Microsoft

Workshop Cloud Experts

« Introduction au Machine Learning »

Mercredi 5 juin 2019

Vos interlocuteurs Microsoft

Mouhamadou Diallo
mdiallo@microsoft.com

Serge Retkowsky
serge.retkowsky@microsoft.com

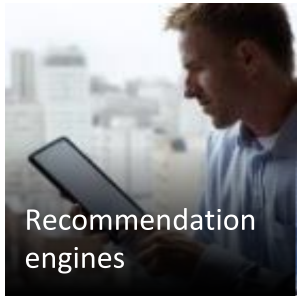
What is Machine Learning ?

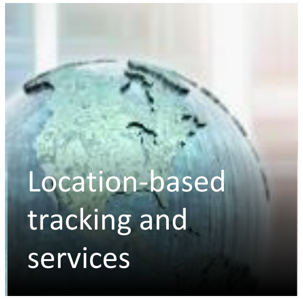
Machine Learning - using past data to predict the future

Imagine what Machine Learning could do to your business

Churn analysis

Social network analysis

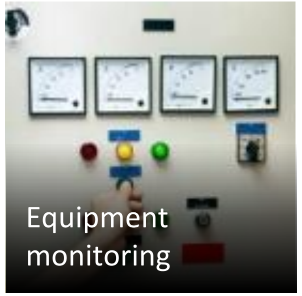

Recommendation engines


Location-based tracking and services

Vision Analytics

Weather forecasting for business planning

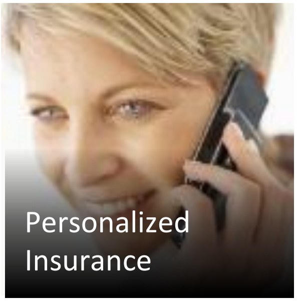
Legal discovery and document archiving


Equipment monitoring

Advertising analysis

Pricing analysis

Fraud detection


Personalized Insurance

Georges Box

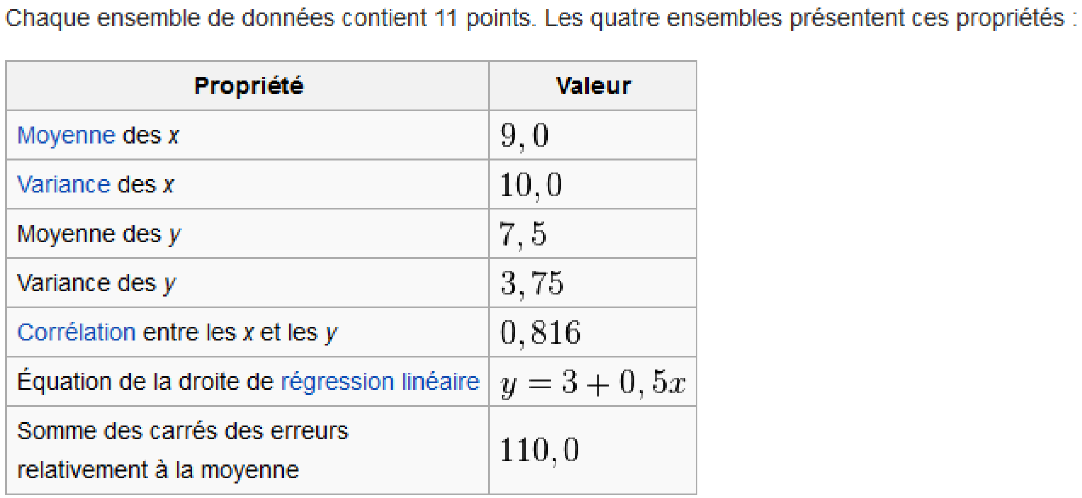
« Tous les modèles sont faux, mais certains sont utiles »

Un modèle n'est qu’une simplification de la réalité destinée à la faire comprendre et à obtenir des prévisions convenables.

Le quartet d’Ascombe ou l’importance de l’analyse exploratoire graphique

Le quartet d’Ascombe ou l’importance de l’analyse exploratoire graphique

- **Quatre ensembles** de données de **2 variables** (X et Y) et de **onze observations** ont les mêmes caractéristiques de distribution.
- Chaque ensemble de données est caractérisé comme suit :



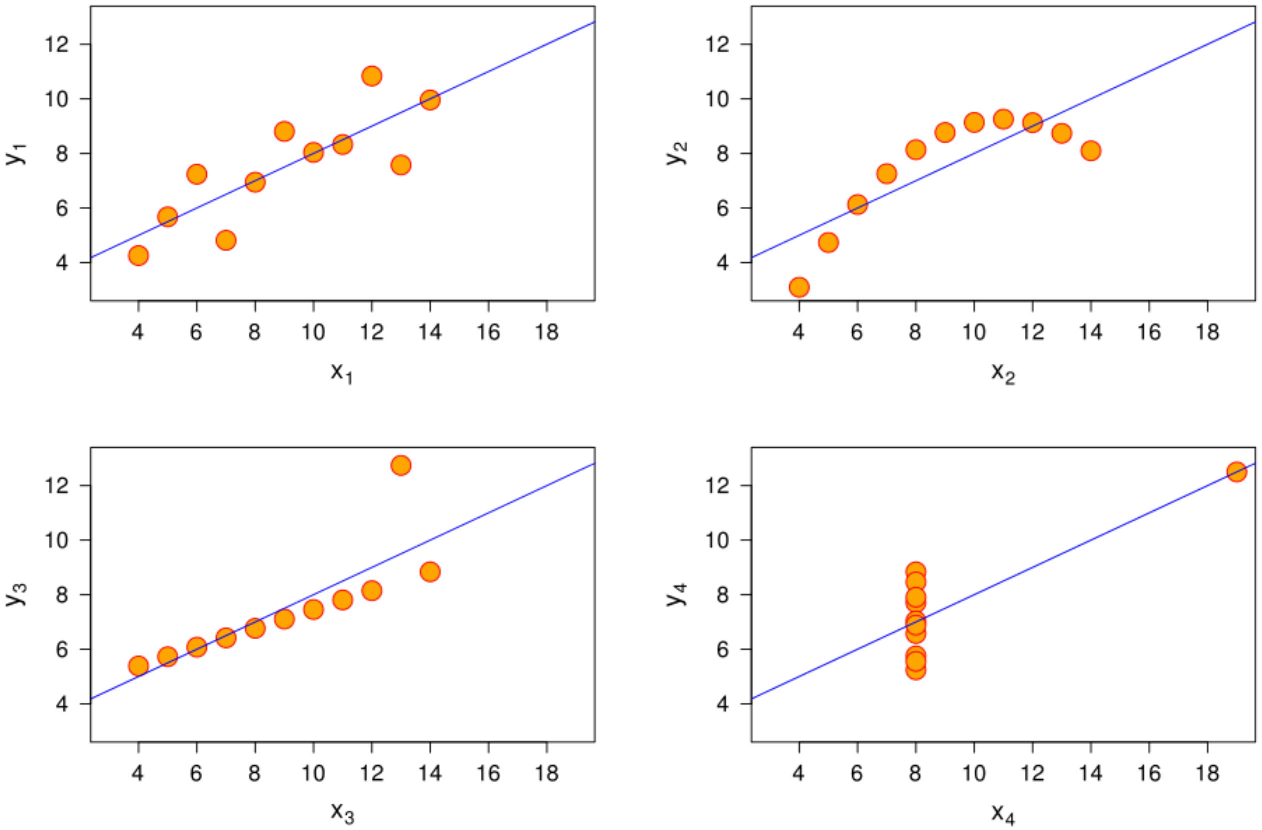
Chaque ensemble de données contient 11 points. Les quatre ensembles présentent ces propriétés :

Propriété	Valeur
Moyenne des *x*	$9,0$
Variance des *x*	$10,0$
Moyenne des *y*	$7,5$
Variance des *y*	$3,75$
Corrélation entre les *x* et les *y*	$0,816$
Équation de la droite de régression linéaire	$y = 3 + 0,5x$
Somme des carrés des erreurs relativement à la moyenne	$110,0$

- Question : Est-ce que ces quatre populations sont comparables ?

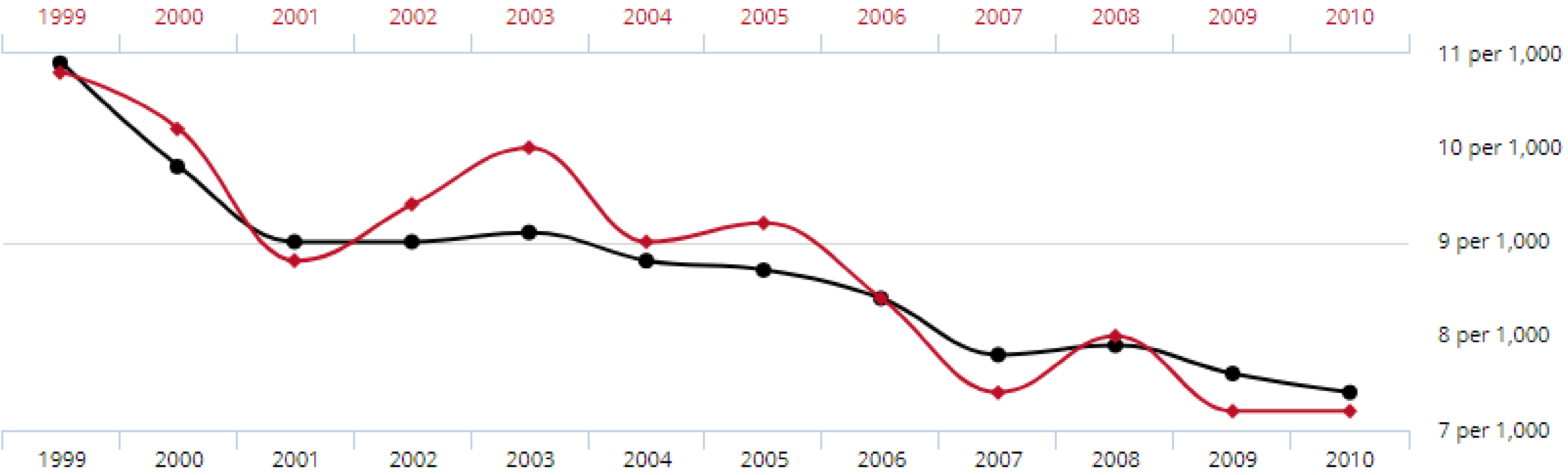
Le quartet d’Ascombe ou l’importance de l’analyse exploratoire graphique

Est-ce que ces 4 populations sont identiques ? Votre avis ?



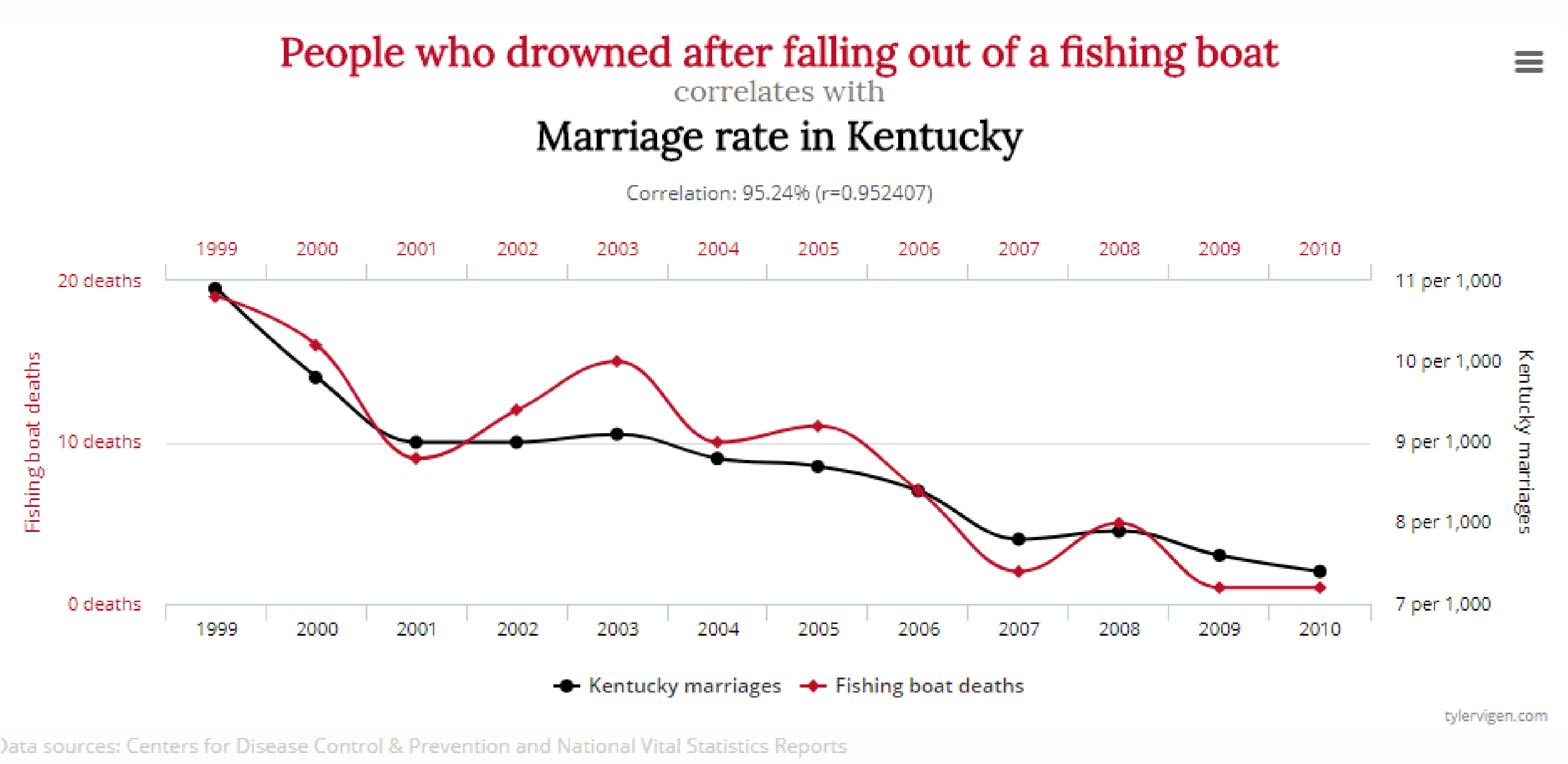
Corrélation / Causalité ?

R = 0.952407



Corrélation / Causalité ?

R = 0.952407



http://tylervigen.com/spurious-correlations

Le paradoxe de Simpsons

Le paradoxe de Simpsons (effet de Yule-Simpson)
Exemple : Analyse de la perte de poids de patients

Le **paradoxe de Simpson** ou **effet de Yule-Simpson** est un paradoxe statistique dans lequel un phénomène observé de plusieurs groupes semble s'inverser lorsque les groupes sont combinés.

Ce résultat qui semble impossible au premier abord est lié à des éléments qui ne sont pas pris en compte (comme la présence de variables non-indépendantes ou de différences d'effectifs entre les groupes, etc.).

	Régime	Activité sportive
Hommes	10/40 (25%)	22/80(27,5%)

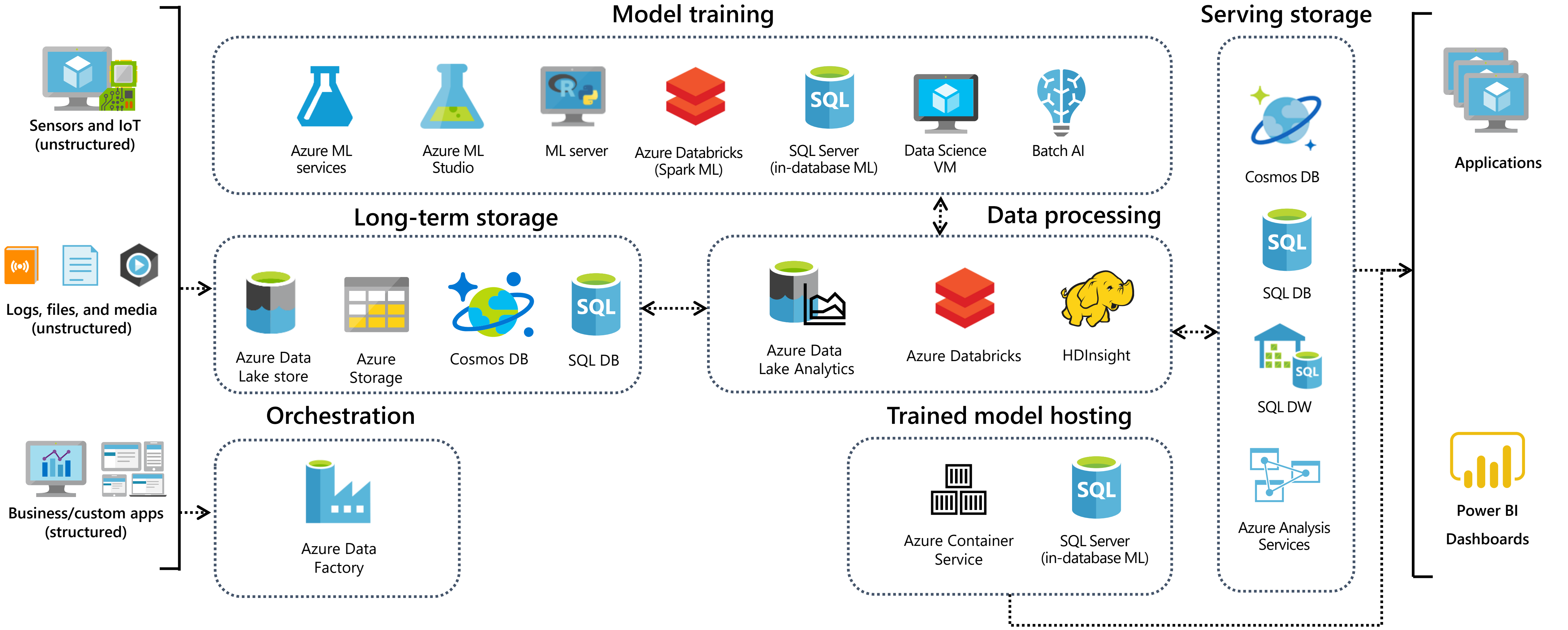
	Régime	Activité sportive
Hommes	10/40 (25%)	22/80(27,5%)
Femmes	60/80 (75%)	35/40 (87,5%)

	Régime	Activité sportive
Hommes	10/40 (25%)	22/80(27,5%)
Femmes	60/80 (75%)	35/40 (87,5%)
Total	**70/120 (58,33%)**	**57/120 (47,5%)**

https://fr.wikipedia.org/wiki/Paradoxe_de_Simpson

Advanced analytics pattern in Azure

Data collection and understanding, modeling, and deployment



Sensors and IoT (unstructured)

Logs, files, and media (unstructured)

Business/custom apps (structured)

Model training

Azure ML services

Azure ML Studio

ML server

Azure Databricks (Spark ML)

SQL Server (in-database ML)

Data Science VM

Batch AI

Long-term storage

Azure Data Lake store

Azure Storage

Cosmos DB

SQL DB

Data processing

Azure Data Lake Analytics

Azure Databricks

HDInsight

Orchestration

Azure Data Factory

Trained model hosting

Azure Container Service

SQL Server (in-database ML)

Serving storage

Cosmos DB

SQL DB

SQL DW

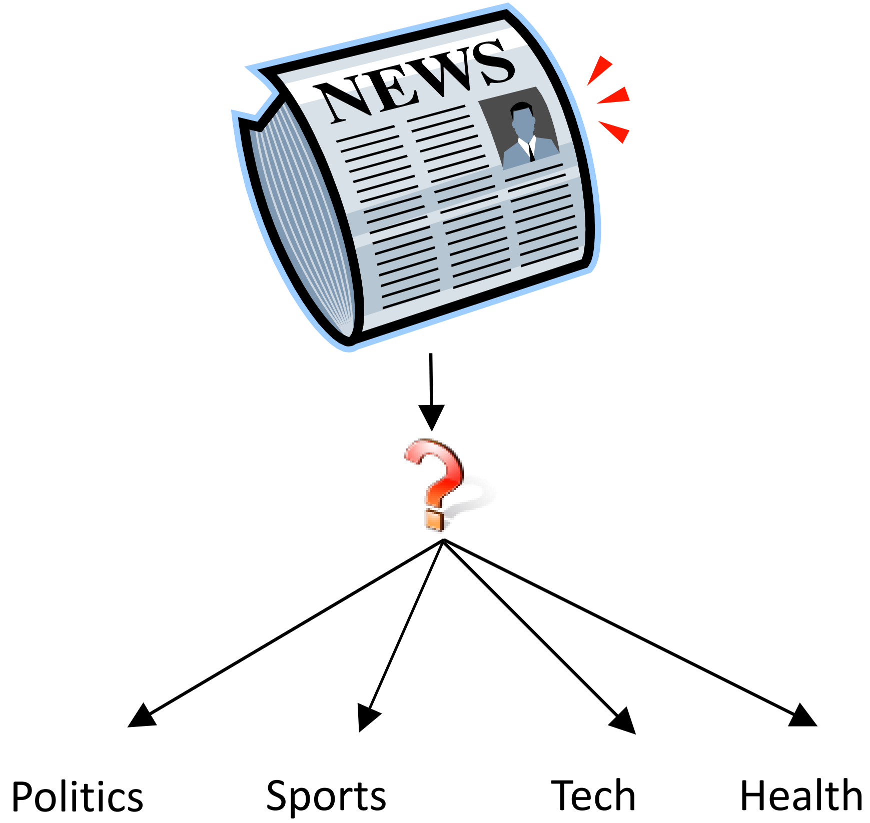
Azure Analysis Services

Applications

Power BI Dashboards

Example - Model (Classification)

Classify a news article as (politics, sports, technology, health, …)



Politics

Sports

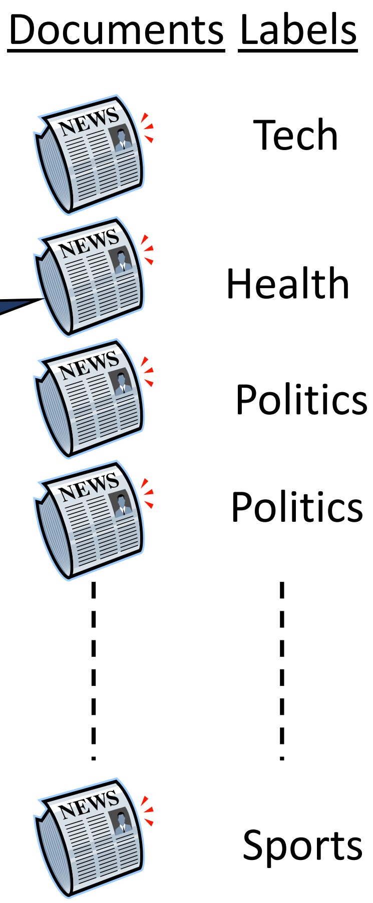
Tech

Health

Using **known data**, develop a **model** to predict **unknown data**.

Known data (Training data)

Documents consist of unstructured text. Machine learning typically assumes a more structured format of examples

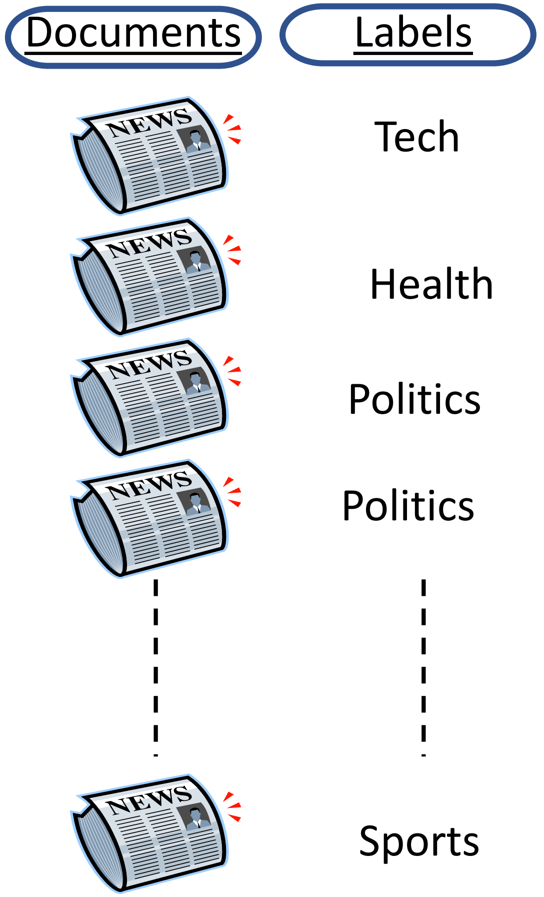


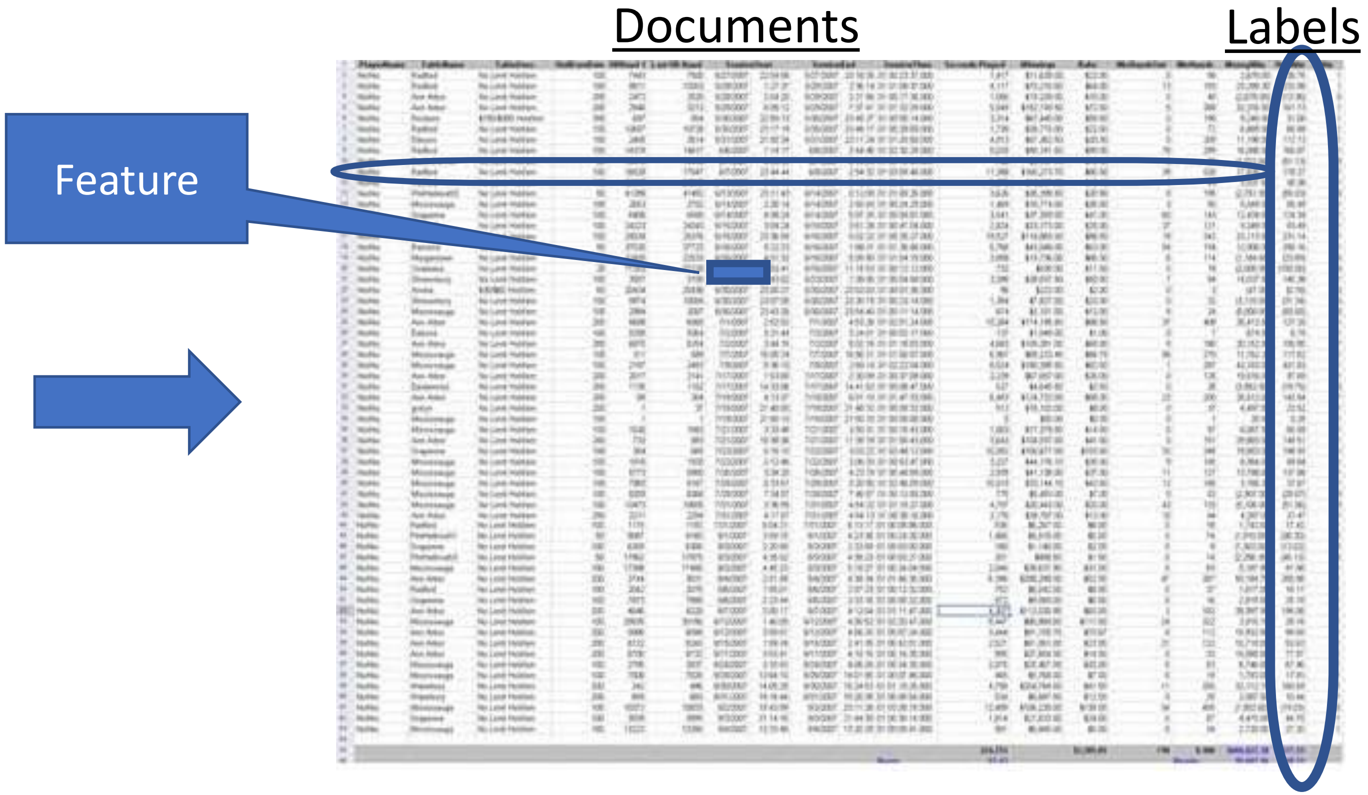
Process the raw data

Using **known data**, develop a **model** to <u>predict</u> **unknown data**.

Known data (Training data)

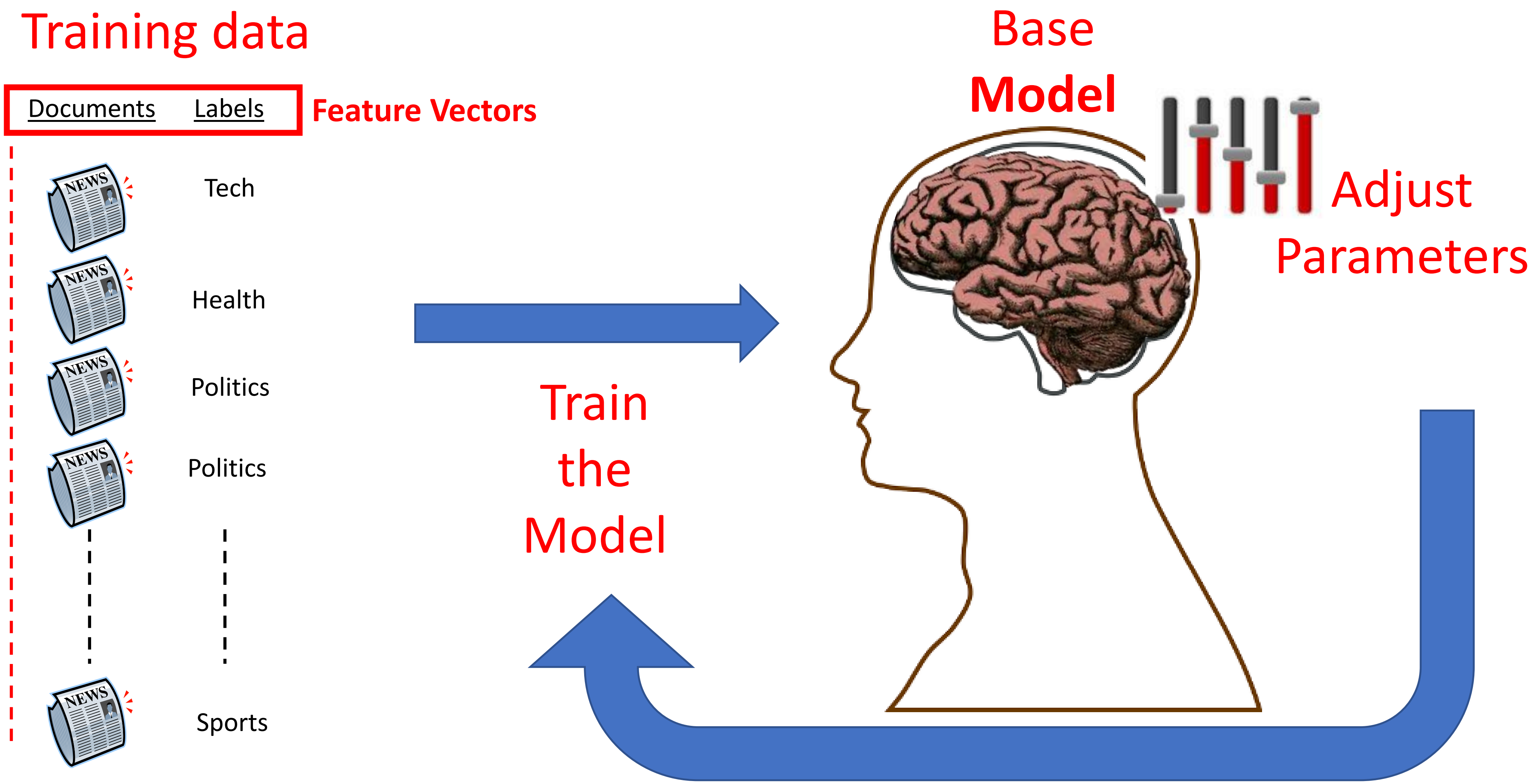
Process each data instance to represent it as a feature vector





Using **known data**, develop a **model** to <u>predict</u> **unknown data**.

Developing a Model



Training data

Documents Labels

Feature Vectors

Tech

Health

Politics

Politics

Sports

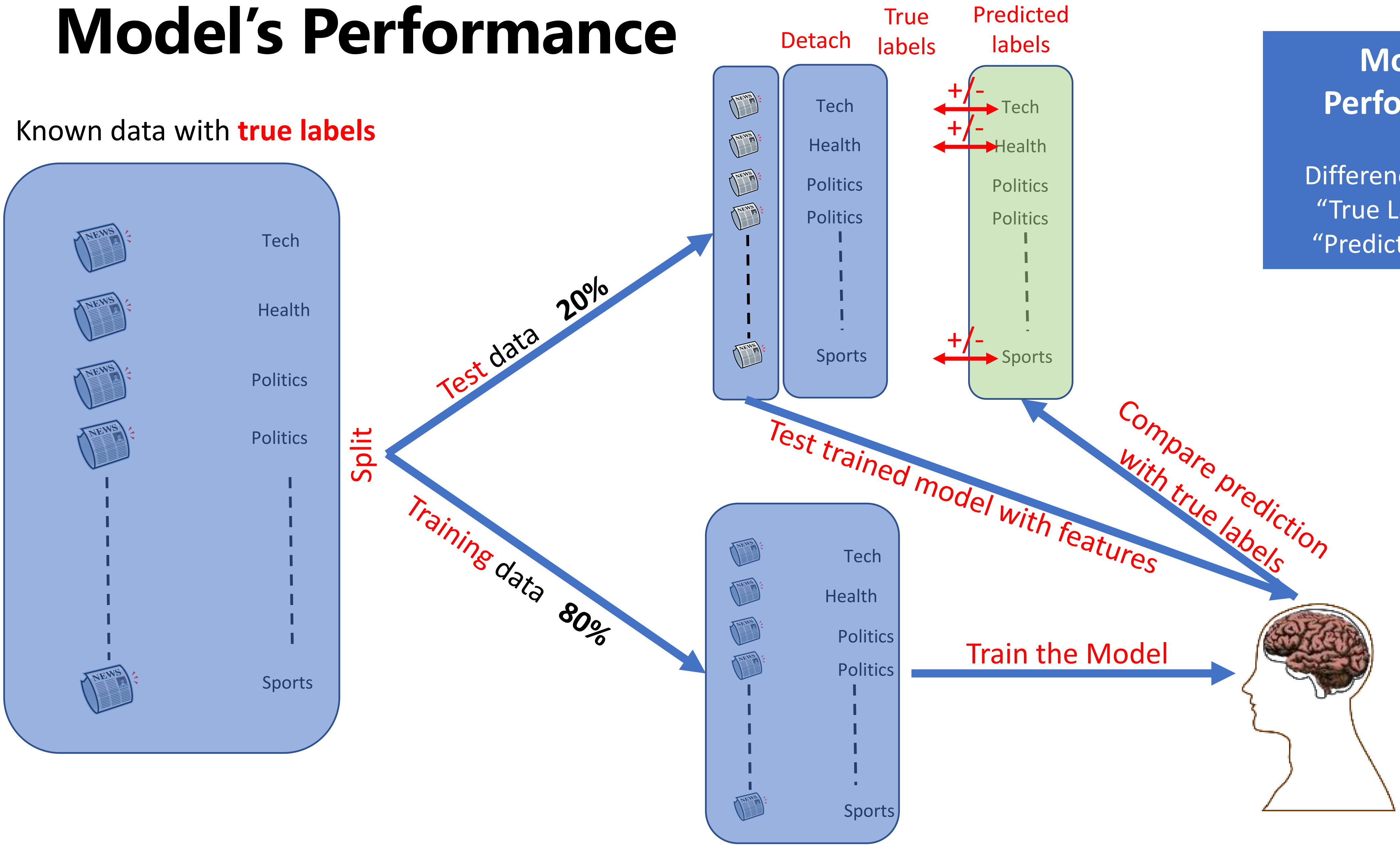
Base **Model**

Adjust Parameters

Train the Model

Using **known data**, develop a **model** to predict **unknown data**.

Model's Performance



Model's Performance

Difference between "True Labels" and "Predicted Labels"

Different Families of Modeling

Supervised, Unsupervised, Semi-supervised

- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values.
- Example: Buy/Don’t buy

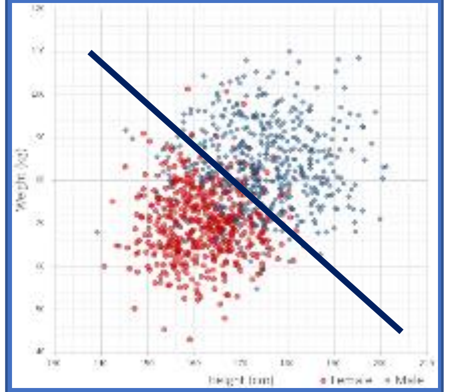
- **Unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data.
- Example: Clustering

- **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data.
- Example: Fraud detection
- **Reinforcement learning**

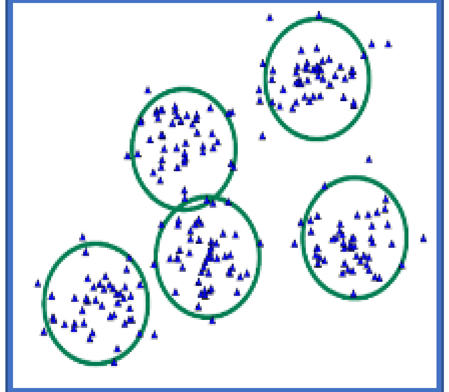
Common Classes of Algorithms

(Supervised|Unsupervised)

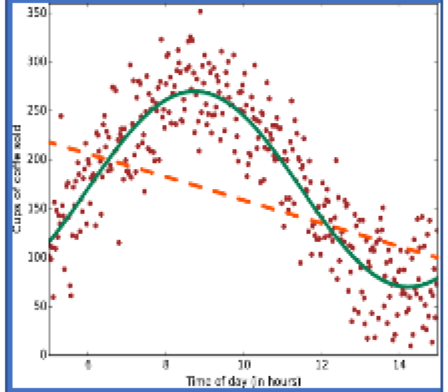
Classification



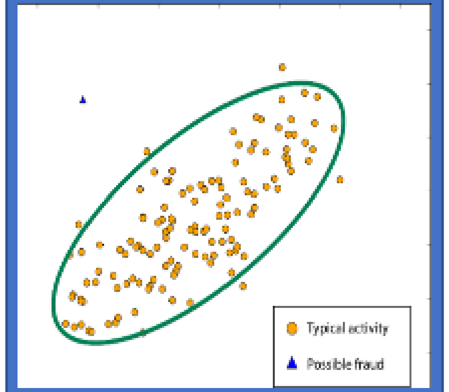
Clustering



Regression



Anomaly Detection



Why you need to know these algorithms?

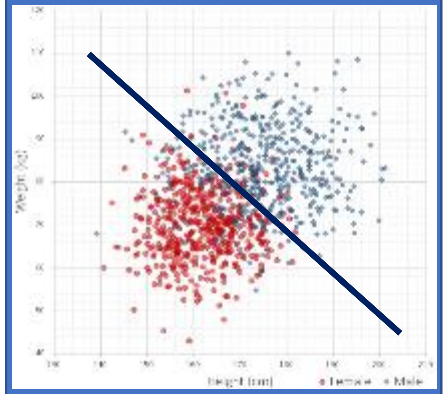
- If you want to answer a YES|NO question, it is **classification**
- If you want to predict a numerical value, it is **regression**
- If you want to group data into similar observations, it is **clustering**
- If you want to recommend an item, it is **recommender system**
- If you want to find anomalies in a group, it is **anomaly detection**
- and many other ML algorithms for specific problem

Classification

Scenarios:

- Which customer are more likely to buy, stay, leave (churn analysis)
- Which transactions|actions are fraudulent
- Which quotes are more likely to become orders
- Recognition of patterns: speech, speaker, image, movement, etc.

Classification



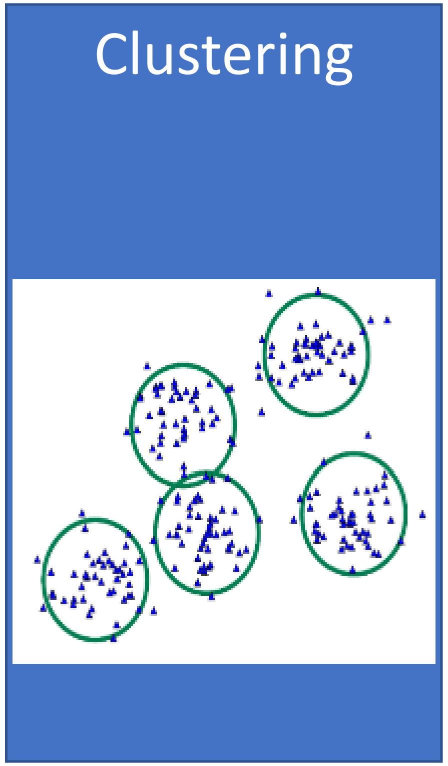
Algorithms: Boosted Decision Tree, Decision Forest, Decision Jungle, Logistic Regression, SVM, ANN, etc.

Clustering

Scenarios:

- Customer segmentation: divide a customer base into groups of individuals that are similar in specific ways relevant to marketing, such as age, gender, interests, spending habits, etc.
- Market segmentation
- Quantization of all sorts, such as, data compression, color reduction, etc.
- Pattern recognition

Algorithms: K-means


Clustering

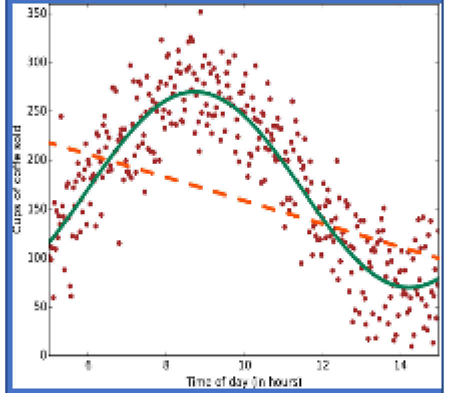
Regression

Scenarios:

- Stock prices prediction
- Sales forecasts
- Premiums on insurance based on different factors
- Quality control: number of complaints over time based on product specs, utilization, etc.
- Workforce prediction
- Workload prediction

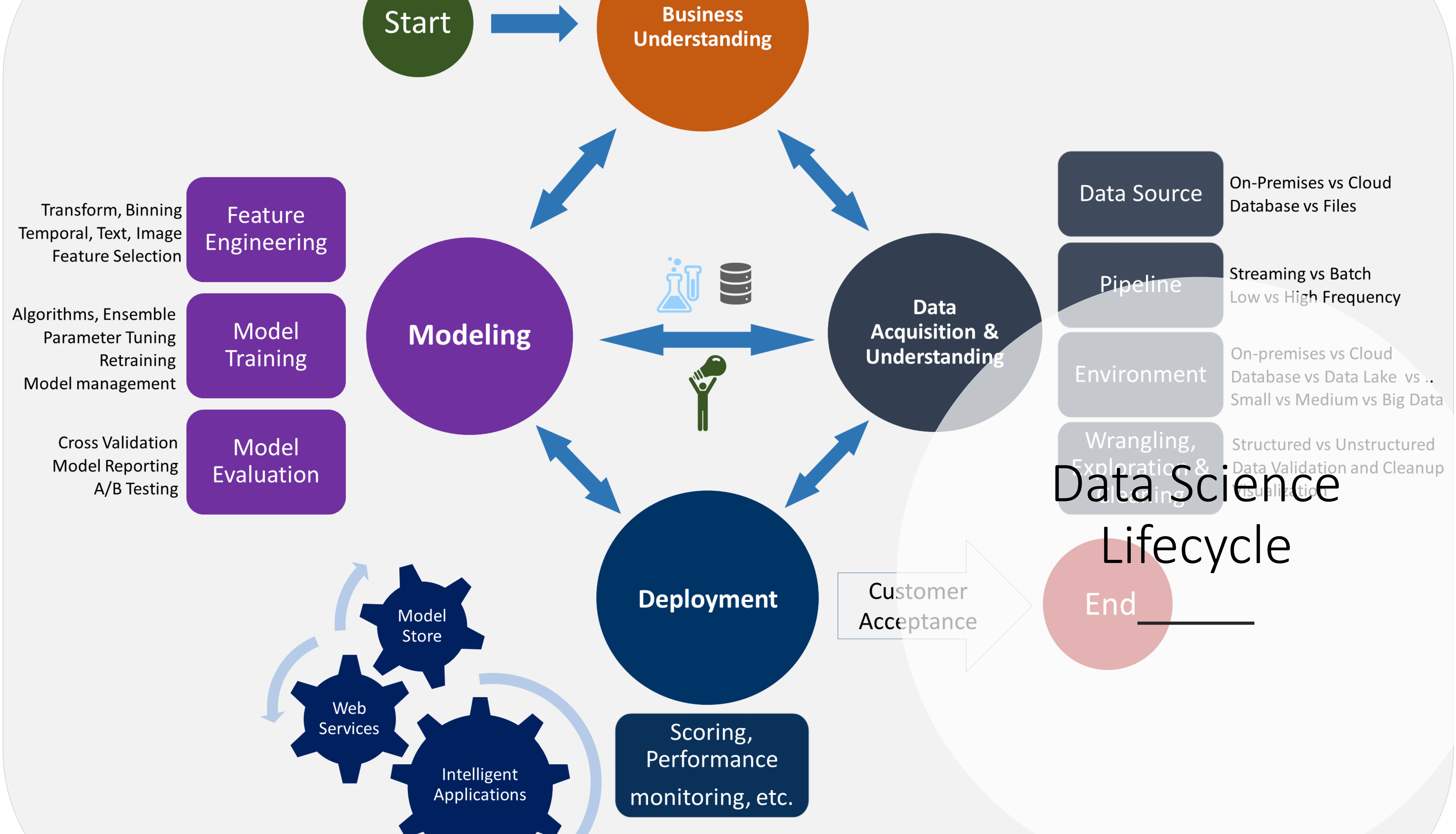
Algorithms: Bayesian Linear, Linear Regression, Ordinal Regression, ANN, Boosted Decision Tree, Decision Forest

Regression



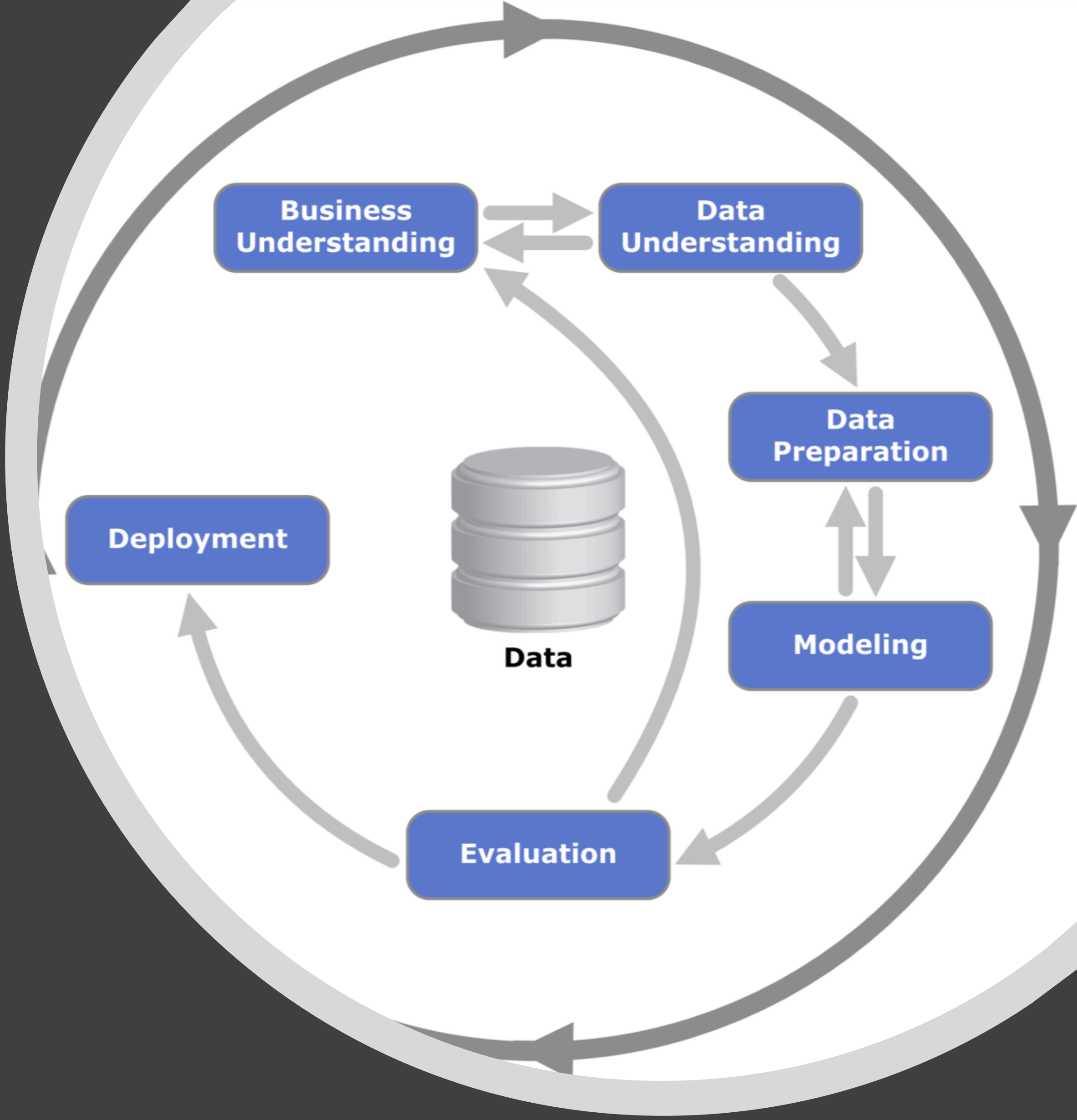
Modeling Process

Data Science Lifecycle



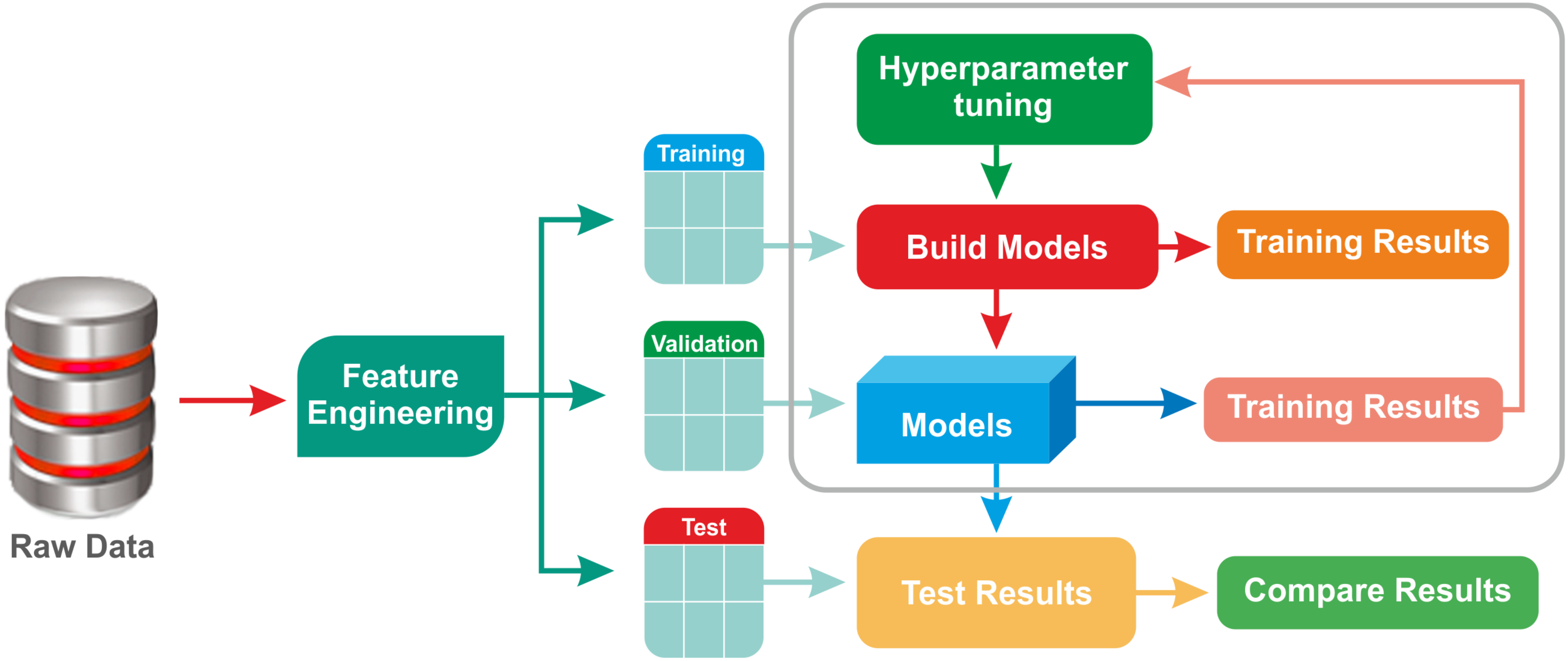
Crisp-DM

- CRoss Industry Standard Process for Data Mining – 1996
  - Non-proprietary
  - Application/Industry neutral
  - Tool neutral
  - Focus on business issues
    - As well as technical analysis
  - Framework for guidance
  - Experience base
    - Templates for Analysis

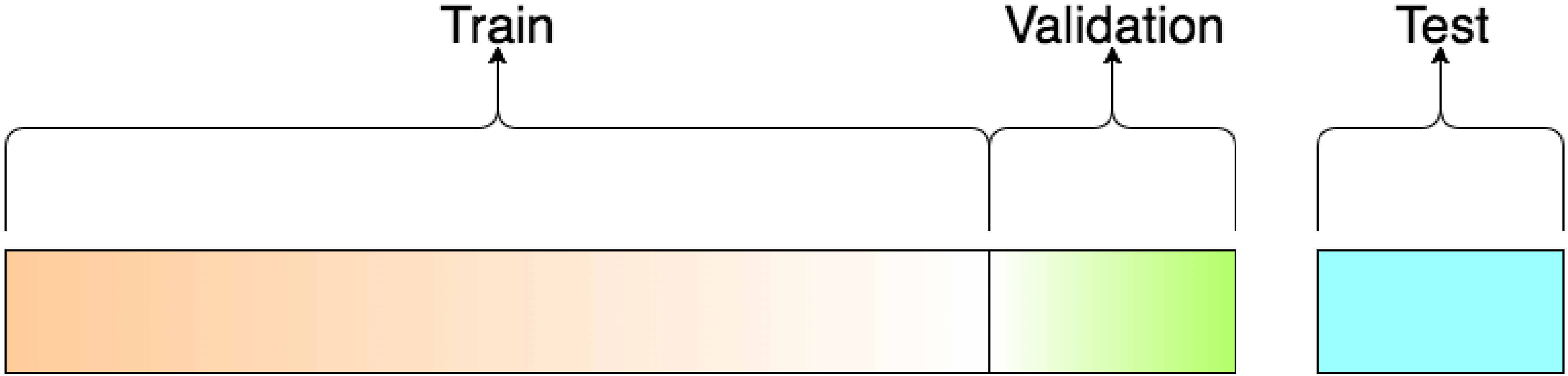


Initialize and Train Machine Learning Models

Machine Learning

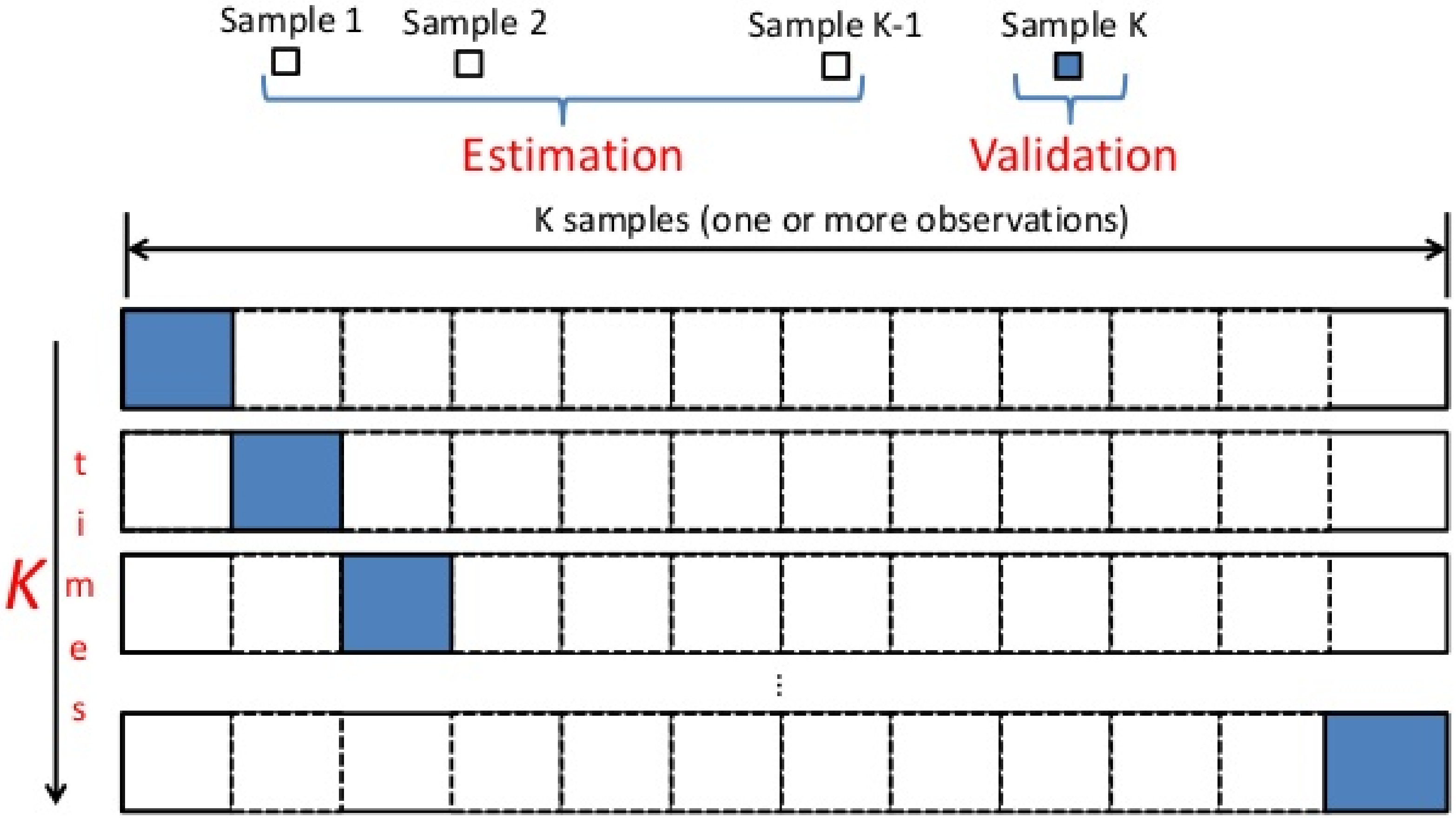


Data Partition

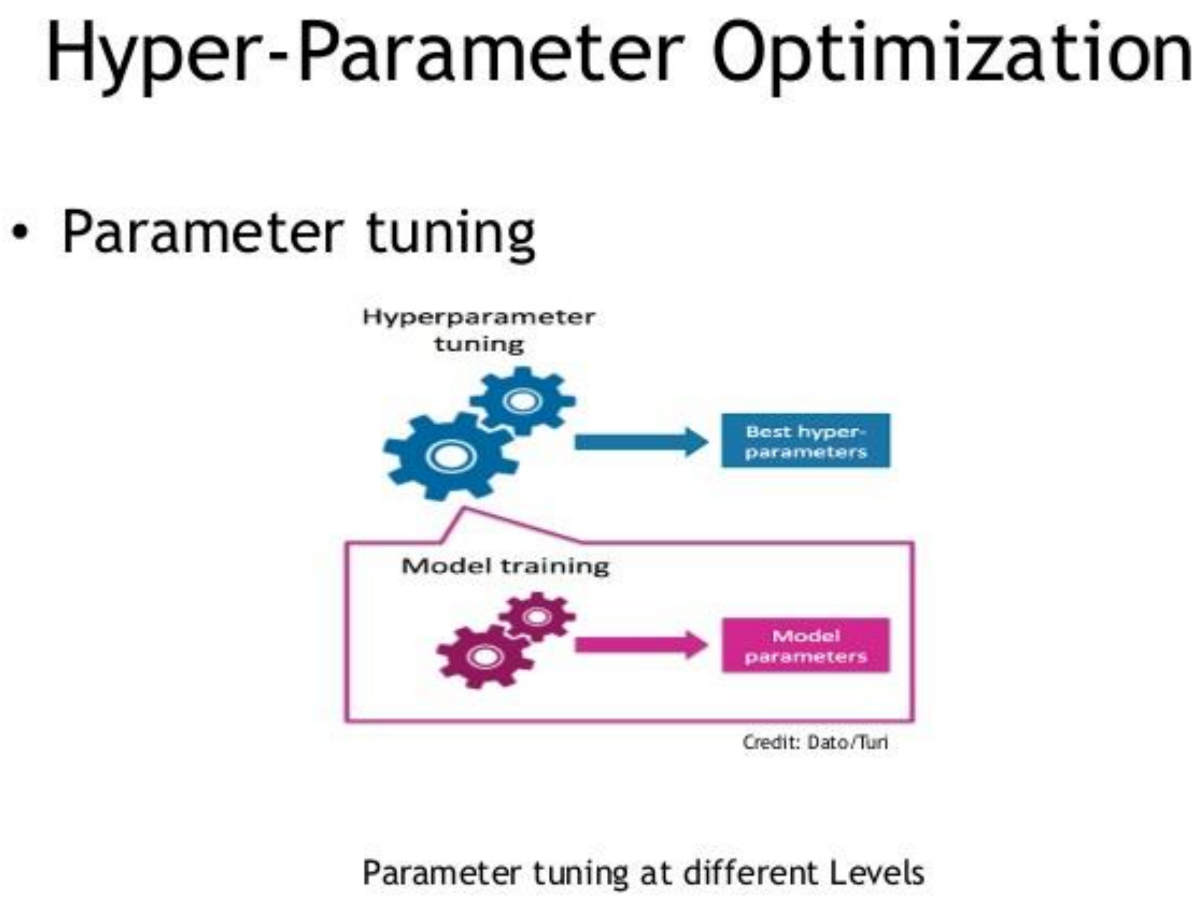


Cross Validation

- K-fold cross-validation:



Hyperparameter tuning



Hyper-Parameter Optimization

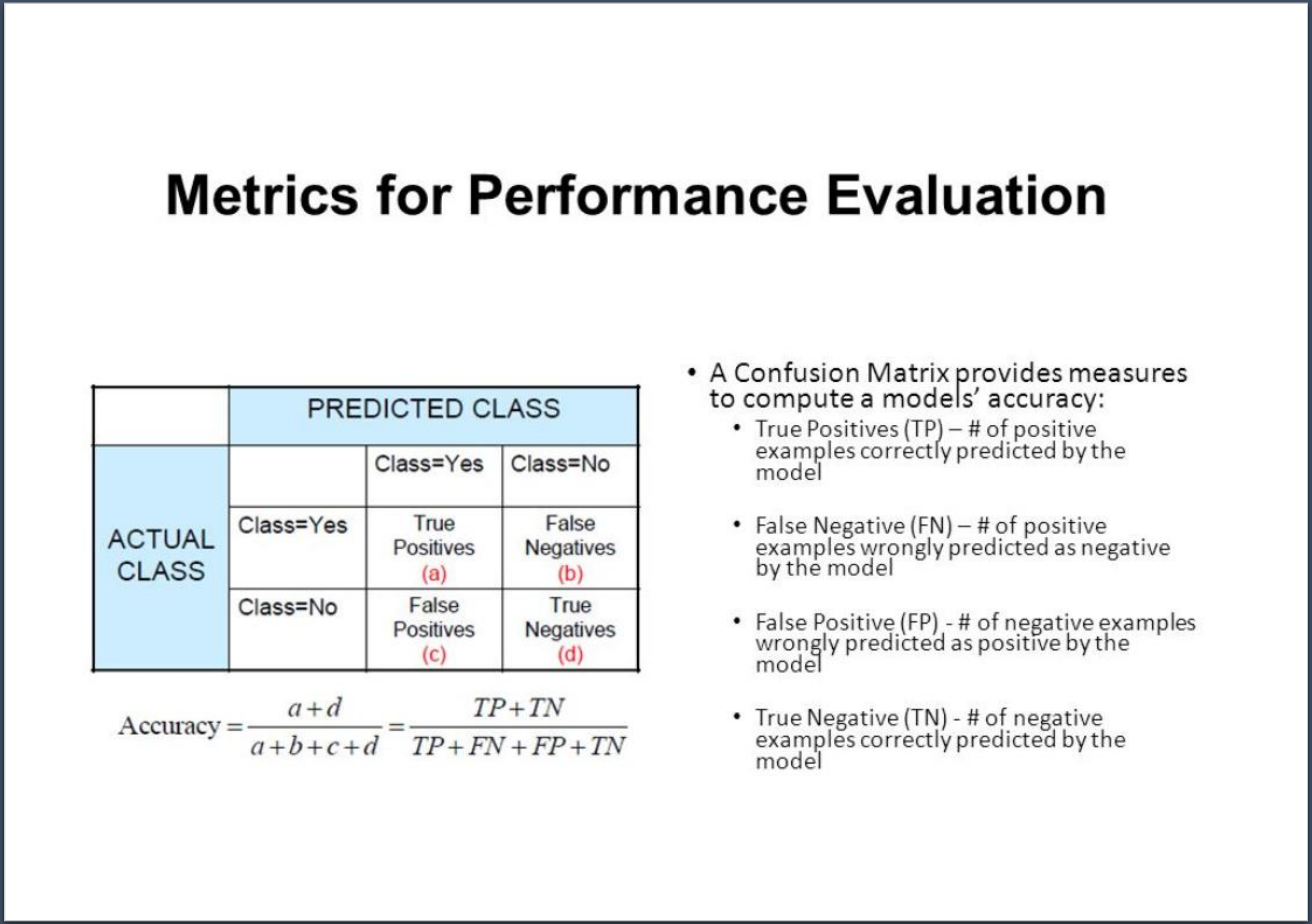
- Parameter tuning

Credit: Dato/Turi

Parameter tuning at different Levels

Validate Models

Evaluation Metrics for Classification



Metrics for Performance Evaluation

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	True Positives (a)	False Negatives (b)
	Class=No	False Positives (c)	True Negatives (d)

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+FN+FP+TN}$$

- A Confusion Matrix provides measures to compute a models' accuracy:
  - True Positives (TP) – # of positive examples correctly predicted by the model
  - False Negative (FN) – # of positive examples wrongly predicted as negative by the model
  - False Positive (FP) - # of negative examples wrongly predicted as positive by the model
  - True Negative (TN) - # of negative examples correctly predicted by the model

Evaluation metrics for classification

- **Accuracy**

- Measures the proportion of correctly classified cases. It gives misleading results on unbalanced datasets (unbalanced datasets contain a large percentage of samples of a unique class).

- For example, if 90% of our samples are from class 1, our algorithm could have a 90% accuracy just by predicting that all examples are from class 1. The value goes from 0 to 1 where 1 is the ideal value. It is very common to express accuracy in percentage.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{Correct}{Total\ population}$$

Evaluation metrics for classification

- **Precision**

- It answers the question: from the cases that have been classified as positives (TP and FP), which ones were classified correctly (TP)?

- The value goes from 0 to 1 where 1 is the ideal value. This measure is often used when the cost of a false positive is high.

$$Precision = \frac{TP}{TP + FP}$$

Evaluation metrics for classification

- **Recall**
- It answers the question: from the cases that should have been predicted as positives (TP and FN) which ones were classified correctly (TP)?
- The value goes from 0 to 1 where 1 is the ideal value. This measure is typically used when the cost of a false negative is high.

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

Evaluation metrics for classification

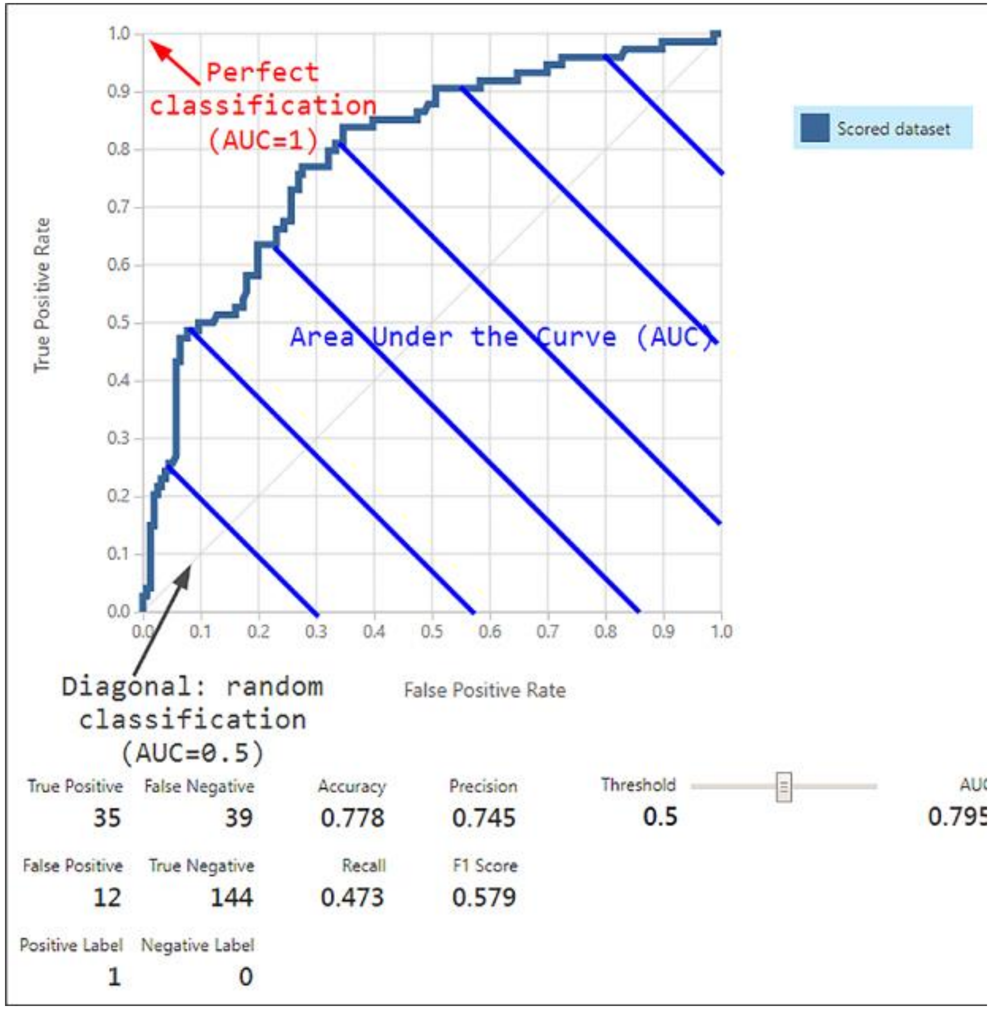
- **F1-Score (Also F-Score)**

- It is the harmonic mean of precision and recall. The value goes from 0 to 1 where 1 is the ideal value. Very commonly used when the dataset is unbalanced (skewed).

$$F_1 = 2\,\frac{Precision \cdot Recall}{Precision + Recall}$$

Evaluation metrics for classification

- **AUC**
- Measures the Area Under the ROC Curve. The ROC curve shows how the performance (true positive rate and false positive rate) of a binary classifier changes when the discrimination threshold is varied. The AUC value varies between 0 and 1.



Perfect classification (AUC=1)

Area Under the Curve (AUC)

Diagonal: random classification (AUC=0.5)

True Positive 35	False Negative 39	Accuracy 0.778	Precision 0.745	Threshold 0.5	AUC 0.795
False Positive 12	True Negative 144	Recall 0.473	F1 Score 0.579		
Positive Label 1	Negative Label 0				

Evaluation metrics for regression

- **Mean absolute error (MAE)**

- Measures how close the predictions of the model are to the target value. This metric is calculated by averaging the subtraction of the predicted value and the actual value in absolute value over all training set examples. The lower the value the better.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |y_i - \widehat{y_i}|}{N}$$

Evaluation metrics for regression

- **Root mean squared error (RMSE)**

- This is very similar to MAE, but instead of taking the absolute value of the difference between predicted and target values, this metric squares the difference. After taking the average, the square root is computed. This metric should be used when you want to avoid large errors (because the error is squared). This and the previous metrics depend on the magnitude of the values you are predicting. If you are predicting small values (like values near 0), the error of this metric will be small, but if you are predicting values in the range of millions, the error will be bigger.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}}$$

Evaluation metrics for regression

- **Relative absolute error (RAE)**

- This is a normalized absolute error between expected and target values. It is very similar to MAE but with normalization. Because it is a relative metric, it goes from 0 to 1. It is better the nearer it comes to zero.

$$\mathrm{RAE} = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{\sum_{i=1}^{N} |y_i - \bar{y}|}$$

Evaluation metrics for regression

- **Relative squared error (RSE)**
- The normalized version of the RMSE. Because it is a relative measure, it goes from 0 to 1. It is better the nearer it comes to zero.

$$\mathrm{RSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Evaluation metrics for regression

- **Coefficient of determination (R2)**

- This represents the predictive power of the model as a value between 0 and 1. 0 means that the model is random (explains nothing); 1 means there is a perfect fit. However, caution should be used in interpreting R2 values, because R2 can be low even if the model is good, and R2 can be high even if the model is not good.

$$SS_{res} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

$$SS_{tot} = \sum_{i=1}^{N} (y_i - \bar{y})^2$$

$$R2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Evaluation metrics for clustering

Since there is not a unique correct answer for clustering problems (unsupervised learning), the results are more difficult to evaluate than results from a supervised learning model.

In order to evaluate the clusters, you must use the information returned by the evaluation of the model: number of examples per cluster, separation between clusters, and how close the samples of the center are of the cluster (commonly called centroid) to which they belong.

Using this information, different metrics can be computed.

Evaluation metrics for clustering

Simplified Silhouette

This metric is a measure of how similar an object is to its own cluster compared to other clusters. Values range from -1 to 1. Higher values are better.

Davies-Bouldin

This metric is defined as a ratio between the dispersion and the separation between each cluster. As you want slightly scattered and separated clusters, the lower values are better.

Dunn

As the other metrics, the aim of this metric is to identify sets of clusters compact and well separated. Generally, a higher value for this metric indicates better clustering.

Average deviation

The metric represents the average distance between each sample and its centroid. It decreases as the number of clusters increase, so it is not the best choice if you want to find the optimal number of clusters.

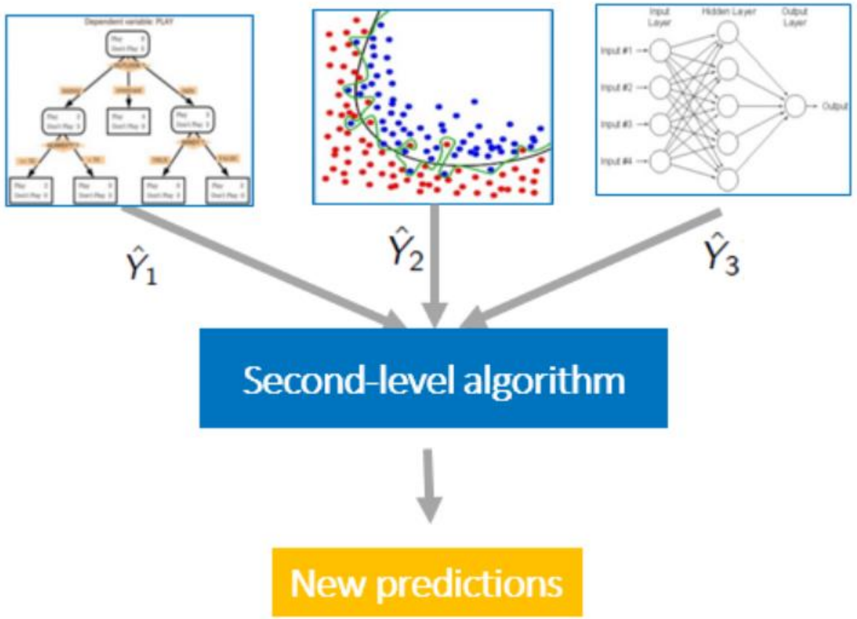
How to improve the quality of a model?

Ensemble Modeling

- Ensemble modeling is the process of running two or more related but different analytical models and then synthesizing the results into a single score or spread in order to improve the accuracy of predictive analytics and data mining applications.

- Certain models have an inherent ensemble, for example, some may be based on forests (a collection of trees).

Stacking



$\hat{Y}_1$ $\hat{Y}_2$ $\hat{Y}_3$

Second-level algorithm

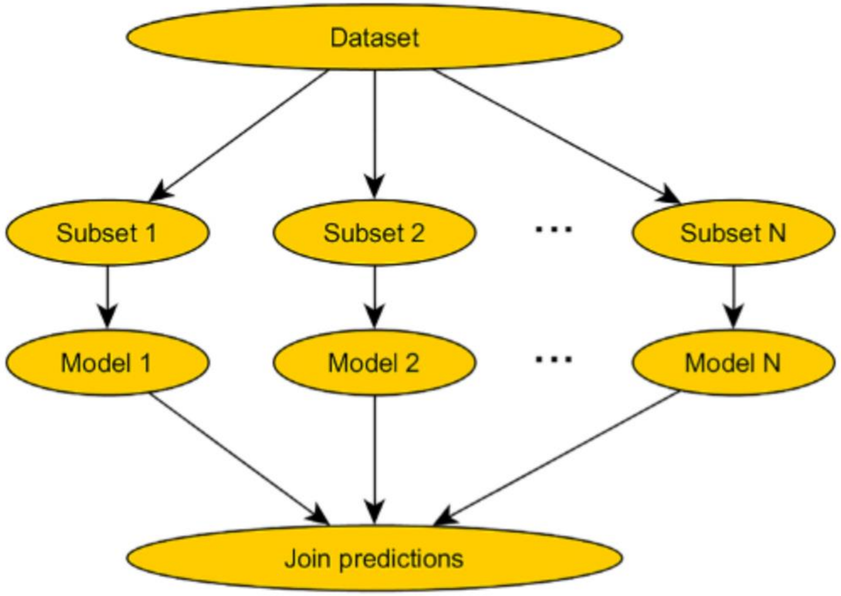
New predictions

The most direct form of ensemble is stacking, where the outputs of several models (base models) trained on the entire dataset are passed through another element that, in some way, weighs the results returned by the other models.

Using this method usually results in an improvement over the use of a single classifier.

The “Join predictions” node can be implemented, for example, calculating the arithmetic mean of each of the outputs from the previous models.

Bagging

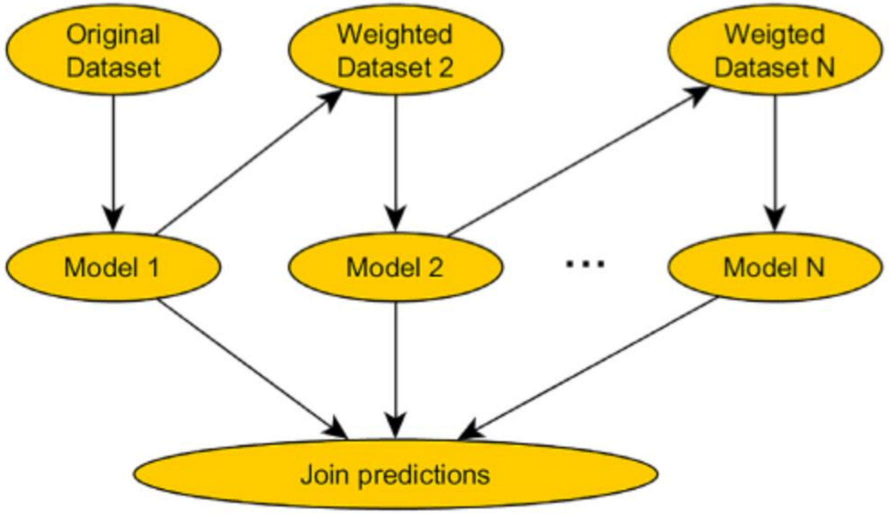


Bagging ensemble method

Other types of ensembles are those based on bagging (also referred to as bootstrap aggregation).

This technique divides the training set in **different subsets that can be overlapped** (sampling with replacement). A different model is trained on each of the subset and, like in stacking, the results of each classifier are combined. The Random Forest algorithm is very similar to a bagging of Trees, but with some additions.

Boosting



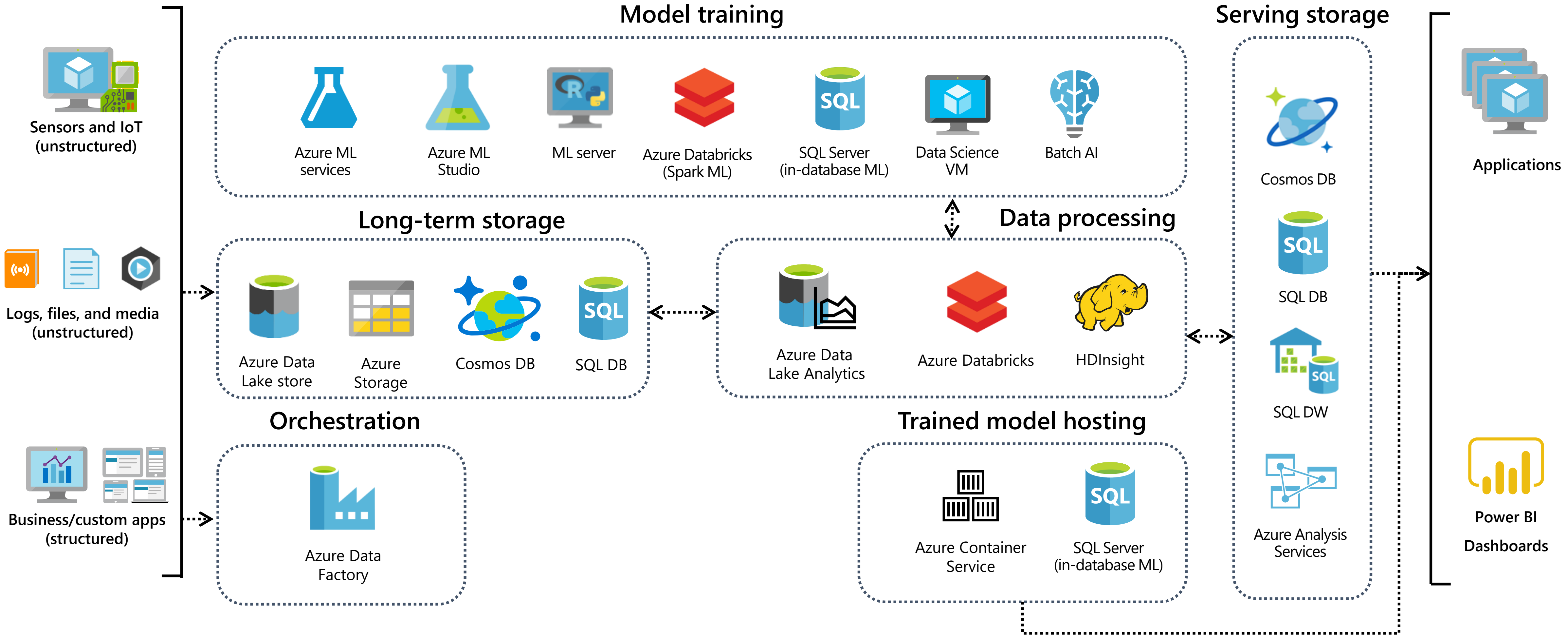
Boosting ensemble method

Another common type of ensemble is boosting. This type of ensemble trains a first model on the initial dataset.

Then, looking at the errors that this first model has made, it generates another dataset in which it **assigns more weight to those samples where the first model has failed**. In this way, the second model will pay more attention to the errors of the first model. The process is repeated N times and the outputs of the N models are combined.

Advanced analytics pattern in Azure

Data collection and understanding, modeling, and deployment



Microsoft