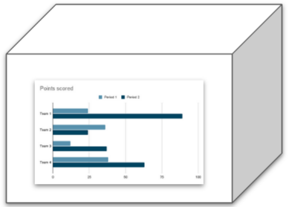
# Interpretable Machine Learning

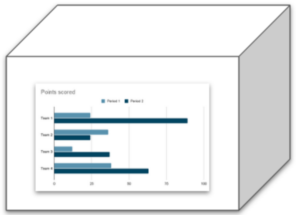
## Inherently Interpretable Models - Motivation



**Learning goals**

- Why should we use interpretable models?
- Advantages and disadvantages of interpretable models

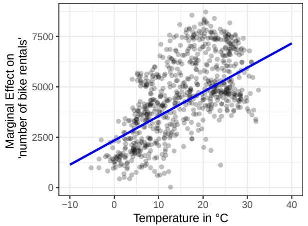
Interpretable Machine Learning
Inherently Interpretable Models - Motivation

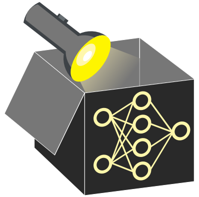


Learning goals
- Why should we use interpretable models?
- Advantages and disadvantages of interpretable models

# MOTIVATION

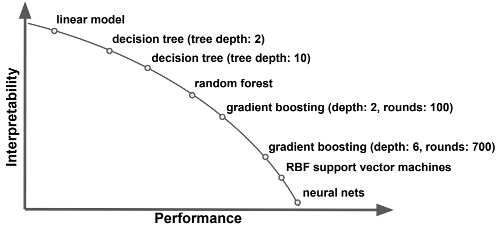
- Achieving interpretability by using interpretable models is the most straightforward approach
- Classes of models deemed interpretable:
  - (Generalized) linear models (LM, GLM)
  - Generalized additive models (GAM)
  - Decision trees
  - Rule-based learning
  - Model-based / component-wise boosting
  - ...



⇝ LM provides straightforward interpretation

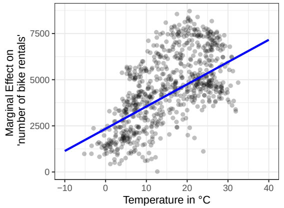
- Often there is a trade-off between interpretability and model performance

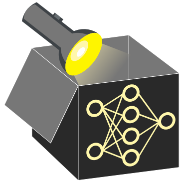





# MOTIVATION

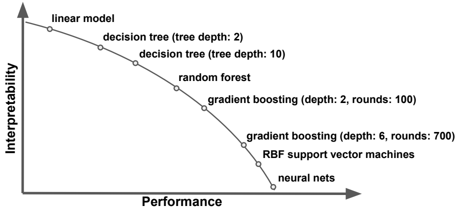
- Achieving interpretability by using interpretable models is the most straightforward approach
- Classes of models deemed interpretable:
  - (Generalized) linear models (LM, GLM)
  - Generalized additive models (GAM)
  - Decision trees
  - Rule-based learning
  - Model-based / component-wise boosting
  - ...



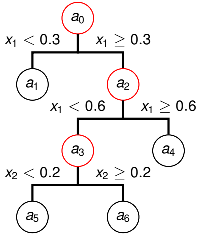
⇝ LM provides straightforward interpretation

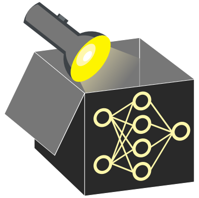
- Often there is a trade-off between interpretability and model performance



# ADVANTAGES

- Interpretable models are transparent by design, making many model-agnostic explanation methods unnecessary
  $\rightsquigarrow$ Eliminates an extra source of estimation error

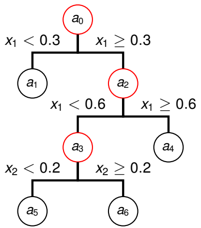


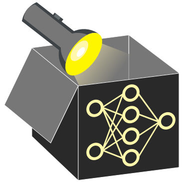





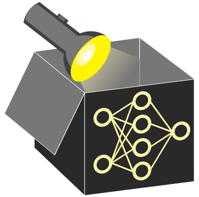
# ADVANTAGES

- Interpretable models are transparent by design, making many model-agnostic explanation methods unnecessary
  $\rightsquigarrow$ Eliminates an extra source of estimation error

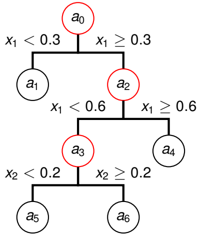


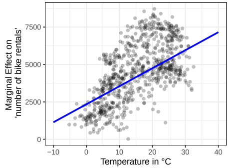



# ADVANTAGES



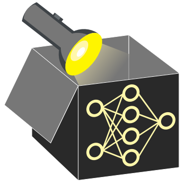
- Interpretable models are transparent by design, making many model-agnostic explanation methods unnecessary
  $\rightsquigarrow$ Eliminates an extra source of estimation error
- They often have few hyperparameters and are structurally simple (e.g., linear, additive, sparse, monotonic)
  $\rightsquigarrow$ Easy to train, fast to tune, and straightforward to explain



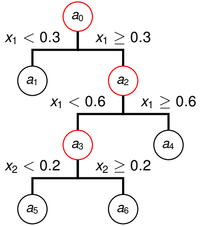



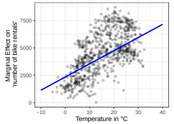



# ADVANTAGES

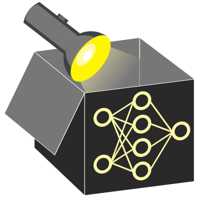


- Interpretable models are transparent by design, making many model-agnostic explanation methods unnecessary
  $\rightsquigarrow$ Eliminates an extra source of estimation error
- They often have few hyperparameters and are structurally simple (e.g., linear, additive, sparse, monotonic)
  $\rightsquigarrow$ Easy to train, fast to tune, straightforward to explain

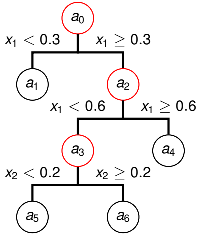


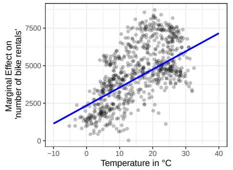



# ADVANTAGES



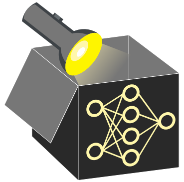
- Interpretable models are transparent by design, making many model-agnostic explanation methods unnecessary
  ⇝ Eliminates an extra source of estimation error
- They often have few hyperparameters and are structurally simple (e.g., linear, additive, sparse, monotonic)
  ⇝ Easy to train, fast to tune, and straightforward to explain
- Many people are familiar with simple interpretable models
  ⇝ Increases trust, facilitates communication of results



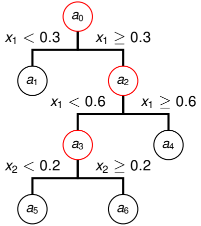



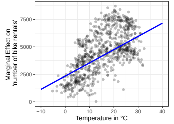



# ADVANTAGES



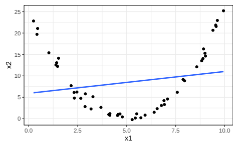
- Interpretable models are transparent by design, making many model-agnostic explanation methods unnecessary
  ⇝ Eliminates an extra source of estimation error
- They often have few hyperparameters and are structurally simple (e.g., linear, additive, sparse, monotonic)
  ⇝ Easy to train, fast to tune, straightforward to explain
- Many people are familiar with simple interpretable models
  ⇝ Increases trust, facilitates communication of results

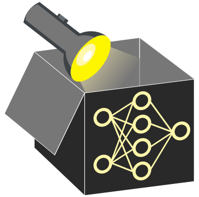




# DISADVANTAGES & LIMITATIONS

- Often require assumptions about data / model structure
  ⇝ If assumptions are wrong, models may perform bad

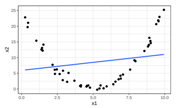


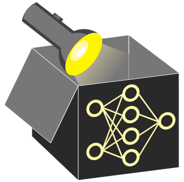





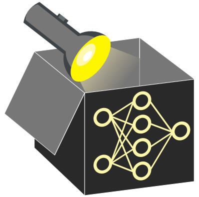
# DISADVANTAGES & LIMITATIONS

- Often require assumptions about data / model structure
  ⇝ If assumptions are wrong, models may perform bad

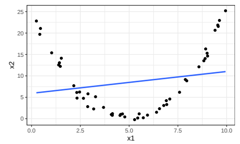




# DISADVANTAGES & LIMITATIONS



- Often require assumptions about data / model structure
  ⇝ If assumptions are wrong, models may perform bad

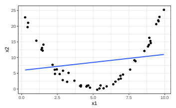


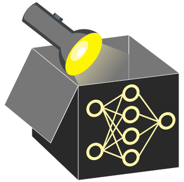

- Interpretable models may also be hard to interpret, e.g.:
  - LM with lots of features and interactions
  - Decision trees with huge tree depth



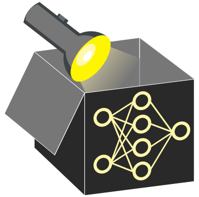
# DISADVANTAGES & LIMITATIONS

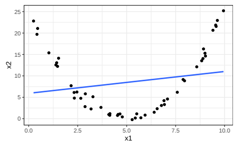
- Often require assumptions about data / model structure
  ⇝ If assumptions are wrong, models may perform bad
- Interpretable models may also be hard to interpret, e.g.:
  - LM with lots of features and interactions
  - Decision trees with huge tree depth





# DISADVANTAGES & LIMITATIONS

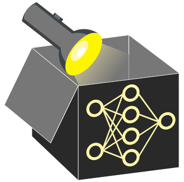


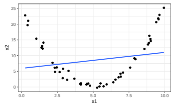



- Often require assumptions about data / model structure
  ⇝ If assumptions are wrong, models may perform bad
- Interpretable models may also be hard to interpret, e.g.:
  - LM with lots of features and interactions
  - Decision trees with huge tree depth
- Often do not automatically model complex relationships due to limited flexibility
  e.g., high-order main or interaction effects need to be specified manually in an LM



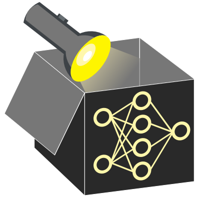
# DISADVANTAGES & LIMITATIONS

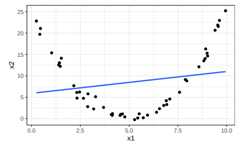




- Often require assumptions about data / model structure
  ⇝ If assumptions are wrong, models may perform bad
- Interpretable models may also be hard to interpret, e.g.:
  - LM with lots of features and interactions
  - Decision trees with huge tree depth
- Often do not automatically model complex relationships due to limited flexibility
  e.g., high-order main or interaction effects need to be specified manually in an LM

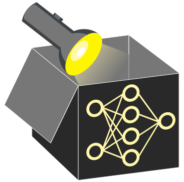
# DISADVANTAGES & LIMITATIONS

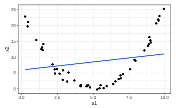





- Often require assumptions about data / model structure
  ⇝ If assumptions are wrong, models may perform bad
- Interpretable models may also be hard to interpret, e.g.:
  - LM with lots of features and interactions
  - Decision trees with huge tree depth
- Often do not automatically model complex relationships due to limited flexibility
  e.g., high-order main or interaction effects need to be specified manually in an LM
- Inherently interpretable models do not address all explanation needs
  ⇝ Complementary model-agnostic methods (e.g., counterfactuals) remain valuable for specific tasks



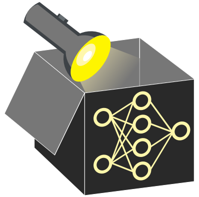




# DISADVANTAGES & LIMITATIONS

- Often require assumptions about data / model structure
  ⇝ If assumptions are wrong, models may perform bad
- Interpretable models may also be hard to interpret, e.g.:
  - LM with lots of features and interactions
  - Decision trees with huge tree depth
- Often do not automatically model complex relationships due to limited flexibility
  e.g., high-order main or interaction effects need to be specified manually in an LM
- Inherently interpretable models do not address all explanation needs
  ⇝ Complementary model-agnostic methods (e.g., counterfactuals) remain valuable for specific tasks

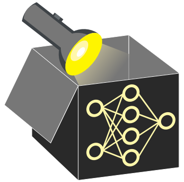
# FURTHER COMMENTS



- Some researchers advocate for inherently interpretable models instead of explaining black boxes after training ▸ Rudin 2019
  - Built-in interpretation ⇒ fewer risks from misleading post-hoc explanations
  - Good performance possible with effort on preprocessing / feat. engineering
  - But interpretability depends on meaning of created features
    ⇝ E.g., PCA keeps models linear, but yields hard-to-interpret components

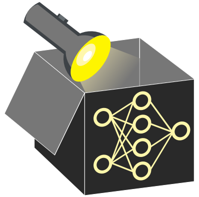


# FURTHER COMMENTS



- Some researchers advocate for inherently interpretable models instead of explaining black boxes after training ▸ Rudin 2019
  - Built-in interpretation
    ⇝ fewer risks from misleading post-hoc explanations
  - Good performance possible with effort on preprocessing and/or feature engineering
  - But interpretability depends on meaning of created features
    ⇝ E.g., PCA keeps models linear, but yields hard-to-interpret components

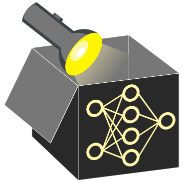
# FURTHER COMMENTS



- Some researchers advocate for inherently interpretable models instead of explaining black boxes after training ▸ Rudin 2019
  - Built-in interpretation ⇒ fewer risks from misleading post-hoc explanations
  - Good performance possible with effort on preprocessing / feat. engineering
  - But interpretability depends on meaning of created features
    ⇝ E.g., PCA keeps models linear, but yields hard-to-interpret components
- Limitation: Less suited for complex data where end-to-end learning is crucial
  - Applies to image, text, or sensor data where features must be learned
  - Manual extraction of interpretable features is difficult
    ⇒ Information loss and lower performance

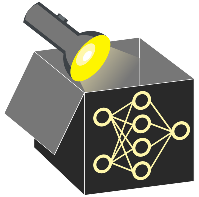


# FURTHER COMMENTS



- Some researchers advocate for inherently interpretable models instead of explaining black boxes after training ▸ Rudin 2019
  - Built-in interpretation
    ⇝ fewer risks from misleading post-hoc explanations
  - Good performance possible with effort on preprocessing and/or feature engineering
  - But interpretability depends on meaning of created features
    ⇝ E.g., PCA keeps models linear, but yields hard-to-interpret components
- Limitation: Less suited for complex data complex data requiring end-to-end learning
  - Applies to image, text, or sensor data where features must be learned from raw input
  - Manual extraction of interpretable features is difficult
    ⇒ Information loss and lower performance

# RECOMMENDATION



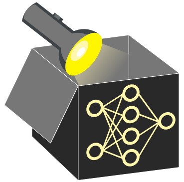
- Begin with the simplest model appropriate for the task
- Increase complexity only if necessary to meet performance requirements
  $\rightsquigarrow$ Typically reduces interpretability and requires model-agnostic explanations
- Choose the simplest model with sufficient accuracy $\rightsquigarrow$ Occam's razor

**Bike Data, 4-fold CV**

| Model | RMSE | $R^2$ |
|---|---|---|
| LM | 800.15 | 0.83 |
| Tree | 981.83 | 0.74 |
| Random Forest | 653.25 | 0.88 |
| Boosting (tuned) | 638.42 | 0.89 |



# RECOMMENDATION



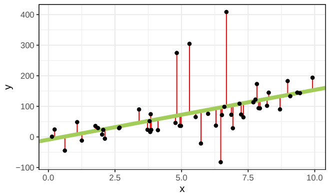
- Begin with the simplest model appropriate for the task
- Increase complexity only if necessary to meet performance requirements
  $\rightsquigarrow$ Typically reduces interpretability and requires model-agnostic explanations
- Choose the simplest model with sufficient accuracy $\rightsquigarrow$ Occam's razor

**Bike Data, 4-fold CV**

| Model | RMSE | $R^2$ |
|---|---|---|
| LM | 800.15 | 0.83 |
| Tree | 981.83 | 0.74 |
| Random Forest | 653.25 | 0.88 |
| Boosting (tuned) | 638.42 | 0.89 |

# Interpretable Machine Learning

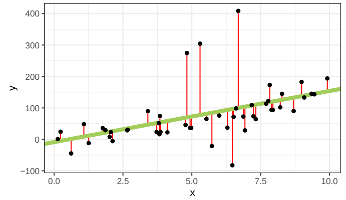
## Linear Regression Model




### Learning goals

- LM basics and assumptions
- Interpretation of main effects in LM
- What are significant features?


Interpretable Machine Learning
Linear Regression Model

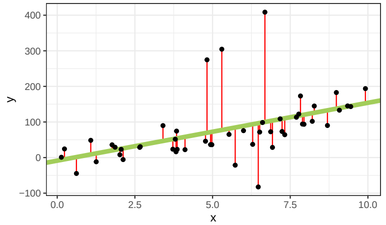





Learning goals
- LM basics and assumptions
- Interpretation of main effects in LM
- What are significant features?

# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

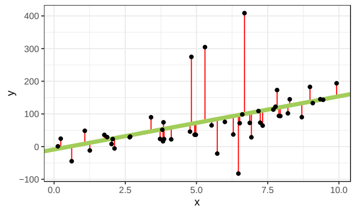



- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\leadsto$ model consists of $p + 1$ weights




# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

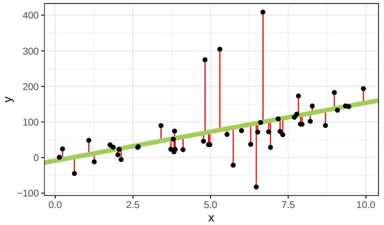





- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\leadsto$ model consists of $p + 1$ weights

# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$




- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

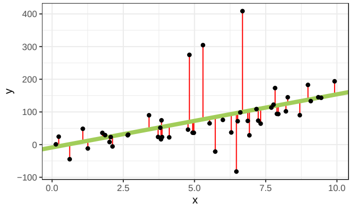
**Properties and assumptions** ▸ Faraway (2002), Ch. 7 ▸ Checking assumptions in R & Python

- **Linear** relationship between features and target



# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$



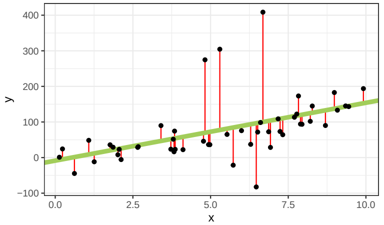

- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

**Properties and assumptions** ▸ “Faraway, Ch. 7” 2002

- **Linear** relationship between features and target

# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$




- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

## Properties and assumptions

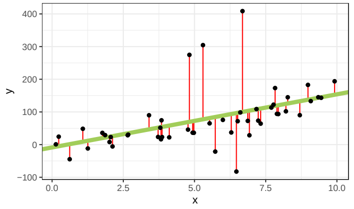
▸ Faraway (2002), Ch. 7 ▸ Checking assumptions in R & Python

- **Linear** relationship between features and target
- $\epsilon$ and $y|\mathbf{x}$ are **normally** distributed with **constant variance** (homoscedastic)
  $\rightsquigarrow \epsilon \sim N(0, \sigma^2) \;\Rightarrow\; (y|\mathbf{x}) \sim N(\mathbf{x}^\top\theta, \sigma^2)$
  $\rightsquigarrow$ if violated, inference-based metrics (e.g., p-values) are invalid



# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$




- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

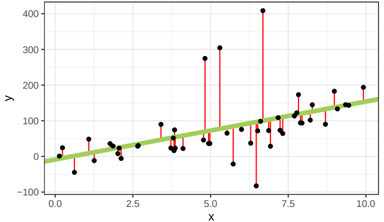
## Properties and assumptions

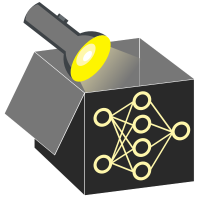
▸ "Faraway, Ch. 7" 2002

- **Linear** relationship between features and target
- $\epsilon$ and $y|\mathbf{x}$ are **normally** distributed with **constant variance** (homoscedastic)
  $\rightsquigarrow \epsilon \sim N(0, \sigma^2) \;\Rightarrow\; (y|\mathbf{x}) \sim N(\mathbf{x}^\top\theta, \sigma^2)$
  $\rightsquigarrow$ if violated, inference-based metrics (e.g., p-values) are invalid

# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$



- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

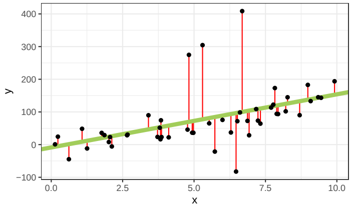
## Properties and assumptions ▸ Faraway (2002), Ch. 7 ▸ Checking assumptions in R & Python

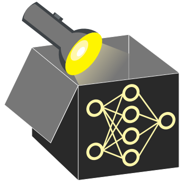
- **Linear** relationship between features and target
- $\epsilon$ and $y|\mathbf{x}$ are **normally** distributed with **constant variance** (homoscedastic)
  $\rightsquigarrow$ $\epsilon \sim N(0, \sigma^2) \;\Rightarrow\; (y|\mathbf{x}) \sim N(\mathbf{x}^\top\theta, \sigma^2)$
  $\rightsquigarrow$ if violated, inference-based metrics (e.g., p-values) are invalid
- Independence of observations (e.g., no repeated measurements)



# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$



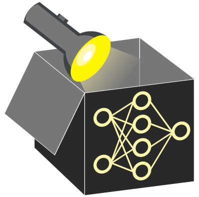



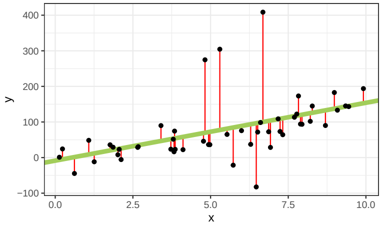
- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

## Properties and assumptions ▸ "Faraway, Ch. 7" 2002

- **Linear** relationship between features and target
- $\epsilon$ and $y|\mathbf{x}$ are **normally** distributed with **constant variance** (homoscedastic)
  $\rightsquigarrow$ $\epsilon \sim N(0, \sigma^2) \;\Rightarrow\; (y|\mathbf{x}) \sim N(\mathbf{x}^\top\theta, \sigma^2)$
  $\rightsquigarrow$ if violated, inference-based metrics (e.g., p-values) are invalid
- Independence of observations (e.g., no repeated measurements)

# LINEAR REGRESSION






$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

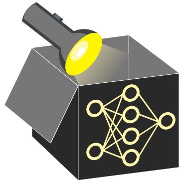
- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

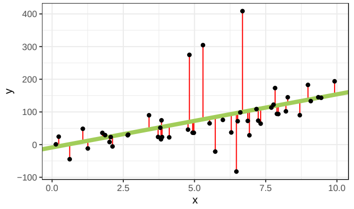
**Properties and assumptions** ‣ Faraway (2002), Ch. 7 ‣ Checking assumptions in R & Python

- **Linear** relationship between features and target
- $\epsilon$ and $y|\mathbf{x}$ are **normally** distributed with **constant variance** (homoscedastic)
  $\rightsquigarrow \epsilon \sim N(0, \sigma^2) \;\Rightarrow\; (y|\mathbf{x}) \sim N(\mathbf{x}^\top\theta, \sigma^2)$
  $\rightsquigarrow$ if violated, inference-based metrics (e.g., p-values) are invalid
- Independence of observations (e.g., no repeated measurements)
- Features $x_j$ independent from error term $\epsilon$



# LINEAR REGRESSION






$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

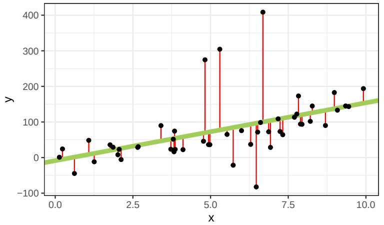
- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

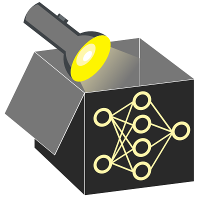
**Properties and assumptions** ‣ "Faraway, Ch. 7" 2002

- **Linear** relationship between features and target
- $\epsilon$ and $y|\mathbf{x}$ are **normally** distributed with **constant variance** (homoscedastic)
  $\rightsquigarrow \epsilon \sim N(0, \sigma^2) \;\Rightarrow\; (y|\mathbf{x}) \sim N(\mathbf{x}^\top\theta, \sigma^2)$
  $\rightsquigarrow$ if violated, inference-based metrics (e.g., p-values) are invalid
- Independence of observations (e.g., no repeated measurements)
- Features $x_j$ independent from error term $\epsilon$

# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$



- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

## Properties and assumptions

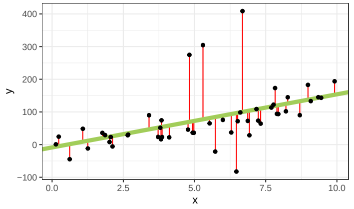
▸ Faraway (2002), Ch. 7 ▸ Checking assumptions in R & Python

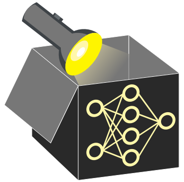
- **Linear** relationship between features and target
- $\epsilon$ and $y|\mathbf{x}$ are **normally** distributed with **constant variance** (homoscedastic)
  $\rightsquigarrow \epsilon \sim N(0, \sigma^2) \;\Rightarrow\; (y|\mathbf{x}) \sim N(\mathbf{x}^\top \theta, \sigma^2)$
  $\rightsquigarrow$ if violated, inference-based metrics (e.g., p-values) are invalid
- Independence of observations (e.g., no repeated measurements)
- Features $x_j$ independent from error term $\epsilon$
- No or little multicollinearity (i.e., no strong feature correlations)



# LINEAR REGRESSION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$



- $y$: target / output
- $\epsilon$: remaining error / residual
- $\theta_j$: weight of input feature $x_j$ (intercept $\theta_0$)
  $\rightsquigarrow$ model consists of $p+1$ weights

## Properties and assumptions

▸ "Faraway, Ch. 7" 2002

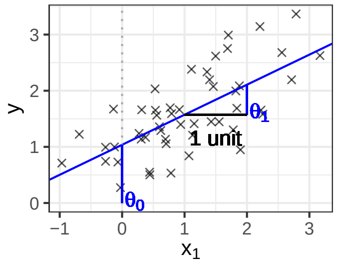
- **Linear** relationship between features and target
- $\epsilon$ and $y|\mathbf{x}$ are **normally** distributed with **constant variance** (homoscedastic)
  $\rightsquigarrow \epsilon \sim N(0, \sigma^2) \;\Rightarrow\; (y|\mathbf{x}) \sim N(\mathbf{x}^\top \theta, \sigma^2)$
  $\rightsquigarrow$ if violated, inference-based metrics (e.g., p-values) are invalid
- Independence of observations (e.g., no repeated measurements)
- Features $x_j$ independent from error term $\epsilon$
- No or little multicollinearity (i.e., no strong feature correlations)

# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

Interpretation of weights (**feature effects**) depend on type of feature:

- **Numerical** $x_j$: Increasing $x_j$ by one unit changes outcome by $\theta_j$, ceteris paribus (*ceteris paribus* (c.p.) means "everything else held constant".)





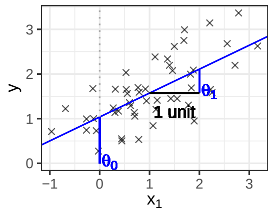


# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

Interpretation of weights (**feature effects**) depend on type of feature:

- **Numerical** $x_j$: Increasing $x_j$ by one unit changes outcome by $\theta_j$, ceteris paribus
(*ceteris paribus* (c.p.) means "everything else held constant".)

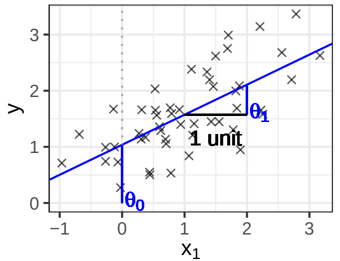



# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

Interpretation of weights (**feature effects**) depend on type of feature:

- **Numerical** $x_j$: Increasing $x_j$ by one unit changes outcome by $\theta_j$, ceteris paribus (*ceteris paribus* (c.p.) means "everything else held constant".)
- **Binary** $x_j$: Weight $\theta_j$ is active or not (multiplication with 1 or 0) $\rightsquigarrow$ reference category $x_j = 0$



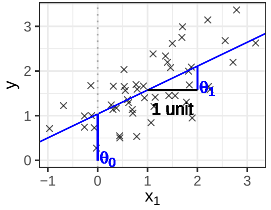



# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

Interpretation of weights (**feature effects**) depend on type of feature:

- **Numerical** $x_j$: Increasing $x_j$ by one unit changes outcome by $\theta_j$, ceteris paribus (*ceteris paribus* (c.p.) means "everything else held constant".)
- **Binary** $x_j$: Weight $\theta_j$ is active or not (multiplication with 1 or 0) $\rightsquigarrow$ reference category $x_j = 0$



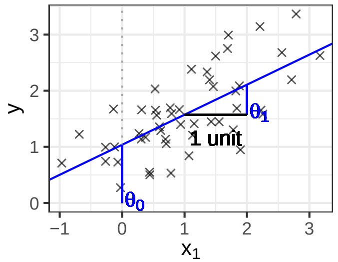
# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

Interpretation of weights (**feature effects**) depend on type of feature:

- **Numerical** $x_j$: Increasing $x_j$ by one unit changes outcome by $\theta_j$, ceteris paribus (*ceteris paribus* (c.p.) means "everything else held constant".)
- **Binary** $x_j$: Weight $\theta_j$ is active or not (multiplication with 1 or 0)
  $\rightsquigarrow$ reference category $x_j = 0$
- **Categorical feature** $x_j$ **with** $L$ **categories**:
  - Create $L - 1$ one-hot-encoded features $x_{j,1}, \ldots, x_{j,L-1}$ (each having its own weight)
  - Left out cat. is reference ($\hat{=}$ dummy encoding)

  $\rightsquigarrow$ Interpretation: Outcome changes by $\theta_{j,i}$ for category $i$ compared to reference cat., c.p.







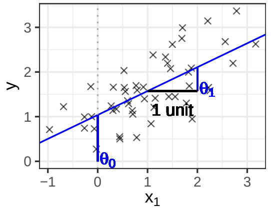
# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

Interpretation of weights (**feature effects**) depend on type of feature:

- **Numerical** $x_j$: Increasing $x_j$ by one unit changes outcome by $\theta_j$, ceteris paribus (*ceteris paribus* (c.p.) means "everything else held constant".)
- **Binary** $x_j$: Weight $\theta_j$ is active or not (multiplication with 1 or 0)
  $\rightsquigarrow$ reference category $x_j = 0$
- **Categorical feature** $x_j$ **with** $L$ **categories**:
  - Create $L - 1$ one-hot-encoded features $x_{j,1}, \ldots, x_{j,L-1}$ (each having its own weight)
  - Left out cat. is reference ($\hat{=}$ dummy encoding)

  $\rightsquigarrow$ Interpretation: Outcome changes by $\theta_{j,i}$ for category $i$ compared to reference cat., c.p.

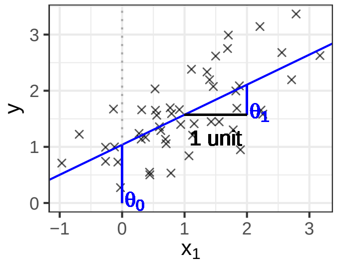



# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

Interpretation of weights (**feature effects**) depend on type of feature:

- **Numerical** $x_j$: Increasing $x_j$ by one unit changes outcome by $\theta_j$, ceteris paribus (*ceteris paribus* (c.p.) means "everything else held constant".)
- **Binary** $x_j$: Weight $\theta_j$ is active or not (multiplication with 1 or 0)
  $\rightsquigarrow$ reference category $x_j = 0$
- **Categorical feature** $x_j$ **with** $L$ **categories**:
  - Create $L-1$ one-hot-encoded features $x_{j,1}, \ldots, x_{j,L-1}$ (each having its own weight)
  - Left out cat. is reference ($\widehat{=}$ dummy encoding)
  
  $\rightsquigarrow$ Interpretation: Outcome changes by $\theta_{j,i}$ for category $i$ compared to reference cat., c.p.
- **Intercept** $\theta_0$: Expected outcome if all feature values are set to 0



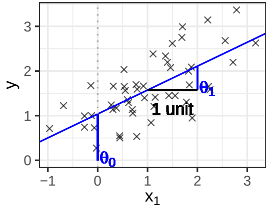



# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

Interpretation of weights (**feature effects**) depend on type of feature:

- **Numerical** $x_j$: Increasing $x_j$ by one unit changes outcome by $\theta_j$, ceteris paribus
  (*ceteris paribus* (c.p.) means "everything else held constant".)
- **Binary** $x_j$: Weight $\theta_j$ is active or not (multiplication with 1 or 0)
  $\rightsquigarrow$ reference category $x_j = 0$
- **Categorical feature** $x_j$ **with** $L$ **categories**:
  - Create $L-1$ one-hot-encoded features $x_{j,1}, \ldots, x_{j,L-1}$ (each having its own weight)
  - Left out cat. is reference ($\widehat{=}$ dummy encoding)
  
  $\rightsquigarrow$ Interpretation: Outcome changes by $\theta_{j,i}$ for category $i$ compared to reference cat., c.p.
- **Intercept** $\theta_0$: Expected outcome if all feature values are set to 0



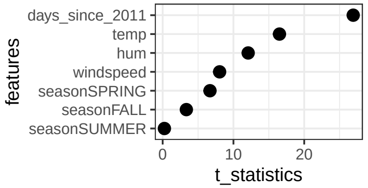
# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

**Feature importance**:

- Absolute **t-statistic** value: $\hat{\theta}_j$ scaled with standard error ($SE(\hat{\theta}_j)$ $\hat{=}$ reliability of estimate)

$$|t_{\hat{\theta}_j}| = \left| \frac{\hat{\theta}_j}{SE(\hat{\theta}_j)} \right|$$




- High $t$-values $\Rightarrow$ important (significant) feat.



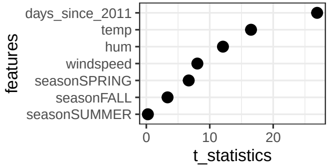
# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

**Feature importance**:

- Absolute **t-statistic** value: $\hat{\theta}_j$ scaled with standard error ($SE(\hat{\theta}_j)$ $\hat{=}$ reliability of estimate)

$$|t_{\hat{\theta}_j}| = \left| \frac{\hat{\theta}_j}{SE(\hat{\theta}_j)} \right|$$




- High $t$-values $\Rightarrow$ important (significant) feat.

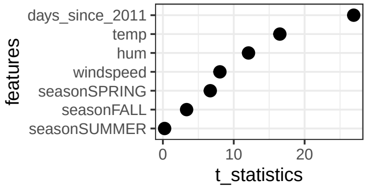
# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

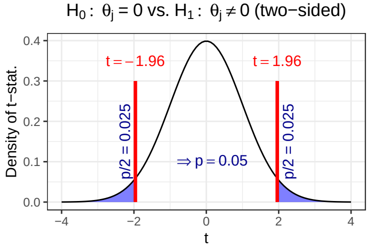
**Feature importance**:

- Absolute **t-statistic** value: $\hat{\theta}_j$ scaled with standard error ($SE(\hat{\theta}_j)$) $\hat{=}$ reliability of estimate)

$$|t_{\hat{\theta}_j}| = \left| \frac{\hat{\theta}_j}{SE(\hat{\theta}_j)} \right|$$




- High $t$-values $\Rightarrow$ important (significant) feat.
- **p-value**: probability of obtaining a more extreme test statistic assuming $H_0$ is correct (here: $\theta_j = 0$, i.e., feat. $j$ not significant) $\rightsquigarrow$ High $|t|$ $\Rightarrow$ small p-val. (speak against $H_0$)







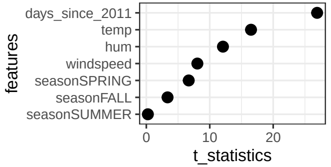
# LINEAR REGRESSION - INTERPRETATION

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon$$

**Feature importance**:

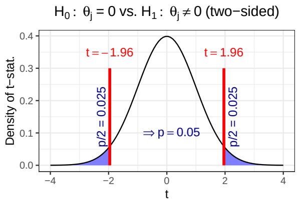
- Absolute **t-statistic** value: $\hat{\theta}_j$ scaled with standard error ($SE(\hat{\theta}_j)$) $\hat{=}$ reliability of estimate)

$$|t_{\hat{\theta}_j}| = \left| \frac{\hat{\theta}_j}{SE(\hat{\theta}_j)} \right|$$

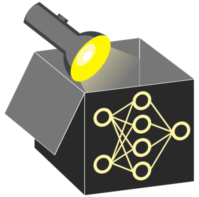


- High $t$-values $\Rightarrow$ important (significant) feat.
- **p-value**: probability of obtaining a more extreme test statistic assuming $H_0$ is correct (here: $\theta_j = 0$, i.e., feat. $j$ not significant) $\rightsquigarrow$ High $|t|$ $\Rightarrow$ small p-val. (speak against $H_0$)

# EXAMPLE: LINEAR REGRESSION - MAIN EFFECTS

**Bike data**: predict no. of rented bikes using 4 numeric, 1 categorical feat. (season)



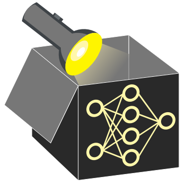
$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 \mathbb{1}_{x_{season}=SPRING} + \hat{\theta}_2 \mathbb{1}_{x_{season}=SUMMER} + \hat{\theta}_3 \mathbb{1}_{x_{season}=FALL} + \hat{\theta}_4 x_{temp} + \hat{\theta}_5 x_{hum} + \hat{\theta}_6 x_{windspeed} + \hat{\theta}_7 x_{days\_since\_2011}$$

| | Weights | SE | t-stat. | p-val. |
|---|---|---|---|---|
| (Intercept) | 3229.3 | 220.6 | 14.6 | 0.00 |
| seasonSPRING | 862.0 | 129.0 | 6.7 | 0.00 |
| seasonSUMMER | 41.6 | 170.2 | 0.2 | 0.81 |
| seasonFALL | 390.1 | 116.6 | 3.3 | 0.00 |
| temp | 120.5 | 7.3 | 16.5 | 0.00 |
| hum | -31.1 | 2.6 | -12.1 | 0.00 |
| windspeed | -56.9 | 7.1 | -8.0 | 0.00 |
| days_since_2011 | 4.9 | 0.2 | 26.9 | 0.00 |



# EXAMPLE: LIN. REGRESSION - MAIN EFFECTS

**Bike data**: predict no. of rented bikes using 4 numeric, 1 cat. feat. (season)



$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 \mathbb{1}_{x_{season}=SPRING} + \hat{\theta}_2 \mathbb{1}_{x_{season}=SUMMER} + \hat{\theta}_3 \mathbb{1}_{x_{season}=FALL} + \hat{\theta}_4 x_{temp} + \hat{\theta}_5 x_{hum} + \hat{\theta}_6 x_{windspeed} + \hat{\theta}_7 x_{days\_since\_2011}$$

| | Weights | SE | t-stat. | p-val. |
|---|---|---|---|---|
| (Intercept) | 3229.3 | 220.6 | 14.6 | 0.00 |
| seasonSPRING | 862.0 | 129.0 | 6.7 | 0.00 |
| seasonSUMMER | 41.6 | 170.2 | 0.2 | 0.81 |
| seasonFALL | 390.1 | 116.6 | 3.3 | 0.00 |
| temp | 120.5 | 7.3 | 16.5 | 0.00 |
| hum | -31.1 | 2.6 | -12.1 | 0.00 |
| windspeed | -56.9 | 7.1 | -8.0 | 0.00 |
| days_since_2011 | 4.9 | 0.2 | 26.9 | 0.00 |

# EXAMPLE: LINEAR REGRESSION - MAIN EFFECTS

**Bike data**: predict no. of rented bikes using 4 numeric, 1 categorical feat. (season)

$$
\begin{aligned}
\hat{y} = &\hat{\theta}_0 + \hat{\theta}_1 \mathbb{1}_{x_{season}=SPRING} + \\
&\hat{\theta}_2 \mathbb{1}_{x_{season}=SUMMER} + \\
&\hat{\theta}_3 \mathbb{1}_{x_{season}=FALL} + \hat{\theta}_4 x_{temp} + \\
&\hat{\theta}_5 x_{hum} + \hat{\theta}_6 x_{windspeed} + \\
&\hat{\theta}_7 x_{days\_since\_2011}
\end{aligned}
$$

| | Weights | SE | t-stat. | p-val. |
|---:|---:|---:|---:|---:|
| (Intercept) | 3229.3 | 220.6 | 14.6 | 0.00 |
| seasonSPRING | 862.0 | 129.0 | 6.7 | 0.00 |
| seasonSUMMER | 41.6 | 170.2 | 0.2 | 0.81 |
| seasonFALL | 390.1 | 116.6 | 3.3 | 0.00 |
| temp | 120.5 | 7.3 | 16.5 | 0.00 |
| hum | -31.1 | 2.6 | -12.1 | 0.00 |
| windspeed | -56.9 | 7.1 | -8.0 | 0.00 |
| days_since_2011 | 4.9 | 0.2 | 26.9 | 0.00 |

**Interpretation:**

- **Intercept**: If all feature values are 0 (and season is `WINTER` $\hat{=}$ reference cat.), the expected number of bike rentals is $\hat{\theta}_0 = 3229.3$



# EXAMPLE: LIN. REGRESSION - MAIN EFFECTS

**Bike data**: predict no. of rented bikes using 4 numeric, 1 cat. feat. (season)

$$
\begin{aligned}
\hat{y} = &\hat{\theta}_0 + \hat{\theta}_1 \mathbb{1}_{x_{season}=SPRING} + \\
&\hat{\theta}_2 \mathbb{1}_{x_{season}=SUMMER} + \\
&\hat{\theta}_3 \mathbb{1}_{x_{season}=FALL} + \hat{\theta}_4 x_{temp} + \\
&\hat{\theta}_5 x_{hum} + \hat{\theta}_6 x_{windspeed} + \\
&\hat{\theta}_7 x_{days\_since\_2011}
\end{aligned}
$$

| | Weights | SE | t-stat. | p-val. |
|---:|---:|---:|---:|---:|
| (Intercept) | 3229.3 | 220.6 | 14.6 | 0.00 |
| seasonSPRING | 862.0 | 129.0 | 6.7 | 0.00 |
| seasonSUMMER | 41.6 | 170.2 | 0.2 | 0.81 |
| seasonFALL | 390.1 | 116.6 | 3.3 | 0.00 |
| temp | 120.5 | 7.3 | 16.5 | 0.00 |
| hum | -31.1 | 2.6 | -12.1 | 0.00 |
| windspeed | -56.9 | 7.1 | -8.0 | 0.00 |
| days_since_2011 | 4.9 | 0.2 | 26.9 | 0.00 |

**Interpretation:**

- **Intercept**: If all feature values are 0 (and season is `WINTER` $\hat{=}$ reference cat.), the expected number of bike rentals is $\hat{\theta}_0 = 3229.3$

# EXAMPLE: LINEAR REGRESSION - MAIN EFFECTS

**Bike data**: predict no. of rented bikes using 4 numeric, 1 categorical feat. (season)

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 \mathbb{1}_{x_{season}=SPRING} + \hat{\theta}_2 \mathbb{1}_{x_{season}=SUMMER} + \hat{\theta}_3 \mathbb{1}_{x_{season}=FALL} + \hat{\theta}_4 x_{temp} + \hat{\theta}_5 x_{hum} + \hat{\theta}_6 x_{windspeed} + \hat{\theta}_7 x_{days\_since\_2011}$$

| | Weights | SE | t-stat. | p-val. |
|---|---|---|---|---|
| (Intercept) | 3229.3 | 220.6 | 14.6 | 0.00 |
| seasonSPRING | 862.0 | 129.0 | 6.7 | 0.00 |
| seasonSUMMER | 41.6 | 170.2 | 0.2 | 0.81 |
| seasonFALL | 390.1 | 116.6 | 3.3 | 0.00 |
| temp | 120.5 | 7.3 | 16.5 | 0.00 |
| hum | -31.1 | 2.6 | -12.1 | 0.00 |
| windspeed | -56.9 | 7.1 | -8.0 | 0.00 |
| days_since_2011 | 4.9 | 0.2 | 26.9 | 0.00 |

**Interpretation:**

- **Intercept**: If all feature values are 0 (and season is `WINTER` $\hat{=}$ reference cat.), the expected number of bike rentals is $\hat{\theta}_0 = 3229.3$
- **Categorical**: Rentals in `SPRING` are by $\hat{\theta}_1 = 862$ higher than in `WINTER`, c.p.



# EXAMPLE: LIN. REGRESSION - MAIN EFFECTS

**Bike data**: predict no. of rented bikes using 4 numeric, 1 cat. feat. (season)

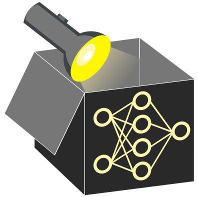
$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 \mathbb{1}_{x_{season}=SPRING} + \hat{\theta}_2 \mathbb{1}_{x_{season}=SUMMER} + \hat{\theta}_3 \mathbb{1}_{x_{season}=FALL} + \hat{\theta}_4 x_{temp} + \hat{\theta}_5 x_{hum} + \hat{\theta}_6 x_{windspeed} + \hat{\theta}_7 x_{days\_since\_2011}$$


| | Weights | SE | t-stat. | p-val. |
|---|---|---|---|---|
| (Intercept) | 3229.3 | 220.6 | 14.6 | 0.00 |
| seasonSPRING | 862.0 | 129.0 | 6.7 | 0.00 |
| seasonSUMMER | 41.6 | 170.2 | 0.2 | 0.81 |
| seasonFALL | 390.1 | 116.6 | 3.3 | 0.00 |
| temp | 120.5 | 7.3 | 16.5 | 0.00 |
| hum | -31.1 | 2.6 | -12.1 | 0.00 |
| windspeed | -56.9 | 7.1 | -8.0 | 0.00 |
| days_since_2011 | 4.9 | 0.2 | 26.9 | 0.00 |


**Interpretation:**

- **Intercept**: If all feature values are 0 (and season is `WINTER` $\hat{=}$ reference cat.), the expected number of bike rentals is $\hat{\theta}_0 = 3229.3$
- **Categ.**: Rentals in `SPRING` are by $\hat{\theta}_1 = 862$ higher than in `WINTER`, c.p.

# EXAMPLE: LINEAR REGRESSION - MAIN EFFECTS



**Bike data**: predict no. of rented bikes using 4 numeric, 1 categorical feat. (season)

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 \mathbb{1}_{x_{season}=SPRING} + \hat{\theta}_2 \mathbb{1}_{x_{season}=SUMMER} + \hat{\theta}_3 \mathbb{1}_{x_{season}=FALL} + \hat{\theta}_4 x_{temp} + \hat{\theta}_5 x_{hum} + \hat{\theta}_6 x_{windspeed} + \hat{\theta}_7 x_{days\_since\_2011}$$

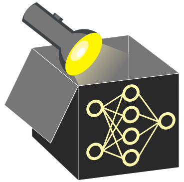
| | Weights | SE | t-stat. | p-val. |
|---|---|---|---|---|
| (Intercept) | 3229.3 | 220.6 | 14.6 | 0.00 |
| seasonSPRING | 862.0 | 129.0 | 6.7 | 0.00 |
| seasonSUMMER | 41.6 | 170.2 | 0.2 | 0.81 |
| seasonFALL | 390.1 | 116.6 | 3.3 | 0.00 |
| temp | 120.5 | 7.3 | 16.5 | 0.00 |
| hum | -31.1 | 2.6 | -12.1 | 0.00 |
| windspeed | -56.9 | 7.1 | -8.0 | 0.00 |
| days_since_2011 | 4.9 | 0.2 | 26.9 | 0.00 |

**Interpretation:**

- **Intercept**: If all feature values are 0 (and season is `WINTER` $\hat{=}$ reference cat.), the expected number of bike rentals is $\hat{\theta}_0 = 3229.3$
- **Categorical**: Rentals in `SPRING` are by $\hat{\theta}_1 = 862$ higher than in `WINTER`, c.p.
- **Numerical**: Rentals increase by $\hat{\theta}_4 = 120.5$ if `temp` increases by 1 °C, c.p.



# EXAMPLE: LIN. REGRESSION - MAIN EFFECTS



**Bike data**: predict no. of rented bikes using 4 numeric, 1 cat. feat. (season)

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 \mathbb{1}_{x_{season}=SPRING} + \hat{\theta}_2 \mathbb{1}_{x_{season}=SUMMER} + \hat{\theta}_3 \mathbb{1}_{x_{season}=FALL} + \hat{\theta}_4 x_{temp} + \hat{\theta}_5 x_{hum} + \hat{\theta}_6 x_{windspeed} + \hat{\theta}_7 x_{days\_since\_2011}$$

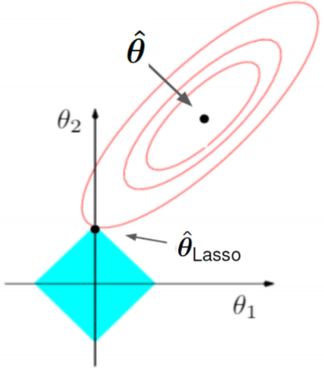
| | Weights | SE | t-stat. | p-val. |
|---|---|---|---|---|
| (Intercept) | 3229.3 | 220.6 | 14.6 | 0.00 |
| seasonSPRING | 862.0 | 129.0 | 6.7 | 0.00 |
| seasonSUMMER | 41.6 | 170.2 | 0.2 | 0.81 |
| seasonFALL | 390.1 | 116.6 | 3.3 | 0.00 |
| temp | 120.5 | 7.3 | 16.5 | 0.00 |
| hum | -31.1 | 2.6 | -12.1 | 0.00 |
| windspeed | -56.9 | 7.1 | -8.0 | 0.00 |
| days_since_2011 | 4.9 | 0.2 | 26.9 | 0.00 |

**Interpretation:**

- **Intercept**: If all feature values are 0 (and season is `WINTER` $\hat{=}$ reference cat.), the expected number of bike rentals is $\hat{\theta}_0 = 3229.3$
- **Categ.**: Rentals in `SPRING` are by $\hat{\theta}_1 = 862$ higher than in `WINTER`, c.p.
- **Numerical**: Rentals increase by $\hat{\theta}_4 = 120.5$ if `temp` increases by 1 °C, c.p.

# Interpretable Machine Learning

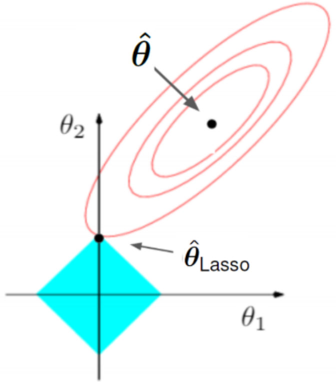
# Extensions of Linear Regression Models




**Learning goals**

- Inclusion of high-order and interaction effects
- Regularization via LASSO


# Interpretable Machine Learning
# Extensions of Linear Regression Models




**Learning goals**

- Inclusion of high-order and interaction effects
- Regularization via LASSO

# INTERACTION AND HIGH-ORDER EFFECTS

LM Equation: $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon$

Equation above can be extended (polynomial regression) by including

- **high-order effects** which have their own weights
  $\rightsquigarrow$ e.g., quadratic effect: $\theta_{x_j^2} \cdot x_j^2$
- **interaction effects** as the product of multiple feat.
  $\rightsquigarrow$ e.g., 2-way interaction: $\theta_{x_i,x_j} \cdot x_i \cdot x_j$

**Bike Data**

| Method | $R^2$ | adj. $R^2$ |
|---|---|---|
| Simple LM | 0.85 | 0.84 |
| High-order | 0.87 | 0.87 |
| Interaction | 0.96 | 0.93 |




# INTERACTION AND HIGH-ORDER EFFECTS

LM Equation: $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon$

Equation above can be extended (polynomial regression) by including

- **high-order effects** which have their own weights
  $\rightsquigarrow$ e.g., quadratic effect: $\theta_{x_j^2} \cdot x_j^2$
- **interaction effects** as the product of multiple feat.
  $\rightsquigarrow$ e.g., 2-way interaction: $\theta_{x_i,x_j} \cdot x_i \cdot x_j$

**Bike Data**

| Method | $R^2$ | adj. $R^2$ |
|---|---|---|
| Simple LM | 0.85 | 0.84 |
| High-order | 0.87 | 0.87 |
| Interaction | 0.96 | 0.93 |

# INTERACTION AND HIGH-ORDER EFFECTS

LM Equation: $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon$

Equation above can be extended (polynomial regression) by including

- **high-order effects** which have their own weights
  $\rightsquigarrow$ e.g., quadratic effect: $\theta_{x_j^2} \cdot x_j^2$
- **interaction effects** as the product of multiple feat.
  $\rightsquigarrow$ e.g., 2-way interaction: $\theta_{x_i, x_j} \cdot x_i \cdot x_j$

**Bike Data**

| Method | $R^2$ | adj. $R^2$ |
|---|---|---|
| Simple LM | 0.85 | 0.84 |
| High-order | 0.87 | 0.87 |
| Interaction | 0.96 | 0.93 |

Implications of including high-order and interaction effects:

- Both make the model more flexible but also less interpretable
  $\rightsquigarrow$ More weights to interpret
- Both need to be specified manually (inconvenient and sometimes infeasible)
  $\rightsquigarrow$ Other ML models often learn them automatically
- Marginal effect of a feature cannot be interpreted by single weights anymore
  $\rightsquigarrow$ Feature $x_j$ occurs multiple times (with different weights) in equation



# INTERACTION AND HIGH-ORDER EFFECTS

LM Equation: $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p + \epsilon$

Equation above can be extended (polynomial regression) by including

- **high-order effects** which have their own weights
  $\rightsquigarrow$ e.g., quadratic effect: $\theta_{x_j^2} \cdot x_j^2$
- **interaction effects** as the product of multiple feat.
  $\rightsquigarrow$ e.g., 2-way interaction: $\theta_{x_i, x_j} \cdot x_i \cdot x_j$

**Bike Data**

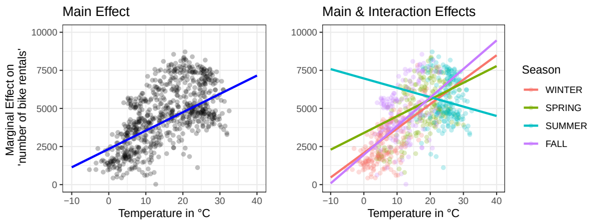
| Method | $R^2$ | adj. $R^2$ |
|---|---|---|
| Simple LM | 0.85 | 0.84 |
| High-order | 0.87 | 0.87 |
| Interaction | 0.96 | 0.93 |

Implications of including high-order and interaction effects:

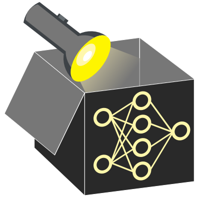
- Both make the model more flexible but also less interpretable
  $\rightsquigarrow$ More weights to interpret
- Both need to be specified manually (inconvenient, sometimes infeasible)
  $\rightsquigarrow$ Other ML models often learn them automatically
- Marginal effect of a feat. cannot be interpreted by single weights anymore
  $\rightsquigarrow$ Feature $x_j$ occurs multiple times (with different weights) in equation

# EXAMPLE: INTERACTION EFFECT

**Example**: Interaction between `temp` and `season` will affect marginal effect of `temp`



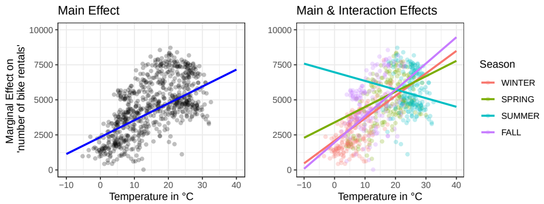

| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |





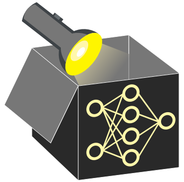
# EXAMPLE: INTERACTION EFFECT

**Ex.**: Interaction between `temp` and `season` will affect marginal effect of `temp`



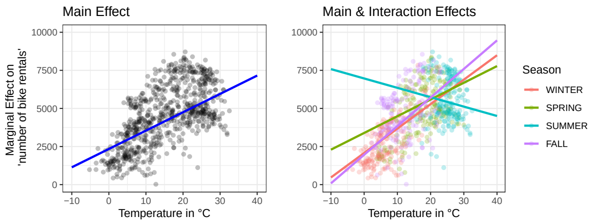


| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |

# EXAMPLE: INTERACTION EFFECT

**Example**: Interaction between `temp` and `season` will affect marginal effect of `temp`




| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |

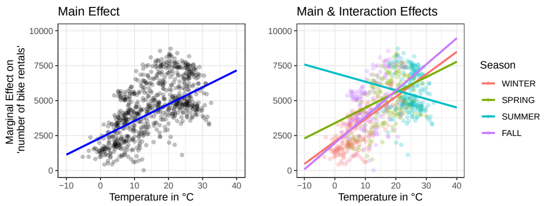
**Interpretation**: If `temp` increases by 1 °C, bike rentals

- increase by 160.5 in `WINTER` (reference)



# EXAMPLE: INTERACTION EFFECT

**Ex.**: Interaction between `temp` and `season` will affect marginal effect of `temp`



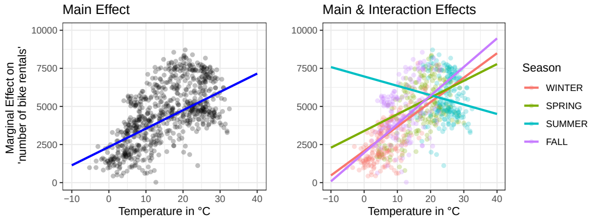

| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |

**Interpretation**: If `temp` increases by 1 °C, bike rentals

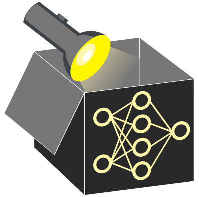
- increase by 160.5 in `WINTER` (reference)

# EXAMPLE: INTERACTION EFFECT

**Example**: Interaction between `temp` and `season` will affect marginal effect of `temp`




| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |



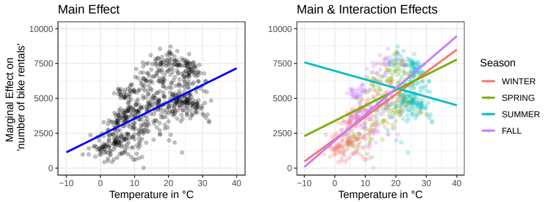
**Interpretation**: If `temp` increases by 1 °C, bike rentals

- increase by 160.5 in `WINTER` (reference)
- increase by 109.8 (= 160.5 - 50.7) in `SPRING`

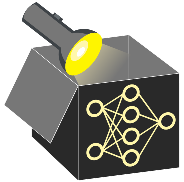


# EXAMPLE: INTERACTION EFFECT

**Ex.**: Interaction between `temp` and `season` will affect marginal effect of `temp`




| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |

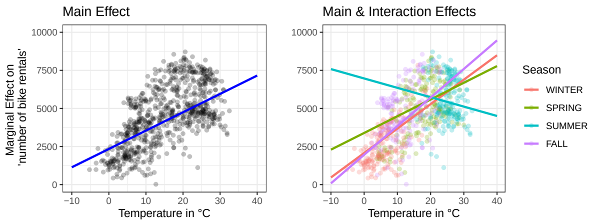


**Interpretation**: If `temp` increases by 1 °C, bike rentals

- increase by 160.5 in `WINTER` (reference)
- increase by 109.8 (= 160.5 - 50.7) in `SPRING`

# EXAMPLE: INTERACTION EFFECT

**Example**: Interaction between `temp` and `season` will affect marginal effect of `temp`




| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |

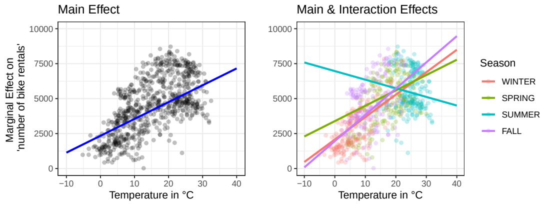
**Interpretation**: If `temp` increases by 1 °C, bike rentals

- increase by 160.5 in `WINTER` (reference)
- increase by 109.8 (= 160.5 - 50.7) in `SPRING`
- decrease by -61.5 (= 160.5 - 222) in `SUMMER`



# EXAMPLE: INTERACTION EFFECT

**Ex.**: Interaction between `temp` and `season` will affect marginal effect of `temp`





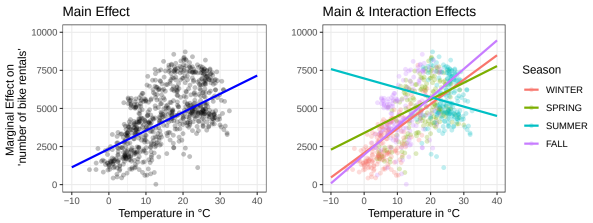
| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |


**Interpretation**: If `temp` increases by 1 °C, bike rentals


- increase by 160.5 in `WINTER` (reference)
- increase by 109.8 (= 160.5 - 50.7) in `SPRING`
- decrease by -61.5 (= 160.5 - 222) in `SUMMER`

# EXAMPLE: INTERACTION EFFECT

**Example**: Interaction between `temp` and `season` will affect marginal effect of `temp`




| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |

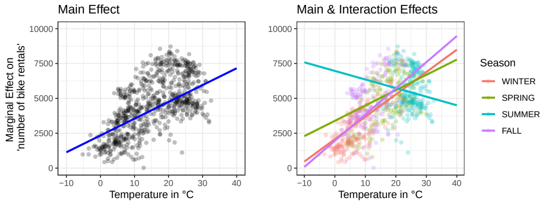
**Interpretation**: If `temp` increases by 1 °C, bike rentals

- increase by 160.5 in `WINTER` (reference)
- increase by 109.8 (= 160.5 - 50.7) in `SPRING`
- decrease by -61.5 (= 160.5 - 222) in `SUMMER`
- increase by 187.7 (= 160.5 + 27.2) in `FALL`



# EXAMPLE: INTERACTION EFFECT

**Ex.**: Interaction between `temp` and `season` will affect marginal effect of `temp`



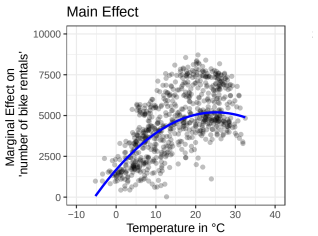

| | Weights |
|---|---|
| (Intercept) | 3453.9 |
| seasonSPRING | 1317.0 |
| seasonSUMMER | 4894.1 |
| seasonFALL | -114.2 |
| temp | 160.5 |
| hum | -37.6 |
| windspeed | -61.9 |
| days_since_2011 | 4.9 |
| seasonSPRING:temp | -50.7 |
| seasonSUMMER:temp | -222.0 |
| seasonFALL:temp | 27.2 |

**Interpretation**: If `temp` increases by 1 °C,
bike rentals

- increase by 160.5 in `WINTER` (reference)
- increase by 109.8 (= 160.5 - 50.7) in `SPRING`
- decrease by -61.5 (= 160.5 - 222) in `SUMMER`
- increase by 187.7 (= 160.5 + 27.2) in `FALL`

# EXAMPLE: QUADRATIC EFFECT

**Example**: Adding quadratic effect for `temp`




| | Weights |
|---|---|
| (Intercept) | 3094.1 |
| seasonSPRING | 619.2 |
| seasonSUMMER | 284.6 |
| seasonFALL | 123.1 |
| hum | -36.4 |
| windspeed | -65.7 |
| days_since_2011 | 4.7 |
| temp | 280.2 |
| temp$^2$ | -5.6 |

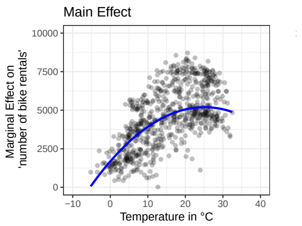
**Interpretation**: Not linear anymore!

- `temp` depends on two weights: $280.2 \cdot x_{temp} - 5.6 \cdot x_{temp}^2$



# EXAMPLE: QUADRATIC EFFECT

**Ex.**: Adding quadratic effect for `temp`



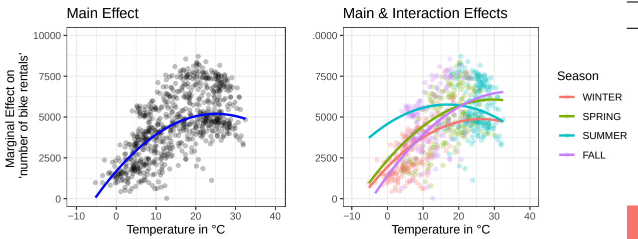

| | Weights |
|---|---|
| (Intercept) | 3094.1 |
| seasonSPRING | 619.2 |
| seasonSUMMER | 284.6 |
| seasonFALL | 123.1 |
| hum | -36.4 |
| windspeed | -65.7 |
| days_since_2011 | 4.7 |
| temp | 280.2 |
| temp$^2$ | -5.6 |

**Interpretation**: Not linear anymore!

- `temp` depends on two weights: $280.2 \cdot x_{temp} - 5.6 \cdot x_{temp}^2$

# EXAMPLE: QUADRATIC EFFECT

**Example**: Adding quadratic effect for `temp` (left) and interaction with `season` (right)




| | Weights |
|---|---|
| (Intercept) | 3802.1 |
| seasonSPRING | -1345.1 |
| seasonSUMMER | -6006.3 |
| seasonFALL | -681.4 |
| hum | -38.9 |
| windspeed | -64.1 |
| days_since_2011 | 4.8 |
| temp | 39.1 |
| temp$^2$ | 8.6 |
| seasonSPRING:temp | 407.4 |
| seasonSPRING:temp$^2$ | -18.7 |
| seasonSUMMER:temp | 801.1 |
| seasonSUMMER:temp$^2$ | -27.2 |
| seasonFALL:temp | 217.4 |
| seasonFALL:temp$^2$ | -11.3 |

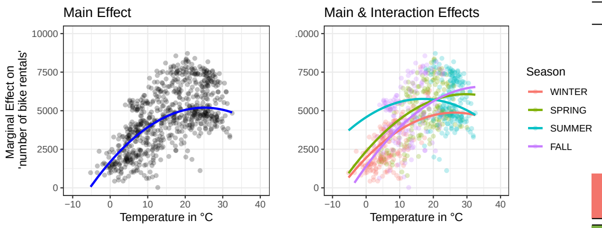
**Interpretation**: Not linear anymore!

- `temp` depends on multiple weights due to `season`:
  - $\rightsquigarrow$ `WINTER`: $39.1 \cdot x_{temp} + 8.6 \cdot x^2_{temp}$
  - $\rightsquigarrow$ `SPRING`: $(39.1+407.4) \cdot x_{temp} + (8.6-18.7) \cdot x^2_{temp}$
  - $\rightsquigarrow$ `SUMMER`: $(39.1+801.1) \cdot x_{temp} + (8.6-27.2) \cdot x^2_{temp}$
  - $\rightsquigarrow$ `FALL`: $(39.1+217.4) \cdot x_{temp} + (8.6-11.3) \cdot x^2_{temp}$



# EXAMPLE: QUADRATIC EFFECT

**Ex.**: Adding quadratic effect for `temp` (left) and interaction with `season` (right)




| | Weights |
|---|---|
| (Intercept) | 3802.1 |
| seasonSPRING | -1345.1 |
| seasonSUMMER | -6006.3 |
| seasonFALL | -681.4 |
| hum | -38.9 |
| windspeed | -64.1 |
| days_since_2011 | 4.8 |
| temp | 39.1 |
| temp$^2$ | 8.6 |
| seasonSPRING:temp | 407.4 |
| seasonSPRING:temp$^2$ | -18.7 |
| seasonSUMMER:temp | 801.1 |
| seasonSUMMER:temp$^2$ | -27.2 |
| seasonFALL:temp | 217.4 |
| seasonFALL:temp$^2$ | -11.3 |

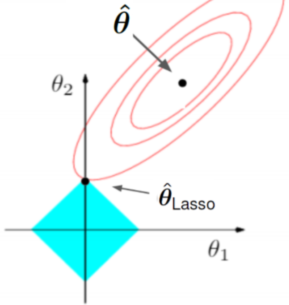
**Interpretation**: Not linear anymore!

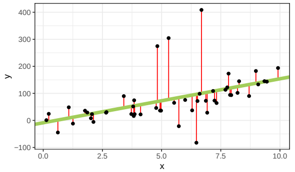
- `temp` depends on multiple weights due to `season`:
  - $\rightsquigarrow$ `WINTER`:
    $39.1 \cdot x_{temp} + 8.6 \cdot x^2_{temp}$
  - $\rightsquigarrow$ `SPRING`:
    $(39.1+407.4) \cdot x_{temp} + (8.6-18.7) \cdot x^2_{temp}$
  - $\rightsquigarrow$ `SUMMER`:
    $(39.1+801.1) \cdot x_{temp} + (8.6-27.2) \cdot x^2_{temp}$
  - $\rightsquigarrow$ `FALL`:
    $(39.1+217.4) \cdot x_{temp} + (8.6-11.3) \cdot x^2_{temp}$

# REGULARIZATION VIA LASSO ▸ Tibshirani (1996)

- LASSO adds an $L_1$-norm penalization term ($\lambda||\theta||_1$) to least squares optimization problem
  - ⇝ Shrinks some feature weights to zero (feature selection)
  - ⇝ Sparser models (fewer features): more interpretable
- Penalization parameter $\lambda$ must be chosen (e.g., by CV)

$$min_\theta \left( \underbrace{\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \mathbf{x}^{(i)\top}\theta)^2}_{\text{Least square estimate for LM}} + \lambda||\theta||_1 \right)$$



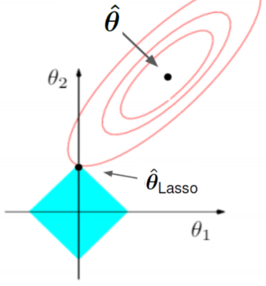



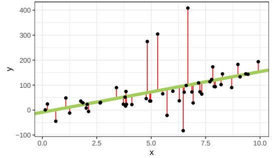



# REGULARIZATION VIA LASSO ▸ TIBSHIRANI

- LASSO adds an $L_1$-norm penalization term ($\lambda||\theta||_1$) to least squares optimization problem
  - ⇝ Shrinks some feature weights to zero (feature selection)
  - ⇝ Sparser models (fewer features): more interpretable
- Penalization parameter $\lambda$ must be chosen (e.g., by CV)

$$min_\theta \left( \underbrace{\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \xi^\top\theta)^2}_{\text{Least square estimate for LM}} + \lambda||\theta||_1 \right)$$

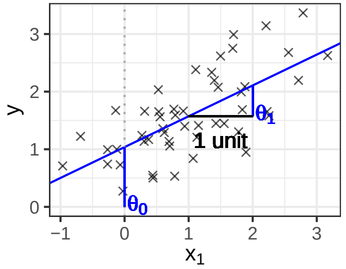


# REGULARIZATION VIA LASSO ▸ Tibshirani (1996)

**Example** (interpretation of weights analogous to LM):

- LASSO with main effects and interaction `temp` with `season`
- $\lambda$ is chosen $\rightsquigarrow$ 6 selected features ($\neq 0$)
- LASSO shrinks weights of single categories separately (due to dummy encoding)
  $\rightsquigarrow$ No feature selection of whole categorical features (only w.r.t. category levels)
  $\rightsquigarrow$ Solution: group LASSO ▸ Yuan and Lin (2006)



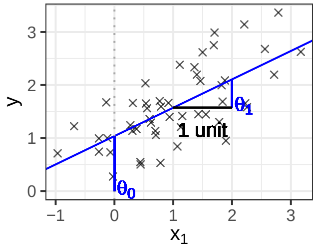

| | Weights |
|---|---|
| (Intercept) | 3135.2 |
| seasonSPRING | 767.4 |
| seasonSUMMER | 0.0 |
| seasonFALL | 0.0 |
| temp | 116.7 |
| hum | -28.9 |
| windspeed | -50.5 |
| days_since_2011 | 4.8 |
| seasonSPRING:temp | 0.0 |
| seasonSUMMER:temp | 0.0 |
| seasonFALL:temp | 30.2 |



# REGULARIZATION VIA LASSO ▸ TIBSHIRANI

**Example** (interpretation of weights analogous to LM):

- LASSO with main effects and interaction `temp` with `season`
- $\lambda$ is chosen $\rightsquigarrow$ 6 selected features ($\neq 0$)
- LASSO shrinks weights of single categories separately (due to dummy encoding)
  $\rightsquigarrow$ No feature selection of whole categorical features (only w.r.t. category levels)
  $\rightsquigarrow$ Solution: group LASSO ▸ Yuan and Lin 2006

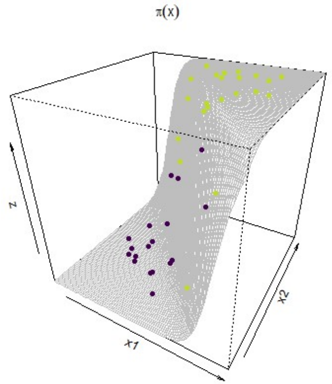




| | Weights |
|---|---|
| (Intercept) | 3135.2 |
| seasonSPRING | 767.4 |
| seasonSUMMER | 0.0 |
| seasonFALL | 0.0 |
| temp | 116.7 |
| hum | -28.9 |
| windspeed | -50.5 |
| days_since_2011 | 4.8 |
| seasonSPRING:temp | 0.0 |
| seasonSUMMER:temp | 0.0 |
| seasonFALL:temp | 30.2 |

# Interpretable Machine Learning

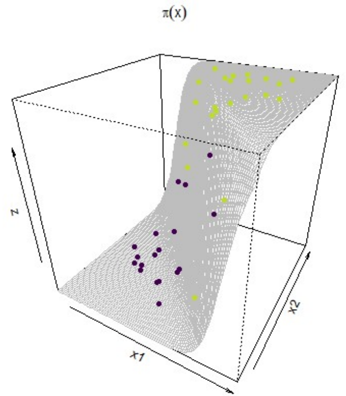
## Generalized Linear Models




**Learning goals**

- Definition of GLMs
- Logistic regression as example
- Interpretation in logistic regression

# Interpretable Machine Learning
## Generalized Linear Models (GLMs)



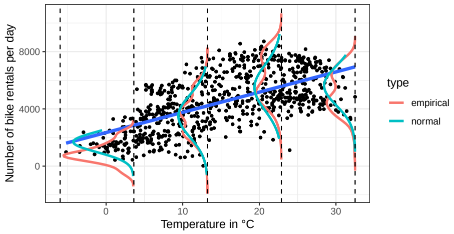

**Learning goals**

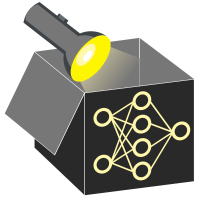
- Definition of GLMs
- Logistic regression as example
- Interpretation in logistic regression

# GENERALIZED LINEAR MODEL (GLM) ▸ Nelder and Wedderburn 1972

**Problem**: Target variable given feat. not always normally dist. ⇝ LM not suitable

- Target is binary (e.g., disease classification)
  ⇝ Bernoulli / Binomial distribution
- Target is count variable
  (e.g., number of sold products)
  ⇝ Poisson distribution
- Time until an event occurs
  (e.g., time until death)
  ⇝ Gamma distribution



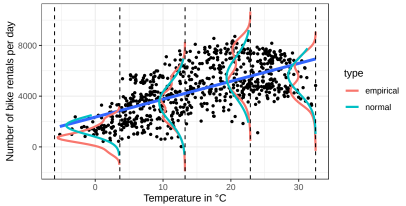



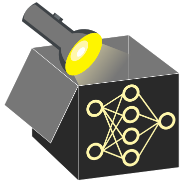


# GLM ▸ NELDER_WEDDERBURN

**Problem**: Target variable given feat not always normally distributed
⇝ LM not suitable

- Target is binary (e.g., disease classif.)
  ⇝ Bernoulli / Binomial distribution
- Target is count variable
  (e.g., number of sold products)
  ⇝ Poisson distribution
- Time until an event occurs
  (e.g., time until death)
  ⇝ Gamma distribution

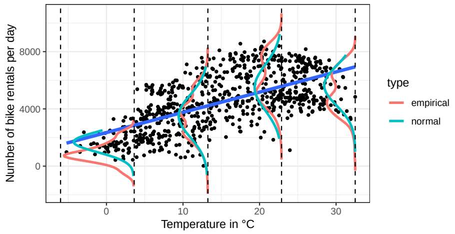


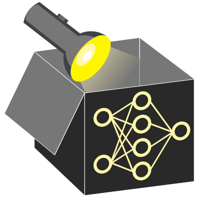



# GENERALIZED LINEAR MODEL (GLM) ▸ Nelder and Wedderburn 1972

**Problem**: Target variable given feat. not always normally dist. ⇝ LM not suitable

- Target is binary (e.g., disease classification)
  ⇝ Bernoulli / Binomial distribution
- Target is count variable
  (e.g., number of sold products)
  ⇝ Poisson distribution
- Time until an event occurs
  (e.g., time until death)
  ⇝ Gamma distribution



**Solution**: GLMs - extend LMs by allowing other distributions from exponential family

$$g(\mathbb{E}(y \mid \mathbf{x})) = \mathbf{x}^\top \boldsymbol{\theta} \;\Leftrightarrow\; \mathbb{E}(y \mid \mathbf{x}) = g^{-1}(\mathbf{x}^\top \boldsymbol{\theta})$$

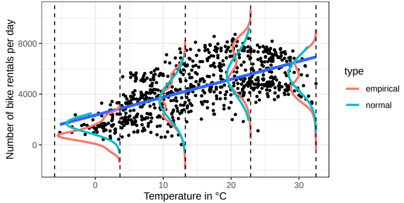
- Link function $g$ links linear predictor $\mathbf{x}^\top\boldsymbol{\theta}$ to expectation of distribution of $y \mid \mathbf{x}$
  ⇝ LM is special case: Gaussian distribution for $y \mid \mathbf{x}$ with $g$ as identity function
- Link function $g$ and distribution need to be specified
- High-order and interaction effects can be manually added as in LMs
- Note: Interpretation of weights depend on link function and distribution

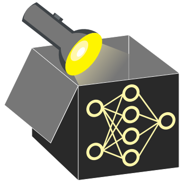


# GLM ▸ NELDER_WEDDERBURN

**Problem**: Target variable given feat not always normally distributed
⇝ LM not suitable

- Target is binary (e.g., disease classif.)
  ⇝ Bernoulli / Binomial distribution
- Target is count variable
  (e.g., number of sold products)
  ⇝ Poisson distribution
- Time until an event occurs
  (e.g., time until death)
  ⇝ Gamma distribution



**Solution**: GLMs - extend LMs by allowing other distrib.-s from exp. family

$$g(\mathbb{E}(y \mid \mathbf{x})) = \mathbf{x}^\top \boldsymbol{\theta} \;\Leftrightarrow\; \mathbb{E}(y \mid \mathbf{x}) = g^{-1}(\mathbf{x}^\top \boldsymbol{\theta})$$

- Link function $g$ links linear predictor $\mathbf{x}^\top\boldsymbol{\theta}$ to expectation of distrib. of $y \mid \mathbf{x}$
  ⇝ LM is special case: Gaussian distrib. for $y \mid \mathbf{x}$ with $g$ as identity func.
- Link function $g$ and distribution need to be specified
- High-order and interaction effects can be manually added as in LMs
- Note: Interpretation of weights depend on link function and distribution

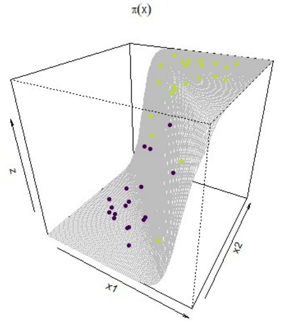
# GLM - LOGISTIC REGRESSION

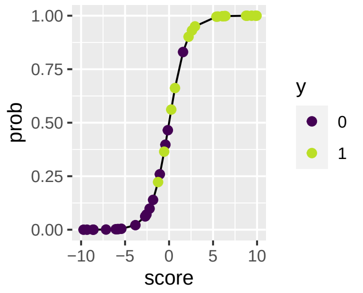
- Logistic regression $\hat{=}$ GLM with Bernoulli distribution and logit link function:

$$g(x) = \log\left(\frac{x}{1-x}\right) \Rightarrow g^{-1}(x) = \frac{1}{1+\exp(-x)}$$

- Models probabilities for binary classification by

$$\pi(\mathbf{x}) = \mathbb{E}(y \mid \mathbf{x}) = P(y=1) = g^{-1}(\mathbf{x}^\top \boldsymbol{\theta}) = \frac{1}{1+\exp(-\mathbf{x}^\top \boldsymbol{\theta})}$$





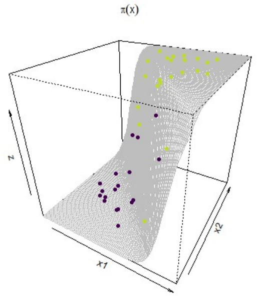

# GLM - LOGISTIC REGRESSION

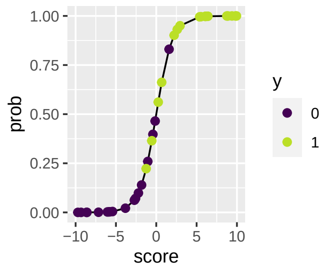
- Logistic regression $\hat{=}$ GLM with Bernoulli distribution and logit link function:

$$g(x) = \log\left(\frac{x}{1-x}\right) \Rightarrow g^{-1}(x) = \frac{1}{1+\exp(-x)}$$

- Models probabilities for binary classification by

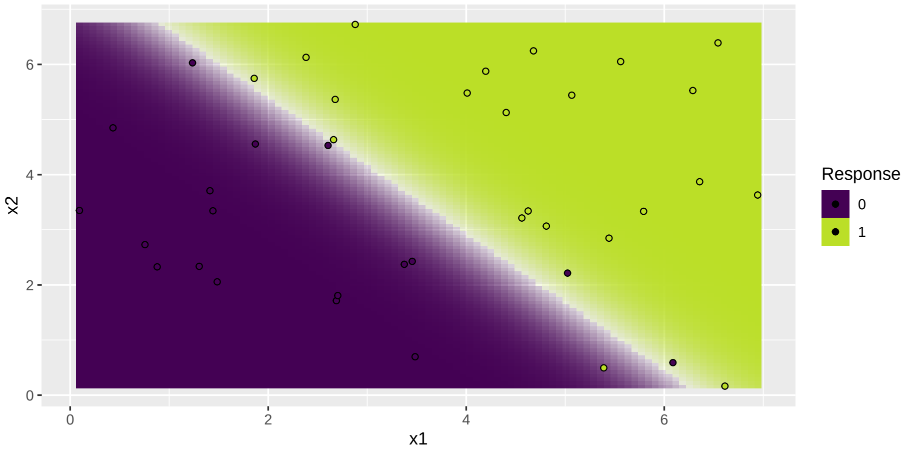
$$\pi(\mathbf{x}) = \mathbb{E}(y \mid \mathbf{x}) = P(y=1) = g^{-1}(\mathbf{x}^\top \boldsymbol{\theta}) = \frac{1}{1+\exp(-\mathbf{x}^\top \boldsymbol{\theta})}$$





# GLM - LOGISTIC REGRESSION

- Typically, we set the threshold to 0.5 to predict classes, e.g.,
  - Class 1 if $\pi(\mathbf{x}) > 0.5$
  - Class 0 if $\pi(\mathbf{x}) \leq 0.5$

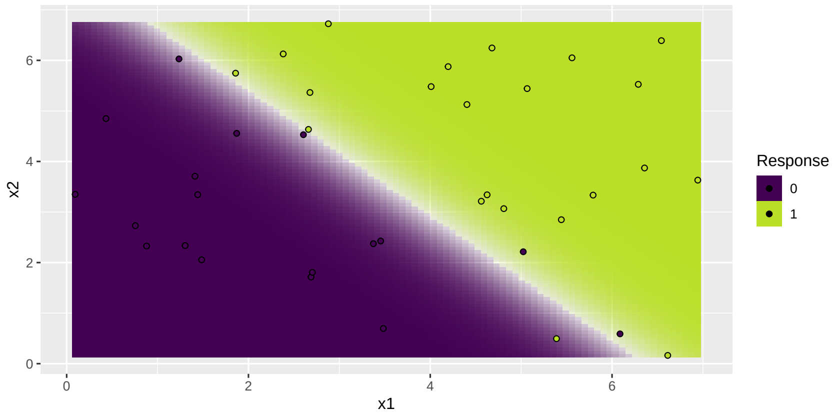






# GLM - LOGISTIC REGRESSION

- Typically, we set the threshold to 0.5 to predict classes, e.g.,
  - Class 1 if $\pi(\mathbf{x}) > 0.5$
  - Class 0 if $\pi(\mathbf{x}) \leq 0.5$

# GLM - LOGISTIC REGRESSION - INTERPRETATION

- **Recall:** Odds is ratio of two probabilities, odds ratio compares ratio of two odds
- Weights $\theta_j$ are interpreted linear as in LM (but w.r.t. log-odds)
  $\rightsquigarrow$ difficult to comprehend

$$\textit{log-odds} = \log\left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right) = \log\left(\frac{P(y=1)}{P(y=0)}\right) = \theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p$$

**Interpretation:**
Changing $x_j$ by one unit, changes log-odds of class 1 compared to class 0 by $\theta_j$



# GLM - LOG. REGRESSION - INTERPRETATION

- **Recall:** Odds is ratio of two probabilities, odds ratio is ratio of two odds
- Weights $\theta_j$ are interpreted linear as in LM (but w.r.t. log-odds)
  $\rightsquigarrow$ difficult to comprehend

$$\textit{log-odds} = \log\left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right) = \log\left(\frac{P(y=1)}{P(y=0)}\right) = \theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p$$

**Interpretation:** Changing $x_j$ by one unit, changes log-odds of class 1 compared to class 0 by $\theta_j$

# GLM - LOGISTIC REGRESSION - INTERPRETATION

- **Recall:** Odds is ratio of two probabilities, odds ratio compares ratio of two odds
- Weights $\theta_j$ are interpreted linear as in LM (but w.r.t. log-odds)
  $\rightsquigarrow$ difficult to comprehend

$$\textit{log-odds} = \log\left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right) = \log\left(\frac{P(y=1)}{P(y=0)}\right) = \theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p$$

**Interpretation:**
Changing $x_j$ by one unit, changes log-odds of class 1 compared to class 0 by $\theta_j$

- Odds for class 1 vs. class 0: $\textit{odds} = \dfrac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})} = \exp(\theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p)$
- Instead of interpreting changes w.r.t. log-odds, *odds ratio* is more common

$$= \frac{\textit{odds}_{x_j+1}}{\textit{odds}} = \frac{\exp(\theta_0 + \theta_1 x_1 + \ldots + \theta_j(x_j+1) + \ldots + \theta_p x_p)}{\exp(\theta_0 + \theta_1 x_1 + \ldots + \theta_j x_j + \ldots + \theta_p x_p)} = \exp(\theta_j)$$

**Interpretation**: Changing $x_j$ by one unit, changes the **odds ratio** for class 1 (compared to class 0) by the **factor** $\exp(\theta_j)$



# GLM - LOG. REGRESSION - INTERPRETATION

- **Recall:** Odds is ratio of two probabilities, odds ratio is ratio of two odds
- Weights $\theta_j$ are interpreted linear as in LM (but w.r.t. log-odds)
  $\rightsquigarrow$ difficult to comprehend

$$\textit{log-odds} = \log\left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right) = \log\left(\frac{P(y=1)}{P(y=0)}\right) = \theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p$$

**Interpretation:** Changing $x_j$ by one unit, changes log-odds of class 1 compared to class 0 by $\theta_j$

- Odds for cls 1 vs. cls 0: $\textit{odds} = \dfrac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})} = \exp(\theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p)$
- Instead of interpreting changes w.r.t. log-odds, it is more common to use *odds ratio*

$$= \frac{\textit{odds}_{x_j+1}}{\textit{odds}} = \frac{\exp(\theta_0 + \theta_1 x_1 + \ldots + \theta_j(x_j+1) + \ldots + \theta_p x_p)}{\exp(\theta_0 + \theta_1 x_1 + \ldots + \theta_j x_j + \ldots + \theta_p x_p)} = \exp(\theta_j)$$

**Interpretation**: Changing $x_j$ by one unit, changes the **odds ratio** for class 1 (compared to class 0) by the **factor** $\exp(\theta_j)$

# GLM - LOGISTIC REGRESSION - EXAMPLE

- Create a binary target variable for bike rental data:
  - Class 1: “high number of bike rentals” $>$ 70% quantile (i.e., `cnt` $>$ 5531)
  - Class 0: “low to medium number of bike rentals” (i.e., `cnt` $\leq$ 5531)
- Fit a logistic regression model (GLM with Bernoulli distribution and logit link)

| | Weights | SE | p-value |
|---|---|---|---|
| (Intercept) | -8.52 | 1.21 | 0.00 |
| seasonSPRING | 1.74 | 0.60 | 0.00 |
| seasonSUMMER | -0.86 | 0.77 | 0.26 |
| seasonFALL | -0.64 | 0.55 | 0.25 |
| temp | 0.29 | 0.04 | 0.00 |
| hum | -0.06 | 0.01 | 0.00 |
| windspeed | -0.09 | 0.03 | 0.00 |
| days_since_2011 | 0.02 | 0.00 | 0.00 |



# GLM - LOGISTIC REGRESSION - EXAMPLE

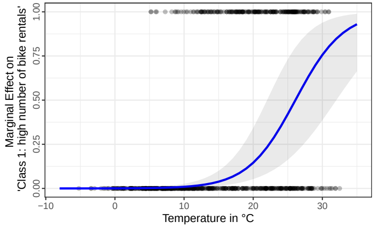
- Create a binary target variable for bike rental data:
  - Class 1: “high number of rentals” $>$ 70% quantile (i.e., `cnt` $>$ 5531)
  - Class 0: “low to medium number of rentals” (i.e., `cnt` $\leq$ 5531)
- Fit a logistic regression model (GLM with Bernoulli distri. and logit link)

| | Weights | SE | p-value |
|---|---|---|---|
| (Intercept) | -8.52 | 1.21 | 0.00 |
| seasonSPRING | 1.74 | 0.60 | 0.00 |
| seasonSUMMER | -0.86 | 0.77 | 0.26 |
| seasonFALL | -0.64 | 0.55 | 0.25 |
| temp | 0.29 | 0.04 | 0.00 |
| hum | -0.06 | 0.01 | 0.00 |
| windspeed | -0.09 | 0.03 | 0.00 |
| days_since_2011 | 0.02 | 0.00 | 0.00 |

# GLM - LOGISTIC REGRESSION - EXAMPLE

- Create a binary target variable for bike rental data:
  - Class 1: “high number of bike rentals” $>$ 70% quantile (i.e., $\texttt{cnt} > 5531$)
  - Class 0: “low to medium number of bike rentals” (i.e., $\texttt{cnt} \leq 5531$)
- Fit a logistic regression model (GLM with Bernoulli distribution and logit link)

| | Weights | SE | p-value |
|---|---|---|---|
| (Intercept) | -8.52 | 1.21 | 0.00 |
| seasonSPRING | 1.74 | 0.60 | 0.00 |
| seasonSUMMER | -0.86 | 0.77 | 0.26 |
| seasonFALL | -0.64 | 0.55 | 0.25 |
| temp | 0.29 | 0.04 | 0.00 |
| hum | -0.06 | 0.01 | 0.00 |
| windspeed | -0.09 | 0.03 | 0.00 |
| days_since_2011 | 0.02 | 0.00 | 0.00 |




### Interpretation

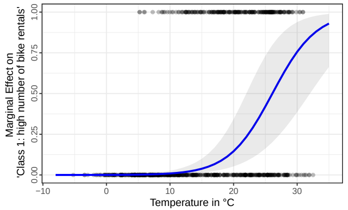
- If $\texttt{temp}$ increases by $1°C$, odds ratio for class 1 increases by factor $\exp(0.29) = 1.34$ compared to class 0, c.p. ($\hat{=}$ “high number of bike rentals” now 1.34 times more likely)



# GLM - LOGISTIC REGRESSION - EXAMPLE

- Create a binary target variable for bike rental data:
  - Class 1: “high number of rentals” $>$ 70% quantile (i.e., $\texttt{cnt} > 5531$)
  - Class 0: “low to medium number of rentals” (i.e., $\texttt{cnt} \leq 5531$)
- Fit a logistic regression model (GLM with Bernoulli distri. and logit link)

| | Weights | SE | p-value |
|---|---|---|---|
| (Intercept) | -8.52 | 1.21 | 0.00 |
| seasonSPRING | 1.74 | 0.60 | 0.00 |
| seasonSUMMER | -0.86 | 0.77 | 0.26 |
| seasonFALL | -0.64 | 0.55 | 0.25 |
| temp | 0.29 | 0.04 | 0.00 |
| hum | -0.06 | 0.01 | 0.00 |
| windspeed | -0.09 | 0.03 | 0.00 |
| days_since_2011 | 0.02 | 0.00 | 0.00 |

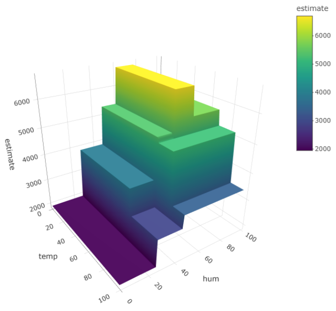



### Interpretation

- If $\texttt{temp}$ increases by $1°C$, odds ratio for class 1 increases by factor $\exp(0.29) = 1.34$ compared to class 0, c.p. ($\hat{=}$ “high number of bike rentals” now 1.34 times more likely)

# Interpretable Machine Learning

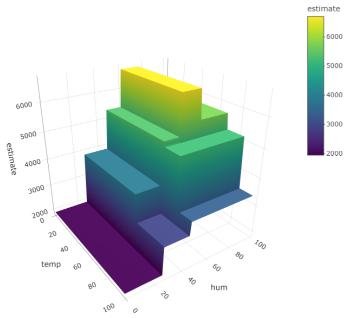
# Rule-based Models




**Learning goals**

- Decision trees
- RuleFit
- Decision rules


# Interpretable Machine Learning
# Rule-based Models



**Learning goals**

- Decision trees
- RuleFit
- Decision rules

# DECISION TREES ▸ Breiman et al. (1984)

**Idea**: Partition data into axis-aligned regions via greedy search for feature cut points (minimizing a split criterion), then predict a constant mean $c_m$ in each leaf region $\mathcal{R}_m$:

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{1}_{\{x \in \mathcal{R}_m\}}$$

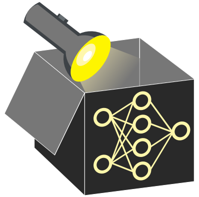


# DECISION TREES ▸ BREIMAN

**Idea**: Partition data into axis-aligned regions via greedy search for feature cut points (minimizing a split criterion), then predict a constant mean $c_m$ in each leaf region $\mathcal{R}_m$:

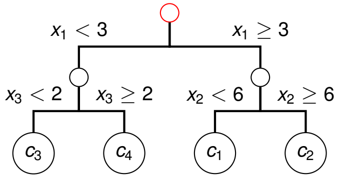
$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{1}_{\{x \in \mathcal{R}_m\}}$$

# DECISION TREES ▸ Breiman et al. (1984)

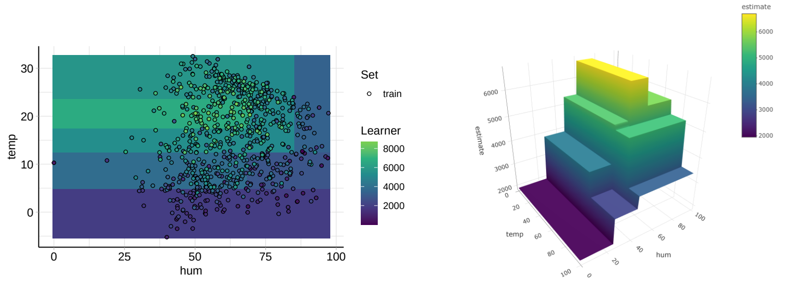


**Idea**: Partition data into axis-aligned regions via greedy search for feature cut points (minimizing a split criterion), then predict a constant mean $c_m$ in each leaf region $\mathcal{R}_m$:

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{1}_{\{x \in \mathcal{R}_m\}}$$

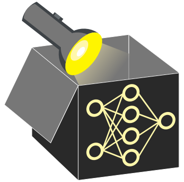



- Applicable to regression and classification
- Models interactions and non-linear effects
- Handles mixed feature spaces & missing values



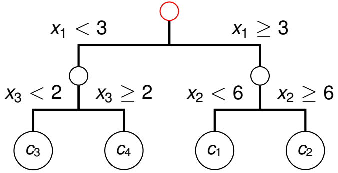



# DECISION TREES ▸ BREIMAN

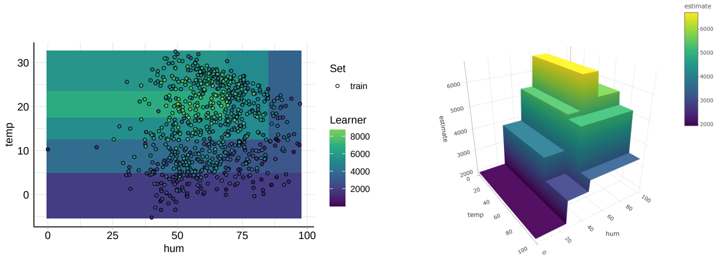


**Idea**: Partition data into axis-aligned regions via greedy search for feature cut points (minimizing a split criterion), then predict a constant mean $c_m$ in each leaf region $\mathcal{R}_m$:

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{1}_{\{x \in \mathcal{R}_m\}}$$

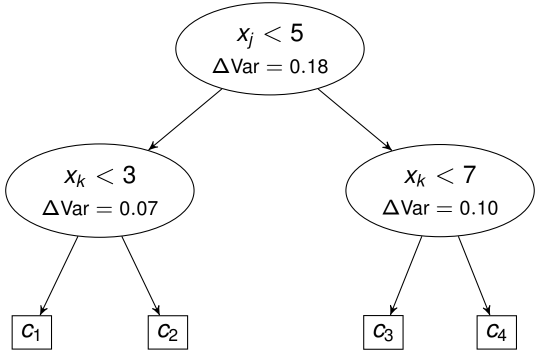



- Applicable to regression and classification
- Models interactions and non-linear effects
- Handles mixed feat, spaces & missing values



# INTERPRETATION OF TREE-BASED MODELS

- Interpretation via path of decision rules along tree branches
- **Feature importance** (quantifies how often and how usefully $x_j$ is used):
  - For each split on feature $x_j$, record the decrease in the split criterion
  - Aggregate this over the tree: sum or average over all splits involving $x_j$
  - Split criterion: variance (regression), Gini index / entropy (classification)



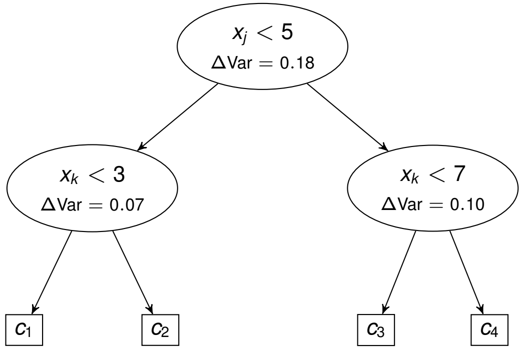

- Each $\Delta$Var is assigned to the splitting feature
- Feature importance = sum of all $\Delta$Var for that feature:

  $x_j$: $0.18$
  $x_k$: $0.07 + 0.10 = 0.17$



# INTERPRETATION OF TREE-BASED MODELS

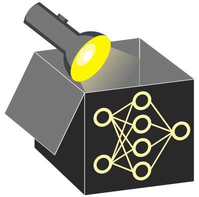
- Interpretation via path of decision rules along tree branches
- **Feature importance** (quantifies how often and how usefully $x_j$ is used):
  - For each split on feature $x_j$, record the decrease in the split criterion
  - Aggregate this over the tree: sum or avg. over all splits involving $x_j$
  - Split criterion: variance (regression), Gini index / entropy (classif.)




- Each $\Delta$Var is assigned to the splitting feature
- Feature importance = sum of all $\Delta$Var for that feat.:

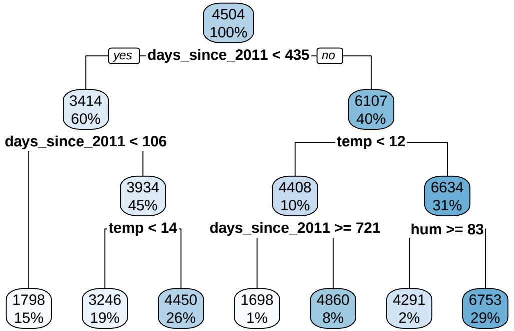
  $x_j$: $0.18$
  $x_k$: $0.07 + 0.10 = 0.17$

# DECISION TREES - EXAMPLE



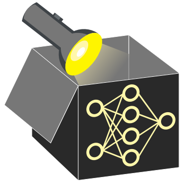
- Fit decision tree with tree depth of 3 on bike data
- E.g., mean prediction for the first 105 days since 2011 is 1798
  $\rightsquigarrow$ Applies to $\hat{=}$15% of the data (leftmost branch)
- `days_since_2011`: highest feature importance (explains most of variance)

| Feature | Importance |
|---|---|
| days_since_2011 | 79.53 |
| temp | 17.55 |
| hum | 2.92 |



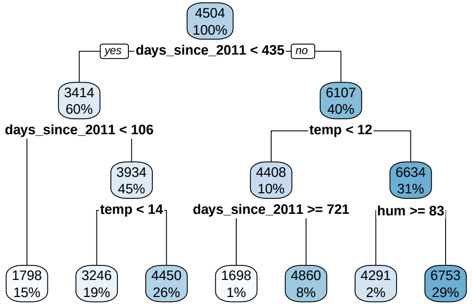



# DECISION TREES - EXAMPLE



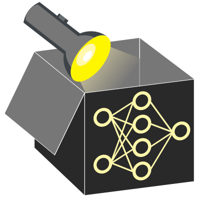
- Fit decision tree with tree depth of 3 on bike data
- E.g., mean prediction for the first 105 days since 2011 is 1798
  $\rightsquigarrow$ Applies to $\hat{=}$15% of the data (leftmost branch)
- `days_since_2011`: highest feat. importance (explains most of variance)

| Feature | Importance |
|---|---|
| days_since_2011 | 79.53 |
| temp | 17.55 |
| hum | 2.92 |



# UNBIASED RECURSIVE PARTITIONING

▸ Hothorn et al. (2006) ▸ Zeileis et al. (2008) ▸ Strobl et al. (2007)



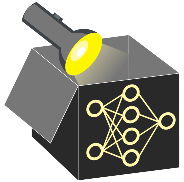
**Problems** with CART (Classification and Regression Trees):

1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance $\rightsquigarrow$ prone to overfitting



# UNBIASED RECURSIVE PARTITIONING

▸ Hothorn 2006 ▸ Zeileis 2008 ▸ Strobl 2007

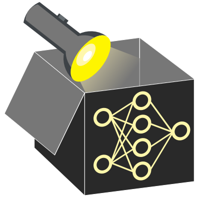


**Problems** with CART (Classification and Regression Trees):

1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance $\rightsquigarrow$ prone to overfitting

# UNBIASED RECURSIVE PARTITIONING

▸ Hothorn et al. (2006) ▸ Zeileis et al. (2008) ▸ Strobl et al. (2007)



**Problems** with CART (Classification and Regression Trees):

1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance $\rightsquigarrow$ prone to overfitting

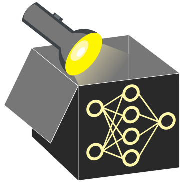
**Unbiased recursive partitioning** via conditional inference trees (`ctree`) or model-based recursive partitioning (`mob`):

1. Separate selection of **feature used for splitting** and **split point**
2. Hypothesis test as stopping criteria



# UNBIASED RECURSIVE PARTITIONING

▸ Hothorn 2006 ▸ Zeileis 2008 ▸ Strobl 2007



**Problems** with CART (Classification and Regression Trees):

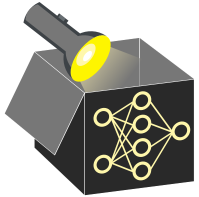
1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance
   $\rightsquigarrow$ prone to overfitting

**Unbiased recursive partitioning** via conditional inference trees (`ctree`) or model-based recursive partitioning (`mob`):

1. Separate selection of **feature used for splitting** and **split point**
2. Hypothesis test as stopping criteria

# UNBIASED RECURSIVE PARTITIONING

▸ Hothorn et al. (2006) ▸ Zeileis et al. (2008) ▸ Strobl et al. (2007)



**Problems** with CART (Classification and Regression Trees):

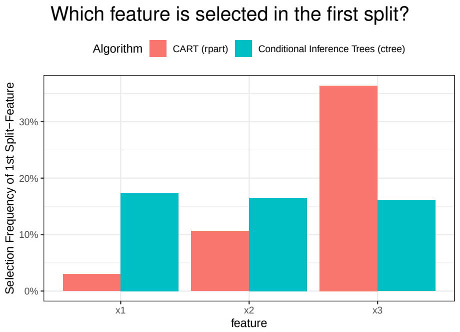
1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance ⇝ prone to overfitting

**Unbiased recursive partitioning** via conditional inference trees (`ctree`) or model-based recursive partitioning (`mob`):

1. Separate selection of **feature used for splitting** and **split point**
2. Hypothesis test as stopping criteria

**Example (selection bias)**:
Simulate data ($n = 200$) with $Y \sim N(0, 1)$ and 3 features of different cardinality independent from $Y$ (repeat 500 times):

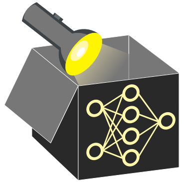
- $X_1 \sim Binom(n, \frac{1}{2})$
- $X_2 \sim M(n, (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}))$
- $X_3 \sim M(n, (\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}))$






# UNBIASED RECURSIVE PARTITIONING

▸ Hothorn 2006 ▸ Zeileis 2008 ▸ Strobl 2007



**Problems** with CART (Classification and Regression Trees):

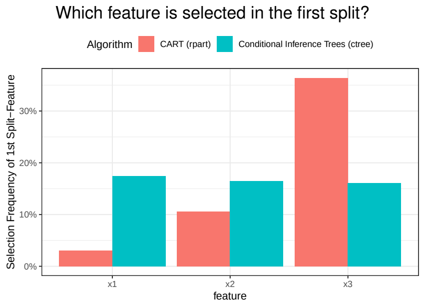
1. Selection bias towards high-cardinal/continuous features
2. Splits on any improvement, regardless of significance ⇝ prone to overfitting

**Unbiased recursive partitioning** via conditional inference trees (`ctree`) or model-based recursive partitioning (`mob`):

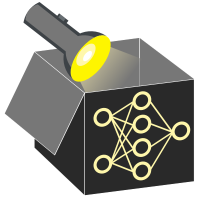
1. Separate selection of **feature used for splitting** and **split point**
2. Hypothesis test as stopping criteria

**Example (selection bias)**:
Simulate data ($n = 200$), $Y \sim N(0, 1)$ and 3 features of different cardinality indep. from $Y$ (repeat 500 times):

- $X_1 \sim Binom(n, \frac{1}{2})$
- $X_2 \sim M(n, (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}))$
- $X_3 \sim M(n, (\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}))$



# UNBIASED RECURSIVE PARTITIONING

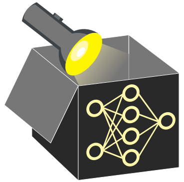


Differences to CART:

- Two-step approach (1. find most significant split feature, 2. find best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leave nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point



# UNBIASED RECURSIVE PARTITIONING



Differences to CART:

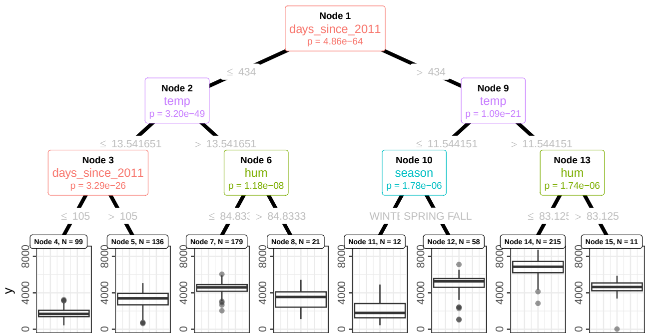
- Two-step approach (finds 1. most significant split feat., 2. best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leaf nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

# UNBIASED RECURSIVE PARTITIONING

Differences to CART:

- Two-step approach (1. find most significant split feature, 2. find best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leave nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

**Example** (`ctree`): Bike data (constant model in final nodes)




Train error (MSE):
758,844.0 (`ctree`)
742,244.4 (`mob`)

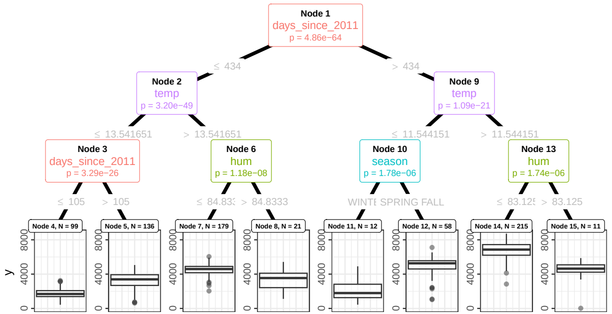


# UNBIASED RECURSIVE PARTITIONING

Differences to CART:

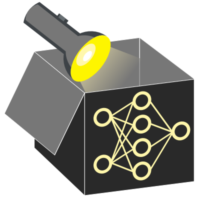
- Two-step approach (finds 1. most significant split feat., 2. best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leaf nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

**Example** (`ctree`): Bike data (constant model in final nodes)



Train MSE:
758,844 (`ctree`)
742,244 (`mob`)

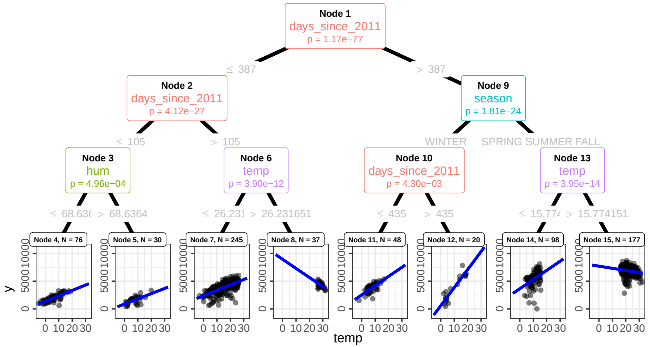
# UNBIASED RECURSIVE PARTITIONING



Differences to CART:

- Two-step approach (1. find most significant split feature, 2. find best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leave nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

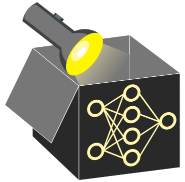
**Example** (`mob`): Bike data (linear model with `temp` in final nodes)




Train error (MSE):
758,844.0 (`ctree`)
742,244.4 (`mob`)



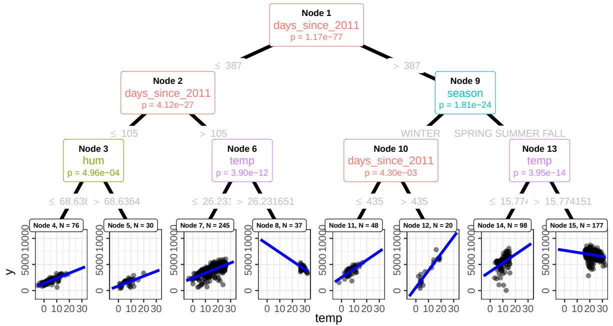
# UNBIASED RECURSIVE PARTITIONING



Differences to CART:

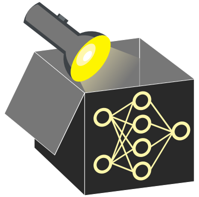
- Two-step approach (finds 1. most significant split feat., 2. best split point)
- Parametric model (e.g. LM instead of constant) can be fitted in leaf nodes
- Significance of split (p-value) given in each node
- `ctree` and `mob` differ in hypothesis test used for selecting the split feature (independence test vs. fluctuation test) and how to find the best split point

**Example** (`mob`): Bike data (linear model with `temp` in final nodes)




Train MSE:
758,844 (`ctree`)
742,244 (`mob`)

# OTHER RULE-BASED MODELS

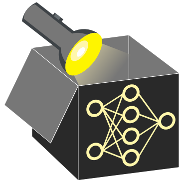


## Decision Rules ▸ Holte 1993

- Flat list of simple "if – then" statements
  ⇝ very intuitive and easy-to-interpret
- Mainly devised for classification
  (support for regression is limited)
- Numeric features are typically discretised

| | | |
|---|---|---|
| IF $x_1 \leq 2.3$ AND $x_4 = $ "A" | THEN | y = 1 |
| ELSE IF $x_2 > 5.0$ | THEN | y = 2 |
| ELSE | | y = 3 |





# OTHER RULE-BASED MODELS

## Decision Rules ▸ Holte 1993

- Flat list of simple "if – then" statements
  ⇝ very intuitive and easy-to-interpret
- Mainly devised for classification
  (support for regression is limited)
- Numeric features are typically discretised

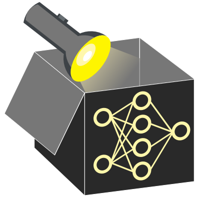
| | | |
|---|---|---|
| IF $x_1 \leq 2.3$ AND $x_4 = $ "A" | THEN | y = 1 |
| ELSE IF $x_2 > 5.0$ | THEN | y = 2 |
| ELSE | | y = 3 |

# OTHER RULE-BASED MODELS

## Decision Rules ▸ Holte 1993

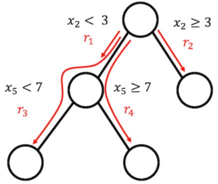
- Flat list of simple "if – then" statements
  $\rightsquigarrow$ very intuitive and easy-to-interpret
- Mainly devised for classification
  (support for regression is limited)
- Numeric features are typically discretised

IF $x_1 \leq 2.3$ AND $x_4 =$ "A" THEN y = 1
ELSE IF $x_2 > 5.0$ THEN y = 2
ELSE y = 3



## RuleFit ▸ Friedman & Popescu 2008

- Extract binary rules $r_m(\mathbf{x}) \in \{0, 1\}$ from many shallow trees (one per root-to-leaf path)
- Fit an $L_1$-regularized LM
  $\hat{f}(\mathbf{x}) = \beta_0 + \sum_m \beta_m r_m(\mathbf{x}) + \sum_j \gamma_j x_j$
- Regularization retains only a few rules
  $\Rightarrow$ sparse, non-linear, interaction-aware
- Coefficients relate to rule/feature importance




▸ Molnar 2022

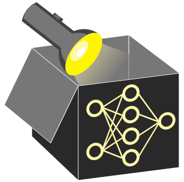



# OTHER RULE-BASED MODELS

## Decision Rules ▸ Holte 1993

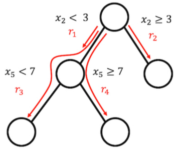
- Flat list of simple "if – then" statements
  $\rightsquigarrow$ very intuitive and easy-to-interpret
- Mainly devised for classification
  (support for regression is limited)
- Numeric features are typically discretised

IF $x_1 \leq 2.3$ AND $x_4 =$ "A" THEN y = 1
ELSE IF $x_2 > 5.0$ THEN y = 2
ELSE y = 3


## RuleFit ▸ Friedman and Popescu 2008

- Extract binary rules $r_m(\mathbf{x}) \in \{0, 1\}$ from many shallow trees (one per root-to-leaf path)
- Fit an $L_1$-regularized LM
  $\hat{f}(\mathbf{x}) = \beta_0 + \sum_m \beta_m r_m(\mathbf{x}) + \sum_j \gamma_j x_j$
- Regularization retains only a few rules
  $\Rightarrow$ sparse, non-linear, interaction-aware
- Coefficients relate to rule/feature importance


▸ Molnar 2022