# CSC461
# INTRODUCTION TO DATA SCIENCE

**Dr. Muhammad Sharjeel**

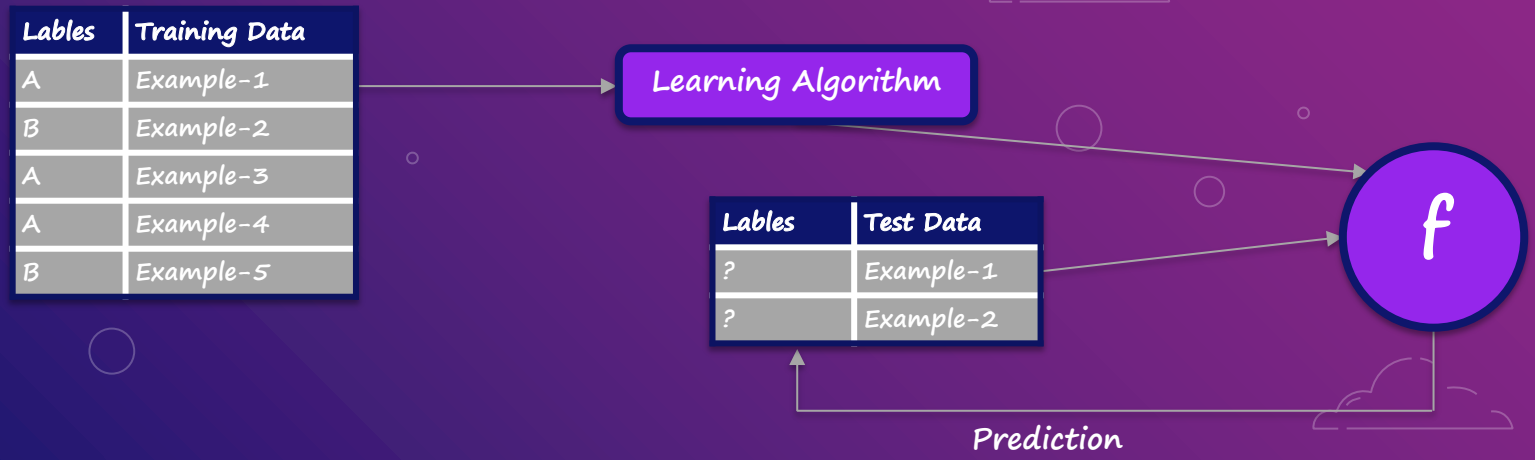
https://muhmmadsharjeel.github.io/

08

# INTRODUCTION TO MACHINE LEARNING

# MACHINE LEARNING

- Machine Learning is about predicting the future based on the past
- More formally, “Machine Learning (ML) is a field of study that gives computers the ability to learn without being explicitly programmed.”
- It is a branch of Artificial Intelligence (AI) in which computers are trained to learn from data, identify patterns, and make decisions with minimal human intervention



# MACHINE LEARNING

- ML is the study of how to design computer programs whose performance at some task improves through experience
  - Goal is to develop a machine that behaves like a human
- In ML, a computer program is said to learn
  - from experience E
  - with respect to some class of tasks T, and
  - performance measure P
- ML program performance is evaluated at tasks in T as measured by P improves with experience E

# MACHINE LEARNING

- Example:
- Learn to drive a car
  - Class of tasks (T)
    - Starting a car, changing gear, control on accelerator and breaks, parking a car etc.
  - Performance Measure (P)
    - Efficiency
  - Experience (E)
    - Number of hours spent in driving the car

- A person (program) is said to have learned to drive a car:
  - If his/her performance P on task(s) T is improving with experience E

# LEARNING ANALOGY

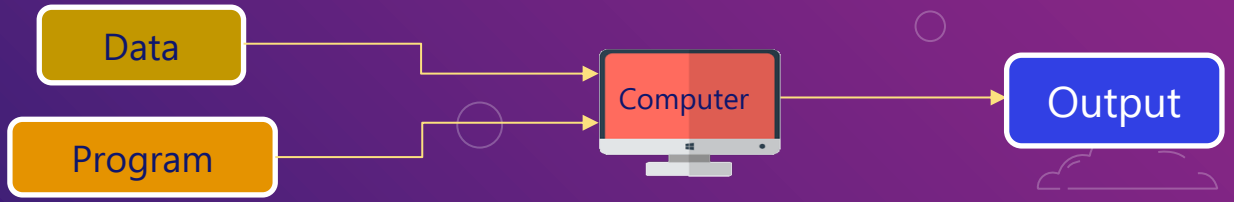
- To learn something is the one's ability to use previous knowledge to perform future actions
- Suppose you took a new course this semester (e.g., Mathematics)
- You expect to “learn” something from that course
- What is a common way to judge how well you do?
  - You did well at learning, if you do well on the exam

# LEARNING ANALOGY

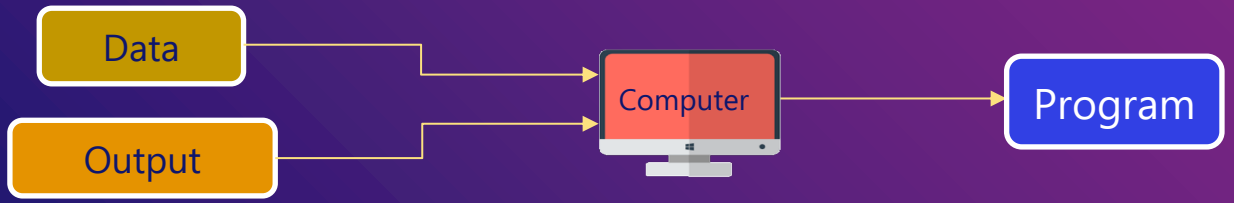
- What makes a reasonable exam?
  - If it has chemistry questions, it’s not representative of your learning
    - Remember the course was mathematics
  - If it only has questions that were already solved in the lectures, that’s a bad test of your learning
- The best practice would be
  - You study and understand the concepts with examples during the lectures
  - The exam then have “new” but “related” questions
- The good exam would test your ability to “generalize”

# MACHINE LEARNING

- Traditional programming approach




- Machine Learning approach



# WHY STUDY ML

- Technological or engineering motivation
  - To build computer systems that can improve their performance at tasks with experience (data)
- Cognitive science motivation
  - To understand better how humans learn by modeling the learning process

- To understand better properties of various algorithms for function (or concept to be learned) approximation
  - How much data is required and how to represent that data?
  - How accurate they can be?
  - How to choose optimal data for training?

# TYPES OF LEARNING

- Deductive Learning:
  - Theory
  - Hypothesis
  - Observation
  - Confirmation

- Inductive Learning:
  - Information
  - Pattern
  - Tentative Hypothesis
  - Theory

# TYPES OF LEARNING

- Deductive Learning:
  - Works on existing facts and knowledge
  - Does not generate "new" knowledge at all
  - Makes the reasoning system more efficient
- Example:
- Concept to be learned - Throwing a ball in air
- How Deductive Learning works?
  - We know Newton's Law of Gravitation
  - So, we conclude that if we let a ball go, it will certainly fall downwards

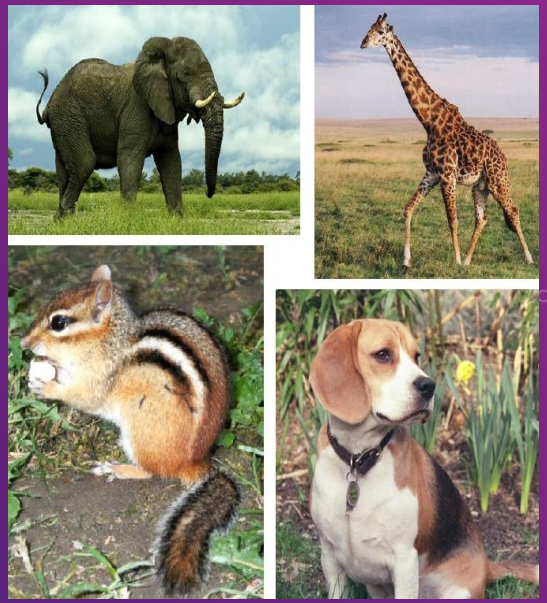
- Inductive Learning:
  - Takes examples of a concept and generalizes rather than starting with existing knowledge
  - Generates “new” knowledge
  - Has “scope of error”

# TYPES OF LEARNING

- Inductive Learning:
- Example:
- Concept to be learned - Throwing a ball in air
- How Inductive Learning works?
  - Take examples of the concept to be learned
  - 1 example – 1 time we throw a ball in the air
  - 50 examples – 50 times we throw a ball in the air
  - 100 examples – 100 times we throw a ball in the air
  - Learn from examples, we throw a ball 100 times in the air and learned that every time we throw the ball in the air, it falls downward
  - Generalize the concept learned from examples
  - We conclude that if we let a ball go, it will certainly fall downwards

# HOW LEARNING IS REPRESENTED?

- Lets try out an example



# HOW LEARNING IS REPRESENTED?

- How would you write a program to distinguish a picture of yourself from a picture of someone else?
  - Provide example pictures of yourself and pictures of other people and let a classifier learn to distinguish the two
- How would you write a program to distinguish cancerous cells from normal cells?
  - Provide examples of cancerous and normal cells and let a classifier learn to distinguish the two
- These are known as a classification tasks

# TYPES OF ML

- ML algorithms can be broadly classified into 3 types
- Supervised Learning
  - Classification (Fraud Detection, Sentiment Analysis, Gender Prediction)
  - Regression (Weather Forecasting, Population Growth Estimation, House Price Prediction)
- Unsupervised Learning
  - Dimensionality Reduction (Meaningful Compression, Big Data Visualisation)
  - Clustering (Customer Segmentation, Recommender Systems, Text Categorization)
- Reinforcement Learning
  - Real-time Decisions, Robot Navigation, Skill Acquisition

# DATA

- Data is raw facts and figures (text, image, video, audio)
- Information is processed form of data
- Data can be:
  - Structured data
  - Unstructured data
  - Semi-Structured data
- Quantitative data can be described using numbers
  - Mathematical operations possible
- Qualitative data cannot be described using numbers
  - Mathematics operations not possible
  - It is generally thought of as being described using "natural" categories and language
- Example: Data of a coffee Shop
  - Name of the coffee shop (qualitative)
  - Monthly revenue (quantitative)
  - Zip code (qualitative)
  - Average monthly customers (quantitative)
  - Country of the coffee origin (qualitative)

# DATA ANNOTATIONS FOR ML

- Data annotation (or tagging), is the process of labeling data to make it usable for data science
  - Performed by domain experts (humans – annotators or taggers)
  - Requires a lot of effort, time, and cost
- Example: Customer reviews annotated as positive, negative, or neutral sentiments

| Raw Data |
| --- |
| iPhone10 is a good mobile |
| Battery of this phone is bad |
| I am using iPhone10 |

| Annotated Data | |
| --- | --- |
| Comment / Review | Sentiment |
| iPhone10 is a good mobile | Positive |
| Battery of this phone is bad | Negative |
| I am using iPhone10 | Neutral |

- Data is available as
  - Annotated data
  - Un-Annotated data
  - Semi-Annotated data

# DATA ANNOTATIONS FOR ML

- For accurate learning, it is important to have balanced data
  - For each class, the dataset must contain the same number of instances
- Data must be large in quantity, good quality and balanced
- For any learning setting, test data must be annotated to evaluate the performance of a model

# HOW EXAMPLES IN ML ARE REPRESENTED

- Examples in ML are called instances, or data points, or observations
- An ML example is nothing but input + output, which is represented as “attribute-value” pair
- Input could be single valued, but mostly it is vector valued
- Examples of attribute(s) and value(s):
  - Categorical /Ordinal, e.g., Male, Female, Yes, No, etc.
  - Numeric
    - Discrete – e.g., 10, 25, 10000, etc.
    - Continuous – e.g., 3.5, 5.9, etc.
- Representation of input, set of attributes with possible values
  - The key is to try to identify the most discriminating attributes for a learning problem

| Attribute | Possible Values | | |
|---|---|---|---|
| Height | Short | Medium | Tall |
| Weight | Small | Medium | Heavy |
| Beard | Yes | No | – |

- Representation of output
  - Single attribute with possible values, e.g., gender - male, female

# HOW EXAMPLES IN ML ARE REPRESENTED

- Note the difference between “attribute” and “value”
  - “Height” is an attribute and “Tall” is the value of that attribute
- Instance (example) is a vector of attribute values
  - 3 possible instances for the gender identification learning problem

| Instance No. | Height | Weight | Beard | Gender |
|---|---|---|---|---|
| 1 | Short | Medium | No | Female |
| 2 | Tall | Heavy | Yes | Male |
| 3 | Medium | Medium | No | Female |

# HOW EXAMPLES IN ML ARE REPRESENTED

- Learn from input to predict output

| Instance no. | Height | Weight | Beard | Hair Length | Scarf | Gender |
|---|---|---|---|---|---|---|
| 1 | 180.3 | 96 | No | Bald | No | Male |
| 2 | 170 | 60 | No | Long | No | Female |
| 3 | 178.5 | 89 | Yes | Short | No | Male |
| 4 | 163.2 | 75 | Yes | Long | No | Male |
| 5 | 175 | 85 | No | Short | No | Male |
| 6 | 165 | 64 | No | Medium | Yes | Female |

Set of input vectors

Output

# PHASES OF ML

- Training phase
  - ML algorithm tries to learn from the training data and outputs a ML trained model which can then be used to make predictions
  - Training phase creates the “model” using the “data” (training data)
- Testing phase
  - Performance of the trained model (created in the training phase) is evaluated (on the test data) using the evaluation measure(s)
  - Testing phase checks the “error” in the “model” using “data” (test data)
- Trained model (or model) is used to make predictions on new (unseen) data
- If a model produces an accuracy of 80% on a large test set, then presumably it will correctly classify 80% of unseen data
  - If a trained model performs well on large test data, it will perform well on real-time data

# PHASES OF ML

- For both training and testing, we need data, therefore, we split the available data into
  - Train data (or train set)
  - Test data (or test set)
- Standard approach for data split (mostly used)
  - 2/3 train set
  - 1/3 test set
- Important: Train set and test set must be disjoint, i.e., example(s) in the train set should not occur in the test set and vice versa
- Two main approaches to data split
  - Random split
  - Class balanced split

# PHASES OF ML

- Balanced dataset = 600 instances (A = 300, B = 300)
- Train/Test split ratio is 67%-33%
- Random split
  - Train set = 400 instances (A = 175, B = 225)
  - Test set = 200 instances (A = 125, B = 75)
- Class balanced split
  - Train set = 400 instances (A = 200, B = 200)
  - Test set = 200 instances (A = 100, B = 100)

- Un-balanced dataset = 900 instances (A = 600, B = 300)
- Train/Test split ratio is 67%-33%
- Random split
  - Train set = 600 instances (A = 500, B = 100)
  - Test set = 300 instances (A = 100, B = 200)
- Class balanced split
  - Train set = 600 instances (A = 400, B = 200)
  - Test set = 300 instances (A = 200, B = 100)

# ML TRAINING REGIMES

- Batch method
  - All training examples are used at once to compute the hypothesis
- Incremental method
  - All training examples are used iteratively to refine a current hypothesis, one at time and randomly
- On-line method
  - The training examples are used as they become available, one at a time

# ML MODELS EVALUATION

- Cross-validation is a statistical method used to estimate the performance of ML models
- There are numerous ways to cross-validate a model
  - Aims is to keep test data aside from the training data
- 3 steps are involved in the process of cross-validation
  - Separate a part of the data from the rest
  - Utilize the rest of the data to train the ML model
  - Once the model is ready, validate it using the data that was separated earlier

# ML MODELS EVALUATION

- Different types of cross-validation
  - Leave p-out cross-validation
  - Leave one-out cross-validation
  - Holdout cross-validation
  - k-fold cross-validation
  - Stratified k-fold cross-validation
  - Repeated random subsampling validation
  - Rolling cross-validation

# ML MODELS EVALUATION METRICS

- Which evaluation metric should be used to measure the performance of a ML model?
- How reliable are the predicted results?
- How much to believe on what was learned?
- If two models are equal in performance, which one to prefer?

- Focus is on the predictive capability of a model, rather than how fast it classifies

# ML MODELS EVALUATION METRICS

- Different metric exists to evaluate the performance of a ML classification model
  - Accuracy
  - Precision
  - Recall
  - F-measure
- All of these rely on the confusion matrix

# ML MODELS EVALUATION METRICS

- Confusion matrix is a table of (m x n) that is often used to describe the performance of a ML classification model on test data (for which the class labels are known)

Predicted class

| Actual class | Yes | No | Total |
|---|---|---|---|
| Yes | TP | FN | P |
| No | FP | TN | N |
| Total | P' | N' | P+N |

- TP (True Positives): Positive instances that were correctly labelled by the model
- TN (True Negatives): Negative instances that were correctly labelled by the model
- FP (False Positives): Negative instances that were incorrectly labelled by the model (as positives)
- FN (False Negatives): Positive instances that were incorrectly labelled by the model (as negatives)

# ML MODELS EVALUATION METRICS

- Accuracy is the percentage of test instances that are correctly classified (or recognition rate)
  - Accuracy = TP + TN / P + N
- Error rate (or misclassification rate) = 1 – accuracy

- Limitations:
  - Suppose a binary classification problem, where total instances are 1000
  - 990 are positive, and 10 are negative
- If the model predicts everything to be positive, what will be the accuracy?
- Accuracy = 990 / 1000 = 99%
  - Model does not detect any negative instances, however, it is still 99% accurate

# ML MODELS EVALUATION METRICS

- Accuracy is the function of sensitivity and specificity
- Sensitivity is referred to as true positive rate (TPR), i.e., the proportion of positive instances that are correctly classified
  - Sensitivity = TP / P
- Specificity is referred to as true negative rate (TNR), i.e., the proportion of negative instances that are correctly classified
  - Specificity = TN / N

# ML MODELS EVALUATION METRICS

- An example to understand precision and recall:
- Suppose this morning, you got a phone call
- A stranger on the line said, “Congratulations! You have won a lottery of 100 Crores! I just need you to provide me your bank account details, and the money will be deposited in your bank account right way.”
- What would you do?
  - Tricky, right?
- You assumed the call is a prank (or scam), and deny to provide any information
  - If the assumption is right, you have saved yourself
  - If it is wrong, the decision would cost you 100 Crores!

# ML MODELS EVALUATION METRICS

- An example to understand precision and recall:
- If you would have believed the stranger and provided your bank details, and the call was in fact a scam
  - You would have committed a type I error, also known as a false positive
- If you would have ignored the stranger's request, but later found out that the call was not a scam
  - You would have committed a type II error, or a false negative

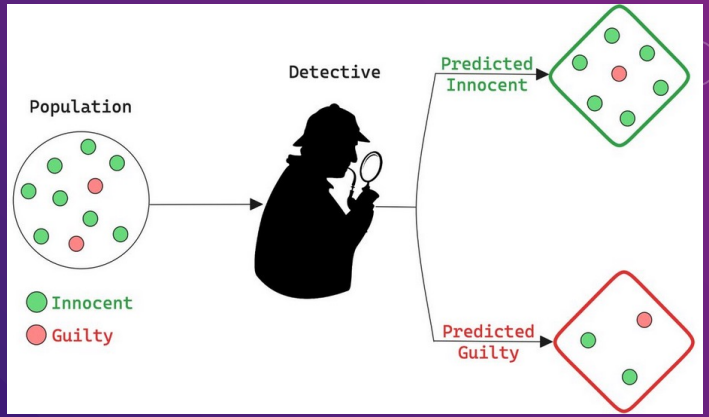
- Precision (or exactness) is the percentage of instances predicted as positives are actually positive (or are relevant)
  - Precision = TP / TP + FP
- Recall (or completeness) is the percentage of positive instances predicted as positives
  - Recall = TP / TP + FN

# ML MODELS EVALUATION METRICS

- An example to understand precision and recall:
- You called the bank to ensure your existing accounts were safe and all your credits were secure
- After listening to your story, the bank representative informed you that all your accounts were safe
- However, in order to ensure that there is no future risk, the bank representative asked you to recall all instances in the last six months wherein you might have shared your account details with another person
- What are the chances that you will be able to recall all such instances precisely?
- Let's say there are 10 such instances in reality
- However, you narrated 20 instances to finally spell out the 10 correct instances
- The recall will be a 100%, but precision will only be 50%

# ML MODELS EVALUATION METRICS

- Another example to understand precision and recall:
- Let's say there are 10 people in a town and 2 of them have committed a crime
- So, 8 are innocent, 2 are guilty
- A detective was hired to catch (predict) the guilty
- Detective accuses 3 people of being guilty and 7 innocent
  - Out of the 3 predicted accused, 1 is guilty in reality
  - Out of the 7 predicted innocent, 1 is guilty in reality



# ML MODELS EVALUATION METRICS

- Another example to understand precision and recall:
- Detective accuses people of a crime, precision is about how many of those accused are truly guilty
- We don't want false accusations
- TP = 1, FP = 2
- Precision = TP / TP + FP = 1 / (1 + 2) = 0.33

- Recall, focuses on completeness
- For the detective, it's not just about correctly accusing the guilty, but ensuring no guilty person goes free
- Detective was able to find only half of the guilty persons
- TP = 1, FN = 1
- Recall = TP / TP + FN = 1 / (1 + 1) = 0.50

| | Guilty | Innocent |
|---|---|---|
| Guilty | TP = 1 | FP = 2 |
| Innocent | FN = 1 | TN = 6 |

# ML MODELS EVALUATION METRICS

- Trade-off between precision and recall
- Detective's work (prediction) is to trade-off between precision (avoiding false accusations) and recall (catching all guilty)
- In a small peaceful town, false accusations might cause social unrest, so aim for higher precision
- In a crime-ridden town, catching all criminals is crucial, favouring higher recall

# ML MODELS EVALUATION METRICS

- Trade-off between precision and recall
- If you have to recall everything, you will have to keep generating results which are not accurate, hence lowering your precision
- If a customer is shown a lot of irrelevant results and very few relevant results against his search query
- He will not keep browsing each and every product forever to finally find the one he intends to buy
- Hence the underlying model would need a fix to balance the recall and precision

- Similar thing happens when a model tries to maximize precision

# ML MODELS EVALUATION METRICS

- A simpler metric which takes into account both precision and recall is known as F1-score
  - It is the harmonic mean of precision and recall
- F1-score = 2 * (Precision * Recall / Precision + Recall)

# ML MODELS EVALUATION METRICS

- How to calculate accuracy for regression?
  - Accuracy is a measure for classification, not regression
  - Not possible to calculate accuracy for regression
- Performance of a regression model is calculated as an error in the predicted vs actual values
- Three error (evaluation) metrics commonly used are:
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Error (MAE)

# ML MODELS EVALUATION METRICS

- Mean Squared Error (MSE)
  - MSE = $1/n \sum(y_i - y_i')^2$
  - $y_i$ is the *i*'th expected value and $y_i'$ is the *i*'th predicted value
- Root Mean Squared Error (RMSE)
  - RMSE = $\sqrt{}$ MSE
- Mean Absolute Error (MAE)
  - MAE = $1/n (\sum ABS(y_i - y_i'))$

# THANKS