LIVE ONLINE TRAINING

Business Analytics With Python Bootcamp

Week 5: Predictive Business Analytics with Python

Agenda

1. **Recap & Intro (15 minutes)**

2. **Inferential statistics (45 minutes)**
   - Presentation: Introduction to inferential statistics
   - Interactive labs
   - Q&A / Break

3. **Machine learning (60 minutes)**
   - Presentation: Introduction to machine learning
   - Interactive labs
   - Q&A / Break

4. **Model performance (60 minutes)**
   - Presentation: Evaluating machine learning models
   - Interactive labs
   - Q&A / Break

5. **Time series forecasting (30 minutes)**
   - Presentation: Introduction to time series forecasting
   - Interactive lab: Forecast Time Series with ARIMA Models

6. **Wrap-Up / Outlook (5 minutes)**

Recap

Bootcamp overview

Last week

Week 1 Jan 31	Week 2 Feb 7	Week 3 Feb 14	Week 4 Feb 21	Week 5 Feb 28	Week 6 Mar 7
Business Analytics Fundamentals	**Data Analysis With Python Fundamentals**	**Descriptive Business Analytics With Python**	**Diagnostic Business Analytics With Python**	**Predictive Business Analytics With Python**	**Prescriptive Business Analytics With Python**
Goal: Get the theoretical foundations for doing business analytics	**Goal:** Get the practical foundations for using Python for data analysis	**Goal:** Learn how to explore & describe your data (What happened?)	**Goal**: Learn how to get insights from your data (Why did it happen?)	**Goal:** Learn how to predict the future, given historical data (What will happen?)	**Goal:** Suggest better actions based on data (What to do?)

NOTE: With today's registration, you'll be signed up for all six sessions. Although you can attend any of the sessions individually, it's recommended participating in all six weeks.

Quiz time!

What is the best way to avoid selection bias in business analytics?

A. Collect as much data as possible.

B. Use data only from sources that are convenient or easily accessible.

C. Ensure that the data being analyzed is representative of the population of interest.

D. Only analyze data that supports the desired outcome

Which of the following is an example of survivorship bias in business analytics?

A. An analysis of successful startup companies to identify the characteristics that led to their success.

B. An analysis of failed marketing campaigns to learn from mistakes.

C. An analysis of customer satisfaction ratings for a product that has been discontinued.

D. An analysis of employee turnover rates at a company that has recently experienced significant growth.

Which of these statements about correlation and causation is true?

A) Correlation generally implies causation

B) Causation generally implies correlation

C) Causation and correlation have nothing to do with each other

D) Causation and correlation are the same things

What’s the meaning of the parameter k in k-Means Clustering

A) The total number of clusters

B) The total number of iterations

C) The learning rate of the clustering algorithm

D) The minimum distance between clusters

What’s NOT a good use case for Association Rule Mining to find relationships between items?

A) Understand user behavior and improve website design

B) Recommend items based on a user's previous choices

C) Identify suspicious activities and transactions

D) Identify product associations and cross-selling opportunities

Bootcamp overview

Learning goals:

- Derive actionable insights from data
- Perform exploratory data analysis and create meaningful visualizations
- Use value-based analysis techniques and create association rules for effective decision support
- Apply clustering techniques to discover segments in your data, e.g. different customer groups
- Build predictive models for regression and classification tasks
- Understand the key criteria for evaluating the performance of a predictive model
- Suggest specific business actions that will lead to better results

Bootcamp overview

This Week

Week 1 Jan 31	Week 2 Feb 7	Week 3 Feb 14	Week 4 Feb 21	Week 5 Feb 28	Week 6 Mar 7
Business Analytics Fundamentals	**Data Analysis With Python Fundamentals**	**Descriptive Business Analytics With Python**	**Diagnostic Business Analytics With Python**	**Predictive Business Analytics With Python**	**Prescriptive Business Analytics With Python**
Goal: Get the theoretical foundations for doing business analytics	**Goal:** Get the practical foundations for using Python for data analysis	**Goal:** Learn how to explore & describe your data (What happened?)	**Goal**: Learn how to get insights from your data (Why did it happen?)	**Goal:** Learn how to predict the future, given historical data (What will happen?)	**Goal:** Suggest better actions based on data (What to do?)

NOTE: With today's registration, you'll be signed up for all six sessions. Although you can attend any of the sessions individually, it's recommended participating in all six weeks.

Bootcamp overview

Learning Goals Week 5

- ❑ Learn to work with data under uncertainty
- ❑ Introduction to inferential statistics incl. regression modeling
- ❑ Do inferential statistics in Python
- ❑ Understand the fundamentals of Machine Learning and the difference to inferential statistics
- ❑ Learn different types of ML incl. decision trees
- ❑ Evaluate machine learning models
- ❑ Create predictive models in Python for regression and classification tasks
- ❑ Introduction to time series forecasting
- ❑ Understand the concepts of trends, seasonality, and randomness
- ❑ Run a time series forecast using ARIMA Models in Python

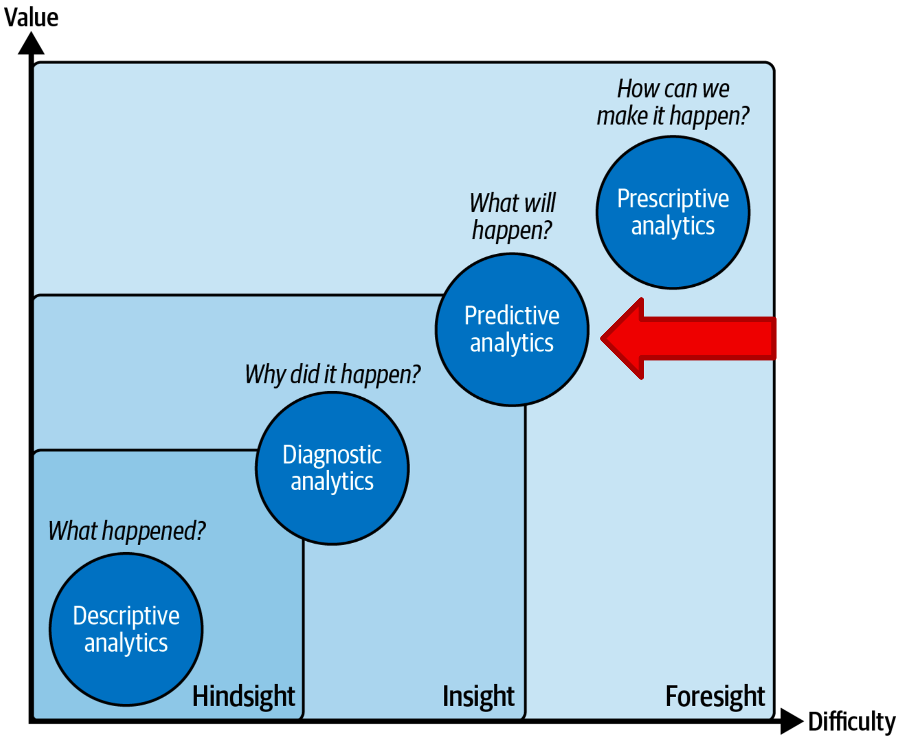
Inferential Statistics

Predictive Analytics – Introduction

- Predictive analytics is the process of using data, statistical algorithms, and/or machine learning techniques to identify the **likelihood of future outcomes** based on **patterns in historical data**.

→ **Being able to quantify uncertainty!**

- Can be used to solve a wide range of business problems and improve decision-making.
- However, predictive analytics requires careful planning, data preparation, and model building in order to be effective.



Descriptive vs. Inferential statistics

Descriptive statistics

- Descriptive statistics is a tool to summarize and describe the main features of a data set, such as measures of central tendency (e.g., mean, median) and dispersion (e.g., standard deviation, range).
- It can be used to prepare data for inferential statistics, but it is not its only purpose.

→ **Descriptive statistics describe what happened**

Inferential statistics

- Inferential statistics is a tool used to make generalizations about a population based on a sample by e.g., testing hypotheses, estimating parameters, and making predictions.
- Inferential statistics can be used for predictive analytics, but there are other methods as well (e.g. machine learning)

→ **Inferential statistics attempt to infer the population or data generation process from a sample.**

Inferential Statistics

Goal: Investigate whether findings found in samples can be generalized and quantify the uncertainty of this finding.

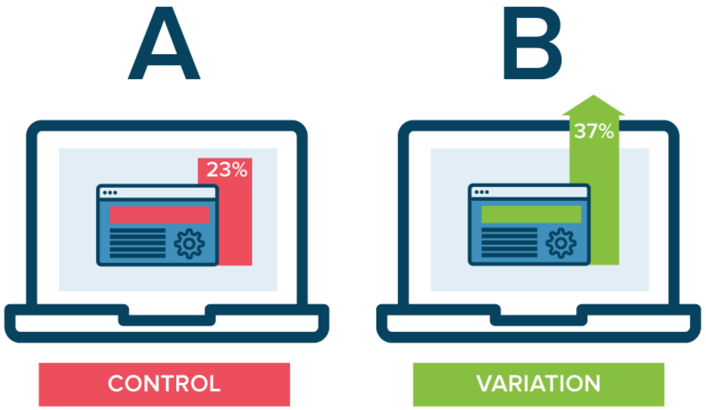
→ **Method**: Hypothesis test

- H0: There is no effect (difference, relationship, ...).
- H1: There is an effect

→ Hypotheses must be testable and falsifiable.

Important in practical application:

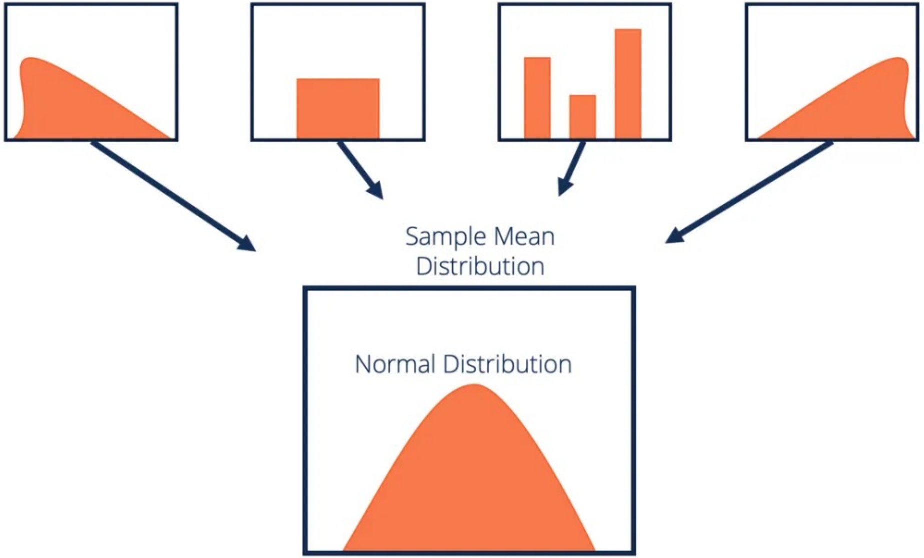
- There must be a defined standard action
- It is clear what case must occur for the default action to be replaced by another.
- These things must be defined before the analysis!



Initial situation: "There is no significant difference between the green and red buttons, and I'll continue with the red button until I'm convinced otherwise."

The Central Limit Theorem

- The Central Limit Theorem (CLT) is an essential statistical concept
- It states that the sample mean distribution of a random variable approaches a normal distribution as the size of the sample increases, regardless of the shape of the original population distribution.
- **If the sample is representative of the population and the sample size is sufficiently large, the sample mean can be a good estimate of the population mean.**



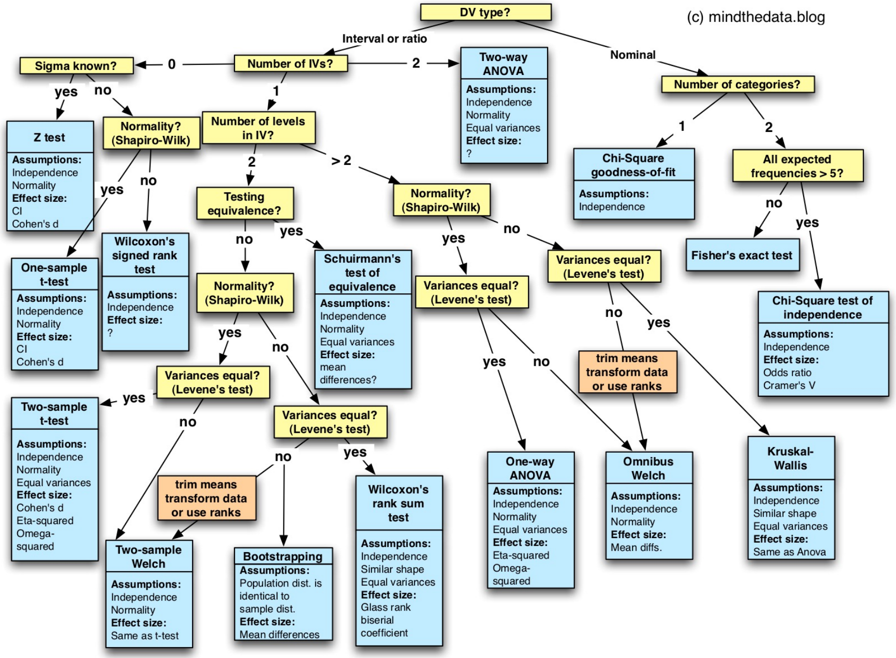
- Source: https://corporatefinanceinstitute.com/resources/data-science/central-limit-theorem/

Hypothesis testing

Statistical hypothesis testing is a framework for making decisions about population parameters based on sample statistics.

Steps include:

1. Formulating null (H0) and alternative hypotheses (H1)
2. Choosing a significance level (α)
3. Selecting an appropriate statistical test (t-tests, ANOVA, chi-square tests, etc.)
4. Collecting data and calculating test statistics
5. Interpreting the results



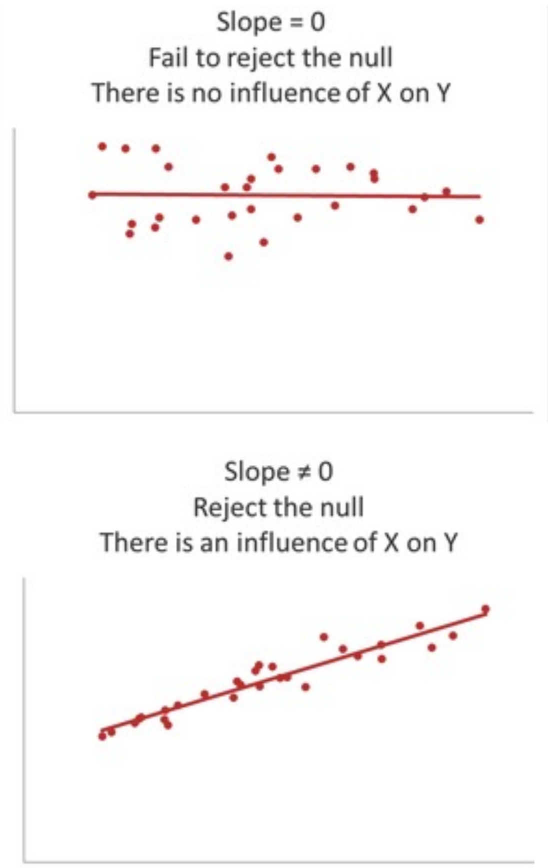
- Source: Inna Tsirlin, https://mindthedata.blog/2020/05/14/statistics-cheat-sheet-part-1/

Labs: Inferential Statistics

Regression

- Unlike correlation, regression indicates not only the strength but also the effect of the linear relationship.
- That is, if variable X is changed by a certain number of units, what is the effect on variable Y?

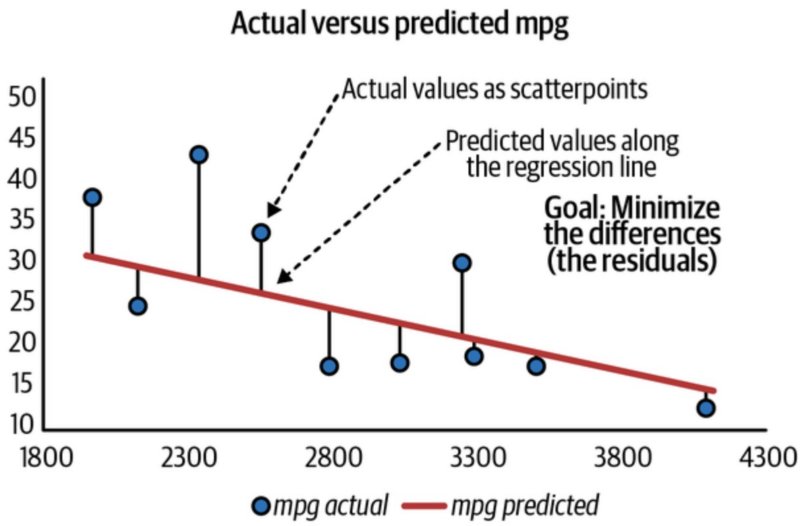
$$Y = \beta_0 + \beta_1 \times X + \epsilon$$



- Source: Mount, G., Advancing into Analytics (2021), O'Reilly

Regression

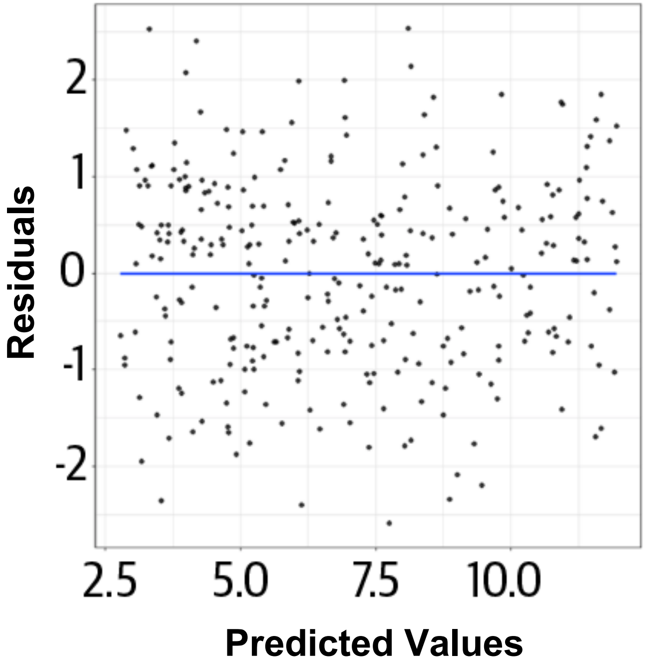
- Quality criteria of the regression: **Analysis of residuals**
- Graphical and/or computational, e.g.:
- R-squared = percentage of variance in Y explained by a linear model.
- Basically, the higher the R-squared, the better the model fits the data.
- If possible: visualize residuals.



- Source: Mount, G., Advancing into Analytics (2021), O'Reilly

Regression – Residuals

- Summary statistics are great to compare models to one another at scale. But it's impossible to understand a regression model from these alone.
- A good approach to understand your model better is to look at the distribution of the **residuals** (errors)
- A residual diagram plots the predicted values against the residuals. Assumptions:
  - **Linearity**: residuals are randomly scattered without any notable pattern.
  - **Homoscedasticity**: The spread of the residuals is more or less the same for all values.



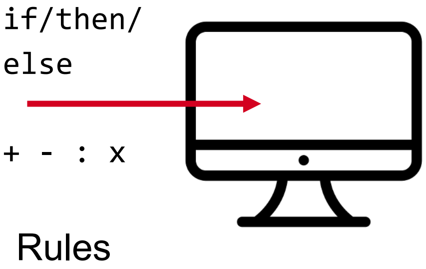
Labs: Regression

Machine Learning

What is Machine Learning (ML)?

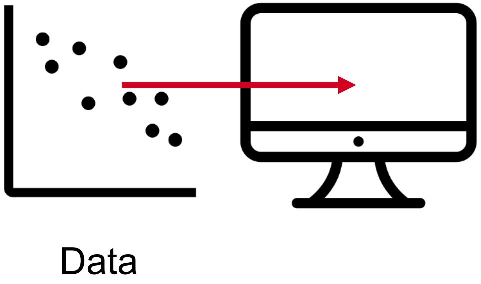
- **Core idea**: find patterns in historical data to solve a specific task, e.g. putting observations into categories, scoring probabilities, finding similarities between items, etc.
- ML typically has two phases: **training** (learning) and **inference** (also called testing or prediction).
- ML is a set of tools for writing software by turning examples into instructions.

Traditional



The computer can perform the task it has been *instructed* to do.

Machine Learning



The computer can perform the task it has *learned* to do.

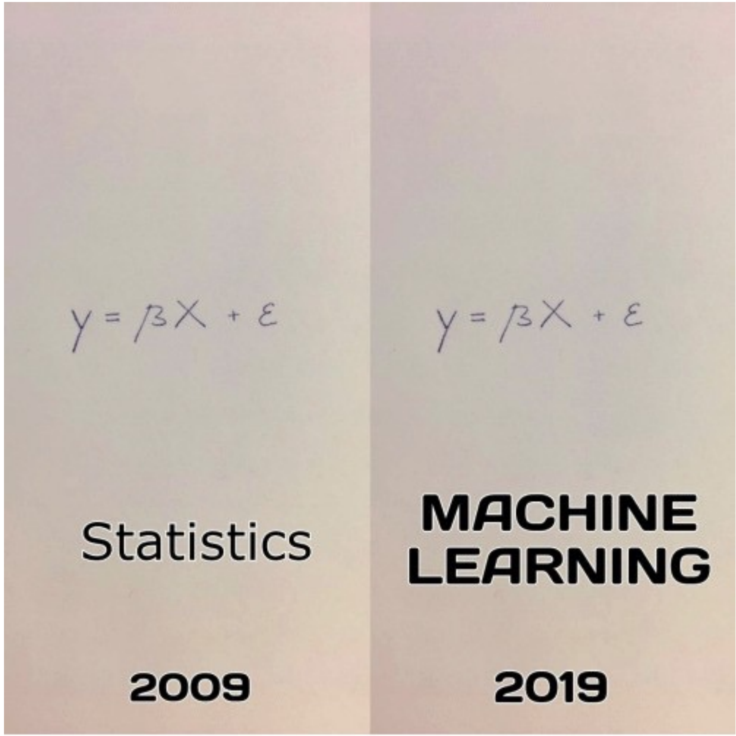
Statistics vs. Machine Learning

What’s the difference between statistics and machine learning?

Isn’t it all the same?

→ (Frequentist) Statistics and Machine Learning are two different data modeling mindsets

Let’s explain this using a story…

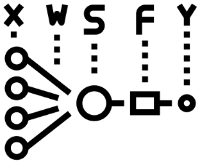


Statistics vs. Machine Learning

Imagine the following scenario:

A business analyst collects data on customer revenues, their company tenure, and some demographic information. He then flags all customers that abandoned the company with 1 and everyone else with 0. The business analysts gives this data to a data scientist to “do something with it”

Three days later the data scientist comes back with a logistic regression model that expresses how customer churn depends on revenues, tenure and demographic information.

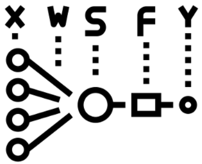


What can the business analyst do with this model?

→ Without any further information about the mindset behind the modeling process, we can’t clearly answer that!

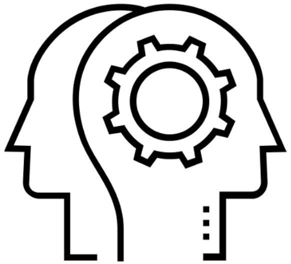
- Adapted from Modeling Mindsets, C. Molnar

Statistics vs. Machine Learning



- Can the business analyst use the model to accurately predict customer churn?
  → *Only if the model has a low generalization error and was evaluated properly*

- Can the business analyst interpret the p-values to say whether the effect of the demographic information is significant?
  → *Only if model assumptions hold and relying on a frequentist interpretation of probability.*



The mindset behind the model processing dictates what we can do with the model.

- Adapted from Modeling Mindsets, C. Molnar

Frequentist Mindset

How would a frequentist statistician build the regression model?

In frequentist inference the data scientist might have made assumptions about how data is distributed, decided on a linear regression model, fitted the model, and diagnosed the model.

Now the business analyst may interpret the parameter estimates, including confidence intervals and hypothesis tests.

→ This model is optimized to infer the original distribution of the data and all metrics that are coming with it.

The fact that you actually apply this model to new data without examining this data thoroughly is something that scares every frequentist statistician!

Machine Learning Mindset

How would a data scientist with a machine learning mindset build the regression model?

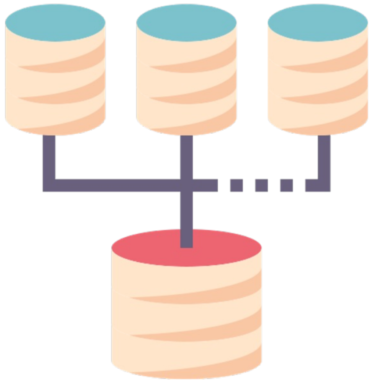
In machine learning, the regression model came out of a “contest” between multiple models.

The logistic regression model **just happened to be the most performative** one.

Performance is usually measured by comparing the performance a model had on a training dataset vs. the performance a model had on a new test or validation dataset, often using a technique called cross-validation.

The application and evaluation of the model on new, real data is actually nothing that ML practitioners fear, but something they embrace as that’s one of the main goals.

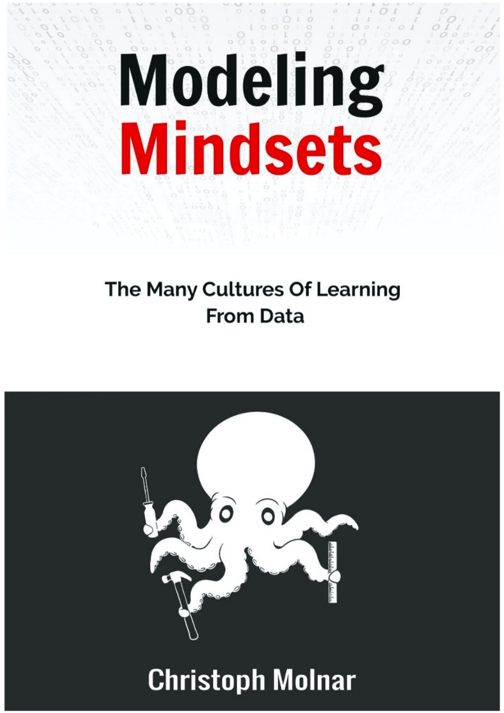
In fact, machine learning heavily relies on out of sample metrics to evaluate the performance of a machine learning model.



Summary

- Frequentist statistics try to **infer the original distribution** of data given the data you have observed.
- Machine Learning practitioners try to build a model that works well on new data, i.e. it generalizes well enough to **predict new data points**.
- Knowing which modeling mindset was involved during the modeling process dictates what you can do with the model: Assess metrics for the current data or make predictions for new data.

→ There are many statistical modeling mindsets (Bayesian, Causal, …)

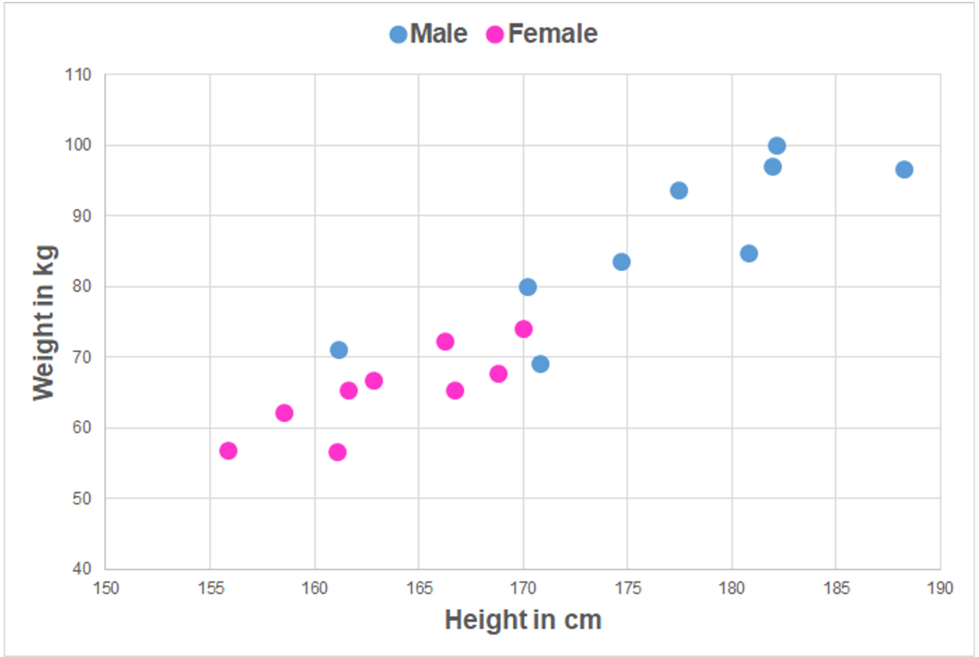


https://book.modeling-mindsets.com/

Let’s put our Machine Learning hat on!

Let's look at some data

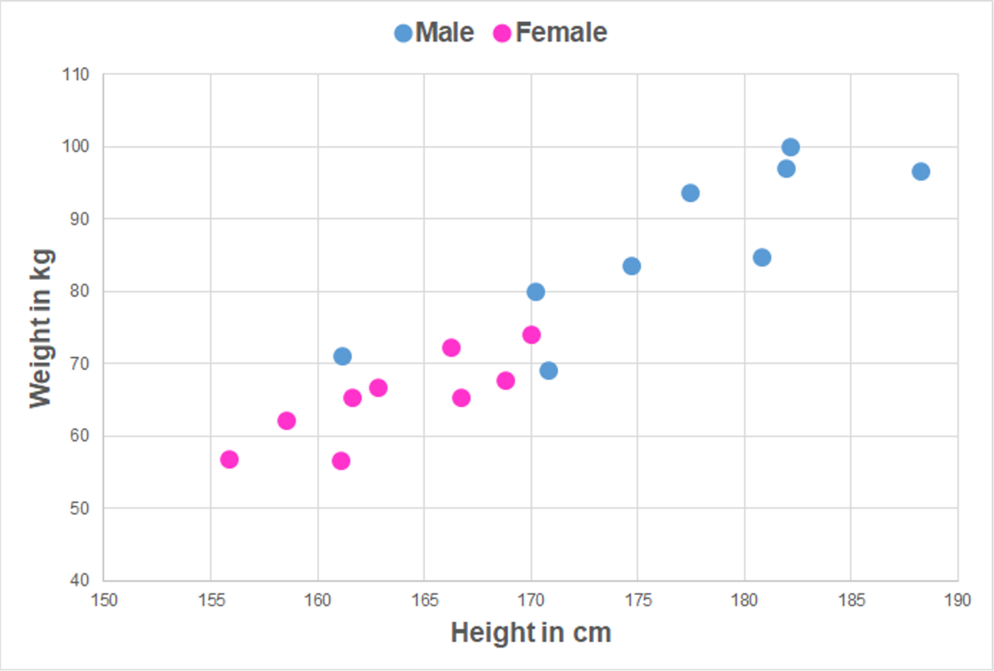
Gender	Height in cm	Weight in cm
Male	188	97
Male	182	100
Male	161	71
Male	181	85
Male	182	97
Female	148	45
Female	167	65
Female	166	72
Female	161	57
Female	156	57
Female	163	67
...	...	...



Possible Problems

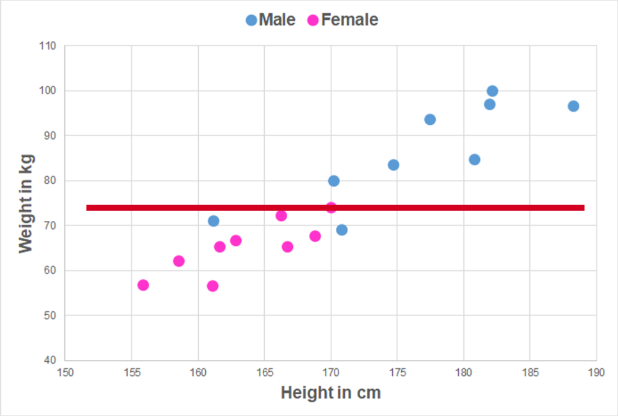
- Can we predict the gender, given height and weight?
- Can we predict the height, given the weight?
- Do the points naturally fall into groups?

...



Most common model classes

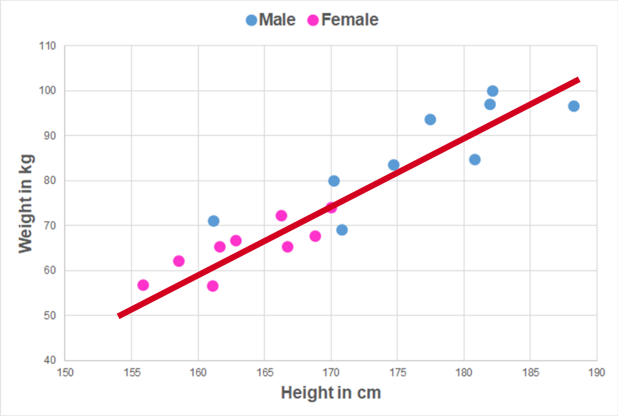
Classification



Which class?

Example: Logistic Regression

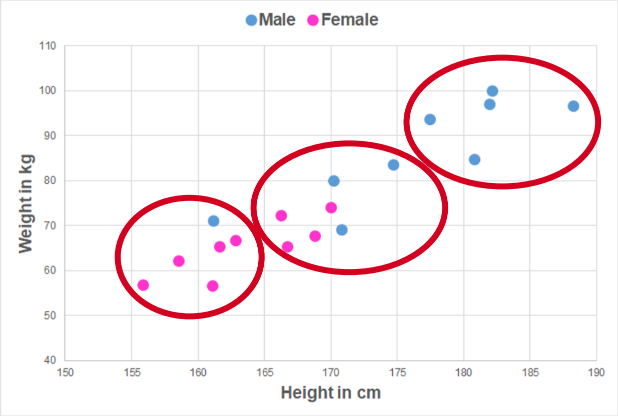
Regression



Which numeric value?

Example: Linear Regression

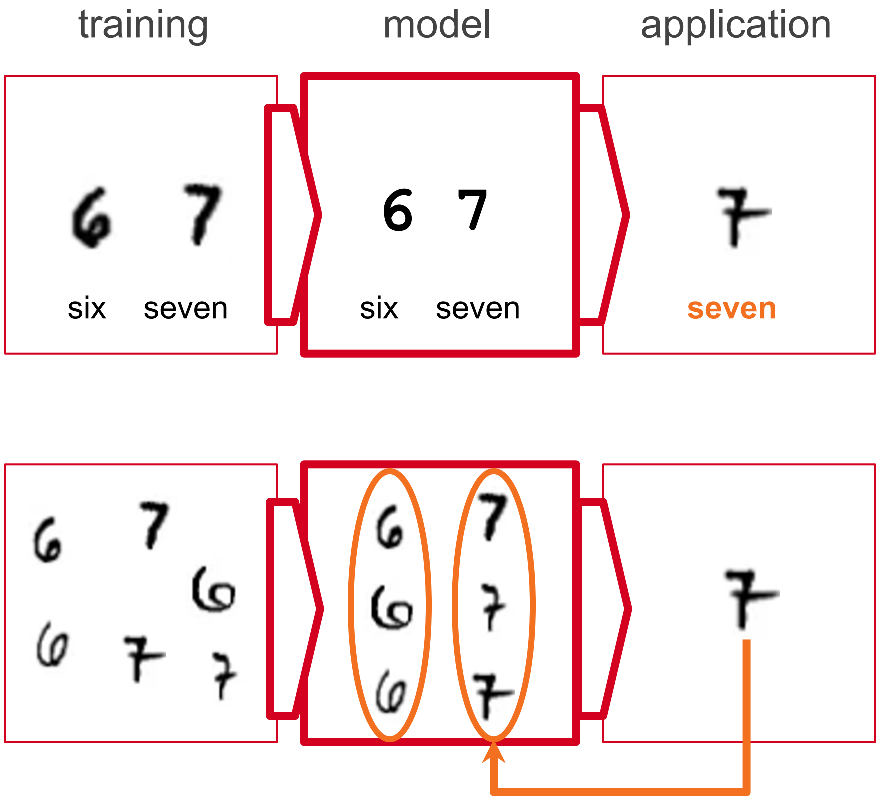
Clustering



Any groups?

Example: k-Means Clustering

Learning Types: Supervised vs. Unsupervised



Supervised

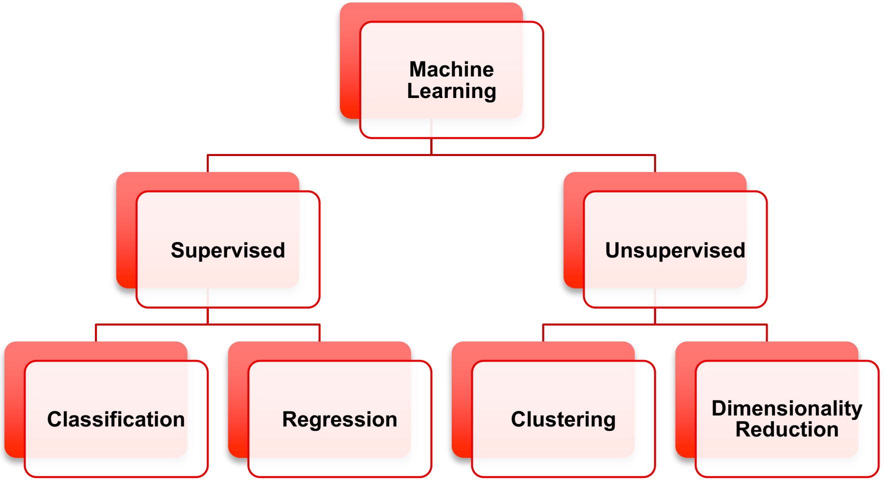
- Instances are **labeled**: For each observation the value of the target variable is known.
- Train a model that predicts the output variable (target) based on the input variables (**features**).

Unsupervised

- Instances are **unlabeled**: The data contains only features, no labels.
- Model training based on relationships amongst instances.

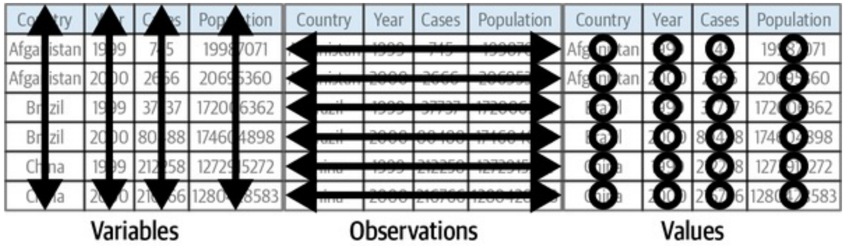
The Supervised Machine Learning Process

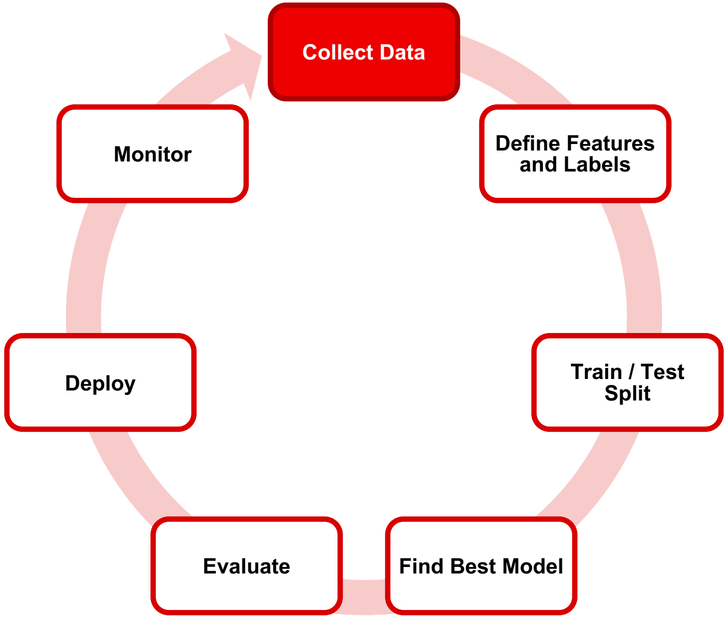
- **Most enterprise ML problems are supervised**
- *Supervised machine learning* is a process of training an ML model based on historical data when the ground truth is known.
- “Features” used to calculate the “label”
- Supervised Machine Learning follows a repeatable pattern



Supervised ML Step 1: Collect Historical Data

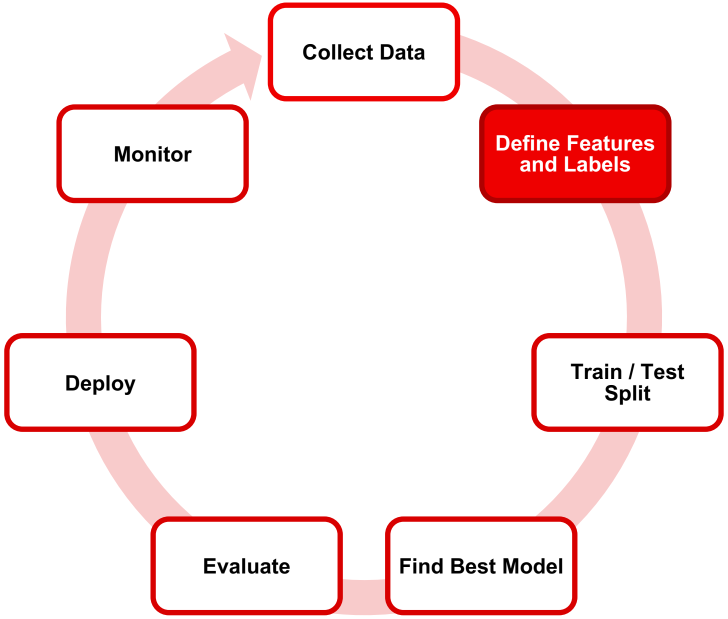
- Get historical data in a form that ML algorithms can learn from them
- Tidy Data
  - Every observation is in its own row.
  - Every variable is in its own column.
  - Every measurement is a cell.





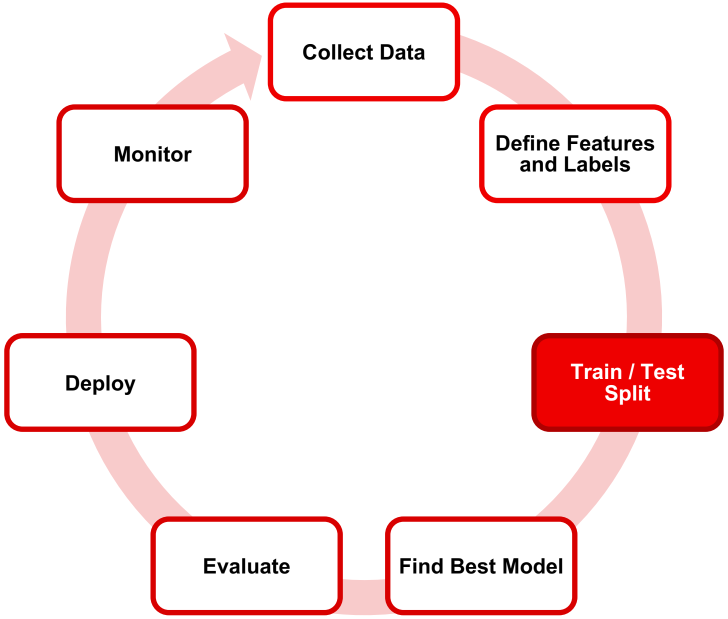
Supervised ML Step 2: Define Features and Labels

- ML algorithms trying to find a function (*model*) that will predict some output given some inputs.
  - **Labels** (y, targets, outputs, or dependent variables) = variables you want to predict
  - **Features** (x, attributes, inputs, or independent variables) = variables used for prediction
- Training data: Historical dataset containing features and labels for a given number of observations (training examples).
- Numeric label = Regression problem
- Categorical label = Classification problem



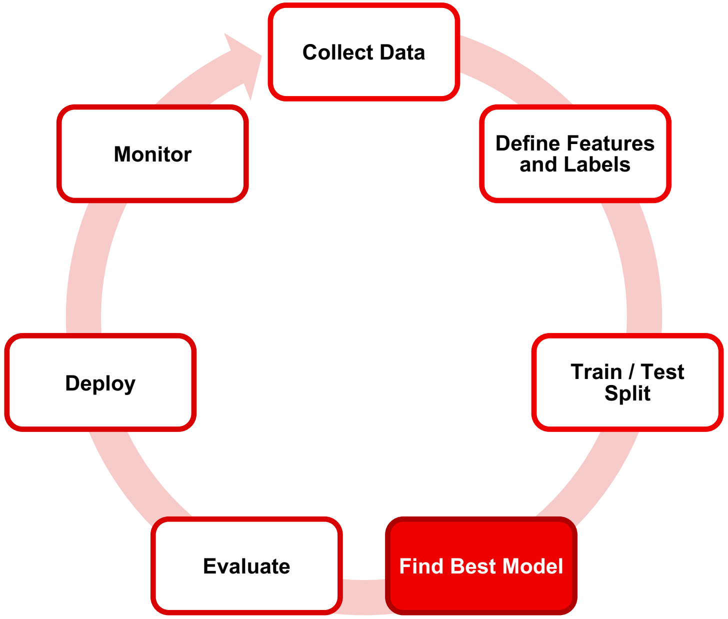
Supervised ML Step 3:
Split Data into Training and Test Sets

- Training set: part of the historical data to train the model (usually between 70% to 80%).
- Test set (holdout set): Rest of the historical data that will be used for the evaluation of the ML model.
- Split is done to make sure the model has "predictive power ", e.g. it will not only perform well on the data it knows, but also on new, unseen data from the same distribution
- Performance of ML models measured on test set



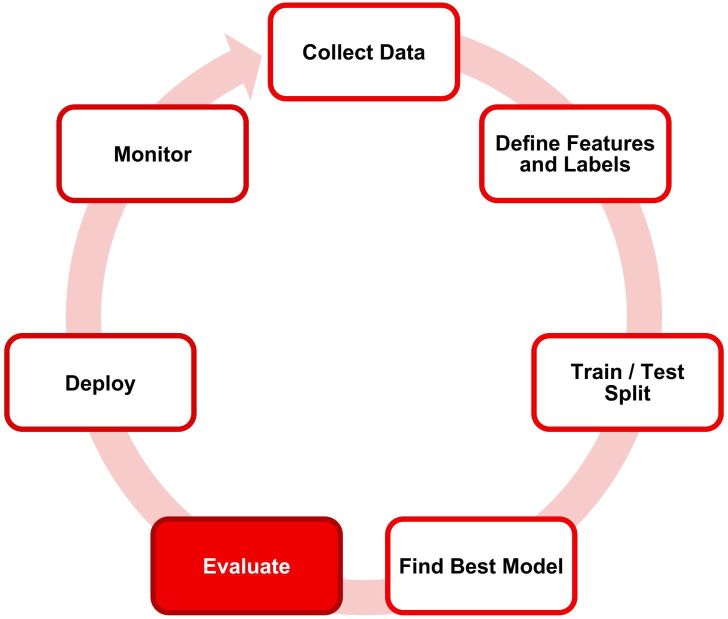
Supervised ML Step 4:
Use Algorithms to Find the Best Model

- Find the best model that represents your data with the highest predictive power.
- Try out various ML algorithms with different parameters to find best model.
- The model is the final deliverable of the ML training. Essentially a function that calculates y for x:
  - $y = 200x + 1000$
- Example: linear regression algorithm $y = b1x + b0$
  - Computer uses training data to find the best parameters b1 and b0
  - Parameter estimation = learning/training.



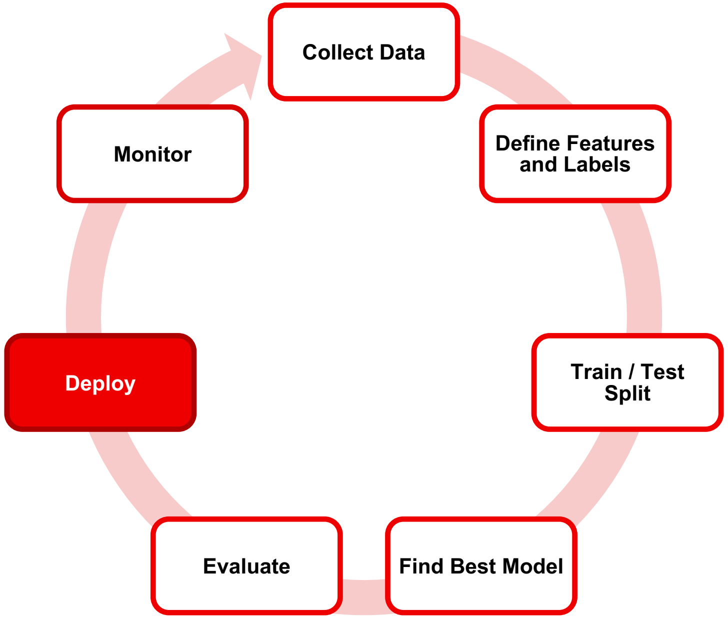
Supervised ML Step 5: Evaluate the Model

- Evaluation metrics are calculated for two reasons:
  - Compare models to one another in order to find the best predictive model
  - Continuously measure the model's true performance on new, unseen data
- Core Idea: Compare the values that the model predicts to the values that the model should have predicted according to the ground truth (labels).
  → Sounds simple, but quickly gets complicated.
- Different evaluation metrics are used for classification and regression tasks.



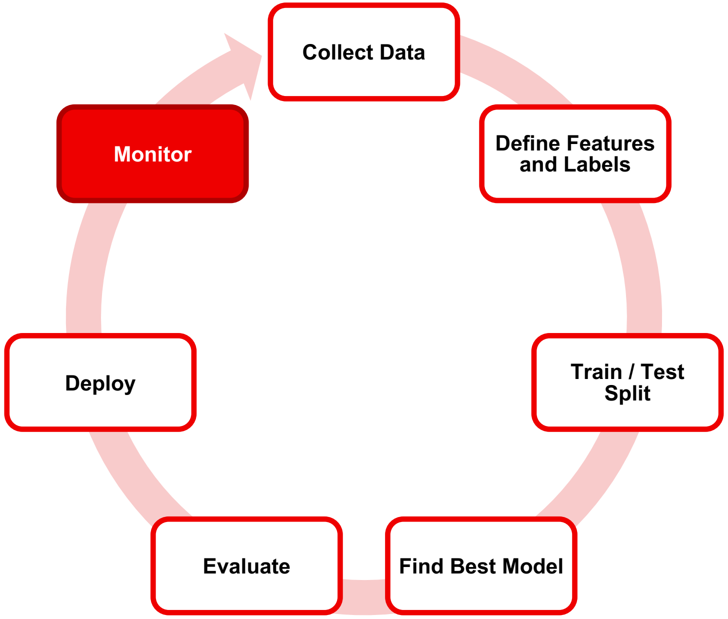
Supervised ML Step 6: Deploy the Model

- Goal: Users or applications can use the model.
- “Usage” is called inference, prediction, or scoring.
- At this stage, the model doesn’t learn anymore but only calculates outputs for new input values.
- Two broad types:
  - **Online Prediction**: Model is hosted as an HTTP API that takes input data and returns the predictions immediately (speed)
  - **Batch Prediction**: Model is used to score a lot of data at once and outputs are written to a file or database (volume)



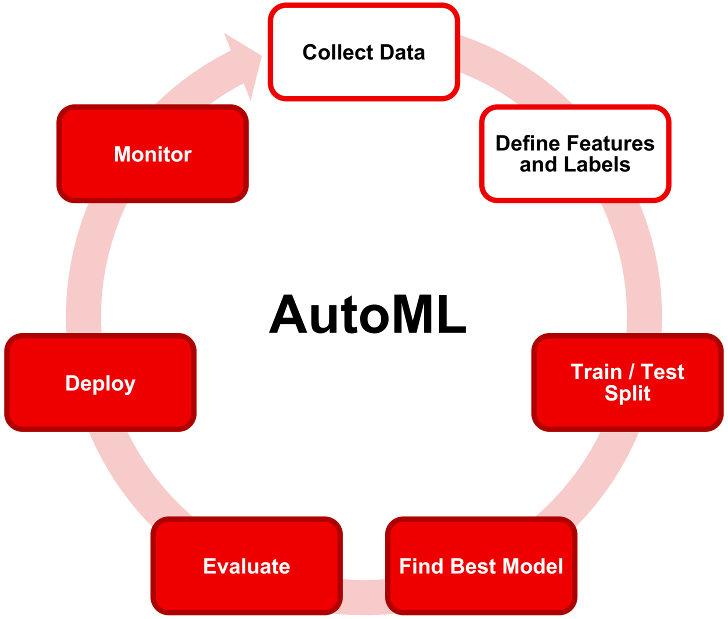
Supervised ML Step 7: Monitor and Maintenance

- The ML process never ends!
- As data patterns change, models need to be retrained, and we need to look at whether the initial performance of the ML model can be kept high over time.
- ML models need a considerable amount of maintenance after deployment, which you should consider in your feasibility analysis.



Automated Machine Learning (Auto ML)

- AutoML abstracts and automates most of the steps of the supervised Machine Learning process:
  - Finding the best algorithm
  - Finding the best learning parameters
  - Optimize the features (feature engineering)
  - Splitting your data into training % test set
  - Deploy your model & monitor performance
- **Things you still need to do:**
  - Provide clean data
  - Define the problem (incl. labels & features)
  - Interpret the results critically



Scikit-Learn

- Popular machine learning library in Python
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable
- https://scikit-learn.org/

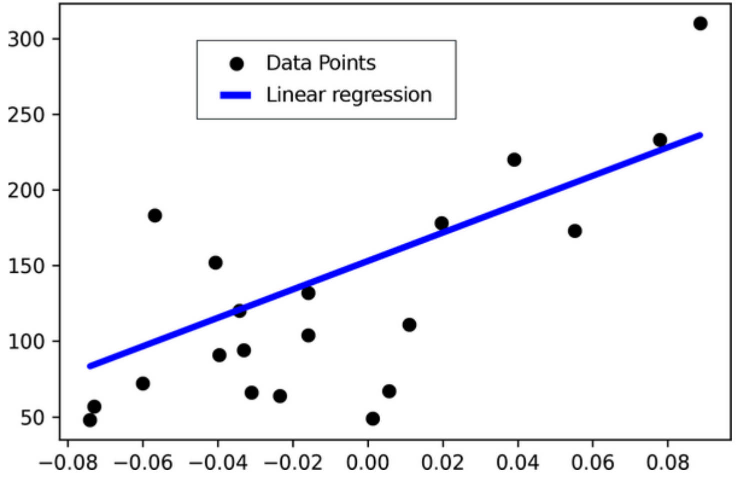
```
pip install scikit-learn
```

Interactive Labs: Machine Learning

Further Readings

Linear Regression

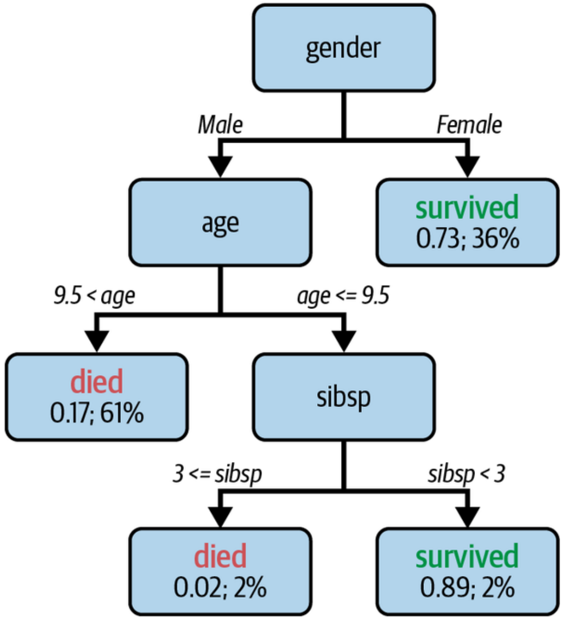
- Essential algorithm in ML: logistic regression, regression trees, time-series analysis, neural networks
- Easy to learn, hard to master: Given numeric input data, we find a linear function that calculates numeric output data.
- Main assumptions:
  - Features are independent from one another (low feature correlation).
  - Features and targets have linear relationship (e.g., when one variable increases, the output variable also increases or decreases at a fixed rate)



Classification Algorithms: Decision Trees

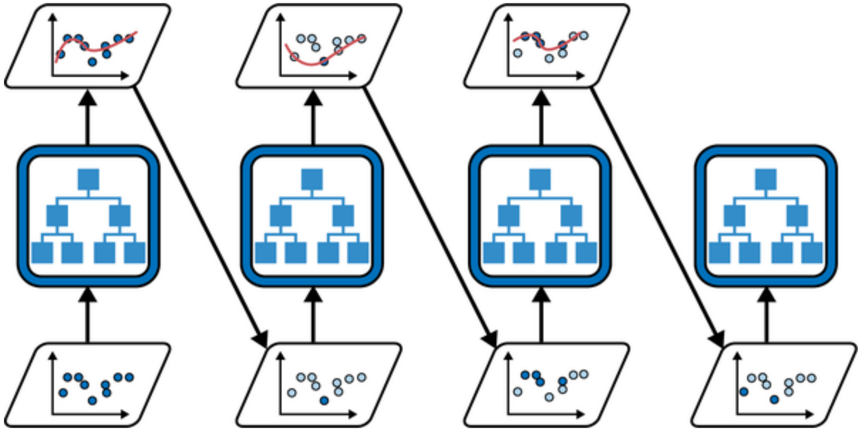
- Decision trees work with categorical and numerical data. They are very popular for classification tasks.
- Decision trees split data at various levels, variable after variable, until the slice becomes small enough for prediction (hierarchical order of if-then rules)
- Advantages: Good all-round algorithms with robust performance, easy to explain and understand
- Problem: “Greediness”: Every split will only be done once at the point where the data shows the highest discrepancy. This process isn’t ideal in all situations.

Survival of passengers on the *Titanic*



Ensemble Models

- Ensemble models combine multiple (weak) models into a strong one
- Popular example: Boosted trees
- Boosted trees train multiple single decision trees where each tree tries to correct the erros of the previous tree.
- The result is a complex, usually high performing model which is harder to interpret and explain than a single decision tree.



Evaluating (Machine Learning) Models

How to validate statistical models in the mindset of a machine learning practitioner:

Statistical Evaluation

- Running the right tests
- Splitting data correctly
- Choosing the right metrics

→ Does this model work?
→ “Doing things right”

Semantic Evaluation

- Do I understand the results?
- Are the results helpful?
- How can I do better?

→ Is this model useful?
→ “Doing the right thing”

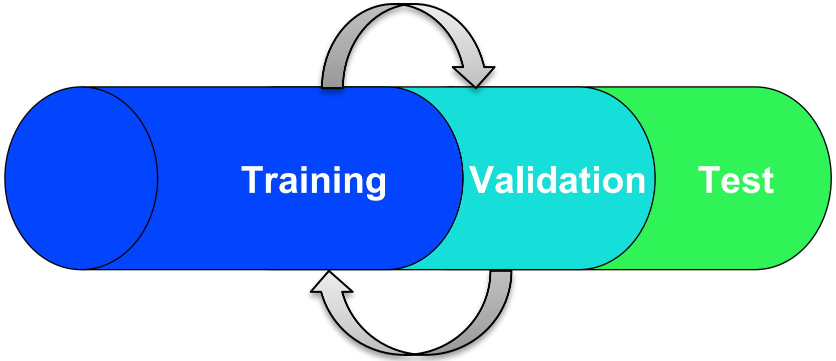
→ Don’t build models that are statistically correct, but **don’t solve a problem**. (waste of resources)

→ Don’t build models that seem to solve a problem, **but lack the statistical groundwork**. (danger zone!)

→ Build models that are **statistically robust AND help you solve a concrete (business) problem**!

The Trick: Don’t Use All Data to Build a Model

Split your data typically into three pieces:



Training Data
- Used to fit models (estimate parameters)
- Typically 60-80% of the data set

Validation Data
- Used to measure the **error of candidate models**
- Training and validation are performed iteratively until model achieves the desired performance

Test Data
- Used to measure the **performance of the final selected model**
- True blind test as validation data was used multiple times during training

Golden Rule: Train and evaluate on different datasets!

Statistical Evaluation Process

1. **Blind test on new data**
   a. Create model on one part of the data (training set)
   b. Validate on an unseen, independent part of the data (validation set)

2. **Calculate a performance metric**
   a. Accuracy
   b. RMSE
   c. …

3. **Improve the performance**
   a. Adjust data (add/remove features, get more data)
   b. Adjust model (change model type and/or hyper parameters)

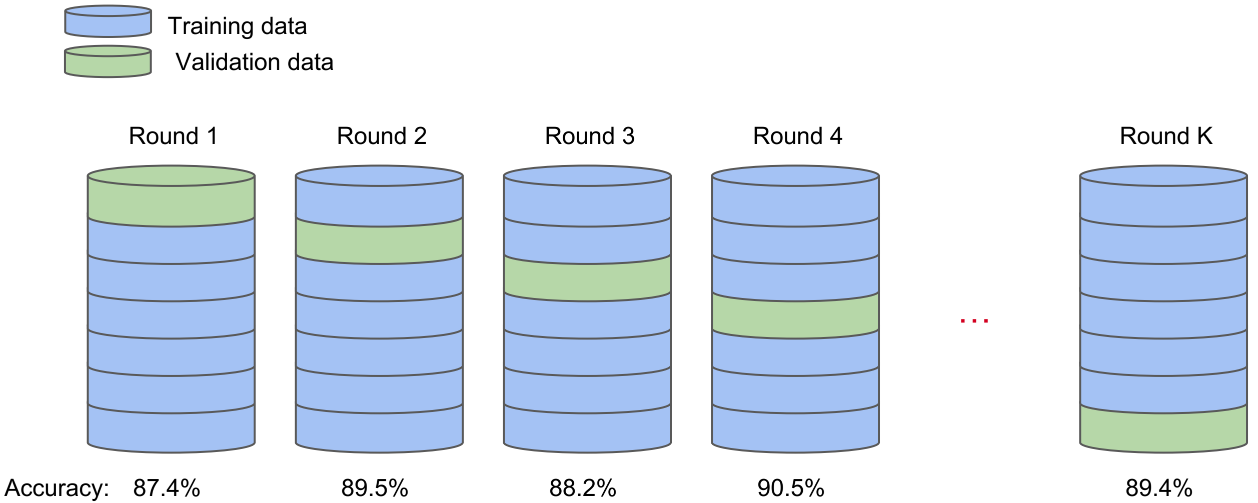
Requirement: Baseline definition! If you don't have a baseline defined upfront, you don't know when your model is "good enough". **Try to have a non-ML baseline first!**

Golden Rule: Simulate the **practical use** of the model as closely as possible!

Cross Validation

→ Popular method to prevent overfitting but at the same time try out many different models.

1. Split the data into K folds
2. K-1 folds are used for training, 1 fold for validation
3. Shuffle the validation fold k times



→ Final accuracy is the average accuracy over all rounds

→ Compare different models

→ Train best model on full dataset

→ Stratified Cross Validation: Keep representation of target variable fixed

Approaches for splitting data

Random splits

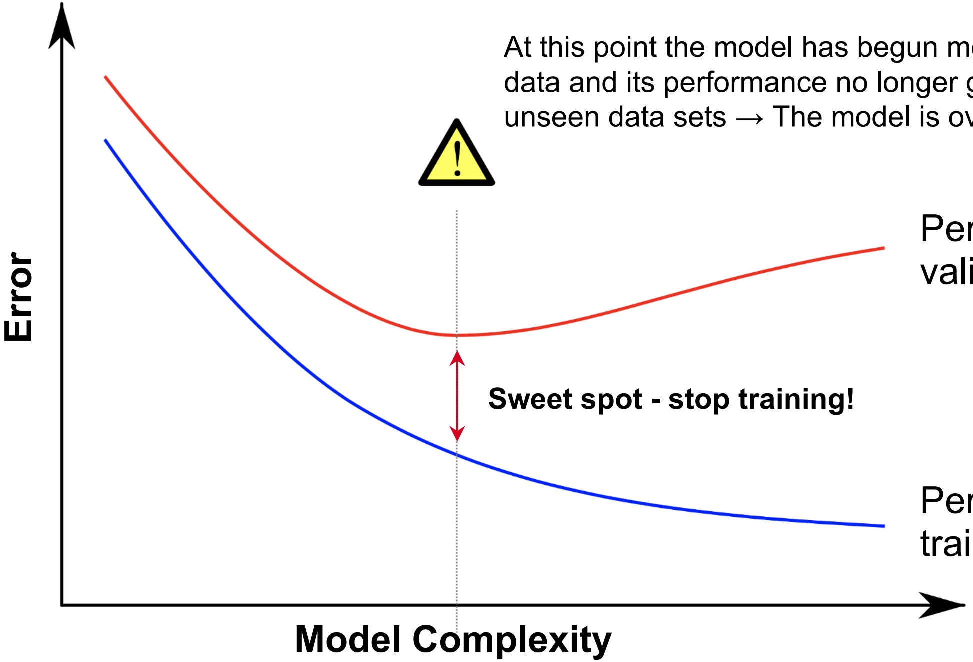
- Assign individual observations randomly to training/validation and test set.
- Pros:
  - Easy to implement and easy to understand
  - Works well with very large datasets
- Cons:
  - Target leakage when data has high autocorrelation! (e.g. time series, model might memorize data from training instead of capturing the actual distribution)
  - Need to verify that target distribution is the same for all sets (especially imbalanced classes)

Stratified splits:

- Stratified random sampling divides the data into smaller groups (strata) based on shared characteristics
- Pros:
  - Keeps the same proportions/ratios of targets (and features) across all sets
- Cons:
  - Groups need to be defined upfront
  - Harder to implement for numerical data (binning)

Why Evaluating the Model on Unseen Data?

There is a high risk of overfitting when a model is not evaluated on data which was not used to generate the model. Generally, the risk of overfitting increases along with the model complexity.



Overfitting

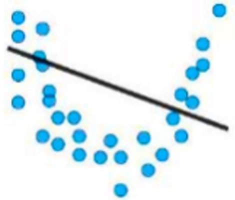
Overfitting = Model (almost) perfectly fits the data upon which it was created but cannot estimate beyond this data.

→ Algorithm memorizes data instead of learning meaningful dependencies

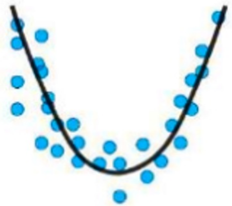
- **Solutions**:
  - Aim at trade-off between fit and generalization
  - Build and validate model on different subsets of data - training and validation / testing sets, ideally **cross-validated**

Overfit
Poor generalization

Underfit
Strong generalization



Good Fit
Useful generalization



Sources of Error: Bias, Variance and Noise

Bias:

- The truth is different from what the technique can capture. Since the truth is unknown, you have to make assumptions

Variance

- Since the algorithm has only finite data, it cannot find the optimal model.

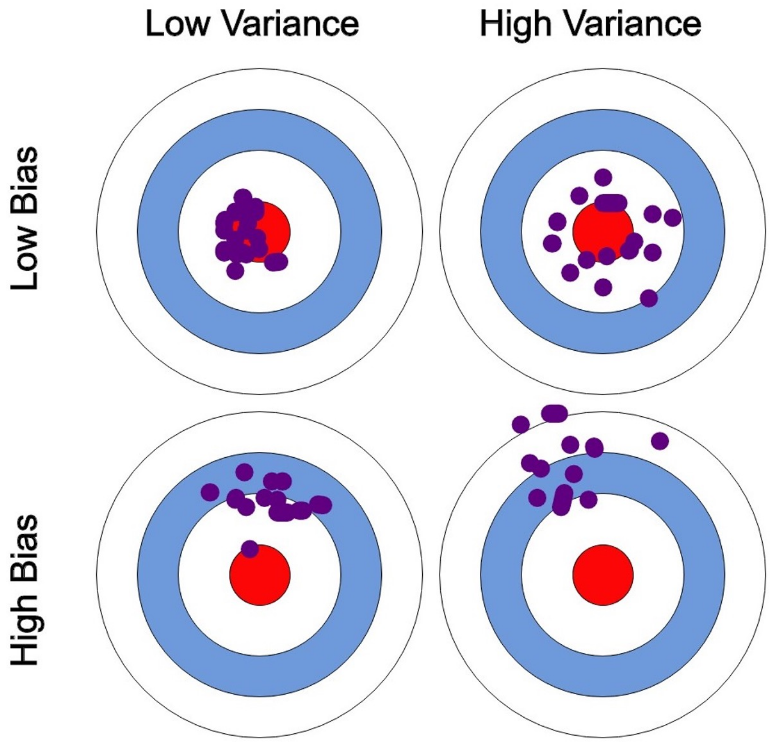
Noise

- Data contains random noise which cannot (and should not) be captured by the model

→ A model with low bias and low variance is rare.

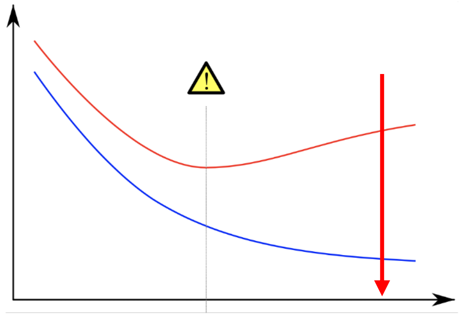
→ **Reducing model bias increases variance and vice versa.**

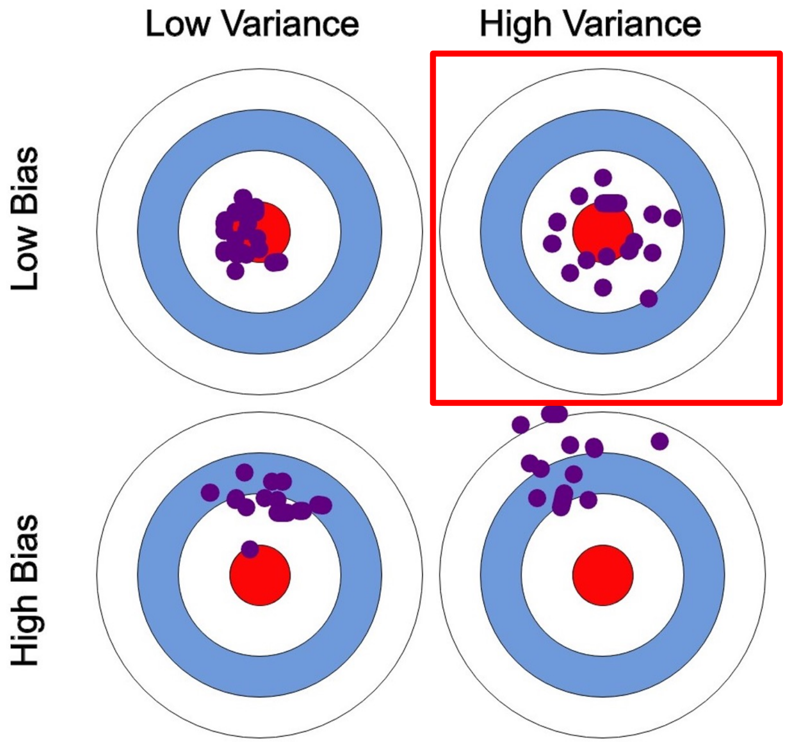
→ The goal is to find the model which best fits the business goal. This typically requires a tradeoff between bias and variance.



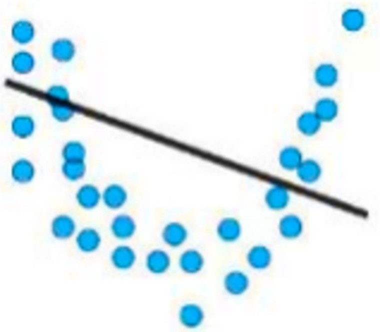
Bias vs. Variance – Examples

This model has **high variance and low bias**. It is **overfitting** the training data and will very likely show a large error on new data.

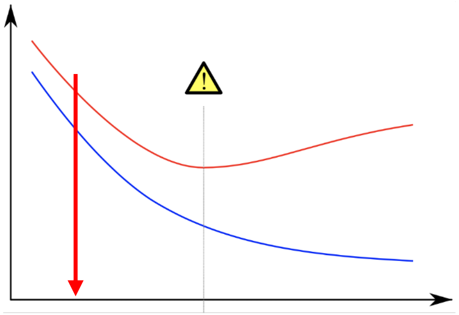


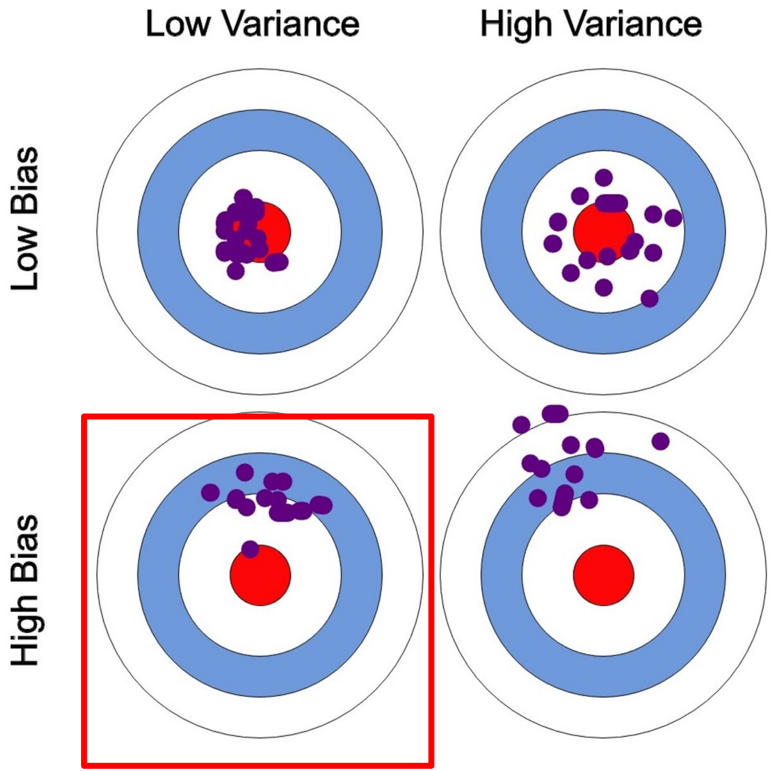


Bias vs. Variance - Examples



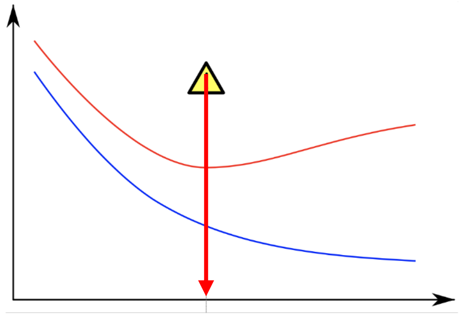
This model has **low variance and high bias**. It is **underfitting** the training data and has a large error on training and validation data.

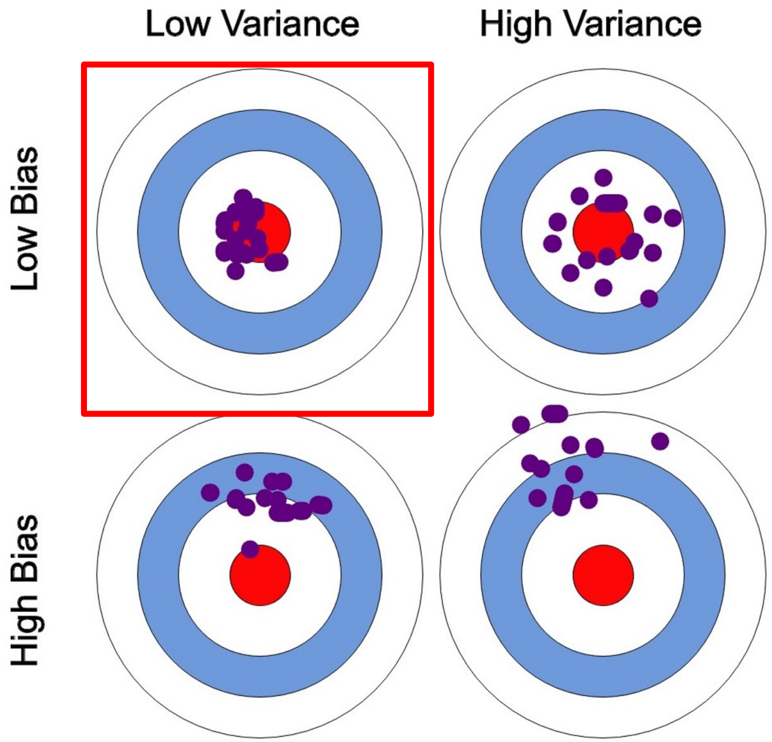




Bias vs. Variance – Examples

This model has a **good trade-off between variance and bias**. It shows a **good fit for the data.**





Popular Performance Metrics

→ Regression and classification tasks use different evaluation metrics

Most popular:

- Classification
  - Accuracy, Precision, Recall, **F1-Score**
- Regression
  - Mean Absolute Error, **Root-Mean-Square Error**, **R^2**

Classification: Confusion Matrix

Confusion Matrix		Predicted Class		
		Red	Green	Blue
Actual Class	Red	4	3	2
	Green	5	4	1
	Blue	2	2	8

↓ **Positive class = "Red"**

Confusion Matrix		Predicted Class		
		Red	Green	Blue
Actual Class	Red	**TP**	**FN**	
	Green	**FP**	**TN**	
	Blue			

True positives (TP): Actually positive cases that were correctly assigned positive class.

→ Example: 4 actual "Red" cases were classified as "Red"

True negatives (TN): Actually negative cases that were correctly assigned negative class.

→ Example: 15 other colors correctly classified as not "Red"

False positives (FP): Actually negative cases that were wrongly assigned positive class (Type I error).

→ Example: 5 "Green" and 2 "Blue" cases were wrongly assigned "Red'

False negatives (FN): Actually positive cases that were wrongly assigned negative class (Type II Error).

→ Example: 3 "Green" and 2 "Blue" cases were wrongly assigned as not "Red" although they were actually "Red"

Classification: Confusion Matrix

Confusion Matrix		Predicted Class		
		Red	Green	Blue
Actual Class	Red	4	3	2
	Green	5	4	1
	Blue	2	2	8

↓ **Positive class = “Red”**

Confusion Matrix		Predicted Class		
		Red	Green	Blue
Actual Class	Red	**TP = 4**	**FN = 5**	
	Green	**FP = 7**	**TN = 15**	
	Blue			

Accuracy: How often is the model correct?
→ (TP + TN) / Total = (4 + 15) / 31 = 61.3%

Precision: How exact is the model? (what % of cases predicted as positive are actually positive?)
→ TP / (TP + FP) = 36.4%

Recall/Sensitivity: How well does the model recognize positive cases?
→ TP / (TP + FN) = 44.4%

F1-Score: How exact is the model and how well does the model recognize positive cases?
→ 2 TP / (2TP + FN + FP) = 28.6%

→ **Which of these metric to prioritize depends on the problem you’re trying to solve!**

Regression: Performance Metrics

ID	Contract years (y)	Predicted Contract years ($\hat{y}$)	Residual ($\hat{y} - y$)
1	5	5	0
2	1	1.2	0.2
3	0.5	0.3	-0.2
4	4	4.4	0.4
5	8	9	1
6	6	5.9	-0.1

n

- **Mean absolute error (MAE)**: Large errors have the same power as small errors

$$\frac{1}{n} \cdot \Sigma_n |\hat{y} - y| = \mathbf{0.32}$$

- **Root-mean-square error (RMSE)**: Assigns more weight to large errors

$$\sqrt{\frac{1}{n} \cdot \Sigma_n (\hat{y} - y)^2} = \mathbf{0.46}$$

- **R²**: How much variability in Y can be explained by the model?

$$R^2 = 1 - \frac{RSS}{TSS}$$

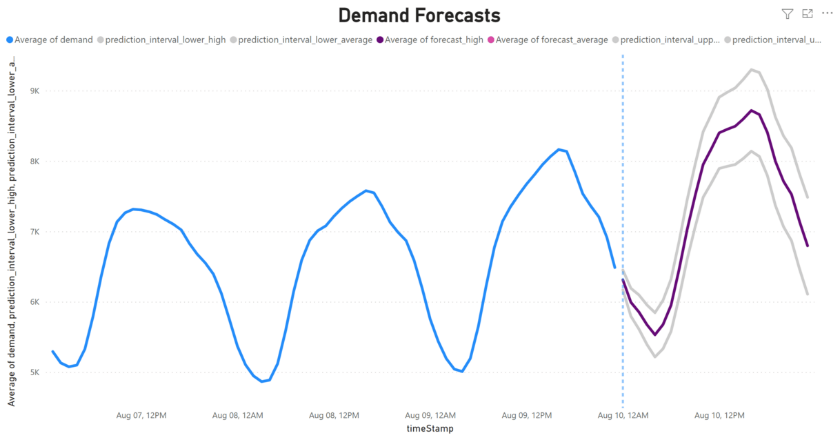
Q&A

How do you feel?

Time Series Forecasting

Introduction to Time Series Forecasting

- Time series forecasting = predict future values of a time series based on its past behavior.
- Time series data is characterized by its temporal ordering, meaning that the order of the data points matters.
- → Contrast to regression where each observation are typically independent (order does not matter).
- Many application areas in business!



Time Series – A Special Regression

- Time series problems are essentially regression problems where not only the features influnce the target, but also the **previous records.**

Regression problem:

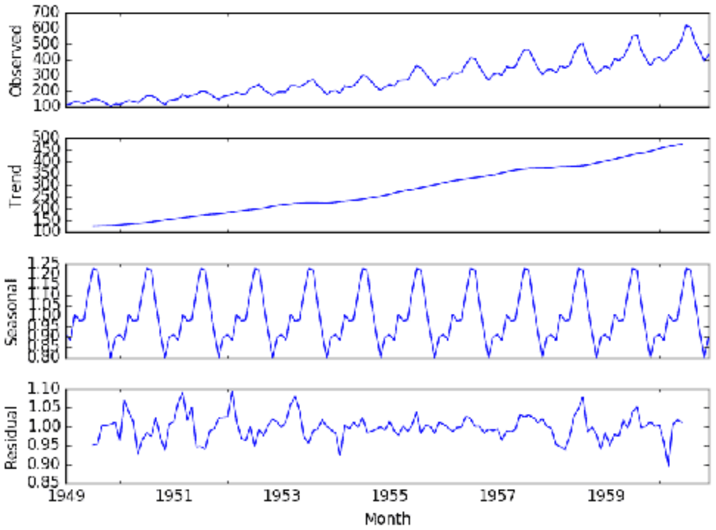
Precipitation	Temperature	Energy Demand
120	74	5,430
120	86	6,120
20	74	5,100
…	…	…

Time series regression problem:

Demand t-2	Temperature t-2	Demand t-1	Temperature t-1	Precipitation t0	Temperature t0	Energy Demand
5,100	74	6,120	86	20	74	5,430
…	…	…	…	…	…	…

Time Series – A Special Regression

- Time series can be decomposed into at least three components:
  - observation
  - trends
  - seasonality
- Different algorithms such as ARIMA (Autoregressive integrated moving average) or SARIMA (ARIMA + Seasonality) account for these patterns



Interactive Lab: Forecast Time Series with ARIMA Models

Q&A
How do you feel?

Wrap-up

What did we learn today?

- Learn to work with data under uncertainty
- Introduction to inferential statistics incl. regression modeling
- Do inferential statistics in Python
- Understand the fundamentals of Machine Learning and the difference to inferential statistics
- Learn different types of ML incl. decision trees
- Evaluate machine learning models
- Run predictive modeling in Python for regression and classification tasks
- Introduction to time series forecasting
- Understand the concepts of trends, seasonality, and randomness
- Run a time series forecast using ARIMA Models in Python

Outlook for next week

Week 6: Predictive Business Analytics with Python

- ❑ Understand Prescriptive Analytics
- ❑ Differentiate between automated vs. manual decision-making
- ❑ Get an intuition about recommender systems and reinforcement learning
- ❑ Get hands-on experience with reinforcement learning
- ❑ Learn the most critical AI services for business analytics
- ❑ Identify AI use cases for analytics using the AI for BI value framework
- ❑ Set up and run an Automated Machine Learning workflow

Thank you!

Keep in touch:

linkedin.com/in/tobias-zwingmann
tobias@rapyd.ai