**Strathmore University**

**Tools to use:** python -m jupyterlab or python -m notebook.

**Online IDE**

JetBrains

<https://datalore.jetbrains.com/notebook/Auv3NM6JmTq575D7671W9M/U2XZLznvlo9swv5SwERlOJ/>

Goggle Collab

<https://colab.research.google.com/drive/1isgUPuBYdNY9ZYtHqiflJ7K_lTG1TNVr#scrollTo=GJEsWysMNTmv>

**Offline-IDE** -Vs-Code and PyCharm

**Database**:

MySQL workbench/ MariaDB with Xamp

**Deployment:**

-Django Framework and Stream lit Package

-CSS , HTML and JS

-Power BI.

**Database Tutorials.**

<https://www.youtube.com/playlist?list=PLZoTAELRMXVNMRWlVf0bDDSxNEn38u9Cl>

**Recommender systems Tutorials:**

<https://www.youtube.com/watch?v=_hf_y-_sj5Y&list=PLZoTAELRMXVN7QGpcuN-Vg35Hgjp3htvi>

**Feature Engineering Tutorials:**

<https://www.youtube.com/watch?v=6WDFfaYtN6s&list=PLZoTAELRMXVPwYGE2PXD3x0bfKnR0cJjN>

<https://www.youtube.com/watch?v=F-X82zhIfBo&list=PLZoTAELRMXVPzj1D0i_6ajJ6gyD22b3jh&index=2>

Machine Learning Tutorial

<https://www.jovian.ai/>

https://jovian.ai/learn/machine-learning-with-python-zero-to-gbms/lesson/linear-regression-with-scikit-learn

**Python Data Visualization Tutorials:**

https://jovian.ai/aakashns/python-matplotlib-data-visualization

**Power Bi Tutorials**

**Timetable-master’s in data science and Analytics.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Time** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** | **Sunday** |
| **8:30PM**  **|**  **10:00PM** | **ML** | **Thesis** | **Thesis** | **DB/WebSP** | **Web Dev**  **Django** |  | **ML** |
| **12-5PM** | **PowerBi/DB/WebSP** | | | | |  |  |
| **8-11PM** |  |  |  |  |  | **Thesis** | **ML** |
| **1-4PM** |  |  |  |  |  | **ML** |  |

**Machine Learning Unit.**

**Lesson 2:**

Two types of machine learning.

1. Supervised Machine Learning
2. Unsupervised Machine Learning

**Supervised**

**-**Main types of supervised machine learning are **Regression** and **Classification**.

**Regression**.

The common algorithms are

* **Linear regression**
* **Decision Tree**
* **Random Forest**- is a collection of a large number of uncorrelated decision trees each tree comes from bootstrap sample of the training data

Prediction is by running the new- record through the pre-grown forest and averaging the end-node values

The process is a very successful.

* **Gradient-Boosting**- A collection of weak learners and converted into a strong learner through gradual improvement of one tree to the next.

This method is generally better than random forest.

Model Ensembling; Random Forest- **Bagging**, GB=**Boosting**

* Neural Networks and Deep Learning.
* Lasso regression
* Ridge Regression
* Support Vector regression.

**Classification**.

Classification types are grouped into

1. Binary classification-Instance belongs to one of 2 classes (Churner and Non-churner)

Common algorithm

* Logistic regression.
* K-Nearest Neighbors
* Decision Tree
* Support Vector Machine
* Naïve Bayes

1. Multi-Class Classification-Instance belong to one of the multiple classes (High spender or mid spender or Low spender.
2. Multi-label classification- Instance can be assigned to more than one class (A movie is Horror and Romance and Adventure)

Common algorithms are:

* Multi-label decision tree
* Multi-label random Forests
* Multi-label; Gradient Boosting

**Unsupervised Machine Learning**

To establish the relationship among records without defined target variable (unlabeled data).

**Clustering-** Group similar pieces

* K-Means Clustering

**Association Rule Mining**- Uncovering the relationship between objects e.g Market Basket Analysis

* Apriori Algorithm

Others are:

* Neural networks (SOM self-organizing maps)
* Hierarchy clustering
* Principal Components Analysis
* Anomaly detection
* KNN

**Semi-Supervised Machine Learning**

**Reinforcement Learning**

**INSTANCE -BASED LEARNING vs MODEL-BASED LEARNING**

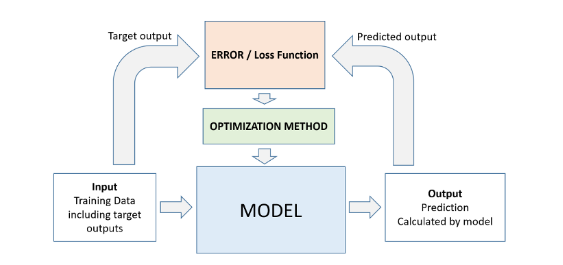
**Instance-Based** – Uses the entre dataset as model e.g KNN.

**Model-Based**- Uses training data to create a model that has parameters learned from the training dataset. E.g., Build a rule-set or parametrized model using training example Linear Regression.

**NB.**

Every Machine learning Problem has three components

* Model
* Cost Function
* Optimizer



**How to Approach a Machine Learning Problem**

Here's a strategy you can apply to approach any machine learning problem:

1. Explore the data and find correlations between inputs and targets
2. Pick the right model, loss functions and optimizer for the problem at hand
3. Scale numeric variables and one-hot encode categorical data
4. Set aside a test set (using a fraction of the training set)
5. Train the model
6. Make predictions on the test set and compute the loss

We'll apply this process to several problems in future tutorials.