# 1 General info

## 1.1 Sum Square Error Formula(SSE)

$E(\theta) = \sum_i^n (\hat{y}_i - y_i)^2$<br>
Since the formula for the predicted value is
$\hat{y} = \theta_1 * x_i + \theta_2$
the SSE formula can be summarized into
$E(\theta) = \sum_i^n (\theta_1 * x_i + \theta_2 - y_i)^2$

## 1.2 Canonical machine learning optimization problem

?? Formula not understandable.

## 1.3 vocabulary

1.Hypothesis function: The function used to predict values
2.Objective function: The function is used to identify the difference between the predicted values from the hypothesis funciton and the real values
3.Optimization problem: Is used to optimize the hypothesis parameters to reduce the objective function

### 1.3.1 Signs

1.Input features: $x^i$
2.Target features: $y^i$
3.Model parameters: $\theta$
4.Hypothesis function: $h_\theta = h_\theta(x) = \sum_{i=1}^n \theta_j * x_j$
5.Objective funciton: $\ell$

## 1.4 Regression with canonical formulation and matrix

1.Hypothesis function
$h_\theta(x) = \sum_{i=1}^n \theta_j * x_j$
$\theta$ j is the intercept for x = 0, for x¿0 it is the slope
2.Objective function
$\arg\min_\theta$: Means that $\theta$ (the parameter vector) should be optimized regarding min error $\theta = l(\hat{y}, y) = 1/2(\hat{y} - y)^2$
3.Optimization problem
$\theta$: Vector of parameters that need to be optimized (Slope)
n: number of features (Of each instance)
m: number of instances (Number of data points in the dataset)

$$\theta = \arg\min_\theta \sum_{i=1}^m \left( \sum_{j=1}^n \theta_j x_j^i - y^i \right)^2$$

### 1.4.1 Optimization problem formula explained

$\theta = \arg\min_{\theta}$: The to be optimized value is all parameter vector $\theta$, so the Slope, to minimize errors
$\sum_{i=1}^{m}$: The sum of errors for all data points in the dataset
$\theta_j x_j^i$: $\theta$ is the slope for a specific feature $x_j$ at position j at the i'th instance
$\sum_{j=1}^{n} \theta_j x_j^i$: Sums up the slope j multiplied with the feature j for the data point i, so $\hat{y}^i$ is calculated
$(\sum_{j=1}^{n} \theta_j x_j^i - y^i)^2$: Calculates the squared difference between the predicted value $\hat{y}^i = \sum_{j=1}^{n} \theta_j x_j^i$ and the actual value $y^i$ for the data point i

# 2 Minimizing Loss function

$l(\theta) = \sum_{i=1}^{n} (\theta_j x_j^i - y^i)^2$

# 3 Gradient descent

Negative Derivated error function and step by step insert x-values to find the near zero y-value (error value)
The stepsize indicates the difference between the current and the next entered x-value. If its to high, the lowest point might be skipepd.
The optimal $\theta$ is to be found by this formula:
$\theta_j := \theta_j - \propto \sum_{i=1}^{m} \left( \theta^{\top} x^{(i)} - y^{(i)} \right) x_j^{(i)} ==$
$\theta = (X^T X)^{-}1 X^T y$
X: Is the data matrix, each row is a data point and each column a feature
y: Is the target vector, containing the target values to predict based on X
$X^T$: Is the transposed data matrix, allowing matrix multiplication leading to combination of all features and sampels.
$X^T X$ (Gram matrix): Summarizes the relationships between features across all samples
$X^T y$: Captures relationship between features in X and outcomes in y.

# 4 Bias and Variance

Bias: Describes how the model performs on training data. Low Bias = Good fit for training data
Variance: Describes how the model performs on test data. High variance = Bad fit for new data (Overfitting)

# 5 Underfit and overfit

Underfit: Low variance and high bias
Overfit: High variance and low bias With increasing model complexity, the training loss decreases. The generalization/prediction starts to increase because of overfit

# 6 Absolute Regression Test Metrics

Signs:
e: Error
n: Amount of data points

## 6.1 Mean Square Error/Deviation

$\frac{1}{n}\sum_{i=1}^{n} e_i^2$
Is the average of the squares of the differences between the actual values and the predicted values. Use: Larger errors are penalized more heavily. Therefore it is good for an overview, but might be skewed by outliers. Lower errors are better

## 6.2 Root Mean Square Error

Term
$\sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}$
Is the root of the mean square error, but because of the root, the units stay the same

## 6.3 Mean Absolute Error/Deviation

Term
$\frac{1}{n}\sum_{i=1}^{n} |e_i|$
It is more robust to outliers and gives a straightforward interpretation of the average error magnitude.

## 6.4 Average Error

Term
$\frac{1}{n}\sum_{i=1}^{n} e_i$
Indicates whether the predictions are on average over- orunderpredicting the target response

# 7 Relative Regression Test Metrics

Are used, measure error or performance in ratios or percent. More usefull when comparing datasets with different units and scales.

## 7.1 r-Squared

Returns a value between r¡0 and r¡=1
1: The model has a perfect fit
0: The model has no fit
¡0: Worse then a horizontal line

# 8 Model training cycle

1.Divide data into training, validation, test set (e.g. 50, 20, 30)
2.Train model on training set
3.Use model on validation data to see performance and adjust hyperparameters (polynomial degree)
4.Retrain system on training and validation dataset
5.Evaluate performance on test set

## 8.1 Test data leakage

Test data leakage describes the usage of test data to adjust the model (a.e. adjusting hyperparameters). Therefore the test data set should be isolated.

### 8.1.1 Solving test data leakage

1: Recollect data if overfitting to test set is suspected
2: Dont look at test set

# 9 Regularization

The degree of polynomial can be seen as complexity of the model
Regularization is used to prevent overfitting by penalizing large coefficient values.
The best $\theta$ is to be found.

### 9.1 L2 Regularization/Ridge regression

This method first takes the cost function with $\lambda$ and then derives it after $\theta$
Afterwards, the derived function gets set to 0, therefore looking for the lowest error point.
This dervied function then gets rearranged to solve for $\theta$, so the $\theta$ creating the lowest error gets found.
Through ridge regression, the bias usually rises but the variance decreases.
The l2 regularization works like this:
1. Set a $\lambda$ value.The higher this value, the harder $\theta$ will be penalized, so $\theta$ will be lower with a higher $\lambda$.
2. Adjust $\theta$ to minimize the cost function for chosen $\lambda$. A too small $\theta$ will result in the error term increasing (underfitting), a too high one will increase the penalty
The goal is to find a $\theta$ that balances the error and also keeps the coefficients small. $\arg\min_\theta \frac{1}{m}\sum_{i=1}^{m} l(h_0(x^i), y^i) + \lambda \sum_{i=1}^{n} \theta^2$
Using a gradient helps finding the optimal $\theta$:
$\theta = (X^T X + \lambda I)^- 1 X^T y$
Side Note: The Identity matrix I is a matrix with ones on the diagonal and zeros elsewhere. It makes the term always invertable.

### 9.2 L1 Regularization/Lasso regression

The main difference is, that the total value of $\theta$ is used. Therefore, the penalty term can be zero, leading to the identification of features essentially not relevant to the model.
$\arg\min_\theta \frac{1}{m}\sum_{i=1}^{m} l(h_0(x^i), y^i) + \lambda \sum_{i=1}^{n} |\theta|$

## 10 Support Vector Machine

## 11 PROBLEMS

Lecture 3 Slide 10
Lection 3 Slide 15
Lection 4 slide 7
R Squared
Lection 4 slide 24