Big Data Machine Learning

Dr. Rajiv Misra
Dept. of Computer Science & Engg.
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Preface

Content of this Lecture:

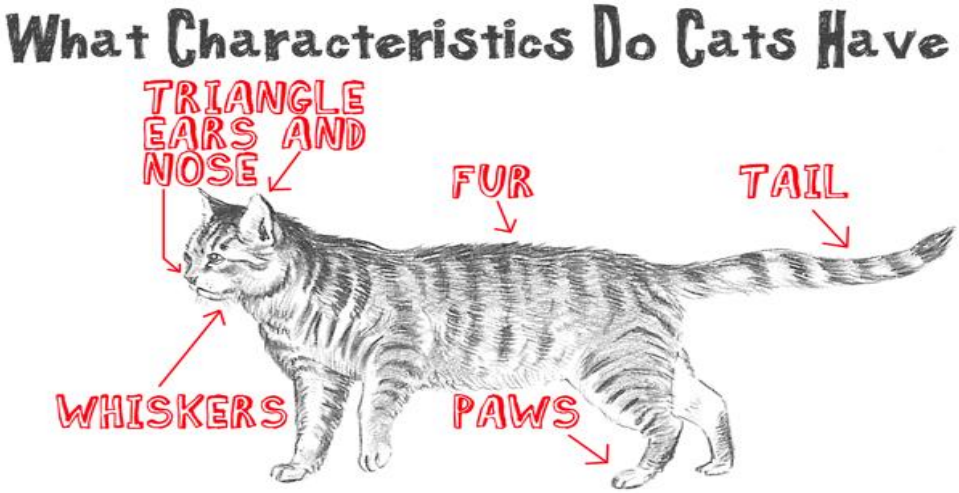
- In this lecture, we will provide an overview of machine learning techniques to explore, analyze, and leverage data.
- We will also discuss tools and algorithms that you can use to create machine learning models that learn from data, and to scale those models up to big data problems.

Machine Learning Overview

What is Machine Learning?

- **Learning from data:**

- Machine learning is the field of study that focuses on computer systems that can learn from data. That is the system's often called models can learn to perform a specific task by analyzing lots of examples for a particular problem. For example, a machine learning model can learn to recognize an image of a cat by being shown lots and lots of images of cats.



What is Machine Learning ?

1. Learning from Data

2. **No explicit programming:** This notion of learning from data means that a machine learning model can learn a specific task without being explicitly programmed. In other words, the machine learning model is not given the step by step instructions on how to recognize the image of a cat.

- Instead, the model learns what features are important in determining whether it picture contains a cat from the data that has analyzed. Because the model learns to perform this task from data it's good to know that the amount and quality of data available for building the model are important factors in how well the model learns the task.

What is Machine Learning ?

- **Discovering hidden patterns:** Because machine learning models can learn from data that can be used to discover hidden patterns and trends in the data.

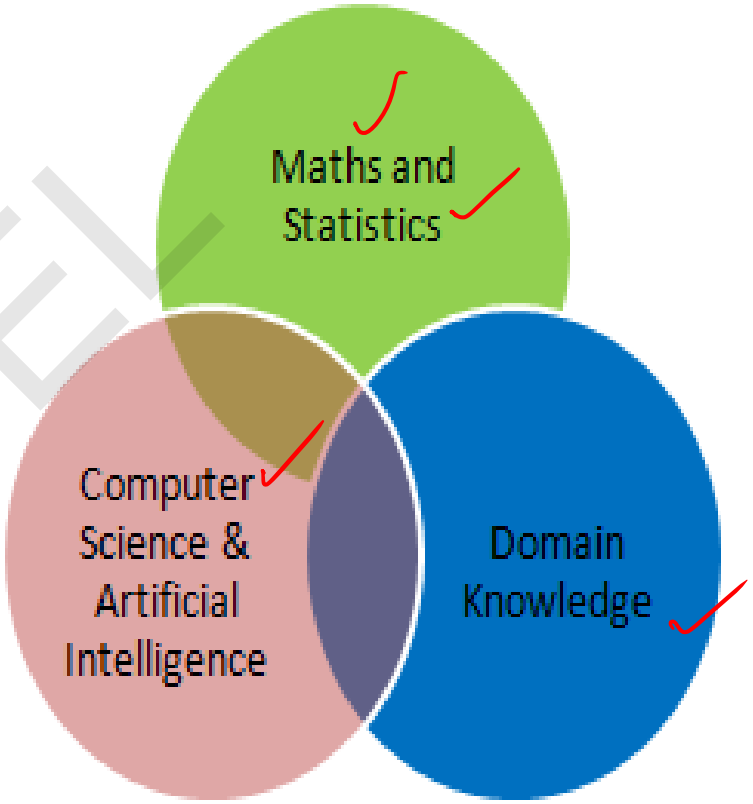
- **Data-driven decisions:** These patterns and trends lead to valuable insights into the data. Thus the use of machine learning allows for data driven decisions to be made for a particular problem.

- So to summarize, the field of machine learning focuses on the study and construction of computer systems that can learn from data without being explicitly programmed. Machine learning algorithms and techniques are used to build models, to discover hidden patterns and trends in the data allowing for data-driven decisions to be made.

Machine Learning (ML) is an Interdisciplinary Field

In applying machine learning to a problem, domain knowledge is essential to the success of end results. By domain knowledge we mean an understanding of the application or business domain.

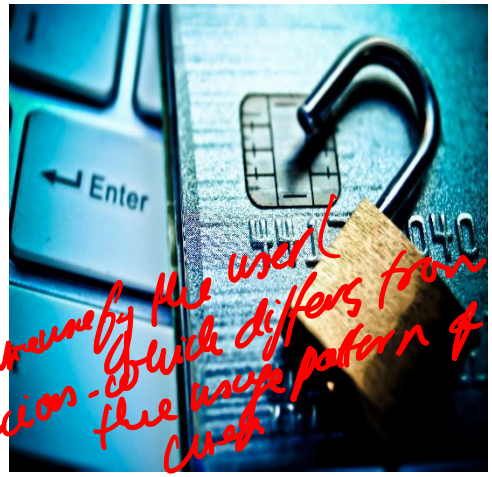
Knowledge about the application, the data related to the application, and how the outcomes will be used are crucial to driving the process of building the machine learning model. So domain knowledge is also an integral part of a machine learning solution.



Machine learning is an interdisciplinary fields – Statistics, Mathematics, Computer Science, AI & other Domains

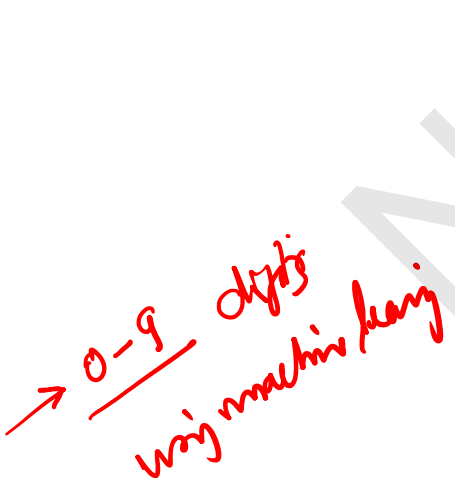
Example Application of Machine Learning

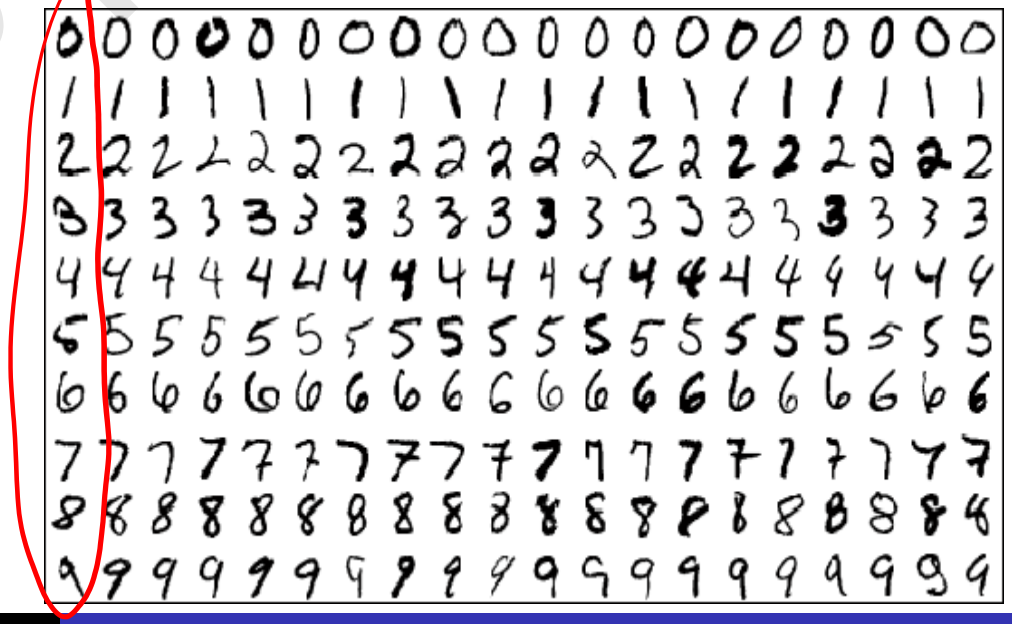
- **Credit card fraud detection:** One application of machine learning that you likely used this past weekend, or even just today, is credit card fraud detection. Every time you use your credit card, the current purchase is analyzed against your history of credit card transactions to determine if the current purchase is a legitimate transaction or a potentially fraudulent one. If the purchase is very different from your past purchases, such as for a big ticket item in a category that you had never shown an interest in or when the point of sales location is from another country, then it will be flagged as a suspicious activity.



Example Application of Machine Learning

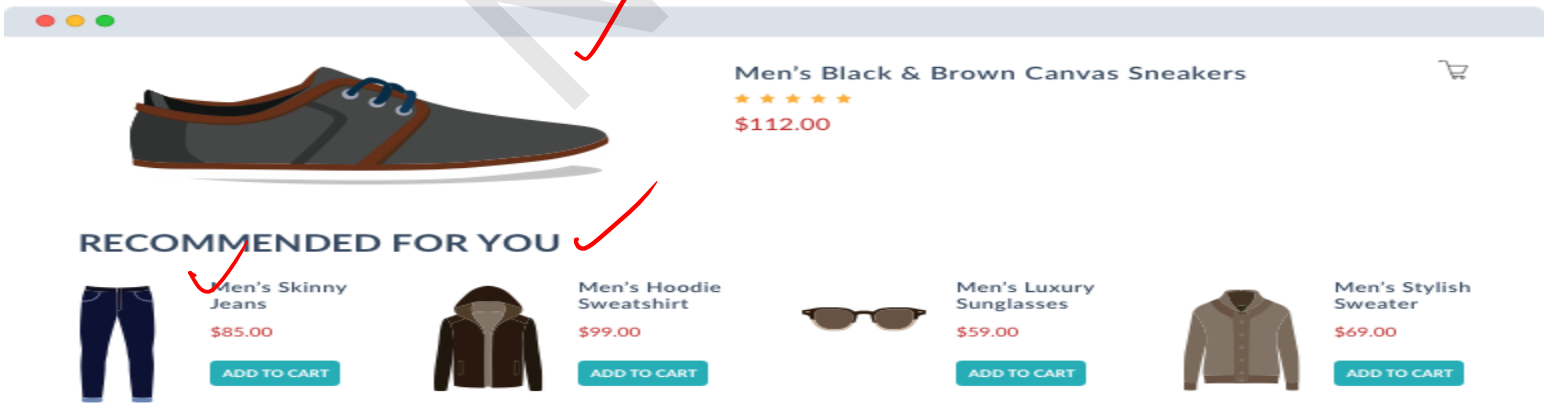
- **Handwritten digit recognition:** When you deposit a hand-written check into an ATM, a machine learning process is used to read the numbers written on the check to determine the amount of the deposit. Handwritten digits are trickier to decipher than typed digits due to the many variations in people's handwriting.





Example Application of Machine Learning

- **Recommendations on websites:** After you buy an item on a website you will often get a list of related items. Often this will be displayed as customers who bought this item also bought these items, or you may also like.
- These related items have been associated with the item you purchased by a machine learning model, and are now being shown to you since you may also be interested in them.



More Applications of Machine Learning

- Targeted ads on mobile apps
- Sentiment analysis
- Climate monitoring
- Crime pattern detection
- Drug effectiveness analysis

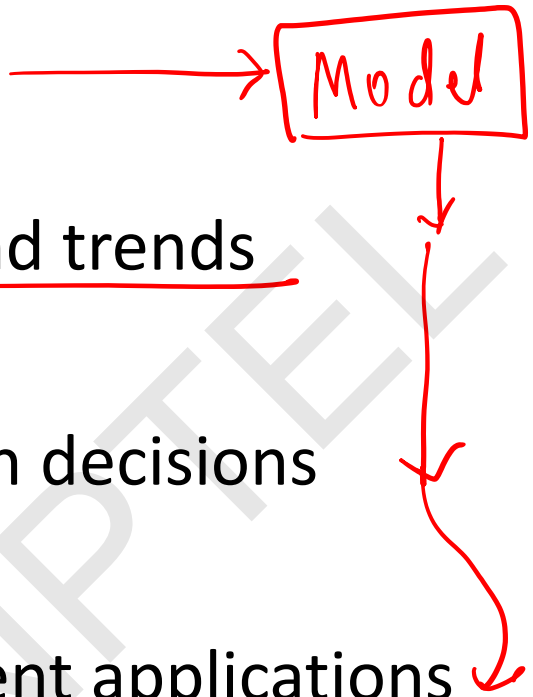
What's in a Name of Machine Learning?

- **Machine learning:** Machine learning has its roots since statistics, artificial intelligence, and computer science among other fields. Machine learning encompasses the algorithms and techniques used to learn from data.
- **Data mining:** The term data mining became popular around the time that the use databases became common place. So data mining was used to refer to activities related to finding patterns in databases and data warehouses. There are some practical data management aspects to data mining related to accessing data from databases. But the process of finding patterns in data is similar, and can use the same algorithms and techniques as machine learning.
- **Predictive analytics** refers to analyzing data in order to predict future outcomes. This term is usually used in the business context to describe activities such as sales forecasting or predicting the purchasing behavior of a customer.
- **Data science** is a new term that is used to describe processing and analyzing data to extract meaning. Again machine learning techniques can also be used here. Because the term data science became popular at the same time that big data began appearing, data science usually refers to extracting meaning from big data and so includes approaches for collecting, storing and managing big data.

ML can be applied in Data Science, Predictive Analytics whereas the Algorithms of Data Mining is used in ML.

Machine Learning Models

- Learn from data
- Discover patterns and trends
- Allow for data-driven decisions
- Used in many different applications



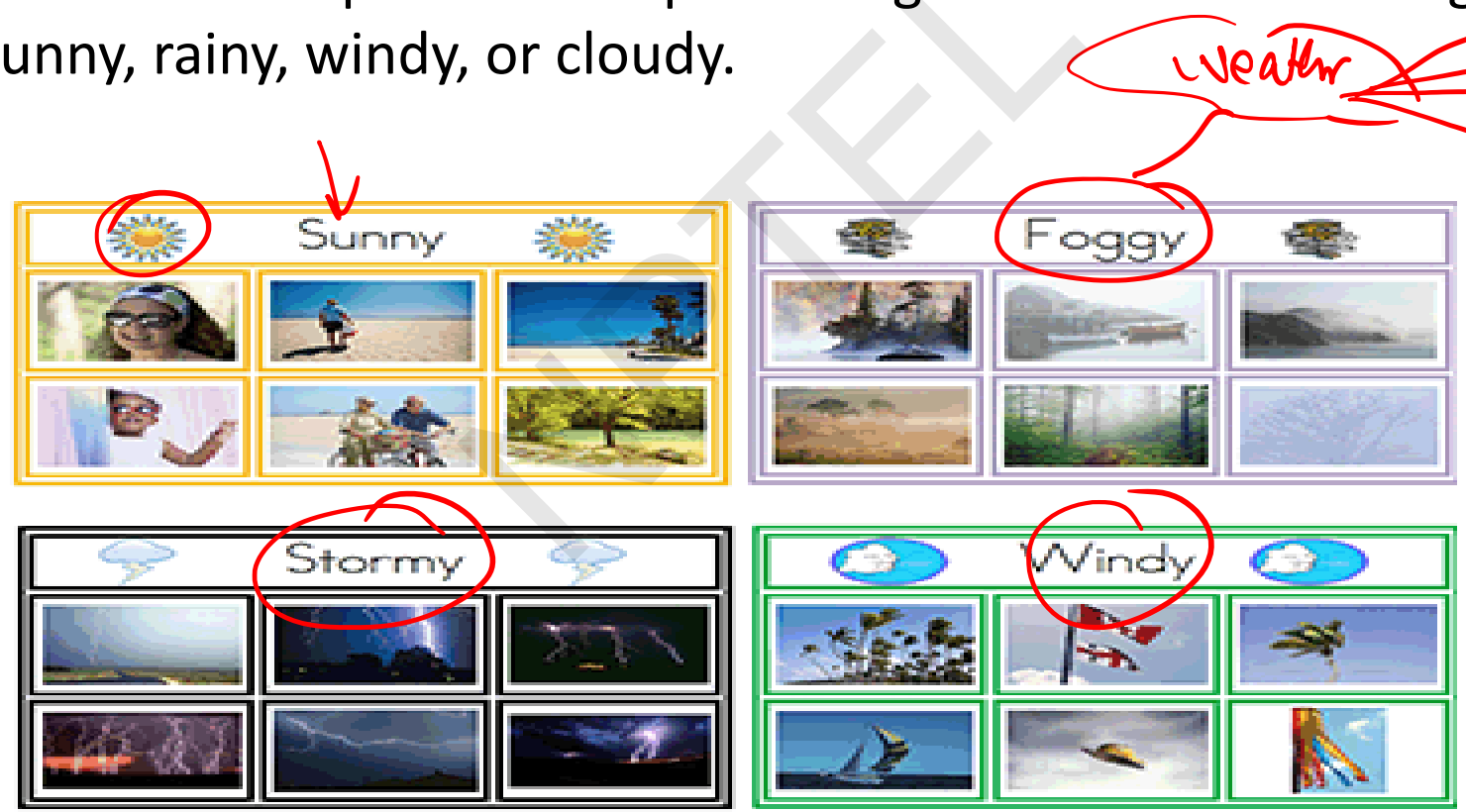
Categories Of Machine Learning Techniques

Categories Of Machine Learning Techniques

- Classification
- Regression
- Cluster Analysis
- Association Analysis

Classification

- **Goal:** Predict category
- In classification, the goal is to predict the category of the input data. An example of this is predicting the weather as being sunny, rainy, windy, or cloudy.



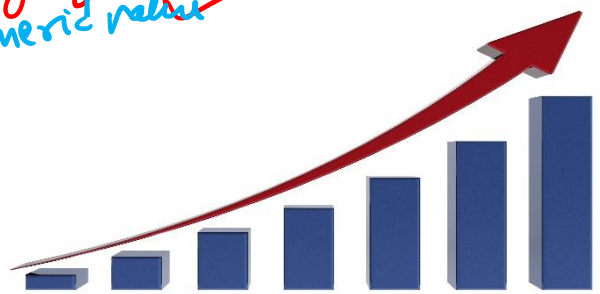
Classification

- Some more examples of classification are classifying a tumor from a medical image as being benign or malignant.
- Predicting whether it will rain the next day.

- Determining if a loan application is high-risk, medium-risk or low-risk.
- Identifying the sentiment of a tweet or review as being positive, negative, or neutral.

Regression

- When your model has to predict a numeric value instead of a category, then the task becomes a regression problem.
- An example of regression is to predict the price of a stock. The stock price is a numeric value, not a category. So this is a regression task instead of a classification task.
- If you were to predict whether the stock price will rise or fall, then that would be a classification problem. But if you're predicting the actual price of the stock, then that is a regression problem.
- That is the main difference between classification and regression. In classification, you're predicting a category and in regression, you're predicting a numeric value.

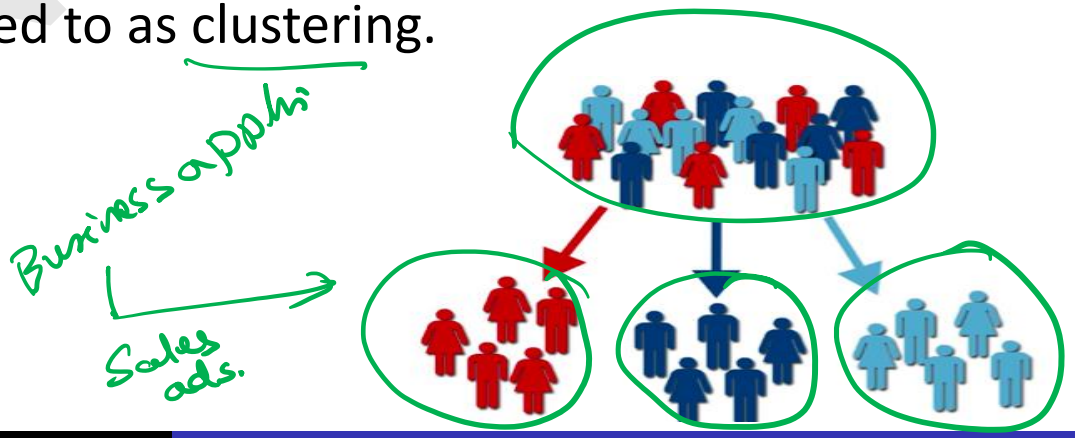


Regression Examples

- Estimating the demand of a product based on time or season of the year.
- Predicting a score on a test.
- Determining the likelihood of how effective a drug will be for a particular patient.
- Predicting the amount of rain for a region.

Cluster Analysis & ML.

- In cluster analysis, the goal is to organize similar items in your data set into groups. A very common application of cluster analysis is referred to as customer segmentation. This means that you're separating your customer base into different groups or segments based on customer types.
- For example it would be very beneficial to segment your customers into seniors, adults and teenagers. These groups have different likes and dislikes and have different purchasing behaviors. By segmenting your customers to different groups you can more effectively provide marketing adds targeted for each groups particular interests. Note that cluster analysis is also referred to as clustering.



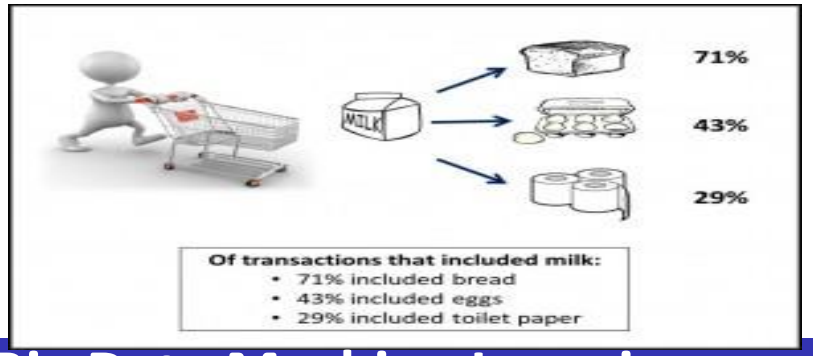
Cluster Analysis Examples

- Some other examples of cluster analysis are:
- Identifying areas of similar topography, such as desert region, grassy areas, mountains etc.
- Categorizing different types of tissues from medical images. Determining different groups of weather patterns, such as snowy, dry, monsoon and
- Discovering hot spots for different types of crime from police reports.

- Grouping together similar data items are called Cluster Analysis or Clustering.

Association Analysis

- The goal in association analysis is to come up with a set of rules to capture associations between items or events. The rules are used to determine when items or events occur together.
- A common application of association analysis is known as market basket analysis. Which is used to understand customer purchasing behavior.
- For example, association analysis can reveal that banking customers who have CDs, or Certificates of Deposits, also tend to be interested in other investment vehicles such as money market accounts.
- This information can be used for cross selling. If you advertise money market accounts to your customers with CDs they are likely to open such an account.



Association Analysis Examples

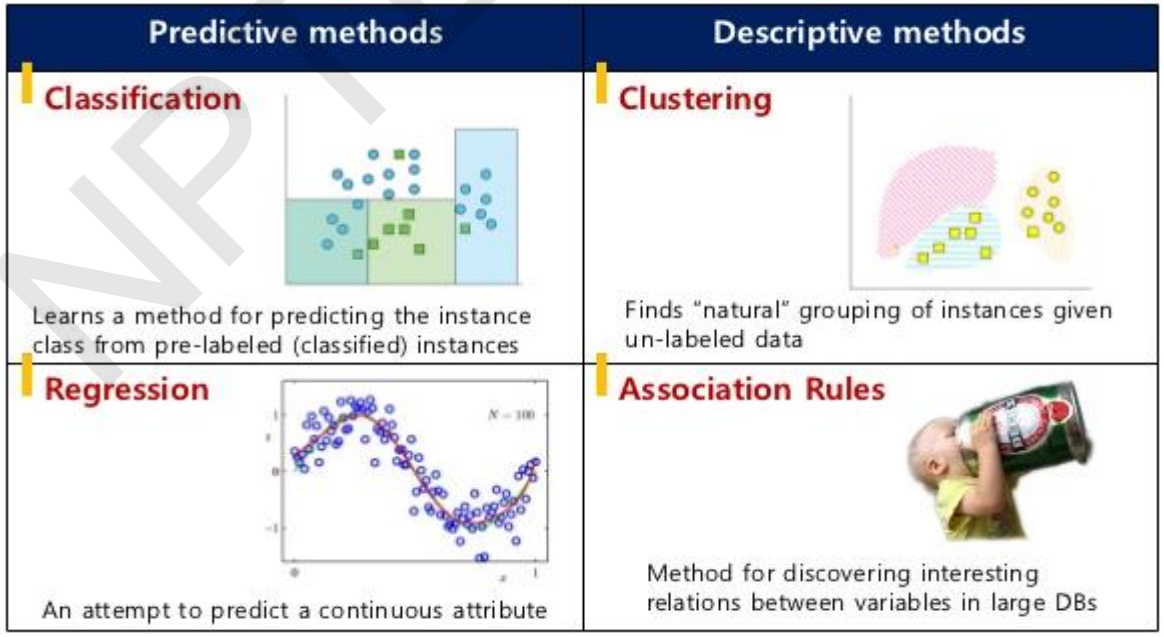
- Some other applications of association analysis are recommending similar items based on the purchasing behavior or browsing histories of customers.

- Finding items that are often purchased together, such as garden hose and potting soil, and offer sales on these related items at the same time to drive sales of both items.

- Identifying web pages that are often accessed together so that you can more efficiently offer up these related web pages at the same time.

Classification of Machine Learning Techniques

- So the different categories of machine learning techniques are classification, regression, cluster analysis, and association analysis.



Supervised vs. Unsupervised

- **In supervised approaches** the target, which is what the model is predicting, is provided. This is referred to as having labeled data because the target is labeled for every sample that you have in your data set.

- Referring back to our example of predicting a weather category of sunny, windy, rainy or cloudy, every sample in the data set is labeled as being one of these four categories. So the data is labeled and predicting the weather categories is a supervised task. In general, classification and regression are supervised approaches.

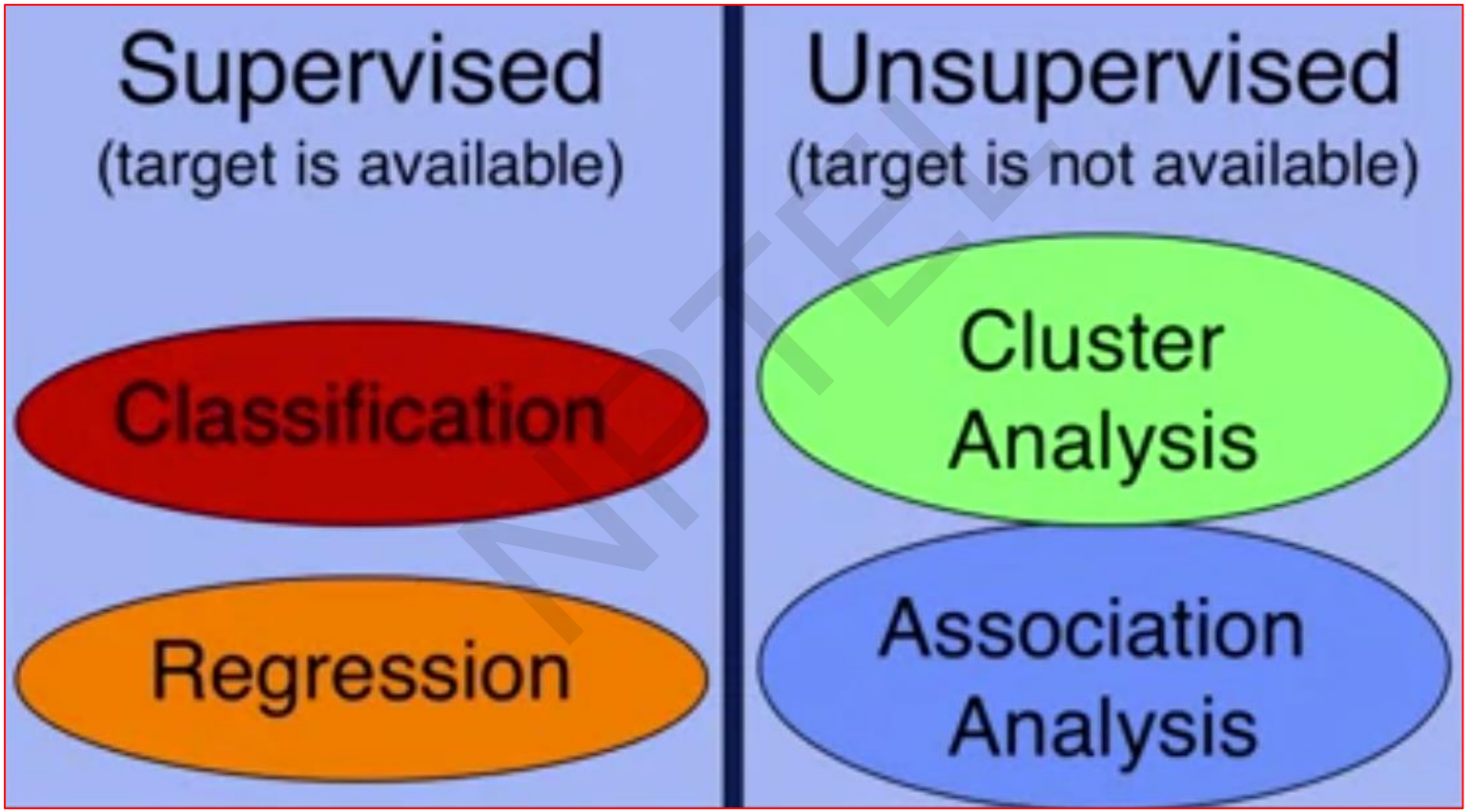
Supervised vs. Unsupervised

- **In unsupervised approaches** on the other hand, the target that the model is predicting is unknown or unavailable. This means that you have unlabeled data.

- Remember the cluster analysis example of segmenting customers into different groups. The samples in your data are not labeled with the correct group. Instead, the segmentation is performed using a clustering technique to group items based on characteristics that they have in common.

- Thus, the data is unlabeled and the task of grouping customers into different segments is an unsupervised one. In general, cluster analysis and association analysis are unsupervised approaches.

Classification of Machine Learning Techniques



Machine Learning Process

Machine Learning Process

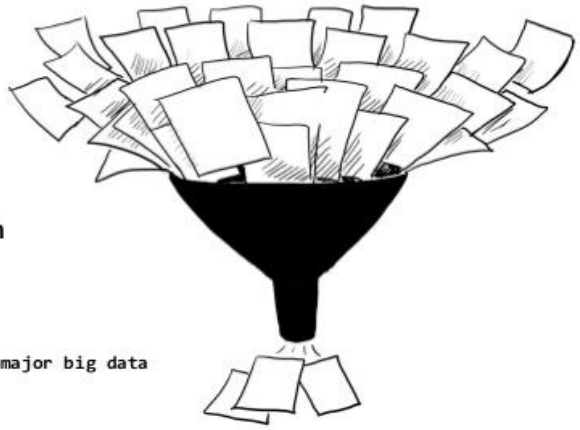
- This diagram illustrates the steps in the machine learning process.
- It should be kept in mind that all of these steps need to be carried out with a clear purpose in mind. That is, the problem or opportunity that is being addressed must be defined with clearly stated goals and objectives.
- For example, the purpose of a project may be to study customer purchasing behavior to come up with a more effective marketing strategy in order to increase sales revenue. The purpose behind the project will drive the machine learning process.

Step 1: Acquire Data

- The first step in the machine learning process is to get all available data related to the problem at hand. Here, we need to identify all data sources, collect the data, and finally integrate data from these multiple sources.

What is Data Acquisition

It is the process of gathering, filtering and cleaning the data before it is put in a data warehouse or any other storage solution on which data analysis can be carried out.



Data acquisition is one of the major big data challenges

Step 2: Prepare Data

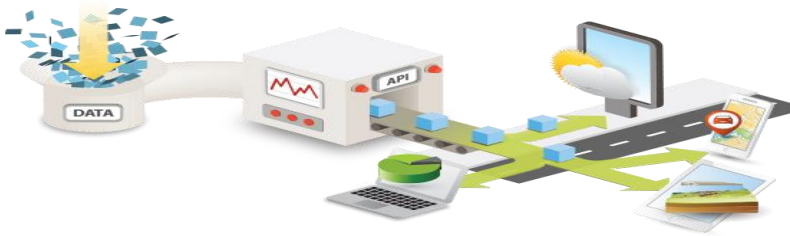
- **This step is further divided into two parts, explore data and pre-process data.**

Data Exploration

- The first part of data preparation involves preliminary exploration of the data to understand the nature of the data that we have to work with. Things we want to understand about the data are its characteristics, format, and quality. A good understanding of the data leads to a more informed analysis and a more successful outcome.

Preprocessing

- Once we know more about the data through exploratory analysis, the next part is pre-processing of the data for analysis. This includes cleaning data, selecting the variables to use, and transforming data to make the data more suitable for analysis in the next step.

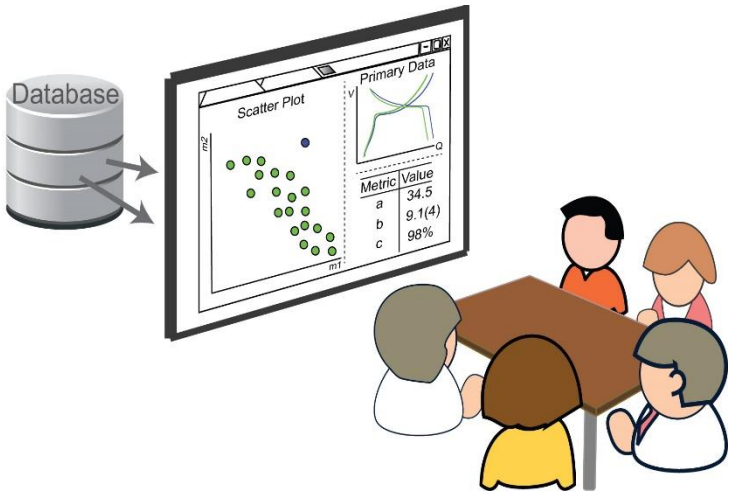


Step 3: Analyze Data

- The prepared data then would be passed on to the analysis step. This step involves selecting the analytical techniques to use, building a model using the data, and assessing the results.

Step 4: Communicate Results

- This includes evaluating the results with respect to the goal set for the project. Presenting the results in a easy to understand way and communicating the results to others.



Step 5: Apply the Results

- The last step is to apply the results. This brings us back to the purpose of the project. How can the insights from our analysis be used to provide effective marketing to increase sales revenue?

- Determining actions from insights gained from analysis is the main focus of the act step.

Iterative Process

- Note that the machine learning process is a very iterative one. Findings from one step may require a previous step to be repeated with new information.

- For example, during the prepare step, we may find some data quality issues that may require us to go back to the acquire step to address some issues with data collection or to get additional data that we didn't include in the first go around.

Goals and Activities in the Machine Learning Process

Goals and Activities in the Machine Learning Process

- Here we will describe the goals of each step and the key activities performed in each step.

Acquire Data

- The first step in the data science process is to acquire the data. The goal of the step is to identify and obtain all data related to the problem at hand. First, we need to identify all related data and the sources. Keep in mind, that data can come from different sources such as files, databases, the internet, mobile devices. So remember to include all data related to the problem you are addressing.

Data sources related to problem at hand

- After you've identified your data and data sources, the next step is to collect the data and integrate data from the different sources. This may require conversion, as data can come in different formats. And it may also require aligning the data, as data from different sources may have different time or spatial resolutions. Once you've collected and integrated your data, you now have a coherent data set for your analysis.

Prepare Data

- The next step after acquiring data is to prepare it to make it suitable for analysis.

- There are two parts to this step, explore data and preprocess data.

Step-2-A: Explore

- In data exploration, you want to do some preliminary investigation in order to gain a better understanding of the specific characteristics of your data. This in turn will guide the rest of the process. With data exploration, you'll want to look for things like correlations, general trends, outliers, etc.

- Correlations provide information about the relationship between variables in your data. Trends in your data will reveal if the variable is moving in a certain direction, such as transaction volume increasing throughout the year.

- Outliers indicate potential problems with the data, or may indicate an interesting data point that needs further examination. Without this data exploration activity, you will not be able to use your data effectively.

Describe your Data

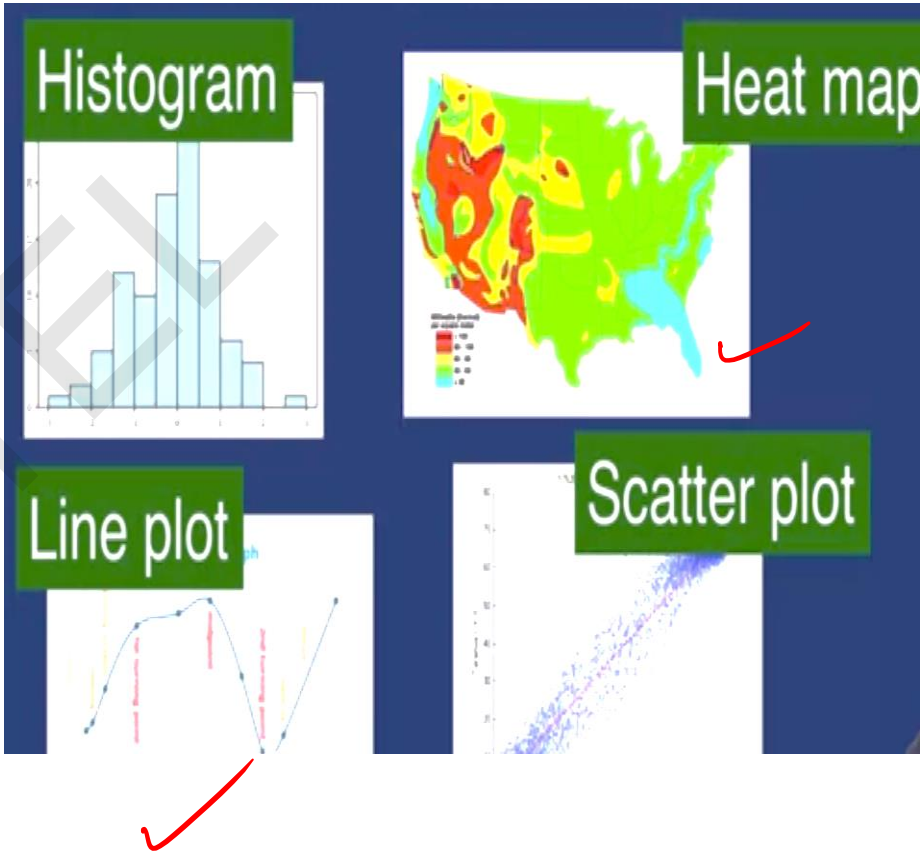
- One way to explore your data is to calculate summary statistics to numerically describe the data.

- Summary statistics are quantities that capture various characteristics of a set of values with a single number, or a small set of numbers. Some basic summary statistics that you should compute for your data set are mean, median, mode, range and standard deviation. Mean and median are measures of the location of a set of values. Mode is the value that occurs most frequently in your data set, and range and standard deviation are measures of spread in your data. Looking at these measures will give you an idea of the nature of your data. They can tell you if there's something wrong with your data.

- For example, if the range of the values for age in your data includes negative numbers, or a number much greater than a hundred, there's something suspicious in the data that needs to be examined

Visualize Your Data

- Visualization techniques also provide quick and effective ways to explore your data. Some examples are, a histogram, such as the plot shown here, shows the distribution of the data and can show skewness or unusual dispersion in outliers.
- A line plot, like the one in the lower left, can be used to look at trends in the data, such as, the change in the price of a stock. A heat map can give you an idea of where the hot spots are.
- A scatter plot effectively shows correlation between two variables. Overall, there are many types of plots to visualize data. They are very useful in helping you understand the data you have.



Step-2-B: Pre-Process

- The second part of the prepare step is preprocess. So, after we've explored the data, we need to preprocess the data to prepare it for analysis.
- The goal here is to create the data that will be used for analysis.
- The main activities on this part are to clean the data, select the appropriate variables to use and transform the data as needed.

Data Cleaning

- A very important part of data preparation is to clean the data to address quality issues. Real world data is nothing. There are many examples of quality issues with data from real applications including missing values, such as income in a survey, duplicate data, such as two different records for the same customer with different addresses.

- Inconsistent or invalid data, such as a six digit zip code. Noise in the collection of data that distorts the true values. Outliers, such as a number much larger than 100 for someone's age. It is essential to detect and address these issues that can negatively affect the quality of the data.

Feature Selection

- Feature selection refers to choosing the set of features to use that is appropriate for the application. Feature selection can involve removing redundant or irrelevant features, combining features, or creating new features.
- During the exploring data step, you may have discovered that two features are very correlated. In that case, one of these features can be removed without negatively effecting the analysis results.
- For example, the purchase price of a product and the amount of sales tax pain are very likely to be correlated. Eliminating the sales tax amount then will be beneficial. Removing redundant or irrelevant features will make the subsequent analysis simpler.
- You may also want to combine features or create new ones. For example, adding the applicants education level as a feature to a loan approval application would make sense. There are also algorithms to automatically determine the most relevant features based on various mathematical properties.

Feature Transformation

- Feature transformation maps the data from one format to another. Various transformation operations exist. For example, scaling maps the data values to a specified range to prevent any one feature from dominating the analysis results.

- Filtering or aggregation can be used to reduce noise and variability in the data.

- Dimensionality reduction maps the data to a smaller subset of dimensions to simplify the subsequent analysis. We will discuss techniques to prepare data in more detail later in this course.

Step-3: Analyze

- After preparing the data to address data quality issues and preprocess it to get it in the appropriate format, the next step in the machine learning process is to analyze the data. The goals of the staff are to build a machine learning model, to analyze the data and to evaluate the results that you get from the model.

- The analyze steps starts with this determining the type of problem you have. You begin by selecting appropriate machine learning techniques to analyze the data.

- Then you construct the model using the data that you've prepared. Once the model is built, you will want to apply it to new data samples to evaluate how well the model performs. Thus data analysis involves selecting the appropriate technique for your problem, building the model, then evaluating the results.

Step-4: Report

- The next step in the machine learning process is reporting results from your analysis. In reporting your results, it is important to communicate your insights and make a case for what actions should follow.
- In reporting your results, you will want to think about what to present, as well as how to present. In deciding what to present, you should consider what the main results are, what insights were gained from your analysis, and what added value do these insights bring to the application.
- Keep in mind that even negative results are valuable lessons learned, and suggest further avenues for additional analysis. Remember that all findings must be presented so that informs decisions can be made for next steps.
- In deciding how to present, remember that visualization is an important tool in presenting your results.
- Plots and summary statistics discussing the explore step can be used effectively here as well. You should also have tables with details from your analysis as backup, if someone wants to take a deeper dive into the results.
- In summary, you want to report your findings by presenting your results and the value added with graphs using visualization tools.

Step-5: Act

- The final step in the machine loading process is to determine what action should be taken based on the insights gained.

- What action should be taken based on the results of your analysis? Should you market certain products to a specific customer segment to increase sales? What inefficiency is can be removed from your process? What incentives would be effective in attracting new customers?

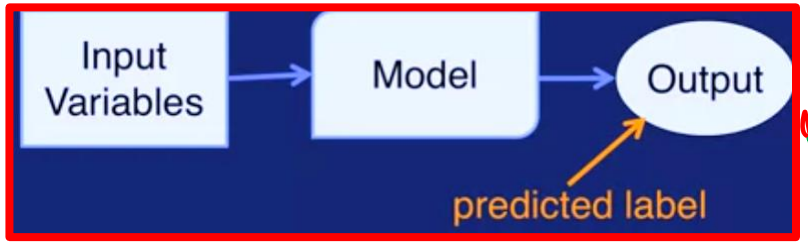
- Once a specific action has been determined, the next step is to implement the action. Things to consider here include, how can the action be added to your application? How will end users be affected?

- Assessing the impact of the implemented action is then necessary to evaluate the benefits gained. The results of this assessment determine next steps, which could suggest additional analysis or further opportunities, which would begin another cycle of the machine learning process.

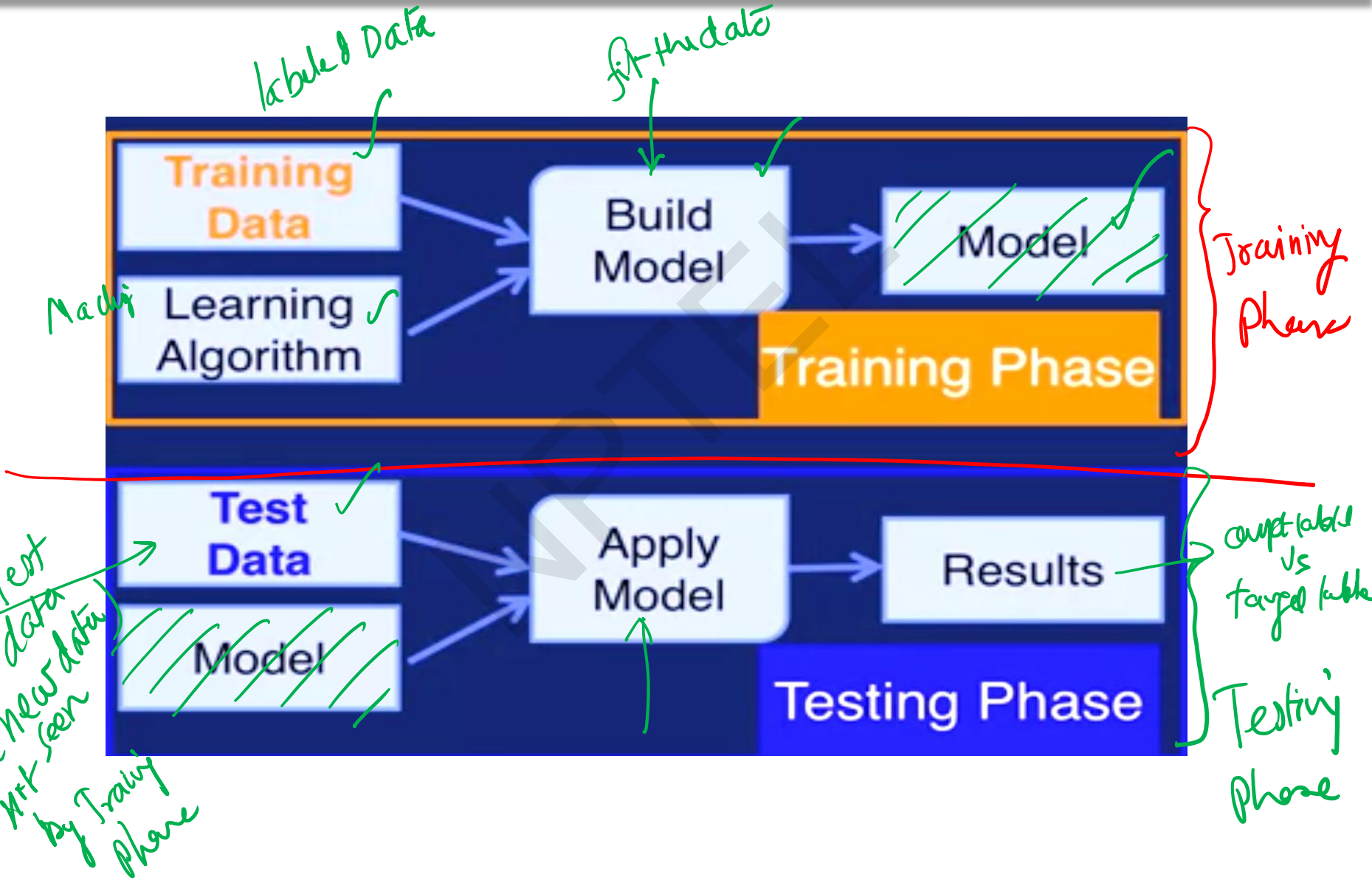
Generalization and Overfitting

Errors in Classification

- Recall that a machine learning model maps the input it receives to an output. For a classification model, the model's output is the predicted class label for the input variables and the true class label is the target.
- Then if the classifier predicts the correct classes label for a sample, that is a success. If the predicted class label is different from the true class label, then that is an error.
- The error rate, then, is the percentage of errors made over the entire data set. That is, it is the number of errors divided by the total number of samples in a data set.
- Error rate is also known as misclassification rate, or simply error.



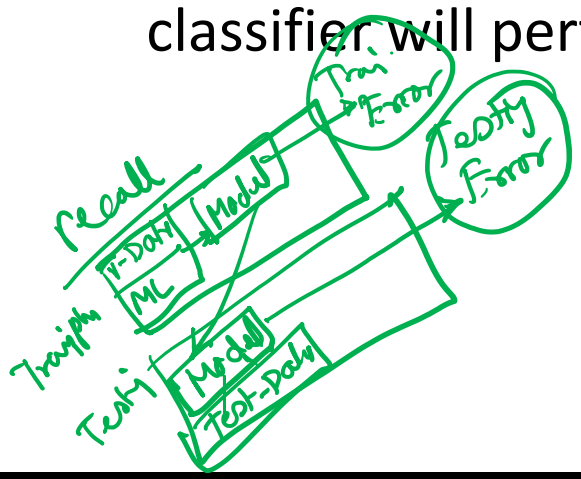
Training vs. Testing Phase



Errors in Classification

- The model is built using training data and evaluated on test data. The training and test data are two different data sets. The goal in building a machine learning model is to have the model perform well on training, as well as test data.

- Error rate, or simply error, on the training data is referred to as training error, and the error on test data is referred to as test error. The error on the test data is an indication of how well the classifier will perform on new data.



Generalization

- This is known as generalization. Generalization refers to how well your model performs on new data, that is data not used to train the model.

- You want your model to generalize well to new data. If your model generalizes well, then it will perform well on data sets that are similar in structure to the training data, but doesn't contain exactly the same samples as in the training set.

- Since the test error indicates how well your model generalizes to new data, note that the test error is also called generalization error.

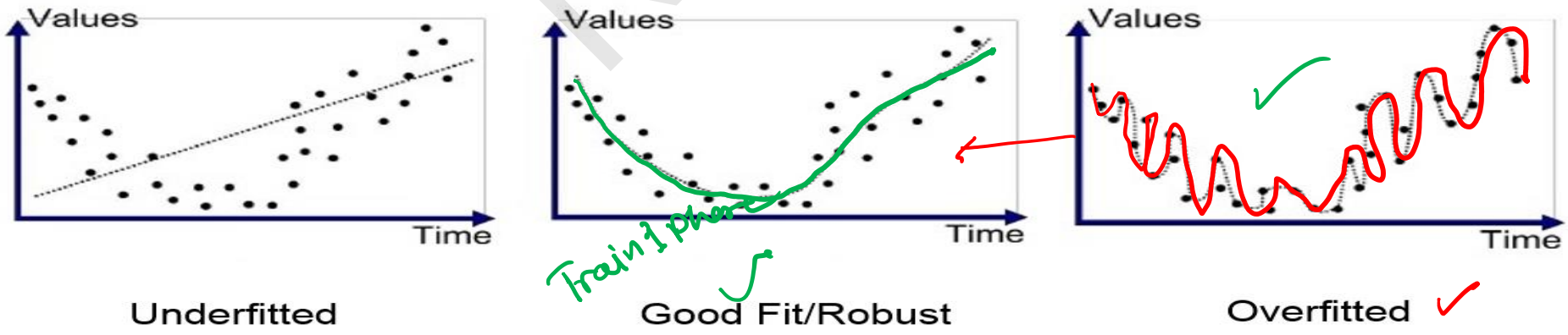
Overfitting

- A related concept to generalization is overfitting. If your model has very low training error but high generalization error, then it is overfitting.

- This means that the model has learned to model the noise in the training data, instead of learning the underlying structure of the data.

Overfitting

- These plots illustrate what happens when a model overfits. Training samples are shown as points, and the input to output mapping that the model has learned is indicated as a curve. The plot on the left shows that the model has learned the underlying structure of the data, as the curve follows the trend of the sample point as well. The plot on the right, however, shows that the model has learned to model the noise in a data set.
- The model tries to capture every sample point, instead of the general trend of the samples together. The training error and the generalization error are plotted together, during model training.



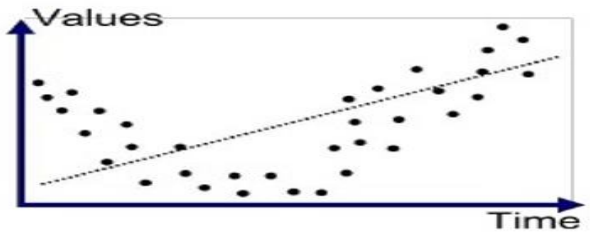
Connection between overfitting and generalization

- A model that overfits will not generalize well to new data.
- So the model will do well on just the data it was trained on, but given a new data set, it will perform poorly.

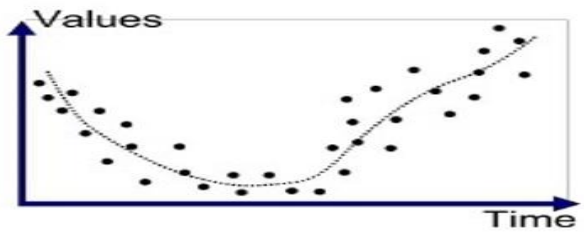
- A classifier that performs well on just the training data set will not be very useful. So it is essential that the goal of good generalization performance is kept in mind when building a model.

Overfitting and Underfitting

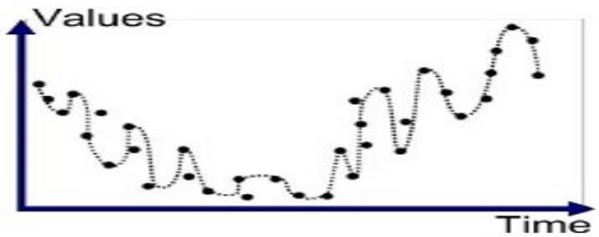
- Overfitting occurs when the model is fitting to the noise in the training data. This results in low training error and high test error.
- Underfitting on the other hand, occurs when the model has not learned the structure of the data. This results in high training error and high test error.
- Both are undesirable, since both mean that the model will not generalize well to new data. Overfitting generally occurs when a model is too complex, that is, it has too many parameters relative to the number of training samples. So to avoid overfitting, the model needs to be kept as simple as possible, and yet still solve the input/output mapping for the given data set.



Underfitted



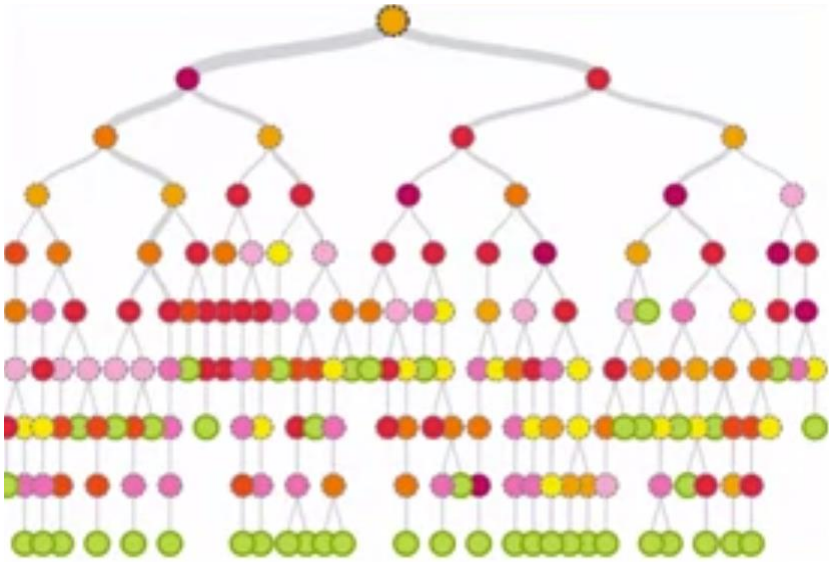
Good Fit/Robust



Overfitted

What Causes Overfitting

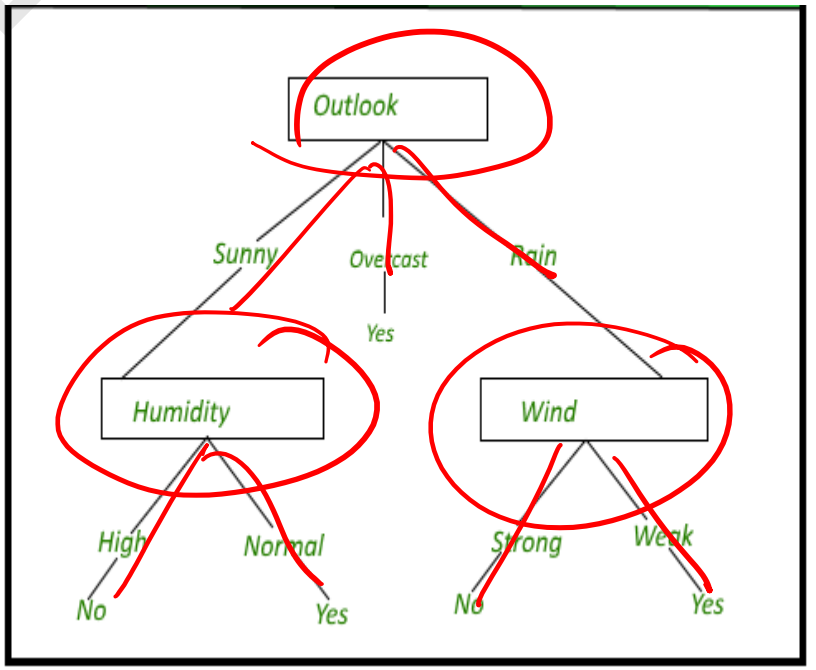
- In summary, overfitting is when your model has learned the noise in the training data instead of the underlying structure of the data. You want to avoid overfitting so that your model will generalize well to new data.



Overfitting in Decision Trees

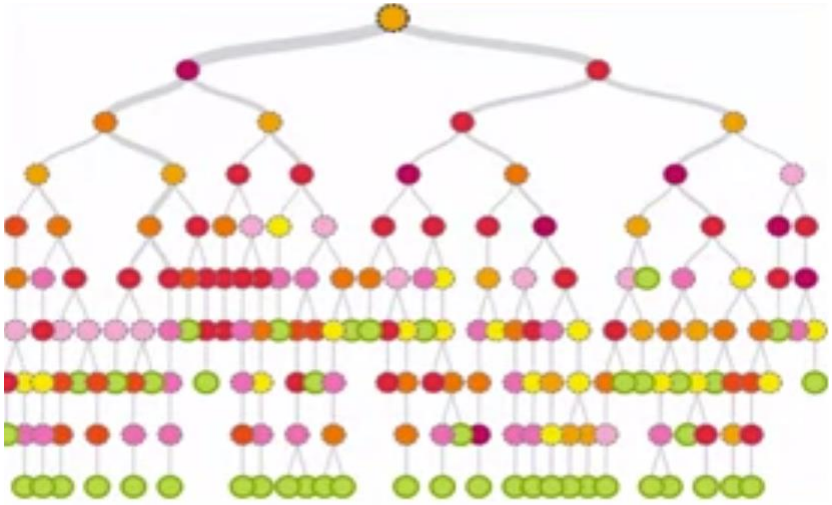
Decision Tree Induction

- A decision tree, also referred to as tree induction, the tree repeatedly splits the data in a node in order to get successively paired subsets of data.

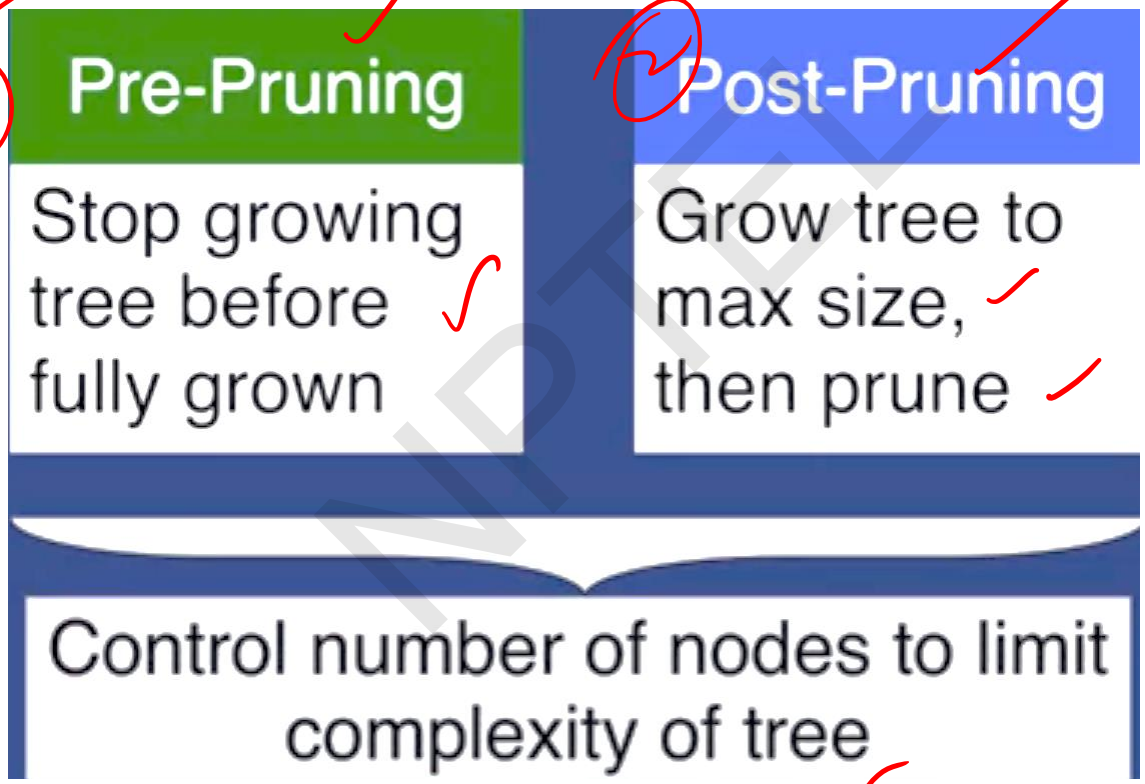


Overfitting in Decision Trees

- Note that a decision tree classifier can potentially expand its nodes until it can perfectly classify samples in the training data.
- But if the tree grows nodes to fit the noise in the training data, then it will not classify a new sample well.
- This is because the tree has partitioned the input space according to the noise in the data instead of to the true structure of a data. In other words, it has overfit.



Avoiding Overfitting in Decision Trees



Pre-pruning

- With pre-pruning, the idea is to stop tree induction before a fully grown tree is built that perfectly fits the training data.
-
- To do this, restrictive stopping conditions for growing nodes must be used. For example, a nose stops expanding if the number of samples in the node is less than some minimum threshold.
- Another example is to stop expanding a note if the improvement in the impurity measure falls below a certain threshold.

Post-pruning

- In post-pruning, the tree is grown to its maximum size, then the tree is pruned by removing nodes using a bottom up approach.

- That is, the tree is trimmed starting with the leaf nodes. The pruning is done by replacing a subtree with a leaf node if this improves the generalization error, or if there is no change to the generalization error with this replacement.

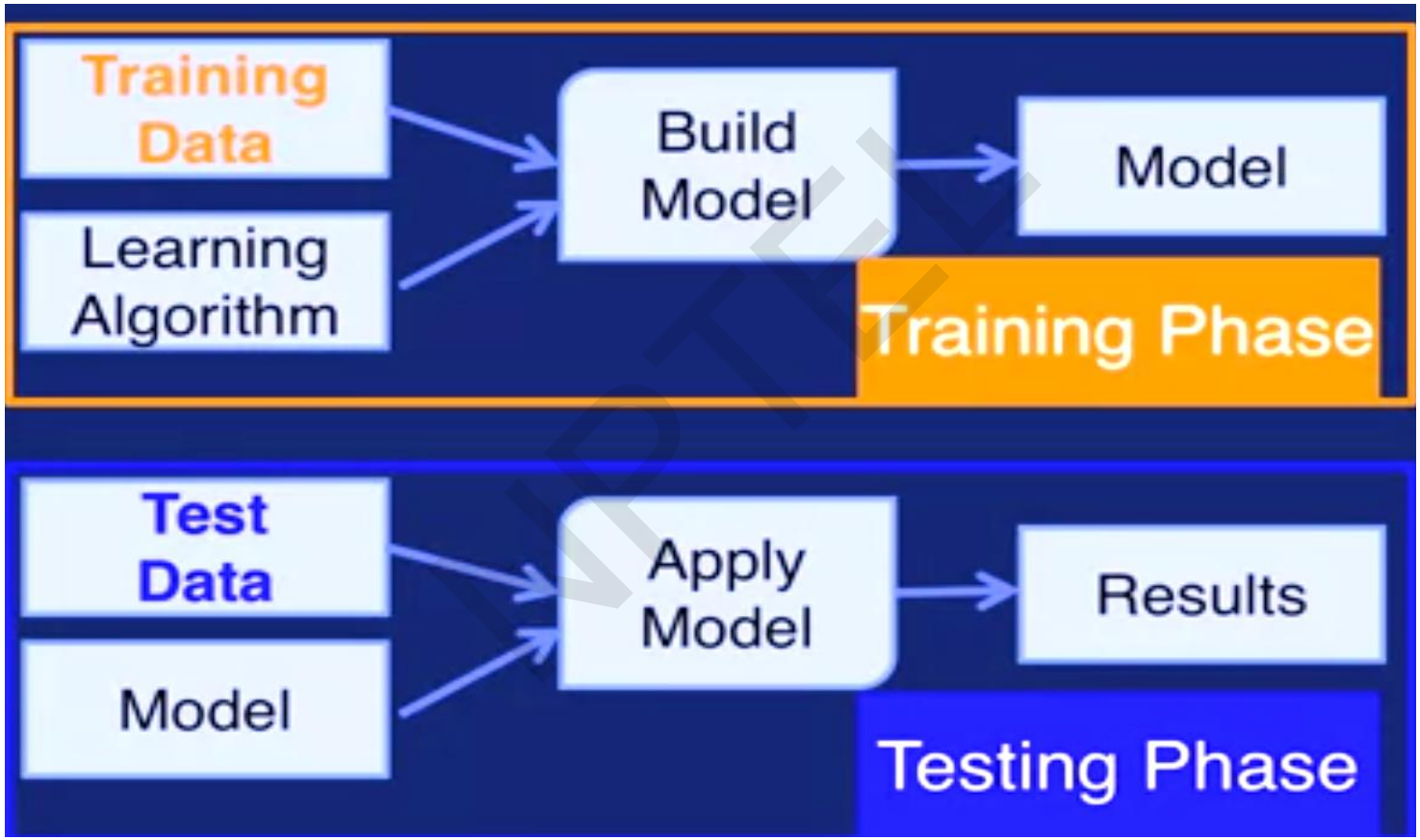
Overfitting in Decision Trees

- In other words, if removing a subtree does not have a negative effect on the generalization error, then the nodes in that subtree only add to the complexity of the tree, and not to its overall performance.

- So those nodes should be removed. In practice, post-pruning tends to give better results. This is because pruning decisions are based on information from the full tree. Pre-pruning, on the other hand, may stop the tree growing process prematurely. However, post-pruning is more computationally expensive since the tree has to be expanded to its full size.

Validation Set

Training vs. Testing Phase

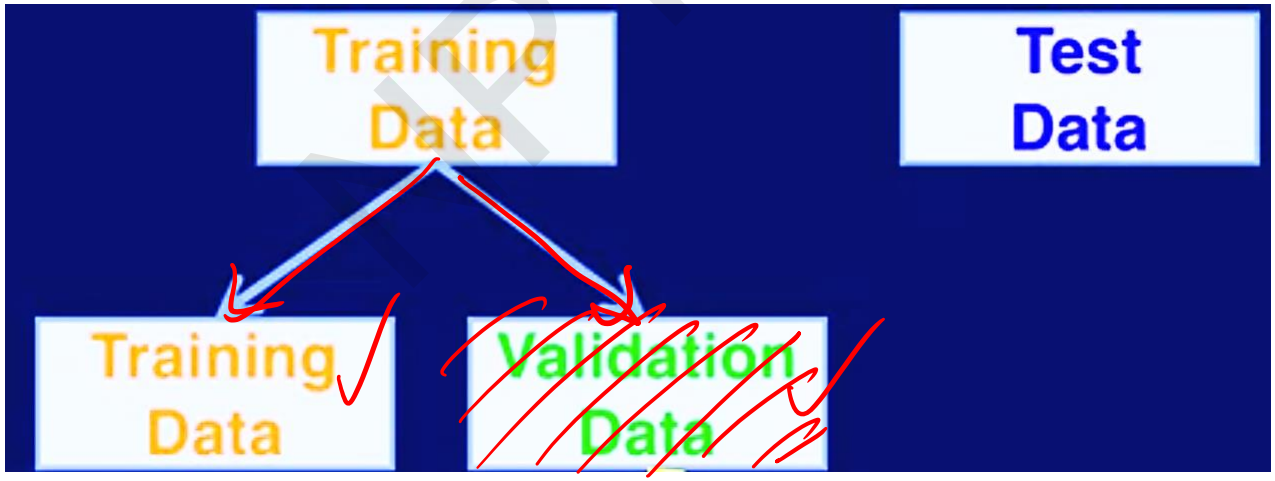


Avoiding Overfitting

- Recall that a model that overfits does not generalize well to new data.
- Recall also that overfitting generally occurs when a model is too complex.
- So to have a model with good generalization performance, model training has to stop before the model gets too complex.
- How do you determine when this should occur?

Validation Set

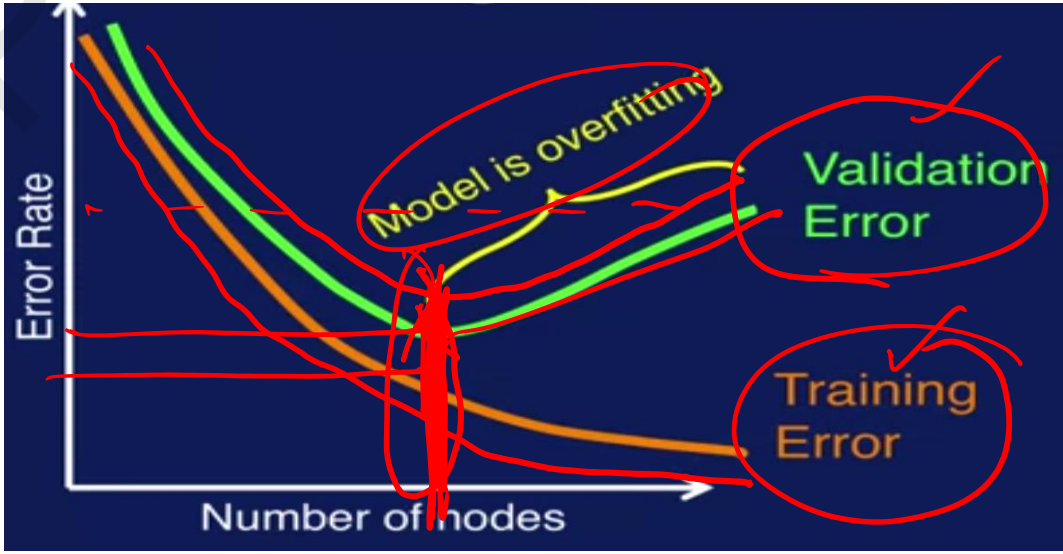
- A validation set can be used to guide the training process to avoid overfitting and deliver good generalization performance.
- We have discussed having a training set and a separate test set. The training set is used to build a model and the test set is used to see how the model performs a new data.



Training and Validation Error

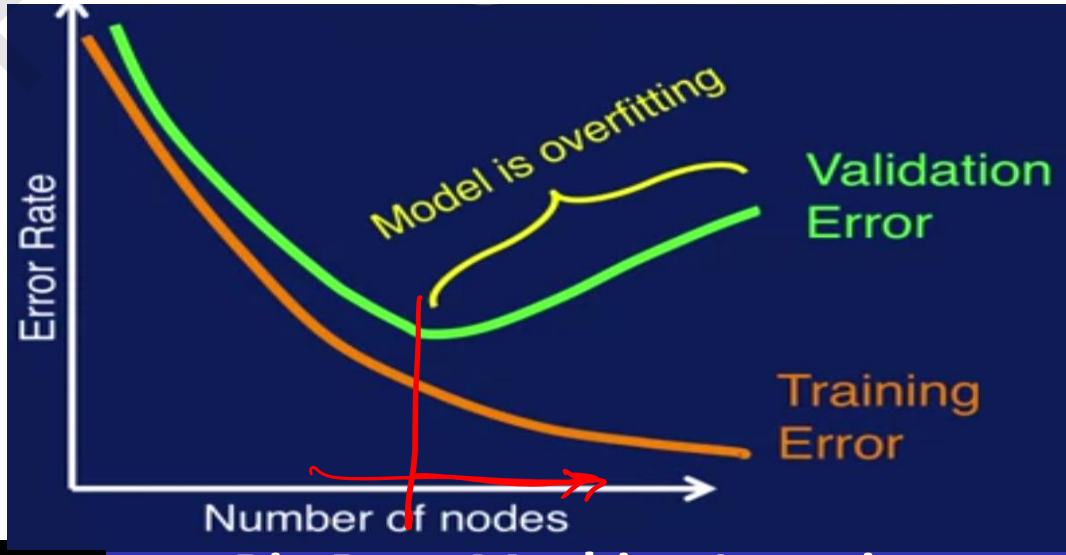
- Now we want to further divide up the training data into a training set and a validation set.

- The training set is used to train the model as before and the validation set is used to determine when to stop training the model to avoid overfitting, in order to get the best generalization performance.



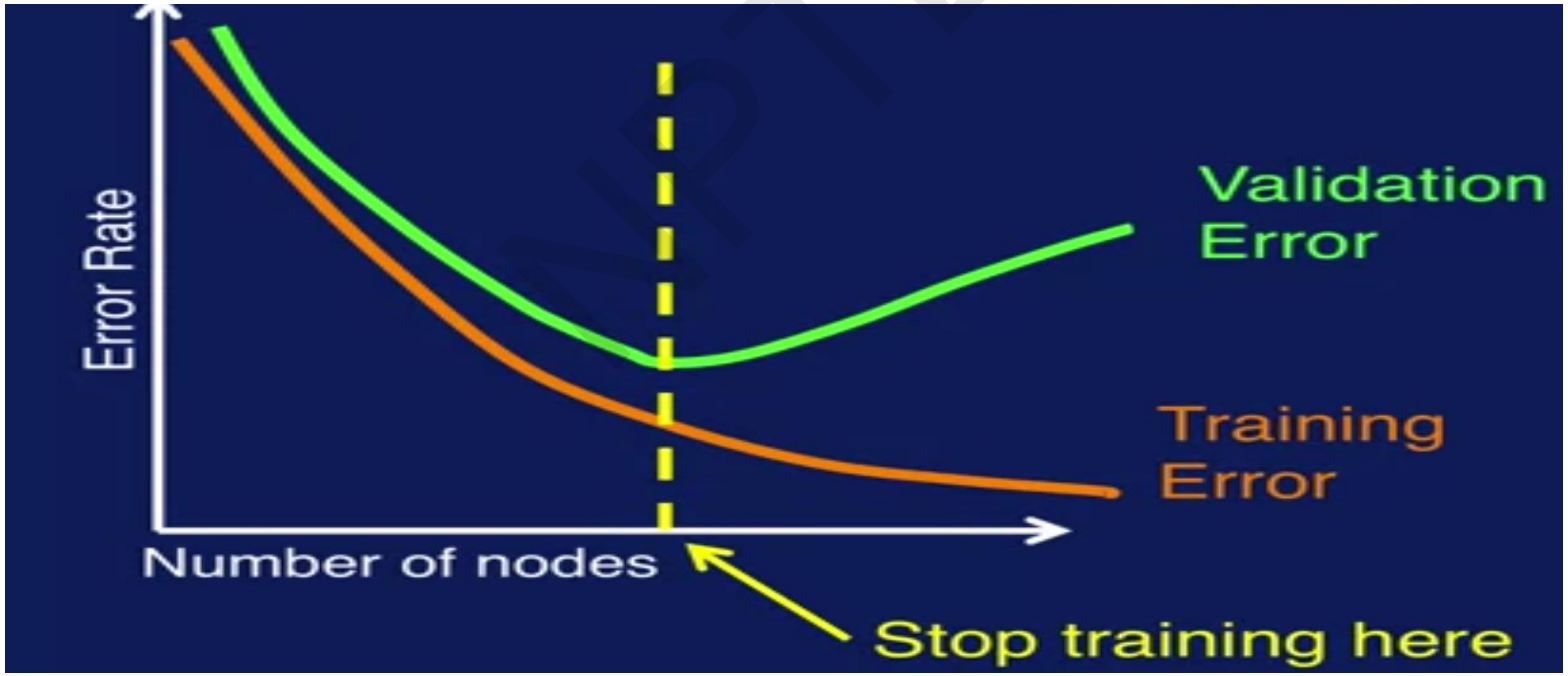
Training and Validation Error

- The idea is to look at the errors on both training set and validation set during model training as shown here. The orange solid line on the plot is the training error and the green line is the validation error. We see that as model building progresses along the x-axis, the number of nodes increases. That is the complexity of the model increases. We can see that as the model complexity increases, the training error decreases. On the other hand, the validation error initially decreases but then starts to increase.
- When the validation error increases, this indicates that the model is overfitting, resulting in decreased generalization performance.



When to Stop Training ?

This can be used to determine when to stop training. Where validation error starts to increase is when you get the best generalization performance, so training should stop there. This method of using a validation set to determine when to stop training is referred to as model selection since you're selecting one from many of varying complexities. Note that this was illustrated for a decision tree classifier, but the same method can be applied to any type of machine learning model.

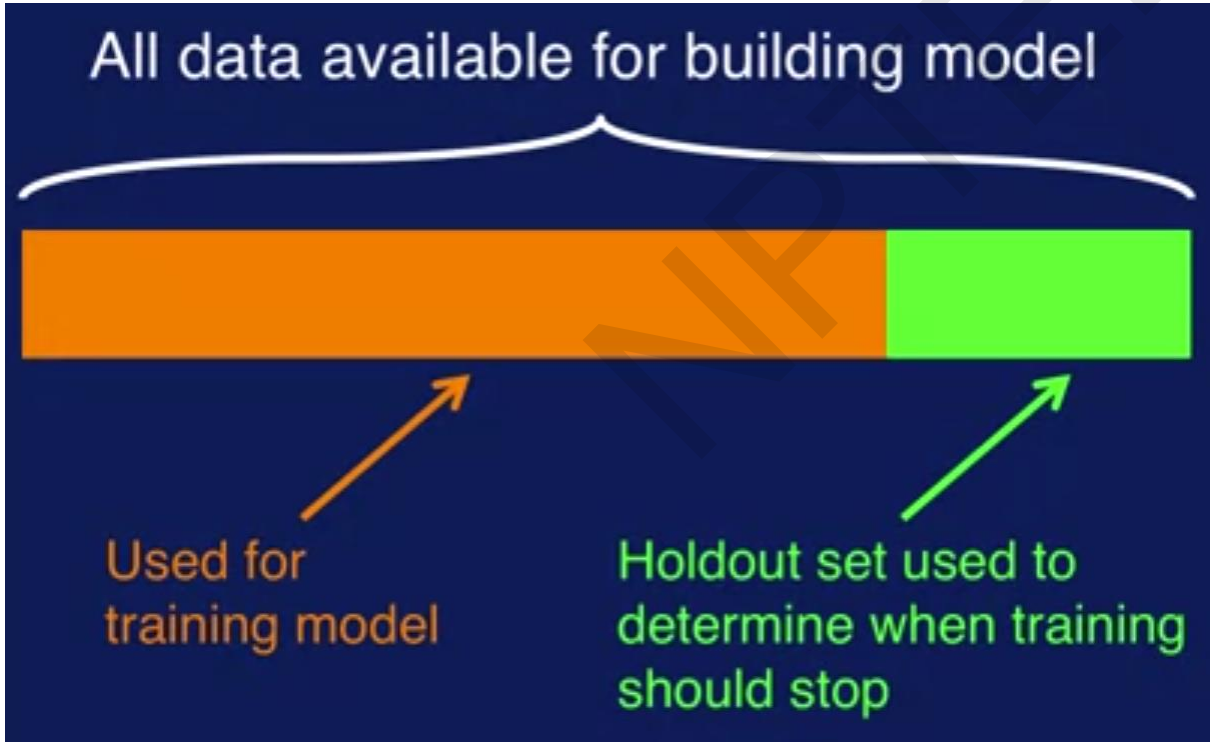


Ways to create and use the validation set

- There are several ways to create and use the validation set to avoid overfitting. The different methods are:
- Holdout method
- Random subsampling
- K-fold cross-validation, and
- Leave-one-out cross-validation

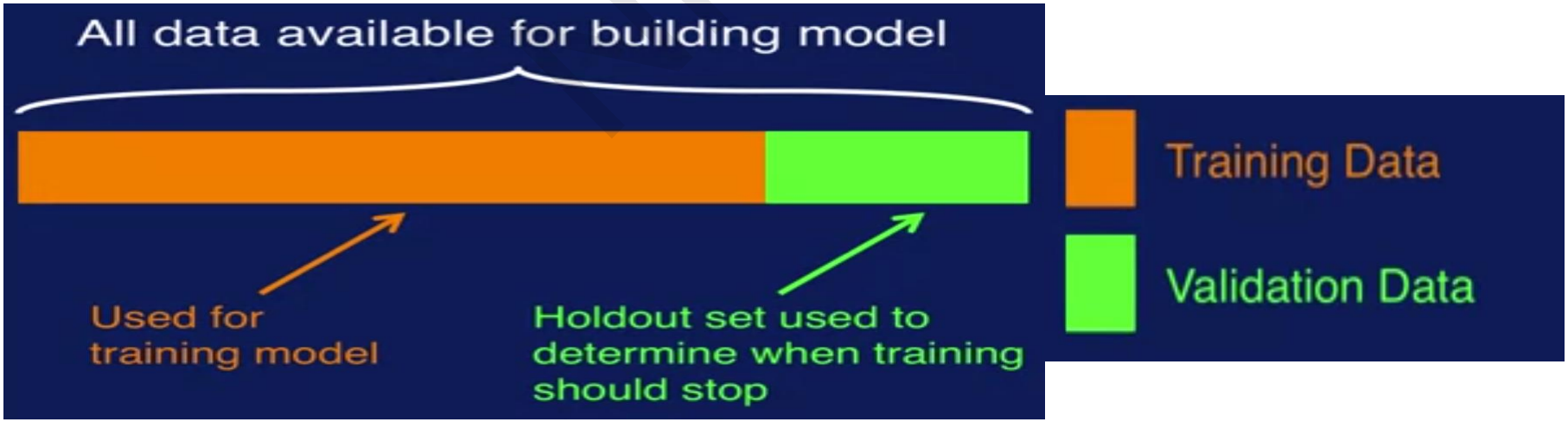
Holdout Method

- The first way to use a validation set is the holdout method. This describes the scenario that we have been discussing, where part of the training data is reserved as a validation set. The validation set is then the holdout set.



Holdout Method

- Errors on the training set and the holdout set are calculated at each step during model training and plotted together as we've seen before. And the lowest error on the holdout set is when training should stop. This is the just the process that we have described here before. There's some limitations to the holdout method however.
- First, since some samples are reserved for the holdout validation set, the training set now has less data than it originally started out with.
- Secondly, if the training and holdout sets do not have the same data distributions, then the results will be misleading. For example, if the training data has many more samples of one class and the holdout dataset has many more samples of another class.

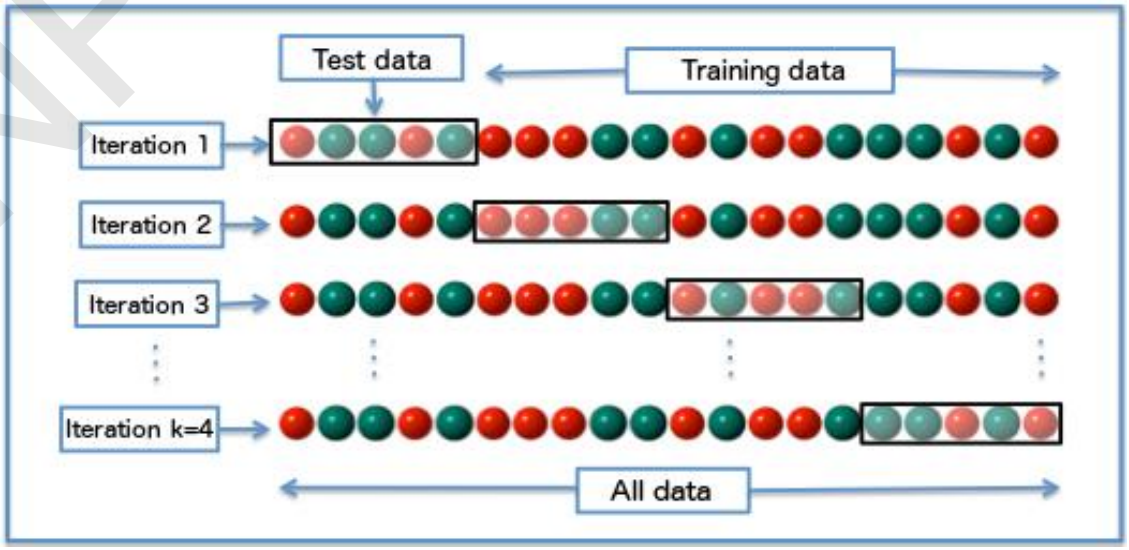


Repeated Holdout Method

- Repeating holdout method several times
- Randomly select different hold out set each iteration
- Average validation errors over all repetitions.

K-Fold Cross-Validation

- A way to improve on the repeated holdout method is use cross-validation. Cross-validation works as follows. Segment the data into k number of disjoint partitions. During each iteration, one partition is used as the validation set. Repeat the process k times. Each time using a different partition for validation. So each partition is used for validation exactly once. This is illustrated in this figure. In the first iteration, the first partition, is used for validation. In the second iteration, the second partition is used for validation and so on.



K-Fold Cross-Validation

- The overall validation error is calculated by averaging the validation errors for all k iterations.

- The model with the smallest average validation error then is selected. The process we just described is referred to as k-fold cross-validation. This is a very commonly used approach to model selection in practice.

- This approach gives you a more structured way to divide available data up between training and validation datasets and provides a way to overcome the variability in performance that you can get when using a single partitioning of the data.

Leave-One-Out Cross-Validation

- Leave-one-out cross-validation is a special case of k-fold cross-validation where k equals N, where N is the size of your dataset.
- Here, for each iteration the validation set has exactly one sample. So the model is trained to using N minus one samples and is validated on the remaining sample.
- The rest of the process works the same way as regular k-fold cross-validation.
- Note that cross-validation is often abbreviated CV and leave-one-out cross-validation is in abbreviated L-O-O-C-V and pronounced LOOCV.

Uses of Validation Set

- Note that the validation error that comes out of this process can also be used to estimate generalization performance of the model.

- In other words, the error on the validation set provides an estimate of the error on the test set.

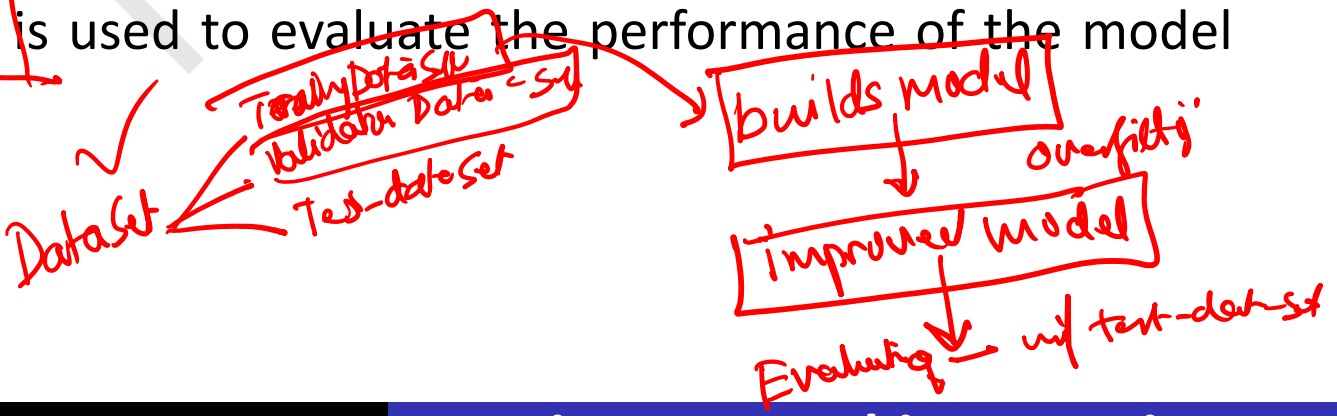
Uses of Datasets

- With the addition of the validation set, you really need three distinct datasets when you build a model. Let's review these datasets.

1. The training dataset is used to train the model, that is to adjust the parameters of the model to learn the input to output mapping.

2. The validation dataset is used to determine when training should stop in order to avoid overfitting.

3. The test data set is used to evaluate the performance of the model on new data.



Uses of Datasets

- Note that the test data set should never, ever be used in any way to create or tune the model. It should not be used, for example, in a cross-validation process to determine when to stop training.

- The test dataset must always remain independent from model training and remain untouched until the very end when all training has been completed. Note that in sampling the original dataset to create the training, validation, and test sets, all datasets must contain the same distribution of the target classes.

- For example, if in the original dataset, 70% of the samples belong to one class and 30% to the other class, then this same distribution should approximately be present in each of the training, validation, and test sets. Otherwise, analysis results will be misleading.

Validation Set Summary

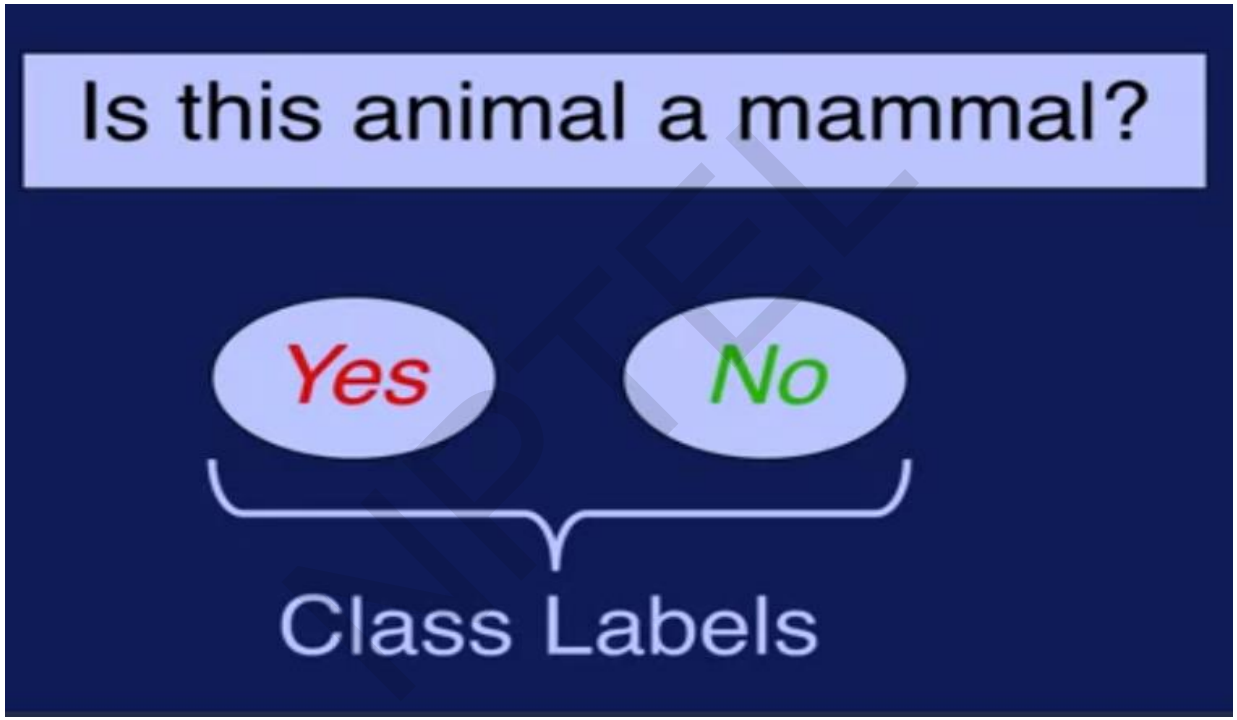
- We have discuss the need for three different datasets in building model. A training set to train the model, a validation set to determine when to stop training, and a test to evaluate performance on new data.

- We learned how a validation set can be used to avoid overfitting and in the process, provide an estimate of generalization performance.

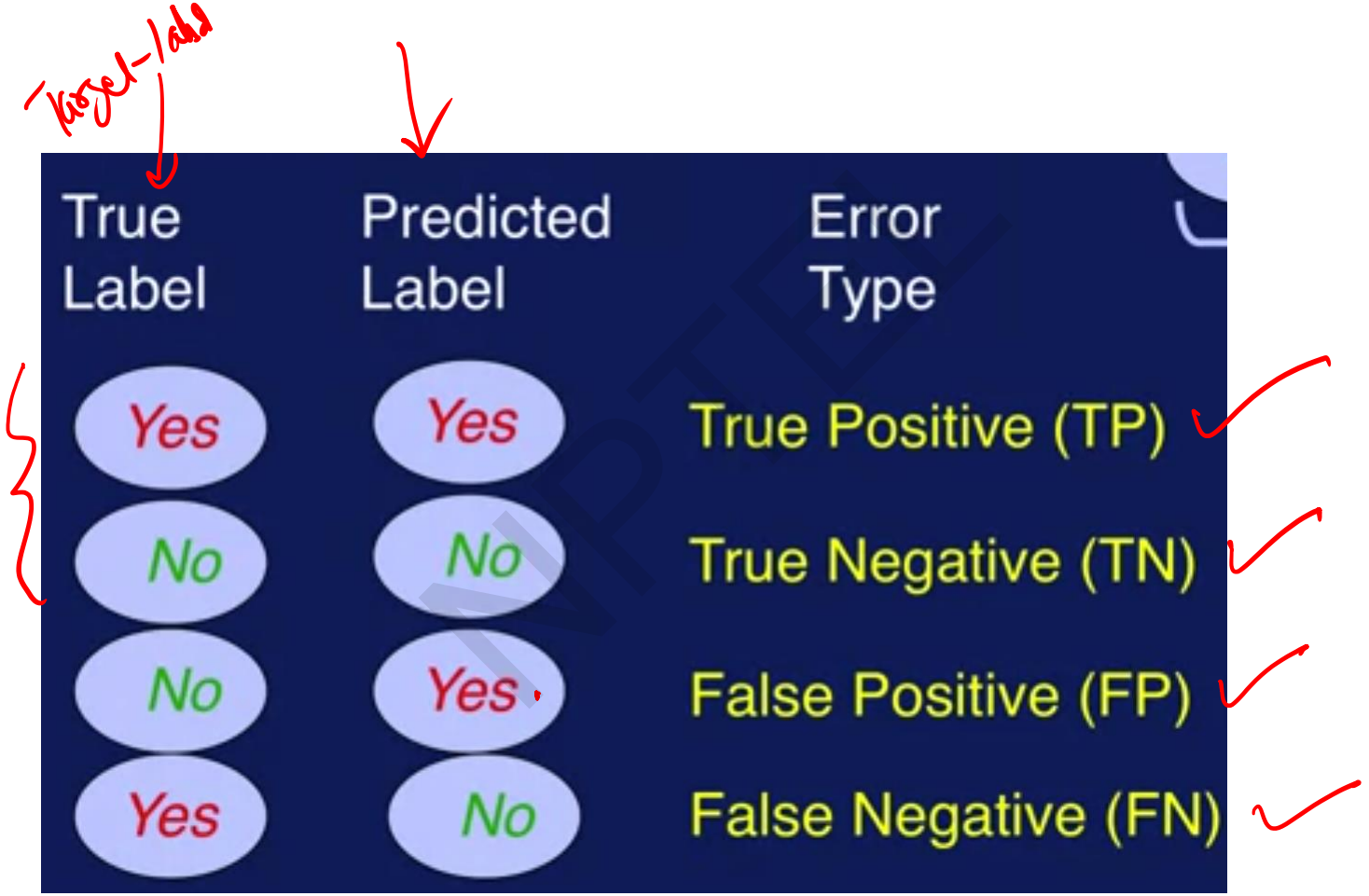
- And we covered different ways to create and use a validation set such as k-fold cross-validation.

Metrics to Evaluate Model Performance

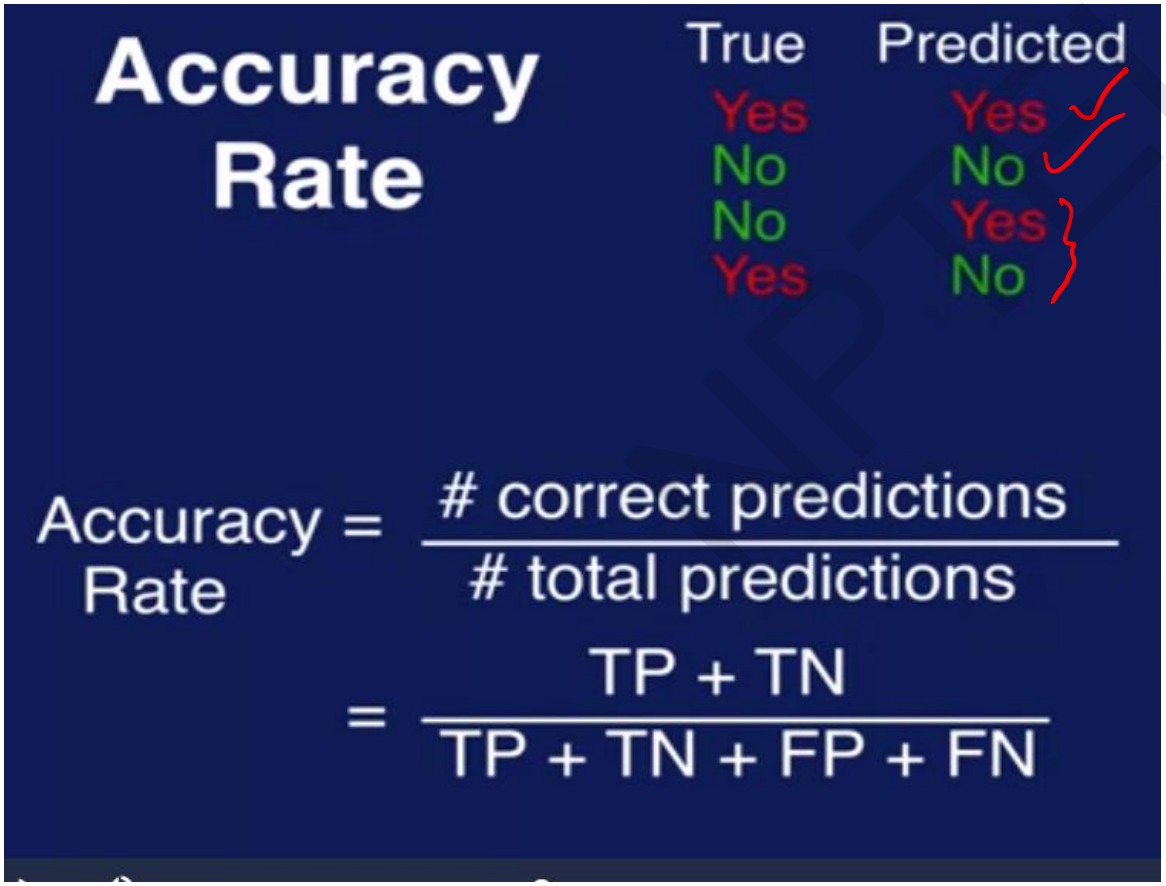
Class Labels



Types of Classification Errors



Accuracy Rate



$\frac{2}{4}$

Error
True Positive (TP)
True Negative (TN)
False Positive (FP)
False Negative (FN)

Error Rate

Error Rate

True	Predicted
Yes	Yes
No	No
No	Yes
Yes	No

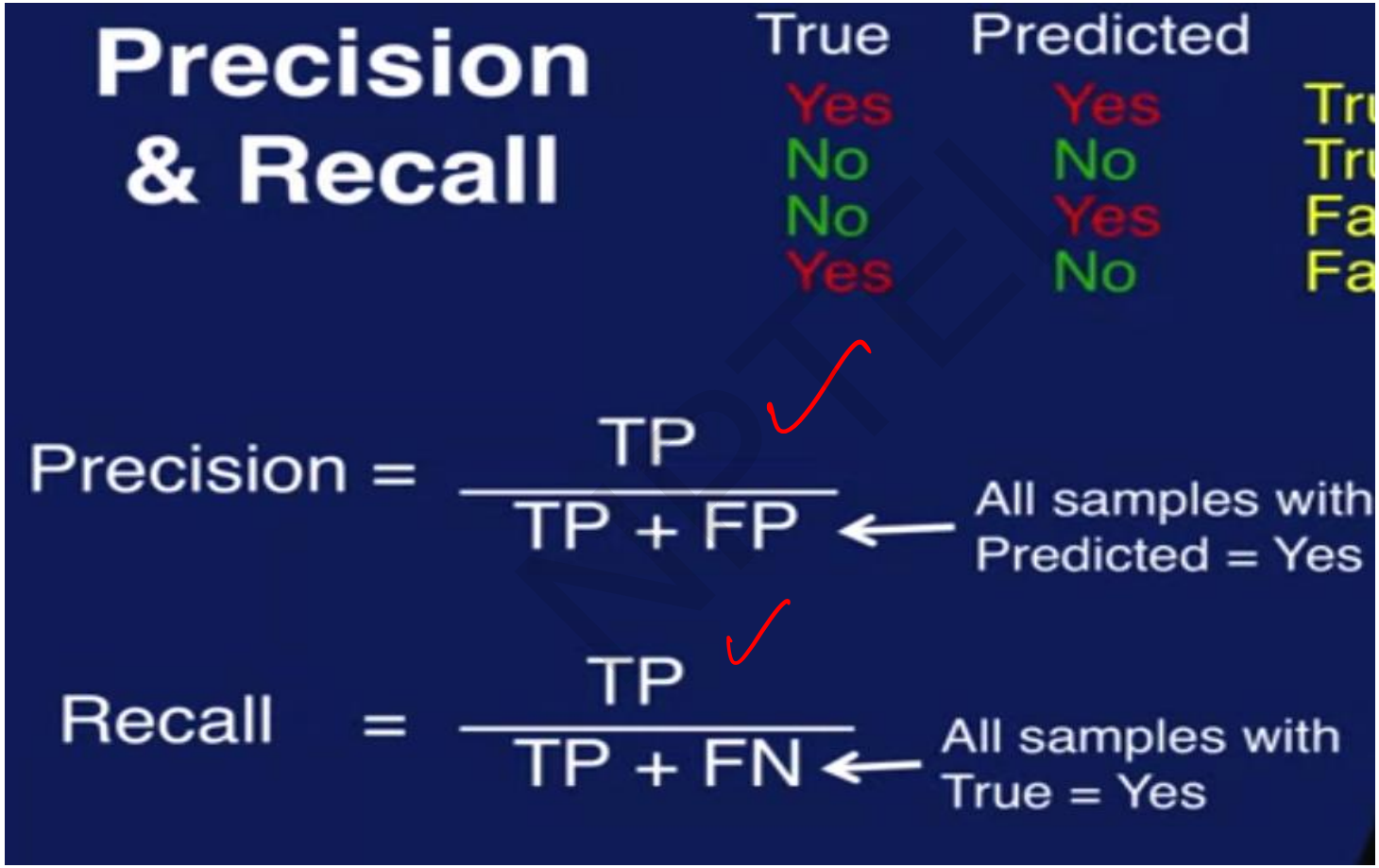
$$\text{Error Rate} = \frac{\#\text{ incorrect predictions}}{\#\text{ total predictions}}$$

$$= 1 - \text{Accuracy Rate}$$

$$= 1 - 0.7 = 0.3$$

Error
True Positive (TP)
True Negative (TN)
False Positive (FP)
False Negative (FN)

Precision and Recall

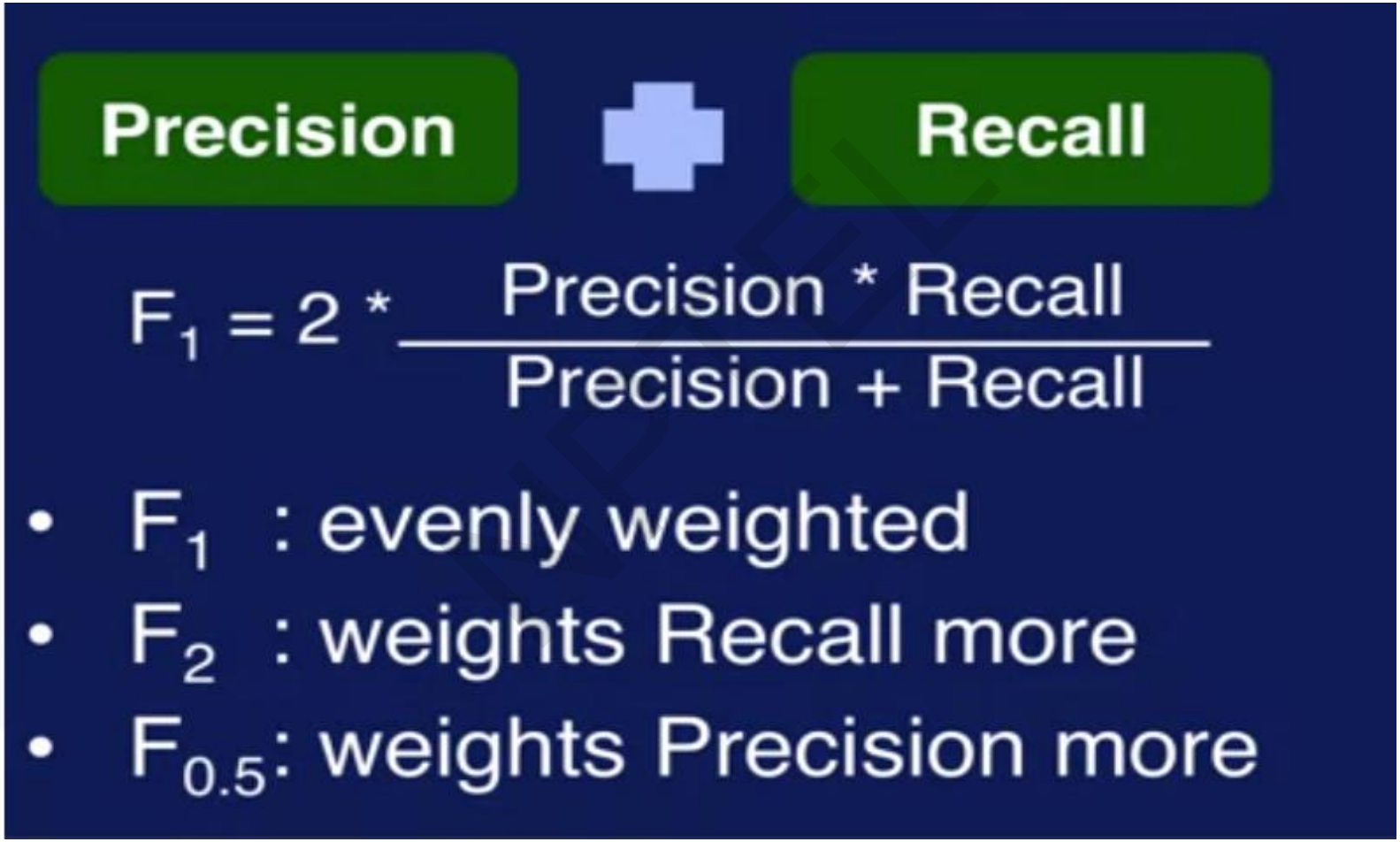


Precision and Recall

- Precision is considered a measure of exactness because it calculates the percentage of samples predicted as positive, which are actually in a positive class.

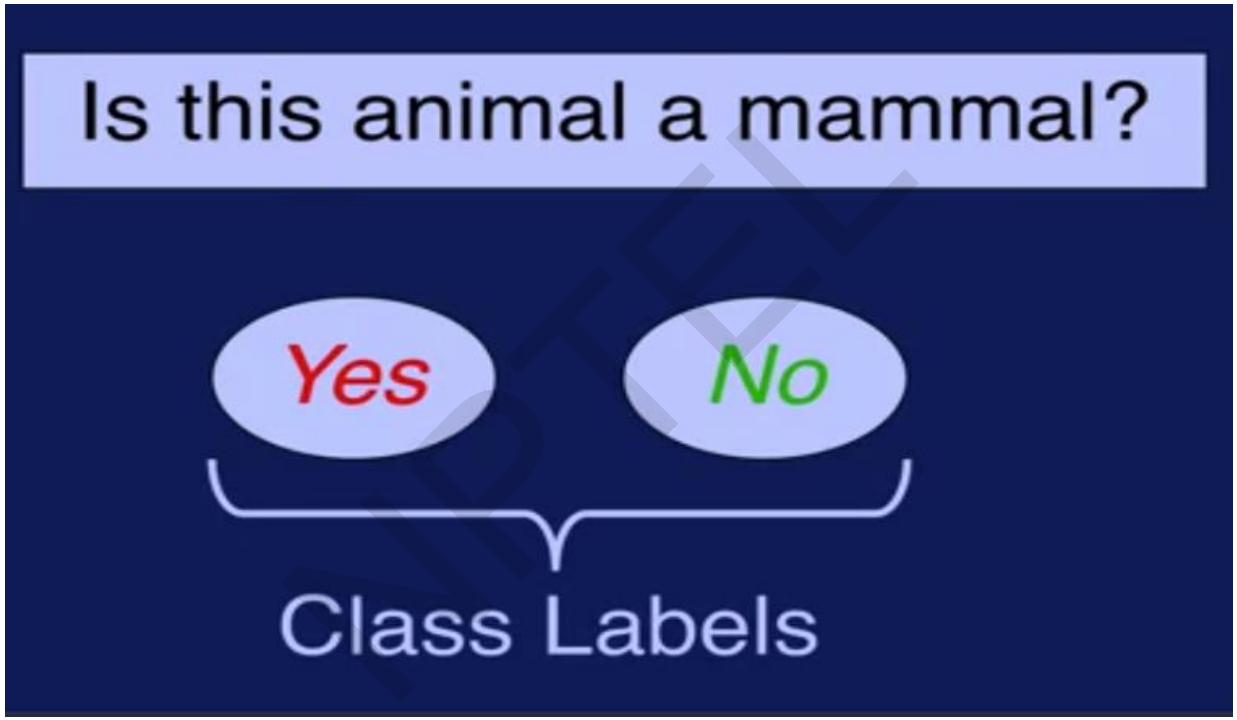
- Recall is considered a measure of completeness, because it calculates the percentage of positive samples that the model correctly identified.

F-Measure

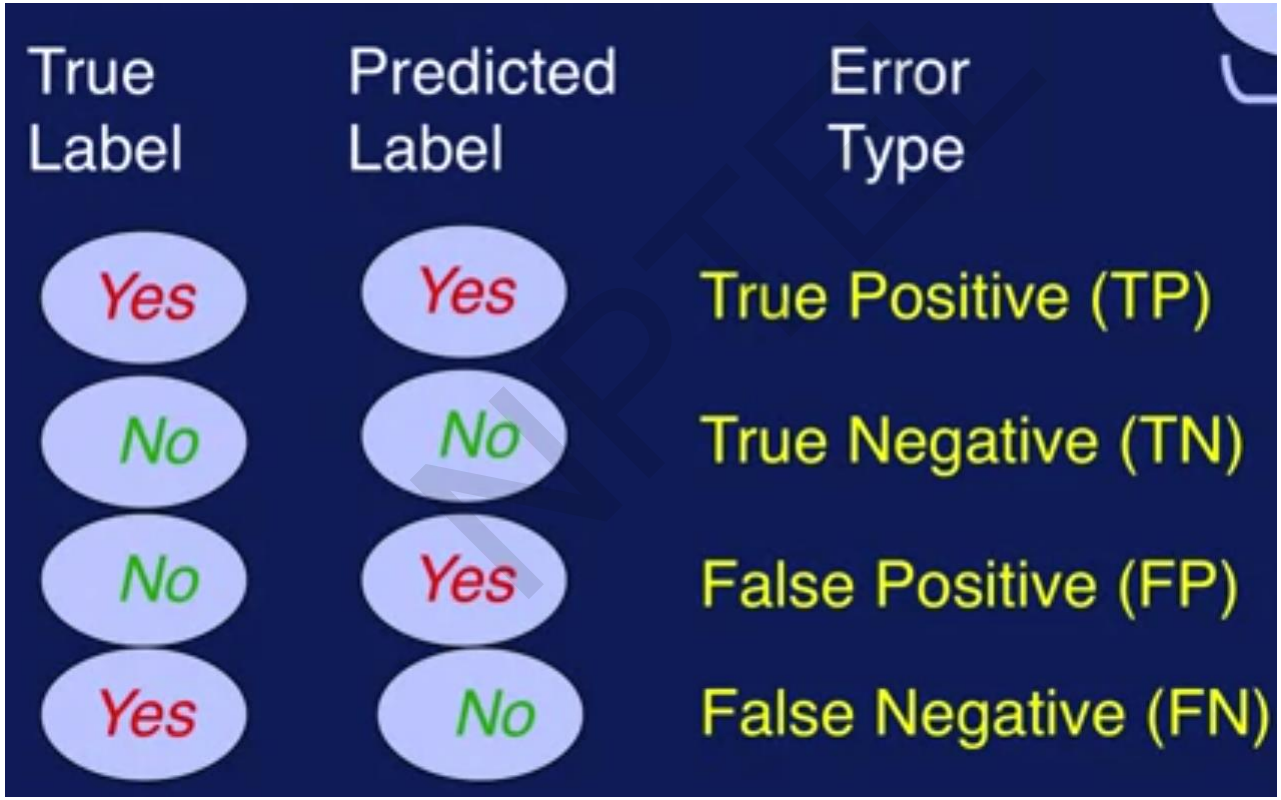

Precision
Recall
$F_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$
• F_1 : evenly weighted
• F_2 : weights Recall more
• $F_{0.5}$: weights Precision more

Confusion Matrix

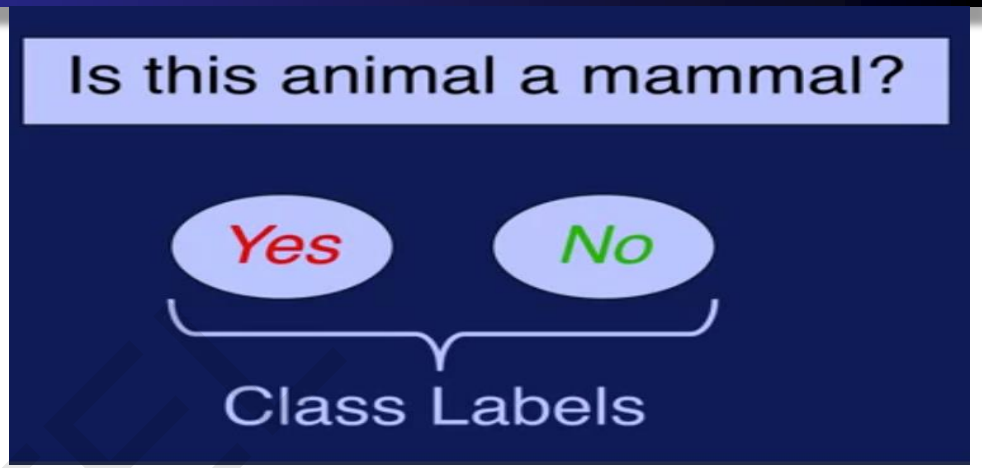
Classification



Types of Classification Errors



Confusion Matrix



		Predicted Class Label	
		Yes	**No**
True Class Label	**Yes**	True Positive (TP)	False Negative (FN)
	No	False Positive (FP)	True Negative (TN)

True Label	Predicted Label
Yes	No
No	No
No	No
Yes	Yes
Yes	Yes
No	No
Yes	No
Yes	Yes
No	No
No	Yes

		Predicted Class Label	
		Yes	No
True Class Label	Yes	True Positive (TP) 3	False Negative (FN) 2
	No	False Positive (FP) 1	True Negative (TN) 4

Confusion matrix

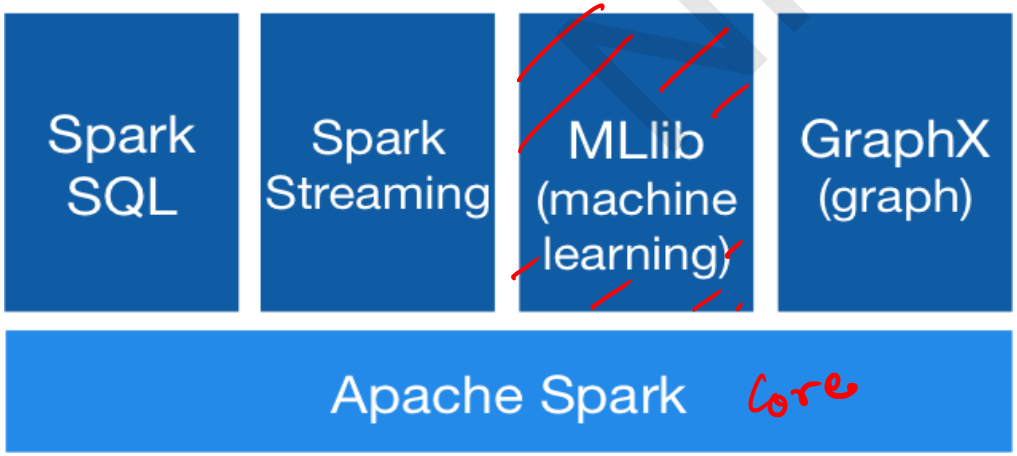
10%

70%

Machine Learning Tool

Spark MLlib

- **Spark MLlib** is a distributed machine-learning framework on top of Spark Core.
- MLlib is Spark's scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction.



Spark MLlib Component

Algorithms
- Classification
- Regression
- Clustering
- Collaborative Filtering

Pipeline
- Constructing
- Evaluating
- Tuning
- Persistence

Featurization
- Extraction
- Transformation

Utilities
- Linear algebra
- Statistics

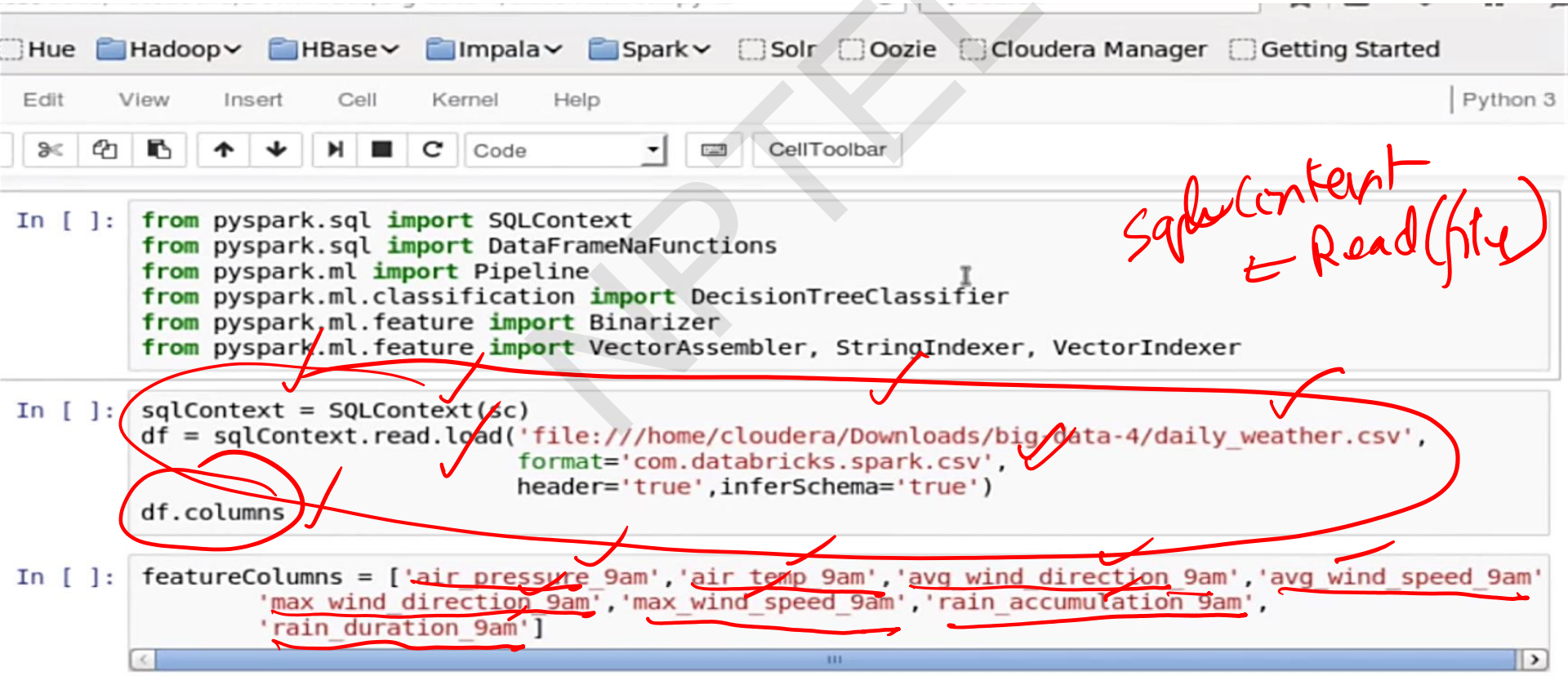
Classification in Spark

Steps for Classification in Spark

1. Load weather data into DataFrame (Read Data)
2. Drop unused and missing data. (Data preparation)
3. Create a categorical variable for low humidity days (Transform)
4. Aggregate features used to make predictions. (Feature Selection)
5. Split the data into training and test sets. (Split Data-sets into Train & Test)
6. Create and train the decision tree.
7. Save the predictions to a CSV file

1. Load weather data into DataFrame

The first cell contains the classes we need to load to run this exercise. Next we create SQL context and load the weather data CSV into a data frame.
The second cell also prints all the columns in this data frame.
The third cell defines the columns in the weather data we will use for the decision tree classifier.



Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started

Edit View Insert Cell Kernel Help Python 3

Code CellToolbar

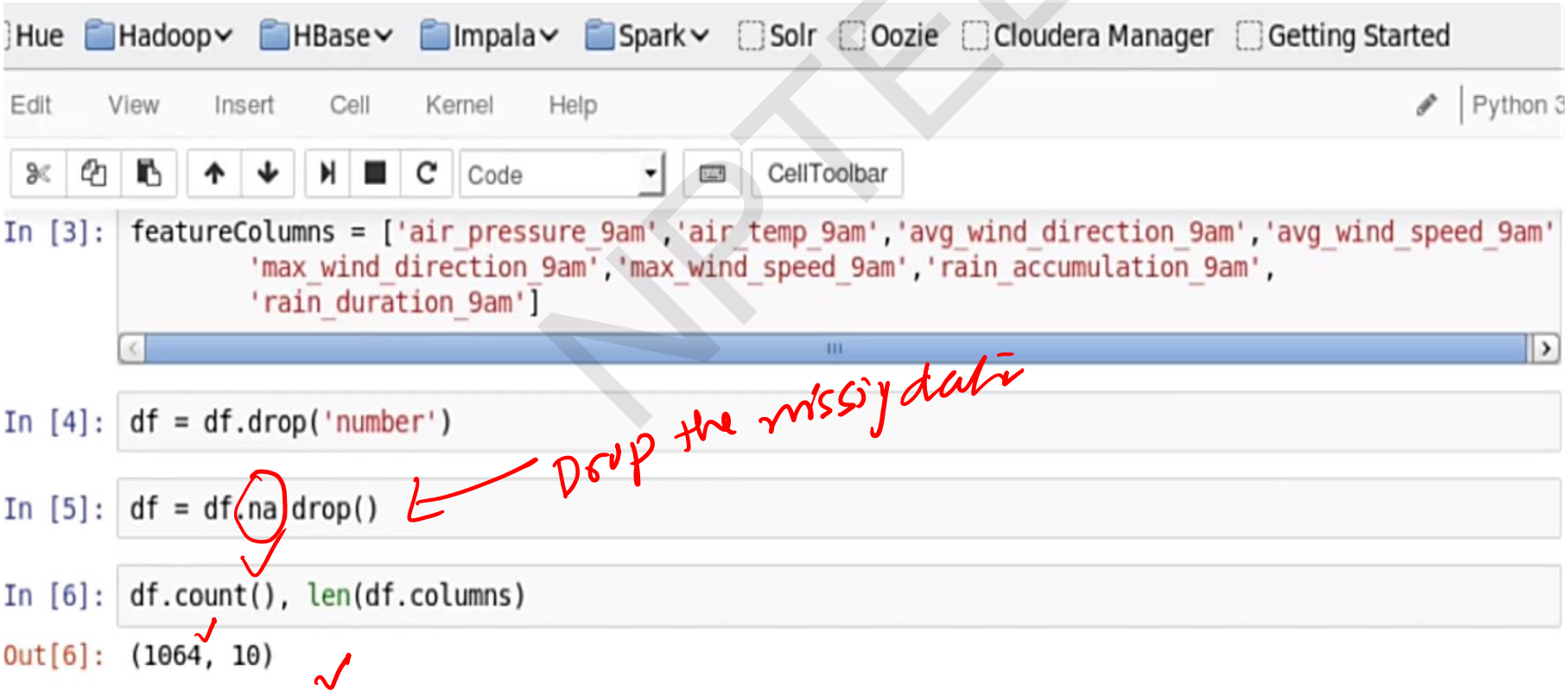
```
                                header='true',inferSchema='true')
df.columns
```

Out[2]:
```
['number',
 'air_pressure_9am',
 'air_temp_9am',
 'avg_wind_direction_9am',
 'avg_wind_speed_9am',
 'max_wind_direction_9am',
 'max_wind_speed_9am',
 'rain_accumulation_9am',
 'rain_duration_9am',
 'relative_humidity_9am',
 'relative_humidity_3pm']
```

In [ ]:
```
featureColumns = ['air_pressure_9am','air_temp_9am','avg_wind_direction_9am','avg_wind_speed_9am'
        'max_wind_direction_9am','max_wind_speed_9am','rain_accumulation_9am',
        'rain_duration_9am']
```

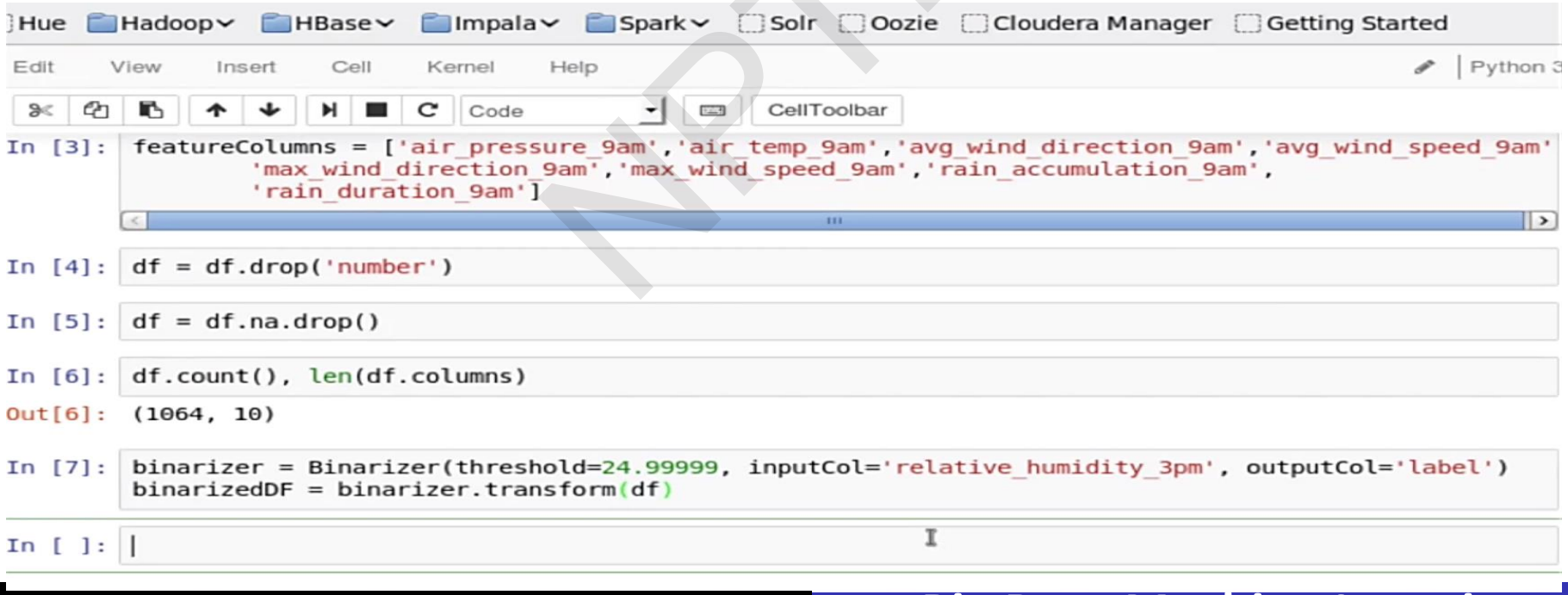
2. Drop unused and missing data

Use the column name number.
Drop that from the data frame, df = df.drop Number.
Rows with missing data, df = df.na.drop.
Print the number of rows and columns in our resulting data frame, df.count(), len(df.columns).



3. Create a categorical variable for low humidity days

We will enter binarizer = Binarizer (). The first argument specifies a threshold value for the variable. We want the categorical variable to be 1, if the humidity is greater than 25%. So we'll enter a threshold=24.9999. The next argument specifies the column to use to create the categorical variable. We'll input, inputCol = relative_humidity_3pm. The final argument specifies the new column name, outputCol = label. A new data frame is created with this categorical variable. binarizeredDF = binarizer.transform df.



4. Aggregate features used to make predictions.

The relative humidity in the first row is greater than 25% and the label is 1. The relative humidity in the second, third, and fourth rows are less than 25%, and the label is 0. For aggregating the features we will use to make predictions into a single col, assembler = VectorAssember( ). The first argument is a list of the columns to be aggregated. inputCols=featureColumns, and the second argument is the name of the new column containing the aggregated features, outputCol=features. We can create the new data frame by running assembled=assembler.transform binarizedDF.

```
Out[6]: (1064, 10)

In [7]: binarizer = Binarizer(threshold=24.99999, inputCol='relative_humidity_3pm',
        binarizedDF = binarizer.transform(df)

In [8]: binarizedDF.select('relative_humidity_3pm', 'label').show(4)

        +----------------------+-----+
        |relative_humidity_3pm|label|
        +----------------------+-----+
        |    36.160000000000494|  1.0|
        |      19.4265967985621|  0.0|
        |    14.460000000000045|  0.0|
        |    12.742547353761848|  0.0|
        +----------------------+-----+
        only showing top 4 rows

In [ ]:
```

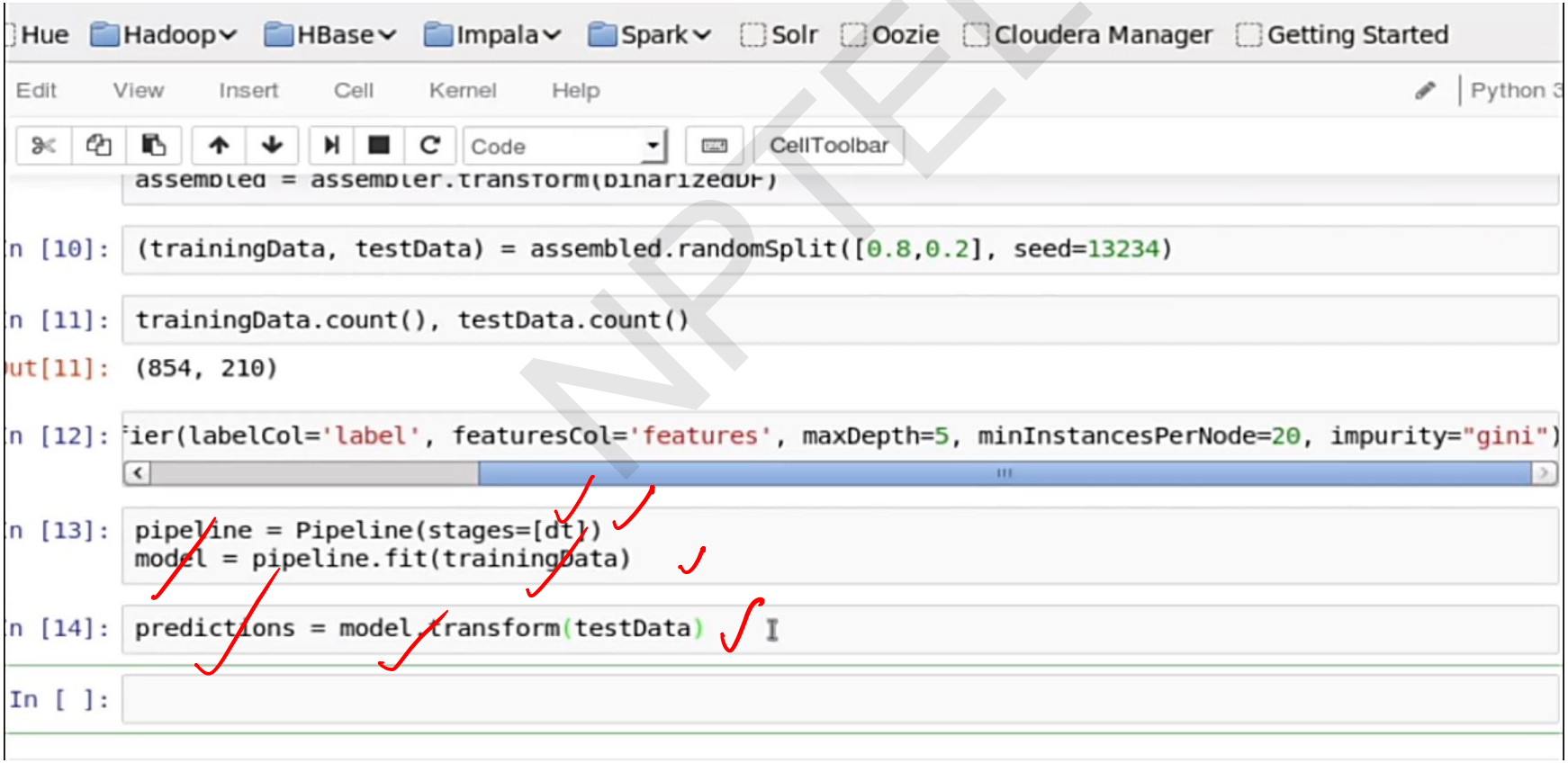
5. Split the data into training and test sets.

Next split the data set into two parts, one for training data and one for test data. We can do this by entering (training Data, testData)=assembled.randomSplit([0.8, 0.2], seed=13234). We can see the size of the two sets by running count, trainingData.count(), testData.count ().

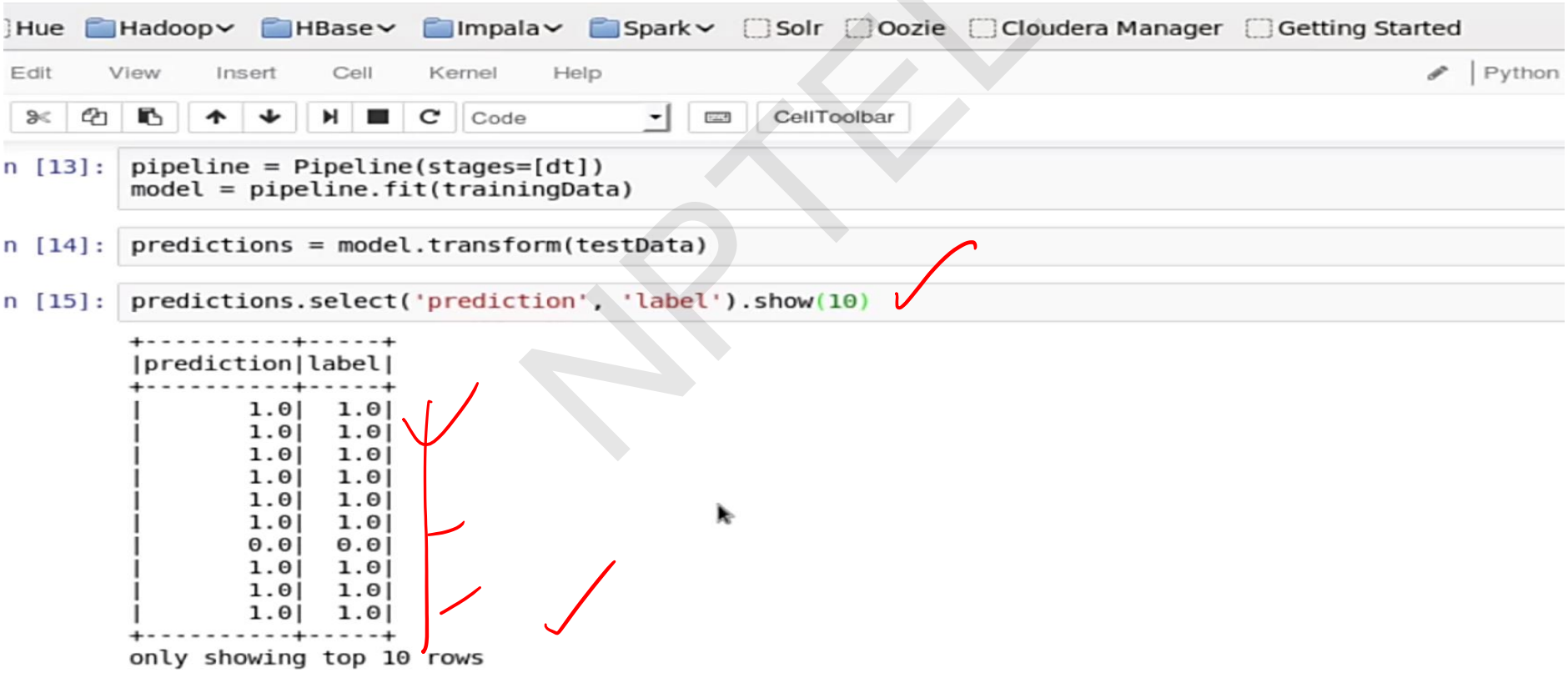
```
In [9]: assembler = VectorAssembler(inputCols=featureColumns, outputCol='features')
        assembled = assembler.transform(binarizedDF)

In [10]: (trainingData, testData) = assembled.randomSplit([0.8,0.2], seed=13234)

In [11]: trainingData.count(), testData.count()

Out[11]: (854, 210)

In [ ]:
```

6. Create and train the decision tree.

Next, we can create a model by training the decision tree. We can do this by executing it in a pipeline. Enter pipeline=Pipeline (stages=[dt]). We will create them all by putting a training data, model = pipeline.fit trainingData. Let's run this Now, we can make predictions using our test data. Enter predictions = model.transform(testData).

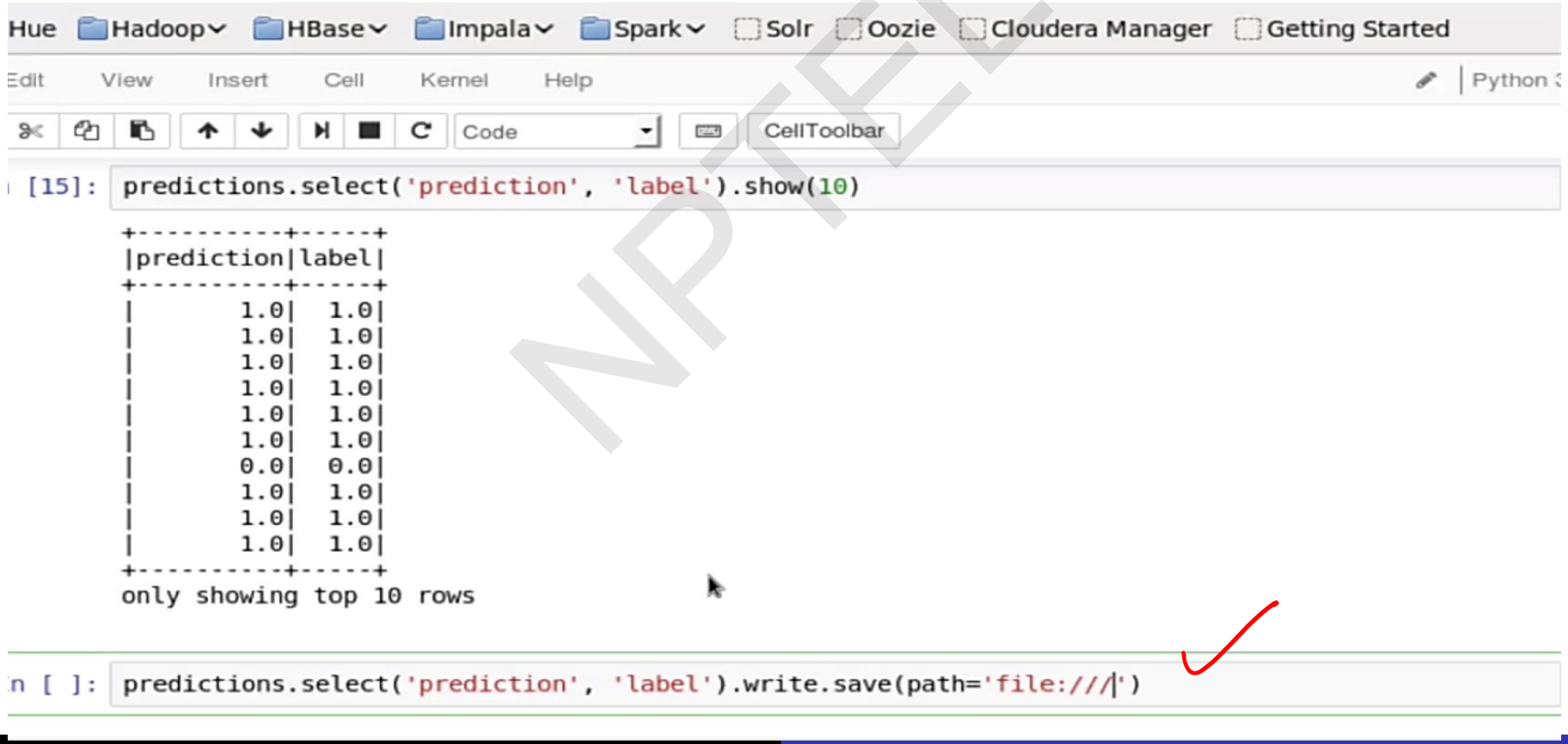


Look at the first 10 rows of the prediction by running, predictions.select('prediction', label') show(10). Here we can see in the first ten rows, the prediction matches the label.



7. Save the predictions to a CSV file

Save the predictions to a CSV file. We can save it by running predictions.select('prediction', 'label').write.save(path=‘). Format='com.databricks.spark.csv.



Conclusion

- In this lecture, we have discussed the machine learning techniques.

- We have also discussed tools and algorithms that you can use to create machine learning models that learn from data, and to scale those models up to big data problems.

Machine Learning Algorithm Parallel K-means using Map Reduce for Big Data Analytics

Dr. Rajiv Misra

Dept. of Computer Science & Engg.

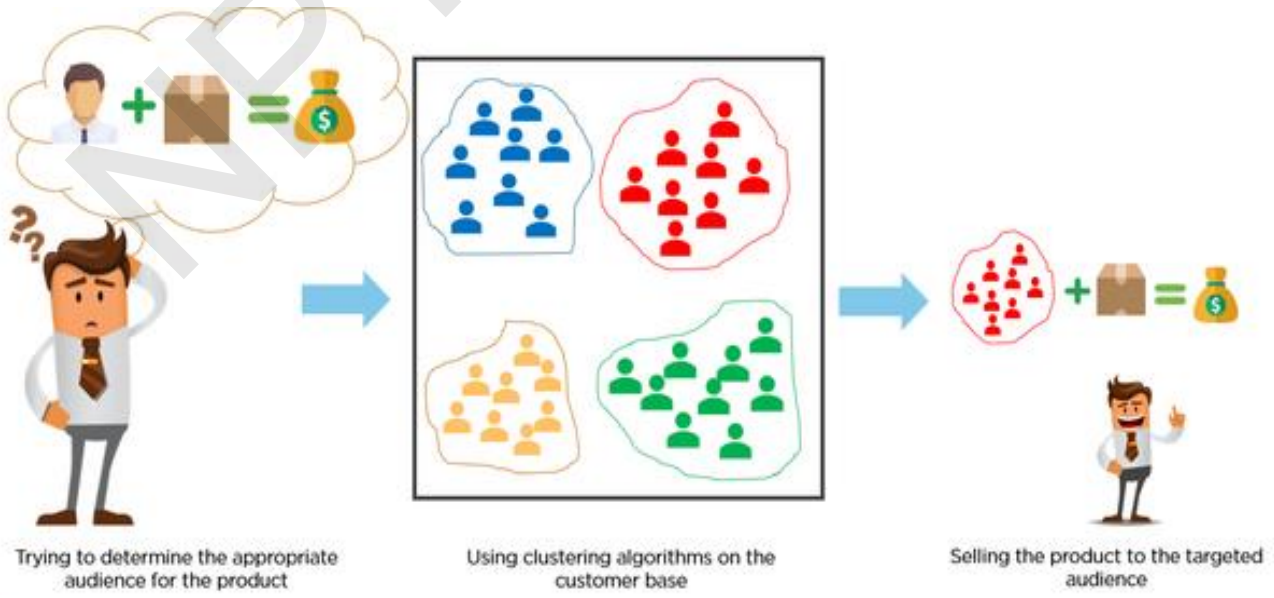
Indian Institute of Technology Patna

rajivm@iitp.ac.in

Preface

Content of this Lecture:

- In this lecture, we will discuss machine learning classification algorithm k-means using mapreduce for big data analytics

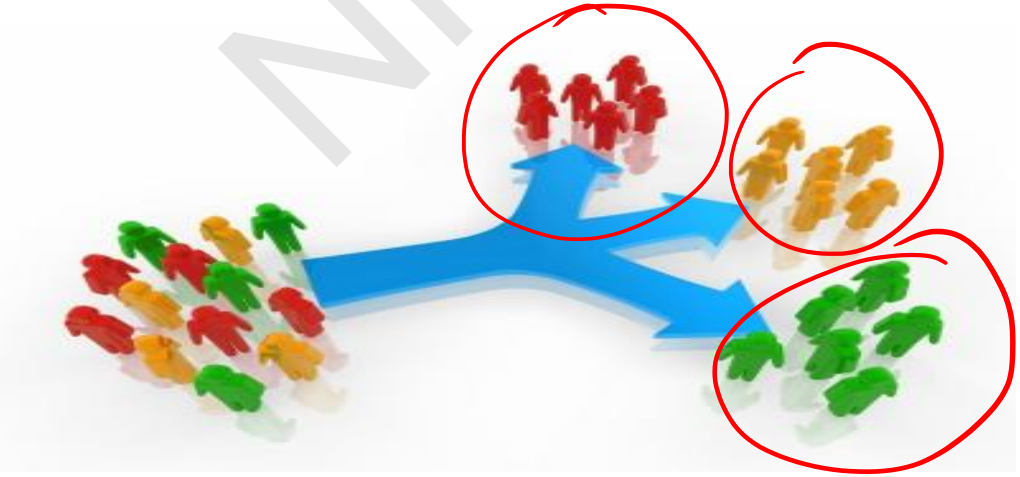


Cluster Analysis Overview

- **Goal:** Organize similar items into groups.

- In cluster analysis, the goal is to organize similar items in given data set into groups or clusters. By segmenting given data into clusters, we can analyze each cluster more carefully.

- Note that cluster analysis is also referred to as clustering.



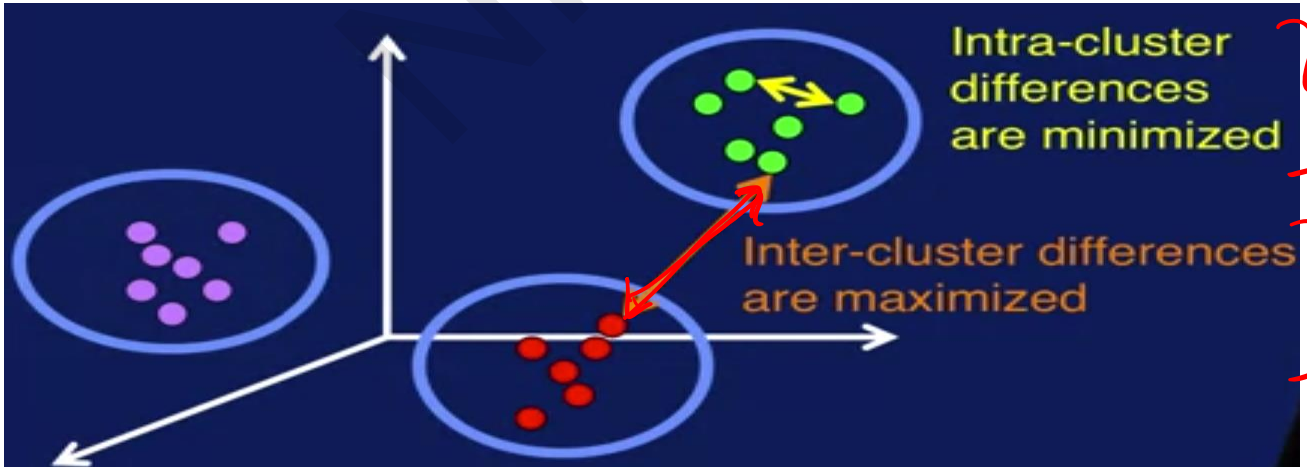
Segmentation of people can be used for targetting the shopping

Applications: Cluster Analysis

- **Segment customer base into groups:** A very common application of cluster analysis is to divide your customer base into segments based on their purchasing histories. For example, you can segment customers into those who have purchased science fiction books and videos, versus those who tend to buy nonfiction books, versus those who have bought many children's books. This way, you can provide more targeted suggestions to each different group.
- **Characterize different weather patterns for a region:** Some other examples of cluster analysis are characterizing different weather patterns for a region.
- **Group news articles into topics:** Grouping the latest news articles into topics to identify the trending topics of the day.
- **Discover crime hot spots:** Discovering hot spots for different types of crime from police reports in order to provide sufficient police presence for problem areas.

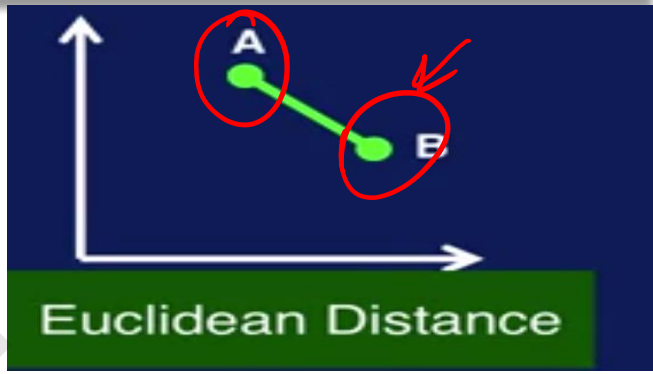
Cluster Analysis

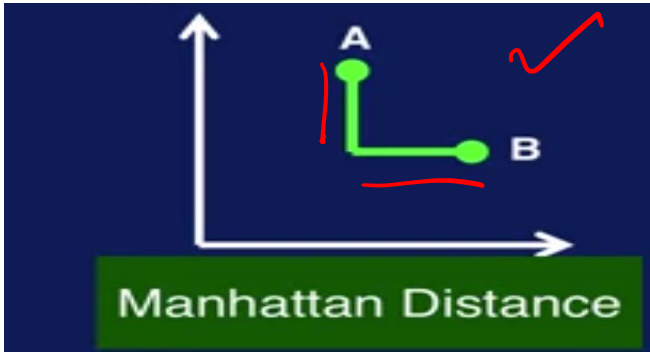
- **Divides data into clusters:** Cluster analysis divides all the samples in a data set into groups. In this diagram, we see that the red, green, and purple data points are clustered together. Which group a sample is placed in is based on some measure of similarity.
- **Similar items are placed in same cluster:** The goal of cluster analysis is to segment data so that differences between samples in the same cluster are minimized, as shown by the yellow arrow, and differences between samples of different clusters are maximized, as shown by the orange arrow. Visually, we can think of this as getting samples in each cluster to be as close together as possible, and the samples from different clusters to be as far apart as possible.

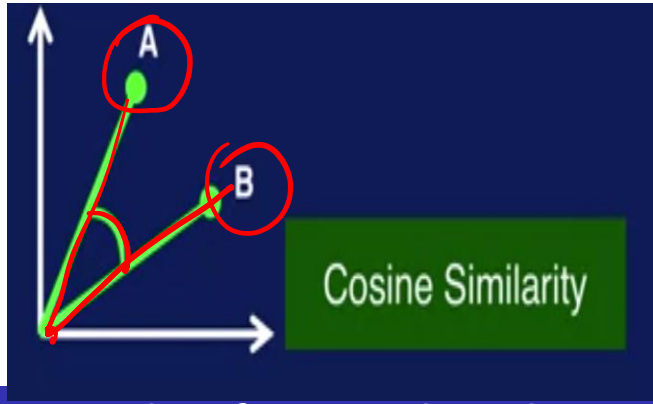


Similarity Measures

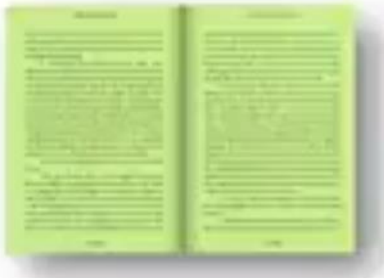
- Cluster analysis requires some sort of metric to measure similarity between two samples. Some common similarity measures are **Euclidean distance**, which is the distance along a straight line between two points, A and B, as shown in this plot.
- **Manhattan distance**, which is calculated on a strictly horizontal and vertical path, as shown in the right plot. To go from point A to point B, you can only step along either the x-axis or the y-axis in a two-dimensional case. So the path to calculate the Manhattan distance consists of segments along the axes instead of along a diagonal path, as with Euclidean distance.
- **Cosine similarity** measures the cosine of the angle between points A and B, as shown in the bottom plot. Since distance measures such as Euclidean distance are often used to measure similarity between samples in clustering algorithms, note that it may be necessary to normalize the input variables so that no one value dominates the similarity calculation.







Another natural inner product measure



x_q

1	0	0	0	5	3	0	0	1	0	0	0	0
3	0	0	0	2	0	0	1	0	1	0	0	0

Similarity

$$= \mathbf{x}_i^T \mathbf{x}_q$$

$$= \sum_{j=1}^{d} \mathbf{x}_i[j]\, \mathbf{x}_q[j]$$

x_i

$= 13$

Another natural inner product measure

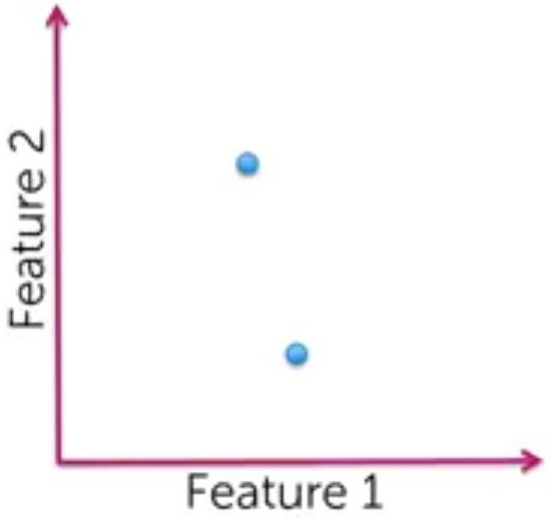
1	0	0	0	5	3	0	0	1	0	0	0	0
0	0	1	0	0	0	9	0	0	6	0	4	0

Similarity = 0

Cosine similarity- normalize

$$\text{Similarity} = \frac{\sum_{j=1}^{d} \mathbf{x}_i[j]\,\mathbf{x}_q[j]}{\sqrt{\sum_{j=1}^{d} (\mathbf{x}_i[j])^2}\sqrt{\sum_{j=1}^{d} (\mathbf{x}_q[j])^2}}$$

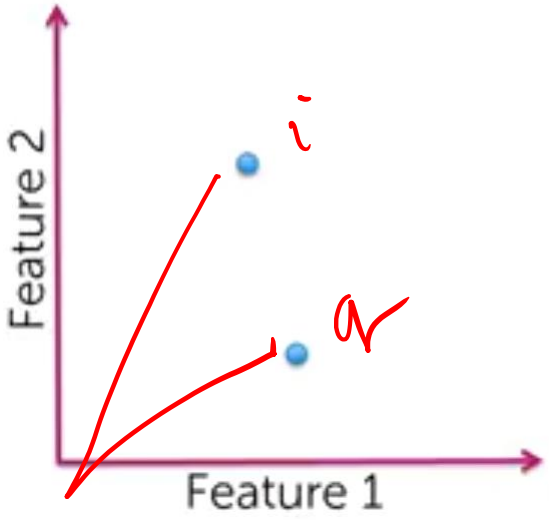
$$= \frac{\mathbf{x}_i^T \mathbf{x}_q}{||\mathbf{x}_i||\,||\mathbf{x}_q||} = \cos(\theta)$$



Cosine similarity- normalize

$$\text{Similarity} = \frac{\sum_{j=1}^{d} \mathbf{x}_i[j]\,\mathbf{x}_q[j]}{\sqrt{\sum_{j=1}^{d} (\mathbf{x}_i[j])^2}\sqrt{\sum_{j=1}^{d} (\mathbf{x}_q[j])^2}}$$

$$= \frac{\mathbf{x}_i^T \mathbf{x}_q}{||\mathbf{x}_i||\,||\mathbf{x}_q||} = \cos(\theta)$$



- Not a proper distance metric
- Efficient to compute for sparse vecs

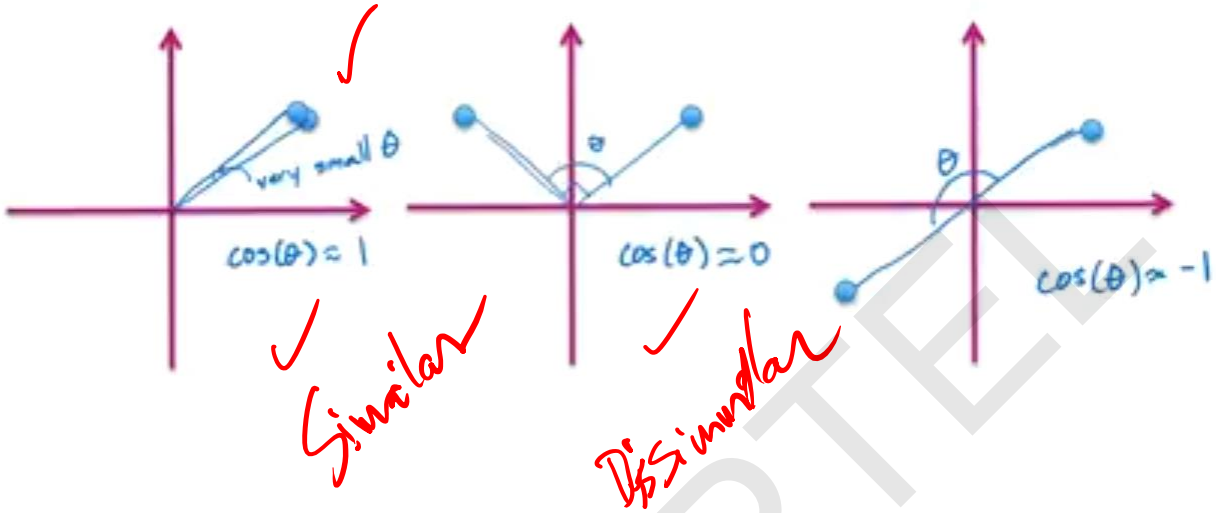
Normalize

1	0	0	0	5	3	0	0	1	0	0	0	0

$$\sqrt{(1^2 + 5^2 + 3^2 + 1^2)}$$

1/6	0	0	0	5/6	3/6	0	0	1/6	0	0	0	0

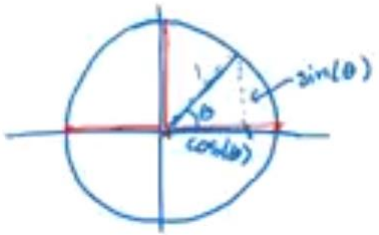
Cosine Normalize



In general, -1 < similarity < 1

For positive features (like tf-idf)

0 < similarity < 1



Define **distance = 1-similarity**

Cluster Analysis Key Points

Unsupervised

Cluster analysis is an unsupervised task. This means that there is no target label for any sample in the data set.

There is no 'correct' clustering

In general, there is no correct clustering results. The best set of clusters is highly dependent on how the resulting clusters will be used.

Clusters don't come with labels

Clusters don't come with labels. You may end up with five different clusters at the end of a cluster analysis process, but you don't know what each cluster represents. Only by analyzing the samples in each cluster can you come out with reasonable labels for your clusters. Given all this, it is important to keep in mind that interpretation and analysis of the clusters are required to make sense of and make use of the results of cluster analysis.

Interpretation and analysis required to make sense of clustering results!

Uses of Cluster Results

- **Data segmentation**
- Analysis of each segment can provide insights: There are several ways that the results of cluster analysis can be used. The most obvious is data segmentation and the benefits that come from that. If we segment your customer base into different types of readers, the resulting insights can be used to provide more effective marketing to the different customer groups based on their preferences. For example, analyzing each segment separately can provide valuable insights into each group's likes, dislikes and purchasing behavior, just like we see science fiction, non-fiction and children's books preferences here.

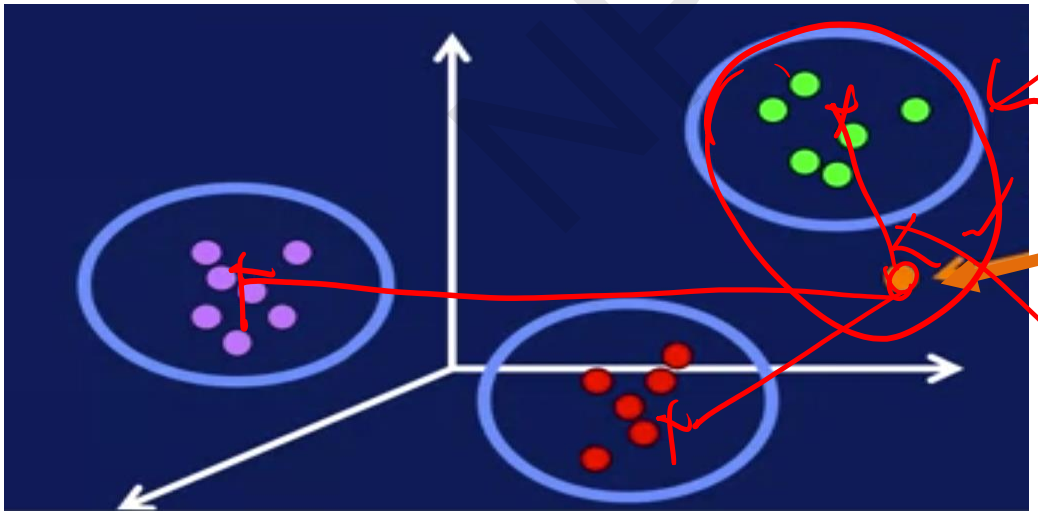
Science fiction

non-fiction

children's

Uses of Cluster Results

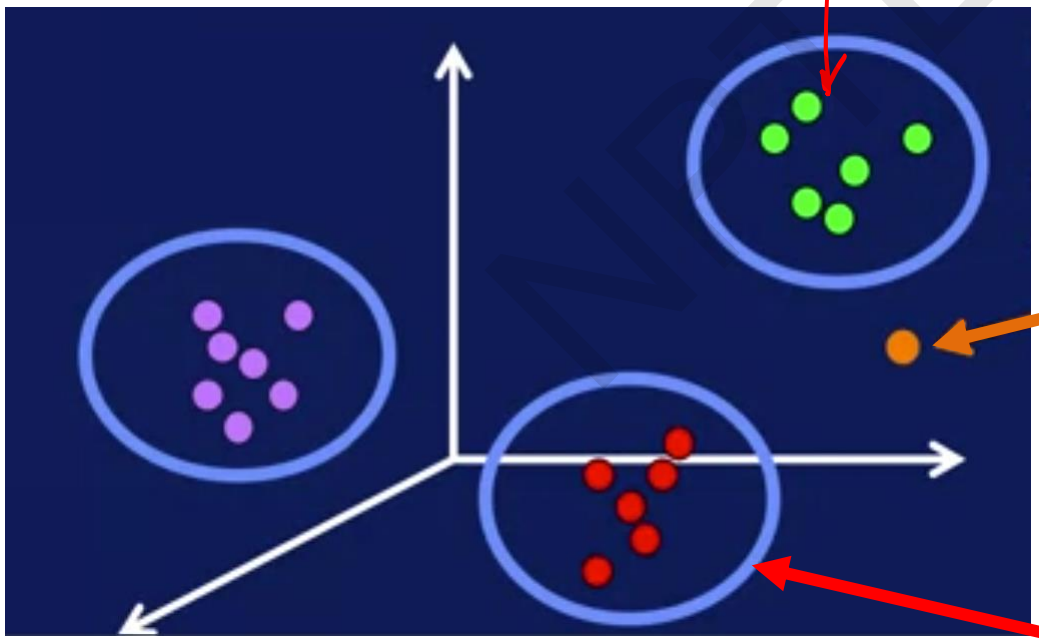
- **Categories for classifying new data**
- New sample assigned to closest cluster: Clusters can also be used to classify new data samples. When a new sample is received, like the orange sample here, compute the similarity measure between it and the centers of all clusters, and assign a new sample to the closest cluster. The label of that cluster, manually determined through analysis, is then used to classify the new sample. In our book buyers' preferences example, a new customer can be classified as being either a science fiction, non-fiction or children's books customer depending on which cluster the new customer is most similar to.



Label of closest cluster used to classify new sample

Uses of Cluster Results

- **Labeled data for classification**
- Cluster samples used as labeled data: Once cluster labels have been determined, samples in each cluster can be used as labeled data for a classification task. The samples would be the input to the classification model. And the cluster label would be the target class for each sample. This process can be used to provide much needed labeled data for classification.

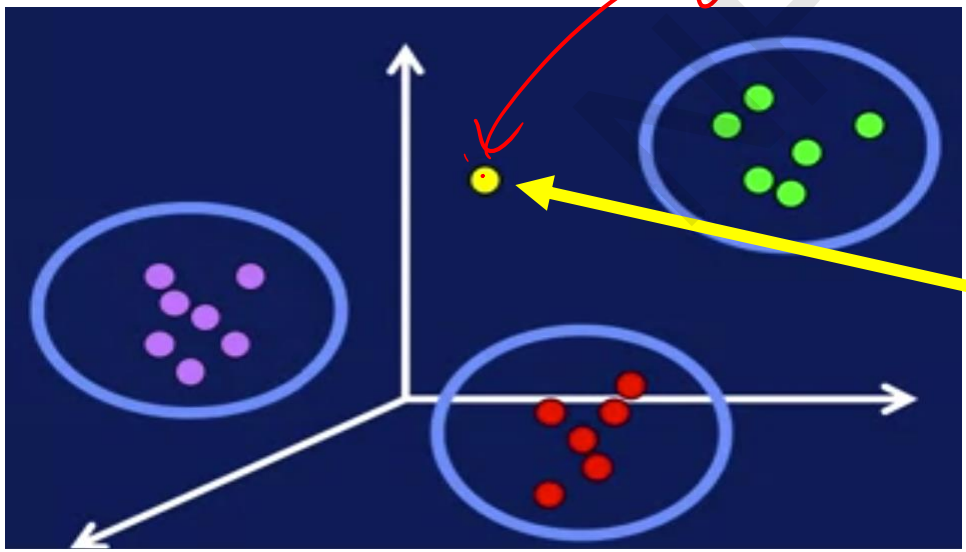


Label of closest cluster used to classify new sample

Labeled samples for science fiction customers

Uses of Cluster Results

- **Basis for anomaly detection**
- Cluster outliers are anomalies: Yet another use of cluster results is as a basis for anomaly detection. If a sample is very far away, or very different from any of the cluster centers, like the yellow sample here, then that sample is a cluster outlier and can be flagged as an anomaly. However, these anomalies require further analysis. Depending on the application, these anomalies can be considered noise, and should be removed from the data set. An example of this would be a sample with a value of 150 for age.

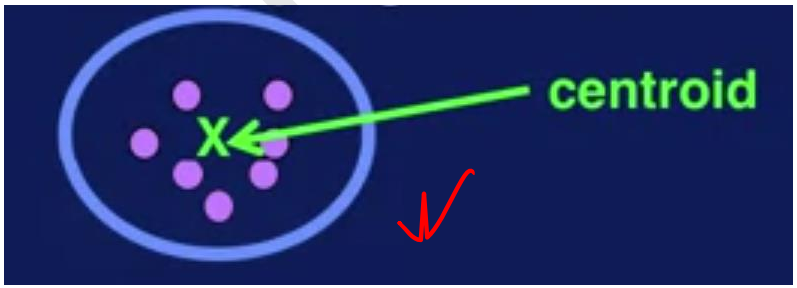


Cluster Analysis Summary

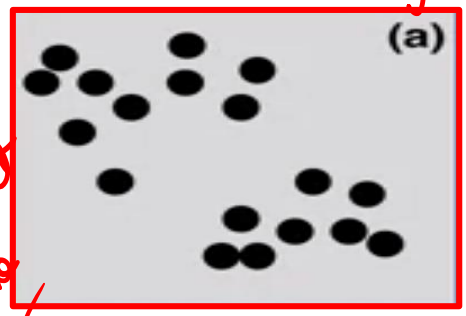
- Organize similar items into groups
- Analyzing clusters often leads to useful insights about data
- Clusters require analysis and interpretation

k-Means Algorithm

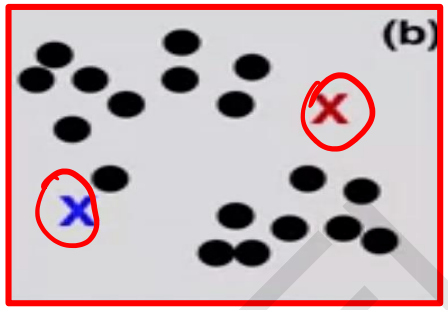
- Select k initial centroids (cluster centers)
- Repeat
  - Assign each sample to closest centroid
  - Calculate mean of cluster to determine new centroid
- Until some stopping criterion is reached



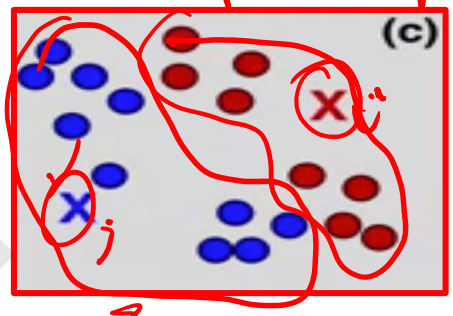
k-Means Algorithm: Example



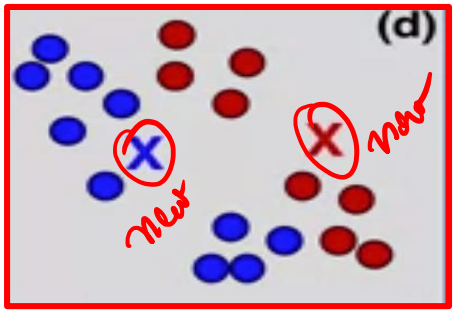
Original samples



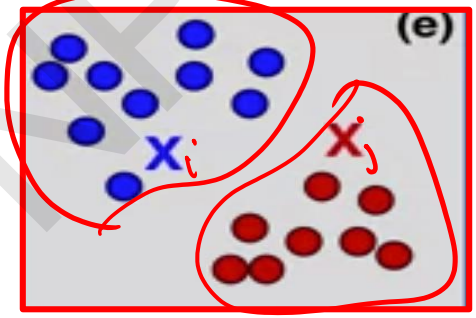
Initial centroids



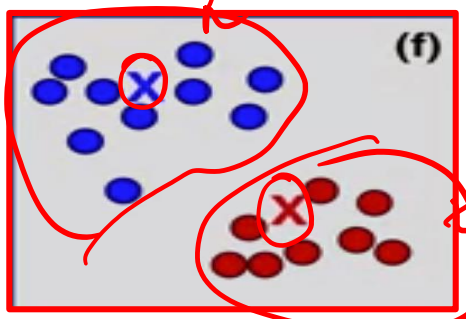
Assign samples



Re-calculate centroids



Assign samples



Re-calculate centroids

Choosing Initial Centroids

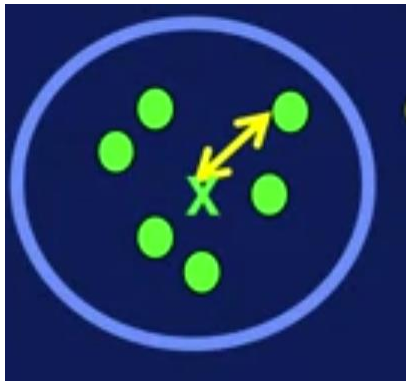
- **Issue:**

  Final clusters are sensitive to initial centroids

- **Solution:**

  Run k-means multiple times with different random initial centroids, and choose best results

Evaluating Cluster Results



error=distance between sample & centroid

Squared error = $error^2$

Sum of squared errors
between all samples & centroid

Sum over all clusters → WSSE
(Within-Cluster Sum
of Squared Error)

Using WSSE

- **WSSE1 < WSSE2** → WSSE1 is better numerically

- **Caveats:**

  Does not mean that cluster set 1 is more 'correct' than cluster set 2

  Larger values for k will always reduce WSSE

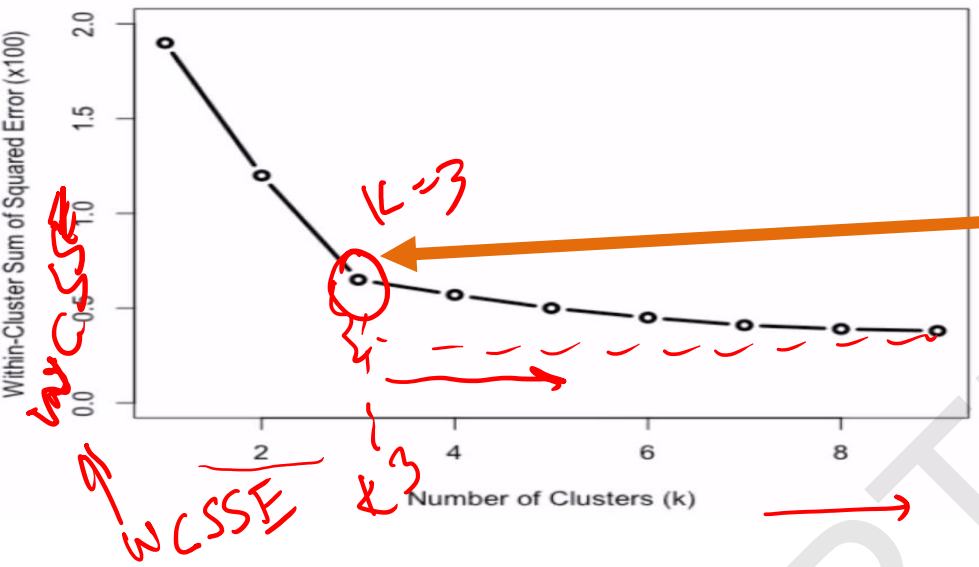
Choosing Value for k

k = ?

Approaches: Choosing the optimal value for k is always a big question in using k-means. There are several methods to determine the value for k.

- Visualization: Visualization techniques can be used to determine the dataset to see if there are natural groupings of the samples. Scatter plots and the use of dimensionality reduction are useful here, to visualize the data.
- Application-Dependent: A good value for k is application-dependent. So domain knowledge of the application can drive the selection for the value of k. For example, if you want to cluster the types of products customers are purchasing, a natural choice for k might be the number of product categories that you offer. Or k might be selected to represent the geographical locations of respondents to a survey. In which case, a good value for k would be the number of regions your interested in analyzing.
- Data-Driven: There are also data-driven method for determining the value of k. These methods calculate symmetric for different values of k to determine the best selections of k. One such method is the elbow method.

Elbow Method for Choosing k



"Elbow" suggests value for k should be 3

The elbow method for determining the value of k is shown on this plot. As we saw in the previous slide, WSSE, or within-cluster sum of squared error, measures how much data samples deviate from their respective centroids in a set of clustering results. If we plot WSSE for different values for k, we can see how this error measure changes as a value of k changes as seen in the plot. The bend in this error curve indicates a drop in gain by adding more clusters. So this elbow in the curve provides a suggestion for a good value of k.

Note that the elbow can not always be unambiguously determined, especially for complex data. And in many cases, the error curve will not have a clear suggestion for one value, but for multiple values. This can be used as a guideline for the range of values to try for k.

Stopping Criteria

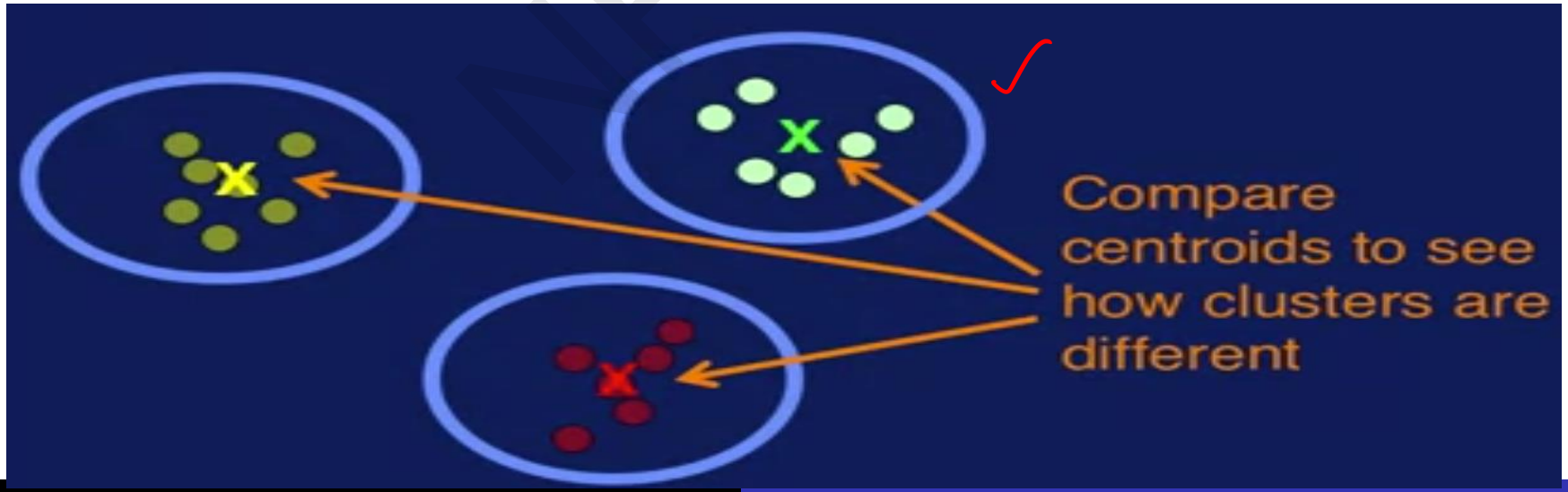
When to stop iterating ?

- No changes to centroids: How do you know when to stop iterating when using k-means? One obviously stopping criterion is when there are no changes to the centroids. This means that no samples would change cluster assignments. And recalculating the centroids will not result in any changes. So additional iterations will not bring about any more changes to the cluster results.
- Number of samples changing clusters is below threshold: The stopping criterion can be relaxed to the second stopping criterion: which is when the number of sample changing clusters is below a certain threshold, say 1% for example. At this point, the clusters are changing by only a few samples, resulting in only minimal changes to the final cluster results. So the algorithm can be stopped here.

Interpreting Results

Examine cluster centroids

- How are clusters different ? At the end of k-means we have a set of clusters, each with a centroid. Each centroid is the mean of the samples assigned to that cluster. You can think of the centroid as a representative sample for that cluster. So to interpret the cluster analysis results, we can examine the cluster centroids. Comparing the values of the variables between the centroids will reveal how different or alike clusters are and provide insights into what each cluster represents. For example, if the value for age is different for different customer clusters, this indicates that the clusters are encoding different customer segments by age, among other variables.



K-Means Summary

- Classic algorithm for cluster analysis
- Simple to understand and implement and is efficient
- Value of k must be specified
- Final clusters are sensitive to initial centroids

Parallel K-means using Map Reduce on Big Data Cluster Analysis

Apply MapReduce on K-means Algorithm
- Data Parallel K-means Algorithm using MapReduce
- Bigdata
- Extract the insights of Big data

MapReducing 1 iteration of k-means

- **Classify:** Assign observations to closest cluster center

$$z_i \leftarrow \arg\min_j ||\mu_j - \mathbf{x}_i||_2^2$$

Map: For each data point, given ($\{\mu_j\}$,x_i), emit(z_i,x_i)

- **Recenter:** Revise cluster centers as mean of assigned observations

$$\mu_j = \frac{1}{n_j} \sum_{i:z_i=k} \mathbf{x}_i$$

Reduce: Average over all points in cluster j (z_i=k)

Classification step as Map

- **Classify:** Assign observations to closest cluster center

$$z_i \leftarrow \arg\min_j ||\mu_j - \mathbf{x}_i||_2^2$$

```
map([μ1, μ2,..., μk], xi)
    zi ← arg min_j ||μj − xi||_2^2
    emit(zi, xi)
```

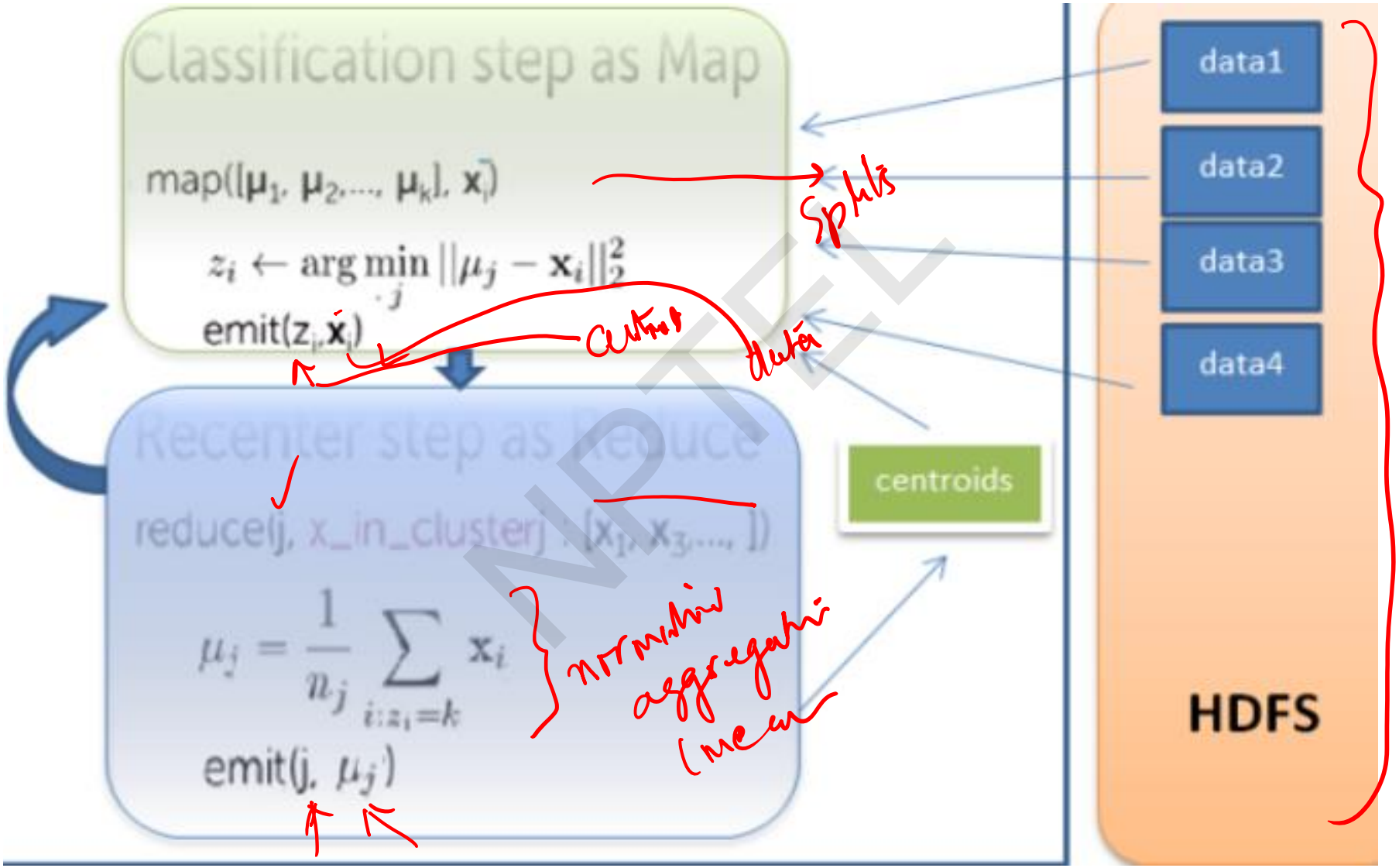
Recenter step as Reduce

- **Recenter:** Revise cluster centers as mean of assigned observations

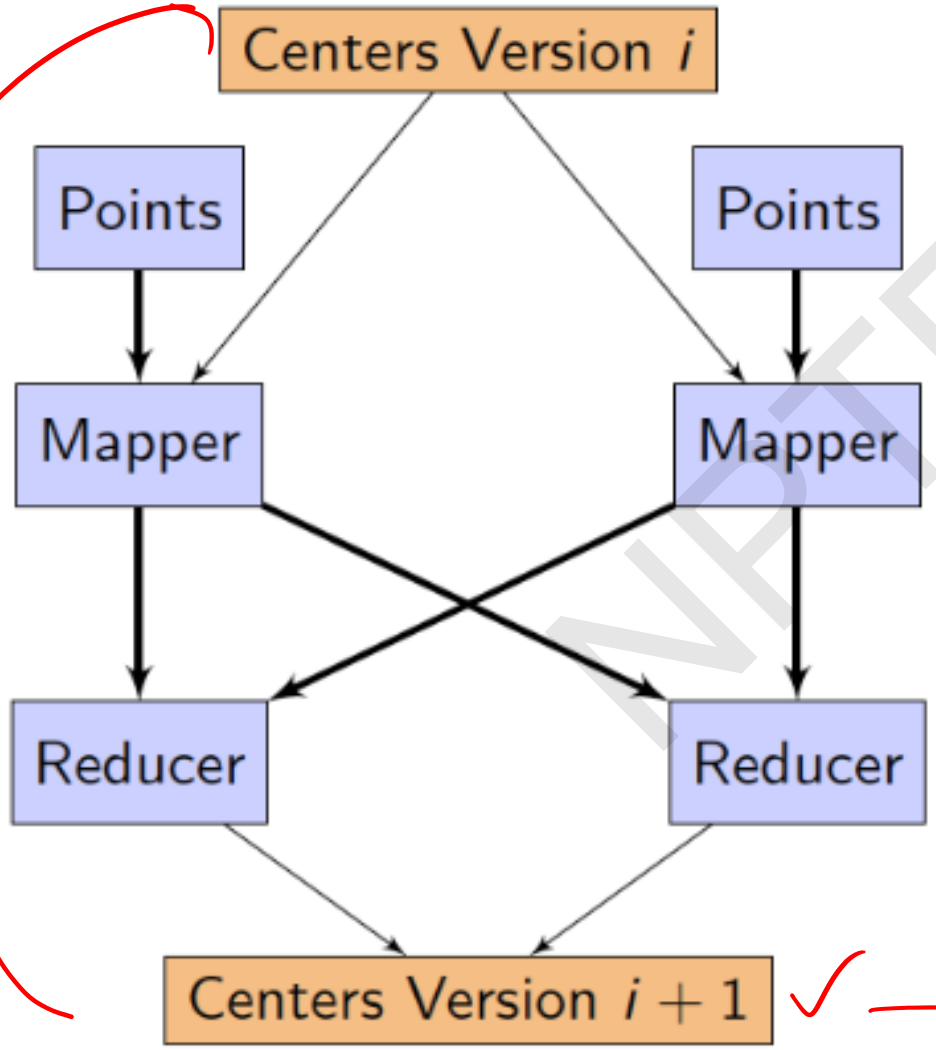
$$\mu_j = \frac{1}{n_j} \sum_{i:z_i=k} \mathbf{x}_i$$

```
reduce(j, x_in_cluster j : [x1, x3,..., ])
    sum = 0
    count = 0
    for x in x_in_cluster j
    sum += x
    count += 1
    emit(j, sum/count)
```

Distributed KMeans Iterative Clustering



Distributed KMeans Iterative Clustering



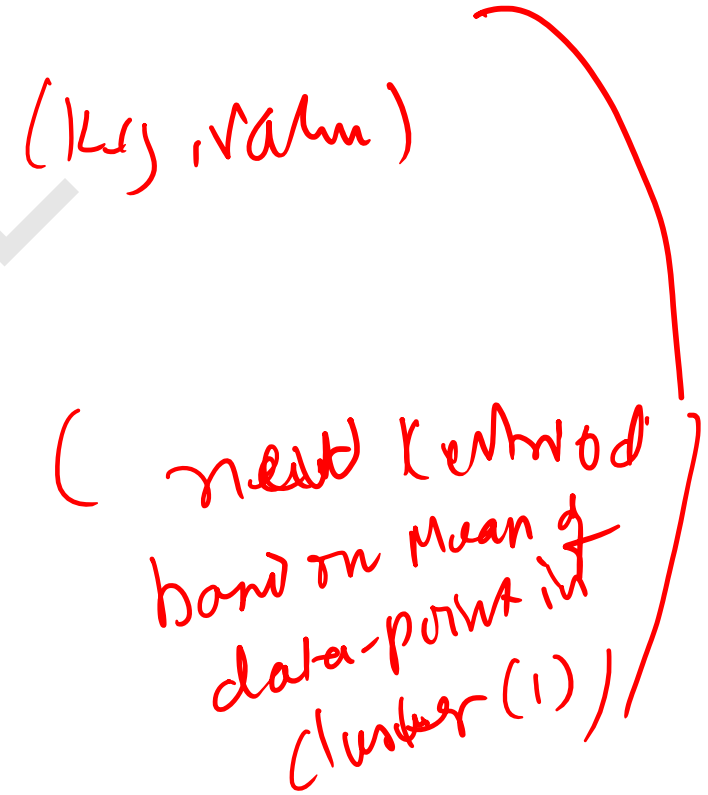
Find Nearest Center

Key is Center, Value is Movie

Average Ratings

Summary of Parallel k-means using MapReduce

- **Map: classification step;**
  - **data parallel over data points**
- **Reduce: recompute means;**
  - **data parallel over centers**



Some practical considerations

- k-means needs an iterative version of MapReduce

  Not standard formulation

Spark.

- Mapper needs to get data point and all centers

  A lot of data!

  Better implementation:

  mapper gets many data points

Spark

Conclusion

- In this lecture, we have given an overview of cluster analysis and also discussed machine learning classification algorithm k-means using Mapreduce for big data analytics