Autonomous cars and how teams at the likes of Tesla have used computer vision technology in a way to navigate through the traffic keeping pedestrians and other vehicles in mind.

Similar to creating world-class solutions for business problems that may seem impossible at first, the other responsibilities that a data scientist takes care of are as follows:

1. Leadership skills to manage teams and keep the entire Data Science process running with efficiency for any given business problem.

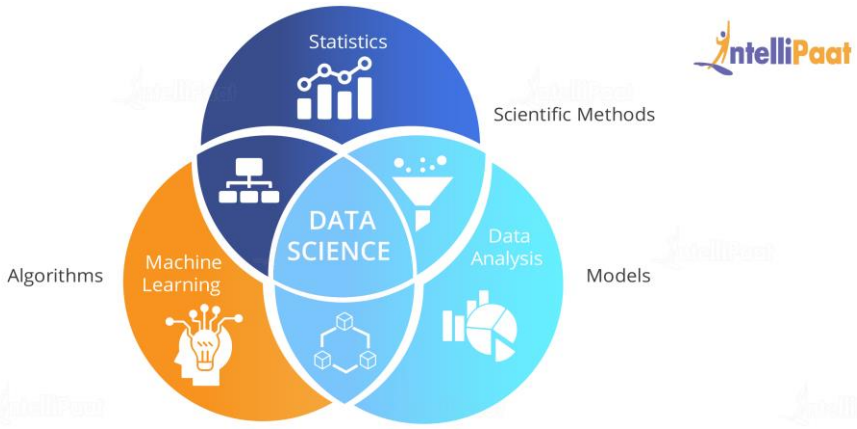
2. Project Management Skills to plan entire end-to-end projects from inception to conclusion with optimized problem-solving approaches.

3. Stakeholder Management to be able to convey the requirements to the concerned teams and be on the same page while delivering the solutions to the respective stakeholders.

Since we have discussed the various roles and responsibilities of a data scientist, let us also discuss why becoming a data scientist is a good roadmap for your career.

Why Data Science?

Currently, across industries, there is a huge need for skilled and certified data scientists. They are among the highest-paid professionals in the IT industry. According to Glassdoor, a data scientist is the best job in America with an average annual salary of $110,000. Only a few people process the skills to derive valuable insights out of raw data.

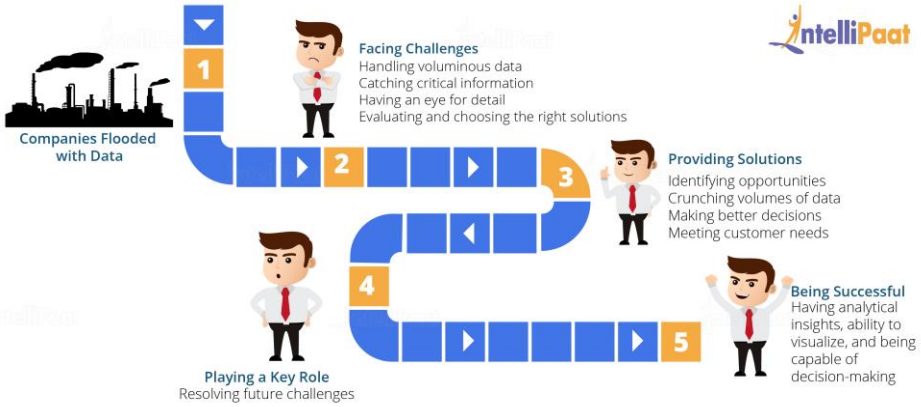


Furthermore, looking at the ever-increasing requirements, McKinsey has predicted that there will be a 50 percent gap in the demand and supply of data scientists in the upcoming years.

In recent years, there has been huge growth in the field of the Internet of Things (IoT), which has led to the generation of 90 percent of data being generated today. Every day, 2.5 quintillion bytes of data are generated, and it is accelerated with the growth of IoT.

This data comes from all possible sources such as

- Sensors used in shopping malls to gather the shoppers' information
- Posts on social media platforms
- Digital pictures and videos captured on phones
- Purchase transactions made through e-commerce

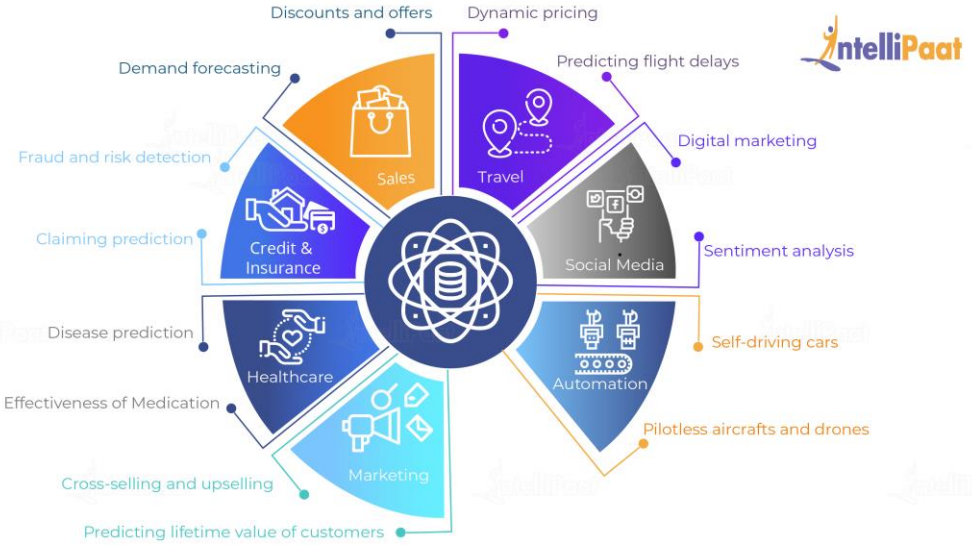


The preceding picture represents the concept of Data Science. It brings together a lot of skills such as statistics, mathematics, and business domain knowledge, and helps organizations find ways to:

- Reduce costs
- Get into new markets
- Tap into different demographics
- Gauge the effectiveness of marketing campaigns
- Launch new products or services

And the list is endless!

Therefore, regardless of the industry vertical, data science is likely to play a key role in your organization's success.



Thank you