ReDI School of
Digital Integration

Crash course into supervised machine learning

/oana

We use tech to connect human potential and
opportunity with dignity & humility

Recap: Descriptive Analytics

Looks at data to examine, understand, and describe something that **has already happened**.

Looks at events in the past → Identify specific patterns
Commonly used visuals: pie charts, bar charts, tables, line graphs

Example:
How many sales occurred in the last quarter?

Recap: Predictive Analytics

Relies on historical data, past trends, and assumptions to answer questions about what **is likely to happen** in the future.

Techniques:
Regression analysis, forecasting, multivariate statistics, predictive modeling

Requirements for these techniques are large amounts of high-quality and historic data. They are also done using programming such as R and Python.

Example:
How will the next marketing campaign impact the customer engagement?

Recap: Prescriptive Analytics

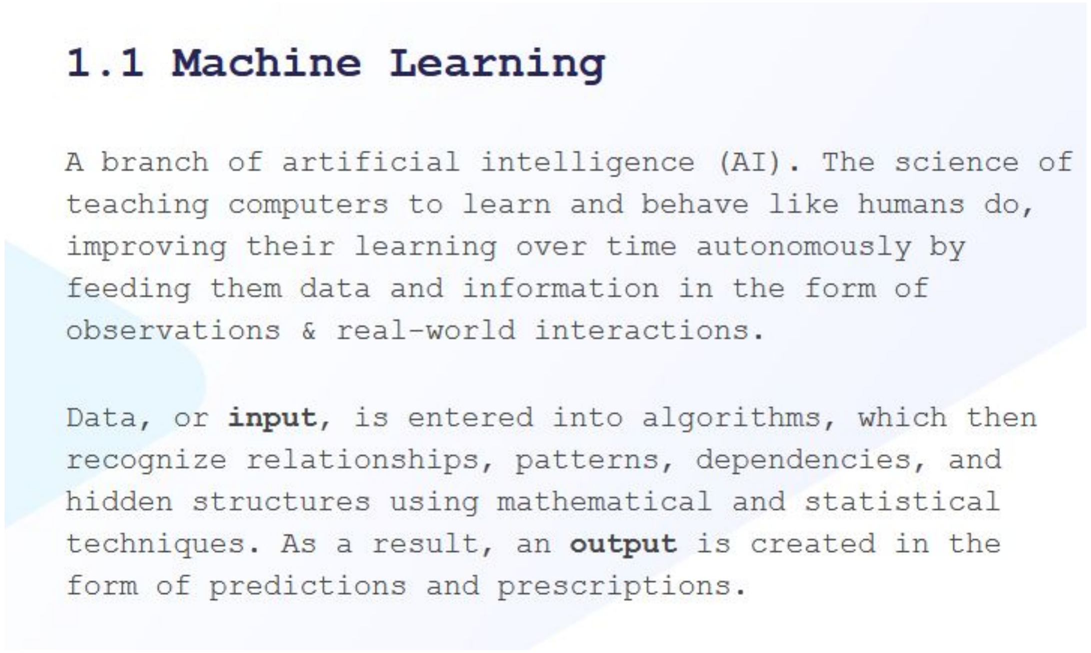
Identifies specific actions an individual or organization should take to **reach future targets or goals**.

Techniques:
Graph analysis, simulation, neural networks, machine learning

This is one of the most advanced level of analytics and it heavily depends on the accuracy of the descriptive, diagnostic, and predictive analysis. It also heavily relies on quality of data and the appropriate data architecture and expertise.

Organization will be able to **make decisions based on highly analyzed facts** rather than instinct.

Machine learning /vs AI



1.1 Machine Learning

A branch of artificial intelligence (AI). The science of teaching computers to learn and behave like humans do, improving their learning over time autonomously by feeding them data and information in the form of observations & real-world interactions.

Data, or **input**, is entered into algorithms, which then recognize relationships, patterns, dependencies, and hidden structures using mathematical and statistical techniques. As a result, an **output** is created in the form of predictions and prescriptions.

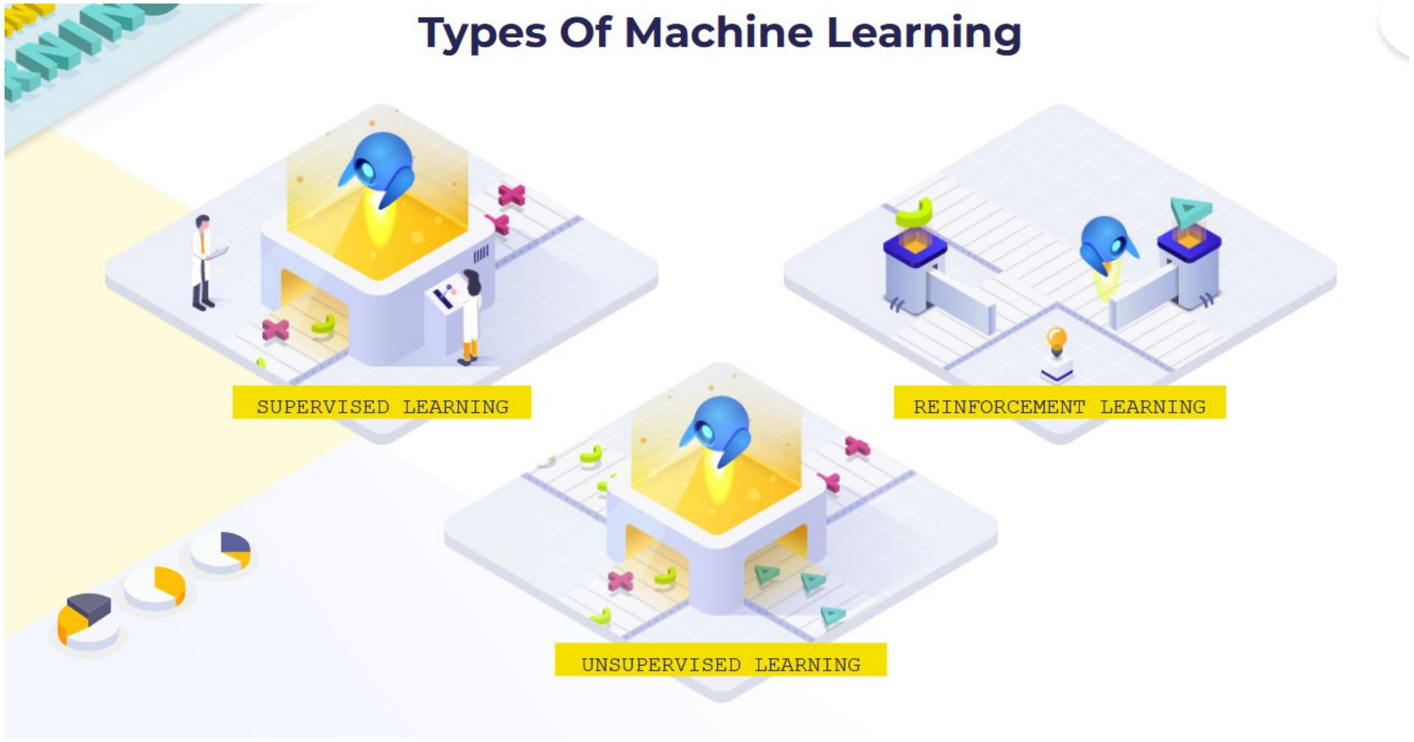
Machine Learning is...

-
  ... learning from data
  ... no explicit programming
  ... discovering hidden patterns
  ... essential for data-driven decisions

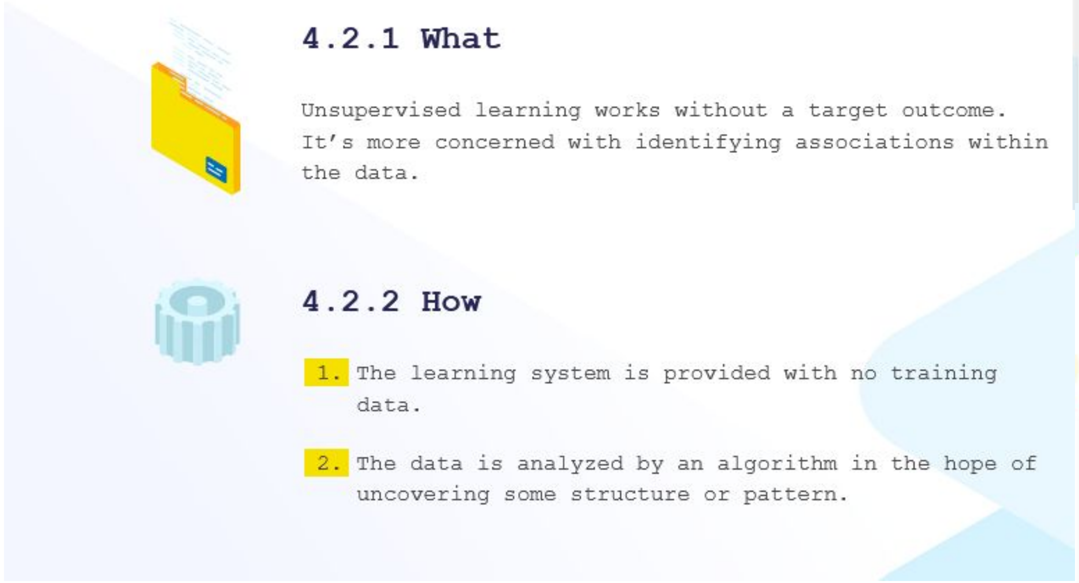
- e.g.
- Credit card fraud detection
- Handwritten digit recognition
- Recommendations on websites

Types of machine learning

Types of ml


Types Of Machine Learning
SUPERVISED LEARNING
REINFORCEMENT LEARNING
UNSUPERVISED LEARNING

Unsupervised ML

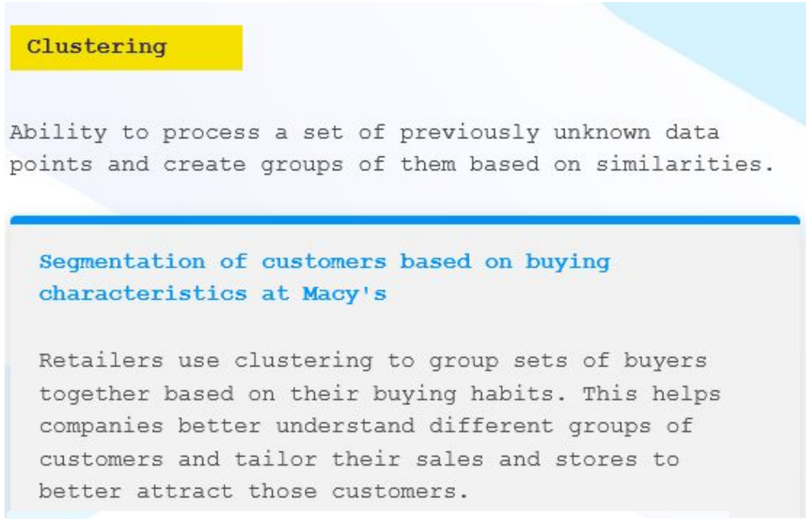


4.2.1 What

Unsupervised learning works without a target outcome. It’s more concerned with identifying associations within the data.

4.2.2 How

1. The learning system is provided with no training data.
2. The data is analyzed by an algorithm in the hope of uncovering some structure or pattern.

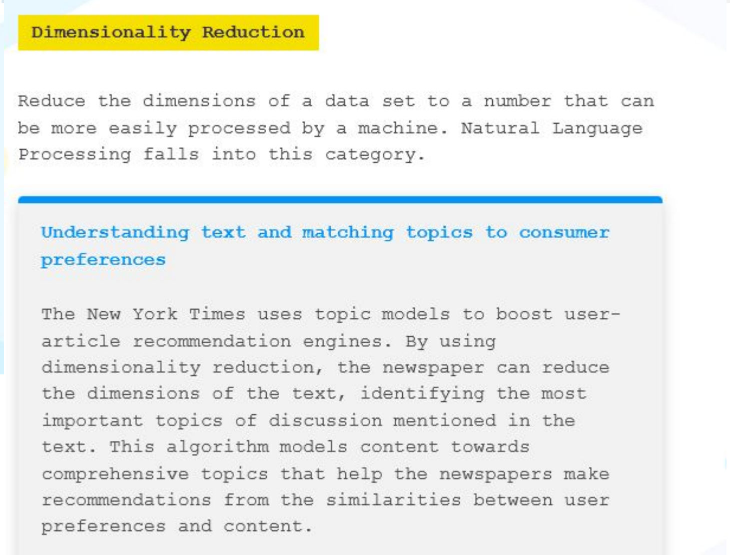


Clustering

Ability to process a set of previously unknown data points and create groups of them based on similarities.

Segmentation of customers based on buying characteristics at Macy's

Retailers use clustering to group sets of buyers together based on their buying habits. This helps companies better understand different groups of customers and tailor their sales and stores to better attract those customers.



Dimensionality Reduction

Reduce the dimensions of a data set to a number that can be more easily processed by a machine. Natural Language Processing falls into this category.

Understanding text and matching topics to consumer preferences

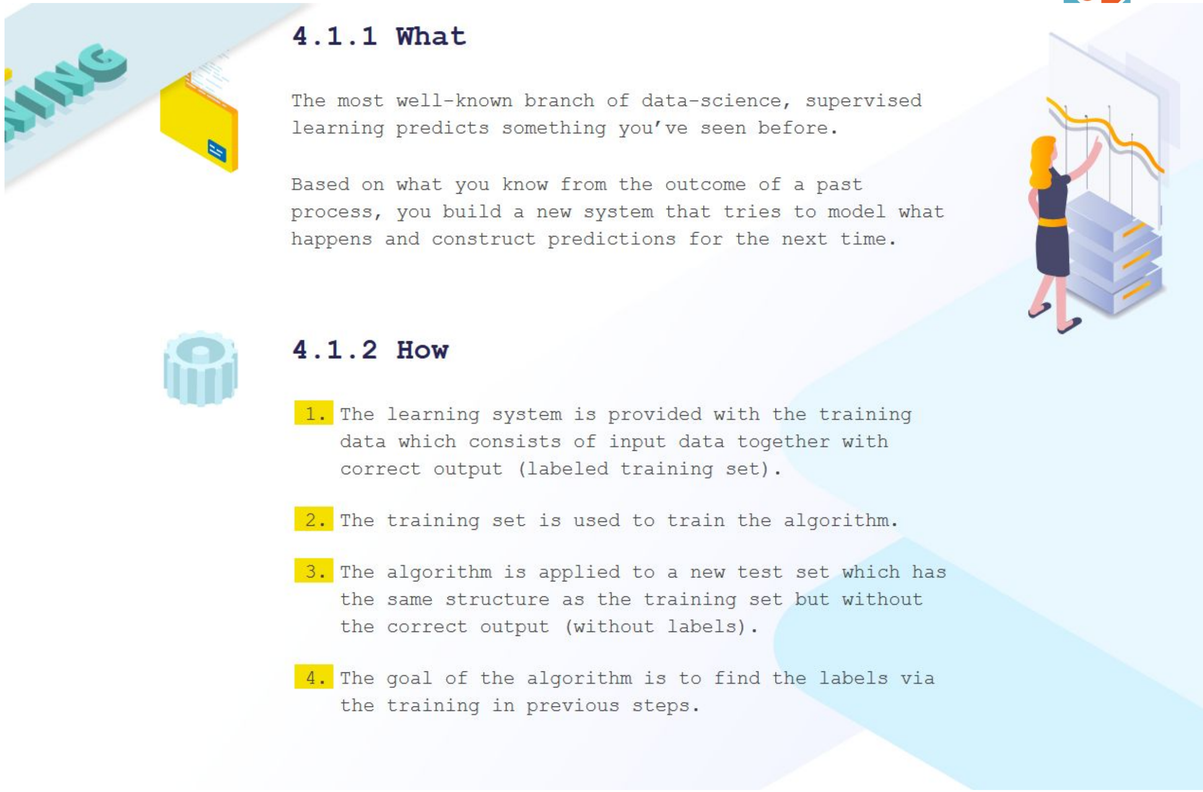
The New York Times uses topic models to boost user-article recommendation engines. By using dimensionality reduction, the newspaper can reduce the dimensions of the text, identifying the most important topics of discussion mentioned in the text. This algorithm models content towards comprehensive topics that help the newspapers make recommendations from the similarities between user preferences and content.

ReDI School of
Digital Integration

Supervised Machine Learning

We use tech to connect human potential and opportunity with dignity & humility

Supervised ML



4.1.1 What

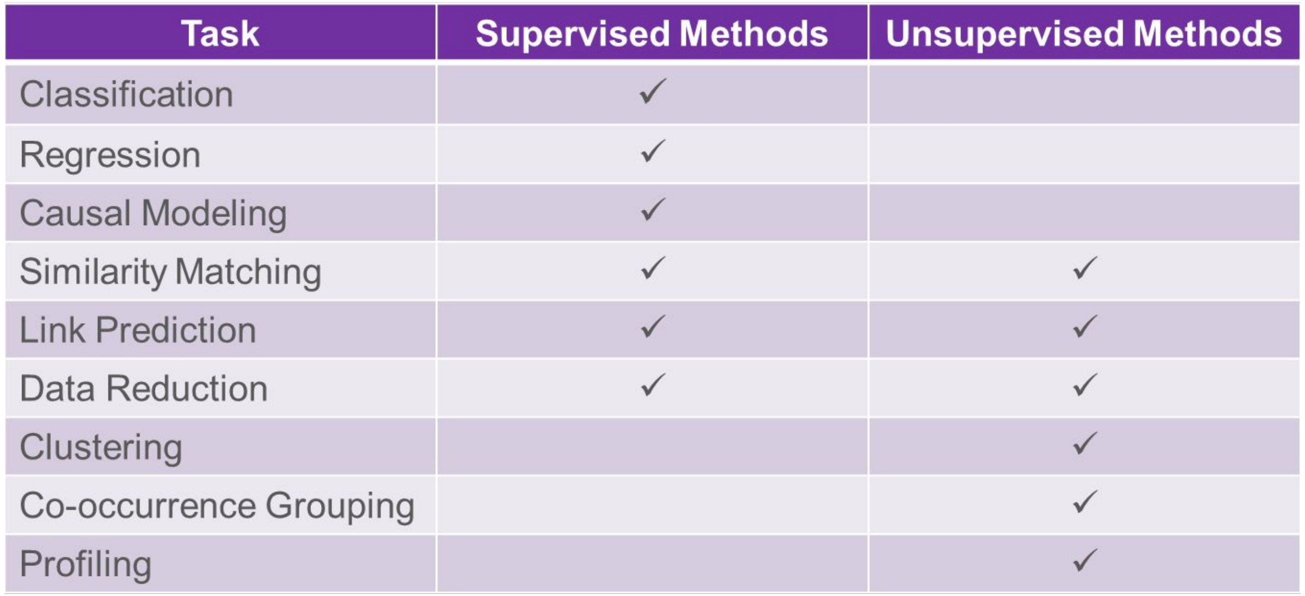
The most well-known branch of data-science, supervised learning predicts something you've seen before.

Based on what you know from the outcome of a past process, you build a new system that tries to model what happens and construct predictions for the next time.

4.1.2 How

1. The learning system is provided with the training data which consists of input data together with correct output (labeled training set).
2. The training set is used to train the algorithm.
3. The algorithm is applied to a new test set which has the same structure as the training set but without the correct output (without labels).
4. The goal of the algorithm is to find the labels via the training in previous steps.

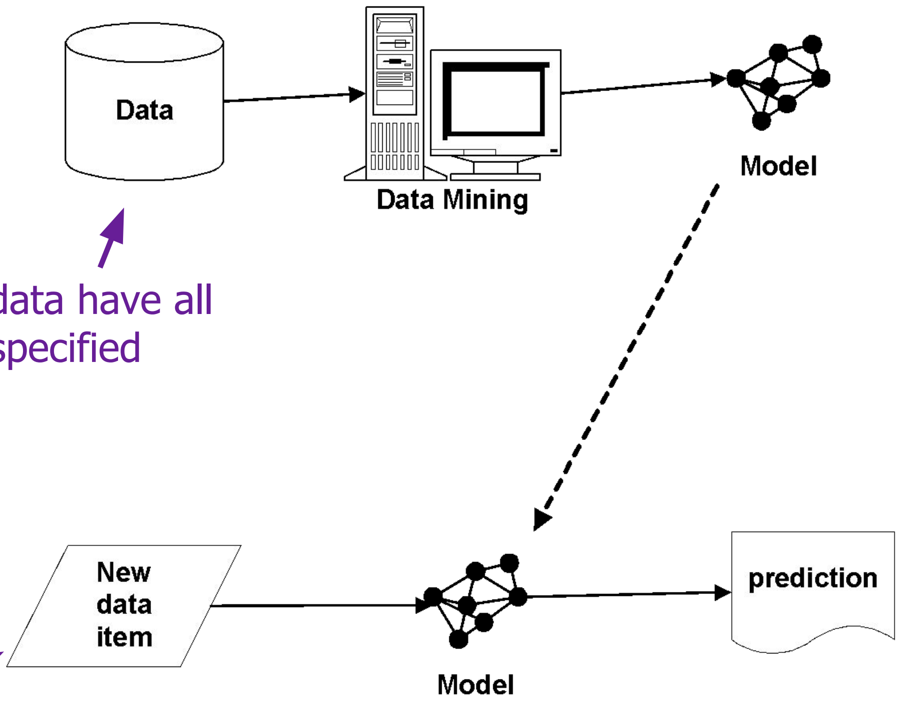
Common Data analytics Tasks



Task	Supervised Methods	Unsupervised Methods
Classification	✓	
Regression	✓	
Causal Modeling	✓	
Similarity Matching	✓	✓
Link Prediction	✓	✓
Data Reduction	✓	✓
Clustering		✓
Co-occurrence Grouping		✓
Profiling		✓

Data Mining versus Use of the Model

"Supervised" modeling:



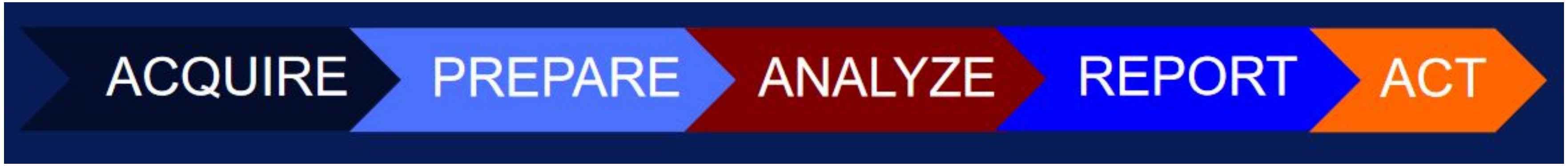
"Training" data have all values specified

New data item has some value unknown (e.g., will she leave?)

Supervised versus Unsupervised Methods

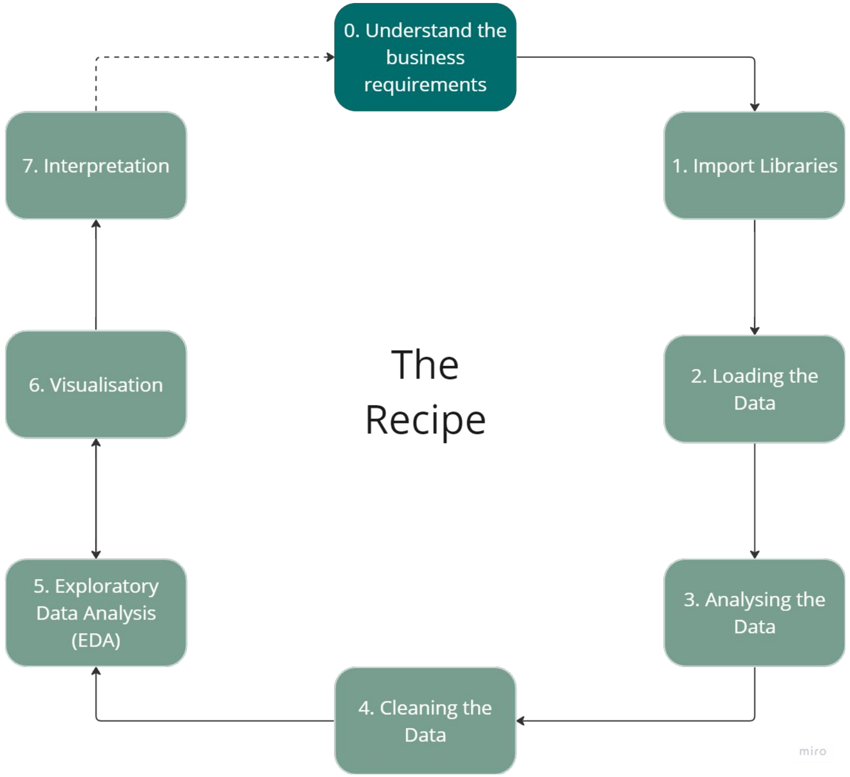
- “Do our customers naturally fall into different groups?”
    - No guarantee that the results are meaningful or will be useful for any particular purpose
- “Can we find groups of customers who have particularly high likelihoods of canceling their service soon after contracts expire?”
    - **A specific purpose**
    - Much more useful results (usually)
    - Different techniques
    - **Requires data on the target**
        - The individual’s label

Machine learning pipeline/recipe (works also for unsupervised)



Machine learning pipeline (recipe)

The recipe for data analysis



Step 6(2): Analyze Data

- Select analytical techniques
  - Supervised, unsupervised
    - Supervised: Based on your target variable: classifiers or regressions
      - Split the data
    - ~~Unsupervised: cluster analysis, association analysis~~
      - ~~Doesn’t require data split~~
- Build models using train data
- Assess results using test data

What is a model?

A *simplified** *representation* of reality created for a *specific purpose*

**based on some assumptions*

- *Examples: map, prototype, Black-Scholes model, etc.*
- *Data analytics example:*

  “formula” for predicting probability of customer attrition at contract expiration

  → “classification model” or “class-probability estimation model”

Classification

- Goal: given input, predict category

- Classify tumor as benign or malignant
  Predict if it will rain tomorrow
  Determine if loan application is high-,medium-, or low-risk
  Identify sentiment as positive, negative, or neutral

- Examples: LDA, SVM, decision tree, KNN, naïve bayes

Regression

- Goal: Given input variables, predict numeric value

- Forecast high temperature for next day
  Estimate average house price for a region
  Determine demand for a new product
  Predict power usage

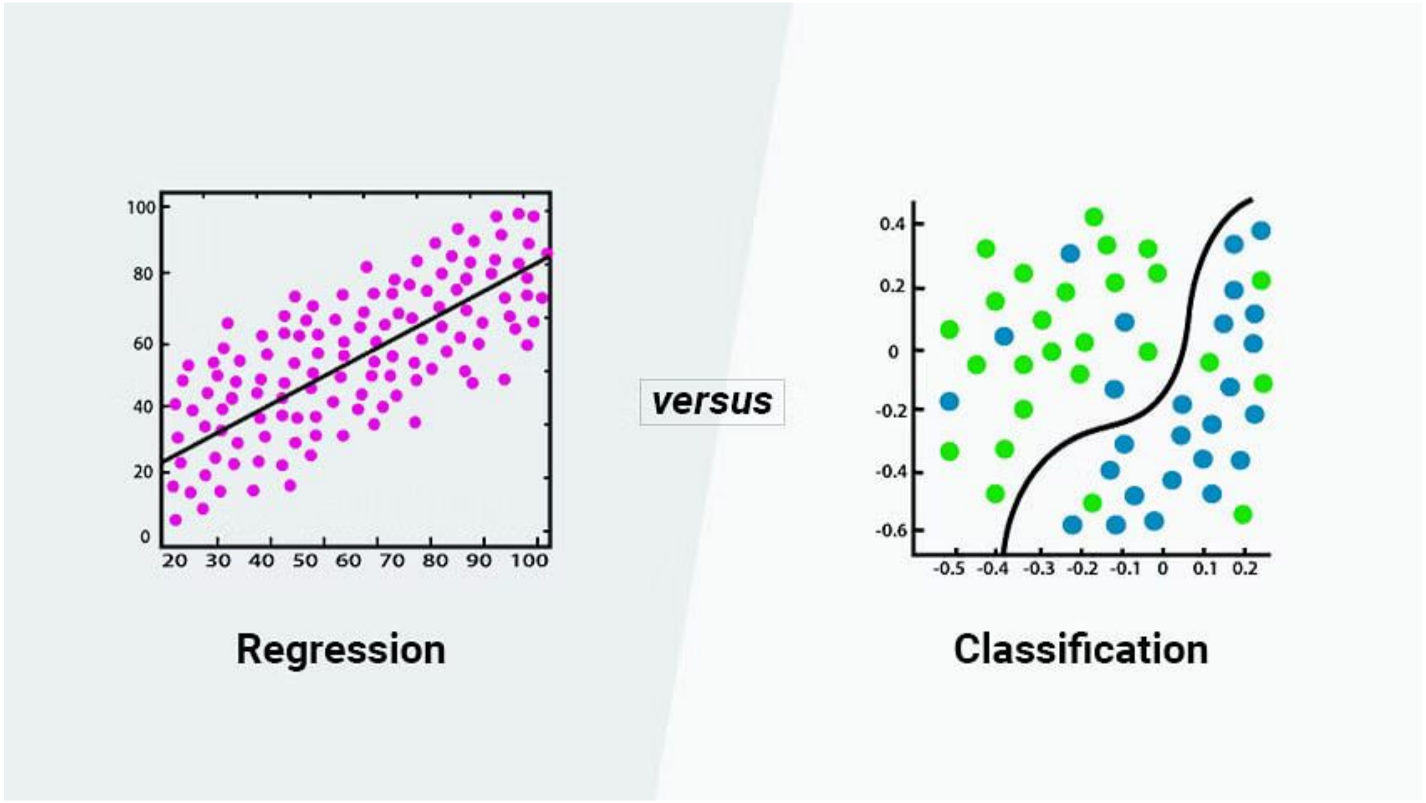
- Examples: linear regression, non linear regression, decision tree

Build models

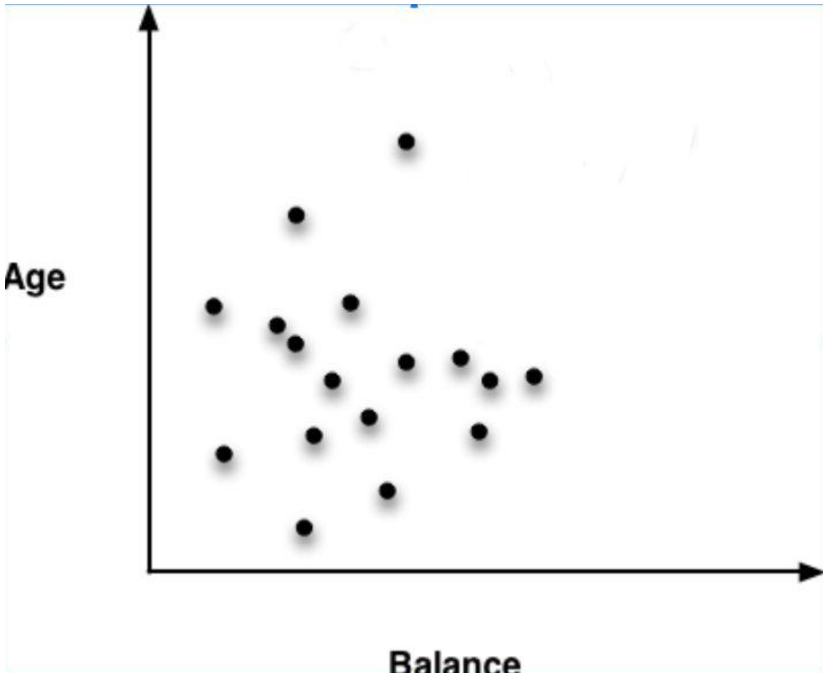
- 1. models – where do they come from
- 2. how to choose
    - loss functions
    - Explainability: do you need to explain to stakeholders?
    - Accuracy

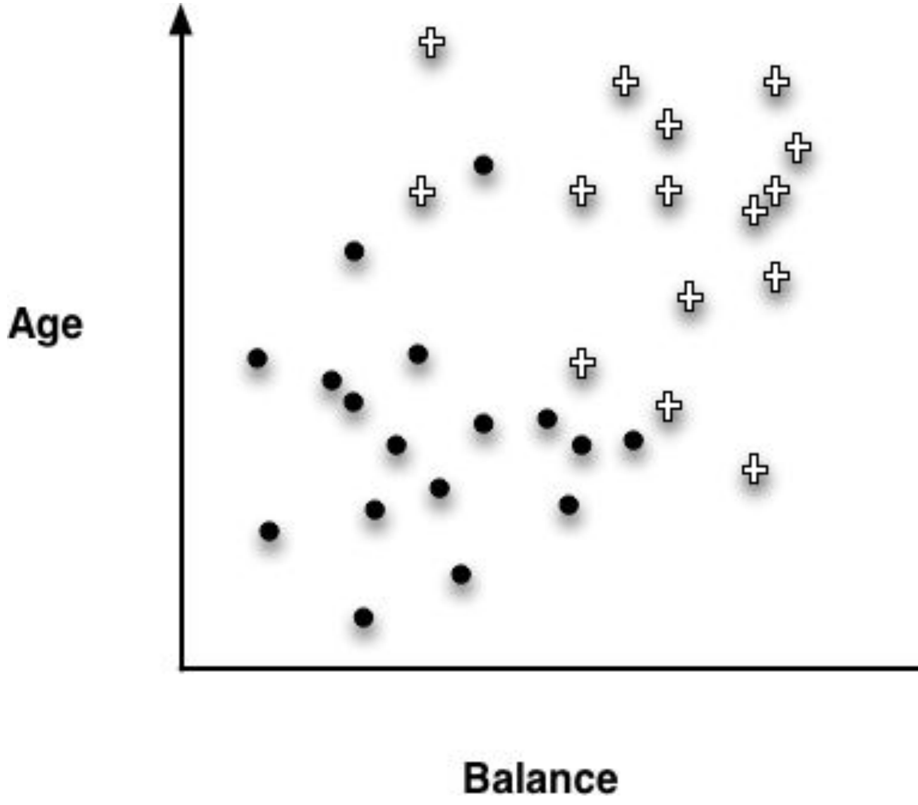
Models, where do they come from

Regression vs classification

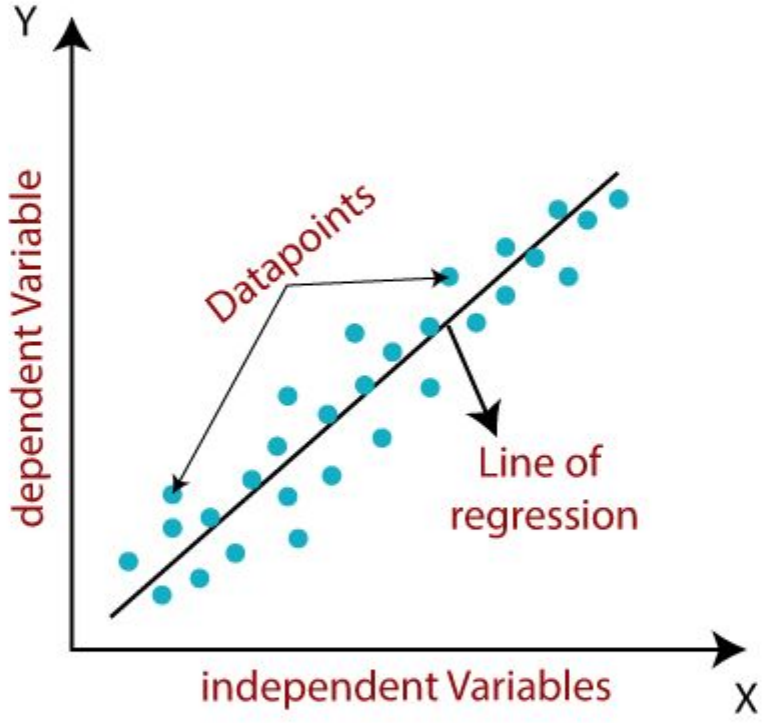


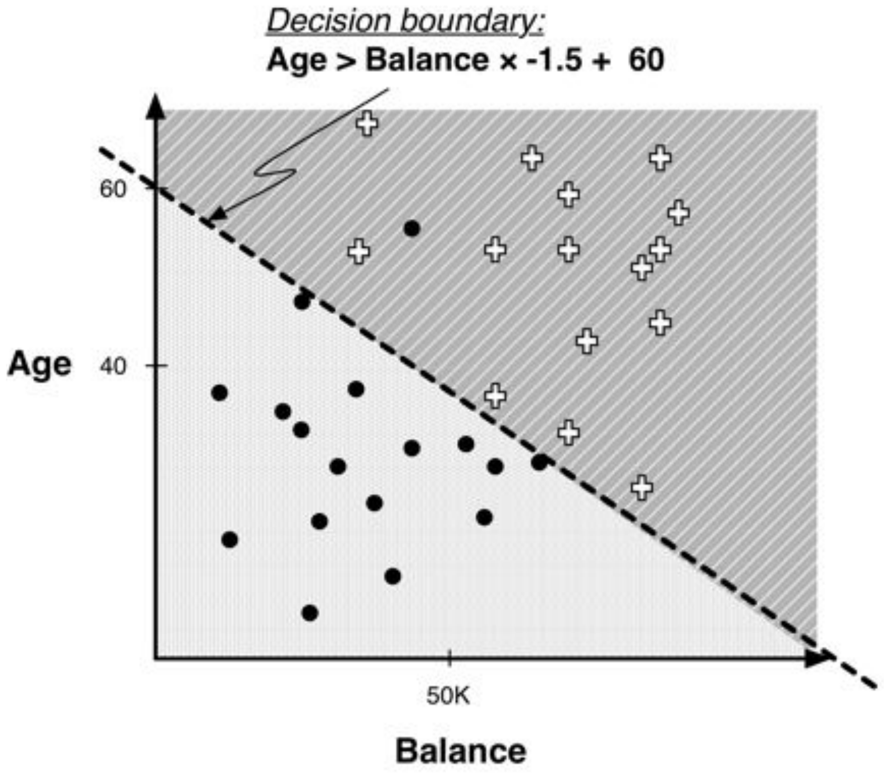
Instance Space regression vs linear classifier





Regression vs linear classfier





Example of regression Function

Linear regression:
$Y = 1.0 \times Age - 1.5 \times Balance + 60$

- We now have a parameterized model: the weights of the linear function are the parameters

- The weights are often loosely interpreted as importance indicators of the features

Example of Classification Function

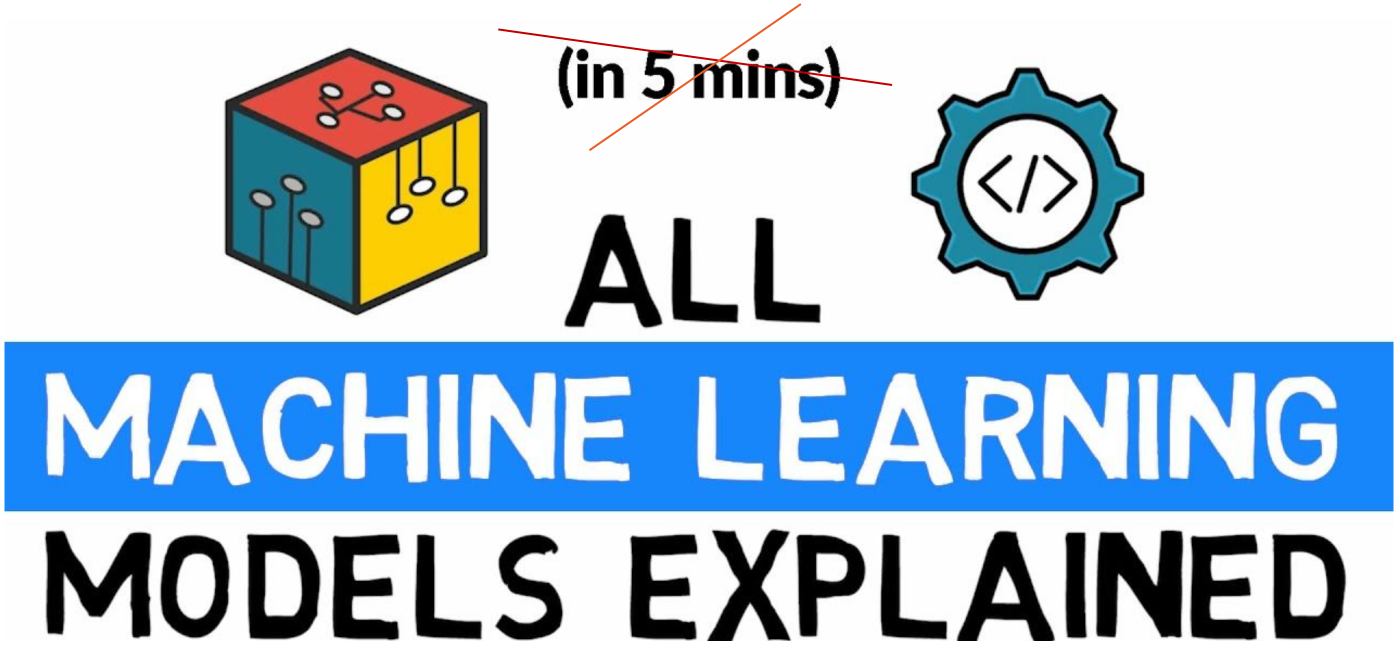
Linear discriminant:

$$class(x) = \begin{cases} + \text{ if } 1.0 \times Age - 1.5 \times Balance + 60 > 0 \\ \bullet \text{ if } 1.0 \times Age - 1.5 \times Balance + 60 \leq 0 \end{cases}$$

We now have a parameterized model: the weights of the linear function are the parameters
• The weights are often loosely interpreted as importance indicators of the features
• A different sort of multivariate supervised segmentation
• The difference from DTs is that the method for taking multiple attributes into account is to create a mathematical function of them

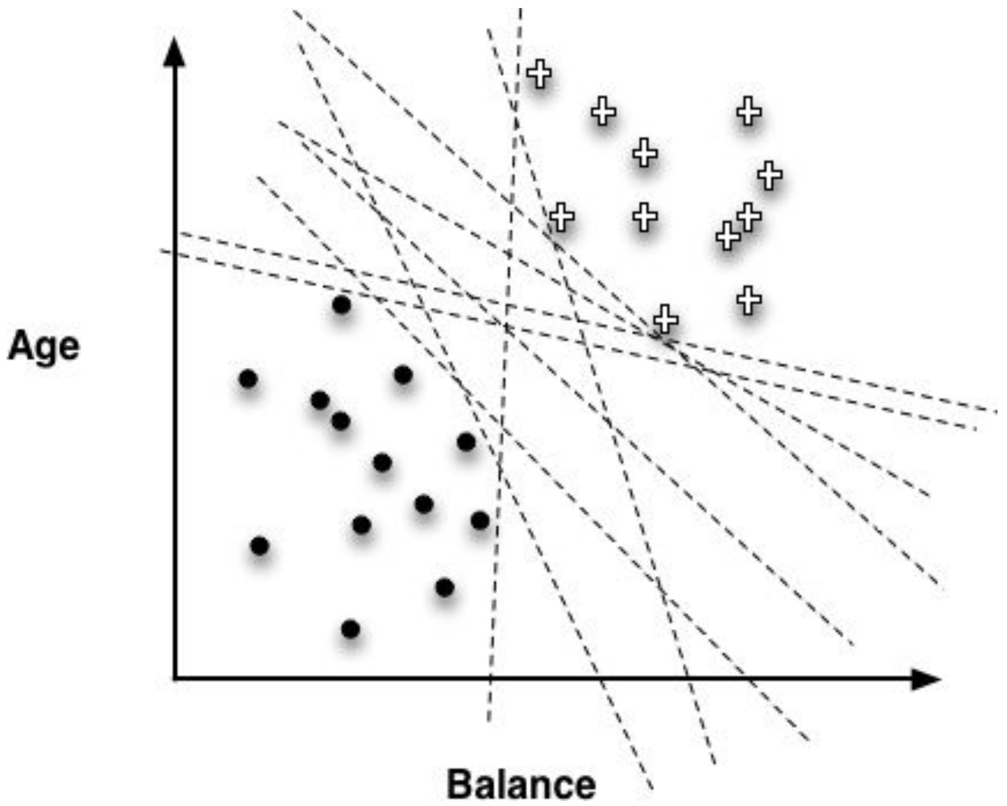
Other models

There are MANY MANY more, e.g. random forests, decision trees, neural networks etc.

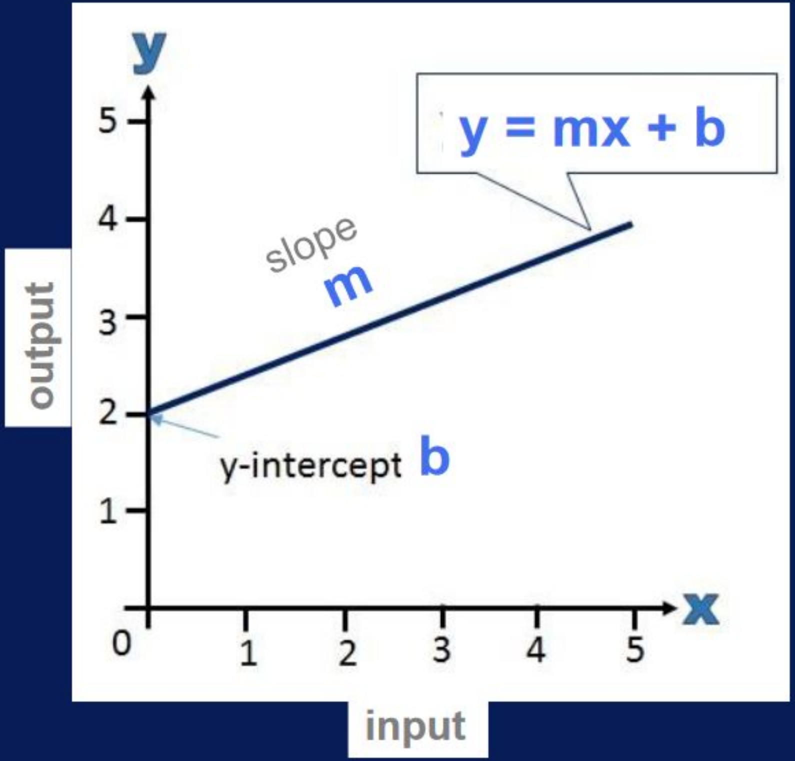


Models, how to choose

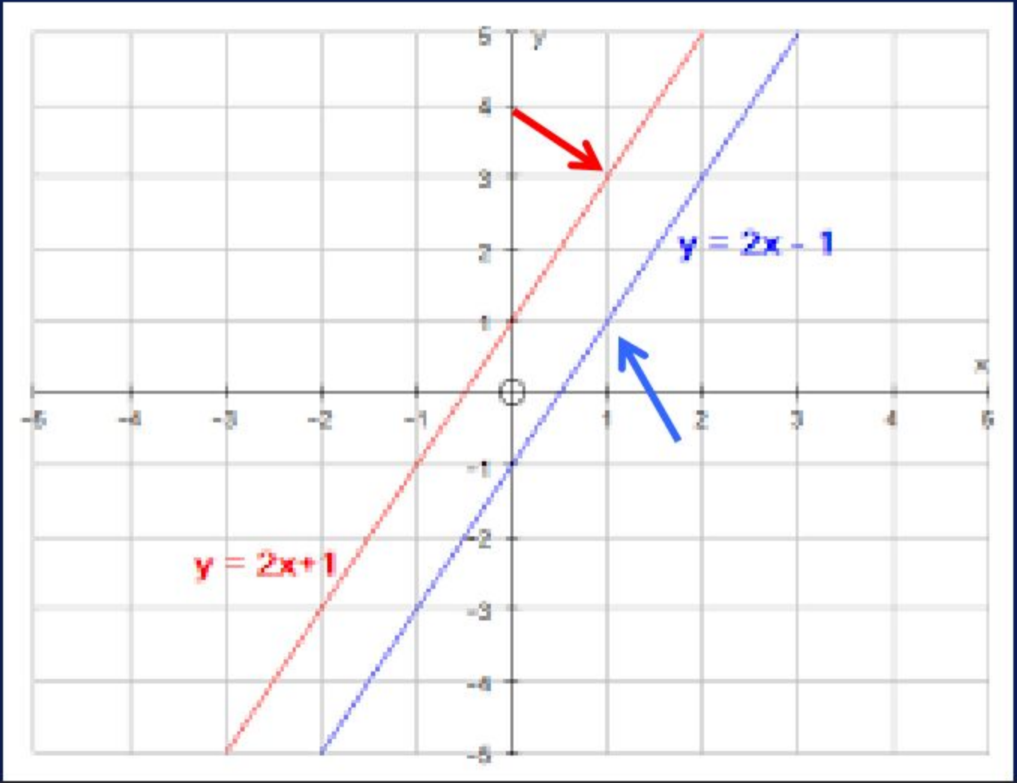
Choosing the “best” line



Example of Model



Adjusting Model Parameters

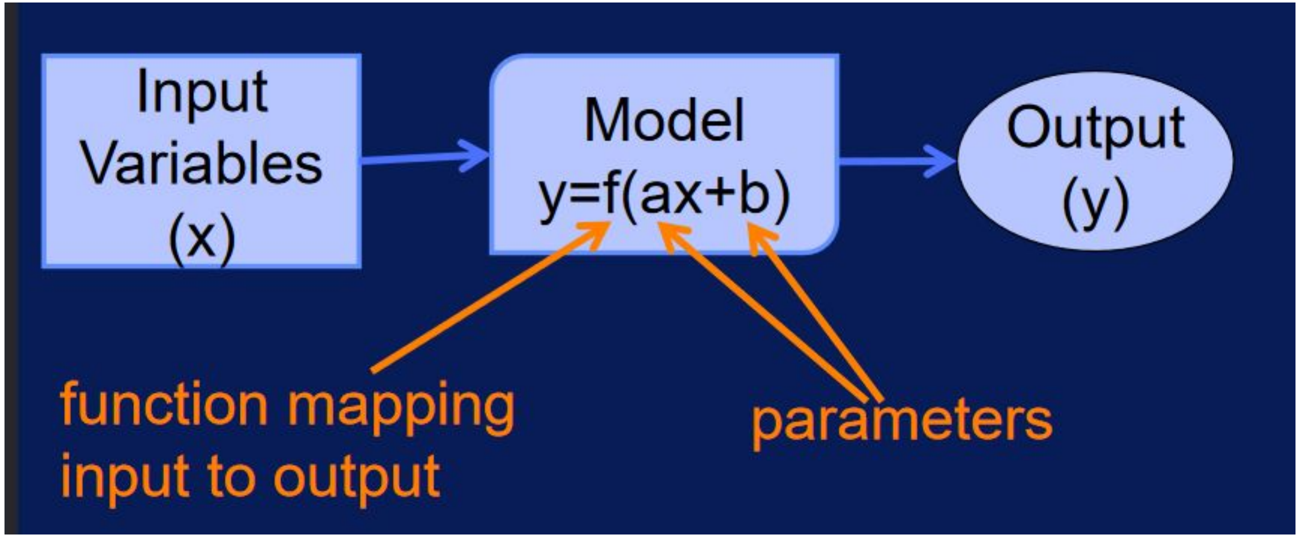


slope m = 2
y-intercept b = -1
x=1 => y=2*1-1=1

slope m = 2
y-intercept b = +1
x=1 => y=2*1+1= 3

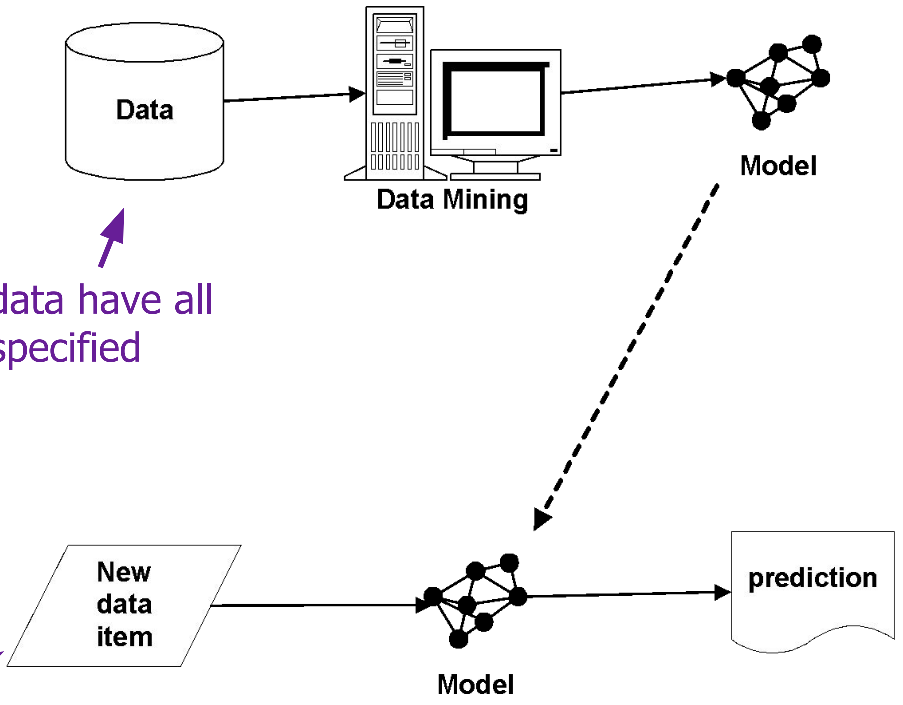
Building Machine Learning Model

- Model parameters are adjusted during model training to change input-output mapping via loss functions



Data Mining versus Use of the Model

"Supervised" modeling:



"Training" data have all values specified

New data item has some value unknown (e.g., will she leave?)

Phew. We made it!

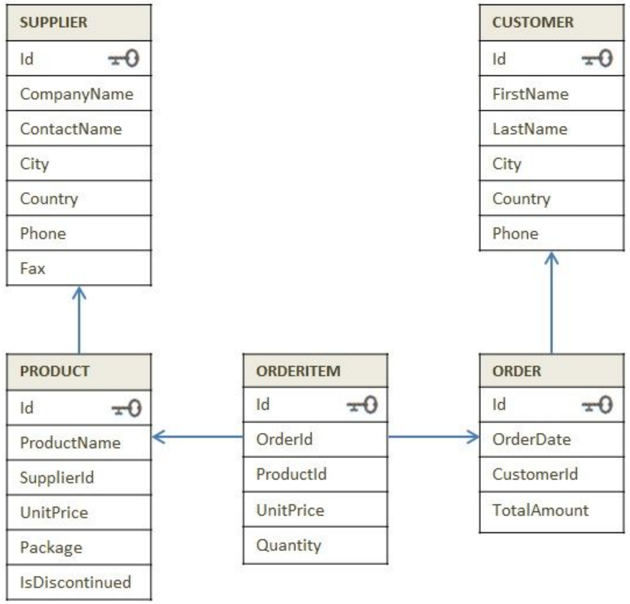
Data storage solutions

Data storage solutions

- Data storage solutions are essential for managing and organizing large amounts of data.
- There are several types of data storage solutions, including SQL databases, data silos, data warehouses, cloud storage, and file storage.
- Choosing the right data storage solution depends on factors such as the size and complexity of the data, the need for scalability, and the level of accessibility required.

SQL Databases

- SQL (Structured Query Language) is a programming language used to manage and manipulate relational databases.
- SQL databases store data in tables with rows and columns, allowing for easy querying and retrieval of data.
- SQL databases are widely used in a variety of applications, including business operations, customer relationship management, and e-commerce.



Data Silos vs. Data Warehouses + File vs. cloud storage

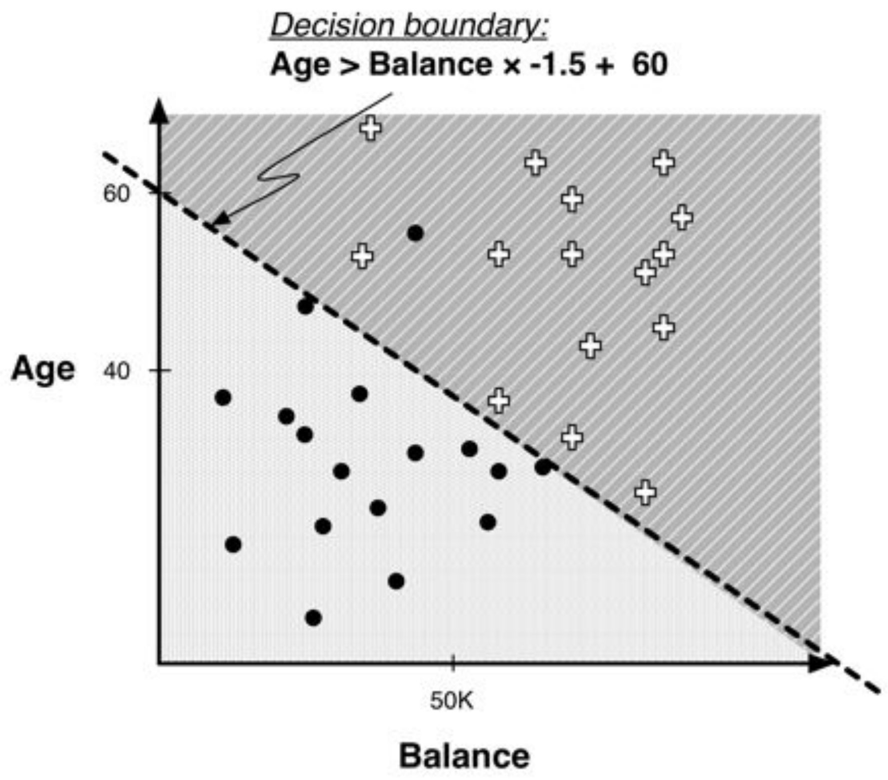
- A data silo is a repository of fixed data that remains under the control of one department and is isolated from the rest of the organization.
- A data warehouse is a large, centralized repository of data that is used to support business intelligence and decision-making activities.
- Unlike data silos, data warehouses integrate data from multiple sources and make it accessible to users across the organization.
- Data warehouses also support advanced analytics and reporting capabilities.

- Cloud storage involves storing data on remote servers that can be accessed over the internet.
- File storage involves storing data on local servers or devices.
- Cloud storage offers several advantages over file storage, including scalability, accessibility, and cost-effectiveness.
- Cloud storage also provides greater flexibility and can be easily integrated with other cloud-based services.

Phew. We made it x 2 :P!

Evaluating classifiers

Linear Classifier



Evaluating Classifiers: The Confusion Matrix

- A confusion matrix for a problem involving n classes is an $n \times n$ matrix,
  - with the columns labeled with actual classes and the rows labeled with predicted classes
- It separates out the decisions made by the classifier,
  - making explicit how one class is being confused for another

	p	n
Y	True Positives	False Positives
N	False Negatives	True Negatives

- The errors of the classifier are the false positives and false negatives

Evaluating Classifiers: Plain Accuracy

-
- $$\text{Accuracy} = \frac{\text{Number of correct decisions made}}{\text{Total number of decisions made}}$$
-

$$= 1 - \text{error rate}$$

- *Too* simplistic..
- y?
- Because often times, high accuracy is easy to get, but at what cost? 99.99% of people making transactions are NOT committing fraud. So if your algorithm fails to identify the one, it will still have 99.99% accuracy..

Other Evaluation Metrics

- Precision $= \frac{TP}{TP+FP}$ (positive predicted value)

Precision evaluates the fraction of correct classified instances among the ones classified as positive

- Recall $= \frac{TP}{TP+FN}$(sensitivity)

Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made.

- $\mathrm{F-measure} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$

  - true positive (TP) A test result that correctly indicates the presence of a condition or characteristic
  - true negative (TN) A test result that correctly indicates the absence of a condition or characteristic
  - false positive (FP) A test result which wrongly indicates that a particular condition or attribute is present
  - false negative (FN) A test result which wrongly indicates that a particular condition or attribute is absent

For those who want to play with the python scripts

We won’t go through these slides, and they are roughly the same as the text in the jupyter notebook, just with a bit more visuals

We need to split our data before creating a model

Conventions

Put the data you want to predict (the label) in a `DataFrame` called `y`

Put the data that influence the label (the features) in a `DataFrame` called `X`

```python
y = delays_df.loc[:,['ARR_DELAY']]
X = delays_df.loc[:,['DISTANCE', 'CRS_ELAPSED_TIME']]
```

index	DISTANCE	CRS_ELAPSED_TIME	ARR_DELAY
0	1670	225	7
1	580	105	-4
2	425	96	5
…	…	…	…
299645	1500	210	73

Features

Label

DataFrame X

index	DISTANCE	CRS_ELAPSED_TIME
0	1670	225
1	580	105
…	…	…
299645	1500	210

DataFrame y

index	ARR_DELAY
0	7
1	-4
…	…
299645	74

You need to split your data into training and test data sets before you train your model

DataFrame X

index	DISTANCE	CRS_ELAPSED_TIME
0	1670	225
1	580	105
2	425	96
…	…	…
299645	1500	210

train will be used to train the model

test will be used to test the model

Typically put aside 20 to 30% of your records for testing

X_train

index	DISTANCE	CRS_ELAPSED_TIME
0	1670	225
2	425	96
…	…	…
299645	1500	210

X_test

index	DISTANCE	CRS_ELAPSED_TIME
1	580	105
17	750	130
…	…	…
299643	425	100

scikit-learn

Open source library for preprocessing data and model training

`train_test_split` to split each of the X and y DataFrames into training and test DataFrames

```
from sklearn.model_selection import train_test_split
```

Now we can call train_test_split

```
X_train, X_test, y_train, y_test =
   train_test_split(X, y, test_size=0.3, random_state=42)
```

- **X_train, X_test, y_train, y_test** – names of DataFrames to create to store the training and test data
- **X** – name of the DataFrame containing your features
- **y** – name of the DataFrame containing your labels
- **test_size** – what portion of the rows to put into the test data frames
- **random_state** – sets the seed for random numbers - every time the code runs the same "random" sets of rows will be returned

What started in one DataFrame, is now split into four DataFrames

X_train

index	DISTANCE	CRS_ELAPSED_TIME
0	1670	225
1	580	105
…	…	…
299645	1500	210

y_train

index	ARR_DELAY
0	7
1	-4
…	…
299645	74

X_test

index	DISTANCE	CRS_ELAPSED_TIME
2	425	96
17	750	130
…	…	…
299643	425	100

y_test

index	ARR_DELAY
2	5
17	-1
…	…
299645	11

index	DISTANCE	CRS_ELAPSED_TIME	ARR_DELAY
0	1670	225	7
1	580	105	-4
2	425	96	5
…	…	…	…
299645	1500	210	73

Once your data is prepared you can use it to train a model

X_train

index	DISTANCE	CRS_ELAPSED_TIME
0	1670	225
1	580	105
…	…	…
299645	1500	210

y_train

index	ARR_DELAY
0	7
1	-4
…	…
299645	74

X_test

index	DISTANCE	CRS_ELAPSED_TIME
2	425	96
17	750	130
…	…	…
299643	425	100

y_test

index	ARR_DELAY
2	5
17	-1
…	…
299645	11

There are different libraries and classes you can use to train a model

The **`scikit-learn`** library contains a number of different classes to train different types of models

The **`LinearRegression`** class fits a linear model

Call the **`fit`** method to fit your data to a linear model which can be used to predict the outcomes for new data

The code to train a model usually follows the same pattern

import the desired class from the library

```
from sklearn.linear_model import LinearRegression
```

Create an instance of the class

```
regressor = LinearRegression()
```

use the **`fit`** method to train the model using your training data

```
regressor.fit(X_train, y_train)
```

Once your model is trained you can test it using your test data

X_train

index	DISTANCE	CRS_ELAPSED_TIME
0	1670	225
1	580	105
…	…	…
299645	1500	210

y_train

index	ARR_DELAY
0	7
1	-4
…	…
299645	74

X_test

index	DISTANCE	CRS_ELAPSED_TIME
2	425	96
17	750	130
…	…	…
299643	425	100

y_test

index	ARR_DELAY
2	5
17	-1
…	…
299645	11

`predict` returns the values the trained model predicts for new data

Predict values for the test data in X_test

Compare the predicted values to the actual values in y_test to see how well your model performs

```
y_pred = regressor.predict(X_test)
```

y_test

index	ARR_DELAY
291483	-5.0
164800	20.0
…	…
60706	-6.0

y_pred

ARR_DELAY
3.48
5.80
…
6.08

It's impractical to visually compare predicted and actual values across thousands of rows

y_test   y_pred

index	ACTUAL_ARR_DELAY	PREDICTED_ ARR_DELAY
291483	-5.0	3.48
164800	20.0	5.80
…	…	…
60706	-6.0	6.08

Instead we use standard calculations to get a sense of overall accuracy

Mean Squared Error (MSE) is the average error

MSE = mean((actual – predicted)**2)

You could write a python function to loop through each row, and perform this calculation

OR

You could use `mean_squared_error` from the `scikit-learn` library

```
from sklearn import metrics
metrics.mean_squared_error(y_test, y_pred)
```

Root Mean Squared Error (RMSE) is the square root of the mean squared error

$\text{RMSE} = \sqrt{\text{MSE}}$

scikit-learn doesn't have a method for RMSE but…

`scikit-learn` has the `mean_squared_error` method to calculate MSE

numpy has a `sqrt` method to calculate square root

```
from sklearn import metrics
import numpy as np
np.sqrt(metrics.mean_squared_error(y_test, y_pred))
```

Different models have different metrics to measure accuracy

numpy provides standard mathematical functions

`scikit-learn` provides a number of functions for assessing prediction error

Together they allow us to measure model accuracy