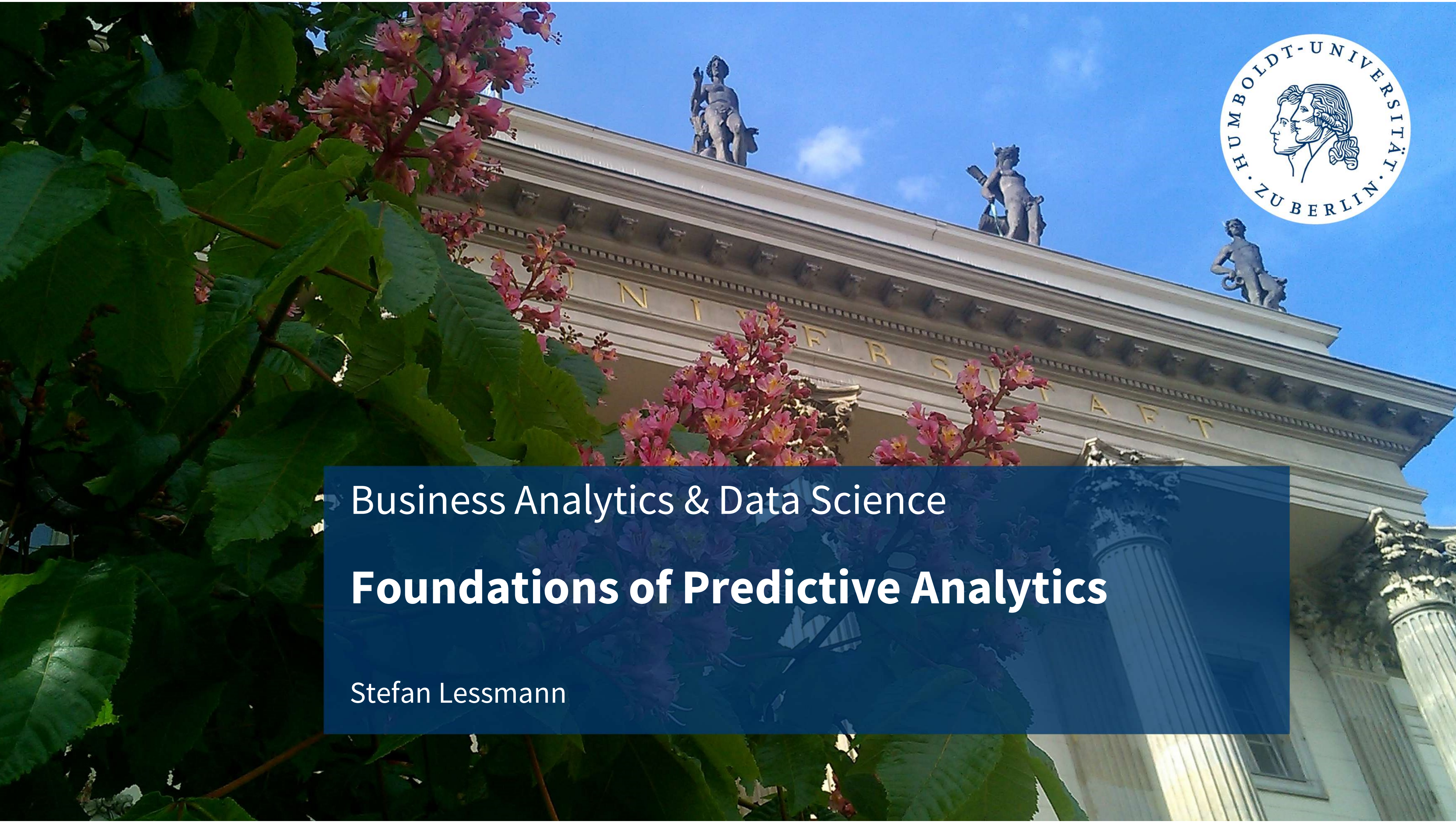


Business Analytics & Data Science

Foundations of Predictive Analytics

Stefan Lessmann

Recap: Data Science / Machine Learning Algorithms

Modeling approaches

Predictive

Regression

- Linear Regression
- Neural network
- CART
- …

Classification

- Logistic regression
- Nearest neighbor
- Boosting
- …

Descriptive

Segmentation

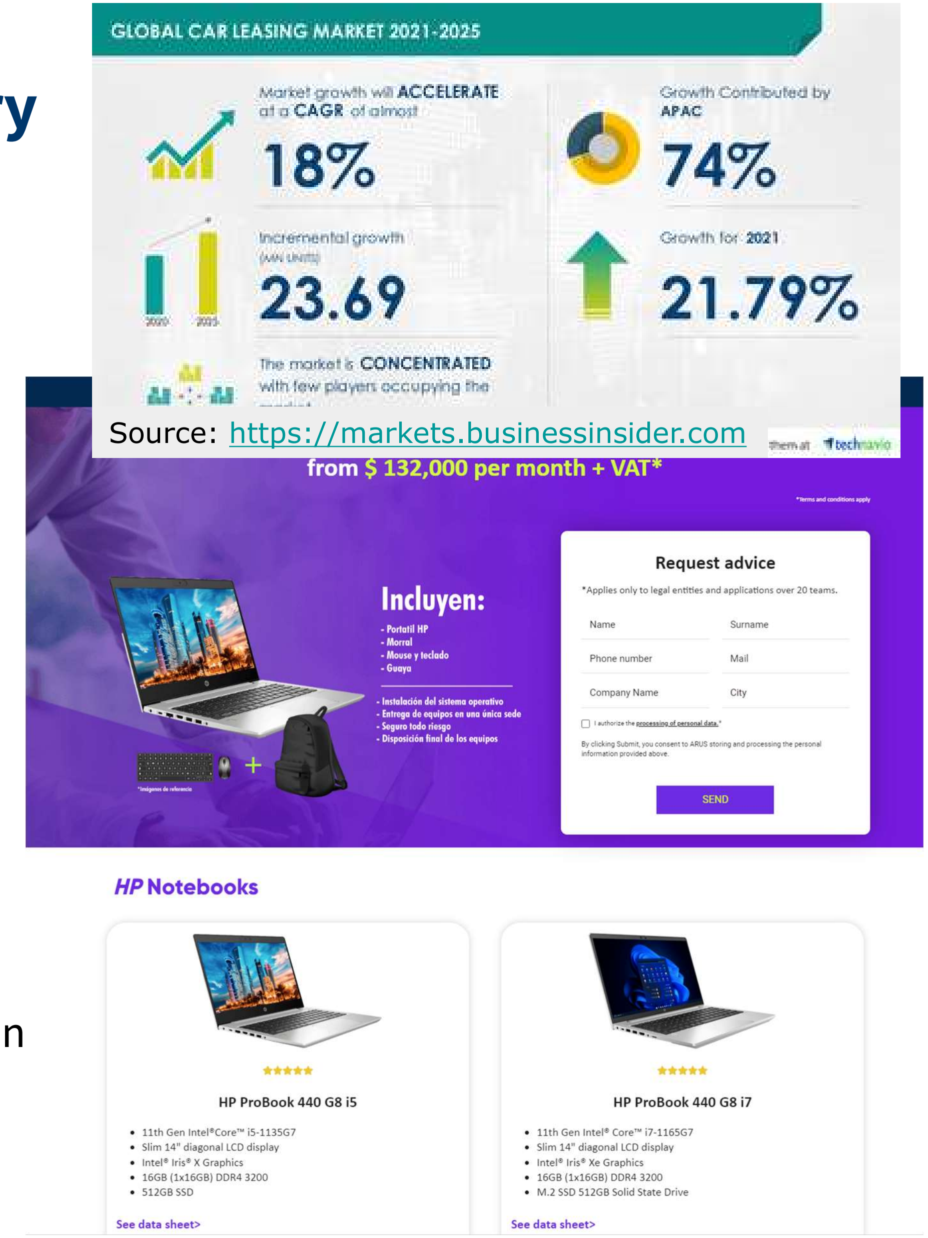
- Hierarchical
- K-means
- DBSCAN
- …

Rule/sequence mining

- Apriori-algorithm
- FP-growth
- PrefixSpan
- …

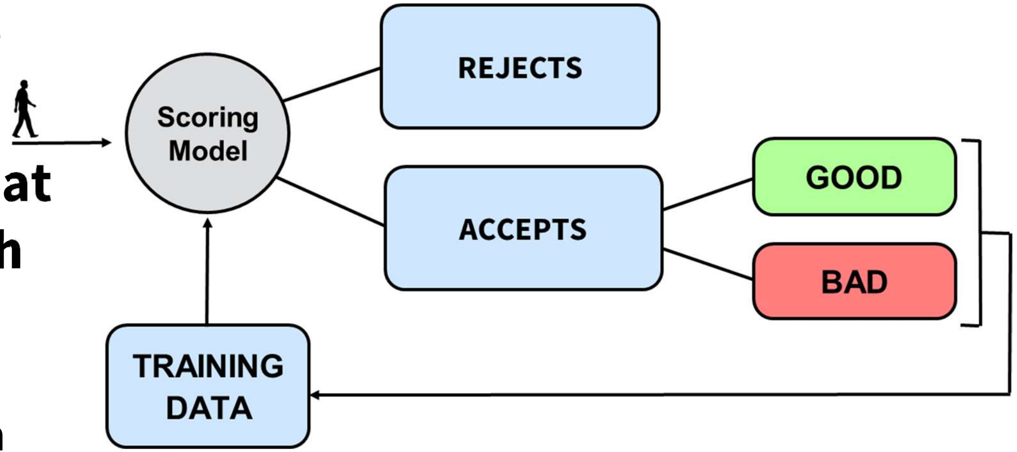
Recap: Business Use Case: Leasing Industry

- **Important channel to market durables**
  - Most prominently cars
  - Machinery, IT equipment, etc.
- **Typical setup**
  - Clients lease equipment for a given period
  - Provider receives monthly fee
  - Client returns the item when contract expires
  - Provides resales the used item in the second-hand market
- **Prediction-based decision support**
  - Predict resale price in the second-hand market
  - Set leasing rate so as to cover all costs including depreciation



Business Use Case: Credit Scoring

- **Huge variety of financial products from traditional loans to on-demand rate payment**
- **Lender sets interest rate and earns a fixed, contract-agreed fee…**
- **… but faces the risk of default, meaning that the borrower does not pay back their depth**
- **Prediction-based decision support**
  - Estimate probability to default (PD) using data from the application form
  - Decide on the application based on estimated default risk



Agenda

Introduction
Principles of prediction, regression & classification

The Linear Regression Model
Parametric model, loss function and residuals, model estimation

Summary

Introduction

Principles of prediction, regression & classification

Predictive Analytics Using Supervised Machine Learning

Estimate functional relationship between features and a target

- **Data includes values for attributes a value for the target variable**
- **Two flavors depending on the type of the target variable**
  - Discrete target variable → classification (e.g., credit risk modeling)
  - Numerical target variable → regression (e.g., resale price forecasting)

BUREAU SCORE	…	**DEFAULT** **(e.g., 90 days late)**
650	…	**No**
280	…	**Yes**
750	…	**No**
600	…	**No**
575	…	**No**
715	…	**No**
580	…	**Yes**
410		**No**

PRODUCT	…	**RESALE PRICE [$]**
Dell XPS 15’	…	347
Dell XPS 15’	…	416
Dell XPS 17’	…	538
HP Envy 17’	…	121
HP EliteBook 850	…	172
Lenovo Yoga 11’	…	88
Lenovo Yoga 13’	…	266
…	…	…

$$f(X)$$

$$X \xrightarrow{\text{map}} Y$$

input output

Formalization of Supervised Machine Learning

Resale price forecasting example

- **We aim at forecasting resale prices (our target variable) denoted by Y**
- **We assume that resale prices Y depend on features X**
  - We do not know how exactly resale prices depend on feature values
  - But we have access to historical data $\mathcal{D} = \{Y_i, \boldsymbol{X}_i\}_{i=1}^{n}$ that exemplifies the relationship
- **At decision time, i.e., when we need a forecast, we can observe X but not Y**
- **We use algorithms to learn a model f that maps from features to target $f(\boldsymbol{X}) \longrightarrow Y$**

PRODUCT	LIST PRICE [$]	AGE [month]	CLIENT INDUSTRY	…
Dell XPS 15’	2,500	36	Mining	…
Dell XPS 15’	2,500	24	Health	…
Dell XPS 17’	3,000	36	Manufacturing	…
Lenovo Yoga 11’	799	12	Office	…
Lenovo Yoga 13’	1,100	12	Office	…
…	…	…	…	…

$$\boldsymbol{X} = (X_1, X_2, \dots, X_m) \in \mathbb{R}^m \xrightarrow{f(X)} Y \in \mathbb{R}$$

OBSERVED RESALE PRICE [$]
347
416
538
266
…

Two-Stage Paradigm of Supervised Machine Learning

Together with some more terminology

Learning Algorithm

Stage 1: Model Training

Data-driven development of a predictive model using **labelled** data $\mathcal{D} = \{Y_i, \boldsymbol{X}_i\}_{i=1}^{n}$

Training data incl. Y

i	Y	X_1	X_2	…	X_m
1	…	…	…	…	…
2	…	…	…	…	…
…	…	…	…	…	…
n	…	…	…	…	…

Model

Stage 2: Model Application

Apply trained model to novel data w/o known output to obtain a forecast

New data w/o Y

i	X_1	X_2	…	X_m
$n+1$	…	…	…	…
$n+2$	…	…	…	…
…	…	…	…	…
N	…	…	…	…

Forecasts of Y

i	$\hat{Y}$
$n+1$	…
$n+2$	…
…	…
N	…

Implications of the Supervised ML Paradigm
Supervised ML requires labeled data

- **Defining the target variable**
  - Credit scoring example: 90 days past due
  - Resale price forecasting example: actual resale price
- **Observing the value of the target variable**
  - The value of the target is what we aim to predict
  - Two possible scenarios
    - The value cannot be known when it is needed
    - It is not practical to measure the value when it is needed
  - Have access to the feature values and leverage these to predict the target (i.e., stage 2)
  - Training set (i.e., stage 1) comprises **past** data
- **Is past data representative for the future?**

Algorithms for Supervised Learning (Selection)

A subjective view and a bit of guidance

Supervised learning algorithms

Tree- and prototype-based algorithms (non-parametric)

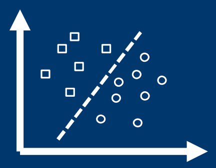
- CART
- CHAID
- C4.5
- Bagging / Random Forest
- Gradient Boosting / XGB
- Nearest neighbors
- …

Individual trees: CART, CHAID, C4.5

Ensembles (many trees): Bagging / Random Forest, Gradient Boosting / XGB

Regression-type algorithms (semi-/parametric)

- Linear regression
- Generalized linear models (GLM)
- Generalized additive models (GAM)
- Artificial neural networks (ANN)
- Support vector machines (SVM
- …



The Linear Regression Model

Parametric model, loss function and residuals, model estimation

Linear Regression Model

Postulates a linear, additive feature-to-target relationship

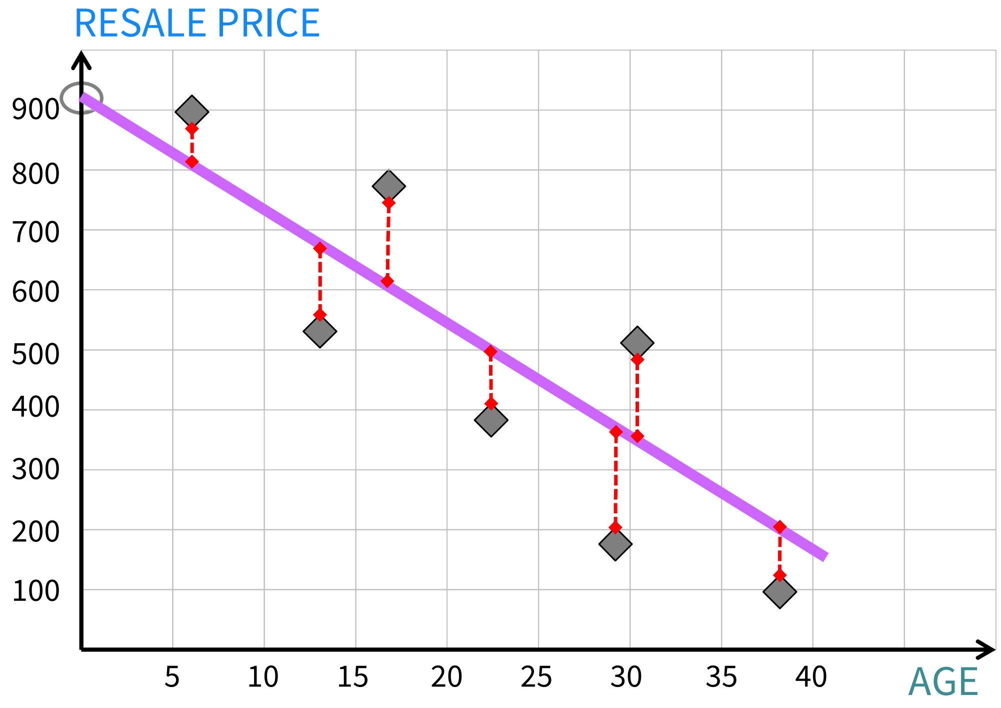
- **Famous regression equation in a resale price forecasting context**

$$\text{RESALE PRICE} = \textit{bias} + w_1 \text{LIST PRICE} + w_2 \text{AGE} + \cdots + w_m \text{INDUSTRY} + \textit{residual}$$

- **Simplification for plotting:**

$$\text{RESALE PRICE} = \textit{bias} + w_1 \text{AGE} + \epsilon$$

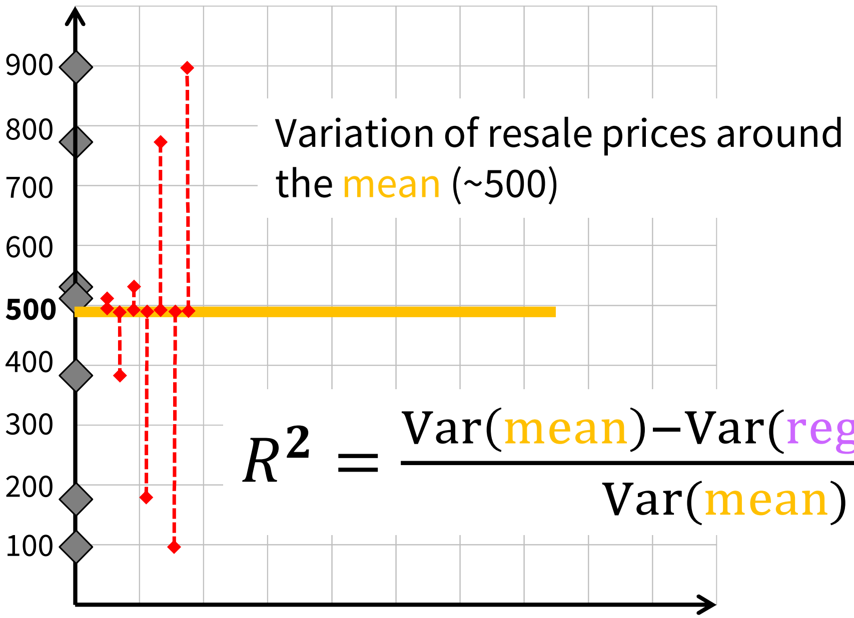
…	AGE [MONTH]	…	RESALE PRICE [$]
…	6	…	900
…	13	…	515
…	17	…	890
…	23	…	395
…	29	…	180
…	31	…	501
…	38	…	100
…	…	…	…



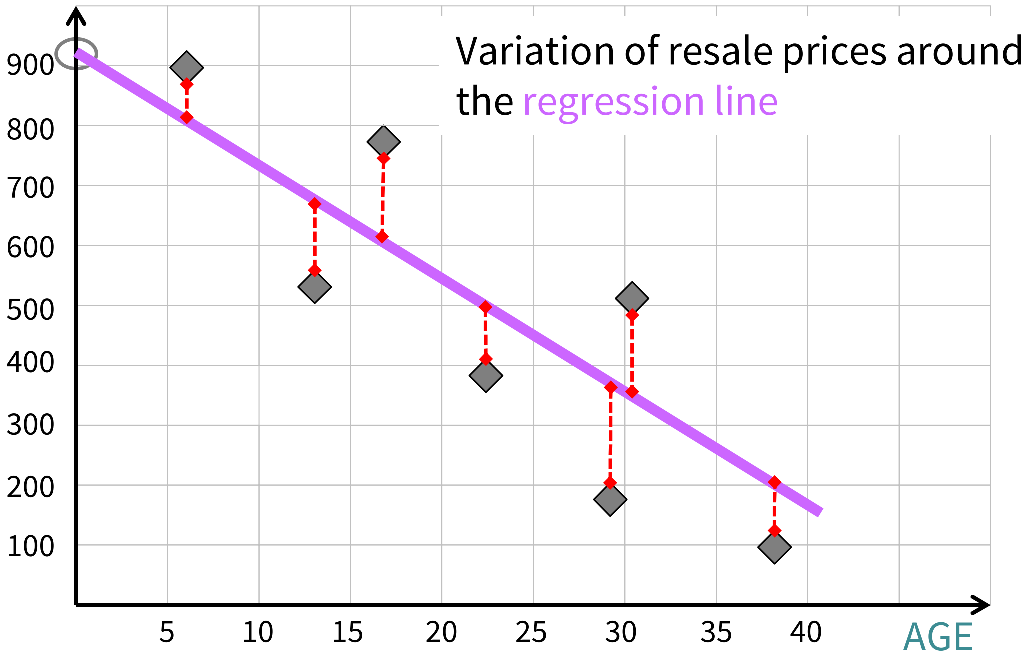
Linear Regression Model

Postulates a linear, additive feature-to-target relationship

- **Regression model explains variation in resale prices (target) by differences in the Age of resold items (feature values)**
- **The famous R^2 statistic captures how much of the variation in resale prices the regression explains**



$$R^2 = \frac{\mathrm{Var}(\text{mean}) - \mathrm{Var}(\text{regression})}{\mathrm{Var}(\text{mean})}$$



Regression Model Estimation
Determine the free parameters of the regression function

- **Linear regression belongs to the family of parametric models**
  - We assume we know the true dependency of the target variable and the features
  - We specify a function expressing the assumed relationship (e.g., linear and additive)
  - We incorporate free parameters that govern the shape of the function
- **Formally**

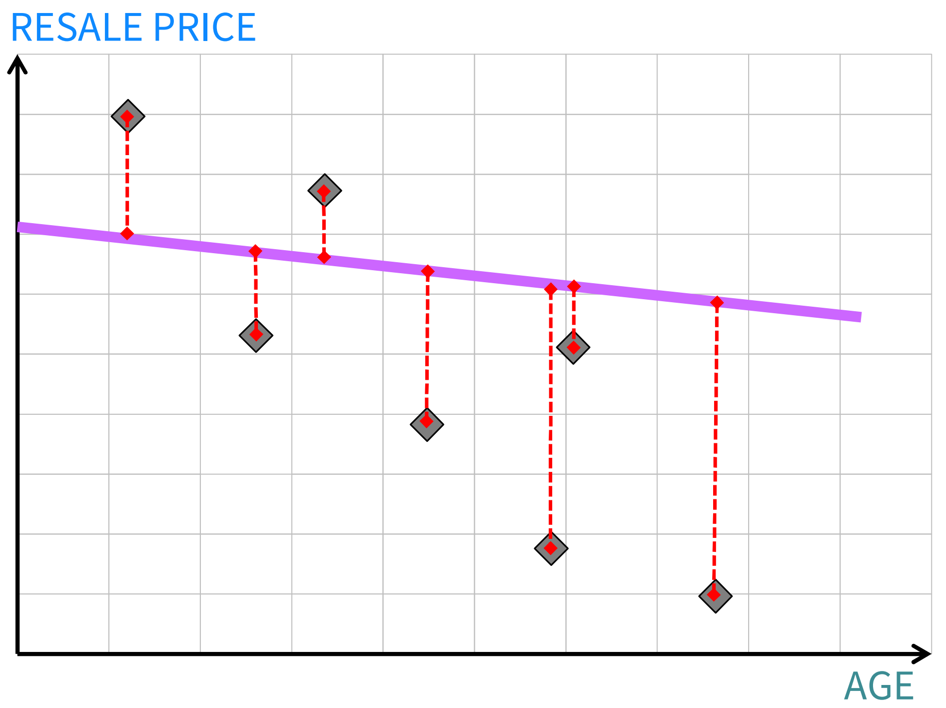
$$\text{RESALE PRICE} = \mathit{bias} + w_1 \text{LIST PRICE} + w_2 \text{AGE} + \cdots + w_m \text{INDUSTRY} + \mathit{residual}$$

$$Y = b + w_1 X_1 + w_2 X_2 + \cdots + w_m X_m + \epsilon$$

- **Model estimation (aka training, fitting, development)**
  - Find *suitable* values for the free parameters (here denoted by w)
  - Introduce a measure that captures what is meant by *suitable*
  - Set parameters such this measure signals an optimal fit of the model to the data

Regression Model Estimation Intuition
Reach steady-state with minimal energy

- **The data points are fixed at their location**
- **The position of the regression line is flexible and depends on two parameters**
  - The intercept and the slope
  - Denoted by $bias$ and w_1 above
- **A spring connects each data point with the regression line**
- **Springs exerts a force on the regression line, pulling it toward 'their' data point**
- **The more we bend a spring the higher the force it exerts**

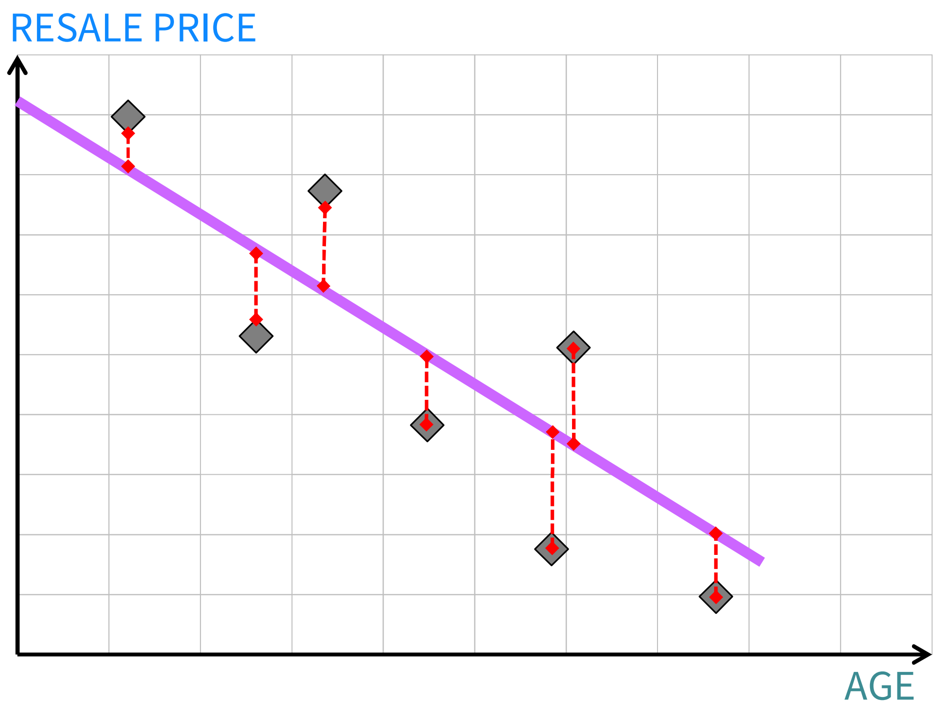


Not an equilibrium solution!

Regression Model Estimation Intuition

Reach steady-state with minimal energy

- **The data points are fixed at their location**
- **The position of the regression line is flexible and depends on two parameters**
  - The intercept and the slope
  - Denoted by $bias$ and w_1 above
- **A spring connects each data point with the regression line**
- **Springs exerts a force on the regression line, pulling it toward 'their' data point**
- **The more we bend a spring the higher the force it exerts**



Equilibrium solution!

Regression Model Estimation

Maximizing model fit through minimizing a loss function

- **A loss function J measures the degree to which the model output $\hat{Y}$ agrees with the true value of the target Y**
- **Squared-error loss**

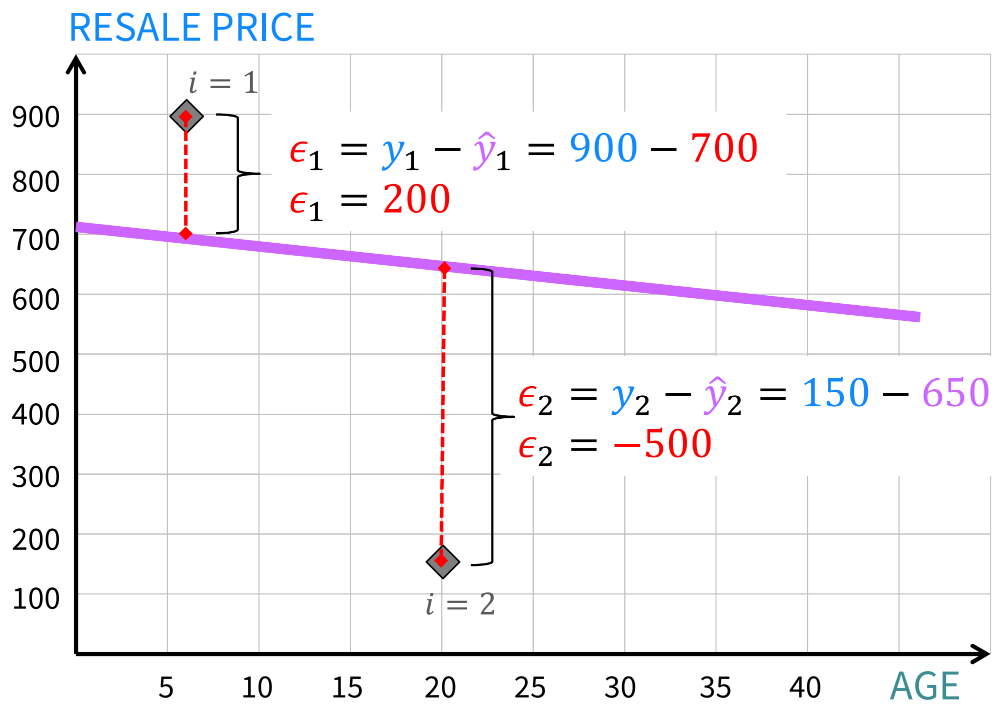
$$J = \sum_{i=1}^{n} (\epsilon_i)^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- **Whereby the model output depends on the free parameters w**

$$\hat{y}_i = b + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_m x_{im}$$

$$\hat{y}_i = b + \sum_{j=1}^{m} w_j x_{ij}$$

1



So here, the value of the loss function J is:

$$\sum_{i=1}^{n} (\epsilon_i)^2 = (\epsilon_1)^2 + (\epsilon_2)^2 = 200^2 + (-500)^2 = 290{,}000$$

Regression Model Estimation

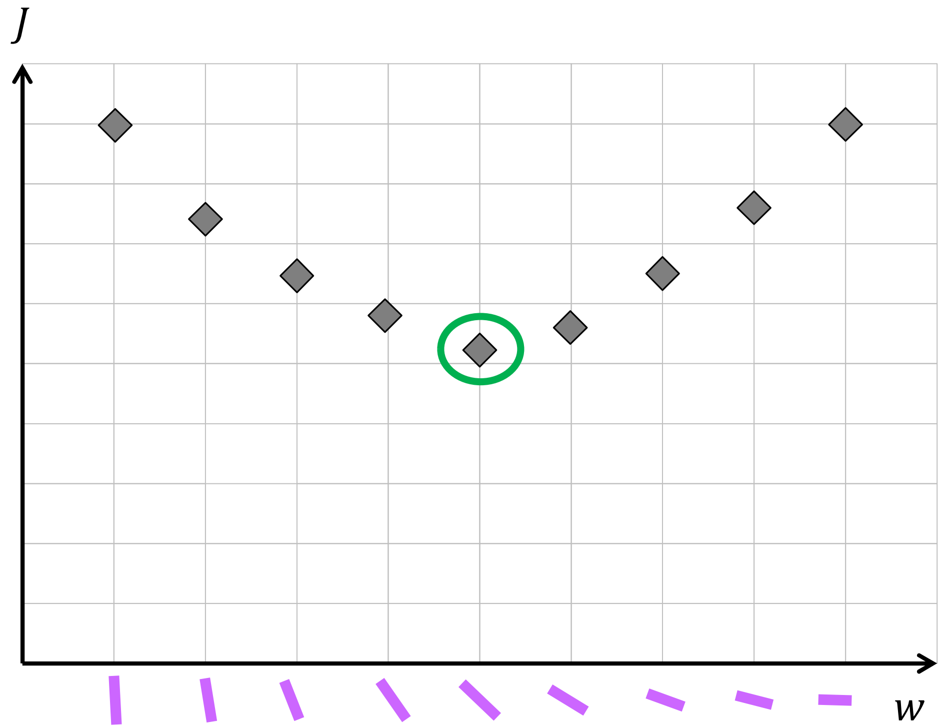
Maximizing model fit through minimizing a loss function

- **Given the model output depends on the free parameters w (and the bias b)**

$$\hat{y}_i = b + \sum_{j=1}^{m} w_j x_{ij}$$

1

- **We can adjust the model output by adjusting w (or b)**
  - Let's focus, for simplicity, on w
  - Changing w will change the slope of the regression line
- **For each slope we can calculate the loss (e.g., sum of squared residuals)**
- **Eventually, we know which slope (i.e., value of w) gave the lowest loss; our *least-squares solution***

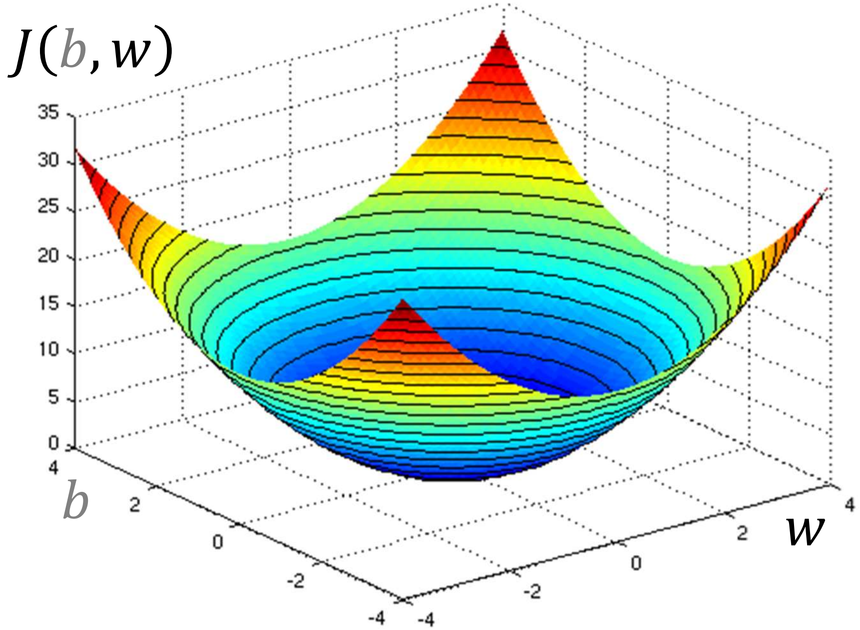


Regression Model Estimation
Least squares loss function

- **Given data set** $\mathcal{D} = \{y_i, \boldsymbol{x}_i\}_{i=1}^{n}$
- **Squared-error loss**

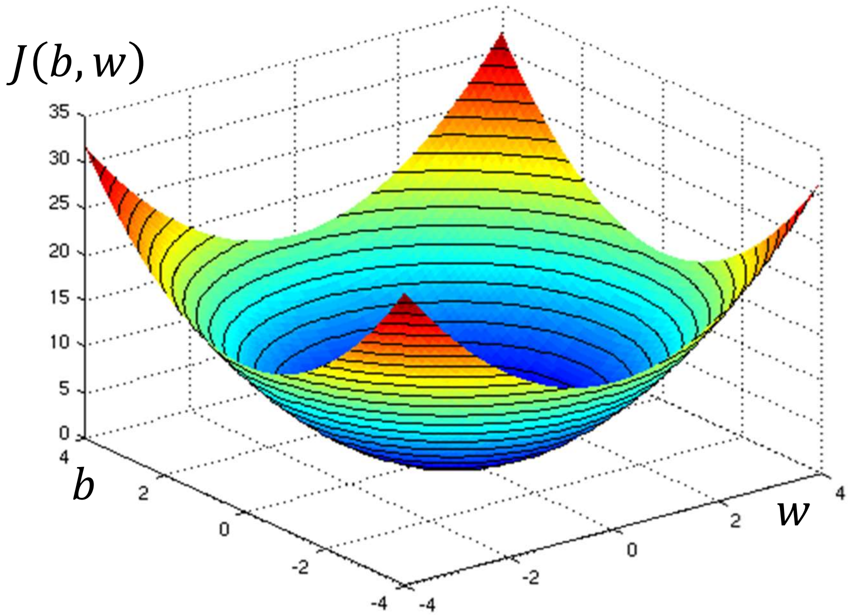
$$J(b, \boldsymbol{w}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \left(y_i - \left(b + \sum_{j=1}^{m} w_j x_{ij} \right) \right)^2$$

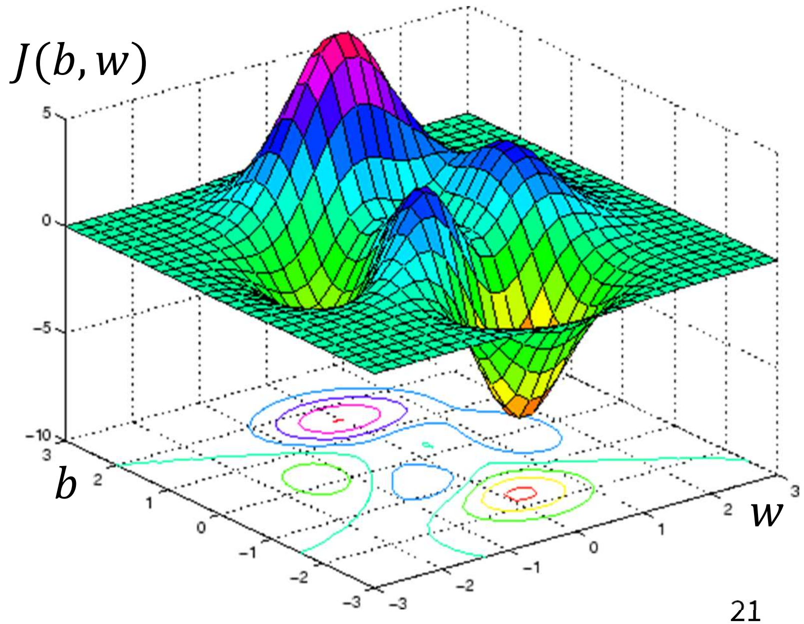
- **Visualization for a single feature** X
  - Simply think of X as a placeholder for some data you believe will influence your target variable
  - For ex. the feature could measure the price of a product when aiming to forecast sales (Y) based on prices



Regression Model Estimation
Finding the optimal solution

- **Formalization of model estimation (aka training)**
  - Given data set $\mathcal{D} = \{y_i, \boldsymbol{x}_i\}_{i=1}^{n}$
  - Select a suitable loss function $J(\boldsymbol{w}, \mathcal{D})$
  - Minimize that loss over the adjustable parameters
    $\widehat{\boldsymbol{w}} \leftarrow \mathrm{argmin}_{\boldsymbol{w}} \; J(\boldsymbol{w}, \mathcal{D})$
- **Special case**
  - Linear regression with squared error loss function
  - Optimal solution is easily found analytically
    - Calculate partial derivatives of $J(\boldsymbol{w})$ wrt $\boldsymbol{w}$ and set to zero
    - Gives famous normal equation $\widehat{\boldsymbol{w}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$
- **General case**
  - Same principle: minimize loss over parameters
  - Use some iterative algorithm for (loss) function minimization (e.g., gradient descent)





Linear Regression as Supervised Learning Algorithm

- **Model specification**
  - Continuous target variable
  - Linear, additive relationship
  - Random variation
- **Model estimation**
  - Determine free parameter $\boldsymbol{w}$
  - Find $\hat{\boldsymbol{w}}$ that maximizes model fit
  - Objective: minimize least-squares loss
- **Model**
  - Estimated coefficients
  - Facilitates forecasting

$$Y = b + \boldsymbol{w}\boldsymbol{X} + \epsilon$$

$$\hat{\boldsymbol{w}} \leftarrow \underset{\boldsymbol{w},b}{\operatorname{argmin}} \left(y_i - (b + \boldsymbol{w}\boldsymbol{x}_i) \right)^2, \quad i = 1, \ldots, n$$

$$\hat{Y} = \hat{b} + \hat{\boldsymbol{w}}\boldsymbol{X}$$

Training data incl. Y

i	Y	X_1	X_2	…	X_m
1	…	…	…	…	…
2	…	…	…	…	…
…	…	…	…	…	…
n	…	…	…	…	…

Learning Algorithm

Model $\hat{\boldsymbol{w}}$

New data w/o Y

i	X_1	X_2	…	X_m
$n+1$	…	…	…	…
$n+2$	…	…	…	…

Forecasts of Y

i	$\hat{Y}$
$n+1$	…
$n+2$	…
…	…
N	…

Two-Stage Paradigm in Supervised ML

- **Learning algorithm**
  - (Semi-)Parametric approaches mimic linear regression
  - Nonparametric approaches make no assumptions about DGP*
- **Model training**
  - Empirical risk minimization: maximize model fit on training data
  - Structural risk minimization: balance model fit vs. complexity
  - Minimize a loss function
- **Model**
  - Form varies across algorithms
  - Function with estimated parameters
  - Decision rules or tree-structure

Training data incl. Y

i	Y	X_1	X_2	…	X_m
1	…	…	…	…	…
2	…	…	…	…	…
…	…	…	…	…	…
n	…	…	…	…	…

Lear-ning Algo-rithm

New data w/o Y

i	X_1	X_2	…	X_m
$n+1$	…	…	…	…
$n+2$	…	…	…	…

Model

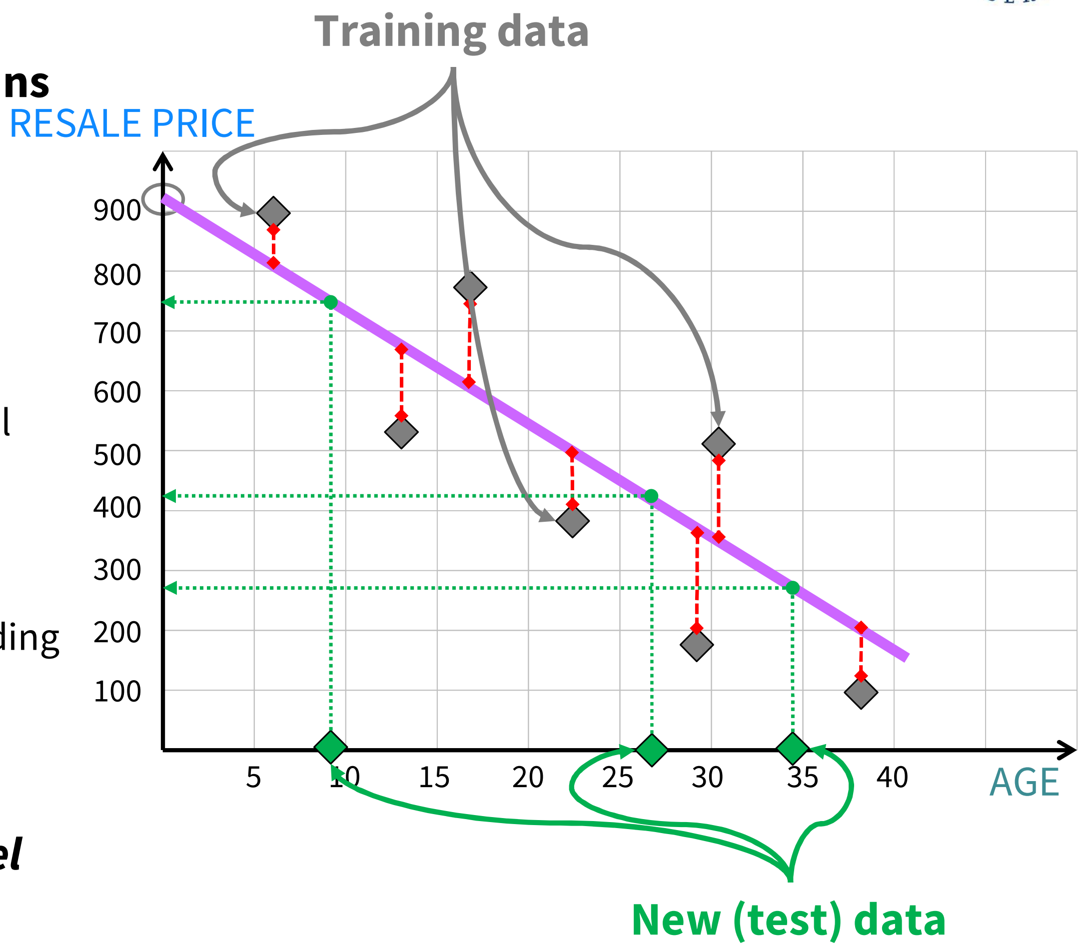
Forecasts of Y

i	$\hat{Y}$
$n+1$	…
$n+2$	…
…	…
N	…

*DGP: Data generating process

The Two Faces of Linear Regression

- **We discussed how the regression function explains variation in resale prices by age**
- **Because of this feature, linear regression is an *explanatory model***
  - Clarifies the relationship between features and the target
  - Can work out the strength of the effect of a feature
  - Can calculate elasticities, i.e., how a 1% change in age will change resale prices
- **However, linear regression also facilitates prediction**
  - Given a value of age, we can easily predict the corresponding resale price using the estimated coefficients
  - Just evaluate regression equation
  - $\text{Resale Price Forecast} = bias + w_1 \text{Age}$
- **Hence, linear regression is also a *predictive model***



Summary

Summary

Learning goals

- Predictive analytics (PA) principle & implications
- Fundamentals of linear regression

Findings

- PA requires past data with labels / target variable
- Regression predicts a numeric target
- Classification predicts a discrete target
- Two-step approach: model training and testing
- Linear regression supports both, explanatory and predictive modeling
  - Assume linear, additive relationship
  - Determine parameters by minimizing the sum of squared residuals

What next

- How to prepare data for analytics
- Preprocessing pipeline & techniques

Thank you for your attention!

Stefan Lessmann

Chair of Information Systems
School of Business and Economics
Humboldt-University of Berlin, Germany

Tel. +49.30.2093.5742
Fax. +49.30.2093.5741

stefan.lessmann@hu-berlin.de
http://bit.ly/hu-wi

www.hu-berlin.de