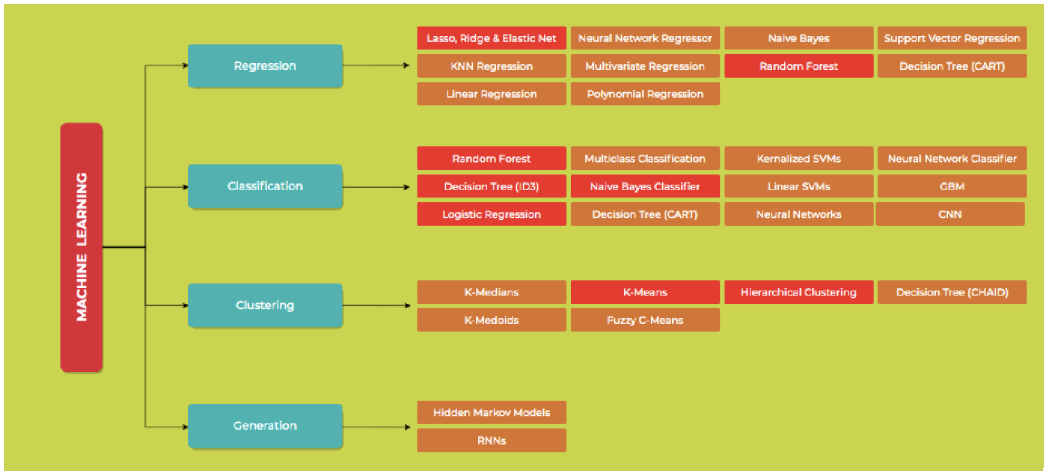
Using machine learning to study literature

In this activity, we will use Machine Learning (ML) to study literature from different famous authors.

In machine learning, the procedure is to train the computer to recognize objects using one of a variety of algorithms. The **training set** contains some number of objects with their association, for example, this artwork is associated with this painter.

Once the machine has been trained, then the training is applied to a **test set** of related objects. In our case, we will ask the trained computer to determine which of the three authors -- Shakespeare, Victor Hugo, or Oscar Wilde -- the literature most resembles. Our test set will include a number of famous writings.

There are four categories of approaches, or algorithms, that are common to machine learning: regression, classification, clustering, and generation. The graphic below captures these in a concise manner. As can be seen, each of the categories has some number of specific types. The most common types are: logistic regression, random forest, naive Bayes, and neural networks. We will use those four types in this activity.



Sample Data:

In[1673]:=

```
Othello = Import["http://www.gutenberg.org/cache/epub/2267/pg2267.txt"];
Hamlet = Import["http://www.gutenberg.org/cache/epub/2265/pg2265.txt"];
Macbeth = Import["http://www.gutenberg.org/cache/epub/2264/pg2264.txt"];
```

In[1676]:=

```
TheImportanceOfBeingEarnest =
  Import["http://www.gutenberg.org/cache/epub/844/pg844.txt"];
ThePictureofDorianGray =
  Import["http://www.gutenberg.org/cache/epub/174/pg174.txt"];
AnIdealHusband = Import["http://www.gutenberg.org/files/885/885-0.txt"];
```

In[1679]:=

```
LesMiserables = Import["http://www.gutenberg.org/cache/epub/135/pg135.txt"];
NotreDamedeParis =
  Import["http://www.gutenberg.org/cache/epub/2610/pg2610.txt"];
TheManWhoLaughs = Import["http://www.gutenberg.org/cache/epub/12587/pg12587.txt"];
```

Generate an author classifier from these texts:

Set up different methods of machine learning:

In[1682]:=

```
trainingset = <|
   "Shakesphere" → {Othello, Hamlet, Macbeth},
   "Widle" →
    {TheImportanceOfBeingEarnest, ThePictureofDorianGray, AnIdealHusband},
   "Hugo" → {LesMiserables, NotreDamedeParis, TheManWhoLaughs}
   |>;
methods = {"RandomForest", "NearestNeighbors", "LogisticRegression", "Markov"};
```

Run the ML using the “Classify” command and related associations:

In[1684]:=

```
author1 = Classify[trainingset, Method → methods⟦1⟧]
author2 = Classify[trainingset, Method → methods⟦2⟧]
author3 = Classify[trainingset, Method → methods⟦3⟧]
author4 = Classify[trainingset, Method → methods⟦4⟧]
```

Out[1684]=

ClassifierFunction[Input type: Text
Classes: Hugo, Shakesphere, Widle]

Data not saved. Save now

Out[1685]=

ClassifierFunction[Input type: Text
Classes: Hugo, Shakesphere, Widle]

Data not saved. Save now

Out[1686]=

ClassifierFunction[Input type: Text
Classes: Hugo, Shakesphere, Widle]

Data not saved. Save now

Out[1687]=

ClassifierFunction[Input type: Text
Classes: Hugo, Shakesphere, Widle]

Data not saved. Save now

Get information about your model, accuracy, evaluation time, etc.

In[1688]:=

```
Information[author1]
Information[author2]
Information[author3]
Information[author4]
```

Out[1688]=

Classifier information

Data type	Text
Classes	Hugo, Shakesphere, Widle
Method	RandomForest
Single evaluation time	256. ms /example
Batch evaluation speed	3.87 examples /s
Model memory	6.94 MB
Training examples used	9 examples
Training time	2.07 s

Out[1689]=

Classifier information

Data type	Text
Classes	Hugo, Shakes phere, Widle
Method	NearestNeighbors
Single evaluation time	264. ms /example
Batch evaluation speed	3.89 examples /s
Model memory	6.86 MB
Training examples used	9 examples
Training time	2.08 s

Out[1690]=

Classifier information

Data type	Text
Classes	Hugo, Shakesphere, Widle
Accurac y	(71. ±14.)%
Method	LogisticRe gression
Single evaluation time	257. ms /example
Batch evaluation speed	3.83 examples /s
Loss	0.800 ± 0.46
Model memory	6.97 MB
Training examples used	9 examples
Training time	2.56 s

Out[1691]=

Classifier information

Data type	Text
Classes	Hugo, Shakesphere, Widle
Method	Markov
Single evaluation time	1.17 s /example
Batch evaluation speed	0.861 examples /s
Model memory	3.41 MB
Training examples used	9 examples
Training time	4.53 s

Test your model:

In[1692]:=

```
test = {Hamlet, Macbeth, LesMiserables};
author1[test]
author2[test]
author3[test]
author4[test]
```

Out[1693]=

```
{Shakesphere, Shakesphere, Shakesphere}
```

Out[1694]=

```
{Shakesphere, Shakesphere, Hugo}
```

Out[1695]=

```
{Shakesphere, Shakesphere, Hugo}
```

Out[1696]=

```
{Shakesphere, Shakesphere, Hugo}
```

Find some other literature:

In[1697]:=

```
anotherHugo = Import["http://www.gutenberg.org/files/2523/2523-0.txt"];
joyce = Import["http://www.gutenberg.org/files/4300/4300-0.txt"];
irving = Import["http://www.gutenberg.org/files/41/41-0.txt"];
```

Test your model for the new data:

In[1700]:=

```
books = {anotherHugo, joyce, irving};
actual = {Hugo, Joyce, Irving};
results = Table[
   {
    actual[[image]],
    author3[books[[image]]]
   },
   {image, 1, Length[books]}];
Grid[Prepend[results, {"Actual", "Looks most like: "}], Frame → All]
```

Out[1703]=

Actual	Looks most like:
Hugo	Hugo
Joyce	Hugo
Irving	Hugo