AREIX

Everyone Worthy, Everyone Wealthy

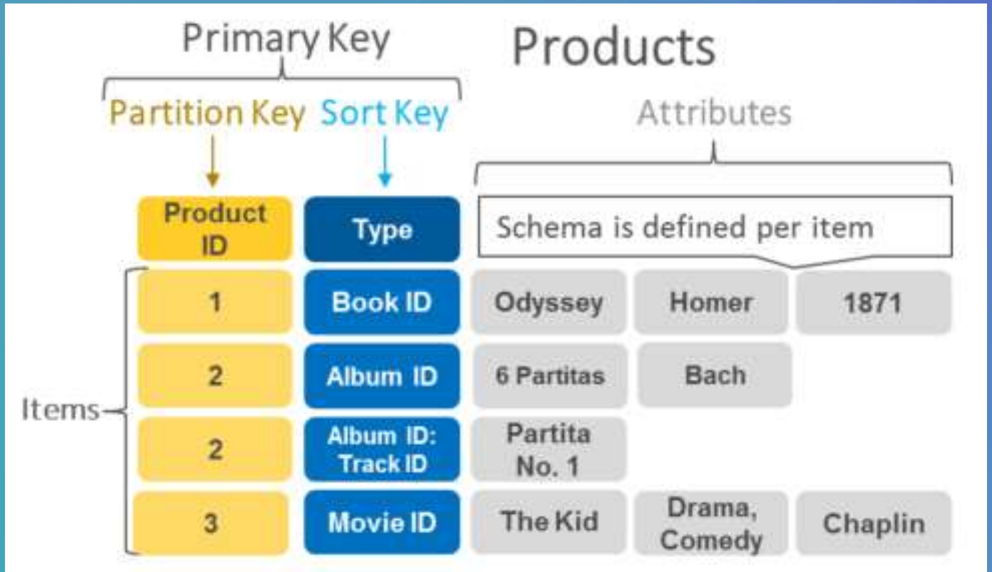
Modelling in Practice

Charon Guo
HEAD OF APPLICATION & BACKEND DEVELOPMENT

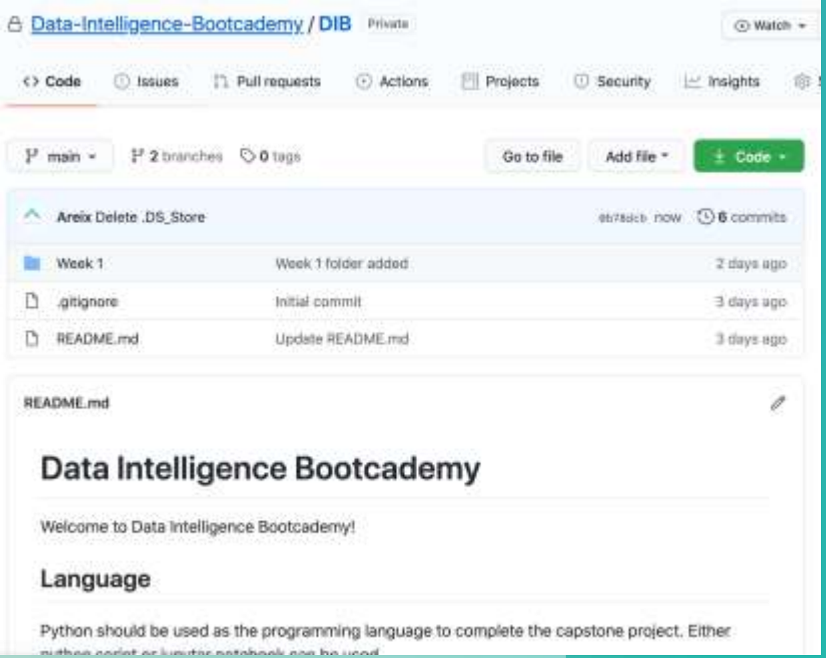
Tools

AREIX

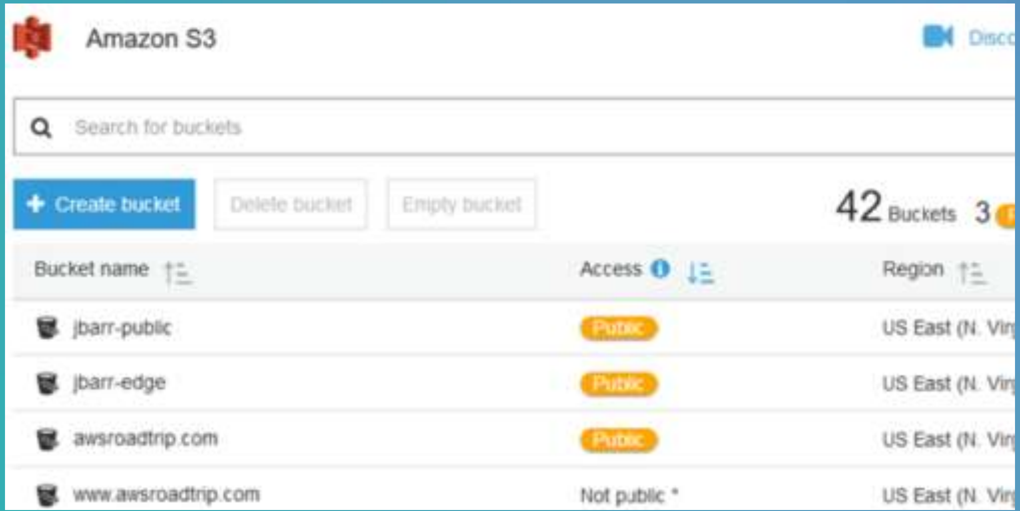
AWS DynamoDB



Github



AWS S3



What is Machine Learning?

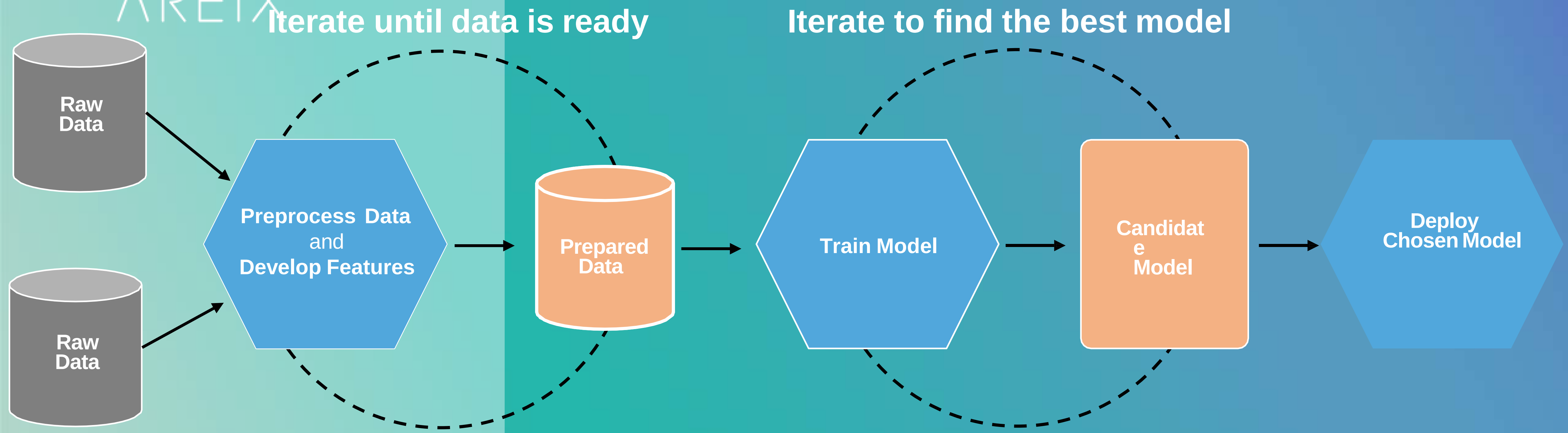
AREIX

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

--- Mitchell, T. (1997). Machine Learning, McGraw Hill

Machine learning is a technique of data science that helps computers **learn from existing data** in order to **forecast future behaviors, outcomes, and trends**.

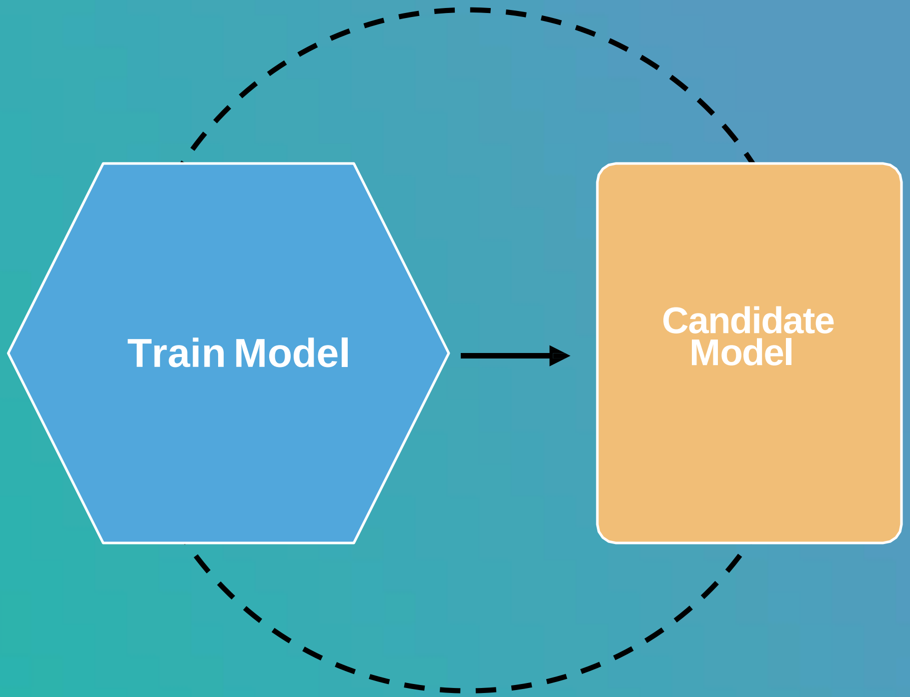
The process of Machine Learning



Amazon, http://docs.aws.amazon.com/machine-learning/latest/dg/training-the-model.html

Modelling

Iterate to find the best model

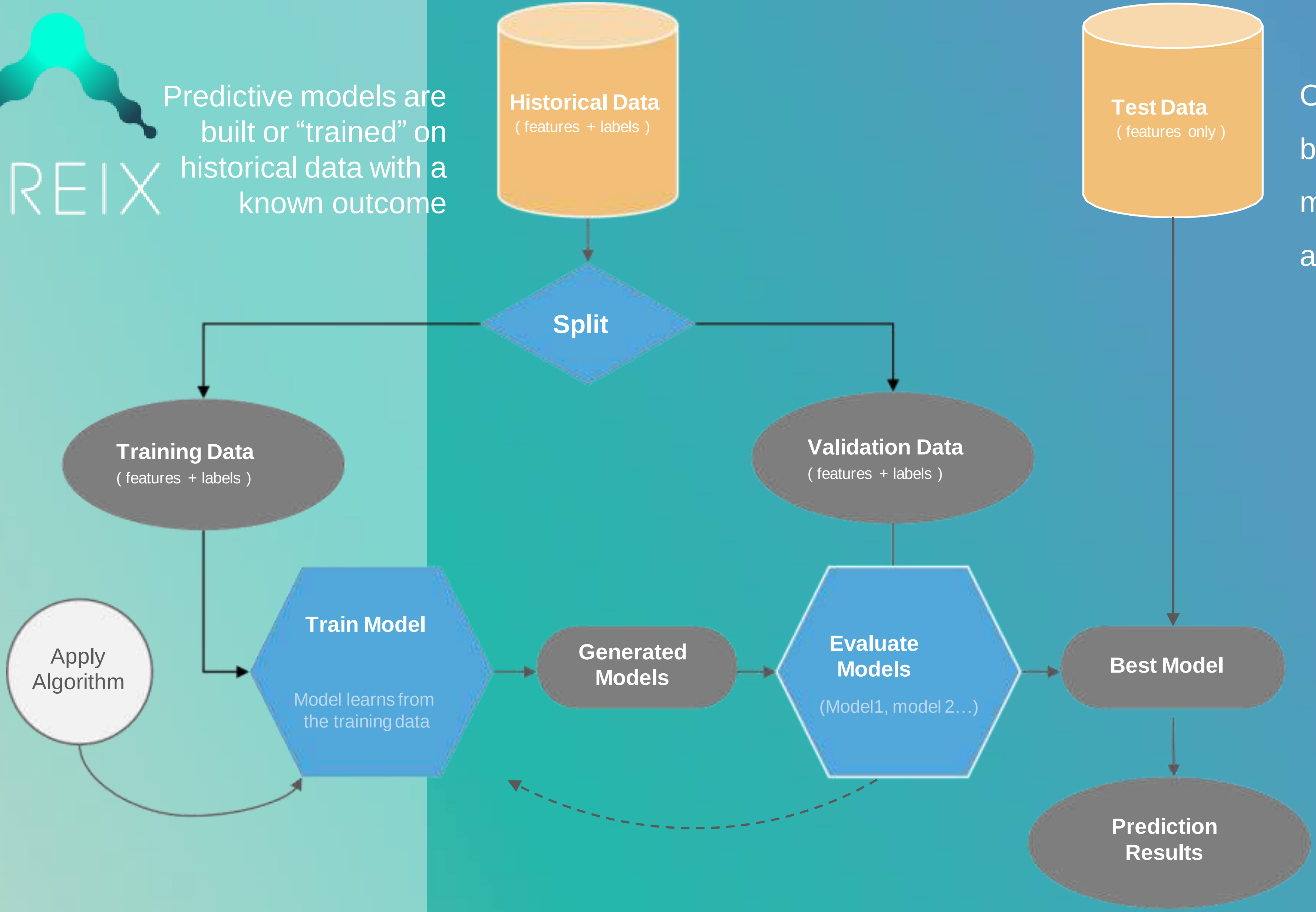


Train model, Evaluate Model & Optimization

- Find the model that answers the question most accurately by comparing their success metrics
- Determine if your model is suitable for production

How to develope model?

Predictive models are built or “trained” on historical data with a known outcome



Once the model has been built, it’s applied onto new, more recent data which has an unknown outcome

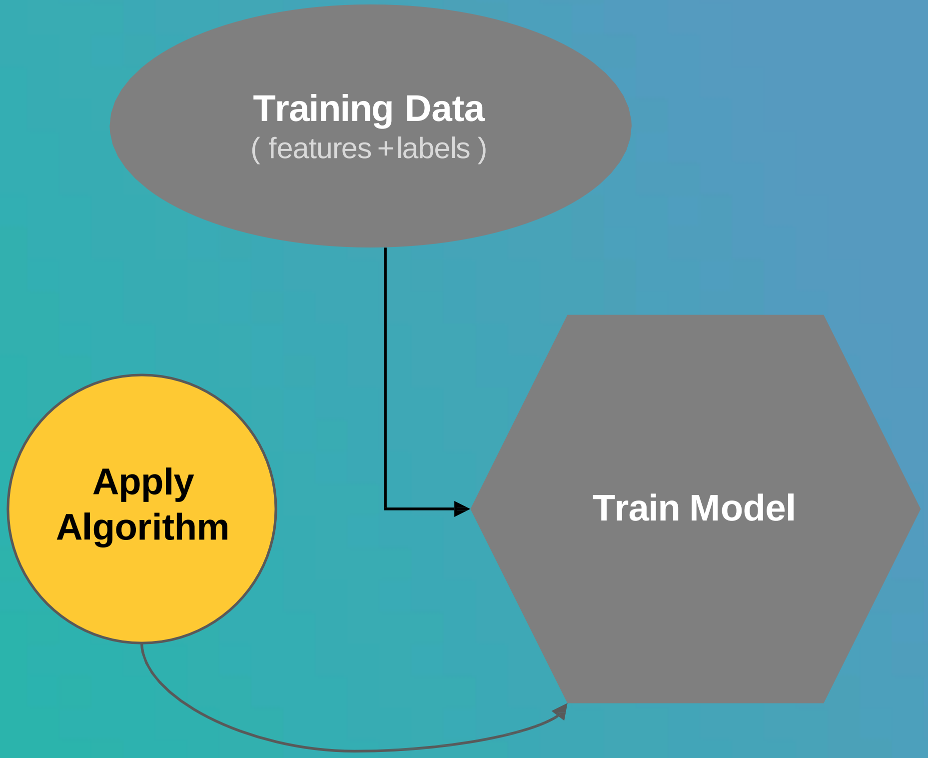
SAP: https://open.sap.com/courses/ds1

Train the model

Train a model using algorithms and training data

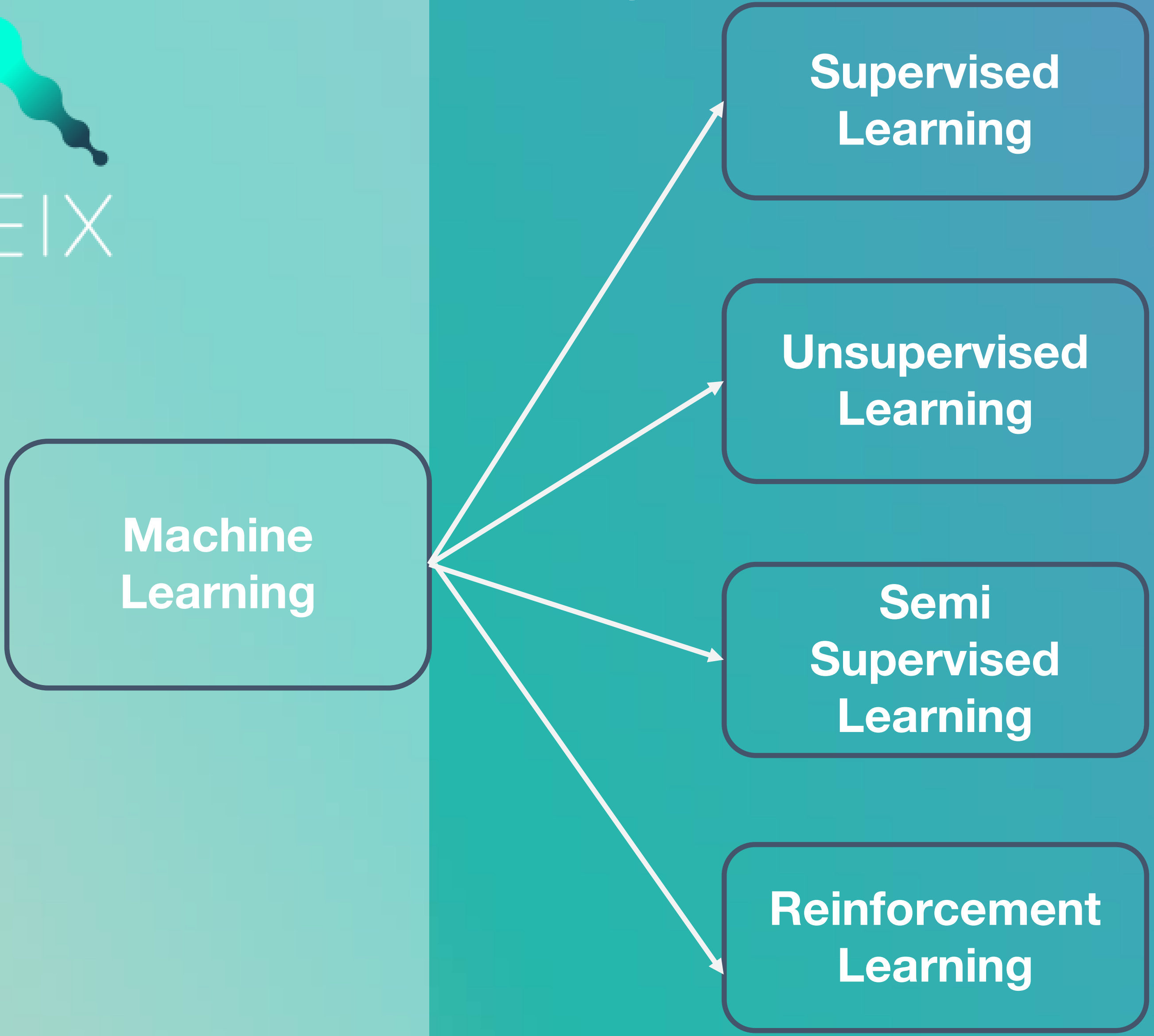
Algorithm is a set of rules used to solve problems through data processing, math, or automated reasoning.

Machine learning algorithms use computational methods to “learn” information directly from data without relying on a predetermined equation as a model

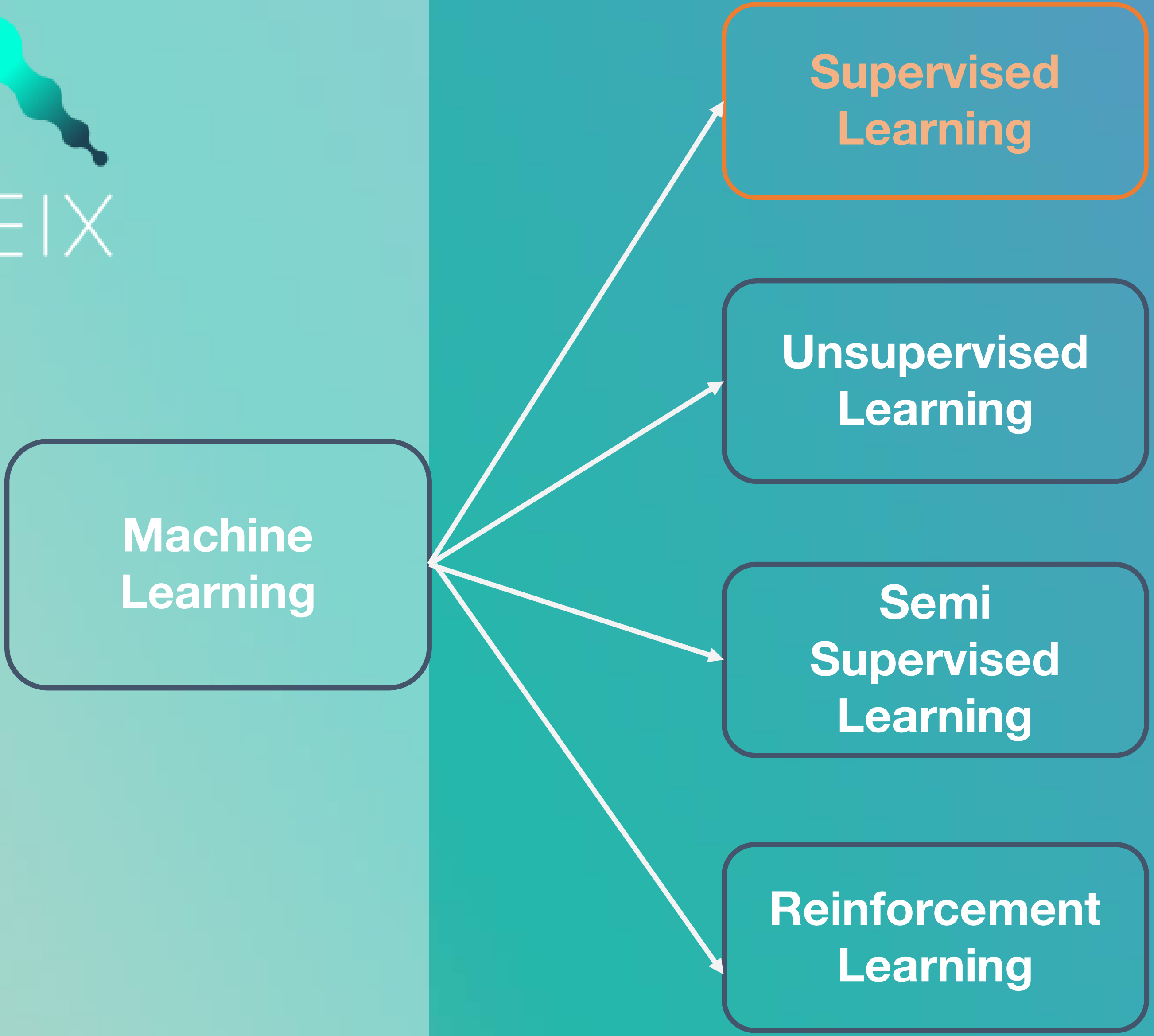


The algorithm will learn from the training data patterns that map the variables(features) to the targets(labels), and it will output a model that captures these relationships

The categories of machine learning algorithms



The categories of machine learning algorithms

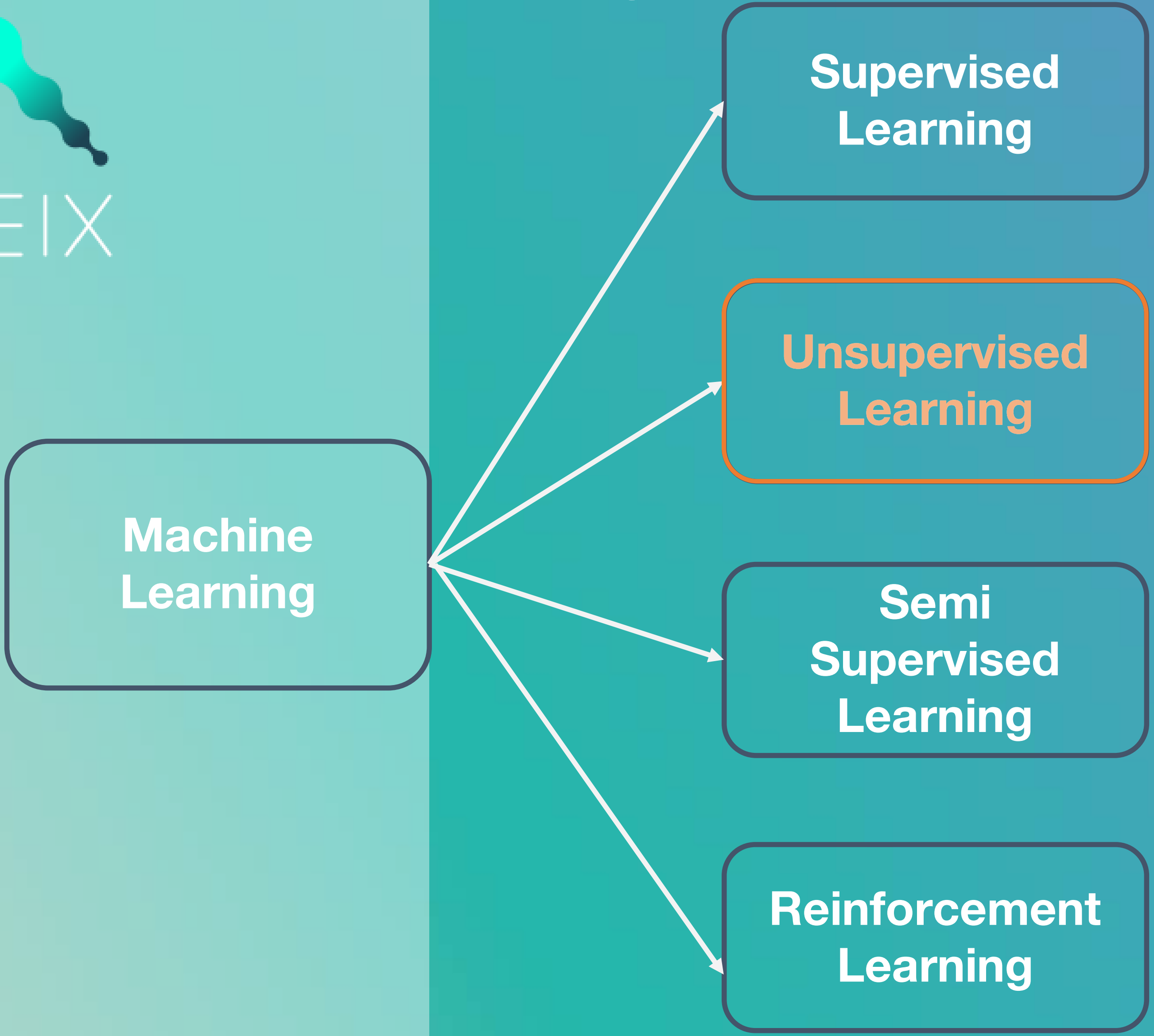


Train the model with labeled data

A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions

(Most machine learning is supervised)

The categories of machine learning algorithms

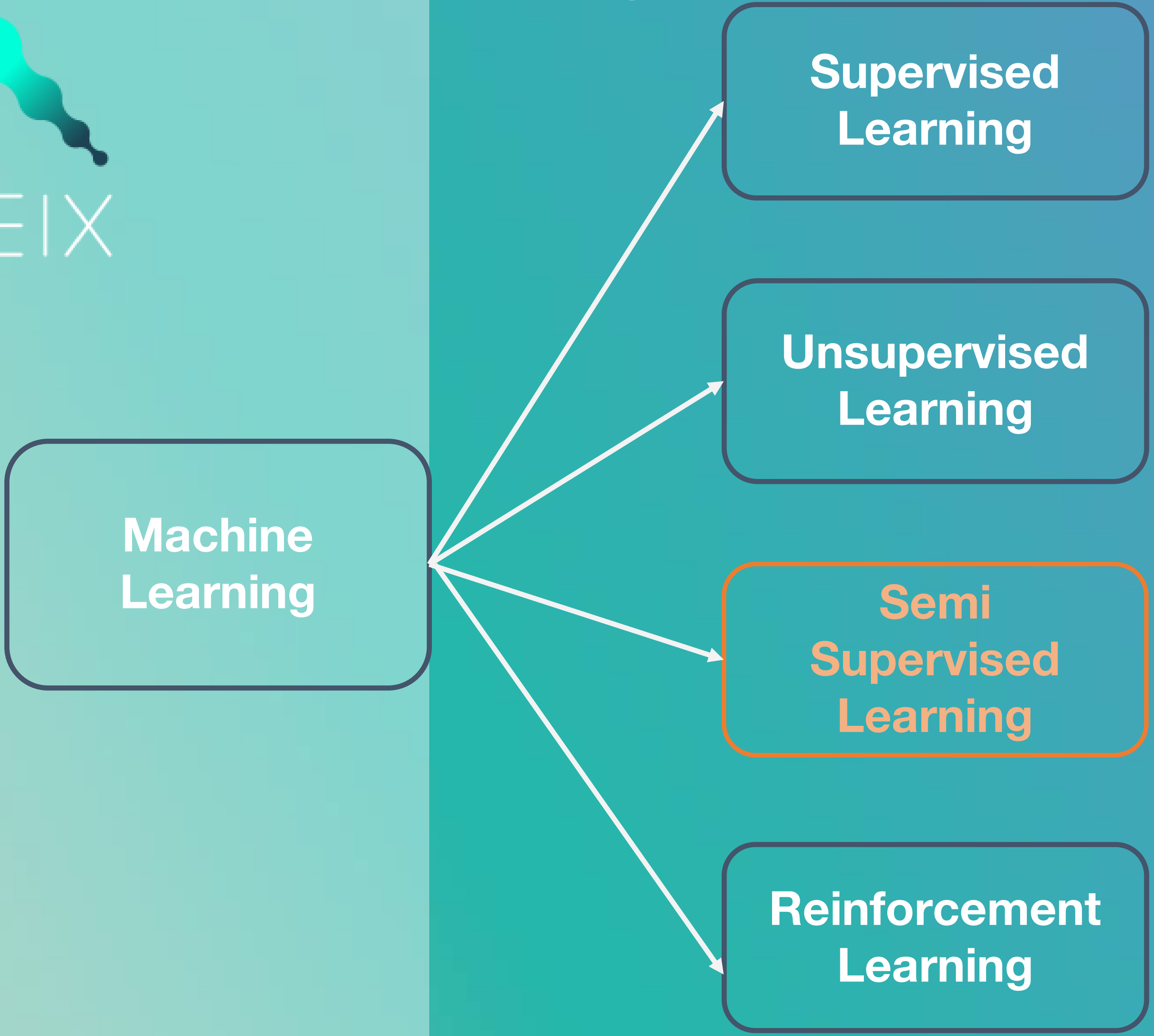


Train the model with unlabeled data

The goal is to find patterns or intrinsic structures in the data. It's also a good way to simplify data somehow (reduce dimensions, remove unnecessary variables or detect anomalies).

Clustering is the most common unsupervised learning technique

The categories of machine learning algorithms

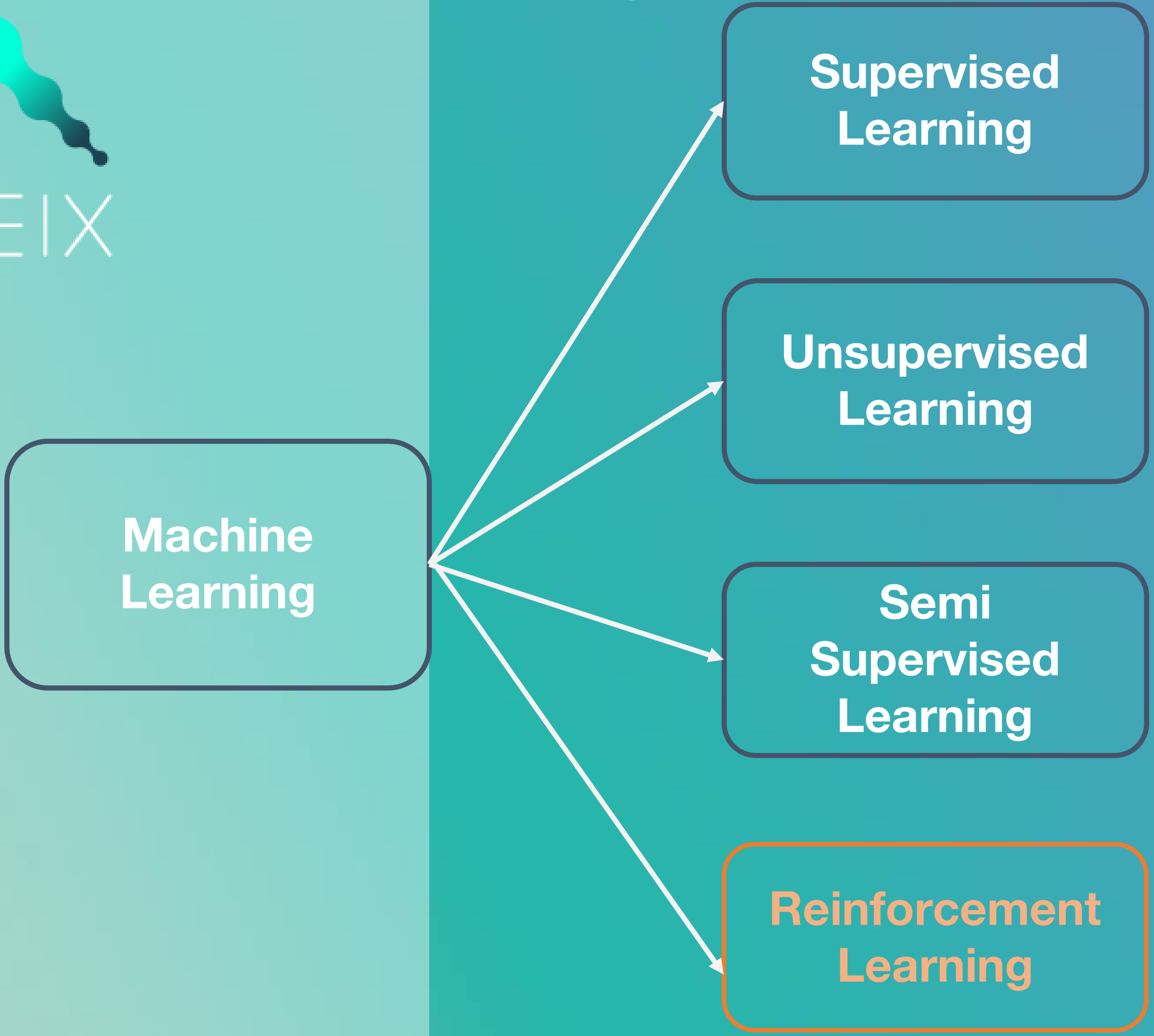


Train the model with partially-labeled data

Typically a small amount of labeled data with a large amount of unlabeled data during the training phase.

The trained models that result from this training set can be highly accurate and less expensive to train compared to using all labeled data

The categories of machine learning algorithms



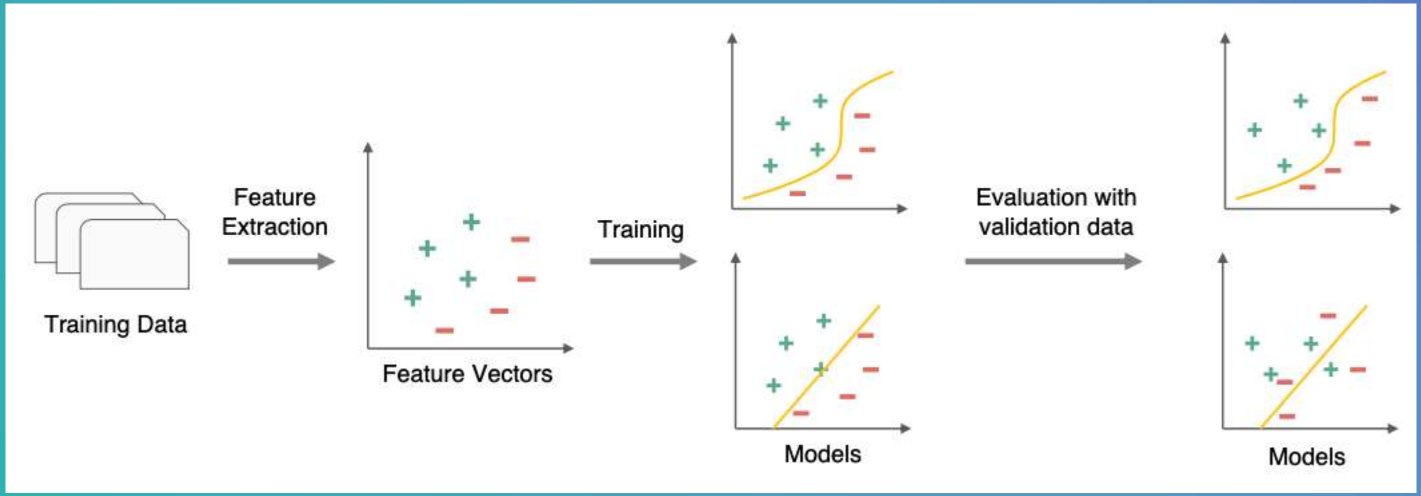
Train the model from a series of 'reward function'

The model is learned from a series of actions by maximizing a “reward function”. The reward function can either be maximized by penalizing “bad actions” and/or rewarding “good actions”.

Supervised Learning tasks

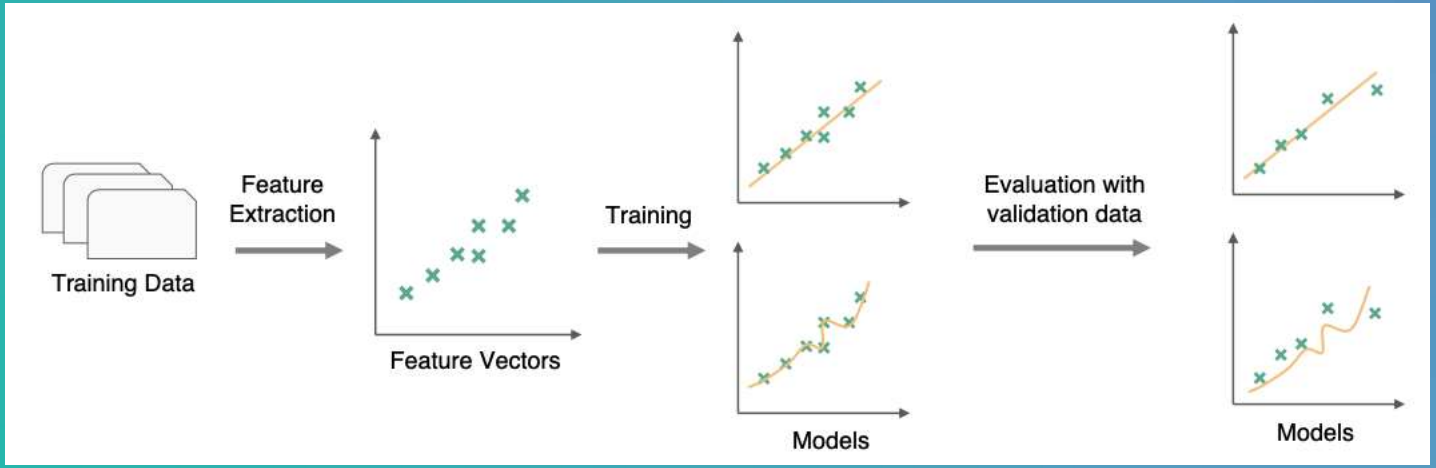
Classification (Predict the category)

Identifing to which cataegoty the objecct belongs to



Regression (Predict the value)

Predicting a cointinuous-valued attribute associated with an object



Example for Regression problem

Example: Predict house’s price using linear regression

Suppose we have a dataset giving the living areas and prices of n houses from House Sales in HongKong. Given data like this, we can learn to predict the prices of other houses in Hong Kong

latitude	longtitude	bedrooms	Living Area (Feet^2)	Price ($)
-32.432	64.342	2	1180	221,900
34.543	43.532	3	2570	538,000
54.34	54.53	2	770	180,000
-12.432	324.53	3	1960	604,000
-43.432	5.345	2	1680	510,000
54.543	23.423	5	5420	1,225,000
56.32	53.525	4	1715	257,500
-93.54	98.34	1	1060	291,850
45.65	54.89	2	1780	229,500
76.63	654.54	3	1890	323,000
25.654	543.63	1	3560	662,500
75.53	43.22	2	1160	468,000
-54.00	43.3543	4	1430	310,000
543.2	65.654	3	1370	400,000
54.6	63.435	4	1810	530,000
...	...	...	...	...

Dataset

How much for this house ?

living area = 4876 feet^2
bedrooms = 4
latitude = -34.244
logtitude = 31.42

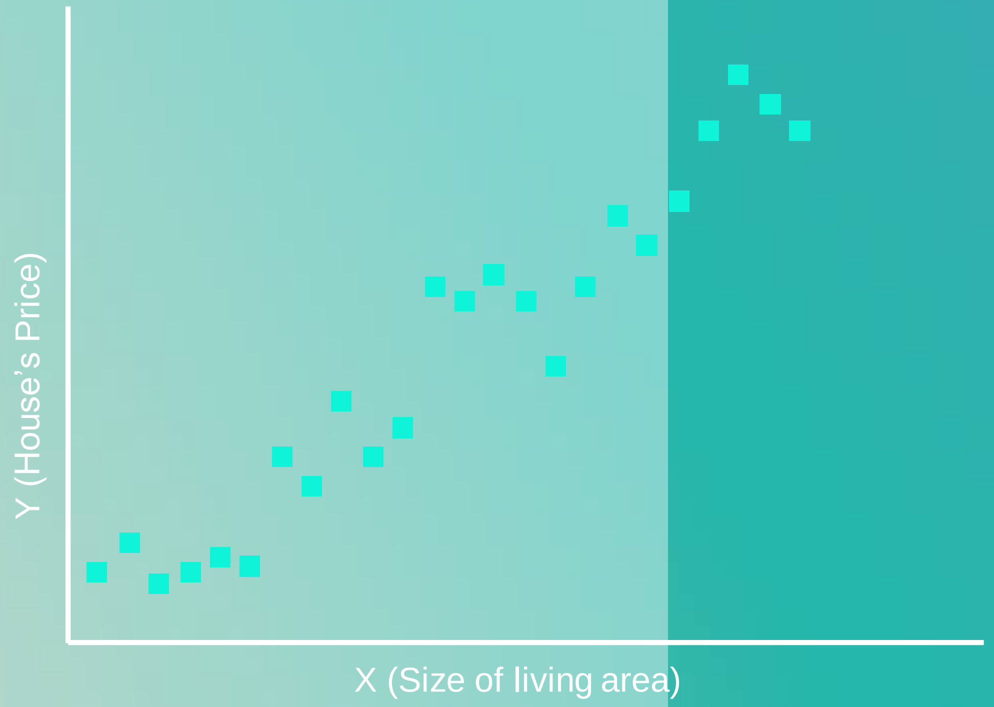
A “mathy” approach where we weight each feature by how important it is then use a weighted sum to estimate housing prices:

$$\text{Housing price} = \text{w1*bedrooms} + \text{w2*size} + \text{w3*latitude} + \text{w4*longtitude} + \text{min\_price}$$

Example for Regression problem

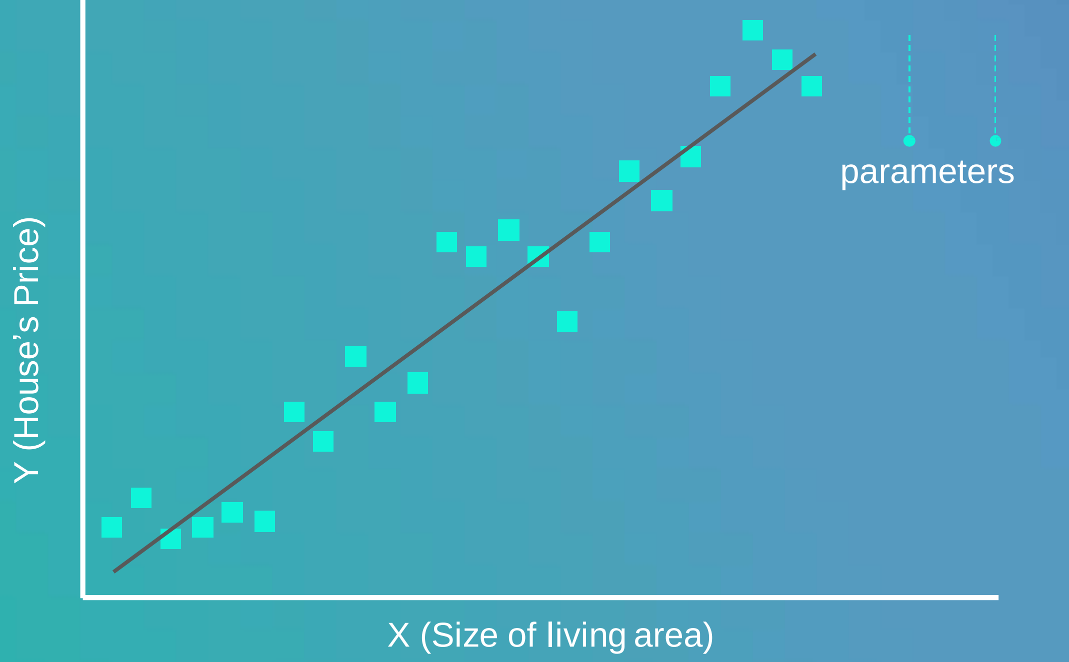
Use linear regression algorithm to approximate the relationship between x and y

Take linear regression as example, the algorithm is trying to find a best-fit line to represent the relationship between the input feature x and target y



Regression line (model) can be presented as :

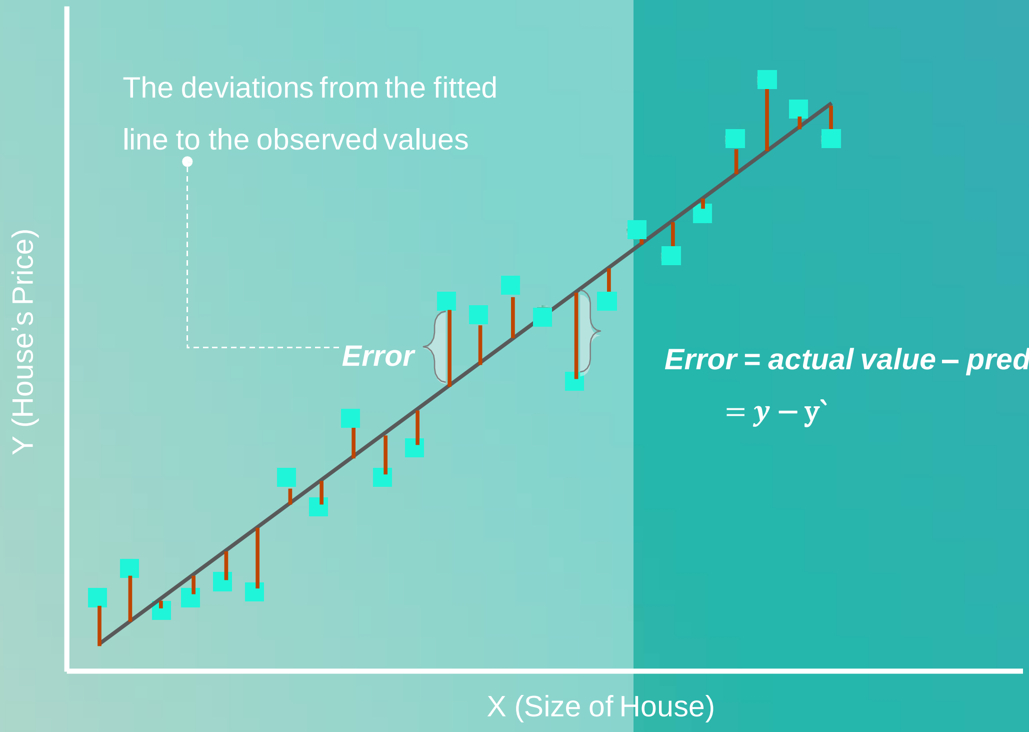
$$y` = \theta_0 + \theta_1 x$$



Train the model to minimize the loss/error

The deviations indicate how bad the model's prediction was on the training examples

Loss (i.e. error) is a number indicating how bad the model's prediction was on a single example. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater



Error = actual value – predicted value

$= y - y`$

Mean square error (MSE) is a commonly-used function to measure how large the loss is. It's called as **Loss function** or **Cost function**.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - y`)^2$$

y` is the prediction
y is the actual value

Mean square error (MSE) is the average squared loss per example over the whole dataset.

The smaller the Mean square error, the better the fit of the line to the data.

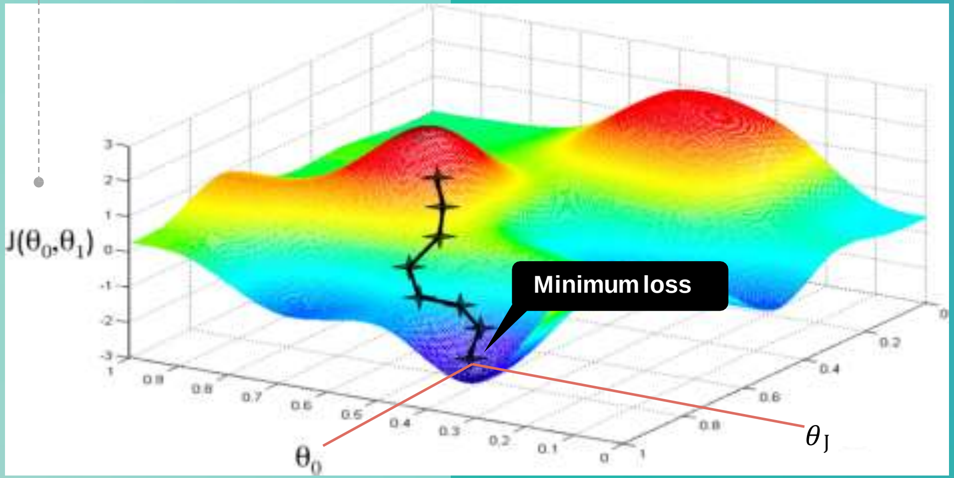
How does the model find the “best” parameters ?

Gradient Descent is one of the most common algorithms to find the good parameters

A Machine Learning model is trained by starting with an initial guess for the parameters (e.g. weights and bias in neural network) and iteratively adjusting those guesses until learning parameters with the lowest possible loss

Loss function/cost function: (error)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - y`)^2$$



Parameter 1

Parameter 2

With these 2 specific parameter value, the loss (i.e. MSE) is almost smallest.

Usually iterate until overall loss stops changing or at least changes extremely slowly.

When that happens, we say that the model has **converged**.

$$w := w - \alpha\frac{\delta J(w,b)}{\delta w}$$
$$b := b - \alpha\frac{\delta J(w,b)}{\delta b}$$

Other optimization algo:

Gradient Descent (with momentum)

Mini-batch Gradient Descent

Stochastic Gradient Descent

Adam

Adagrad

RMSprop

....

Example for classification problem

AREIX

Classification Algorithm

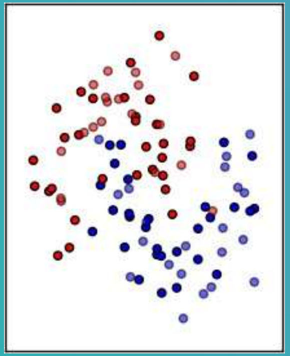
Identify what category new information belongs in

Predict Categories

Predict between two categories	Predict between several categories
Binary-Class Classification	**Multi-Class Classification**
It answers simple two-choice questions, like Yes-or-no, true-or-false	It answers complex questions with multiple possible answers
Example: Use CT scan to identify whether has diabetes (True, False)	Example: Use CT scan to identify which type of diabetes (Type1, Type2, Type3..)

Example: how to classify the data points ?

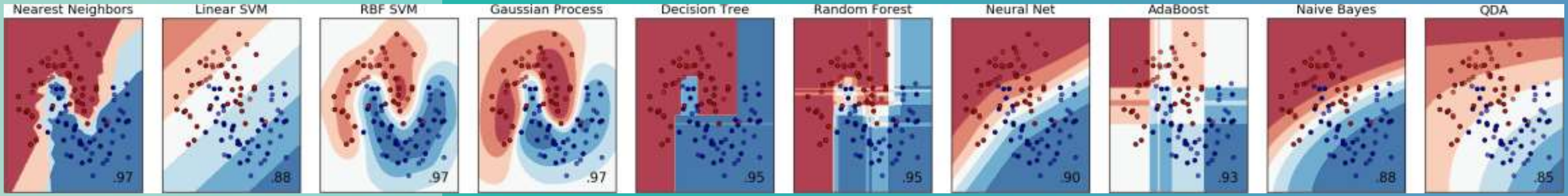
Input data



How to classify this dataset into 2 categories ?

“red” and “blue”

Apply different algorithms on the same data set



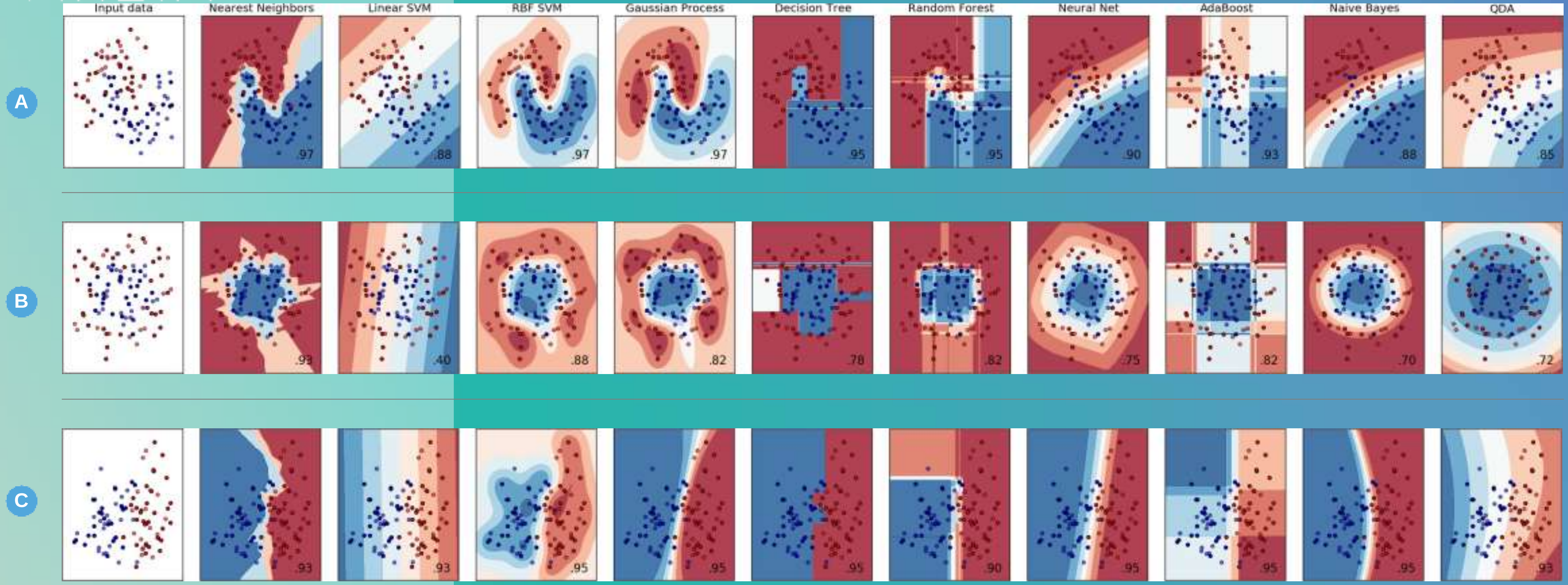
The plots show training points in solid colors and testing points semi-transparent.
The lower right shows the classification accuracy on the test set.

Source: Classifier comparison http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html#

Example: how to classify the data points ?

Use different algorithms to classify the data set.

In this example, there are 3 different data sets- A, B, C. You can see how they're classified with different algorithms.



The plots show training points in solid colors and testing points semi-transparent.
The lower right shows the classification accuracy on the test set.

Source: Classifier comparison http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html#

Example: Stock Prediction

AREIX

- Data collection & construction
  - API & web scrape
  - Label: outperform S&P500 10%
- Feature engineering
  - Standarlization
  - Feature augmentation
- Model selection & training & evaluation & optimization
  - Model:
    - RandomForest (Bagging)
    - Lightgbm (Boosting)
  - Metrics: Accuracy, Annualized Return, Volatility...
  - HP tunning: RandomizedSearch & 10-fold Crocss Validation

Date	Unix	Ticker	Price	stock_p_change	SP500	SP500_p_change	Market Cap	Enterprise Value	Trailing P/E	Forward P/E	...	200-Day Moving Average	Avg Vol (3 month)
2007-07-11 5:59:28	1.184105e+09	ctas	33.538876	-33.02	122.084023	-15.90	6.350000e+09	7.080000e+09	19.13	17.79	...	39.30	1064470.0
2013-04-26 7:42:40	1.366969e+09	ctas	41.474506	30.78	143.779022	20.10	5.480000e+09	6.550000e+09	18.46	16.09	...	42.70	580981.0
2006-10-23 3:27:49	1.161581e+09	ctas	34.183594	-12.08	108.906029	12.46	6.630000e+09	7.360000e+09	20.60	16.72	...	39.51	803655.0
2006-09-05 5:34:29	1.157442e+09	ctas	31.392931	-3.79	103.854393	14.30	6.110000e+09	6.670000e+09	19.58	15.57	...	39.94	894611.0
2008-11-02 1:33:44	1.225633e+09	vz	17.552139	5.26	79.804276	10.53	8.428000e+10	1.273800e+11	13.56	10.83	...	34.48	19325800.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2007-06-08 5:38:43	1.181288e+09	vno	57.148098	-16.96	120.800285	-8.01	1.746000e+10	2.693000e+10	33.66	18.98	...	122.43	1181170.0
2013-04-22 1:17:35	1.366565e+09	vno	54.257736										
2008-05-12	1.210607e+09	vno	48.502396										

```
X_train = training_data[features].values
# Generate the labels: '1' if a stock beats the S&P500 by more than 10%, else '0'.
y_train = (training_data['stock_p_change']-training_data['SP500_p_change'] >= 10)
X_train
```

```
array([[6.350e+09, 7.080e+09, 1.913e+01, ..., 4.200e+00, 3.700e+00,
        4.490e+06],
       [5.480e+09, 6.550e+09, 1.846e+01, ..., 8.000e+00, 5.600e+00,
        5.860e+06],
       [6.630e+09, 7.360e+09, 2.060e+01, ..., 5.600e+00, 2.900e+00,
        4.220e+06],
       ...,
       [1.465e+10, 2.688e+10, 2.959e+01, ..., 4.500e+00, 4.900e+00,
        6.730e+06],
       [8.020e+09, 1.999e+10, 1.277e+01, ..., 2.000e+00, 7.700e+00,
        2.180e+04],
       [1.613e+10, 1.631e+10, 2.935e+01, ..., 3.300e+00, 2.200e+00,
        3.460e+06]])
```

```
y_train
```

```
Date
2007-07-11 05:59:28    False
2013-04-26 17:42:40     True
2006-10-23 13:27:49    False
2006-09-05 15:34:29    False
2008-11-02 21:33:44    False
                        ...
2007-06-08 15:38:43    False
2013-04-22 01:17:35    False
2008-05-12 23:47:41    False
2009-01-12 23:29:46    False
```

Cond.

AREIX

Random Forest

```python
def predict_stocks(test_data):
    X_train, y_train = build_data_set()
    clf = RandomForestClassifier(n_estimators=100)
    clf.fit(X_train, y_train)

    test_data.dropna(axis=0, how="any", inplace=True)
    features = test_data.columns[6:]
    X_test = test_data[features].values
    z = test_data["Ticker"].values

    # Get the predicted tickers
    y_pred = clf.predict(X_test)
    if sum(y_pred) == 0:
        print("No stocks predicted!")
    else:
        invest_list = z[y_pred].tolist()
        print(
            f"{len(invest_list)} stocks predicted to outperform the S&P500 by more than {OUTPERFORMANCE}%:"
        )
        print(" ".join(invest_list))
        print(f"y_pred length: {len(y_pred)}")
        return invest_list

if __name__ == "__main__":
    print("Building dataset and predicting stocks...")
    predict_stocks()
```

```
Building dataset and predicting stocks...
28 stocks predicted to outperform the S&P500 by more than 10%:
CNX OI BAX SWK WGO MAC LH SNA LNC BIIB BWA GES GWW AIZ GNW VIAB DNR R BIG PBI BLK DLX GTN AMP X BBBY LM APD
y_pred length: 286
```

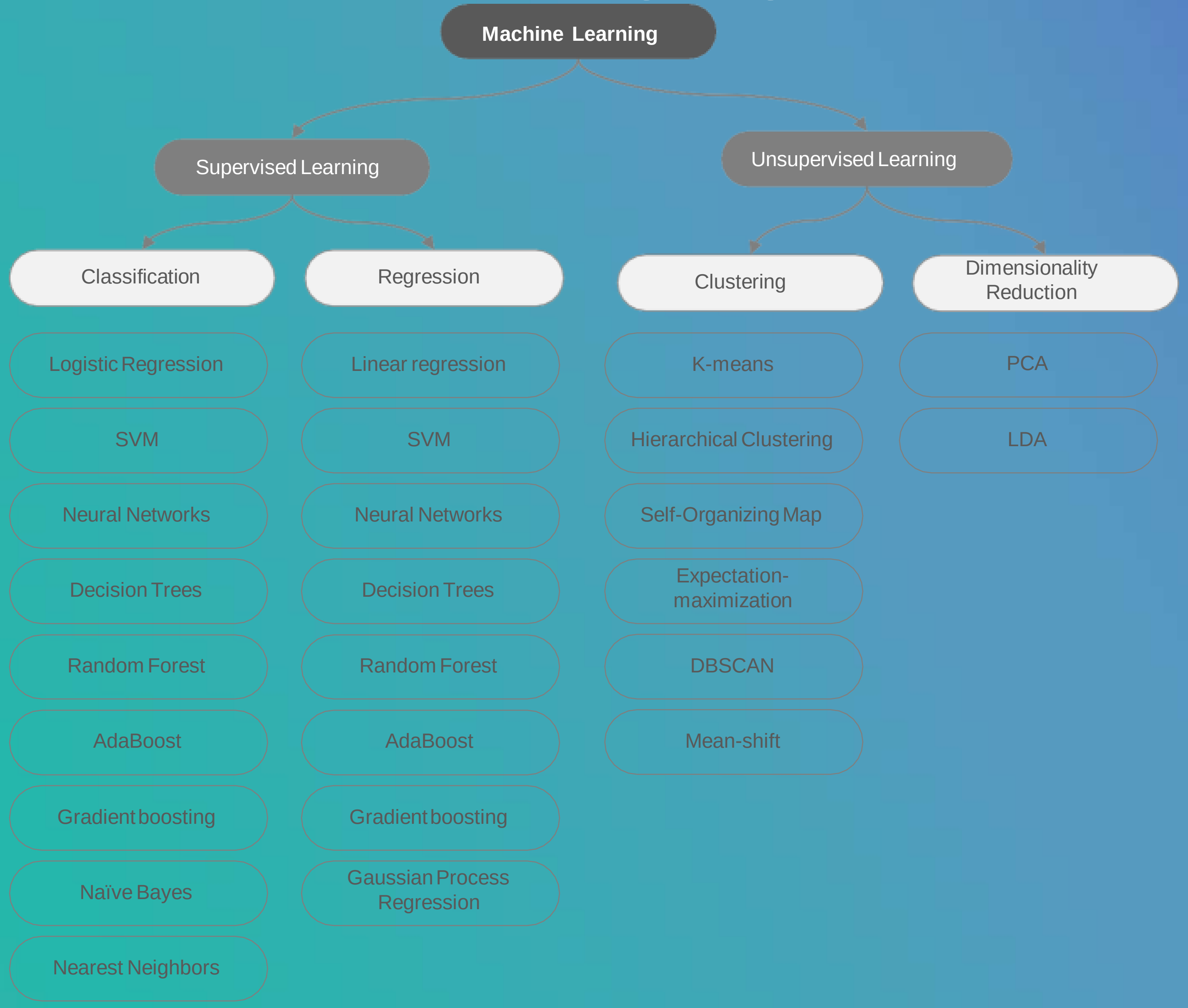
LightGBM

```python
import lightgbm as lgb
from sklearn import metrics

param = {'num_leaves': 31,
         'min_data_in_leaf': 20,
         'max_depth': 15,
         'num_leaves': 20,
         'objective': 'binary',
         'learning_rate': 0.06,
         "boosting": "gbdt",
         'feature_fraction': 0.8,
         'subsample': 0.2,
#          "bagging_freq": 1,
#          "bagging_seed": 11,
#          'objective':'multiclass',
         "metric": 'None',
         "verbosity": -1}

st = time.time()
trn_data = lgb.Dataset(X_train, y_train)
num_round =1000
lgb_clf = lgb.train(param, trn_data, num_round, verbose_eval=300)
pred_y = lgb_clf.predict(X_test, num_iteration=lgb_clf.best_iteration
```

```python
        'column': features,
        'importance': lgb_clf.feature_importance(),
    }).sort_values(by='importance', ascending=False)
```

	column	importance
4	PEG Ratio	741
29	Beta	705
3	Forward P/E	695
15	Quarterly Revenue Growth	659
24	Total Debt/Equity	627
38	Short Ratio	595
8	Enterprise Value/EBITDA	583
40	Shares Short (prior month	581
32	Avg Vol (3 month)	571
20	Quarterly Earnings Growth	557
37	Shares Short	548
25	Current Ratio	547
30	50-Day Moving Average	521
22	Total Cash Per Share	514

Some of machine learning algorithms



Q: How to select the right algorithm ?
A: The answer to the question varies depending on many factors, including:

- The size, quality, and nature of data.
- The available computational time.
- The urgency of the task.
- What you want to do with the data.

Model Evaulation

Use different metrics to measure the performance of the model

By using Metrics and scoring to quantify the quality of predictions

- **For Classfication**
  - Accuracy
  - Precision
  - Recall
  - F1
  - ROC_AUC
  - Jaccard Similarity
  - ....
- **For Regression**
  - Max error
  - Mean square error
  - R^2 score
  - ....
- **For Clustering**
  - Mutual Information
  - ....

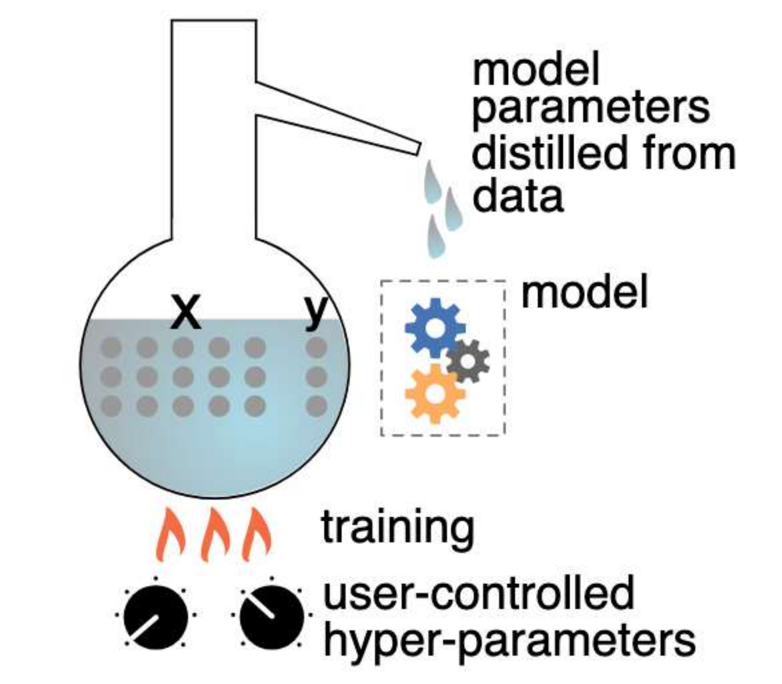
Precision = TP / (TP + FP)
Recal = TP / (TP + FN)
F1 = 2 * (precision * recall) / (precision + recall)
ROC_AUC : TPR VS FPR

https://scikit-learn.org/stable/modules/model_evaluation.html

Parameters and Hyper-parameters

Iteratively tuning the Hyperparameter so that the model can learn the "best" Parameters from data

The hyper-parameters are specified by the developer/data scientist while parameters are computed from the data via the algorithms.



- X is the feature vectors
- Y is the target variable

- Model's parameters are the variables that your chosen machine learning technique uses to adjust to your data. They are internal to the model. They are estimated or learned from data. They are often not set manually by the practitioner.
- Hyperparameters control how a machine learning algorithm fits the model to the data. Hyper-parameters are specified by the programmer, not computed from the training data, and are often used to tune a model to improve accuracy for a particular data set

The examples of hyper-parameters :

- Number of layers/units, learning-rate, dropout rate weight d ecay, activation function... in Neural network
- Number of trees, max depth... in Random Forest

https://mlbook.explained.ai/intro.html#sec:2.2.3

Model Optimization

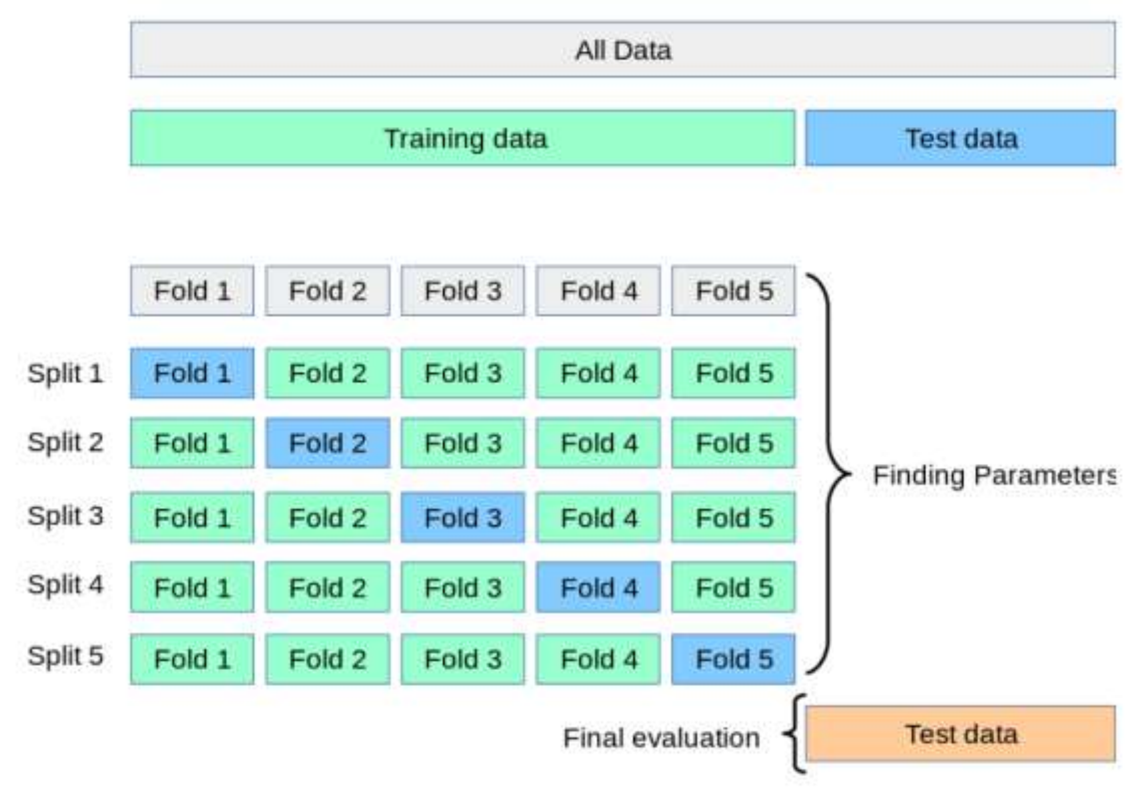
Model Optimization aka hypterparameter tunning is one of the key step to optimize the performance of the model

- **Grid Search**
- • Gaussian Process based
- • Easy to try, but some crucial drawbacks
- **Random Search**
- • Often leads better result than grid search
- **Bayesian Optimazation**
- • Random Forest based
- • Deep Neural Net based
- • Tree Parzan Estimators based

- **Recommend Lib: Sklearn, Skopt, Hyperopt**

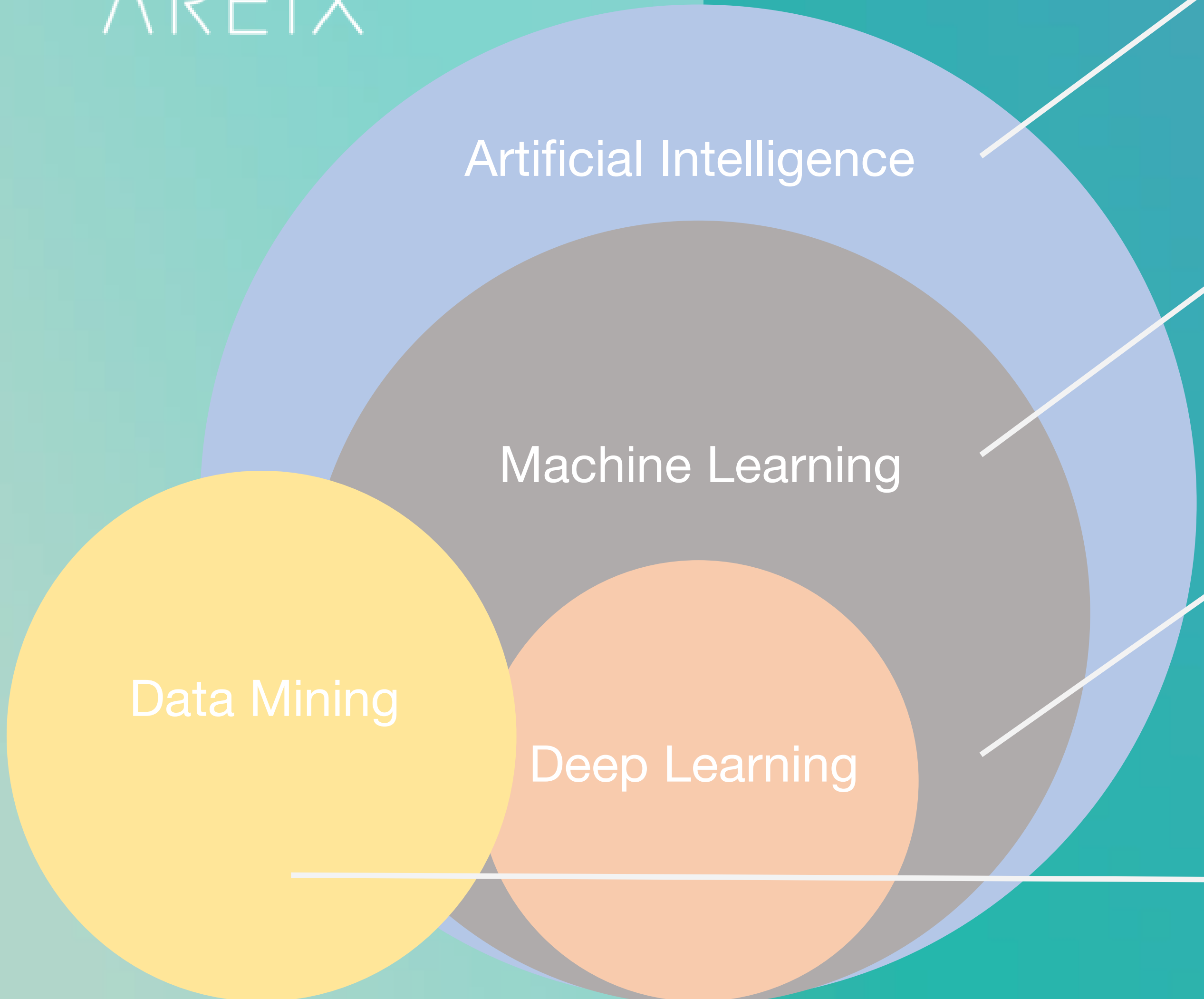
N-fold Cross Validation

- Use n-1 of the folds as training data
- Validate on the remaining data
- Averagre the values computed in the loop



https://scikit-optimize.github.io/stable/

About AI , Machine Learning and Deep Learning



A technique which enables machines to mimic human behaviour

Subset of AI technique which use statistical methods to enable machines to improve with experience

Subset of ML algorithms which make the computation of multi-layer neural network feasible

The process of extracting useful information from a vast amount of data

Wrap up

- Machine Learning uses historical data to make predictions
- ML process
- Modelling involved training, evaluation & optimization
- Loss function: Measure the error
- Parameters: Computed from training data (automatically trained)
- Hyper-parameters: Human-defined and need (manually fine-tuned)
- Classfication & Regression
- Model evaluation by metrics and scoring
- Tunning HP techniques

- ML VS DM
  - Both are good at pattern recognition and learning from data
  - But serves different purpose

- ML VS DL
  - Structured data & unstructured data
  - Less feature engineering