27.11.25

OPENCAMPUS.sh

Introduction to Data Science and Machine Learning

INTRODUCTION TO LINEAR REGRESSION

- Baseline Models
- Definition of linear regression
- Cost functions
- Optimization functions
- Project requirements

BASELINE MODELS

- Metrics alone often don't reveal how well a model performs.
- Always compare results to a baseline for meaningful evaluation.

Typical Baselines:

- Previous models on the same dataset
- Models on similar datasets
- Popular example for time series: Naïve Forecasting

BASELINE MODEL SELECTION

- **Should be easy to implement, with a reasonable chance of delivering decent results and a very low risk of overfitting.**

- **Should be interpretable to improve understanding of the data and enable better feature engineering.**

Ameisen, E. (2018, March 6). *Always start with a stupid model, no exceptions.* Medium. https://blog.insightdatascience.com/always-start-with-a-stupid-model-no-exceptions-3a22314b9aaa

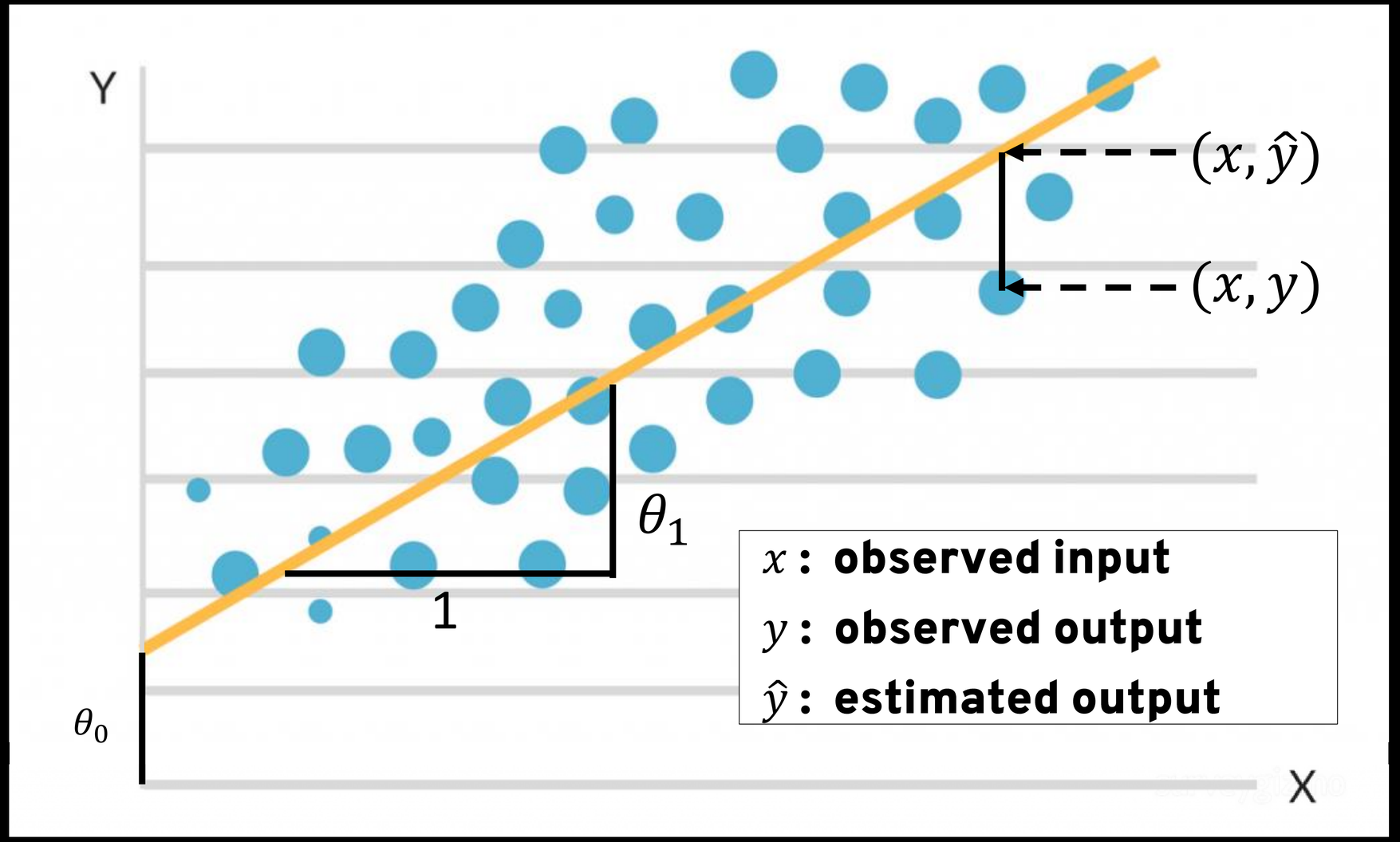
NAÏVE FORECASTING

- Simple baseline model for time series analysis
- Forecast = last observed value
- Seasonal Naïve Forecasting:
  Forecast = last observed value from the same seasonal period

LINEAR MODEL

$$\hat{y} = \theta_0 + \theta_1 x$$

$$= h_x\,(\theta_0, \theta_1)$$



COST FUNCTION (LOSS FUNCTION)

For calculating the function with the optimal parameters θ_0 und θ_1:

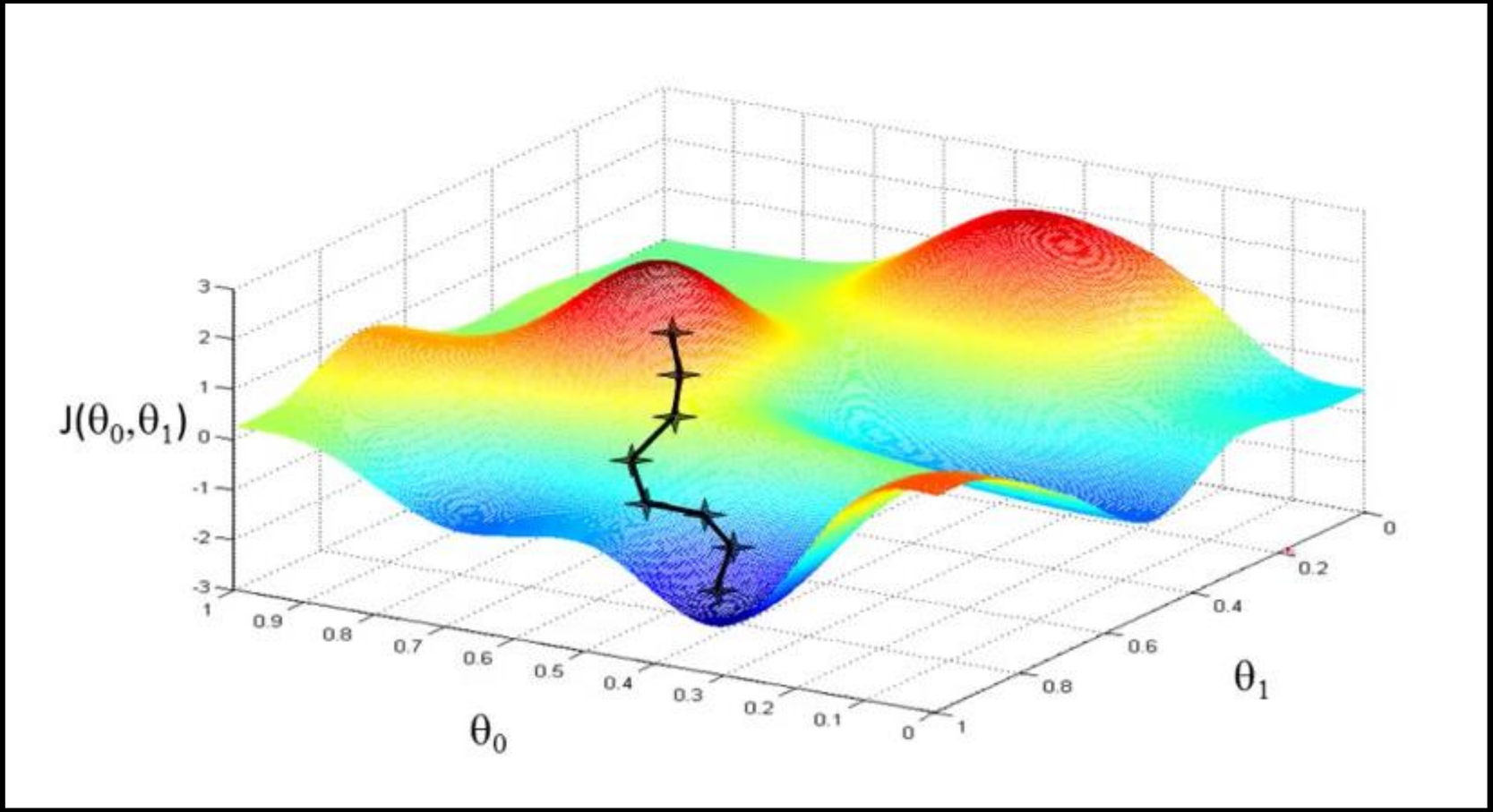
$$J_x(\theta_0, \theta_1) = \qquad h_x(\theta_0, \theta_1) - y$$

Mean Absolute Error (MAE)

$$J_x(\theta_0, \theta_1) = \frac{1}{m} \sum (h_x(\theta_0, \theta_1) - y)^2$$

Mean Squared Error (MSE)

OPTIMIZATION FUNCTION (OPTIMIZER)



- **Iterative method (Gradient Descent) to find the minimum of the cost function.**
- **The learning rate ("learning parameter") describes the step size for approaching the minimum.**

Quelle: https://www.coursera.org/learn/machine-learning

OPTIMIZATION OF THE MODEL PARAMETERS

Forward Propagation

- **Computation of predictions for a dataset using the current model parameters (weights and biases)**
- **Computation of the total cost or loss based on the difference between predictions and actual values**

Backward Propagation

- **Calculation of the contribution of each model parameter to the total loss**
- **Computation of gradients (i.e., partial derivatives of the loss with respect to parameter) to determine how to update the parameters to reduce the loss**

Gradient Descent (Optimization Step)

- **Use the derivates to update ethe parameters**
- **Typical update rule:**

```
New Value = Old Value – Learning Rate × Partial Derivative
```

MODEL PARAMETERS VS. HYPER PARAMETERS

Model Parameters

- **Parameters that are optimized during training (the weights and biases).**

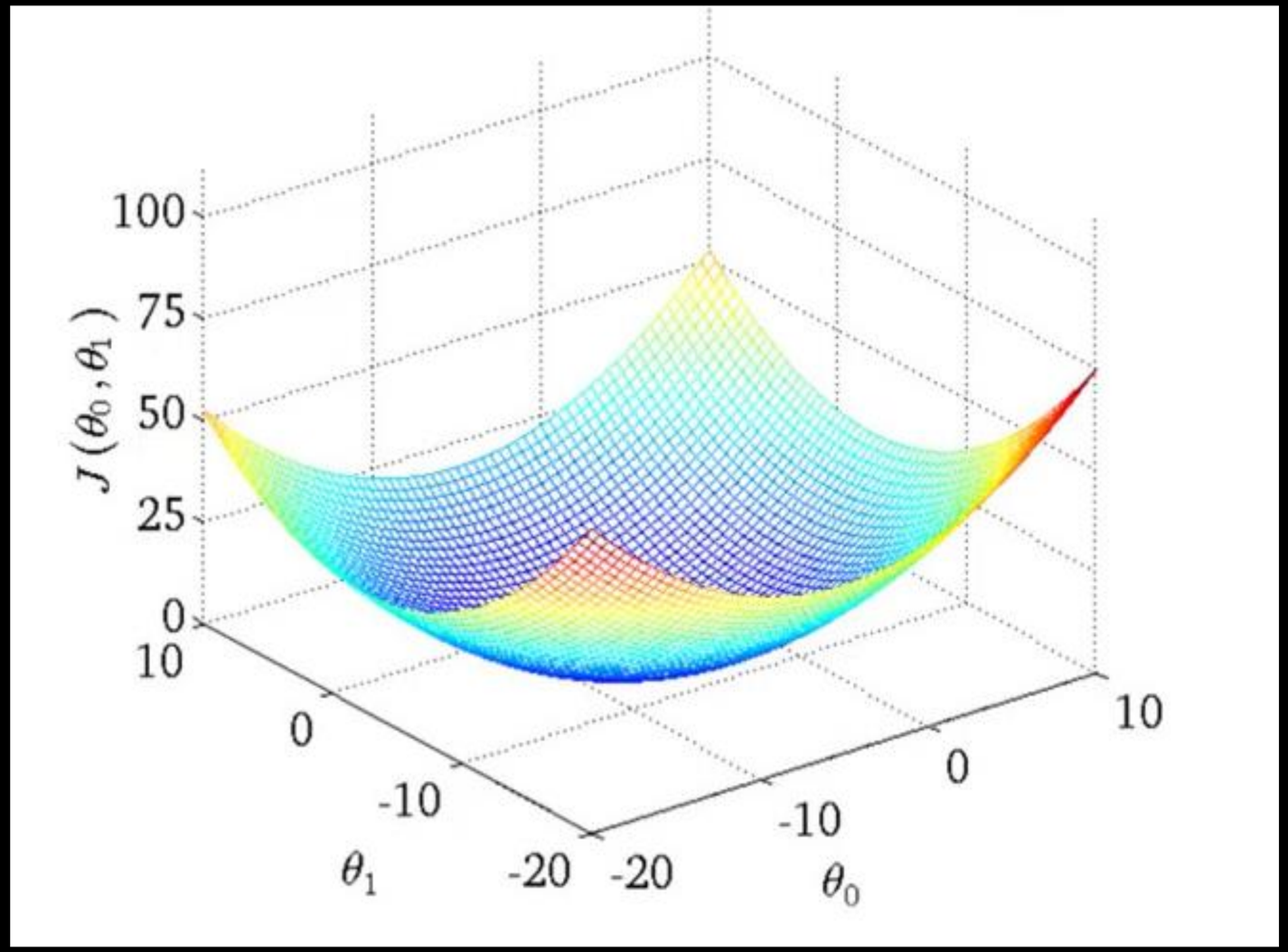
Hyper Parameters

- **Parameters set before training (e.g., on how the parameters are optimized, how many are optimized, or how they relate to each other).**

FEATURES, LABELS, PARAMETER

Machine Learning	Statistics	Description
Feature, Input Variable	Independent Variable	Input data used for prediction
Label, Target Variable, Output	Dependent Variable	Observed outcomes used to train the model
Weights, (Modell) Parameter	Coefficients	Are optimized ("learned") during training

MINIMIZATION IN LINEAR MODELLS



Quelle: https://www.coursera.org/learn/machine-learning

- For linear models, the cost function is convex and has no local minima.
- Here, other statistical methods besides Gradient Descent can also be used, especially when the model includes only a few variables.

EXAMPLE OF A LINEAR MODEL

```
import pandas as pd

import statsmodels.formula.api as smf


# Load the dataset

url = "https://raw.githubusercontent.com/opencampus-sh/einfuehrung-in-data-science-und-ml/main/house_pricing_data/house_pricing_train.csv"

house_pricing = pd.read_csv(url)


# Fit the linear model

mod = smf.ols('price ~ sqft_lot15 + C(condition)', data=house_pricing).fit()


# Print the summary

print(mod.summary())
```

RESULT OF A LINEAR MODEL

OLS Regression Results

Dep. Variable:	price	R-squared:	0.014
Model:	OLS	Adj. R-squared:	0.014
Method:	Least Squares	F-statistic:	49.39
Date:	Thu, 23 May 2024	Prob (F-statistic):	5.82e-51
Time:	08:50:41	Log-Likelihood:	-2.4603e+05
No. Observations:	17290	AIC:	4.921e+05
Df Residuals:	17284	BIC:	4.921e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.261e+05	7.05e+04	4.625	0.000	1.88e+05	4.64e+05
C(condition)[T.2]	-2.305e+04	7.69e+04	-0.300	0.764	-1.74e+05	1.28e+05
C(condition)[T.3]	2.031e+05	7.06e+04	2.878	0.004	6.48e+04	3.41e+05
C(condition)[T.4]	1.8e+05	7.07e+04	2.546	0.011	4.14e+04	3.19e+05
C(condition)[T.5]	2.762e+05	7.12e+04	3.880	0.000	1.37e+05	4.16e+05
sqft_lot15	1.1382	0.103	11.002	0.000	0.935	1.341

Omnibus:	15642.173	Durbin-Watson:	2.004
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1054915.657
Skew:	4.122	Prob(JB):	0.00
Kurtosis:	40.368	Cond. No.	1.69e+06

...

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 1.69e+06. This might indicate that there are strong multicollinearity or other numerical problems.

Explained variance of the model

Model parameters

Significance at p<.05

REGRESSION METRICS

p-Value

- Significance value
- Probability that the corresponding value in the regression is equal to zero.
- Values below 0.05 (equivalent to 5%) are usually considered significant.

(Adjusted) R^2

- Metric for assessing the quality of a regression
- Value between 0 and 1, representing the proportion of explained variation (1 corresponds to 100%):

$$R^2 = \frac{\text{Erklärte Varianz}}{\text{Gesamtvarianz}}$$

- The adjusted R^2 penalizes the addition of extra variables/parameters.

DataCamp tutorial on linear regression:
https://www.datacamp.com/tutorial/essentials-linear-regression-python

EXAMPLE PREDICTION

```
# Create a new house with the following features
new_house = pd.DataFrame({
    'sqft_lot15': [5000],  # Square footage of lot
    'condition': [3]       # Overall condition of the house
})

# Use the model to predict the price of the new house
predicted_price = mod.predict(new_house)

print(f"The predicted price for the new house is: {predicted_price[0]}")
```

BREAKOUT

- Split your dataset into a training set from 01.07.2013 to 31.07.2017, a validation set from 01.08.2017 to 31.07.2018, and a test set from 01.08.2018 to 30.07.2019.

- Define a very simple linear model equation and conduct a linear regression using your training data.

- Use your linear model to make a prediction for the test dataset and evaluate it on Kaggle

  (Ensure that the number of rows and the defined IDs match those in the sample_submission file!)

PROJECT PRESENTATION

Presentation (Powerpoint, Keybote or similar)

Prepare an 8 to 10-minute presentation including:

- Your team members' names on the title slide
- List and brief description of self-created variables
- Bar charts with confidence intervals for two self-created variables
- Linear model optimization: model equation and adjusted R^2
- Type of missing value imputation used
- Neural network optimization:
  - Source code defining the neural network
  - Loss function plots for training and validation sets
  - MAPE scores for the overall validation set and each product group
- Highlight "Worst Fail" and "Best Improvement" cases

Each team member should have a part in the presentation!

Important Notes

- **Presentation duration: approx. 8 minutes per team**
- **Every team member should have a part**
- **Use PowerPoint, Keynote, or similar**

Deadlines

- **By 4 PM on the day of the last course session:**
  Generate predictions for the Kaggle competition test dataset using your best model and upload them
- **By February 28, 2026:**
  - **Add your presentation to the repository**
  - **Set the repository to public**
  - **Complete the repository as described in the READMEs**
  - **One team member uploads the main README to the EduHub platform as instructed**

LEARNING RESOURCES

- Watch the videos in the section „The problem of overfitting“ from Week 3 of the course Supervised Machine Learning: Regression and Classification on DeepLearning.AI.

TASKS

- Split your dataset into a training set from 01.07.2013 to 31.07.2017, a validation set from 01.08.2017 to 31.07.2018, and a test set from 01.08.2018 to 31.07.2019.
- Use a simple linear model to make a prediction for the test dataset and evaluate it on Kaggle. (Ensure that the number of rows and the defined IDs match those in the sample submission file!)
- Further enrich the dataset with additional variables that may be relevant for estimating revenue and formulate a linear model equation that maximizes the adjusted R^2 for your training dataset.
- Document the linear regression calculations in the “Baseline Model” directory of your team repository.