See <https://github.com/skorsens/ai-foundations-ml>

# ML types

* Supervised learning.
  + Tains the model with labeled data – known inputs and outputs. E.g. houses prices vs. house features; or flower images vs flower types.
  + Regression problems – predicting a continuous value. House pricse is a regression problem that predicts the value of a house.
  + Classification problems – predict discrete values, tag data, classify data instance. E.g. given a flower image define its type.
* Unsupervised learning.
  + Does not use labeled data, instead it groups data based on the input only.
  + Clustering. E.g. group customers by their preferences.
* Reinforcement learning. Trial and error learning. Does not rely on a defined data set, but operates in a dynamic environment where the model learns from the experience. Used in robotics, board games

# ML Life Cycle

## Problem formation and understanding.

* Define a problem to solve and if ML can provide a solution.
* Define the inputs and outputs and prediction error rates.

When to use ML:

* Predict an outcome
* Uncover trends and patterns in data
* Have rules and steps that can’t be coded
* The dataset is too large to process by a human.

Frame the problem to solve

* Define the questions for the model.
* Define the inputs (features) and outputs. The outputs can define the learning algorithm.
* Define the success metrics (e.g. how faster a house shall be sold).
* Define the costs for running and maintaining the model.

## Data collection and preparation.

* Source the raw data required.
* Prepare the data: label the data, remove irrelevant features and outliers; transform the data; input missing values.
* Split the data into training (80%), validation (10%), testing (10%).
* Select the ML algorithm. Often the required output defines the learning algorithm.

## Model training and testing.

* Select if to train the model from scratch or if to use a pretrained model (transfer learning – adding a new data on top of a pre-trained model).
  + [ModelZoo](https://modelzoo.co/) is the pre-trained model catalog.
  + [AWS SageMaker](https://aws.amazon.com/sagemaker-ai/?trk=af448c55-a335-4b68-909c-3589fef8d1a7&sc_channel=ps&ef_id=Cj0KCQjwmqPDBhCAARIsADorxIY2FiCdLwA1Fx39Z_xtgqUwESU3YdqdLZzan39i8x--Ctp8aoCp2UAaAj2GEALw_wcB:G:s&s_kwcid=AL!4422!3!724139679532!p!!g!!amazon%20sagemaker%20ai!19574556908!168879877422&gad_campaignid=19574556908&gbraid=0AAAAADjHtp9PA4wy1DgjnimsBImyuWPFm&gclid=Cj0KCQjwmqPDBhCAARIsADorxIY2FiCdLwA1Fx39Z_xtgqUwESU3YdqdLZzan39i8x--Ctp8aoCp2UAaAj2GEALw_wcB).
  + <https://huggingface.co/>.
* Try the learning algorithm on the data trying to produce a model.

## Model deployment and maintenance.

* Deploy the model. Retrain the model and monitor its performance.

# Python libraries for ML

* Pandas. Data structures, matrices, data analysis, reading data from files.
* Numpy. Numerical data processing in arrays and matrices.
* Matplotlib, Seaborn. Visualization.
* Scikit-learn, TensorFlow, MXNext, PyTorch, Keras. Learning algorithms.

# Preparing data for ML

## Obtaining data

Data sources: internal; from clients; open-source; public datastore; commercial.

### E.g. problem. Use supervised learning linear regression to predict the house prices

Features: location (long, lat), house age, total rooms, bedrooms, population, house value etc…

House value is the target output.

# Training an ML model

ML training makes multiple iterations called epochs.

Loss functions are used to measure the accuracy of the trained model (the diff between the predicted value and the actual value) in order to optimize the training:

Loss function evaluation metrics: Mean Square Error (MSE), Accuracy, F1 score, AUC, R^2, etc…

Metrics values are used to decide if it is needed to change the hyper parameters of the training algorithm and start the training again.

Training algorithms examples.

* Linear regression. Solves regression problems; predicts numeric values.
* Linear equation. Establishes relationships between dependent and independent variables and fits them to a regression line.
* Logistic regression. Used to resolve classification problems by predicting the probability between 0 and 1 of a binary value (true/false, yes/no, 0/1) of a dependent variable.
* Decision trees. Used for classification and regression problems.
* Random forest. Set of decision trees.

## Use Logistic regression learning algorithm

Predict if a stop will lead to an arrest using AWS SageMaker XGBoost algorithm.

## Linear regression

Predict house prices using linear regression algorithms Scikit-learn, RandomForestRegressor, XGBoost.

# Evaluating model performance

Metrics are the indicators of model performance.

## Classification metrics

### Accuracy

The fraction of the correct predictions vs the total predictions.

Accuracy = N(correct predictions) / N(total predictions)

Accuracy(Binary Classification) = (TP + TN) / (TP + TN + FP + FN)

T = True, F = False, P = Positive, N = Negative

Accuracy is not a good way to measure the performance of imbalanced datasets.

### 5.1.2. Precision

The number of correct positive predictions vs total number of positive predictions. Used to minimize FP, e.g. minimize the number of e-mail messages that are falsely marked as spam.

Precision = TP / (TP + FP)

### 5.1.3. Recall

Measures the sensitivity of a model. Used to maximize TP, when false positives are ok, but false negatives are not, e.g. checking if a credit card transaction is fraud shall not be FN.

Recall = TP / (TP + FN)

### 5.1.4. F1 Score

Combines precision and recall. Works well for imbalanced data.

F1 = 2\*[Precision \* Recall] / [Precision + Recall]

### 5.1.5. AUC (Area Under the Curve)

Measure accuracy and visualize how the predictions are ranked across true positives and false positives.

### 5.1.6. Additional classification metrics

TBD

## Confusion matrix

Table that shows a summary of prediction results for a classification model. Evaluates the accuracy of a model.

Counters of incorrect and correct predictions per classification class.

For binary classification problems, it is 2x2 matrix. Can be extended to problems with more classes by adding additional rows and columns.

|  |  |
| --- | --- |
| TP | FP |
| FN | TN |

Can be created using

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay, classification\_report

Can be used to calculate the classification metrics.

## Regression metrics.

R Square (R^2), Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE)

### R^2

R^2 calculates the diffs between the actual and predicted value using the residuals - the distances between actual and predicted values. R^2 values are between 0 and 1, expressed as %; values closer to 1 indicate a better model fit. R^2 is a relative measure

### MSE and RMSE

Absolute number of how much the predicted results deviate from the actual ones.

RMSE = sqrt(MSE), to make it easier to interpret.

Measures if there are big differences between the predicted and actual numbers.

Ranges between 0 and infinity, with lower values indicating better model.

### MAE

Sum of all differences between actual and predicted values / total number of predictions. Measures the accuracy of the model on average.

Ranges between 0 and infinity, with lower values indicating better model.

## Determining Feature Importance.

Finding the most important features for predictions.

Improves predictions, reduces training time and cost, clarifies the predictions (model explainability – helps with the confidence in the model).

How to determine the importance of a feature?

There are algorithms with embedded feature importance, e.g. tree-based models, random forest, gradient boost. T.i. there is a function that grades the features by their importance.

## Combating Bias

Providing limited data can create biased model, e.g. if a model is trained only on roses and tulips, it will classify all other flows either as one of those 2.

Bias can be due to

* Data.

Limited data or data with outliers can create a bias.

* Algorithm

It is needed to try and measure the performance of multiple algorithms before selecting one.

* Model

The model in production can drift over time. Including only the most important features mitigates this.

# Optimizing ML Pipeline

Automating the process of data preparation, learning algorithm selection, tune hyperparameters.

Sklearn.pipeline provides Pipeline object.