**Slide1:**

While we are getting started, I would love to know what brings you to this workshop?

**Those were some interesting answers.**

So let’s get started; schedule. I will cover the basics of ML and about classification and clustering quickly so that we have more time for the hands-on.

0:05 - 0:15 About Machine Learning

0:15 - 0:30 About Classification and Clustering

0:30 - 1:20 Hands-on with Jupyter Notebook

1:20 - 1:30 Wrap-up and Discussion

**Slide 2:**

But before going into more details I would like to begin by acknowledging that I .

**Slide 3:**

Active participation makes the session so much fun and gives me and your peers much more energy. Your voices and perspectives enlivens the session. We encourage you to engage with each other and us.

The participants window lists everyone in the session and click the icons at the bottom to communicate with the us.

You can also use the Chat windows to comment or ask questions at any time. It is also a good place to share problems with your audio connection.

**Slide 4:**

This workshop is eligible for the Canadian Certificate in Digital Humanities <https://ccdhhn.ca/>. This is a program that allows you to claim non-credit workshops and training toward a certificate. If you are interested in claiming 2 hours from this workshop please fill out this form: <https://ubc.ca1.qualtrics.com/jfe/form/SV_cwiqRxedypNPhA2>. This is how we track attendance for the certificate, it will only be shared during the workshop and will not be emailed to you after."

**Slide 5:**

So, to touch various viewpoints of machine learning regression, we have the following learning objectives for this workshop:

**Slide 6:**

For hands-on exercises, we will use [Python](https://www.python.org/) on [Jupyter Notebooks](https://jupyter.org/). You don’t need to have Python installed. Please make sure that you have a [UBC Syzygy](https://ubc.syzygy.ca/) or a [Google Colaboratory](https://colab.research.google.com/) account. (You will need a CWL login to access Syzygy.) hands-on exercises, programming tools and libraries, such as [Python] and [scikit-learn] prior familiarity with Python programming is recommended, we do not study the codes in detail

**Slide 7:**  
What comes to your mind when you hear of the word machine learning?

Machine learning is a field of computer science, teaching computers to learn from data, without explicitly defining the rules applicable to the problem.

**Slide 8:**

Let’s do a simple exercise. We see a table with two columns X and Y. They are some random variable. For each value of X there is a corresponding value of Y. Observer if there is a pattern (a relationship between X & Y)? If so what do you think should come in place of “?”.

Well there is no correct answer. All your guesses were good. Interestingly what you did was a prediction the same as what a machine learning model would do: observe the data, try to look for patterns and make predictions following that observed pattern. Specifically, what we did here was nothing but Classification, we will see how.

**Slide 9:**

Let’s do another exercise. This time there is no “Y” so we don’t know what we are predicting. But we can still see there is a pattern in these different X values. Maybe let’s try grouping them in 2 sets. What do you think the two sets would be?

Was this exercise the same as the previous one?   
What was different?

Well, this was clustering.

**Slide 10:**

But before diving into regression, let’s understand more about machine learning. Although I briefly defined the steps of ML in the last slide, its not that simple. There are many other steps involved.

- \*\*Data Collection:\*\*: gathering and preparing data, he quality of the data used in the model directly impacts its accuracy and effectiveness

- \*\*Data Preprocessing:\*\* cleaned, transformed, and formatted into a suitable format

- \*\*Model Training:\*\* Once the data is ready, it is used to train a machine learning model.

- \*\*Model Evaluation:\*\* After the model has been trained, it needs to be evaluated to ensure that it is performing accurately. This is done using a test set of data that was not used during the training process.

- \*\*Model Deployment:\*\* Once the model has been trained and evaluated, it can be deployed for use in a production environment. This involves integrating the model into a larger system and ensuring that it can handle real-world data. This step involves choosing a suitable employment platform.

- \*\*Model Improvement:\*\* Machine learning is an iterative process, so the model may need to be improved over time. This can involve retraining or fine-tuning its parameters the with new data or continues stream of data.

**Slide 11:**

Let’s now look at the type of ML. There are different types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning, each with its own set of algorithms and techniques.

\* Supervised Learning:

an algorithm is trained on a labeled dataset, meaning that the dataset has input features (X) and corresponding output labels (Y). The goal of supervised learning is to learn a function that maps the input features to the output labels. Once the model is trained, it can be used to make predictions on new data. Examples of supervised learning tasks include image classification, speech recognition, and regression analysis.

\* Unsupervised Learning:

Unsupervised learning is a type of machine learning where the algorithm is trained on an unlabeled dataset, meaning that there are no output labels (Y) associated with the input features (X). The goal of unsupervised learning is to learn patterns and structure in the data without the help of a labeled dataset. Examples of unsupervised learning tasks include clustering, anomaly detection, and dimensionality reduction.

\* Reinforcement Learning:

Reinforcement learning is a type of machine learning that involves an agent interacting with an environment to learn how to make decisions that maximize a reward. The agent receives feedback from the environment in the form of rewards or penalties, and its goal is to learn a policy that maximizes the expected long-term reward. Reinforcement learning is often used in robotics, game playing, and control systems.

\* Transfer Learning:

Transfer learning refers to a technique in which a pre-trained model, typically trained on a large dataset, is used as a starting point for training a new model on a smaller dataset or a different but related task. The idea is that the knowledge learned from the pre-trained model can be transferred to the new task, allowing the new model to start with some level of knowledge or "transfer" from the previous task, which can potentially improve its performance and reduce the amount of training data required.

**Slide 12:**  
In this section, we will provide an overview of the steps in machine learning, some algorithms within the various types of ml. What are algorithms? Set of rules in programming in this case to make our machine learning models.

**Slide 13:**

Before going to the main step, it is a good idea to learn a little bit about the steps right before and after the model training step

**Continuous vs Categorical Variables**When working with data, we often encounter two main types of features: continuous and categorical.

* Continuous Features: These are numeric values that can take any real number. Examples include age, income, temperature, and height.
* Categorical Features: These are non-numeric and represent categories or labels. Examples include gender, color, and product type.

Understanding the type of features in your dataset is crucial because it affects how you handle and preprocess them.

**Handling missing values - drop, impute, forward backward fill**Missing data is a common challenge in real-world datasets. How you deal with missing values can greatly impact your model's performance.

Here are some strategies for handling missing data:

* Imputation: Filling in missing values with a suitable estimate (e.g., mean, median, mode).
* Deletion: Removing rows or columns with missing data (use with caution).
* Advanced techniques: Using machine learning to predict missing values based on existing data.

**Feature scaling: standardization, min-max scaling, Feature selection**

Feature scaling is the process of standardizing the range of independent variables or features of data. This step is crucial for many machine learning algorithms, such as those based on distances or gradients.

Common techniques for feature scaling include:

* Min-Max Scaling: Rescales features to a specified range, often [0, 1].
* Standardization (Z-score Scaling): Scales features to have a mean of 0 and a standard deviation of 1.
* Robust Scaling: Scales features based on their median and interquartile range, making it robust to outliers.

Feature scaling ensures that all features contribute equally to the model, preventing biases due to the magnitude of the values.

Not all features are equally valuable for your model. Feature selection is the process of choosing the most relevant features while discarding irrelevant ones.

**Slide 14:**

Those were the steps to kind mind before training your model, but let’s also look at what happens after training your model.

Model evaluation is a critical aspect of machine learning. Two common issues to watch for are:

* Underfitting:
  + Occurs when a model is too simple to capture the underlying patterns in the data.
  + Remedies: Increase model complexity, add more features.
* Overfitting:
  + Occurs when a model is excessively complex.
  + Remedies: Reduce model complexity, increase the amount of training data.

Balancing between underfitting and overfitting is a key challenge in building robust machine learning models. Proper model evaluation helps identify and address these issues for improved performance.

**Slide 15:**

Now get back to understanding the type of ml in detail by looking at various ml methods and algorithms within them.

One of the ml method in Supervised ml is Regression:

* Used to predict a continuous numeric output.
* Examples: Predicting house prices, stock prices, temperature, or any quantity that can take real-number values.

classification:

* Used to categorize data into predefined classes or labels.
* Examples: Spam detection, image recognition, sentiment analysis, and medical diagnosis (e.g., benign/malignant).

Clustering is a machine learning technique used for unsupervised learning. Unlike regression and classification, which involve making predictions or categorizing data, clustering focuses on discovering inherent structures or groupings within the data.

Key points about clustering:

* Objective:
  + The primary goal is to group similar data points together, forming clusters or subgroups.
* Applications:
  + Clustering is widely used in customer segmentation, recommendation systems, anomaly detection, and image segmentation.
  + It can also help in exploratory data analysis, revealing hidden patterns.

**Slide 16:**

For each of those methods that we discussed earlier, there are various algorithms. There are various set of rules and mathematics to achieve those goals.

We won’t get into details of any of those. But further in today’s session we would try to implement multiple types of classification algorithms.

**Slide 17:**

When we implement it in Python, we will use some python libraries. In Python, libraries, also known as modules or packages, are collections of pre-written code and functions that extend the capabilities of the Python programming language.

Numpy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Pandas: pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

Scikit-learn is a free software machine learning library for the Python programming language.

**Slide 18:**

Classification uses supervised learning, where the algorithm is trained on a labeled dataset to learn the relationship between input features and output class labels. The trained model can then be used to predict the class labels of new, unseen data.

Clustering uses unsupervised learning, where the algorithm groups similar data points together based on their similarity or distance from each other. The number of clusters may be predefined or learned from the data.

Classification metrics and algorithms

Clustering algorithms

**Slide 19:**

My apologies for the misunderstanding. Here's a script you can use to explain the confusion matrix and Type 1 and Type 2 errors to beginners during a teaching session:

Introduction to Confusion Matrix

What is a Confusion Matrix?

A confusion matrix is a table that allows us to evaluate the performance of a classification algorithm. It helps us visualize the model's predictions and how they match up with the actual outcomes. It consists of four essential components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

True Positives (TP)

Let's start with True Positives. These are cases where the model correctly predicted a positive outcome when the actual outcome is indeed positive. In simple terms, the model got it right!

True Negatives (TN)

On the other hand, True Negatives are cases where the model correctly predicted a negative outcome when the actual outcome is indeed negative. Another win for the model!

False Positives (FP)

Now, let's move on to False Positives. These are cases where the model incorrectly predicted a positive outcome when the actual outcome is negative. In other words, it made a mistake by saying "positive" when it should have been "negative." This is also known as a Type 1 Error.

False Negatives (FN)

Lastly, we have False Negatives. These are cases where the model incorrectly predicted a negative outcome when the actual outcome is positive. It made a mistake by saying "negative" when it should have been "positive." This is known as a Type 2 Error.

Type 1 and Type 2 Errors

So, what's the difference between Type 1 and Type 2 errors?

* Type 1 Error (False Positive): Think of it as a false alarm. For example, your spam filter incorrectly marking an important email as spam. It's a "positive" prediction when it should be "negative."
* Type 2 Error (False Negative): This is a missed opportunity. For instance, your medical test failing to detect a disease that is actually present. It's a "negative" prediction when it should be "positive."

In summary, the confusion matrix and Type 1 and Type 2 errors help us understand the strengths and weaknesses of our model's predictions. By analyzing these components, we can make necessary adjustments and improvements to our models, ensuring they perform better in real-world scenarios."

**Slide 20:**  
  
Some of the differentiation factors of machine learning models also count as their limits:

\* Garbage In = Garbage Out

In machine learning, the quality of the output model is directly dependent on the quality of the input data used to train it. If the input data is incomplete, noisy, or biased, the resulting model may be inaccurate or unreliable.

For example, suppose a machine learning model is being developed to predict which loan applications are likely to be approved by a bank. If the training dataset only contains loan applications from a particular demographic group or geographic region, the resulting model may be biased towards that group or region and may not generalize well to other groups or regions. This could lead to discrimination and unfair lending practices.

\* Data Limitation

Machine learning algorithms are only as good as the data they are trained on. If the data is biased, incomplete, or noisy, the algorithm may not be able to learn the underlying patterns or may learn incorrect patterns. Also, machine learning models require large amounts of labeled data for training, which can be expensive and time-consuming to obtain.

\* Generalization and overfitting

Machine learning models are typically trained on a specific dataset, and their ability to generalize to new data outside of that dataset may be limited. Overfitting can occur if the model is too complex or if it is trained on a small dataset, causing it to perform well on the training data but poorly on new data. When a model is overfitting, it is essentially memorizing the training data rather than learning the underlying patterns in the data.

\* Inability to explain answers

Machine learning models can be complex and difficult to interpret, making it challenging to understand why they make certain predictions or decisions. This can be a problem in domains such as healthcare or finance where it is important to be able to understand the rationale behind a decision.

\* Ethics and Bias Limitations

Machine learning algorithms can amplify existing biases in the data they are trained on, leading to unfair or discriminatory outcomes. There is a risk of unintended consequences when using machine learning algorithms in sensitive areas such as criminal justice, hiring decisions, and loan applications. One example of bias in machine learning is in facial recognition technology. Studies have shown that facial recognition systems are less accurate in identifying people with darker skin tones and women. This bias can lead to misidentification, which can have serious consequences, such as wrongful arrest or discrimination in hiring. In the context of healthcare, machine learning algorithms can also perpetuate bias and discrimination. For example, if the algorithm is trained on biased data, it may make less accurate predictions for certain demographic groups, such as racial minorities or people with disabilities.

\* Computational Limitations

Machine learning algorithms can be computationally expensive and require a lot of computing power to train and run. This can be a barrier to adoption in applications where real-time or low-power processing is required. One example of computational limitations in machine learning is training deep neural networks. Deep neural networks are a type of machine learning algorithm that can learn complex patterns in data by using many layers of interconnected nodes. However, training these models can be computationally expensive, requiring significant computing power and memory.

For example, training a state-of-the-art natural language processing model like BERT (Bidirectional Encoder Representations from Transformers) can take weeks or even months on a large cluster of GPUs. This limits the ability of smaller organizations or individuals with limited computing resources to develop or use these models.

**Slide 21:**Let’s dive deeper   
  
**Slide 22:**

Ethical concerns & end