Python and ML Bootcamp - Day 3

Evangelia Drakopoulou

PhD students: Dimitris Stavropoulos
Vasilis Tsourapis
George Zarpapis

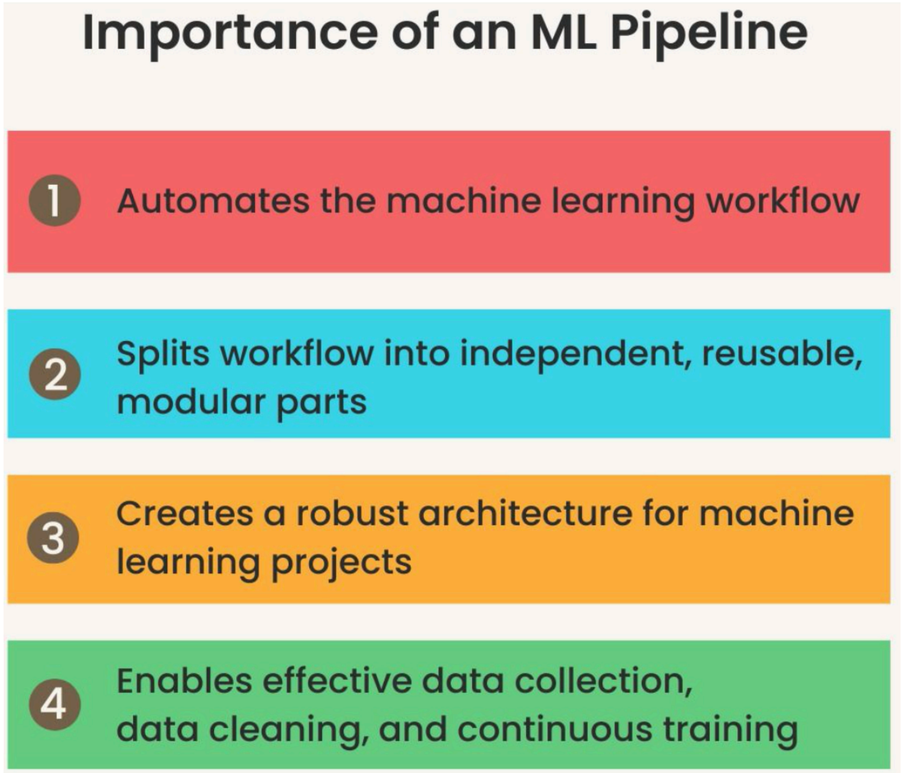
Konstantinos Paschos
Leda Liogka

Feature Selection

Pipelines

By pipeline we mean performing multiple operations (e.g., calling function after function) in a sequence, for each element of an iterable, in such a way that the output of each element is the input of the next.

A machine learning pipeline is a way to automate the end-to-end machine learning workflow.



Task: Let's check the script at GitHub:
…/Day3/feature_engineering/pipeline_example.py

https://www.geeksforgeeks.org/create-a-pipeline-in-pandas/

https://www.turing.com/kb/building-ml-pipeline-in-python-with-scikit-learn#the-necessity-of-a-machine-learning-pipeline

Feature Selection

Feature selection is a process where you automatically select those features in your data that contribute most to the prediction variable or output in which you are interested.

Benefits of performing feature selection before modeling your data are:

- **Reduces Overfitting**: Less redundant data means less opportunity to make decisions based on noise.
- **Improves Accuracy**: Less misleading data means modeling accuracy improves.
- **Reduces Training Time**: Less data means that algorithms train faster.

Task: *Let's explore real datasets!!* You are given two csv files and you know one has events from Class 1 and the other one has events from Class 0.

(i) Read .csv files: vars_class1.csv and vars_class0.csv

(ii) Merge the two data frames and explore how the data look like.

(iii) How many variables are there in the dataset? Which are the ones that could help discriminate/predict if events belong to Class 1 or Class 0?

Hint: plot each var from classes 0 & 1 in the same plot and check the differences?

Feature Selection

Task: *Let's explore real datasets!!* You are given two csv files and you know one has events from Class 1 and the other one has events from Class 0.

(i) Read .csv files: vars_class1.csv and vars_class0.csv

(ii) Merge the two data frames and explore how the data look like.

(iii) How many variables are there in the dataset? Which are the ones that could help discriminate/predict if events belong to Class 1 or Class 0?

Hint: plot each var from classes 0 & 1 in the same plot and check the differences?

Answer (1) :

Check example script: ~/Day3/feature_enginnering/feature_extr_plots.py

Feature Selection

Task: *Let's explore real datasets!!* You are given two csv files and you know one has events from Class 1 and the other one has events from Class 0.

(i) Read .csv files: vars_class1.csv and vars_class0.csv

(ii) Merge the two data frames and explore how the data look like.

(iii) How many variables are there in the dataset? Which are the ones that could help discriminate/predict if events belong to Class 1 or Class 0?

Hint: plot each var from classes 0 & 1 in the same plot and check the differences?

Answer (2) :

Also check the example script: ~/Day3/feature_enginnering/feature_extr_multiple_plots.py

to create multiple plots at once.

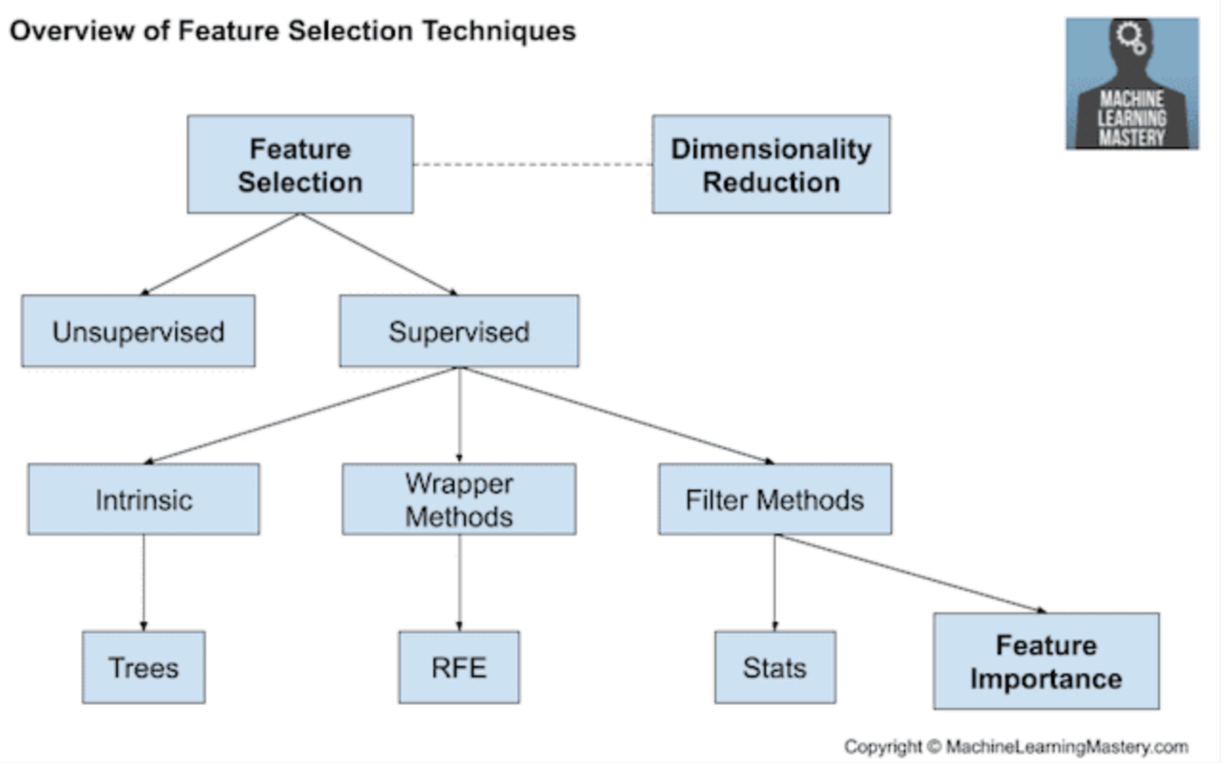
```
from sklearn.utils import shuffle
```

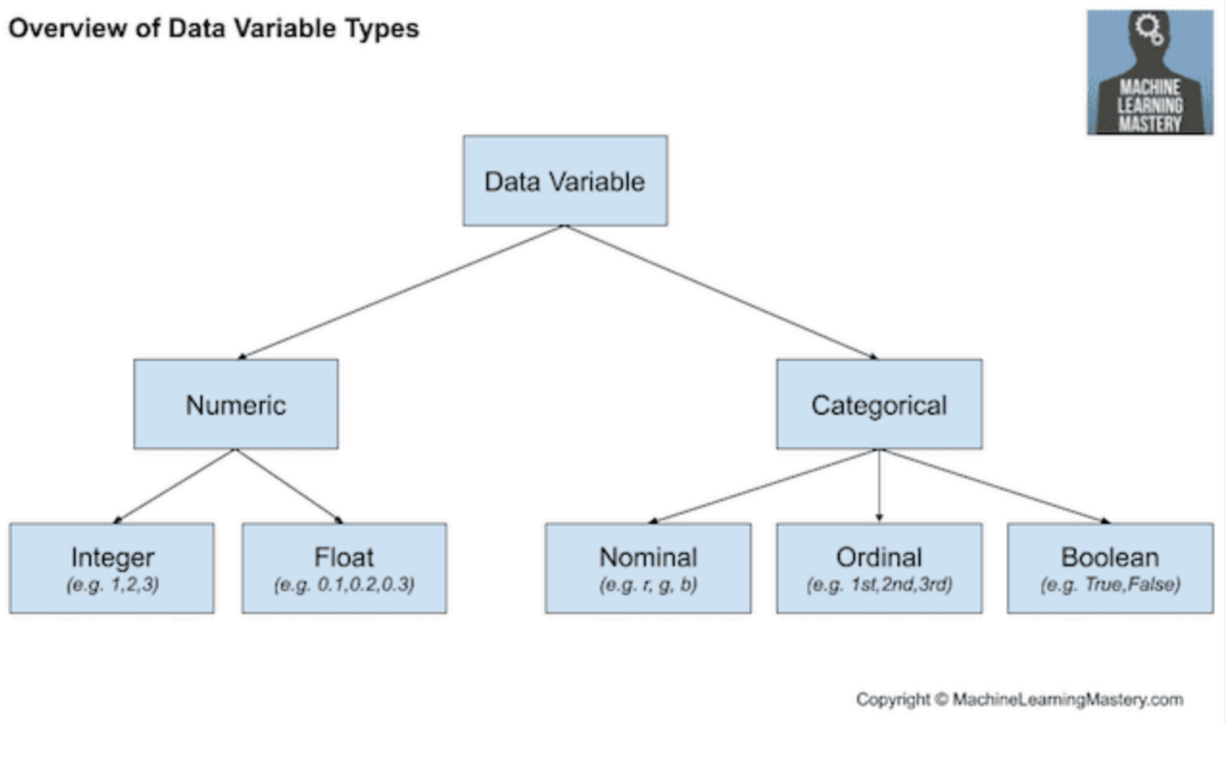
shuffles your data

https://scikit-learn.org/stable/modules/generated/sklearn.utils.shuffle.html

Feature Selection

Feature selection methods are intended to reduce the number of input variables to those that are believed to be most useful to a model in order to predict the target variable.





Let's have an overview of these techniques at the website:

https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

Feature Selection

Feature selection is a process where you automatically select those features in your data that contribute most to the prediction variable or output in which you are interested.

Benefits of performing feature selection before modeling your data are:

- **Reduces Overfitting**: Less redundant data means less opportunity to make decisions based on noise.
- **Improves Accuracy**: Less misleading data means modeling accuracy improves.
- **Reduces Training Time**: Less data means that algorithms train faster.

Let's try to automise this porcedure! Check the following website:

https://machinelearningmastery.com/feature-selection-machine-learning-python/

Feature Selection

Task: *Let's explore real datasets!!* You are given two csv files and you know one has events from Class 1 and the other one has events from Class 0.

(i) Read .csv files: vars_class1.csv and vars_class0.csv

(ii) Merge the two data frames and explore how the data look like.

(iii) Use the techniques described in: https://machinelearningmastery.com/feature-selection-machine-learning-python/ to conclude which variables are important for our classification task.

Which variables should we choose? Is there a conclusive result?

Feature Selection

Task: *Let's explore real datasets!!* You are given two csv files and you know one has events from Class 1 and the other one has events from Class 0.

(i) Read .csv files: vars_class1.csv and vars_class0.csv

(ii) Merge the two data frames and explore how the data look like.

(iii) Use the techniques described in: https://machinelearningmastery.com/feature-selection-machine-learning-python/ to conclude which variables are important for our classification task.

Which variables should we choose? Is there a conclusive result?

Answer :

Check example script: feature_extr_1.py

Feature Selection for Machine Learning Models using Iris dataset

Task: Let's have a look at the following example of selecting input variables and removing outliers from the iris dataset.

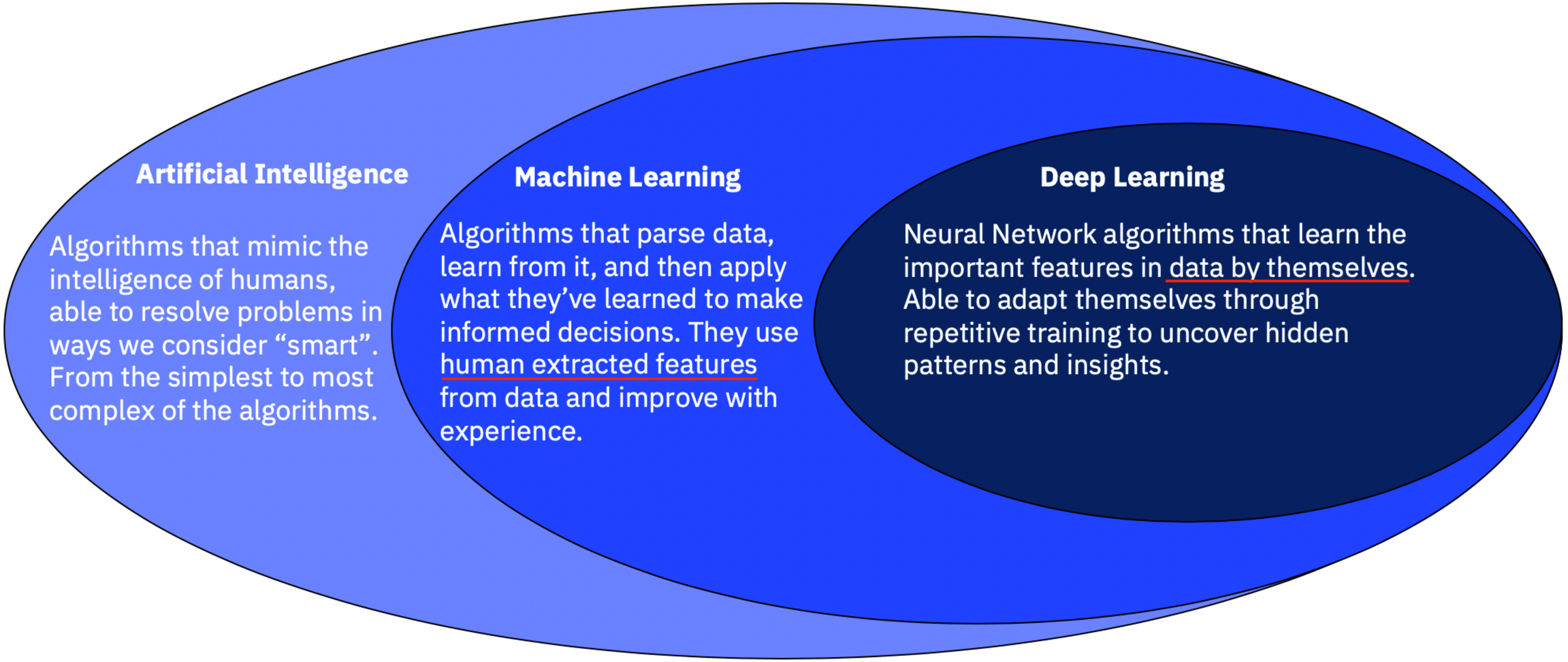
https://medium.com/@KousikRoyDataScientist/feature-selection-in-machine-learning-9e07a4664439

Note: For this step let's understand the code up to **Feature selection** section.

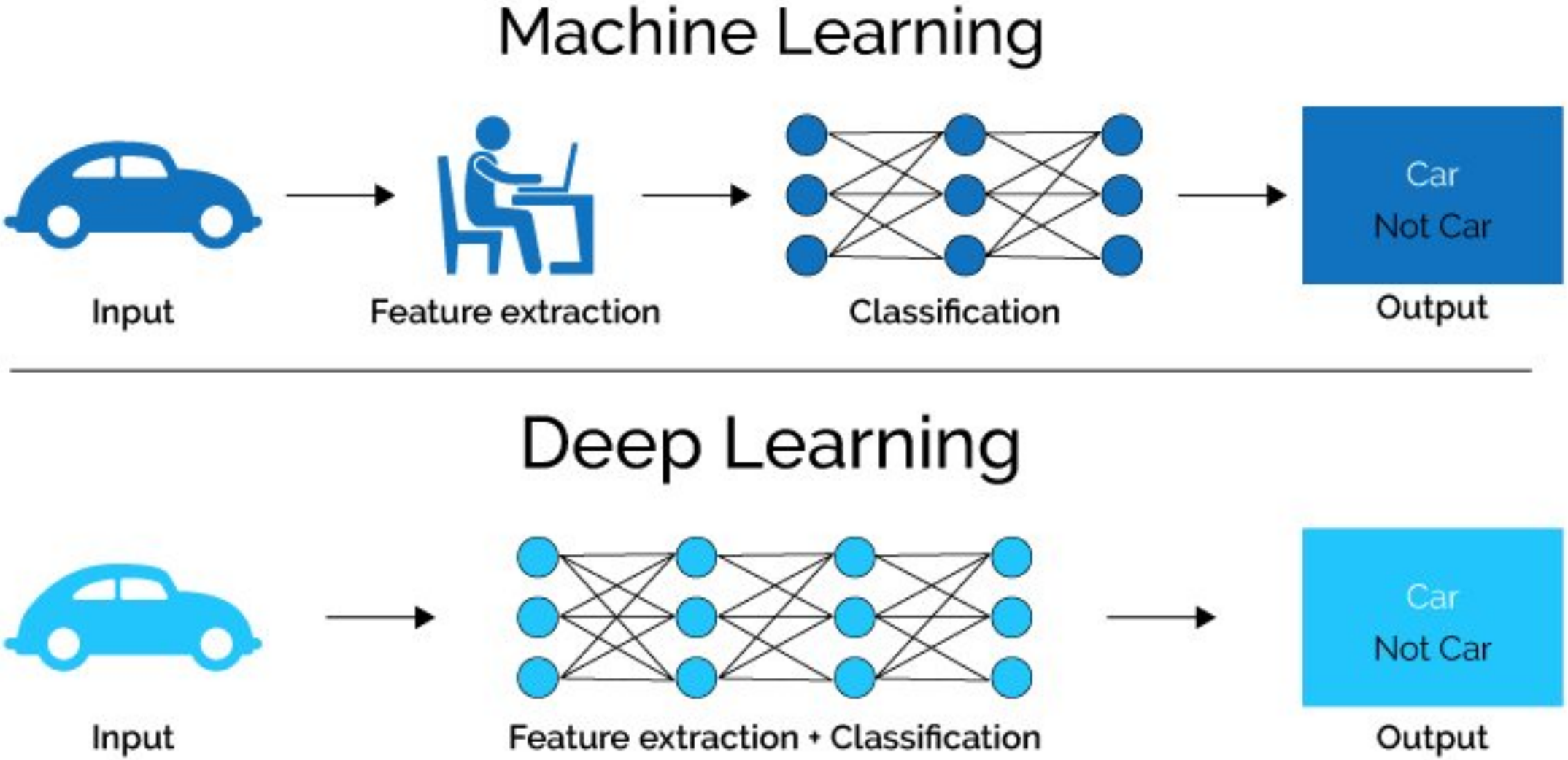
Reference for further reading on feature selection/engineering:
https://www.kaggle.com/code/ryanholbrook/what-is-feature-engineering

Machine Learning

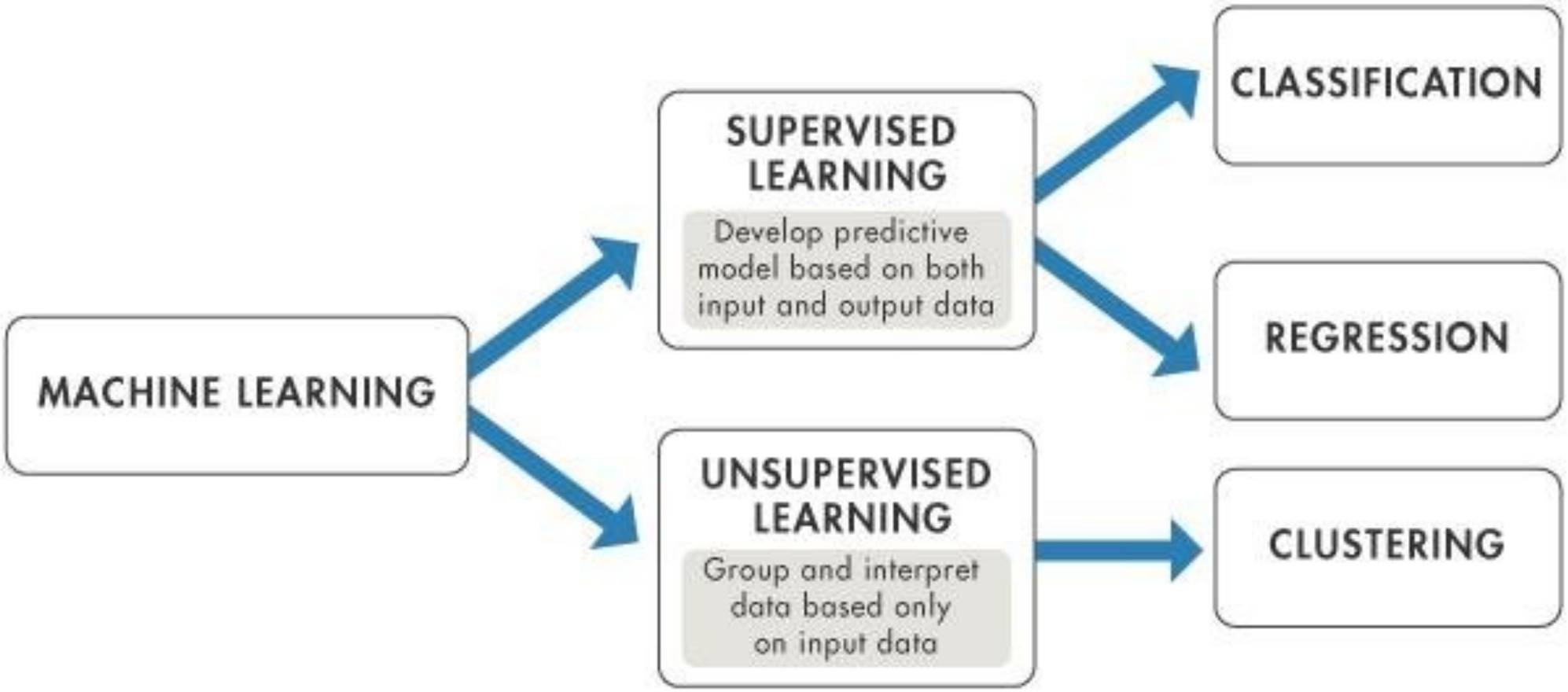
Machine & Deep Learning



Machine & Deep Learning

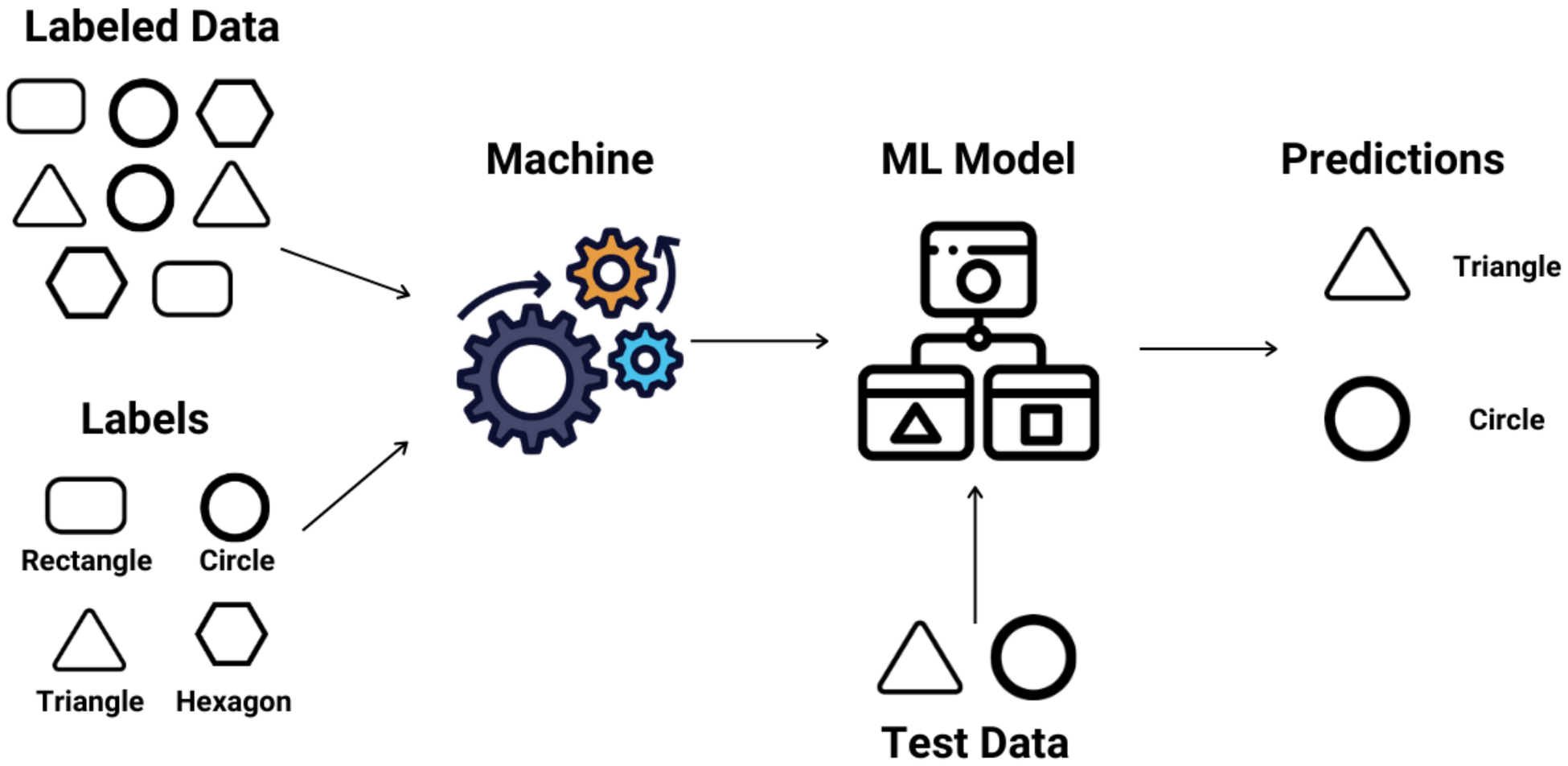


Intro to Machine Learning



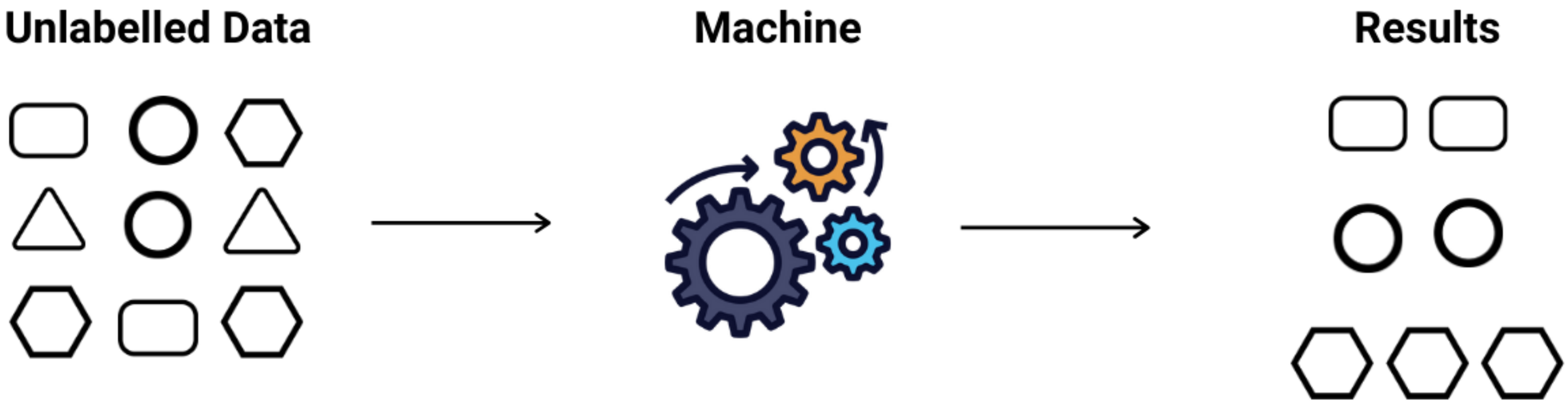
Intro to Machine Learning

Supervised Learning

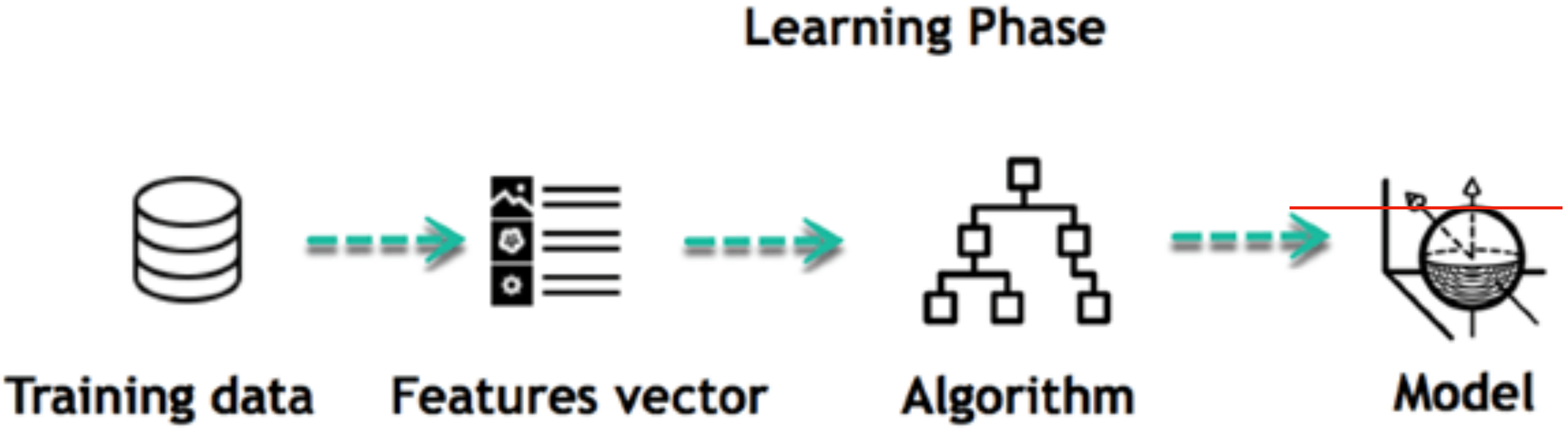


Intro to Machine Learning

Unsupervised Learning



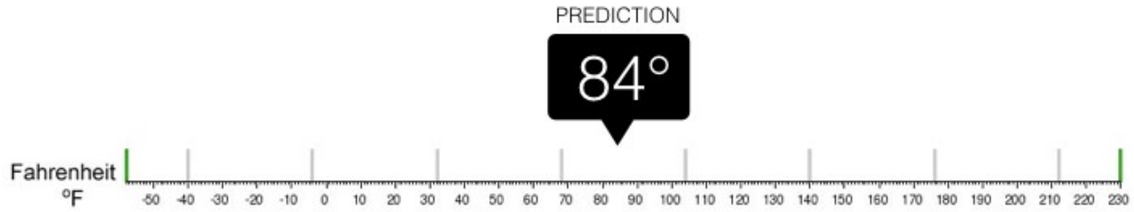
Intro to Machine Learning



Classification vs Regression

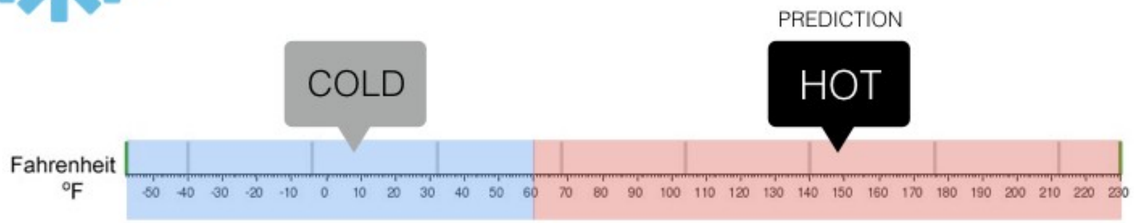
Regression

What is the temperature going to be tomorrow?



Classification

Will it be Cold or Hot tomorrow?



Classification example - iris dataset

Let's understand the first classification example: *the iris dataset*

Run the code: IRISclassification.py

Task:

(i) What happens if we change the number of input variables (e.g. 3 instead of 4)?

(ii) Change the different parameters of sklearn.tree.DecisionTreeClassifier and check the results.

Note: This model builds a single tree. Downside: its instability, which can be improved through ensemble techniques such as random forests, boosting, etc.

For a detailed description of this classification example with different algorithms check:
https://towardsdatascience.com/exploring-classifiers-with-python-scikit-learn-iris-dataset-2bcb490d2e1b

Classification example

Task:

(i) Use the same code for our dataset. Start by:

Reading and merging the data frames and shuffling the data (as in: feature_extr_1.py)

(ii) Then add the Classifier and check which parameters give the best accuracy.

(iii) What happens if we change the number of input variables (i.e. increase/ decrease #input variables)?

Answer :

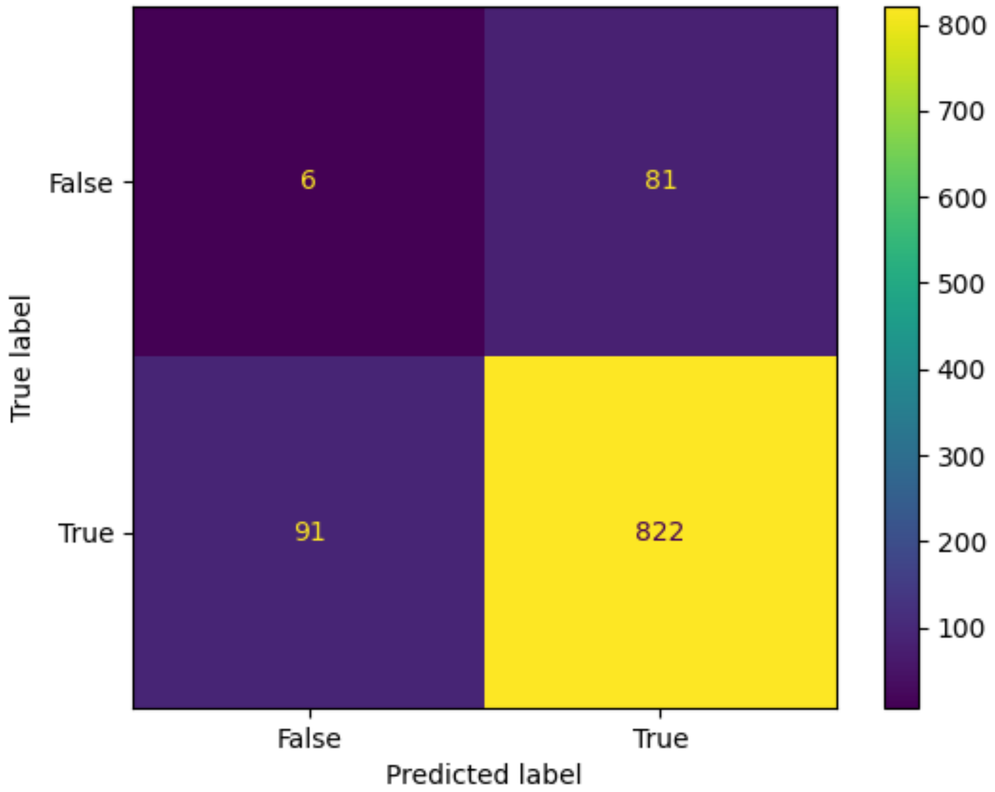
Check script: `DecisionTree_ClassExa.py`

Confusion Matrix

sklearn.metrics.confusion_matrix

sklearn.metrics.**confusion_matrix**(*y_true*, *y_pred*, *, *labels=None*, *sample_weight=None*, *normalize=None*)

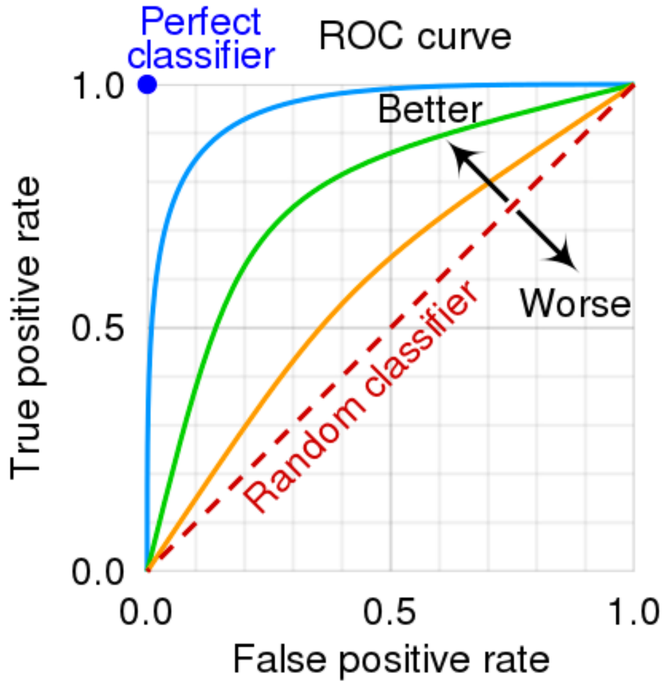
```
import matplotlib.pyplot as plt
import numpy
from sklearn import metrics

actual = numpy.random.binomial(1,.9,size = 1000)
predicted = numpy.random.binomial(1,.9,size = 1000)

confusion_matrix = metrics.confusion_matrix(actual, predicted)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix
confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
```



ROC curve

sklearn.metrics.roc_curve

sklearn.metrics.**roc_curve**(*y_true*, *y_score*, *, *pos_label=None*, *sample_weight=None*, *drop_intermediate=True*)



An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate
- False Positive Rate

True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (**FPR**) is defined as follows: $$FPR = \frac{FP}{FP + TN}$$

Area Under the Curve (AUS), which plots the whole are under the ROC (0.0, 1.0). AUC makes it easy to compare ROC curves. If the AUC of one ROC curve of the model is greater than the AUC of the ROC for another model, one can conclude that first model is better.

More for ROC curves at:
https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html#sphx-glr-auto-examples-model-selection-plot-roc-py

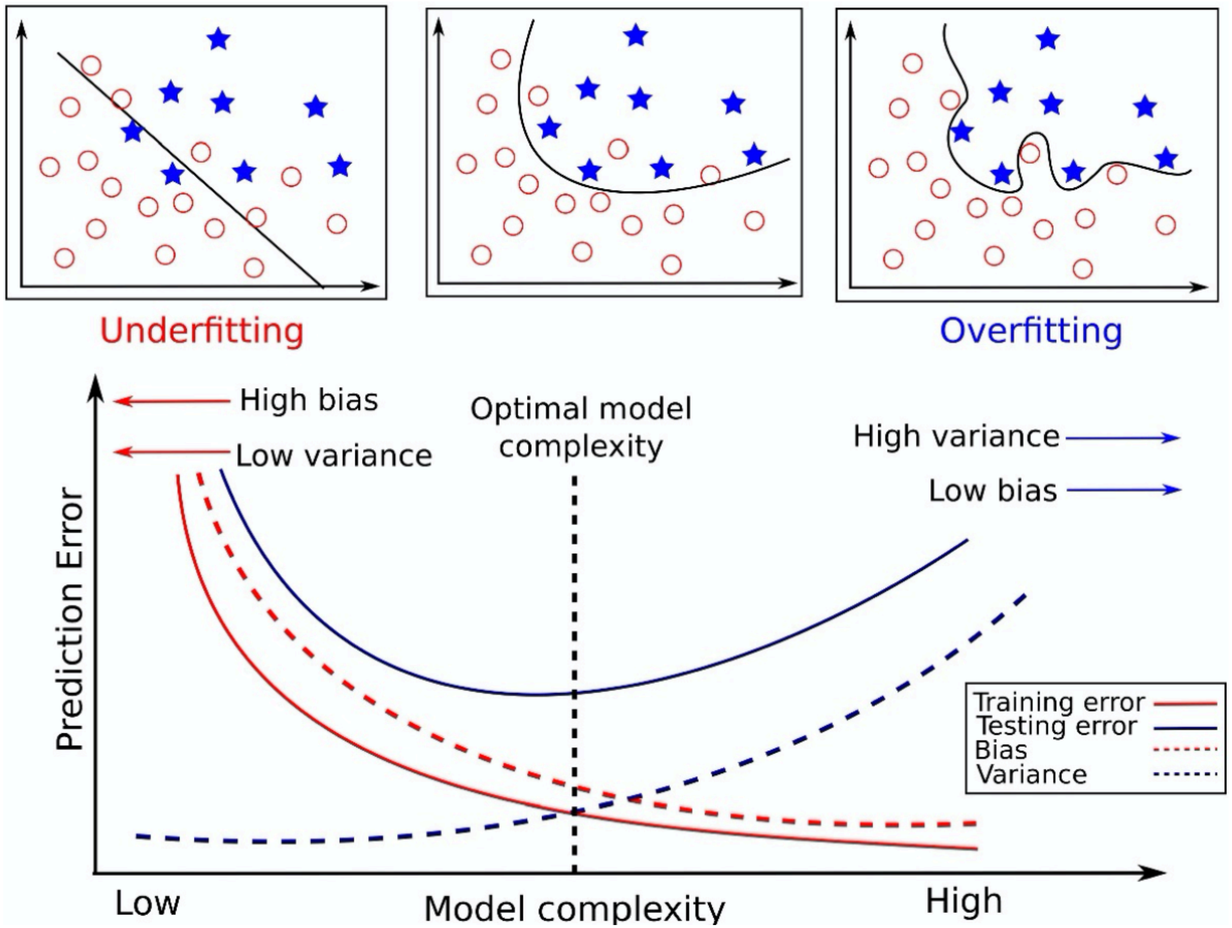
Example on ROC curves

Task: Understand and run the first example at: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

Bias - Variance Trade off

- The *bias* error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

- The *variance* is an error from sensitivity to small fluctuations in the training set. High variance may result from an algorithm modeling the random noise in the training data (overfitting).

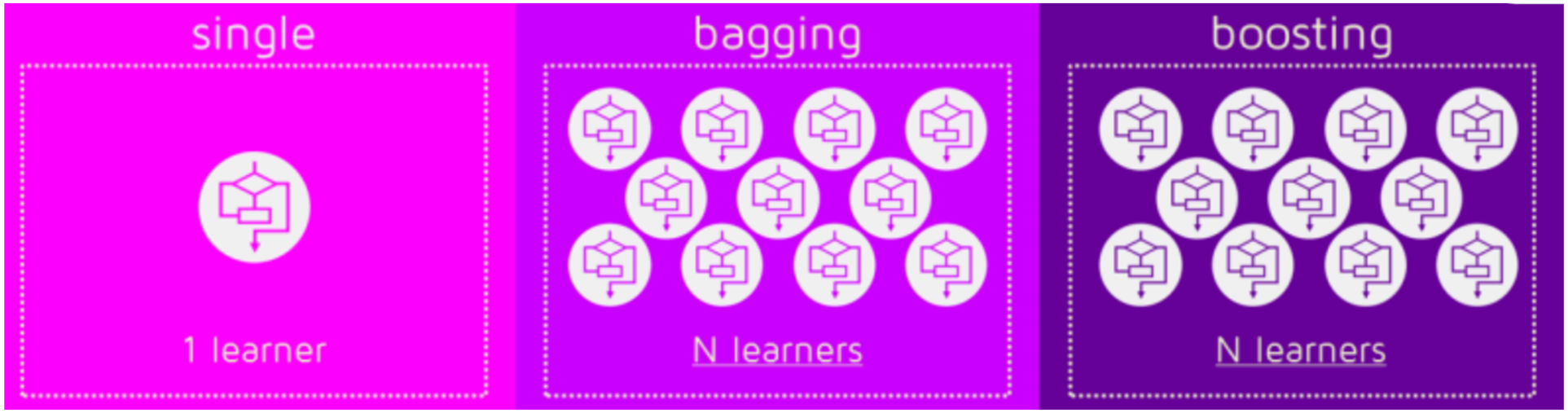


The **bias–variance problem** is the conflict in trying to simultaneously minimize these two sources of error that prevent supervised learning algorithms from generalizing beyond their training set.

Ensemble Methods

Ensemble learning is to learn a set of classifiers (experts) and to allow them to vote. This diversification in Machine Learning is achieved by a technique called **ensemble learning**. The idea here is to train multiple models, each with the objective to predict or classify a set of results.

- **Bagging** and **boosting** are two types of ensemble learning techniques. These two decrease the variance of single estimate as they combine several estimates from different models. So the result may be a model with higher stability.
- **Bagging helps to decrease the model's variance.**
- **Boosting helps to decrease the model's bias.**



https://www.kaggle.com/code/prashant111/bagging-vs-boosting

Classification example

Use an ensemble method **Random Forest**.

What is a Random Forest Classifier?

The Random Forest Classifier algorithm is an ensemble method in that it utilizes the Decision Tree Classifier algorithm but instead of creating just a single Decision Tree, multiple are created. In doing so, it takes advantage of a random sampling of the data as each tree learns from a random sample of the data points which are drawn without replacement. These trees also use only a subset of the features to be able to evaluate the importance of each feature overall. This randomness in generating individual trees minimizes the potential for over-fitting and improves the overall predictive accuracy of the model. This is because the final predictions are made by averaging the predictions of each individual tree, thus following the logic that the performance of the crowd is better than the performance of the individual.

Task:

(i) Use the following script and check the parameters of the Random forest that improve the accuracy:

`RandomForest_ClassExa.py`